

03063



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

**"ESTUDIO DE LA CONSTRUCCIÓN DE SUPERFICIES
CON NURBS"**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN CIENCIAS
(COMPUTACIÓN)**

P R E S E N T A:

DANIEL ALEJANDRO CERVANTES CABRERA

DIRECTOR DE TESIS: DR. PABLO BARRERA SÁNCHEZ

México, D. F.

2005.

m. 344702

Dedicatoria

A mis padres Alberto y Elda.

A mis hermanos Elda, Alberto y Dalia.

Agradecimientos.

Quiero agradecer a mi asesor el Dr. Pablo Barrera Sánchez, por todo su apoyo a lo largo de mis estudios de maestría y en la realización de este trabajo, por tener siempre una actitud abierta para compartir sus conocimientos, por sus valiosos consejos y su amistad incondicional.

A todos mis sinodales Dr. Ernesto Bribiesca Correa, Mat. Carlos Velarde Velázquez y en especial a la M. en C. Ana Irene Ramírez Galarza y el Dr. Pedro González-Casanova Heríquez por sus valiosas correcciones y comentarios.

A los revisores Gloria Olimpia Rivas y Guilmer F. González, por su tiempo dedicado en este trabajo.

A mis padres y hermanos que han estado conmigo en los momentos más importantes de mi vida y en particular en esta etapa.

A Lucía por todo su amor, paciencia y palabras de aliento sobre todo en los momentos cuando creía que nunca iba a terminar este trabajo.

A la Mat. María de Carmen Ramos Nava por su apoyo y comprensión y al M. en C. Luis M. de la Cruz por su amistad incondicional.

A mis cuates de la Facultad de Ciencias, Claudia Alquicira, Erick, Selene, los dos Ivanés y Verónica, de la maestría, Eduardo, Adriana, Toño y Yuriria y a los de DGSCA Porfirio, Miguel, Nino y Victor, que con su amistad pasé muchos momentos gratos durante el tiempo en el que hice esta tesis.

Índice general

1. Funciones polinomiales por pedazos y sus representaciones.	1
1.1. Funciones polinomiales por pedazos.	1
1.2. Representaciones de funciones polinomiales por pedazos.	4
1.2.1. Función de potencia truncada.	4
1.2.2. Base de funciones de potencia truncada.	5
1.2.3. La Base de funciones de potencia truncada puede ser mala.	9
1.2.4. Diferencias divididas.	11
1.2.5. Función B-spline.	19
1.2.6. Base de funciones B-spline.	26
2. Curvas NURBS	33
2.1. Curva de Bézier integral.	33
2.1.1. Propiedades.	39
2.1.2. Subdivisión y diferenciabilidad.	43
2.2. Curvas de Bézier racionales.	50
2.2.1. Propiedades.	52
2.2.2. La curva de Bézier cuadrática racional y cónicas.	54
2.3. Curvas B-spline.	60
2.3.1. Propiedades.	63
2.4. Curvas NURBS.	69
2.4.1. Propiedades.	73
3. Superficies NURBS	75
3.1. Superficies de producto tensor.	75
3.2. Superficies de Bézier integrales.	77
3.2.1. Propiedades	79
3.3. Superficies de Bézier racionales.	82
3.3.1. Propiedades.	84
3.4. Superficies B-spline.	86
3.4.1. Propiedades.	87
3.5. Superficies NURBS.	91
3.5.1. Propiedades.	92
4. Técnicas básicas para la construcción de superficies.	101
4.1. Inserción de nodos.	101
4.1.1. Refinamiento.	109
4.2. Eliminación de nodos.	116
4.3. Elevación de grado.	120

4.4. Interpolación global de curvas B-spline.	126
4.5. Interpolación global con superficies B-spline.	130
5. Técnicas especiales para la construcción de superficies.	135
5.1. Superficies “Swung”	135
5.2. Superficies “Skinned”.	140
5.2.1. Superficies regladas.	140
5.2.2. Superficies “Skinned” generales.	144
5.2.3. Superficies “Skinned Spine”.	147
5.2.4. Superficies “Skinned” con curvas racionales.	153
5.3. Superficies de barrido.	155
Apéndice.	170
A. La biblioteca NURBS++.	171
A.1. Instalación.	172
A.2. Uso de la biblioteca NURBS++.	174
A.2.1. Curvas	174
A.2.2. Superficies	181
Bibliografía	187

Índice de figuras

1.1. Función Polinomial formada por 7 polinomios definidos cada uno en un intervalo.	1
1.2. Función polinomial por pedazos formada por $\{P_i\}_{i=0,\dots,4}$, en donde P_0 y P_1 se unen con continuidad de clase C^1 , P_1 y P_2 de $C^{(2)}$ y P_2 y P_3 de $C^{(0)}$	2
1.3. Funciones polinomiales por pedazos que pertenecen al espacio $P_{1,U,\nu}$ en donde $U = \{u_0, u_1, u_3, u_4\}$ y $\nu = (0, -1)$	3
1.4. La función de potencia truncada para $p = 0, 1, 2, 3$	4
1.5. Funciones que forman la familia $\{\phi_{r,s}\}$ para $p = 2$ y $k = 3$	5
1.6. Función que representa una constante rota.	9
1.7. La función spline en términos de la base de potencias truncadas pierde precisión al aproximar una constante rota.	10
1.8. (a) Función B-spline $N_{i,1}$ sobre U . (b) Función B-spline $N_{i,1}$ sobre V	20
1.9. Funciones B-spline $N_{0,1}$, $N_{1,1}$ y $N_{0,2}$ sobre $U = \{-1, 1, 3, 5\}$	24
1.10. Base de funciones B-spline de grado 2, con $U = \{-1, 2, 5, 6, 9\}$ y $\nu = (-1, 0, 1)$	29
1.11. Función B-spline $N_{0,2}$ definida sobre $U = \{-1, 1, 3, 5\}$ en la parte superior, sobre $U = \{-1, 1, 1, 5\}$ en medio y $U = \{-1, 1, 1, 1\}$ en la parte inferior.	31
2.1. Puntos obtenidos por el algoritmo de Casteljau con P_0, P_1, P_2, P_3 y $t = \frac{1}{2}$ (a) y $t = \frac{2}{3}$ (b).	34
2.2. Curva generada por el algoritmo de Casteljau con puntos de control P_0, P_1, P_2, P_3 y $t \in [0, 1]$	35
2.3. Polinomios Bernstein $B_{i,3}$ para $i=0,2,3$	39
2.4. Curva de Bézier integral subdividida por $\mathbf{B}_0(t)$ con $t \in [0, \frac{1}{2}]$ y $\mathbf{B}_1(t)$ con $t \in [\frac{1}{2}, 1]$	43
2.5. Curva de Bézier subdividida en donde se muestra las distancias proporcionales del polígonos de control y el rango parametral.	44
2.6. Condición de continuidad de clase $C^{(1)}$	48
2.7. Condición de continuidad de clase $C^{(2)}$	49
2.8. Condición de continuidad de clase $C^{(3)}$	49
2.9. Coordenadas homogéneas, el plano proyectivo y la curva B-spline racional.	52
2.10. Elipse y parábola obtenidas con el segmento complementario $\tilde{\mathbf{B}}(t)$, de la curva de Bézier racional.	56
2.11. Hipérbola obtenida con el segmento complementario $\tilde{\mathbf{B}}(t)$, de la curva de Bézier racional.	57
2.12. Cónicas para distintos valores del parámetro s . Si $s = \frac{1}{2}$ tenemos una elipse, si $s = \frac{1}{3}$ una parábola y si $s = \frac{2}{3}$ una hipérbola.	58
2.13. Arco circular representado por una curva de Bézier cuadrática racional.	59
2.14. Curva B-spline cúbica.	60
2.15. Curva B-spline cuadrática y sus funciones base.	62
2.16. (a) Curva B-spline. (b),(c),(d),(e) y (f) Envolturas convexas.	64
2.17. La modificación del punto P_3 cambia las subcurvas C_1 , C_2 y C_3 por C'_1 , C'_2 y C'_3	66

2.18. (a) Curva B-spline cuadrática. (b) Curva B-spline cuadrática en la cual la primer y segunda subcurva se unen con una continuidad de clase $C^{(0)}$. (c) Curva B-spline con modificaciones en los puntos de control, para lograr una continuidad de clase $C^{(1)}$ en la unión de la primer y segunda curva.	67
2.19.	68
2.20. Las curvas NURBS son una generalización de todas las curvas presentadas en este capítulo.	69
2.21. Circunferencia representadas con una curvas NURBS.	71
2.22. Circunferencias representadas con una curvas NURBS.	72
2.23. Curva NURBS de cúbica con 5 puntos de control.	72
3.1. Superficie de productor tensor.	77
3.2. Superficie de productor tensor con funciones base de potencia.	78
3.3. Superficie de Bézier integral.	79
3.4. Superficies base.	80
3.5. El algoritmo de Casteljau aplicado a una superficie de Bézier.	80
3.6. Parche de Bézier racional que representa un cilindro parabólico.	83
3.7. Parches racionales que resultan de la variación de sus pesos. (a) $w_{1,0} = 0.9$ $w_{1,1} = 0.6$ (b) $w_{1,0} = 5$ y $w_{1,1} = 0.2$ (c) $w_{1,0} = 10$ y $w_{1,1} = 1$ (d) $w_{1,0} = 30$ y $w_{1,1} = 0.01$	85
3.8. La Figura superior es la superficie B-spline y los puntos de control las isocurvas, en la inferior la superficie sin puntos de control.	88
3.9. La superficie sólo se altera en un rectángulo al modificarse un punto de control.	89
3.10. La repetición de puntos de control puede causar discontinuidades visuales en la superficie.	89
3.11. Las curvas NURBS son una generalización de todas las curvas presentadas en este capítulo.	91
3.12. Curva NURBS cuadrática en forma de trébol, con la que se forma el cilindro general.	94
3.13. Curva NURBS cúbica, con la que se forma un cilindro general en forma de domo.	95
3.14. Selección de Puntos de control para una superficie de revolución NURBS.	96
3.15. Esfera obtenida al girar un arco circular al rededor del eje de las z	97
3.16. Superficie libre obtenida al girar la curva NURBS cuadrática al rededor del eje de las z	98
3.17. Elipse que resulta de la intersección de la elipsoide y el plano (a) yx , (b) zx , (c) zy	99
3.18. Elipsoide, que resulta de aplicar el factor de escala $F = \{1, 1, 2\}$, a los puntos de control de la esfera del ejemplo anterior.	99
4.1. Inserción del nodo $\bar{u} = 3.5$, en una curva cúbica NURBS.	104
4.2. Las distancias que existen entre el nodo insertado y el nuevo punto de control son proporcionales.	104
4.3. El polígono de control está particionado por el vector de nodos.	105
4.4. El polígono de control de la Figura anterior, pero con otro vector de nodos.	106
4.5. Inserción del nodo $\bar{u} = 3.5$ por segunda ocasión.	107
4.6. Inserción del nodo $\bar{u} = 3.5$ por tercera ocasión.	107
4.7. Aristas del polígono de control que se ven afectadas por el refinamiento del vector de nodos.	109
4.8. Curva NURBS cuadrática original.	111
4.9. Primer paso del algoritmo de refinamiento.	111
4.10. Polígono y vector de nodos intermedios en el algoritmo de refinamiento.	112
4.11. Polígono y vector de nodos intermedios en el algoritmo de refinamiento.	113
4.12. Polígono de control y vector de nodos resultante.	113
4.13. Curva NURBS cuadrática con 5 puntos de control.	114
4.14. Curva NURBS con sus 3 segmentos de Bézier.	115
4.15. Funciones base cuadráticas de la curva NURBS refinada.	116
4.16. Curva NURBS formada por dos curvas que se unen con una continuidad de clase $C^{(2)}$	117
4.17. El vector de nodos $u = \frac{1}{2}$ puede ser eliminado dos veces sin que se pierda diferenciabilidad.	118

4.18. El vector de nodos $u = \frac{1}{2}$ puede ser eliminado 3 veces sin que se pierda diferenciabilidad.	118
4.19. Curva NURBS cúbica.	122
4.20. Se extrae el primer segmento de curva de Bézier.	122
4.21. Elevación de grado de la subcurva de Bézier.	123
4.22. Se extrae la siguiente curva de Bézier.	123
4.23. Se eleva el grado de la segunda curva de Bézier.	124
4.24. Se eliminan los nodos adicionales que se insertaron al extraer la primera curva de Bézier.	124
4.25. Se eleva el grado de la última curva de Bézier.	125
4.26. Se elimina los nodos adicionales que se insertaron para extraer el segundo segmento de Bézier.	125
4.27. Interpolación cúbica con valores parametrales y vector de nodos (a) uniforme, (b) longitud de arco y abscisas de Greville y (c) longitud centripetal y abscisas de Greville.	128
4.28. Curva de interpolación cúbica de ciertos puntos y sus derivadas.	129
4.29. Puntos a interpolar.	131
4.30. Isocurvas $C_{\bar{v}_l}(u)$, para $l = 0, \dots, m$.	132
4.31. Puntos de control resultantes de la interpolación de los puntos de control de las isocurvas $C_{\bar{v}_l}(u)$.	132
4.32. (a) Malla de puntos que se desea interpolar. (b) Superficie de interpolación con $p = 1$ y $q = 1$.	134
4.33. (a) Superficie de interpolación con $p = 1$ y $q = 2$. (b) Superficie de interpolación con $p = 3$ y $q = 2$.	134
5.1. Transformación de un punto de una curva perfil en la superficie "Swung".	136
5.2. Superficie "Swung": (a) Curva nurbs de trayectoria cuadrática. (b) Curva nurbs de perfil cúbica. (c) Superficie resultante.	137
5.3. Superficie "Swung": (a) Curva nurbs de trayectoria cuadrática. (b) Curva nurbs de perfil cúbica. (c) Superficie resultante.	138
5.4. Superficie "Swung": (a) Curva nurbs de trayectoria cuadrática. (b) Curva nurbs de perfil cúbica. (c) Superficie resultante.	138
5.5. Superficies "Swung" que resulta de variar el factor de escala.	139
5.6. Superficie libre "Swung": (a) Curva perfil. (b) Curva trayectoria. (c) Superficie resultante.	139
5.7. Superficie libre "Swung": (a) Curva perfil. (b) Curva trayectoria. (c) Superficie resultante.	139
5.8. (a) Las dos curvas con las que se construirá la superficie reglada. (b) Elevación de grado de la segunda curva.	142
5.9. (c) Refinamiento del vector de nodos. (d) Superficie reglada resultante.	142
5.10. Cono general. (a) Curva NURBS base del cono. (b) Cono resultante.	143
5.11. Isocurvas de la superficie "Skinned".	145
5.12. Superficie "Skinned", obtenida con las curvas anteriores haciendo una interpolación lineal.	145
5.13. Superficie "Skinned", obtenida con las curvas anteriores haciendo una interpolación cuadrática.	145
5.14. Superficie "Skinned" obtenida con las curvas anteriores haciendo una interpolación cúbica.	146
5.15. Cara formada con una superficie "Skinned".	146
5.16. Posicionamiento de las isocurvas sobre la curva "Spine". (a) $\hat{V} = \{0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1\}$ y $\hat{U} = \{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\}$ (b) $\hat{V} = \{0, \frac{1}{6}, \frac{2}{6}, \frac{1}{2}, 1\}$ y $\hat{U} = \{\frac{1}{2}, \frac{1}{6}, 0, 1, \frac{1}{2}\}$.	147
5.17. Las curvas deben ser orientadas de tal manera que su derivada en el punto $C_s(\hat{v}_k)$ se perpendicular al plano que la contiene.	148
5.18. Curva "Spine" y puntos en donde serán posicionadas las isocurvas.	149
5.19. Isocurvas, las cuales serán posicionadas sobre la curva "Spine".	150
5.20. Posicionamiento y orientación de las isocurvas.	150

5.21. Superficie “Skinned Spine” resultante.	151
5.22. Superficie “Skinned Spine” sin interpolación de derivadas.	151
5.23. Otro posicionamiento y orientación de las isocurvas.	152
5.24. Superficie “Skinned Spine”, con interpolación de derivadas.	152
5.25. Superficie “Skinned Spine”, sin interpolación de derivadas.	153
5.26. En la parte superior se presentan las curvas racionales con las que se construye la superficie reglada de la parte inferior.	154
5.27. Superficies “Skinned” mal parametrizada en el parámetro v . En superficie superior $\alpha = 20$ y en la superficie inferior $\alpha = 50$	156
5.28. (a) Curva NURBS que sigue la trayectoria de la curva (b). (c) Superficie de barrido resultante.	157
5.29. Proyección ortogonal para el cálculo del vector B_i	160
5.30. (a) Curva de barrido. (b) Curva Trayectoria.	164
5.31. Curvas posicionadas a través de la curva de trayectoria.	164
5.32. Superficie de barrido.	166
5.33. Superficie de barrido con función de escala s'_v	168
5.34. Superficie de barrido con función de escala s''_v	169
A.1. Página electrónica de la biblioteca NURBS++	171
A.2. Curva NURBS cúbica.	174
A.3. (a) Curva B-spline cúbica. (b) Curva de Bézier cúbica.	177
A.4. (a) Interpolación global. (b) Interpolación global de puntos y derivadas.	179
A.5. Superficie NURBS de grado 3×2 . (a) Superficie en formato ps. (b) Superficie con puntos de control de sus isocurvas. (c) Superficie en formato VRML.	182
A.6. Superficie de Bézier de grado 3×3 . (a) Superficie en formato ps. (b) Superficie con puntos de control de sus isocurvas. (c) Superficie en formato VRML	184
A.7. Superficie de interpolación global de grado 3×2 . (a) Superficie en formato ps. (b) Superficie con puntos de control de sus isocurvas. (c) Superficie en formato VRML	185

Introducción.

La modelación geométrica usando NURBS (“Non Uniform Rational B-spline”) es una área que ha crecido en los últimos 50 años, debido a sus excelentes propiedades y su incorporación en estándares internacionales como “IGES”¹ y “PHIGS+”², siendo ideales para una gran cantidad de aplicaciones a través de los diseños asistidos por computadora, es decir lo que se conoce como los CAD/CAM (“Computer Aided Desing”, “Computer Aided Manufacturing”), las cuales incluyen la animación en películas, realidad virtual, diseño de automóviles, aviones, barcos, etc..

Inicialmente los CAD/CAM no explotaban el potencial de representación de las superficies paramétricas estudiadas en la Geometría Diferencial, fue con los trabajos desarrollados por Casteljaou, Bézier, Coons y Forrest junto con el estudio teórico de las funciones B-spline de Schoenberg, De Boor, Cox, Mansfield, Reiesenfeld y Versprille, los que establecieron nuevas perspectivas para el desarrollo de esta área.

En el estudio de los NURBS, están involucrados la teoría de la aproximación, el análisis numérico y la geometría diferencial. Las principales características que los hacen tan populares son, que ofrecen representaciones precisas tanto de formas analíticas (secciones cónicas, superficies cuadráticas, etc.) como de formas libres (carrocerías, aviones etc.), los algoritmos usados son rápidos y numéricamente estables, son invariantes bajo una amplia gama de transformaciones geométricas y son generalizaciones de las curvas y superficies de Bézier y B-splines racionales o no.

Todo esto se debe a la gran cantidad de propiedades de las funciones B-spline con las que se representan los NURBS, por ejemplo, están definidas sobre un soporte pequeño, satisfacen una relación recursiva simple y tienen diferentes grados de diferenciabilidad. Además se han desarrollado procedimientos para su evaluación económica y reducción o elevación de grado.

Si bien es cierto que la mayoría de fundamentos básicos de los NURBS ya están establecidos, el área sigue siendo motivo de investigación y se requiere el desarrollo de nuevas técnicas para múltiples aplicaciones.

Este trabajo tiene el objetivo de presentar los fundamentos de las funciones base B-spline, las curvas y superficies NURBS. Así como la descripción de ciertos algoritmos y técnicas para la construcción tanto de superficies clásicas, como de superficies libres y finalmente su representación y graficación computacional.

Por lo que consideramos que su principal aportación, es la de proporcionar herramientas tanto teóricas como prácticas a las personas dedicadas a la construcción de superficies avanzadas, ya que en la actualidad es muy común que se utilicen sistemas o bibliotecas basados en NURBS, sin mucho entendimiento de su estructura matemática lo que generalmente limita las aplicaciones, pues en estos “softwares” no se permiten el control total de todos los elementos del NURBS o el usuario no posee mucho conocimiento para su manipulación más avanzada.

¹“Initial Graphics Exchange Specification, Version 3.0“, Doc. No. NBSIR 86-3359, NIST, Gaithersburg, Md., 1986

²“PHIGS+” Functional Description Revision 3.0 Computer Graphics, Vol. 22, Nl. 3, July 1988, pp.125-218

La tesis consta de cinco capítulos y un apéndice que se describen a continuación.

- En el primer capítulo, estudiamos el espacio vectorial formado por todas las funciones polinomiales por pedazos sobre un vector de nodos fijo y dos de sus representaciones. Así se presentan la base formada por funciones de potencia truncada y la base de funciones B-spline con sus respectivas propiedades. Mostramos por qué, en algunas ocasiones la representación de funciones de potencia truncada puede acarrear errores de precisión y cómo eliminarlos a través del uso la base de funciones B-spline.
- En el segundo, definimos las curvas NURBS. Iniciamos con las curvas de Bézier y B-spline ambas no racionales, presentamos sus respectivas propiedades y la relación de las curvas de Bézier racionales cuadráticas y las cónicas, lo que a su vez justifica la definición de las curvas de Bézier racionales y curvas NURBS, mostrando que estas últimas son una generalización de todas las demás.
- En el tercero, definimos las superficies NURBS. Para esto iniciamos con el estudio de las superficies de producto tensor, veremos como éstas, junto con el estudio previo de las curvas NURBS, nos permite definir de forma relativamente simple superficies NURBS las cuales heredan todas las características de las curvas. También comenzamos con la construcción de algunas superficies clásicas, como las de revolución, los cilindros generales y las obtenidas aplicando factores de escala no uniformes.
- En el cuarto, describimos una serie de técnicas básicas para la construcción de superficies NURBS más avanzadas, así estudiamos los algoritmos de refinamiento, inserción y eliminación de nodos, elevación y reducción de grado, y finalizamos con la interpolación global.
- En el quinto y último, hablamos de técnicas especiales para la construcción de superficies NURBS como las llamadas “Swung”, las cuales son generalizaciones de la superficies de revolución, “Skinned”, que se forman a partir de un conjunto de curvas o una curva llamada “Spine” y las de barrido, definidas a partir de una curva que sigue la trayectoria de otra.
- En el apéndice A, se presenta el uso de la biblioteca NURBS++ para C++. Aquí explicamos su instalación en un sistema operativo Linux, así como una serie de ejemplos en donde se muestra como representar y manipular tanto curvas como superficie NURBS.

El código fuente de todos los ejemplos que presentamos en este trabajo viene en el disco anexo, los cuales también se puede obtener en la página de “Internet”

<http://tycho.fciencias.unam.mx/~danielcc>

en donde además se encuentra la versión en formato *pdf* o *ps*, de esta tesis.

Capítulo 1

Funciones polinomiales por pedazos y sus representaciones.

En este capítulo estudiaremos el espacio vectorial formado por todas las funciones polinomiales por pedazos sobre un vector de nodos fijo y dos de sus representaciones. Así presentamos la base formada por funciones de potencia truncada y la base formada por funciones B-spline con sus respectivas propiedades. Mostraremos por qué en algunas ocasiones la representación de funciones de potencia truncada puede acarrear errores de precisión y cómo eliminarlos mediante el uso de la base de funciones B-spline.

1.1. Funciones polinomiales por pedazos.

Una **función polinomial por pedazos (fpp)** es una función formada por secciones de polinomios definidos cada uno en un intervalo. Por ejemplo, en la figura 1.1 observamos una fpp formada por 7 polinomios definidos sobre ciertos intervalos. Es importante notar que este tipo de funciones no necesariamente deben ser continuas, sin embargo para la definición formal siguiente, supondremos que en la unión de cada par de polinomios existe al menos continuidad por la derecha¹.

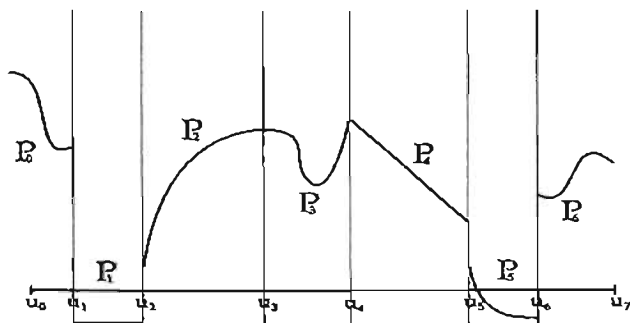


Figura 1.1: Función Polinomial formada por 7 polinomios definidos cada uno en un intervalo.

¹Esta suposición es necesaria, ya que a continuación definiremos el espacio vectorial formada por estas funciones, por lo que no deben existir puntos en donde no estén definidas.

1.1. FUNCIONES POLINOMIALES POR PEDAZOS.

Definición 1 Sean k y p dos enteros positivos, $U = \{u_0, \dots, u_k\}$ un conjunto (llamado vector de nodos) de $k+1$ reales estrictamente creciente (llamados nodos) y $\{P_i\}_{i=0}^{k-1}$ una sucesión de polinomios de grado $\leq p$ entonces una **función polinomial por pedazos** f de grado $\leq p$ (también se dice de orden $p+1$) en $\{u_0, u_k\}$ (o sobre U), es una función de la forma

$$f(u) = P_i(u) \text{ si } u \in [u_i, u_{i+1})$$

para $i = 0, \dots, k-1$.

Un resultado importante, se deduce al suponer que el vector de nodos U es fijo y considerar al conjunto (que denotaremos por $P_{p,U}$) formado por todas las fpp de grado menor o igual que p definidas sobre éste, ya que es posible probar [tesis, pág.10] que forma un espacio vectorial sobre los reales con dimensión $k(p+1)$. La demostración de esta última afirmación se puede deducir notando que cada polinomio que forma una fpp de $P_{p,U}$ es de a lo más grado p , por lo que hay a lo más $p+1$ variables sin especificar de un total de k polinomios.

Una característica deseable de las funciones polinomiales por pedazos, es que la unión de los polinomios que la forman sea suave, es decir, tengan ciertos grados de diferenciabilidad. Por ejemplo, tomemos un vector de nodos $U = \{u_0, u_1, u_2, u_3, u_4\} = \{-1, 1, 3, 5, 7\}$ y consideremos una fpp definida por

$$f(u) = \begin{cases} P_0(u) = \frac{1}{2} + u + \frac{1}{2}u^2 & \text{si } u \in [-1, 1) \\ P_1(u) = \frac{-149}{16} + 25u - \frac{149}{8}u^2 + \frac{11}{2}u^3 - \frac{9}{16}u^4 & \text{si } u \in [1, 3) \\ P_2(u) = \frac{101}{8} - \frac{97}{8}u + \frac{31}{8}u^2 - \frac{3}{8}u^3 & \text{si } u \in [3, 5) \\ P_3(u) = -\frac{1}{2} + \frac{1}{2}u & \text{si } u \in [5, 7) \end{cases}$$

en este caso los polinomios P_0, P_1, P_2 y P_3 son respectivamente de grado 2, 4, 3 y 1, por lo que la fpp $f(u)$, es de grado 4 (o de orden 5). En la Figura 1.2, se puede observar que la unión (representada con una bolita negra) de los tres primeros polinomios que la forman es suave, de hecho es fácil probar que la primera es de clase $C^{(1)}$ y la segunda de clase $C^{(2)}$; en el caso de P_2 y P_3 solamente existe continuidad de clase $C^{(0)}$.

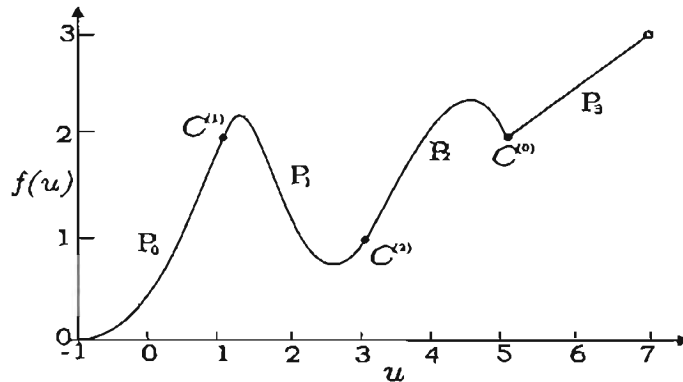


Figura 1.2: Función polinomial por pedazos formada por $\{P_i\}_{i=0,\dots,4}$, en donde P_0 y P_1 se unen con continuidad de clase C^1 , P_1 y P_2 de $C^{(2)}$ y P_2 y P_3 de $C^{(0)}$.

Una manera conveniente de incorporar la diferenciabilidad en una fpp es considerar un vector de $k-1$ enteros $\nu = (\nu_1, \dots, \nu_{k-1})$ en donde cada elemento cumple $-1 \leq \nu_i < p$ (para toda $i = 1$) los cuales representarán el grado de diferenciabilidad en cada u_i , es decir la fpp será de clase $C^{(\nu_i)}$ en u_i y en caso de que $\nu_i = -1$ entenderemos que $f(u)$ es discontinua en u_i .

El conjunto de todas las fpp de a lo más grado p sobre U fijo y grados de diferenciabilidad determinados por ν se denotará por $P_{p,U,\nu}$; éste como se esperaría, es un subespacio vectorial ([tesis, pág.12]) de $P_{p,U}$ con dimensión

$$\dim(P_{p,U,\nu}) = k(p+1) - \sum_{i=1}^{k-1} (\nu_i + 1) \quad (1.1)$$

El resultado se deduce del hecho de que cada condición de continuidad fija una variable (o grado de libertad) en cada polinomio que forma la fpp, disminuyendo en uno la dimensión de $P_{p,U}$ por cada una de estas condiciones.

Ejemplo 1 Supongamos que $U = \{u_0, u_1, u_2, u_3\}$ es un vector de nodos estrictamente creciente y que $p = 1$, así $P_{1,U}$ está formado por todas las fpp de la forma

$$f(u) = P_i(u) = \alpha_i + \beta_i u \quad \text{si } u \in [u_i, u_{i+1})$$

por lo que su dimensión es igual a $k(p+1) = (3)(2) = 6$, es decir las variables α_i y β_i para $i = 0, 1, 2$ son libres.

Ahora, construyamos el subespacio formado por todas las funciones de $P_{1,U}$ las cuales sean de clase $C^{(0)}$ en u_1 y discontinuas por la izquierda en u_2 , es decir $\nu = (0, -1)$ (ver Figura 1.3).

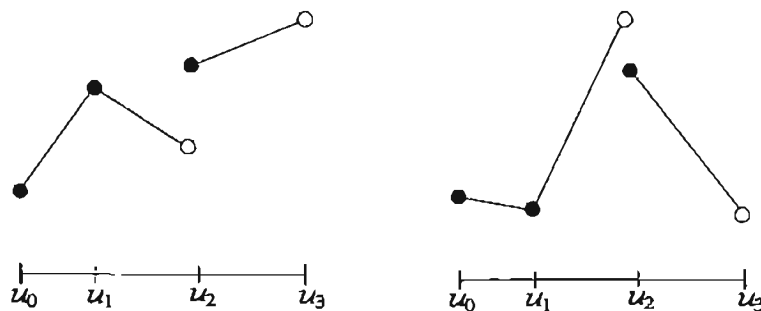


Figura 1.3: Funciones polinomiales por pedazos que pertenecen al espacio $P_{1,U,\nu}$ en donde $U = \{u_0, u_1, u_3, u_4\}$ y $\nu = (0, -1)$.

Así los elementos de este espacio deben cumplir que $P_0(u_1) = P_1(u_1)$ por lo que en caso de que $u_1 \neq 0$ debe suceder que

$$\beta_0 = \frac{\alpha_1 - \alpha_0 + \beta_1 u_1}{u_1}$$

quedando libres las variables $\alpha_0, \alpha_1, \alpha_2, \beta_1$ y β_2 . En el caso $u_1 = 0$ podemos tomar $\alpha_0 = \alpha_1$ en donde las variables libres ahora son $\alpha_1, \alpha_2, \beta_0, \beta_1$ y β_2 .

Es decir, al imponer la condición de continuidad en el nodo u_1 fijamos una de las 6 variables lo que implica que $\dim(P_{p,U,\nu}) = 6 - 1 = 5$. En términos de la igualdad (1.1) obtenemos el mismo resultado

$$\dim(P_{p,U,\nu}) = 6 - 1 - 0 = 5.$$

□

Un importante caso particular de $P_{p,U,\nu}$, es el llamado espacio **spline**, que denotaremos por $S_{p,U}$, el cual se obtiene cuando cada ν_i es igual a $p - 1$ (para toda i), es decir la unión de cualquier par de polinomios tiene suavidad "máxima", según la ecuación (1.1) su dimensión es $p + k$.

1.2. Representaciones de funciones polinomiales por pedazos.

El enfoque de espacio vectorial con el que estamos abordando las fpp, permite entre otras cosas plantear el problema de la búsqueda de bases adecuadas para la representación de sus elementos. Así lo que resta del capítulo lo dedicaremos al estudio de dos bases para $P_{p,k}$ (y sus subespacios), una formada por las llamadas funciones de potencia truncada y la otra por los B-spline.

1.2.1. Función de potencia truncada.

Una función de potencia truncada (fpt), es una función formada por el polinomio idénticamente cero y el polinomio $(u-t)^p$, los cuales están definidos sobre los intervalos $(-\infty, t]$ y (t, ∞) respectivamente. Es decir,

Definición 2 Dado $t \in \mathbb{R}$, una función de potencia truncada de grado $p \geq 0$, que denotaremos por $(u-t)_+^p$, es una función definida por

$$(u-t)_+^p = \begin{cases} (u-t)^p & \text{si } u > t \\ 0 & \text{si } u \leq t \end{cases} \quad (1.2)$$

Así es claro que $(u-t)_+^p \in P_{p,U}$ con $U = \{t\}$. En la Figura (1.4) se muestran las funciones de potencia truncada para $p = 0, 1, 2$ y 3 ; notemos que excepto para $p = 0$, las fpt son continuas en \mathbb{R} , y en particular en t la unión de los polinomios se vuelve más suave a medida que el grado aumenta. Esto en general se puede comprobar con el siguiente resultado.

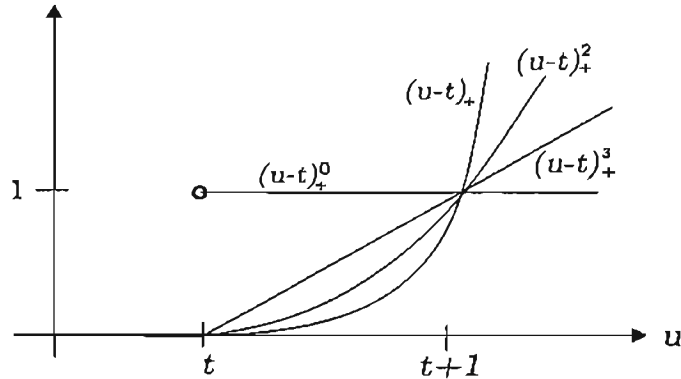


Figura 1.4: La función de potencia truncada para $p = 0, 1, 2, 3$.

Como

$$\frac{d}{du}(u-t)_+^p = \begin{cases} p(u-t)^{p-1} & \text{si } u > t \\ 0 & \text{si } u \leq t \end{cases} = p(u-t)_+^{p-1}$$

tenemos que

$$\frac{d^{p-1}}{du^{p-1}}(u-t)_+^p = \begin{cases} p!(u-t) & \text{si } u > t \\ 0 & \text{si } u \leq t \end{cases} = p!(u-t)_+ \quad (1.3)$$

y

$$\frac{d^p}{du^p}(u-t)_+^p = \begin{cases} p! & \text{si } u > t \\ 0 & \text{si } u \leq t \end{cases} = p!(u-t)_+^0 \quad (1.4)$$

Entonces de la ecuación (1.3) deducimos que $(u - t)_+^p$ tiene $p - 1$ derivadas continuas (es de clase $C^{(p-1)}$) en \mathbb{R} y en particular en t , donde toma el valor de cero, así adicionalmente la función de potencia truncada pertenece a $P_{p,U,\nu}$ y $S_{p,U}$ con $U = \{t\}$ y $\nu = (p - 1)$.

En el caso de la ecuación (1.4), se deduce que la fpt se vuelve discontinua en la p -ésima derivada con un brinco de valor en las ordenadas de $p!$.

1.2.2. Base de funciones de potencia truncada.

Una vez definida la función de potencia truncada estamos en la posibilidad de estudiar la base formada por estas funciones. Para lo cual definiremos una familia de funcionales lineales, que nos facilitará la construcción de representaciones de fpp en términos de esta base.

Definición 3 [DeBoor78, pág.102]

Dado un vector de nodos estrictamente creciente $U = \{u_0, \dots, u_k\}$ con $k > 0$, definimos al conjunto de funciones de potencia truncada $\{\phi_{r,s}(u)\}$ como

$$\phi_{r,s}(u) = \begin{cases} \frac{(u-u_0)_+^s}{s!} & \text{si } r = 0 \\ \frac{(u-u_r)_+^s}{s!} & \text{si } r = 1, \dots, k-1 \end{cases} \quad (1.5)$$

para $s = 0, \dots, p$ con $u \in [u_0, u_k)$.

Ejemplo 2 Supongamos que $p = 2$ y $k = 3$; entonces el conjunto $\{\phi_{r,s}\}$, está formado por las funciones que se muestran en la Figura 1.5

$$\begin{aligned} \phi_{0,0}(u) &= (u - u_0)_+^0 & \phi_{0,1}(u) &= (u - u_0) & \phi_{0,2}(u) &= \frac{(u - u_0)^2}{2!} \\ \phi_{1,0}(u) &= (u - u_1)_+^0 & \phi_{1,1}(u) &= (u - u_1)_+ & \phi_{1,2}(u) &= \frac{(u - u_1)^2}{2!} \\ \phi_{2,0}(u) &= (u - u_2)_+^0 & \phi_{2,1}(u) &= (u - u_2)_+ & \phi_{2,2}(u) &= \frac{(u - u_2)^2}{2!} \end{aligned}$$

□

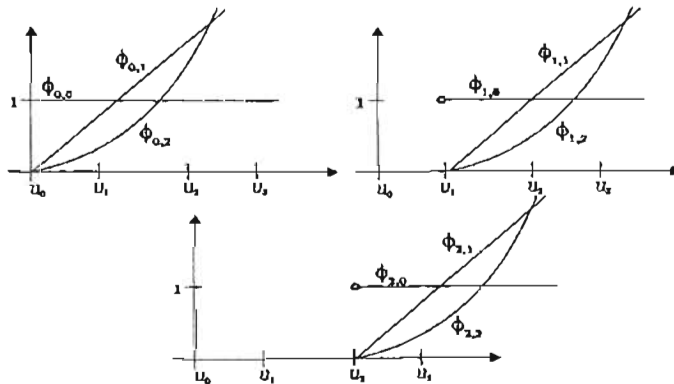


Figura 1.5: Funciones que forman la familia $\{\phi_{r,s}\}$ para $p = 2$ y $k = 3$.

Definición 4 [DeBoor78, pág.102]

Dado un vector de nodos estrictamente creciente $U = \{u_0, \dots, u_k\}$ definimos la familia de funcionales lineales $\{\lambda_{i,j}(f)\}$ del espacio dual de $P_{p,U}$ como

$$\lambda_{i,j}(f(u)) = \begin{cases} f^{(j)}(u_0) & \text{si } i = 0 \\ f^{(j)}(u_i^+) - f^{(j)}(u_i^-) & \text{si } i = 1, \dots, k-1 \end{cases} \quad (1.6)$$

para $j = 0, \dots, p$, con $f(u) \in P_{p,U}$ y $u \in [u_0, u_k]$ en donde

$$f(u_i^+) = \lim_{u \rightarrow u_i^+} f(u),$$

$$f(u_i^-) = \lim_{u \rightarrow u_i^-} f(u),$$

y $f^{(j)}(u_i)$ representa la j -ésima derivada de f evaluada en u_i .

El término lineal de las funcionales se refiere a que de las propiedades de la derivada se verifica fácilmente que cualquier $\lambda_{i,j}$ de (1.6) cumple

$$\lambda_{i,j}(\alpha f + g) = \alpha \lambda_{i,j}(f) + \lambda_{i,j}(g) \quad (1.7)$$

$$\lambda_{i,j}(\dot{f}) = 0 \quad (1.8)$$

en donde \dot{f} representa la función idénticamente cero de $P_{p,U}$, (el cero de este espacio) $f, g \in P_{p,U}$ y $\alpha \in \mathbb{R}$.

Una vez dadas las anteriores definiciones, a continuación probaremos que el conjunto $\{\phi_{r,s}\}$ forma una base de $P_{p,U}$.

Primeramente notemos que cada una de las funciones de $\{\phi_{r,s}\}$ pertenece a $P_{p,U}$ y que la cardinalidad de este conjunto coincide con la dimensión del espacio (es decir $k(p+1)$); por lo que solamente debemos probar que $\{\phi_{r,s}\}$ es linealmente independiente, para esto tenemos el siguiente lema.

Lema 1 $\{\phi_{r,s}\}$ y $\{\lambda_{i,j}\}$ cumplen

$$\lambda_{i,j}(\phi_{r,s}) = \delta_{i,r} \delta_{j,s} = \begin{cases} 1 & \text{si } i = r, j = s \\ 0 & \text{en otro caso.} \end{cases}$$

Demostración.

Por definición

$$\lambda_{i,j}(\phi_{r,s}(u)) = \begin{cases} \phi_{r,s}^{(j)}(u_0) & \text{si } i = 0 \\ \phi_{r,s}^{(j)}(u_i^+) - \phi_{r,s}^{(j)}(u_i^-) & \text{si } i = 1, \dots, k-1 \end{cases} \quad (1.9)$$

para $j = 0, \dots, p$.

Entonces el primer caso de (1.9) se puede escribir como

$$\phi_{r,s}^{(j)}(u_0) = \begin{cases} \left(\frac{(u-u_0)^s}{s!} \right)^{(j)} \Big|_{u=u_0} & \text{si } r = 0 \\ \left(\frac{(u-u_r)^s}{s!} \right)^{(j)} \Big|_{u=u_0} & \text{si } r = 1, \dots, k-1 \end{cases} \quad (1.10)$$

cuyo respectivo primer caso, claramente es igual a cero para toda $j < s$ e igual a 1 si $j = s$ (no olvidemos que $i = r = 0$). El segundo es igual a cero para toda j , ya que $u_0 < u_r$ para $r = 1, \dots, k-1$ (aquí $i \neq r$).

Ahora, el segundo caso de (1.9), es igual a cero para toda $j < s$ con $i \neq r$, debido a la propiedad de diferenciabilidad de las funciones de potencia truncada (1.3). Por lo que sólo cuando $j = s$ con $i = r$ tenemos

$$(u_r^+ - u_r)_+^0 - (u_r^- - u_r)_+^0 = 1 - 0 = 1.$$

■

Teorema 1 $\{\phi_{r,s}\}$ es un conjunto linealmente independiente.

Demostración. Supongamos que

$$\sum_{r=0}^{k-1} \sum_{s=0}^p \alpha_{r,s} \phi_{r,s}(u) = \hat{f}; \quad (1.11)$$

debemos probar que $\alpha_{r,s} = 0 \forall r, s$.

Entonces aplicando el funcional lineal $\lambda_{i,j}$ en ambos lados de (1.11) tenemos

$$\lambda_{i,j} \left(\sum_{r=0}^{k-1} \sum_{s=0}^p \alpha_{r,s} \phi_{r,s}(u) \right) = \lambda_{i,j}(\hat{f})$$

y por (1.7) y (1.8)

$$\sum_{r=0}^{k-1} \sum_{s=0}^p \alpha_{r,s} \lambda_{i,j}(\phi_{r,s}(u)) = 0,$$

finalmente por el lema anterior

$$\sum_{r=0}^{k-1} \sum_{s=0}^p \alpha_{r,s} \delta_{i,r} \delta_{j,s} = 0$$

de donde $\alpha_{i,j} = 0$ para $i = 0, \dots, k-1$ y $j = 0, \dots, p$.

■

Una vez demostrado que $\{\phi_{r,s}\}$ forma una base del espacio de funciones polinomiales por pedazos, el siguiente teorema nos proporciona una manera fácil de obtener representaciones de fpp en términos de la base $\{\phi_{r,s}\}$.

Teorema 2 Dado un vector de nodos $U = \{u_0, \dots, u_k\}$ y $p > 0$, cualquier función $f \in P_{p,U}$ se puede escribir de forma única como

$$f(u) = \sum_{i=0}^{k-1} \sum_{j=0}^p (\lambda_{i,j}(f(u))) \phi_{i,j}(u) \quad (1.12)$$

Demostración. Como $\{\phi_{r,s}\}$ forma una base de $P_{p,U}$, existen coeficientes únicos $\alpha_{r,s} \in \mathbb{R}$ (para $r = 0, \dots, k-1$ y $s = 0, \dots, p$) tales que

$$f(u) = \sum_{r=0}^{k-1} \sum_{s=0}^p \alpha_{r,s} \phi_{r,s}(u)$$

entonces sustituyendo el lado derecho de esta igualdad en el lado derecho de (1.12) tenemos

$$\begin{aligned}
 \sum_{i=0}^{k-1} \sum_{j=0}^p (\lambda_{i,j}(f(u))) \phi_{i,j}(u) &= \sum_{i=0}^{k-1} \sum_{j=0}^p \left(\lambda_{i,j} \left(\sum_{r=0}^{k-1} \sum_{s=0}^p \alpha_{r,s} \phi_{r,s}(u) \right) \right) \phi_{i,j}(u) \\
 &= \sum_{i=0}^{k-1} \sum_{j=0}^p \left(\sum_{r=0}^{k-1} \sum_{s=0}^p \alpha_{r,s} \lambda_{i,j}(\phi_{r,s}(u)) \right) \phi_{i,j}(u) \\
 &= \sum_{i=0}^{k-1} \sum_{j=0}^p \left(\sum_{r=0}^{k-1} \sum_{s=0}^p \alpha_{r,s} \delta_{i,r} \delta_{j,s} \right) \phi_{i,j}(u) \\
 &= \sum_{i=0}^{k-1} \sum_{j=0}^p \alpha_{i,j} \phi_{i,j}(u) = f(u)
 \end{aligned}$$

lo cual es cierto por las propiedades del funcional (1.7) y el lema 1. ■

Finalmente notemos que sustituyendo las definiciones de $\phi_{i,j}$ y $\lambda_{i,j}$ en (1.12), obtenemos la siguiente expresión para f

$$f(u) = \sum_{j=0}^p \frac{f^{(j)}(u_0)(u-u_0)^j}{j!} + \sum_{i=1}^{k-1} \sum_{j=0}^p \frac{(f^{(j)}(u_i^+) - f^{(j)}(u_i^-))(u-u_i)_+^j}{j!} \quad (1.13)$$

la cual tiene la ventaja de que la derivada en la unión de cada par de funciones que forman a la fpp es explícita. Así para el caso cuando $f \in P_{p,U,\nu}$, tenemos que los coeficientes

$$f^{(j)}(u_i^+) - f^{(j)}(u_i^-) = 0$$

para $j = 0, \dots, \nu_i$ con $i = 1, \dots, k-1$, (ya f es de clase $C^{(\nu_i)}$ en u_i ver sección 1.1), por lo que se puede escribir como

$$f(u) = \sum_{j=0}^p \frac{f^{(j)}(u_0)(u-u_0)^j}{j!} + \sum_{i=1}^{k-1} \sum_{j=\nu_i+1}^p \frac{(f^{(j)}(u_i^+) - f^{(j)}(u_i^-))(u-u_i)_+^j}{j!} \quad (1.14)$$

y en particular cualquier $s \in S_{p,U}$

$$\begin{aligned}
 s(u) &= \sum_{j=0}^p \frac{s^{(j)}(u_0)(u-u_0)^j}{j!} + \sum_{i=1}^{k-1} \frac{(s^{(p)}(u_i^+) - s^{(p)}(u_i^-))(u-u_i)_+^p}{p!} \\
 &= \sum_{j=0}^p \frac{s_0^{(j)}(u_0)(u-u_0)^j}{j} + \sum_{i=1}^{k-1} \frac{(s_i^{(p)}(u_i) - s_{i-1}^{(p)}(u_i))(u-u_i)_+^p}{p!}
 \end{aligned} \quad (1.15)$$

en donde $s_i(u)$ representa el i -ésimo polinomio que forma al spline s .

Ejemplo 3 Escribamos la fpp de la Figura 1.2 definida por

$$f(u) = \begin{cases} P_0(u) = \frac{1}{2} + u + \frac{1}{2}u^2 & \text{si } u \in [-1, 1) \\ P_1(u) = \frac{-149}{16} + 25u - \frac{149}{8}u^2 + \frac{11}{2}u^3 - \frac{9}{16}u^4 & \text{si } u \in [1, 3) \\ P_2(u) = \frac{101}{8} - \frac{97}{8}u + \frac{31}{8}u^2 - \frac{3}{8}u^3 & \text{si } u \in [3, 5) \\ P_3(u) = -\frac{1}{2} + \frac{1}{2}u & \text{si } u \in [5, 7) \end{cases}$$

utilizando (1.14). Entonces ya que en este caso $p = 4, k = 4$ y $\nu = (1, 2, 0)$; tenemos

$$\begin{aligned} f(u) &= \sum_{j=0}^4 \frac{f^{(j)}(u_0)(u - u_0)^j}{j!} + \sum_{i=1}^3 \sum_{\nu, +1}^4 \frac{(f^{(\nu)}(u_i^+) - f^{(\nu)}(u_i^-))(u - u_i)_+^j}{j!} \\ &= \frac{(u+1)^2}{2!} + \sum_{j=2}^4 \frac{(P_1^{(j)} - P_0^{(j)})(u-1)_+^j}{j!} \\ &\quad + \sum_{j=3}^4 \frac{(P_2^{(j)}(3) - P_1^{(j)}(3))(u-3)_+^j}{j!} + \sum_{j=1}^4 \frac{(P_3^{(j)}(5) - P_2^{(j)}(5))(u-5)_+^j}{j!} \\ &= \frac{(u+1)^2}{2} - 6(u-1)_+^2 + \frac{39}{12}(u-1)_+^3 - \frac{27}{48}(u-1)_+^4 + \frac{21}{24}(u-3)_+^3 + \\ &\quad + \frac{27}{48}(u-3)_+^4 + 2(u-5)_+ + \frac{7}{4}(u-5)_+^2 + \frac{9}{24}(u-5)_+^3. \end{aligned}$$

□

1.2.3. La Base de funciones de potencia truncada puede ser mala.

Existen casos en donde la representación numérica de una fpp en términos de la base $\{\phi_{i,j}\}$ puede acarrear errores de precisión, a continuación estudiaremos este problema por medio de un ejemplo.

Ejemplo 4 Construyamos una función spline s en términos de la base $\{\phi_{i,j}\}$ como una “constante rota”, es decir deseamos que s aproxime a una función como la que se muestra en la Figura 1.6.

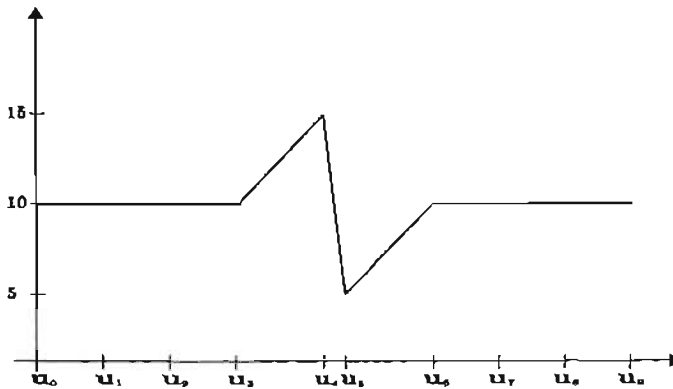


Figura 1.6: Función que representa una constante rota.

Entonces podemos tomar un spline $s \in S_{p,U}$, con $p = 1$ y $U = \{0, 1, 2, 3, u_4, u_5, 6, 7, 8, 9\}$ (u_4 y u_5 se determinarán más adelante) en donde cada $s_i(u) = 10$ para $i = 0, 1, 2, 6, 7, 8, 9$ y

$$s_j(u) = (u - u_j)m_j + s_j(u_j)$$

con

$$m_j = \frac{s_j(u_{j+1}) - s_j(u_j)}{u_{j+1} - u_j} = \frac{s_j(u_{j+1}) - s_j(u_j)}{\Delta_j}$$

1.2. REPRESENTACIONES DE FUNCIONES POLINOMIALES POR PEDAZOS.

para $j = 3, 4, 5$. Lo cual implica que

$$s'_k(u) = \begin{cases} 0 & \text{si } k = 0, 1, 2, 6, 7, 8, 9 \\ m_k & \text{si } k = 3, 4, 5. \end{cases}$$

y por último notemos que s debe cumplir

$$s_3(u_4) = 15 = s_4(u_4)$$

$$s_4(u_5) = 5 = s_5(u_5)$$

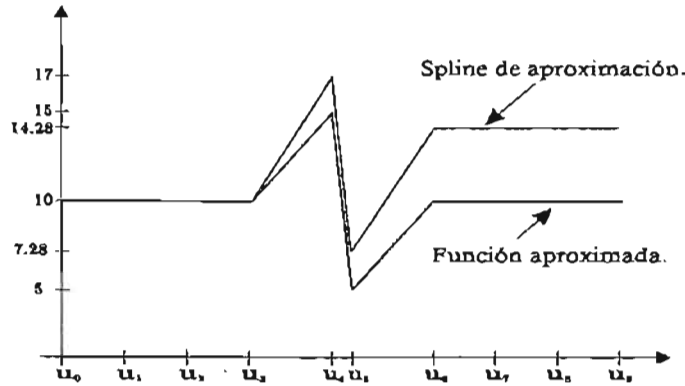


Figura 1.7: La función spline en términos de la base de potencias truncadas pierde precisión al aproximar una constante rota.

De (1.15) la función deseada en términos de la base de potencias truncadas es

$$\begin{aligned} s(u) &= \sum_{j=0}^1 \frac{s_0^{(j)}(u_0)(u-u_0)^j}{j!} + \sum_{i=1}^8 (s'_i(u_i) - s'_{i-1}(u_i))(u-u_i)_+ \\ &= s(u_0) + \sum_{i=3}^6 (s'_i(u_i) - s'_{i-1}(u_i))(u-u_i)_+ \\ &= s_0(u_0) + (m_3 - 0)(u-u_3)_+ + (m_4 - m_3)(u-u_4)_+ \\ &\quad + (m_5 - m_4)(u-u_5)_+ + (0 - m_5)(u-u_6)_+ \\ &= 10 + 5(u-3)_+ + \left(-\frac{10}{\Delta_4} - \frac{5}{\Delta_3}\right)(u-u_4)_+ \\ &\quad + \left(\frac{5}{\Delta_5} + \frac{10}{\Delta_4}\right)(u-u_5)_+ + 5(u-6)_+ \end{aligned}$$

Entonces una vez que tenemos la representación de la constante rota, notemos que cuando $\Delta_4 \approx 0$, es decir u_4 se aproxima a u_5 (o viceversa) implica que $(u-u_4) \approx (u-u_5)$ y

$$-\left(-\frac{10}{\Delta_4} - \frac{5}{\Delta_3}\right) \approx \left(\frac{5}{\Delta_5} + \frac{10}{\Delta_4}\right) \gg 1$$

esto último es cierto ya que los cocientes $-\frac{10}{\Delta_4}$ y $\frac{10}{\Delta_4}$ tienden a infinito y menos infinito respectivamente, así $\frac{5}{\Delta_3}$ y $\frac{5}{\Delta_5}$ no son significativos. En términos geométricos $\Delta_4 \approx 0$ se puede interpretar diciendo que la

pendiente de s_4 (es decir m_4) se vuelve cada vez más vertical lo que “deforma” al spline causando una pérdida de precisión al evaluar ciertos puntos en s (ver Figura 1.7).

Por ejemplo si $u_4 = 4.4$ y $u_5 = 4.6$ tenemos que $\Delta_4 = 0.2$ y usando aritmética de dos dígitos decimales y redondeando al par más cercano el spline es

$$s(u) = 10 + 5(u - 3)_+ + -53.58(u - 4.4)_+ + 53.58(u - 4.6)_+ - 5(u - 6)_+$$

para $0 \leq u \leq 9$.

Ahora si tomamos $u = 8.5$, tenemos que $s(8.5) = 14.28$ sin embargo $s(8.5)$ debería tener un valor de 10.

La razón de esto, se debe a que la evaluación de s en $u = 8.5$ involucra todas las funciones de potencia truncada base que la forman, y los coeficientes $(m_4 - m_3) = (m_5 - m_4) = 53.58$ son bastante grandes en comparación a los demás, alterando el resultado final. En la siguiente sección volveremos a este ejemplo y mostraremos cómo usando funciones B-spline se elimina este error.

□

1.2.4. Diferencias divididas.

A partir de ahora nos enfocaremos en el estudio de la base formada por las funciones B-spline, para esto será necesario antes una breve introducción a las diferencias divididas, ya que una manera de definir a los B-spline, es utilizar la diferencia dividida de una función de potencia truncada.

Entonces supongamos dada la siguiente tabla de valores

$$\begin{array}{c|c|c|c} u & u_0 & \cdots & u_k \\ \hline f(u) & f(u_0) & \cdots & f(u_k) \end{array} \quad (1.16)$$

para alguna función $f : \mathbb{R} \rightarrow \mathbb{R}$ (no necesariamente conocida para todos los \mathbb{R}) y $u_i \in \mathbb{R}$ para toda i .

El problema de encontrar un polinomio $p_k(u)$ de a lo más grado k que interpole a f en los puntos u_i es decir

$$p_k(u_i) = f(u_i) \text{ para } i = 0, \dots, k \quad (1.17)$$

es bastante conocido y estudiado, de hecho uno de los resultados más importantes es el de la existencia y unicidad de este polinomio. Para probarlo supongamos que p_k es de la forma

$$p_k(u) = a_0 + a_1u + \cdots + a_ku^k$$

y entonces para que este polinomio cumpla (1.17), debe suceder

$$\begin{aligned} p_k(u_0) &= a_0 + a_1u_0 + \cdots + a_ku_0^k = f(u_0) \\ p_k(u_1) &= a_0 + a_1u_1 + \cdots + a_ku_1^k = f(u_1) \\ &\vdots \\ p_k(u_k) &= a_0 + a_1u_k + \cdots + a_ku_k^k = f(u_k) \end{aligned}$$

lo que implica un sistema de $k + 1$ ecuaciones con $k + 1$ incógnitas cuya existencia y unicidad de solución depende de que el determinante

$$\begin{vmatrix} 1 & u_0 & \cdots & u_0^k \\ 1 & u_1 & \cdots & u_1^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & u_k & \cdots & u_k^k \end{vmatrix} = \Delta_k$$

(llamado de Vandermonde) sea distinto de cero. El cual además se puede escribir como [Lang][pág. 259]

$$\begin{aligned}\Delta_k &= (u_k - u_{k-1})(u_k - u_{k-2}) \cdots \\ &\quad (u_k - u_0)(u_{k-1} - u_{k-2})(u_{k-1} - u_{k-3}) \cdots (u_1 - u_0) \\ &= \prod_{\substack{i=1 \\ j < i}}^k (u_i - u_j)\end{aligned}$$

de donde se deduce que se puede garantizar la existencia y unicidad del polinomio de interpolación siempre y cuando los valores u_i sean distintos.

Una de las formas clásicas de $p_k(u)$ es el Polinomio de Lagrange, dado por la siguiente expresión

$$p_n(u) = \sum_{i=0}^k \frac{f(u_i)}{\prod_{\substack{j=0 \\ j \neq i}}^k (u_i - u_j)} \prod_{i=0}^k (u - u_j) = \sum_{i=0}^k f(u_i) \ell_i(u) \quad (1.18)$$

en donde el conjunto de funciones base $\{\ell_i\}_{i=0, \dots, k}$ (llamadas cardinales), están dadas por

$$\ell_i(u) = \frac{(u - u_0) \cdots (u - u_{i-1})(u - u_{i+1}) \cdots (u - u_k)}{(u_1 - u_0) \cdots (u_i - u_{i-1})(u_i - u_{i+1}) \cdots (u_i - u_k)}$$

Ya que así

$$p_k(u_j) = \sum_{i=0}^k f(u_i) \ell_i(u_j) = \sum_{i=0}^k f(u_i) \delta_{i,j} = f(u_j)$$

para $j = 0, \dots, k$.

Este esquema nos es recomendable en términos prácticos, pues si se agrega otro punto $(u_{k+1}, f(u_{k+1}))$ a la tabla (1.16), se deberá volver a calcular cada función ℓ_i , lo que es bastante ineficiente.

Un método más adecuado para la representación de $p_k(u)$ en esos términos, es el llamado Polinomio de Newton cuyo planteamiento es el siguiente.

Supongamos que $p_{k-1}(u)$ y $p_k(u)$ son dos polinomios de grado a lo más $k-1$ y k que interpolan a $f(u)$ en u_0, \dots, u_{k-1} y u_0, \dots, u_k respectivamente; así $q_k(u) = p_k(u) - p_{k-1}(u)$ es un polinomio de a lo más grado k que se anula en u_0, \dots, u_{k-1} lo cual implica que se puede escribir como

$$q_k(u) = A_k(u - u_0) \cdots (u - u_{k-1}) = A_k \prod_{i=0}^{k-1} (u - u_i) \quad (1.19)$$

con $A_k \in \mathbb{R}$.

Ahora, notando que $q(u_k) = f(u_k) - p_{k-1}(u_k)$, podemos utilizar (1.19) para obtener A_k realizando el siguiente despeje

$$A_k = \frac{f(u_k) - p_{k-1}(u_k)}{\prod_{i=0}^{k-1} (u_k - u_i)} \quad \text{para } k \geq 1 \quad (1.20)$$

Por lo tanto, si tomamos $p_0(u) = f(u_0)$ (es decir si hacemos que el polinomio de grado cero $p_0(u)$ sea igual a $f(u_0)$) y $A_0 = f(u_0)$ podemos usar (1.20) para calcular A_1 de la siguiente manera

$$A_1 = \frac{f(u_1) - p_0(u_1)}{u_1 - u_0} = \frac{f(u_1) - f(u_0)}{u_1 - u_0}$$

y así $p_1(u)$ se puede escribir como

$$p_1(u) = A_0 + A_1(u - u_0)$$

logrando que $p_1(u_0) = f(u_0)$ y $p_1(u_1) = f(u_1)$.

De forma análoga podemos calcular A_2 y obtener el polinomio $p_2(u)$ que interpola a f en u_0, u_1 y u_2 , y así sucesivamente. Finalmente, el polinomio $p_k(u)$ que interpola a f en los puntos u_0, \dots, u_k tiene la expresión

$$p_k(u) = A_0 + A_1(u - u_0) + A_2(u - u_0)(u - u_1) + \dots + A_k(u - u_0) \cdots (u - u_{k-1}) \quad (1.21)$$

En este caso, si se desea interpolar un nuevo punto $(u_{k+1}, f(u_{k+1}))$ que no esté en la tabla (1.16), el polinomio $p_{k+1}(u)$ se obtiene simplemente sumando a (1.21) el término

$$A_{k+1}(u - u_0) \cdots (u - u_k).$$

El valor A_i es llamado la i -ésima diferencia dividida de $f(u)$ en u_0, \dots, u_i y por conveniencia se denota como $f[u_0, \dots, u_i]$ para indicar la dependencia con u_0, \dots, u_i . Finalmente (1.21) se puede escribir como

$$p_k(u) = \sum_{i=0}^k f[u_0, \dots, u_i] \prod_{j=0}^{i-1} (u - u_j) \quad (1.22)$$

en donde $\prod_{j=0}^{-1} (u - u_j) = 1$ y $f[u_0, \dots, u_k]$ es el coeficiente correspondiente de cada función base $\{\prod_{j=0}^{i-1} (u - u_j)\}_{i=0, \dots, k}$.

Propiedades de las diferencias divididas.

A continuación presentaremos algunas propiedades de la diferencia dividida, pensándola no sólo como el coeficiente del polinomio de interpolación, sino también como una función de los nodos u_i , lo que nos será de gran utilidad para probar importantes resultados de las funciones B-spline más adelante. Por otro lado para simplificar la notación diremos "la diferencia dividida de f en U " para indicar $f[u_0, \dots, u_k]$.

Sea un vector de $k + 1$ nodos estrictamente creciente² $U = \{u_0, \dots, u_k\}$ con $k \geq 0$ y una función $f : [a, b] \rightarrow \mathbb{R}$ en donde $a = \min\{u_0, \dots, u_k\}$ y $b = \max\{u_0, \dots, u_k\}$ entonces.

Teorema 3 *La diferencia dividida de f en U cumple*

$$f[u_0, \dots, u_k] = \sum_{i=0}^k \frac{f(u_i)}{(u_i - u_0) \cdots (u_i - u_{i-1})(u_i - u_{i+1}) \cdots (u_i - u_k)} \quad (1.23)$$

y por lo tanto es una función simétrica respecto a sus argumentos, es decir se verifica que

$$f[u_0, \dots, u_k] = f[u_{j_0}, \dots, u_{j_k}] \quad (1.24)$$

para toda permutación (j_0, \dots, j_k) de los subíndices $0, \dots, k$.

Demostración.

Primeramente notemos que el polinomio de interpolación $p_k(u)$, se puede escribir como

$$p_k(u) = f[u_0, \dots, u_k]u^k + \text{un polinomio de grado } < k$$

de donde

$$f[u_0, \dots, u_k] = \frac{p_k^{(k)}(u)}{k!} \quad (1.25)$$

²Más adelante probaremos que bajo ciertas condiciones es posible suponer que la diferencia dividida puede estar definida con un vector de nodos no decreciente

1.2. REPRESENTACIONES DE FUNCIONES POLINOMIALES POR PEDAZOS.

lo que muestra que la k -ésima diferencia dividida $f[u_0, \dots, u_k]$, es $\frac{1}{k!}$ veces la k -ésima derivada de $p_k(u)$ ³. Ahora de (1.18) $p_k(u)$ también tiene la forma

$$p_k(u) = \sum_{i=0}^k \frac{f(u_i)}{\prod_{\substack{j=0 \\ j \neq i}}^k (u_i - u_j)} u^k + \text{un polinomio de grado } < k$$

y entonces por la unicidad del polinomio de interpolación tenemos que

$$f[u_0, \dots, u_k] = \sum_{i=0}^k \frac{f(u_i)}{\prod_{\substack{j=0 \\ j \neq i}}^k (u_i - u_j)}$$

La propiedad de simetría (1.24) se verifica claramente de (1.23) y la conmutatividad de la suma de números reales; veamos:

$$\begin{aligned} f[u_0] &= f(u_0) \\ f[u_0, u_1] &= \frac{f(u_0)}{u_0 - u_1} + \frac{f(u_1)}{u_1 - u_0} = f[u_1, u_0] \\ f[u_0, u_1, u_2] &= f[u_0, u_2, u_1] = f[u_1, u_2, u_0] = \\ &= \frac{f(u_0)}{(u_0 - u_1)(u_0 - u_2)} + \frac{f(u_1)}{(u_1 - u_0)(u_1 - u_2)} + \frac{f(u_2)}{(u_2 - u_0)(u_2 - u_1)} \\ &= f[u_1, u_0, u_2] = f[u_2, u_1, u_0] = f[u_2, u_0, u_1] \end{aligned}$$

■

Teorema 4 La diferencia dividida de f en U se puede escribir como

$$f[u_0, \dots, u_k] = \frac{f[u_0, \dots, u_{k-1}] - f[u_1, \dots, u_k]}{u_0 - u_k} \quad (1.26)$$

Demostración.

De la ecuación (1.21) sabemos que el polinomio p_k de grado menor o igual que k , que interpola a f en la tabla

$$\begin{array}{c|c|c|c} u & u_k & \cdots & u_0 \\ \hline f(u) & f(u_k) & \cdots & f(u_0) \end{array}$$

está dado por

$$\begin{aligned} p_k(u) &= B_0 + B_1(u - u_k) + B_2(u - u_k)(u - u_{k-1}) \\ &+ \cdots + B_{k-1}(u - u_{k-1}) \cdots (u - u_2) + B_k(u - u_k) \cdots (u - u_1) \end{aligned} \quad (1.27)$$

en donde $B_k = f[u_k, \dots, u_0]$. Ahora por la unicidad del polinomio de interpolación y la propiedad de simetría de las diferencias divididas $B_k = f[u_k, \dots, u_0] = f[u_0, \dots, u_k] = A_k$.

Ahora, reescribiendo (1.27) y (1.21) tenemos

$$\begin{aligned} p_k(u) &= A_k(u - u_0) \cdots (u - u_{k-1}) + A_{k-1}u^{k-1} + p'_{k-2}(u) \\ p_k(u) &= B_k(u - u_k) \cdots (u - u_1) + B_{k-1}u^{k-1} + p''_{k-2}(u) \end{aligned}$$

³Usaremos este resultados más adelante.

Demostración. [DeBoor78, pág 5.] Definamos la función

$$F(u) = \sum_{i=0}^k \prod_{r=0}^{i-1} (u - u_r) g[u_0, \dots, u_i] + \sum_{j=0}^k \prod_{s=0}^j (u - u_{s+j+1}) h[u_j, \dots, u_k] \quad (1.30)$$

en donde $\prod_{k=0}^{i-1} (u - u_k) = 1$ si $i = 0$ y $\prod_{k=0}^{r-1} (u - u_{s+j+1}) = 1$ si $j = k$.

Notemos que la primera suma que compone a $F(u)$, es un polinomio de interpolación en la forma de Newton de una función g evaluada en U , y la segunda es el correspondiente polinomio para h ; por tanto $F(u)$ interpola a f evaluada en U .

