

SEMINARIO DE INVESTIGACIÓN EN MODELAMIENTO
GEOMÉTRICO

LIDA MARIANA CASTAÑEDA CABRERA
RICARDO CALDERÓN SIERRA
ANDRÉS MAURICIO VERA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2010

SEMINARIO DE INVESTIGACIÓN EN MODELAMIENTO
GEOMÉTRICO

LIDA MARIANA CASTAÑEDA CABRERA

RICARDO CALDERÓN SIERRA

ANDRÉS MAURICIO VERA

Trabajo de grado para optar al título de
Ingeniero Mecánico

Director

M.Sc. PEDRO JOSÉ DÍAZ GUERRERO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2010

A Dios, por permitirme alcanzar mis metas.

A mi hijo Nicolas que es la luz de mi vida.

A mi esposo, mi apoyo incondicional.

A mi madre que siempre ha estado a mi lado.

A mi familia por su apoyo.

Mariana.

A mi madre Aminta Sierra, por su fuerza para luchar por sus hijos.

A mi esposa Liceida, por su compañía y apoyo incondicional.

A mi hijo Heydan Ricardo, por su inspiración y cariño.

A Visión Mundial por su oportuna ayuda.

Ricardo.

A mi madre.

Andrés.

AGRADECIMIENTOS

A la Universidad Industrial de Santander por posibilitar nuestro crecimiento profesional.

A la escuela de Ingeniería Mecánica y su cuerpo docente por brindarnos las herramientas necesarias para nuestra formación.

A Pedro José Díaz Guerrero, Ingeniero Mecánico, director del proyecto, por su dedicación, apoyo, asesoría y orientación, lo cual permitió el desarrollo y la culminación exitosa de este proyecto de grado.

A nuestros compañeros y familiares que siempre nos apoyaron.

CONTENIDO

I SEMINARIO DE INVESTIGACIÓN	25
1. GENERALIDADES DEL SEMINARIO DE INVESTIGACIÓN	26
1.1. ¿Qué es el Seminario de Investigación?	26
1.2. Objetivo del Seminario de Investigación	27
1.3. Ventajas del Seminario de Investigación	27
1.4. Características del Seminario de Investigación	28
1.5. Organización del Seminario de Investigación	29
1.6. Tema del Seminario de Investigación	30
1.6.1. Herramientas del modelamiento geométrico	30
1.6.1.1. Interpolación Polinomial	30
1.6.1.2. Curvas de Bézier	31
1.6.1.3. Curvas Spline	32
1.6.1.4. Curvas Racionales	32
1.7. Dirección del Seminario de Investigación	33
1.8. Metodología del Seminario de Investigación	33
2. PLANEACIÓN	34
2.1. Descripción de los subtemas	39
3. EJECUCIÓN	43
3.1. El Acta	45

II	Resultados	49
4.	PRESENTACIONES, DOCUMENTO Y CÓDIGOS	50
4.1.	Presentaciones	50
4.1.1.	Características	50
4.2.	Documento	52
4.3.	Códigos	54
4.3.1.	Plataforma de realización de los códigos	54
	CONCLUSIONES Y RECOMENDACIONES	58
	Bibliografía	63
III	Anexo	64
A.	Teoría de interpolación polinomial	66
A.1.	Preliminares	66
A.1.1.	Consideraciones numéricas	69
A.2.	Interpolación polinomial	69
A.2.1.	Unicidad del polinomio de interpolación	72
A.3.	Método de Lagrange	72
A.3.1.	Propiedades de los polinomios de Lagrange	73
A.4.	Fórmula de Newton	77
A.4.1.	Diferencias divididas	80
A.5.	Error en la interpolación Polinomial	84
A.5.1.	Una expresión para el error	85
A.5.2.	El efecto Runge	86
A.6.	Interpolación de Tchebycheff	86
A.6.1.	Polinomios de Tchebycheff	87
A.6.2.	La propiedad de la norma mínima	88
A.6.3.	Raíces de $T_n(x)$	89
A.6.4.	Método de interpolación de Tchebycheff	89

A.7.	Problemas con interpolación Polinomial	94
A.7.1.	Minimización de la oscilación: Splines cúbicos	96
A.8.	Interpolación de Hermite	97
A.8.1.	Elección del soporte	99
B.	Curvas de Bézier	104
B.1.	Polinomios de Bernstein	105
B.1.1.	Definición polinomios de Bernstein	105
B.1.2.	Propiedades de los polinomios de Bernstein	108
B.2.	Curvas de Bézier simples	109
B.2.1.	Propiedades de las curvas de Bézier	114
B.2.2.	Conversión entre representaciones	123
B.2.3.	Técnicas constructivas para curvas de Bézier simples	128
B.2.4.	Reparametrización de la curvas de Bézier	131
B.2.5.	Curvas de Bézier Compuestas	132
B.3.	Curvas de Bézier Compuestas	137
B.3.1.	Subdivisión	140
B.3.2.	Control local de forma	146
B.3.3.	Interpolantes Cúbicos Compuestos.	147
B.3.3.1.	Interpolante Cúbico de Hermite	148
B.3.3.2.	Estimación de las derivadas	149
C.	Curvas Spline	151
C.1.	Spline cúbico de interpolación	151
C.1.1.	Spline cúbico completo de interpolación	159
C.1.2.	El spline natural	159
C.1.3.	Base de potencias truncadas	163
C.1.4.	Base B-Splines	165
C.1.5.	Nodos múltiples	168
C.1.6.	Nodos múltiples y potencias truncadas	171
C.1.7.	Espacios de los splines cúbicos	172

C.2. Splines generales	172
C.2.1. Bases de potencias truncadas	173
C.2.2. Base de B-Splines	174
C.2.3. Relación de recurrencia para B-Splines	187
C.2.4. Evaluación de B-Splines por recurrencia	194
C.2.5. Propiedades de los B-Splines	196
C.2.6. Polinomios de Bernstein como funciones B-Splines	197
C.3. Curvas B-Spline	198
C.3.1. Propiedades de las curvas B-Spline	199
C.3.2. Inserción de nodos. Algoritmo de Boehm	207
C.3.3. Inserción múltiple. Algoritmo de De Boor	215
C.3.4. Reducción de la multiplicidad	215
C.4. Interpolación con curvas B-Spline	222
C.4.1. Soporte de Interpolación basado en longitud de cuerda	222
C.4.2. Bases de B-Spline para interpolación	223
C.4.3. Interpolación con condiciones en los extremos	229
C.4.4. Interpolación local	231
D. Curvas Racionales	233
D.1. Secciones cónicas	234
D.2. Curvas racionales como proyecciones	238
D.3. Curvas de Bézier racionales	239
D.3.1. Propiedades de las curvas de Bézier Racionales.	241
D.4. Curvas de Bézier racionales compuestas	244
D.4.1. Secciones cónicas como curvas de Bézier racionales compuestas .	246
D.5. NURBS	248
D.5.1. Algoritmos básicos para curvas NURBS	251
D.5.2. Secciones cónicas como NURBS	252
E. Superficies 3D	257
E.1. Introducción	257

E.1.1.	Silueta de una Base	259
E.2.	Superficies producto tensorial	261
E.2.1.	Interpolación con superficies producto tensorial	262
E.2.2.	Superficie Bilineal	266
E.3.	Superficies paramétricas	268
E.3.1.	Curvas Isoparamétricas	269
E.3.2.	Superficies Regladas	270
E.3.3.	Superficies Cilíndricas	271
E.3.4.	Superficies de revolución	272
E.3.5.	Superficies Swing	276
E.3.6.	Superficies generadas por Barrido	279
E.4.	Lofting	286
E.4.1.	V-Lofting	286
E.4.2.	V-Lofting con especificación de derivadas.	295
E.4.3.	Posicionamiento de curvas de sección	299
E.4.4.	Superficies translacionales	299
E.4.5.	Superficies de Coons	301
E.4.6.	Superficies de Gordon. UV-Lofting	302
F.	Superficies rectangulares	307
F.1.	Superficies de Bézier simples	308
F.1.1.	Propiedades	308
F.2.	Superficies de Bézier compuestas	311
F.2.1.	Composición de dos superficies simples	311
F.2.2.	Superficies compuestas rectangulares	313
F.2.3.	Interpolación con superficies de Bézier C^1	314
F.3.	Superficies B-Spline	315
F.3.1.	Propiedades de las superficies B-spline	315
F.3.2.	Interpolación con superficies B-Spline	318
F.4.	Superficies racionales	319
F.4.1.	Patches de Bézier Racionales	321

F.4.2.	Cuádricas	322
F.4.3.	Superficies B-Spline racionales (NURBS)	322
F.5.	Técnicas constructivas	323
F.5.1.	Superficies regladas	323
F.5.2.	Superficies cilíndricas	323
F.5.3.	Superficies de revolución	324
F.5.4.	Superficies de barrido	325

LISTA DE TABLAS

2.1. Subtemas de investigación	37
3.1. Temas y subtemas con su respectivo relator	43
4.1. Códigos relacionados a los subtemas de investigación	56
A.1. Cálculo de las diferencias divididas	82
E.1. Clasificación de las superficies 3D	284

LISTA DE FIGURAS

3.1. Modelo del acta	47
3.2. Ejemplo del acta diligenciada	48
4.1. Formato de presentación	51
4.2. Ejemplo de diapositiva	53
4.3. Interfaz de LyX.	54
A.1. Interactividad	66
A.2. Tres ecuaciones interpolando 2 puntos	67
A.3. Polinomio interpolador	71
A.4. Polinomio de apoyo para x_k	74
A.5. Oscilación del error	85
A.6. Efecto Runge para $n = 5, 9, 13$	86
A.7. Diferente soporte para \sqrt{x}	87
A.8. Soporte por Tchebycheff	91
A.9. Soporte equiespaciado	92
A.10. Soporte Tchebycheff para $n = 8, 12, 16$	93
A.11. La poligonal que une los puntos	96
A.12. Polinomio obtenido por el método de Hermite	99
A.13. Tipos de soporte	101
B.1. Pierre Bézier. Fuente: www.davidsalomon.name/CaS/CaS.html	105
B.2. Polinomios de Bernstein	109
B.3. Curva de Bézier grado 1	110

B.4. Curva de Bézier grado 2	111
B.5. Curva de Bézier grado 3	112
B.6. Curva de Bézier	114
B.7. Derivada de curva de Bézier	116
B.8. Control seudolocal	119
B.9. Poligonal convexa	121
B.10. Disminución de la variación	122
B.11. Curva de Bézier	123
B.12. Discontinuidad de la primera derivada	133
B.13. Continuidad de la dirección	134
B.14. Curva de Bézier compuesta	140
B.15. Subdivisión	143
C.1. Listón elástico. Fuente: http://en.wikipedia.org/wiki/Flat_spline . . .	152
C.2. Spline cúbico completo de interpolación	156
C.3. Spline natural	161
C.4. Potencia truncada	165
C.5. B-Spline	166
C.6. Multiplicidad de un nodo	169
C.7. Variación de la multiplicidad de un nodo	170
C.8. B-Splines de grado 0	176
C.9. B-Splines de grado 1, nodos simples	178
C.10. B-Splines de grado 1, multiplicidad $p + 1$ en los extremos	180
C.11. B-Splines grado 1, multiplicidad $p + 1$ en un nodo interno	181
C.12. B-Spline grado 2, nodos simples	183
C.13. B-Splines grado 2, multiplicidad p en los extremos	185
C.14. B-Spline grado 2, multiplicidad $p + 1$ en los extremos	186
C.15. Curva B-Spline	207
C.16. Inserción de un nodo	214
C.17. Reducción de un nodo	221
C.18. Interpolación mediante B-Splines	229

D.1. puntos de control de una curva racional de Bézier	237
D.2. Curva racional como proyección	239
D.3. Sección cónica como curva de Bézier racional	255
E.1. Nodos formados por los soportes en u y v	263
E.2. Interpolación de las alturas	263
E.3. Superficie Bilineal	266
E.4. Líneas que interceptan por los puntos $P(0, 0), P(0, 1), P(1, 0), P(1, 1)$	267
E.5. Curva isoparamétrica	269
E.6. Familia de curvas	269
E.7. Superficie isoparamétrica	270
E.8. Superficie doblemente reglada	271
E.9. Superficie reglada	271
E.10. Superficie cilíndrica	272
E.11. Superficie de revolución	273
E.12. Rotación alrededor del eje z	274
E.13. Curvas planas contenidas en planos perpendiculares	277
E.14. Superficie swing	278
E.15. Superficie de barrido siguiendo una trayectoria	280
E.16. Superficie de barrido deformada	282
E.17. Curvas planas en un entorno tridimensional	286
E.18. Curvas a interpolar	289
E.19. Curvas de sección	300
E.20. Las dos curvas comparten un vértice	300
F.1. Patch de Bézier	310
F.2. Unión de dos superficies de Bézier	312
F.3. Generatriz	335
F.4. Trayectoria	335
F.5. Superficie de barrido	336

TITULO¹: SEMINARIO DE INVESTIGACIÓN EN MODELAMIENTO GEOMÉTRICO.

AUTORES²: LIDA MARIANA CASTAÑEDA CABRERA

RICARDO CALDERÓN SIERRA

ANDRÉS MAURICIO VERA

PALABRAS CLAVES: Modelamiento geométrico, Curvas de Bézier, Curvas B-Spline, Curvas racionales, Superficies 3D y Superficies rectangulares.

DESCRIPCIÓN:

El modelamiento geométrico es el proceso mediante el cual se puede representar gráficamente un objeto utilizando para ello las herramientas de la geometría.

Tradicionalmente, el dibujo de objetos con partes curvadas se había apoyado en las que se podrían denominar curvas analíticas para las que existían procedimientos exactos de dibujo. Las cónicas, por ejemplo, fueron incorporadas incluso a sistemas de modelado tridimensional computacionales, dando lugar al término lofting cónico, basado en definir secciones de una superficie mediante cónicas para luego recubrir estas secciones.

Gracias al trabajo de unos pocos pioneros provenientes de la industria automotriz se elaboraron los modernos métodos de modelamiento y se adoptó una nueva filosofía de modelamiento basado en curvas no analíticas. También contribuyó el empleo de computadores digitales potentes para que se utilizaran curvas polinomiales compuestas, y aun racionales, descartando otros sistemas de representación.

Los ingenieros P. De Casteljaou de Citroën y P. Bézier de Renault, desarrollaron separadamente y de forma simultánea, lo que pasaron a llamarse Curvas de Bézier, que fueron publicadas en 1966 por el ingeniero de Renault.

En los años 70, algunos autores encontraron la estrecha relación entre las curvas de Bézier y las splines, ya que las primeras eran casos particulares de las conocidas B-Splines. Con esto, se unificó de manera importante la teoría que había hasta el momento.

¹Trabajo de grado.

²Universidad Industrial de Santander, Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería Mecánica, Director: Ing. Mecánico, M.sc. Pedro José Díaz Guerrero

TITLE³: GEOMETRIC MODELING RESEARCH SEMINAR.

AUTHORS⁴: LIDA MARIANA CASTAÑEDA CABRERA

RICARDO CALDERÓN SIERRA

ANDRÉS MAURICIO VERA

KEY WORDS: Geometric Modeling, Bezier curves, B-spline curves, rational curves, 3D surfaces and rectangular surfaces.

DESCRIPTION

The geometric modeling is the process by which an object can be represented graphically using the tools of geometry.

Traditionally, the drawing of objects with curved sides had relied on what one might call analytical curves for that drawing procedures were accurate. The conical, for example, were incorporated even three-dimensional computer modeling systems, giving rise to the term lofting conical sections based on defining a surface and then coating these conic sections.

Thanks to the work of a few pioneers from the automotive industry developed modern methods of modeling and adopted a new philosophy of modeling based on non-analytical curves. He also contributed the use of powerful digital computers were used for composite polynomial curves, and even rational, as other systems of representation. Engineers P. From Casteljau Citroën and p. Bezier of Renault, developed separately and simultaneously, which became known as Bezier curves, which were published in 1966 by Renault engineer.

In the 70s, some authors found the close connection between Bézier curves and splines, because the former were known particular cases of B-Splines. This is an important unifying theory had so far.

³Thesis Project.

⁴Universidad Industrial de Santander, Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería Mecánica, Director: Ing. Mecánico, M.Sc. Pedro José Díaz Guerrero

INTRODUCCIÓN

La escuela de Ingeniería Mecánica en su misión contempla la construcción, aplicación y divulgación del conocimiento científico. Para cumplir con dicha misión es necesario implementar herramientas que faciliten el desarrollo del conocimiento científico y el mejoramiento de los procesos de investigación. Una forma de contribuir al logro de la misión es investigar las bases de las curvas y superficies involucradas en el modelamiento geométrico.

El seminario de investigación en modelamiento geométrico es un proceso reflexivo, sistemático y crítico que tiene como propósito fortalecer en el estudiante las habilidades requeridas en el manejo de la información y la comunicación para desarrollar investigación científica, valiéndose de la formación para el trabajo tanto personal como grupal, e iniciando el estudio de nuevos objetos de investigación de interés para la Escuela mediante una dinámica que comprende actividades de relatoría, correlatoría, discusión y elaboración de un documento de síntesis, en el cual se incluye el estudio de los referentes contextuales de modelamiento geométrico, desarrollando así todo el tema de manera compacta teniendo cohesión entre los subtemas que conforman el seminario. Adicionalmente se genera continuidad y profundización en el estudio de la técnica de modelamiento geométrico.

El investigador se forma haciendo ciencia, produciendo un trabajo científico, que necesariamente tendrá que estar coherente con el nivel y demás posibilidades de los participantes, llegando a ser estos trabajos fundamentales para la ejecución de nuevos proyectos, o el mejoramiento de antiguos proyectos de grado, en los que sean necesarias estas soluciones.

- Documento (recopilación de los subtemas).
- Presentaciones de los subtemas.
- Códigos en MATLAB para los ejemplos de aplicación.

En la primera parte se abordan temas que aclaran y dan una perspectiva del seminario de investigación y cómo el modelamiento geométrico es el tema de investigación.

En la segunda parte se explican los resultados obtenidos por medio del seminario, cumpliendo así con los objetivos.

Parte I

SEMINARIO DE INVESTIGACIÓN

Capítulo 1

GENERALIDADES DEL SEMINARIO DE INVESTIGACIÓN

1.1. ¿Qué es el Seminario de Investigación?¹

El Seminario de Investigación, también conocido como Seminario Alemán, es una actividad académica cuyo origen se dio a finales del siglo XVIII en la Universidad de GOTTINGEN de Alemania, para renovar las estrategias de estudio y formación de los investigadores, además para demostrar que la docencia y la investigación se pueden unir y complementar para obtener mejores resultados.

Consiste en estudiar, discutir e intercambiar experiencias acerca de un tema en particular, en un grupo, en el cual sus participantes se intercomunican exponiendo dicho tema (la Relatoría) complementándolo, evaluándolo (Correlatoría), aportando entre todos (la Discusión), sacando conclusiones y planteando nuevos interrogantes lo que permite que todo ello quede en la memoria escrita (el protocolo).

El Seminario de Investigación se programa por subtemas, estos son seleccionados con la orientación del director del seminario, quien con su experiencia y conocimiento del tema central, guía la selección con la debida pertinencia, actualidad y ubicación en el contexto. Los temas son desarrollados en sesiones planificadas, en las cuales los miembros del grupo deben asumir diferentes roles de acuerdo con la descripción anterior y mantener

¹Universidad Industrial de Santander. Lineamientos para el seminario de investigación como modalidad para el desarrollo del trabajo de grado. Vicerrectoría académica. Septiembre de 2007

una relación de interés y compromiso con el conocimiento sin jerarquías en un clima de colaboración y participación activa.

1.2. Objetivo del Seminario de Investigación

El objetivo es formar a los participantes en la investigación científica mediante el desarrollo de habilidades específicas aplicadas al asumir los diferentes roles dentro del seminario. Dichas habilidades están orientadas a desarrollar la capacidad de lector crítico de resultados de investigación en cualquiera de las áreas del conocimiento, a fortalecer la capacidad de observar e identificar los problemas presentes en temas bajo análisis, a buscar respuestas a preguntas clave y sustentarlas teórica y metodológicamente en forma verbal y por escrito; y a identificar las relaciones del problema objeto de estudio con el contexto económico, político o social a fin de enriquecer, con una mirada de integralidad, el conocimiento para el grupo de estudiantes. Para ello se programan y ejecutan ejercicios estructurados que permiten a los estudiantes desarrollar competencias iniciales de investigador, avanzar en el conocimiento y aportar buenas revisiones y análisis sobre temas que pueden facilitar el desarrollo de la investigación.

Para alcanzar dicho objetivo es preciso que haya una formación desde el trabajo personal hacia el trabajo en equipo; para esto, cada participante debe reconocer sus intereses, estilos de aprendizaje, capacidades para aprender en interacción con pares; además debe apropiarse de la metodología y los instrumentos con los cuales trabajará con el fin de lograr, al interactuar con los demás miembros del grupo en las sesiones del seminario, compartir, criticar y corregir las ideas que surjan de él en un ambiente de colaboración mutua.

Los seminarios de investigación no se enfocan hacia la repetición de trabajos ya realizados, sino hacia la búsqueda de respuestas con nuevos argumentos; por tal razón los trabajos que se deriven del cumplimiento del objetivo del Seminario deben caracterizarse por su originalidad y estar acordes al nivel científico de formación de sus participantes.

1.3. Ventajas del Seminario de Investigación

El seminario de investigación, como herramienta para el desarrollo integral, presenta las siguientes ventajas:

- Permite a los participantes contar con un director (profesor) durante el seminario,

el cual les guía hacia la consecución de los propósitos establecidos y resuelve las dudas e inquietudes, orienta sobre las fuentes de consulta y ayuda a los miembros del grupo en la búsqueda de información para suplir las necesidades que se presenten.

- Fortalece el hábito de documentarse acerca del tema bajo estudio. Para esto los participantes recurren a fuentes bibliográficas, bases de datos y textos de referencia obligada; este ejercicio refuerza el desarrollo de las competencias interpretativas, argumentativas y propositivas; permite apropiarse y aprender de los métodos que emplearon los autores de los artículos y textos, así como también reconocer su valor y aporte a la investigación.
- Permite que los participantes desempeñen diferentes roles dentro del grupo y desarrollen habilidades comunicativas y de relaciones interpersonales complementarias para la formación tanto personal como profesional.
- Fomenta el aprendizaje como una experiencia grupal. Permite experimentar la eficiencia del trabajo en equipo y, si el grupo está conformado por estudiantes de diferentes áreas del conocimiento, la riqueza de la interdisciplinariedad, características todas aplicables y necesarias en el desempeño laboral del mundo de hoy.
- Permite el uso de distintas herramientas didácticas de apoyo al desarrollo de las sesiones, así como un control sobre la planificación establecida de éstas.
- Es una metodología integradora centrada en el estudiante con amplio potencial para fortalecer la habilidad de aprender a aprender, fundamental para tomar el perfil del ciudadano del siglo XXI, quien deberá asumir el compromiso de aprender a lo largo de la vida como la plantea J. Delors².

1.4. Características del Seminario de Investigación

El seminario de investigación consta las siguientes características:

²Jaques Delors et al. La Educación Encierra un Tesoro. Informe a la UNESCO de la Comisión Internacional sobre la educación para el siglo XXI. París, 1996.

- Participación activa de todos los miembros del seminario, puesto que no solo el director (profesor) interviene, sino que también todos los integrantes del grupo realizan su aporte desde el rol que estén desempeñando. En este proceso los discípulos participantes empiezan a recorrer el camino hacia Maestros.
- El Seminario de Investigación está conformado por un grupo reducido, de aprendizaje activo y cooperativo; inducido a investigar, reflexionar, descubrir y concluir información del tema.
- Empleo del dialogo permanente para compartir los conocimientos adquiridos.
- Fomenta un ambiente amable y cooperativo que mejora la participación de los integrantes del grupo.
- Se desarrolla en sesiones que utilizan medios didácticos de apoyo al aprendizaje.
- La estructura, todas las actividades y parámetros para desarrollar el seminario son planificados en la primera sesión.
- El seminario de investigación exige a los participantes una alta responsabilidad para lograr la preparación adecuada que les permita tener bases para llevarlo a cabo.

1.5. Organización del Seminario de Investigación

El Seminario de Investigación se compone de las siguientes actividades: la Relatoría, la Correlatoría, la Discusión y el Protocolo, las cuales deben girar en torno a un tema, del que se desprenden los subtemas que se trataran durante las sesiones.

Dichas actividades son responsabilidad de los integrantes del grupo, por lo cual a cada uno de ellos es asignado un rol de carácter rotativo; es así como una persona que en una sesión asume el rol de correlator podrá ser el protocolante en la siguiente, es decir, el responsable del protocolo.

La organización del seminario también implica establecer el lugar donde se llevará a cabo, el número de sesiones y las fechas para realizarlo; así como la duración de cada una de las actividades (Relatoría, Correlatoría, Discusión y Protocolo).

El seminario de investigación en Modelamiento Geométrico está compuesto por:

Director: M.Sc.Pedro José Díaz Guerrero

Participantes: Lida Mariana Castañeda Cabrera

Ricardo Calderón Sierra

Andrés Mauricio Vera

A los cuales se les asigna un rol para cada sesión. Las sesiones se llevan a cabo en el salón 207 de las instalaciones de la Escuela de Ingeniería Mecánica. El número de sesiones depende del tema y los subtemas mostrados en la tabla 2.1. Las sesiones están programadas para una duración de 2 horas.

1.6. Tema del Seminario de Investigación

Un modelo geométrico describe las propiedades espaciales de un componente geométrico. Entre sus características destacan: su estructura espacial, la conectividad entre elementos, y las propiedades asociadas a componentes espaciales. Las técnicas usadas en modelamiento geométrico son la base fundamental del proceso de diseños asistido por computador, una de las primeras que surgió fue la teoría de interpolación, esta técnica fué enriquecida con la aparición de las curvas de Bézier, los B-Splines y las curvas racionales, que trasladan todas sus propiedades al proceso de modelado de superficies. El modelamiento geométrico tuvo un gran auge en los años 50 cuando los ingenieros de la industria automovilística intentaban desarrollar una aplicación que les permitiera modelar las carrocerías de los autos, para ello se empezaron a desarrollar técnicas de interpolación que con el paso del tiempo fueron mejoradas hasta obtener técnicas de alto rendimiento usadas en los software CAD. Sin lugar a dudas, hoy en día el estudio en modelamiento geométrico es una de las aéreas de desarrollo más importantes debido a la gran explosión de software de CAD-CAM en el mercado mundial. El proceso de construcción de conocimiento en el área de modelamiento geométrico requiere de personal capacitado y experimentado en el tema, tanto desde el punto de vista teórico como desde de la implementación computacional.

1.6.1. Herramientas del modelamiento geométrico

1.6.1.1. Interpolación Polinomial

El problema de la interpolación consiste en estimar el valor de una función en un punto a partir de valores conocidos en puntos cercanos. Para obtener esta estimación se aproxima la función con polinomios ya que son fáciles de evaluar y por el hecho

fundamental de que dados $n + 1$ puntos de abscisa distinta, $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, existe exactamente un polinomio $P_n(x)$ de grado no superior a n , que pasa por dichos puntos, es decir $P_n(x_i) = y_i$ para $i = 0, \dots, n$. Así, el problema de interpolación consiste en la obtención de un polinomio, llamado polinomio de interpolación, de grado menor o igual que n que pasa por $n + 1$ puntos (x_i, y_i) , $i = 0, 1, \dots, n$, también llamados nodos de interpolación.

Se plantean tres formulaciones diferentes para este problema que llevan al mismo polinomio interpolador:

1. Planteando directamente las condiciones anteriores se obtiene un sistema de ecuaciones lineales con solución única, pero generalmente mal condicionado o de difícil solución si el número de puntos es elevado.
2. Los polinomios de Lagrange permiten obtener una expresión explícita del polinomio de interpolación cuyo interés es más bien teórico, pues es difícil de evaluar en puntos concretos.
3. Numéricamente es mucho más útil la Fórmula de Newton del polinomio de interpolación. Aunque no tiene expresión explícita, su obtención es más estable que por los métodos anteriores, su evaluación no presenta los inconvenientes de los polinomios de Lagrange, y sobre todo, se puede actualizar fácilmente si se añaden nuevos nodos de interpolación.

1.6.1.2. Curvas de Bézier

Los polinomios no son tan buenas herramientas para el problema de interpolación. La idea de unir sucesivamente arcos polinomiales nació gracias al trabajo de dos ingenieros ya mencionados anteriormente: P. De Casteljan y P. Bézier: Esta idea se materializó en lo que hoy en día lleva el nombre de Curvas de Bézier.

Las curvas de Bézier se generan a partir de funciones polinómicas de grado tres que permiten la representación de cualquier forma curvada y evitan la complicación innecesaria de cálculos matemáticos que se produciría usando polinomios de mayor grado.

Cualquier trazado de estas características está definido por una serie de puntos por los que pasa la curva y otros exteriores a ella que definen sus puntos de inflexión, es decir, aquellos en que cambia de curvatura, pasando de cóncava a convexa o viceversa.

La definición paramétrica de la curva de Bézier es una función $Q(u)$, la cual está definida en términos del parámetro u donde $0 < u < 1$. Cuando u varía de 0 a 1 en el dominio de la función, el rango de la función encuentra una serie de puntos intermedios entre P_0 y P_3 , los cuales resultan de mezclas y escalas de los 4 puntos de control P_0, P_1, P_2 y P_3 . Esto es, cada punto en la curva es determinado escalando cada punto de control por un polinomio cúbico conocido como base o función de mezcla.

La curva es dada entonces por: Donde P_i es cada punto de control, B_i es el factor cúbico de escala o función de Blending, u es la aproximación o parámetro de la función.

1.6.1.3. Curvas Spline

Un Spline es una banda flexible que se utiliza para producir una curva suave a través de un conjunto de puntos designados. El término curva de Spline refiere a cualquier curva compuesta que se forma con secciones polinómicas que satisfacen condiciones específicas de continuidad.

Una curva de Spline se especifica a partir de un conjunto de posiciones de coordenadas, conocidas como puntos de interpolación, los cuales indican la forma general de la curva. Dado un conjunto de puntos, los métodos de interpolación generan una curva que pasa por todos ellos. En cambio, los métodos de aproximación generan una curva que normalmente no pasa por todos los puntos, excepto, tal vez, por los puntos extremos. Existen tres métodos equivalentes para especificar una representación de Spline particular:

- Se puede establecer el conjunto de condiciones de frontera que se imponen en la Spline.
- Se puede establecer la matriz que caracteriza al Spline.
- Se puede establecer el conjunto de funciones de combinación o funciones base que determinan la forma en que se combinan las restricciones geométricas de la curva para calcular posiciones a lo largo de la misma.

1.6.1.4. Curvas Racionales

Todos los sistemas basados en curvas polinómicas o constituidas por trazos polinómicos comparten el mismo problema: son incapaces de modelar curvas primitivas tan sencillas y utilizadas como las circunferencias, ya que requieren estructuras de datos independientes y técnicas de manipulación específicas.

Las curvas racionales ofrecen una representación exacta para curvas cuadráticas, como las circunferencias o elipses. Son invariables con respecto de una transformación de vista en perspectiva, es decir, que se puede aplicar una transformación de vista de perspectiva en los puntos de control de la curva y obtener la vista correcta de la curva.

1.7. Dirección del Seminario de Investigación

La dirección del Seminario de Investigación en modelamiento geométrico estuvo a cargo del profesor: M.sc. Pedro José Díaz Guerrero que cuenta con la siguiente formación:

Formación
Doctorado en Tecnología Energética Universidad Politécnica de Valencia, UPV, España Título: Estudio y modelado del flujo en tubos capilares adiabáticos en sistemas de refrigeración. Tutor: José Miguel Coberán Salvador Becado de: Universidad Politécnica de Valencia
Pregrado / Universitario en Ingeniería Mecánica Universidad Industrial de Santander, UIS, Colombia Título: Diseño de válvulas de seguridad en sistemas hidráulicos Tutor: Abel Parada Corrales

1.8. Metodología del Seminario de Investigación

Para llevar a cabo el Seminario de Investigación en modelamiento geométrico satisfactoriamente y garantizar la fluidez de las sesiones y cumplimiento de los objetivos planteados, se escogió una metodología reunida en tres grupos: planeación, ejecución y finalización; las cuales se relacionan entre sí dependiendo cada una de la anterior. Grupos que se explicarán en los siguientes capítulos.

Capítulo 2

PLANEACIÓN

En esta etapa se establecen los lineamientos bajo los cuales se desarrolla el seminario, se definen los alcances y resultados que se esperan obtener mediante la selección del tema, el estudio bibliográfico, los subtemas; y la planificación de las sesiones.

Los alcances y objetivos del seminario son:

- Escribir dos (2) documentos en Latex, uno tipo book donde se recopile el contenido total del Seminario de Investigación de acuerdo con los temas especificados en la tabla 2.1, y otro tipo beamer donde se lleven a cabo las presentaciones de cada tema.
- Realizar presentaciones y codigos en MATLAB de los temas investigados durante la duración del Seminario de Investigación. Ver tabla 2.1.
- Realizar la revisión bibliográfica de cada uno de los temas relacionados en la tabla 2.1.

En cuanto a la selección del tema, se escogió uno que fuera de interés para todos los estudiantes participantes. Se hizo un estudio previo del tema, el cual arrojó que la formación de los estudiantes participantes cumple para desarrollar el seminario y el director cuenta con la capacidad de orientación y el perfil para dirigirlo.

Antes de iniciar el seminario se definió la bibliografía para estudiar el tema seleccionado, la cual es:

- CORDERO VALLE, Juan M. y CORTES PAREJO, José. Curvas y Superficies para Modelado Geométrico.

Este libro surgió a raíz de la colaboración que los autores prestaron en la elaboración de una plataforma comercial de desarrollo de software 3D. Es un texto bastante teórico y a la vez práctico sobre técnicas de manipulación de curvas y superficies de Bezier, Splines, B-Splines y NURBS, las cuales son la base del modelamiento geométrico, y que además, han sido incorporadas de manera frecuente en los últimos programas comerciales de dibujo y diseño.