Ahora por ejemplo si desarrollamos (1.30) para $k = 1$ tenemos

$$\begin{aligned} F(u) &= (g[u_0] + (u - u_0)g[u_0, u_1])((u - u_1)h[u_0, u_1] + h[u_1]) \\ &= g[u_0]h[u_1] + g[u_0, u_1]h[u_1](u - u_0) \\ &\quad + g[u_0]h[u_0, u_1](u - u_1) + g[u_0, u_1]h[u_0, u_1](u - u_0)(u - u_1) \end{aligned}$$

Notemos que último término de esta expresión es igual a cero cuando F se evalúa en u_0 o u_1 , de donde el polinomio formado por las tres sumas anteriores forma al polinomio de grado a lo más 1, que interpola a f en u_0, u_1 , cuyo coeficiente del término de grado 1 es

$$g[u_0, u_1]h[u_1] + g[u_0]h[u_1, u_2]$$

En general, (1.30) se puede descomponer al realizar las multiplicaciones correspondientes en dos sumas, en donde en una se encuentren todos los productos que corresponden al polinomio de interpolación y en la otra los que se anulan al evaluarlos en U ; es decir, la primera suma estará compuesta por los productos cuyos índices cumplan que $i \leq j$, ya que estos productos forman monomios de grado a lo más k , y la otra suma estará formada por productos de grado mayor que k los cuales se anularán al ser evaluados en U , es decir

$$F(u) = \sum_{i \leq j} \text{monomios de grado } \leq k + \sum_{i > j} \text{monomios de grado } > k$$

En particular, los coeficientes de los términos de grado k son aquellos para los cuales $i = j$; es decir

$$\begin{aligned} &g[u_0](u - u_1) \cdots (u - u_k)h[u_0, \dots, u_k] \\ &\quad + (u - u_0)g[u_0, u_1](u - u_2) \cdots (u - u_k)h[u_1, \dots, u_k] \\ &\quad + (u - u_0) \cdots (u - u_k)g[u_0, \dots, u_k]h[u_k] \end{aligned}$$

de donde se deduce (1.29). ■

Teorema 7 Sea $\bar{u} \in \mathbb{R}$ tal que $u_0 \leq \bar{u} < u_k$; entonces

$$(u_0 - u_k)f[u_0, \dots, u_k] + (\bar{u} - u_0)f[u_0, \dots, u_{k-1}, \bar{u}] + (u_k - \bar{u})f[\bar{u}, u_1, \dots, u_k] = 0$$

Demostración.

De la ecuación (1.26), el lado izquierdo de la igualdad anterior es

$$\begin{aligned} &(u_0 - u_k) \frac{f[u_1, \dots, u_k] - f[u_0, \dots, u_{k-1}]}{u_k - u_0} \\ &\quad + (\bar{u} - u_0) \frac{f[u_1, \dots, u_{k-1}, \bar{u}] - f[u_0, \dots, u_{k-1}]}{\bar{u} - u_0} \\ &\quad + (u_k - \bar{u}) \frac{f[u_1, \dots, u_k] - f[\bar{u}, u_1, \dots, u_{k-1}]}{u_k - \bar{u}} \end{aligned}$$

de donde tenemos la expresión

$$\begin{aligned} & -f[u_1, \dots, u_k] + f[u_0, \dots, u_{k-1}] + f[u_1, \dots, u_{k-1}, \bar{u}] \\ & -f[u_0, \dots, u_{k-1}] + f[u_1, \dots, u_k] - f[\bar{u}, u_1, \dots, u_{k-1}] \end{aligned}$$

la cual por la simetría de las diferencias divididas es igual a cero. ■

Según lo mencionado al inicio de la sección, la unicidad y existencia del polinomio de interpolación está garantizada siempre y cuando los nodos no se repitan; sin embargo, los siguientes dos teoremas nos proporcionan implicaciones muy interesantes de las diferencias divididas si f es diferenciable o tiene cierto grado de diferenciable continua, cuya principal consecuencia (para nuestros intereses) radica en el hecho, que bajo tal suposición, será posible definir las diferencias divididas sobre nodos repetidos.

Teorema 8 *Supongamos que f es k veces diferenciable en (a, b) . Entonces existe $\xi \in (a, b)$ tal que*

$$f[u_0, \dots, u_k] = \frac{f^{(k)}(\xi)}{k!} \quad (1.31)$$

Demostración. Supongamos inicialmente que $k = 1$, así debemos probar que existe $\xi \in (a, b)$ tal que

$$f[u_0, u_1] = \frac{f(u_1) - f(u_0)}{u_1 - u_0} = f'(\xi)$$

para $u_0, u_1 \in [a, b]$ y f diferenciable en (a, b) , lo cual por el teorema del valor intermedio es cierto.

En general, observemos que según la hipótesis, la función $q_k(u) = f(u) - p_k(u)$ tiene (por lo menos) $k + 1$ ceros distintos u_0, \dots, u_k en $[a, b]$ (ya que p_k es el polinomio de interpolación de grado menor o igual que k de f en estos puntos), de modo que si f es k veces diferenciable en (a, b) , q_k también lo es, y por el teorema de Rolle $q_k'(u)$ tiene por lo menos k ceros en (a, b) ; continuando con este argumento sucesivamente, tenemos que $q_k^{(k)}$ tiene por lo menos un cero en (a, b) , si por ejemplo lo denotamos como ξ , entonces

$$0 = q_k^{(k)}(\xi) = f^{(k)}(\xi) - p_k^{(k)}(\xi)$$

por otro lado de (1.25)

$$p_k^{(k)}(\xi) = f[u_0, \dots, u_k] k!$$

de donde se obtiene (1.31). ■

Claramente, la k -ésima diferencia dividida $f[u_0, \dots, u_k]$ no está definida si algunos (o todos) valores u_i se repiten, ya que por ejemplo si $k = 1$

$$f[u_0, u_1] = \frac{f(u_1) - f(u_0)}{u_1 - u_0}$$

y el lado derecho de esta igualdad no tiene sentido si $u_1 = u_0$. Sin embargo si la función f es diferenciable se puede hacer que

$$\lim_{u_0, u_1 \rightarrow v} \left(\frac{f(u_1) - f(u_0)}{u_1 - u_0} \right) = f'(v)$$

para algún valor $v \in \mathbb{R}$, de modo que tiene sentido definir

$$f[u_0, u_1] = f'(u_0) \text{ si } u_0 = u_1 = v.$$

El siguiente teorema generaliza esta idea.

1.2. REPRESENTACIONES DE FUNCIONES POLINOMIALES POR PEDAZOS.

Teorema 9 Si f es una función de clase $C^{(k)}$ en $[a, b]$ y $v \in [a, b]$, entonces la diferencia dividida de f en U cumple

$$\lim_{u_0, \dots, u_k \rightarrow v} f[u_0, \dots, u_k] = \frac{1}{k!} f^{(k)}(v)$$

Demostración. Aplicando el teorema anterior sobre nuestras hipótesis tenemos que

$$f[u_0, \dots, u_k] = \frac{1}{k!} f^{(k)}(\xi) \text{ para algún } \xi \in (a, b)$$

De modo que si todos los u_0, \dots, u_k se “acercan” a v , el correspondiente ξ también lo hace, por lo que

$$\lim_{u_0, \dots, u_k \rightarrow v} f[u_0, \dots, u_k] = \lim_{\xi \rightarrow v} \frac{1}{k!} f^{(k)}(\xi) = \frac{1}{k!} f^{(k)}(v);$$

la última igualdad es cierta por la continuidad de $f^{(k)}(u)$. ■

Ahora notemos que si $p_k(u)$ es el polinomio de Newton (1.22) que interpola a la función f de clase $C^{(k)}$ en U , según el teorema anterior, $p_k(u)$ se transforma en

$$\tilde{p}_k(u) = \sum_{i=0}^k \frac{1}{k!} f^{(i)}(v) \prod_{j=0}^{i-1} (u - v) = \sum_{i=0}^k \frac{f^{(i)}(v)(u - v)^i}{i!} \quad (1.32)$$

siempre y cuando los puntos de U se aproximen a v . Es decir $\tilde{p}_k(u)$ se vuelve la serie truncada de Taylor para f alrededor de v , por lo que claramente se cumple que

$$\tilde{p}_k^{(j)}(v) = f^{(j)}(v) \text{ para } j = 0, \dots, k$$

Así, el polinomio \tilde{p}_k no sólo interpola a f en v , sino que además la j -ésima derivada (para $j = 0, \dots, k$) de \tilde{p}_k y f coinciden en este punto. Este tipo de interpolación es llamada **osculatoria** por la palabra latina *ósculos* que significa beso, porque \tilde{p}_k no sólo toca a f en v sino que además lo hace tan suavemente como grados de diferenciabilidad continua tenga f .

De todo lo anterior se deduce es posible definir la diferencia dividida de $f(u)$ en $k + 1$ nodos iguales como

$$f[u_0, \dots, u_k] = \frac{f^{(k)}(v)}{k!} \text{ si } u_0 = u_1 = \dots = u_k = v. \quad (1.33)$$

y entonces tenemos la siguiente definición.

Definición 5 Si f es una función de clase $C^{(k)}$ en $[a, b]$, la diferencia dividida de f sobre U se define como sigue:

- Si $u_0 \leq u_1 \leq \dots \leq u_k$ entonces

$$f[u_0, \dots, u_k] = \begin{cases} \frac{f[u_1, \dots, u_k] - f[u_0, \dots, u_{k-1}]}{u_k - u_0} & \text{si } u_0 \neq u_k \\ \frac{f^{(k)}(u_0)}{k!} & \text{si } u_0 = u_k \end{cases} \quad (1.34)$$

- Si no se cumple que $u_0 \leq u_1 \leq \dots \leq u_k$, entonces tomamos cualquier reordenación v_0, \dots, v_k de estos puntos, de tal manera que $f[v_0, \dots, v_k]$ se defina por el primer caso de (1.34).

La siguiente definición será necesaria para el siguiente teorema.

Definición 6 Si f es lo suficientemente diferenciable definimos la "restricción de f en U ", al vector $f|_U$ formado por $k + 1$ elementos (f_0, \dots, f_k) en donde

$$f_i = f^{(r)}(u_i) \text{ con } r = \max\{j \mid u_{i-j} = u_i\}$$

Por ejemplo, si $U = \{0, 0, 1, 2, 2, 2\}$, $f|_U = (f(0), f'(0), f(1), f(2), f'(2), f''(2))$. La importancia de $f|_U$ se debe a que contiene toda la información acerca de la función f necesaria para construir un polinomio de interpolación de grado a lo más k que coincida con f evaluada U .

Teorema 10 Todos los elementos de la tabla de las diferencias divididas (1.28), son funciones del vector $f|_U$. Es decir, para cualesquiera $0 \leq r \leq s \leq k$, existen números reales d_0, \dots, d_k que dependen de r, s y U pero no de f , tales que

$$f[u_r, \dots, u_s] = \sum_{i=0}^k d_i f_i \quad (1.35)$$

Demostración. La prueba será por inducción sobre la diferencia $s - r$.

- Así supongamos que $s - r = 0$, entonces de (1.35) se tiene si $d_i = 0$ para toda i excepto para alguna $d_j = 1$ cuyo respectivo $f_j = f^{(0)}(u_s)$, en donde $j \leq s$ dependiendo solamente de cuántos u_m hay con $m < r$ y $u_m = u_r$. De hecho, un caso más general de esta suposición es $r \leq s$ y $u_r = \dots = u_s$ ya que entonces $f[u_r, \dots, u_s] = f^{(s-r)}(u_r)/(s-r)!$ y (1.35) se tiene si $d_i = 0$ excepto para un $d_j = 1/(s-r)!$.
- Ahora supongamos que la sentencia es correcta para $s - r < k$ y probemos para $s - r = k$ con $u_r < u_s$. Así, por hipótesis de inducción, existen valores d'_i y d''_i , que dependen sólo de r, s y U tales que $f[u_r, \dots, u_{s-1}] = \sum_i d'_i f_i$ y $f[u_{r+1}, \dots, u_s] = \sum_i d''_i f_i$ entonces

$$f[u_r, \dots, u_s] = \frac{f[u_{r+1}, \dots, u_s] - f[u_r, \dots, u_{s-1}]}{u_s - u_r} = \sum_i \frac{d''_i - d'_i}{u_s - u_r} f_i$$

así tomando

$$d_i = \frac{d''_i - d'_i}{u_s - u_r}$$

tenemos (1.35) en donde d_i es independiente de f . ■

1.2.5. Función B-spline.

Existen varias maneras de definir la función B-spline⁴; sin embargo, nosotros abordaremos el enfoque de las diferencias divididas de la función de potencia truncada [DeBoor78],[Cox72], [Scho46], ya que éste nos permite entre otras cosas, probar múltiples propiedades de la función B-spline.

Definición 7 Dado un entero positivo p y una sucesión de nodos no decreciente $U = \{u_i, \dots, u_{i+p+1}\}$, la i -ésima función B-spline normalizada de grado p (orden $p + 1$) sobre U denotada por $N_{i,p}(u)$ se define como

$$N_{i,p}(u) = (u_{i+p+1} - u_i)(\cdot - u)_+^p [u_i, \dots, u_{i+p+1}] \quad (1.36)$$

para todo $u \in \mathbb{R}$.

⁴El nombre B-spline se refiere al término "Basis", ya que gracias a sus propiedades estas funciones forman una de las bases más comunes para el espacio de funciones Spline.

1.2. REPRESENTACIONES DE FUNCIONES POLINOMIALES POR PEDAZOS.

La notación *placeholder* en $(\cdot - u)_+^p$ se usa para indicar que en la diferencia dividida de la función bivariada $(\bar{u} - u)_+^p$ con $\bar{u} \in U$, se fija primero el valor de u y se le considera solamente como función de \bar{u} , y así la expresión resultante está en función de u . Por ejemplo para $g(\cdot, u)|_{u_i, u_{i+1}}$

$$g(\cdot, u)|_{u_i, u_{i+1}} = \frac{g(\cdot, u)|_{u_{i+1}} - g(\cdot, u)|_{u_i}}{u_{i+1} - u_i} = \frac{g(u_{i+1}, u) - g(u_i, u)}{u_{i+1} - u_i}$$

También notemos que en este caso se está suponiendo que el vector de nodos U , es formado por nodos no decreciente; esto se hace por simplicidad ya que por la definición 5, la diferencia dividida puede estar definida sobre cualquier conjunto de números reales no vacío.

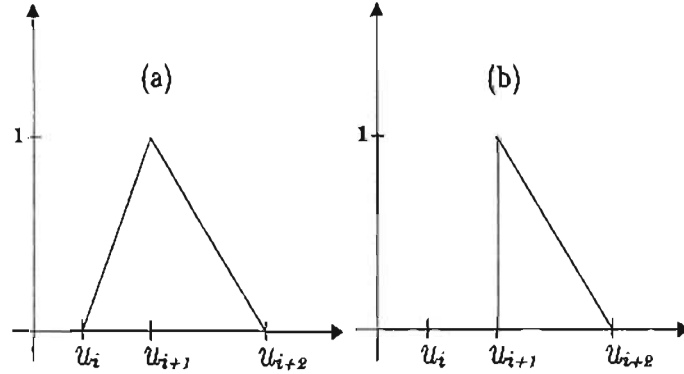


Figura 1.8: (a) Función B-spline $N_{i,1}$ sobre U . (b) Función B-spline $N_{i,1}$ sobre V .

Ejemplo 5 Calculemos la función $N_{i,1}(u)$ sobre los dos vectores de nodos $U = \{u_i, u_{i+1}, u_{i+2}\}$ en donde $u_i < u_{i+1} < u_{i+2}$ y $V = \{u_i, u_{i+1}, u_{i+2}\}$ con $u_i = u_{i+1} < u_{i+2}$.

Así, de (1.36) y (1.26) el B-spline sobre U está dado por

$$\begin{aligned} N_{i,1}(u) &= (u_{i+2} - u_i)(\cdot - u)_+|_{u_i, u_{i+1}, u_{i+2}} \\ &= \frac{(u_{i+2} - u)_+ - (u_{i+1} - u)_+}{u_{i+2} - u_{i+1}} - \frac{(u_{i+1} - u)_+ - (u_i - u)_+}{u_{i+1} - u_i} \\ &= \frac{(u_{i+2} - u)_+}{u_{i+2} - u_{i+1}} - \left(\frac{1}{u_{i+2} - u_{i+1}} + \frac{1}{u_{i+1} - u_i} \right) (u_{i+1} - u)_+ + \frac{(u_i - u)_+}{u_{i+1} - u_i}. \end{aligned}$$

Notemos que en este caso $N_{i,1}(u_i) = N_{i,1}(u_{i+2}) = 0$ y $N_{i,1}(u_{i+1}) = 1$; la gráfica de esta función se muestra en la Figura (1.8) inciso (a).

De manera análoga, el B-spline sobre V es

$$\begin{aligned} N_{i,1}(u) &= (u_{i+2} - u_i)(\cdot - u)_+|_{u_i, u_{i+1}, u_{i+2}} \\ &= (u_{i+2} - u_i) \left[\frac{(\cdot - u)_+|_{u_i, u_{i+2}} - (\cdot - u)_+|_{u_i, u_i}}{u_{i+2} - u_i} \right] \\ &= \frac{(u_{i+2} - u)_+ - (u_i - u)_+}{u_{i+2} - u_i} - (\cdot - u)_+|_{u_i, u_i}. \end{aligned}$$

El segundo término de la última igualdad se calcula utilizando la diferencia dividida para el caso de nodos repetidos en (1.34) así

$$N_{i,1}(u) = \frac{(u_{i+2} - u)_+ - (u_i - u)_+}{u_{i+2} - u_i} - (u_i - u)_+^0 \quad (1.37)$$

y entonces

$$N_{i,1}(u_i) = \frac{u_{i+2} - u_i}{u_{i+2} - u_i} - 0 = 1$$

$$N_{i,1}(u_{i+2}) = \frac{0 - 0}{u_{i+2} - u_i} - 0 = 0;$$

la gráfica de esta función se muestra en la Figura 1.8 inciso (b).

□

De este ejemplo podemos notar que el cálculo de una función B-spline según la definición 7 puede ser bastante engorrosa y en términos numéricos difícil de evaluar; sin embargo, a principios de los setenta de Boor, Cox y Mansfield [Cox72],[DeBoor72],[DeBoor78], descubrieron simultáneamente una expresión recursiva que simplifica enormemente los cálculos.

Teorema 11 Dado un entero $p \geq 0$ y vector de nodos no decreciente $U = \{u_i, \dots, u_{i+p+1}\}$, la función B-spline $N_{i,p}(u)$ cumple

$$N_{i,0}(u) = \begin{cases} 1 & \text{si } u_i \leq u < u_{i+1} \\ 0 & \text{en otro caso.} \end{cases} \quad (1.38)$$

y

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (1.39)$$

Demostración.

La ecuación (1.38) se sigue de las definiciones de la función B-spline, la potencia truncada y la propiedad (1.26)

$$\begin{aligned} N_{i,1}(u) &= (u_{i+1} - u_i)(\cdot - u)_+^0[u_i, u_{i+1}] = (u_{i+1} - u)_+^0 - (u_i - u)_+^0 \\ &= \begin{cases} 1 & \text{si } u \in [u_i, u_{i+1}) \\ 0 & \text{en otro caso.} \end{cases} \end{aligned}$$

Ahora para probar (1.39) notemos que si $\alpha, u \in \mathbb{R}$, la función de potencia truncada $(\alpha - u)_+^p$ cumple

$$(\alpha - u)_+^p = \begin{cases} (\alpha - u)(\alpha - u)_+^{p-1} & \text{si } u < \alpha \\ 0 & \text{en otro caso.} \end{cases} = (\alpha - u)(\alpha - u)_+^{p-1}$$

Entonces, del teorema de Leibniz (ver teorema 6) tenemos que

$$\begin{aligned}
 (\cdot - u)_+^p [u_i, \dots, u_{i+p+1}] &= (u_i - u)(\cdot - u)_+^{p-1} [u_i, \dots, u_{i+p+1}] (\cdot - u)_+^p [u_i, \dots, u_{i+p+1}] \\
 &= \sum_{r=i}^{i+p+1} (\cdot - u) [u_i, \dots, u_r] (\cdot - u)_+^{p-1} [u_r, \dots, u_{i+p+1}] \\
 &= (u_i - u)(\cdot - u)_+^{p-1} [u_i, \dots, u_{i+p+1}] \\
 &\quad + (\cdot - u) [u_i, u_{i+1}] (\cdot - u)_+^{p-1} [u_{i+1}, \dots, u_{i+p+1}] \\
 &\quad + (\cdot - u) [u_i, u_{i+1}, u_{i+2}] (\cdot - u)_+^{p-1} [u_{i+2}, \dots, u_{i+p+1}] \\
 &\quad \vdots \\
 &\quad + (\cdot - u) [u_i, \dots, u_{i+p+1}] (u_{i+p+1} - u)_+^{p-1} \\
 &= (u_i - u)(\cdot - u)_+^{p-1} [u_i, \dots, u_{i+p+1}] + (1)(\cdot - u)_+^{p-1} [u_{i+1}, \dots, u_{i+p+1}]
 \end{aligned}$$

la última igualdad es cierta por el teorema 5, ya que la diferencias divididas de un polinomio de grado $k \geq 2$ que interpola a $(\cdot - u)$ en los números $\{u_i, \dots, u_k\}$ es cero (teorema 5). Ahora

$$\begin{aligned}
 (\cdot - u)_+^p [u_i, \dots, u_{i+p+1}] &= \frac{u_i - u}{u_{i+p+1} - u_i} \left[(\cdot - u)_+^{p-1} [u_{i+1}, \dots, u_{i+p+1}] \right. \\
 &\quad \left. - (\cdot - u)_+^{p-1} [u_i, \dots, u_{i+p}] \right] \\
 &\quad + \frac{(u_{i+p+1} - u_i)(\cdot - u)_+^{p-1} [u_{i+1}, \dots, u_{i+p+1}]}{u_{i+p+1} - u_i} \\
 &= \frac{1}{u_{i+p+1} - u_i} \left[(u - u_i)(\cdot - u)_+^{p-1} [u_i, \dots, u_{i+p}] \right. \\
 &\quad \left. - (u_{i+p+1} - u)(\cdot - u)_+^{p-1} [u_{i+1}, \dots, u_{i+p+1}] \right]
 \end{aligned}$$

y de la definición del B-spline tenemos finalmente

$$(\cdot - u)_+^p [u_i, \dots, u_{i+p+1}] = \frac{1}{u_{i+p+1} - u_i} \left[\frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \right]$$

Es importante notar que en el caso de un vector con nodos repetidos, la ecuación (1.39) puede tener cocientes con denominadores igual a cero, sin embargo eso significa que la función B-spline que multiplica a este cociente, está definida sobre un vector de nodos iguales, de donde por definición es idénticamente cero y por lo tanto el sumando también es cero. ■

Ejemplo 6 Sea la sucesión de nodos $V = \{u_i, u_{i+1}, u_{i+2}\}$ del ejemplo 5; entonces $N_{i,0}(u)$ sobre V está dado por

$$N_{i,1}(u) = \frac{u - u_i}{u_{i+1} - u_i} N_{i,0}(u) + \frac{u_{i+2} - u}{u_{i+2} - u_{i+1}} N_{i+1,0}(u).$$

En este caso el cociente del primer término de la suma está indefinido pero la función $N_{i,0}(u)$ es cero por definición, de donde

$$N_{i,1}(u) = \begin{cases} \frac{u_{i+2} - u}{u_{i+2} - u_{i+1}} & \text{si } u_{i+1} \leq u < u_{i+2} \\ 0 & \text{en otro caso.} \end{cases}$$

Este resultado coincide con (1.37). □

Otra ventaja de la relación recursiva del teorema anterior, es que nos permite calcular o evaluar de forma fácil una función B-spline $N_{i,p}(u)$ realizando los cálculos descritos en la siguiente tabla.

$$\begin{array}{ccccccc}
 N_{i,0}(u) & & & & & & \\
 & N_{i,1}(u) & & & & & \\
 N_{i+1,0}(u) & & N_{i,2}(u) & & & & \\
 & N_{i+1,1}(u) & & & & & \\
 N_{i+2,0}(u) & & & & & & \\
 \vdots & & & & N_{i,p-1}(u) & & \\
 & & & & & N_{i,p}(u) & (1.40) \\
 \vdots & & & & & & \\
 N_{i+p-2,0}(u) & & & & N_{i+1,p-1}(u) & & \\
 & N_{i+p+2,1}(u) & & & & & \\
 N_{i+p-1,0}(u) & & N_{i+p+2,2}(u) & & & & \\
 & N_{i+p-1,1}(u) & & & & & \\
 N_{i+p,0}(u) & & & & & &
 \end{array}$$

Ejemplo 7 Tomemos un vector de nodos $U = \{u_0, u_1, u_2, u_3\} = \{-1, 1, 3, 5\}$ y calculemos la función B-spline $N_{0,2}(u)$ sobre éste. Así primeramente obtenemos

$$N_{i,1}(u) = 1 \quad \text{si } u \in [u_i, u_{i+1})$$

para $i = 0, 1, 2$. Ahora

$$N_{0,1} = \begin{cases} \frac{u+1}{2} & \text{si } u \in [-1, 1) \\ \frac{3-u}{2} & \text{si } u \in [1, 3) \\ 0 & \text{en otro caso.} \end{cases}$$

$$N_{1,1}(u) = \begin{cases} \frac{u-1}{2} & \text{si } u \in [1, 3) \\ \frac{3-u}{2} & \text{si } u \in [1, 2) \\ 0 & \text{en otro caso.} \end{cases}$$

y por último

$$N_{0,2} = \begin{cases} \frac{(u+1)^2}{8} & \text{si } u \in [-1, 1) \\ \frac{(u+1)(3-u)}{8} + \frac{(5-u)(u-1)}{8} & \text{si } u \in [1, 3) \\ \frac{(5-u)^2}{8} & \text{si } u \in [3, 5) \end{cases}$$

Las funciones $N_{0,1}(u)$, $N_{1,1}(u)$ y $N_{0,2}(u)$ se presentan en la Figura 1.9. □

El siguiente teorema nos proporciona una relación importante que nos permite escribir el B-spline como una combinación lineal de B-splines, sobre un vector de nodos "refinado", es decir, sobre un vector con más nodos que el original.

Teorema 12 ([Boehm80]) Dada una sucesión de $p + 2$ nodos no decreciente $U = \{u_i, \dots, u_{i+p+1}\}$ y un nodo \bar{u} tal que $u_k \leq \bar{u} < u_{k+1}$ si definimos

$$\bar{U} = \{\bar{u}_i = u_i, \dots, \bar{u}_k = u_k, \bar{u}_{k+1} = \bar{u}, \bar{u}_{k+2} = u_{k+1}, \dots, \bar{u}_{i+p+2} = u_{i+p+1}\},$$

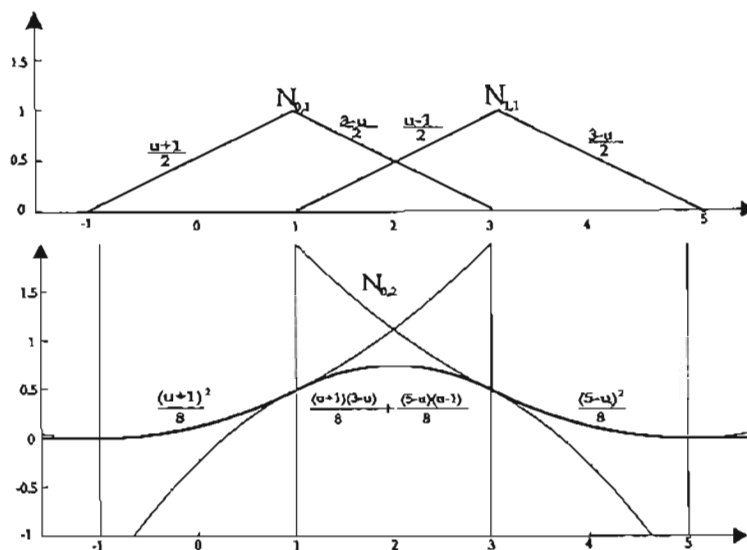


Figura 1.9: Funciones B-spline $N_{0,1}$, $N_{1,1}$ y $N_{0,2}$ sobre $U = \{-1, 1, 3, 5\}$.

la función B-spline $N_{i,p}(u)$ sobre U se puede escribir como

$$N_{i,p}(u) = \frac{\bar{u} - \bar{u}_i}{\bar{u}_{i+p+1} - \bar{u}_i} \bar{N}_{i,p}(u) + \frac{\bar{u}_{i+p+2} - \bar{u}}{\bar{u}_{i+p+2} - \bar{u}_{i+1}} \bar{N}_{i+1,p}(u)$$

en donde $\bar{N}_{i,p}(u)$ y $\bar{N}_{i+1,p}(u)$ son los B-splines definidos en \bar{U} .

Demostración. Del teorema 7, si $f = (\cdot - u)_+^p$ tenemos que

$$\begin{aligned} (u_{i+p+1} - u_i)(\cdot - u)_+^p[u_i, \dots, u_{i+p+1}] &= \\ \frac{\bar{u} - u_i}{\bar{u}_{i+p+1} - \bar{u}_i} (\bar{u}_{i+p+1} - \bar{u}_i)(\cdot - u)_+^p[u_i, \dots, u_{i+p}, \bar{u}] & \\ + \frac{u_{i+p+1} - \bar{u}}{\bar{u}_{i+p+2} - \bar{u}_{i+1}} (\bar{u}_{i+p+2} - \bar{u}_{i+1})(\cdot - u)_+^p[\bar{u}, u_{i+1}, \dots, u_{i+p+1}] & \end{aligned} \quad (1.41)$$

Ahora, de la propiedad de la simetría de las diferencias divididas tenemos

$$\begin{aligned} (\cdot - u)_+^p[u_i, \dots, u_{i+p}, \bar{u}] &= \\ (\cdot - u)_+^p[u_i, \bar{u}, u_{i+2}, \dots, u_{i+p}] &= (\cdot - u)_+^p[\bar{u}_i, \bar{u}_{i+1}, \dots, \bar{u}_{i+p+1}] \end{aligned}$$

y

$$(\cdot - u)_+^p[\bar{u}, u_{i+1}, \dots, u_{i+p+1}] = (\cdot - u)_+^p[\bar{u}_{i+1}, \bar{u}_{i+2}, \bar{u}_{i+3}, \dots, \bar{u}_{i+p+2}]$$

de donde (1.41) se puede escribir como

$$\begin{aligned} (u_{i+p+1} - u_i)(\cdot - u)_+^p[u_i, \dots, u_{i+p+1}] &= \\ \frac{\bar{u} - \bar{u}_i}{\bar{u}_{i+p+1} - \bar{u}_i} (\bar{u}_{i+p+1} - \bar{u}_i)(\cdot - u)_+^p[\bar{u}_i, \dots, \bar{u}_{i+p+1}] & \\ + \frac{\bar{u}_{i+p+2} - \bar{u}}{\bar{u}_{i+p+2} - \bar{u}_{i+1}} (\bar{u}_{i+p+2} - \bar{u}_{i+1})(\cdot - u)_+^p[\bar{u}_{i+1}, \dots, \bar{u}_{i+p+1}] & \end{aligned}$$

y de la definición de la función B-spline tenemos finalmente

$$N_{i,p}(u) = \frac{\bar{u} - \bar{u}_i}{\bar{u}_{i+p+1} - \bar{u}_i} \bar{N}_{i,p}(u) + \frac{\bar{u}_{i+p+2} - \bar{u}}{\bar{u}_{i+p+2} - \bar{u}_{i+1}} \bar{N}_{i+1,p}(u) \quad (1.42)$$

■

Propiedades de las funciones B-spline.

A continuación describiremos algunas de las propiedades más importantes de la función B-spline; para esto supondremos dado un entero $p \geq 0$ y un vector de nodos no decreciente $U = \{u_{i-p}, \dots, u_{i+p+1}\}$.

i) **Soporte local;** $N_{i,p}(u) \neq 0$ si $u \in (u_i, u_{i+p+1})$. Esta propiedad se deduce al observar la tabla (1.40), ya que $N_{i,p}(u)$ se construye a partir de las funciones $N_{i,0}(u)$ las cuales por (1.38) son mayores que cero en $u_i \leq u < u_{i+1}$ para $i = 0, \dots, i+p$.

ii) **No negatividad;** $N_{i,p}(u) \geq 0 \quad \forall i, p, u$. Probaremos esta propiedad por inducción sobre p .

Así por (1.38) es cierta para $p = 0$. Ahora supongamos que es válida para $p-1$, y lo demostraremos para p . Si $u \notin [u_i, u_{i+p+1})$ la propiedad de soporte local implica que

$$N_{i,p-1}(u) = N_{i+1,p-1}(u) = 0$$

Supongamos lo contrario, es decir que $u \in [u_i, u_{i+p+1})$; entonces

$$\frac{u - u_i}{u_{i+p} - u_i}, \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} \geq 0$$

así de (1.39) y la hipótesis de inducción concluimos que $N_{i,p}(u) \geq 0$

iii) **Partición de la unidad.** $\sum_{j=i-p}^i N_{j,p}(u) = 1 \quad \forall u \in [u_i, u_{i+1})$. De la relación recursiva del B-spline esta suma se puede descomponer como

$$\begin{aligned} \sum_{j=i-p}^i N_{j,p}(u) &= \sum_{j=i-p}^i \frac{u - u_j}{u_{j+p} - u_j} N_{j,p-1}(u) \\ &\quad + \sum_{j=i-p}^i \frac{u_{j+p-1} - u}{u_{j+p+1} - u_{j+1}} N_{j+1,p-1}(u) \end{aligned}$$

ahora, cambiando el subíndice de $i-p$ por $i-p+1$ en la segunda suma del lado derecho de la igualdad, podemos escribir

$$\sum_{j=i-p}^i \frac{u - u_j}{u_{j+p} - u_j} N_{j,p-1}(u) + \sum_{j=i-p+1}^{i+1} \frac{u_{j+p} - u}{u_{j+p} - u_j} N_{j,p-1}(u) \quad (1.43)$$

y como $u \in [u_i, u_{i+1})$, la propiedad del soporte local implica que

$$N_{i-p,p-1}(u) = N_{i+1,p-1}(u) = 0$$

y agrupando términos (1.43)

$$\sum_{j=i-p+1}^i \left[\frac{u-u_j}{u_{j+p}-u_j} + \frac{u_{j+p}-u}{u_{j+p}-u_j} \right] N_{j,p-1}(u) = \sum_{j=i-p+1}^i N_{j,p-1}(u)$$

repetiendo el argumento se deduce que

$$\sum_{j=i-p+1}^i N_{j,p-1}(u) = \sum_{i-p+2}^i N_{j,p-2}(u) = \dots = \sum_{j=i}^i N_{j,0}(u) = 1.$$

la última igualdad es cierta por (1.38).

iv) **La Derivada:** La derivada de la función B-spline [Cox82, pág.15],[DeBoor78, pág.138]. Notemos que

$$\frac{d}{du}(\alpha - u)_+^p = -p(\alpha - u)_+^{p-1} \quad \alpha \in \mathbb{R}$$

de donde la derivada respecto a u , de la i -ésima diferencia dividida

$$(\cdot - u)_+^p[u_i, \dots, u_{i+p+1}]$$

es igual a $-p(\alpha - u)_+^{p-1}[u_i, \dots, u_{i+p+1}]$ y entonces

$$\begin{aligned} N'_{i,p}(u) &= (u_{i+p+1} - u_i)(-p)(\cdot - u)_+^{p-1}[u_i, \dots, u_{i+p+1}] \\ &= -p \left[(\cdot - u)_+^{p-1}[u_{i+1}, \dots, u_{i+p+1}] - (\cdot - u)_+^{p-1}[u_i, \dots, u_{i+p}] \right] \end{aligned}$$

de donde

$$N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (1.44)$$

En general si $N_{i,p}^{(j)}(u)$ representa la j -ésima derivada de $N_{i,p}(u)$, la derivación repetida j -veces de (1.44) resulta la fórmula

$$N_{i,p}^{(j)}(u) = p \left(\frac{N_{i,p-1}^{(j-1)}(u)}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}^{(j-1)}(u)}{u_{i+p+1} - u_{i+1}} \right) \quad (1.45)$$

1.2.6. Base de funciones B-spline.

Una vez que se ha definido y discutido las principales características de la función B-spline, es momento de construir la base formada por estas funciones.

Teorema 13 (Curry y Schoenberg) [DeBoor78, pág. 119] Sean k un entero positivo, $U = \{u_0, \dots, u_k\}$ un vector de nodos estrictamente creciente y $\nu = (\nu_1, \dots, \nu_{k-1})$ un vector de enteros (en donde $-1 \leq \nu_i < p$) los cuales representan los grados de diferenciabilidad en cada nodo interior (ver sección 1.1). Entonces, si

$$n = k(p+1) - \sum_{i=1}^{k-1} (\nu_i + 1) = \dim(P_{p,U,\nu})$$

definimos $\bar{U} = \{\bar{u}_0, \dots, \bar{u}_{n+p}\}$ en donde

$$1) \quad \bar{u}_0 \leq \bar{u}_1 \leq \dots \leq \bar{u}_p \leq u_0 \text{ y } u_k \leq \bar{u}_n \leq \dots \leq \bar{u}_{n+p}.$$

11) Cada nodo \bar{u}_{p+i} es igual $p - \nu_i$ veces a u_i para $i = 1, \dots, k - 1$. Es decir

$$\begin{aligned} \bar{u}_{p+1} &= \dots = \bar{u}_{p-\nu_1} = u_1 \\ \bar{u}_{p-\nu_1+1} &= \dots = \bar{u}_{2p-\nu_1-\nu_2} = u_2 \\ &\vdots \\ \bar{u}_{(k-2)p-\sum_{i=1}^{k-2} \nu_i+1} &= \dots = \bar{u}_{(k-1)p-\sum_{i=1}^{k-1} \nu_i} = u_{k-1}. \end{aligned}$$

Entonces el conjunto de funciones B-spline

$$\{N_{i,p}(u)\}_{i=0,\dots,n-1} \quad (1.46)$$

consideradas como funciones sobre $[\bar{u}_p, \bar{u}_n]$ forman una base del espacio $P_{p,U,\nu}$.

Demostración. Inicialmente probemos que cualquier $N_{i,p}(u)$ de (1.46) pertenece al espacio $P_{p,U,\nu}$. El teorema 10, nos indica que si f es cualquier función suficientemente diferenciable existen escalares d_i, \dots, d_{i+p+1} que no dependen de f , tales que

$$f[\bar{u}_i, \dots, \bar{u}_{i+p+1}] = \sum_{r=i}^{i+p+1} d_r f^{(j_r)}(\bar{u}_r)$$

con $j_r = \max\{s \mid r - s \geq i \text{ y } \bar{u}_{r-s} = \bar{u}_r\}$ para $r = i, \dots, i + p + 1$. De donde si $f = (\cdot - u)_+^p$

$$N_{i,p}(u) = (u_{i+p+1} - u_1)(\cdot - u)_+^p|_{u_i, \dots, u_{i+p+1}} = (u_{i+p+1} - u_i) \sum_{r=i}^{i+p+1} d_r (\bar{u}_r - u)_+^{p-j_r} \frac{p!}{(p-j_r)!} \quad (1.47)$$

lo que muestra que $N_{i,p}$ es una función polinomial por pedazos de grado a lo más p con puntos de ruptura $\bar{u}_i, \dots, \bar{u}_{i+p+1}$ los cuales por definición son iguales a ciertos nodos interiores u_1, \dots, u_{k-1} . Además si $\bar{u}_r = u_j$ (para $r \in i, \dots, i + p + 1$ y $j \in 1, \dots, k - 1$), las propiedades de la función de potencia truncada implican que esta función tiene $p - j_r - 1$ grados de diferenciableidad en u_j (es decir, es de clase $C^{(p-j_r-1)}$).

Por otro lado notemos que por definición, j_r cuenta el número de nodos iguales que u_j (no incluyendo a u_j), y $p - \nu_j$ la cantidad de nodos iguales que u_j (incluyéndolo), de donde $p - \nu_i = j_r + 1$ es decir $\nu_j = p - j_r - 1$, así $N_{i,p}$ es de clase $C^{(\nu_j)}$ en u_j y por lo tanto $N_{i,p} \in P_{p,U,\nu}$.

Ahora, lo único que falta probar es que (1.46) es linealmente independiente. Para esto usaremos la fórmula recursiva de los B-splines realizando inducción sobre p .

- Supongamos que $p = 0$; así, si

$$\sum_{i=0}^{n-1} \alpha_i N_{i,0}(u) = 0 \text{ con } \alpha_i \in \mathbb{R}$$

entonces $\alpha_i = 0 \quad \forall i$, ya que

$$N_{i,0}(u) = \begin{cases} 1 & \text{si } u \in [u_i, u_{i+1}) \\ 0 & \text{c.o.c.} \end{cases}$$

lo que implica que $\alpha_i = 0$ para $i = 0, \dots, n - 1$.

- Ahora supongamos cierto para $p - 1$ y probemos para p ;

$$\sum_{i=0}^{n-1} \alpha_i N_{i,p}(u) = 0$$

Entonces si derivamos esta ecuación usando la propiedad (iv), tenemos

$$\begin{aligned} 0 &= \left(\sum_{i=0}^{n-1} \alpha_i N_{i,p}(u) \right)' = \sum_{i=0}^{n-1} \alpha_i N'_{i,p}(u) \\ &= p \sum_{i=0}^{n-1} \alpha_i \left(\frac{N_{i,p-1}(u)}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}(u)}{u_{i+p+1} - u_{i+1}} \right) \end{aligned}$$

entonces

$$0 = \sum_{i=0}^{n-1} \alpha_i \frac{N_{i,p-1}(u)}{u_{i+p} - u_i} - \sum_{i=0}^{n-1} \alpha_i \frac{N_{i+1,p-1}(u)}{u_{i+p+1} - u_{i+1}}$$

ahora notando que por definición $N_{0,p-1}(u) = N_{k-p,p-1}(u) = 0$, y además cambiamos el subíndice de $i+1$ por i en el segundo sumando, tenemos

$$\begin{aligned} 0 &= \sum_{i=1}^{n-1} \alpha_i \frac{N_{i,p-1}(u)}{u_{i+p} - u_i} - \sum_{i=1}^{n-1} \alpha_{i-1} \frac{N_{i,p-1}(u)}{u_{i+p} - u_i} \\ &= \sum_{i=1}^{n-1} \frac{\alpha_i - \alpha_{i-1}}{u_{i+p} - u_i} N_{i,p-1}(u) \end{aligned}$$

de donde por hipótesis de inducción $\alpha_i - \alpha_{i-1} = 0$, y ya que los valores de α_i pueden ser cualesquiera se deduce que $\alpha_i = 0 \quad \forall i$. ■

Ejemplo 8 Si $p = 2$, $U = \{u_0, u_1, u_2, u_3, u_4\} = \{-1, 2, 5, 6, 9\}$ y $\nu = (-1, 0, 1)$, obtengamos la base de funciones B-splines de $P_{2,U,\nu}$ formada por todas las fpp de a lo más grado 2 sobre U las cuales son discontinuas por la izquierda en u_1 , continuas en u_2 y de clase $C^{(1)}$ en u_3 .

Entonces, según la demostración del teorema anterior, la base está formada por $n = \dim(P_{2,U,\nu}) = (3)(4) - 3 = 9$ funciones y por lo tanto $\bar{U} = \{\bar{u}_0, \dots, \bar{u}_{11}\}$ con

1. $\bar{u}_0 \leq \bar{u}_1 \leq \bar{u}_2 \leq u_0 = -1$ y $9 = u_3 \leq \bar{u}_9 \leq \bar{u}_{10} \leq \bar{u}_{11}$.
2. $\bar{u}_3 = \bar{u}_4 = \bar{u}_5 = 2, \bar{u}_8 = \bar{u}_7 = 5, \bar{u}_8 = 6$.

si tomamos $\bar{u}_0 = \bar{u}_1 = \bar{u}_2 = u_0$ y $\bar{u}_9 = \bar{u}_{10} = \bar{u}_{11} = u_4$ entonces

$$\bar{U} = \{-1, -1, -1, 2, 2, 2, 5, 5, 6, 9, 9, 9\}$$

y así la base está formada por el conjunto de funciones

$$\{N_{i,2}(u)\}_{i=0,\dots,8}$$

sobre \bar{U} en donde $u \in [\bar{u}_2, \bar{u}_9) = [-1, 9]$.

Para este caso por ejemplo las funciones $N_{3,2}(u)$, $N_{4,2}(u)$ según la ecuación (1.47) son

$$\begin{aligned} N_{2,2}(u) &= 3[d_2(-1-u)_+^2 + d_3(2-u)_+^2 + 2d_4(2-u)_+ + 2d_5(2-u)_+^0] \\ N_{3,2}(u) &= 3[d_3(2-u)_+^2 + 2d_4(2-u)_+ + d_5(2-u)_+^0 + 2d_6(5-u)_+^2] \end{aligned}$$

las cuales claramente son discontinuas en 2 como se esperaba.

Obtendremos las expresiones explícitas de las funciones que forman la base usando la tabla de los B-splines (1.40). Así tenemos

$$N_{0,0} = N_{1,0} = N_{3,0} = N_{4,0} = N_{6,0} = N_{9,0} = N_{10,0} = 0.$$

$$N_{2,0}(u) = \begin{cases} 1 & \text{si } u \in [-1, 2) \\ 0 & \text{en otro caso} \end{cases} \quad N_{5,0}(u) = \begin{cases} 1 & \text{si } u \in [2, 5) \\ 0 & \text{en otro caso} \end{cases}$$

$$N_{7,0}(u) = \begin{cases} 1 & \text{si } u \in [5, 6) \\ 0 & \text{en otro caso} \end{cases} \quad N_{8,0}(u) = \begin{cases} 1 & \text{si } u \in [6, 9) \\ 0 & \text{en otro caso} \end{cases}$$

Ahora las funciones B-spline de grado 1 son $N_{0,1} = N_{3,1} = N_{9,1} = 0$

$$N_{1,1}(u) = \begin{cases} \frac{2-u}{3} & \text{si } u \in [-1, 2) \\ 0 & \text{en otro caso} \end{cases} \quad N_{2,1}(u) = \begin{cases} \frac{u+1}{3} & \text{si } u \in [-1, 2) \\ 0 & \text{en otro caso} \end{cases}$$

$$N_{4,1}(u) = \begin{cases} \frac{5-u}{3} & \text{si } u \in [2, 5) \\ 0 & \text{en otro caso} \end{cases} \quad N_{5,1}(u) = \begin{cases} \frac{u-2}{3} & \text{si } u \in [2, 5) \\ 0 & \text{en otro caso} \end{cases}$$

$$N_{6,1}(u) = \begin{cases} 6-u & \text{si } u \in [5, 6) \\ 0 & \text{en otro caso} \end{cases} \quad N_{7,1}(u) = \begin{cases} u-5 & \text{si } u \in [5, 6) \\ \frac{9-u}{3} & \text{si } u \in [6, 9) \\ 0 & \text{en otro caso} \end{cases}$$

$$N_{8,1}(u) = \begin{cases} \frac{u-6}{3} & \text{si } u \in [6, 9) \\ 0 & \text{en otro caso} \end{cases}$$

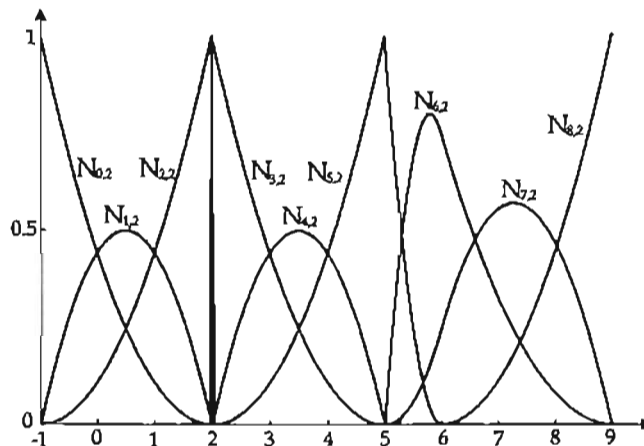


Figura 1.10: Base de funciones B-spline de grado 2, con $U = \{-1, 2, 5, 6, 9\}$ y $\nu = (-1, 0, 1)$.

Por último las funciones B-spline de grado 2 son las siguientes, éstas se presentan en la Figura 1.10.

$$\begin{aligned}
 N_{0,2}(u) &= \begin{cases} \frac{(2-u)^2}{9} & \text{si } u \in [-1, 2) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{1,2}(u) &= \begin{cases} \frac{(u+1)(2-u)}{9} + \frac{(2-u)(u+1)}{9} & \text{si } u \in [-1, 2) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{2,2}(u) &= \begin{cases} \frac{(u+1)^2}{9} & \text{si } u \in [-1, 2) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{3,2}(u) &= \begin{cases} \frac{(5-u)^2}{9} & \text{si } u \in [2, 5) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{4,2}(u) &= \begin{cases} \frac{2(u-2)(5-u)}{9} & \text{si } u \in [2, 5) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{5,2}(u) &= \begin{cases} \frac{(u-2)^2}{9} + (6-u)^2 & \text{si } u \in [5, 6) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{6,2}(u) &= \begin{cases} (u-5)(6-u) + \frac{(9-u)(u-5)}{4} & \text{si } u \in [5, 6) \\ \frac{(9-u)^2}{12} & \text{si } u \in [6, 9) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{7,2}(u) &= \begin{cases} \frac{(u-5)^2}{4} & \text{si } u \in [5, 6) \\ \frac{(u-5)(9-u)}{12} + \frac{(9-u)(u-6)}{9} & \text{si } u \in [6, 9) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{8,2}(u) &= \begin{cases} \frac{(u-6)^2}{9} & \text{si } u \in [6, 9) \\ 0 & \text{en otro caso} \end{cases}
 \end{aligned}$$

□

Tanto de la ecuación (1.47) como del ejemplo anterior, se puede deducir que los grados de diferenciablez de las funciones B-spline de cierto grado p en un nodo interior se reducen a medida de que éste se repite (y por lo tanto la fpp representada por esta base). Es decir si por ejemplo la función $N_{i,p}(u)$ es distinta de cero en un nodo u_i el cual no se repite, entonces ésta es de clase $C^{(p-1)}$ en este nodo, en caso de que se repita exactamente dos veces, tendremos continuidad de clase $C^{(p-2)}$ y así sucesivamente. A continuación presentaremos un ejemplo de esto.

Ejemplo 9 Tomemos la función B-spline $N_{0,2}(u)$ de ejemplo 7, es decir

$$N_{0,2}(u) = \begin{cases} \frac{(u+1)^2}{8} & \text{si } u \in [-1, 1) \\ \frac{(u+1)(3-u)}{8} + \frac{(5-u)(u-1)}{2} & \text{si } u \in [1, 3), \\ \frac{(5-u)^2}{8} & \text{si } u \in [3, 5) \\ 0 & \text{en otro caso} \end{cases}$$

en donde $U = \{-1, 1, 3, 5\}$ que se muestra en la Figura 1.11 (a). La cual se puede probar que es de clase $C^{(1)}$ en 1 y 3. Ahora supongamos que repetimos el nodo 1 reemplazándolo por 3, es decir $U = \{-1, 1, 1, 5\}$; entonces la función resulta

$$N_{0,2}(u) = \begin{cases} \frac{(u+1)^2}{4} & \text{si } u \in [-1, 1) \\ \frac{(5-u)^2}{16} & \text{si } u \in [1, 5) \\ 0 & \text{en otro caso} \end{cases}$$

lo que implica una continuidad de $C^{(0)}$ en 1, como se muestra en la Figura 1.11 (b). Por último, volvamos a insertar 1 ahora reemplazándolo con 5; así $U = \{-1, 1, 1, 1\}$ y entonces tenemos

$$N_{0,2}(u) = \begin{cases} \frac{u+1}{2} & \text{si } u \in [-1, 1) \\ 0 & \text{en otro caso} \end{cases}$$

la cual es discontinua en 1, como se observa en la Figura 1.11 (c).

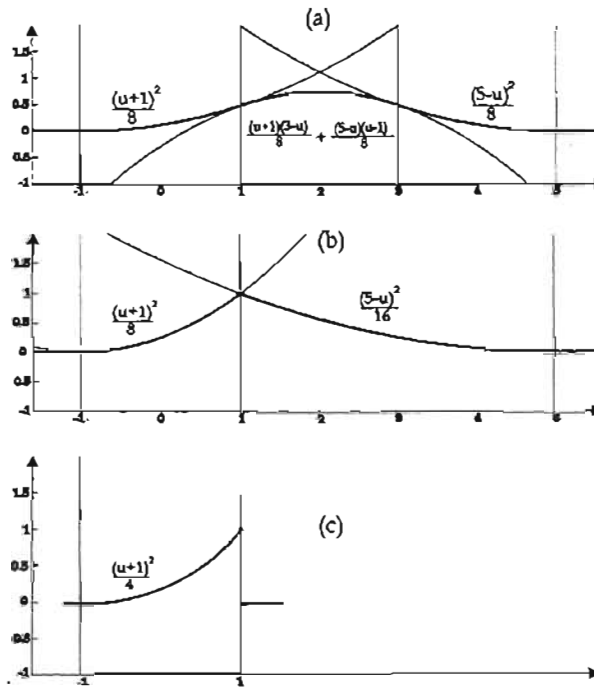


Figura 1.11: Función B-spline $N_{0,2}$ definida sobre $U = \{-1, 1, 3, 5\}$ en la parte superior, sobre $U = \{-1, 1, 1, 5\}$ en medio y $U = \{-1, 1, 1, 1\}$ en la parte inferior.

□

Ejemplo 10 Recordemos que en el ejemplo 4, se representó una “constante rota” usando la base de funciones de potencia truncada, y vimos por qué en este caso se pierde precisión. Ahora representaremos la misma función pero con la base de funciones B-spline.

Aquí el vector de nodos debe ser $U = \{u_0, \dots, u_9\} = \{0, 1, 2, 3, 4.4, 4.6, 6, 7, 8, 9\}$, $p = 1$ y $\nu = (\nu_1, \dots, \nu_8) = (0, \dots, 0)$.

Entonces $n = (2)(9) - 8 = 10$ y $\bar{U} = \{\bar{u}_0, \dots, \bar{u}_{11}\}$. Tomando $\bar{u}_0 = \bar{u}_1 = u_0$ y $\bar{u}_9 = \bar{u}_{10} = u_9$ tenemos que $\bar{U} = \{0, 0, 1, 2, 3, 4.4, 4.6, 6, 7, 8, 9, 9\}$.

Así las funciones B-spline que forma esta base son

$$\begin{aligned}
 N_{0,1}(u) &= \begin{cases} 1-u & \text{si } u \in [0, 1) \\ 0 & \text{en otro caso} \end{cases} & N_{1,1}(u) &= \begin{cases} u & \text{si } u \in [0, 1) \\ 2-u & \text{si } u \in [1, 2) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{2,1}(u) &= \begin{cases} u-1 & \text{si } u \in [1, 2) \\ 3-u & \text{si } u \in [2, 3) \\ 0 & \text{en otro caso} \end{cases} & N_{3,1}(u) &= \begin{cases} u-2 & \text{si } u \in [2, 3) \\ \frac{4.4-u}{1.4} & \text{si } u \in [3, 4.4) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{4,1}(u) &= \begin{cases} \frac{u-3}{1.4} & \text{si } u \in [3, 4.4) \\ \frac{4.6-u}{0.2} & \text{si } u \in [4.4, 4.6) \\ 0 & \text{en otro caso} \end{cases} & N_{5,1}(u) &= \begin{cases} \frac{u-4.4}{0.2} & \text{si } u \in [4.4, 4.6) \\ \frac{6-u}{1.4} & \text{si } u \in [4.6, 6) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{6,1}(u) &= \begin{cases} \frac{u-4.6}{1.4} & \text{si } u \in [4.6, 6) \\ 7-u & \text{si } u \in [6, 7) \\ 0 & \text{en otro caso} \end{cases} & N_{7,1}(u) &= \begin{cases} u-6 & \text{si } u \in [6, 7) \\ 8-u & \text{si } u \in [7, 8) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{8,1}(u) &= \begin{cases} u-7 & \text{si } u \in [7, 8) \\ 9-u & \text{si } u \in [8, 9) \\ 0 & \text{en otro caso} \end{cases} & N_{9,1}(u) &= \begin{cases} u-8 & \text{si } u \in [8, 9) \\ 0 & \text{en otro caso} \end{cases}
 \end{aligned}$$

Ahora la función spline $s \in S_{1,U}$ que representa a la recta rota está dada por $s(u) = \sum_{i=0}^9 \alpha_i N_{i,1}(u)$ para $u \in [0, 9)$ en donde $\alpha_0 = \alpha_1 = \alpha_2 = \alpha_3 = 10$, $\alpha_4 = 5$, $\alpha_5 = 15$ y $\alpha_6 = \alpha_7 = \alpha_8 = \alpha_9 = 10$, en este caso si $u = 8.5$ tenemos

$$s(8.5) = 10N_{8,1}(8.5) + 10N_{9,1}(8.5) = 10(0.5) + 10(0.5) = 10.$$

es decir obtenemos el resultado esperado. Esto se debe a la propiedad del soporte local de las funciones B-spline ya que para esta evaluación solamente 2 funciones base están involucradas.

□

Capítulo 2

Curvas NURBS

Una vez discutidas las principales características de las funciones B-spline, este capítulo lo dedicaremos a estudiar una importante familia de curvas paramétricas definidas en términos de B-splines, llamadas “Non Uniform Rational B-spline” (NURBS).

2.1. Curva de Bézier integral.

La idea de la siguiente exposición es partir de lo más particular a lo más general, así iniciaremos con las curvas de Bézier integrales las cuales en este contexto son las más simples pero a la vez más importantes. Una manera intuitiva y bastante útil para definir las es por medio del llamado algoritmo de Casteljaeu que discutiremos a continuación.

El Algoritmo de Casteljaeu

El algoritmo de Casteljaeu es un proceso recursivo con el que se obtienen puntos ya sea en el plano o espacio de una curva Bézier integral, lo presentaremos usando el siguiente ejemplo.

Ejemplo 11 Sean P_0, P_1, P_2, P_3 puntos en \mathbb{R}^2 o \mathbb{R}^3 y un real $t \in [0, 1]$. Si fijamos $t = \frac{1}{2}$ y denotamos $P_{i,0} = P_i$, (para $i = 0, 1, 2, 3$) el algoritmo de Casteljaeu para este caso consiste en los siguientes pasos:

- **Paso 1:** Calcular el punto medio entre $P_{i,0}$ y $P_{i+1,0}$, que denotaremos por $P_{i,1}(\frac{1}{2})$, es decir

$$P_{i,1}\left(\frac{1}{2}\right) = \frac{1}{2}P_{i,0} + \frac{1}{2}P_{i+1,0}$$

para $i = 0, 1, 2$.

- **Paso 2:** Calcular el punto medio entre $P_{i,1}(\frac{1}{2})$ y $P_{i+1,1}(\frac{1}{2})$ denotado por $P_{i,2}(\frac{1}{2})$

$$P_{i,2}\left(\frac{1}{2}\right) = \frac{1}{2}P_{i,1}\left(\frac{1}{2}\right) + \frac{1}{2}P_{i+1,1}\left(\frac{1}{2}\right)$$

para $i = 0, 1$.

- **Paso 3:** Y finalmente calcular el punto medio entre $P_{0,2}(\frac{1}{2})$ y $P_{1,2}(\frac{1}{2})$ denotado por $P_{0,3}(\frac{1}{2})$

$$P_{0,3}\left(\frac{1}{2}\right) = \frac{1}{2}P_{0,2}\left(\frac{1}{2}\right) + \frac{1}{2}P_{1,2}\left(\frac{1}{2}\right)$$

2.1. CURVA DE BÉZIER INTEGRAL.

Se dice entonces que $P_{0,3}(\frac{1}{2})$ es un punto de parámetro $t = \frac{1}{2}$, sobre la curva de Bézier de integral cúbica definida por P_0, P_1, P_2 y P_3 . Gráficamente, estos tres pasos se presentan en la Figura 2.1(a).

Si ahora tomamos $t = \frac{2}{3}$, los nuevos puntos estarán ubicados en el segundo tercio de la línea recta que une cada par, es decir los pasos son

- **Paso 1:** $P_{i,1}(\frac{2}{3}) = \frac{1}{3}P_{i,0} + \frac{2}{3}P_{i+1,0}$ para $i = 0, 1, 2$.
- **Paso 2:** $P_{i,2}(\frac{2}{3}) = \frac{1}{3}P_{i,1}(\frac{2}{3}) + \frac{2}{3}P_{i+1,1}(\frac{2}{3})$ para $i = 0, 1$.
- **Paso 3:** $P_{0,3}(\frac{2}{3}) = \frac{1}{3}P_{0,2}(\frac{2}{3}) + \frac{2}{3}P_{1,2}(\frac{2}{3})$

Y de nuevo se dice que $P_{0,3}(\frac{2}{3})$ es un punto de parámetro $t = \frac{2}{3}$ sobre la curva de Bézier. éstos se muestran en la Figura 2.1(b).

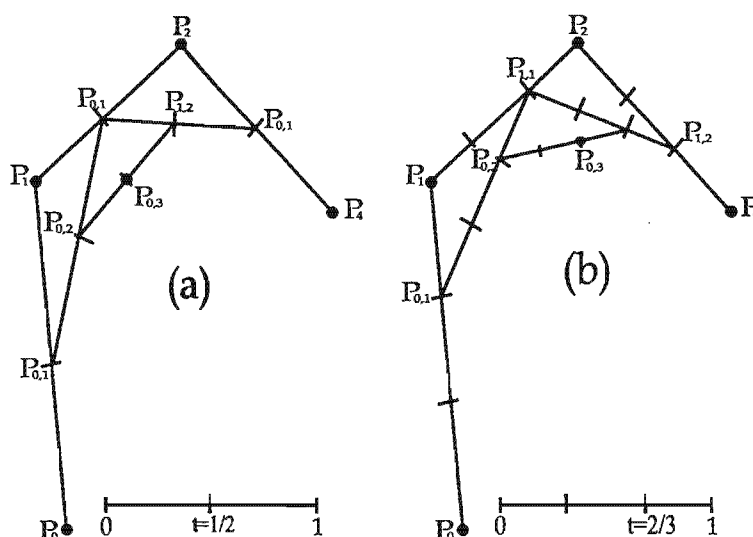


Figura 2.1: Puntos obtenidos por el algoritmo de Casteljau con P_0, P_1, P_2, P_3 y $t = \frac{1}{2}$ (a) y $t = \frac{2}{3}$ (b).

En general, se puede establecer que para cualquier parámetro $t \in [0, 1]$, los pasos que componen al algoritmo son

- **Paso 1:** $P_{i,1}(t) = (1-t)P_{i,0}(t) + tP_{i+1,0}(t)$ para $i = 0, 1, 2$.
- **Paso 2:** $P_{i,2}(t) = (1-t)P_{i,1}(t) + tP_{i+1,1}(t)$ para $i = 0, 1$.
- **Paso 3:** $P_{i,3}(t) = (1-t)P_{0,2}(t) + tP_{1,2}(t)$.

que notemos, se puede escribir como

$$P_{i,r}(t) = (1-t)P_{i,r-1}(t) + tP_{i+1,r-1}(t)$$

para $r = 1, 2, 3$ y $i = 1, \dots, 3 - r$. Así, el conjunto de todos los puntos $P_{0,3}(t)$ para todos los valores $t \in [0, 1]$, forma la curva de Bézier de integral de grado 3, que presentamos en la Figura 2.2. □

En la siguiente definición se formaliza el proceso descrito en este ejemplo.

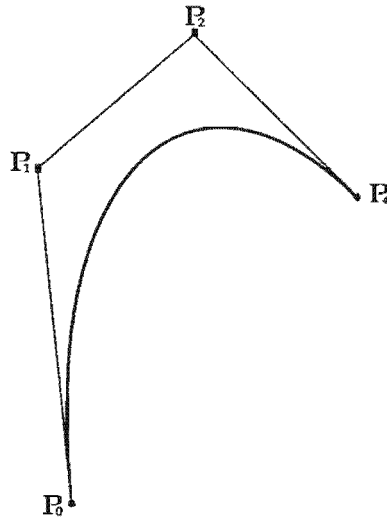


Figura 2.2: Curva generada por el algoritmo de Casteljau con puntos de control P_0, P_1, P_2, P_3 y $t \in [0, 1]$

Definición 8 Dado un conjunto de puntos $P_0, P_1, \dots, P_p \in \mathbb{R}^2$ o \mathbb{R}^3 llamados de control, y $t \in [0, 1]$ el **algoritmo de Casteljau** es el proceso que calcula puntos de una curva de Bézier por medio de la siguiente ecuación

$$P_{i,r}(t) = (1-t)P_{i,r-1}(t) + tP_{i+1,r-1}(t) \quad (2.1)$$

para $r = 1, \dots, p, i = 0 \dots, p-r$ con $P_{j,0}(t) = P_j$ para $j = 0, \dots, p$.

Así, una **curva de Bézier integral** de grado p definida por P_0, P_1, \dots, P_p , que denotaremos por $\mathbf{B}(t)$. Es aquella que está formada por los puntos $P_{0,p}(t)$ para todos los $t \in [0, 1]$. El polígono formado por los puntos de control es llamado **polígono de Control**.

La curva de Bézier y la base de Bernstein.

Es importante notar que los puntos en (2.1), son sumas de productos de t y $1-t$, lo que resulta un polinomio, de donde es claro que $\mathbf{B}(t)$ es una curva polinomial (es decir sus funciones coordenadas son polinomios). En la siguiente discusión probaremos que ésta se puede representar usando una base de funciones polinomiales, llamadas de Bernstein¹ las cuales definimos a continuación.

Definición 9 Dado un entero positivo p y $t \in [0, 1]$, el i -ésimo **polinomio de Bernstein** de grado p se define como

$$B_{i,p}(t) = \binom{p}{i} t^i (1-t)^{p-i} \quad (2.2)$$

con

$$\binom{p}{i} = \begin{cases} \frac{p!}{i!(p-i)!} & \text{si } 0 \leq i \leq p \\ 0 & \text{c.o.c.} \end{cases}$$

para $i = 0, \dots, p$.

Algunas de las principales propiedades de las funciones de Bernstein son las siguientes

- 1) $B_{i,p}(t) \geq 0$. Esto se sigue de (2.2).

¹Que veremos más adelante son un caso particular de los B-splines.

ii) **Partición de la Unidad:** $\sum_{i=0}^p B_{i,p}(t) = 1$.

Por el teorema del binomio tenemos que

$$\sum_{i=0}^p B_{i,p}(t) = \sum_{i=0}^p \binom{p}{i} t^i (1-t)^{p-i} = (t+1-t)^p = 1$$

iii) $B_{0,p}(0) = B_{p,p}(1) = 1$. Como

$$B_{0,p}(t) = \binom{p}{0} t^0 (1-t)^p = (1-t)^p$$

$$B_{p,p}(t) = \binom{p}{p} t^p (1-t)^0 = t^p$$

tenemos que $B_{0,p}(0) = B_{p,p}(1) = 1$.

iv) $B_{i,p}(t) = (1-t)B_{i,p-1}(t) + tB_{i-1,p-1}(t)$.

Esto se sigue del hecho

$$\begin{aligned} B_i^p(t) &= \binom{p-1}{i} t^i (1-t)^{p-i} \\ &= \left[\binom{p-1}{i} + \binom{p-1}{i-1} \right] t^i (1-t)^{p-i} \\ &= \binom{p-1}{i} t^i (1-t)^{p-i} + \binom{p-1}{i-1} t^i (1-t)^{p-i} \\ &= (1-t) \binom{p-1}{i} t^i (1-t)^{p-i-1} + t \binom{p-1}{i-1} t^{i-1} (1-t)^{p-i} \\ &= (1-t) B_{i,p-1}(t) + t B_{i-1,p-1}^p(t) \end{aligned}$$

v) $B'_{i,p}(t) = p[B_{i-1,p-1}(t) - B_{i,p-1}(t)]$

$$\begin{aligned} B'_{i,p}(t) &= \frac{d}{dt} \left[\binom{p}{i} t^i (1-t)^{p-i} \right] \\ &= \binom{p}{i} [it^{i-1}(1-t)^{p-i} - t^i(p-i)(1-t)^{p-i-1}] \\ &= \frac{p!}{(i-1)!(p-i)!} t^{i-1}(1-t)^{p-i} - \frac{p!}{i!(p-i-1)!} t^i(1-t)^{p-i-1} \\ &= p \binom{p-1}{i-1} t^{i-1}(1-t)^{p-i} - p \binom{p-1}{i} t^i(1-t)^{p-i-1} \\ &= p[B_{i-1,p-1}(t) - B_{i,p-1}(t)] \end{aligned}$$

La siguiente proposición da sentido a la afirmación de que los polinomios de Bernstein forman una base.

Proposición 1 Dado un entero positivo p y $t \in (0, 1)$ el conjunto de polinomios de Bernstein

$$\{B_{i,p}(t)\}_{i=0}^p \tag{2.3}$$

forma una base del conjunto de funciones polinomiales por pedazos $P_{p,U}$ (ver sección 1.1) en donde $U = \{u_0, u_1\} = \{0, 1\}$.

Demostración.

Primeramente notemos que por definición cada $B_{i,p} \in P_{p,U}$ para $i = 0, \dots, p$. Ahora probaremos la independencia lineal de (2.3) por inducción sobre p , así si $p = 1$ el conjunto $\{1 - t, t\}$ es linealmente independiente, es decir

$$\alpha_0(1 - t) + \alpha_1 t = 0 \text{ entonces } \alpha_0 = \alpha_1 = 0$$

ya que $t \in (0, 1)$. Supongamos cierto para $p - 1$ y demostremos para p , así supongamos

$$\sum_{i=0}^p \alpha_i B_{i,p}(u) = 0$$

entonces usando la propiedad IV de los polinomios de Bernstein tenemos que

$$\begin{aligned} & \sum_{i=0}^p \alpha_i p [B_{i-1,p-1}(u) - B_{i,p-1}(u)] = \\ & p \left[\sum_{i=0}^p \alpha_i B_{i-1,p-1} - \sum_{i=0}^p \alpha_i B_{i,p-1} \right] = 0 \end{aligned}$$

ahora por definición $B_{-1,p-1}(u) = B_{p,p-1} = 0$, de donde

$$p \left[\sum_{i=1}^p \alpha_i B_{i-1,p-1}(u) - \sum_{i=0}^{p-1} \alpha_i B_{i,p-1}(u) \right] = 0$$

cambiando subíndices

$$\begin{aligned} & p \left[\sum_{i=1}^p \alpha_i B_{i-1,p-1} - \sum_{i=1}^p \alpha_{i-1} B_{i-1,p-1} \right] = \\ & p \left[\sum_{i=1}^p (\alpha_i - \alpha_{i-1}) B_{i-1,p-1}(u) \right] = 0 \end{aligned}$$

y por hipótesis de inducción concluimos que $\alpha_i - \alpha_{i-1} = 0$, y ya que éstos son escalares cualesquiera tenemos que $\alpha_i = 0$ para toda i , es decir (2.3) es un conjunto linealmente independiente.

Por ultimo notemos que según (1.1), la dimensión de este espacio es

$$\dim(P_{p,U}) = 1 \cdot (p + 1) - 0 = p + 1$$

ya que en este caso el valor de k es igual a 1 y no existe un vector de grados de diferenciabilidad continua en las uniones de los polinomios, pues este espacio está formado por las funciones polinomiales de un solo pedazo. Así la dimensión coincide con la cantidad de funciones de (2.3) y concluimos que éste forma una base de $P_{p,U}$. ■

Ahora presentaremos un teorema muy importante, que indica que los puntos obtenidos por el algoritmo de Casteljau se pueden escribir en términos de los polinomios de Bernstein.

Teorema 14 [Far90, página 42] *Dado un entero positivo p y puntos de control P_0, \dots, P_p , de una curva de Bézier de grado p , entonces*

$$P_{i,r}(t) = \sum_{j=0}^r P_{i+j} B_{j,r}(t) \tag{2.4}$$

para $r = 0, \dots, p$ y $i = 0, \dots, p - r$ y $t \in [0, 1]$, en donde $P_{i,r}(t)$ es el i -ésimo punto obtenido en el r -ésimo paso del algoritmo de Casteljau (2.1) con parámetro $t \in [0, 1]$.

Demostración.

Esta demostración se hará por inducción sobre r , usando las propiedades de los polinomios de Bernstein presentadas anteriormente. Así si $r = 0$

$$P_{i,0}(t) = P_i = \sum_{j=0}^0 P_i B_j^0(t)$$

por la propiedad III. Ahora supongamos que se cumple para $r - 1$ es decir

$$P_{i,r-1}(t) = \sum_{j=0}^{r-1} P_{i+j} B_{j,r-1}(t) \quad (2.5)$$

y probemos para r . Entonces del algoritmo de Casteljaou

$$P_{i,r}(t) = (1-t)P_{i,r-1}(t) + tP_{i+1,r-1}(t) \quad (2.6)$$

así sustituyendo (2.5) en (2.6) tenemos

$$\begin{aligned} P_{i,r}(t) &= (1-t) \sum_{j=0}^{r-1} P_{i+j} B_{j,r-1}(t) + t \sum_{j=0}^{r-1} P_{i+j+1} B_{j,r-1}(t) \\ &= (1-t) \sum_{j=i}^{i+r-1} P_j B_{j-i,r-1}(t) + t \sum_{j=i+1}^{i+r} P_j B_{j-i-1,r-1}(t) \end{aligned}$$

y por definición $B_{-1,r-1}(t) = B_{r,r-1}(t) = 0$ entonces

$$P_{i,r} = (1-t) \sum_{j=i}^{i+r} P_j B_{j-i,r-1}(t) + t \sum_{j=i}^{i+r} P_j B_{j-i-1,r-1}(t)$$

por último por la propiedad I

$$\begin{aligned} P_{i,r}(t) &= \sum_{j=i}^{i+r} P_j [(1-t)B_{j-i,r-1}(t) + tB_{j-i-1,r-1}(t)] \\ &= \sum_{j=i}^{i+r} P_j B_{j-i,r}(t) = \sum_{j=0}^r P_{i+j} B_{j,r}(t) \end{aligned}$$

Así finalmente, los puntos obtenidos en el último paso del algoritmo de Casteljaou, se pueden escribir como lo indica el siguiente corolario. ■

Corolario 1 Dado un entero positivo p y un conjunto de puntos de control P_0, \dots, P_p entonces

$$\mathbf{B}(t) = P_{0,p}(t) = \sum_{i=0}^p P_i B_{i,p}(t)$$

Demostración. Claramente la igual es cierta, si en (2.4) $i = 0$ y $r = p$. ■

Ejemplo 12 Representemos la curva del ejemplo 11 usando ahora la base de Bernstein. Así según el teorema anterior esta curva se puede escribir como

$$\mathbf{B}(t) = \sum_{i=0}^3 B_{i,3}(t)P_i$$

en donde

$$\begin{aligned} B_{0,3}(t) &= (1-t)^3 \\ B_{1,3}(t) &= 3t(1-t)^2 \\ B_{2,3}(t) &= 3t^2(1-t) \\ B_{3,3}(t) &= t^3 \end{aligned}$$

estas funciones se presentan en la Figura 2.3

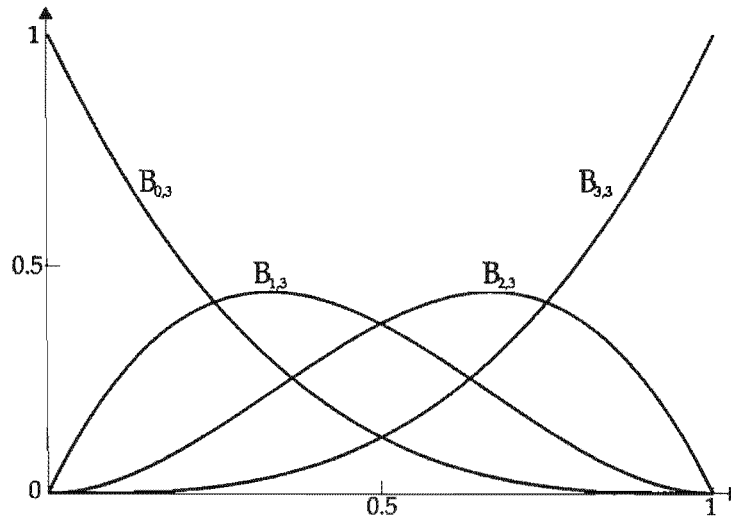


Figura 2.3: Polinomios Bernstein $B_{i,3}$ para $i=0,2,3$.

□

2.1.1. Propiedades.

A continuación describiremos algunas de las principales propiedades de la curva de Bézier integral, para esto supondremos dado un entero positivo p y puntos de control P_0, \dots, P_p que definen a $\mathbf{B}(t)$ con $t \in [0, 1]$.

1) Interpolación de los Puntos Frontera.

De las propiedades III y partición de la unidad los polinomios de Bernstein, tenemos que $B_{j,p}(0) =$

$B_{k,p}(1) = 0$ para $0 < j \leq p$ y $0 \leq k < p$ así

$$\mathbf{B}(0) = \sum_{i=0}^p B_{i,p}(0)P_i = B_{0,p}(0)P_0 = P_0$$

$$\mathbf{B}(1) = \sum_{i=0}^p B_{i,p}(1)P_i = B_{p,p}(1)P_p = P_p$$

Esta propiedad es bastante útil ya que proporciona un mayor control en la curva.

- ii) **Envoltura Convexa.** La curva está contenida en la envoltura convexa de los puntos de control. Sea $t^* \in [0, 1]$, entonces por la no negatividad y partición de la unidad de los polinomios de Bernstein

$$\sum_{i=0}^p B_{i,p}(t^*) = 1$$

$$B_{i,p}(t^*) \geq 0$$

de donde $\mathbf{B}(t^*) = \sum B_{i,p}(t^*)P_i$, pertenece a la envoltura convexa de los puntos P_0, \dots, P_p .

- iii) **Invarianza Afín.** Esta propiedad es muy importante ya que nos indica que es posible aplicar ciertas transformaciones geométricas a $\mathbf{B}(t)$, con sólo aplicarlas a los puntos de control, para describirla necesitaremos la siguiente definición.

Definición 10 Dado un conjunto de reales mayores o iguales que cero $\alpha_0, \dots, \alpha_p$ tales que $\sum_{i=0}^p \alpha_i = 1$ un mapeo $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ es llamado afín si

$$\phi\left(\sum_{i=0}^p \alpha P_i\right) = \sum_{i=0}^p \alpha \phi(P_i)$$

Así la propiedad de la invarianza afín de la curva de Bézier nos indica que

$$\phi\left(\sum B_{i,p}P_i\right) = \sum B_{i,p}\phi(P_i)$$

lo cual es cierto por la propiedad III de los polinomios de Bernstein.

Los mapeos afines que son interesantes para las curvas de Bézier, (y en general para los NURBS) son las transformaciones geométricas básica, como translaciones, rotaciones, proyecciones etc., es decir aquellas de la forma

$$\phi(v) = Av + w \tag{2.7}$$

en donde A es una matriz de 3×3 y w es un vector. Notemos que para este caso la invarianza se puede probar como sigue

$$\begin{aligned} \phi\left(\sum B_{i,p}P_i\right) &= A\left(\sum B_{i,p}P_i\right) + w = \sum B_{i,p}AP_i + \sum B_{i,p}w \\ &= \sum B_{i,p}(AP_i + w) = \sum B_{i,p}\phi(P_i) \end{aligned}$$

- iv) **Invarianza Bajo Transformaciones de Parámetros Afines.**

Los valores parametrales de $\mathbf{B}(t)$, según la definición del algoritmo de Casteljau o los polinomios de Bernstein pertenecen al $[0, 1]$, sin embargo es posible definir a la curva sobre cualquier intervalo $[\alpha, \beta]$ (con $\alpha < \beta$ y $\alpha, \beta \in \mathbb{R}$) por medio del cambio de variable

$$t = \frac{u - \alpha}{\beta - \alpha} \quad (2.8)$$

y así

$$\sum_{i=0}^p B_{i,p}(t)P_i = \sum_{i=0}^p B_{i,p}\left(\frac{u - \alpha}{\beta - \alpha}\right)$$

el algoritmo de Casteljau para este caso, se escribe como

$$P_{i,r}(t) = \frac{\beta - u}{\beta - \alpha} P_{i,r-1}(u) + \frac{u - \alpha}{\beta - \alpha} P_{i+1,r-1}(u)$$

para $r = 0, \dots, p$ y $i = 0, \dots, p - r$.

Es importante notar que la igualdad (2.8) se puede ver como un mapeo $\gamma(t) : [\alpha, \beta] \rightarrow [0, 1]$, la cual se puede probar (ver [tesis, página 75]) también es un mapeo afín de \mathbb{R} en \mathbb{R} , y así se dice que es una transformación de parámetros afines y que $\mathbf{B}(t)$ es invariante bajo ésta.

v) **Derivada.** La derivada de $\mathbf{B}(t)$ está dada por

$$\mathbf{B}'(t) = p \sum_{i=0}^{p-1} \Delta P_i B_{i,p-1}(t) \quad (2.9)$$

en donde $\Delta P_i = P_{i+1} - P_i$.

Esto se sigue de la propiedad v de los polinomios de Bernstein ya que

$$\mathbf{B}'(t) = \sum_{i=0}^p B'_i P_i = p \sum_{i=0}^p [B_{i-1,p-1} - B_{i,p-1}] P_i$$

y por definición $B_{-1,p-1}(t) = B_{p,p-1}(t) = 0$ y así

$$\begin{aligned} \mathbf{B}'(t) &= p \sum_{i=0}^p B_{i-1,p-1}(t) P_i - p \sum_{i=0}^{p-1} B_{i,p-1}(t) P_i \\ &= p \sum_{i=0}^{p-1} B_{i,p-1}(t) P_{i+1} - p \sum_{i=0}^{p-1} B_{i,p-1}(t) P_i \\ &= p \sum_{i=0}^{p-1} (P_{i+1} - P_i) B_{i,p-1}(t) = p \sum_{i=0}^{p-1} \Delta P_i B_{i,p-1}(t). \end{aligned}$$

Para la derivada de orden r (con $0 \leq r \leq p$), es necesario definir la siguiente notación

$$\Delta^r P_i = \Delta^{r-1} P_{i+1} - \Delta^{r-1} P_i$$

en donde $\Delta^0 P_i = P_i$. Por ejemplo

$$\begin{aligned}\Delta^1 P_i &= P_{i+1} - P_i \\ \Delta^2 P_i &= P_{i+2} - 2P_{i+1} + P_i \\ \Delta^3 P_i &= P_{i+3} - 3P_{i+2} + 3P_{i+1} - P_i \\ &\vdots \\ \Delta^r P_i &= P_{i+r} - \binom{r}{1} P_{i+r-1} + \binom{r}{2} P_{i+r-2} - \cdots + P_i\end{aligned}$$

que se puede escribir como

$$\Delta^r P_i = \sum_{j=0}^{r-1} \binom{r-1}{j} (-1)^j \Delta^1 P_{i+r-1-j} \quad (2.10)$$

Entonces si $\mathbf{B}^{(r)}(t)$ denota la r -ésima derivada de la curva, tenemos que

$$\mathbf{B}^{(r)}(t) = \frac{p!}{(p-r)!} \sum_{i=0}^{p-r} \Delta^r P_i B_{i,p-r}(t) \quad (2.11)$$

esta igualdad demostrará por inducción sobre r . Así si $r = 1$

$$\mathbf{B}'(t) = p \sum_{i=0}^{p-1} (P_{i+1} - P_i) B_{i,p-1}(t) = \frac{p!}{(p-1)!} \sum_{i=0}^{p-1} \Delta^1 P_i B_{i,p-1}(t)$$

Ahora supongamos que es cierto para $r - 1$, es decir

$$\frac{d^{r-1}}{dt^{r-1}} \mathbf{B}_p(t) = \frac{p!}{(p-r+1)!} \sum_{i=0}^{p-r+1} \Delta^{r-1} P_i B_{i,p-r+1}(t)$$

y probemos para r . Al derivar el lado derecho de esta última expresión resulta que

$$\begin{aligned}& \frac{p!}{(p-r+1)!} (p-r+1) \sum_{i=0}^{p-r+1} \Delta^{r-1} P_i [B_{i-1,p-r}(t) - B_{i,p-r}(t)] = \\ & \frac{p!}{(p-r)!} \left[\Delta^{r-1} P_i B_{i-1,p-r}(t) - \sum_{i=0}^{p-r} \Delta^{r-1} P_i B_{i,p-r}(t) \right] = \\ & \frac{p!}{(p-r)!} \left[\sum_{i=0}^{p-r} \Delta^{r-1} P_{i+1} B_{i,p-r}(t) - \sum_{i=0}^{p-r} \Delta^{r-1} P_i B_{i,p-r}(t) \right] = \\ & \frac{p!}{(p-r)!} \left[\sum_{i=0}^{p-r} (\Delta^{r-1} P_{i+1} - \Delta^{r-1} P_i) B_{i,p-r}(t) \right] = \\ & \frac{p!}{(p-r)!} \sum_{i=0}^{p-r} \Delta^r P_i B_{i,p-r}(t)\end{aligned}$$

2.1.2. Subdivisión y diferenciabilidad.

Según lo discutido anteriormente, el cálculo de un punto sobre la curva de Bézier integral usando el algoritmo de Casteljau, requiere de una serie de pasos que generan puntos intermedios. A continuación describiremos como algunos de éstos, pueden verse como los puntos de control de dos subcurvas que subdividen a la original y la diferenciabilidad que existe en su unión.

Ejemplo 13 Sean los puntos de control P_0, P_1, P_2 y P_3 del ejemplo 11 y fijemos a $t = \frac{1}{2}$. Entonces la respectiva curva de Bézier integral $\mathbf{B}(t)$ junto con los puntos obtenidos por el algoritmo de Casteljau, (para este parámetro) se presentan en la Figura 2.4. Notemos que $\mathbf{B}(t)$ puede verse como si estuviese formada por las curvas $\mathbf{B}_0(t)$ y $\mathbf{B}_1(t)$ para la cuales su rango y puntos de control son respectivamente $t \in [0, \frac{1}{2}]$, $t \in [\frac{1}{2}, 1]$, Q_0, \dots, Q_3 y Q_3, \dots, Q_5 . Además se observa que estos puntos cumplen que

$$\begin{aligned} Q_i &= P_{i,0}(1/2) \\ Q_{3+i} &= P_{i,3-i}(1/2) \end{aligned}$$

para $i = 0, \dots, 3$.

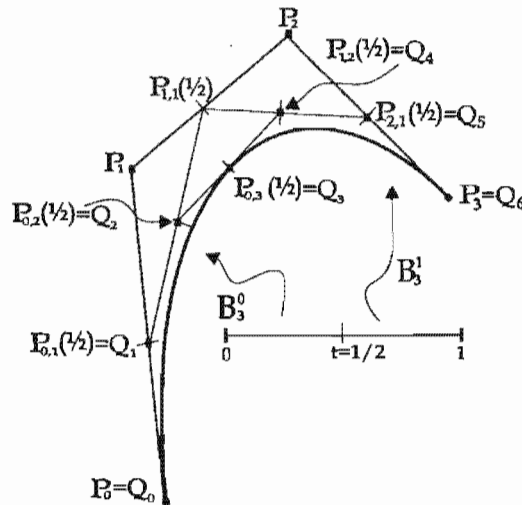


Figura 2.4: Curva de Bézier integral subdividida por $\mathbf{B}_0(t)$ con $t \in [0, \frac{1}{2}]$ y $\mathbf{B}_1(t)$ con $t \in [\frac{1}{2}, 1]$.

□

Así, ya que el algoritmo de Casteljau es un proceso recursivo, las observaciones hechas a la curva del ejemplo anterior se puede “extrapolar” a una curva $\mathbf{B}(t)$ de grado p con puntos P_0, \dots, P_p . Es decir si $t^* \in (0, 1)$, $\mathbf{B}(t)$ está subdividida por $\mathbf{B}_0(t)$ y $\mathbf{B}_1(t)$ cuyos dominios y puntos de control son respectivamente $[0, t^*]$, $[t^*, 1]$, Q_0, \dots, Q_p y Q_p, \dots, Q_{2p} los cuales cumplen

$$\begin{aligned} Q_i &= P_{0,i}(t^*) \\ Q_{p+i} &= P_{i,p-i}(t^*) \end{aligned} \tag{2.12}$$

para $i = 0, \dots, p$.

Por otro lado notemos que la unión de las curvas cúbicas $\mathbf{B}_0(t)$ y $\mathbf{B}_1(t)$ del ejemplo se realiza en el punto Q_3 y en general según (2.12) en Q_p cuando $t = t^*$. Entonces cabe la pregunta acerca de cómo es,

2.1. CURVA DE BÉZIER INTEGRAL.

o que tan suave es esta unión. Para responder a esto, es necesario entender las relaciones geométricas entre los puntos $Q_0, \dots, Q_{2p}, P_0, \dots, P_p$ y sus respectivos polígonos de control.

Tomemos el caso particular del ejemplo anterior y observemos la Figura 2.5. Ya que Q_2, Q_3 y Q_4 son colineales y la distancia entre Q_2, Q_3 y Q_3, Q_4 es proporcional a la distancia que hay entre 0 a t y t a 1 respectivamente, la interpolación lineal (que denotaremos por $Q_{2,1}(u)$) de los puntos Q_2 y Q_3 dada por

$$Q_{2,1}(u) = (1 - 2u)Q_2 + 2uQ_3 \quad -\infty < u < \infty$$

cumple que $Q_{2,1}(1) = Q_4$.

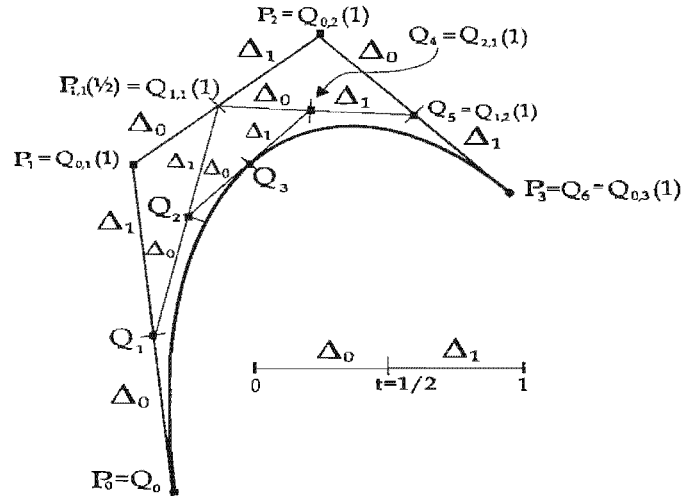


Figura 2.5: Curva de Bézier subdividida en donde se muestra las distancias proporcionales del polígonos de control y el rango parametral.

Esto mismo sucede con las interpolaciones lineales $Q_{1,1}(u)$ y $Q_{1,2}(u)$ de los puntos Q_1, Q_2 y $Q_{1,1}(1), Q_{2,1}(1)$ dadas por

$$\begin{aligned} Q_{1,1}(u) &= Q_1(1 - 2u) + Q_2 2u & -\infty < u < \infty \\ Q_{1,2}(u) &= Q_{1,1}(1)(1 - 2u) + Q_{2,1}(1) 2u & -\infty < u < \infty \end{aligned}$$

es decir

$$\begin{aligned} Q_{1,1}(1) &= P_{1,1}\left(\frac{1}{2}\right) \\ Q_{1,2}(1) &= Q_5 \end{aligned}$$

y por último la interpolación lineal de $Q_{0,2}(1)$ y $Q_{1,2}(1)$ dada por

$$Q_{0,3}(u) = (1 - 2u)Q_{0,2}(1) + 2uQ_{1,2}(1) \quad -\infty < u < \infty$$

cumple

$$Q_{0,3}(1) = Q_6.$$

Es decir tenemos que

$$\begin{aligned} Q_{2,1}(1) &= Q_4 \\ Q_{1,2}(1) &= Q_5 \\ Q_{0,3}(1) &= Q_6 \end{aligned}$$

En la siguiente proposición generalizamos este resultado.

Proposición 2 *Sea un entero positivo p y un conjunto de puntos de control P_0, \dots, P_p de una curva de Bézier integral $\mathbf{B}(t)$. Si $t^* \in (0, 1)$ y $\mathbf{B}_0(t)$ (para $t \in [0, t^*]$) y $\mathbf{B}_1(t)$ (para $t \in [t^*, 1]$) son las curvas de Bézier, que subdividen a $\mathbf{B}(t)$, las cuales tiene puntos de control Q_0, \dots, Q_p y Q_p, \dots, Q_{2p} respectivamente, que (2.12), entonces*

$$Q_{i+p} = Q_{p-i,i}(1) \quad (2.13)$$

para $i = 0, \dots, p$ y $Q_{i,0} = Q_i$.

Demostración

La demostración se hará por inducción sobre i . Así supongamos que $i = 1$. Es decir debemos demostrar que $Q_{p+1} = Q_{p-1,1}(1)$, sin embargo por (2.12) tenemos que $Q_{p+1} = P_{1,p-1}(t^*)$, por lo que es equivalente probar que $P_{1,p-1}(t^*) = Q_{p-1,1}(1)$.

Entonces por el algoritmo de Casteljau y la propiedad IV de las curvas de Bézier integrales tenemos que

$$Q_{p-1,1}(t) = \left(1 - \frac{t}{t^*}\right) Q_{p-1}(t) + \frac{t}{t^*} Q_p(t)$$

y de nuevo por (2.12)

$$Q_{p-1,1}(t) = \left(1 - \frac{t}{t^*}\right) P_{0,p-1}(t^*) + \frac{t}{t^*} P_{0,p}(t^*)$$

así

$$\begin{aligned} Q_{p-1,1}(1) &= \left(1 - \frac{1}{t^*}\right) P_{0,p-1}(t^*) + \frac{1}{t^*} P_{0,p}(t^*) \\ &= \left(1 - \frac{1}{t^*}\right) P_{0,p-1}(t^*) + \frac{1}{t^*} [(1 - t^*) P_{0,p-1}(t^*) + t^* P_{1,p-1}(t^*)] \\ &= P_{1,p-1}(t^*). \end{aligned}$$

Ahora supongamos cierto para $i - 1$ es decir

$$P_{i-1,p-i+1}(t^*) = Q_{p+i-1} = Q_{p-i+1,i-1}(1)$$

y probemos para i . Entonces como $Q_{p+i} = P_{i,p-i}(t^*)$, es equivalente probar que $P_{i,p-i}(t^*) = Q_{p-i,i}(1)$ y

$$Q_{p-i,i}(t) = \left(1 - \frac{t}{t^*}\right) Q_{p-i,i-1}(t) + \frac{t}{t^*} Q_{p-i+1,i-1}(t)$$

de donde

$$\begin{aligned} Q_{p-i,i}(1) &= \left(1 - \frac{1}{t^*}\right) Q_{p-i,i-1}(1) + \frac{1}{t^*} Q_{p-i+1,i-1}(1) \\ &= \left(1 - \frac{1}{t^*}\right) P_{i-1,p-i}(t^*) + \frac{1}{t^*} P_{i-1,p-i+1}(t^*) \\ &= \left(1 - \frac{1}{t^*}\right) P_{i-1,p-i}(t^*) + \frac{1}{t^*} [(1 - t^*) P_{i-1,p-i}(t^*) + t^* P_{i,p-i}(t^*)] \\ &= P_{i,p-i}(t^*) \end{aligned}$$

■

El siguiente teorema nos proporciona la relación entre los puntos de control Q_i y la suavidad de la unión de las subcurvas.

Teorema 15 Sea un entero positivo p y un conjunto de puntos de control P_0, \dots, P_p de una curva de Bézier $\mathbf{B}(t)$. si $t^* \in (0, 1)$ y las curvas $\mathbf{B}_0(t)$ con $t \in [0, t^*]$, y $\mathbf{B}_1(t)$ con $t \in [t^*, 1]$, y puntos de control $Q_0, \dots, Q_p, Q_p, \dots, Q_{2p}$ (respectivamente) subdividen a $\mathbf{B}(t)$, entonces $\mathbf{B}_0(t)$ y $\mathbf{B}_1(t)$ se unen en Q_p (cuando $t = t^*$) con una continuidad de clase $C^{(r)}$ con $0 \leq r \leq p-1$, siempre que

$$Q_{i+p} = Q_{p-i,i}(1) \quad (2.14)$$

para $i = 0, \dots, r$.

Demostración.

Por la propiedad IV de la curva de Bézier integral

$$\begin{aligned} \mathbf{B}_0(t) &= \sum_{i=0}^p Q_i B_{i,p}\left(\frac{t}{t^*}\right) && \text{para } t \in [0, t^*] \\ \mathbf{B}_1(t) &= \sum_{i=0}^p Q_{p+i} B_{i,p}\left(\frac{t-t^*}{1-t^*}\right) && \text{para } t \in [t^*, 1] \end{aligned}$$

ahora por la propiedad V

$$\begin{aligned} \frac{d^r}{dt^r} \mathbf{B}_0(t) &= \frac{p!}{(p-r)!} \sum_{i=0}^{p-r} \Delta^r Q_i B_{i,p-r}\left(\frac{t}{t^*}\right) \cdot \left(\frac{1}{t^*}\right) \\ \frac{d^r}{dt^r} \mathbf{B}_1(t) &= \frac{p!}{(p-r)!} \sum_{i=0}^{p-r} \Delta^r Q_{p+i} B_{i,p-r}\left(\frac{t-t^*}{1-t^*}\right) \cdot \left(\frac{1}{1-t^*}\right) \end{aligned}$$

Así

$$\begin{aligned} \frac{d^r}{dt^r} \mathbf{B}_0(t) &= \frac{p!}{(p-r)!} \Delta^r Q_{p-r} \left(\frac{1}{t^*}\right)^r \\ \frac{d^r}{dt^r} \mathbf{B}_1(t) &= \frac{p!}{(p-r)!} \Delta^r Q_p \left(\frac{1}{1-t^*}\right)^r \end{aligned}$$

de donde, la demostración se reduce a probar que²

$$Q_{i+p} = Q_{p-i,i}(1) \Rightarrow \left(\frac{1}{t^*}\right)^r \Delta^r Q_{p-r} = \left(\frac{1}{1-t^*}\right) \Delta^r Q_p \quad (2.15)$$

para $i = 0, \dots, r$.

Probaremos la expresión 2.15 por inducción sobre r . Si $r = 0$ se cumple trivialmente, si $r = 1$ debemos demostrar que

$$\left(\frac{1}{t^*}\right) \Delta^1 Q_{p-1} = \left(\frac{1}{1-t^*}\right) \Delta^1 Q_p$$

así por el algoritmo de Casteljau y la hipótesis

$$\begin{aligned} \left(\frac{1}{1-t^*}\right) \Delta^1 Q_p &= \frac{1}{1-t^*} (Q_{p+1} - Q_p) = \frac{1}{1-t^*} (Q_{p-1,1}(1) - Q_p) \\ &= \frac{1}{1-t^*} \left(\frac{t^*-1}{t^*} Q_{p-1} + \frac{1}{t^*} Q_p - Q_p\right) \\ &= \frac{1}{t^*} (Q_p - Q_{p-1}) = \left(\frac{1}{t^*}\right) \Delta^1 Q_{p-1} \end{aligned}$$

²De hecho se cumple también la implicación inversa para esto ver [Far90, página 103].

Ahora supongamos cierto para $r - 1$ es decir

$$\left(\frac{1}{t^*}\right)^{r-1} \Delta^{r-1} Q_{p-r+1} = \left(\frac{1}{1-t^*}\right)^{r-1} \Delta^{r-1} Q_p$$

y probemos para r . Entonces por hipótesis de inducción

$$\begin{aligned} \left(\frac{1}{1-t^*}\right) \Delta^r Q_p &= \left(\frac{1}{1-t^*}\right)^r (\Delta^{r-1} Q_{p+1} - \Delta^{r-1} Q_p) \\ &= \left(\frac{1}{1-t^*}\right) \left[\left(\frac{1}{1-t^*}\right)^{r-1} \Delta^{r-1} Q_{p+1} \right. \\ &\quad \left. - \left(\frac{1}{1-t^*}\right)^{r-1} \Delta^{r-1} Q_p \right] \\ &= \left(\frac{1}{1-t^*}\right) \left[\left(\frac{1}{t^*}\right)^{r-1} \Delta^{r-1} Q_{p-r+2} \right. \\ &\quad \left. - \left(\frac{1}{t^*}\right)^{r-1} \Delta^{r-1} Q_{p-r+1} \right] \\ &= \left(\frac{1}{t^*}\right) \left[\left(\frac{1}{1-t^*}\right) \Delta^{r-1} Q_{p-r+2} \right. \\ &\quad \left. - \left(\frac{1}{1-t^*}\right) \Delta^{r-1} Q_{p-r+1} \right] \end{aligned}$$

usando (2.10) la última expresión se puede escribir como

$$\begin{aligned} \left(\frac{1}{t^*}\right)^{r-1} \left[\sum_{j=0}^{r-2} \binom{r-2}{j} (-1)^j \left(\frac{1}{1-t^*}\right) \Delta^1 Q_{p-j} \right. \\ \left. - \sum_{j=0}^{r-2} \binom{j}{r-2} (-1)^j \left(\frac{1}{1-t^*}\right) \Delta^1 Q_{p-1-j} \right] \end{aligned}$$

por el caso cuando $r = 1$

$$\begin{aligned} \left(\frac{1}{t^*}\right)^{r-1} \left[\sum_{j=0}^{r-2} \binom{r-2}{j} (-1)^j \left(\frac{1}{t^*}\right) \Delta^1 Q_{p-1-j} \right. \\ \left. - \sum_{j=0}^{r-2} \binom{j}{r-2} (-1)^j \left(\frac{1}{t^*}\right) \Delta^1 Q_{p-2-j} \right] \\ = \left(\frac{1}{t^*}\right)^r \left[\sum_{j=0}^{r-2} \binom{r-2}{j} (-1)^j \Delta^1 Q_{p-1-j} \right. \\ \left. - \sum_{j=0}^{r-2} \binom{j}{r-2} (-1)^j \Delta^1 Q_{p-2-j} \right] \\ = \left(\frac{1}{t^*}\right)^r (\Delta^{r-1} Q_{p-r+1} - \Delta^{r-1} Q_{p-r}) = \left(\frac{1}{t^*}\right)^r \Delta^r Q_{p-r} \end{aligned}$$

2.1. CURVA DE BÉZIER INTEGRAL.

y por lo tanto

$$\left(\frac{1}{1-t^*}\right) \Delta^r Q_p = \left(\frac{1}{t^*}\right)^r \Delta^r Q_{p-r}$$

■

Claramente, por la propiedad de invarianza bajo transformaciones afines, la curva de Bézier $\mathbf{B}(t)$ del teorema anterior, puede ser definida sobre cualquier intervalo $[\alpha, \beta]$, en donde el respectivo t^* es un valor $u^* \in [\alpha, \beta]$ con $u^* = t^*(\beta - \alpha) + \alpha$. Así en este caso las curvas se escriben como

$$\begin{aligned} \mathbf{B}(u) &= \sum_{i=0}^p P_i B_{i,p} \left(\frac{u-\alpha}{\beta-\alpha} \right) && \text{para } u \in [\alpha, \beta] \\ \mathbf{B}_0(u) &= \sum_{i=0}^p Q_i B_{i,p} \left(\frac{u-\alpha}{u^*-\alpha} \right) && \text{para } u \in [\alpha, u^*] \\ \mathbf{B}_1(u) &= \sum_{i=0}^p Q_{p+i} B_{i,p} \left(\frac{u-u^*}{\beta-u^*} \right) && \text{para } u \in [u^*, \beta] \end{aligned}$$

y entonces $\mathbf{B}_0(u)$ y $\mathbf{B}_1(u)$ se unen en u^* con una continuidad de clase $C^{(r)}$ siempre que

$$Q_{i+p} = Q_{p-i,i}(\beta)$$

para $i = 0, \dots, r$.

Geoméricamente, el teorema nos indica que la diferenciabilidad de la unión de dos curvas de Bézier, está directamente relacionado con el hecho de que sus puntos de control tengan una **geometría definida por el algoritmo de Casteljau**.

Por ejemplo para que dos curvas de Bézier $\mathbf{B}_0(u)$ con $u \in [\alpha, \beta]$ y $\mathbf{B}_1(u)$ con $u \in [\beta, \gamma]$ y puntos de control P_0, \dots, P_p y P_p, \dots, P_{2p} , se unan con una continuidad de clase $C^{(1)}$ en P_p cuando $u = \beta$, debe suceder lo que se muestra en la Figura 2.6

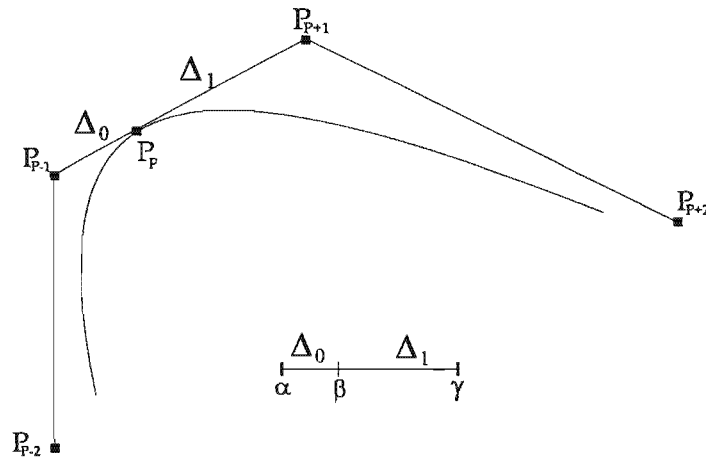


Figura 2.6: Condición de continuidad de clase $C^{(1)}$

Es decir la distancia de P_{p-1} a P_p y P_p a P_{p+1} debe ser proporcional a la distancia de α a β y β a γ .

En el caso de la continuidad de clase $C^{(2)}$ debe existir algún número $d \in \mathbb{R}^3$, de tal manera que se cumpla lo que presenta en la Figura 2.7. Para una continuidad de clase $C^{(3)}$ deben existir dos puntos d_1 y $d_3 \in \mathbb{R}^3$ y suceder lo que se observa en la Figura 2.8. Siguiendo el mismo proceso es posible determinar la geometría de los puntos de control para obtener cualquier grado de diferenciabilidad requerido³.

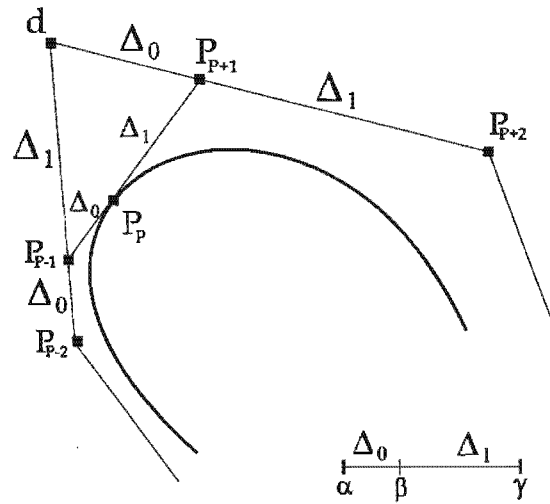


Figura 2.7: Condición de continuidad de clase $C^{(2)}$

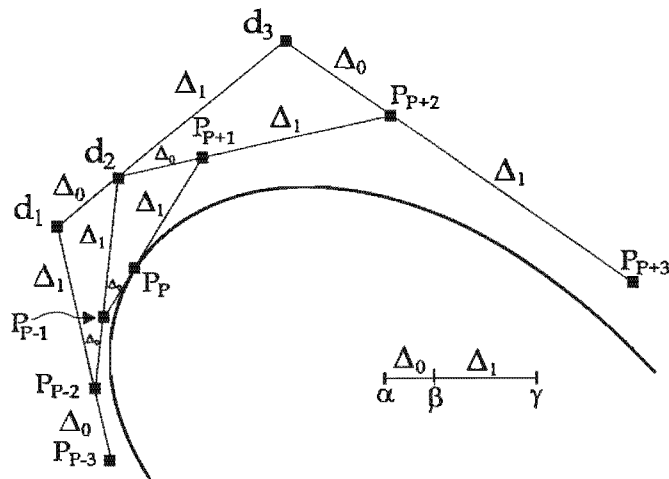


Figura 2.8: Condición de continuidad de clase $C^{(3)}$

³Por supuesto esto también depende con cuantos puntos de control se dispongan.

2.2. Curvas de Bézier racionales.

Una cónica, es una curva polinomial cuadrática definida por la ecuación

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0$$

dentro de la cuales se encuentran las bien conocidas hipérbolas, elipses y parábolas. éstas tiene una característica importante dentro de nuestro contexto llamada *irreductibilidad*, que indica que no se pueden descomponer como el producto de dos factores lineales, lo que además implica que las parametrizaciones polinomiales, deben de tener funciones coordenadas de cocientes de polinomios⁴ llamadas racionales [Marsh, pág. 98-115]. Esto quiere decir, que estas cónicas **no** pueden ser representadas por curvas polinomiales con funciones coordenadas del estilo

$$\begin{aligned} x(u) &= a_0 + a_1u + \dots + a_nu^n \\ y(u) &= b_0 + b_1u + \dots + b_nu^n \end{aligned} \quad (2.16)$$

con $a_i, b_i \in \mathbb{R}$ para toda i . Por ejemplo, supongamos que deseamos representar el círculo con centro en el origen $x^2 + y^2 = 1$ con (2.16)

Entonces, sustituyendo estas dos ecuaciones en

$$x^2 + y^2 - 1 = 0 \quad (2.17)$$

tenemos

$$\begin{aligned} 0 &= (a_0 + a_1u + \dots + a_nu^n)^2 + (b_0 + b_1u + \dots + b_nu^n)^2 - 1 \\ &= (a_0^2 + b_0^2 - 1) + 2(a_0a_1 + b_0b_1)u + (a_1^2 + 2a_0a_2 + b_1^2 + 2b_0b_2)u^2 \\ &\quad + \dots + (a_{n-1}^2 + 2a_{n-2}a_n + b_{n-1}^2 + 2b_{n-2}b_n)u^{2n-2} \\ &\quad + 2(a_na_{n-1} + b_nb_{n-1})u^{2n-1} + (a_n^2 + b_n^2)u^{2n} \end{aligned}$$

Es decir, para que las funciones coordenadas puedan representar al círculo, es necesario que esta igualdad se cumpla para toda $u \in \mathbb{R}$, lo cual es cierto si todos sus coeficientes son cero, lo que implica que $0 = a_0^2 + b_0^2 - 1$, $a_i = 0$ y $b_i = 0$ para $1 \leq i \leq n$. Esto se sigue de

1. $a_n^2 + b_n^2 = 0 \Rightarrow a_n = b_n = 0$
2. $a_{n-1}^2 + 2a_{n-2}a_n + b_{n-1}^2 + 2b_{n-2}b_n = 0$ y el paso 1 $\Rightarrow a_{n-1}^2 + b_{n-1}^2 = 0 \Rightarrow a_{n-1} = b_{n-1} = 0$.
- \vdots
- n. $a_1^2 + 2a_0a_2 + b_1^2 + 2b_0b_2 = 0$ y el paso $n-1 \Rightarrow a_1 = b_1 = 0$.

Entonces $x(u) = a_0$ y $y(u) = b_0$, las cuales claramente no satisfacen (2.17) para toda $u, a_0, b_0 \in \mathbb{R}$ y por lo tanto con (2.16) no se puede representar la circunferencia.

Ahora, ya que las curvas de Bézier integral están definidas en términos de los polinomios de Bernstein, se deduce que éstas excluyen la posibilidad de representar las hipérbolas y elipses, así surge la necesidad de definir la curvas de Bézier racional.

⁴Excepto las parábolas.

Definición 11 Dado un entero positivo p , puntos de control P_0, \dots, P_p en \mathbb{R}^3 y reales mayores o iguales que cero llamados pesos w_0, \dots, w_p , una curva de **Bézier racional** de grado p , está dada por

$$\mathbf{B}(t) = \frac{\sum_{i=0}^p w_i B_{i,p}(t) P_i}{\sum_{j=0}^p w_j B_{j,p}(t)} \text{ con } t \in [0, 1] \quad (2.18)$$

en donde cada función $B_{i,p}(t)$ es un polinomio de Bernstein definido en (2.2), en el caso de que algún $w_i = 0$, el punto $w_i P_i$ es remplazado por P_i . Sin embargo para esta definición supondremos que al menos hay un peso distinto de cero.

Ahora si denotamos

$$R_{i,p}(t) = \frac{w_i B_{i,p}(t)}{\sum_{j=0}^p w_j B_{j,p}(t)} \quad (2.19)$$

$\mathbf{B}(t)$ se puede escribir como

$$\mathbf{B}(t) = \sum_{i=0}^n R_{i,p}(t) P_i$$

es decir, la curva se puede representar como combinación lineal de las funciones racionales $R_{i,p}(t)$.

Las curva de Bézier racional, tienen una interpretación geométrica en el contexto de Geometría Proyectiva [Marsh, pág. 19-27] ⁵ muy útil, con la que $\mathbf{B}(t)$ puede ser representada en términos de una curva integral.

Para esto definamos las coordenada homogénea de un punto en el espacio $P_i = (x_i, y_i, z_i)$ como

$$P_i^w = \begin{cases} (w_i x_i, w_i y_i, w_i z_i, w_i) & \text{si } w_i \neq 0 \\ (x_i, y_i, z_i, 0) & \text{si } w_i = 0 \end{cases} \quad (2.20)$$

y la proyección $H : \mathbb{R}^4 \rightarrow \mathbb{R}^3$ como

$$H(x, y, z, w) = \begin{cases} \left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w} \right) & \text{si } w \neq 0 \\ \text{dirección de } (x, y, z) & \text{si } w = 0 \end{cases} \quad (2.21)$$

en donde el caso **dirección de** (x, y, z) , se refiere al concepto de **punto al infinito**, que nos indica que éste representa un único punto en dirección de (x, y, z) .

Por entender mejor esto, consideremos la línea en el espacio $(x(t), y(t), z(t)) = (tx + a, ty + b, tz + c)$ que pasa a través de (a, b, c) en dirección de (x, y, z) , la cual tienen una representación en coordenadas homogéneas $(tx + a, ty + b, tz + c, w')$ para algún $w' \in \mathbb{R} \neq 0$, y multipliquemosla por $\frac{1}{t}$ (con $t \neq 0$) resultando $(x + \frac{a}{t}, y + \frac{b}{t}, z + \frac{c}{t}, \frac{w'}{t})$. Entonces si ahora tomamos el límite cuando t tiende a infinito tenemos

$$\lim_{t \rightarrow \infty} \left(x + \frac{a}{t}, y + \frac{b}{t}, z + \frac{c}{t}, \frac{w'}{t} \right) = (x, y, z, 0)$$

De donde resulta natural la interpretación anterior.

Con estos elementos notemos que la curva (2.18), se puede escribir en la siguiente manera

$$\mathbf{B}^w(t) = \sum_{i=0}^p B_{i,p}(t) P_i^w$$

la cual, es la expresión que representa a una curva de Bézier integral, que por lo general es más simple de manipular⁶, además de que la mayoría de los algoritmos para curvas de Bézier están diseñados para

⁵la cual para los alcances de este trabajo no necesitaremos abordar.

⁶En términos computacionales esto es muy claro, ya que las curvas de Bézier racionales involucran el cálculo de cocientes lo que generalmente causa pérdida de precisión.

curvas integrales.⁷ Así una vez realizada alguna transformación o cálculo a $\mathbf{B}^w(t)$, es posible aplicar la proyección H al plano $w = 1$ y obtener nuevamente la versión racional (ver Figura 2.9).

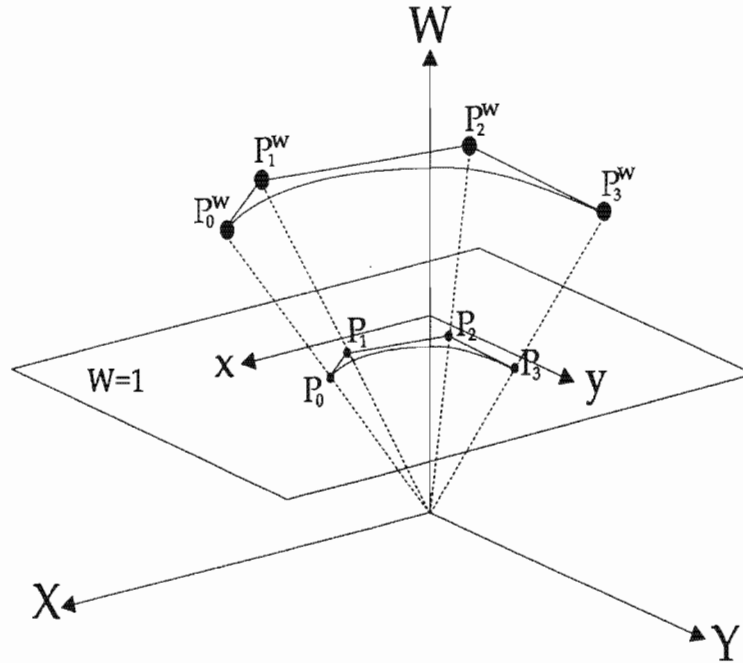


Figura 2.9: Coordenadas homogéneas, el plano proyectivo y la curva B-spline racional.

2.2.1. Propiedades.

A continuación presentaremos algunas de las principales propiedades de las funciones $R_{i,p}(t)$ y la curva de Bézier racional $\mathbf{B}(t)$.

- i) **No negatividad;** $R_{i,p}(t) \geq 0 \forall i, p$ y $t \in [0, 1]$.

Esto se sigue del hecho de que $B_{i,p}(t) \geq 0$ y que cada $w_i \geq 0$.

- ii) **Partición de la unidad.** $\sum_{i=0}^p R_{i,p}(t) = 1$.

Notemos que

$$\sum_{i=0}^p R_{i,p}(t) = \frac{\sum_{i=0}^p w_i B_{i,p}(t)}{\sum_{j=0}^p w_j B_{j,p}(t)} = 1$$

- iii) $R_{0,p}(0) = R_{p,p}(1) = 1$.

$$R_{0,p}(0) = \frac{w_0 B_{0,p}(0)}{\sum_{j=0}^p w_j B_{j,p}(0)} = \frac{w_0 B_{0,p}(0)}{w_0 B_{0,p}(0)} = 1$$

$$R_{p,p}(1) = \frac{w_p B_{p,p}(1)}{\sum_{j=0}^p w_j B_{j,p}(1)} = \frac{w_p B_{p,p}(1)}{w_p B_{p,p}(1)} = 1$$

⁷ Lo que no quiere decir que siempre que se trabaja con curvas racionales se tenga que realizar esta transformación.

iv) Si $w_i = c$ para c constante, entonces $R_{i,p}(t) = B_{i,p}(t)$.

Esto se tiene por la propiedad de la partición de la unidad de los polinomios de Bernstein, ya que así

$$R_{i,p}(t) = \frac{cB_{i,p}(t)}{c \sum_{j=0}^p B_{j,p}(t)} = \frac{cB_{i,p}}{c} = B_{i,p}(t)$$

v) **Interpolación de los puntos frontera.**

Se sigue de III.

vi) **Envoltura convexa.** $\mathbf{B}(t)$ está contenida en la envoltura convexa de sus puntos de control. Esto se tiene por I y II.

vii) **Invarianza Afin.** $\mathbf{B}(t)$ es invariante bajo transformaciones afines. Se sigue de II.

viii) Una curva de Bézier integral es un caso particular de la curva de Bézier racional.

Se sigue de IV.

ix) **Derivada.**

Para obtener la derivada de la curva de Bézier racional notemos que si $F(t) = f(t)/g(t)$

$$F'(t) = \frac{g(t)f'(t) - g'(t)f(t)}{g^2(t)} = \frac{f'(t) - g'(t)F(t)}{g(t)}$$

Ahora por la regla de Leibniz la r -ésima derivada de $f(t) = g(t)F(t)$

$$\begin{aligned} f^{(r)} &= \sum_{i=0}^r \binom{r}{i} g^{(i)}(t) F^{(r-i)}(t) \\ &= g(t)F^{(r)}(t) + \sum_{i=1}^r \binom{r}{i} g^{(i)}(t) F^{(r-i)}(t) \end{aligned}$$

lo que implica que

$$F^{(r)} = \frac{f^{(r)}(t) - \sum_{i=1}^r \binom{r}{i} g^{(i)}(t) F^{(r-i)}(t)}{g(t)} \quad (2.22)$$

Ahora si tomamos $f(t) = \sum_{i=0}^p w_i B_{i,p}(t) P_i$ y $g(t) = \sum_{j=0}^p w_j B_{j,p}(t)$, es posible calcular la r -ésima derivada de $\mathbf{B}(t)$ usando (2.22), por ejemplo si $r = 1$ y $r = 2$

$$\begin{aligned} \mathbf{B}'(t) &= \frac{(\sum_{i=0}^p w_i B_{i,p}(t) P_i)' - (\sum_{j=0}^p w_j B_{j,p}(t))' \mathbf{B}(t)}{\sum_{j=0}^p w_j B_{j,p}(t)} \\ \mathbf{B}''(t) &= \frac{(\sum_{i=0}^p w_i B_{i,p}(t) P_i)'' - 2(\sum_{i=0}^p w_j B_{j,p}(t))' \mathbf{B}'(t) - (\sum_{j=0}^p w_j B_{j,p}(t))'' \mathbf{B}(t)}{\sum_{j=0}^p w_j B_{j,p}(t)} \end{aligned}$$

aquí $B'_{i,p}$ está dada por (2.9) y (2.11).

2.2.2. La curva de Bézier cuadrática racional y cónicas.

Al inicio de esta sección, se mencionó que la principal razón para definir la curva de Bézier racional, es que ésta, a diferencia de la curva integral, permite la representación las cónicas. Claramente, la manera más natural de hacerlo es utilizar la curva racional cuadrática, dada por

$$\begin{aligned} \mathbf{B}(t) &= \frac{w_0(1-t)^2 P_0 + 2w_1 t(1-t) P_1 + w_2 t^2 P_2}{w_0(1-t)^2 + 2w_1 t(1-t) + w_2 t^2} \\ &= R_{0,2}(t) P_0 + R_{1,2}(t) P_1 + R_{2,2}(t) P_2 \end{aligned} \quad (2.23)$$

A continuación describiremos ([Lee87]), como probar que esta ecuación, representa una cónica y una manera de clasificarla.

Supongamos que los puntos de control P_0, P_1, P_2 de $\mathbf{B}(t)$, no son colineales y definamos $S = P_0 - P_1$ y $T = P_2 - P_1$, así los vectores S y T son linealmente independientes y podemos considerar un sistema coordenado oblicuo definido por $\{P_1, S, T\}$, en donde P_1 es el centro y S y T los ejes. Entonces por la propiedad de la envoltura convexa de las curvas de Bézier racional, tenemos que para cada $t \in [0, 1]$, existen escalares $\alpha(t)$ y $\beta(t)$ tales que

$$\begin{aligned} \mathbf{B}(t) &= P_1 + \alpha(t)S + \beta(t)T \\ &= \alpha(t)P_0 + (1 - \alpha(t) - \beta(t))P_1 + \beta(t)P_2 \end{aligned} \quad (2.24)$$

Ahora comparando esta ecuación con (2.23) se deduce que

$$\begin{aligned} \alpha(t) &= R_{0,2}(t) \\ \beta(t) &= R_{2,2}(t) \\ (1 - \alpha(t) - \beta(t)) &= R_{1,2}(t) \end{aligned}$$

además, notando que

$$B_{0,2}(t)B_{2,2}(t) = \left(\frac{B_{1,2}(t)}{2} \right)^2$$

tenemos

$$\begin{aligned} \alpha(t)\beta(t) &= R_{0,2}(t)R_{2,2}(t) = \frac{w_0 w_2 B_{0,2}(t)B_{2,2}(t)}{w^2(t)} \\ &= \frac{w_0 w_2 B_{1,2}^2(t)}{4w^2(t)} = \frac{w_0 w_2 (w_1 B_{1,2})^2}{4w_1^2 w^2(t)} \\ &= \frac{w_0 w_2}{4w_1^2} R_{1,2}^2(t) \end{aligned}$$

es decir

$$\alpha(t)\beta(t) = k(1 - \alpha(t) - \beta(t))^2 \quad (2.25)$$

donde

$$k = \frac{w_0 w_2}{4w_1^2} \quad (2.26)$$

Esto nos indica, que las coordenadas oblicuas de todo punto de (2.24), satisfacen la ecuación cuadrática (2.25), por lo que representan un cónica.

Si definimos el segmento de curva $\tilde{\mathbf{B}}(t)$ como

$$\tilde{\mathbf{B}}(t) = \frac{w_0(1-t)^2 P_0 - 2w_1(1-t)t P_1 + w_2 t^2 P_2}{\tilde{w}(t)} \quad (2.27)$$

con

$$\tilde{w}(t) = w_0(1-t)^2 - 2w_1(1-t)t + w_2 t^2$$

tenemos que

$$\begin{aligned} \tilde{\alpha}(t) &= \frac{w_0 B_{0,2}(t)}{\tilde{w}(t)} \\ \tilde{\beta}(t) &= \frac{w_2 B_{2,2}(t)}{\tilde{w}(t)} \\ 1 - \tilde{\alpha}(t) - \tilde{\beta}(t) &= \frac{-w_1 B_{1,2}(t)}{\tilde{w}(t)} \end{aligned}$$

y por lo tanto, $\tilde{\alpha}(t)\tilde{\beta}(t) = k(1 - \tilde{\alpha}(t) - \tilde{\beta}(t))^2$, es decir, $\tilde{\mathbf{B}}(t)$ también es una cónica.

Ahora notemos que por definición

$$\tilde{w}(t)\tilde{\mathbf{B}}(t) - w(t)\mathbf{B}(t) = -2w_1 P_1 = (\tilde{w}(t) - w(t))P_1$$

entonces

$$\tilde{w}(t)\tilde{\mathbf{B}}(t) - \tilde{w}(t)P_1 = w(t)\mathbf{B}(t) - w(t)P_1$$

y suponiendo que $\tilde{w}(t) \neq 0$ tenemos que

$$\tilde{\mathbf{B}}(t) - P_1 = \frac{w(t)}{\tilde{w}(t)}(\mathbf{B}(t) - P_1)$$

esto nos indica que los puntos P_1 , $\mathbf{B}(t)$ y $\tilde{\mathbf{B}}(t)$ son colineales, y así $\tilde{\mathbf{B}}(t)$ describe un segmento de curva complementaria a $\mathbf{B}(t)$, la cual nos permitirá determinar el tipo de cónica que representa. Para esto, notemos que el denominador de (2.27) se puede descomponer como

$$\tilde{w}(t) = (w_0 + 2w_1 + w_2)t^2 - 2(w_0 + w_1)t + w_0$$

de donde sus raíces están dadas por

$$t_i = \frac{w_0 + w_1 \pm w_1 \sqrt{1 - 4k}}{w_0 + 2w_1 + w_2} \quad i = 1, 2.$$

ahora de (2.26), sabemos que w_0 y w_2 pueden tener cualquier valor mayor o igual que cero y que $w_1 \neq 0$, entonces si tomamos $w_0 = w_2 = 1$ (lo cual es llamado *parametrización normal*) podemos determinar la clasificación de la cónica (2.23) considerando lo siguiente

- Si $4k > 1$, $\tilde{w}(t)$ no tiene raíces reales y por lo tanto $\tilde{\mathbf{B}}(t)$ no tiene puntos en el infinito, de donde está acotada para toda $t \in [0, 1]$ y se deduce que $\mathbf{B}(t)$ es una elipse (ver Figura 2.10 (a)).
- Si $4k = 1$, $\tilde{w}(t)$ tiene una raíz real y por (2.26), $w_1 = 1$ o $w_1 = -1$, sin embargo si $w_1 = -1$, tenemos que $\tilde{w}(t) = 1$, la cual es una función que no tiene raíces reales. Así debe suceder que $w_1 = 1$ y entonces $\tilde{w}(t) = (2t - 1)^2 \geq 0$ para toda t , por lo que $\tilde{\mathbf{B}}(t)$ está contenido en el primer cuadrante del sistema oblicuo $\{P_1, S, T\}$ y no está acotado. Notemos que cuando t se aproxima a t_1 por la izquierda o por la derecha, $\tilde{\mathbf{B}}(t)$ crece infinitamente. Esto nos indica que en este caso $\mathbf{B}(t)$ representa una parábola (ver Figura 2.10 (b)).

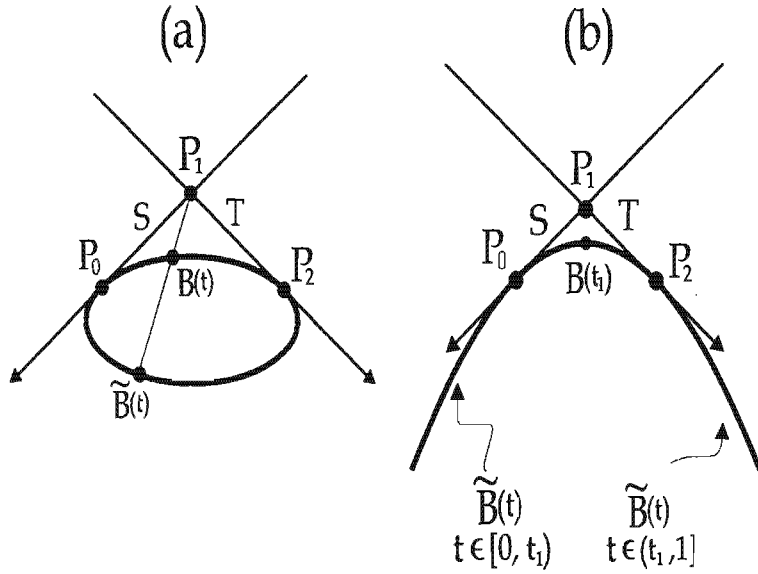


Figura 2.10: Elipse y parábola obtenidas con el segmento complementario $\tilde{\mathbf{B}}(t)$, de la curva de Bézier racional.

- Finalmente cuando $4k < 1$, $\tilde{w}(t)$ tiene raíces distintas $t_1 < t_2$. Así a medida de que t se aproxima por la izquierda a t_1 o a 1, $\tilde{\mathbf{B}}(t)$ crece infinitamente en el cuadrante ++. Cuando $t_1 < t < t_2$, $\tilde{\mathbf{B}}(t)$ pertenece al tercer cuadrante (- -), de tal manera que cuando t se aproxima a t_1 o a t_2 , esta curva crece infinitamente. Es decir tenemos una hipérbola (ver Figura 2.11).

Esta clasificación se puede escribir en términos del peso w_1 de la siguiente manera

- Si $4k > 1$, $\Rightarrow w_0 w_1 > w_1^2 \Rightarrow w_1^2 < 1 \Rightarrow 0 < w_1 < 1$, por lo que en este caso tenemos una elipse.
- Si $4k = 1$, $\Rightarrow w_1^2 \Rightarrow w_1 = 1$, tenemos una parábola.
- Si $4k < 1$, $\Rightarrow w_1^2 \Rightarrow w_1 > 1$, tenemos una hipérbola.

Es importante notar que cuando $w_1 = 0$, tenemos que

$$\mathbf{B}(t) = \frac{(1-t)^2 P_0 + t^2 P_2}{(1-t)^2 + t^2}$$

en donde $\mathbf{B}(0) = P_0$ y $\mathbf{B}(1) = P_2$, es decir esta curva resulta en una recta que une sus puntos de control.

El *shoulder point*

Otra herramienta útil para determinar el tipo de cónica que forma (2.23), es con el llamado “*shoulder point*”, el cual se deduce de la siguiente manera.

Sea $t^* = \frac{1}{2}$ (no necesariamente tiene que ser este valor) y denotemos $S = \mathbf{B}(t^*)$, entonces continuando con la suposición de una parametrización normal ($w_0 = w_2 = 1$), de la ecuación (2.23) tenemos que

$$\mathbf{B}(t^*) = \frac{1}{1+w_1} \frac{P_0 + P_1}{2} + \frac{w_1}{1+w_1} P_1 = \frac{M}{1+w_1} + \frac{w_1}{1+w_1} P_1 \quad (2.28)$$

la cual se puede escribir como

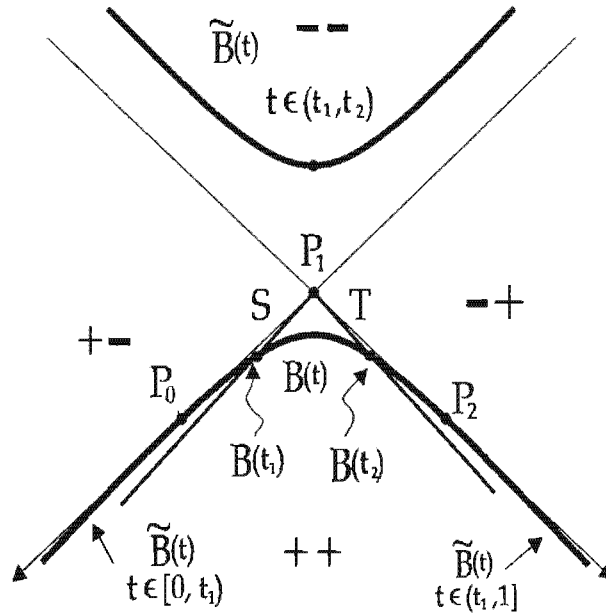


Figura 2.11: Hipérbola obtenida con el segmento complementario $\tilde{B}(t)$, de la curva de Bézier racional.

$$S = (1 - s)M + sP_1 \tag{2.29}$$

con

$$s = \frac{w_1}{1 + w_1} \tag{2.30}$$

Entonces, basados en la clasificación usando el peso w_1 , podemos establecer que

- Si $s = 0$, la curva (2.27) representa un segmento de recta.
- Si $0 < s < \frac{1}{2}$, entonces tenemos una elipse.
- Si $s = \frac{1}{2}$, una parábola.
- Y finalmente si $\frac{1}{2} < s < 1$, una hipérbola.

en la Figura 2.12, se muestran ejemplos de curvas resultantes al variar este valor.

El arco circular.

Ya que el círculo es un caso particular de una elipse, debe existir algún valor de w_1 , con el que podamos representar un arco circular. Para esto, notemos que por simetría, es claro que para lograr que la curva Bézier racional pueda representar este arco, debe suceder que su polígono de control forme un triángulo isósceles, así, supongamos que los puntos P_0, P_1 y P_2 tienen esta forma.

Ahora, si denotamos el centro del círculo que contiene al arco (ver figura 2.13) como O , r el radio, M el punto medio de la cuerda P_0P_2 , X la intersección del círculo con el segmento MP_1 y 2α el ángulo en el punto P_1 . Se puede deducir que el valor de w_1 debe cumplir que

$$\frac{|MX|}{|MP_1|} = \frac{w_1}{1 + w_1}$$

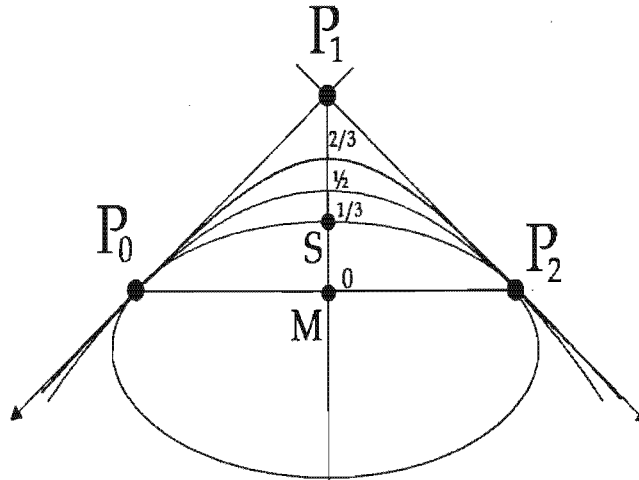


Figura 2.12: Cónicas para distintos valores del parámetro s . Si $s = \frac{1}{2}$ tenemos una elipse, si $s = \frac{1}{3}$ una parábola y si $s = \frac{2}{3}$ una hipérbola.

Entonces, notando que del triángulo OMP_0 se deduce que $|OM| = r \sin(\alpha)$ entonces

$$|MX| = |OX| - |OM| = r - r \sin(\alpha) = r(1 - \sin(\alpha))$$

además, del triángulo OP_0P_1 se deduce que $|OP_1| = \frac{r}{\sin(\alpha)}$ y así MP_1 está dado por

$$|MP_1| = |OP_1| - |OM| = \frac{r}{\sin(\alpha)} - r \sin(\alpha) = \frac{r(1 - \sin^2(\alpha))}{\sin(\alpha)}$$

con lo que finalmente

$$\frac{w_1}{1 + w_1} = \frac{|MX|}{|MP_1|} = \frac{\sin(\alpha)}{1 + \sin(\alpha)}$$

y $w_1 = \sin(\alpha)$.

Todo esto nos indica que para obtener un arco circular debemos hacer que w_1 tenga el valor del seno de la mitad del ángulo de la arista del punto de control P_1 . Por ejemplo para un arco de 60° debemos hacer que $w_1 = \sin(30^\circ) = \frac{1}{2}$, para un arco de 90° $w_1 = \sin(45^\circ) = \frac{\sqrt{2}}{2} = \frac{1}{\sqrt{2}}$.

Variación del peso.

Como se puede observar en la Figura 2.12, la variación de w_1 (y a su vez la de s), modifica la curva siguiendo una recta, este comportamiento se puede aprovechar para modificar⁸ la forma de una curva, lo que nos proporciona mayor flexibilidad en el diseño libre tanto de curvas como superficies, a continuación presentaremos un teorema que formaliza esto.

Teorema 16 Dado un entero positivo p , pesos w_0, \dots, w_p y puntos de control P_0, \dots, P_p de una curva de Bézier racional $\mathbf{B}(t)$. Supongamos que algún peso w_k (con $0 \leq k \leq p$) es modificado a $w_k + \delta w_k$, para

⁸La alteración de toda la curva por medio de la variación de un sólo punto o peso no siempre es lo deseado, sobre todo cuando ya se tiene completamente diseñada la curva y se desea hacer sólo un ligero cambio, las curvas NURBS discutidas en la siguiente sección evitan esta complicación.

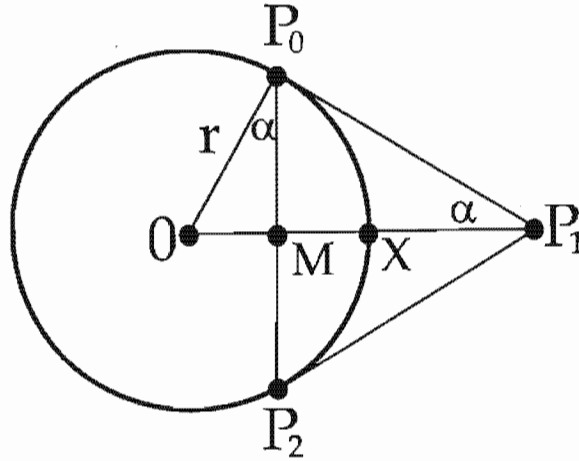


Figura 2.13: Arco circular representado por una curva de Bézier cuadrática racional.

algún real $\delta \geq 0$ entonces cualquier punto $P^* = \mathbf{B}(t^*)$ (con $t^* \in [0, 1]$), cambia a $P' = (1 - \alpha)P^* + \alpha P_k$ donde

$$\alpha = \frac{\delta w_k B_{k,p}(t^*)}{\sum_{j=0}^p w_j B_{j,p}(t^*) + \delta w_k B_{k,p}(t^*)}$$

Demostración. Primeramente notemos que

$$1 - \alpha = \frac{\sum_{j=0}^p w_j B_{j,p}(t^*)}{\sum_{j=0}^p w_j B_{j,p}(t^*) + \delta w_k B_{k,p}(t^*)}$$

Ahora por definición de la curva racional, el punto P' que resulta de modificar w_k por $w_k + \delta w_k$ está dado por

$$\begin{aligned} P' &= \frac{w_0 B_{0,p}(t^*) P_0 + \cdots + (w_k + \delta w_k) B_{k,p}(t^*) P_k + \cdots + w_p B_{p,p}(t^*) P_p}{w_0 B_{0,p}(t^*) + (w_k + \delta w_k) B_{k,p}(t^*) + \cdots + w_p B_{p,p}(t^*)} \\ &= \frac{\sum_{j=0}^p w_j B_{j,p}(t^*) P_j + \delta w_k B_{k,p}(t^*) P_k}{\sum_{j=0}^p w_j B_{j,p}(t^*) + \delta w_k B_{k,p}(t^*)} \end{aligned}$$

ahora

$$\begin{aligned} P' &= \frac{\sum_{j=0}^p w_j B_{j,p}(t^*) P_j}{\sum_{j=0}^p w_j B_{j,p}(t^*) + \delta w_k B_{k,p}(t^*)} + \frac{\delta w_k B_{k,p}(t^*) P_k}{\sum_{j=0}^p w_j B_{j,p}(t^*) + \delta w_k B_{k,p}(t^*)} P_k \\ &= \frac{\sum_{j=0}^p w_j B_{j,p}(t^*)}{\sum_{j=0}^p w_j B_{j,p}(t^*) + \delta w_k B_{k,p}(t^*)} \left(\frac{\sum_{j=0}^p w_j B_{j,p}(t^*) P_j}{\sum_{j=0}^p w_j B_{j,p}(t^*)} \right) \\ &\quad + \frac{\delta w_k B_{k,p}(t^*)}{\sum_{j=0}^p w_j B_{j,p}(t^*) + \delta w_k B_{k,p}(t^*)} P_k \\ &= (1 - \alpha) P^* + \alpha P_k \end{aligned}$$

Claramente si $\delta = 0$ entonces $\alpha = 0$ y $P' = P^*$, es decir no hay modificación alguna a la curva. Además si $w_k = 0$ el respectivo factor $B_{k,p}(t^*)$ en el numerador y denominador de (2.18) se elimina, ■

resultando una curva de grado $p - 1$ y puntos de control $P_0, \dots, P_{k-1}, P_{k+1}, \dots, P_p$. Por último si δ tiende a infinito tenemos que

$$\begin{aligned} \lim_{\delta \rightarrow \infty} \alpha &= \lim_{\delta \rightarrow \infty} \frac{\delta w_k B_{k,p}(t^*)}{\sum_{j=0}^p w_j B_{j,p}(t^*) + \delta w_k B_{k,p}(t^*)} \\ &= \lim_{\delta \rightarrow \infty} \frac{w_k B_{k,p}(t^*)}{\sum_{j=0}^p w_j B_{j,p}(t^*)/\delta + w_k B_{k,p}(t^*)} = \frac{w_k B_{k,p}(t^*)}{w_k B_{k,p}(t^*)} = 1. \end{aligned}$$

es decir $P' = P_k$.