Es un libro que se puede utilizar en cursos de iniciación, induciendo a los estudiantes al dar los primeros pasos en el manejo y comprensión de las bases matemáticas y geométricas del funcionamiento de los CAD/CAM /CAE fomentando creación de las herramientas pertinentes para la comprensión de temas de modelamiento geométrico para cursos avanzados. También incluye algoritmos que ayudan a la completa implementación de las técnicas tratadas.

Por estas razones, y teniendo en cuenta que son pocos los libros que se encuentran sobre este tema en nuestro idioma, hemos escogido este libro como nuestro texto guía en el desarrollo del Seminario de Investigación en modelamiento geométrico.

- MAX K. AGOSTON. Computer Graphics and Geometric Modeling. Implementation and Algorithms.

Este libro es complementario a nuestro texto guía porque cuenta con una serie de ejercicios bastante prácticos y algoritmos bien elaborados acerca de los temas que nos conciernen.

- MICHAEL E. MORTENSON. Geometric Modeling.

Este libro describe y compara la importancia de todos los métodos matemáticos para modelar curvas, superficies y sólidos. Prepara al lector para temas más avanzados, como el modelado 3D, CAD / CAM, animación y visualización científica. Incorpora referencias a lo largo del texto para dirigir al lector a los tratamientos más especializados de los temas. Cuidadosamente diseñado ilustraciones y ejercicios de apoyo de cada concepto matemático presentado. Ofrece cientos de ejercicios para poner a prueba la comprensión de los lectores.

- GERARLD FARIN. Curves and Surfaces for CAGD: A Practical Guide.

Este libro proporciona un tratamiento unificado de todos los importantes métodos de curva de diseño y de superficie, y está muy centrado en el movimiento de la teoría a la aplicación.

- DAVID SALOMON. Curves and Surfaces for Computer Graphics.

El libro ofrece una introducción a la superficie curva y la representación de modelos y técnicas que se utilizan en los campos de la modelización geométrica, de diseño geométrico asistido por ordenador (CAGD), y gráficos por ordenador.

- DIETER ROLLER, BERNADETTA KWINTIANA ANE. Geometric constraint modelling Boundary planar B-spline curves and control polyhedra for 5-axes response surface graph. Paper. 2009.
- PUNEET TANDON, MD. RAJIK KHAN. Three dimensional modeling and finite element simulation of a generic end mill. Paper. 2009.

Cada uno de los participantes consultó dicha bibliografía y su información se complementó con el avance de cada etapa en el desarrollo del seminario y, en caso de que esta información no fuera suficiente, el participante recurrió a otras fuentes que aportaron algo nuevo a la investigación del tema.

Una vez realizado el estudio bibliográfico fue notorio observar que cuenta con subtemas comunes, en un mismo orden y profundiza en aspectos relevantes que permiten comprender de manera integral la técnica del modelamiento geométrico. Se tiene en cuenta que ésta técnica está en una etapa introductoria para los participantes del seminario, y en general para la escuela de Ingeniería Mecánica, por tanto se seleccionaron los subtemas de la tabla 2.1.

El número de sesiones se determinó de acuerdo a los subtemas seleccionados, se asignó a cada participante un subtema para exponer arbitrariamente, de forma que quedaran equilibradas las cargas en los subtemas, a la vez se definieron los roles de cada uno. Al tener definido el número de sesiones se organizaron las respectivas fechas de tal modo que hubiera un lapso de tiempo previo para la preparación de los subtemas. También se deja claro qué participante se escoge como relator en cada sesión. Se escogió el martes para realizar las sesiones por su asequibilidad horaria, tanto para el director como para los participantes, y también por la disponibilidad de herramientas necesarias para el

desarrollo de cada sesión como el vídeobeam y el salón.

Todo los subtemas se resume como se muestra en la tabla 2.1

Cuadro 2.1: Subtemas de investigación

Tabla 2.1
1. Interpolación. <ul style="list-style-type: none">▪ Preliminares.▪ Interpolación polinomial.▪ Método de Lagrange.▪ Fórmula de Newton.▪ Error en la interpolación polinomial.▪ Interpolación de Tchebycheff.▪ Problemas con interpolación polinomial.▪ Interpolación de Hermite.▪ Interpolación en forma paramétrica.
2. Curvas de Bézier. <ul style="list-style-type: none">▪ Polinomios de Bernstein.▪ Curvas de Bézier simples.▪ Curvas de Bézier compuestas.

continúa en la página siguiente

Tabla 2.1

3. Curvas Spline.

- Spline cúbico de interpolación.
- Splines generales.
- Curvas B-Spline.
- Interpolación con curvas B-Splines.

4. Curvas racionales.

- Curvas racionales como proyecciones.
- Secciones cónicas.
- Curvas de Bézier racionales.
- Curvas de Bézier racionales compuestas.
- NURBS.

5. Superficies 3D

- Introducción.
- Superficies producto tensorial.
- Superficies paramétricas.
- Lofting.

continúa en la página siguiente

Tabla 2.1

6. Superficies rectangulares.

- Superficies de Bézier simples.
- Superficies de Bézier compuestas.
- Superficies B-Spline.
- Superficies Racionales.
- Técnicas constructivas.

2.1. Descripción de los subtemas

Interpolación polinomial: el propósito de empezar con el estudio de la interpolación polinomial es entender los principios matemáticos del proceso interpolatorio y su funcionalidad, básicamente se trata de esbozar una curva que pase (interpole) unos puntos en el espacio que a su vez describen una forma específica, lo que da como resultado ecuaciones matemáticas de interpolación otorgadas por los diferentes autores de los diferentes métodos que exponen varias técnicas de interpolación como el método de Lagrange, la fórmula de Newton, interpolación de Tchebycheff, interpolación de Hermite además de exponer los problemas que resultan de la interpolación polinomial así como el error característico de este método.

Es importante entender el significado y lo principal de cada método para la interpretación de las ventajas y desventajas de cada uno de ellos.

Curvas de Bézier: Las curvas de Bézier son elementos geométricos que se expresan en una base polinomial especial conocida como los polinomios de Bernstein, que gozan de excelentes propiedades que permiten la manipulación de la forma de la curva sin que esto signifique un alto coste computacional, la forma de estas curvas está influenciada por la presencia de unos puntos de control, que como su nombre lo indica controlan la forma de la curva ya que la atraen hacia ellos sin tocarla, solo los puntos extremos interpolan la curva. Para minimizar cálculos la base Bernstein será siempre de grado $n = 3$, ya que al aumentar el grado de la curva también aumenta el número de cálculos

que se deben llevar a cabo para la solución del sistema; para describir formas más complejas lo que se hace es pegar consecutivamente curvas de Bézier simples de acuerdo con las necesidades del problema de modelado, sin embargo esto acarrea serias restricciones en cuanto al tema de la continuidad entre curvas, las curvas de Bézier gozan de las propiedades de los polinomios de Bernstein, lo que contribuye con su facilidad de cálculo y transformación.

Curvas B-Spline:

Buscando construir curvas de Bézier de cualquier grado y con cualquier orden de diferenciabilidad en los nodos, se encontró que las condiciones que habría que ponerle a los puntos de control harían intratable el problema. Entonces, la curva buscada no debía formularse en términos de puntos de control. Al hacer esto, se prescindió de la formulación de Bézier, lo que permite construir una curva polinomial compuesta a través de un número restringido de coeficientes que no influyen en su diferenciabilidad. Una de estas curvas es el spline cúbico. Un spline cúbico es entonces, una curva definida en un intervalo dado a la cual se le ha impuesto que sea de continuidad dos, y en cada partición de su soporte está definida como un polinomio de tercer grado; por lo tanto, se dice que es una curva polinomial a trozos de tercer grado.

Con los splines cúbicos se trabajan bases sencillas que son nulas en todo su dominio excepto en un pequeño intervalo de este; una de estas bases se denominan B-Splines y tienen la propiedad de que para cada valor de su soporte solamente se necesitan cuatro B-Splines para definir la curva. Entonces, una curva con esta base pasa a llamarse curva B-Spline que además tiene otra propiedad muy importante y es que sus nodos de interpolación pueden ser múltiples y de esta manera modificar su continuidad de manera local. Pero, no solamente se pueden definir B-splines cúbicos, sino que el grado puede ser cualquiera; aprovechando la relación de recurrencia para B-Splines se puede obtener todas las curvas básicas relacionadas a un soporte dado. Aquí es conveniente mencionar que es posible representar un B-spline como polinomio de Bernstein y que comparte algunas de sus propiedades, entre ellas, la no negatividad y la partición de la unidad.

La construcción de una curva B-Spline se hace entonces combinando linealmente cada B-Spline (de grado determinado) de la base asociada a un soporte (también determinado), hallado por recurrencia, con cada coeficiente que controla la curva que para el caso pueden pasar a llamarse también puntos de control, pero a diferencia de los puntos

de control en las curvas de Bézier, estos están más cercanos a la curva y por tanto la controlan mejor. A los nodos de las curvas B-spline se les puede aumentar su multiplicidad tanto que interpole los puntos de control, lo cual se puede aprovechar para aproximar el polígono de control a la curva y que se pueda dibujar directamente. Esto es muy utilizado en la práctica para realizar el rendering de curvas B-spline. Ahora bien, el proceso de interpolación con curvas B-Spline empieza con la construcción del soporte de interpolación a partir de los puntos a interpolar dados. Luego, se calculan los nodos que servirán para hallar los B-splines de la base y finalmente encontrar los puntos de control utilizando la combinación lineal entre los B-splines y los puntos de interpolación .

Curvas Racionales: Las curvas racionales, solucionan una gran carencia que presentan que las curvas de Bézier y las B-Spline y es describir formas tan primitivas como las geometrías cónicas que en su representación paramétrica están identificadas por el cociente de dos funciones polinomiales, así que las curvas racionales están dadas por el cociente de dos curvas que pueden estar expresadas en bases, polinomiales, Bernstein o B-Spline (NURBS), esa base va acompañada de unos pesos w que en el caso de los puntos de control modifican la influencia que tienen estos sobre las curvas, por ello así como a cada base le corresponde un punto de control, cada punto de igual forma tiene su propio peso.

Casi todo lo que puede hacerse con una curva NURBS puede conseguirse con curvas B-Splines polinomiales y a menor coste. Es deseable que un sistema de modelado basado en NURBS opere en lo posible con curvas polinomiales y sólo convertir a formato NURBS cuando se deseen ajustes finos (modificando los pesos).

Superficies 3D: Son una extensión de los clásicos procedimientos utilizados con las curvas de Bézier, las B-spline y las curvas racionales por lo tanto gozan de sus mismas propiedades. Las superficies producto tensorial están dadas por la variación de los parámetros u y v , y la base en la que se expresa esta superficie es el resultado de una combinación lineal convexa de dos bases univariadas en el espacio, obtenida a partir del producto tensorial entre dichas bases, este producto se puede definir como todas las posibles combinaciones entre una base y otra, la ventaja que presenta este método es la posibilidad de interpolar en una variable mientras se mantiene constante la otra, es decir interpolar una curva en 2D, en un plano ubicado en el espacio resultado de

hacer la otra variable igual a una constante, esta interpolación se puede llevar a cabo por cualquiera de los métodos ya estudiados.

Las superficies Paramétricas, tienen una enorme ventaja ya que cuando se pasa a esta representación se expresan los valores de x , y y z en funciones de un espacio vectorial, con una base producto tensorial, lo que permite ir modificando los valores de los parámetros y de esta manera se recorre la superficie, obteniéndose las coordenadas de los puntos que están sobre ella.

La técnica usada por las superficies Lofting, se basa en posicionar curvas planas en un entorno tridimensional para a continuación recubrirlas generando así una superficie, su nombre puede ser v-lofting, u-lofting o uv-lofting, dependiendo del parámetro sea u o v del que dependan las curvas que se van a interpolar, cuando se habla de la superficie uv-lofting es porque se tienen curvas en v y en u respectivamente, esta interpolación se puede realizar por medio de cualquiera de las técnicas antes vistas, ya sea asociar a cada curva un polinomio de Lagrange, interpolación polinomial, o interpolación con superficies producto tensorial, siendo este último el más utilizado en la práctica.

Superficies rectangulares: Una superficie rectangular es aquella que se obtuvo usando como bases los polinomios de Bernstein o B-Splines; o de manera más general, usando curvas NURBS, con lo cual se puede definir un sistema de modelado basado en NURBS aplicable a las técnicas vistas en el subtema llamado *Superficies 3D*.

Capítulo 3

EJECUCIÓN

En la tabla 3.1 se muestra cómo se desarrolló el orden de las actividades. Cada sesión, compuesta por su relator (según tabla 3.1), correlator y los asistentes según el protocolo. El protocolo consta de los siguientes pasos:

- Levantar un acta en la cual se deja por escrito el número del acta, fecha de la sesión, lugar, roles, tema desarrollado, observaciones y conclusiones.
- Hacer un documento del tema expuesto (anexos).
- Dejar un archivo de las presentaciones del tema expuesto por el relator.

Cuadro 3.1: Temas y subtemas con su respectivo relator

Tabla 3.1	Relator
1. Interpolación Polinomial. Preliminares. Interpolación polinomial. Fórmula de Newton. Método de Lagrange Error en la interpolación polinomial. Interpolación de Tchebycheff.	Andrés Mauricio Vera

continúa en la página siguiente

Tabla 3.1	Relator
Problemas con interpolación polinomial. Interpolación de Hermite. Interpolación en forma paramétrica	
2. Curvas de Bézier Polinomios de Bernstein. Curvas de Bézier simples. Curvas de Bézier compuestas.	Mariana Castañeda
3. Curvas Spline Spline cúbico de interpolación. Splines generales. Curvas B-Spline. Interpolación con curva B-Splines	Ricardo Calderón
4. Curvas racionales Curvas racionales como proyecciones. Secciones cónicas. Curvas de Bézier racionales. Curvas de Bézier racionales compuestas. NURBS.	Andrés Mauricio Vera
5. Superficies 3D Introducción. Superficies producto tensorial Superficies paramétricas Lofting	Mariana Castañeda
6. Superficies rectangulares. Superficies de Bézier simples. Superficies de Bézier compuestas. Superficies B-Spline. Superficies Racionales Técnicas constructivas.	Ricardo Calderón

continúa en la página siguiente

3.1. El Acta

El acta es un documento escrito donde se hace constar la relación de los roles de los participantes en cada sesión del seminario de investigación. El documento modelo que se llevó durante todas las sesiones se observa en la figura 3.1.

La estructura del acta está determinada por su contenido, ya que cada apartado se corresponde con cada uno de los aspectos de la reunión. En el acta mostrada en la figura 3.1 sobresalen los siguientes apartados:

- **Numeración:** Cada acta se llevó con un número consecutivo que le correspondía, en total fueron 12 actas para 6 subtemas expuestos.
- **Fecha:** Incluye la fecha en que se realiza la reunión; se tuvo en cuenta que va primero el día, luego el mes y el año completo.
- **Hora:** Se anota la hora de inicio, el lapso de duración de la reunión fue dos horas, así se sabe la hora de finalización, por lo tanto no se anota.
- **Relator:** Es el expositor del tema correspondiente en la sesión del seminario, siempre tuvo como misión principal enriquecer, como resultado de su investigación y estudio, el saber de los demás, buscar por medio de una argumentación rigurosa y aportar algo nuevo que permitiera avanzar en el conocimiento sobre el objeto de estudio. Este relator siempre incentivo al grupo a la discusión y despertó interés de participación a todos los demás integrantes a través de explicaciones claras, un argumento riguroso e hizo planteamientos desde su punto de vista. El relator es el que va plasmando lo expuesto al documento informativo y su presentación se da con ayuda del correlator. Esta persona tiene toda la libertad para organizar su material de trabajo, puede escoger la bibliografía a su gusto, preparar la sesión antes y, con la ayuda de herramientas como el vídeobeam, puede desempeñar su rol como debe ser.
- **Correlator:** Es el que complementa y enriquece la relatoría. En este seminario se trabajó de manera que el correlator fuese el que hiciera la presentación de las sesiones para poder tener una idea amplia acerca del subtema, éstas presentaciones eran mostradas al relator con anticipación para la verificación y corrección de posibles errores. El correlator siempre estuvo atento a lo que decía el relator evaluando, de manera paralela, el contenido, la claridad y el manejo

de la exposición con el fin de plantear su reflexión personal de manera crítica y motivando a los participantes al debate.

- **Discusión:** Básicamente aquí están nombrados todos los participantes, puesto que la discusión es donde se confrontan las concepciones y apreciaciones sobre el tema. En las actas pasadas, durante el transcurso del seminario, siempre se nombró una persona con el fin de dar claridad la participación de cada estudiante en cada uno de los roles. La discusión fue de una forma oportuna y respetuosa hacia las opiniones de los demás; pues la crítica, es una parte muy importante que permite a los participantes fortalecer procesos cognitivos a partir de la escucha, la reflexión y la toma de notas sobre lo que se desea expresar para luego hablar.
- **Protocolo:** en la sesión se encargó a una persona de escribir la evidencia de cada subtema y un insumo para la elaboración del informe final, éste es el protocolo. El acta se leyó antes de comenzar cada sesión, ya que es un documento de registro de lo ocurrido en la anterior sesión y de los compromisos establecidos. Para la elaboración del protocolo, el protocolante utilizó una redacción clara y concisa, además plasmó las actividades tal y como fueron desarrolladas.
- **Tema desarrollado:** Se redactaron y describieron de manera precisa los diferentes subtemas tratados, estos se expusieron en orden como lo muestra la tabla 3.1.
- **Observaciones y conclusiones:** Se anotaron en el acta para cada una de las sesiones.

El protocolo o acta permite en su realización el ejercicio de la concentración para percibir y captar lo fundamental, los momentos más trascendentales de la sesión y plasmarlo por escrito.

Figura 3.1: Modelo del acta

SESIÓN _____ **SEMINARIO DE INVESTIGACIÓN EN MODELAMIENTO GEOMÉTRICO**

TEMA: _____

Fecha: _____

1. *Asistentes y funciones*
2. *Evaluación de la relatoría y la correlatoría*
3. *Desarrollo de la discusión.*
4. *Anotaciones y preguntas de los participantes*
5. *Reglas metodológicas utilizadas*
6. *Conclusiones*
7. *Tema de la siguiente sesión*

Firma de los asistentes

Director

Protocolante

Relator

Correlator

Figura 3.2: Ejemplo del acta diligenciada

**SESIÓN I SEMINARIO DE INVESTIGACIÓN EN MODELAMIENTO GEOMÉTRICO
INTERPOLACIÓN POLINOMIAL**

28 de Agosto de 2009

1. *Asistentes y funciones*

ANDRÉS MAURICIO VERA	Relator
RICARDO CALDERÓN	Correlator
MARIANA CASTAÑEDA	Protocolante
PEDRO JOSÉ DÍAZ	Director

2. *Evaluación de la relatoría y la correlatoría*

Se realizó la presentación del relator quien explico los siguientes temas:

- Teoría de interpolación
- Interpolación en espacios generales

3. *Desarrollo de la discusión.*

Se planteo la pregunta de qué es una función

4. *Anotaciones y preguntas de los participantes*

- La nomenclatura de los comentarios para el código de MATLAB deben ir en mayúscula y tabulados.
- El Director pidió al Relator y Correlator demostrar que para una función de grado 2 se cumple la teoría interpolación.

5. *Reglas metodológicas utilizadas*

Se siguió la metodología del seminario alemán, para la preparación y desarrollo de la sesión. Se utilizó Lyx para la edición del texto final, y MATLAB para generar los códigos de cada uno de los casos.

6. *Conclusiones*

- La teoría de interpolación desarrolló varios métodos que a partir de puntos en un plano que describen la forma deseada por el diseñador, esbozan la figura resultante, llevando a cabo un proceso interpolatorio de dichos puntos.

7. *Tema de la siguiente sesión*

Continuación interpolación Polinomial

Firma de los asistentes

PEDRO JOSÉ DÍAZ
Director

MARIANA CASTAÑEDA
Protocolante

ANDRÉS M. VERA
Relator

RICARDO CALDERÓN
Correlator

Parte II

Resultados

Capítulo 4

PRESENTACIONES, DOCUMENTO Y CÓDIGOS

4.1. Presentaciones

La presentación de trabajos de investigación en foros académicos en las modalidades oral, póster y resumen son la culminación del esfuerzo realizado por los investigadores. Si no se siguen lineamientos especiales de elaboración de presentaciones, el trabajo de investigación, al momento de su presentación, no podrá ser valorado por el auditorio de forma completa. Es por esto que se estableció una guía precisa para este seminario como se discute a continuación.

Los subtemas expuestos en cada sesión fueron desarrollados por medio de presentaciones con el siguiente formato:

4.1.1. Características

La diapositiva es un medio que facilita una exposición, pero para ello es necesario tener en mente los objetivos que se pretenden alcanzar y que básicamente son: 1- Servir como guía al expositor. 2- Evitar o minimizar el uso de apuntes. 3- Dar un orden lógico a la presentación. 4- Favorecer la captación del mensaje por el auditorio. A continuación se exponen las características que ayudan a cumplir estos objetivos:

- La comunicación humana es verbal y corporal. Mientras las palabras envían un mensaje, el cuerpo emite mensajes adicionales que pueden ser complementarios, afirmativos o, incluso, contradictorios del primero. El balance entre ambas formas de comunicación acompañadas con la exposición de imágenes, gráficas y textos favorecen que el mensaje llegue, por alguna de las vías de percepción del ser humano, a la audiencia que nos observa. Es por ello que el formato resulta

Figura 4.1: Formato de presentación



Capítulo 2. Curvas de Bézier

Mariana Castañeda,
Andrés Vera,
Ricardo Calderón.

Ingeniería Mecánica
Universidad Industrial de Santander

Dir. Pedro José Díaz



relevante. El color blanco del fondo ayuda a que las personas que están presentes durante la presentación centren su atención en esta y no cansen la vista durante la conferencia, además de dar tono y cuerpo a la exposición.

- Se establecieron márgenes en las diapositivas que restringen el uso de textos recargados y quien elabora la diapositiva se ve en la tarea de preparar bien su presentación al solo incluir en éstas ideas específicas y cortas.
- Se utilizó un tipo y tamaño de letra adecuado para facilitar la lectura y no dar motivo para confusiones.
- Una distribución simple y abierta para lograr máxima efectividad.
- Las imágenes cumplen una función nemotécnica, son más atractivas y seductoras que las palabras y los números, por lo que son más fácilmente recordadas por quienes las observan. Sirven a la audiencia como elemento “recordable” que, ligado

a la comunicación verbal y no verbal, permiten que las personas a las que se dirige el mensaje lo aprendan y puedan más tarde reproducirlo. El color y tamaño de las imágenes se seleccionaron de tal forma que combinaran con el fondo para dar una apariencia más agradable a la presentación.

- El título del capítulo y el tema están presentes en cada diapositiva para que el auditorio siempre sepa de qué tema se está tratando y no pierda el hilo de la exposición.
- El nombre de los participantes da sentido de pertenencia con el trabajo realizado en cada sesión y se convierte en un compromiso el trabajar en equipo.
- La información se organiza de forma lógica, dividida en ideas sencillas que facilitan su exposición, de acuerdo a la estrategia comunicativa más adecuada para transmitir el mensaje.

La figura 4.2 es una demostración de las diapositivas utilizadas durante las sesiones.

La totalidad de las diapositivas resultantes de la sesiones llevadas a cabo en el transcurso del seminario están recopiladas de forma digital; debido al gran número, y anexadas a este informe.

4.2. Documento

El documento escrito fue elaborado con todas las normas y técnicas de la metodología científica y éste no consiste en una serie de puntos de vista, de opiniones, de ideas, etc., sino que es producto de la acción de investigar, cuestionar, fundamentar y contribuir realmente al avance del seminario y se desarrolló, afinó, aclaró y enriqueció con los aportes individuales, la discusión de grupo y con la asesoría del director durante el transcurso de las sesiones.

El documento está dividido en seis (6) capítulos y es una recopilación de todos los subtemas expuestos durante las sesiones contenidas en la tabla 3.1 siguiendo el mismo orden para tener continuidad y conexión entre ellos; profundizando por medio de secciones y subsecciones sus contenidos aclarando las técnicas del modelamiento geométrico.

Dicho documento fue elaborado en un editor de texto **LyX**, que ofrece ventajas por sus cualidades interactivas de alto rendimiento y que está íntimamente relacionado con

Figura 4.2: Ejemplo de diapositiva

Introducción Polinomios de Bernstein Curvas de Bézier simples Curvas de Bézier compuestas	Propiedades de las curvas de Bézier simples Conversión entre representaciones Técnicas constructivas Reparametrización de curvas de Bézier
Propiedades de las curvas de Bézier simples V	

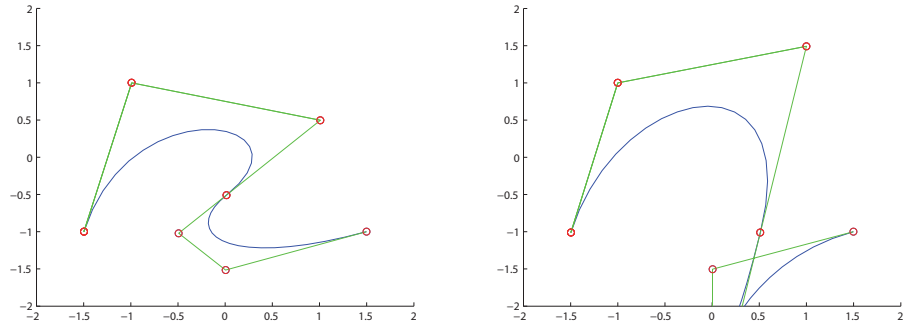


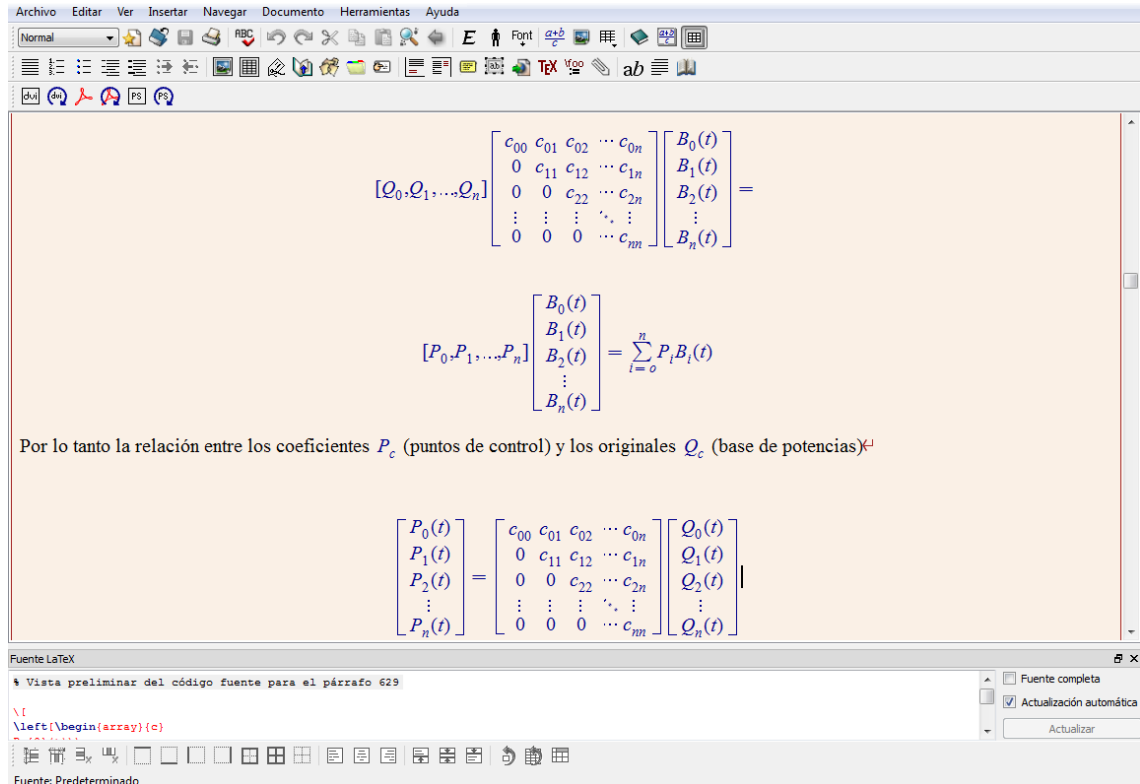
Figura: Control seudolocal

\LaTeX^1 , ya que se aprovecha de las características del mismo para producir textos de alta calidad en su formato. Gracias a estas funciones es posible ocupar en \LyX , directamente comandos que son propios de \LaTeX , siendo ambos completamente compatibles. Incluso es posible señalar que \LyX es un editor de documentos que sirve de interfaz a \LaTeX . Como sucede con LaTeX , en \LyX sólo es necesario preocuparse por el contenido, ya que los detalles del formato los entrega la aplicación, lo que convierte a \LyX en un software ideal para documentos de gran tamaño como libros o tesis. De hecho \LyX se ha convertido en un estándar para publicaciones científicas de alto nivel. Como se ha mencionado anteriormente, \LyX trabaja como lo hace \LaTeX , es decir bajo la filosofía WYSIWYM ("lo que ves es lo que quieres decir"), a diferencia de otros software que se basan en WYSIWYG ("lo que ves es lo que obtienes"). Esto es así ya que en \LyX solo se señala bajo ciertas etiquetas que es lo que se desea hacer y es la aplicación la que

¹Procesador de textos que está compuesto en su mayoría por órdenes (macros)

realiza los pasos necesarios para dar el formato correcto al documento. Convirtiéndolo en la herramienta ideal para publicaciones específicas que requieren un alto grado de complejidad como una publicación científica o una tesis doctoral.

Figura 4.3: Interfaz de LyX.



4.3. Códigos

Durante el desarrollo del seminario se realizaron códigos de algunos de los subtemas que se plantearon, que sirven para ampliar y entender de mejor forma las diferentes técnicas utilizadas en el modelamiento geométrico. En la tabla 4.1 se muestra los diferentes subtemas que cuentan con un código.

4.3.1. Plataforma de realización de los códigos²

Todos estos códigos se realizaron en el entorno de MATLAB que es un programa que permite desarrollar numerosas aplicaciones de cálculo, la representación y la

²Fuente: SUPERFICIES EN MATLAB COMO RECURSO DIDÁCTICO DE COMPRENSIÓN DE CONCEPTOS DEL CÁLCULO DE VARIAS VARIABLES Juan Carlos Molina García.

programación. MATLAB como en cualquier otro lenguaje de programación, y/o Asistente Matemático, se utilizan variables. Las variables deben tener un nombre según ciertas reglas. Estas reglas son:

- No pueden comenzar con un número, aunque si pueden tener números (variable1 es un nombre válido).
- Las máyusculas y mínusculas se diferencian en los nombres de variables. (A y a son dos variables diferentes)
- Los nombres de variables no pueden contener operadores ni puntos. (No es válido usar /, *, -, +, ...)
- Si se trabaja con complejos sólo puede utilizarse uno de los nombres i y/o j para variables.
- No es necesario definir el tipo de variable o tamaño (si se usa un vector y despues se expande, no hay problema)

Lo que hace verdaderamente poderoso al MATLAB es la facilidad para realizar operaciones matemáticas con elementos como:

- **MATRICES**

Asignación. Manejo de subíndices. Operaciones matemáticas simples (sumar, restar, multiplicar). Comandos matemáticos para matrices (invertir, determinante, valores propios...).

- **VECTORES**

Asignación. Manejo de subíndices. Operaciones matemáticas simples (sumar, restar, producto punto). Comandos matemáticos para vectores (producto cruz, longitud, normas ...).

- **NÚMEROS REALES Y COMPLEJOS**

Asignación de valores a variables. Nota importante sobre el uso de las variables i y j al trabajar con complejos. Operaciones matemáticas simples (sumar, restar, multiplicar, potenciación). Comandos matemáticos para reales y complejos (valor absoluto, raíz cuadrada, ángulo (para trabajar en polares), norma).

■ INTEGRALES DEFINIDAS

Se solucionan numéricamente por medio del comando TRAPZ.