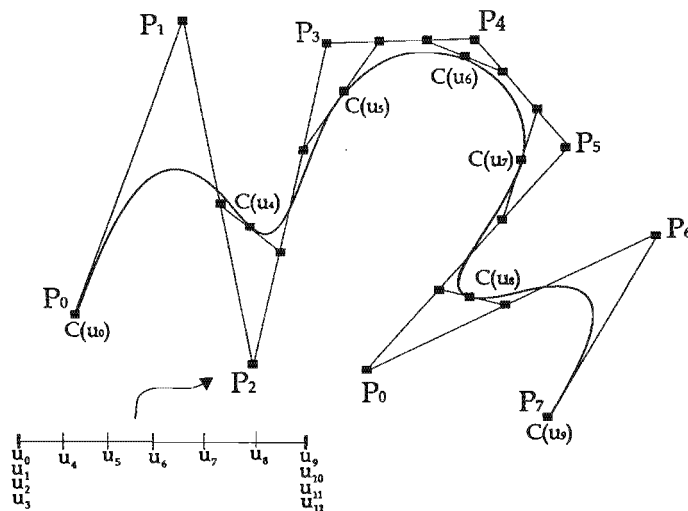


Figura 2.14: Curva B-spline cúbica.

2.3. Curvas B-spline.

Una **curva spline** de grado p , es una curva paramétrica, formada por secciones de curvas con funciones coordenadas polinomiales de a lo más grado p , que se unen con una continuidad de clase $C^{(p-1)}$. Así según lo discutido en la sección anterior, es intuitivamente claro que esta curva puede representarse por medio de secciones de curvas de Bézier definidas en cierto intervalo, uniéndose con la diferenciabilidad requerida. Por ejemplo en la Figura 2.14 se presenta una curva spline cúbica formada por 6 curvas de Bézier denotadas por $B_i(t)$ con $u \in [u_{i+3}, u_{i+4}]$ para $i = 0, \dots, 5$, que se unen con una continuidad de clase $C^{(2)}$ en los nodos u_4, \dots, u_8 .

Este enfoque no es completamente trivial, sin embargo será más evidente cuando estudiemos el algoritmo de Boor, que digamos es el equivalente para los splines al de Casteljaou para las curvas de Bézier.

En la representación analítica de una curva spline, por lo general se usan las funciones B-spline⁹, debido a su gran cantidad de propiedades, además como probaremos más adelante los polinomios de Bernstein son un caso particular de estas funciones lo que comprueba la idea geométrica anterior.

Una curva spline representada usando la base de funciones B-spline es llamada **curva B-spline** y se define como sigue.

⁹Sin embargo esto no quiere decir que se la única forma de hacerlo, por ejemplo también es posible usar la base de funciones de potencia truncada discutidas en la sección 1.2.2.

Definición 12 Dados tres enteros positivos n, m, p y puntos de control P_0, \dots, P_p , una curva B-spline de grado p está dada por

$$C(u) = \sum_{i=0}^n N_{i,p}(u) P_i \text{ para } u \in [a, b]$$

en donde cada $N_{i,p}(u)$ es una función base B-spline definida por (1.39) sobre un vector de $m + 1$ nodos estrictamente crecientes y no uniformes de la forma

$$U = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\} \quad (2.31)$$

La multiplicidad de los nodos de los extremos de U , se debe a que esto permite (como se mostrará más adelante) la interpolación de los puntos de control P_0 y P_1 , justo como una curva de Bézier, lo que proporciona un mayor control de la curva, sin embargo a diferencia de las curvas de Bézier, los B-splines permiten un control local, es decir la modificación de algún punto de control sólo afecta a ciertas secciones de ésta.

Los nodos u_i con $i = p + 1, \dots, m - p - 1$ (llamados interiores), la definición indica que éstos deben cumplir que $u_i < u_{i+1}$, sin embargo en algunos casos es necesario una desigualdad no estricta, es decir que algunos (o todos) nodos se puedan repetir, lo que por propiedades de los B-spline implica una pérdida de diferenciabilidad y estrictamente hablando para estos casos la curva no es un spline.

El término “no uniforme” de U se refiere a que no se exige cierta distancia fija entre los nodos, y por supuesto es posible definir a la curva sobre múltiples tipos de vector (para más sobre esto ver en [Rog01, páginas, 60-73]) sin embargo a menos que se indique lo contrario nosotros supondremos que el vector de nodos tiene la forma de la definición. A continuación presentaremos algunos ejemplos de curvas B-spline.

Ejemplo 14 Construyamos una curva B-spline cuadrática con 7 puntos de control. Así si tomamos el vector de nodos $U = \{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$ la respectivas funciones B-spline están dadas por

$$N_{0,2}(u) = \begin{cases} (1-u)^2 & \text{si } u \in [0, 1) \\ 0 & \text{en otro caso.} \end{cases}$$

$$N_{1,2}(u) = \begin{cases} u(1-u) + \frac{(2-u)u}{2} & \text{si } u \in [0, 1) \\ \frac{(2-u)^2}{2} & \text{si } u \in [1, 2) \\ 0 & \text{en otro caso.} \end{cases}$$

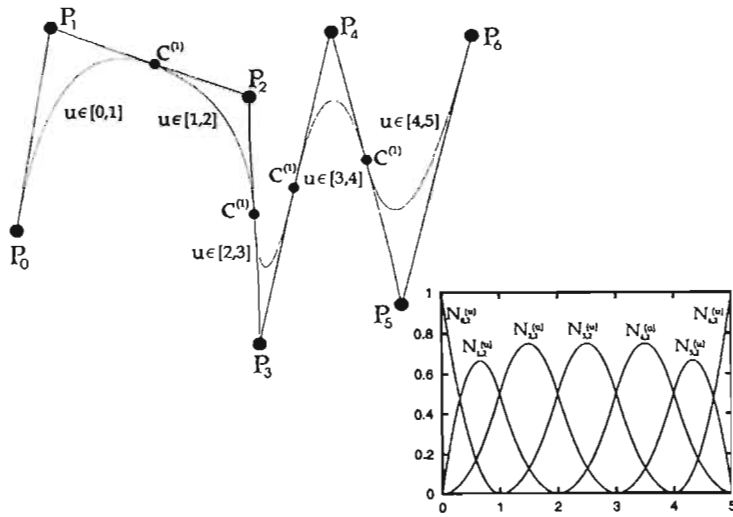


Figura 2.15: Curva B-spline cuadrática y sus funciones base.

$$N_{2,2}(u) = \begin{cases} \frac{u^2}{2} & \text{si } u \in [0, 1) \\ \frac{u(2-u)^2}{2} + \frac{(3-u)(u-1)}{2} & \text{si } u \in [1, 2) \\ \frac{(3-u)^2}{2} & \text{si } u \in [2, 3) \\ 0 & \text{en otro caso.} \end{cases}$$

$$N_{3,2}(u) = \begin{cases} \frac{(u-1)^2}{2} & \text{si } u \in [1, 2) \\ \frac{(u-1)(3-u)}{2} + \frac{(4-u)(u-2)}{2} & \text{si } u \in [2, 3) \\ \frac{(4-u)^2}{2} & \text{si } u \in [3, 4) \\ 0 & \text{en otro caso.} \end{cases}$$

$$N_{4,2}(u) = \begin{cases} \frac{(u-2)^2}{2} & \text{si } u \in [2, 3) \\ \frac{(u-2)(4-u)}{2} + \frac{(5-u)(u-3)}{2} & \text{si } u \in [3, 4) \\ \frac{(5-u)^2}{2} & \text{si } u \in [4, 5) \\ 0 & \text{en otro caso.} \end{cases}$$

$$N_{5,2}(u) = \begin{cases} \frac{(u-3)^2}{2} & \text{si } u \in [3, 4) \\ \frac{(u-3)(5-u)}{2} + (5-u)(u-4) & \text{si } u \in [4, 5) \\ 0 & \text{en otro caso.} \end{cases}$$

$$N_{6,2}(u) = \begin{cases} (u-4)^2 & \text{si } u \in [4, 5) \\ 0 & \text{en otro caso.} \end{cases}$$

□

2.3.1. Propiedades.

Claramente por definición la mayoría de las propiedades de las curvas B-spline son una implicación de las funciones base B-spline. A continuación presentamos las más importantes.

I) **Interpolación de los puntos frontera;** $\mathbf{C}(0) = P_0$ y $\mathbf{C}(1) = P_n$.

Esto es cierto ya que para el vector de nodos U tenemos que $N_{0,p}(a) = 1$, $N_{n,p}(b) = 1$ y así la partición de la unidad de los B-splines implica que

$$\begin{aligned}\mathbf{C}(a) &= \sum_{i=0}^n P_i N_{i,p}(a) = P_0 \\ \mathbf{C}(b) &= \sum_{i=0}^n P_i N_{i,p}(b) = P_n\end{aligned}$$

Para probar que $N_{0,p}(a) = 1$, demostraremos la siguiente igualdad por inducción sobre i

$$N_{p-i,i}(a) = 1 \text{ para } 0 \leq i \leq p$$

Si $i = 0$

$$N_{p,0}(u) = \begin{cases} 1 & \text{si } u \in [u_p, u_{p+1}) = [a, u_{p+1}) \\ 0 & \text{en otro caso.} \end{cases}$$

Ahora supongamos cierto para $i = p - 1$ y probemos para $i = p$ así

$$\begin{aligned}N_{0,p}(u) &= \frac{u - u_0}{u_0 - u_p} N_{0,p-1}(u) + \frac{u_{p+1} - u}{u_{p+1} - u_1} N_{1,p-1}(u) \\ &= \frac{u - a}{a - a} N_{0,p-1}(u) + \frac{b - u}{b - a} N_{1,p-1}(u) \\ &= \frac{b - u}{b - a} N_{1,p-1}(u)\end{aligned}$$

y entonces

$$N_{0,p}(a) = \frac{b - a}{b - a} N_{1,p-1}(a) = 1$$

por hipótesis de inducción. $N_{n,p}(b) = 1$ es análogo.

II) **Propiedad Fuerte de Envoltura Convexa.** La curva está contenida en la envoltura convexa de su polígono de control, además si $u \in [u_i, u_{i+1})$ para $p \leq i \leq m - p - 1$, entonces $\mathbf{C}(u)$ está contenida en la envoltura convexa de los puntos P_{i-p}, \dots, P_i .

El término "fuerte" de esta propiedad, se refiere al hecho de que la curva no sólo está contenida en la envoltura convexa de sus puntos de control (como en el caso de las curvas de Bézier racionales o no) sino que además, cada subcurva de Bézier que la forma está contenida en su propia envoltura convexa.

Esto se sigue del hecho de que si $u \in [u_i, u_{i+1})$

$$\mathbf{C}(u) = \sum_{j=0}^n N_{j,p}(u) P_j = N_{i-p,p}(u) P_{i-p} + \dots + N_{i,p}(u) P_i$$

con $N_{i,p} \geq 0$ y $\sum_{j=i-p}^i N_{j,p}(u) = 1$.

En la Figura 2.16 se muestra las envolturas convexas de cada subcurva que forman al B-spline del ejemplo anterior.

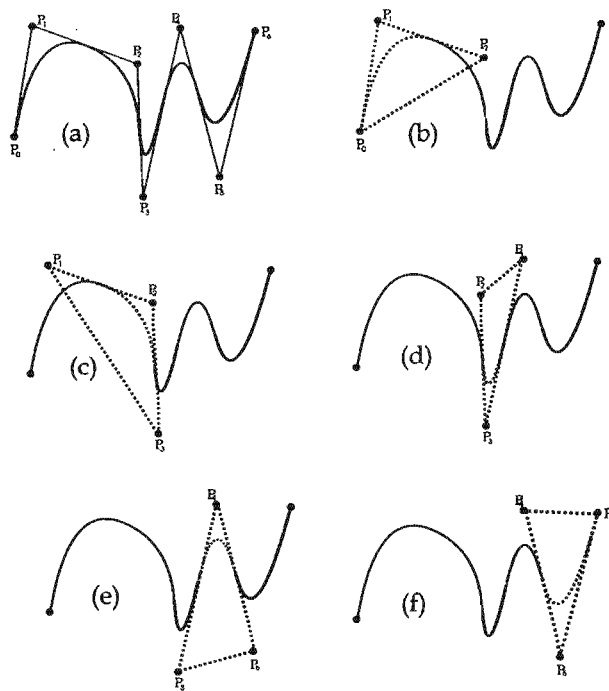


Figura 2.16: (a) Curva B-spline. (b),(c),(d),(e) y (f) Envolturas convexas..

III) Invarianza Afin.

Una curva B-spline es invariante bajo transformaciones afines (ver definición 10), es decir para aplicar ciertas transformaciones geométricas a la curva, sólo hay que hacerlo a los puntos de control.

Esto se tiene por la partición de la unidad de las funciones B-spline, ya que si $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ es un mapeo afín

$$\Phi(C(u)) = \Phi \left(\sum_{i=0}^n P_i N_{i,p}(u) \right) = \sum_{i=0}^n N_{i,p}(u) \Phi(P_i)$$

- IV) Si $n = p$ y $U = \{ \underbrace{a, \dots, a}_{p+1}, \underbrace{b, \dots, b}_{p+1} \}$, $C(u)$ es una curva de Bézier $\mathbf{B}(u)$ (con $u \in [a, b]$), de grado p y puntos de control P_0, \dots, P_p .

Esto se debe a que para este vector de nodos la base B-spline es equivalente a la formada por los polinomios de Bernstein, para demostrarlo probemos la siguiente igualdad por inducción sobre j .

$$N_{i+p-j,j}(u) = B_{i,j} \left(\frac{u-a}{b-a} \right) \tag{2.32}$$

para $j = 0, \dots, p$ y $i = 0, \dots, j$.

Si $j = 0$, por definición tenemos que

$$N_{p,0}(u) = \begin{cases} 1 & \text{si } u \in [u_p, u_{p+1}] = [a, b] \\ 0 & \text{en otro caso.} \end{cases}$$

y

$$B_{0,0} \left(\frac{u-a}{b-a} \right) = \binom{0}{0} \left(\frac{u-a}{b-a} \right)^0 \left(\frac{b-u}{b-a} \right)^0 = 1 \text{ con } u \in [a, b]$$

Ahora supongamos cierto para $j-1$ en donde $0 < j-1 < p$ y probemos para j . Como

$$\begin{aligned} N_{i+p-j,j}(u) &= \frac{u-u_{i+p-j}}{u_{i+p}-u_{i+p-j}} N_{i+p-j,j-1}(u) \\ &\quad + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+p-j+1}} N_{i+p-j+1,j-1}(u) \end{aligned}$$

Si $i=0$ tenemos que $N_{p-j,j-1}(u) = 0$, ya que $u \in [u_{p-j}, u_p] = [a, a)$, entonces

$$N_{p-j,j}(u) = \frac{u_{p+1}-u}{u_{p+1}-u_{p-j+1}} N_{p-j+1,j-1}(u) = \frac{b-u}{b-a} N_{p-(j-1),j-1}(u)$$

y por hipótesis de inducción

$$\frac{b-u}{b-a} N_{p-(j-1),j-1}(u) = \frac{b-u}{b-a} B_{0,j-1} \left(\frac{u-a}{b-a} \right) = B_{0,j} \left(\frac{u-a}{b-a} \right)$$

Si $i=j$ tenemos que $N_{p+1,j-1}(u) = 0$ ya que $u \in [u_{p+1}, u_{p+j+1}] = [b, b)$ entonces

$$\begin{aligned} N_{p,j}(u) &= \frac{u-u_p}{u_{j+p}-u_p} N_{p,j-1}(u) = \frac{u-a}{b-a} N_{(j-1)+p-(j-1),j-1}(u) \\ &= \frac{u-a}{b-a} B_{j-1,j-1} \left(\frac{u-a}{b-a} \right) = B_{j,j} \left(\frac{u-a}{b-a} \right) \end{aligned}$$

Ahora supongamos que $0 < j < i$, entonces

$$\begin{aligned} N_{i+p-j,j}(u) &= \frac{u-u_{i+p-j}}{u_{i+p}-u_{i+p-j}} N_{i+p-j,j-1}(u) \\ &\quad + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+p-j+1}} N_{i+p-j+1,j-1}(u) \\ &= \frac{u-a}{b-a} N_{i-1+p-(j-1),j-1}(u) \\ &\quad + \frac{b-u}{b-a} N_{i+p-(j-1),j-1}(u) \\ &= \frac{u-a}{b-a} B_{i-1,j-1} \left(\frac{u-a}{b-a} \right) \\ &\quad + \frac{b-u}{b-a} B_{i,j-1} \left(\frac{u-a}{b-a} \right) \\ &= B_{i,j} \left(\frac{u-a}{b-a} \right) \end{aligned}$$

Una vez probada la igualdad, notemos que si $j=p$

$$N_{i,p}(u) = B_{i,p} \left(\frac{u-a}{b-a} \right)$$

para $i=0, \dots, p$.

- v) El número de puntos de control, la cantidad de nodos y el grado están relacionados por $m = n + p + 1$.

Esto se debe a la propiedad de no negatividad de las curvas B-spline, es decir cada función $N_{i,p}(u) \geq 0$ si $u \in [u_i, u_{i+p+1})$, en particular si $i = n$ la última función base cumple $N_{n,p}(u) \geq 0$ si $u \in [u_n, u_{n+p+1})$ de donde $u_{n+p+1} = u_m$ y $m = n + p + 1$.

vi) **Modificaciones Locales.**

A diferencia de las curvas de Bézier las curvas B-spline tienen la propiedad de control local, es decir la modificación de algún punto de control altera la curva sólo en ciertas secciones. Esta propiedad es útil sobre todo en el diseño de curvas o superficies de forma libre ya que permite hacer ligeros cambios sin alterar el diseño completo. Así, si modificamos el punto P_i por P'_i , la curva se cambia sobre el rango parametral en donde la respectiva función base $N_{i,p}(u)$ es distinta de cero, lo cual se tiene en (u_i, u_{i+p+1}) .

Por ejemplo en la curva de la Figura 2.17 se modifica el valor del punto P_3 y ésta se ve afectada en el rango parametral $[u_3, u_6)$, alterando las curvas C_1, C_2 y C_3 correspondientes a los intervalos $[u_3, u_4), [u_4, u_5)$ y $[u_5, u_6)$ respectivamente.

En general cualquier modificación al punto P_i , afecta en el rango parametral $[u_i, u_{i+p+1})$ y sólo $p + 1$ curvas definidas sobre los intervalos $[u_i, u_{i+1}), \dots, [u_{i+p}, u_{i+p+1})$ cambian.

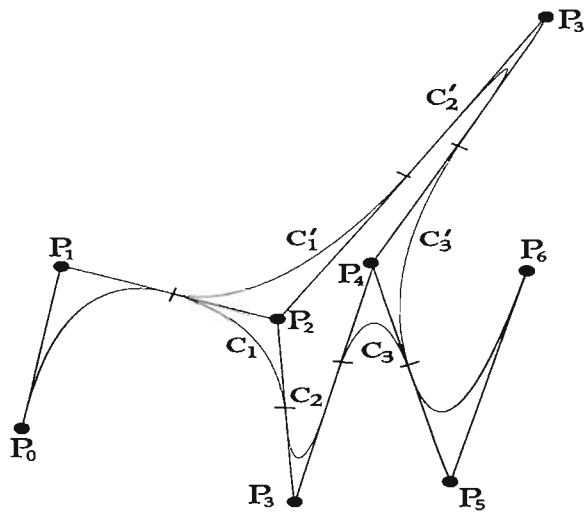


Figura 2.17: La modificación del punto P_3 cambia las subcurvas C_1, C_2 y C_3 por C'_1, C'_2 y C'_3 .

vii) **Diferenciabilidad.**

Ya que $C(u)$ es una combinación lineal de funciones B-spline, su diferenciabilidad está sujeta a la diferenciabilidad de éstas. Así de la demostración del teorema 13 (capítulo 1), sabemos que cada $N_{i,p}(u)$ tiene una continuidad de clase $C^{(p-k)}$ en algún nodo $u \in [u_i, u_{i+p+1})$ en donde k es el número de veces que se repite (es de multiplicidad k). De igual modo una curva B-spline tiene una continuidad de clase $C^{(p-k)}$ sobre algún $u \in U$ de multiplicidad k , en el caso de que u sea algún nodo de los extremos ($u = a$ o $u = b$) su multiplicidad es $p + 1$ y las funciones $N_{0,p}(u)$ y $N_{n,p}(u)$ son discontinuas en éste, sin embargo su evaluación es igual a 1, logrando la interpolación los puntos frontera como se muestra en el inciso i .

Por ejemplo en la Figura 2.18 (a) se muestra la curva B-spline cuadrática del ejemplo 14, y en (b) la misma curva pero ahora con un vector de nodos $U = \{0, 0, 0, 1, 1, 3, 4, 5, 5, 5\}$ y así la unión de la primer y segunda subcurva (que se realiza en el nodo $u = 1$) es ahora de clase $C^{(0)}$ en el punto P_3 .

Es posible hacer modificaciones a los puntos de control de tal manera que se obtenga alguna diferenciabilidad deseada (por supuesto esto también depende de cuantos puntos de control se tengan), por ejemplo si en la Figura 2.18 (b) se modifican los puntos P_2 y P_4 de tal manera que las distancias de P_2 a P_3 y P_3 a P_4 sean proporcionales a las distancias de u_2 a $u_3 = u_4$ y de u_4 a u_5 respectivamente, se logra una continuidad de clase $C^{(1)}$ en P_3 , esto se muestra en 2.18 (c).

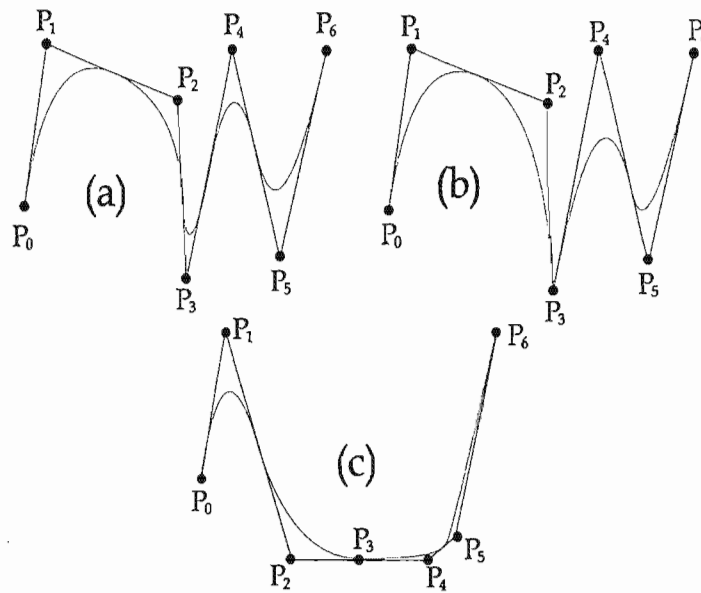


Figura 2.18: (a) Curva B-spline cuadrática. (b) Curva B-spline cuadrática en la cual la primer y segunda subcurva se unen con una continuidad de clase $C^{(0)}$. (c) Curva B-spline con modificaciones en los puntos de control, para lograr una continuidad de clase $C^{(1)}$ en la unión de la primer y segunda curva.

VIII) Repetición de Puntos de Control.

Es posible y algunas veces útil repetir puntos de control, por ejemplo en la Figura 2.19 (b) el punto P_3 es igual a P_2 , y por la propiedad de la envoltura convexa las curvas C_1 y C_2 que se muestran en 2.19 (a) son líneas rectas, ya que $C_1(u)$ está contenida en la envoltura convexa de P_1 , P_2 y P_3 y C_2 en la envoltura convexa de P_2 , P_3 y P_4 . Además aunque visualmente exista un pico, la unión de estas dos curvas es de clase $C^{(1)}$. En 2.19 (c) se hace que el punto P_4 sea igual P_2 y entonces la curva C_2 coincide con este punto.

IX) Derivada.

Si $C(u)^{(k)}$ denota la k -ésima derivada la curva B-spline de $C(u)$, entonces claramente para u fijo tenemos que

$$C(u)^{(k)} = \sum_{i=0}^n N_{i,p}^{(k)}(u)P_i$$

Por ejemplo para $k = 1$ por (1.44)

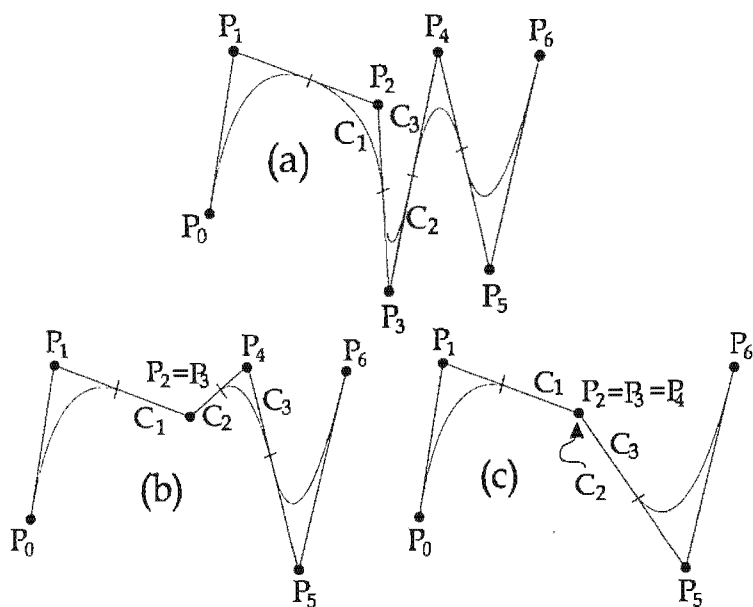


Figura 2.19:

$$\begin{aligned}
 C'(u) &= \sum_{i=0}^n \left(p \frac{N_{i,p-1}(u)}{u_{i+p} - u_i} - p \frac{N_{i+1,p-1}(u)}{u_{i+p+1} - u_{i+1}} \right) P_i \\
 &= \left(\sum_{i=-1}^{n-1} p \frac{N_{i+1,p-1}(u)}{u_{i+p+1} - u_{i+1}} \right) P_{i+1} - \left(\sum_{i=0}^n p \frac{N_{i+1,p-1}(u)}{u_{i+p+1} - u_{i+1}} \right) P_i \\
 &= p \frac{N_{0,p-1}(u)}{u_p - u_0} P_0 + \left(\sum_{i=0}^{n-1} p \frac{N_{i+1,p-1}(u)}{u_{i+p+1} - u_{i+1}} \right) (P_{i+1} - P_i) \\
 &\quad - p \frac{N_{n+1,p-1}(u)}{u_{n+p+1} - u_{n+1}} P_n
 \end{aligned}$$

sin embargo notemos que por definición de U , $u_0 = u_p$ y $u_{n+1} = u_{n+p+1}$ de donde el primer y último término son eliminados ¹⁰ y tenemos que

$$C'(u) = p \sum_{i=0}^{n-1} \frac{N_{i+1,p-1}(u)(P_{i+1} - P_i)}{u_{i+p+1} - u_{i+1}} = \sum_{i=0}^{n-1} N_{i+1,p-1}(u) Q_i$$

con

$$Q_i = p \frac{P_{i+1} - P_i}{u_{i+p+1} - u_{i+1}}$$

Si ahora definimos U' como el vector de nodos obtenido al eliminar el primer y último nodo de U , es decir

¹⁰Tanto en el ejemplo 6 del capítulo 1, como en su comentario previo, se explica por qué para estos casos, es posible eliminar factores en donde existan cocientes no definidos.

$$U' = \underbrace{\{0, \dots, 0\}}_p, u_p, \dots, u_{m-p-1}, \underbrace{\{1, \dots, 1\}}_p$$

es claro que las funciones $N_{i+1,p-1}(u)$ sobre U coinciden con $N_{i,p-1}(u)$ sobre U' para $i = 0, \dots, n-1$, así podemos escribir

$$C'(u) = \sum_{i=0}^{n-1} N_{i,p-1}(u)Q_i \tag{2.33}$$

En general usando el mismo procedimiento podemos obtener las derivadas de orden mayor. Es decir si escribimos $P_i^{(0)} = P_i$ y

$$C(u) = C^{(0)}(u) = \sum_{i=0}^n N_{i,p}(u)P_i^{(0)}$$

entonces

$$C^{(k)}(u) = \sum_{i=0}^{n-k} N_{i,p-k}(u)P_i^{(k)} \tag{2.34}$$

con

$$P_i^{(k)} = \begin{cases} P_i & k = 0 \\ \frac{p-k+1}{u_{i+p+1}-u_{i+k}} (P_{i+1}^{(k-1)} - P_i^{(k-1)}) & k > 0 \end{cases}$$

y

$$U^{(k)} = \underbrace{\{0, \dots, 0\}}_{p-k+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{\{1, \dots, 1\}}_{p-k+1}$$

2.4. Curvas NURBS.

De manera análoga al caso de las curvas de Bézier racionales e integrales, las curvas “Non Uniform Rational B-splines” (NURBS) son una generalización de las curvas B-spline, con las que además se pueden representar cónicas por medio de secciones de curvas racionales que se unen con ciertos grados de diferenciabilidad. Así las siglas NURBS indican que la curva tiene funciones coordenadas racionales en términos de funciones base B-splines definidas sobre un vector de nodos no uniforme.

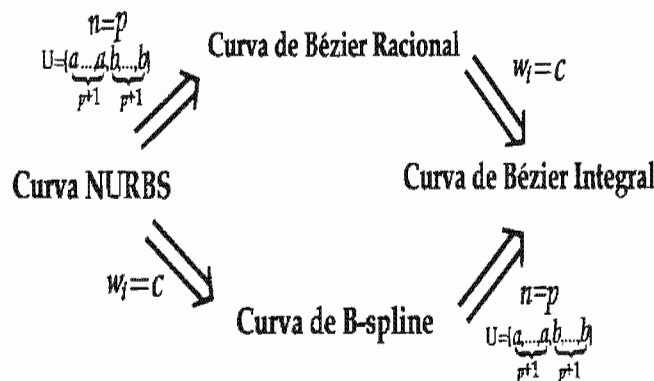


Figura 2.20: Las curvas NURBS son una generalización de todas las curvas presentadas en este capítulo.

2.4. CURVAS NURBS.

Definición 13 Dados los enteros positivos n, p, m , puntos de control P_0, \dots, P_n y pesos w_0, \dots, w_n , entonces una curva NURBS de grado p está dada por

$$C(u) = \frac{\sum_{i=0}^n w_i N_{i,p}(u) P_i}{\sum_{j=0}^n w_j N_{j,p}(u)} \text{ para } u \in [a, b] \quad (2.35)$$

en donde cada $N_{i,p}(u)$ es una función base B-spline definida por (1.39) sobre un vector de $m+1$ nodos no decrecientes y no uniformes

$$U = \{ \underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1} \}$$

En este caso a diferencia de la curva B-spline en U no se exige que los nodos sean estrictamente crecientes, ya que una curva spline es un caso particular de un NURBS.

Ahora si tomamos $U = \{ \underbrace{0, \dots, 0}_{p+1}, \underbrace{1, \dots, 1}_{p+1} \}$ y $n = p$, por la propiedad IV de las curvas B-spline, (2.35)

se escribe como

$$C(u) = \frac{\sum_{i=0}^p w_i B_{i,p} P_i}{\sum_{j=0}^p w_j B_{j,p}(u)} \text{ para } u \in [0, 1]$$

la cual es una expresión que define a la de la curva de Bézier racional.

Por otro lado si $w_i = c$, con c constante para toda i , la propiedad de partición de la unidad de las funciones B-spline implica que (2.35)

$$C(u) = \frac{c \sum_{i=0}^n N_{i,p}(u) P_i}{c \sum_{i=0}^n N_{i,p}(u)} = \sum_{i=0}^n N_{i,p}(u) P_i$$

cuya ecuación coincide con la de la curva B-spline. Y finalmente del hecho de que la curva de Bézier integral sea un caso particular de un B-spline integral y una curva de Bézier racional, deducimos que los NURBS son una generalización de todas las curvas estudiadas en este capítulo (ver Figura 2.20).

De igual forma que para el caso de las curvas de Bézier racionales si definimos

$$R_{i,p}(u) = \frac{w_i N_{i,p}(u)}{\sum_{j=0}^n w_j N_{j,p}(u)} \quad (2.36)$$

para $i = 0, \dots, n$, podemos reescribir (2.35) como

$$C(u) = \sum_{i=0}^n R_{i,p}(u) P_i \quad (2.37)$$

Además la representación en coordenadas homogéneas P_i^w , definidas por (2.20) está dada por

$$\begin{aligned} C^w(u) &= \sum_{i=0}^n N_{i,p}(u) P_i^w \\ &= \left(\sum_{i=0}^n w_i N_{i,p}(u) x_i, \sum_{i=0}^n w_i N_{i,p}(u) y_i, \sum_{i=0}^n w_i N_{i,p}(u) z_i, \sum_{i=0}^n w_i N_{i,p}(u) \right) \end{aligned}$$

Ya que aplicando el mapeo (2.21) tenemos que

$$\begin{aligned} H(C^w(u)) &= \left(\frac{\sum_{i=0}^n w_i N_{i,p}(u) x_i}{\sum_{j=0}^n w_j N_{j,p}(u)}, \frac{\sum_{i=0}^n w_i N_{i,p}(u) y_i}{\sum_{j=0}^n w_j N_{j,p}(u)}, \frac{\sum_{i=0}^n w_i N_{i,p}(u) z_i}{\sum_{j=0}^n w_j N_{j,p}(u)} \right) \\ &= C(u) \end{aligned}$$

Ejemplo 15 Es posible representar una circunferencia usando 3 o 4 secciones de curvas de Bézier racional cuadráticas, en el caso de una representación con 3 secciones podemos tomar el vector de nodos y los pesos siguientes

$$U = \{0, 0, 0, 1, 1, 2, 2, 3, 3, 3\}, w_i = 1, w_j = \frac{1}{2}$$

para $i = 0, 2, 4, 6$ y $j = 1, 3, 5$.

Para la representación con 4 curvas podemos tomar

$$U = \{0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4\}, w_i = 1, w_j = \frac{\sqrt{2}}{2}$$

para $i = 0, 2, 4, 6, 8$ y $j = 1, 3, 5, 7$.

Estas circunferencias se muestra en la Figuras 2.21 y 2.22.

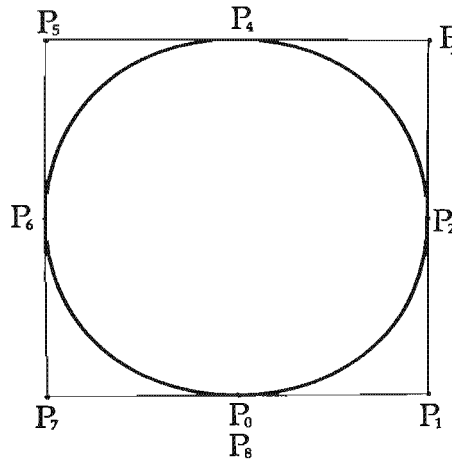


Figura 2.21: Circunferencia representadas con una curvas NURBS.

Ejemplo 16 Construyamos una curva NURBS cúbica con 6 puntos de control y pesos $w_0 = 1, w_1 = 2, w_2 = 1, w_3 = 3, w_4 = 0,5, w_5 = 1$. Entonces por definición tenemos que $p = 3, n = 5$ y $U = \{0, 0, 0, 0, 1, 2, 3, 3, 3\}$. Ahora las funciones $R_{i,3}(u)$ dadas por (2.36), tienen como funciones b-spline

$$N_{0,3}(u) = \begin{cases} (1-u)^3 & \text{si } u \in [0, 1) \\ 0 & \text{en otro caso.} \end{cases}$$

$$N_{1,3}(u) = \begin{cases} u(1-u)^2 + \frac{(2-u)u(1-u)}{2} + \frac{u(2-u)^2}{4} & \text{si } u \in [0, 1) \\ \frac{(2-u)^3}{4} & \text{si } u \in [1, 2) \\ 0 & \text{en otro caso.} \end{cases}$$

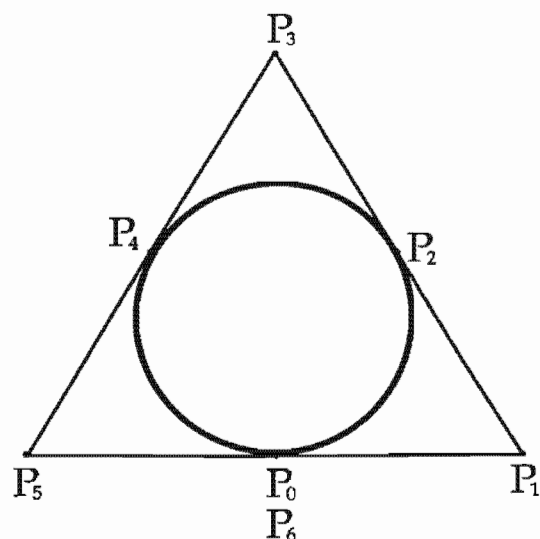


Figura 2.22: Circunferencias representadas con una curvas NURBS.

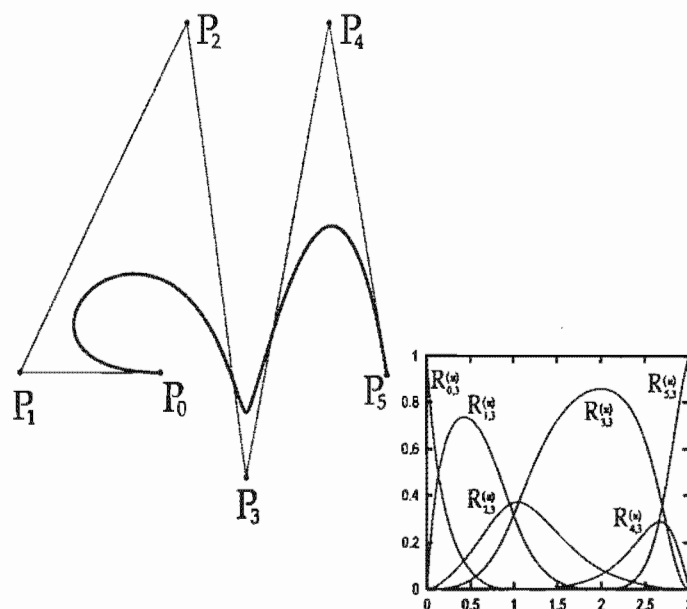


Figura 2.23: Curva NURBS de cúbica con 5 puntos de control.

$$N_{2,3}(u) = \begin{cases} \frac{u^2(1-u)}{2} + \frac{u^2(2-u)}{4} + \frac{(3-u)u^2}{6} & \text{si } u \in [0, 1) \\ \frac{u(2-u)^2}{6} + \frac{(3-u)u(2-u)}{6} + \frac{(3-u)^2(u-1)}{6} & \text{si } u \in [1, 2) \\ \frac{(3-u)^3}{6} & \text{si } u \in [2, 3) \\ 0 & \text{en otro caso.} \end{cases}$$

$$N_{3,3}(u) = \begin{cases} \frac{u^3}{6} & \text{si } u \in [0, 1) \\ \frac{u^2(2-u)}{6} + \frac{u(3-u)(u-1)}{6} + \frac{(3-u)(u-1)^2}{6} & \text{si } u \in [1, 2) \\ \frac{u(3-u)^2}{6} + \frac{(u-1)(3-u)^2}{4} + \frac{(3-u)^2(u-2)}{2} & \text{si } u \in [2, 3) \\ 0 & \text{en otro caso.} \end{cases}$$

$$N_{4,3}(u) = \begin{cases} \frac{(u-1)^3}{4} & \text{si } u \in [1, 2) \\ \frac{(u-1)^2(3-u)}{4} + \frac{(u-1)(3-u)(u-2)}{2} + (3-u)(u-2)^2 & \text{si } u \in [2, 3) \\ 0 & \text{en otro caso.} \end{cases}$$

$$N_{5,3}(u) = \begin{cases} (u-2)^3 & \text{si } u \in [2, 3) \\ 0 & \text{en otro caso.} \end{cases}$$

Todas las $R_{i,3}(u)$ como la curva NURBS se presenta en la Figura 2.23.

Notemos que el segmento de curva cercano al punto P_3 tiene un “ligero pico”, y sin embargo teóricamente esta curva tiene una diferenciabilidad de clase C^2 en $(1, 3)$, esto se debe a que el peso $w_3 = 3$, hace que la curva se “acerque” a este punto.

2.4.1. Propiedades.

Al igual que en el caso de las curvas B-spline, las propiedades de las funciones $R_{i,p}(u)$ y las curvas NURBS que presentamos a continuación, se tienen gracias a las funciones base B-spline presentadas en el capítulo 1.

- I) **Soporte local.** $R_{i,p}(u) \neq 0 \forall u \in (u_i, u_{i+p+1})$.
- II) **No negatividad.** $R_{i,p}(u) \geq 0 \forall i, p$ y u .
- III) **Partición de la unidad;** $\sum_{i=0}^n R_{i,p}(u) = 1$.
- IV) $R_{0,p}(a) = R_{n,p}(b) = 1$
- V) $R_{i,p}(u)$ es $p - k$ veces continuamente diferenciable para algún nodo $u \in [u_i, u_{i+p+1})$, en donde k es la multiplicidad de éste.
- VI) **Interpolación de los puntos frontera.** $C(a) = P_0$ y $C(b) = P_n$.
- VII) **Propiedad fuerte de la envoltura convexa.** $C(u)$ está contenida en la envoltura convexa de sus puntos de control, además si $u \in [u_i, u_{i+1})$ entonces $C(u)$ está contenida dentro de la envoltura convexa de los puntos P_{i-p}, \dots, P_i para $p \leq i \leq n - p - 1$.
- VIII) **Invarianza Afín.** Una transformación afín (ver definición 10) es aplicada a $C(u)$, haciéndolo a sus puntos de control.

- ix) **Modificaciones Locales.** Si un punto de control P_i es movido, o el peso w_i es cambiado, $\mathbf{C}(u)$ sólo se ve afectada en el intervalo $[u_i, u_{i+p+1})$. Si $u^* \in [u_i, u_{i+p+1})$ entonces el incremento (decremento) de w_i hace que el punto $\mathbf{C}(u^*)$ se acerque (aleje) a P_i a través de una recta.
- x) **Diferenciabilidad** $\mathbf{C}(u)$ tiene una continuidad de clase $C^{(p-k)}$ en algún nodo interior $u \in U$ de multiplicidad k .
- xi) **Repetición de puntos de control.** Es posible repetir puntos de control con lo que se puede obtener picos visuales sin perder diferenciabilidad.
- xii) **Derivada.** Se puede deducir una formula para la derivada de las curvas NURBS usando la ecuación (2.22) y haciendo $f(u) = \sum_{i=0}^n w_i N_{i,p}(u) P_i$ y $g(u) = \sum_{j=0}^n w_j N_{j,p}(u)$. Las derivadas de $f(u)$ y $g(u)$ se obtienen aplicando la formula (2.34) considerando $w_i P_i$ los puntos de control de $f(u)$ y w_i los de $g(u)$.

Capítulo 3

Superficies NURBS

Como su nombre lo indica, en este capítulo estudiaremos las superficies NURBS. Veremos que utilizando el esquema de superficie de producto tensor, tanto la definición de la superficie como las propiedades son similares al caso de las curvas NURBS, en particular mostraremos que éstas, son la generalización de las superficies de Bézier y B-spline.

3.1. Superficies de producto tensor.

Hay varias maneras de representar una superficie paramétrica, las cuales por lo general, difieren en el tipo de funciones coordenadas y dominio. Sin embargo, el esquema más comúnmente usado es llamado producto tensor¹, el cual es un modelo de curvas bidireccional, que usa funciones bivariadas, obtenidas del producto de funciones base univariadas, y puntos en el plano o espacio como coeficientes.

Definición 14 Sean n, m enteros positivos, un conjunto de puntos $P_{i,j} \in \mathbb{R}^3$ y una familia de funciones base $f_i : [a, b] \rightarrow \mathbb{R}$ y $g_j : [c, d] \rightarrow \mathbb{R}$ (para $i = 1, \dots, n, j = 1, \dots, m$ y $a, b, c, d \in \mathbb{R}$). Entonces una superficies de producto tensor, es un mapeo de $[a, b] \times [c, d] \rightarrow \mathbb{R}^3$, dado por

$$S(u, v) = (x(u, v), y(u, v), z(u, v)) = \sum_{i=0}^n \sum_{j=0}^m f_i(u)g_j(v)P_{i,j} \quad (3.1)$$

Este tipo de superficies, también se puede escribir en la forma matricial siguiente

$$S(u, v) = [f_0(u) \cdots f_n(u)] \begin{bmatrix} P_{0,0} & \cdots & P_{0,m} \\ \vdots & & \vdots \\ P_{n,0} & \cdots & P_{n,m} \end{bmatrix} \begin{bmatrix} g_0(v) \\ \vdots \\ g_m(v) \end{bmatrix}$$

notando que la matriz tiene como elementos puntos en \mathbb{R}^3 y así la multiplicación por el renglón y columna de funciones base, son multiplicaciones de un real por un vector.

Esta representación es útil en muchos cálculos sobre las superficies de producto tensor.

¹Otro posibilidad esquema triangular, que se presenta en [HL97, páginas 287-329], [Far90, Cap.18].

Ahora, supongamos que fijamos el parámetro $u = u_0$, entonces

$$\begin{aligned} S(u_0, v) &= C_{u_0}(v) = \sum_{i=0}^n \sum_{j=0}^m f_i(u_0) g_j(v) P_{i,j} \\ &= \sum_{j=0}^m g_j(v) \left(\sum_{i=0}^n f_i(u_0) P_{i,j} \right) \\ &= \sum_{j=0}^m g_j(v) Q_j(u_0) \end{aligned}$$

con $Q_j(u_0) = \sum_{i=0}^n f_i(u_0) P_{i,j}$.

Es decir la superficie se vuelve una curva paramétrica llamada *isocurva para el parámetro v* con puntos de control $Q_j(u_0)$ (para $j = 0, \dots, m$). Análogamente la respectiva isocurva de parámetro u $C_{v_0}(u)$, se obtiene fijando $v = v_0$, la cual tiene puntos de control $Q_i(v_0) = \sum_{j=0}^m g_j(v_0) P_{i,j}$. Notemos que estas dos curvas se intersectan en el punto (u_0, v_0) .

Ejemplo 17 Sean las funciones base² $f_0(u) = 1 - u$, $f_1(u) = u$, $g_0(v) = 1 - v$ y $g_1(v) = v$ con $0 \leq u, v \leq 1$. Entonces, la correspondiente superficie de producto tensor con puntos $P_{i,j}$ para $i, j = 0, 1$, está dada por

$$S(u, v) = (1 - u)(1 - v)P_{0,0} + (1 - u)vP_{0,1} + u(1 - v)P_{1,0} + uvP_{1,1}$$

y su respectiva representación matricial es

$$S(u, v) = \begin{bmatrix} 1 - u & u \end{bmatrix} \begin{bmatrix} P_{0,0} & P_{0,1} \\ P_{1,0} & P_{1,1} \end{bmatrix} \begin{bmatrix} 1 - v \\ v \end{bmatrix}$$

Ahora si fijamos $u = u_0$

$$S(u_0, v) = C_{u_0}(v) = (1 - v)Q_0(u_0) + vQ_1(u_0)$$

con $Q_j(u_0) = (1 - u_0)P_{0,j} + u_0P_{1,j}$ (para $j = 0, 1$), es decir la superficie se convierte en una interpolación lineal de los puntos $Q_0(u_0)$ y $Q_1(u_0)$, que a su vez resultan de la interpolación lineal de $P_{0,0}$, $P_{1,0}$ y $P_{0,1}$, $P_{1,1}$ respectivamente (ver Figura 3.1). La isocurva $C_{u_0}(u)$ se deduce de forma análoga fijando $v = v_0$. Notemos finalmente, que la superficie interpola a los puntos $\{P_{i,j}\}$, es decir

$$\begin{aligned} S(0, 0) &= P_{0,0} & S(0, 1) &= P_{0,1} \\ S(1, 0) &= P_{1,0} & S(1, 1) &= P_{1,1} \end{aligned}$$

□

Ejemplo 18 Sea la base de funciones de potencia $f_i(u) = u^i$ y $g_j(v) = v^j$ con $0 \leq u, v \leq 1$ y los puntos $P_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$ para $i = 0, \dots, n$, $j = 0, \dots, m$. Entonces una superficie de producto tensor para este caso es

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m u^i v^j P_{i,j} \tag{3.2}$$

la cual tiene una representación matricial dada por

²Podemos afirma esto, ya que notemos que estas funciones son completamente análogas a las funciones B-spline de grado 1, presentadas en el capítulo 1, ejemplo 7.

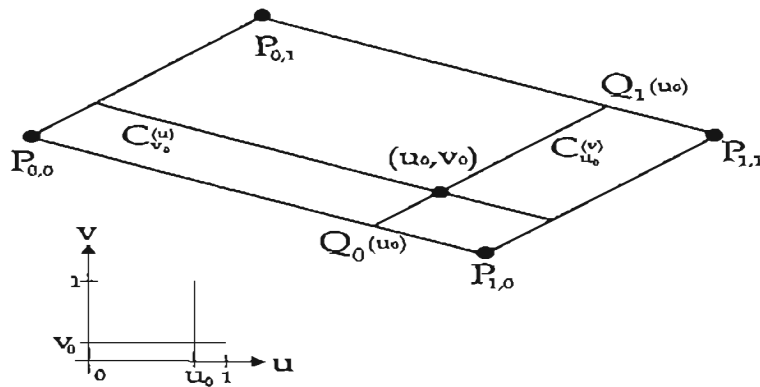


Figura 3.1: Superficie de productor tensor.

$$S(u, v) = \begin{bmatrix} 1 & u & \cdots & u^n \end{bmatrix} \begin{bmatrix} P_{0,0} & \cdots & P_{0,m} \\ \vdots & & \vdots \\ P_{n,0} & \cdots & P_{n,m} \end{bmatrix} \begin{bmatrix} 1 \\ v \\ \vdots \\ v^n \end{bmatrix}$$

Por ejemplo si tomamos $n = 2$ y $m = 3$, (3.2) sería

$$\begin{aligned} S(u, v) &= \sum_{i=0}^2 \sum_{j=0}^3 u^i v^j P_{i,j} \\ &= P_{0,0} + v P_{0,1} + v^2 P_{0,2} + v^3 P_{0,3} \\ &\quad + u P_{1,0} + uv P_{1,1} + uv^2 P_{1,2} + uv^3 P_{1,3} \\ &\quad + u^2 P_{2,0} + u^2 v P_{2,1} + u^2 v^2 P_{2,2} + u^2 v^3 P_{2,3} \end{aligned}$$

Esta superficie se presenta en la Figura 3.2.

□

3.2. Superficies de Bézier integrales.

Una superficie de Bézier integral, es una superficie de productor tensor, formada por funciones bivariables definidas por productos de polinomios de Bernstein (ver capítulo 2). A continuación presentamos la definición formal.

Definición 15 *Dados dos enteros positivos p, q y un conjunto de puntos de control $\{P_{i,j}\} \in \mathbb{R}^2$ o \mathbb{R}^3 , (para $i = 0, \dots, p$ y $j = 0, \dots, q$) una Superficie de Bézier integral de grado p en dirección de s y de grado q en dirección de t está dada por la superficie de producto tensor*

$$S(s, t) = \sum_{i=0}^p \sum_{j=0}^q B_{i,p}(s) B_{j,q}(t) P_{i,j} \text{ con } 0 \leq s, t \leq 1$$

donde $B_{i,p}(s)$ y $B_{j,q}(t)$ son polinomios de Bernstein (2.2), para toda i, j .

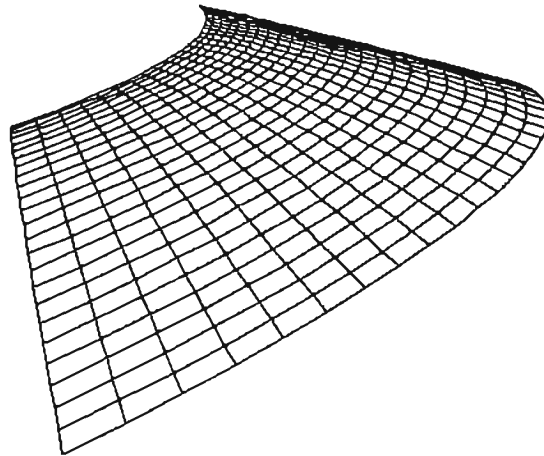


Figura 3.2: Superficie de productor tensor con funciones base de potencia.

Para este caso notemos que si fijamos $s = s_0$,

$$\begin{aligned}
 S(s_0, t) = C_{s_0}(t) &= \sum_{i=0}^p \sum_{j=0}^q B_{i,p}(s_0) B_{j,q}(t) P_{i,j} \\
 &= \sum_{j=0}^q B_{j,q}(t) \left(\sum_{i=0}^p B_{i,p}(s_0) P_{i,j} \right) \\
 &= \sum_{j=0}^q B_{j,q}(t) Q_j(s_0)
 \end{aligned} \tag{3.3}$$

donde

$$Q_j(s_0) = \sum_{i=0}^p B_{i,p}(s_0) P_{i,j} \tag{3.4}$$

es decir, la isocurva de parámetro t , $C_{s_0}(t)$, es una curva de Bézier integral de grado q con puntos de control $Q_j(s_0)$ para $j = 0, \dots, q$.

Análogamente la isocurva de parámetro s , que se obtiene fijando $t = t_0$ está dada por

$$C_{t_0}(s) = \sum_{i=0}^p B_{i,p}(s) Q_i(t_0) \tag{3.5}$$

que es una curva de Bézier integral de grado p con puntos de control

$$Q_i(t_0) = \sum_{j=0}^q B_{i,q}(t_0) P_{i,j} \tag{3.6}$$

Ejemplo 19 Construyamos una superficie de Bézier integral tomando, $p = 3$, $q = 3$ y los puntos de control

Puntos de control				
i/j				
	(15, 0, 10)	(5, 0, 15)	(-5, 0, 15)	(-15, 0, 10)
	(15, 5, 15)	(5, 5, 25)	(-5, 5, 25)	(-15, 5, 15)
	(15, 0, 15)	(5, 10, 25)	(-5, 10, 25)	(-15, 10, 15)
	(15, 15, 10)	(5, 15, 15)	(-5, 15, 15)	(-15, 15, 10)

entonces según la definición anterior la superficie, está dada por

$$S(s, t) = \sum_{i=0}^3 \sum_{j=0}^3 B_{i,3}(s) B_{j,3}(t) P_{i,j}$$

la cual se muestra en la Figura (3.3) y sus respectivas funciones base $B_{1,3}(s)B_{2,3}(t)$ y $B_{3,3}(s)B_{1,3}(t)$ en la Figura (3.3).

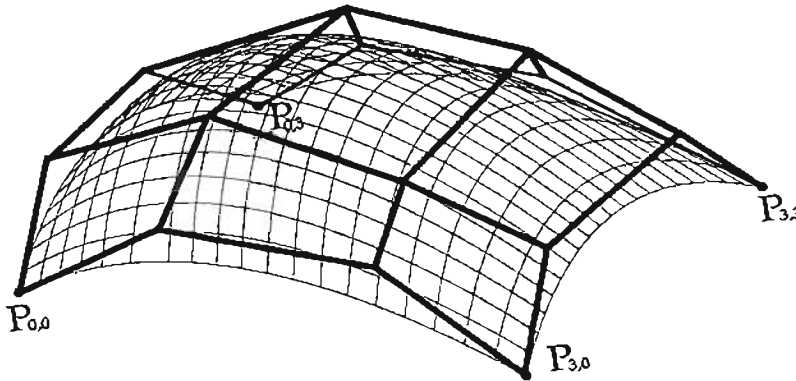


Figura 3.3: Superficie de Bézier integral.

□

Es útil notar, que el algoritmo de Casteljau (ver capítulo 2), puede ser aplicado fácilmente en el cálculo de los puntos de control de las isocurvas (3.3) y (3.5). Esto se debe, a que (3.4), representa un punto de una curva de Bézier integral, el cual sabemos se puede calcular con este algoritmo.

El proceso inicia aplicando (2.1), a los puntos $P_{i,j}$ para $i = 0, \dots, p$ con $j = 0$, con el que se obtiene el primer punto de control de la isocurva, después se hace lo mismo para $j = 1$ y así sucesivamente. Esto se presenta en la Figura (3.5). Claramente el caso de (3.6) es análogo.

3.2.1. Propiedades

Las siguientes propiedades de las funciones $B_{i,p}(s)B_{j,q}(t)$ y la superficie $S(s, t)$, se deducen claramente de las respectivas para los polinomios de Bernstein y las curvas de Bézier discutidas en la sección 2.1 del capítulo 2.

i) No negatividad.

$$B_{i,p}(s)B_{j,q}(t) \geq 0, \forall i, j, s, t.$$

ii) Partición de la unidad.

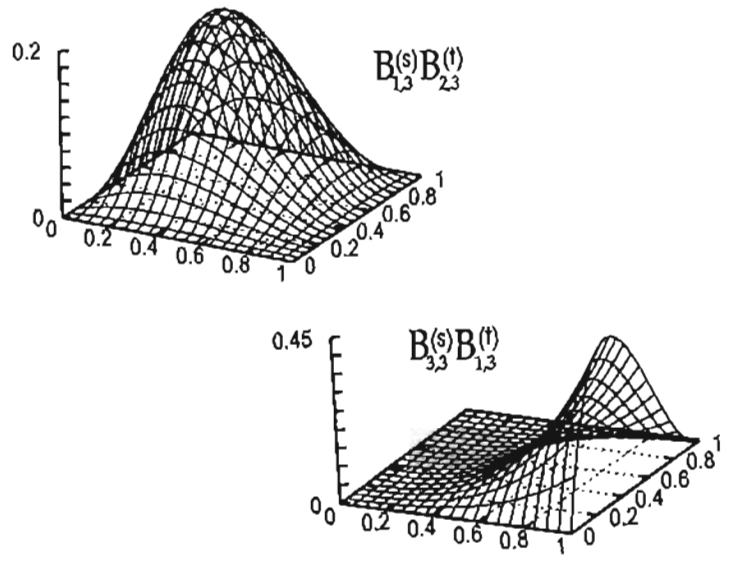


Figura 3.4: Superficies base.

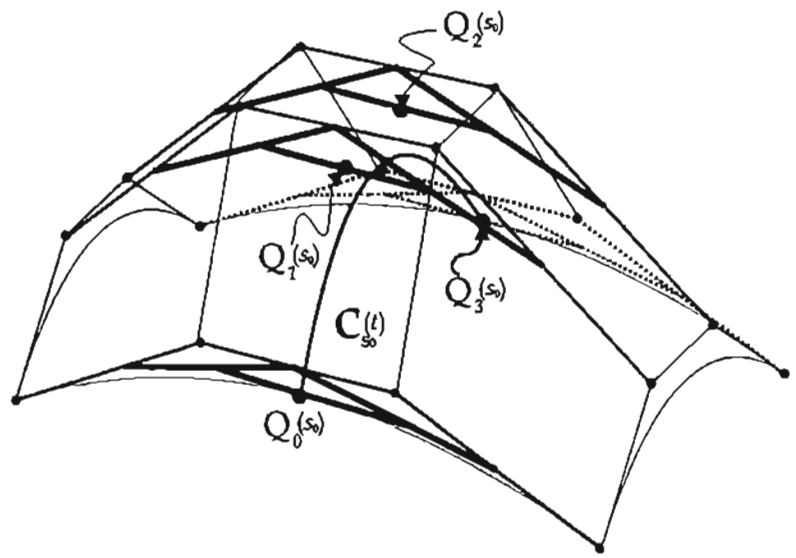


Figura 3.5: El algoritmo de Casteljau aplicado a una superficie de Bézier.

$\sum_{i=0}^p \sum_{j=0}^q B_{i,p}(s)B_{j,q}(t) = 1$. Esta propiedad se sigue de

$$\sum_{i=0}^p \sum_{j=0}^q B_{i,p}(s)B_{j,q}(t) = \left(\sum_{i=0}^p B_{i,p}(s) \right) \left(\sum_{j=0}^q B_{j,q}(t) \right) = 1 \cdot 1 = 1.$$

iii) $B_{0,p}(0)B_{0,q}(0) = B_{p,p}(1)B_{0,q}(0) = B_{0,p}(0)B_{q,q}(1) = B_{p,p}(1)B_{q,q}(1) = 1$.

iv) Interpolación de los cuatro puntos de las esquinas. $S(i, j) = P_{i,j}$ para $i, j = 0, 1$.

Esta propiedad se tiene por iii

v) Envoltura convexa.

$S(s, t)$ está contenida en el poliedro formado por los puntos de control. Por ii y i.

vi) Invarianza afín. Una transformación afín (ver definición 10) es aplicada a $S(s, t)$, haciéndolo a sus puntos de control. Esta propiedad es cierta por ii y i.

vii) Invarianza bajo Transformaciones de parámetros afines.

Si $\alpha, \beta, \gamma, \delta \in \mathbb{R}$ tal que $\alpha < \beta$ y $\gamma < \delta$

$$\begin{aligned} S(s, t) &= \sum_{i=0}^p \sum_{j=0}^q B_{i,p}(s)B_{j,q}(t)P_{i,j} \\ &= \sum_{i=0}^p \sum_{j=0}^q B_{i,p} \left(\frac{u - \alpha}{\beta - \alpha} \right) B_{j,q} \left(\frac{v - \gamma}{\delta - \gamma} \right) P_{i,j} \\ &= S(u, v) \end{aligned}$$

con $\alpha \leq u \leq \beta$ y $\gamma \leq v \leq \delta$.

viii) Derivada.

De (2.9) tenemos que

$$\begin{aligned} \frac{\partial S(s, t)}{\partial s} &= \sum_{j=0}^q p \left(\sum_{i=0}^{p-1} (P_{i+1,j} - P_{i,j} B_{i,p-1}(s)) \right) B_{j,q}(t) \\ &= p \sum_{j=0}^q \sum_{i=0}^{p-1} \Delta_i P_{i,j} B_{i,p-1}(s) B_{j,q}(t) \end{aligned}$$

con $\Delta_i P_{i,j} = P_{i+1,j} - P_{i,j}$.

y

$$\begin{aligned} \frac{\partial S(s, t)}{\partial t} &= \sum_{i=0}^p q \left(\sum_{j=0}^{q-1} (P_{i,j+1} - P_{i,j}) B_{j,q-1}(t) \right) B_{i,p}(s) \\ &= q \sum_{i=0}^p \sum_{j=0}^{q-1} \Delta_j P_{i,j} B_{j,q-1}(t) B_{i,p}(s) \end{aligned}$$

con $\Delta_j P_{i,j} = P_{i,j+1} - P_{i,j}$.

3.3. SUPERFICIES DE BÉZIER RACIONALES.

En general si $0 \leq e \leq p$ y $0 \leq f \leq q$ por (2.10)

$$\frac{\partial^{e+f} S(s, t)}{\partial s^e \partial t^f} = \frac{p!}{(p-e)!} \frac{q!}{(q-f)!} \sum_{i=0}^{p-e} \sum_{j=0}^{q-f} \Delta^{e,f} P_{i,j} B_{i,p-e}(s) B_{j,q-f}(t)$$

en donde

$$\Delta^{e,f} P_{i,j} = \sum_{l=0}^e \sum_{m=0}^f \binom{e}{l} \binom{f}{m} (-1)^l (-1)^m P_{i+e-l, j+f-m}$$

3.3. Superficies de Bézier racionales.

De forma análoga al caso de las curvas de Bézier integrales, y de la definición de la superficie de producto tensor, una superficie de Bézier integral no puede realizar una representación precisa (tanto de forma analítica como geométrica) de secciones (también llamados parches) de superficies cónicas. Así el equivalente de una curva de Bézier racional para superficies es la superficie de Bézier racional.

Definición 16 *Dados dos enteros positivos p, q , un conjunto de puntos de control $\{P_{i,j}\}$ y pesos $\{w_{i,j}\}$ para $i = 0, \dots, p$, $j = 0, \dots, q$, una Superficie de Bézier racional de grado p en dirección s y grado q en dirección t se define como*

$$S(s, t) = \frac{\sum_{i=0}^p \sum_{j=0}^q w_{i,j} B_{i,p}(s) B_{j,q}(t) P_{i,j}}{\sum_{i=0}^p \sum_{j=0}^q w_{i,j} B_{i,p}(s) B_{j,q}(t)} \quad 0 \leq s, t \leq 1$$

en donde $B_{i,p}(s)$ y $B_{j,q}(t)$ son los polinomios de Bernstein definidos por (2.3) para toda i, j .

Ahora la representación de esta superficie en coordenadas homogéneas definidas por (2.20), es

$$S^w(s, t) = \sum_{i=0}^p \sum_{j=0}^q B_{i,p}(s) B_{j,q}(t) P_{i,j}^w$$

ya que aplicado el mapeo (2.21) tenemos

$$\begin{aligned} H(S^w(s, t)) &= \frac{\sum_{i=0}^p \sum_{j=0}^q w_{i,j} B_{i,p}(s) B_{j,q}(t) P_{i,j}}{\sum_{i=0}^p \sum_{j=0}^q w_{i,j} B_{i,p}(s) B_{j,q}(t)} = S(s, t) \\ &= \sum_{i=0}^p \sum_{j=0}^q R_{i,j,p,q}(s, t) P_{i,j} \end{aligned}$$

en donde

$$R_{i,j,p,q}(s, t) = \frac{w_{i,j} B_{i,p}(s) B_{j,q}(t)}{\sum_{l=0}^p \sum_{m=0}^q B_{l,p}(s) B_{m,q}(t) w_{l,m}} \quad (3.7)$$

sin embargo por simplicidad escribiremos $R_{i,j}(s, t)$ en vez de $R_{i,j,p,q}(s, t)$.

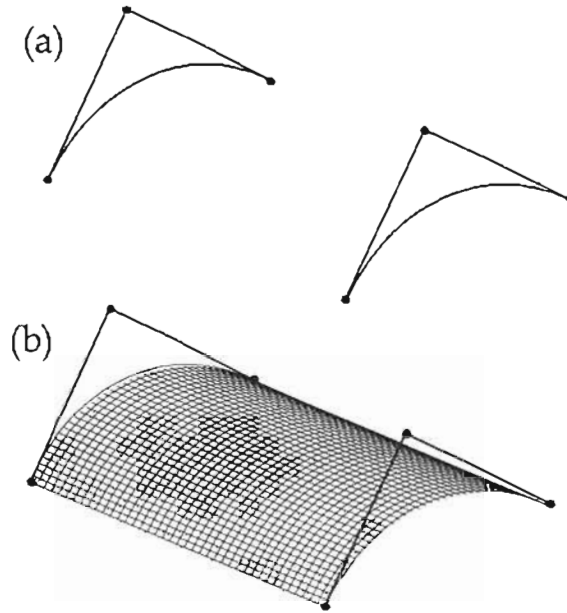


Figura 3.6: Parche de Bézier racional que representa un cilindro parabólico.

Ejemplo 20 Construyamos una superficie de Bézier racional que represente un cilindro que tenga como curvas límites dos arcos circulares (ver Figura 3.6 (a)).

Entonces de la sección de cónicas y curvas de Bézier racionales (2.2.2), sabemos que es posible obtener la representación un arco circular de 90 grados usando una curva de Bézier racional cuadrática. Así, supongamos que estas están dadas por

$$B_k(s) = \frac{\sum_{i=0}^2 B_{i,2}(s)w_i^k P_i^k}{\sum_{i=0}^2 B_{i,2}(s)w_i^k}$$

en donde $w_i^k = 1$ para $i = 0, 2$ y $w_1^k = \frac{\sqrt{2}}{2}$ para $k = 0, 1$.

Por lo que la superficie buscada se obtiene haciendo una interpolación lineal entre estas dos curvas, que notemos se puede escribir en términos de polinomios de Bernstein en la siguiente forma

$$(1-t)P_0 + tP_1 = \sum_{j=0}^1 B_{j,1}(t)P_j$$

en donde P_0 y P_1 son puntos de $B_0(s)$ y $B_1(s)$ respectivamente.

Así se deduce que el cilindro esta dado por

$$S(s, t) = \frac{\sum_{i=0}^2 \sum_{j=0}^1 B_{i,2}(s)B_{j,1}(t)w_{i,j}P_{i,j}}{\sum_{i=0}^2 \sum_{j=0}^1 B_{i,2}(s)B_{j,1}(t)w_{i,j}}$$

con $P_{i,0} = P_i^0$, $P_{i,1} = P_i^1$, pesos

$$\begin{array}{ll} w_{0,0} = 1 & w_{0,1} = 1 \\ w_{1,0} = \frac{\sqrt{2}}{2} & w_{1,1} = \frac{\sqrt{2}}{2} \\ w_{2,0} = 1 & w_{2,1} = 1 \end{array}$$

En la Figura 3.6 (b), se presenta la superficie resultante.

3.3.1. Propiedades.

Las siguientes propiedades se deducen de las respectivas para las funciones base de las curvas de Bézier racional.

- i) **No negatividad;** $R_{i,j}(s, t) \geq 0 \forall s, t, i, j$.
 ii) **Partición de la unidad;** $\sum_{i=0}^p \sum_{j=0}^q R_{i,j}(s, t) = 1$.

$$\frac{\sum_{i=0}^p \sum_{j=0}^q B_{i,p}(s) B_{j,q}(t) w_{i,j}}{\sum_i \sum_j B_{i,p}(s) B_{j,q}(t) w_{i,j}} = 1$$

iii) $R_{0,0}(0, 0) = R_{p,0}(1, 0) = R_{0,q}(0, 1) = R_{p,q}(1, 1) = 1$.

iv) Si $w_{i,j} = c$ con c constante para toda i, j , entonces $R_{i,p}(s, t) = B_{i,p}(s) B_{j,q}(t)$.

Esto se sigue de lo siguiente

$$\begin{aligned} R_{i,p}(s, t) &= \frac{w_{i,j} B_{i,p}(s) B_{j,q}(t)}{\sum_{i=0}^p \sum_{j=0}^q w_{i,j} B_{i,p}(s) B_{j,q}(t)} = \frac{B_{i,p}(s) B_{j,q}(t)}{\sum_{i=0}^p \sum_{j=0}^q B_{i,p}(s) B_{j,q}(t)} \\ &= B_{i,p}(s) B_{j,q}(t). \end{aligned}$$

- v) **Interpolación de los cuatro puntos de las esquinas.** $S(i, j) = P_{i,j}$ para $i, j = 0, 1$. Se tiene por iii).
 vi) **Envoltura convexa.** $S(s, t)$ está contenida en el la envoltura convexa formada por los puntos de control. Se tiene por i y ii).
 vii) **Invarianza afín.** Una transformación afín es aplicada a $S(s, t)$, haciéndolo a sus puntos de control. Se tiene por ii).
 viii) **Invarianza bajo transformaciones de parámetros afines.** Es decir si, $\alpha, \beta, \gamma, \delta \in \mathbb{R}$

$$\begin{aligned} S(s, t) &= \frac{\sum_{i=0}^p \sum_{j=0}^q w_{i,j} B_{i,p}(s) B_{j,q}(t) P_{i,j}}{\sum_{i=0}^p \sum_{j=0}^q w_{i,j} B_{i,p}(s) B_{j,q}(t)} \\ &= \frac{\sum_{i=0}^p \sum_{j=0}^q w_{i,j} B_{i,p}\left(\frac{u-\alpha}{\beta-\alpha}\right) B_{j,q}\left(\frac{v-\gamma}{\delta-\gamma}\right) P_{i,j}}{\sum_{i=0}^p \sum_{j=0}^q w_{i,j} B_{i,p}\left(\frac{u-\alpha}{\beta-\alpha}\right) B_{j,q}\left(\frac{v-\gamma}{\delta-\gamma}\right)} \\ &= S(u, v) \end{aligned}$$

con $\alpha \leq u \leq \beta$ y $\gamma \leq v \leq \delta$.

- ix) La superficie de Bézier integral es un caso particular de la superficie de Bézier racional. Se sigue de iv).
 x) Si $w_{i,j}$ se incrementa (o disminuye), la superficie $S(s, t)$ se acerca (o aleja) a $P_{i,j}$, estos movimientos se realizan siguiendo una línea recta. Como ejemplo de esto, tomemos la superficie del ejemplo 20 y modifiquemos los pesos que determinan el tipo de cónica de las curvas de los extremos, es decir $w_{1,0}$ y $w_{1,1}$. En la Figura 3.7 se presentan las distintas superficies resultantes al realizar estas variaciones. Notemos por ejemplo en la Figura (d), que una de las curvas de los extremos es muy aproximada a su polígono de control y la otra a una recta, esto se debe a que $w_{1,0} = 30$ y $w_{1,1} = 0.01$.

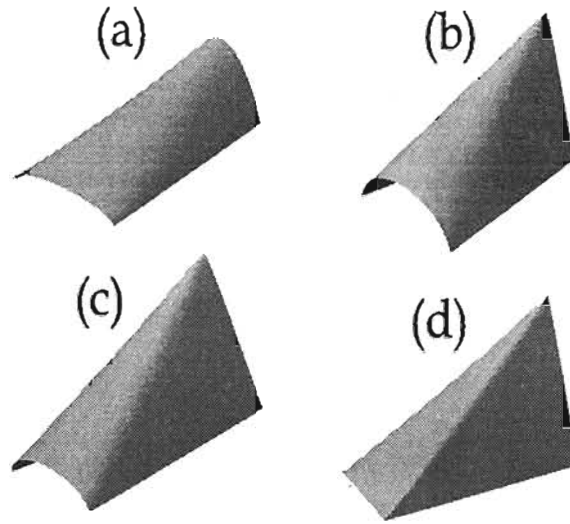


Figura 3.7: Parches racionales que resultan de la variación de sus pesos. (a) $w_{1,0} = 0.9$ $w_{1,1} = 0.6$ (b) $w_{1,0} = 5$ y $w_{1,1} = 0.2$ (c) $w_{1,0} = 10$ y $w_{1,1} = 1$ (d) $w_{1,0} = 30$ y $w_{1,1} = 0.01$.

x1) Derivada. Supongamos que

$$F(s, t) = \frac{f(s, t)}{g(s, t)}$$

entonces si denotamos $f_\alpha(s, t) = \frac{\partial f(s, t)}{\partial \alpha}$, con $\alpha = s$ o t , tenemos

$$\begin{aligned} F_\alpha(s, t) &= \frac{g(s, t)f_\alpha(s, t) - g_\alpha f(s, t)}{g^2(s, t)} \\ &= \frac{f_\alpha(s, t) - g_\alpha(s, t)F(s, t)}{g(s, t)} \end{aligned}$$

Ahora si $F = F(s, t)$, $g = g(s, t)$ y $\frac{\partial^{k+l} f(s, t)}{\partial s^k \partial t^l} = f^{(k, l)}$ tenemos

$$\begin{aligned} f^{(k, l)} &= [(gF)^k]^l = \left(\sum_{i=0}^k \binom{k}{i} g^{(i, 0)} F^{(k-i, 0)} \right)^l \\ &= \sum_{i=0}^k \binom{k}{i} \sum_{j=0}^l \binom{l}{j} g^{(i, j)} F^{(k-i, l-j)} \\ &= g^{(0, 0)} F^{(k, l)} + \sum_{i=1}^k g^{(i, 0)} F^{(k-i, l)} \\ &\quad + \sum_{j=1}^l \binom{l}{j} g^{(0, j)} F^{(k, l-j)} + \sum_{i=1}^k \binom{k}{i} \sum_{j=1}^l \binom{l}{j} g^{(i, j)} F^{(k-i, l-j)} \end{aligned}$$

entonces

$$F^{(k,l)} = \frac{f^{(k,l)}}{g} - \frac{\sum_{i=1}^k \binom{k}{i} g^{(i,0)} F^{(k-i,l)}}{g} - \frac{\sum_{j=1}^l \binom{l}{j} g^{(0,j)} F^{(k,l-j)}}{g} - \frac{\sum_{i=1}^k \binom{k}{i} \sum_{j=1}^l \binom{l}{j} g^{(i,j)} F^{(k-i,l-j)}}{g} \quad (3.8)$$

Finalmente si tomamos

$$f = \sum_{i=0}^p \sum_{j=0}^q w_{i,j} B_{i,p}(s) B_{j,q}(t) P_{i,j}$$

y

$$g = \sum_{i=0}^p \sum_{j=0}^q w_{i,j} B_{i,p}(s) B_{j,q}(t)$$

la superficie de Bézier racional está dada por $S = \frac{f}{g}$ y por ejemplo

$$S_{s,t} = S^{(1,1)} = \frac{f^{(1,1)} - g^{(1,0)} S^{(0,1)} - g^{(0,1)} S^{(1,0)} - g^{(1,1)} S}{g}$$

$$S_{s,s} = S^{(2,0)} = \frac{f^{(2,0)} - 2g^{(1,0)} S^{(1,0)} - g^{(2,0)} S}{g}$$

$$S_{t,t} = S^{(0,2)} = \frac{f^{(0,2)} - 2g^{(0,1)} S^{(0,1)} - g^{(0,2)} S}{g}$$

3.4. Superficies B-spline.

Una superficie B-spline, es una superficie de producto tensor la cual tiene como funciones base, las funciones B-spline (ver capítulo 1) y por lo tanto sus propiedades junto con las equivalentes, de las curvas B-spline (ver capítulo 2), a continuación presentamos la definición formal.

Definición 17 *Dados los enteros positivos p, q, n, m y un conjunto de puntos de control $\{P_{i,j}\}$, (para $i = 0, \dots, n$ y $j = 0, \dots, m$) una Superficie de B-spline de grado p en dirección u y grado q en dirección v , está dada por*

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{i,j} \quad a \leq u, v \leq b \quad (3.9)$$

en donde cada $N_{i,p}(u)$ y $N_{j,q}(v)$ son funciones B-spline (1.99), definidas respectivamente sobre los vectores de $r+1$ y $s+1$ nodos no decrecientes

$$U = \underbrace{\{a, \dots, a\}}_{p+1}, \underbrace{\{u_{p+1}, \dots, u_{r-p-1}, b, \dots, b\}}_{p+1}$$

$$V = \underbrace{\{a, \dots, a\}}_{q+1}, \underbrace{\{v_{q+1}, \dots, v_{s-q-1}, b, \dots, b\}}_{q+1}$$

U y V no necesariamente tienen que tener los nodos de los extremos iguales, sin embargo por simplicidad generalmente se supone que sí.

Claramente para este caso, la isocurva de parámetro v , $C_{u_0}(v)$, que se obtiene fijando el valor de $u = u_0$

$$C_{u_0}(v) = S(u_0, v) = \sum_{j=0}^m N_{j,q}(v)Q_j(u_0)$$

con

$$Q_j(u_0) = \sum_{i=0}^n N_{i,p}(u_0)P_{i,j}$$

es una curva B-spline de grado q con puntos de control $Q_j(u_0)$ para $j = 0, \dots, m$ sobre un vector de nodos U . Análogamente para $C_{v_0}(u)$.

Ejemplo 21 Construyamos una superficie de B-spline de grado 3 en dirección de u y 2 en dirección de v y puntos de control

Puntos de control					
i/j					
	$(-10, -10, 5)$	$(-10, -5, 5)$	$(-10, 0, 5)$	$(-10, 5, 5)$	$(-10, 10, 5)$
	$(-5, -10, 5)$	$(-5, -5, 5)$	$(-5, 0, 5)$	$(-5, 5, 5)$	$(-5, 10, 5)$
	$(0, -10, 5)$	$(0, -5, -10)$	$(0, 0, 15)$	$(0, 5, -10)$	$(0, 10, 5)$
	$(5, -10, 5)$	$(5, -5, 5)$	$(5, 0, 5)$	$(5, 5, 5)$	$(5, 10, 5)$
	$(10, -10, 5)$	$(10, -5, 5)$	$(10, 0, 5)$	$(10, 5, 5)$	$(10, 10, 5)$

Así en este caso

$$S(u, v) = \sum_{i=0}^3 \sum_{j=0}^2 N_{i,3}(u)N_{j,2}(v)P_{i,j}$$

y tomemos los vectores de nodos

$$U = \{0, 0, 0, \frac{1}{2}, 1, 1, 1, 1\}$$

$$V = \{0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1\}$$

La superficie se muestra en la Figura (3.8).

3.4.1. Propiedades.

- i) **Soporte local.** $N_{i,p}(u)N_{j,q}(v) \neq 0$ si $u \in (u_i, u_{i+p+1})$ y $v \in (v_j, v_{j+q+1})$.
- ii) **No negatividad:** $N_{i,p}(u)N_{j,q}(v) \geq 0$ para toda i, j, p, q, u, v .
- iii) **Partición de la unidad:** $\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(v) = 1$ para $u \in (u_i, u_{i+p+1})$ y $v \in (v_j, v_{j+q+1})$.
- iv) $N_{0,p}(a)N_{0,q}(a) = N_{0,p}(a)N_{m,q}(b) = N_{n,p}(b)N_{0,q}(a) = N_{n,p}(b)N_{m,q}(b) = 1$.
- v) Si $n = p, m = q, U = \{\underbrace{a, \dots, a}_{p+1}, \underbrace{b, \dots, b}_{p+1}\}$ y $V = \{\underbrace{a, \dots, a}_{q+1}, \underbrace{b, \dots, b}_{q+1}\}$ entonces $N_{i,p}(u)N_{j,q}(v) = B_{i,p}(u)B_{j,q}(v) \forall i, j, p, q, u, v$.
- vi) La superficie interpola los cuatro puntos de control de las esquinas. Es decir $S(a, a) = P_{0,0}$, $S(b, a) = P_{n,0}$, $S(a, b) = P_{0,m}$ y $S(b, b) = P_{n,m}$. Esta propiedad se tiene por iv.

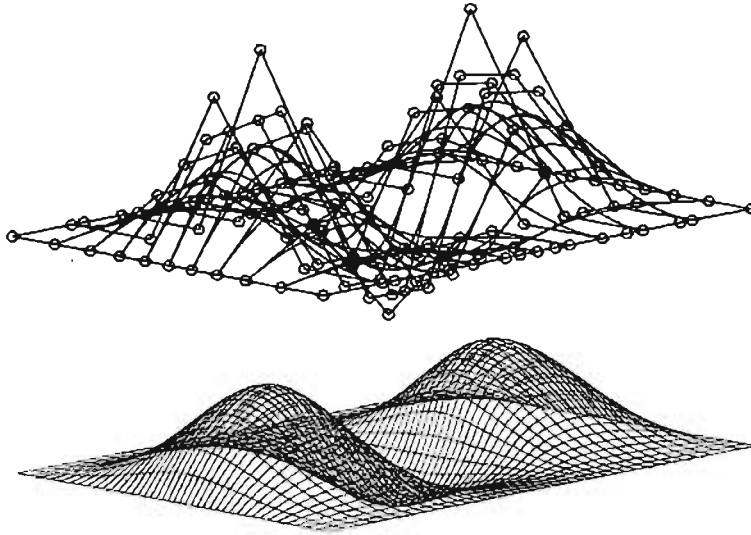


Figura 3.8: La Figura superior es la superficie B-spline y los puntos de control las isocurvas, en la inferior la superficie sin puntos de control.

- vii) **Propiedad fuerte de la envoltura convexa.** Si $u \in [u_{i_0}, u_{i_0+1})$ y $v \in [v_{j_0}, v_{j_0+1})$ entonces $S(u, v)$ está contenido en la envoltura convexa de los puntos de control $\{P_{i,j}\}$, con $i_0 - p \leq i \leq i_0$ y $j_0 - q \leq j \leq j_0$.
- viii) **Invarianza afín.** Una transformación afín es aplicada a la superficie aplicándola a sus puntos de control.
- ix) Si $n = p$ y $m = q$, $U = \underbrace{\{a, \dots, a\}}_{p+1}, \underbrace{\{b, \dots, b\}}_{p+1}$ y $V = \underbrace{\{a, \dots, a\}}_{p+1}, \underbrace{\{b, \dots, b\}}_{p+1}$ entonces $S(u, v)$, es una superficie de Bézier integral.
- x) **Modificaciones locales.** Si algún punto de control $P_{i,j}$ es un modificado, esto afecta a la superficie sólo en el rectángulo $[u_i, u_{i+p+1}) \times [v_j, v_{j+p+1})^3$. Un ejemplo de esto se presenta en la Figura 3.9.
- xi) **Diferenciabilidad.** $S(u, v)$ es $(p - k)$ veces diferenciable en un nodo interior u , de multiplicidad k y es $(q - k)$ veces diferenciable en un nodo v de multiplicidad k .
- xii) **Repetición de puntos de control.** Es posible utilizar puntos de control que coincidan, lo cual permite obtener discontinuidades visuales. Un ejemplo se presenta en la Figura 3.10.
- xiii) **Derivada.** Notemos que

$$\begin{aligned}
 S_u(u, v) &= \frac{\partial}{\partial u} S(u, v) = \sum_{j=0}^m N_{j,q}(v) \left(\frac{\partial}{\partial u} \sum_{i=0}^n N_{i,p}(u) P_{i,j} \right) \\
 &= \sum_{j=0}^m N_{j,q}(v) \left(\frac{\partial}{\partial u} C_j(u) \right)
 \end{aligned} \tag{3.10}$$

³En donde $A \times B = \{(x, y) | x \in A \text{ y } y \in B\}$.

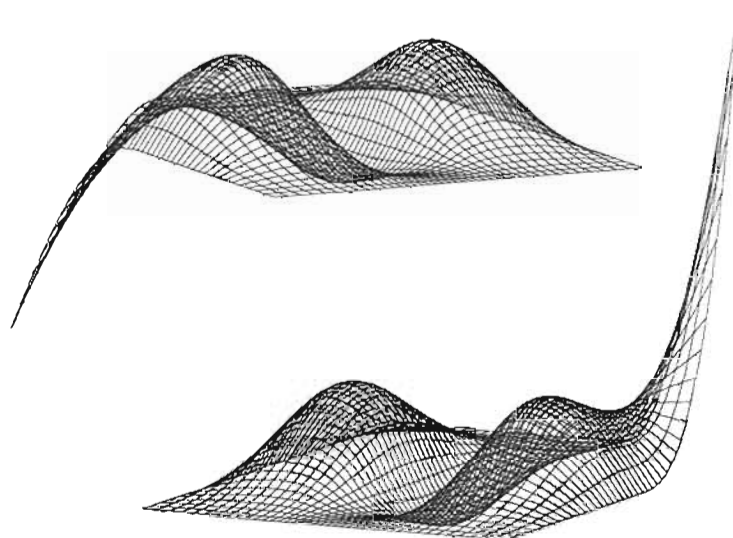


Figura 3.9: La superficie sólo se altera en un rectángulo al modificarse un punto de control.

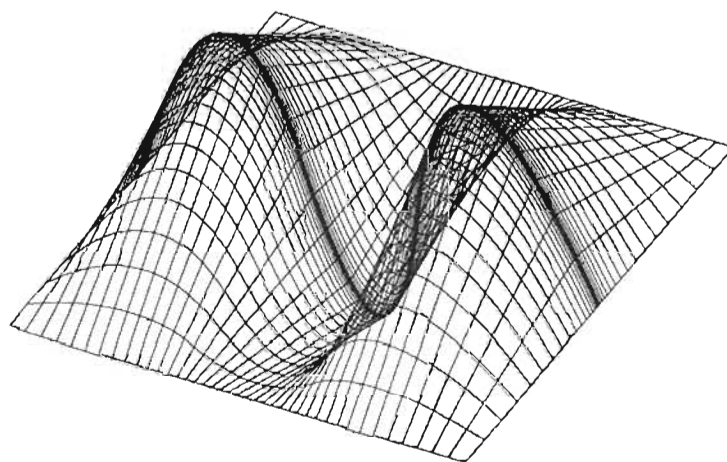


Figura 3.10: La repetición de puntos de control puede causar discontinuidades visuales en la superficie.

3.4. SUPERFICIES B-SPLINE.

donde $C_j(u) = \sum_{i=0}^n N_{i,p}(u)P_{i,j}$ para $j = 0, \dots, m$, son curva B-spline. Ahora aplicando la ecuación (2.33), a cada $C_j(u)$ y sustituyendo en (3.10) tenemos

$$S_u(u, v) = \sum_{i=0}^{n-1} \sum_{j=0}^m N_{i,p-1}(u)N_{j,q}(v)P_{i,j}^{(1,0)} \quad (3.11)$$

donde

$$P_{i,j}^{(1,0)} = p \frac{P_{i+1,j} - P_{i,j}}{u_{i+p+1} - u_{i+1}}$$

$$U^{(1)} = \left\{ \underbrace{a_1, \dots, a_p}_{p}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{b_1, \dots, b_p}_{p} \right\}$$

$$V^{(0)} = V$$

Análogamente

$$S_v(u, v) = \sum_{i=0}^n \sum_{j=0}^{m-1} N_{i,p}(u)N_{j,q-1}(v)P_{i,j}^{(0,1)} \quad (3.12)$$

donde

$$P_{i,j}^{(0,1)} = q \frac{P_{i,j+1} - P_{i,j}}{v_{j+q+1} - v_{j+1}}$$

$$U^{(0)} = U$$

$$V^{(1)} = \left\{ \underbrace{a_1, \dots, a_q}_q, v_{q+1}, \dots, v_{s-q-1}, \underbrace{b_1, \dots, b_q}_q \right\}$$

Ahora aplicando (3.11) y (3.12)

$$S_{u,v} = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} N_{i,p-1}(u)N_{j,q-1}(v)P_{i,j}^{(1,1)}$$

donde

$$P_{i,j}^{(1,1)} = q \frac{P_{i,j+1}^{(1,0)} - P_{i,j}^{(1,0)}}{v_{j+q+1} - v_{j+1}}$$

$U^{(1)}$ y $V^{(1)}$ como antes.

En general tenemos que

$$\frac{\partial^{k+l}}{\partial^k u \partial^l v} S(u, v) = \sum_{i=0}^{n-k} \sum_{j=0}^{m-l} N_{i,p-k}(u)N_{j,q-l}(v)P_{i,j}^{(k,l)}$$

donde

$$P_{i,j}^{(k,l)} = (q-l+1) \frac{P_{i,j+1}^{(k,l-1)} - P_{i,j}^{(k,l-1)}}{v_{j+q+1} - v_{j+l}}$$

$$U^{(k)} = \{ \underbrace{a, \dots, a}_{p+1-k}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{b, \dots, b}_{p+1-k} \}$$

$$V^{(l)} = \{ \underbrace{a, \dots, a}_{q+1-l}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{b, \dots, b}_{q+1-l} \}$$

3.5. Superficies NURBS.

Después de todo el trabajo previo nos encontramos en el punto en el cual es posible definir de forma muy clara y concisa las superficies NURBS. Las cuales son una generalización de todas las superficies estudiadas en el capítulo, y es fácil probar que las implicaciones que se muestran en la Figura 3.11.

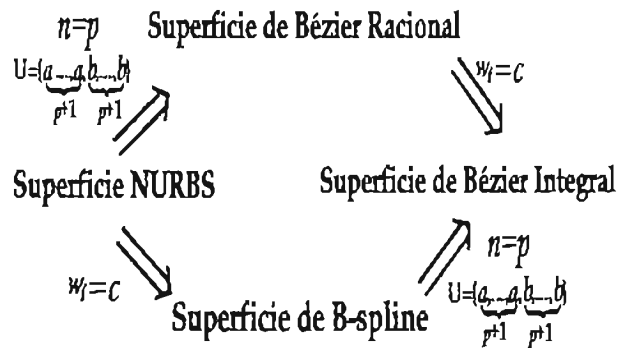


Figura 3.11: Las curvas NURBS son una generalización de todas las curvas presentadas en este capítulo.

Definición 18 Una superficie NURBS de grado p en dirección u y grado q en dirección v es una función racional por piezas bivariada de la forma

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}} \quad 0 \leq u, v \leq 1 \quad (3.13)$$

Los puntos $\{P_{i,j}\}$ forman una red bidireccional de puntos de control, los $\{w_{i,j}\}$ son los pesos y $\{N_{i,p}(u)\}$ y $\{N_{j,q}(v)\}$ son dos sucesiones de funciones base B-spline sobre los vectores

$$U = \{ \underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{b, \dots, b}_{p+1} \}$$

$$V = \{ \underbrace{a, \dots, a}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{b, \dots, b}_{q+1} \}$$

donde $r = n + p + 1$ y $s = m + q + 1$, respectivamente.

Si escribimos

$$R_{i,j;p,q}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{k=0}^n \sum_{l=0}^m N_{k,p}(u) N_{l,q}(v) w_{k,l}} \quad (3.14)$$

la ecuación (3.13) toma la forma mas compacta

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j;p,q}(u, v) P_{i,j} \quad (3.15)$$

Escribiremos $R_{i,j}(u, v)$ en vez de $R_{i,j;p,q}(u, v)$.

3.5.1. Propiedades.

- i) **No negatividad;** $R_{i,j}(u, v) \geq 0$ para toda $(u, v) \in [a, b] \times [a, b]$.
- ii) **Partición de la unidad;** $\sum_{i=0}^p \sum_{j=0}^q R_{i,j}(s, t) = 1$, para toda $(u, v) \in [a, b] \times [a, b]$.
- iii) $R_{0,0}(0; 0) = R_{p,0}(1, 0) = R_{0,q}(0, 1) = R_{p,q}(1, 1) = 1$.
- iv) **Diferenciabilidad de $R_{i,j}(u, v)$.** En un nodo u (o v) interior de multiplicidad k , $R_{i,j}(u, v)$ es $p - k$ (o $q - k$) veces diferenciable en dirección u (o v).
- v) Si $w_{i,j} = c$ con $c \neq 0$ constante para toda i, j , entonces $R_{i,p}(s, t) = N_{i,p}(s)N_{j,q}(t)$.
- vi) **Interpolación de los cuatro puntos de las esquinas.** $S(i, j) = P_{i,j}$ para $i, j = 0, 1$. Se tiene por III.
- vii) **Propiedad fuerte de envoltura convexa.** Si $w_{i,j} \geq 0$ para toda i, j . Si $(u, v) \in [u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, entonces $S(u, v)$ está contenida en el la envoltura convexa formada por los puntos de control, $P_{i,j}$, $i_0 - p \leq i \leq i_0$ y $j_0 - q \leq j \leq j_0$.
- viii) **Invarianza afín.** Una transformación afín es aplicada a $S(u, v)$, haciéndolo a sus puntos de control. Se tiene por II.
- ix) **Invarianza bajo transformaciones de parámetros afines.**
- x) La superficie B-spline es un caso particular de la superficie de NURBS. Esto es cierto por v.
- xi) Si $P_{i,j}$ es movido o $w_{i,j}$ cambiado, esto afecta a la superficie sólo en el rectángulo $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$.
- xii) **Diferenciabilidad de $S(u, v)$.** La superficie NURBS, es $p - k$ (o $q - k$) veces diferenciable con respecto al nodo u (o v) de multiplicidad k .
- xiii) **Derivada.** Es posible calcular la derivada de esta función, tomando

$$f = \sum_{i=0}^n \sum_{j=0}^m w_{i,j} N_{i,p}(u) N_{j,q}(v) P_{i,j}$$

$$g = \sum_{i=0}^n \sum_{j=0}^m w_{i,j} N_{i,p}(u) N_{j,q}(v)$$

con lo que $S(u, v) = F = \frac{f}{g}$ y entonces podemos usar (3.8).

A continuación describiremos la construcción de algunas superficies estándares, de la geometría clásica usando NURBS.

Ejemplo 22 Cilindros generales. Un cilindro general, es una superficie que resulta de “barrer” una curva, a cierta distancia a través de un vector. Es decir, si denotamos a este vector (unitario) por W y d la distancia. Dada una curva NURBS

$$C(u) = \sum_{i=0}^n R_{i,p}(u)P_i$$

la superficie NURBS $S(u, v)$, que representa un cilindro general con $C(u)$, a través de W y distancia d , denotando el parámetro de barrido con v ($0 \leq v \leq 1$), debe satisfacer las siguientes condiciones

1. Para \bar{u} fijo, $S(\bar{u}, v)$ es un segmento de línea recta de $C(\bar{u})$ a $C(\bar{u}) + dW$.
2. Para \bar{v} fijo,

$$S(u, \bar{v}) = C(u) + \bar{v}dW = \sum_{i=0}^n R_{i,p}(P_i + \bar{v}dW)$$

Así el cilindro debe ser de la forma

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^1 R_{i,p}(u)R_{j,1}(v)P_{i,j} \tag{3.16}$$

con vectores de nodos U y V en donde $V = \{0, 0, 1, 1\}$ y U es el vector de nodos de $C(u)$. Los puntos de control están dados por

$$\begin{aligned} P_{i,0} &= P_i \\ P_{i,1} &= P_i + dW \end{aligned} \tag{3.17}$$

y los pesos

$$w_{i,0} = w_{i,1} = w_i \tag{3.18}$$

Por ejemplo en la Figura 3.12 se presenta un cilindro general obtenido al barrer la curva NURBS cuadrática en forma de trébol, cuyos puntos de control son

<i>Puntos de control</i>		
i		
	(-10, -10, 0)	(-30, -40, 0)
	(-30, 40, 0)	(-30, -40, 0)
	(30, 40, 0)	(-10, -10, 0)
	(10, 10, 0)	(-30, -30, 0)
	(30, 30, 0)	(-30, 30, 0)
	(30, -30, 0)	(-10, 10, 0)
	(10, -10, 0)	

vector de nodos $U = \{0, 0, 0, 1, 2, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, 8\}$ y $w_i = 1$ para toda i .

Entonces tomando $d = 10$, $W = (0, 0, 10)$ y calculando (3.17), (3.18), la superficie 3.16 está dada por

$$S(u, v) = \sum_{i=0}^{12} \sum_{j=0}^1 R_{i,j,2,1}(u, v)P_{i,j}$$

En la Figura 3.13, se presenta una superficie que representa un domo. Los puntos de la curva NURBS son

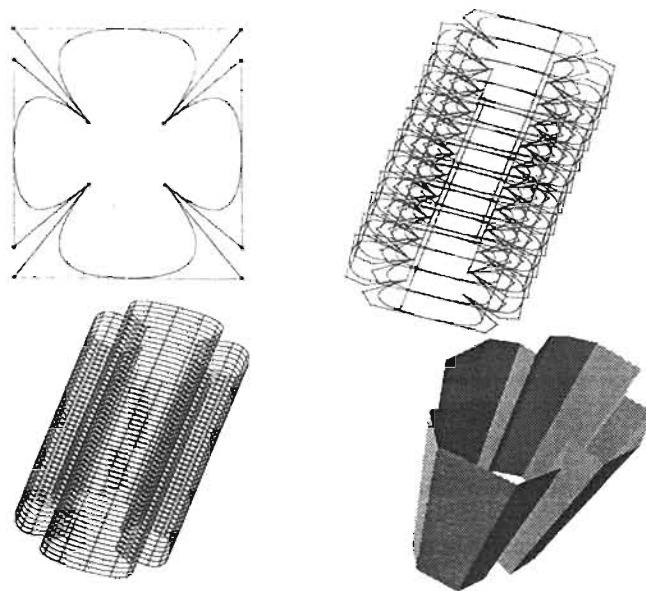


Figura 3.12: Curva NURBS cuadrática en forma de trébol, con la que se forma el cilindro general.

Puntos de control	
<i>i</i>	
	(20, 15, 10)
	(20, -10, 20)
	(20, -5, 10)
	(20, 0, 20)
	(20, 5, 10)
	(20, 10, 20)
	(20, 15, 10)

en este caso $U = \{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$, $w_i = 1$ para toda i , $d = 10$ y $W = (-10, 0, 0)$. Y así la superficie está dada por

$$S(u, v) = \sum_{i=0}^6 \sum_{j=0}^1 R_{i,j;3,1}(u, v) P_{i,j}$$

La implementación de los programas con los que se obtuvieron estas superficies se incluye en este trabajo de tesis.

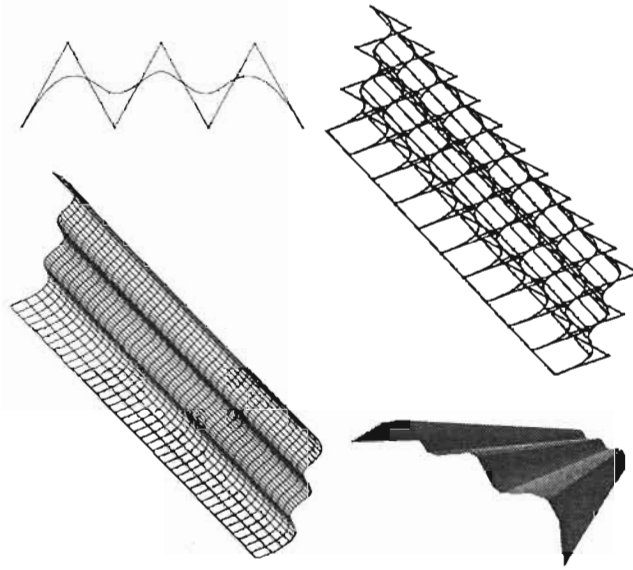


Figura 3.13: Curva NURBS cúbica, con la que se forma un cilindro general en forma de domo.

Ejemplo 23 Superficies de revolución.

Dado una curva NURBS $C(v) = \sum_{j=0}^m R_{j,q}(v)P_j$ con vector de nodos V , llamada *generatriz* que supondremos está contenida en el plano xz . No es difícil encontrar la representación NURBS de una superficie de revolución obtenida al girar $C(v)$ 360° al rededor del eje de las z . Ya que notemos ésta debe cumplir que

1. para \bar{u} fija, $S(\bar{u}, v)$ deber ser una curva $C(v)$, rotada algún ángulo al rededor del eje z .
2. para \bar{v} fija, $S(u, \bar{v})$ deber ser un círculo completo en un plano perpendicular al eje z , con centro en este eje.

Entonces, podemos usar los 9 puntos de control como los que representaron la una circunferencia del ejemplo 15⁴, y tomar $U = \{0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1\}$, con pesos $\{1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1\}$. Así la superficie requerida tendrá la forma

$$S(u, v) = \sum_{i=0}^8 \sum_{j=0}^m R_{i,j,2,q}(u, v)P_{i,j} \tag{3.19}$$

en donde sus vectores de nodos, son U y V , los puntos de control se obtiene como se muestra en la Figura 3.14 y los pesos son $w_{0,j} = w_j, w_{1,j} = (\frac{\sqrt{2}}{2})w_j, w_{2,j} = w_j, w_{3,j} = (\frac{\sqrt{2}}{2})w_j, \dots, w_{8,j} = w_j$.

En la Figura (3.15), se obtiene una esfera, haciendo girar al rededor del eje de las z , una curva NURBS que representa la mitad de una circunferencia. Los puntos de control de la superficie son

⁴También podríamos usar la representación de sólo 7 puntos.

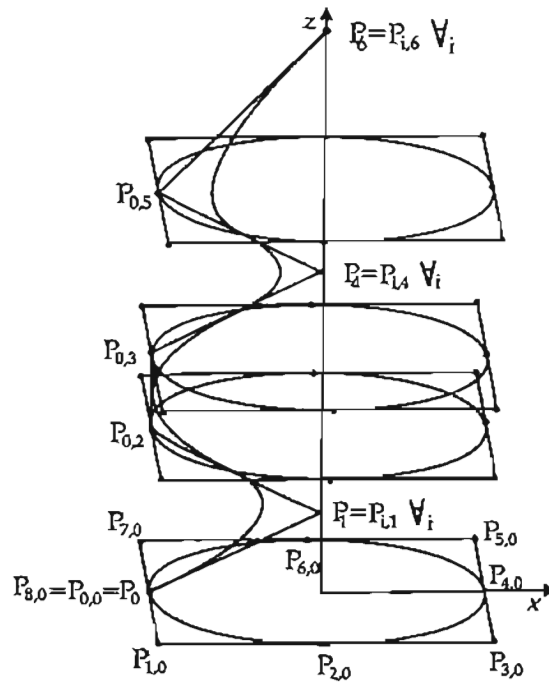


Figura 3.14: Selección de Puntos de control para una superficie de revolución NURBS.

Puntos de control					
i/j					
	(0, 0, 30)	(0, -30, 30)	(0, -30, 0)	(0, -30, -30)	(0, 0, -30)
	(0, 0, 30)	(30, -30, 30)	(30, -30, 0)	(30, -30, -30)	(0, 0, -30)
	(0, 0, 30)	(30, 0, 30)	(30, 0, 0)	(30, 0, -30)	(0, 0, -30)
	(0, 0, 30)	(30, 30, 30)	(30, 30, 0)	(30, 30, -30)	(0, 0, -30)
	(0, 0, 30)	(0, 30, 30)	(0, 30, 0)	(0, 30, -30)	(0, 0, -30)
	(0, 0, 30)	(-30, 30, 30)	(-30, 30, 0)	(-30, 30, -30)	(0, 0, -30)
	(0, 0, 30)	(-30, 0, 30)	(-30, 0, 0)	(-30, 0, -30)	(0, 0, -30)
	(0, 0, 30)	(-30, -30, 30)	(-30, -30, 0)	(-30, -30, -30)	(0, 0, -30)
	(0, 0, 30)	(0, -30, 30)	(0, -30, 0)	(0, -30, -30)	(0, 0, -30)

para $i = 0, \dots, 8$. Con $V = \{0, 0, 0, \frac{1}{2}, 1, 1, 1\}$ y la matriz de pesos de (9×5) es

$$W = \begin{pmatrix} 1 & \frac{\sqrt{2}}{2} & 1 & \frac{\sqrt{2}}{2} & 1 \\ \frac{\sqrt{2}}{2} & \frac{1}{2} & \frac{\sqrt{2}}{2} & \frac{1}{2} & \frac{\sqrt{2}}{2} \\ 1 & \frac{\sqrt{2}}{2} & 1 & \frac{\sqrt{2}}{2} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \frac{\sqrt{2}}{2} & 1 & \frac{\sqrt{2}}{2} & 1 \end{pmatrix}$$

Y así la superficie de revolución (3.19), tiene la forma

$$S(u, v) = \sum_{i=0}^8 \sum_{j=0}^4 R_{i,j,2,2}(u, v) P_{i,j}$$

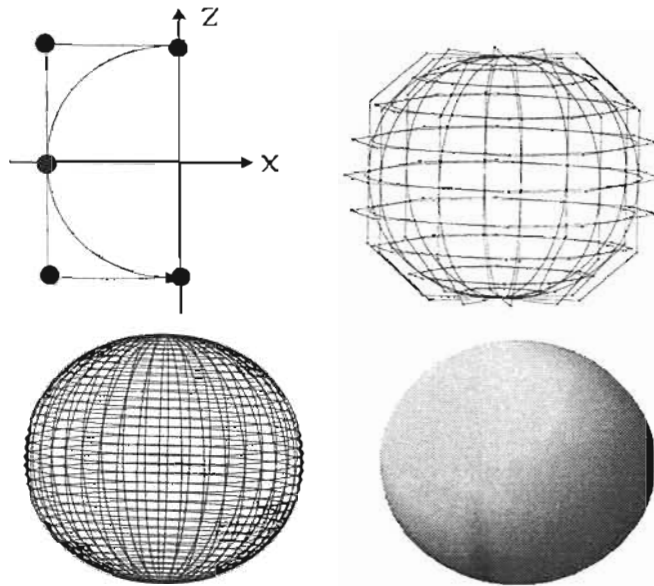


Figura 3.15: Esfera obtenida al girar un arco circular al rededor del eje de las z.

En la Figura 3.16, se presenta una superficie de revolución usando una curva NURBS, la cual tiene puntos de control

Puntos de control							
i/j							
	(10, 0, 0)	(0, 0, 5)	(10, 0, 10)	(10, 0, 15)	(0, 0, 20)	(10, 0, 25)	(0, 0, 35)
	(10, 10, 0)	(0, 0, 5)	(10, 10, 10)	(10, 10, 15)	(0, 0, 20)	(10, 10, 25)	(0, 0, 35)
	(0, 10, 0)	(0, 0, 5)	(0, 10, 10)	(0, 10, 15)	(0, 0, 20)	(0, 10, 25)	(0, 0, 35)
	(-10, 10, 0)	(0, 0, 5)	(-10, 10, 10)	(-10, 10, 15)	(0, 0, 20)	(-10, 10, 25)	(0, 0, 35)
	(-10, 0, 0)	(0, 0, 5)	(-10, 0, 10)	(-10, 0, 15)	(0, 0, 20)	(-10, 0, 25)	(0, 0, 35)
	(-10, -10, 0)	(0, 0, 5)	(-10, -10, 10)	(-10, -10, 15)	(0, 0, 20)	(-10, -10, 25)	(0, 0, 35)
	(0, -10, 0)	(0, 0, 5)	(0, -10, 10)	(0, -10, 15)	(0, 0, 20)	(0, -10, 25)	(0, 0, 35)
	(10, -10, 0)	(0, 0, 5)	(10, -10, 10)	(10, -10, 15)	(0, 0, 20)	(10, -10, 25)	(0, 0, 35)
	(10, 0, 0)	(0, 0, 5)	(10, 0, 10)	(10, 0, 15)	(0, 0, 20)	(10, 0, 25)	(0, 0, 35)

con vector de nodos $V = \{0, 0, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1, 1, 1\}$ y matriz de pesos (de 9×7) dada por

$$W = \begin{pmatrix} 1 & \dots & 1 \\ \frac{\sqrt{2}}{2} & \dots & \frac{\sqrt{2}}{2} \\ \vdots & & \vdots \\ \frac{\sqrt{2}}{2} & \dots & \frac{\sqrt{2}}{2} \\ 1 & \dots & 1 \end{pmatrix}$$

y así (3.19) para este caso es

$$S(u, v) = \sum_{i=0}^8 \sum_{j=0}^6 R_{i,j,2,2}(u, v) P_{i,j}$$

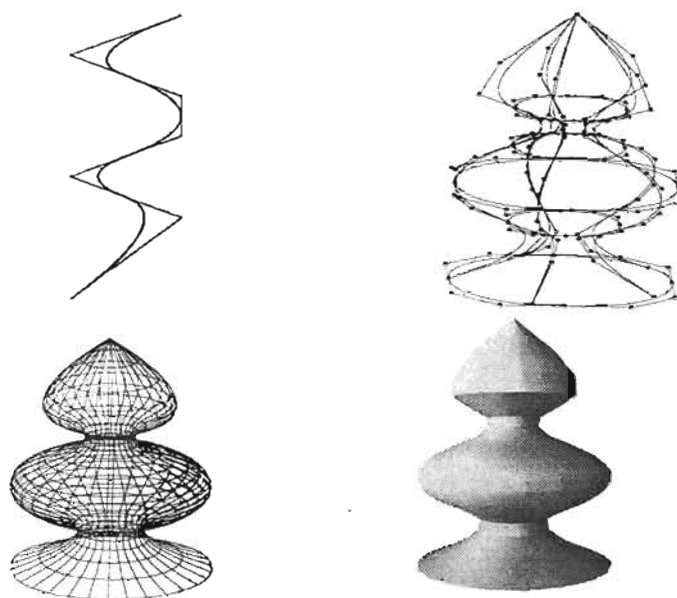


Figura 3.16: Superficie libre obtenida al girar la curva NURBS cuadrática al rededor del eje de las z .

Ejemplo 24 Superficies con factor de escala no uniforme.

Una herramienta simple pero útil en la construcción de superficies, es la aplicación de factores de escala, a continuación estudiaremos la representación NURBS de este tipo de superficies.

Dado un punto en el espacio $P = (p_1, p_2, p_3)$ y un conjunto de reales $F = \{f_1, f_2, f_3\}$. Se dice que \bar{P} tiene un factor de escala F al rededor del origen si

$$\bar{P} = (f_1 p_1, f_2 p_2, f_3 p_3) \quad (3.20)$$

Análogamente, si $C = (c_1, c_2, c_3)$ es un punto en el espacio, \bar{P} tiene un factor de escala F al rededor de C si

$$\bar{P} = (\bar{p}_1, \bar{p}_2, \bar{p}_3) \quad (3.21)$$

en donde

$$\bar{p}_i = f_i p_i + (1 - f_i) c_i \quad i = 1, 2, 3.$$

Los factores de escala, pueden ayudarnos en la construcción de interesantes superficies NURBS, ya que éstos "expanden o contraen" sus puntos en los tres ejes coordenados.

Por ejemplo, es posible construir una elipsoide con centro en el origen de manera muy simple aplicando ciertos factores de escala a los puntos de control de la esfera construida en el ejemplo anterior.

Para esto, recordemos que la ecuación de una elipsoide con centro en el origen y semiejes a , b y c , está dada por

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

en donde la intersección de esta superficie con el plano xy , zx y zy , resultan las elipses que se presentan en la Figura 3.17, esto nos indica que al tomar $F = \{a, b, c\}$, los puntos de control de la esfera resultarán, en los respectivos para el elipsoide. En la Figura 3.18, se presenta el elipsoide resultante tomando $F = \{1, 1, 2\}$.

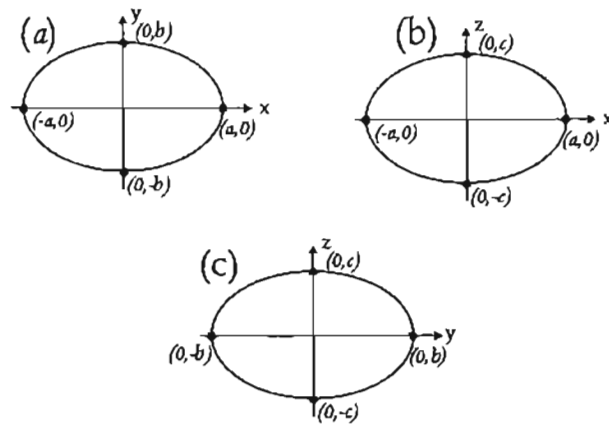


Figura 3.17: Elipse que resulta de la intersección de la elipsoide y el plano (a) yx , (b) zx , (c) zy .

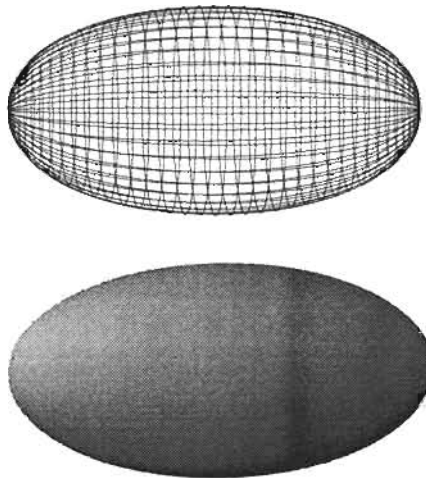


Figura 3.18: Elipsoide, que resulta de aplicar el factor de escala $F = \{1, 1, 2\}$, a los puntos de control de la esfera del ejemplo anterior.

En general, cuando los puntos estén en el espacio de tres dimensiones, se les puede aplicar las ecuaciones (3.20) y (3.21) directamente, dejando sin cambio los pesos. Cuando estos sean de la forma $P^w = (wx, wy, wz, w)$, sólo las tres primeras coordenadas cambian, es decir para (3.20) tendremos $\bar{P}^w = (f_1wx, f_2wy, f_3wz, w)$ y para (3.21), $\bar{P}^w = (f_1wx + (1 - f_1)wc_1, f_2wy + (1 - f_2)wc_2, f_3wz + (1 - f_3)wc_3)$.

Capítulo 4

Técnicas básicas para la construcción de superficies.

En este capítulo estudiaremos varios algoritmos y técnicas importantes en la teoría de las curvas y superficies NURBS, las cuales nos serán de gran utilidad para la construcción de superficies más avanzadas en el siguiente.

Así discutiremos los algoritmos de inserción y refinamiento, la eliminación de nodos y la elevación de grado y finalizaremos con la interpolación global de curvas y superficies.

4.1. Inserción de nodos.

El algoritmo de inserción de nodos, es uno de los más importantes para las curvas y superficies NURBS, ya que nos permite entre otras cosas, una evaluación sencilla, subdivisión¹ y la posibilidad de agregar puntos de control adicionales para incrementar la flexibilidad de éstas.

Sea una curva NURBS

$$C^w(u) = \sum_{i=0}^n N_{i,p}(u) P_i^w \quad (4.1)$$

con vector de nodos $U = \{u_0, \dots, u_m\}$. Así supongamos que se inserta un nodo $u \in [u_k, u_{k+1})$ (para $k \in \{0, \dots, m\}$) en U , obteniendo el nuevo vector de nodos

$$\bar{U} = \{\bar{u}_0 = u_0, \dots, \bar{u}_k = u_k, \bar{u}_{k+1} = \bar{u}, \bar{u}_{k+2} = u_{k+1}, \dots, \bar{u}_{m+1} = u_m\}$$

entonces el proceso de inserción de nodos se refiere a obtener puntos $Q_i^w \in \mathbb{R}^3$ tales que

$$C^w(u) = \sum_{i=0}^{n+1} \bar{N}_{i,p}(u) Q_i^w \quad (4.2)$$

en donde $\bar{N}_{i,p}(u)$ (para $i = 0, \dots, n+1$) son las funciones B-spline definidas en \bar{U} .

Así notemos, que como $u \in [u_k, u_{k+1})$ y por la propiedad de soporte local de la funciones B-spline (ver sección 1.2.5), podemos escribir (4.1) y (4.2) como

¹De forma equivalente a como se realiza para las curvas de Bézier.

$$C^w(u) = \sum_{i=k-p}^k N_{i,p}(u) P_i^w \quad (4.3)$$

$$C^w(u) = \sum_{i=k-p}^{k+1} \bar{N}_{i,p}(u) Q_i^w \quad (4.4)$$

Ahora, igualándolas a cero y sustituyendo (4.1), tenemos

$$\begin{aligned} \sum_{i=0}^{k-p-1} (P_i^w - Q_i^w) N_{i,p}(u) &= 0 \\ \sum_{i=k+1}^n (P_i^w - Q_i^w) N_{i,p}(u) &= 0 \end{aligned}$$

y por la independencia lineal de las funciones B-spline (ver capítulo 1 sección 1.2.6)

$$P_i^w = Q_i^w \quad \text{para } i = 0, \dots, k-p-1 \quad (4.5)$$

$$P_j^w = Q_j^w \quad \text{para } j = k+1, \dots, n \quad (4.6)$$

Por otro lado, del teorema 12 del capítulo 1, sabemos que es posible expresar las funciones $N_{i,p}(u)$ en términos de $\bar{N}_{i,p}(u)$ en la siguiente manera

$$N_{i,p}(u) = \frac{\bar{u} - \bar{u}_i}{\bar{u}_{i+p+1} - \bar{u}_i} \bar{N}_{i,p}(u) + \frac{\bar{u}_{i+p+2} - \bar{u}}{\bar{u}_{i+p+2} - \bar{u}_{i+1}} \bar{N}_{i+1,p}(u) \quad (4.7)$$

entonces sustituyendo (4.7) en (4.3) e igualando con (4.4) tenemos que (escribiremos \bar{N}_i en vez de $\bar{N}_{i,p}(u)$)

$$\begin{aligned} & \left(\frac{\bar{u} - \bar{u}_{k-p}}{\bar{u}_{k+1} - \bar{u}_{k-p}} \bar{N}_{k-p} + \frac{\bar{u}_{k+2} - \bar{u}}{\bar{u}_{k+2} - \bar{u}_{k-p+1}} \bar{N}_{k-p+1} \right) P_{k-p}^w \\ & + \left(\frac{\bar{u} - \bar{u}_{k-p+1}}{\bar{u}_{k+2} - \bar{u}_{k-p+1}} \bar{N}_{k-p+1} + \frac{\bar{u}_{k+3} - \bar{u}}{\bar{u}_{k+3} - \bar{u}_{k-p+2}} \bar{N}_{k-p+2} \right) P_{k-p+1}^w \\ & \vdots \\ & + \left(\frac{\bar{u} - \bar{u}_k}{\bar{u}_{k+p+1} - \bar{u}_k} \bar{N}_k + \frac{\bar{u}_{k+p+2} - \bar{u}}{\bar{u}_{k+p+2} - \bar{u}_{k+1}} \bar{N}_{k+1} \right) P_k^w \\ & = \bar{N}_{k-p} Q_{k-p}^w + \dots + \bar{N}_{k+1} Q_{k+1}^w \end{aligned}$$

Ahora factorizando y usando los nodos del vector U en vez de los de \bar{U}

$$\begin{aligned} 0 &= \bar{N}_{k-p} \left(Q_{k-p}^w - \frac{u_k - u_{k-p-1}}{u_k - u_{k-p-1}} P_{k-p}^w \right) \\ & + \bar{N}_{k-p+1} \left(Q_{k-p+1}^w - \frac{\bar{u} - u_{k-p+1}}{u_{k+1} - u_{k-p+1}} P_{k-p+1}^w - \frac{u_{k+1} - \bar{u}}{u_{k+1} - u_{k-p+1}} P_{k-p}^w \right) \\ & \vdots \\ & + \bar{N}_k \left(Q_k^w - \frac{\bar{u} - u_k}{u_{k+p} - u_k} P_k^w - \frac{u_{k+p} - \bar{u}}{u_{k+p} - u_k} P_{k-1}^w \right) + \bar{N}_{k+1} (Q_{k+1}^w - P_k^w) \end{aligned}$$

y de nuevo por la independencia lineal de las funciones B-spline

$$\begin{aligned}
 Q_{k-p}^w &= P_{k-p}^w \\
 Q_{k-p+1}^w &= \frac{\bar{u} - u_{k-p+1}}{u_{k+1} - u_{k-p+1}} P_{k-p+1}^w + \frac{u_{k+1} - \bar{u}}{u_{k+1} - u_{k-p+1}} P_{k-p}^w \\
 &\vdots \\
 Q_k^w &= \frac{\bar{u} - u_k}{u_{k+p} - u_k} P_k^w - \frac{u_{k+p} - \bar{u}}{u_{k+p} - u_k} P_{k-1}^w \\
 Q_{k+1}^w &= P_k^w
 \end{aligned} \tag{4.8}$$

Si denotamos

$$\alpha_i = \frac{\bar{u} - u_i}{u_{i+p} - u_i} \quad 1 - \alpha_i = \frac{u_{i+p} - \bar{u}}{u_{i+p} - u_i}$$

(4.8) se puede escribir como

$$Q_i^w = \alpha_i P_i^w + (1 - \alpha_i) P_{i-1}^w \tag{4.9}$$

para $i = k - p + 1, \dots, k$.

Además, notemos que usando (4.5) y (4.6) si definimos

$$\alpha_i = \begin{cases} 1 & i \leq k - p \\ \frac{\bar{u} - u_i}{u_{i+p} - u_i} & k - p + 1 \leq i \leq k \\ 0 & i \leq k + 1 \end{cases} \tag{4.10}$$

la ecuación (4.9) también es válida para todo $i \in \{0, \dots, n + 1\}$.

Ejemplo 25 Sea una curva NURBS $C^w(u)$, de grado $p = 3$ con puntos de control P_0^w, \dots, P_7^w y vector de nodos $U = \{0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 5, 5\}$. Entonces supongamos que deseamos obtener los puntos de control de curva que resulta de insertar un nodo $\bar{u} = 3.5$ en U .

Entonces los nuevos puntos están dados por (4.9), en donde en este caso $k = 6$ y

$$\alpha_i = \begin{cases} 1 & \text{si } i \leq 3 \\ \frac{\bar{u} - u_i}{u_{i+3} - u_i} & 4 \leq i \leq 6 \\ 0 & \text{si } i \geq 7 \end{cases} \tag{4.11}$$

con $\alpha_4 = \frac{5}{8}$, $\alpha_5 = \frac{1}{2}$ y $\alpha_6 = \frac{1}{4}$. En la Figura 4.1 se muestra los puntos de control antes y después de la inserción de \bar{u} .

□

Es de gran utilidad entender la relación geométrica entre los puntos de control Q_i^w , el vector de nodos U , y el nodo \bar{u} . Así notemos que de la ecuación (4.10), los valores de cada α_i (para $k - p + 1 \leq i \leq k$) representan la distancia relativa al intervalo $[u_i, u_{i+p}]$ entre u_i y \bar{u} . Por lo que de (4.9), cada Q_i^w se encuentra a una distancia de P_{i-1}^w y P_i^w , proporcional a la que existe de \bar{u} a u_i y u_{i+p} (ver Figura 4.2).

Esto además nos indica, que *los elementos del vector de nodos particionan el polígono de control*, es decir si L_i (para $i = 1, \dots, n$) representa el segmento de recta entre P_{i-1}^w y P_i^w y d_i su distancia, a cada L_i se le puede asociar los nodos u_{i+j} (para $j = 0, \dots, p$) de tal manera que si definimos

$$\lambda_i^j = \frac{u_{i+j+1} - u_{i+j}}{u_{i+p} - u_i} \quad \text{para } j = 0, \dots, p - 1$$

podemos decir que esta recta está formada por los segmentos L_i^j cuya distancia es $d_i^j = \lambda_i^j d_i$.

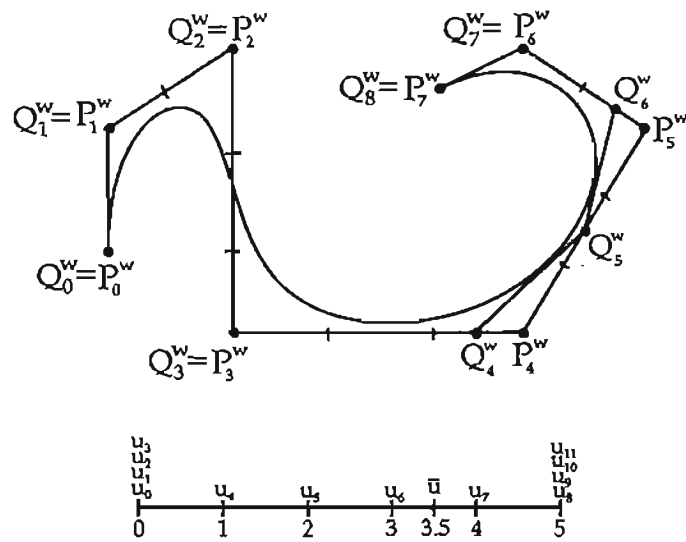


Figura 4.1: Inserción del nodo $\bar{u} = 3.5$, en una curva cúbica NURBS.

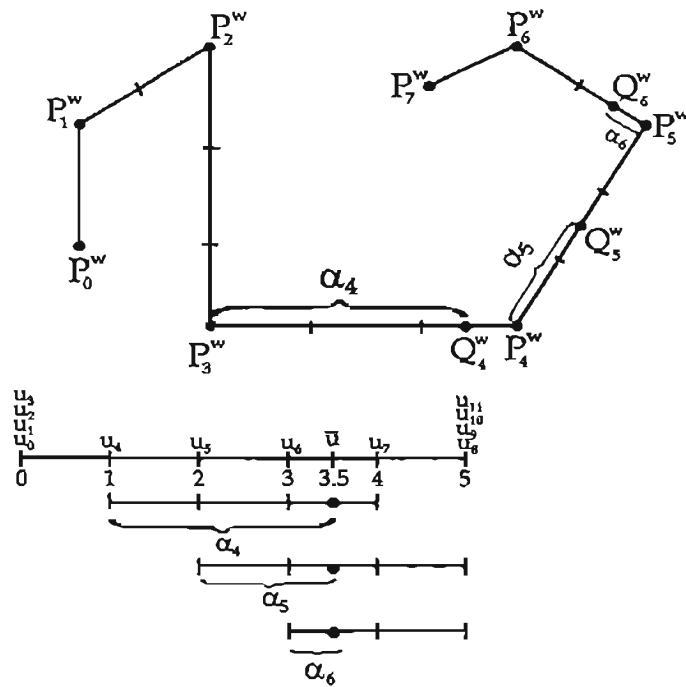


Figura 4.2: Las distancias que existen entre el nodo insertado y el nuevo punto de control son proporcionales.

En la Figura 4.3 se presentan los segmentos que subdividen al polígono de control del ejemplo anterior, notemos la relación de cada L_i y sus nodos asociados, también que $L_1 = L_1^2$ y $\lambda_1^0 = \lambda_1^1 = 0$ por lo que $d_1^0 = d_1^1 = 0$. Adicionalmente, observemos que los nuevos puntos de control cumplen que $Q_4 \in L_4^2$, $Q_5 \in L_5^1$ y $Q_6 \in L_6^0$.

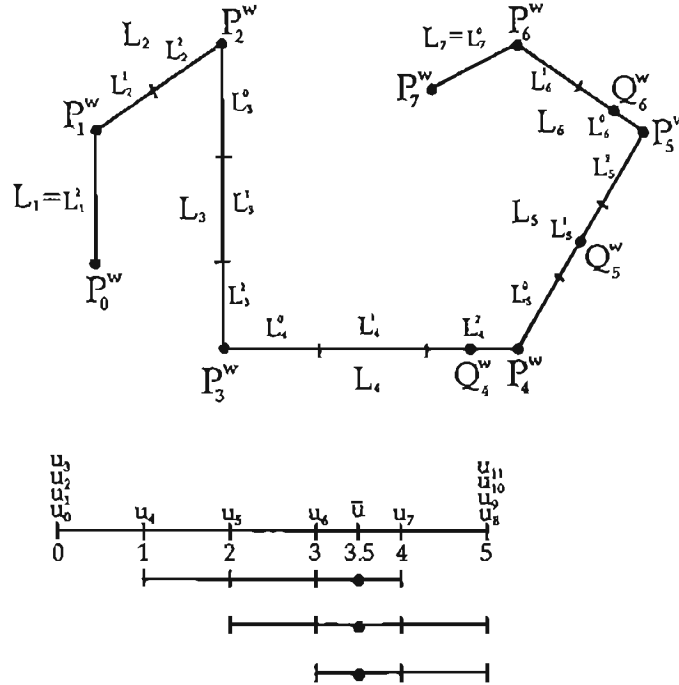


Figura 4.3: El polígono de control está particionado por el vector de nodos.

El mismo polígono de control pero con un vector de nodos no uniforme

$$U = \{0, 0, 0, 0, 1, 1.5, 2.3, 3.9, 4.2, 4.2, 4.2, 4.2\}$$

es particionado como se muestra en la Figura 4.4.

El siguiente ejemplo nos ayudará a estudiar los efectos que tiene en la curva la inserción repetida de un nodo.

Ejemplo 26 Tomemos la curva NURBS del ejemplo 25 (con \bar{u} ya insertado) con puntos $P_i^w = Q_i^w$ (para $i = 0, \dots, 8$) y vector $U = \{0, 0, 0, 0, 1, 2, 3, 3.5, 4, 5, 5, 5\}$ e insertemos de nuevo $\bar{u} = 3.5$.

Entonces para este caso $k = 7$, y los puntos dados por (4.9) tiene

$$\alpha_i = \begin{cases} 1 & \text{si } i \leq 4 & \text{a)} \\ \frac{\bar{u} - u_i}{u_{i+3} - u_i} & \text{si } 5 \leq i \leq 7 & \text{b)} \\ 0 & \text{si } i \geq 8 & \text{c)} \end{cases}$$

en donde $\alpha_5 = \frac{3}{4}$, $\alpha_6 = \frac{1}{4}$ y $\alpha_7 = 0$. Con éstos valores observemos que es posible cambiar el rango de i en b) y escribir $5 \leq i \leq 6$, y para c), $i \geq 7$, además si por un momento suponemos que nuestro vector de nodos U es el del ejemplo 25, la igualdad anterior es

$$\alpha_i = \begin{cases} 1 & \text{si } i \leq 4 \\ \frac{\bar{u} - u_i}{u_{i+2} - u_i} & \text{si } 5 \leq i \leq 6 \\ 0 & \text{si } i \geq 7 \end{cases} \quad (4.12)$$

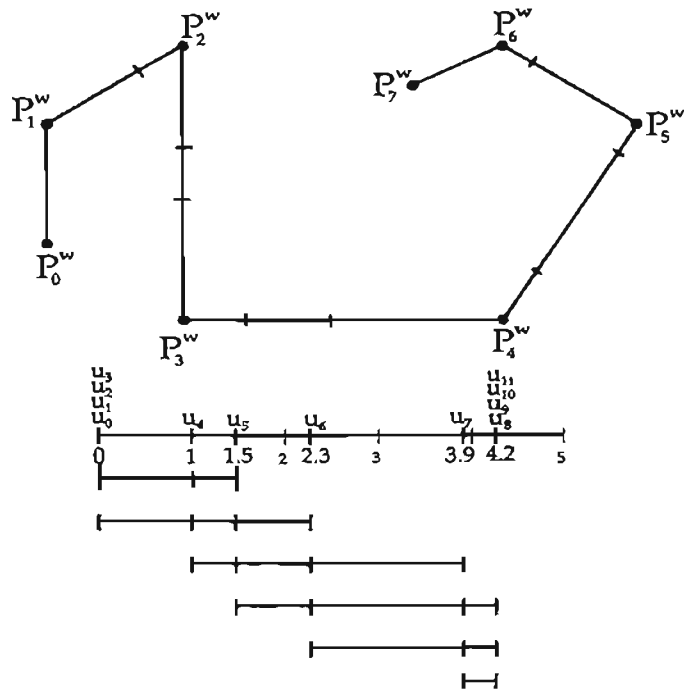


Figura 4.4: El polígono de control de la Figura anterior, pero con otro vector de nodos.

Los nuevos puntos de control Q_i^w se presentan en la Figura 4.5, notemos que para este caso $Q_5 \in L_3^2$ y $Q_6 \in L_6^1$ y $Q_7^w = P_6^w$.

Finalmente si reinsertamos una vez más $\bar{u} = 3.5$ pero ahora en

$$U = \{0, 0, 0, 0, 1, 2, 3.5, 3.5, 4, 5, 5, 5, 5\}$$

tenemos que $k = 8$, y los nuevos puntos de control dados por (4.9) tienen

$$\alpha_i = \begin{cases} 1 & \text{si } i \leq 5 \\ \frac{\bar{u} - u_i}{u_{i+3} - u_i} & \text{si } 6 \leq i \leq 8 \\ 0 & \text{si } i \geq 9 \end{cases} \quad (4.13)$$

con $\alpha_6 = \frac{1}{2}$, $\alpha_7 = \alpha_8 = 0$. Haciendo la observación y suposición anterior, tenemos que esta igualdad también se puede escribir como

$$\alpha_i = \begin{cases} 1 & \text{si } i \leq 5 \\ \frac{\bar{u} - u_i}{u_{i+1} - u_i} & \text{si } 6 \leq i \leq 6 \\ 0 & \text{si } i \geq 7 \end{cases} \quad (4.14)$$

Los puntos de control resultantes de esta inserción se presentan en la Figura 4.6, aquí $Q_6 \in L_6^2$. □

Siguiendo las ideas de los ejemplos anteriores, se deduce que la inserción de un nodo $\bar{u} \in [u_k, u_{k+1})$ de multiplicidad s (para $0 \leq s \leq p - 1$) en U , corta las esquinas del polígono de control de la curva NURBS (4.1), en los puntos $P_{k-p+1}, \dots, P_{k-s-1}$ y los $p - s$ nuevos puntos de control cumplen

$$Q_{k-j}^w \in L_{k-j}^j \text{ para } j = p - 1, \dots, s \quad (4.15)$$

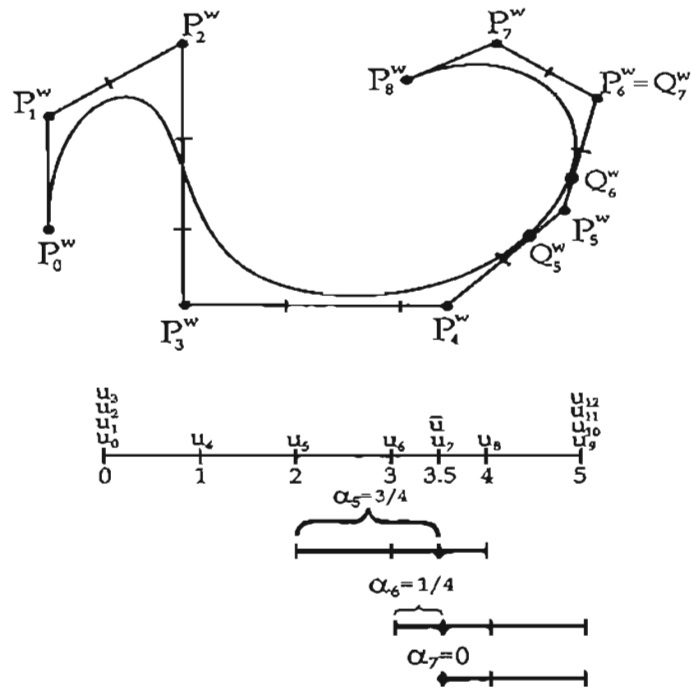


Figura 4.5: Inserción del nodo $\bar{u} = 3.5$ por segunda ocasión.

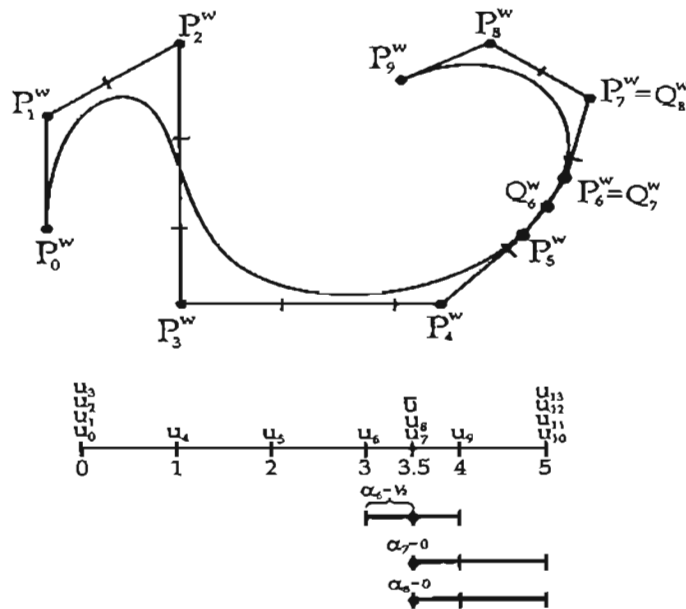


Figura 4.6: Inserción del nodo $\bar{u} = 3.5$ por tercera ocasión.

así en el ejemplo 25, las equinas cortadas tienen como puntos P_4^w, P_5^w (ver Figura 4.1) y en el ejemplo 26 el punto es P_5^w (ver Figura 4.5).