Una vez establecida la estructura y la plataforma de realización de los códigos, se desarrollaron los siguientes códigos en cumplimiento a lo estipulado en la tabla 4.1:

Cuadro 4.1: Códigos relacionados a los subtemas de investigación

Tabla 4.1	Código
Interpolación Polinomial	X
Método de Lagrange	X
Fórmula de Newton	X
Interpolación de Tchebycheff	X
Problemas con interpolación polinomial.	X
Interpolación de Hermite.	X
Interpolación en forma paramétrica	X
Polinomios de Bernstein.	X
Control seudolocal	X
Derivas de las curvas de Bézier	X
Aumento del grado de la curva de Bézier	X
Disminución del grado de la curva de Bézier	X
Conversión entre representaciones:	
- Polinómico a Bézier	X
- Bézier a Polinómico	X
Curva de Bézier Simple	X
Curva de Bézier Compuesto	X
Subdivisión de la Curva de Bézier	X
Spline cúbico	X
Spline natural	X
Potencias truncadas	X
Base B-Spline	X
Nodos múltiples	X
B-Splines por recurrencia	X

continúa en la página siguiente

Tabla 4.1	Código
Evaluación de B-Splines por recurrencia	X
Evaluación de curvas B-Spline	X
Inserción de nodos	X
Reducción de nodos	X
Interpolación general con B-Splines	X
Lofting solución por superficies de producto tenso- rial	X
Superficies rectangulares swing	X

continúa en la página siguiente

CONCLUSIONES Y RECOMENDACIONES

- El seminario es una herramienta valiosa para la formación de investigadores y maestros en una disciplina. Sin dejar de reconocer, que siguen siendo válidas las cátedras magistrales; y lo serán mucho más si se evalúan críticamente y se aprovechan responsablemente las ventajas que poseen.
- En el desarrollo del seminario se partió del supuesto que se aprende a investigar investigando y que dicha tarea no simplemente fue la aplicación de ciertas técnicas, si no un proceso a través del cual se integraron conocimientos teóricos y prácticos en función de resolver los problemas planteados en la investigación.
- Gracias a los códigos generados en MATLAB fue posible encontrar resultados satisfactorios en la solución de problemas de interpolación, y modelado de formas curvas y superficies, así mismo se pudo mostrar las ventajas y desventajas de cada uno de los métodos estudiados.
- La teoría de interpolación desarrolló varios métodos, a partir de puntos en un plano, que describen la forma deseada por el diseñador y esbozan la figura resultante, llevando a cabo un proceso interpolatorio de dichos puntos.
- Una de las principales preocupaciones en interpolación polinomial es la elección del soporte, ya que de ello depende el desempeño del proceso interpolatorio, con miras a satisfacer las necesidades del diseñador, teniendo en cuenta el comportamiento de la curva, que presentará mayor o menor oscilación entre los puntos interpolados, así como la rapidez del computador en el momento de realizar los cálculos pertinentes.
- Se estudiaron 3 tipos de soportes y se evaluó su desempeño en la resolución de problemas de interpolación polinomial, los cuales se pudieron caracterizar así:

- El soporte centripetal maneja menor grado de oscilación de la curva entre los puntos de interpolación, aunque también es el que maneja mayor costo computacional.
 - El soporte uniforme (equiespaciado) genera un menor costo computacional, pero los resultados obtenidos no son los más recomendables debido a los altos niveles de oscilación de la curva entre los puntos de interpolación.
 - El soporte basado en la longitud de cuerda brinda muy buenos resultados y su costo computacional no es muy elevado dejándolo como la mejor opción para el desarrollo de problemas de interpolación que no requieran de total exactitud.
- El proceso de interpolación polinomial presenta varios inconvenientes que llevaron a la necesidad de investigar y posteriormente desarrollar técnicas más elaboradas que superaran las debilidades de este proceso.
 - Las curvas de Bézier son un sistema que permite una excelente interpretación geométrica de los valores explícitos de la ecuación de la curva, estos valores son los ya nombrados puntos de control que atraen hacia ellos la curva y ésta los interpola sólo en el punto inicial y final.
 - Por muchos años las curvas de Bézier fueron la herramienta de modelado más utilizada en la industria automotriz, sin embargo se necesitaría de un experto dibujante que supiera exactamente en donde ubicar los puntos de control para que la curva describiera la forma que finalmente se necesitaría obtener.
 - Como en interpolación polinomial el aumento del grado de la base Bernstein implica el aumento de los puntos de control, así como el control sobre la curva, sin embargo esto implicaría un elevado número de cálculos lo que se vería reflejado en una lenta respuesta del sistema modelador.
 - Las curvas de Bézier compuestas son una buena herramienta para describir formas complejas, sin embargo solo se pueden generar con una continuidad C^1 por el gran número de condiciones de interpolación que se deben cumplir en los puntos de unión de tramos consecutivos.
 - El splines cúbico se desarrolla para solventar las limitaciones de las curvas de Bézier: falta de control local, la laboriosidad requerida para imponer continuidad

C^2 y el hecho de que el número de puntos de control de una curva de Bézier imponga su grado.

- La multiplicidad de los nodos controla el orden de continuidad del mismo y por tanto la suavidad con que la curva es dibujada en un determinado nodo, ya que en la medida que se aumenta la multiplicidad de un nodo disminuye el número de trozos de polinomio que esbozan la forma requerida por el diseñador.
- Las curvas de Bézier y B-Splines no son tipos distintos de curvas, sino reparametrizaciones distintas para curvas constituidas por tramos polinomiales y con ciertos órdenes de continuidad en la unión de dos tramos.
- El polígono de control obtenido tras la inserción de un nodo constituye una aproximación a la curva mejor de lo que era el polígono inicial. Con una inserción apropiada de nuevos nodos, el polígono de control se hará parte de la curva y podrá ser dibujado en vez de la curva.
- Las curvas racionales se desarrollaron como técnicas del modelamiento geométrico para cubrir una gran carencia que presentaban las curvas de Bézier y las B-Spline: dificultad de representar secciones cónicas, cuya forma paramétrica está dada por el cociente de dos funciones.
- Las curvas racionales pueden expresarse en cualquiera de las bases estudiadas, e introducen el concepto de pesos asociado a los puntos de control o coeficientes (dependiendo de la base elegida).
- Una curva racional muy especial es la denominada NURBS. Casi todo lo que puede hacerse con una curva NURBS puede conseguirse con curvas B-Splines polinomiales y a menor costo. Es deseable que un sistema de modelado basado en NURBS opere en lo posible con curvas polinomiales y sólo convertir a formato NURBS cuando se deseen ajustes finos (modificando los pesos)
- Una superficie de producto tensorial es aquella que constituye una combinación lineal de todos los productos posibles entre las funciones de dos bases arbitrarias, cada una en una de las variables x e y , lo que ofrece la facilidad de resolver los problemas en dos pasos, primero en la variable x manteniendo constante el valor de y , y a continuación llevar a cabo el mismo procedimiento en la variable y .

- Uno de los principales limitantes en el proceso de interpolación de superficies 3D, es la elección de una base que no contenga demasiadas funciones, ya que esto está relacionado directamente con el número de cálculos necesarios para esbozar la superficie resultante.
- Las superficies paramétricas permiten definir las coordenadas de la superficie en función de parámetros variantes, ofreciendo un sistema flexible, aunque no orientables, ya que definir la ubicación espacial de las coordenadas a partir de expresiones en función de dos parámetros variantes resulta complicado.
- Las superficies regladas ofrecen grandes aplicaciones dentro de las funciones del proceso de diseño asistido por computador, ya que la mayor parte de las superficies generadas a partir de la técnica propuesta por este tipo de superficies, permiten manipular las características de la superficie para adaptarse a las necesidades de estudio de las diferentes industrias que demandan este tipo de aplicaciones.
- Una de las principales ventajas de las superficies 3D, es que se pueden expresar en cualquiera de las bases ya estudiadas, lo que implica que gozan de las propiedades de dichas bases, facilitando su transformación y manipulación, para acercarse más a las necesidades del diseñador.
- Las superficies lofting son la técnica más usada para la representación de superficies, ya que pueden ser solucionadas por cualquier método de interpolación o haciendo uso de las superficies de producto tensorial, una de las principales aplicaciones se da en el desarrollo de herramientas de recubrimiento de los software CAD-CAM.
- Una de las más grandes ventajas que tienen las superficies derivadas de las curvas simples, es la capacidad de trasladarse, rotarse o deformarse, aplicando la transformación deseada solo a los puntos de control, lo que implica un bajo costo computacional ya que la base no debe ser recalculada sino elegida a decisión del diseñador.
- Una de las principales desventajas al trabajar las superficies 3D es la incapacidad de los métodos estudiados de predecir con exactitud la forma de las superficies resultante para los siguientes casos:
 - Superficies de barrido:

- Si la curva generatriz y trayectoria no son planas y están contenidas en planos perpendiculares.
 - Si la curva se deforma, traslada y rota al mismo tiempo.
- Las propiedades de las superficies B-Splines polinomiales son compartidas por las racionales. Las superficies NURBS son generalizaciones de las superficies racionales y no racionales, así como de los B-Splines polinomiales.
- Las superficies NURBS constituyen el esquema de modelado de superficies más general posible entre los descritos. Sin embargo, algunos problemas prácticos no necesitan la generalidad y flexibilidad de las superficies NURBS, ya que la interpolación discreta y la transinfinita se resuelven de forma más eficiente con superficies o curvas no racionales.

Bibliografía

- [1] CORDERO VALLE, Juan M. y CORTES PAREJO, José. Curvas y Superficies para Modelado Geométrico.
- [2] MAX K. AGOSTON. Computer Graphics and Geometric Modeling. Implementation and Algorithms.
- [3] MICHAEL E. MORTENSON. Geometric Modeling.
- [4] GERARLD FARIN. Curves and Surfaces for CAGD: A Practical Guide.
- [5] DAVID SALOMON. Curves and Surfaces for Computer Graphics.
- [6] WINTIANA ANE. Geometric constraint modelling Boundary planar B-spline curves and control polyhedra for 5-axes response surface graph. Paper. 2009.
- [7] PUNEET TANDON, MD. RAJIK KHAN. Three dimensional modeling and finite element simulation of a generic end mill. Paper. 2009.

Parte III

Anexo

MODELAMIENTO GEOMÉTRICO

Apéndice A

Teoría de interpolación polinomial

A.1. Preliminares

En modelado computacional si se desea dibujar una curva (posiblemente el contorno de una figura) se inicia especificando unos puntos del plano que describan la forma de la curva que se desea dibujar. Cuando se lleva a cabo el proceso de interpolación de dos puntos: a manera superficial muchas soluciones, incluso infinitas pueden satisfacer las condiciones especificadas

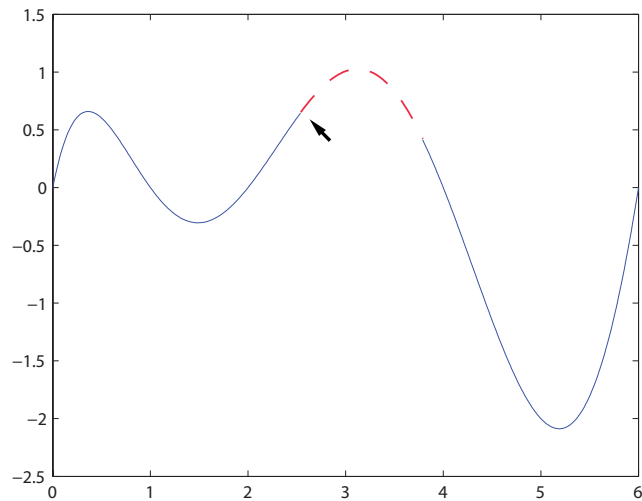


Figura A.1: Interactividad

$$\begin{aligned}
 y &= -x + 3 \\
 y &= x^2 - 2x + 3 \\
 y &= \frac{\cos(\pi * x + 5)}{2}
 \end{aligned}
 \tag{A.1}$$

Estas 3 ecuaciones satisfacen los puntos de interpolación (0,3) y (1,2) pues son las condiciones dadas ver fig(A.1)

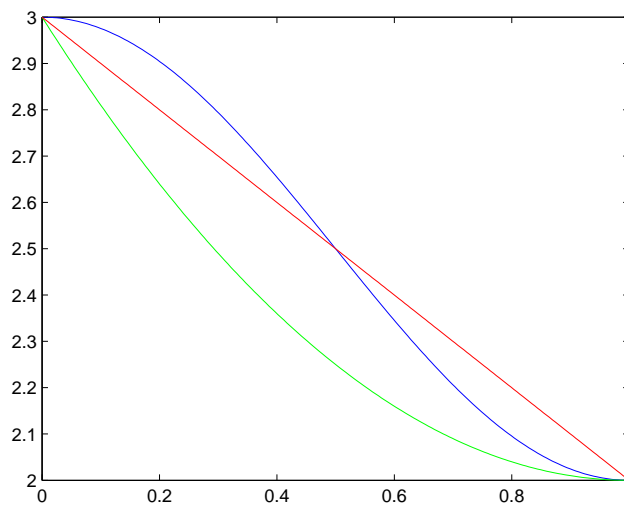


Figura A.2: Tres ecuaciones interpolando 2 puntos

Aún cuando los datos son puntos del plano (*parejas de coordenadas*) se deben discriminar entre abscisas y ordenadas. Si los datos son 2 puntos, el polinomio más sencillo es una recta que los una, o sea un polinomio de grado 1.

$$\begin{aligned}
 P(x_0) &= ax_0 + b = y_0 \\
 P(x_1) &= ax_1 + b = y_1
 \end{aligned}
 \tag{A.2}$$

Una solución puede ser $a = 0$, entonces

$$P(x) = b = y
 \tag{A.3}$$

Si son 3 puntos el planteamiento es el mismo 3 ecuaciones de la forma $ax^2 + bx + c = y$. Observando los anteriores casos parece razonable que al tener $(n + 1)$ puntos

$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ se debe buscar un polinomio de grado n y este dará la solución más sencilla al problema de interpolación, entonces $n + 1$ ecuaciones y $n + 1$ incógnitas, que de generar una solución dará el polinomio de interpolación buscado. Como se vió una solución puede ser cero, al interpolar $n + 1$ puntos se busca en el espacio de los polinomios de grado menor ó igual a n

$$n + 1 \Rightarrow p \leq n$$

Si no se desea interpolar polinomios sino otro tipo de funciones: trigonométricas, exponenciales ó racionales, se consideran funciones similares. Debería estar claro que para realizar una interpolación de $n + 1$ puntos con funciones diferentes a polinomios el espacio de trabajo debería ser de dimensión $n + 1$.

$$n + 1 \Rightarrow \mathcal{B}(n + 1)$$

Un espacio vectorial de funciones dimensión $n + 1$ con una base:

$$\mathcal{B} = (B_0(x), B_1(x), \dots, B_n(x)) \tag{A.4}$$

Una función cualquiera de este espacio

$$P(x) = a_0 B_0(x) + a_1 B_1(x) + \dots + a_n B_n(x) \tag{A.5}$$

Una condición de interpolación

$$\begin{aligned} P(x_0) &= a_0 B_0(x_0) + a_1 B_1(x_0) + \dots + a_n B_n(x_0) = y_0 \\ P(x_1) &= a_0 B_0(x_1) + a_1 B_1(x_1) + \dots + a_n B_n(x_1) = y_1 \\ &\vdots \\ P(x_n) &= a_0 B_0(x_n) + a_1 B_1(x_n) + \dots + a_n B_n(x_n) = y_n \end{aligned}$$

En forma matricial

$$\begin{bmatrix} B_0(x_0) & B_1(x_0) & \cdots & B_n(x_0) \\ B_0(x_1) & B_1(x_1) & \cdots & B_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_0(x_n) & B_1(x_n) & \cdots & B_n(x_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{A.6}$$

Si $\det(\mathcal{B}) \neq 0$ ¹ El sistema dará una solución única y se obtendrá la función interpoladora $P(x)$, el soporte (x_0, x_1, \dots, x_n) de la matriz \mathcal{B} puede perfectamente dar $\det(\mathcal{B}) \neq 0$, pero para otra elección del soporte $(x_a, x_b, \dots, x_z) \Rightarrow \det(\mathcal{B}) = 0$, por lo tanto se debe verificar la condición $\det(\mathcal{B}) \neq 0$, un ejemplo sencillo de anulación de $\det(\mathcal{B})$ ocurre en un espacio de funciones trigonométricas $1, \cos x, \cos 2x, \dots, \cos nx$ si el soporte tiene al menos 2 puntos separados 2π existirían 2 filas iguales $\det(\mathcal{B}) = 0$. De (A.6) $\mathcal{B}a = y$.

Es evidente que la matriz depende exclusivamente del soporte (x_0, x_1, \dots, x_n) y el termino independiente, función de las ordenadas (y_0, y_1, \dots, y_n) razón principal para discriminar (considerar de forma separada) abscisas y ordenadas. Esta cuestión se vuelve más practica por ejemplo cuando es necesario hacer varias interpolaciones con el mismo soporte común, entonces \mathcal{B} de (A.6) sería siempre igual y sólo se variaría el termino independiente y .

A.1.1. Consideraciones numéricas

Qué sucede cuando en la práctica se encuentran problemas numéricos en ciertos casos tipos de interpolación que son bastante comunes, esto ha llevado a mostrar la inviabilidad de dichos planteamientos. Hay que anotar que con mucha frecuencia los elementos de \mathcal{B} sobrepasan la precisión numérica del computador, por ejemplo en una interpolación de 100 puntos, algunos elementos podrían ser números con 200 cifras significativas, estos números son difíciles de operar y almacenar.²

Se sabe que el número de operaciones necesarias para resolver un sistema de n ecuaciones con n incógnitas es del orden $N = \frac{2}{3}n^3$. Para que el movimiento de cualquier elemento en una pantalla sea fluido, sin parpadeos la imagen debe ser recalculada cada $\frac{1}{25}$ de segundo. Un computador con procesador de un GB realiza 10^9 operaciones por segundo $\Rightarrow 391,48$ incógnitas.

El computador podría calcular interpolaciones interactivamente si el número de puntos es ≤ 400 .

A.2. Interpolación polinomial

Teniendo en cuenta que los polinomios a buscar serán de grado $\leq n$ se elige la base

¹Se asume que $\det(\mathcal{B}) \neq 0$, ahora bien, si posee infinitas soluciones se plantea una solución numérica para el cálculo de alguna de las soluciones, aparte de los problemas de no unicidad

²A menos que se opere con aritmética de alta precisión la matriz almacenada en el computador diferirá de la verdadera \Rightarrow soluciones altamente erróneas

$$\mathcal{B} = [1, x, x^2, \dots, x^n]$$

para llegar al polinomio habitual:

$$P_n(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n \quad (\text{A.7})$$

De (A.6) se transforma a la llamada matriz de *Vandermonde*:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \quad (\text{A.8})$$

Y su determinante viene dado por:

$$\det(\mathcal{B}) = \prod_{i \neq j} (x_i - x_j) \quad (\text{A.9})$$

Esta matriz de *Vandermonde* permite ver un problema muy posible con un soporte $S = [0, 1, 2, \dots, 98, 99]$ números que serian en (A.8) $99^{99}, 98^{98}$ etc, enteros con 200 dígitos decimales que no son faciles de guardar, al menos de forma usual.

Ejemplo Considerar los puntos de interpolación $(0,3), (1,5), (2,1), (2,0)$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 1 \\ 0 \end{bmatrix}$$

La solución es:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1,5 \\ -7,5 \\ 8 \\ 3 \end{bmatrix}$$

La solución del polinomio interpolador:

$$P_n(x) = \frac{3}{2}x^3 - \frac{15}{2}x^2 + 8x + 3$$

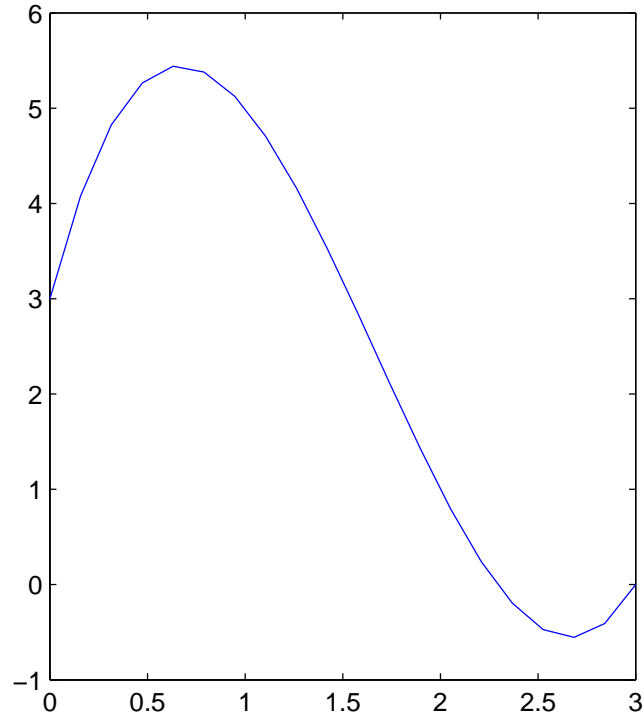


Figura A.3: Polinomio interpolador

Algoritmo: Interpolación polinomial

```

abs=[0 1 2 3]           % SOPORTE (PUNTOS EN X)
ord=[2;1;0;5]          % ABCISAS (PUNTOS EN Y)
n=length(abs)          % TAMAÑO DEL SOPORTE
for j=1:n               % CONSTRUCCION MATRIZ DE POTENCIAS
    for i=1:n
        m(j,i)=abs(j)^(i-1)
    end
end

a=inv(m)*ord           % SOLUCION AL SISTEMA DE ECUACIONES
z=a';v=length(z);y=z(v:-1:1) % ORDENANDO EL POLINOMIO

```

```
x=linspace(0,3,100);
yo=polyval(y,x);
plot(x,yo)
```

A.2.1. Unicidad del polinomio de interpolación

Es importante verificar que el polinomio calculado sea el mismo sin importar otros posibles métodos a seguir. Esta unicidad es solamente posible dentro del espacio P_n , polinomio³ de grado $\leq n$.

A.3. Método de Lagrange

De (A.8) el vector solución a simbólicamente $\Rightarrow a = B^{-1} y = Ly$, a esta inversa suele llamársele *matriz de Lagrange*, y este método conduce a un cálculo sencillo sin necesidad de la inversa de B .

Con el polinomio de interpolación $P_n(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n$ escrito de forma:

$$P_n(x) = \begin{bmatrix} 1 & x & x^2 & \dots & x^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}$$

Asumiendo que conocemos la matriz L

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = Ly = \begin{bmatrix} l_{00} & l_{01} & \dots & l_{0n} \\ l_{10} & l_{11} & \dots & l_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{n0} & l_{n1} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

³Recordando: con 2 puntos el polinomio solución de interpolación es de grado ≤ 1 que es único, por el contrario existen ∞ polinomios de grado 2 que los interpolan

Reemplazando el vector a en el polinomio de interpolación.

$$P_n = \begin{bmatrix} 1 & x & x^2 & \dots & x^n \end{bmatrix} \begin{bmatrix} l_{00} & l_{01} & \dots & l_{0n} \\ l_{10} & l_{11} & \dots & l_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{n0} & l_{n1} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Se tiene:

$$P_n(x) = [L_0(x) \ L_1(x) \ \dots \ L_n(x)] \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \quad (\text{A.10})$$

Multiplicando formalmente (A.10) se llega a la **Fórmula de Lagrange**. Estos polinomios L_k se asumen conocidos y la solución se halla sin tener que operar la matriz de interpolación,

$$P_n(x) = y_0 L_0(x) + y_1 L_1(x) + \dots + y_n L_n(x) \equiv \sum_{i=0}^n y_i L_i(x) \quad (\text{A.11})$$

A.3.1. Propiedades de los polinomios de Lagrange

$$L_{i,n}(x) = L_{i,n} \begin{bmatrix} 1 & x & x^2 & \dots & x^n \end{bmatrix} = \prod_{j=0, j \neq i}^n \left(\frac{x - x_j}{x_i - x_j} \right)$$

Los $L_k(x)$ dependen del soporte de interpolación, por lo tanto sean cuales sean las coordenadas y_i los polinomios de *Lagrange* están determinados por las abscisas x_i .

Se considera interpolar los puntos:

$$(x_0, 0) \ (x_1, 0) \ \dots \ (x_{k-1}, 0) \ (x_k, 1) \ (x_{k+1}, 0) \ \dots \ (x_n, 0) \quad (\text{A.12})$$

Se busca el polinomio P^n de grado $\leq n$ que se anula en todos los puntos de interpolación menos x_k donde toma el valor de 1. En la formula de *Lagrange* dice

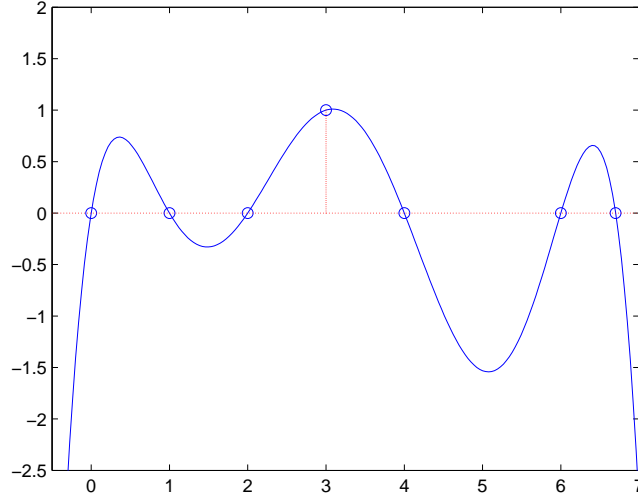


Figura A.4: Polinomio de apoyo para x_k

que el polinomio buscado es:

$$\begin{aligned}
 P_n(x) &= 0L_0(x) + \dots + 0L_{k-1}(x) + 1L_k(x) + 0L_{k+1}(x) + \dots + 0L_n(x) = L_k(x) \\
 L_k(x_i) &= 0 \quad i \neq k \\
 L_k(x_k) &= 1
 \end{aligned}
 \tag{A.13}$$

Un polinomio de grado n anulándose en todos los nodos de interpolación excepto x_k es:

$$R_n(x) = (x - x_0)(x - x_1) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_n)$$

Y su valor en el nodo x_k es:

$$R_n(x_k) = (x_k - x_0)(x_k - x_1) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)$$

Si se divide $R_n(x)$ entre el valor $R_n(x_k)$ se tendrá un polinomio que se anula en todos los nodos x_i menos en x_k , que equivale a 1, este polinomio será $L_k(x)$

$$L_k(x) = \frac{R_n(x)}{R_n(x_k)} = \frac{(x - x_0) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_n)}{(x_k - x_0) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)} \tag{A.14}$$

Con (A.14) se halla cada polinomio de *Lagrange* explícitamente como la combinación lineal entre los $L_i(x)$ y los valores de las abscisas.

Ejemplo

Considerar los puntos de interpolación $(0, 3)$, $(1, 5)$, $(2, 1)$, $(3, 0)$

Con la ecuación A.14 se hallan los polinomios de *Lagrange*:

$$\begin{aligned} L_0(x) &= \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} = \frac{(x-1)(x-2)(x-3)}{(-1)(-2)(-3)} \\ &= \frac{x^3 - 6x^2 + 11x - 6}{-6} = -\frac{1}{6}x^3 + x^2 - \frac{11}{6}x + 1 \end{aligned}$$

$$\begin{aligned} L_1(x) &= \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} = \frac{x(x-2)(x-3)}{(1)(-1)(-2)} \\ &= \frac{x^3 - 5x^2 + 6x}{2} = \frac{1}{2}x^3 - \frac{5}{2}x^2 + 3x \end{aligned}$$

$$\begin{aligned} L_2(x) &= \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} = \frac{x(x-1)(x-3)}{(2)(1)(-1)} \\ &= \frac{x^3 - 4x^2 + 3x}{-2} = -\frac{1}{2}x^3 + 2x^2 - \frac{3}{2}x \end{aligned}$$

$$\begin{aligned} L_3(x) &= \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} = \frac{x(x-1)(x-2)}{(3)(2)(1)} \\ &= \frac{x^3 - 3x^2 + 2x}{6} = \frac{1}{6}x^3 - \frac{1}{2}x^2 + \frac{1}{3}x \end{aligned}$$

$$\begin{aligned} P_3(x) &= y_0L_0(x) + y_1L_1(x) + y_2L_2(x) + y_3L_3 = 3L_0(x) + 5L_1(x) + 1L_2(x) \\ &= 3 \left[-\frac{1}{6}x^3 + x^2 - \frac{11}{6}x + 1 \right] + 5 \left[\frac{1}{2}x^3 - \frac{5}{2}x^2 + 3x \right] + \left[-\frac{1}{2}x^3 + 2x^2 - \frac{3}{2}x \right] \end{aligned}$$

Reduciendo:

$$P_n(x) = \frac{3}{2}x^3 - \frac{15}{2}x^2 + 8x + 3$$

Como se puede notar es el mismo polinomio que el hallado por interpolación polinomial.

Algoritmo: Método de Lagrange

```

clear all
clc

abscisas=[0 1 2 3];           %SOPORTE
ordenadas=[2 1 0 5];         %ORDEADAS
n=length(ordenadas);

for j=1:n                       % CREACION DE LA MATRIZ
                                De ABCISAS (SIN Xii)
    for i=1:n-1
        m(j,i)=abscisas(i);
        if i>=j
            m(j,i)=abscisas(i+1);
        end
    end
end

for j=1:n                       %CREACION DE MATRIZ CON
                                POLINOMIOS DE LAGRANGE
    for i=1:n-1
        a(i)=m(j,i);
    end
    Rn=poly(a);                 %POLINOMIO QUE NO SE ANULA EN Xk
    Rk=1;                       %VARIABLE PARA MULTIPLICAR
    for k=1:n-1
        Rk=Rk*(abscisas(j)-a(k)); %EVALUACION DE Xk EN Rn
    end
    nk=Rn/Rk;                   %POLINOMIO DE LAGRANGE EVALUADO EN Xk
    for h=1:n
        L(j,h)=nk(h);          %MATRIZ DE POLINOMIOS DE LAGRANGE
    end
end

Pn=0;
for j=1:n                       %CONSTRUCCION DE LA FORMULA DE LAGRANGE

```

```

for i=1:n
a(i)=L(j,i);
end
Pn=Pn+ordenadas(j)*a
end

x=linspace(0,3,100);
yo=polyval(Pn,x);
plot(x,yo)

```

%GRAFICA DEL POLINOMIO INTERPOLADOR

A.4. Fórmula de Newton

Un problema del método de *Lagrange* se debe a la naturaleza del modelado computacional en el momento de añadir un nuevo punto y rehacer los cálculos desde el principio. La fórmula de Newton soluciona este inconveniente expresando el polinomio de interpolación para los $k+1$ puntos en función del polinomio de los primeros k puntos. Otra gran ventaja es que el nuevo punto del soporte a añadir no necesita ser mayor que los k anteriores.

Se llama:

$$\begin{aligned}
P_0(x) &= y_0 \text{ polinomio cte. grado } 0 \\
P_1(x) & \text{ polinomio de interpolación de } (x_0, y_0)(x_1, y_1) \\
P_2(x) & \text{ polinomio de interpolación de } (x_0, y_0)(x_1, y_1)(x_2, y_2) \\
P_n(x) & \text{ polinomio de interpolación de } (x_0, y_0)(x_1, y_1) \dots (x_n, y_n)
\end{aligned}$$

Donde no necesariamente $x_0 < x_1 < x_2 \dots < x_n$ Para cada punto k se puede expresar el $P_k(x)$ en función de P_{k-1} en la forma:

$$P_k(x) = P_{k-1} + r_k(x) \tag{A.15}$$

Donde k va desde $1 \rightarrow n$ y r_k es un polinomio de grado $\leq k$

$$r_k(x) = c_k(x - x_0)(x - x_1) \dots (x - x_{k-1}) = c_k \prod_{i=0}^{k-1} (x - x_i) \tag{A.16}$$

r_k se anula en k abscisas x_0, x_1, \dots, x_{k-1} , c_k es una constante que se puede tratar recordando que $P_k(x)$ interpola x_k, y_k

$$y_k = P_k(x_n) = P_{k-1}(x_k) + r_k(x)$$

Por (A.17):

$$y_k = P_{k-1}(x_k) + c_k \prod_{i=0}^{k-1} (x - x_i)$$

Y de aquí se despeja c_k :

$$c_k = \frac{y_k - P_{k-1}(x_k)}{\prod_{i=0}^{k-1} (x_k - x_i)} \quad k = [1 \dots n] \quad (\text{A.17})$$

Las ecuaciones (A.15)(A.16) y (A.17) más la definición $P_0(x) = y_0$ permiten calcular el polinomio de interpolación de $n + 1$ puntos en función de un polinomio de interpolación con menos puntos.

Los polinomios intermedios pueden variar dependiendo del orden de ingreso de los datos pero el polinomio final $P_n(x)$ será siempre el mismo, este desarrollo se hace interactivo en la medida en que se añaden cada vez mas puntos de interpolación, que aprovechando los cálculos anteriores calcula el nuevo polinomio de interpolación.

Ejemplo

Considerar los siguientes puntos de interpolación

(0, 3), (2, 1), (3, 0), (1, 5)

Se han de ingresar los puntos en desorden y así se corroborará que no influye en la solución.

P_2 interpola (0, 3), (2, 1), (3, 0) que están alineados

$$c_3 = \frac{y_3 - P_2(x_3)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} = \frac{5 - 2}{(1)(-1)(-2)} = \frac{3}{2}$$

$$r_3 = c_3(x - x_0)(x - x_1)(x - x_2) = \frac{3}{2}x(x - 2)(x - 3)$$

$$P_3(x) = P_2(X) + r_3(x) = 3 - x + \frac{3}{2}x(x - 2)(x - 3)$$

Y se llega al mismo polinomio interpolador al que se ha llegado

$$P_n(x) = \frac{3}{2}x^3 - \frac{15}{2}x^2 + 8x + 3$$

Algoritmo: Método de Newton

```

clear all
close all
clc
abs=[0]
ord=[3]
n=length(abs)
o=1
p=ord(1)                                %EL POLINOMIO ES Y o PARA UN SOLO VALOR

while o~=0                               %MIENTRAS LA RESTA SEA DIFERENTE
                                         DE CERO

a=input('ingresa punto en X: ')        %INGRESO DE DATOS
b=input('ingresa punto en Y: ')
n=n+1                                    %CONTADOR DE PUNTOS INSERTADOS
ord(n)=b                                 %CONSTRUCCION DE VECTORES DE DATOS
abs(n)=a
for i=1:n-1
    o=(abs(n)-abs(i))*o                 %COMPROBACION DEL
                                         VALOR CERO
end
if o~=0                                  %COMPROBACION DE NO
                                         REPETICION DE ABCISA

d=1
for i=1:n-1
    d=(abs(n)-abs(i))*d                 %PRODUCTORIA DE ABCISAS
end
c=(ord(n)-polyval(p,abs(n)))/d         %CALCULO DE LA CONSTANTE C
for i=1:n-1

```

```

    e(i)=abs(i)                                %ELIMINACION DE LA
                                                ULTIMA ABCISA DEL VECTOR
end
rk=c*poly(e)                                  %MULTIPLICACION DE C
                                                POR LA PRODUCTORIA
p=[0,p]+rk                                     %AUMENTO DE UNA POSICION
                                                AL VECTOR PARA SUMAR
end
x=linspace(min(abs),max(abs),20);
yo=polyval(p,x);
plot(x,yo)
end

```

A.4.1. Diferencias divididas

Las ecuaciones (A.15), (A.16) y (A.17) son útiles, pero aún pueden mejorarse en relación al cálculo de los coeficientes c_k en (A.17), estos coeficientes reciben el nombre de *diferencias divididas*, pueden calcularse de una forma más sencilla que con el método anterior. Dado que el coeficiente c_k depende únicamente de los puntos $(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k)$ se usa la notación

$$c_k = y[x_0, x_1, \dots, x_k] \quad c_0 = y[x_0] = y_0$$

Que hace referencia explícita a los puntos de los que depende.