Si se continua con este proceso τ veces, (con $r + s \leq p$ no tiene sentido práctico hacerlo para valores mayores que p ya que la curva no tiene ninguna modificación) y $Q_{i,r}^w$ denota el i -ésimo punto de control del r -ésimo paso de inserción (con $Q_{i,0}^w = P_i^w$) entonces (4.10) y (4.9) se pueden generalizar con el llamado Algoritmo de Boor dado por

$$Q_{i,r}^w = \alpha_{i,r} Q_{i,r-1}^w + (1 - \alpha_{i,r}) Q_{i-1,r-1}^w \quad (4.16)$$

con

$$\alpha_{i,r} = \begin{cases} 1 & \text{si } i \leq k - p + r - 1 \\ \frac{\bar{u} - u_i}{u_{i+p-r+1} - u_i} & \text{si } k - p + r \leq i \leq k - s \\ 0 & \text{si } i \geq k - s + 1 \end{cases} \quad (4.17)$$

Notemos que cuando $s = 0$ y $r = p$, este algoritmo genera la tabla

$$\begin{array}{ccc} & & Q_{k-p+1,1}^w \\ & & Q_{k-p+2,2}^w \\ & & Q_{k-p+2,1}^w \\ & & \vdots \\ & & \vdots \\ & & Q_{k-1,1}^w \\ & & Q_{k,2}^w \\ & & Q_{k,1}^w \end{array} \quad \begin{array}{ccc} & & \\ & & \\ & & \\ & & \\ & & Q_{k,p}^w \\ & & \\ & & \\ & & \end{array}$$

en donde los puntos de control resultantes son $Q_{k-p+1,1}^w, Q_{k-p+2,2}^w, \dots, Q_{k,p}^w, \dots, Q_{k,2}^w, Q_{k,1}^w$. Así en cada inserción el número total de nuevos puntos es

$$p - s + r - 1 \quad (4.18)$$

los cuales remplazan a

$$p - s - 1 \quad (4.19)$$

que inician en el índice

$$k - p + 1 \quad (4.20)$$

Por ejemplo para la triple inserción del nodo $\bar{u} = 3.5$, la correspondiente tabla sería

$$\begin{array}{ccc} & & Q_{4,1}^w \\ & & Q_{5,2}^w \\ & & Q_{5,1}^w \\ & & Q_{6,3}^w \\ & & Q_{6,2}^w \\ & & Q_{6,1}^w \end{array}$$

en donde (4.11), (4.12) y (4.14) son respectivamente las $\alpha_{i,1}$, $\alpha_{i,2}$ y $\alpha_{i,3}$ del algoritmo de Boor. En la primera inserción, los 3 ($3 - 0 + 1 - 1 = 3$) nuevos puntos son $Q_{4,1}^w, Q_{5,1}^w, Q_{6,1}^w$, en la segunda, los 2 ($3 - 1 + 1 - 1 = 2$) nuevos puntos son $Q_{5,2}^w, Q_{6,2}^w$ los cuales remplazan a $Q_{5,1}^w$ ($3 - 1 - 1 = 1$) el cual tiene índice $7 - 3 + 1 = 5$.

4.1.1. Refinamiento.

Sea una curva NURBS

$$C^w(u) = \sum_{i=0}^n N_{i,p}(u)P_i^w$$

con vector de nodos $U = \{u_0, \dots, u_m\}$ y un conjunto de números reales $X = \{x_0, \dots, x_r\}$ tales que $x_i \leq x_{i+1}$ y $u_0 < x_i < u_m$ para toda i . Entonces supongamos que se desea realizar la inserción de los elementos de X en U (con lo que resultarán los nuevos puntos de control $Q_0^w, \dots, Q_{n+r+1}^w$ y un vector \bar{U}). Claramente el proceso a seguir, sería aplicar el algoritmo de inserción anterior a cada uno de los elementos del conjunto hasta terminarlos, esto es correcto, sin embargo existen algoritmos más eficientes para realizarlo. Uno de ellos es el llamado de Oslo diseñado por Riesenfeld, Cohen y Lyche [Cohen], otro más simple y también eficiente es el desarrollado por Boehm y Prautzsch [Boehm85] el cual presentamos a continuación.

La idea consiste en encontrar índices a y b tales que $u_a \leq x_i \leq u_b$, para toda i , y con éstos determinar los segmentos del polígono de control y los puntos que serán alterados, es decir los de L_{a-p+1} a L_{b-1} (ver Figura 4.7).

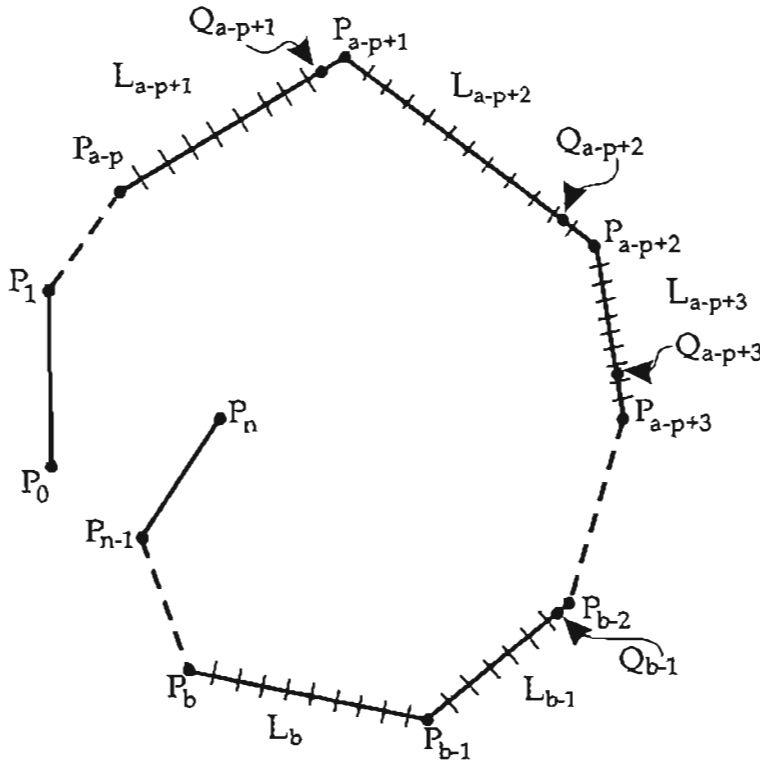


Figura 4.7: Aristas del polígono de control que se ven afectadas por el refinamiento del vector de nodos.

Esto implica además, que los puntos que no cambian son P_0^w, \dots, P_{a-p}^w y P_{b-1}^w, \dots, P_n^w y que el número de nuevos puntos a calcular está dada por

$$(n + r + 2) - [(a - p + 1) + (n - b + 2)] = r + p + b - a - 1$$

El algoritmo, calcula los nuevos puntos usando un ciclo de inserción, que puede iniciar con x_0 (con

4.1. INSERCIÓN DE NODOS.

el que obtiene Q_{a-p+1}^w) o con x_r (con el que obtiene Q_{b+r-1}^w). Nosotros presentaremos esta última².

```

RefineKnotVectCurve(n, p, U, Pw, X, r,  $\bar{U}$ , Qw)
{
  /* Entrada : n, p, U, Pw, X, r */
  /* Salida :  $\bar{U}$ , Qw */

  m = n + p + 1
  a = Encuentra subíndice a tal que  $u_a \leq x_0$ 
  b = Encuentra subíndice b tal que  $x_r \leq u_b$ 

  for(j=0; j<=a-p; j++) Qjw = Pjw
  for(j=b-1; j<=n; j++) Qj+r+1w = Pjw
  for(j=0; j<=a; j++)  $\bar{U}_j = U_j$ 
  for(j=0; j<=m; j++)  $\bar{U}_{j+r+1} = U_j$ 

  i = b + p + 1 k = b + p + r
  for(j=r; j>=0; j--)
  {
    while(Xj <= Ui && i > a)
    {
      Qk-p-1w = Pi-p-1w
       $\bar{U}_k = U_i$ 
      k=k-1 i=i-1
    }
    Qk-p-1w = Qk-pw
    for(l=1; l<=p; l++)
    {
      ind = k-p+1
       $\alpha = \bar{U}_{k+1} - X_j$ 
      if( $\alpha == 0.0$ )
        Qind-1w = Qindw
      else
      {
         $\alpha = \frac{\alpha}{U_{k+1} - U_{i-p+1}}$ 
        Qind-1w =  $\alpha Q_{ind-1}^w + (1 - \alpha)Q_{ind}^w$ 
      }
    }

     $\bar{U}_k = X_j$ 
    k=k-1
  }
}

```

²Boehm y Hartmut, también proponen el algoritmo para el caso de una curva NURBS periódica[Boehm85].

Ahora presentamos un ejemplo sencillo para mostrar su funcionamiento.

Ejemplo 27 Sea la curva NURBS cuadrática que se presenta en la Figura 4.8, la cual tiene un vector de nodos $U = \{0, 0, 0, 1, 2, 2, 2\}$. Entonces supongamos que se desea refinar U con el conjunto $X = \{\frac{1}{2}, \frac{3}{2}\}$.

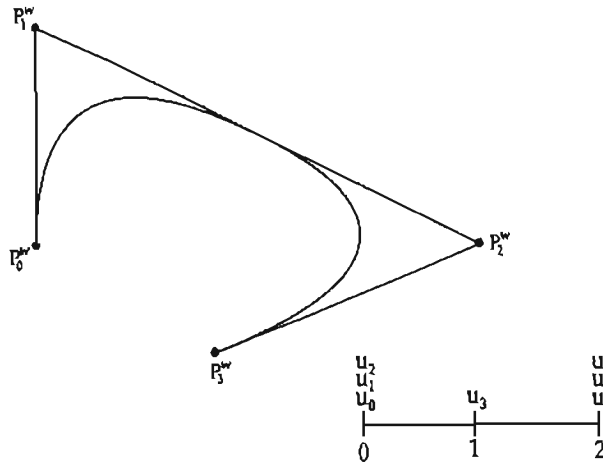


Figura 4.8: Curva NURBS cuadrática original.

Para este caso notemos que $a = 2$, $b = 4$ y $r = 1$, así el primer paso del algoritmo es (ver Figura 4.9)

$$Q_0^w = P_0^w, \quad Q_5^w = P_3^w$$

$$\bar{u}_0 = u_0, \quad \bar{u}_1 = u_1, \quad \bar{u}_2 = u_2, \quad \bar{u}_8 = u_6.$$

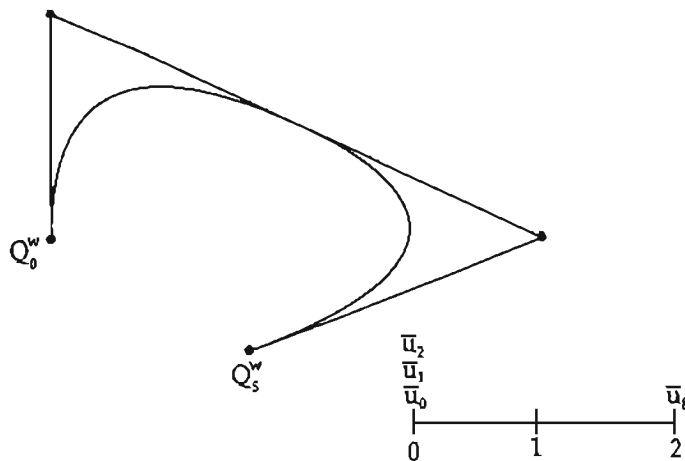


Figura 4.9: Primer paso del algoritmo de refinamiento.

Ahora entrando al for principal tenemos

$$Q_4^w = P_2^w, \quad \bar{u}_7 = u_5, \quad k = 6, \quad i = 4$$

$$Q_3^w = P_1^w, \quad \bar{u}_6 = u_4, \quad k = 6, \quad i = 4$$

$$Q_2^w = Q_3^w$$

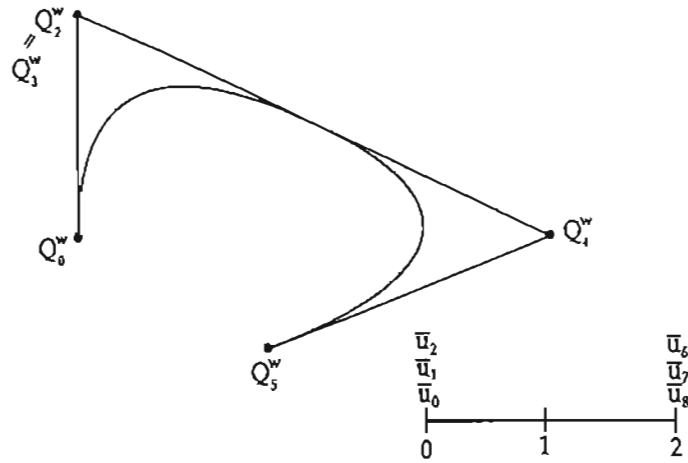


Figura 4.10: Polígono y vector de nodos intermedios en el algoritmo de refinamiento.

esto se muestra en la Figura 4.10. A continuación se calcula (ver Figura 4.11)

$$\alpha = \bar{u}_6 - x_1 = \frac{1}{2}$$

$$\alpha = \frac{\alpha}{\bar{u}_6 - u_2} = \frac{1}{4}$$

$$Q_3^w = \alpha Q_3^w + (1 - \alpha) Q_4^w$$

$$\alpha = \bar{u}_7 - x_1 = \frac{1}{2}$$

$$\alpha = \frac{\alpha}{\bar{u}_7 - u_3} = \frac{1}{2}$$

$$Q_4^w = \alpha Q_4^w + (1 - \alpha) Q_5^w$$

$$\bar{u}_5 = x_1, k = 4$$

$$Q_1^w = P_0, \bar{u}_4 = u_3, k = 3, i = 2.$$

Y finalmente tenemos

$$\alpha = \bar{u}_4 - x_0 = \frac{1}{2}$$

$$\alpha = \frac{\alpha}{\bar{u}_4 - u_1} = \frac{1}{2}$$

$$Q_1^w = \alpha Q_1^w + (1 - \alpha) Q_2^w$$

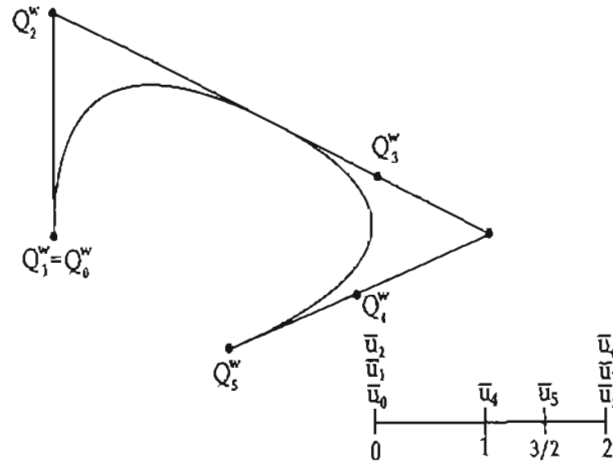


Figura 4.11: Polígono y vector de nodos intermedios en el algoritmo de refinamiento.

$$\alpha = \bar{u}_5 - x_0 = 1$$

$$\alpha = \frac{\alpha}{\bar{u}_5 - u_2} = \frac{2}{3}$$

$$Q_2^w = \alpha Q_2^w + (1 - \alpha) Q_3^w$$

$$\bar{u}_3 = x_0$$

En la Figura 4.12, se muestra la curva con su vector de nodos refinado y el polígono de control resultante.

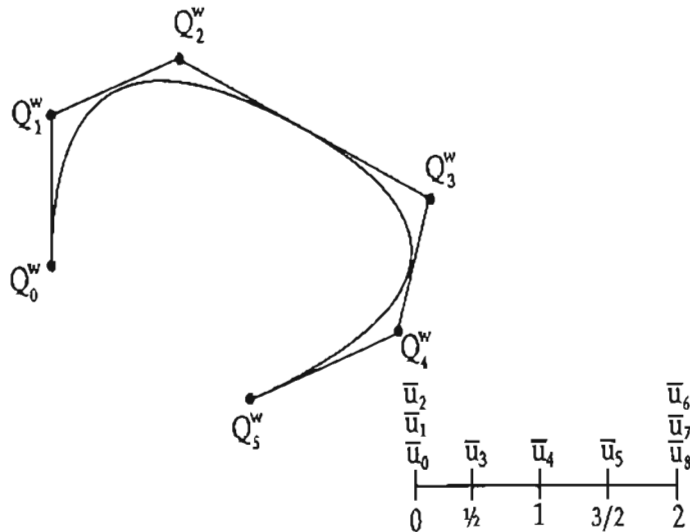


Figura 4.12: Polígono de control y vector de nodos resultante.

□

4.1. INSERCIÓN DE NODOS.

Una importante aplicación del refinamiento de nodos, es la descomposición de una curva NURBS en sus respectivos segmentos de Bézier. Para lograr esto, se deben insertar nodos iguales a los interiores, de tal manera la multiplicidad de todos sea p , es decir

$$U = \underbrace{\{a, \dots, a\}}_{p+1}, \underbrace{u_p, \dots, u_{m-p-1}}_{p+1}, \underbrace{\{b, \dots, b\}}_{p+1} \quad (4.21)$$

se debe convertir en

$$U = \underbrace{\{a, \dots, a\}}_{p+1}, \underbrace{u_p, \dots, u_p}_p, \dots, \underbrace{u_{m-p-1}, \dots, u_{m-p-1}}_p, \underbrace{\{b, \dots, b\}}_{p+1}$$

Esto permitirá que la unión de las subcurvas tenga una continuidad de clase $C^{(0)}$ (ver la propiedad x) de la curvas NURBS sección 2.4), resultando las curvas de Bézier que la componen.

Ejemplo 28 Supongamos que se desea descomponer la curva NURBS cuadrática que se presenta en la Figura 4.13, en sus respectivas curvas de Bézier. Así ya que su vector de nodos es

$$U = \{0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1\}$$

debemos insertar nodos $\frac{1}{3}$ y $\frac{2}{3}$ de tal manera que tengamos

$$U = \{0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1\}$$

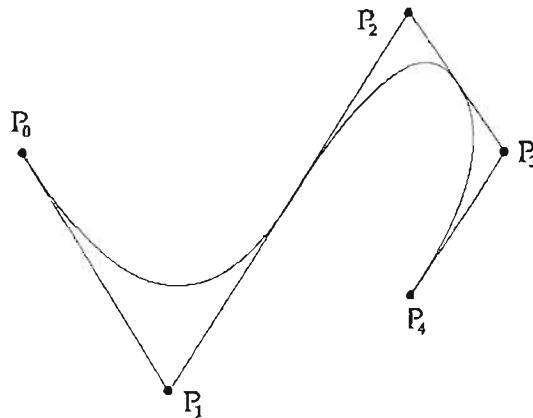


Figura 4.13: Curva NURBS cuadrática con 5 puntos de control.

Una vez que realizado el proceso de inserción, la curva NURBS y las curvas de Bézier resultantes son las que se muestran en la Figura 4.13

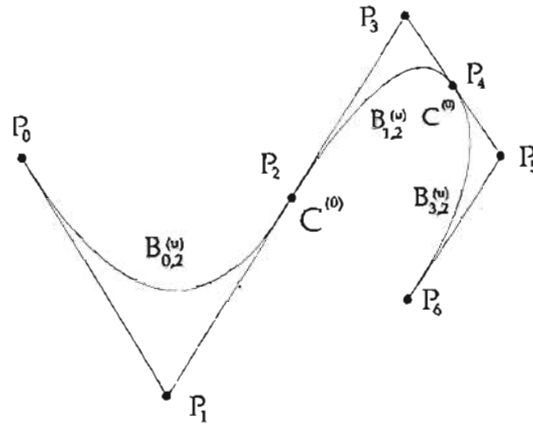


Figura 4.14: Curva NURBS con sus 3 segmentos de Bézier.

En donde sus respectivas funciones base son

$$\begin{aligned}
 N_{0,2}(u) &= \begin{cases} (1-3u)^2 & \text{si } u \in [0, \frac{1}{3}) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{1,2}(u) &= \begin{cases} 6u(1-3u)^2 & \text{si } u \in [0, \frac{1}{3}) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{2,2}(u) &= \begin{cases} 9u^2 & \text{si } u \in [0, \frac{1}{3}) \\ (2-3u)^2 & \text{si } u \in [\frac{1}{3}, \frac{2}{3}) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{3,2}(u) &= \begin{cases} 2(2-3u)(3u-1) & \text{si } u \in [\frac{1}{3}, \frac{2}{3}) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{4,2}(u) &= \begin{cases} (3u-1)^2 & \text{si } u \in [\frac{1}{3}, \frac{2}{3}) \\ (3-3u)^2 & \text{si } u \in [\frac{2}{3}, 1) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{5,2}(u) &= \begin{cases} 2(3u-2)(3-3u) & \text{si } u \in [\frac{2}{3}, 1) \\ 0 & \text{en otro caso} \end{cases} \\
 N_{6,2}(u) &= \begin{cases} 3u-1 & \text{si } u \in [\frac{2}{3}, 1) \\ 0 & \text{en otro caso} \end{cases}
 \end{aligned}$$

las cuales se presenta en la Figura 4.15, notemos que éstas son iguales a los polinomios de Bernstein cuadráticos en cada intervalo $[0, \frac{1}{3}]$, $[\frac{1}{3}, \frac{2}{3}]$ y $[\frac{2}{3}, 1]$.

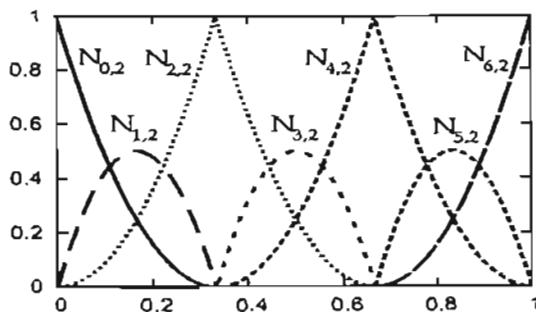


Figura 4.15: Funciones base cuadráticas de la curva NURBS refinada.

4.2. Eliminación de nodos.

El proceso de eliminación de nodos es el inverso al de inserción, y nos será de gran utilidad para el algoritmo de elevación de grado que estudiaremos en la siguiente sección.

Sea una curva NURBS

$$C^w(u) = \sum_{i=0}^n N_{i,p}(u) P_i^w \quad (4.22)$$

con vector de nodos U y $u_r \in U$, un nodo interior de multiplicidad s . Si suponemos que U_t denota el vector de nodos obtenido al quitar u_r , en t ocasiones de U ($1 \leq t \leq s$). Diremos que u_r se puede eliminar t veces de la curva $C^w(u)$, si ésta se puede escribir en la forma

$$C^w(u) = \sum_{i=0}^{n-t} \tilde{N}_{i,p}(u) Q_i^w \quad (4.23)$$

donde $\tilde{N}_{i,p}(u)$ son las funciones base definidas en U_t .

Para esto, recordemos del capítulo 1, que las funciones B-spline $N_{i,p}(u)$ distintas de cero en u_r tiene una continuidad de clase $C^{(p-s)}$. Así, aunque generalmente no se espera más que ésta, es posible que el polígono de control tengan una geometría definida por el algoritmo de Casteljau (ver sección 2.1.2) de tal manera que su continuidad sea mayor y nos permita eliminar una o más veces este nodo, sin ninguna alteración. En otras palabras, u_r se puede eliminar t veces si y sólo si la curva $C^w(u)$ es de clase $C^{(p-s+t)}$ en u_r .³

De acuerdo a esto deduciremos las ecuaciones para determinar si un nodo se puede eliminar usando el siguiente ejemplo. Por simplicidad de notación omitiremos el superíndice w de los puntos de control y lo remplazaremos por un entero que representará el paso del proceso de eliminación, así inicialmente este valor será 0.

Ejemplo 29 Sea una curva NURBS cúbica, con puntos de control P_0^0, \dots, P_6^0 y vector de nodos

$$U = \{u_0, \dots, u_{11}\} = \{0, 0, 0, 0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 1, 1, 1\}$$

que se presenta en la Figura (4.16). Notemos que ésta, es curva NURBS subdividida por dos de Bézier racionales $C_0^w(u)$ y $C_1^w(u)$ (por simplicidad de notación las denotaremos así, en vez de $B_0(u)$ y $B_1(u)$), las cuales según la ecuación (2.15), para que su unión sea de continuidad de clase $C^{(1)}$ y así $u = \frac{1}{2}$ pueda ser eliminado una vez, los puntos P_2^0, P_3^0 y P_4^0 deben ser colineales y la distancia de P_2 a P_3 y P_3 a P_4 ,

³Es importante notar que la continuidad debe ser respecto a $C^w(u)$ y no a su proyección $C(u)$, la cual puede ser continua pero $C^w(u)$ no.

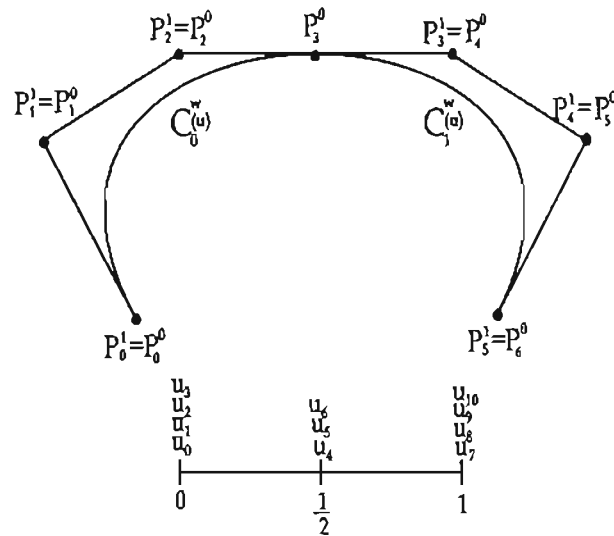


Figura 4.16: Curva NURBS formada por dos curvas que se unen con una continuidad de clase $C^{(2)}$

proporcional a la que hay entre los nodos u_6 a u_3 y u_7 a u_4 respectivamente (ver Figura 4.16), lo cual se puede escribir como

$$\frac{p}{u_6 - u_3} (P_3^0 - P_2^0) = \frac{p}{u_7 - u_4} (P_4^0 - P_3^0)$$

haciendo $u = u_4 = u_6$ y reordenando términos esta ecuación tiene la forma

$$P_3^0 = \frac{u - u_3}{u_7 - u_3} P_4^0 + \frac{u_7 - u}{u_7 - u_3} P_2^0 \quad (4.24)$$

si definimos $P_i^1 = P_i^0$ para $i = 1, 2$, y $P_j^1 = P_{j+1}^0$ para $j = 3, 4$ podemos escribir

$$P_3^0 = \alpha_3 P_3^1 + (1 - \alpha_3) P_2^1 \quad (4.25)$$

con

$$\alpha_3 = \frac{u - u_3}{u_7 - u_3}$$

Ahora, para lograr que la unión de $C_0^w(u)$ y $C_1^w(u)$ sea de continuidad de clase $C^{(2)}$ y $u = \frac{1}{2}$ pueda ser removido por segunda vez, debe existir un punto P_2^2 (ver Figura 4.17) tal que

$$\begin{aligned} P_2^1 &= \alpha_2 P_2^2 + (1 - \alpha_2) P_1^2 \\ P_3^1 &= \alpha_3 P_3^2 + (1 - \alpha_3) P_2^2 \end{aligned} \quad (4.26)$$

con

$$\alpha_i = \frac{u - u_i}{u_{i+p+2} - u_i}$$

definiendo $P_1^2 = P_1^1$, $P_3^2 = P_4^1$ para $i = 2, 3$.

Finalmente, para la continuidad de clase $C^{(2)}$ y $u = \frac{1}{2}$ pueda ser removido tres veces, deben existir puntos P_1^3 y P_2^3 (ver Figura 4.18) tales que

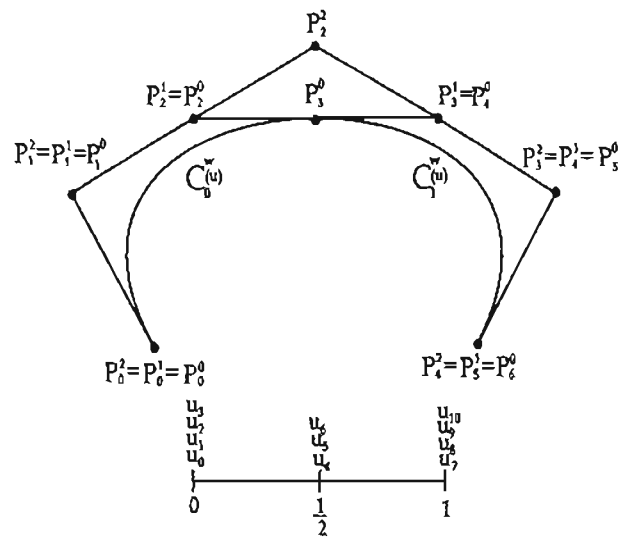


Figura 4.17: El vector de nodos $u = \frac{1}{2}$ puede ser eliminado dos veces sin que se pierda diferenciabilidad.

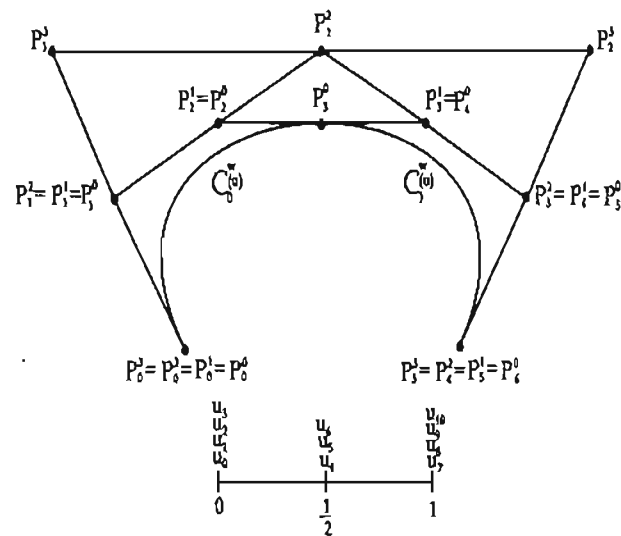


Figura 4.18: El vector de nodos $u = \frac{1}{2}$ puede ser eliminado 3 veces sin que se pierda diferenciabilidad.

$$\begin{aligned} P_1^2 &= \alpha_1 P_1^3 + (1 - \alpha_1) P_0^3 \\ P_2^2 &= \alpha_2 P_2^3 + (1 - \alpha_2) P_1^3 \\ P_3^2 &= \alpha_3 P_3^3 + (1 - \alpha_3) P_2^3 \end{aligned} \quad (4.27)$$

con

$$\alpha_i = \frac{u - u_i}{u_{i+p+3} - u_i}$$

para $i = 1, 2, 3$, haciendo $P_0^3 = P_0^0$, $P_3^3 = P_0^0$.

Resumiendo, la ecuación (4.25), nos indica que tanto $u = \frac{1}{2}$ como P_0^3 (que será reemplazado por P_2^1 y P_3^1) pueden ser eliminados siempre y cuando $\alpha_3 P_3^1 + (1 - \alpha_3) P_2^1$ y P_0^3 coincidan dentro de cierta tolerancia.

Si en (4.26), despejamos P_2^2 , tenemos

$$P_2^2 = \frac{P_2^1 - (1 - \alpha_2) P_1^2}{\alpha_2} \quad P_2^2 = \frac{P_3^1 - \alpha_3 P_3^2}{1 - \alpha_3}$$

y así, u se puede eliminar dos veces y los P_2^1 y P_3^1 ser reemplazados por P_2^2 , si éstos cocientes son iguales, dentro de cierta tolerancia.

Finalmente, de la primera y tercera ecuación de (4.27) tenemos que

$$P_1^3 = \frac{P_1^2 - (1 - \alpha_1) P_0^3}{\alpha_1} \quad P_2^3 = \frac{P_3^2 - \alpha_3 P_3^3}{1 - \alpha_3}$$

y por lo tanto, u podrá ser eliminado tres veces y P_1^2 , P_2^2 y P_3^2 reemplazados por P_1^3 y P_2^3 , si el punto que resulta de sustituir éstos últimos en el lado derecho de la segunda ecuación de (4.27), coinciden dentro de cierta tolerancia con P_2^2 .

□

En general si tenemos un número par de ecuaciones como en (4.26), es posible determinar si un nodo puede ser eliminado del vector U de la curva NURBS (4.22), si al calcular los nuevos puntos de control iniciando con la primera y última ecuación (después con la segunda y penúltima y así sucesivamente terminando con dos ecuaciones) los últimos dos coinciden dentro de cierta tolerancia.

Análogamente para el caso impar, es posible determinar si un nodo puede ser eliminado, si al calcular los nuevos puntos iniciando con la primera y última ecuación y así sucesivamente, el lado derecho de la ecuación de en medio, al ser sustituidos los puntos obtenidos en las ecuaciones anterior (o de arriba) y posterior (o de abajo), coinciden dentro de cierta tolerancia con el punto del lado izquierdo.

Si $u = u_r \neq u_{r+1}$, los nuevos puntos de control en el proceso de eliminación de nodos en el t -ésimo paso están dados por

$$\begin{aligned} P_i^t &= \frac{P_i^{t-1} - (1 - \alpha_i) P_{i-1}^t}{\alpha_i} & r - p - t + 1 \leq i \leq \frac{1}{2}(2r - p - s - t) \\ P_j^t &= \frac{P_{j+1}^{t-1} - \alpha_{j+1} P_{j+1}^t}{1 - \alpha_{j+1}} & \frac{1}{2}(2r - p - s + t + 1) \leq j \leq r - s + t - 1 \end{aligned}$$

con

$$\alpha_i = \frac{u - u_i}{u_{i+p+t} - u_i} \quad \alpha_j = \frac{u - u_{j-t+1}}{u_{j+p+1} - u_{j-t+1}}$$

4.3. Elevación de grado.

En esta sección estudiaremos el algoritmo de elevación de grado, el cual nos permite entre otras cosas, empatar el grado de distintas curvas NURBS sin alterarlas, esto nos será de gran utilidad en el siguiente capítulo ya que con estas curvas es posible construir distintas superficies.

Sea $C_p^w(u) = \sum_{i=0}^n N_{i,p}(u)P_i^w$ una curva NURBS con vector de nodos U . Entonces por definición, sabemos que es una curva polinomial por pedazos y por lo tanto es posible obtener su representación con grado mayor que p ⁴. Es decir, debe existir un vector de nodos \hat{U} , un entero positivo \hat{n} , y un conjunto de puntos $\{Q_i^w\}$ tales que

$$C_p^w(u) = C_{p+1}^w(u) = \sum_{i=0}^{\hat{n}} \hat{N}_{i,p+1}(u)Q_i^w \quad (4.28)$$

en donde $\hat{N}_{i,p+1}(u)$, es la función B-spline de grado $p+1$, definida en \hat{U} para toda i .

Si suponemos que U tiene la forma

$$U = \{u_0, \dots, u_m\} = \underbrace{\{a, \dots, a\}}_{p+1}, \underbrace{\{u_1, \dots, u_1\}}_{m_1}, \dots, \underbrace{\{u_s, \dots, u_s\}}_{m_s}, \underbrace{\{b, \dots, b\}}_{p+1}$$

entonces $C_p^w(u)$ es de al menos de continuidad de clase $C^{(p-m_i)}$ en cada nodo interior u_i (para $i = 1, \dots, s$) y por lo tanto se espera que $C_{p+1}^w(u)$ también tenga esta continuidad, de donde la multiplicidad de sus nodos interiores debe de ser $m_i + 1$ y en los extremos tener un nodo más, es decir

$$\hat{U} = \{u_0, \dots, u_{\hat{m}}\} = \underbrace{\{a, \dots, a\}}_{p+2}, \underbrace{\{u_1, \dots, u_1\}}_{m_1+1}, \dots, \underbrace{\{u_s, \dots, u_s\}}_{m_s+1}, \underbrace{\{b, \dots, b\}}_{p+2}$$

por lo que $\hat{m} = m + s + 2$ y

$$\hat{n} = \hat{m} - (p+1) - 1 = m + s + 2 - p - 2 = n + p + 1 + s - p = n + s + 1.$$

Para el cálculo de los puntos $\{Q_i^w\}$, existen distintos métodos como los propuestos por Prautzsch [Prau84], Cohen [Cohen85] y Prautzsch y Piper [Prau91]. Este último es el más eficiente para el caso general, sin embargo nosotros presentaremos el propuesto por Piegl y Tiller [PT97, pág.200] el cual matemáticamente es más simple y compite el de Prautzsch y Piper, particularmente en el caso donde el grado es elevado a más de uno [Prau91].

El algoritmo consiste en la iteración de los siguientes pasos

1. Extraer la i -ésima curva de Bézier de $C^w(u)$.
2. Elevar el grado de de ésta.
3. Eliminar los nodos innecesarios para separar la curva $i - 1$ con la i .

Tanto el primer paso como el tercero fueron discutidos en las secciones (4.1.1) y (4.2) respectivamente. A continuación desarrollaremos segundo [Far90, pág.65].

Sea una curva de Bézier racional de grado p

$$B_p^w(u) = \sum_{i=0}^p B_{i,p}(u)P_i^w \quad (4.29)$$

⁴Esto es claro en el polinomio $a_p x^p + \dots + a_0$, ya que se puede escribir como $a_{p+1} x^{p+1} + a_p x^p + \dots + a_0$, haciendo $a_{p+1} = 0$

y una de grado $p + 1$

$$B_{p+1}^w(u) = \sum_{i=0}^{p+1} B_{i,p+1}(u) Q_i^w \quad (4.30)$$

Entonces multiplicando $(u + (1 - u)) = 1$ por (4.29) e igualando la a (4.30) tenemos que

$$\begin{aligned} \sum_{i=0}^{p+1} B_{i,p+1}(u) Q_i^w &= (u + (1 - u)) \sum_{i=0}^p B_{i,p}(u) P_i \\ &= \sum_{i=0}^p ((1 - u) B_{i,p}(u) + u B_{i,p}(u)) P_i \end{aligned}$$

ahora por definición

$$\begin{aligned} \sum_{i=0}^{p+1} \binom{p+1}{i} u^i (1 - u)^{p+1-i} Q_i^w &= \sum_{i=0}^p \binom{p}{i} ((1 - u)^{p+1-i} u^i + u^{i+1} (1 - u)^{p-i}) P_i \\ &= \sum_{i=0}^p \binom{p}{i} u^i (1 - u)^{p+1-i} P_i + \sum_{i=1}^{p+1} \binom{p}{i-1} u^i (1 - u)^{p+1-i} P_{i-1} \end{aligned}$$

entonces los coeficientes de $u^i (1 - u)^{p+1-i}$ cumplen

$$\binom{p+1}{i} Q_i^w = \binom{p}{i} P_i + \binom{p}{i-1} P_{i-1}$$

y ya que

$$\begin{aligned} \binom{p}{i} / \binom{p+1}{i} &= \frac{p! i! (p+1-i)!}{i! (p-i)! (p+1)!} = \frac{p+1-i}{p+1} = 1 - \frac{i}{p+1} \\ \binom{p}{i-1} / \binom{p+1}{i} &= \frac{p! i! (p+1-i)!}{(i-1)! (p+1-i)! (p+1)!} = \frac{i}{p+1} \end{aligned}$$

tenemos finalmente

$$Q_i = (1 - \alpha_i) P_i + \alpha_i P_{i-1} \quad (4.31)$$

donde

$$\alpha_i = \frac{i}{p+1}$$

para $i = 0, \dots, p+1$.

4.3. ELEVACIÓN DE GRADO.

Ejemplo 30 Sea una curvas NURBS cúbica con puntos de control P_0^w, \dots, P_5^w y vector de nodos $U = \{0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1\}$, la cual se presenta en la Figura 4.19. Entonces apliquemos el algoritmo anterior para cambiar su grado a 4.

Así, lo primero que debemos hacer, es extraer el segmento inicial de Bézier insertando dos veces el nodo $u = \frac{1}{3}$ en U . Esto se presenta en la Figura 4.20.

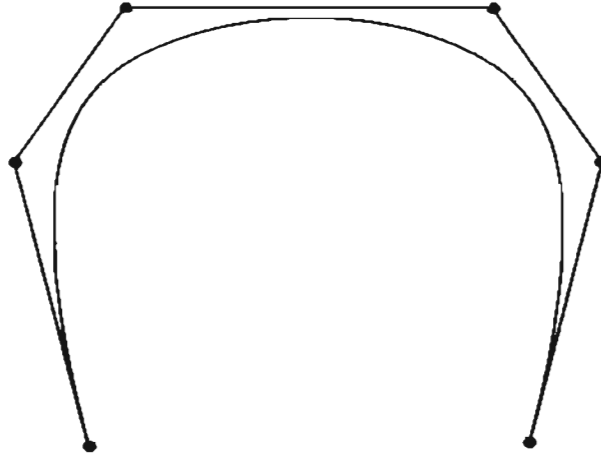


Figura 4.19: Curva NURBS cúbica.

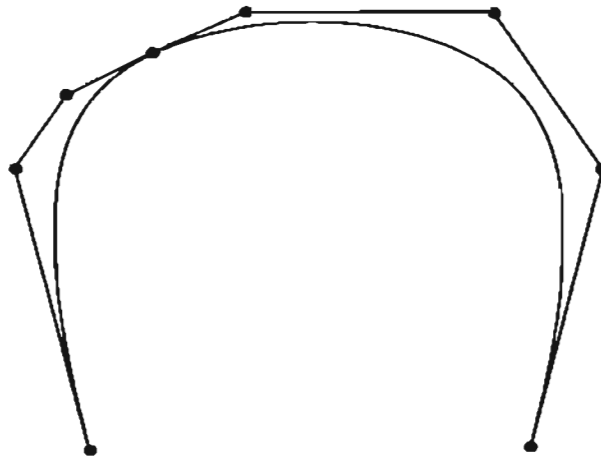


Figura 4.20: Se extrae el primer segmento de curva de Bézier.

Una vez hecho esto, elevemos el grado de la subcurva de Bézier utilizando la ecuación (4.31) (ver Figura 4.21) y obtengamos la segunda subcurva de Bézier insertando dos veces el nodo $u = \frac{2}{3}$ (ver Figura 4.22).

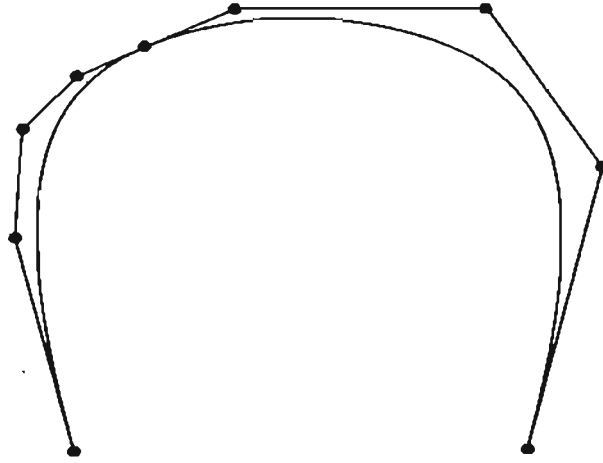


Figura 4.21: Elevación de grado de la subcurva de Bézier.

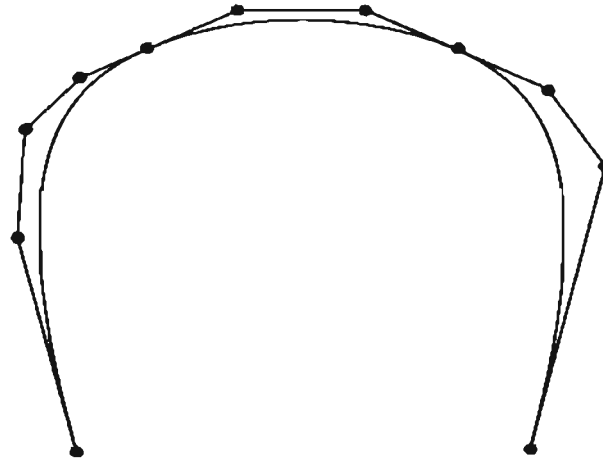


Figura 4.22: Se extrae la siguiente curva de Bézier.

4.3. ELEVACIÓN DE GRADO.

Ahora elevemos el grado de la segunda curva de Bézier (ver Figura 4.23) y eliminemos las dos repeticiones del nodo $u = \frac{1}{3}$ (ver Figura 4.24).

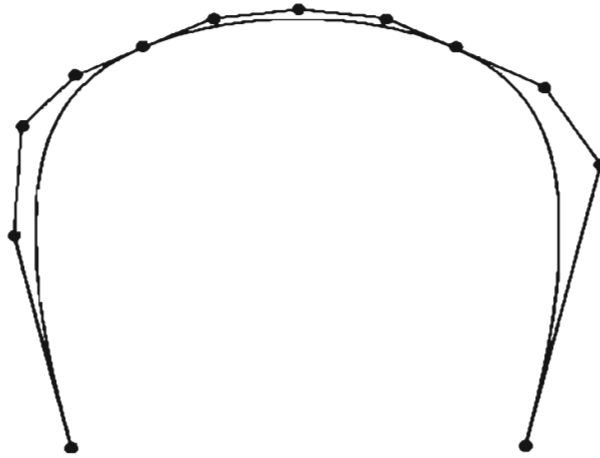


Figura 4.23: Se eleva el grado de la segunda curva de Bézier.

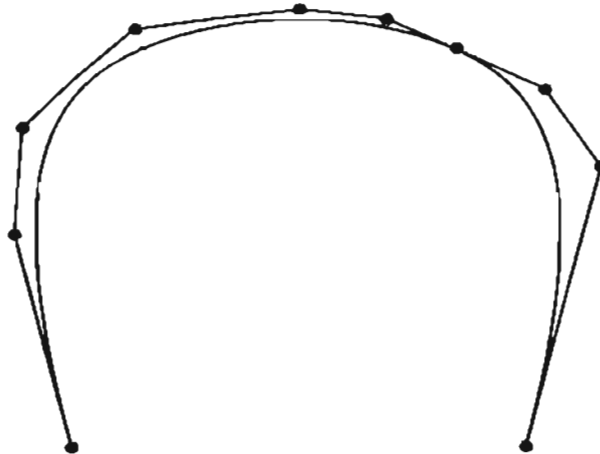


Figura 4.24: Se eliminan los nodos adicionales que se insertaron al extraer la primera curva de Bézier.

Finalmente elevemos el grado de la última subcurva de Bézier (4.25) y eliminemos dos ocurrencias del nodo $u = \frac{2}{3}$, con lo que tenemos la curva deseada que se muestra en la Figura 4.26.

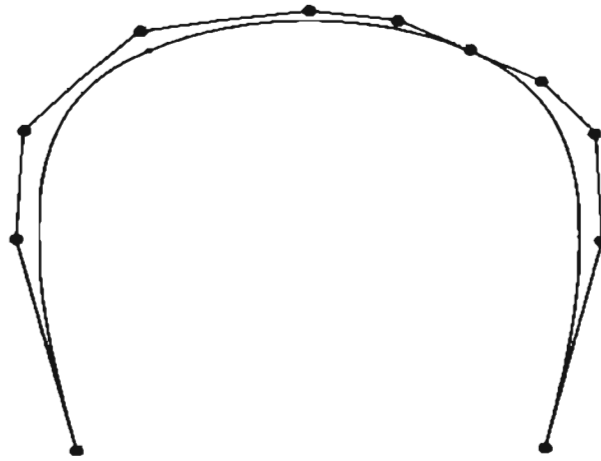


Figura 4.25: Se eleva el grado de la última curva de Bézier.

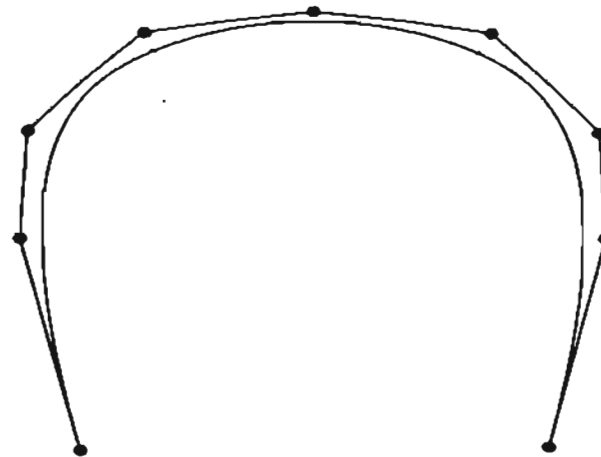


Figura 4.26: Se elimina los nodos adicionales que se insertaron para extraer el segundo segmento de Bézier.

□

4.4. Interpolación global de curvas B-spline.

Una de las técnicas más usadas en la aproximación polinomial es por supuesto la interpolación. A continuación describiremos como podemos realizarla usando una curva B-spline. Las interpolaciones usando curvas racionales, son un poco más complicadas sobre todo por las variables adicionales de los pesos w_i . Nosotros siguiendo el objetivo de este trabajo, no necesitaremos abordar este tema, sin embargo se puede encontrar algo en [Far91, pág. 59-74].

Sea un conjunto de puntos $\{Q_k\}$ para $k = 0, \dots, n$ en \mathbb{R}^3 , los cuales se desea interpolar con una curva B-spline de grado p . Si asignamos un valor parametral \bar{u}_k a cada Q_k y seleccionamos un vector de nodos apropiado $U = \{u_0, \dots, u_m\}$, es posible establecer un sistema de $(n+1) \times (n+1)$ ecuaciones

$$Q_k = C(\bar{u}_k) = \sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \quad (4.32)$$

en donde las incógnitas son los puntos de control de la curva B-spline de interpolación.

Así el único problema es la selección de los parámetros \bar{u}_k y el vector U , los cuales determinarán la forma de la curva.

Existen varias posibilidades para la selección de \bar{u}_k , a continuación presentaremos algunas posibilidades

- **Igualmente espaciado.**

$$\bar{u}_0 = 0 \quad \bar{u}_n = 1 \quad \text{y} \quad \bar{u}_k = \frac{k}{n} \quad k = 1, \dots, n-1 \quad (4.33)$$

Este método no es del todo recomendable, ya que puede producir formas que no describen la tendencia de los puntos interpolados.

- **Longitud de arco.**

Si d representa la longitud total de arco, es decir

$$d = \sum_{k=1}^n |Q_k - Q_{k-1}|$$

entonces en este caso se toman

$$\bar{u}_0 = 0 \quad \bar{u}_n = 1 \quad \text{y} \quad \bar{u}_k = \bar{u}_{k-1} + \frac{|Q_k - Q_{k-1}|}{d} \quad k = 1, \dots, n-1 \quad (4.34)$$

Este método, es el más comúnmente usado y generalmente es adecuado ya que su parametrización "refleja" la forma de la curva.

- **Centripetal.**

Aquí tomamos

$$d = \sum_{k=1}^n \sqrt{|Q_k - Q_{k-1}|}$$

y

$$\bar{u}_0 = 0 \quad \bar{u}_n = 1 \quad \text{y} \quad \bar{u}_k = \bar{u}_{k-1} + \frac{\sqrt{|Q_k - Q_{k-1}|}}{d} \quad k = 1, \dots, n-1. \quad (4.35)$$

Para el vector de nodos U , una posibilidad es hacer que los nodos interiores sean igualmente espaciados, es decir

$$\begin{aligned} u_0 = \dots = u_p = 0 & & u_{m-p} = \dots = u_m = 1 \\ u_{j+p} = \frac{j}{n-p+1} & & j = 1, \dots, n-p \end{aligned} \quad (4.36)$$

Sin embargo esto no es recomendable si se usa en conjunción con (4.34) o (4.35), ya que puede implicar un sistema de ecuaciones singular.

Otra posibilidad son las llamadas abscisas de Greville definidas por

$$\begin{aligned} u_0 = \dots = u_p = 0 & & u_{m-p} = \dots = u_m = 1 \\ u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p+1} \bar{u}_i & & j = 1, \dots, n-p \end{aligned} \quad (4.37)$$

las cuales reflejan la distribución de \bar{u}_k . Además usándolas en combinación con (4.34) o (4.35), el sistema de ecuaciones que resultante es definido positivo y bandeado [DeBoor78] y puede ser resuelto usando el método de eliminación Gaussiana sin pivoteo.

Ejemplo 31 Construyamos una curva de interpolación B-spline cúbica para los puntos $Q_0 = (2, 3, 0)$, $Q_1 = (6, 4, 0)$, $Q_2 = (4, 14, 0)$, $Q_3 = (12, 18, 0)$, $Q_4 = (15, 8, 0)$, $Q_5 = (18, 3, 0)$.

Entonces usando las ecuaciones (4.33) y (4.36), los valores parametrales y el vector de nodos están dados por

$$\bar{u}_0 = 0, \bar{u}_1 = 0.2, \bar{u}_2 = 0.4, \bar{u}_3 = 0.6, \bar{u}_4 = 0.8, \bar{u}_5 = 1$$

$$U = \{u_0, \dots, u_9\} = \{0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1\}$$

Para el caso de las ecuaciones (4.34) y (4.37), tenemos que

$$\begin{aligned} |Q_1 - Q_0| &= 4.123 & |Q_2 - Q_1| &= 10.198 \\ |Q_3 - Q_2| &= 8.944 & |Q_4 - Q_3| &= 10.440 & |Q_5 - Q_4| &= 5.831 \end{aligned}$$

entonces $d = 39.537$ y

$$\bar{u}_0 = 0, \bar{u}_1 = 0.104, \bar{u}_2 = 0.362, \bar{u}_3 = 0.588, \bar{u}_4 = 0.853, \bar{u}_5 = 1$$

$$U = \{0, 0, 0, 0, u_4, u_5, 1, 1, 1, 1\}$$

en donde $u_4 = \frac{1}{3}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3) = 0.352$ y $u_5 = \frac{1}{3}(\bar{u}_2 + \bar{u}_3 + \bar{u}_4) = 0.601$

Por último para (4.35) y (4.37)

$$\begin{aligned} \sqrt{|Q_1 - Q_0|} &= 2.030 & \sqrt{|Q_2 - Q_1|} &= 3.193 \\ \sqrt{|Q_3 - Q_2|} &= 2.990 & \sqrt{|Q_4 - Q_3|} &= 3.231 & \sqrt{|Q_5 - Q_4|} &= 2.414 \end{aligned}$$

y $d = 13.860$

$$\bar{u}_0 = 0, \bar{u}_1 = 0.146, \bar{u}_2 = 0.377, \bar{u}_3 = 0.593, \bar{u}_4 = 0.826, \bar{u}_5 = 1$$

$$U = \{0, 0, 0, 0, 0.352, 0.601, 1, 1, 1, 1\}$$

4.4. INTERPOLACIÓN GLOBAL DE CURVAS B-SPLINE.

Una vez calculado u_k y U , debemos resolver el sistema de ecuaciones (4.32), con matriz de coeficientes dados por

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ N_{0,3}(\bar{u}_1) & N_{1,3}(\bar{u}_1) & N_{2,3}(\bar{u}_1) & N_{3,3}(\bar{u}_1) & 0 & 0 \\ 0 & N_{1,3}(\bar{u}_2) & N_{2,3}(\bar{u}_2) & N_{3,3}(\bar{u}_2) & N_{4,3}(\bar{u}_2) & 0 \\ 0 & 0 & N_{2,3}(\bar{u}_3) & N_{3,3}(\bar{u}_3) & N_{4,3}(\bar{u}_3) & N_{5,3}(\bar{u}_3) \\ 0 & 0 & N_{2,3}(\bar{u}_4) & N_{3,3}(\bar{u}_4) & N_{4,3}(\bar{u}_4) & N_{5,3}(\bar{u}_4) \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

en la Figura (4.27), se presentan las tres curvas de interpolación con los distintos valores de u_k y U .

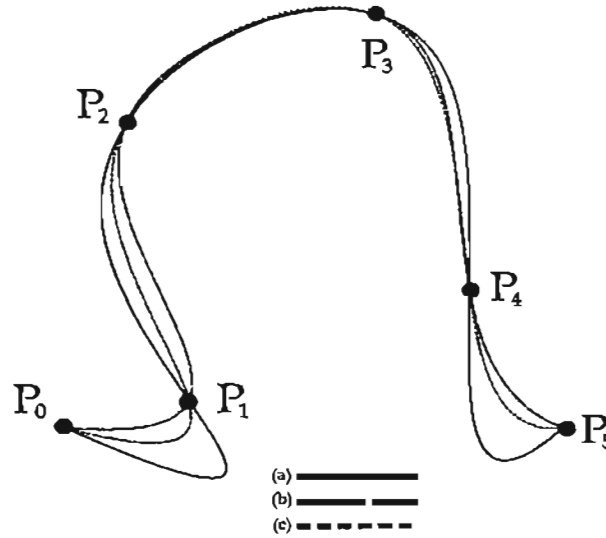


Figura 4.27: Interpolación cúbica con valores parametrales y vector de nodos (a) uniforme, (b) longitud de arco y abscisas de Greville y (c) longitud centripetal y abscisas de Greville.

□

En muchas aplicaciones es común necesitar una curva que además de interpolar los puntos $\{Q_k\}$, también lo haga con su primer derivada (o de orden mayor) $\{D_k\}$ (para $k = 0, \dots, n$) en éstos puntos.

Para esto, es necesario que en (4.32) se incluyan puntos adicionales y se tengan más ecuaciones, lo que además implica nuevos nodos (ya que tenemos más puntos a interpolar). Es decir las $n+1$ ecuaciones para los puntos Q_k ahora están dadas por

$$Q_k = C(\bar{u}_k) = \sum_{i=0}^{2n+1} N_{i,p}(\bar{u}) P_i \quad (4.38)$$

y usando la ecuación (2.33), las $n+1$ ecuaciones para las derivadas son

$$D_k = C'(\bar{u}_k) = \sum_{i=0}^{2n+1} N'_{i,p}(\bar{u}) P_i \quad (4.39)$$

Los parámetros \bar{u}_k se pueden calcular como antes, sin embargo U , ahora estará compuesto de $2(n+1) + p + 1$ nodos, los cuales pueden ser seleccionados de manera que reflejen la distribución de los \bar{u}_k ,

una posibilidad para los casos $p = 2$ y $p = 3$ son respectivamente

$$U = \{0, 0, 0, \bar{u}_1, \frac{\bar{u}_1 + \bar{u}_2}{2}, \bar{u}_2, \dots, \bar{u}_{n-1}, \frac{\bar{u}_{n-1} + 1}{2}, 1, 1, 1\} \quad (4.40)$$

y

$$U = \{0, 0, 0, 0, \frac{\bar{u}_1}{2}, \frac{2\bar{u}_1 + \bar{u}_2}{3}, \frac{\bar{u}_1 + 2\bar{u}_2}{3}, \dots, \frac{\bar{u}_{n-2} + 2\bar{u}_{n-1}}{3}, \frac{\bar{u}_{n-1} + 1}{2}, 1, 1, 1, 1\} \quad (4.41)$$

Finalmente las ecuaciones (4.38) y (4.39) se colocan de forma alternada lo que implica un sistema de $2(n + 1) \times 2(n + 1)$ ecuaciones.

Ejemplo 32 Construyamos una curva de interpolación B-spline cúbica que interpole los puntos del ejemplo anterior y a sus derivadas en estos puntos dadas en la siguiente tabla

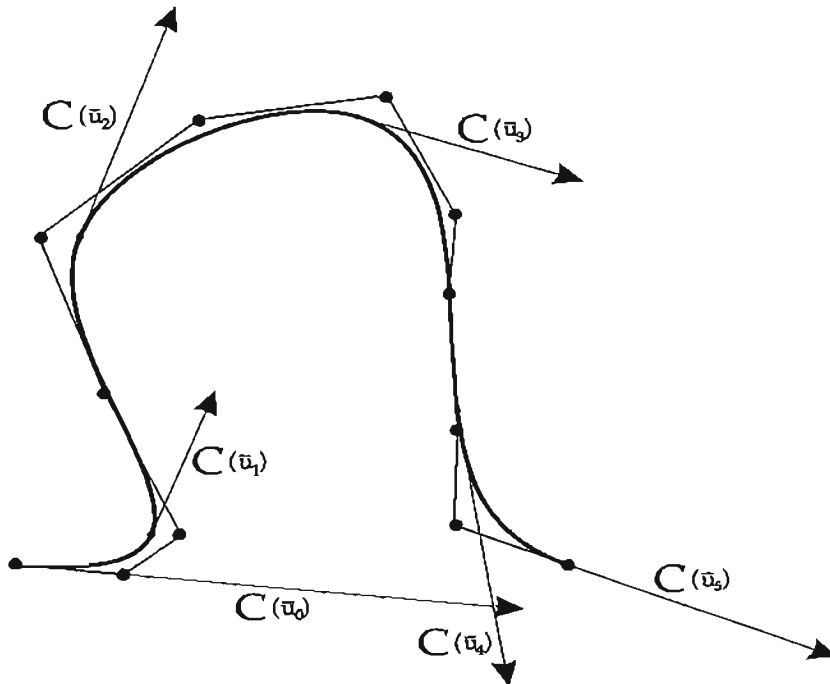


Figura 4.28: Curva de interpolación cúbica de ciertos puntos y sus derivadas.

Derivadas en los puntos de control		
$D_0 = (73.781, -7.368, 0)$	$D_1 = (9.308, 24.083, 0)$	$D_2 = (13.854, 38.409, 0)$
$D_3 = (32.523, -10.763, 0)$	$D_4 = (7.355, -46.020, 0)$	$D_5 = (38.784, -15.812, 0)$

Así si tomamos

$$\bar{u}_0 = 0, \bar{u}_1 = 0.104, \bar{u}_2 = 0.362, \bar{u}_3 = 0.588, \bar{u}_4 = 0.853, \bar{u}_5 = 1$$

el vector de nodos U es

$$U = \{0, 0, 0, 0, 0.130, 0.233, 0.357, 0.475, 0.595, 0.720, 0.833, 0.747, 1, 1, 1, 1\}$$

la matriz de coeficientes del sistema de ecuaciones tiene la forma

$$\begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ -\frac{3}{u_1} & \frac{3}{u_1} & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & N_{1,3}(\bar{u}_1) & N_{2,3}(\bar{u}_1) & N_{3,3}(\bar{u}_1) & N_{4,3}(\bar{u}_1) & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & N'_{1,3}(\bar{u}_1) & N'_{2,3}(\bar{u}_1) & N'_{3,3}(\bar{u}_1) & N'_{4,3}(\bar{u}_1) & \cdots & \cdots & \cdots & \cdots & 0 \\ & & & & \vdots & & & & & \\ 0 & \cdots & \cdots & \cdots & 0 & N_{7,3}(\bar{u}_4) & N_{8,3}(\bar{u}_4) & N_{9,3}(\bar{u}_4) & N_{10,3}(\bar{u}_4) & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & N'_{7,3}(\bar{u}_4) & N'_{8,3}(\bar{u}_4) & N'_{9,3}(\bar{u}_4) & N'_{10,3}(\bar{u}_4) & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & -\frac{3}{1-u_{11}} & \frac{3}{1-u_{11}} \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 \end{pmatrix}$$

y la curva resultante se muestra en la Figura (4.28).

□

4.5. Interpolación global con superficies B-spline.

De la definición de una superficie de producto tensor (ver capítulo 3), es relativamente simple extender la técnica de construcción de una curva de interpolación B-spline a las superficies B-spline.

Dado un conjunto de puntos $\{Q_{k,l}\}$ para $k = 0, \dots, n$ y $l = 0, \dots, m$ en \mathbb{R}^3 , deseamos construir una superficie B-spline de grados p y q en dirección de los parámetros u y v (respectivamente) que los interpole. Es decir determinar los puntos $P_{i,j}$ en \mathbb{R}^3 tales que

$$Q_{k,l} = S(\bar{u}_k, \bar{v}_l) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(\bar{u}_k) N_{j,q}(\bar{v}_l) P_{i,j} \quad (4.42)$$

para ciertos valores u_l y v_k para toda l y k .

Como en el caso de la curva de interpolación, lo primero que debemos obtener son los valores parametrales (\bar{u}_k, \bar{v}_l) y los vectores de nodos U y V .

Una posibilidad para \bar{u}_k , es usar (4.34) o (4.35) y para cada k en $\{Q_{k,l}\}$ calcular

$$\bar{u}_0^l, \dots, \bar{u}_n^l \text{ para } l = 0, \dots, m$$

y entonces

$$\bar{u}_k = \frac{\bar{u}_k^0 + \cdots + \bar{u}_k^m}{m+1} \text{ para } k = 0, \dots, n \quad (4.43)$$

Análogamente para \bar{v}_l tenemos

$$\bar{v}_0^k, \dots, \bar{v}_m^k \text{ para } k = 0, \dots, n$$

y

$$\bar{v}_l = \frac{\bar{v}_l^0 + \cdots + \bar{v}_l^n}{n+1} \text{ para } l = 0, \dots, m \quad (4.44)$$

Los vectores de nodos U y V , pueden ser calculados con las abscisas de Greville (4.37) usando (4.43) y (4.44).

Para obtener los puntos $\{P_{i,j}\}$, notemos que si fijamos la variable l en (4.42) la superficie tiene la forma

$$Q_{k,l} = \sum_{i=0}^n N_{i,p}(\bar{u}_k) \left(\sum_{j=0}^m N_{j,q}(\bar{v}_l) P_{i,j} \right) = \sum_{i=0}^n N_{i,p}(\bar{u}_k) R_{i,l} \quad (4.45)$$

con

$$R_{i,l} = \sum_{j=0}^m N_{j,q}(\bar{v}_l) P_{i,j} \tag{4.46}$$

es decir resulta la isocurva $C_{\bar{v}_l(u)}$ y una curva B-spline de interpolación de los puntos $\{Q_{k,l}\}$ para $k = 0, \dots, n$.

Ahora observemos que si en (4.46) dejamos fijo el valor de i y variamos l , resulta una curva B-spline de interpolación de los puntos $\{R_{i,l}\}$ (para $l = 0, \dots, m$) con la que obtendremos los puntos de control buscados $P_{i,0}, \dots, P_{i,m}$.

En resumen, si suponemos que los puntos que se desea interpolar son como los que se muestra en la Figura (4.29). El proceso de construcción de la superficie se puede establecer en los siguiente pasos.

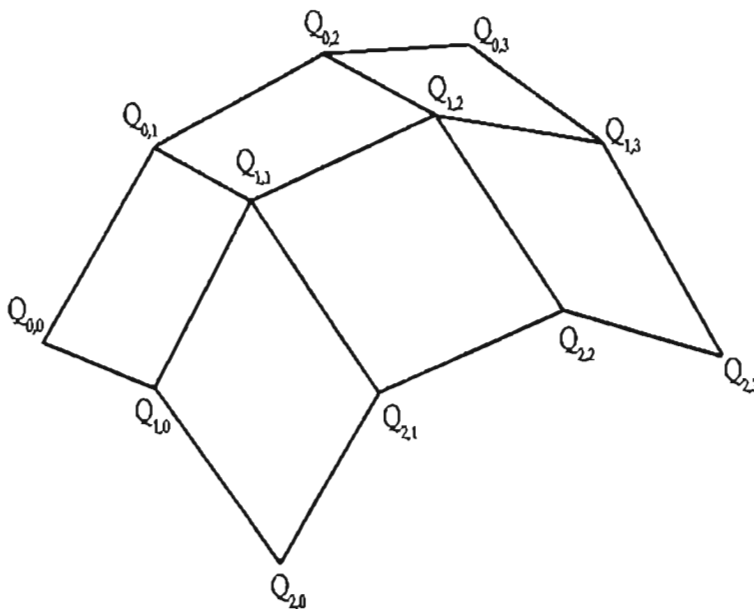


Figura 4.29: Puntos a interpolar.

1. Usando U y los valores \bar{u}_k , obtener $m + 1$ curvas de interpolación para los puntos $Q_{0,l}, \dots, Q_{k,l}$ para $l = 0, \dots, m$, con las que resulta los puntos $R_{i,l}$, ver Figura 4.30.
2. Usando V y los valores \bar{v} , obtener $n + 1$ curvas de interpolación a través de $R_{i,0}, \dots, R_{i,m}$ para $i = 0, \dots, n$, con las que resultan los puntos $P_{i,j}$, ver Figura 4.31.

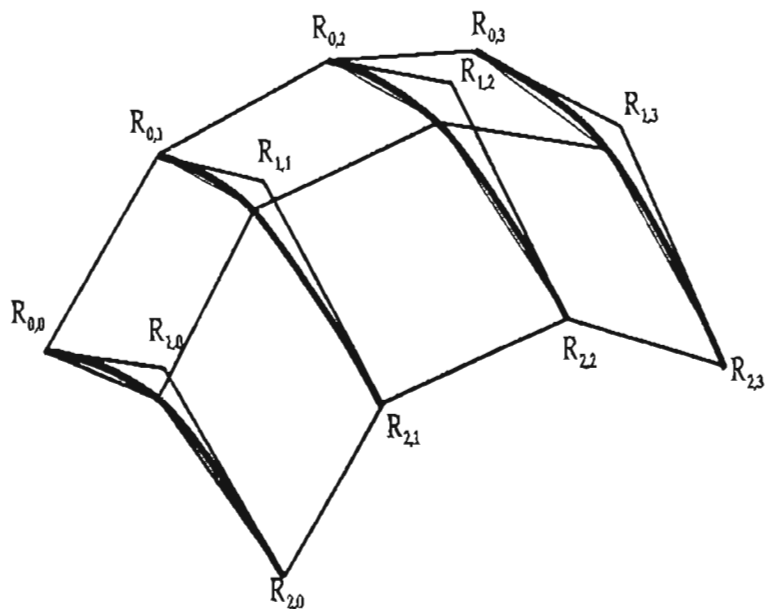


Figura 4.30: Isocurvas $C_{0,l}(u)$, para $l = 0, \dots, m$.

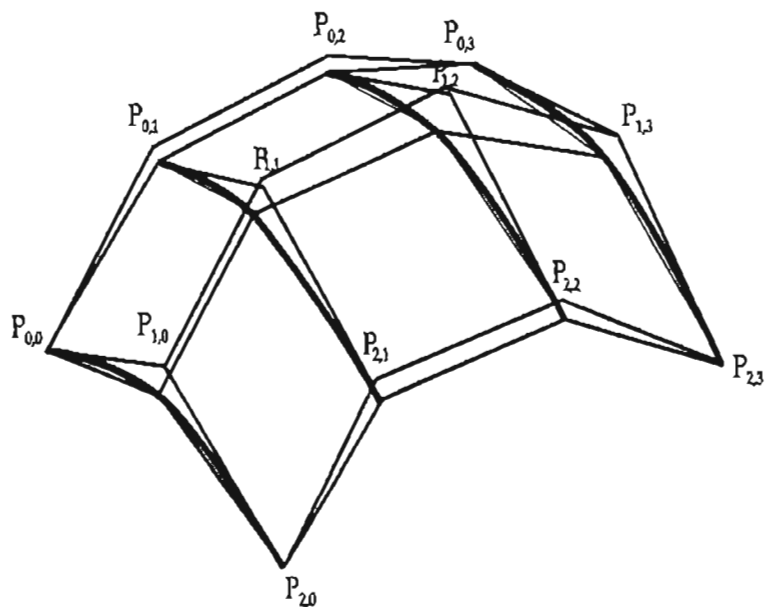


Figura 4.31: Puntos de control resultantes de la interpolación de los puntos de control de las isocurvas $C_{0,l}(u)$.

Ejemplo 33 Supongamos que se desea interpolar con una superficie B-spline los siguientes puntos, los cuales forman la malla que se muestra en la Figura 4.32 (a).

Puntos a interpolar					
i/j					
	(13, 15, 40)	(13, 5, 40)	(13, 0, 40)	(13, -5, 40)	(13, -10, 40)
	(10, 15, 40)	(10, 5, 43)	(10, 0, 40)	(10, -5, 40)	(10, -10, 40)
	(7, 15, 40)	(7, 5, 43)	(7, 0, 40)	(7, -5, 43)	(7, -10, 40)
	(-3, 15, 40)	(-3, 5, 40)	(-3, 0, 40)	(-3, -5, 40)	(-3, -10, 40)
	(-6, 15, 40)	(-6, 5, 42)	(-6, 0, 40)	(-6, -5, 35)	(-6, -10, 40)
	(-9, 15, 40)	(-9, 5, 40)	(-9, 0, 40)	(-9, -5, 40)	(-9, -10, 40)

Así usando (4.43) y (4.44) los valores parametrales están dados por

$$\begin{aligned} \bar{u}_0 = 0 \quad \bar{u}_1 = 0.13635 \quad \bar{u}_2 = 0.271186 \quad \bar{u}_3 = 0.698947 \quad \bar{u}_4 = 0.849474 \quad \bar{u}_5 = 1 \\ \bar{v}_0 = 0 \quad \bar{v}_1 = 0.385707 \quad \bar{v}_2 = 0.587684 \quad \bar{v}_3 = 0.793842 \quad \bar{v}_4 = 1 \end{aligned}$$

y entonces en el caso de una interpolación con $p = 1$ y $q = 1$, los vectores de nodos son

$$\begin{aligned} U &= \{0, 0, 0.13635, 0.271186, 0.698947, 0.849474, 1, 1\} \\ V &= \{0, 0, 0.385707, 0.587684, 0.793842, 1, 1\} \end{aligned}$$

esta superficie se presenta en la Figura 4.32 (b). Ahora supongamos que se desea realizar una interpolación con $p = 1$ y $q = 2$ entonces para este caso los vectores serían

$$\begin{aligned} U &= \{0, 0, 0.13635, 0.271186, 0.698947, 0.849474, 1, 1\} \\ V &= \{0, 0, 0, 0.486695, 0.690763, 1, 1, 1\} \end{aligned}$$

la superficie se muestra en la Figura 4.33 (a). Finalmente para $p = 3$ y $q = 2$ tenemos

$$\begin{aligned} U &= \{0, 0, 0, 0, 0.368828, 0.606536, 1, 1, 1, 1\} \\ V &= \{0, 0, 0, 0.486695, 0.690763, 1, 1, 1\} \end{aligned}$$

la cual se presenta en la Figura 4.33 (b).

□

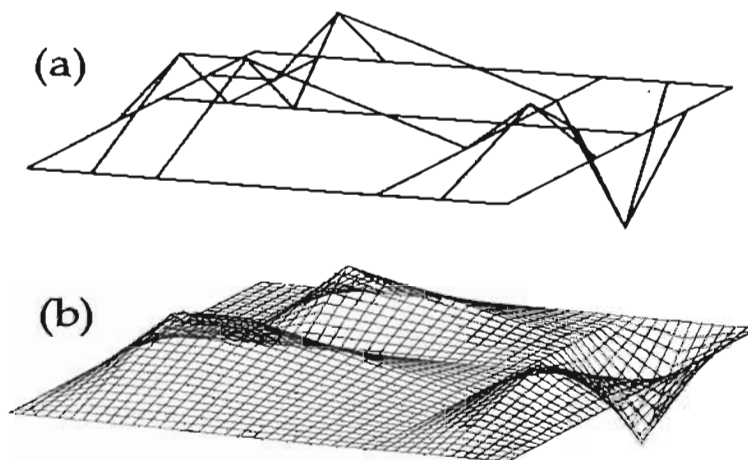


Figura 4.32: (a) Malla de puntos que se desea interpolar. (b) Superficie de interpolación con $p = 1$ y $q = 1$.

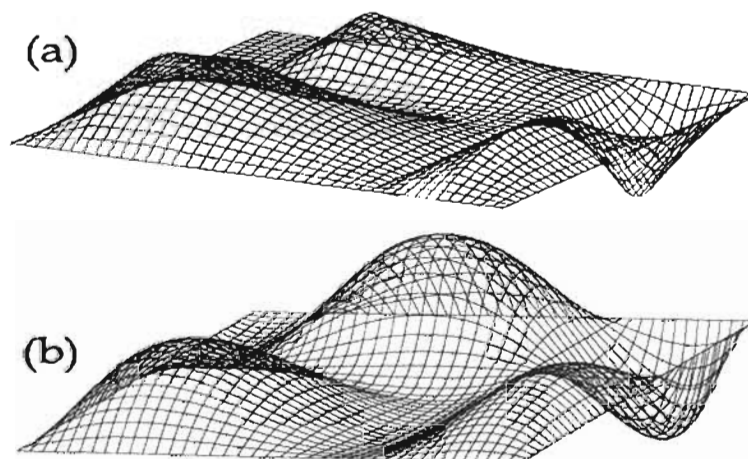


Figura 4.33: (a) Superficie de interpolación con $p = 1$ y $q = 2$. (b) Superficie de interpolación con $p = 3$ y $q = 2$.

Capítulo 5

Técnicas especiales para la construcción de superficies.

Finalizaremos este trabajo presentando varias técnicas para la construcción de superficies NURBS un poco más complicadas de las que hasta ahora hemos hecho, para lo cual utilizaremos los algoritmos descritos en el capítulo anterior.

Estas superficies tienen múltiples aplicaciones sobre todo en las áreas de los diseños gráficos asistidos por computadora, es decir los llamados “Computer Geometry Aided Design” (CAGD).

5.1. Superficies “Swung”

La palabra en Inglés “Swung”, se puede traducir como “oscilación”, “balanceo” o “dar vueltas libremente al rededor de algo”, significados que describen adecuadamente la definición matemática de esta superficie. Ya que como veremos, son el caso general de una superficie de revolución pues resulta de la rotación de una curva al rededor del eje z siguiendo la trayectoria de otra. Adicionalmente se le puede aplicar ciertos factores de escala.

Así, supongamos que la curva perfil (“profile”), definida en el plano xz con vector de nodos U , es

$$P(u) = \sum_{i=0}^n R_{i,p}(u)P_i = \left(\sum_{i=0}^n R_{i,p}(u)x_i, 0, \sum_{i=0}^n R_{i,p}(u)z_i \right) = (P_x(u), 0, P_z(u))$$

y la curva (“trajectory”) está definida en el plano xy con vector de nodos V , es

$$T(v) = \sum_{j=0}^m R_{j,q}(v)T_j = \left(\sum_{j=0}^m R_{j,q}(v)x_j, \sum_{j=0}^m R_{j,q}(v)y_j, 0 \right) = (T_x(v), T_y(v), 0)$$

Entonces una superficie “Swung” definida por Woodward [Wood87] está dada por

$$S(u, v) = (\alpha P_x(u)T_x(v), \alpha P_x(u)T_y(v), P_z(u)) \quad (5.1)$$

con $\alpha \in \mathbb{R}$.

La interpretación geométrica de (5.1) se deduce notando que al fijar $u = \bar{u}$

$$S(\bar{u}, v) = C_{\bar{u}}(v) = (\beta T_x(v), \beta T_y(v), P_z(\bar{u}))$$

con $\beta = \alpha P(\bar{u})$, es decir la superficie "Swung" es una isocurva de forma similar a $T(v)$, pero con un factor de escala β para la entrada en x y y en un plano paralelo al xy de altura $P_z(\bar{u})$.

Análogamente si fijamos $v = \bar{v}$ tenemos

$$S(u, \bar{v}) = C_{\bar{v}}(u) = (\beta_x P_x(u), \beta_y P_y(u), P_z(u))$$

con $\beta_x = \alpha T_x(\bar{v})$ y $\beta_y = \alpha T_y(\bar{v})$. Que resulta en una isocurva con forma similar a $P(u)$, pero rotada hacia un plano perpendicular a xy que contiene el vector (β_x, β_y) , con factores de escalamiento β_x para la entrada x y β_y para la entrada y , por ejemplo en la Figura 5.1, se muestra una transformación de este tipo sobre algún punto $(P_x(u), 0, P_z(u))$.

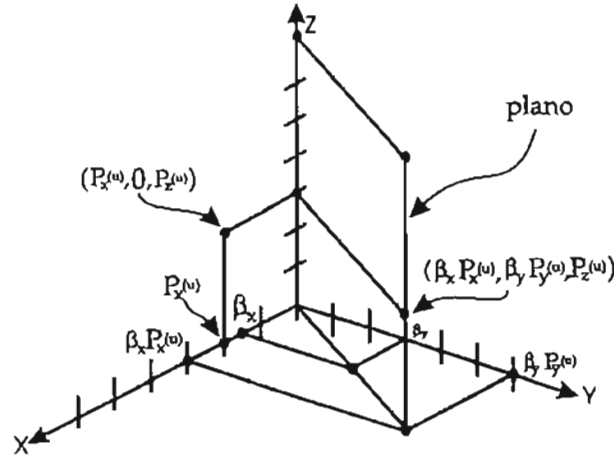


Figura 5.1: Transformación de un punto de una curva perfil en la superficie "Swung".

En el caso cuando $\alpha = 0$ la superficie resulta

$$S(u, v) = (0, 0, P_z(u)) = (0, 0, \sum_{i=0}^n R_{i,p}(u) z_i)$$

que es un conjunto de puntos sobre el eje z .

Entonces por la propiedad de invarianza bajo transformaciones afines de la superficie NURBS, se deduce que es posible obtener una representación de la superficie "Swung" realizando la transformación (5.1) a los puntos de control, es decir la superficie NURBS está dada por

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R(u, v)_{i,p,j,q} Q_{i,j}$$

en donde

$$Q_{i,j} = (\alpha P_{i,x} T_{j,x}, \alpha P_{i,x} T_{j,y}, P_{i,z})$$

y

$$w_{i,j} = w_i w_j$$

En las Figuras (5.2), (5.3) y (5.4) se presentan tres ejemplos de superficies "Swung", en donde las curvas de trayectoria están etiquetadas con (a), las curvas perfil con (b) y la superficies resultantes con (c), todas con un factor de escala $\alpha = 0.15$, los puntos de control para la curva perfil y de trayectoria están dadas en las siguiente tablas.

Puntos de control de la superficie de la Figura 5.2					
P_i	(10, 0, 5)	(5, 0, 5)	(6, 0, 15)	(5, 0, 15)	
T_i	(5, 0, 0)	(5, -2, 0)	(10, -2, 0)	(10, 3, 0)	(10, 3, 0)
	(10, 8, 0)	(10, 8, 0)	(10, 13, 0)	(5, 13, 0)	(5, 13, 0)
	(3, 13, 0)	(3, 13, 0)	(-2, 13, 0)	(-2, 8, 0)	(0, 8, 0)

Puntos de control de la superficie de la Figura 5.3					
P_i	(10, 0, 5)	(5, 0, 5)	(10, 0, 13)	(6, 0, 15)	(5, 0, 15)
T_i	(5, 0, 0)	(5, -2, 0)	(10, -2, 0)	(10, 3, 0)	(9, 3, 0)
	(9, 8, 0)	(10, 8, 0)	(10, 13, 0)	(5, 13, 0)	(5, 12, 0)
	(3, 12, 0)	(3, 13, 0)	(-2, 13, 0)	(-2, 8, 0)	(0, 8, 0)

Puntos de control de la superficie de la Figura de la superficie 5.4					
P_i	(10, 0, 5)	(5, 0, 5)	(0, 0, 10)	(10, 0, 13)	(6, 0, 15)
	(5, 0, 15)				
T_i	(5, 0, 0)	(5, -2, 0)	(10, -2, 0)	(10, 3, 0)	(5, 3, 0)
	(5, 8, 0)	(10, 8, 0)	(10, 13, 0)	(5, 13, 0)	(5, 10, 0)
	(3, 10, 0)	(3, 13, 0)	(-2, 13, 0)	(-2, 8, 0)	(0, 8, 0)

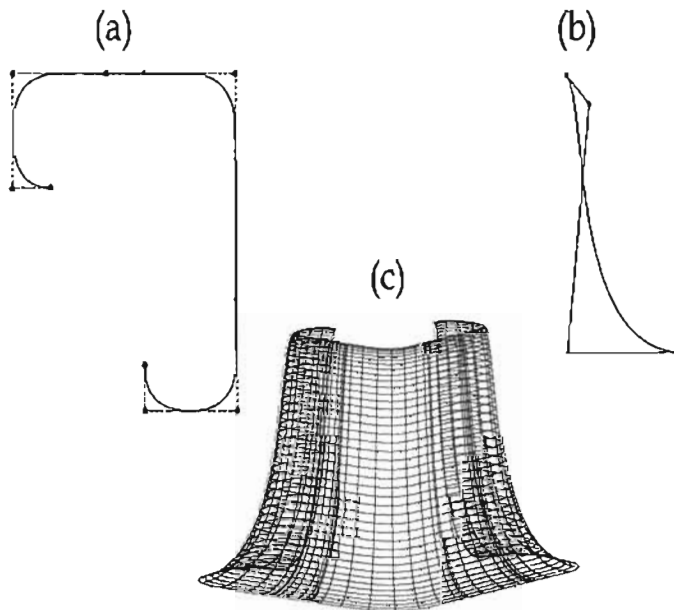


Figura 5.2: Superficie "Swung": (a) Curva nurbs de trayectoria cuadrática. (b) Curva nurbs de perfil cúbica. (c) Superficie resultante.

Por supuesto también es posible hacer variaciones del factor de escalar α de tal manera que obtengamos distintas superficies, en la Figura (5.5) se muestran distintos valores para este valor y las superficies resultantes. En particular notemos que si tomamos $\alpha = 1$ y la curva de trayectoria $T(v)$ es el círculo unitario con centro en el origen, entonces la superficie "Swung", es la superficie de revolución que presentamos en el capítulo 3, ejemplo 23.

En las Figura (5.6) se diseñaron dos superficies libres, en la primera se representa un florero, y en la segunda una muela.

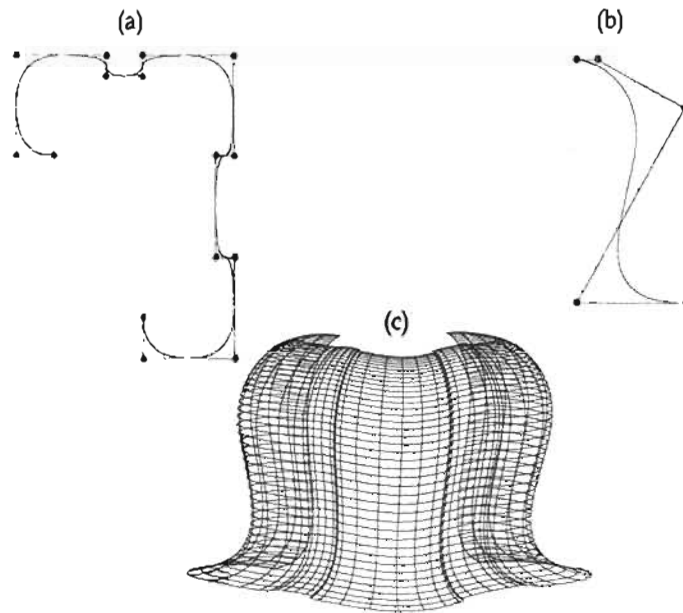


Figura 5.3: Superficie "Swung": (a) Curva nurbs de trayectoria cuadrática. (b) Curva nurbs de perfil cúbica. (c) Superficie resultante.

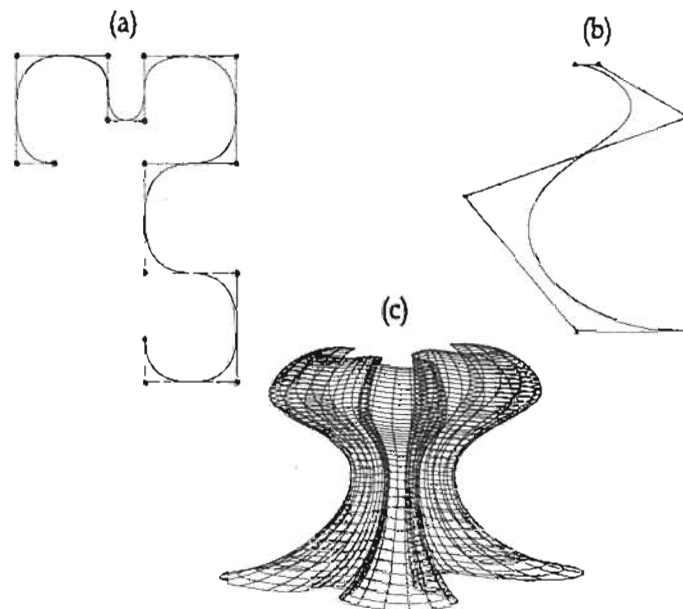


Figura 5.4: Superficie "Swung": (a) Curva nurbs de trayectoria cuadrática. (b) Curva nurbs de perfil cúbica. (c) Superficie resultante.

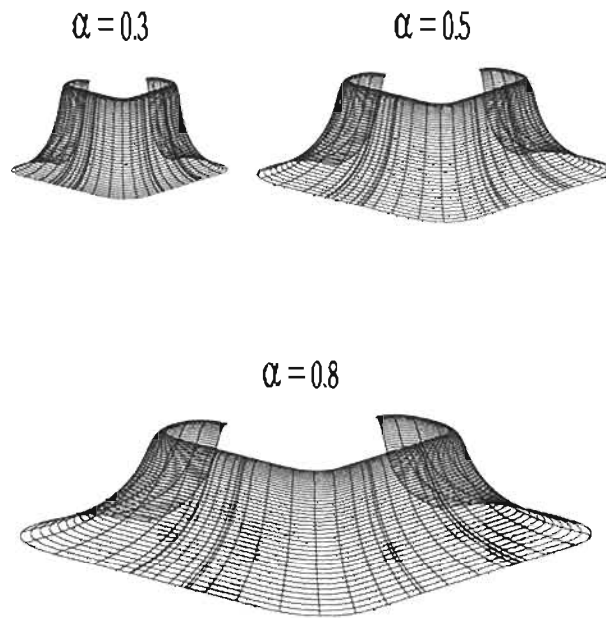


Figura 5.5: Superficies "Swung" que resulta de variar el factor de escala.

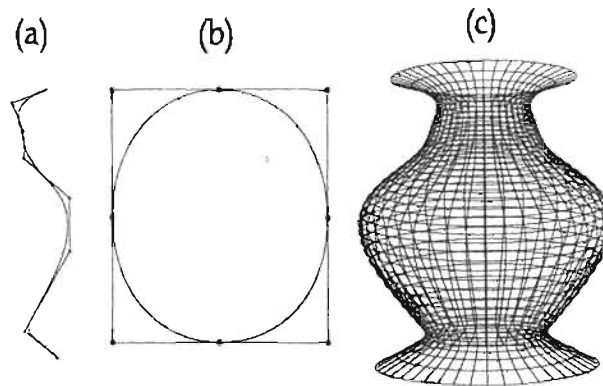


Figura 5.6: Superficie libre "Swung": (a) Curva perfil. (b) Curva trayectoria. (c) Superficie resultante.

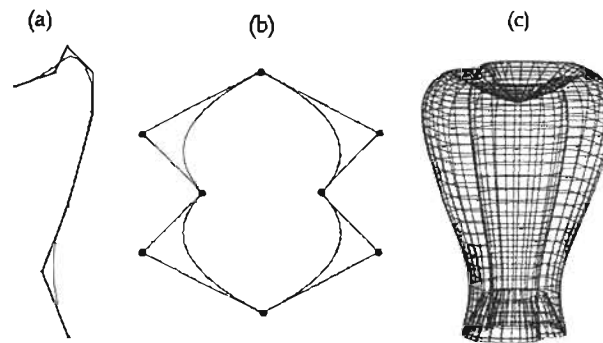


Figura 5.7: Superficie libre "Swung": (a) Curva perfil. (b) Curva trayectoria. (c) Superficie resultante.

5.2. Superficies "Skinned".

Una superficie "skinned", es aquella que resulta de la interpolación de los puntos de control de un conjunto de curvas NURBS en el espacio, de tal manera que éstas se convierten en sus isocurvas.

Así aunque intuitivamente su construcción parezca simple, notemos que no se está suponiendo que las curvas NURBS deban de cumplir ciertas características, es decir tanto sus grados como el número de puntos de control pueden ser cualesquiera, sin embargo por definición de una superficie de producto tensor esto no debe ser así, por lo que es necesario transformarlas sin modificar su forma.

5.2.1. Superficies regladas.

La superficie "skinned" más simple es la llamada reglada, la cual está formada sólo por dos curvas NURBS, iniciaremos estudiandola ya que su construcción nos permitirá generalizar de forma muy simple las "skinned" generales.

Sean dos curvas NURBS

$$C_k(u) = \sum_{i=0}^{n_k} R_{i,p_k}(u) P_i^k$$

definidas sobre los vectores de nodos $U^k = \{u_0^k, \dots, u_{m_k}^k\}$ para $k = 1, 2$.

Entonces, una superficie reglada $S(u, v)$ es aquella que tiene como isocurvas de los extremos en dirección del parámetro u , las curvas $C_1(u)$, $C_2(u)$, y para $u = \bar{u}$ fijo, $S(\bar{u}, v)$ es una interpolación lineal entre los puntos $C_1(\bar{u})$ y $C_2(\bar{u})$.

Ahora si hacemos $P_0 = C_1(\bar{u})$ y $P_1 = C_2(\bar{u})$, de la sección 3.1 sabemos que esta interpolación se puede representar en términos NURBS como

$$C(v) = P_0(1-v) + P_1v = \sum_{i=0}^1 R_{i,1}(v) P_i$$

con $w_0 = w_1 = 1$ y $V = \{0, 0, 1, 1\}$.

Entonces se deduce que una superficie reglada debe tener la forma

$$S(u, v) = R_{i,p_i,j}(u, v) P_{i,j} \quad (5.2)$$

en donde $P_{i,0} = P_i^1$ y $P_{i,1} = P_i^2$ y el grado p está en función de p_0 y p_1 .

Por otro lado, como se mencionó anteriormente, no se está suponiendo que ni los grados p_k ni el número de puntos n_k (para $k = 1, 2$) coincidan, por lo que es necesario igualarlos. Para esto se utilizan el algoritmo de elevación de grado (ver sección 4.3) y el de refinamiento (ver sección 4.1.1), utilizando el proceso que se presenta en los siguientes pasos.

- i) En caso de ser necesario redefinir los vectores de nodos U_1 y U_2 , de tal manera que tengamos el mismo rango parametral para u , es decir debe suceder que $u_0^1 = u_0^2$ y $u_{m_1}^1 = u_{m_2}^2$.
- ii) Empatar los grados. Si $p_1 = p_2$ entonces tomar $p = p_1$. En caso contrario tomar $p = \max\{p_1, p_2\}$ y elevar el grado de la curva que sea menor a p .
- iii) Si U_1 y U_2 no son iguales, mezclarlos para obtener U , de tal manera que si u_j está en U entonces debe suceder que u_j esté en U_1 o U_2 . Además la multiplicidad máxima de cierto nodo u_j en U_1 o U_2 también debe estar en U .
- iv) Usando U , aplicar el algoritmo de refinamiento sobre ambas curvas, con lo que finalmente obtendremos el valor de n , los pesos $w_{i,j}$ y los puntos de control $\{P_{i,j}\}$.

Ejemplo 34 Construyamos una superficie reglada, con dos curvas NURBS $C_1(u)$ y $C_2(u)$ de grados $p_1 = 3$ y $p_2 = 2$, pesos $w_i^1 = w_j = 1$ para toda i, j , vectores de nodos y puntos de control (ver Figura 5.8 (a)).

$$U^1 = \{0, 0, 0, 0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1, 1, 1, 1\}$$

$$U^2 = \{0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1\}$$

Puntos de control			
$P_0^1 = (6, -5, 0)$	$P_1^1 = (1, -5, 0)$	$P_2^1 = (7, -5, 10)$	$P_3^1 = (9, -5, 3)$
$P_4^1 = (12, -5, 10)$	$P_5^1 = (14, -5, 0)$	$P_6^1 = (15, -5, 13)$	$P_0^2 = (6, 5, 0)$
$P_1^2 = (1, 5, 0)$	$P_2^2 = (7, 5, 10)$	$P_3^2 = (9, 5, -3)$	$P_4^2 = (12, 5, 10)$

Entonces siguiendo los pasos anteriores, lo primero que tenemos que hacer en este caso, es empatar los grados $C_1(u)$ y $C_2(u)$, es decir elevar a 3 el grado de $C_2(u)$ ¹. Aplicando el proceso de elevación de grado, resulta que la nueva curva tiene el vector de nodos y puntos de control (ver Figura 5.8 (b))

$$U^2 = \{0, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1, 1\}$$

Puntos de control			
$P_0^2 = (6, 5, 0)$	$P_1^2 = (2.667, 5, 0)$	$P_2^2 = (2, 5, 1.667)$	$P_3^2 = (6, 5, 8.333)$
$P_4^2 = (7.333, 5, 7.833)$	$P_5^2 = (8.666, 5, -0.833)$	$P_6^2 = (10, 5, 1.333)$	$P_7^2 = (12, 5, 10)$

Ahora mezclando los vectores de nodos U^1 y U^2 tenemos que

$$U = \{0, 0, 0, 0, \frac{1}{4}, \frac{1}{3}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{2}{3}, \frac{3}{4}, 1, 1, 1, 1\}$$

Finalmente aplicando el algoritmo de refinamiento, los puntos de control de las dos curvas son

Puntos de control			
$P_0^1 = (6, -5, 0)$	$P_1^1 = (1, -5, 0)$	$P_2^1 = (5, -5, 6.667)$	$P_3^1 = (6.925, -5, 5.037)$
$P_4^1 = (8.130, -5, 3.260)$	$P_5^1 = (9.092, -5, -0.592)$	$P_6^1 = (10.351, -5, 3.260)$	$P_7^1 = (11.333, -5, 5.037)$
$P_8^1 = (12.667, -5, 6.667)$	$P_9^1 = (14, -5, 0)$	$P_{10}^1 = (15, -5, 13)$	
$P_0^2 = (6, 5, 0)$	$P_1^2 = (3.5, 5, 0)$	$P_2^2 = (2.1667, 5, 1.25)$	$P_3^2 = (3.5, 5, 4.167)$
$P_4^2 = (5, 5, 6.667)$	$P_5^2 = (6.667, 5, 8.083)$	$P_6^2 = (7.667, 5, 5.667)$	$P_7^2 = (8.1667, 5, 2.417)$
$P_8^2 = (9, 5, -0.301)$	$P_9^2 = (10.5, 5, 3.5)$	$P_{10}^2 = (12, 5, 10)$	

en donde además $w_{i,j} = 1$ para toda i, j . (ver Figura 5.9(a)). La superficie resultante se muestra en la Figura 5.9 (d).

□

¹Claramente también es posible reducir el grado de $C_1(u)$ este algoritmo se describe en [PT97, pág. 212]

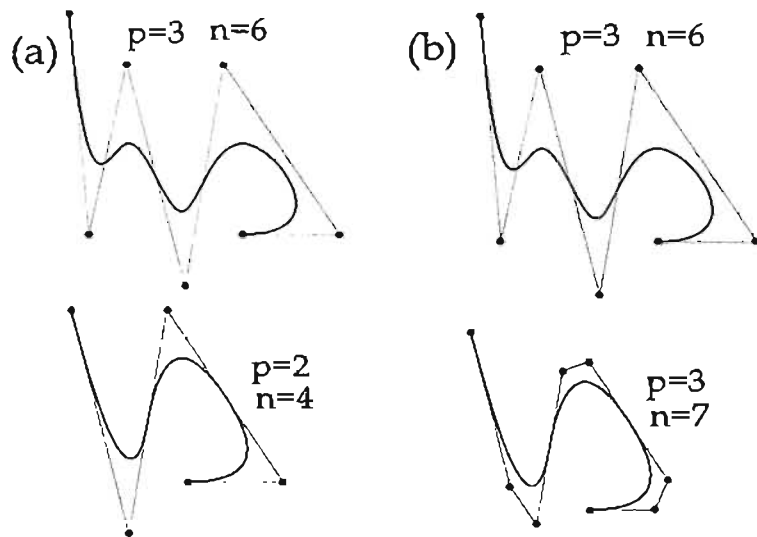


Figura 5.8: (a) Las dos curvas con las que se construirá la superficie reglada. (b) Elevación de grado de la segunda curva.

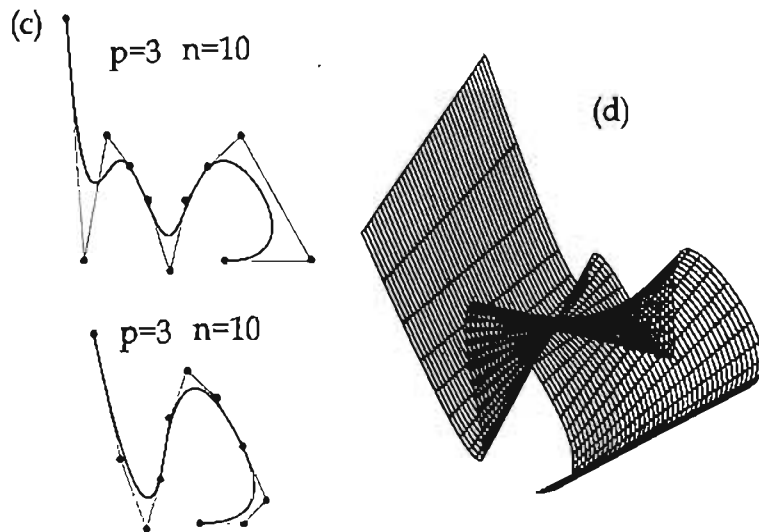


Figura 5.9: (c) Refinamiento del vector de nodos. (d) Superficie reglada resultante.

Un caso particular de una superficie reglada son los conos generales. Si por ejemplo suponemos que P es el vértice del cono y $C_1(u)$ es la curva base, entonces por la propiedad de la envoltura convexa de las curvas NURBS podemos hacer $C_2(u)$ represente a P tomando $P_i^2 = P$ y $w_i^2 = w_i^1$ para toda i . Así el cono deseado resulta de construir la superficie reglada con estas dos curvas.

Ejemplo 35 Tomemos los puntos de control de la curva $C_1(u)$ como

Puntos de control			
(6, 0, 0)	(1, 0, 0)	(7, 10, 0)	(7, -1, 0)
(12, 10, 0)	(14, 0, 0)	(6, 0, 0)	

Y los respectivos para $C_2(u)$ como $P_i^2 = (12, 0, 0)$. El cono resultante se presenta en la Figura 5.10.

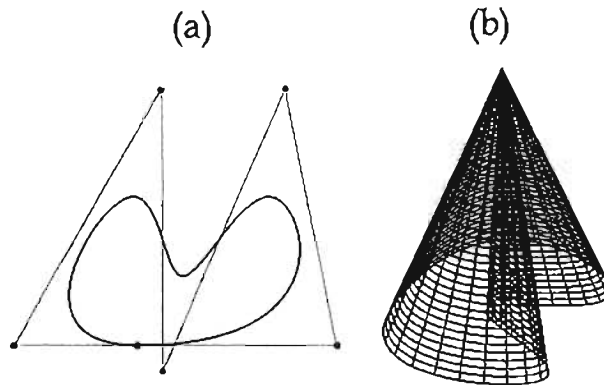


Figura 5.10: Cono general. (a) Curva NURBS base del cono. (b) Cono resultante.

□

5.2.2. Superficies "Skinned" generales.

Una vez discutido el caso más particular de la superficies "Skinned", es bastante directo establecer la construcción de la superficie "Skinned" en general. Dado un conjunto de K curvas NURBS

$$C_k^w(u) = \sum_{i=0}^n R_{i,p}(u) P_{i,k}^w \tag{5.3}$$

para $k = 0, \dots, K$, que supondremos están definidas sobre un mismo vector de nodos U y son del mismo grado p , ya que en caso de no ser así, podemos utilizar los pasos descritos para las superficies regladas.

Una superficie "Skinned" $S(u, v)$, está formada por la interpolación sobre el parámetro v , de los puntos de control de todas las $C_k^w(u)$. Con lo que según la sección 4.5 implica que las curvas (5.3) son isocurvas de la superficie.

Es importante notar que la técnica de interpolación que estudiamos está diseñada para curvas no racionales, y así para calcular la interpolación de los puntos de control se deben usar las versiones homogéneas. Esto como veremos más adelante, puede traer ciertos problemas en la forma final de la superficie.

La única restricción para la elección de un grado común p es que $0 < p \leq K$, ya que en caso contrario el sistema de ecuaciones que resulta de las interpolaciones tendrían más ecuaciones que incógnitas. Los parámetros $\{\bar{v}_k\}$ se puede seleccionar de distintas maneras, una posibilidad es tomar [PT97, pág.462]

$$\bar{v}_k = \bar{v}_{k-1} + \frac{1}{n+1} \sum_{i=0}^n \frac{|P_{i,k} - P_{i,k-1}|}{d_i} \quad \text{para } k = 1, \dots, K-1 \tag{5.4}$$

donde d_i denota la longitud de arco total de los puntos $P_{i,0}, \dots, P_{i,k}$. Por último el vector de nodos U se puede obtener usando las abscisas de Greville (4.37) usando los valores $\{\bar{v}_k\}$.

Ejemplo 36 Construyamos una superficie "Skinned", con las NURBS $C_0^w(u)$, $C_2^w(u)$, $C_3^w(u)$, $C_4^w(u)$ de grados 1,2,3,2 y 1 respectivamente, las cuales tienen como puntos de control (ver Figura 5.11)

Puntos de control curvas que forman la superficie "Skinned".						
C_0^w	(13, 15, 40)	(10, 15, 40)	(7, 15, 40)	(-3, 15, 40)	(-6, 15, 40)	(-9, 15, 40)
C_1^w	(13, 5, 40)	(12, 5, 46)	(5, 5, 45)	(-3, 5, 37)	(-7, 5, 48)	(-9, 5, 40)
C_2^w	(13, 0, 40)	(12, 0, 49)	(7, 0, 42)	(-5, 0, 35)	(-6, 0, 52)	(-9, 0, 40)
C_3^w	(13, -5, 40)	(10, -5, 39)	(6, -5, 45)	(-1, -5, 40)	(-7, -5, 40)	(-9, -5, 40)
C_4^w	(13, -10, 40)	(10, -10, 40)	(7, -10, 40)	(-3, -10, 40)	(-6, -10, 40)	(-9, -10, 40)

En las Figura 5.12 se presenta la superficie resultante usando una interpolación lineal, en la Figura 5.13 con interpolación cuadrática y en la Figura 5.15 con interpolación cúbica.

□

Como mencionamos al inicio de capítulo, este tipo de superficies tienen múltiples aplicaciones en los diseños gráficos asistidos por computadora, por ejemplo en la Figura 5.15 se presenta una superficie "Skinned" que representa un "rostro humano" la cual se construyó usando el "software" Rhino².

²Para más información a cerca de software ver en <http://www.rhino3d.com/>.

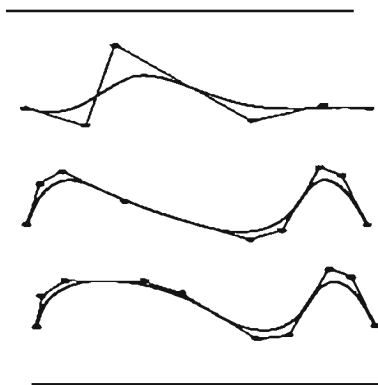


Figura 5.11: Isocurvas de la superficie "Skinned".

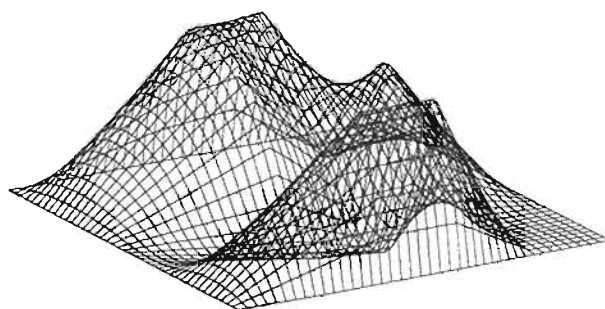


Figura 5.12: Superficie "Skinned", obtenida con las curvas anteriores haciendo una interpolación lineal.

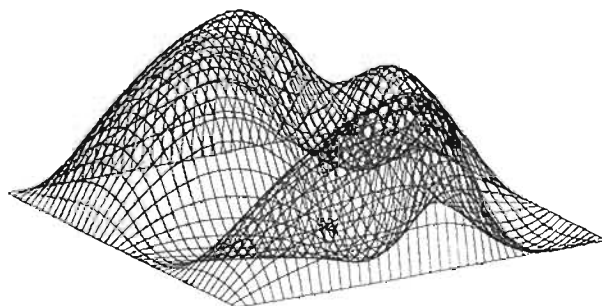


Figura 5.13: Superficie "Skinned", obtenida con las curvas anteriores haciendo una interpolación cuadrática.

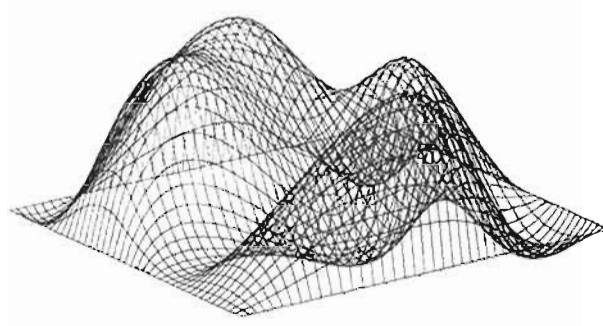


Figura 5.14: Superficie "Skinned" obtenida con las curvas anteriores haciendo una interpolación cúbica.

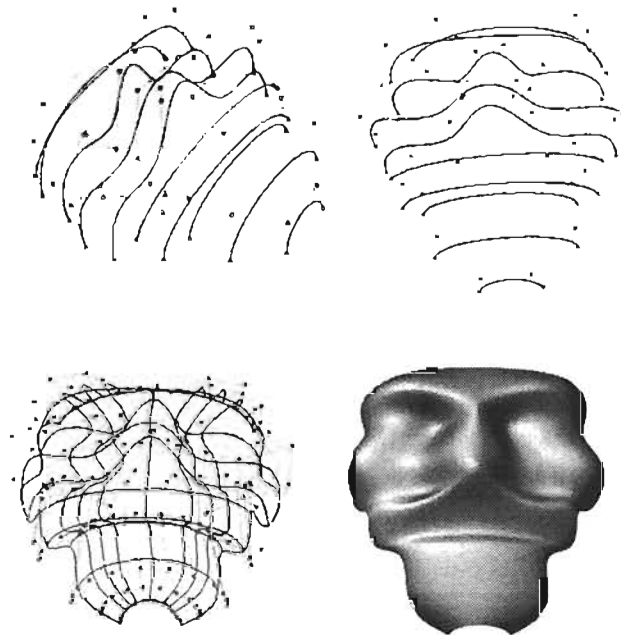


Figura 5.15: Cara formada con una superficie "Skinned".

5.2.3. Superficies “Skinned Spine”.

Un método útil en la construcción de las superficie “Skinned”, que ayuda a posicionar y orientar las curvas (5.3), es el que utiliza la llamada curva “Spine”, dada por

$$C_s(v) = \sum_{i=0}^{n_s} R_{i,p_s} P_i$$

Con ésta, las curvas (5.3) son colocadas sobre los puntos $C_s(\hat{v}_k)$, en ciertas posiciones $C_k^w(\hat{u}_k)$, en donde los parámetros \hat{v}_k y \hat{u}_k con $k = 0, \dots, K$, pueden ser seleccionados por el diseñador de la superficie. Sin embargo es importante mencionar que que sus valores determinarán en gran medida la forma de la superficie resultante, y así se debe tener cuidado con su selección.

Para el caso de \hat{v}_k , también es posible usar (5.4) o algún método descrito en la sección 4.4, usando los puntos de control de la curva “Spine”, sin embargo esto puede ser peligroso en los casos cuando la curva tenga una mala parametrización. En la Figura 5.16 se presentan dos formas de seleccionar \hat{v}_k y \hat{u}_k , suponiendo que el rango parametral de todas es el intervalo $[0, 1]$.

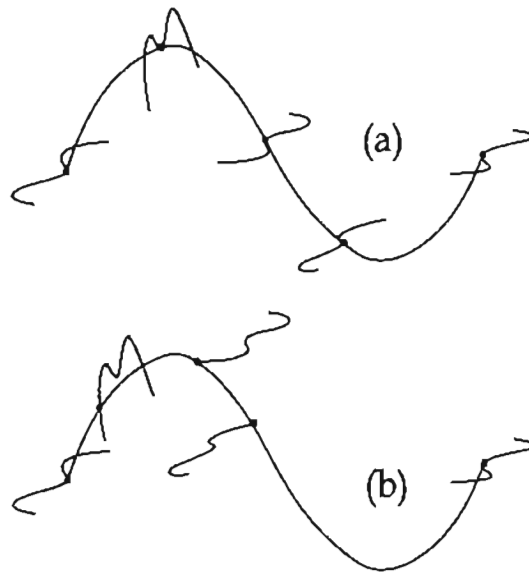


Figura 5.16: Posicionamiento de las isocurvas sobre la curva “Spine”. (a) $\hat{V} = \{0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1\}$ y $\hat{U} = \{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\}$ (b) $\hat{V} = \{0, \frac{1}{6}, \frac{2}{6}, \frac{1}{2}, 1\}$ y $\hat{U} = \{\frac{1}{2}, \frac{1}{6}, 0, 1, \frac{1}{2}\}$.

Es interesante notar que la curva “Spine”, puede contribuir con información para la forma de la superficie “Skinned”. Por ejemplo es posible hacer que las curvas (5.3) se oriente de tal manera que el plano que las contenga³ sea perpendicular al vector tangente del punto $C_s(\hat{v}_k)$, lo cual permitirá que la superficie resultante sea más acorde a la estructura de la curva, un ejemplo se muestra en la Figura 5.17.

También, es posible hacer que las curvas de interpolación en el parámetro v , no sólo interpolen los puntos de control $P_{i,0}, \dots, P_{i,k}$, (en donde cada $P_{i,j}$ es el i -ésimo punto de control de la j -ésima curva para $j = 0, \dots, K$) sino que también lo hagan con ciertos vectores $D_{i,0}, \dots, D_{i,k}$ paralelos a las derivadas de $C_s(\hat{v}_k)$.

³Siempre y cuando las curvas sean planares, en caso de no ser así el diseñador deberá decidir si trabajar con las versiones planares o con otro método.

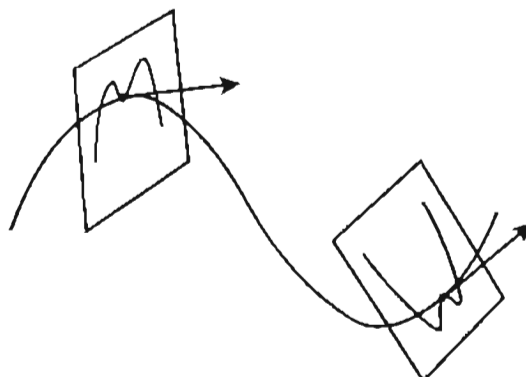


Figura 5.17: Las curvas deben ser orientadas de tal manera que su derivada en el punto $C_s(\hat{v}_k)$ se perpendicular al plano que la contiene.

Es decir, si tomamos $P_k = C_s(\hat{v}_k)$, $D_k = C'_s(\hat{v}_k)$ para $k = 0, \dots, K$ y

$$d_k = |P_k - P_{k-1}| \quad k = 1, \dots, K$$

los $D_{i,k}$ pueden ser calculados haciendo

$$D_{i,k} = \frac{|P_{i,k+1} - P_{i,k}| + |P_{i,k} - P_{i,k-1}|}{d_{k+1} + d_k} D_k \quad (5.5)$$

y

$$\begin{aligned} D_{i,0} &= \frac{|P_{i,1} - P_{i,0}|}{d_1} D_0 \\ D_{i,K} &= \frac{|P_{i,K} - P_{i,K-1}|}{d_K} D_K \end{aligned} \quad (5.6)$$

Con lo que las derivadas de las v -isocurva en \hat{v}_k , serán paralelas a los vectores D_k , es decir

$$S_v(u, \bar{v}_k) \parallel D_k \quad \text{para toda } u \text{ y } k = 0, \dots, K \quad (5.7)$$

Ahora si, $m = 2K + 1$, cada i -ésima curva de interpolación, determinará el sistema de $m + 1$ ecuaciones

$$\begin{aligned} P_{i,0} &= \sum_{j=0}^m N_{j,q}(\bar{v}_0) Q_{i,j} \\ D_{i,0} &= \sum_{j=0}^m N'_{j,q}(\bar{v}_0) Q_{i,j} \\ &\vdots \\ P_{i,K} &= \sum_{j=0}^m N_{j,q}(\bar{v}_K) Q_{i,j} \\ D_{i,K} &= \sum_{j=0}^m N'_{j,q}(\bar{v}_K) Q_{i,j} \end{aligned}$$

en donde el vector de nodos de cada una de estas curvas, debe contener $(m+1)+q+1 = m+q+2$, para calcularlos es posible usar las ecuaciones (4.40) y (4.41) para los casos $q = 2$ y $q = 3$ respectivamente.

Ejemplo 37 Construyamos una superficie "Skinned Spine", en donde la curva "Spine" es la que se presenta en la Figura 5.18, la cual es una curva NURBS de interpolación de los puntos

Puntos de interpolación de la curva "Spine".			
$(0, -80, 80, 1)$	$(0, -50, 100, 1)$	$(0, 30, -50, 1)$	$(0, 70, -20, 1)$

Las isocurvas serán las que se presentan en la Figura 5.19, éstas tiene como puntos de control orientados de acuerdo a los planos perpendiculares a la derivadas los siguientes valores, las cuales colocaremos tomando $\tilde{U} = \{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\}$ y $\tilde{V} = \{0, \frac{1}{4}, \frac{3}{4}, 1\}$ como se muestra en la Figura 5.20.

Puntos de control curvas que forman la superficie "Skinned Spine".			
$C_0^w(u)$	$C_1^w(u)$	$C_2^w(u)$	$C_3^w(u)$
$(90, -90, 95, 1)$	$(90, -20, 111, 25, 1)$	$(90, 30, -23, 75, 1)$	$(90, 70, -20, 1)$
$(50, -80, 80, 1)$	$(50, -10, 123, 98, 1)$	$(50, 20, -41, 528, 1)$	$(70, 80, -33, 33, 1)$
$(30, -100, 110, 1)$	$(20, -20, 111, 25, 1)$	$(30, 50, 11, 806, 1)$	$(50, 60, -6, 66, 1)$
$(0, -80, 80, 1)$	$(0, -47, 5, 76, 25, 1)$	$(0, 30, -23, 75, 1)$	$(0, 70, -20, 1)$
$(-30, -100, 110, 1)$	$(-20, -20, 111, 25, 1)$	$(-30, 50, 11, 81, 1)$	$(-50, 60, -6, 66, 1)$
$(-50, -80, 80, 1)$	$(-50, -10, 124, 1)$	$(-50, 20, -41, 53, 1)$	$(-70, 80, -33, 33, 1)$
$(-90, -90, 95, 1)$	$(-90, -20, 111, 25, 1)$	$(-90, 30, -23, 75, 1)$	$(-90, 70, -20, 1)$

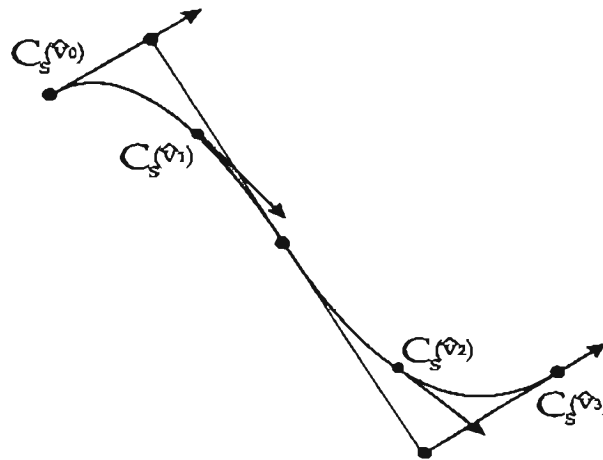


Figura 5.18: Curva "Spine" y puntos en donde serán posicionas las isocurvas.

Una vez hecho esto, las superficie resultante se muestra en la Figura 5.21. En la Figura 5.22 se muestra la misma superficie pero en este caso no se realizó interpolación de vectores paralelos a las derivas.

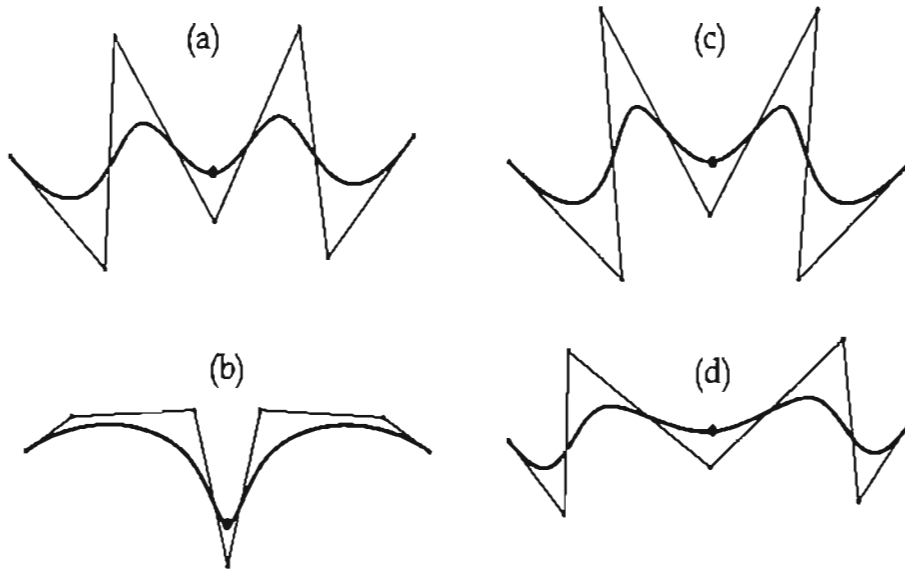


Figura 5.19: Isocurvas, las cuales serán posicionadas sobre la curva "Spine".

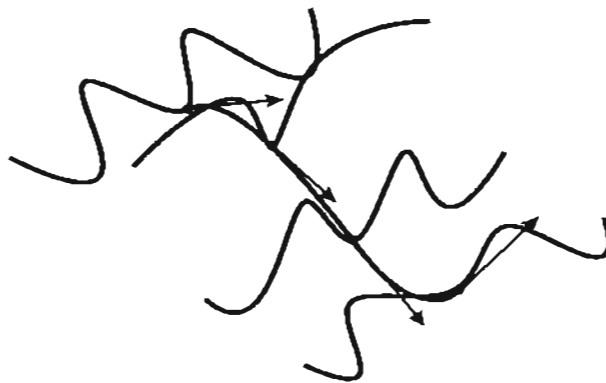


Figura 5.20: Posicionamiento y orientación de las isocurvas.

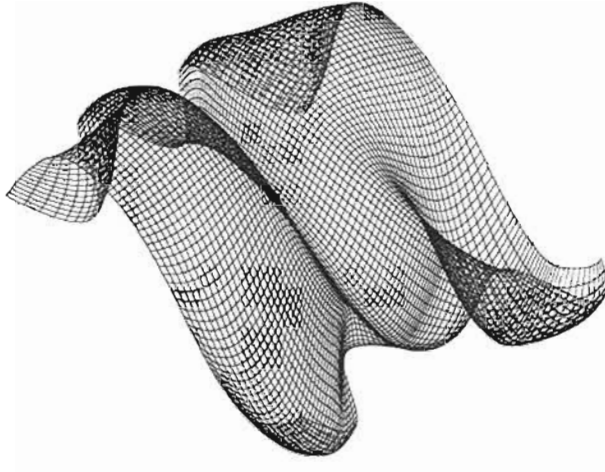


Figura 5.21: Superficie "Skinned Spine" resultante.

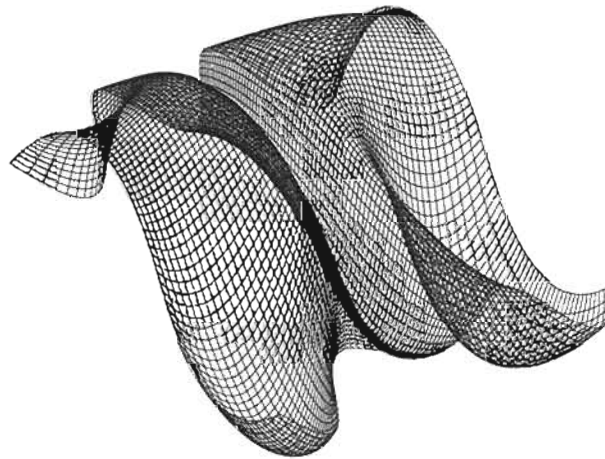


Figura 5.22: Superficie "Skinned Spine" sin interpolación de derivadas.

5.2. SUPERFICIES "SKINNED".

Si por ejemplo ahora tomamos $\hat{V} = \{0, \frac{1}{4}, \frac{3}{4}, 1\}$ los puntos resultantes son

Puntos de control curvas que forman la superficie "Skinned Spine".			
$C_0^w(u)$	$C_1^w(u)$	$C_2^w(u)$	$C_3^w(u)$
(90, -90, 95, 1)	(90, 9, 2, 60)	(90, 0, 8)	(90, 70, -20, 1)
(50, -80, 80, 1)	(50, 19, 6, 66, 5)	(50, -10, 1, 4)	(70, 80, -33, 33, 1)
(30, -100, 110, 1)	(20, 9, 2, 60)	(30, 20, 21)	(50, 60, -6, 66, 1)
(0, -80, 80, 1)	(0, -19, 8, 42, 3)	(0, 0, 8)	(0, 70, -20, 1)
(-30, -100, 110, 1)	(-20, 9, 2, 60)	(-30, 20, 21)	(-50, 60, -6, 66, 1)
(-50, -80, 80, 1)	(-50, 19, 6, 66, 5)	(-50, -10, 1, 4)	(-70, 80, -33, 33, 1)
(-90, -90, 95, 1)	(-90, 9, 2, 60)	(-90, 0, 8)	(-90, 70, -20, 1)

el nuevo posicionamiento y orientación se presenta en la Figura 5.23 y las superficies resultantes en las figuras 5.24 y 5.25.



Figura 5.23: Otro posicionamiento y orientación de las isocurvas.

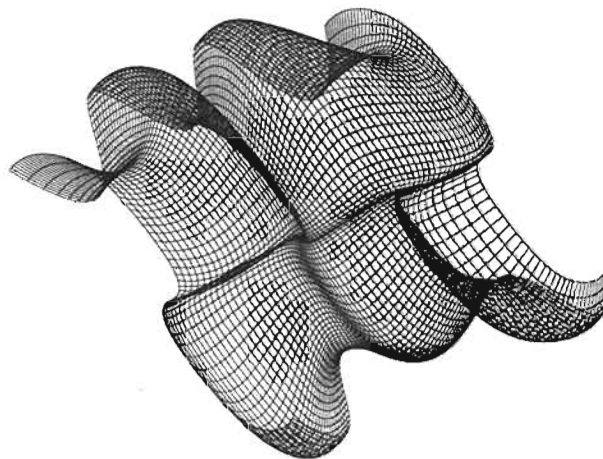


Figura 5.24: Superficie "Skinned Spine", con interpolación de derivadas.

□

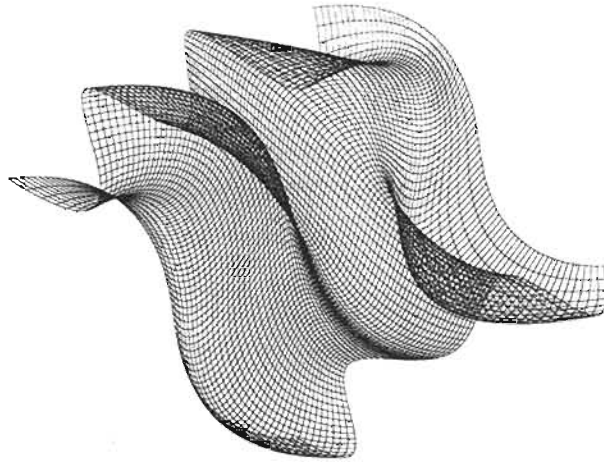


Figura 5.25: Superficie “Skinned Spine”, sin interpolación de derivadas.

5.2.4. Superficies “Skinned” con curvas racionales.

En el caso cuando algunas (o todas) de las curvas de (5.3) sean racionales, las interpolaciones de los puntos deben realizarse en el espacio homogéneo y los parámetros \bar{v}_k ser calculados con los puntos en este espacio. Es decir

$$\bar{v}_0 = 0 \quad \bar{v}_k = 1$$

$$\bar{v}_k = \bar{v}_{k-1} + \frac{1}{n+1} \sum_{i=0}^n \frac{|P_{i,k}^w - P_{i,k-1}^w|}{d_i^w}$$

para $k = 1, \dots, K-1$ en donde d_i^w denota la longitud de arco de $P_{i,0}^w, \dots, P_{i,k}^w$. Los nodos $\{v_i\}$ se tienen usando las abscisas de Greville (4.37) y los puntos finales de la superficie resultan aplicando el mapeo (2.21). Generalmente estas superficies resultan como se esperaba, sin embargo pueden existir algunos problemas como lo mostraremos a continuación.

Ejemplo 38 Sea la superficie reglada que se presenta en la parte inferior de la Figura 5.26, definida por las curvas NURBS cuadráticas $C_0^w(u)$ y $C_1^w(u)$ de la parte superior, que tiene como puntos de control

Puntos de control en el espacio homogéneo	
$C_0^w(u)$	$C_1^w(u)$
$P_i^w = (x_i^0, y_i^0, z_i^0, w_i^0)$	$P_i^w = (x_i^1, y_i^1, z_i^1, w_i^1)$
(20, -25, 0, 2)	(20, 25, 0, 2)
(10, -25, 20, 1)	(10, -25, 20, 1)
(-10, -25, 20, 1)	(-10, 25, 20, 1)
(-10, -25, 0, 2)	(-10, 25, 0, 2)

Es claro, que si multiplicamos algún conjunto de éstos, la respectiva curva en el espacio no homogéneo permanecerá sin cambio, es decir si por ejemplo multiplicamos por $\alpha \in \mathbb{R}$ los puntos de control de $C_0^w(u)$, por definición

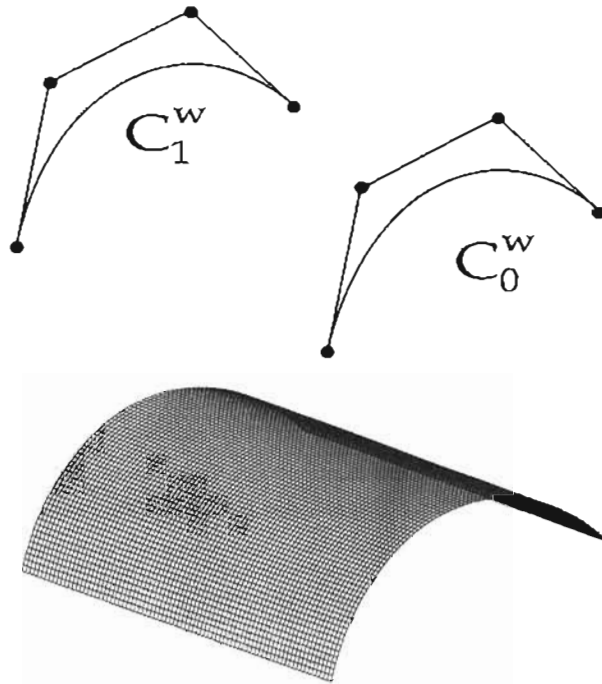


Figura 5.26: En la parte superior se presentan la curvas racionales con las que se construye la superficie reglada de la parte inferior.

$$\begin{aligned}
 C_0^w(u) &= \sum_{i=0}^3 N_{i,2}(u) \alpha P_i^w \\
 &= \left(\sum_{i=0}^3 N_{i,2}(u) w_i^0 \alpha x_i^0, \sum_{i=0}^3 N_{i,2}(u) w_i^0 \alpha y_i^0, \sum_{i=0}^3 N_{i,2}(u) w_i^0 \alpha z_i^0, \sum_{i=0}^3 N_{i,2}(u) w_i^0 \alpha \right)
 \end{aligned}$$

entonces aplicando el mapeo (2.21) tenemos que

$$\begin{aligned}
 H(C_0^w(u)) &= \\
 &= \left(\frac{\sum_{i=0}^3 N_{i,2}(u) w_i^0 \alpha x_i^0}{\sum_{i=0}^3 N_{i,2}(u) w_i^0 \alpha}, \frac{\sum_{i=0}^3 N_{i,2}(u) w_i^0 \alpha y_i^0}{\sum_{i=0}^3 N_{i,2}(u) w_i^0 \alpha}, \frac{\sum_{i=0}^3 N_{i,2}(u) w_i^0 \alpha z_i^0}{\sum_{i=0}^3 N_{i,2}(u) w_i^0 \alpha} \right) \\
 &= \left(\frac{\sum_{i=0}^3 N_{i,2}(u) w_i x_i^0}{\sum_{i=0}^3 N_{i,2}(u) w_i^0}, \frac{\sum_{i=0}^3 N_{i,2}(u) w_i y_i^0}{\sum_{i=0}^3 N_{i,2}(u) w_i^0}, \frac{\sum_{i=0}^3 N_{i,2}(u) w_i z_i^0}{\sum_{i=0}^3 N_{i,2}(u) w_i^0} \right) \\
 &= C_0(u)
 \end{aligned}$$

Sin embargo para un valor grande de α , la superficie resultante no tiene una buena parametrización en la dirección de v . Es decir, ya que la superficie reglada en términos de coordenadas homogéneas tiene

la forma

$$\begin{aligned} S^w(u, v) &= \sum_{i=0}^3 \sum_{j=0}^1 N_{i,2}(u) N_{j,1}(v) P_{i,j}^w \\ &= (1-v) \sum_{i=0}^3 N_{i,2}(u) P_{i,0}^w + v \sum_{i=0}^3 N_{i,2}(u) P_{i,1}^w \end{aligned}$$

en donde $P_{i,0}^w = (\alpha x_i^0, \alpha y_i^0, \alpha z_i^0, \alpha w_i^0)$ y $P_{i,1}^w = (x_i^1, y_i^1, z_i^1, w_i^1)$.

Así si fijamos $u = 0$, $S^w(u, v)$ se convierte en una interpolación lineal de los puntos $P_{0,0}$ a $P_{0,1}$ es decir

$$C_{u=0}^w(v) = (1-v)P_{0,0}^w + vP_{0,1}^w$$

la cual no tiene una buena parametrización para algún valor de α grande.

Por ejemplo supóngase que $\alpha = 20$ y evaluemos $C_{u=0}^w(v)$ en $v = \frac{1}{2}$, entonces se esperaría que el punto resultante estuviera en medio de la recta, es decir para nuestro ejemplo tendría que ser el valor $(10, 5, 0)$ (ya que el parámetro varía del 0 a 1) sin embargo notemos que

$$C_{u=0}^w\left(\frac{1}{2}\right) = \frac{1}{2} \begin{bmatrix} 400 \\ -200 \\ 0 \\ 40 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 20 \\ 10 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 200 \\ -100 \\ 0 \\ 20 \end{bmatrix} + \begin{bmatrix} 10 \\ 5 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 210 \\ -95 \\ 0 \\ 21 \end{bmatrix}$$

así aplicando el mapeo H definido en la ecuación (2.21), este punto es igual a $(10, -4.5, 0)$. La superficie completa se muestra en la parte superior de la Figura 5.27. En general, a medida de que el valor de α se incrementa las v -isocurvas se aproximarán más a $C_0(u)$ por ejemplo en la parte inferior de la Figura 5.27 se toma $\alpha = 50$.

5.3. Superficies de barrido.

Finalizaremos el capítulo, estudiando la representación NURBS de una interesante superficie llamada de barrido, la cual resulta de trasladar o "barrer" una curva siguiendo la trayectoria descrita por otra.

Tomemos dos curvas NURBS $T(v)$ y $C(u)$ en donde $T(v)$ representa la trayectoria y $C(u)$ es la curva que será trasladada (nosotros la denominaremos de barrido), entonces una superficie de barrido es aquella que tiene la forma general

$$S(u, v) = T(v) + C(u) \tag{5.8}$$

Esta definición es clara, notando que para un valor fijo de $v = \bar{v}$, la superficie se convierte en la curva

$$S(u, v) = C_{\bar{v}}(u) = T(\bar{v}) + C(u)$$

la cual es una copia de $C(u)$ trasladada por $T(\bar{v})$. De aquí además se puede deducir que esta clase de superficies son un caso general de los cilindros discutidos en el ejemplo 22 del capítulo 3, los cuales resultan de (5.8) cuando $T(v)$ es una recta.

Si ahora fijamos $u = \bar{u}$ tenemos que

$$S(\bar{u}, v) = T_{\bar{u}}(v) = T(v) + C(\bar{u})$$

la cual es una curva similar a $T(v)$ trasladada por $C(\bar{u})$. Aquí $T_{\bar{u}}(v)$ se convierte en isocurva de la superficie, si para algún valor \bar{u} , $C(\bar{u}) = 0$, ya que en este caso $S(\bar{u}, v) = T(v)$ para v .

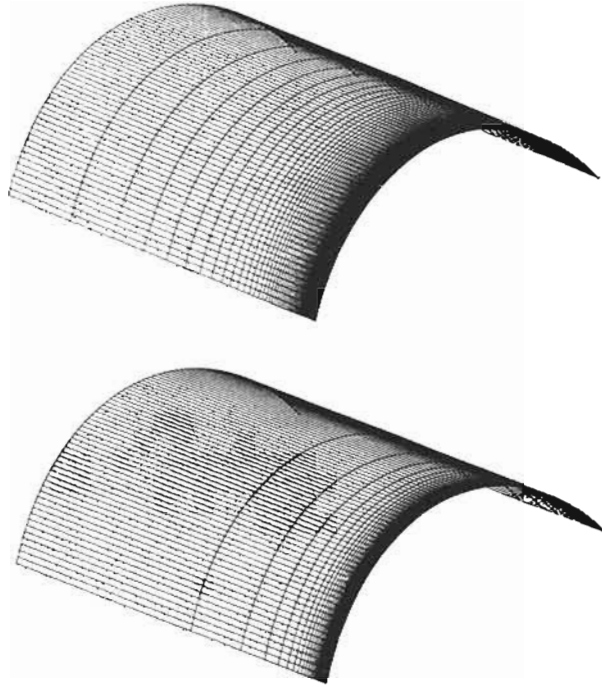


Figura 5.27: Superficies “Skinned” mal parametrizada en el parámetro v . En superficie superior $\alpha = 20$ y en la superficie inferior $\alpha = 50$.

Entonces si

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u)w_i^C Q_i}{\sum_{i=0}^n N_{i,p}(u)w_i^C}$$

con vector de nodos $U = \{u_0, \dots, u_r\}$. Y

$$T(v) = \frac{\sum_{j=0}^m N_{j,q}(v)w_j^T T_j}{\sum_{j=0}^m N_{j,q}(v)w_j^T}$$

con vector de nodos $V = \{v_0, \dots, v_s\}$.

La superficie NURBS que representa a (5.8) esta dada por

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(v)w_{i,j} P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(v)w_{i,j}} \quad (5.9)$$

en donde los vectores de nodos son U y V , los puntos de control

$$P_{i,j} = T_j + Q_i,$$

y los pesos $w_{i,j} = w_i^C w_j^T$ para $i = 0, \dots, n$ y $j = 0, \dots, m$.

A continuación presentaremos un ejemplo, de este tipo de superficies.

Ejemplo 39 Construyamos una superficie de barrido, tomando como curva trayectoria la curva NURBS cuadrática de la Figura 5.28 inciso (b) y como curva de barrido la curva NURBS cúbica del inciso (a), las cuales tiene los puntos de control,

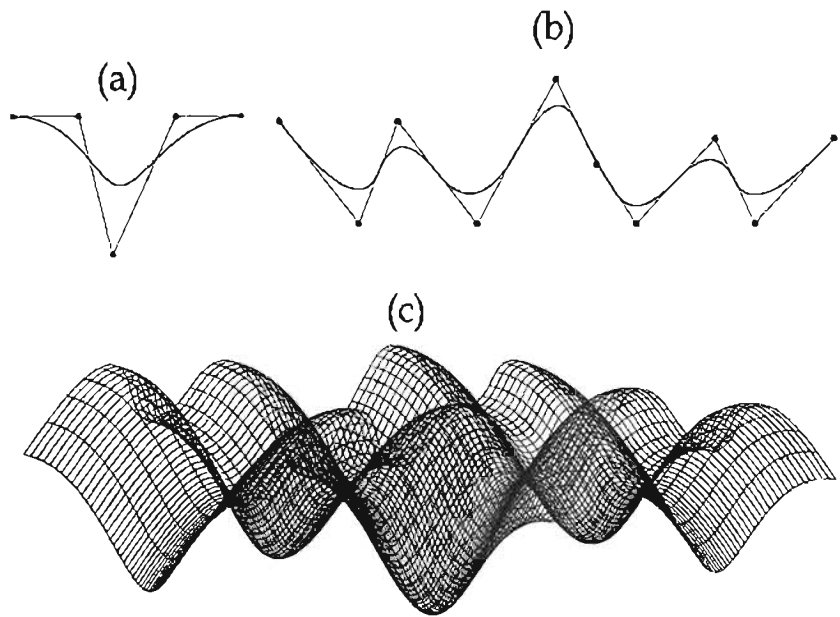


Figura 5.28: (a) Curva NURBS que sigue la trayectoria de la curva (b). (c) Superficie de barrido resultante.

Puntos de control.					
Q_i	$(-10, 0, 5)$	$(0, 0, 5)$	$(10, 0, 25)$	$(15, 0, 5)$	$(25, 0, 5)$
	$(0, -30, 5)$	$(0, -20, -7)$	$(0, -15, 5)$	$(0, -5, -7)$	$(0, 5, 10)$
T_j	$(0, 10, 0)$	$(0, 15, -7)$	$(0, 25, 3)$	$(0, 30, -7)$	$(0, 40, 3)$

vectores de nodos $U = \{0, 0, 0, 0, \frac{1}{2}, 1, 1, 1, 1, 1\}$, $V = \{0, 0, 0, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, 1, 1, 1\}$ y pesos $w_i^T = 1, w_j^C = 1$ para toda i, j .

Entonces por (5.9), la superficie es

$$S(u, v) = \frac{\sum_{i=0}^4 \sum_{j=0}^9 N_{i,3}(u) N_{j,2}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^4 \sum_{j=0}^9 N_{i,2}(u) N_{j,3}(v) w_{i,j}}$$

con $P_{i,j} = T_j + Q_i$, $w_{i,j} = w_i^T w_j^C$ para $i = 0, \dots, 4$ y $j = 0, \dots, 9$. Esta superficie se muestra en la Figura 5.28 inciso (c).