Partiendo entonces de $y[x_0] = y_0$ se calcula c_i por (A.17)

$$y[x_0, x_1] = c_1 = \frac{y_1 - P_0(x_1)}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0}$$

Si se denota por $y[x_1] = y_1$ (y en general $y[x_k] = y_k$) se tiene:

$$y[x_0, x_1] = \frac{y[x_1] - y[x_0]}{x_1 - x_0} \tag{A.18}$$

Por analogía con (A.18):

$$y[x_1, x_2] = \frac{y[x_2] - y[x_1]}{x_2 - x_1}$$

El siguiente coeficiente es más elaborado, pues precisa del cálculo previo de $P_1(x)$ y

algunas operaciones:

$$\begin{aligned}
P_1(x) &= P_0(x) + r_1(x) = y_0 + c_1(x - x_0) = y[x_0] + y[x_0, x_1](x - x_0) \\
y[x_0, x_1, x_2] &= c_2 = \frac{y_2 - P_1(x_2)}{(x_2 - x_0)(x_2 - x_1)} \\
y[x_0, x_1, x_2] &= \frac{y[x_2] - y[x_0] - y[x_0, x_1](x_2 - x_1)}{(x_2 - x_0)(x_2 - x_1)} \\
y[x_0, x_1, x_2] &= \frac{(y[x_2] - y[x_1]) + (y[x_1] - y[x_0]) - y[x_0, x_1](x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \\
y[x_0, x_1, x_2] &= \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0}
\end{aligned} \tag{A.19}$$

Escribiendo (A.19) para tres puntos arbitrarios

$$y[x_i, x_j, x_k] = \frac{y[x_i, x_k] - y[x_i, x_j]}{x_k - x_j}$$

Se puede expresar $c_3 \equiv y[x_0, x_1, x_2, x_3]$ operando con la relación (A.19) para obtener:

$$y[x_0, x_1, x_2, x_3] = \frac{y[x_1, x_2, x_3] - y[x_0, x_1, x_2]}{x_3 - x_0} \tag{A.20}$$

Empieza a verse la recurrencia que permite obtener una *diferencia dividida* correspondiente a $k + 1$ puntos en función de dos *diferencias divididas* de orden inferior, con k puntos:

$$y[x_0, x_1, \dots, x_k] = \frac{y[x_1, x_2, \dots, x_k] - y[x_0, x_1, x_{k-1}]}{x_k - x_0} \tag{A.21}$$

Teniendo en cuenta la analogía entre las fórmulas (A.18) a (A.21) resulta el esquema de cálculo del cuadro (A.4.1).

Como la tabla no requiere el cómputo de c_k mediante (A.16) (que requiere el cálculo de $P_{k-1}(x_k)$) se simplifica la construcción del polinomio de interpolación operando recurrentemente en (A.15) y utilizando (A.17):

$$\begin{aligned}
P_n(x) &= P_{n-1}(x) + r_n(x) \\
P_n(x) &= P_{n-2}(x) + r_{n-1}(x) + r_n(x) \\
P_n(x) &= P_{n-3}(x) + r_{n-2}(x) + r_{n-1}(x) + r_n(x) \\
&\dots = P_0(x) + r_1(x) + r_2(x) + \dots + r_n(x)
\end{aligned}$$

x_i	y_i	$y[x_i, x_j]$	$y[x_i, x_j, x_k]$	$y[x_i, x_j, x_k, x_h]$
x_0	$y_0 \searrow$	$\frac{y_1 - y_0}{x_1 - x_0} = y[x_0, x_1] \searrow$		
x_1	$y_1 \swarrow \searrow$	$\frac{y_2 - y_1}{x_2 - x_1} = y[x_1, x_2] \swarrow \searrow$	$y[x_0, x_1, x_2] \searrow$	$y[x_0, x_1, x_2, x_3]$
x_2	$y_2 \swarrow \searrow$	$\frac{y_3 - y_2}{x_3 - x_2} = y[x_2, x_3] \swarrow \searrow$	$y[x_1, x_2, x_3] \swarrow$	
x_3	$y_3 \swarrow$			
\vdots		\vdots	\vdots	\vdots

Cuadro A.1: Cálculo de las diferencias divididas

Sustituyendo cada $r_k(x)$ por su expresión según (A.17) y recordando que $P_0(x) = y_0$:

$$P_n(x) = y_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_n(x - x_0)(x - x_1) \dots (x - x_{n-1})$$

Y finalmente:

$$P_n(x) = y_0 + y[x_0, x_1](x - x_0) + y[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + y[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_{n-1}) \quad (\text{A.22})$$

Algoritmo: Diferencias Divididas

```
close all;
clear all;
clc

x=[0 1 2 3];           %ENTRADA DE DATOS A INTERPOLAR
y=[3 5 1 0];

xa=x;
ya=y;

d=zeros(length(y));   %FORMACION DE LAS
```

DIFERENCIAS DIVIDIDAS

```

d(:,1)=y';

for k=2:length(x)
    for j=1:length(x)+1-k
        d(j,k)=(d(j+1,k-1)-d(j,k-1))/(x(j+k-1)-x(j));
    end
end

for w=1:length(x)           %FORMACION DEL POLINOMIO
    ds=num2str(abs(d(1,w)));
    if w>1
        if x(w-1)<0
            sg1='+';
        else
            sg1='-';
        end
    end
    if d(1,w)<0
        sg2='-';
    else
        sg2='+';
    end
    if w==1
        acum=num2str(d(1,1));
    elseif w==2
        polact=['(x' sg1 num2str(abs(x(w-1))) ')'];
        actual=[ds '*' polact];
        acum=[acum sg2 actual];
    else
        polact=[polact '.*' '(x' sg1 num2str(abs(x(w-1))) ')'];
        actual=[ds '*' polact];
        acum=[acum sg2 actual];
    end
end

```

```

end

fprintf(' \n Valores de X y Y \n ');    %PRESENTACION DE RESULTADOS
disp(xa);
disp(ya);
fprintf('\n Polinomio : %s \n',acum);
x=input(' Ingrese X = ');
if x>max(xa)|x<min(xa)
    fprintf('\t Punto fuera de rango \n');
end
xinterp=x;
yinterp=eval(acum);
fprintf(' Y(%g) = %g \n',x,yinterp);

% GRAFICA DE LOS PUNTOS
fprintf(' Pulse una tecla para graficar \n');
pause
xg=linspace(min(xa),max(xa));
x=xg;yg=eval(acum);
plot(xg,yg,xa,ya,'.r',xinterp,yinterp,'or');
grid

```

A.5. Error en la interpolación Polinomial

No se hace referencia a los errores de aproximación y redondeos propios del cálculo sea cuál sea el método, se supone que el polinomio $P_n(x)$ se ha calculado exactamente. En muchos casos se aproxima una función (compleja y costosa) calculando algunos puntos con el soporte (cálculo complejo y costoso) $(x_i, f(x_i))$, finalmente se usa $P_n(x)$ para evaluar $f(x)$ en n abscisas \neq abscisas x_k . Para aplicaciones graficas, el diseñador, no usa una función $f(x)$ para describir una curva, hace “click” en los puntos por donde desea que pase la curva, por lo tanto se busca una función matemática desconocida, pero que su forma esta medianamente representada en la mente del diseñador.

$$\varepsilon(x) = f(x) - P_n(x)$$

El estudio del error de interpolación, aún cuando $f(x)$ se conoce de forma aproximada,

puede generar errores indeseables, que no necesariamente descalifican a la *interpolación polinómica* como un método inadecuado para modelado computacional.

A.5.1. Una expresión para el error

Suponiendo que la función desconocida $f(x)$ es $n+1$ veces diferenciable⁴ en un intervalo $[a, b]$ y contiene el soporte x_0, x_1, \dots, x_n .

Para $P_n(x)$ y $f(x)$ $\varepsilon(\bar{x}) = f(\bar{x}) - P_n(\bar{x})$ en un punto $\bar{x} \in [a, b]$ en función de la derivada $n+1$ de $f(x)$.

$$\varepsilon(\bar{x}) = w(\bar{x}) \frac{f^{n+1}(c\bar{x})}{(n+1)!} \quad \left\{ \text{si } |f^{n+1}(x)| \forall x \in (a, b) \right\} \quad (\text{A.23})$$

$$|\varepsilon(\bar{x})| \leq |w(\bar{x})| \frac{M_{n+1}}{(n+1)!} \quad \text{donde } M_{n+1} \geq |f^{n+1}(x)| \quad n+1 \forall (a, b) \quad (\text{A.24})$$

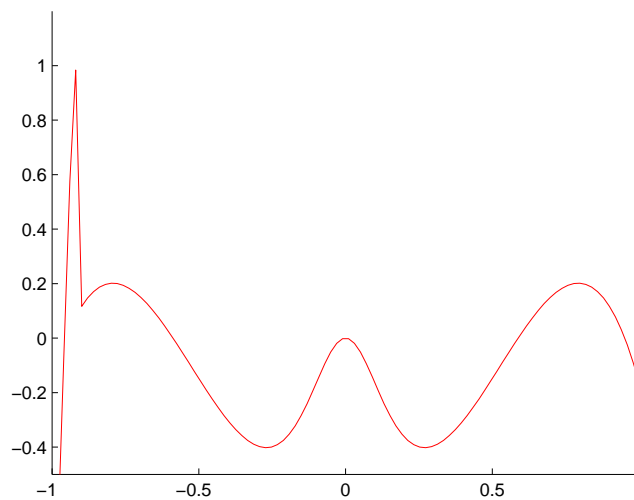


Figura A.5: Oscilación del error

Se puede notar que $w(x)$ cambiará de signo cada vez que alcance un nodo de interpolación y por lo tanto también lo hará $\varepsilon(x)$, esto significa que el polinomio de interpolación $P(x)$ va por encima de $F(x)$ en un intervalo y por debajo en el siguiente. Si quisiéramos fijar una tolerancia para el error, este quedaría en función del número

⁴Se ve la relación entre los puntos de interpolación y el orden diferenciable exigido a $f(x)$

de puntos de interpolación n .

A.5.2. El efecto Runge

El matemático Runge consideró la representación gráfica de la función $\frac{1}{x^2+1}$, parecida a e^{-x^2} , pero más sencilla de analizar, para probar que se interpola en el intervalo $[-5, 5]$ en $n + 1$ abscisas equiespaciadas, cuanto mayor es n , peor es la aproximación de $P_n(x)$ a $f(x)$ en las cercanías a los extremos del intervalo. Este resultado se muestra en la fig(A.5.2) conocido como efecto *Runge*.

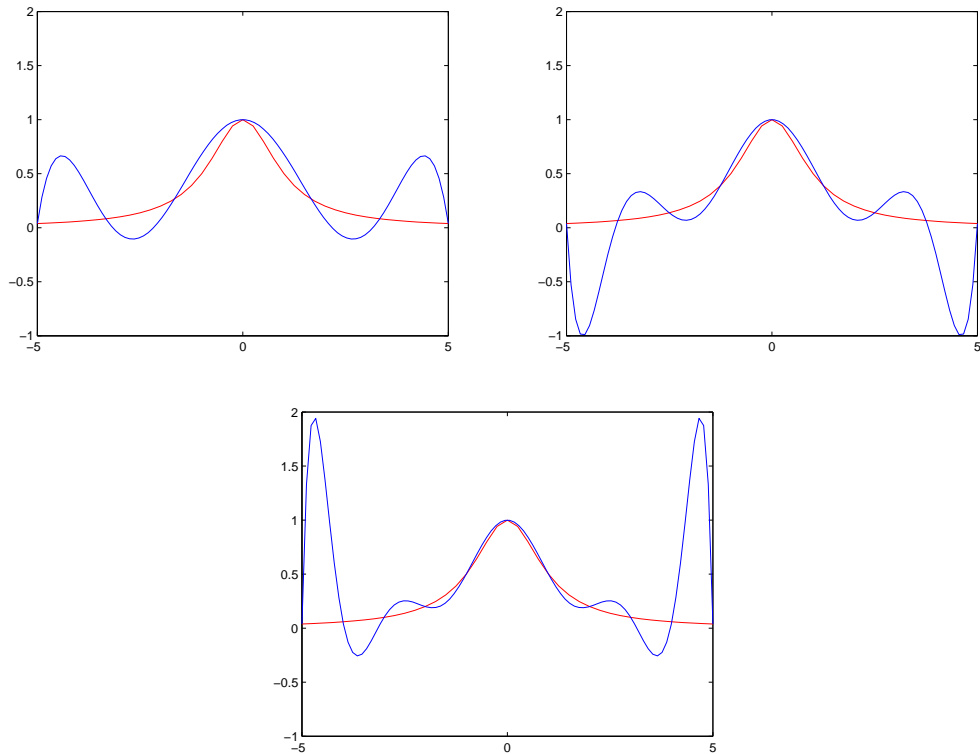


Figura A.6: Efecto Runge para $n = 5, 9, 13$

A.6. Interpolación de Tchebycheff

La expresión para el error dada por (A.23), y ya que no se conoce c_x se expande a:

$$|\varepsilon(x)| \leq |w(x)| \frac{M_{n+1}}{(n+1)!} = (x - x_0)(x - x_1) \dots (x - x_n) \frac{M_{n+1}}{(n+1)!} \quad (\text{A.25})$$

Haciendo una elección distinta del soporte (con las mismas abscisas) los polinomios de interpolación de la función serán diferentes y producirán un error diferente.

La fig(A.6) muestra dos polinomios de interpolación de grado 3 a la función \sqrt{x} , en soportes diferentes.

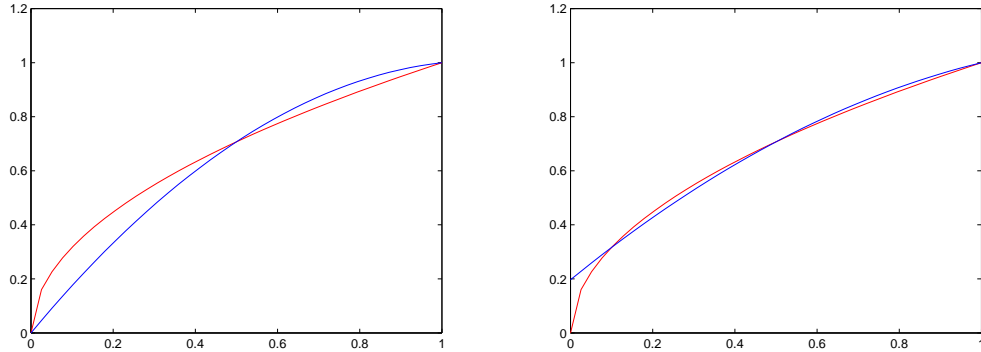


Figura A.7: Diferente soporte para \sqrt{x}

Después de observar los resultados, aparece inmediatamente la pregunta, sobre la elección ideal del soporte de $n + 1$ puntos para que la magnitud $\|w(x)\|_\infty = \|(x - x_0)(x - x_1) \dots (x - x_n)\|_\infty$ sea mínima. Esta situación fué solucionada por *P. Tchebycheff* en el intervalo $[-1, 1]$ (sin generar ninguna restricción) y la solución consiste en tomar como nodos de interpolación las raíces de un polinomio $T_{n+1}(x)$ llamado por obvias razones *polinomio de Tchebycheff* de grado $n + 1$.

A.6.1. Polinomios de Tchebycheff

La función $F_n = \cos(nt)$ puede expresarse como combinación de potencias de $\cos(t)$ usando relaciones trigonométricas.

$$\cos 2t = 2 \cos^2 t - 1$$

$$\cos 3t = 4 \cos^3 t - 3 \cos t$$

$$\cos 4t = 8 \cos^4 t - 8 \cos^2 t + 1$$

$$\cos 5t = 16 \cos^5 t - 20 \cos^3 t + 5 \cos t$$

$\cos(nt)$ es un polinomio de $\cos t$ llamado $T_n(x)$ reemplazando $x = \cos t$ se tienen los polinomios de *Tchebycheff* de primera especie.

$$\begin{aligned}
T_0(x) &= 1 \Rightarrow \cos(0t) = 1 \\
T_1(x) &= x \Rightarrow \cos(1t) = x \\
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
T_5(x) &= 16x^5 - 20x^3 + 5x \\
&\vdots
\end{aligned}$$

Estos polinomios de $T_n(x)$ se pueden obtener por una relación muy simple en función de los dos polinomios inmediatamente inferiores.

$$\begin{aligned}
\cos(n+1)t &= \cos(nt)\cos t - \sin(nt)\sin t \\
\cos(n-1)t &= \cos(nt)\cos t + \sin(nt)\sin t
\end{aligned}$$

Sumándolos:

$$\cos(n+1)t + \cos(n-1)t = 2\cos(nt)\cos t \quad (\text{A.26})$$

$$T_{n+1}(x) + T_{n-1}(x) = 2T_n(x)T_1(x) \Rightarrow T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad (\text{A.27})$$

Por el momento sólo se usarán los polinomios de *Tchebycheff* de primera especie para interpolación, aclarando que además son fundamentales en la teoría de aproximación de funciones.

A.6.2. La propiedad de la norma mínima

Tchebycheff planteó y resolvió el siguiente importante resultado: *Entre todos los polinomios mónicos de grado n , $\frac{1}{2^{n-1}}T_n(x)$ es el de menor norma.* Entendiendo la norma ∞ ó del máximo:

$$\left\| \frac{1}{2^{n-1}}T_n(x) \right\|_{\infty} \leq \|Q_n(x)\|_{\infty}$$

Para todo polinomio $Q_n(x) = x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0$ el signo es igual sólo se produce

$$Q_n(x) \equiv T_n(x).$$

Se observa que:

$$\|T_n(x)\|_{\infty} = \max_{x \in [0, \pi]} |\cos(nt)| = 1$$

Como consecuencia, el valor mínimo que alcanza la norma de un polinomio mónico de grado n es:

$$\min \|Q_n(x)\|_\infty = \frac{1}{2^{n-1}} \quad (\text{A.28})$$

A.6.3. Raíces de $T_n(x)$

Resolver $T_n(x) = 0$ es igual, en el intervalo $[-1, 1]$ a resolver:

$$\cos(nt) = 0 \quad (\text{A.29})$$

Cuyas raíces corresponden a $nt = \frac{2k+1}{2}\pi$ con $k = 0, \pm 1, \pm 2, \dots$ y obtener los (infinitos) valores de t verificando (A.29):

$$t_k = \frac{2k+1}{2n}\pi \quad k = 0, \pm 1, \pm 2, \dots$$

$$x_k = \cos\left(\frac{2k+1}{2n}\pi\right) \quad k = 0, 1, 2, \dots, n-1 \quad (\text{A.30})$$

De este modo $T_n(x)$ tiene n raíces diferentes en $[-1, 1]$, y son todas las raíces pues $T_n(x)$ es de grado n (A.30) entrega las raíces de derecha a izquierda en orden ascendente se reemplaza $j = n - k - 1$

$$x_j = \cos\left(\pi - \frac{2j+1}{2n}\pi\right) = -\cos\left(\frac{2j+1}{2n}\pi\right)$$

Estas raíces son simétricas respecto al origen.

A.6.4. Método de interpolación de Tchebycheff

El polinomio de apoyo del soporte con norma mínima es quien brinda una óptima interpolación esto se logra con el polinomio de *Tchebycheff* de grado $n + 1$ mónico.

$$w(x) = \frac{1}{2^n} T_{n+1}(x) \quad (\text{A.31})$$

$$x_k = \cos\left(\frac{2k+1}{2n+2}\pi\right) \quad k = 0, 1, 2, \dots, n \quad (\text{A.32})$$

Definido el soporte la interpolación se puede hacer por cualquiera de los métodos ya vistos: resolución del sistema de ecuaciones, método de *Lagrange*, ó formula de *Newton*, considerada la base formulada por los $n + 1$ primeros polinomios de *Tchebycheff*⁵.

$$\mathcal{B} = \left(\frac{1}{\sqrt{2}}, T_1(x), T_2(x), \dots, T_n(x) \right) \quad (\text{A.33})$$

Y el polinomio de interpolación será:

$$P_n(x) = \left(\frac{a_0}{\sqrt{2}} + a_1 T_1(x) + a_2 T_2(x) + \dots + a_n T_n(x) \right) \quad (\text{A.34})$$

Se realiza un desarrollo matemático partiendo de:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & T_1(x_0) & T_2(x_0) & \cdots & T_n(x_0) \\ \frac{1}{\sqrt{2}} & T_1(x_1) & T_2(x_1) & \cdots & T_n(x_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\sqrt{2}} & T_1(x_n) & T_2(x_n) & \cdots & T_n(x_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \quad (\text{A.35})$$

Quedando resuelto el sistema (A.35)

$$\begin{aligned} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} &= B^{-1} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} = \frac{2}{n+1} B^t \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} = \\ \frac{2}{n+1} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ T_1(x_0) & T_1(x_1) & \cdots & T_1(x_n) \\ \vdots & \vdots & \ddots & \vdots \\ T_n(x_0) & T_n(x_1) & \cdots & T_n(x_n) \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \end{aligned} \quad (\text{A.36})$$

En la fig(A.6.4) se muestra un polinomio grado 4 con el soporte calculado por Tchebycheff a modo comparativo el polinomio de la fig(A.6.4) está calculado con un soporte de paso constante.

⁵ $T_0(x) = 1$, se sustituye por $\frac{1}{\sqrt{2}}$ para simplificar formulas. solo alteraría a a_0 que también es constante

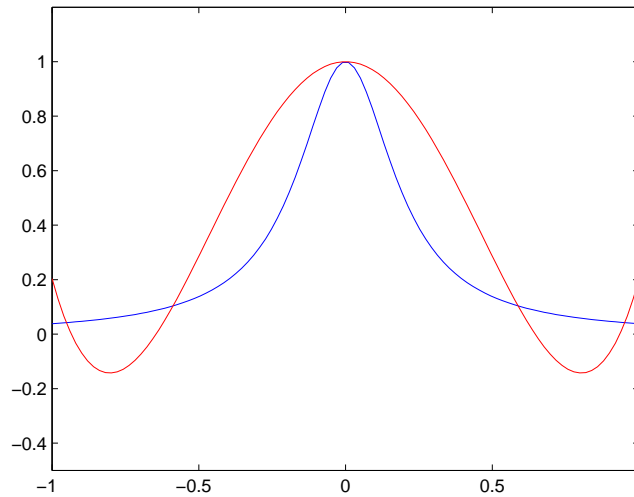


Figura A.8: Soporte por Tchebycheff

En la fig(A.6.4) se muestran los polinomios de interpolación de $f(x) = \frac{1}{25^2+1}$ en $[-5, 5]$ para $n = 8, 12, 16$.

Algoritmo: Interpolación de Tchebycheff

```

clear all
close all
clc

xo=linspace(-1,1,100);
s=(25*xo.^2+1).^-1;           %FUNCIÓN COSTOSA
plot(xo,s)
axis([-1,1,-0.5,1.2])
hold on

ptos=5;                       %USAR PARA NUMERO
                               DE PUNTOS: 5,9,13 0 17

for k=1:ptos

```

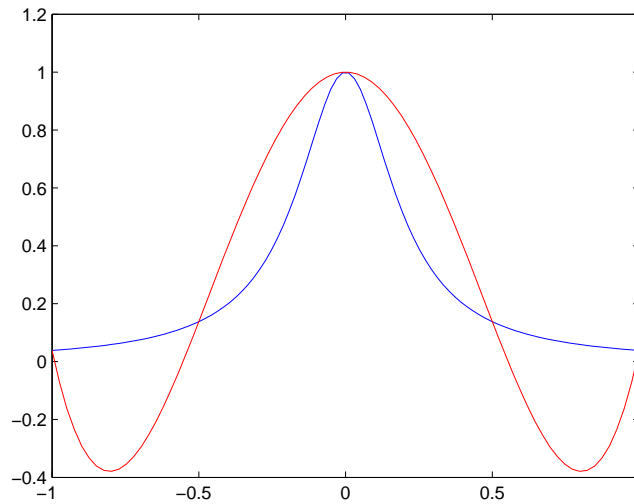


Figura A.9: Soporte equiespaciado

```

s(k)=-cos((2*(k-1)+1)*pi/(2*(ptos-1)+2));    %SOPORTE DE INTERPOLACION
y(k)=1/(25*s(k)^2+1);                        %FUNCIÓN A APROXIMAR
end

B=zeros(ptos);
B(1,ptos)=1;                                %PRIMER POLINOMIO DE TCHEBYCHEFF
B(2,ptos-1)=1;                               %SEGUNDO POLINOMIO DE TCHEBYCHEFF

for h=1:ptos
    c(h)=B(1,h);                              %CONSTRUCCIÓN DEL POLINOMIO COMO FILA
    b(h)=B(2,h);
    d(h)=B(3,h);
end

for i=3:ptos
    for e=1:ptos-1
        d(e)=2*b(e+1);                        %SE MULTIPLICA POR "2x", SE REUBICA
                                                AUMENTA Y MULTIPLICA POR 2
    end
    a=d-c;                                     %SEGUN LA FORMULA DE RECURRENCIA FUNCION DE

```

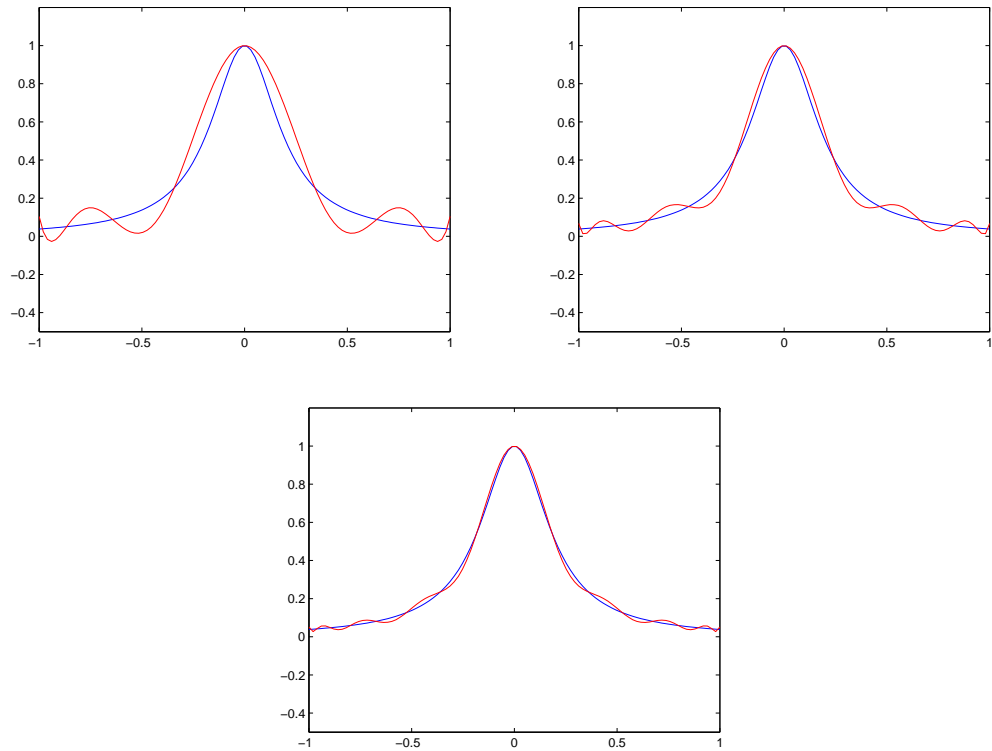


Figura A.10: Soporte Tchebycheff para $n = 8, 12, 16$

```

                                LOS 2 ANTERIORES

for v=1:ptos
    B(i,v)=a(v);                %INGRESO DEL NUEVO VECTOR EN LA MATRIZ DE
                                POLINOMIOS DE TCHEBYCHEFF
end
c=b;
b=a;
end

B(1,ptos)=2^(-0.5);           %CAMBIO DEL PRIMER POLINOMIO=1
                                POR 1/RAIZ DE(2)

m=zeros(ptos);

for w=1:ptos
    m(1,w)=2^(-0.5);           %LLENADO DE LA PRIMERA FILA DE

```

```

                                LA MATRIZ TRANSFORMADA

end

for x=2:ptos
    for z=1:ptos
        q(z)=B(x,z);           %LLAMANDO LA FILA DE POLINOMIOS
                                DE TCHEBYCHEFF
    end
    for d=1:ptos
        m(x,d)=polyval(q,s(d)); %SE EVALUA EL SOPORTE DEL POLINOMIO
                                %SE LLENA LA MATRIZ TRANSFORMADA
    end
end

A=2/ptos*m*y';                %CALCULO DE LOS COEFICIENTES
                                DE INTERPOLACIÓN

p=0;                           %INICIANDO EL POLINOMIO INTERPOLADOR
                                EN CERO PARA IR SUMANDO

for J=1:ptos
    for I=1:ptos
        C(I)=B(J,I);           %LLAMANDO LOS POLINOMIOS DE TCHEBYCHEFF
    end
    p=p+C.*A(J);               %CONSTRUCCIÓN DE LA CURVA
end

xo=linspace(-1,1,100);
yo=polyval(p,xo);
plot(xo,yo,'r')

```

A.7. Problemas con interpolación Polinomial

Las secciones anteriores han tratado diferentes métodos de cálculo del polinomio de interpolación, lo cual ha permitido observar algunas de las importantes carencias del propio proceso interpolatorio. Estas y algunas otras se pueden resumir en los siguientes puntos:

1. Costo computacional elevado, sobre todo si se requiere una interpolación interactiva añadiendo sucesivamente puntos. El método de Newton soluciona parcialmente ese problema.
2. Necesidad de operar con polinomios de grado elevado. En efecto, el grado del polinomio de interpolación es una unidad menos que el número de puntos, de modo que por ejemplo para 50 puntos se obtiene un polinomio de grado 49. Estos polinomios son complicados de manejar y su evaluación plantea problemas numéricos, tanto de costo computacional como de estabilidad. Por ejemplo, tabular un polinomio de grado 49 (para dibujarlo) en 1000 valores de la abscisa requiere en el mejor de los casos 100000 operaciones, lo cual impide su dibujo en tiempo real.
3. Error de gran magnitud entre sucesivos puntos de interpolación. El polinomio no sigue la forma dictada por los puntos de interpolación.
4. Oscilación del polinomio. Problema estrechamente relacionado con el anterior: el polinomio de interpolación oscila a ambos lados de la curva teórica que forman los puntos de interpolación. Esto lo hace altamente inadecuado para el modelado computacional.
5. Carácter global del proceso interpolatorio: si se modifica un solo punto de interpolación, el nuevo, polinomio difiere del original en todo el intervalo de definición (excepto en los propios puntos de interpolación no modificados). Esto hace inviable *retocar* localmente una curva interpoladora. Dejando de lado por el momento los tres primeros problemas que, al menos parcialmente, pueden aliviarse operando con polinomios de grado bajo; la cuestión de la oscilación, es probablemente la principal desventaja en relación al modelado computacional.

Planteando el problema de la elección de espacios vectoriales compuesto por funciones distintas de los polinomios, surge en principio la idea de utilizar funciones más complicadas que los simples polinomios; sin embargo esto no necesariamente deberá simplificar las cosas, más bien se trata de exponer la existencia de espacios de funciones que se comportan mucho mejor que los polinomios con respecto al problema de interpolación.

A.7.1. Minimización de la oscilación: Splines cúbicos

Un conjunto de puntos $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ con $x_0 < x_1 < \dots < x_n$, como se mencionó, existen infinitas curvas pasando por estos puntos, una de ellas es el polinomio de interpolación, que ya se ha venido asimilando y otro la poligonal que une los puntos

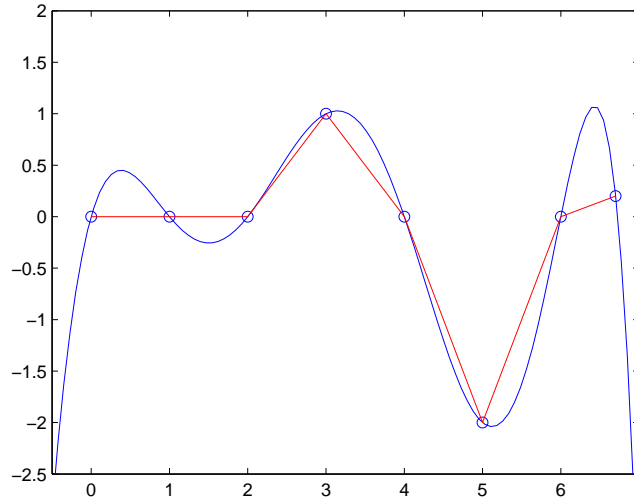


Figura A.11: La poligonal que une los puntos

La poligonal es la que menos oscila pero es inútil como curva interpoladora por la discontinuidad de la primera derivada. Se toma la poligonal en cada intervalo $[x_i, x_{i+1}]$ y estudia cuanto se aparta una curva $y = f(x)$ (oscilación de y) respecto a la poligonal. Considerando que esta recta es el polinomio de interpolación de grado 1 a la curva $f(x)$. La ecuación (A.23) permite expresar el error entre $f(x)$ y la recta:

$$\varepsilon(x) = f(x) - r(x) = w(x) \frac{f''(c_x)}{2!} = \frac{(x - x_i)(x - x_{i+1})f''(c_x)}{2!}$$

Se puede interpretar que el valor $f''(x)$ controla el $\varepsilon(x)$ entre $f(x)$ y la recta entre cada dos puntos, $f''(x)$ esto se puede interpretar como una medida de la oscilación de $f(x)$ que debe ser al menos dos veces diferenciable (por lo menos cúbica). De momento no se prueba la existencia de una función $s(x)$ hecha por trozos de polinomios de tercer grado, que deben estar conectados de modo que $s(x)$ en conjunto sea de clase C^2 , con las condiciones $s''(x_0) = s''(x_n) = 0$, pues probar la existencia de esta función es una labor que se llevará a cabo en el capítulo 3.