□

Una característica deseable de la superficie de barrido, es que la curva $C(u)$, pueda ser modificada a medida que se va trasladando, es decir, se le puedan aplicar ciertas transformaciones geométricas como rotaciones, escalamientos, etc., cuando el parámetro $v \in [v_0, v_s]$ cambie. Para este objetivo, se han desarrollado varias técnicas como la propuesta por Bloomenthal y Riesenfeld [Blom91] o Coquillar [Coqu87]. Nosotros presentaremos un algoritmo desarrollado por Piegl y Tiller [PT97, pág. 475], que usa técnicas "Skinned" (ver sección anterior) para su construcción.

Su procedimiento permite aplicar a la curva de barrido los factores de escala deseados a $K \in \mathbb{Z}^+$ copias de $C(u)$ por medio de una función que dependa de v , una vez hecho esto, las curvas son posicionadas en K puntos de la curva trayectoria de acuerdo a ciertos sistemas coordenados locales, y finalmente la superficie se construye interpolando los puntos de control de éstas, logrando así que las curvas sean isocurvas.

El valor de K es un parámetro que el diseñador de la superficie selecciona de acuerdo a sus intereses, y claramente entre mayor sea, la superficie resultante describirá mejor la trayectoria de la curva. Así, es recomendable al menos se tenga una copia de $C(u)$ relativa a cada punto de control de $T(v)$, por lo que si K es menor que $m + 1$ (condición que notemos también se puede expresar en términos de nodos como $s + 1 > K + q + 1$) haremos $K = m$.

En los casos cuando $K > m + 1$ (que en términos de nodos se escribe $s + 1 \leq K + q$) será necesario insertar⁴ $K - (m + 1) = K + q - r$ nodos más a V , los cuales pueden ser colocados en los puntos medios de los intervalos de longitud mayor.

Por otro lado, si denotamos las K copias de $C(u)$ como

$$C_k(u) = \frac{\sum_{i=0}^n N_{i,p}(u) w_i^C Q_i^k}{\sum_{i=0}^n N_{i,p}(u) w_i^C} \quad \text{con} \quad Q_{i,k} = (x_i^k, y_i^k, z_i^k) \quad (5.10)$$

y su respectivas versiones homogéneas

$$C_k^w(u) = \sum_{i=0}^n N_{i,p}(u) Q_i^w \quad \text{con} \quad Q_i^w = (w_i^C x_i^k, w_i^C y_i^k, w_i^C z_i^k, w_i^C)$$

con vector de nodos a V .

Podemos definir a la función de escala $s_v : [v_0, v_s] \rightarrow \mathbb{R}^3$ como

$$s_v = (f_x(v), f_y(v), f_z(v))$$

en donde $f_x, f_y, f_z : [v_0, v_s] \rightarrow \mathbb{R}$. Y aplicar los factores de escala a los $n + 1$ puntos de control $Q_i^k = (x_i^k, y_i^k, z_i^k)$ de la k -ésima copia, haciendo

$$(f_x(\bar{v}_k)x_i, f_y(\bar{v}_k)y_i, f_z(\bar{v}_k)z_i) = s_{v_k} \cdot Q_i^k$$

y para el caso homogéneo

$$(w_i^C f_x(\bar{v}_k)x_i, w_i^C f_y(\bar{v}_k)y_i, w_i^C f_z(\bar{v}_k)z_i, w_i^C) = s_{v_k} \cdot Q_{i,k}^w \quad (5.11)$$

con $i = 0, \dots, n$ y $k \in \{0, \dots, K - 1\}$.

Cada \bar{v}_k , representa el valor parametral con el que al evaluar $T(\bar{v}_k)$ resultará el punto en donde $C_k(u)$ (o $C_k^w(u)$) será posicionada. Para esto, se selecciona $\bar{v}_0 = v_q$ y $\bar{v}_{K-1} = v_m$ (es decir el valor máximo y mínimo de V), de tal manera que dos copias de $C(u)$ se colocarán de acuerdo en el primer y último punto de control de $T(v)$. Los valores intermedios se pueden obtener convenientemente promediando los nodos relativos, es decir

$$\bar{v}_k = (v_{k+1} + \dots + v_{k+q})/q \quad (5.12)$$

para $k = 1, \dots, K - 2$.

Los sistemas coordenados locales, representan un mapeo de rotación que toma el espacio de tres dimensiones que denotaremos por $\{0, X, Y, Z\}$ a un espacio local $\{o(v_k), x(v_k), y(v_k), z(v_k)\}$ en donde por conveniencia tomaremos

$$o(\bar{v}_k) = T(\bar{v}_k) \quad y(\bar{v}_k) = \frac{T'(\bar{v}_k)}{|T'(\bar{v}_k)|} \quad (5.13)$$

$$z(\bar{v}_k) = \frac{B(\bar{v}_k)}{|B(\bar{v}_k)|} \quad (5.14)$$

$$x(\bar{v}_k) = z(\bar{v}_k) \times y(\bar{v}_k) \quad (5.15)$$

⁴Sólo insertaremos los nodos en el vector sin calcular los nuevos puntos de control de $T(v)$.

Aquí $B(v)$ es una función de valores vectoriales que cumple que $y(v) \cdot B(v) = 0$ para toda v , es decir son ortogonales para toda v .

La definición de $B(v)$ no es trivial y determinarla es uno de los puntos más importantes en la construcción de las superficies de barrido. Una solución natural, sería usar las ecuaciones *Frenet-Serret* descritas en Geometría Diferencial [Klok] {DoCa}

$$B(v) = \frac{T'(v) \times T''(v)}{|T'(v) \times T''(v)|} \quad (5.16)$$

y

$$N(v) = B(v) \times T'(v) \quad (5.17)$$

Sin embargo para esto es necesario suponer que $T(v)$ es doblemente diferenciable, lo cual en términos prácticos tiene las siguientes desventajas

- Para segmentos de línea recta o en puntos de inflexión, $B(v)$ no está definida, es decir cuando $T'(v)$ y $T''(v)$ no existan o sean paralelos, ya que en este caso $T'(v) \times T''(v) = 0$.
- $B(v)$ puede cambian abruptamente en dirección opuesta en puntos de inflexión.
- Para trayectorias en el espacio de tres dimensiones, los vectores proporcionados por $B(v)$ pueden rotar causando excesivos giros.

Siltanen y Woodward [Silt], proponen un método llamado “projection normal” que elimina estos problemas. Su técnica consiste en construir la función $B(v)$ por medio de una interpolación o aproximación de ciertos vectores B_i definidos para cada \bar{v} , los cuales se calculan tomando un vector arbitrario B_0 unitario y ortogonal a $T'(v_0)$ y entonces para $k = 1, \dots, K - 1$

$$\begin{aligned} T_i &= \frac{T'(v_i)}{|T'(v_i)|} \\ b_i &= B_{i-1} - (B_{i-1} \cdot T_i)T_i \\ B_i &= \frac{b_i}{|b_i|} \end{aligned} \quad (5.18)$$

Es decir, B_i se obtiene de normalizar la diferencia del vector que resulta de la proyección ortogonal de B_{i-1} sobre T_i y B_{i-1} (ver figura 5.29), lo cual garantiza que $B_i \cdot T_i = 0$, ya que

$$\begin{aligned} B_i \cdot T_i &= \left[\frac{(B_{i-1} - B_{i-1} \cdot T_i)T_i}{|b_i|} \right] \cdot T_i \\ &= \frac{B_{i-1} \cdot T_i - (B_{i-1} \cdot T_i)|T_i|^2}{|b_i|} \\ &= \frac{B_{i-1} \cdot T_i - (B_{i-1} \cdot T_i)}{|b_i|} = 0 \end{aligned}$$

Una vez determinados el origen y los ejes del sistema local para cada \bar{v}_k

$$\begin{aligned} o(\bar{v}_k) &= (o_x, o_y, o_z) \\ x(\bar{v}_k) &= (x_x, x_y, x_z) \\ y(\bar{v}_k) &= (y_x, y_y, y_z) \\ z(\bar{v}_k) &= (z_x, z_y, z_z) \end{aligned}$$

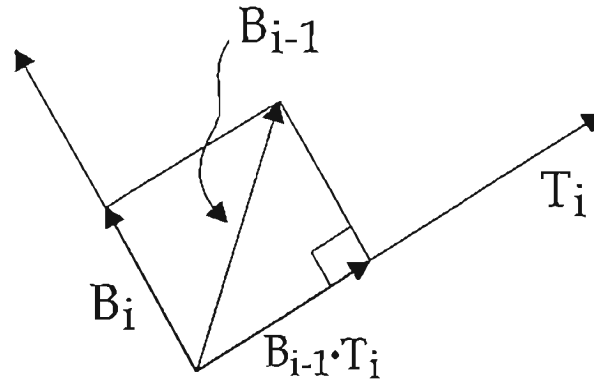


Figura 5.29: Proyección ortogonal para el cálculo del vector B_i .

es necesario rotar y trasladar cada punto de control de (5.10) de acuerdo a éste. Así, es importante notar que esta rotación representa un cambio de sistema coordenado, en donde por definición, cada terna $\{x(\bar{v}_k), y(\bar{v}_k), z(\bar{v}_k)\}$ es una base del nuevo sistema local (ya que son linealmente independientes).

Entonces del álgebra lineal clásica, sabemos que este tipo de transformaciones se realiza usando la matriz de cambio de coordenadas [Friedberg], que se obtiene de la representación esta base en términos de alguna para $\{0, X, Y, Z\}$, así, si por ejemplo tomamos la base canónica tenemos

$$\begin{aligned} x(\bar{v}_k) &= e_1 x_x + e_2 x_y + e_3 x_z \\ y(\bar{v}_k) &= e_1 y_x + e_2 y_y + e_3 y_z \\ z(\bar{v}_k) &= e_1 z_x + e_2 z_y + e_3 z_z \end{aligned}$$

con $e_1 = (1, 0, 0)$, $e_2 = (0, 1, 0)$, $e_3 = (0, 0, 1)$. De donde la matriz de transformación que denotaremos por R , resulta

$$R = \begin{pmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{pmatrix}$$

Por lo que cada Q_i^k se transforma haciendo la multiplicación

$$\begin{aligned} R \cdot (Q_i^k)^t &= \begin{pmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{pmatrix} \begin{pmatrix} x_i^k \\ y_i^k \\ z_i^k \end{pmatrix} \\ &= \begin{pmatrix} x_x x_i^k + y_x y_i^k + z_x z_i^k \\ x_y x_i^k + y_y y_i^k + z_y z_i^k \\ x_z x_i^k + y_z y_i^k + z_z z_i^k \end{pmatrix} \end{aligned}$$

una vez hecho esto la translación se obtiene

$$R \cdot (Q_i^k)^t + o(\bar{v}_k) = \begin{pmatrix} x_x x_i^k + y_x y_i^k + z_x z_i^k + o_x \\ x_y x_i^k + y_y y_i^k + z_y z_i^k + o_y \\ x_z x_i^k + y_z y_i^k + z_z z_i^k + o_z \end{pmatrix}$$

y así el punto de resultante es

$$\begin{aligned}
 & (x_x x_i^k + y_x y_i^k + z_x z_i^k + o_x, \\
 & x_y x_i^k + y_y y_i^k + z_y z_i^k + o_y, \\
 & x_z x_i^k + y_z y_i^k + z_z z_i^k + o_z)
 \end{aligned} \tag{5.19}$$

Una manera conveniente de realizar estas dos transformaciones es considerar la versión homogénea del punto de control Q_i^k es decir $Q_{i,k}^w$ y realizar la composición de transformaciones homogéneas siguiente [Marsh]

$$A = \begin{pmatrix} 1 & 0 & 0 & o_x \\ 0 & 1 & 0 & o_y \\ 0 & 0 & 1 & o_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_x & y_x & z_x & 0 \\ x_y & y_y & z_y & 0 \\ x_z & y_z & z_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} x_x & y_x & z_x & o_x \\ x_y & y_y & z_y & o_y \\ x_z & y_z & z_z & o_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.20}$$

De tal manera que

$$\begin{aligned}
 A \cdot (Q_{i,k}^w)^t &= \begin{pmatrix} x_x & y_x & z_x & o_x \\ x_y & y_y & z_y & o_y \\ x_z & y_z & z_z & o_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w_i^C x_i^k \\ w_i^C y_i^k \\ w_i^C z_i^k \\ w_i^C \end{pmatrix} \\
 &= \begin{pmatrix} x_x w_i^C x_i^k + y_x w_i^C y_i^k + z_x w_i^C z_i^k + o_x w_i^C \\ x_y w_i^C x_i^k + y_y w_i^C y_i^k + z_y w_i^C z_i^k + o_y w_i^C \\ x_z w_i^C x_i^k + y_z w_i^C y_i^k + z_z w_i^C z_i^k + o_z w_i^C \\ w_i^C \end{pmatrix}
 \end{aligned}$$

ahora si definimos

$$\begin{aligned}
 A \cdot Q_{i,k} &= (x_x w_i^C x_i^k + y_x w_i^C y_i^k + z_x w_i^C z_i^k + o_x w_i^C, \\
 & x_y w_i^C x_i^k + y_y w_i^C y_i^k + z_y w_i^C z_i^k + o_y w_i^C, \\
 & x_z w_i^C x_i^k + y_z w_i^C y_i^k + z_z w_i^C z_i^k + o_z w_i^C, \\
 & w_i^C)
 \end{aligned}$$

y aplicamos el mapeo de perspectiva (2.21) a este punto, es decir

$$H(A \cdot Q_{i,k})$$

obtenemos (5.19).

Siguiendo las consideraciones anteriores, el proceso de cálculo de los puntos de control $P_{i,j}^w$ de la superficie de barrido así como su vector de nodos V del parámetro v (el vector de nodos U será el mismo de la curva de barrido), se puede determinar con el siguiente algoritmo.

SweepSurface($T(v), C(u), B(v), S(v), K, V, P^w$) {

Entrada: $T(v), C(u), B(v), S(v), K$

Salida: P^w Puntos de control de la superficie de barrido

V Vector de nodos del parámetro v de la superficie de barrido.

q = grado de la curva $T(v)$

ktv = número de nodos de la curva $T(v)$ //Notemos que entonces $ktv = s + 1$

V = Vector de nodos de $T(v)$

5.3. SUPERFICIES DE BARRIDO.

```

nsect= K

if(ktv <= nsect+q){ //Es necesario refinar el vector de nodos
m =nsect+q - ktv + 1
  Debemos insertar m nodos más a V.
  Una posibilidad es insertarlos en los puntos medios
  de los intervalos con longitud más grande.
}

else { //No es necesario modificar el vector de nodos V.

  V = vector de nodos de T(v)

  if (ktv > nsect+q + 1) //Demos incrementar el número de copias de C(u)
    nsect = ktv - q - 1
}

v̄₀ = v_q
v̄_{nsect-1} = v_m

//Calcular los valores parametrales promediando los nodos.
for(k=1;k<nsect-1;k++)
  v̄_k = (v_{k+1} + ... + v_{k+q})/q

//Cálculo de la función B(v)
B₀ = vector arbitrario ortonormal a T'(v₀)
for(i=1;i<nsect;i++){
  T_i = T'(v_i)
  B_i = B_{i-1} - (B_{i-1} · T_i)T_i
  B_i = B_i / |B_i|
}
B(v) = función que resulta de interpolar o aproximar los puntos {B₀, ..., B_{nsect-1}}

//for principal
for(k=0; k<nsect; k++){

//Escalar los puntos de control Q_{i,k}^w para i = 0, ..., n y k = 0, ..., nsect - 1
  for(i=0; i<=n; i++){
    Q_{i,k}^w = S_{v_k} · Q_{i,k}^w //ecuación (5.10)
  }

//Cálculo de los sistemas ortonormales locales.
o(v̄_k) = T(v̄_k) = (o_x, o_y, o_z)
y(v̄_k) = T'(v̄_k) / |T'(v̄_k)| = (y_x, y_y, y_z)
z(v̄_k) = B(v̄_k) / |B(v̄_k)| = (z_x, z_y, z_z)
x(v̄_k) = z(v̄_k) × y(v̄_k) = (x_x, x_y, x_z)

```

```
//Cálculo de la matriz de transformación
R(v_k) = \begin{pmatrix} x_x & y_x & z_x & 0 \\ x_y & y_y & z_y & 0 \\ x_z & y_z & z_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T(v_k) = \begin{pmatrix} 1 & 0 & 0 & o_x \\ 0 & 1 & 0 & o_y \\ 0 & 0 & 1 & o_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A(v_k) = T(v_k) \cdot R(v_k)

//Transformación de puntos de control de cada curva.
for(i=0; i<=n; i++){
    Q_{i,k}^w = A(v_k) \cdot (Q_{i,k}^w)^t
}
} //Termina for principal

for(i=0; i<n; i++){
    {Q'_{0,...}, Q'_{nsect-1}} = Puntos de control de
    la curva de interpolación global a través los puntos Q_{i,0}^w, \dots, Q_{i,nsect-1}^w
    con vector paramétrico \bar{V} y grado q.
    for(k=0; k<nsect; k++){
        P_{i,k} = Q'_{k}
    }
}
}
```

Ejemplo 40 Construyamos una superficie de barrido con las curvas NURBS cúbicas que se presentan en la Figura 5.30 colocando 9 copias de $C(u)$ a lo largo de $T(v)$ (ver figura 5.31), en donde sus vectores de nodos están dados por

$$U = \{0, 0, 0, 0, \frac{1}{2}, 1, 1, 1, 1\}$$

$$V = \{0, 0, 0, 0, \frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}, 1, 1, 1, 1\}$$

los pesos $w_i^C = 1$ para $i = 0, 2, 4$ y $w_i^C = 2$ para $i = 1, 3$ y $w_j^T = 1$ para $j = 0, \dots, 8$, y los puntos de control son

Puntos de control					
Q_i	$(-3, 0, 0)$	$(-2, 0, 4)$	$(0, 0, 0)$	$(2, 0, 4)$	$(3, 0, 0)$
T_i	$(0, -10, 0)$	$(0, -10, 4)$	$(0, 4, 2)$	$(0, 3, 15)$	$(0, 18, 20)$
	$(0, 19, 6)$	$(0, 33, 10)$	$(0, 35, 19)$	$(0, 40, 19)$	

Primeramente notemos que el número de puntos de control de $T(v)$ coincide con la cantidad de curvas que deseamos insertar (K es igual a 9), por lo que la primer condición de nuestro algoritmo es falsa, es decir ktv (igual a 13) no es menor o igual que $nsect + q$ (igual a 12) y así el vector de nodos de la superficie de barrido en el parámetro v , será exactamente igual al de la curva de trayectoria. También observese que no es necesario incrementar el número de copias de la curva, ya que ktv es igual a $nsect + q + 1$.

Ahora tomando $\bar{v} = v_q = 0$ y $\bar{v}_{nsect-1} = v_8 = v_m = 1$, podemos calcular los valores parametrales en donde serán posicionadas las copias de la curva de barrido usando la formula (5.12), es decir

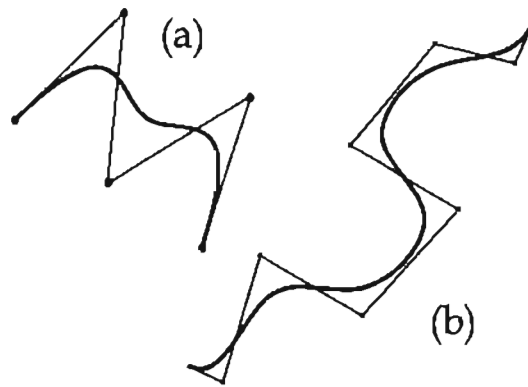


Figura 5.30: (a) Curva de barrido. (b) Curva Trayectoria.

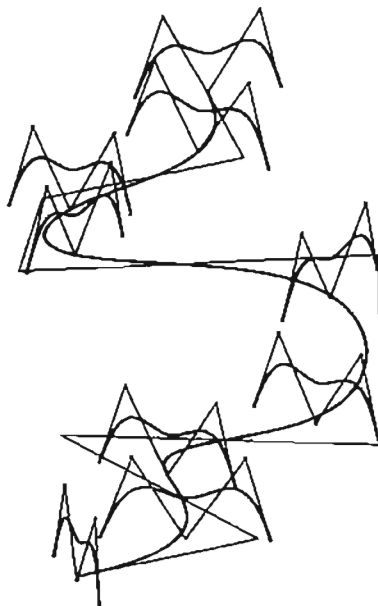


Figura 5.31: Curvas posicionadas a través de la curva de trayectoria.

$$\begin{aligned} \bar{v}_1 &= \frac{v_2 + v_3 + v_4}{3} \\ &\vdots \\ \bar{v}_7 &= \frac{v_8 + v_9 + v_{10}}{3} \end{aligned}$$

entonces

$$\vec{V} = \{0, 0.056, 0.167, 0.333, 0.5, 0.667, 0.833, 0.944, 1.0\}$$

Por otro lado como $T'(\bar{v}_0) = (0, 0, 72)$ podemos tomar $B_0 = (1, 0, 0)$. Y ya que en este caso la curva trayectoria pertenece al plano yz por (5.18) todos los vectores $B_i = (1, 0, 0)$, por lo que la función de aproximación o interpolación deberá ser la función constante $B(v) = (1, 0, 0)$ para toda v .

Si inicialmente suponemos que la función de escala es $s_v = (1, 1, 1)$ para toda v (es decir no realizamos ningún escalamiento), el siguiente paso del algoritmo es calcular los sistemas coordenados locales usando (5.13), (5.14) y (5.15) y su respectiva matriz de cambio de coordenada (5.20), por ejemplo el caso para \bar{v}_0 , la implementación del algoritmo resulta

$$\begin{aligned} o(\bar{v}_0) &= (0, -10, 0) & x(\bar{v}_0) &= (0, -1, 0) \\ y(\bar{v}_0) &= (0, 0, 1) & z(\bar{v}_0) &= (1, 0, 0) \end{aligned}$$

y por lo tanto la matriz de transformación es

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

entonces

$$\begin{aligned} \begin{pmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -3 \\ 0 \\ 0 \\ 1 \end{pmatrix} &= \begin{pmatrix} 0 \\ -7 \\ 0 \\ 1 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -4 \\ 0 \\ 8 \\ 2 \end{pmatrix} &= \begin{pmatrix} 8 \\ -16 \\ 0 \\ 2 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} &= \begin{pmatrix} 0 \\ -10 \\ 0 \\ 1 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 0 \\ 8 \\ 2 \end{pmatrix} &= \begin{pmatrix} 8 \\ -24 \\ 0 \\ 2 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \\ 0 \\ 1 \end{pmatrix} &= \begin{pmatrix} 0 \\ -13 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$

5.3. SUPERFICIES DE BARRIDO.

es decir los puntos de control de $C_0(u)^w$ son

$$\begin{aligned} Q_{0,0}^w &= (0, -7, 0, 1) & Q_{1,0}^w &= (8, -16, 0, 2) & Q_{2,0}^w &= (0, -10, 0, 1) \\ Q_{3,0}^w &= (8, -24, 0, 2) & Q_{4,0}^w &= (0, -13, 0, 1) \end{aligned}$$

Este proceso continua para $k = 1, \dots, K - 1$ y finalmente se realiza la interpolación de $Q_{i,0}^w, \dots, Q_{i,k-1}^w$, con lo que se obtienen los puntos de control de $P_{i,0}^w, \dots, P_{i,K-1}^w$ para $i = 0, \dots, n$ de la superficie de barrido, la cual se muestra en la Figura 5.32.

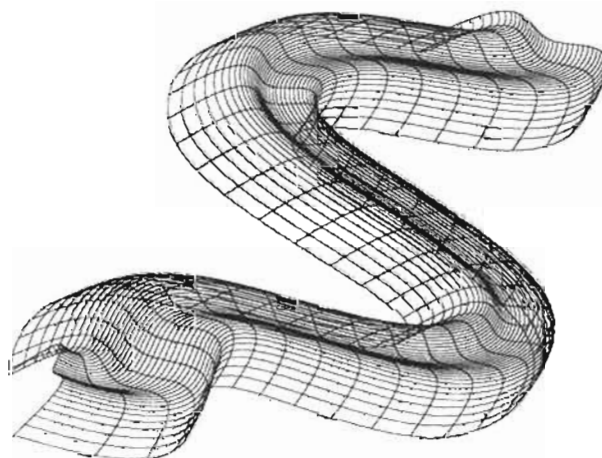


Figura 5.32: Superficie de barrido.

Si ahora tomamos una función de escala $s'_v(u)$ construida con una curva NURBS cuadrática, que interpole los puntos $\{(1, 1, 1), (0.1, 0.1, 0.1), (1, 1, 1)\}$ de tal manera que

$$\begin{aligned} s'_{\bar{v}_0} &= (1, 1, 1) \\ s'_{\bar{v}_4} &= (0.1, 0.1, 0.1) \\ s'_{\bar{v}_8} &= (1, 1, 1) \end{aligned}$$

lograremos que las curvas de barrido en los parámetros $\bar{v}_0, \dots, \bar{v}_4$ se escalen con factores de que van del 1 al 0.1 y en los parámetros de $\bar{v}_4, \dots, \bar{v}_8$ con factores de 0.1 a 1. Por ejemplo para $\bar{v}_4 = 0.5$, los puntos de control escalonados son

$$\begin{aligned} s_{\bar{v}_4} Q_{0,k}^w &= (-0.3, 0, 0, 1) \\ s_{\bar{v}_4} Q_{1,k}^w &= (-0.4, 0, 0.8, 2) \\ s_{\bar{v}_4} Q_{2,k}^w &= (0, 0, 0, 1) \\ s_{\bar{v}_4} Q_{3,k}^w &= (0.4, 0, 0.8, 2) \\ s_{\bar{v}_4} Q_{4,k}^w &= (0.3, 0, 0, 1) \end{aligned}$$

el sistema coordenado local tiene como origen y ejes los puntos

$$\begin{aligned} o(\bar{v}_4) &= (015.666716.8333) & x(\bar{v}_4) &= (-00.4902610.871576) \\ y(\bar{v}_4) &= (00.871576 - 0.490261) & z(\bar{v}_4) &= (1, 0, 0) \end{aligned}$$

y así la matriz de transformación queda

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0.490261 & 0.871576 & 0 & 15.6667 \\ 0.871576 & -0.490261 & 0 & 16.8333 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

por lo que las rotaciones al nuevo sistema se calculan haciendo

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0.490261 & 0.871576 & 0 & 15.6667 \\ 0.871576 & -0.490261 & 0 & 16.8333 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -0.3 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 15.5196 \\ 16.5719 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0.490261 & 0.871576 & 0 & 15.6667 \\ 0.871576 & -0.490261 & 0 & 16.8333 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -0.4 \\ 0 \\ 0.8 \\ 2 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 31.1372 \\ 33.318 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0.490261 & 0.871576 & 0 & 15.6667 \\ 0.871576 & -0.490261 & 0 & 16.8333 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 15.6667 \\ 16.8333 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0.490261 & 0.871576 & 0 & 15.6667 \\ 0.871576 & -0.490261 & 0 & 16.8333 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.4 \\ 0 \\ 0.8 \\ 2 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 31.5294 \\ 34.0153 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0.490261 & 0.871576 & 0 & 15.6667 \\ 0.871576 & -0.490261 & 0 & 16.8333 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.3 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 15.8137 \\ 17.0948 \\ 1 \end{pmatrix}$$

la superficie resultante se muestra en la Figura 5.33.

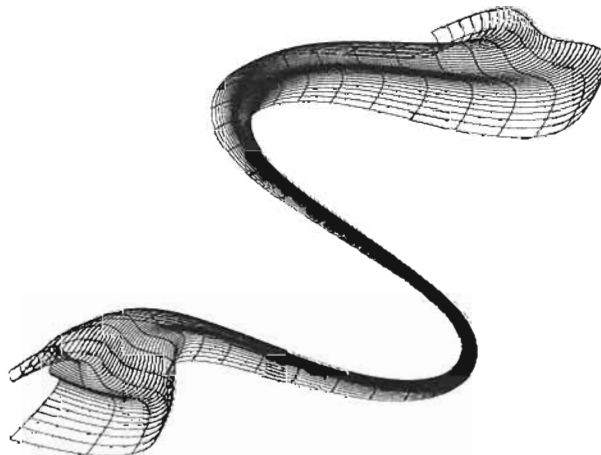


Figura 5.33: Superficie de barrido con función de escala s'_v .

Finalmente si construimos una función de escala s'' usando una curva NURBS, de tal manera que interpole los puntos $\{(1, 1, 1), (2, 2, 2), (1, 1, 1)\}$, en los parámetros \bar{v}_0, \bar{v}_4 y \bar{v}_8 respectivamente, es decir

$$s''_{\bar{v}_0} = (1, 1, 1)$$

$$s''_{\bar{v}_4} = (2, 2, 2)$$

$$s''_{\bar{v}_8} = (1, 1, 1)$$

La superficie resultante se muestra en la Figura 5.34.

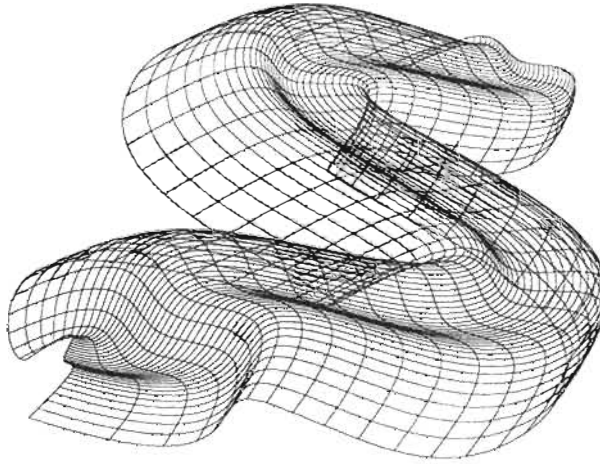


Figura 5.34: Superficie de barrido con función de escala s''_v .

□

Apéndice A

La biblioteca NURBS++.

La biblioteca **NURBS++**, es un conjunto de rutinas con un diseño orientado a objetos implementadas con el lenguaje de programación C++ por el Dr.Philippe Lavoie, la cual proporciona clases con las se pueden representar curvas y superficies NURBS y tiene implementados varios algoritmos descritos en libro “The NURBS Book” ([PT97]). Además proporciona módulos para la manipulación de matrices, métodos numéricos (como aproximación mínima cuadrada, SVD y funciones estadísticas), y permite el manejo de distintos formatos de imágenes con los que se puede realizar impresiones de curvas y superficies usando **OpenGL** , **VRML** y **Post-Script**.

NURBS++ se puede obtener libremente bajo los términos de la licencia **GNU** en la página de “Internet”(ver figura A.1) <http://libnurbs.sourceforge.net/index.shtml>.

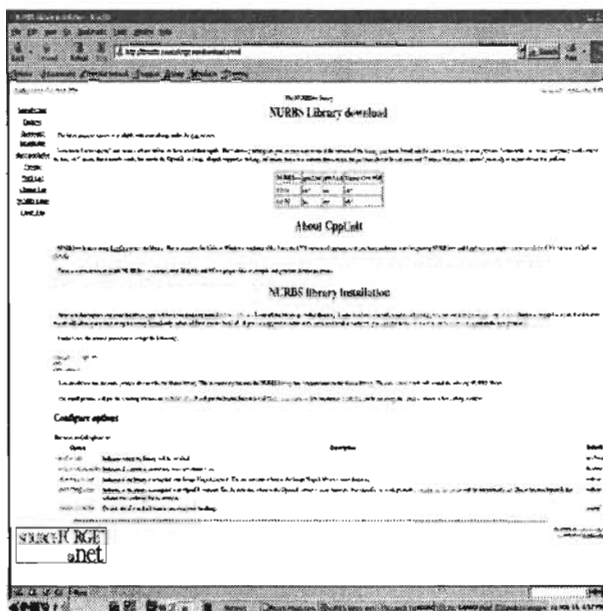


Figura A.1: Página electrónica de la biblioteca NURBS++

En ésta, además se puede consultar las instrucciones para instalación de la biblioteca tanto en los sistemas operativos basados en “Unix”(como linux) o “Windows”de Microsoft y así como información relacionada a este proyecto.

A.1. Instalación.

Existen varias manera de realizar la instalación dependiendo del “software” preinstalado y el sistema operativo con el que se trabaje. Sin embargo, independientemente de éste, la biblioteca **NURBS++** requiere instaladas las bibliotecas gráficas de **OpenGL** y **GLUT**¹ y un compilador del lenguaje C++²

A continuación presentaremos las instrucciones básicas para la instalación de la versión 3.11, suponiendo que se está trabajando bajo sistema operativo linux.

Primeramente se debe bajar el paquete, en la página de **NURBS++** presionando en la liga “files”, la cual nos mandará a la base de datos de “software” libre “sourceforge.net”³ para esta biblioteca, en esta página es posible seleccionar el paquete **NURBS++** comprimido en varios formatos, para esta explicación nosotros seleccionaremos el formato `tar.gz`, es decir `nurbs++-3.0.11.tar.gz`. Una vez hecho, se abrirá otra página en donde se puede seleccionar el dominio desde donde se copiará el archivo, en este caso es recomendable seleccionar el más cercano a nuestro lugar de trabajo, por ejemplo nosotros seleccionaremos cualquiera de norte América.

Cuando el paquete esté copiado, debemos proceder a descomprimirlo e instalarlo, en este caso supondremos que esto se realizará en la carpeta `/home/usuario/`, en donde `usuario`, es el “login” de nuestra respectiva cuenta en linux. Entonces tecleemos los siguientes comandos

```
[usuario]$ tar -zxf nurbs++-3.0.11.tar.gz
[usuario]$ cd nurbs++-3.0.11
[usuario]$ ./configure
[usuario]$ make
[usuario]$ su
[root]# make install
```

La primera instrucción descomprimirá la biblioteca, con la segunda entraremos a la carpeta en donde se encuentra **NURBS++** y la siguiente ejecutará al programa `configure` que sirve para configurar el “software” en la plataforma que se va instalar, a este programa es posible pasarle varios parámetros para realizar una instalación personalizada, por ejemplo si escribimos

```
[usuario]$ ./configure --prefix=/home/usuario/nurbs++-3.0.11/lib
```

las bibliotecas ya compiladas se instalarán en la ruta `/home/usuario/nurbs++-3.0.11/lib` en vez de `/usr/local/lib` (por “default”), esto es útil cuando por ejemplo no se conoce el “password” de “root” o se desea realizar una instalación de las biblioteca sólo en nuestra cuenta de Linux. Otros posibles parámetros al programa `configure` se describen en la página.

La instrucción `make`, compilará todas las bibliotecas de **NURBS++**, su ejecución tardará unos cuantos minutos dependiendo del procesador de nuestra máquina.

La instrucción `su`, permite en un sistema Linux cambiar a modo superusuario o administrador, con ésta podremos ejecutar la siguiente `make install`, con la que se copiarán las bibliotecas recién compiladas a la carpeta `/usr/local/lib`, estás dos instrucciones sólo se podrán realizar si se conoce el “password” de “root”. En caso de que no sea así, es recomendable como se mencionó anteriormente, realizar la instalación local pasándole al programa `configure` el parámetro `--prefix`, en cuyo caso estas dos últimas instrucciones no se deberán ejecutarse.

Una manera fácil de comprobar si nuestra instalación funciona, es tratar de correr los ejemplos incluidos en la biblioteca. Por ejemplo probemos los de curvas y superficies **NURBS** escribiendo

```
[usuario]$ cd /home/usuario/nurbs++-3.0.11/examples/nurbs
[usuario]$ make
```

¹Es posible encontrar más información en <http://www.opengl.org/>

²En la página se indica cuales son la versiones adecuadas.

³Para más acerca de este proyecto ver en <http://sourceforge.net/>

Así una vez que todos estén compilados ejecutemos por ejemplo

```
[usuario]$ ./tnsSweep
```

este programa creará varias superficies de barrido en el formato **VRML**, para poder verlas es necesario tener un visualizador de esta clase de archivos, uno fácil de instalar es **Rational Reducer**, el cual tiene una versión de evaluación gratuita en la dirección <http://www.sim.no/products/evaluate/>

En algunas versiones de linux, es posible que al momento de hacer la compilación de los ejemplos no se encuentre ciertas bibliotecas de **OpenGL** y **GLUT** o sus versiones no sean compatibles con las requeridas por **NURBS++**. Una manera fácil de resolver esto, es instalar la última versión de **OpenGL** o su versión libre **Mesa**. A continuación explicaremos la instalación de la versión 6.2 de **Mesa** suponiendo que se realizará en la carpeta `/home/usuario`.

En la página de Mesa <http://www.mesa3d.org/>, bajar los archivos `MesaLib-6.2.1.tar.gz` y `MesaDemos-6.2.1.tar.gz` y descomprimirlos con las instrucciones

```
[usuario]$ tar -zxf MesaLib-6.2.1.tar.gz
[usuario]$ tar -zxf MesaDemos-6.2.1.tar.gz
```

una vez hecho esto se creará la carpeta `Mesa-6.2.1` y entonces tecleemos

```
[usuario]$ cd Mesa-6.2.1
[usuario]$ chmod a+x bin/mklib
[usuario]$ make linux
```

para comprobar que la instalación tuvo éxito, probemos uno de los ejemplos que vienen con la distribución, así escribamos

```
[usuario]$ cd lib
[usuario]$ export LD_LIBRARY_PATH=${PWD}
[usuario]$ cd ../progs/demos
[usuario]$ ./gears
```

si **Mesa** se instaló correctamente, se presentará una animación de varios engranes moviéndose. En caso de tener algún problema es recomendable revisar con cuidado la página de "Internet", en donde se explica con detalle la instalación anterior.

Una vez que hayamos instalado **Mesa** lo único que debemos hacer para que los ejemplos de **NURBS++** funcionen, es indicar al compilador la ruta de las nuevas bibliotecas de **GLUT** y **OpenGL**, las cuales están en la carpeta `/home/usuario/Mesa-6.2.1/lib`, una manera de hacerlo es modificando el `Makefile` de `/home/usuario/nurbs++-3.0.11/examples/nurbs` reemplazando las líneas

```
OPENGL_LIBS = -lglut $(GL_LFLAGS) $(GL_LIBS) $(X_EXTRA_LIBS)
ALLLIBS = -lglut $(packagedir)/nurbs/libnurbsf.la $(MATRIXLIB)
```

por

```
OPENGL_LIBS = -L/home/usuario/Mesa-6.3.1/glut $(GL_LFLAGS) $(GL_LIBS) $(X_EXTRA_LIBS)
ALLLIBS = -L/home/usuario/Mesa-6.3.1/glut $(packagedir)/nurbs/libnurbsf.la $(MATRIXLIB)
```

A.2. Uso de la biblioteca NURBS++.

En esta sección describiremos brevemente el uso de la biblioteca NURBS++ para la representación y manipulación de curvas y superficies NURBS.

A.2.1. Curvas

La clase principal para la definición de un objeto NURBS en el caso de la curva es la clase “template” NurbsCurve la cual se deriva de la clase ParaCurve y está definida en el archivo nurbs.h. Los prototipos de sus constructores son

```
NurbsCurve() ;
NurbsCurve(const NurbsCurve<T,N>& nurb) ;
NurbsCurve(const Vector< HPoint_nD<T,N> >& P1, const Vector<T> &U1, int deg=3);
NurbsCurve(const Vector< Point_nD<T,N> >& P1, const Vector<T> &W, const Vector<T> &U1,
            int deg=3);
```

Además se definen los tipos de datos

```
typedef NurbsCurve<float,3> NurbsCurvef;
typedef NurbsCurve<double,3> NurbsCurved;
typedef NurbsCurve<float,2> NurbsCurve_2Df;
typedef NurbsCurve<double,2> NurbsCurve_2Dd;
```

los cuales nos permite por ejemplo, construir una curva NURBS usando el constructor NurbsCurvef, en donde <float,3> nos indica que la curva manipulará números representados en punto flotante y puntos de tres dimensiones, análogamente con los otros tipos de datos.

El siguiente programa muestra cómo se puede definir una curva NURBS cúbica con 5 puntos de control, vector de nodos $U = \{0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1\}$ y pesos $W = \{1, 1, 1, 2, 1, 3\}$. La curva resultante se muestra en A.2.

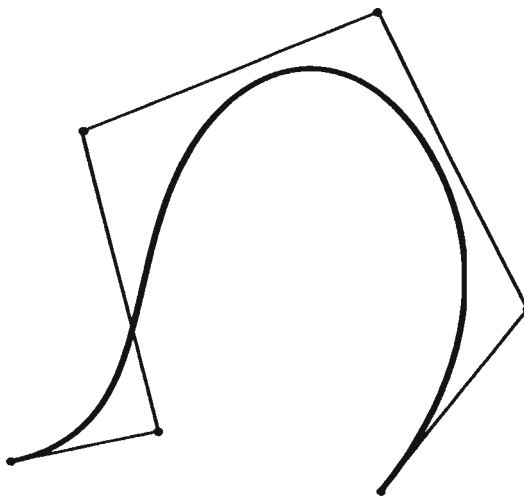


Figura A.2: Curva NURBS cúbica.

```

//definida la clase NurbsCurve
#include <nurbs.h>
int main(){

    //Permiten escribir NurbsCurvef
    //en vez de PLib::NurbsCurvef
    using namespace PLib ;

    //-----
    //Grado y número de puntos de la curva
    //-----

    int p = 3 ; // Grado de la curva.
    int n = 5; // n+1 es el número de puntos.
    int i,j;    // Variables auxiliares.

    //-----
    // Definición de los puntos de control y pesos
    //-----

    //Vector de puntos de tres dimensiones
    // en donde se almacenan los puntos
    // de control de la curva.

    Vector_Point3Df P(n+1);
    P[0] = Point3Df(2,3,0);
    P[1] = Point3Df(6,4,0);
    P[2] = Point3Df(4,14,0);
    P[3] = Point3Df(12,18,0);
    P[4] = Point3Df(16,8,0);
    P[5] = Point3Df(12,2,0);

    //Vector de números de punto flotante
    //en el que almacenaremos los pesos de
    // la curva.

    Vector_FLOAT W(n+1);
    W[0]=1.0;
    W[1]=1.0;
    W[2]=1.0;
    W[3]=2.0;
    W[4]=2.0;
    W[5]=1.0;

    //También es posible usar la versión de
    //los puntos en coordenadas homogéneas
    //haciendo

    Vector_HPoint3Df PH(n+1);
    PH[0] = HPoint3Df(2,3,0,1);
    PH[1] = HPoint3Df(6,4,0,1);
    PH[2] = HPoint3Df(4,14,0,1);
    PH[3] = HPoint3Df(12,18,0,2);
    PH[4] = HPoint3Df(16,8,0,2);
    PH[5] = HPoint3Df(12,2,0,1);

    //-----
    //Definición del vector de nodos
    //-----

    // subíndice del último vector de nodos.

```


A.2. USO DE LA BIBLIOTECA NURBS++.

```
int m = n+p+1;

// Vector de nodos
Vector_FLOAT U(m+1) ;
for(i=0;i<=p;++i)
    U[i] = 0;
for(i=p+1,j=1;i<=n ;i++,j++)
    U[i] = (float)j/float(n-p+1);
for(i=n+1;i<=m;++i)
    U[i] = 1;

//-----
// Declaración de la curva nurbs
//-----

NurbsCurvef curva_nurbs1(PH,U,p);
NurbsCurvef curva_nurbs2(P,W,U,p);

//-----
//Impresión en formato ps usando el método writePS
//-----
// Parámetros:

//El primer argumento es el nombre del
//archivo postscript.

//cp: sirve, para indicar si
//se desea imprimir los puntos de control
//en donde si cp=0 no y cp=1 si.
int cp = 1;

//magFact: sirve, para indicar el factor
//de tamaño de la imagen postscript.
float magFact = 40.5;

//dash: sirve para determinar los espacios
//en blanco de las líneas que unen los puntos
//de control de la curva.
//Si este valor es menor o igual que 0
//la línea no tendrá espacios.
float dash =0.0;

curva_nurbs1.writePS("cnurbs1.ps",cp,magFact,dash) ;
curva_nurbs2.writePS("cnurbs2.ps",cp,magFact,dash) ;
//-----
//Impresión en formato VRML usando el método writeVRML
//-----

// Parámetros:

//El primer argumento es el nombre del
//archivo en formato VRML.

//radio: Radio de la línea.
float radio = 2.0;

//K: Número mínimo de interpolación.
int K=18;

//color: Color de la línea.
Color color = blueColor;
```

```
// Nu: Número de puntos en el círculo.
int Nu = 20;

//Nv: Número de puntos a través del camino.
int Nv = 30;

curva_nurbs1.writeVRML("cnurbs1.wrl",radio,K,color,Nu,Nv) ;
curva_nurbs2.writeVRML("cnurbs2.wrl",radio,K,color,Nu,Nv) ;

return 0 ;
}
```

Del capítulo 2, sabemos que una curva NURBS

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u)w_i P_i}{\sum_{i=0}^n N_{i,p}(u)w_i}$$

con vector de nodos

$$U = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_n, \underbrace{b, \dots, b}_{p+1}\}$$

es una curva B-spline si $w_i = c$ con c constante para toda i , y una curva de Bézier si $n = p$ y

$$U = \{\underbrace{a, \dots, a}_{p+1}, \underbrace{b, \dots, b}_{p+1}\}$$

en particular si $a = 0$ y $b = 1$. En el siguiente programa creamos dos objetos uno que representa una curva B-spline y otro una curva de Bézier.

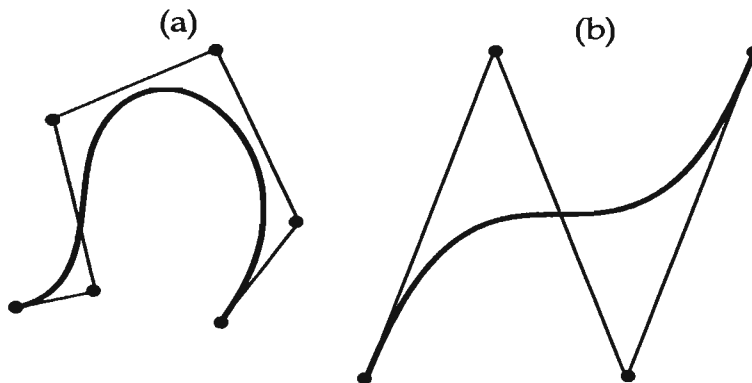


Figura A.3: (a) Curva B-spline cúbica. (b) Curva de Bézier cúbica.

```
#include <nurbs.h>
int main(){

int p = 3; // Grado de las dos curvas.
int n1 = 5; // n1+1 es el número de
// puntos de la curva B-spline.
int n2 = 3; // Número de puntos de la curva de Bézier.
int i,j; // Variables auxiliares.

using namespace PLib;
```

A.2. USO DE LA BIBLIOTECA NURBS++.

```
//Puntos de control
Vector_Point3Df P1(n1+1);
P1[0] = Point3Df(2,3,0);
P1[1] = Point3Df(6,4,0);
P1[2] = Point3Df(4,14,0);
P1[3] = Point3Df(12,18,0);
P1[4] = Point3Df(16,8,0);
P1[5] = Point3Df(12,2,0);

//Pesos
Vector_Point3Df Pd(n1+1);
Vector_FLOAT W1(n1+1);
W1[0]=1.0;
W1[1]=1.0;
W1[2]=1.0;
W1[3]=1.0;
W1[4]=1.0;
W1[5]=1.0;

// subíndice del último vector de nodos.
int m1 = n1+p+1;

// Vector de nodos
Vector_FLOAT U1(m1+1) ;
for(i=0;i<=p;++i)
    U1[i] = 0;
for(i=p+1,j=1;i<=n1 ;i++,j++)
    U1[i] = (float)j/float(n1-p+1);
for(i=n1+1;i<=m1;++i)
    U1[i] = 1;

// Declaración de la curva B-spline
PlNurbsCurvef curva_bspline(P1,W1,U1,p);
curva_bspline.writePS("curva_bspline.ps",1,25.0,0) ;

/*****
*****/

//Puntos de control
Vector_Point3Df P2(n2+1);
P2[0] = Point3Df(0,-20,0) ;
P2[1] = Point3Df(15,25,0) ;
P2[2] = Point3Df(30,-20,0) ;
P2[3] = Point3Df(45,25,0) ;

//Pesos

Vector_FLOAT W2(n2+1);
W2[0]=1.0;
W2[1]=1.0;
W2[2]=1.0;
W2[3]=1.0;

// subíndice del último vector de nodos.
int m2 = n2+p+1;
```

```

// Vector de nodos
Vector_FLOAT U2(m2+1) ;
for(i=0;i<=p;++i)
    U2[i] = 0;
for(i=n2+1;i<=m2;++i)
    U2[i] = 1;

// Declaración de la curva de Bézier
PlNurbsCurvef curva_de_bezier(P2,W2,U2,p);
curva_de_bezier.writePS("curva_de_bezier.ps",1,25.0,0) ;

return 0;
}

```

Como mencionamos al inicio de este apéndice, la biblioteca **NURBS++**, tiene implementados muchos de los algoritmos descritos en el libro “The NURBS Book”, uno útil es el de interpolación global, el cual se describió en la sección 4.4 del capítulo 4. Esta biblioteca crea una curva NURBS de interpolación usando el método `globalInterp`, como mostraremos en el siguiente programa.

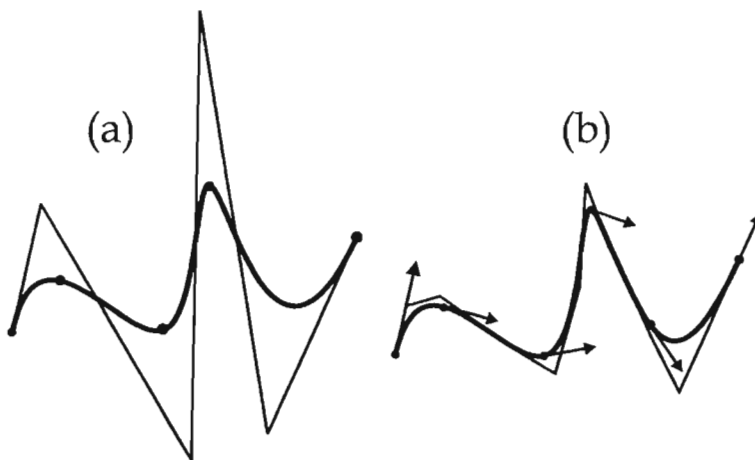


Figura A.4: (a) Interpolación global. (b) Interpolación global de puntos y derivadas.

```

#include <nurbs.h>

using namespace PLib ;

//Definición de funciones auxiliares
double normaEuclidiana(Point3Df P1, Point3Df P2){
    return sqrt((P2.x()-P1.x())*(P2.x()-P1.x())+
                (P2.y()-P1.y())*(P2.y()-P1.y())+
                (P2.z()-P1.z())*(P2.z()-P1.z()));
}

double longitudArco(const Vector_Point3Df & P){
    int n = P.size();
    double larc=0.0;

    for(int i=0;i<n-1;i++)
        larc+= normaEuclidiana(P[i],P[i+1]);
}

```

```

    return larc;
}

int main(){

    int i ;
    int n = 5;

    // Puntos de control
    Vector_Point3Df points(n+1) ;
    points[0] = Point3Df(0,0,0) ;
    points[1] = Point3Df(5,5,0) ;
    points[2] = Point3Df(15,0,0) ;
    points[3] = Point3Df(20,15,0) ;
    points[4] = Point3Df(25,5,0) ;
    points[5] = Point3Df(35,10,0) ;

    // Cálculo del vector Ubar, usando la
    // longitud de arco. Ver sección 4.4 del
    // capítulo 4.

    double larc = longitudArco(points);
    std::cout << larc << std::endl;

    Vector_FLOAT Ubar(n+1);
    Ubar[0] = 0.0;
    Ubar[n] = 1.0;

    for(int i=1;i<n;i++){
        Ubar[i]= Ubar[i-1]+normaEuclidiana(points[i-1],points[i])/larc;

    }

    // Declaración de las curvas NURBS
    NurbsCurvef curva1,curva2;

    Vector_Point3Df deriv(n+1) ;
    Vector_Point3Df empty ;

    //Creación de la curva de interpolación
    //cúbica, usando el vector Ubar.
    curva1.globalInterp(points,Ubar,3);

    //Cálculo de los vectores de derivadas
    //en los puntos de interpolación.
    Vector_Point3Df dtemp(2);
    for(int i=0;i<=n;i++){
        curva1.deriveAt(Ubar[i],1,dtemp);
        deriv[i]= dtemp[1];
    }

    //Creación de la curva de interpolación
    //cúbica, la cual además de interpolar los
    //puntos, también lo hace con las derivadas.
    //Ver sección 4.4 del capítulo 4.

```

```

curva2.globalInterpD(points,deriv,3,0) ;

//Impresión de la curva, en la cual también se
//presentan los puntos de interpolación.
curva1.writePSP("nurbsinterp1.ps",points,empty,1,25.0,0) ;

//Impresión de la curva, en la cual también se
//presentan los puntos de interpolación y sus
//derivadas.
curva2.writePSP("nurbsinterp2.ps",points,deriv,1,25.0,0) ;

return 0;
}

```

A.2.2. Superficies

En el caso de las superficies, la clase principal para su representación es `NurbsSurface`, la cual se deriva de `ParaSurface` y se encuentra en el archivo `nurbsS.h`. Tiene como constructores

```

NurbsSurface() ;
NurbsSurface(const NurbsSurface<T,N>& nS) ;
NurbsSurface(int DegU, int DegV, const Vector<T>& Uk,
              const Vector<T>& Vk, const Matrix< HPoint_nD<T,N> >& Cp) ;
NurbsSurface(int DegU, int DegV, Vector<T>& Uk,
              Vector<T>& Vk, Matrix< Point_nD<T,N> >& Cp, Matrix<T>& W) ;

```

y los tipos definidos para la declaración de sus objetos son

```

typedef NurbsSurface<float,3> NurbsSurfacef ;
typedef NurbsSurface<double,3> NurbsSurfaced ;

```

con éstos, se pueden definir superficies NURBS, manejando tipos de datos de punto flotante y de doble precisión respectivamente.

El siguiente programa define una superficie NURBS de grados $p = 3$ y $q = 2$, puntos de control

Puntos de control					
i/j					
	(20,-10,10)	(10,-10,10)	(0,-10,10)	(-5,-10,10)	(-10,-10,10)
	(20,0,0)	(10,0,10)	(0,0,25)	(-5,0,10)	(-10,0,10)
	(20,5,10)	(10,5,10)	(0,5,10)	(-5,5,10)	(-10,5,10)
	(20,10,10)	(10,10,10)	(0,10,10)	(-5,10,10)	(-10,10,10)

pesos

$$W = \begin{pmatrix} 1 & 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 \\ 2 & 1 & 1 & 1 & 1 \end{pmatrix}$$

y vectores de nodos

$$U = \{0, 0, 0, 0, 1, 1, 1, 1\}$$

$$V = \{0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1\}$$

las superficies resultantes para cada método de impresión se muestran en la figura A.5.

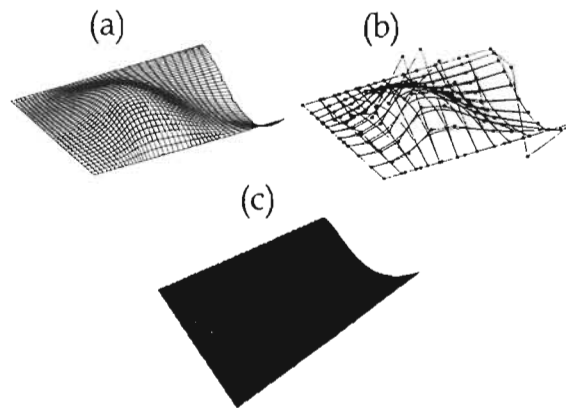


Figura A.5: Superficie NURBS de grado 3×2 . (a) Superficie en formato ps. (b) Superficie con puntos de control de sus isocurvas. (c) Superficie en formato VRML.

```

#include <nurbsS.h>

int main()
{
    // Superficie B-spline

    int i,j;

    using namespace PLib ;

    int p = 3, q = 2;
    int n = 3, m = 4; // Número de puntos, 4(0,...,3) en dirección u y 5 (0,...,4) en dirección v
    int a = 0, b = 1; // intervalo de los nodos

    Matrix_Point3Df Pts(n+1,m+1);
    Pts(0,0) = Point3Df(20,-10,10);
    Pts(1,0) = Point3Df(20,0,0);
    Pts(2,0) = Point3Df(20,5,10);
    Pts(3,0) = Point3Df(20,10,10);

    Pts(0,1) = Point3Df(10,-10,10);
    Pts(1,1) = Point3Df(10,0,10);
    Pts(2,1) = Point3Df(10,5,10);
    Pts(3,1) = Point3Df(10,10,10);

    Pts(0,2) = Point3Df(0,-10,10);
    Pts(1,2) = Point3Df(0,0,25);
    Pts(2,2) = Point3Df(0,5,10);
    Pts(3,2) = Point3Df(0,10,10);

    Pts(0,3) = Point3Df(-5,-10,10);
    Pts(1,3) = Point3Df(-5,0,10);
    Pts(2,3) = Point3Df(-5,5,10);
    Pts(3,3) = Point3Df(-5,10,10);

    Pts(0,4) = Point3Df(-10,-10,10);
    Pts(1,4) = Point3Df(-10,0,10);
    Pts(2,4) = Point3Df(-10,5,10);
    Pts(3,4) = Point3Df(-10,10,10);

```

```

//Pesos
Matrix_FLOAT W(n+1,m+1);
for(i=0;i<n+1;i++)
    for(j=0;j<m+1;j++)
        W[i][j]=1.0;

W[1][1] = 2;
W[2][4] = 2;
W[0][1] = 2;
W[3][0] = 2;

// Vector de nodos U y V
int r = p+n+1;
Vector_FLOAT U(r+1) ;

for(i=0;i<=p;i++)
    U[i] = a;
for(i=p+1,j=1; i<=r-p-1;i++,j++)
    U[i]=a+j*(float(b-a)/(r-2*p) );
for(i=r-p;i<=r;i++)
    U[i] = b ;

int s = q+m+1;
Vector_FLOAT V(s+1);

for(i=0;i<=q;i++)
    V[i]=a;
for(i=q+1,j=1;i<=s-q-1;i++,j++)
    V[i] = a+j*(float(b-a)/(s-2*q));
for(i=s-q;i<=s;i++)
    V[i] = b ;

PlNurbsSurfacef sbspline(p,q,U,V,Pts,W);
sbspline.writePS("SupBspline.ps",40,40,Point3Df(0,0,0),Point3Df(1.5,1.5,1.5),false,15,5);
sbspline.writePS("SupBspline2.ps",10,10,Point3Df(0,0,0),Point3Df(1.5,1.5,1.5),true,15,5);
sbspline.writeVRML("SupBspline.wrl",blueColor);
}

```

El siguiente programa define una superficie de Bézier bicúbica con puntos de control

Puntos de control				
i/j				
	(15,0,10)	(5,0,15)	(-5,0,15)	(-15,0,10)
	(15,5,15)	(5,5,25)	(-5,5,25)	(-15,5,15)
	(15,10,15)	(5,10,25)	(-5,10,25)	(-15,10,15)
	(15,15,10)	(5,15,15)	(-5,15,15)	(-15,15,10)

```

#include <nurbsS.h>

int main(){

    int p = 3, q = 3;
    int npr = p+1, npc = q+1;
    // Superficie de Bézier

    int i,j,k; // variables auxiliares

    using namespace PLib ;

```

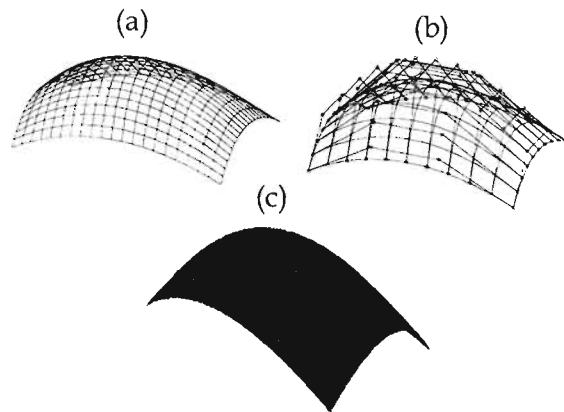



Figura A.6: Superficie de Bézier de grado 3×3 . (a) Superficie en formato ps. (b) Superficie con puntos de control de sus isocurvas. (c) Superficie en formato VRML

```

Matrix_Point3Df Pts(npr,npc);

Pts(0,0) = Point3Df(15,0,10);
Pts(1,0) = Point3Df(15,5,15);
Pts(2,0) = Point3Df(15,10,15);
Pts(3,0) = Point3Df(15,15,10);

Pts(0,1) = Point3Df(5,0,15);
Pts(1,1) = Point3Df(5,5,25);
Pts(2,1) = Point3Df(5,10,25);
Pts(3,1) = Point3Df(5,15,15);

Pts(0,2) = Point3Df(-5,0,15);
Pts(1,2) = Point3Df(-5,5,25);
Pts(2,2) = Point3Df(-5,10,25);
Pts(3,2) = Point3Df(-5,15,15);

Pts(0,3) = Point3Df(-15,0,10);
Pts(1,3) = Point3Df(-15,5,15);
Pts(2,3) = Point3Df(-15,10,15);
Pts(3,3) = Point3Df(-15,15,10);

Vector_Point3Df Vp(npr* npc);

for(k=j=0;k<npr* npc && j<npr;j++)
    for(i=0;i<npc;i++,k++)
        Vp[k] = Pts(i,j);

//Pesos
Matrix_FLOAT W(npr,npc);

for(i=0;i<npr;i++)
    for(j=0;j<npc;j++)

```

```

    W[i][j]=1.0;

    // Vector de nodos U y V

    Vector_FLOAT U(2*p+2) ;
    for(i=0;i<p+1;i++)
        U[i] = 0;
    for(i=p+1 ;i<U.n();i++)
        U[i] = 1.0 ;

    Vector_FLOAT V(2*q+2);
    for(i=0;i<q+1;i++)
        V[i]=0;
    for(i=q+1;i<V.n();i++)
        V[i] = 1.0;

    P1NurbsSurfacef s(p,q,U,V,Pts,W);

    s.writePS("SupBezInt1.ps",10,10,Point3Df(10,10,10),Point3Df(0,0,0),true,15,5);
    s.writePS("SupBezInt2.ps",20,20,Point3Df(10,10,10),Point3Df(0,0,0),false,15,5);
    s.writeVRML("SupBezInt.wrl",blueColor);

    return 0;
}

```

En el siguiente programa se construye una superficie NURBS, realizando la interpolación global de los puntos de la tabla del ejemplo anterior.

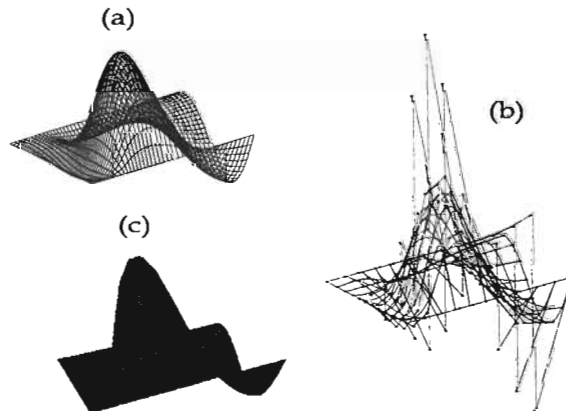


Figura A.7: Superficie de interpolación global de grado 3×2 . (a) Superficie en formato ps. (b) Superficie con puntos de control de sus isocurvas. (c) Superficie en formato VRML

```

#include <nurbsS.h>

int main()
{
    // Superficie B-spline

    int i,j;

    using namespace PLib ;

```

```

int p = 3, q = 2;
int n = 3, m = 4; // Número de puntos, 4(0,...,3) en dirección u y 5 (0,...,4) en dirección v
int a = 0, b = 1; // intervalo de los nodos

Matrix_Point3Df Pts(n+1,m+1);

Pts(0,0) = Point3Df(20,-10,10);
Pts(1,0) = Point3Df(20,0,0);
Pts(2,0) = Point3Df(20,5,10);
Pts(3,0) = Point3Df(20,10,10);

Pts(0,1) = Point3Df(10,-10,10);
Pts(1,1) = Point3Df(10,0,10);
Pts(2,1) = Point3Df(10,5,10);
Pts(3,1) = Point3Df(10,10,10);

Pts(0,2) = Point3Df(0,-10,10);
Pts(1,2) = Point3Df(0,0,25);
Pts(2,2) = Point3Df(0,5,10);
Pts(3,2) = Point3Df(0,10,10);

Pts(0,3) = Point3Df(-5,-10,10);
Pts(1,3) = Point3Df(-5,0,10);
Pts(2,3) = Point3Df(-5,5,10);
Pts(3,3) = Point3Df(-5,10,10);

Pts(0,4) = Point3Df(-10,-10,10);
Pts(1,4) = Point3Df(-10,0,10);
Pts(2,4) = Point3Df(-10,5,10);
Pts(3,4) = Point3Df(-10,10,10);

NurbsSurfacef sbspline;
sbspline.globalInterp(Pts,3,2);
sbspline.writePS("SupInterp1.ps",40,40,Point3Df(0,0,0),Point3Df(1.5,1.5,1.5),false,15,5);
sbspline.writePS("SupInterp2.ps",10,10,Point3Df(0,0,0),Point3Df(1.5,1.5,1.5),true,15,5);
sbspline.writeVRML("SupInterp.wrl",blueColor);

return 0;

```

Bibliografía

- [Blom91] *Bloomenthal M. y Riesenfeld R.F.*, **Approximation of Sweep Surfaces by Tensor Product NURBS**, Curves and Surfaces in Computer Vision and Graphics II, Society of Photo-Optical Instrumentation Engineers, Vol. 1610 páginas 132-144, 1991.
- [Boehm80] *Wolfgang Boehm*, **Inserting New Knots Into B-splines Curves**, Computer Aided Design (CAD), Vol. 12, No. 4, páginas 199-202, 1980.
- [Boehm85] *Boehm, W., y Prautzsch, H.* **The insertion algorithm**. Computer Aided Design (CAD), Vol. 17, No. 2, páginas 58-59, 1985.
- [Cohen] *Cohen E., Lyche T. y Riesenfeld R.* **Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics**, Computer Graphics and Image Processing, Vol. 14 páginas 87-111, 1980.
- [Cohen85] *Cohen E., Lyche, T., and Schumaker, L.L.*, **Algorithms for degree-raising of splines**, ACM TOG, Vol. 4, No. 3, pp. 171-181, 1985.
- [Coqu87] *Coquillart S.*, **A Control Point Based Sweeping Technique**, IEEE Comput. Graph and Appl., Vol 7, No.11, páginas 36-45, 1987.
- [Cox72] *M.G. Cox.*, **The Numerical Evaluation of B-splines** Jour. Inst. Math. Aplic., Vol. 10, pp. 134 – 149, 1972.
- [Cox82] *M.G. Cox.*, **Practical Spline Aproximation**, NPL Report Division of Information Technology and Computing, 1982.
- [DeBoor72] *De Boor, C.*, **On calculating with B-splines**, Jour. Approx. Theory, Vol. 6, pág.50-62, 1972.
- [DeBoor78] *De Boor, C.*, **A Practical Guide to Splines**, New York: Springer-Verlag, 1978.
- [DoCa] *Englewood Cliffs*, **Differential Geometry of Curves and Surfaces**, NJ: Prentice-Hall, 1976.
- [Far90] *Gerald Farin*, **Curves and Surfaces for CAGD, a practical guide**, Academic Press, Inc. Third Edition, 1990.
- [Friedberg] *Stephen H. Friedberg, Arnold J. Insel, Lawrence E. Spence.*, **Álgebra Lineal**, Publicaciones cultural, S.A.. Primera edición, 1982. pág.104-108.
- [Far91] *Gerald Farin*, **NURBS for curve and surface design**, SIAM Activity Group on Geometric Design, 1991.
- [GGM95] *Gasca González Mariano*, **Cálculo Numérico I** Universidad Nacional de Educación a Distancia, Madrid 1995.
- [HL97] *Josef Hoschek, Dieter Lasser*, **Computer Aided Geometric Design**, A. K. Peters, Wellesley, Massachusetts, 1997.
- [Klok] *Klok, F.*, **Two moving coordinate frames for sweeping along a 3D trajectory**, Computer Aided Geometry Desing, Vol. 3, pág. 217-229, 1986.
- [Lang] *Serge Lang.*, **Algebra**, Addison-Wesly Publishing Company, 1993.
- [Lee87] *Eugene T. Y. Lee.*, **Rational quadratic Bézier representation for conics**, in *Geometric Modeling: Algorithms and New Trends*, Faring, G.E., Ed., Philadelphia:SIAM, pp. 3-19, 1987.
- [Marsh] *Marsh Duncan.*, **Applied Geometry for Computer Graphics and CAD.**, Springer-Verlag London Berlin Heidelberg, pág. 42-47.
- [Pie94] *Piegl, L. y Tiller, W.*, **Software engineering approach to degree elevation of B-splines curves**. Computer Aided Design, Vol. 26, No. 1, páginas: 17-28, 1994.
- [Prau84] *Prautzsch H.*, **Degree elevation of B-spline curves**. Computer Aided Design, Vol. 1, No. 1, páginas: 193-198, 1984.
- [Prau91] *Prautzsch, H., y Piper, B.*, **A fast algorithm to raise the degree of spline curves**. Computer Aided Geometry Design, Vol. 8, páginas: 253-265, 1991.

- [PT97] *Les Piegl, Wayne Tiller, The NURBS Book*, Springer-Verlang 1997.
- [Rog01] *David F. Rogers, An Introduction to NURBS With Historical Perspective*, Morgan Kaufmann Publishers 2001.
- [Scho46] *Schoenberg, I.J., Contributions to the problem of approximation of equidistant data by analytic functions*, Quart. Appl. Math, Vol. 4, pág. 45-99, 1946.
- [Silt] *Siltanen, P., and Woodward, C., Normal orientation methods for 3D offset curves, sweep surfaces and skinning*, Proc. Eurographics 92, Vol. 11, No.3, pág. C-449-C-457, 1992.
- [tesis] *Cervantes Cabrera Daniel, Diseño de Curvas en Forma Libre con B-splines*, Tesis de Licenciatura, Facultad de Ciencias UNAM, 2000.
- [Wood87] *Charles D. Woodward, Cross-Sectional Design of B-splines Surfaces*. Computer and Graphics, Vol. 11, No. 2, páginas 193-201, 1987.