A.8. Interpolación de Hermite

En caso de interpolar puntos, y asignar valores al vector tangente de la curva en cada punto de interpolación se habla de un método llamado interpolación de Hermite que se aplica para solucionar estas situaciones. Teniendo: $(x_0, y_0), \dots, (x_n, y_n)$ y las pendientes y'_0, y'_1, \dots, y'_n asociadas a cada punto, se busca un polinomio:

$$\begin{aligned} P(x_i) &= y_i \\ P'(x) &= y'_i \end{aligned} \quad i = 0, 1, \dots, n \quad (\text{A.37})$$

(A.37) impone $2n + 2$ condiciones y será suficiente tratar con un polinomio de grado $2n + 1$, ahora se lleva a cabo una metodología similar a la interpolación polinomial. Se escribe el polinomio en su respectiva base de potencias:

$$\begin{aligned} P_{2n+1}(x) &= a_{2n+1}x^{2n+1} + a_{2n}x^{2n} + \dots + a_1x + a_0 \\ P'_{2n+1}(x) &= (2n + 1)a_{2n+1}x^{2n} + 2na_{2n}x^{2n-1} + \dots + 2a_2x + a_1 \end{aligned} \quad (\text{A.38})$$

Con las condiciones (A.37)

$$\begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^{2n+1} \\ 1 & x_1 & x_1^2 & x_1^3 & \dots & x_1^{2n+1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & \dots & x_n^{2n+1} \\ 0 & 1 & 2x_0 & 3x_0^2 & \dots & (2n + 1)x_0^{2n} \\ 0 & 1 & 2x_1 & 3x_1^2 & \dots & (2n + 1)x_1^{2n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 2x_n & 3x_n^2 & \dots & (2n + 1)x_n^{2n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \\ \vdots \\ a_{2n+1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \\ y'_0 \\ \vdots \\ y'_n \end{bmatrix} \quad (\text{A.39})$$

Aquí como en interpolación polinomial, la matriz es dependiente de las abscisas e independiente de las ordenadas y_i así como de las derivadas y'_i , además se puede afirmar que el determinante será siempre diferente de 0.

Algoritmo: Interpolación de Hermite

```
x=[0 1 2]      %ABSCISAS
y=[1; 2; -1]  %ORDENADAS
d=[-1; 0 ; -1] %DERIVADAS
n=length(x)
for i=1:n      %ADICION DE LAS DERIVADAS AL VECTOR ABSCISAS
    y(i+n)=d(i)
end

m=zeros(2*n)
for I=1:n     %SOLO LAS PRIMERAS n FILAS
    for j=1:2*n
        m(I,j)=x(I)^(j-1) %COMO SI FUESE UN SISTEMA DE ECUACIONES
    end
end

for k=(n+1):2*n
    for l=2:2*n      %LLENADO DESDE LA SEGUNDA COLUMNA
        m(k,l)=(l-1)*x(k-n)^(l-2) %MULTIPLICANDO POR POTENCIA
                                %DEL ELEMENTO SIGUIENTE
    end
end

a=inv(m)*y      %SOLUCIONANDO EL SISTEMA
a=a(2*n:-1:1)  %INVIRTIENDO EL POINOMIO
p=a'
```



```
x=linspace(0,2,100);
yo=polyval(p,x);
plot(x,yo)
```

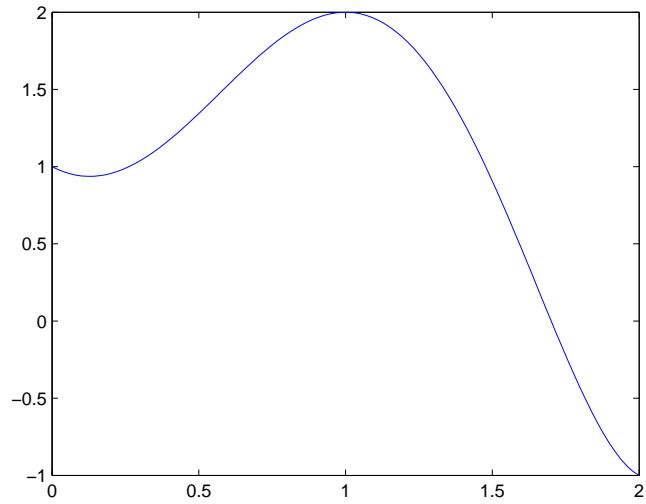


Figura A.12: Polinomio obtenido por el método de Hermite

A.8.1. Elección del soporte

Se verán tres métodos para precalcular un soporte de interpolación, el primero es independiente de los puntos de interpolación y los otros dos se basan en la geometría de sus puntos a interpolar. En general no existe ninguna restricción en considerar siempre $t_i \in [0, 1]$.

1. **Soporte uniforme** Simplemente es una partición regular (paso constante) del intervalo $[0, 1]$

$$t_i = \frac{i}{n} \quad i = 0, 1, \dots, n$$

2. **Soporte basado en longitud de cuerda**

El espaciado entre nodos es proporcional a las distancias entre los puntos de interpolación.

$$t_i - t_{i-1} = h \|P - i - P_{i-1}\| \quad h = cte \text{ y } P_k \equiv (x_k, y_k)$$

Haciendo $t_0 = 0$ tenemos:

$$\begin{aligned}t_1 &= h\|P_1 - P_0\| \\t_2 &= t_1 + h\|P_2 - P_1\| \\t_3 &= t_2 + h\|P_3 - P_2\| \\&\vdots \\t_i &= t_{i-1} + h\|P_i - P_{i-1}\|\end{aligned}$$

El último punto

$$\begin{aligned}t_n &= 1 \\1 &= t_n \\1 &= t_{n-1} + h\|P_n - P_{n-1}\| \\&\vdots \\1 &= t_0 + h(\|P_1 - P_0\| + \dots + \|P_n - P_{n-1}\|) \\1 &= hd\end{aligned}$$

La constante $h = \frac{1}{d}$, y d es la longitud total de la poligonal que une todos los puntos de interpolación

$$d = \|P_1 - P_0\| + \|P_2 - P_1\| + \dots + \|P_n - P_{n-1}\|$$

Y finalmente se calcula el soporte:

$$t_i = t_{i-1} + \frac{\|P_i - P_{i-1}\|}{d} \quad i = 1, 2, \dots, n \quad t_0 = 0$$

3. Soporte centrípeta

El cuadrado del espaciado entre los nodos es proporcional a las distancias entre los puntos de interpolación.

$$(t_i - t_{i-1})^2 = h\|P_i - P_{i-1}\|$$

La idea de esta elección se basa en que si el parámetro t es el tiempo, el movimiento de un punto sobre la curva interpoladora optimiza la fuerza centrípeta de modo que la trayectoria tienda a no alejarse mucho de la recta. Despejamos y ejecutamos

el procedimiento anterior:

$$d = \sqrt{\|P_1 - P_0\|} + \dots + \sqrt{\|P_n - P_{n+1}\|}$$

$$t_0 = 0$$

$$t_i = t_{i-1} + \frac{\sqrt{\|P_1 - P_0\|}}{d}$$

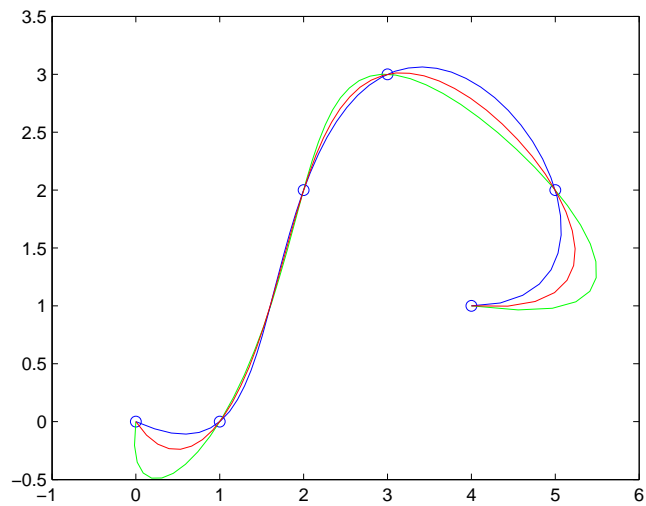


Figura A.13: Tipos de soporte

Algoritmo: Soporte de Interpolación

```
close all
```

```
clear all
```

```
clc
```

```
x=[0 1 2 3 5 4]
```

```
y=[0 0 2 3 2 1]
```

```
plot(x,y,'o')
```

```
%GRAFICANDO LOS PUNTOS
```

```
hold on
```

```

n=length(x)

d2=0; %CONTADORES EN CERO d2, d3
d3=0;
t2=0; %PRIMEROS DATOS DE LOS SOPORTES EN CERO
t3=0;

for i=2:n
d2=d2+((x(i)-x(i-1))^2+(y(i)-y(i-1))^2)^0.5; %LONGITUD DE LA POLIGONAL
d3=d3+((x(i)-x(i-1))^2+(y(i)-y(i-1))^2)^0.25; %SUMATORIA
end

for i=1:n-1 %SOPORTE EQUIESPACIADO
t1(i+1)=i/(n-1)
t2(i+1)=t2(i)+(((x(i+1)-x(i))^2+(y(i+1)-y(i))^2)^0.5)/d2
t3(i+1)=t3(i)+(((x(i+1)-x(i))^2+(y(i+1)-y(i))^2)^0.25)/d3
end

%----- t1 -----
for j=1:n
for i=1:n
m(j,i)=t1(j)^(i-1);
end
end

a=inv(m)*x';
b=inv(m)*y';
z=a';v=length(z);X=z(v:-1:1); %POLINOMIO X(t)
z=b';v=length(z);Y=z(v:-1:1); %POLINOMIO Y(t)
t=linspace(0,1,50);
xo=polyval(X,t);
yo=polyval(Y,t);
plot(xo,yo,'g')

%----- t2 -----

```

```

abs=t2;
for j=1:n
    for i=1:n
        m(j,i)=t2(j)^(i-1);
    end
end

a=inv(m)*x'
b=inv(m)*y'
z=a';v=length(z);X=z(v:-1:1);
z=b';v=length(z);Y=z(v:-1:1);
xo=polyval(X,t);
yo=polyval(Y,t);
plot(xo,yo,'b')

%----- t3 -----
abs=t3;
for j=1:n
    for i=1:n
        m(j,i)=abs(j)^(i-1);
    end
end

a=inv(m)*x'
b=inv(m)*y'
z=a';v=length(z);X=z(v:-1:1);
z=b';v=length(z);Y=z(v:-1:1);
xo=polyval(X,t);
yo=polyval(Y,t);
plot(xo,yo,'r')

```

Apéndice B

Curvas de Bézier

Las curvas son sencillas de expresar matemáticamente, en cambio es muy complicado modelarlas mediante polígonos. Ya se vió que los polígonos no son, por muchas razones, las mejores herramientas para solucionar el problema de interpolación, también se vió que la mejor aproximación y de menor oscilación en cierto sentido es el spline cúbico.

Sin poder probar su existencia, se sabe que el spline cúbico es una curva conformada por trozos de polinomios de tercer grado, de forma tal que están unidos por el punto en donde la derivada segunda coincide para los dos fragmentos de curva, haciéndola continua para así definir una curva en C^2 .

Se denomina CURVA DE BÉZIER al sistema desarrollado hacia los años 60, para el trazado de dibujos técnicos, en el diseño aeronáutico y de automóviles. Su denominación es en honor a Pierre Bézier, quien ideó un método de descripción matemática de las curvas que se comenzó a utilizar con éxito en los programas de CAD.

En 1959 De Casteljaou ingeniero de Citroën, idea una formulación matemática para diseñar las carrocerías de los automoviles, posteriormente en 1959 por el ingeniero de origen francés Pierre Bézier de Renault llega a las mismas conclusiones, partiendo de otro desarrollo matemático para después publicar sus resultados. Las curvas fueron desarrolladas por Paul de Casteljaou usando el algoritmo que lleva su nombre. Se trata de un método numéricamente estable para evaluar las curvas de Bézier.

Una curva de Bézier es la consideración matemática de una base especial en el espacio de un polinomio, que permite una excelente interpretación geométrica de los coeficientes de este polinomio cuando se expresan en función de dicha base. Esta base es la denominada base *Bernstein*, formulada en 1952 por el matemático Ucraniano Sergei Natanovich,



Figura B.1: Pierre Bézier. Fuente: www.davidsalomon.name/CaS/CaS.html

quien formando una combinación convexa la emplea en el contexto de la aproximación uniforme a funciones continuas. La idea de definir geoméricamente una determinada forma no es demasiado compleja, como se sabe, un punto del plano puede definirse por coordenadas, por ejemplo, un punto A tiene unas coordenadas (x_1, y_1) y a un punto B le corresponde (x_2, y_2) . Para trazar una recta entre ambos basta con conocer su posición.

B.1. Polinomios de Bernstein

Considerando los polinomios para n fijo

$$B_k(x) = \binom{n}{k} x^k (1-x)^{n-k} \quad (\text{B.1})$$

B.1.1. Definición polinomios de Bernstein

Sea $f(0, 1)$ R una función continua, el polinomio n -ésimo de Bernstein $B(n, f, k)$ de la función f se define como:

$$B(n, f, x) = B_n(f, x) = \sum_{k=0}^n f \binom{n}{k} x^k (1-x)^{n-k} \quad \text{con } 0 \leq k \leq n \quad (\text{B.2})$$

Donde:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (\text{B.3})$$

Son los coeficientes binómicos

Una construcción de los polinomios de Bernstein para la función de $f(x) = 1$ sobre el intervalo $(0, 1)$ se hace de la siguiente forma:

Obsérvese que:

$$1^n = 1$$

Para todo n que pertenece a los N por lo tanto x pertenece a los R .

Se cumple que:

$$1 = (x + 1 - x)^n = \sum_{k=0}^n \binom{n}{k} x^k (1-x)^{n-k} \quad (\text{B.4})$$

Lo cual nos dice que el n -ésimo término de la función constante $f(x) = 1$ coincide con f es decir equivale a 1. Generalmente sólo se definen para $x \in (0, 1)$ que es donde se manifiestan sus propiedades.

Ejemplos

Para $n = 1$

$$B_0(x) = 1 - x$$

$$B_1(x) = x$$

Para $n = 2$

$$B_0(x) = (1 - x)^2$$

$$B_1(x) = 2x(1 - x)$$

$$B_2(x) = x^2$$

Para $n = 3$

$$B_0(x) = (1 - x)^3$$

$$B_1(x) = 3x(1 - x)^2$$

$$B_2(x) = 3x^2(1 - x)$$

$$B_3(x) = x^3$$

Código en MATLAB

```
n=3          % QUE GRADO DE BERNSTEIN SE TRABAJARA
t(n+1)=1     % INICIA CON EL TRIANGULO DE PASCAL
             % PARA IR CREANDO C/SERIE DE COEFICIENTES
             % BINOMICOS SE USA LA ANTERIOR FILA

base=t pascal=t
for j=2:n+1
for i=1:n
t(i)=base(i)+base(i+1)
pascal(j,i)=t(i)
end
pascal(j,n+1)=1          % EL ULTIMO ES 1
base=t
end                      % LAS POTENCIAS IMPARES
                          % SON NEGATIVAS ->(1-X)

for i=n:-2:1
pascal(:,i)=pascal(:,i)*-1
end                      % EL PRIMER POLINOMIO DE
                          % BERNSTEIN ES EL ULTIMO
                          % POLINOMIO DE PASCAL CON
                          % SIGNO INCLUIDO

bern=pascal(n+1,:)
matriz=bern             % SE LLENA LA MATRIZ CON
                          % LOS POLINOMIOS DE BERNSTEIN

for j=2:n
bern=pascal(n+2-j,:).*t(j)
apoyo=bern
```

```

for i=1:n-j+2
ber(i)=apoyo(i+j-1)           % REUBICA TODOS LOS VALORES A LA
                               IZQUIERDA n "j-1" VECES-> POR X^n
matriz(j,i)=ber(i)
end
clear ber
end                               % EL ULTIMO POLINOMIO ES UN
                                   MONOMIO A LA MAXIMA POTENCIA

matriz(n+1,1)=1
x=linspace(0,1,20)
for i=1:n+1
fx=matriz(i,:)
y=polyval(fx,x)
plot(x,y)
hold on
end

```

B.1.2. Propiedades de los polinomios de Bernstein

A continuación se listan algunas de las propiedades, siempre referidas al intervalo $(0, 1)$:

1. $B_k(x) \geq 0 \quad \forall x, \forall n$
2. $B_k(x)$ posee exactamente un máximo y se produce en $x = \frac{k}{n}$
3. $\sum_{k=0}^n B_k = 1 \quad \forall n$ Esto resulta inmediatamente del desarrollo de *Newton*:

$$\sum_{k=0}^n B_k = \left(\begin{matrix} n \\ k \end{matrix} \right) x^k (1-x)^{n-k} = [x + (1-x)]^n = 1$$

4. $B_0(0) = 1, B_k(0) = 0, \quad \text{para } 1 \leq k \leq n$
 $B_n(0) = 1, B_k(1) = 0, \quad \text{para } 1 \leq k \leq n$

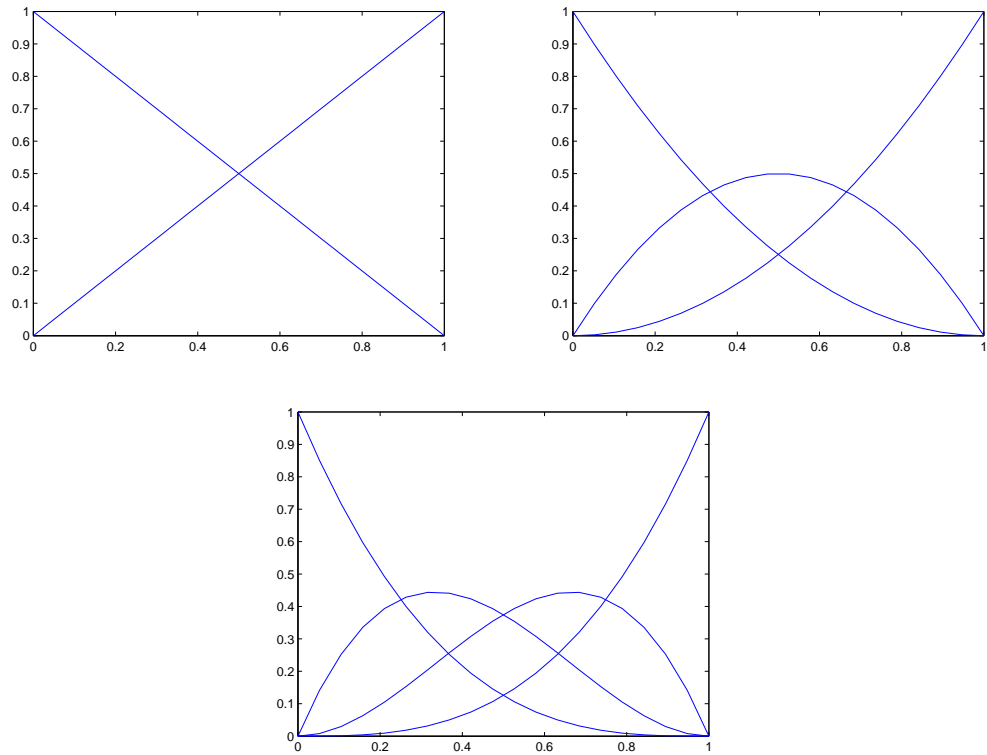


Figura B.2: Polinomios de Bernstein

5. Recurrencia: si añadimos un superíndice a $B_k(x)$ indicando el grado:

$$B_k^n(x) = (1-x)B_k^{n-1}(x) + xB_{k-1}^{n-1}(x) \quad n \geq 1 \quad 1 \leq k \leq n$$

Lo que permite evaluar fácilmente los polinomios de *Bernstein* sin necesidad de usar la ecuación (B.1)

6. Derivada:

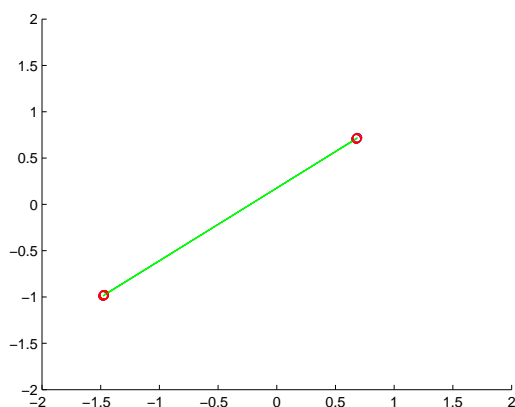
$$[B_k^n(x)]' = n[B_{k-1}^{n-1}(x) - B_k^{n-1}(x)]$$

Las propiedades 1 y 3 indican que para cada x , $B_k(x)$ constituye una partición de la unidad

B.2. Curvas de Bézier simples

Cuando un polinomio arbitrario de grado menor ó igual que n se expresa como una combinación lineal de los polinomios de *Bernstein*, suele denominarse curva de *Bézier*

Figura B.3: Curva de Bézier grado 1



simple ó de un tramo.

Utilizando la representación en forma paramétrica

$$C(t) = \sum_{k=0}^n P_k B_k(x) \quad 0 \leq t \leq 1 \quad (\text{B.5})$$

Donde:

$$P_k = (x_k, y_k) \quad \text{si la curva es de } R^2$$

$$P_k = (x_k, y_k, z_k) \quad \text{si la curva es de } R^3$$

Los coeficientes P_k respecto a la base *Bernstein* serán denominados puntos de control de la curva de *Bézier*.

Curvas lineales (Grado 1)

Para $n=1$ sólo se tiene dos puntos de control, se caracterizan por ser líneas rectas, se puede recorrer la curva con un parámetro $t \in (0, 1)$ que recorre la curva de P_1 a P_2

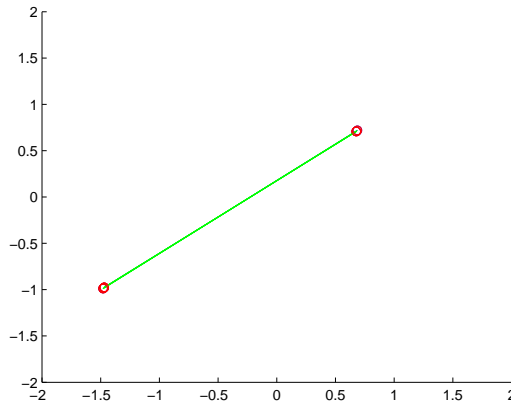


Figura B.4: Curva de Bézier grado 2

$$\begin{aligned}
 C(t) &= \sum_{k=0}^n P_k B_k(x) = P_0 B_0(t) + P_1 B_1(t) = (1-t)P_0 + tP_1 \\
 C(0) &= P_0 \\
 C(1) &= P_1
 \end{aligned}
 \tag{B.6}$$

Curva de Bézier cuadrática (Grado 2)

Se construyen dos curvas lineales entre P_0 y P_1 y P_1 y P_2 . Se construye una tercera curva lineal entre las dos anteriores, el parámetro t que recorre esta tercera recta forma la curva de *Bézier*.

Para $n=2$

$$\begin{aligned}
 C(0) &= P_0 B_0(t) + P_1 B_1(t) + P_2 B_2(t) \\
 C(1) &= (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2 \\
 C(2) &= P_1
 \end{aligned}
 \tag{B.7}$$

Donde $C(t)$ posee la siguiente propiedad geométrica

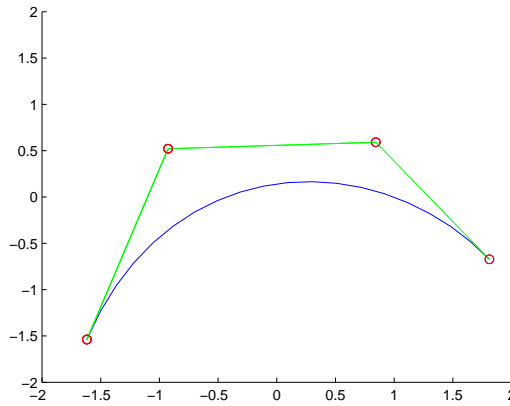


Figura B.5: Curva de Bézier grado 3

$$\begin{aligned}
 C'(t) &= -2(1-t)P_0 + 2(1-2t)P_1 + 2tP_2 \\
 C'(0) &= 2(P_1 - P_0) \\
 C'(1) &= 2(P_2 - P_1)
 \end{aligned}
 \tag{B.8}$$

$C'(0)$ y $C'(1)$ son vectores tangentes a la curva en $C(t)$ en 0 y 1

Se puede observar como los tres puntos determinan de manera bastante precisa la geometría de la curva.

Curvas de Bézier cúbicas (Grado 3)

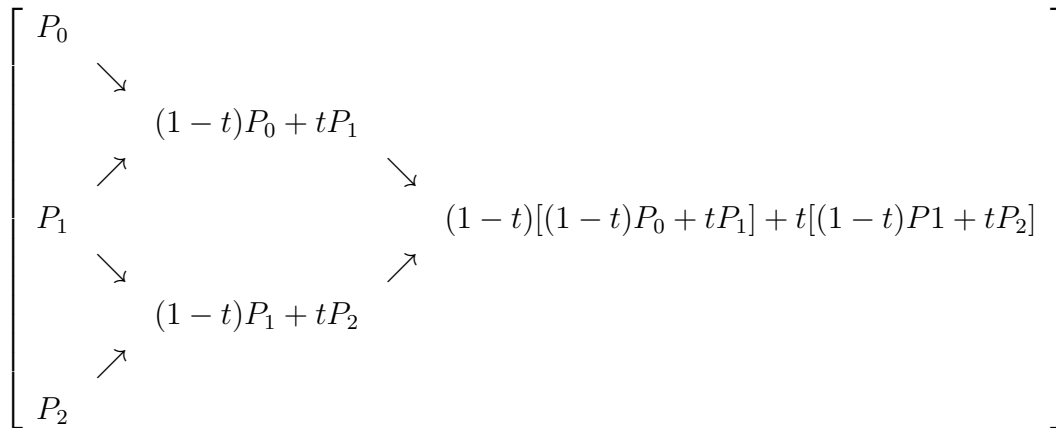
Para $n=3$ se tienen cuatro puntos de control P_0 , P_1 , P_2 y P_3 estas curvas son las más utilizadas, se aplica el mismo principio de inducción partiendo inicialmente de tres curvas de *Bézier* lineales, de esta manera se pueden generar curvas de cualquier grado.

$$\begin{aligned}
 C(t) &= P_0B_0(t) + P_1B_1(t) + P_2B_2(t) + P_3B_3(t) \\
 &= (1-t)^3P_0 + 3t(1-t)^2P_1 + 3t^2(2-3t)P_2 + t^3P_3
 \end{aligned}
 \tag{B.9}$$

Los puntos final e inicial de la curva de Bézier están dados por:

$$\begin{aligned}
C(0) &= P_0 \\
C(1) &= P_2 \\
C'(t) &= -3(1-t)^2P_0 + 3(1-t)(1-3t)P_1 + 3t(2-3t)P_2 + 3t^2P_3 \\
C'(0) &= 3(P_1 - P_0) \\
C'(1) &= 2(P_2 - P_1)
\end{aligned}$$

Formulación de *Casteljau*



Primera columna: Puntos de control

Segunda columna: Curva lineal

Tercera columna: Curva de Bézier

Formulación con Polinomios de Bernstein

$$(x(t); y(t)) = \sum_{i=0}^n B_t P_i$$

P_i Puntos de control

$B_{i,n}$ Polinomios de Bernstein

Al evaluar cada polinomio de Bernstein, se va modelando la forma de la curva.

Código en MATLAB

```

xy=[ 0 0 ; -1 3 ; 2 5 ; 3 5 ; 4 -2 ; 5 1.5 ] %puntos de control
n=length(xy)-1 % CODIGO ANTERIOR

```

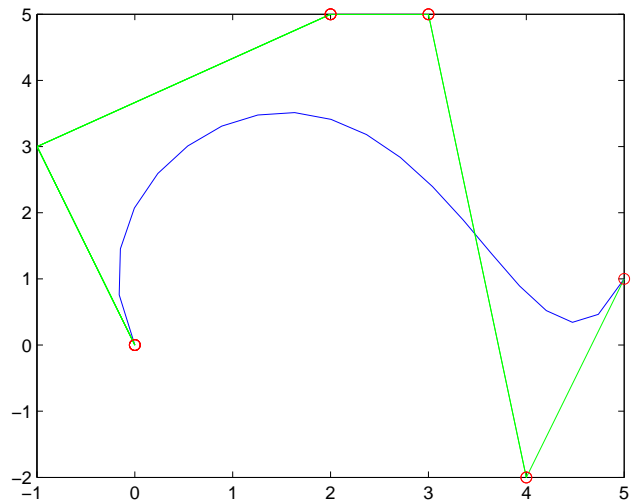


Figura B.6: Curva de Bézier

```

cx=0
cy=0
for i=1:n+1
cx=cx+xy(i,1)*matriz(i,:)
cy=cy+xy(i,2)*matriz(i,:)
end
Y=polyval(cx,x)
X=polyval(cy,x)
plot(Y,X)
hold on % PARA DIBUJAR LOS PTOS DE CTRL
for i=1:n+1
xc=xy(i,1)
yc=xy(i,2)
plot(xc,yc,'ro')
end

```

B.2.1. Propiedades de las curvas de Bézier

1. Interpolación de los puntos de control extremos:

$$C(0) = P_0$$

$$C(t) = P_n$$

Solo se interpolan los puntos de control

2. Tangencia al Polígono en los puntos de control en sus extremos

$$C'(0) = n(P_1 - P_0)$$

$$C'(t) = n(P_n - P_{n-1})$$

Solo el punto inicial y final son tangentes al polígono de control ya que la curva solo pasa por estos dos puntos.

3. Derivadas

De la definición $B_k(t)$ es inmediata la expresión

$$C(t) = n \sum_{k=0}^{k-1} P_{k-1} - P_k (B_k)^n(x)$$

Llamando:

$$\Delta P_j = P_{j+1} - P_j$$

$$C'(t) = n \sum_{k=0}^{n-1} \Delta P_k B_k^{n-1}(t)$$

Se comprueba que $C'(t)$ es una curva de Bézier de grado $n - 1$ con puntos de control $n\Delta P_k$.

En particular las derivadas en los puntos extremos son

$$C^r(0) = n(n-1)\dots(n-r+1)\Delta^r P_0$$

$$C^r(1) = n(n-1)\dots(n-r+1)\Delta^r P_{n-r}$$

La derivada solo se puede hallar en los puntos final e inicial, ya que solo por estos puntos es por donde pasa la curva a dibujar; esta curva es n veces derivable, siendo n el grado

de la curva.

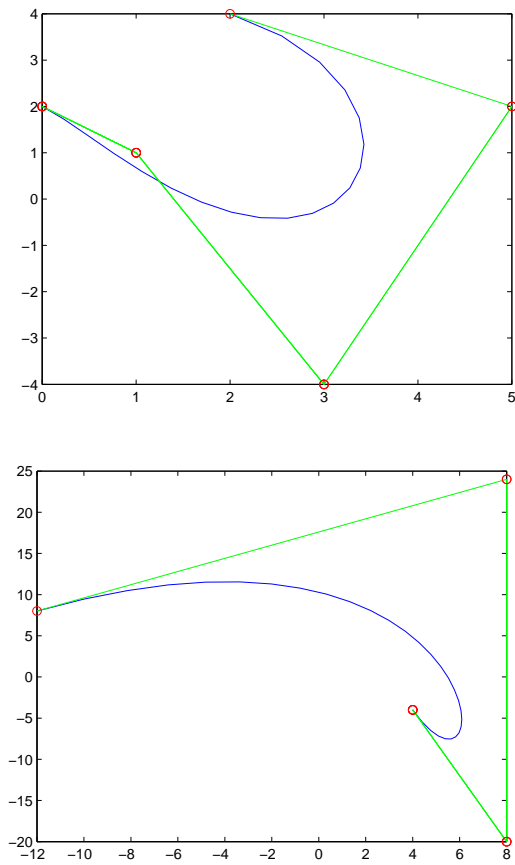


Figura B.7: Derivada de curva de Bézier

Código en MATLAB

```
clear all
close all
clc                                     % ENTRADA DE LOS PUNTOS DE CONTROL
xy=[ 0 2;
    1 1;
    3 -4;
    5 2 ;
    2 4    ]
n=length(xy)-2                          %SIENDO n EL GRADO DE LOS POLINOMIOS
```

```

                                BERNSTEIN A TRABAJARA
t(n+1)=1                        % EL 1ER VALOR DEL TRIANGULO DE PASCAL ES 1
base=t                          % SE CONSTRUYE UN VECTOR DE REFERENCIA
pascal=t                        %LA PRIMERA FILA DE LA MATRIZ DE PASCAL ES 1
for j=2:n+1                      %SE INICIA EN LA SEGUNDA FILA HASTA n+1
for i=1:n
t(i)=base(i)+base(i+1)          %FORMULA DE RECURRENCIA PARA PASCAL
pascal(j,i)=t(i)                % INSERTAMOS LOS COEFICIENTES EN LA MATRIZ
                                DEL TRIANGULO DE PASCAL
end
pascal(j,n+1)=1
base=t
end
for
i=n:-2:1                        %SE LE DA EL SIGNO CORRESPONDIENTE A LOS
                                COEFICIENTES DEL TRIANGULO DE PASCAL
pascal(:,i)=pascal(:,i)*-1
end matriz=pascal(n+1,:)
for j=2:n
bern=pascal(n+2-j,:).*t(j)      %SE TOMAN LOS COEFICIENTES DE
                                (X-X_0)^(n+2-J) Y SE MULTIPLICAN POR SU
                                COEFICIENTE BINOMICO
for i=1:n-j+2
matriz(j,i)=bern(i+j-1)        %MULTIPLICAR POR X^k EQUIVALE A AUMENTAR
                                k POSICIONES LOS COEFICIENTES DEL VECTOR
end
clear ber
end
matriz(n+1,1)=1
derivx=0
derivy=0
for
i=1:n+1
derivx=derivx+matriz(i,:).*(xy(i+1,1)-xy(i,1))*(n+1)
derivy=derivy+matriz(i,:).*(xy(i+1,2)-xy(i,2))*(n+1)

```

```

end x=linspace(0,1,20)
Y=polyval(derivx,x);

% SE EVALUAN LAS Y EN EL SOPORTE x (0,1)

cx(x) X=polyval(derivy,x);
% SE EVALUAN LAS X EN EL SOPORTE
x (0,1) cy(x)
plot(Y,X)

```

4.Control seudolocal

Si dos curvas de Bézier $C(t)$ y $\bar{C}(t) = B_k(t)(P_k - \bar{P}_k)$ tienen los mismos puntos de control excepto 1, entonces:

$$C(t) - \bar{C}(t) = B_k(t)(P_k - \bar{P}_k)$$

Dado que $B_k(t)$ alcanza su valor máximo en $t = \frac{k}{n}$, la máxima diferencia entre ambas curvas sucede en la zona correspondiente a las cercanías al valor del parámetro $t = \frac{k}{n}$ decreciendo la diferencia hacia 0 y coincidiendo en $t = 0$ y $t = 1$

Por lo tanto, el movimiento o modificación de 1 punto de control afecta a la forma de la curva, siendo más notorio en las cercanías al punto movido.

Código en MATLAB

```

a=ginput(1)
s=ones(length(xy),2)
s(:,1)=s(:,1)*a(1)
s(:,2)=s(:,2)*a(2)
eLe=xy-s
for i=1:length(xy)
ele=(eLe(i,1)^2+eLe(i,2)^2)^0.5
end
nod=find(ele==max(ele))

```

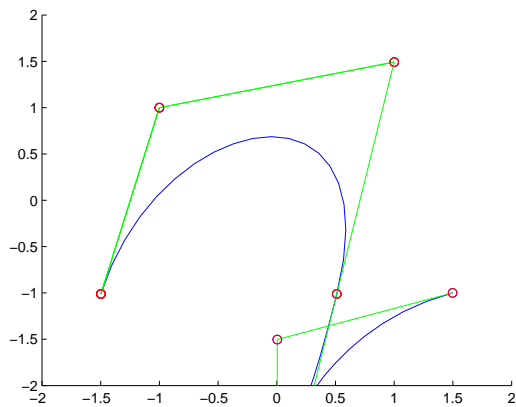
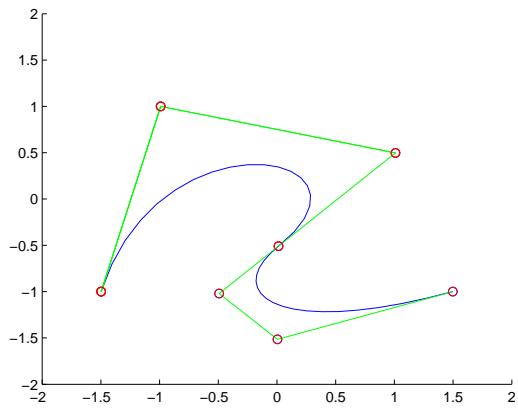


Figura B.8: Control seudolocal

```

plot(xy(nod), 'y*')           % # de nodo a cambiar
                               %nod=6 % --> (puntos - 2)

a=ginput(1)
if not(mod((nod-1)/3,1))
dx=a(1)-xy(nod,1);
% evalua cuanto avanza en x, y
dy=a(2)-xy(nod,2);
xy(nod,1)=a(1);
xy(nod,2)=a(2);
xy(nod-1,1)=xy(nod-1,1)+dx;
xy(nod-1,2)=xy(nod-1,2)+dy;
xy(nod+1,1)=xy(nod+1,1)+dx;
xy(nod+1,2)=xy(nod+1,2)+dy;

```

```

elseif
not(mod((nod-2)/3,1))
xy(nod-2,1)=xy(nod-1,1)+(xy(nod-2,1)-xy(nod-1,1))*
(a(1)-xy(nod-1,1))/(xy(nod,1)-xy(nod-1,1));
xy(nod-2,2)=xy(nod-1,2)+(xy(nod-2,2)-xy(nod-1,2))*
(a(2)-xy(nod-1,2))/(xy(nod,2)-xy(nod-1,2));
xy(nod,1)=a(1);
xy(nod,2)=a(2);
elseif
not(mod((nod-3)/3,1))
xy(nod+2,1)=(xy(nod+2,1)-xy(nod+1,1))/
(xy(nod+1)-xy(nod,1))*(xy(nod+1)-a(1))+xy(nod+1,1);
xy(nod+2,2)=(xy(nod+2,2)-xy(nod+1,2))
/(xy(nod+2)-xy(nod,2))*(xy(nod+2)-a(2))+xy(nod+1,2);
xy(nod,1)=a(1);
xy(nod,2)=a(2);
end
n=length(xy)
for o=0:n
cx=0;cy=0;
for i=1:4          % CONSTRUIR CURVA
cx=cx+xy(3*o+i,1)*matriz(i,:);
cy=cy+xy(3*o+i,2)*matriz(i,:);
end                %para dibujar curva
Y=polyval(cx,x);
X=polyval(cy,x);
plot(Y,X)         % DIBUJAR LINEAS DE CONTROL
for i=1:4
xc(i)=xy(3*o+i,1);
yc(i)=xy(3*o+i,2);
end
plot(xc,yc,'y')
plot(xc,yc,'b.')
axis([min(xy(:,1))-5,max(xy(:,1))+5,min(xy(:,2))-5,max(xy(:,2))+5])
end

```

5. Restricción de la envolvente convexa.

Una curva de Bézier con puntos de control P_0, P_1, \dots, P_n , es siempre interior a la envolvente convexa de dichos puntos. la envolvente convexa de un polígono de vértices P_0, P_1, \dots, P_n es la intersección de todos los conjuntos convexos que contienen el polígono, es decir el menor de los polígonos convexos que contienen al dado.

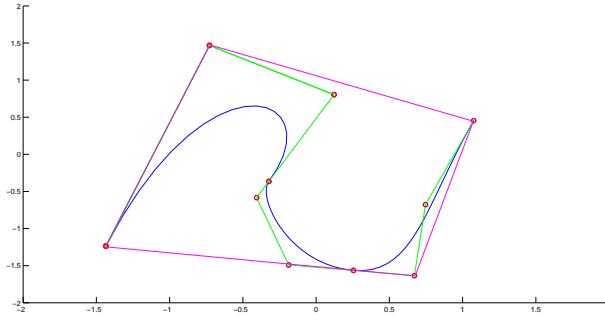


Figura B.9: Poligonal convexa

Esta propiedad obliga a la curva $c(t)$ a permanecer cercana a los puntos de control, es decir que la curva no escapa de la envolvente convexa.

6. Disminución de la variación.

Una recta arbitraria no intercepta a una curva de Bézier en más puntos de los que intercepta a su polígono de control. por lo tanto el número de veces que la recta corta al polígono de control es cota superior del número de veces que corta a la curva. Geométricamente esta propiedad establece que una curva de Bézier oscila menos veces que la poligonal que la define. Con esta propiedad se puede encontrar una cota para las raíces reales de un polinomio, si el polinomio esta expresado en la base Bernstein el número de cambios de signo en la ordenada del punto de control es precisamente esta cota.

7. Invarianza Afín

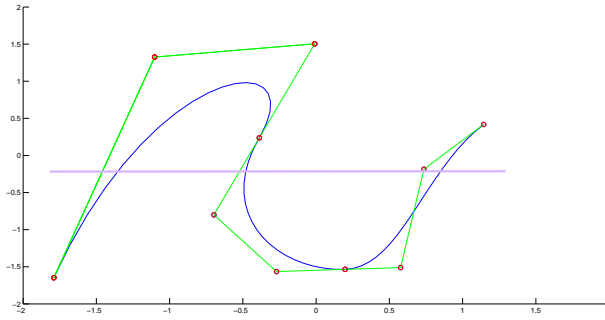


Figura B.10: Disminución de la variación

Dados los puntos P_0, P_1, \dots, P_n del plano afín f , buscamos un método para construir curvas a partir de los puntos P_0, P_1, \dots, P_n aplicando afinidad a estos puntos:

$$\bar{P} = AP + H$$

Donde:

$A \Rightarrow$ una matriz no singular, es decir una matriz cuadrada invertible.

$H \Rightarrow$ un vector columna representando una translación.

$$\bar{C}(t) = AC(t) + H = A[\sum_{k=0}^n P_k B_k(x)] + H$$

Entonces

$$\bar{C}(t) = [\sum_{k=0}^n \bar{P}_k B_k(x)] + H \Rightarrow \text{Transformada Afín de la curva original.}$$

8. Recurrencia

Considerando una curva de Bézier simple de segundo grado

$$C(t) = (1-t)^2 P_0 + 2(1-t)t P_1 + t^2 P_2$$

Puede ser expresada así:

$$C(t) = (1-t)[(1-t)P_0 + tP_1] + t[(1-t)P_1 + tP_2]$$

Se observa que:

$[(1-t)P_0 + tP_1]$ es una curva de Bézier de grado 1 para P_0 y P_1 y $[(1-t)P_1 + tP_2]$ es una curva de Bézier de grado 1 para P_1 y P_2

Considerando un valor fijo para $t = \bar{t}$

$$P_0^{(0)} = (1 - E)P_0 + EP_1$$

$$P_0^{(1)} = (1 - E)P_1 + EP_2$$

Con lo que se demuestra que:

$$C(E) = (1 - E)P_0^{(1)} + EP_1^{(1)}$$

Lo que nos dice que el punto $C(E)$ sobre la curva de Bézier de segundo grado puede considerarse como un punto de control sobre la curva de Bézier de primer grado con puntos de control $P_0^{(1)}$ y $P_1^{(1)}$

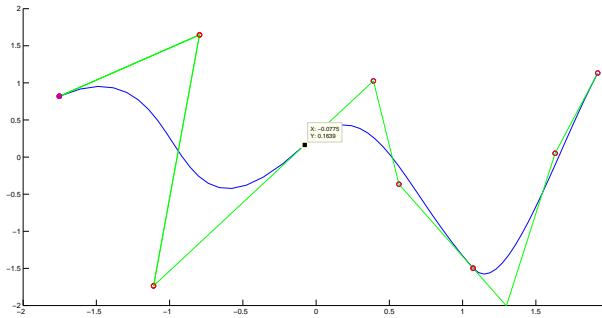


Figura B.11: Curva de Bézier

B.2.2. Conversión entre representaciones

En muchas ocasiones un polinomio de grado menor puede venir expresado en base de potencias, y resultaría conveniente expresarlo en la base de Bernstein, convertirlo a una curva de Bézier y modificar su forma al manipular sus puntos de control. El problema teóricamente es sencillo se trata de un cambio de base, esto es la expresión de cada polinomio de una base en función de los polinomios de otra.

Conversión a Potencias de Bernstein.

$$t^i = \sum_{k=1}^n \binom{n-i}{k-i} t^k (1-t)^{n-k} \quad (\text{B.10})$$

Teniendo en cuenta la definición de los polinomios de Bernstein.

$$t^i = \sum_{k=1}^n \frac{\binom{n-i}{k-i}}{\binom{n}{k}} B_k(t) \quad (\text{B.11})$$

Se sabe que $t^0 = 1$, se expresa como una combinación de todos los polinomios de Bernstein. Según la relación ya conocida

$$t^0 = 1 = \sum_{k=1}^n B_k(t)$$

Por otra parte

$t^1 \rightarrow$ combinación lineal de $B_1(t), \dots, B_n(t)$

$t^2 \rightarrow$ combinación de $B_2(t), \dots, B_n(t)$

t^n que sólo es función de $B_n(t) = t^n$

$$c_{ik} = \frac{\binom{n-i}{k-i}}{\binom{n}{k}} \quad \left(\begin{array}{l} 0 \leq i \leq n \\ i \leq k \leq n \end{array} \right)$$

$$c_{ik} = 0 \quad \text{para } i > k$$

(B.10) expresado en forma matricial

$$\begin{bmatrix} 1 \\ t \\ t^2 \\ \vdots \\ t^n \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} & c_{02} & \cdots & c_{0n} \\ 0 & c_{11} & c_{12} & \cdots & c_{1n} \\ 0 & 0 & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & c_{nn} \end{bmatrix} \begin{bmatrix} B_0(t) \\ B_1(t) \\ B_2(t) \\ \vdots \\ B_n(t) \end{bmatrix} \quad (\text{B.12})$$

$c(t)$ expresada como un polinomio de potencias:

$$c(t) = \sum_{i=0}^n Q_i t^i$$

$$c(t) = [Q_0, Q_1, \dots, Q_n] \begin{bmatrix} 1 \\ t \\ t^2 \\ \vdots \\ t^n \end{bmatrix}$$

$$[Q_0, Q_1, \dots, Q_n] \begin{bmatrix} c_{00} & c_{01} & c_{02} & \cdots & c_{0n} \\ 0 & c_{11} & c_{12} & \cdots & c_{1n} \\ 0 & 0 & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & c_{nn} \end{bmatrix} \begin{bmatrix} B_0(t) \\ B_1(t) \\ B_2(t) \\ \vdots \\ B_n(t) \end{bmatrix} =$$

$$[P_0, P_1, \dots, P_n] \begin{bmatrix} B_0(t) \\ B_1(t) \\ B_2(t) \\ \vdots \\ B_n(t) \end{bmatrix} = \sum_{i=0}^n P_i B_i(t)$$

Por lo tanto la relación entre los coeficientes P_c (puntos de control) y los originales Q_c (base de potencias)

$$\begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ \vdots \\ P_n(t) \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} & c_{02} & \cdots & c_{0n} \\ 0 & c_{11} & c_{12} & \cdots & c_{1n} \\ 0 & 0 & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & c_{nn} \end{bmatrix} \begin{bmatrix} Q_0(t) \\ Q_1(t) \\ Q_2(t) \\ \vdots \\ Q_n(t) \end{bmatrix}$$

Ejemplo

para $n=3$ $c(t) = (0, 1) + t(3, 6) + t^2(3, -3) + t^3(-1, -2)$

$$\begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{1}{3} & 0 & 0 \\ 1 & \frac{1}{3} & \frac{1}{3} & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 3 & 6 \\ 3 & -3 \\ -1 & -2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 3 \\ 3 & 4 \\ 5 & 2 \end{bmatrix}$$

Por lo tanto el mismo polinomio en base *Bernstein* es:

$$c(t) = (1 - 3)^3(0, 1) + 3t(1 - t)^2(1, 3) + 3t^2(1 - t)(3, 4) + t^3(5, 2)$$

Donde ahora los coeficientes tienen el significado de puntos de control.

Código en MATLAB

```
% SE CORRE DESPUES DE CALCULADO EL POLINOMIO DE BEZIER
for k=0:n
for i=0:n
if i<=k
PtoB(k+1,i+1)=(factorial(k)*factorial(n-i))/(factorial(k-i)*factorial(n))
end
end
end
Qx=cx(n+1:-1:1);

% COEFICIENTES POLINOMICOS DE BEZIER (x)
Qx=Qx'

% INVERTIDOS & COLUMNA
Qy=cy(n+1:-1:1);
Qy=Qy' Px=PtoB*Qx

% PUNTOS DE CONTROL=MATRIZ*COEFICIENTES BEZIER (x)
Py=PtoB*Qy
% PUNTOS DE CONTROL=MATRIZ*COEFICIENTES BEZIER (y)

% SE CORRE DESPUES DE CALCULADO EL POLINOMIO DE BEZIER
for k=0:n
for i=0:n
if i<=k
PtoB(k+1,i+1)=(factorial(k)*factorial(n-i))/(factorial(k-i)*factorial(n))
end
```

```

end
end
Qx=cx(n+1:-1:1);
% COEFICIENTES POLINOMICOS DE BEZIER (x)
Qx=Qx'
% INVERTIDOS & COLUMNA
Qy=cy(n+1:-1:1);
Qy=Qy' Px=PtoB*Qx
% PUNTOS DE CONTROL=MATRIZ*COEFICIENTES BEZIER (x)
Py=PtoB*Qy
% PUNTOS DE CONTROL=MATRIZ*COEFICIENTES BEZIER (y)

```

Conversión de Bernstein a potencias

En el polinomio de Bernstein se desarrolla el factor $(1-t)^{n-k}$, por la fórmula binomial se obtiene:

$$B_n(t) = \sum_{k=1}^n (-1)^{j-k} \binom{n}{j} \binom{j}{n} t^j \quad (\text{B.13})$$

Se expresa cada polinomio de Bernstein según la base de potencias $B_n(t) = t^n \rightarrow$ función sólo de la última potencia.

Donde:

$$d_{kj} = (-1)^{j-k} \binom{n}{j} \binom{j}{k}$$

$$\begin{bmatrix} B_0(t) \\ B_1(t) \\ B_2(t) \\ \vdots \\ B_n(t) \end{bmatrix} = \begin{bmatrix} d_{00} & d_{01} & d_{02} & \dots & d_{0n} \\ 0 & d_{11} & d_{12} & \dots & d_{1n} \\ 0 & 0 & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & d_{nn} \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ \vdots \\ t^n \end{bmatrix}$$

El polinomio $c(t)$ expresado en la base Bernstein

$$c(t) = \sum_{k=0}^n P_k B_k(t)$$

Su expresión en base de potencias

$$c(t) = \sum_{k=0}^n \phi_k t^k$$

Y la transformación de polinómico a los puntos de control se da así.

$$\begin{bmatrix} Q_0(t) \\ Q_1(t) \\ Q_2(t) \\ \vdots \\ Q_n(t) \end{bmatrix} = \begin{bmatrix} c_{00} & 0 & 0 & \cdots & 0 \\ c_{01} & c_{11} & 0 & \cdots & 0 \\ c_{02} & c_{12} & c_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{0n} & c_{1n} & c_{2n} & \cdots & c_{nn} \end{bmatrix} \begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ \vdots \\ P_n(t) \end{bmatrix}$$

B.2.3. Técnicas constructivas para curvas de Bézier simples

Se describen dos técnicas básicas, que serán fundamentales para el desarrollo de operadores de alto nivel para manipulación de curvas y superficies. Se trata de procedimientos para elevar y disminuir el grado de la curva de Bézier.

Elevación de grado

Si una curva polinomial de grado n se expresa en base de potencias

$$c(t) = \sum_{k=0}^n \phi_k t^k$$

Escribiendo una de grado $n + 1$

$$c(t) = \sum_{k=0}^{n+1} \bar{\phi}_k t^k$$

$$\bar{\phi}_k = \phi_k \quad \text{para } k = 0, \dots, n \quad \text{y} \quad \bar{\phi}_{n+1} = 0$$

Esto es con el fin de lograr un coeficiente más para controlar la curva, y cuando dicho coeficiente sea modificado, la curva será de grado $n + 1$ con lo que se logra el cambio de grado en la curva

En las curvas con base Bernstein el papel que juegan los coeficientes es fundamental cuando se define la geometría de la curva, caso contrario al de las curvas expresadas en base de Potencias en donde estos coeficientes no ofrecen mayor información sobre la

forma de la curva, razón por la cual el proceso de elevación del grado para una curva representada en base de potencias carece de sentido.

Una curva de grado n se expresa en Base de Potencias:

$$c(t) = \sum_{k=0}^n Q_k B_k^n(t)$$

También podrá escribirse como una de grado $n + 1$

$$c(t) = \sum_{k=0}^n \bar{Q}_k B_k^{n+1}(t)$$

Igualando estas dos curvas expresadas en los polinomios de Bernstein.

$$\sum_{k=0}^{n+1} \bar{P}_k \binom{n+1}{k} t^k (1-t)^{n+1-k} = \sum_{k=0}^n P_k \binom{n}{k} t^k (1-t)^{n-k}$$

Lo que se pretende es que estas dos curvas sean idénticas en su forma, simplemente van a estar expresadas en bases distintas, básicamente lo que sucede es que no sólo se añade un nuevo punto de control, sino que se reubican todos los puntos existentes a excepción del primero y del último que permanecen inamovibles.

$$\begin{aligned} \hat{P}_k &= \frac{k}{n+1} P_{k-1} + \left[1 - \frac{k}{n+1}\right] P_k \\ \hat{P}_0 &= P_0 \\ \hat{P}_{n+1} &= P_n \end{aligned} \quad k = 1, \dots, n \quad (\text{B.14})$$

Ejemplo

$$c(t) = (1-t)^3(0, 1) + 3t(1-t)^2(1, 3) + 3t^2(1-t)(3, -4) + t^3(5, 2)$$

coordenadas de los puntos de control:

$$P_0 = (0, 1) \quad P_1 = (1, 3) \quad P_2 = (3, -4) \quad P_3 = (5, 2)$$

$$\widehat{P}_0 = P_0 = (0, 1)$$

$$\widehat{P}_1 = \frac{4}{3}P_1 + \frac{1}{3}P_0 = \frac{4}{3}(1, 3) + \frac{1}{3}(0, 1) = \left(\frac{4}{3}, 5\right)$$

$$\widehat{P}_2 = \frac{2}{4}P_2 - \frac{2}{2}\widehat{P}_1 = 2(3, -4) - (4, 5) = (5, -11)$$

$$\widehat{P}_3 = 4P_3 - 3\widehat{P}_2 = 4(5, 2) - 3(5, -11) = (11, 20)$$

$$\widehat{P}_3 = P_3 = (5, 2)$$

La nueva curva de mayor grado esta dada por:

$$\widehat{c}(t) = (1-t)^4(0, 1) + 4t(1-t)^3\left(\frac{4}{3}, 5\right) + 6t^2(1-t)^2(5, -11) + 4t^3(11, 20) + t^4(5, 2)$$

Código en MATLAB

```
% aumento de grado
Pmax(1)=xy(1,1),
Pmax(n+2)=xy(n+1,1)
Pmay(1)=xy(1,2),
Pmay(n+2)=xy(n+1,2)
for i=1:n
Pmax(i+1)=i/(n+1)*xy(i,1)+(1-i/(n+1))*xy(i+1,1)
Pmay(i+1)=i/(n+1)*xy(i,2)+(1-i/(n+1))*xy(i+1,2)
end
```

Reducción de grado

Se trata del proceso inverso al descrito en la sección anterior y evidentemente no siempre es posible. La reducción se aplica a partir de una curva de Bézier de grado n .

$$c(t) = \sum_{k=0}^n P_k B_k^n(t)$$

a) Determinar si realmente degenera al grado $n - 1$

$$c(t) = \sum_{k=0}^{n-1} \overline{P}_k B_k^{n-1}(t)$$

Para resolver

Expresar el cambio de base Bernstein \Leftrightarrow potencias
 Calcular el coeficiente de t_n y comprobar si es nulo

$$Q_n = d_{0n}P_0 + d_{1n}P_1 + \dots + d_{nn}P_n$$

$$d_{in} = (-1)^{n-i} \binom{n}{i} \quad (\text{B.15})$$

Código en MATLAB

```
% disminución de grado
Pmix(1)=Pmax(1)
for k=1:n
Pmix(k+1)=(n*Pmax(k+1)-(k-1)*Pmix(k))/(n-k)
end
```

B.2.4. Reparametrización de la curvas de Bézier

Los polinomios de Bernstein $B_k(t)$ disfrutan de importantes propiedades en el intervalo $[0, 1]$. Dichas propiedades son transferidas a las curvas de Bézier en el intervalo $[0, 1]$
 Obsérvese que, se pueden definir curvas de Bézier en cualquier intervalo $[a, b]$, mediante un cambio de variable.

$$\bar{C}(s) = \sum_{k=0}^n P_k B_k \frac{s-a}{b-a} \quad a \leq s \leq b \quad (\text{B.16})$$

Donde llamando $t = \frac{s-a}{b-a}$, verificando que para $s = a$ se obtiene $t = 0$, para $s = b$, obtenemos $t=1$ y siempre $0 \leq t \leq 1$

Está claro que esta curva y la definida por

$$C(s) \sum_{k=0}^n P_k B_k(t) \quad 0 \leq t \leq 1 \quad (\text{B.17})$$

Son idénticas el procesos ha consistido simplemente en realizar un cambio de variable, o una reparametrización.

Ejemplo

Considerando una curva de Bézier cuadrática

$$C(t) = (1 - t)^2(1, 0) + 2t(1 - t)(0, 2) + t^2 \quad t \in [0, 1]$$

Y suponiendo que se quiere que este definida en el intervalo $[3, 5]$, la curva reparametrizada es:

$$\bar{C}(s) = \left(1 - \frac{s-3}{2}\right)^2(-1, 0) + 2\frac{s-3}{2}\left(1 - \frac{s-3}{2}\right)(0, 2) + \left(\frac{s-3}{2}\right)^2(1, 0) \quad s \in [3, 5]$$

Estas curvas son geoméricamente iguales

B.2.5. Curvas de Bézier Compuestas

En la sección anterior, se dejó claro que las curvas de Bézier simples o de un tramo, no son más que un polinomio (en cada componente) expresado en la base Bernstein.

Al margen de las importantes propiedades que relacionan geoméricamente la forma de la curva con sus coeficientes o puntos de control, se pueden ver los diferentes inconvenientes al realizar el proceso de interpolación, principalmente en lo referido al elevado grado de las curvas, necesario para generar formas complejas, los problemas se hacen más notorios en estos casos, ya que se generan altas oscilaciones cuando se obliga a pasar la curva por los puntos de interpolación, e incluso la modificación de un coeficiente cuando se habla de la base Bernstein cambia totalmente la forma de la curva.

La solución a este tipo de problemas consiste en pegar sucesivamente varias curvas de Bézier simples de bajo grado, para construir lo que en adelante se denominará *curva de Bézier compuesta*, o simplemente una curva de Bézier.

Composición de dos curvas de Bézier Simples

Primero se consideran la composición geométrica de dos curvas de Bézier simples, sean $P_0, P_1, P_2, \dots, P_n$ puntos de control para la primera curva y $R_0, R_1, R_2, \dots, R_n$ los de la

segunda curva. En general no tendrán que ser del mismo grado las curvas a analizar, pero si es conveniente que lo sean, por cuestiones de simplicidad en los cálculos y notaciones.

De las curvas de Bézier y sus propiedades se sabe que, la primera curva interpolará P_0 y P_n , y la segunda lo hará para R_0 y P_n , en donde $P_n = R_0$ para que esta curva compuesta sea continua.

Esta es la primera condición para unir dos curvas de Bézier, si se observa la gráfica de esta curva se puede percibir que en general la tangente no será continua en el punto de unión.

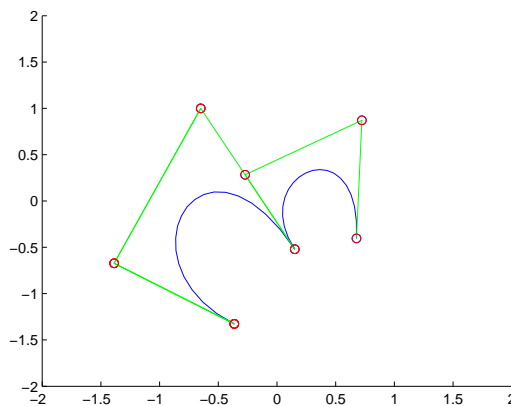


Figura B.12: Discontinuidad de la primera derivada

Ahora se tratará de conseguir suavidad en el punto de unión de las dos curvas, se sabe que el vector tangente de estas dos curvas viene dado por:

$$T_0 = C'(0) = n(P_1 - P_0)$$

$$T_n = C'(1) = n(P_n - P_{n-1})$$

Lo importante es que la dirección de los vectores de las curvas a unir, compartan el mismo sentido, para la primera curva se tiene que el vector estará en la dirección de $P_n - P_{n-1}$, mientras que el vector tangente a la segunda curva tiene la dirección $R_1 - R_0 = R_1 - P_n$.

Por lo tanto la condición que deben cumplir estas dos curvas para que el vector tangente tenga la misma dirección para las dos curvas es que los vectores $P_n - P_{n-1}$ y $R_1 - R_0$ sean proporcionales, es decir que los puntos P_n y R_n estén alineados, esto se

logra, haciendo que la poligonal del punto P_n , sea una prolongación de la poligonal del punto R_0 .

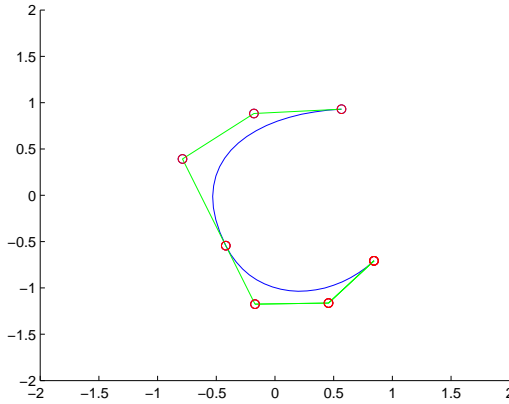


Figura B.13: Continuidad de la dirección

Esta construcción solo garantiza la continuidad en la dirección de los vectores tangentes, pero no del propio vector, se observa que la derivada de la suma compuesta no es necesariamente continua, para este caso se dice que se tiene continuidad geométrica o visual de clase G^1 .

Escribiendo una expresión análoga para las tangentes a la segunda curva:

$$\bar{T}_0 = C'(0) = n(R_1 - R_0)$$

$$\bar{T}_n = C'(1) = n(R_n - R_{n-1})$$

Aquí las derivadas son respecto al parámetro de la segunda curva, es decir $s \in [0, 1]$, pues ambas curvas se suponen definidas en la Base de Bernstein con el parámetro variando entre 0 y 1.

Para que la curva compuesta sea de clase C^1 , es decir que su derivada sea continua, lo primero que se debe hacer es elegir una parámetro global respecto al cual calcular las derivadas. Se sabe que una curva de Bézier puede definirse sobre cualquier intervalo $[a, b]$, usando el cambio de variable $t = \frac{s-a}{b-a}$ con $s \in [a, b]$, este cambio de variable no

altera la naturaleza de la curva, pero si cambia sus derivadas primeras y las de orden superior.

Para unir dos curvas de Bézier y proporcionar un parámetro global para ambas, se deben elegir intervalos consecutivos por ejemplo $[u_0, u_1]$ en donde está definida la primera curva y $[u_1, u_2]$ en donde está definida la segunda curva:

$$C_1(u) = \sum_{k=0}^n P_k B_k\left(\frac{u-u_0}{u_1-u_0}\right) \quad \text{Donde } \implies u \in [u_0, u_1] \quad (\text{B.18})$$

$$C_2(u) = \sum_{k=0}^n R_k B_k\left(\frac{u-u_1}{u_2-u_1}\right) \quad \text{Donde } \implies u \in [u_1, u_2] \quad (\text{B.19})$$

Esta curva es continua si y solo si $P_n = R_0$

Ahora está claro que la continuidad de las curvas depende de los parámetros utilizados para definir las curvas a unir y que el proceso de reparametrización no cambia la geometría de la curva.

En efecto se tiene:

$$C_1'(u) = \frac{1}{u-u_0} \sum_{k=0}^n P_k B_k'\left(\frac{u-u_0}{u_1-u_0}\right)$$

$$C_2'(u) = \frac{1}{u-u_1} \sum_{k=0}^n R_k B_k'\left(\frac{u-u_1}{u_2-u_1}\right)$$

Donde B_k' indica la derivación del polinomio de Bernstein $B_k(t) = \frac{n}{k} t^k (1-t)^{n-k}$ respecto a t .

En el punto de unión de las dos curvas (único en $[u_0, u_2]$ en el que la derivada puede ser continua)

$$C_1'(u) = \frac{1}{u-u_0} \sum_{k=0}^n P_k B_k'(1) = \left(\frac{u-u_0}{u_1-u_0}\right) n(P_n - P_{n-1})$$

$$C_2'(u) = \frac{1}{u-u_1} \sum_{k=0}^n R_k B_k'(1) = \left(\frac{u-u_1}{u_2-u_1}\right) n(R_1 - R_0)$$

Por lo tanto la curva compuesta tendrá derivada continua si y solo si:

$$\frac{1}{u_1-u_0} n(P_n - P_{n-1}) = \frac{1}{u_2-u_1} n(R_1 - R_0)$$

Esta condición puede escribirse:

$$R_1 - R_0 = \lambda(P_n - P_{n-1}) \text{ con } \lambda = \frac{u_2 - u_1}{u_1 - u_0} = \frac{\|R_1 - R_0\|}{\|P_n - P_{n-1}\|}$$

Donde la primera ecuación expresa que los puntos $P_{n-1}, P_n = R_0, R_1$ están alineados y la segunda, que las longitudes del primer segmento en la segunda curva y del último segmento en la primera curva deben ser proporcionales a las amplitudes de los subintervalos de definición de la curva.

Es de gran importancia destacar que si cualquier elección de puntos (u_0, u_1, u_2) tales que $u_2 - u_1$ y $u_1 - u_0$ cumplen las condiciones requeridas, la suma total será de clase C^1 en $[u_0, u_1]$, en cambio si no cumplen las condiciones de proporcionalidad, estarán definidas en G^1 .

Ejemplo para la construcción de una curva de Bézier para dos tramos simples:

Sean los puntos

$P_0 = (0, 0), P_1 = (-2, 2), P_2 = (2, 4)$ puntos de control para la primera curva

$R_0(2, 4), R_1(3, 4), R_2(6, 2)$ para la segunda curva

$C_1(t) = (1-t)^2(0, 0) + 2t(1-t)(-2, 2) + t^2(2, 4) \longrightarrow t \in [0, 1]$

$C_2(t) = (1-t)^2(2, 4) + 2t(1-t)(3, 4) + t^2(6, 2) \longrightarrow t \in [0, 1]$

Se observa que los puntos $P_1, P_2 = R_0, R_1$, son colineales. Sin embargo estas dos curvas son totalmente independientes, cada una con su definición sobre el intervalo $[0, 1]$, para construir una curva de Bézier a partir de $C_1(t)$ y $C_2(t)$ se deben elegir los subintervalos $[u_0, u_1]$ y $[u_1, u_2]$ en los que variará el parámetro global u .

Se aplica la ecuación.

$$\frac{u_2 - u_1}{u_1 - u_0} = \frac{\|R_1 - R_0\|}{\|P_n - P_{n-1}\|} = \frac{4, 2}{3, 2} \quad (\text{B.20})$$

Por tanto debe ser $u_2 - u_1 = \frac{1}{2}(u_1 - u_0)$

Ahora se puede definir la curva en el intervalo que deseemos, dando los valores correspondientes para u_0, u_1 y u_2 quedando definida la curva para el intervalo $[u_0, u_2]$

B.3. Curvas de Bézier Compuestas

Como ya se vio, se pueden imponer condiciones para los puntos de control, y de esta manera "pegar" curvas de Bézier para modelar un objeto; siempre que se cumplan las condiciones antes mencionadas.

$$C''(0) = n(n-1)\Delta^2 P_0 = n(n-1)(P_2 - 2P_1 + P_0)$$

$$C''(1) = n(n-1)\Delta^2 P_{n-2} = n(n-1)(P_n - 2P_{n-1} + P_{n-2})$$

Teniendo en cuenta que las curvas de Bézier simples están definidas sobre los intervalos $[u_0, u_1]$ y $[u_1, u_2]$, para esta curva será continua la segunda derivada si y solo si:

$$\frac{1}{(u_1 - u_0)^2}(P_n - 2P_{n-1} + P_{n-2}) = \frac{1}{(u_2 - u_1)^2}(R_2 - 2R_1 + R_0)$$

Este tipo de continuidad de la segunda derivada impone muchas restricciones a los puntos de control de las curvas, ya que no solo debe ser colineales, sino que deben estar ubicados a una distancia proporcional, que sugiere una gran complicación, por ellos cuando se tratan curvas de Bézier solo se habla de continuidad C^1 .

Curvas de Bézier C^1 de varios tramos.

Como se sabe, para unir dos curvas de Bézier definidas respecto a un parámetro $t \in [0, 1]$

$$C(t) = \sum_{k=0}^n P_k B_k(t) \quad t \in [0, 1]$$

$$C(t) = \sum_{k=0}^n R_k B_k(t) \quad t \in [0, 1]$$

Para ello basta que se cumpla la condición de que $P_n = R_0$ y que P_{n-1}, P_n, R_0 y R_1 estén alineados.

Seguidamente se debe seleccionar el rango en el cual se va a definir la curva $[u_0, u_2]$, para el parámetro global u esto se logra por medio de un cambio de variable, y dos subintervalos consecutivos sobre los que se definen las dos curvas a unir $[u_0, u_1]$ y $[u_1, u_2]$, para una curva compuesta de grado C^1 es suficiente que se cumpla:

$$\frac{u_2 - u_1}{u_1 - u_0} = \frac{\| R_1 - R_0 \|}{\| P_n - P_{n-1} \|} \quad (\text{B.21})$$

Si se dan primero los subintervalos, entonces se imponen condiciones sobre el punto R_1 , en cambio si se dan los puntos de control de las curvas, se imponen restricciones sobre las amplitudes de los subintervalos, usar el segundo enfoque es mas práctico, ya que las restricciones sobre los puntos de control quedan reducidas al mínimo.

$$\begin{aligned} C_1(t) & \sum_{k=0}^n P_k^{(1)} B_k(t) \\ C_2(t) & \sum_{k=0}^n P_k^{(2)} B_k(t) \quad t \in [0, 1] \\ & \dots\dots\dots \\ C_m(t) & \sum_{k=0}^n P_k^{(m)} B_k(t) \end{aligned}$$

Se sabe que la condición para que la curva sea de continuidad C^1 es:

$$\begin{aligned} P_n^1 &= P_0^2 \\ P_{n-1}^1, P_n^1, P_0^2, P_1^2 & \text{ están alineados} \\ P_n^2 &= P_0^3 \\ P_{n-1}^2, P_n^2, P_0^3, P_1^3 & \text{ están alineados} \\ P_n^{m-1} &= P_0^m \\ P_{n-1}^{m-1}, P_n^{m-1}, P_0^m, P_1^m & \text{ están alineados} \end{aligned}$$

Una vez que se cumplen estas restricciones se deben definir cada uno de los intervalos en donde se va a definir la curva C_i de forma que además se tenga:

$$\frac{u_{i+1} - u_i}{u_i - u_{i-1}} = \frac{\| P_1^{i+1} - P_0^{i+1} \|}{\| P_n^i - P_{n-1}^i \|}$$

para $i = 1, \dots, m - 1$

Código en MATLAB

```
%curvas de Bézier compuestas

axis([-2,2,-2,2])
hold on pascal=[ 0 0 0 1; 0 0 -1 1; 0 1 -2 1; -1 3 -3 1];
matriz=[-1 3 -3 1; 3 -6 3 0; -3 3 0 0; 1 0 0 0];
for i=1:4
```

```

a=ginput(1)
plot(a(1),a(2),'o') xy(i,1)=a(1) xy(i,2)=a(2)
end
x=linspace(0,1,20)
cx=0;cy=0
for i=1:4
cx=cx+xy(i,1)*matriz(i,:)
cy=cy+xy(i,2)*matriz(i,:)
end Y=polyval(cx,x)
X=polyval(cy,x)
plot(Y,X)
for i=1:4
xc(i)=xy(i,1) yc(i)=xy(i,2); plot(xc,yc,'g') plot(xc,yc,'ro')
end
% HASTA ACA BEZIER SIMPLE
for o=1:2 a=ginput(1)
% TOMO EL SEGUNDO PUNTO
% DE LA SEGUNDA CURVA
xy(3*o+2,1)=a(1)
% SU "X", PERO EL "Y" SE
% HACE PASAR POR LA RECTA POR
% LOS 2 PTOS ANTERIORES FORMAN

xy(3*o+2,2)=0.5*(xy(3*o+1,2)-xy(3*o,2))/(xy(3*o+1,1)
-xy(3*o,1))*(2*a(1)-xy(3*o+1,1)-xy(3*o,1))+
0.5*(xy(3*o+1,2)+xy(3*o,2))
plot(a(1),xy(3*o+2,2),'o')
for i=1:2
% LOS 2 PTOS QUE FALTAN
a=ginput(1)
plot(a(1),a(2),'o') xy(3*o+2+i,1)=a(1)
xy(3*o+2+i,2)=a(2)
end cx=0;cy=0
for i=1:4
% CONSTRUIR CURVA
cx=cx+xy(3*o+i,1)*matriz(i,:)
cy=cy+xy(3*o+i,2)*matriz(i,:)
end
% DIBUJAR CURVA
Y=polyval(cx,x)
X=polyval(cy,x)

```

```

plot(Y,X)                                % DIBUJAR LINEAS DE CONTROL
for i=1:4 xc(i)=xy(3*o+i,1)
yc(i)=xy(3*o+i,2)
end plot(xc,yc,'g')
plot(xc,yc,'ro')
end

```

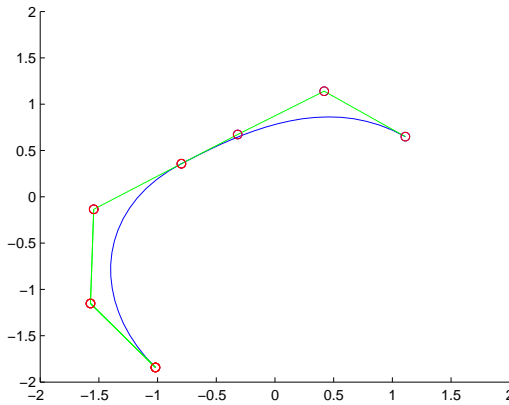


Figura B.14: Curva de Bézier compuesta

B.3.1. Subdivisión

Considerando una curva de Bézier de grado n ,

$$C(u) = \sum_{k=0}^n P_k B_k \frac{u-a}{b-a}$$

Considerando a c un punto sobre el intervalo $[a, b]$, es posible determinar dos subintervalos $[a, c]$; $[c, b]$, entonces la curva se define como un arco C_1 definido para el intervalo $[a, c]$ y otro arco C_2 definido para el intervalo $[c, b]$ siendo C_c el punto de unión de ambas curvas.

Como $C_1(u)$ es un polinomio de grado n definido en $[a, c]$ se puede expresar en la Base Bernstein,

$$C_1(u) = \sum_{k=0}^n P_k B_k \frac{u-a}{c-a}$$

Ahora el problema consiste en encontrar puntos de control para esta curva, a este proceso se le llama subdivisión de la curva $C(u)$ o refinamiento del intervalo $[a, b]$, se observa que la segunda curva $C_2(u)$ es otra curva de Bézier con puntos de control, de esta forma puede considerarse que la curva $C(u)$ esta compuesta por dos curvas de Bézier simples.

Aplicaciones del proceso de subdivisión.

1. Partición de una curva de Bézier compuesta en dos curvas, para esto se selecciona un punto de corte, se determina el tramo al que pertenece dicho punto, y finalmente se subdivide el tramo.
2. Incremento del control; mediante la subdivisión se puede disponer de más puntos de control en la zona subdividida.
3. Mayor exactitud; los nuevos puntos de control se aproximan mas a la curva, de lo que lo hacían los puntos de control originales respecto a la curva original, constituyendo una mejor aproximación a la curva completa $C(u)$.

Este proceso puede repetirse en cada tramo hasta lograr una aproximación cada vez más fina, a esto se le conoce como la fase rendering de la curva, así en lugar de obtener puntos próximos sobre la curva y unirlos mediante rectas para dibujarla, se calcula una poligonal suficientemente próxima y esta es la que se dibuja. Este procedimiento es mucho más económico y tiene la ventaja de ser adaptativo.

La subdivisión se lleva a cabo así:

$$C(u) = \sum_{k=0}^n P_k B_k(u) \quad 0 \leq u \leq 1$$

$$C_1(u) = \sum_{k=0}^n R_k B_k\left(\frac{u}{c}\right) \quad 0 \leq u \leq c$$

Para relacionar los puntos R_k con los puntos iniciales P_k se necesita hallar una expresión adecuada,

$$B_k(\lambda u) = \sum_{j=k}^n (-1)^{j-k} \binom{n}{j} \binom{j}{k} (\lambda u)^j \quad (\text{B.22})$$

$$\begin{aligned} B_k(\lambda u) &= \sum_{j=k}^n (-1)^{j-k} \\ &= \sum_{j=k}^n (-1)^{j-k} \left[\sum_{i=j}^n \binom{i-k}{j-k} \binom{i}{k} B_i(u) \right] \end{aligned}$$

Poniendo $\binom{n}{i} = 0$ si $n < i$ o $i < n$, se pueden iniciar la dos sumatorias desde 0 e intercambiando los términos se obtiene:

$$\begin{aligned} B_k(\lambda u) &= \sum_{i=0}^n (-1) \binom{i}{k} B_i(u) \left[\sum_{j=0}^n (-1)^{j-k} \binom{i-k}{j-k} \lambda^j \right] \\ &= B_k(\lambda u) = \sum_{i=k}^n (-1) \binom{i}{k} B_i(u) \left[\sum_{j=k}^i (-1)^{j-k} \binom{i-k}{j-k} \lambda^j \right] \quad (\text{B.23}) \\ &\quad \sum_{j=k}^i (-1)^{j-k} \binom{i-k}{j-k} \lambda^j = \sum_{h=0}^i (-1)^h \binom{i-k}{h} \lambda^{k+h} = \\ &\quad \lambda^k \sum_{h=0}^{i-k} (-1)^h \binom{i-k}{h} \lambda^h = \lambda^k (1-\lambda)^{i-k} = \frac{1}{\binom{i}{k}} B_k^i(\lambda) \end{aligned}$$

Siendo $B_k^i(\lambda)$ el polinomio de Bernstein de grado i

$$B_k^i(\lambda u) = \sum_{i=k}^i (\lambda) B_i(u) \quad (\text{B.24})$$

Finalmente relacionando las dos expresiones,

$$\lambda^k \sum_{k=0}^n P_k B_k(u) = \sum_{k=0}^n P_k B_k\left(\frac{u}{c}\right) = \sum_{k=0}^n P_k \left[\sum_{i=k}^n B_k^i(c) B_i\left(\frac{u}{c}\right) \right] =$$

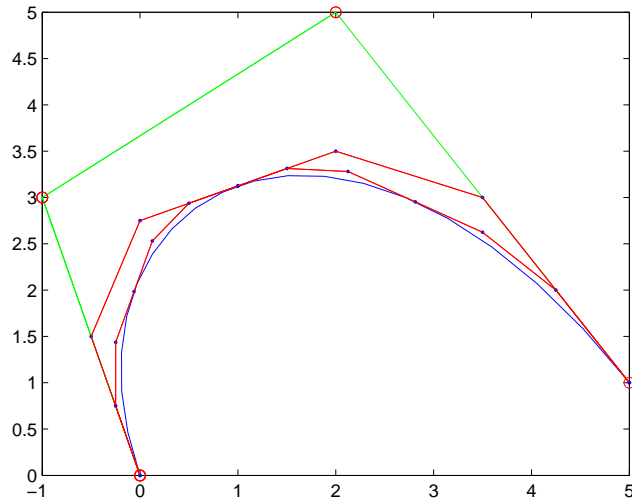


Figura B.15: Subdivisión

$$\begin{aligned}
 &= \sum_{i=0}^n B_i \left(\frac{u}{c} \right) \left[\sum_{k=0}^i P_k B_k^i(c) \right] \\
 &R_i \sum_{k=0}^i P_k B_k^i(c) \tag{B.25}
 \end{aligned}$$

Se observa que R_i es la curva de Bézier correspondiente a los puntos de control P_0, P_1, \dots, P_n , calculada en el punto de subdivisión R_i es el primero de los puntos calculados en la i -ésima etapa del algoritmo de Casteljaou.

$$R_i = P_0^{(0)} \quad i = 0, 1, \dots, n$$

Se puede comprobar a sí mismo, que los puntos de la última fila de esta tabla son los puntos de control del segundo tramo, en el que queda subdividida la curva $C(u)$, por lo tanto la formulación de Casteljaou al punto c de subdivisión en el espacio paramétrico, proporciona los puntos de control de las nuevas curvas en las que se subdividió la curva $C(u)$

$$Q_i = P_I^{(n-i)} \quad i = 0, 1, \dots, n$$

Ejemplo:

Partiendo de la curva de Bézier.

$$C_u = (1-u)3(1,2) + 3u(1-u)^2(3,5) + 3u^2(1-u)(6,6) + (7,1)u^3$$

Donde los puntos de control son $P_0 = (1,2); P_1 = (3,5); P_2 = (6,6); P_3 = (7,1);$ se escoge un nodo de subdivisión $c = \frac{1}{3}$

$$P_0^{(1)} = (1 - \frac{1}{3})P_0 + \frac{1}{3}P_1 = \frac{2}{3}(1, 2) + \frac{1}{3}(3, 5) = (\frac{5}{3}, 3)$$

$$P_1^{(1)} = (1 - \frac{1}{3})P_1 + \frac{1}{3}P_2 = \frac{2}{3}(3, 5) + \frac{1}{3}(6, 6) = (4, \frac{16}{3})$$

$$P_2^{(1)} = (1 - \frac{1}{3})P_2 + \frac{1}{3}P_3 = \frac{2}{3}(6, 6) + \frac{1}{3}(7, 1) = (\frac{19}{3}, \frac{13}{3})$$

$$P_0^{(2)} = (1 - \frac{1}{3})P_0^{(1)} + \frac{1}{3}P_1^{(1)} = \frac{2}{3}(\frac{5}{3}, 3) + \frac{1}{3}(4, \frac{16}{3}) = (\frac{22}{9}, \frac{34}{9})$$

$$P_1^{(2)} = (1 - \frac{1}{3})P_1^{(1)} + \frac{1}{3}P_2^{(1)} = \frac{2}{3}(4, \frac{16}{3}) + \frac{1}{3}(\frac{19}{3}, \frac{13}{3}) = (\frac{43}{9}, 5)$$

$$P_0^{(3)} = (1 - \frac{1}{3})P_0^{(2)} + \frac{1}{3}P_1^{(2)} = \frac{2}{3}(\frac{22}{9}, \frac{34}{9}) + \frac{1}{3}(\frac{43}{9}, 5) = (\frac{87}{27}, \frac{113}{27})$$

Para lo cuál $(\frac{87}{27}, \frac{113}{27})$ es el punto sobre la curva que subdividirá la curva C_u , quedando definida en dos tramos representados por los puntos de control correspondientes así:

Para el primer tramo

$$P_0 = (1, 2)$$

$$P_0^{(1)} = (\frac{5}{3}, 3)$$

$$P_0^{(2)} = (\frac{22}{9}, \frac{34}{9})$$

$$P_0^{(3)} = (\frac{87}{27}, \frac{113}{27})$$

Para el segundo tramo,

$$P_0^{(3)} = (\frac{87}{27}, \frac{113}{27})$$

$$P_1^{(2)} = (\frac{43}{9}, 5)$$

$$P_2^{(1)} = (\frac{19}{3}, \frac{13}{3})$$

$$P_3 = (7, 1)$$

Código en MATLAB

```
c=0.5                                % PUNTO DENTRO DEL SOPORTE
o=(length(xy)-1)/3
```

```

for i=0:o-1                                % CASTEJAN
a1x=xy(3*i+1,1)*(1-c)+c*xy(3*i+2,1)
a2x=xy(3*i+2,1)*(1-c)+c*xy(3*i+3,1)
a3x=xy(3*i+3,1)*(1-c)+c*xy(3*i+4,1)
a4x=a1x*(1-c)+c*a2x
a5x=a2x*(1-c)+c*a3x
a6x=a4x*(1-c)+c*a5x

% SE UBICAN LOS PUNTOS EN LA NUEVA MATRIZ

Pxy(6*i+1,1)=xy(3*i+1,1)
Pxy(6*i+2,1)=a1x
Pxy(6*i+3,1)=a4x
Pxy(6*i+4,1)=a6x
Pxy(6*i+5,1)=a5x
Pxy(6*i+6,1)=a3x
Pxy(6*i+7,1)=xy(3*i+4,1)
a1y=xy(3*i+1,2)*(1-c)+c*xy(3*i+2,2)
a2y=xy(3*i+2,2)*(1-c)+c*xy(3*i+3,2)
a3y=xy(3*i+3,2)*(1-c)+c*xy(3*i+4,2)
a4y=a1y*(1-c)+c*a2y
a5y=a2y*(1-c)+c*a3y
a6y=a4y*(1-c)+c*a5y
Pxy(6*i+1,2)=xy(3*i+1,2)
Pxy(6*i+2,2)=a1y
Pxy(6*i+3,2)=a4y
Pxy(6*i+4,2)=a6y
Pxy(6*i+5,2)=a5y
Pxy(6*i+6,2)=a3y
Pxy(6*i+7,2)=xy(3*i+4,2)

% GUARDAR LOS NUEVOS PTOS DE CTROL

for u=1:7
pxc(u)=Pxy(6*i+u,1)

```

```

pyc(u)=Pxy(6*i+u,2)
end plot(pxc,pyc,'r')
plot(pxc,pyc,'b. ')
end x=Pxy(:,1)
y=Pxy(:,2)
plot(x,y,'r') %,'y*')
xy=Pxy
% SE RENOMBRAN LOS PUNTOS DE CONTROL XY

```

B.3.2. Control local de forma

Existen 3 casos para los cuales se expone el comportamiento de la curva debido a la modificación de uno de los puntos de control, siendo una curva de Bézier la unión de dos o mas curvas de Bézier simples las cuales cumplen las condiciones de continuidad y colinealidad de los puntos ubicados en las tangentes de los extremos de las curvas simples, cumpliendo la continuidad geométrica o G^1 sobre cualquier intervalo $[a, b]$ y con una partición arbitraria o soporte de nodos.

Dependiendo del tipo de punto modificado:

1. Punto de unión de los dos tramos $P_0^{i+1} = P_n^i$.
2. Penúltimo punto de un tramo, excepto el tramo $m : P_{n-1}^i$.
3. Segundo punto de un tramo, excepto el tramo 1 : P_1^{i+1} .
4. Resto de los puntos.

En los tres primeros casos la modificación de uno de esos puntos obliga al re-posicionamiento de por lo menos otro punto,

- Si el punto que se modifica es la unión de dos tramos $P_0^{(i+1)} = P_n^i$, es decir se trata de un punto de interpolación, se trasladan los puntos anterior y posterior, por medio de un vector de translación, recuerde que estos puntos son colineales, es decir están ubicados sobre un mismo vector y para mover el punto de interpolación se le debe sumar un vector que traslade totalmente el vector sin perder la condición de colinealidad.

$$\begin{aligned}\bar{P}_0^{(i+1)} &= P_0^{(i+1)} + h \\ \bar{P}_n^{(i)} &= P_n^{(i)} + h = \bar{P}_0^{(i+1)}\end{aligned}$$

Modificando el punto anterior y el siguiente se obtiene:

$$\begin{aligned}\bar{P}_{n-1}^{(i)} &= P_{(n+1)}^{(i)} + h \\ \bar{P}_1^{(i+1)} &= P_n^{(i)} + h = \bar{P}_0^{(i+1)}\end{aligned}$$

- Si se modifica el penúltimo punto del tramo $P_{n-1}^{(i)}$, esto obligará a mover el segundo punto del siguiente tramo, $P_1^{(i+1)}$ (excepto si se trata del ultimo tramo, esto es $i = m$, de forma que se mantenga alineado con $\bar{P}_{(n-1)}$ y $P_n^{(i)}$ (que permanece invariable):

$$\bar{P}_{n-1}^{(i)} = P_n^{(i)} \frac{\|\bar{P}_1^{(i+1)} - \bar{P}_0^{(i+1)}\|}{\|\bar{P}_n^{(i)} - \bar{P}_{(n-i)}^{(i)}\|} [P_n^{(i)} - \bar{P}_{n-i}^{(i)}]$$

- Si se modifica el segundo punto de un tramo, $P_1^{(n+1)}$, igual que en el caso anterior debe modificarse el penúltimo punto del tramo anterior, para que se mantengan alineados, de esta forma:

$$\bar{P}_1^{(i+1)} = P_n^{(i)} \frac{\|\bar{P}_1^{(i+1)} - \bar{P}_0^{(i+1)}\|}{\|\bar{P}_n^{(i)} - \bar{P}_{(n-i)}^{(i)}\|} [P_n^{(i)} - \bar{P}_{n-i}^{(i)}]$$

- El movimiento de uno de los puntos no mencionados anteriormente, no conlleva al re posicionamiento de de ningún otro, ni tampoco implica cambios en el soporte, por lo que puede ser realizado sin ningún calculo adicional.

B.3.3. Interpolantes Cúbicos Compuestos.

Se observó anteriormente la utilidad que prestan las curvas de Bézier en la labor de modelar curvas, a su vez se mostró la función de los puntos de control en las connotaciones geométricas que tendrá la curva, pero también quedó claro que para la definición de dichas curvas es necesario definir primero los puntos de control que la componen, siendo esta ultima condición una de sus principales desventajas, pues se debería ser un experto dibujante, para saber exactamente donde posicionar los puntos de control que definan la curva que describe la trayectoria deseada.

Por el contrario una herramienta eficiente para realizar el modelado de curvas, de manera interactiva debería permitir el trazo de la misma a partir de los puntos por donde se desee que esta pase, y que el sistema calculara automáticamente los puntos de control que definen la curva deseada.

Para ello se desarrollará un método que permita dibujar las curvas de Bézier a partir de los puntos de interpolación y generar los puntos de control internos de la manera mas apropiada para describir una trayectoria los mas aproximada posible a los deseos del modelador. Para ello solo se trabajará con curvas de tercer grado, de esta manera una curva de Bézier Cúbica de m tramos estará definida de la siguiente manera:

$$\begin{array}{ll}
 P_0, P_1, P_2, P_3 & \text{tramo 1} \\
 P_3, P_4, P_5, P_6 & \text{tramo 2} \\
 P_{3i-3}, P_{3i-2}, P_{3i-1}, P_{3i} & \text{tramo } i \\
 P_{3m-3}, P_{3m-2}, P_{3m-1}, P_{3m} & \text{tramo } m
 \end{array}$$

De esta forma el i -ésimo termino de la curva de Bézier es:

$$C_{(u)} = (1 - u)^3 P_{3i-3} + 3u(1 - u)^2 P_{3i-2} + 3u^2(1 - u) P_{3i-1} + P_{3i} u^3$$

Sean entonces los puntos de interpolación R_0, \dots, R_n de la curva de Bézier es decir que $R_i = P_{3i}$ $i = 1, \dots, m - 1$. Los puntos de control interiores, exceptuando el primero del primer tramo y el segundo del ultimo tramo deben guardar colinealidad con el respectivo punto de interpolación.

$$P_{3i-1}, P_{3i} = R_i, P_{3m+1} \quad \text{alineados para } i = 1, \dots, m - 1$$

B.3.3.1. Interpolante Cúbico de Hermite

Este es el nombre que recibe la curva que se acaba de describir y que cumple las condiciones mencionadas, es decir esta definida por los puntos de interpolación y la colinealidad del punto anterior y siguiente a dicho punto de interpolación, para ello se

definen las derivadas para cada uno de estos puntos, estas derivadas son estimadas por el programador.

Recordando la ecuación para hallar las derivadas de los puntos extremos de una curva de Bézier,

$$C'(u_i) = \frac{3}{u_i - u_{i-1}}(P_{3i} - P_{3i-1}) = \frac{3}{u_{i+1} - u_i}(P_{3i+1} - P_{3i})$$

De esta expresión se deducen las siguientes ecuaciones para encontrar los puntos de control internos a la curva de Bézier que describe la trayectoria marcada para los puntos de interpolación deseados.

$$\begin{aligned} P_{3i} &= R_i & i=0, \dots, m-1 \\ P_{3i-1} &= R_i - \frac{u_i - u_{i-1}}{3} D_i & i=0, \dots, m-1 \\ P_{3i+1} &= R_i - \frac{u_{i+1} - u_i}{3} D_i & i=0, \dots, m-1 \end{aligned}$$

De esta manera se pueden calcular todos los puntos de control internos a las curvas de Bézier cúbicas que se están generando a partir de los puntos de interpolación y las derivadas estimada para cada punto.

B.3.3.2. Estimación de las derivadas

Para encontrar las derivadas debemos suponer el soporte de interpolación conocido y este a su vez es dependiente de los puntos de interpolación, para ello se usará la elección de soporte centripetal que fue el que arrojó los mejores resultados para el proceso de interpolación mencionado en el capítulo anterior.

Para la estimación de las derivadas se usaran propondrán los siguientes métodos

- Derivada de Catmull-Rown

Se trata de tomar como tangente un promedio de los vectores que unen los puntos de interpolación, aunque este método proporciona una buena aproximación, se debe realizar una corrección en función del soporte.

la expresión para encontrar la derivada por este método esta dada por:

$$D_i = \frac{R_{i+1} - R_{i-1}}{u_{i+1} - u_{i-1}}$$

- Interpolante de Bessel

Consiste en la construcción de una parábola que una los tres puntos de interpolación R_{i-1} , R_i , R_{i+1} , y a partir de ella tomar una aproximación para las tangentes de cada punto, esta dada por la expresión;

$$D_i = (1 - \alpha_i) \frac{R_{i+1} - R_i}{u_{i+1} - u_i} + \alpha_i \frac{R_i - R_{i-1}}{u_i - u_{i-1}} \text{ donde } \alpha_i = \frac{u_i - u_{i-1}}{u_{i+1} - u_{i-1}}$$

- Interpolación Akima

$$D_m = 2 \frac{R_i - R_{i-1}}{u_i - u_{m-1}} - D_{m-1}$$

El cálculo de la derivada del primer punto esta dada por la expresión:

$$Q'_1 = \frac{u_1 - u_0}{u_2 - u_0} \frac{R_2 - R_1}{u_2 - u_1} + \frac{u_2 - u_1}{u_2 - u_0} \frac{R_1 - R_0}{u_1 - u_0}$$

Apéndice C

Curvas Spline

La mayor dificultad de trabajar con Curvas de Bézier se encuentra en la manipulación de los puntos de control aunque se siguen manteniendo las condiciones de continuidad en los puntos de unión de tramos consecutivos. Entonces, para superar esta dificultad, la primera cuestión sería determinar cuantos parámetros (grados de libertad) quedan tras satisfacer las condiciones de diferenciabilidad. A continuación, encontrar el tipo de dato que debe proporcionarse, en igual número al de parámetros disponibles, y finalmente, dejar que las correspondientes fórmulas calculen los puntos de control de la curva de Bézier.

Bastaría por tanto, encontrar $2m+2$ curvas básicas y expresar cualquier otra curva como combinación lineal de ellas. Consiguiendo ésto se habría logrado representar la curva justo por el número de coeficientes necesitados y en consecuencia, se habría evitado la formulación de Bézier.

En este capítulo se hará el estudio y la construcción de bases que desliguen las condiciones de diferenciabilidad de los coeficientes y que incorpore estas condiciones a la propia definición de la base. Este estudio permitirá construir una curva polinomial compuesta a través de un número restringido de coeficientes cuya manipulación arbitraria no influirá en la diferenciabilidad de la curva de cada nodo.

C.1. Spline cúbico de interpolación

Los splines son curvas polinómicas por trozos continuamente diferenciables hasta un orden prescrito. El ejemplo más sencillo es el spline C^0 , o sea, lineal por trozos. Este spline es simplemente una poligonal en el plano o en el espacio.

El nombre “spline” es una palabra en idioma inglés que significa “listón elástico”. Estos

listones eran usados para crear curvas, que describen superficies a construir, como cascos de barcos y fusilajes de aviones. Constrained por pesos, estos listones elásticos o splines asumen una forma que minimiza su energía elástica, propiedad que heredan en forma aproximada los splines matemáticos C^2 de grado tres.

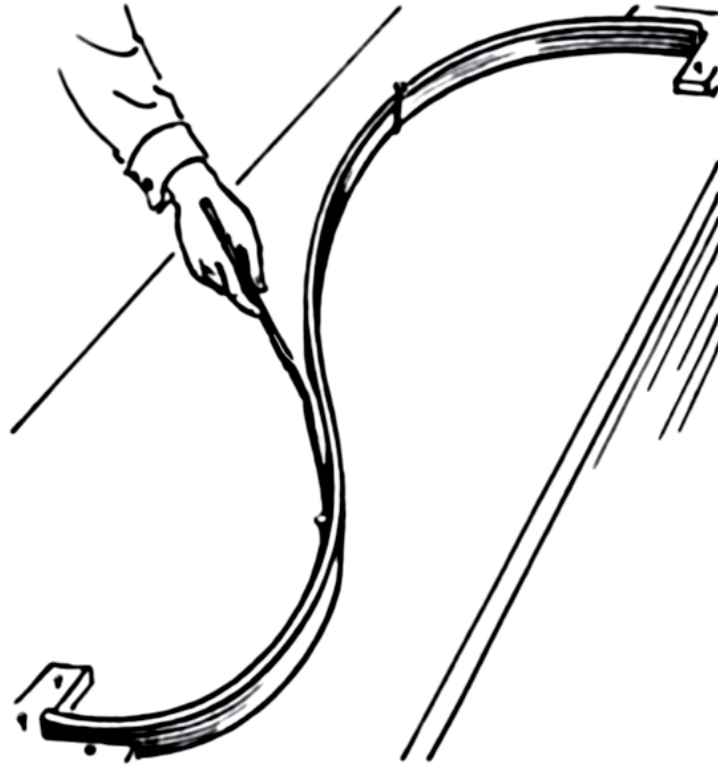


Figura C.1: Listón elástico. Fuente: http://en.wikipedia.org/wiki/Flat_spline

La herramienta de los splines se desarrolla para solventar las limitaciones de las curvas de Bézier: falta de control local, la laboriosidad requerida para imponer continuidad C^2 y el hecho de que el número de puntos de control de una curva de Bézier impone su grado.

Considerando un soporte de nodos:

$$S = \{u_0, \dots, u_m\}$$

y un conjunto de puntos de interpolación Q_0, \dots, Q_m . El soporte S es fijo y especificado de antemano, antes de concretar los puntos de interpolación Q_i .

Si además se conoce la derivada D_i en cada punto de interpolación Q_i , se podría obtener

una curva interpoladora $C(u)$ de clase $C^1[u_0, u_m]$. Sin embargo, ahora no interesa proporcionar D_i como dato, sino imponer que la curva sea de clase C^2 en $[u_0, u_m]$.

Para ello, se estudia la derivada segunda de una curva de Bézier a izquierda y derecha de cada nodo u_i , estableciendo que ambas sean iguales. Entonces se tiene:

$$\frac{P_{3i} - 2P_{3i-1} + P_{3i-2}}{(u_i - u_{i-1})^2} = \frac{P_{3i+2} - 2P_{3i+1} + P_{3i}}{(u_{i+1} - u_i)^2} \quad i = 1, \dots, m - 1 \quad (\text{C.1})$$

Utilizando las expresiones que relacionan a los puntos de control con los puntos de interpolación y las derivadas en ellos:

$$\begin{aligned} P_{3i} &= Q_i \\ P_{3i-1} &= Q_i - \frac{u_i - u_{i-1}}{3} D_i \\ P_{3i-2} &= Q_{i-1} + \frac{u_i - u_{i-1}}{3} D_{i-1} \\ P_{3i+2} &= Q_{i+1} - \frac{u_{i+1} - u_i}{3} D_{i+1} \\ P_{3i+1} &= Q_i + \frac{u_{i+1} - u_i}{3} D_i \end{aligned}$$

con lo cual, sustituyendo:

$$h_i D_{i-1} + 2(h_{i-1} + h_i) D_i + h_{i-1} D_{i+1} = 3 \left[\frac{h_i}{h_{i-1}} (Q_i - Q_{i-1}) + \frac{h_{i-1}}{h_i} (Q_{i+1} - Q_i) \right] \quad (\text{C.2})$$

donde $i = 1, \dots, m - 1$ y $h_i = u_{i+1} - u_i$.

Como se puede ver esta relación expresa la continuidad de $C''(u)$ no es en función de los puntos de control, sino que impone una restricción sobre cada tres derivadas primeras consecutivas. Ahora, se puede convertir este sistema en uno cuadrado con lo que queda en forma matricial

$$AB = C \quad (\text{C.3})$$

donde:

$$A = \begin{bmatrix} 2(h_0 + h_1) & h_0 & 0 & \dots & \dots & 0 \\ h_2 & 2(h_1 + h_2) & h_1 & & & \vdots \\ 0 & h_3 & 2(h_2 + h_3) & h_2 & & \vdots \\ \vdots & & & \ddots & & 0 \\ \vdots & & & & h_{m-2} & 2(h_{m-3} + h_{m-2}) & h_{m-3} \\ 0 & \dots & \dots & 0 & h_{m-1} & 2(h_{m-2} + h_{m-1}) \end{bmatrix}$$

$$B = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_{m-2} \\ D_{m-1} \end{bmatrix} \quad C = \begin{bmatrix} b_1 - h_1 D_0 \\ b_2 \\ b_3 \\ \vdots \\ b_{m-2} \\ b_{m-1} - h_{m-2} D_m \end{bmatrix}$$

con:

$$b_i = 3 \left[\frac{h_i}{h_{i-1}} (Q_i - Q_{i-1}) + \frac{h_{i-1}}{h_i} (Q_{i+1} - Q_i) \right] \quad (\text{C.4})$$

Esta matriz es triangular y de diagonal dominante. Es tipo de matrices son conocidas como *matrices de Hadamard*. Con esto se ha probado que existe una curva polinomial compuesta, de tercer grado, y de clase $C^2[u_0, u_m]$. Más aún, dado que D_0 y D_m pueden especificarse, tenemos que en realidad existen infinitas de tales curvas.

Ejemplo:

Considerando el soporte $S = \{0, 1, 2, 3, 4\}$ los puntos de interpolación $Q_0 = (0, 0)$, $Q_1 = (1, 2)$, $Q_2 = (2, 1)$, $Q_3 = (3, 1)$, $Q_4 = (4, 0)$. Las derivadas en los puntos extremos son $D_0 = [1, 0]$, $D_m = [1, -2]$.

Tomando en cuenta que $h_i = u_{i+1} - u_i = 1 \forall i$. El sistema queda de la siguiente manera:

$$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} = \begin{bmatrix} b_1 - D_0 \\ b_2 \\ b_3 - D_4 \end{bmatrix}$$

Ahora se hallan los valores de b_i :

$$b_1 = 3[(2, 1) - (0, 0)] = (6, 3)$$

$$b_2 = 3[(3, 1) - (1, 2)] = (6, -3)$$

$$b_3 = 3[(4, 0) - (2, 1)] = (6, -3)$$

También se halla:

$$b_1 - D_0 = (6, 3) - [1, 0] = (5, 3)$$

$$b_3 - D_4 = (6, -3) - [1, 2] = (5, -1)$$

y el sistema es:

$$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} = \begin{bmatrix} 5 & 3 \\ 6 & -3 \\ 5 & -1 \end{bmatrix}$$

Resolviendo se encuentran las tangentes intermedias:

$$D_1 = [1, 1] \quad D_2 = [1, -1] \quad D_3 = [1, 0]$$

Las pendientes juntos con los puntos de interpolación definen unívocamente la curva cúbica de clase 2. En la figura C.2 se muestra el spline para el ejemplo.

Algoritmo 4.01. Spline Cúbico definiendo derivadas primeras en los extremos:

```
clear all
clc
hold on
n=5; %NODOS
u3=[0 1 2 3 4]           %SOPORTE
                           % DERIVADA INICIAL Y FINAL DADAS

deriv(5,1)=1;
deriv(5,2)=-2;
```

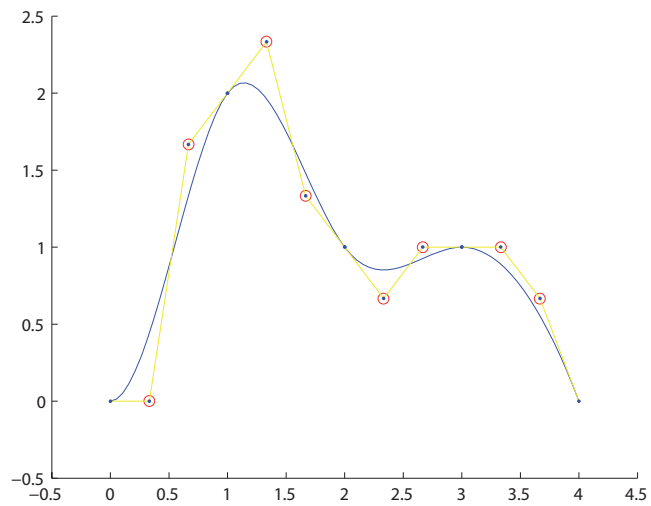


Figura C.2: Spline cúbico completo de interpolación

```

deriv(1,1)=1;
deriv(1,2)=0;
% PUNTOS DE INTERPOLACIÓN DADOS
Q=[
0 0;
1 2;
2 1;
3 1;
4 0
]
for i=1:n-2
a(i,i)=2*(u3(i+2)-u3(i));
if i<n-2
a(i,i+1)=u3(i+1)-u3(i);
end
if i>1
a(i,i-1)=u3(i+2)-u3(i+1);
end
end
% EL PRIMER DATO ES bi - hiDo,

```

```

EL ULTIMO DATO ES bn-1 - hn-2Dn

C(1,1)=3*((u3(3)-u3(2))/(u3(2)-u3(1))*(Q(2,1)-Q(1,1))
+(u3(2)-u3(1))/(u3(3)-u3(2))*(Q(3,1)-Q(2,1)))-(u3(3)-u3(2))*deriv(1,1);
C(1,2)=3*((u3(3)-u3(2))/(u3(2)-u3(1))*(Q(2,2)-Q(1,2))
+(u3(2)-u3(1))/(u3(3)-u3(2))*(Q(3,2)-Q(2,2)))-(u3(3)-u3(2))*deriv(1,2);

% LOS DEMÁS DATOS SON bi

for i=2:n-3
C(i,1)=3*((u3(i+2)-u3(i+1))/(u3(i+1)-u3(i))*(Q(i+1,1)-Q(i,1))
+(u3(i+1)-u3(i))/(u3(i+2)-u3(i+1))*(Q(i+2,1)-Q(i+1,1)));
C(i,2)=3*((u3(i+2)-u3(i+1))/(u3(i+1)-u3(i))*(Q(i+1,2)-Q(i,2))
+(u3(i+1)-u3(i))/(u3(i+2)-u3(i+1))*(Q(i+2,2)-Q(i+1,2)));
end
C(n-2,1)=3*((u3(n)-u3(n-1))/(u3(n-1)-u3(n-2))*(Q(n-1,1)-Q(n-2,1))
+(u3(n-1)-u3(n-2))/(u3(n)-u3(n-1))*(Q(n,1)-Q(n-1,1)))-
(u3(n)-u3(n-1))*deriv(n,1);
C(n-2,2)=3*((u3(n)-u3(n-1))/(u3(n-1)-u3(n-2))*(Q(n-1,2)-Q(n-2,2))
+(u3(n-1)-u3(n-2))/(u3(n)-u3(n-1))*(Q(n,2)-Q(n-1,2)))
-(u3(n)-u3(n-1))*deriv(n,2);
D=inv(a)*C;
%SE AÑADEN A LA MATRIZ DE DERIVADAS
for i=2:n-1
deriv(i,1)=D(i-1,1);
deriv(i,2)=D(i-1,2);
end
% INTERPOLACIÓN CUBICA COMPUESTA SIN CALCULO DE DERIVADAS
for i=0:n-1
% EN UN PUNTO DE INTERPOLACIÓN
xy(3*i+1,1)=Q(i+1,1);
xy(3*i+1,2)=Q(i+1,2);
plot(xy(3*i+1,1),xy(3*i+1,2),'g')
if i>0
% DESPUES DE UN PUNTO DE INTERPOLACIÓN

```

```

xy(3*i,1)=Q(i+1,1)-(u3(i+1)-u3(i))/3*deriv(i+1,1);
xy(3*i,2)=Q(i+1,2)-(u3(i+1)-u3(i))/3*deriv(i+1,2);
plot(xy(3*i,1),xy(3*i,2),'ro')
plot(xy(3*i,1),xy(3*i,2),'g')
end
if i < n-1
% ANTES DE UN PUNTO DE INTERPOLACIÓN
xy(3*i+2,1)=Q(i+1,1)+(u3(i+2)-u3(i+1))/3*deriv(i+1,1);
xy(3*i+2,2)=Q(i+1,2)+(u3(i+2)-u3(i+1))/3*deriv(i+1,2);
plot(xy(3*i+2,1),xy(3*i+2,2),'ro')
plot(xy(3*i+2,1),xy(3*i+2,2),'g')
end
end

% PARA DIBUJAR CUALQUIER TIPO DE DATO XY
matriz=[-1 3 -3 1;
3 -6 3 0;
-3 3 0 0;
1 0 0 0];
x=linspace(0,1,20);
n=length(xy);
for o=0:(n-1)/3-1
cx=0;cy=0;
for i=1:4
cx=cx+xy(3*o+i,1)*matriz(i,:);
cy=cy+xy(3*o+i,2)*matriz(i,:);
end
Y=polyval(cx,x);
X=polyval(cy,x);
plot(Y,X)
for i=1:4
xc(i)=xy(3*o+i,1);
yc(i)=xy(3*o+i,2);
end
plot(xc,yc,'y')

```

```

plot(xc,yc,'b. ')
axis ([-0.5 4.5 -0.5 2.5])
end

```

C.1.1. Spline cúbico completo de interpolación

Dados $m + 1$ puntos de interpolación y un soporte asociado, aún se puede especificar dos condiciones más para tener el sistema de ecuaciones no singular que proporcione unívocamente, a través de las derivadas intermedias, el spline cúbico de clase C^2 . El spline así constituido suele denominarse *spline cúbico completo de interpolación* y posee la propiedad de ser la curva que menos oscila entre todas las de la clase C^2 .

C.1.2. El spline natural

Ahora se verá que pasa si en lugar de especificar derivadas primeras en los extremos se puede hacer lo mismo con las derivadas segundas en los extremos.

El planteamiento que se hace ahora será proporcionar dos ecuaciones más en la que intervengan las derivadas segundas en los extremos.

Partiendo de:

$$C'''(u_0) = 6 \frac{P_2 - 2P_1 + P_0}{(u_1 - u_0)^2}$$

$$C'''(u_m) = 6 \frac{P_{3m} - 2P_{3m-1} + P_{3m-2}}{(u_m - u_{m-1})^2}$$

Expresando los puntos de control P_i involucrados en función de Q_i y D_i :

$$P_0 = Q_0$$

$$P_1 = Q_0 + \frac{u_1 - u_0}{3} D_0$$

$$P_2 = Q_1 - \frac{u_1 - u_0}{3} D_1$$

$$P_{3m-2} = Q_{m-1} - \frac{u_m - u_{m-1}}{3} D_{m-1}$$

$$P_{3m-1} = Q_m + \frac{u_m - u_{m-1}}{3} D_m$$

$$P_m = Q_m$$

donde:

$$\begin{aligned}
 b_0 &= 3 \frac{Q_1 - Q_0}{h_0} - \frac{h_0}{2} C''(u_0) \\
 b_m &= 3 \frac{Q_m - Q_{m-1}}{h_{m-1}} + \frac{h_{m-1}}{2} C''(u_m) \\
 b_i &= 3 \left[\frac{h_i}{h_{i-1}} (Q_i - Q_{i-1}) + \frac{h_{i-1}}{h_i} (Q_{i+1} - Q_i) \right]
 \end{aligned}
 \tag{C.5}$$

En el caso particular $C''(u_0) = C''(u_m) = 0$ se obtiene el spline conocido como *spline cúbico natural*. Este caso implica que el spline es casi recto en las proximidades de los extremos. Para el ejemplo visto anteriormente el spline natural es el mostrado en la figura C.3.

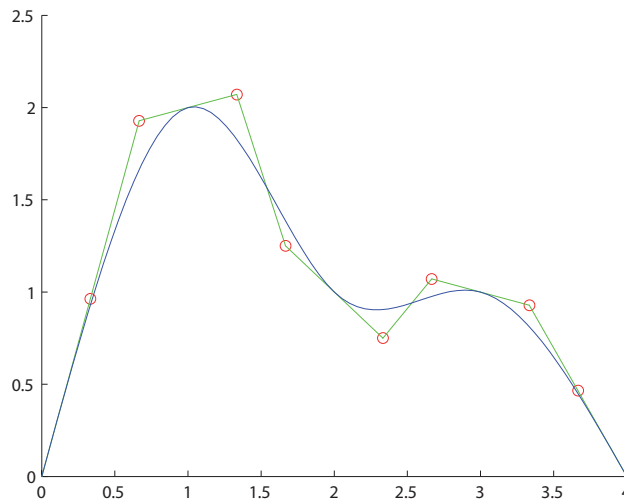


Figura C.3: Spline natural

Algoritmo 4.02. Spline Cúbico definiendo derivadas segundas en los extremos.

```

clear all
close all
clc

```

```

axis([-1,11,-1,10])
hold on
n=5
C1=0 % 2da derivada del 1er punto
Cn=0 % 2da derivada del ultimo punto
u3=(0:n-1); % soporte de paso 1 con n datos
A(1,1)=2;A(n,n)=2;A(1,2)=1;A(n,n-1)=1; % 1er y ultima fila
for i=2:n-1 % desde la 2 a la penultima fila
A(i,i)=2*(u3(i+1)-u3(i-1));
A(i,i+1)=u3(i)-u3(i-1);
A(i,i-1)=u3(i+1)-u3(i);
end
Q=[0 0;1 2;2 1;3 1;4 0]
C(1,1)=3*(Q(2,1)-Q(1,1))/(u3(2)-u3(1))-(u3(2)-u3(1))/2*C1;
C(1,2)=3*(Q(2,2)-Q(1,2))/(u3(2)-u3(1))-(u3(2)-u3(1))/2*C1;
C(n,1)=3*(Q(n,1)-Q(n-1,1))/(u3(n)-u3(n-1))-(u3(n)-u3(n-1))/2*Cn;
C(n,2)=3*(Q(n,2)-Q(n-1,2))/(u3(n)-u3(n-1))-(u3(n)-u3(n-1))/2*Cn;
for i=2:n-1
C(i,1)=3*((u3(i+1)-u3(i))/(u3(i)-u3(i-1))*(Q(i,1)-Q(i-1,1))
+(u3(i)-u3(i-1))/(u3(i+1)-u3(i))*(Q(i+1,1)-Q(i,1)));
C(i,2)=3*((u3(i+1)-u3(i))/(u3(i)-u3(i-1))*(Q(i,2)-Q(i-1,2))
+(u3(i)-u3(i-1))/(u3(i+1)-u3(i))*(Q(i+1,2)-Q(i,2)));
end
B=inv(A)*C;

% SE CONSTRUYE LOS PUNTOS DE CONTROL Y SE DIBUJA EL POLIGONO DE CONTROL

for i=0:n-1
p(3*i+1,1)=Q(i+1,1);
p(3*i+1,2)=Q(i+1,2);
plot(p(3*i+1,1),p(3*i+1,2),'g');
if i>0
p(3*i,1)=Q(i+1,1)-(u3(i+1)-u3(i))/3*B(i+1,1);
p(3*i,2)=Q(i+1,2)-(u3(i+1)-u3(i))/3*B(i+1,2);
plot(p(3*i,1),p(3*i,2),'ro')

```

```

plot(p(3*i,1),p(3*i,2), 'g')
end
if i < n-1
p(3*i+2,1)=Q(i+1,1)+(u3(i+2)-u3(i+1))/3*B(i+1,1);
p(3*i+2,2)=Q(i+1,2)+(u3(i+2)-u3(i+1))/3*B(i+1,2);
plot(p(3*i+2,1),p(3*i+2,2), 'ro')
plot(p(3*i+2,1),p(3*i+2,2), 'g')
end
end
plot(p(:,1),p(:,2), 'g')

% solo los puntos de control se recalculan las curvas compuestas
% para dibujar cualquier tipo de datos xy

xy=p;
matriz=[-1 3 -3 1;
3 -6 3 0;
-3 3 0 0;
1 0 0 0];
x=linspace(0,1,20);
n=length(xy);
for o=0:(n-1)/3-1
cx=0;cy=0;
for i=1:4
cx=cx+xy(3*o+i,1)*matriz(i,:);
cy=cy+xy(3*o+i,2)*matriz(i,:);
end
Y=polyval(cx,x);
X=polyval(cy,x);
plot(Y,X)
end

```

C.1.3. Base de potencias truncadas

Con el objeto de generalizar el spline cúbico, se va a considerar de una forma más abstracta. Para ello, se utilizará la representación explícita, ya que es la más conveniente.

Sea un soporte de nodos $S = \{x_0, \dots, x_m\}$ con $x_0 < x_1 < \dots < x_m$. Un spline cúbico asociado al soporte es cualquier curva $y = C(x)$ de clase C^2 en $[x_0, x_m]$ y tal que, restringida a cada intervalo $[x_i, x_{i+1}]$, es un polinomio de tercer grado (o menor). Se puede comprobar que, si se considera el conjunto de todos los splines cúbicos asociados al mismo soporte S con $m + 1$ nodos, este conjunto es un espacio vectorial.

Entonces, considerando el espacio vectorial S_3 de los splines cúbicos asociados al soporte $S = \{x_0, \dots, x_m\}$, su dimensión es $N = m + 3$, pues un miembro de este espacio está determinado por $m + 3$ parámetros independientes.

Ahora se puede elegir una base del espacio constituida por $m + 3$ splines en concreto, $\{B_0(x), \dots, B_{m+2}(x)\}$ y cualquier spline en S_3 quedaría expresado como combinación lineal de los $B_k(x)$:

$$C(x) = \sum_{k=0}^{m+2} p_k B_k(x)$$

Con esto, el conjunto de los polinomios de grado menor o igual que tres es un subespacio propio de S_3 , por lo que todo polinomio de grado tres es un spline cúbico. Al tratarse de polinomios se eligen bases sencillas para estudiar sus propiedades. En este caso, una base de este tipo es la que se denomina base de potencias truncadas, constituidas por funciones de la forma:

$$y \equiv (x - x_k)_+^3 = \begin{cases} 0 & x \in (-\infty, x_k] \\ (x - x_k)^3 & x \in (x_k, \infty) \end{cases} \quad (\text{C.6})$$

considerando solo la parte positiva de la función $(x - x_k)^3$. Esta curva es continua y dos veces diferenciable. La figura C.4 muestra un ejemplo de este tipo de funciones.

Dos cualesquiera de estas funciones, $(x - x_i)_+^3$ y $(x - x_j)_+^3$, en un intervalo en el que ambas sean no nulas, son linealmente independientes. Por tanto, eligiendo de forma apropiada $m + 3$ potencias truncadas, se dispondrá de una base para S_3 . Por ejemplo, podemos considerar:

$$\{(x - x_{-1})_+^3, (x - x_{-2})_+^3, (x - x_{-3})_+^3, (x - x_0)_+^3, (x - x_1)_+^3, \dots, (x - x_{m-1})_+^3\}$$

que son $m + 3$ funciones linealmente independientes en $[x_0, x_m]$. Con lo cual un spline

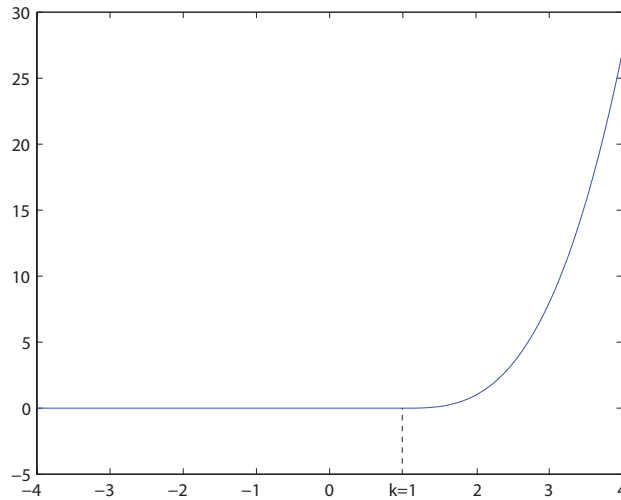


Figura C.4: Potencia truncada

cúbico quedaría representado como combinación lineal de $m + 3$ potencias truncadas:

$$C(x) = \sum_{k=-3}^{m-1} p_k (x - x_k)_+^3 \quad (\text{C.7})$$

La base de potencias es una herramienta teórica cuya generalización permitirá construir finalmente la clase de splines que se pretenden en este capítulo.

C.1.4. Base B-Splines

Cuál es el spline cúbico que puede construirse anulando e mayor número posible de subintervalos de $[x_0, x_m]$. La resultante función es lo que se denomina un spline básico o B-spline, y la figura C.5 muestra un ejemplo de uno de ellos:

En primer lugar, se verá el número mínimo de subintervalos en los que un spline cúbico puede no ser nulo. Para ello, se considera el primero y el último tramo del intervalo no nulo de la curva y la idea será unirlos primero con un tramo, y luego, unirlos con dos tramos.

Primer tramo no nulo (polinomio de tercer grado): $P_i(x) = \lambda(x - x_i)^3$ en el intervalo $[x_i, x_{i+1})$.

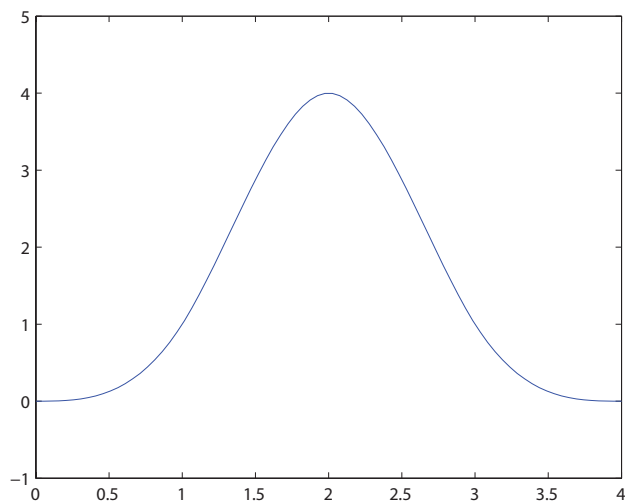


Figura C.5: B-Spline

Ultimo tramo no nulo (polinomio de tercer grado): $P_j(x) = \mu(x - x_j)^3$ en el intervalo $(x_{j-1}, x_j]$.

Si ahora, se unen estos dos tramos con un tramo intermedio: $j = i + 3$, y el polinomio resultante que une los dos tramos es (de tercer grado también):

$$P_{i+1}(x) = a_3(x - x_{i+1})^3 + a_2(x - x_{i+1})^2 + a_1(x - x_{i+1}) + a_0$$

Ahora, imponiendo que $P_i(x)$ y $P_{i+1}(x)$ coincidan hasta la derivada segunda en $x = x_{i+1}$, y que $P_{i+1}(x)$ y $P_{i+2}(x)$ coincidan hasta la derivada segunda en $x = x_{i+2}$, se llega a:

$$\begin{bmatrix} h_{i+1}^3 & (h_i + h_{i+1})^3 & h_{i+2}^3 \\ h_{i+1}^2 & (h_i + h_{i+1})^2 & -h_{i+2}^2 \\ h_{i+1} & (h_i + h_{i+1}) & h_{i+2} \end{bmatrix} \begin{bmatrix} a_3 - \lambda \\ \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Lo que muestra que la solución es nula y que los polinomios $P_i(x)$, $P_{i+1}(x)$ y $p_{i+2}(x)$ son nulos, y que el correspondiente spline es la función nula. Lo que hace concluir que no puede existir un spline cúbico que sea nulo en todo $[x_0, x_m]$, excepto en tres subintervalos.

Ahora, se unirán los dos tramos que se tenían inicialmente con dos tramos más. Es decir, se verá si existe un spline cúbico nulo excepto en cuatro subintervalos. Para ello

se opera de manera análoga al caso anterior y ahora $j = i + 4$:

$$\begin{aligned}
P_i(x) &= \lambda(x - x_i)^3 \\
P_{i+1}(x) &= a_3(x - x_{i+1})^3 + 3\lambda h_i(x - x_{i+1})^2 + 3\lambda h_i^2(x - x_{i+1}) + \lambda h_i^3 \\
P_{i+2}(x) &= b_3(x - x_{i+3})^3 - 3\mu h_{i+3}(x - x_{i+3})^2 + 3\mu h_{i+3}^2(x - x_{i+3}) - \mu h_{i+3}^3 \\
P_{i+3}(x) &= \mu(x - x_{i+4})^3
\end{aligned}$$

Después de las operaciones matemáticas se llega a:

$$\begin{bmatrix} h_{i+1}^3 & (h_i + h_{i+1})^3 & h_{i+2}^3 \\ h_{i+1}^2 & (h_i + h_{i+1})^2 & -h_{i+2}^2 \\ h_{i+1} & (h_i + h_{i+1}) & h_{i+2} \end{bmatrix} \begin{bmatrix} a_3 - \lambda \\ \lambda \\ b_3 - \mu \end{bmatrix} = \begin{bmatrix} -(h_{i+2} + h_{i+3})^3 \\ (h_{i+2} + h_{i+3})^2 \\ -(h_{i+2} + h_{i+3}) \end{bmatrix} \quad (\text{C.8})$$

Este sistema posee única solución para cada valor de μ , lo que indica que el spline cúbico puede existir mínimo con cuatro subintervalos no nulos.

Llamando entonces $B_i(x)$ al B-spline que es no nulo (sólo) en $[x_i, x_{i+4}]$, se puede construir:

$$B_0(x), B_1(x), \dots, B_{m-4}(x)$$

que son los $m - 3$ B-splines necesarios para que exista el spline cúbico. Pero como la dimensión de S_3 es $m + 3$, entonces se definen seis B-splines más para completar la base. Entonces, se tendrá construida la base de S_3 y cualquier spline cúbico asociados a los nodos x_{-3}, \dots, x_{m+3} podrá expresarse:

$$C(x) = \sum_{k=-3}^{m-1} p_k B_k(x) \quad (\text{C.9})$$

La propiedad más relevante de la base de los B-splines se deriva de su pequeño soporte (*soporte local*), es decir, solo es distinto de cero en un pequeño intervalo y una vez determinado dicho intervalo, el cálculo de un spline va a ser una tarea trivial y por tanto eficiente. De aquí también se deriva que la modificación de un coeficiente p_k se traduce en una modificación local (*control local*).

C.1.5. Nodos múltiples

En esta sección es de interés estudiar lo que sucede cuando un nodo (o dos) se hace tender hacia otro nodo, para ello se realizará un ejemplo en el que dos de los nodos serán variables (movibles).

Sea $x_i = 0$, $x_{i+1} = 1$, $x_{i+2} = 2$, $x_{i+3} = 3$, $x_{i+4} = 4$. Entonces $h_0 = h_1 = h_2 = h_3 = 1$. Lo que da:

$$\begin{bmatrix} 1 & 8 & 1 \\ 1 & 4 & -1 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} a_3 - \lambda \\ \lambda \\ b_3 - \mu \end{bmatrix} = \mu \begin{bmatrix} -8 \\ 4 \\ -2 \end{bmatrix} \quad (\text{C.10})$$

de donde:

$$a_3 + \mu = -4\lambda$$

$$\mu = -\lambda$$

$$b_3 - \mu = 4\mu$$

entonces, los cuatro tramos de polinomio no nulos del B-spline son, para $\lambda = 1$:

$$\begin{aligned} P_0(x) &= x^3 & 0 \leq x < 1 \\ P_1(x) &= x^3 - 4(x-1)^3 & 1 \leq x < 2 \\ P_2(x) &= -(x-4)^3 + 4(x-3)^3 & 2 \leq x < 3 \\ P_3(x) &= -(x-4)^3 & 3 \leq x < 4 \end{aligned} \quad (\text{C.11})$$

Ahora sí, se dejan los nodos $x = 2$ y $x = 3$ variables, es decir, ahora $x_{i+2} = p$ y $x_{i+3} = q$. Entonces $h_0 = 0$, $h_1 = p - 1$, $h_2 = q - p$, $h_3 = 4 - q$. Resolviendo el sistema que se genera con estas variables, con $\lambda = 1$, se obtiene el B-spline:

$$\begin{aligned}
P_0(x) &= x^3 & 0 \leq x < 1 \\
P_1(x) &= x^3 - \frac{4pq}{3(p-1)(q-1)}(x-1)^3 & 1 \leq x < p \\
P_2(x) &= \frac{-pq}{3(4-p)(4-q)}(x-4)^3 + \frac{4p}{(q-p)(4-q)(q-1)}(x-q)^3 & p \leq x < q \\
P_3(x) &= \frac{-pq}{3(4-p)(4-q)}(x-4)^3 & q \leq x < 4
\end{aligned}$$

Si $p \rightarrow 1$, el segundo tramo $P_1(x)$ tiende a ser vacío. Si el nodo p se hace coincidir con el nodo 1 (nodo doble) entonces la curva ahora sólo tiene tres tramos no nulos (no es spline cúbico) y la curva ya no es de clase C^2 en $x = 1$ sino sólo de clase C^1 .

Si $q \rightarrow 1$, el tercer tramo $P_2(x)$ tiende a ser vacío. Si el nodo q se hace coincidir con el nodo 1 (nodo triple) entonces la curva ahora sólo tiene dos tramos no nulos y la curva ya no es de clase C^1 en $x = 1$ sino sólo de clase C^0 (continua).

Este problema sirve para mostrar como controlar el orden de continuidad en un nodo. En la figura C.6 se muestra la forma como la curva va perdiendo su orden de continuidad desde C^2 hasta C^0 .

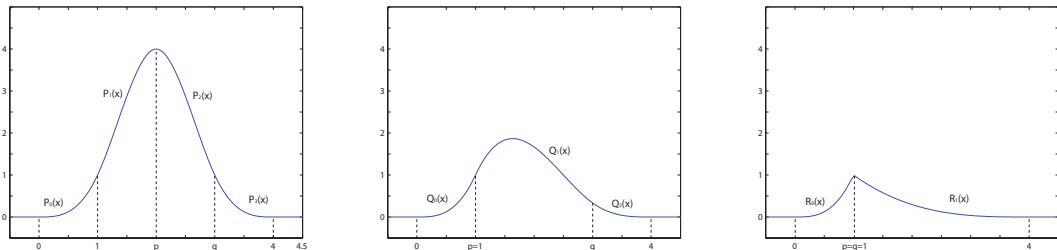


Figura C.6: Multiplicidad de un nodo

Algoritmo 4.03. Variación de la multiplicidad de un nodo:

```

clear all
clc
n=100
p=1
q=1
u=[0 1 p q 4]

```

```

x=linspace(u(1)-0.5,u(5)+0.5,n);
for i=1:n
if x(i)<u(1)
y(i)=0;
elseif x(i)>u(5)
y(i)=0;
elseif x(i)>u(4)
y(i)=-p*q/(3*(4-p)*(4-q))*(x(i)-4)^3
elseif x(i)>u(3)
y(i)=-p*q/(3*(4-p)*(4-q))*(x(i)-4)^3+4*p/((q-p)*(4-q)*(q-1))*(x(i)-q)^3
elseif x(i)>u(2)
y(i)=x(i)^3-4*p*q/(3*(p-1)*(q-1))*(x(i)-1)^3
elseif x(i)>u(1)
y(i)=x(i)^3
end
end
plot (x,y)
axis([u(1)-0.5 u(5)+0.5 -0.5 5])
hold on

```

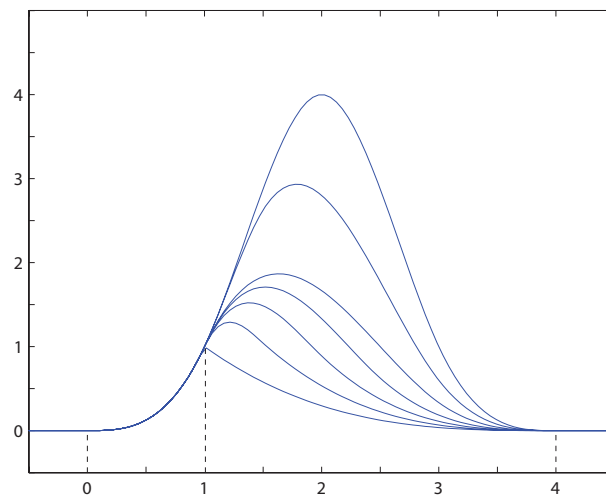


Figura C.7: Variación de la multiplicidad de un nodo

C.1.6. Nodos múltiples y potencias truncadas

Se vio que con $m + 3$ funciones de potencias truncadas se podía definir una base del espacio de los splines cúbicos definidos sobre el soporte. Ahora lo que se va a hacer es expresar de forma sencilla la continuidad de clase C^2 en un nodo x_i utilizando estas potencias.

Entonces, del caso anterior, con el soporte $\{0, 1, 2, 3, 4\}$ y $\lambda = 1$, donde se han variado dos de los nodos, la forma del spline hallado en forma de potencias truncadas es:

$$P_3(x) = (x - 0)^3 - \frac{4pq}{3(p-1)(q-1)}(x-1)^3 + \frac{4q}{(p-1)(4-p)(q-p)}(x-p)^3 + \frac{4p}{(q-1)(4-q)(q-p)}(x-q)^3$$

Ahora, si $p \rightarrow 1$:

$$Q_2(x) = \lim_{p \rightarrow 1} P_3(x) = (x-0)_+^3 + \left[\frac{4q(5-2q)}{9(q-1)^2}(x-1)_+^3 - \frac{4q}{q-1}(x-1)_+^2 \right] - \frac{4}{(q-1)^2(4-q)}(x-q)^3$$

Se puede observar que ha aparecido dos veces el factor $x - 1$ en la forma de potencias truncadas. De nuevo, con $q \rightarrow 1$:

$$R_1(x) = \lim_{q \rightarrow 1} Q_2(x) = (x-0)_+^3 - \left[\frac{28}{27}(x-1)_+^3 + \frac{8}{3}(x-1)_+^2 + 4(x-1)_+ \right]$$

Lo que se puede observar en esta expresión es que el factor $x - 1$ aparece tres veces. Con esto ya se puede concluir que, a medida que se vaya aumentando la multiplicidad a un nodo, se traduce en la aparición de la misma cantidad de potencias truncadas referidas al nodo. En la expresión, el nodo $x = 1$ es triple (lo que muestra 3 potencias truncadas de la forma $(x - 1)^i$).

En resumen:

- Nodo simple (multiplicidad 1) \iff continuidad C^2
- Nodo doble (multiplicidad 2) \iff continuidad C^1
- Nodo triple (multiplicidad 3) \iff continuidad C^0

Con el objeto de complementar:

- Nodo cuádruple (multiplicidad 4) \iff discontinuidad C^{-1}
- Nodo inexistente (multiplicidad 0) \iff continuidad C^3

Con lo cual, llamando a σ_i al orden de continuidad de un nodo y r_i a la multiplicidad asociada se tiene: $\sigma_i = 3 - r_i$.

C.1.7. Espacios de los splines cúbicos

Para una curva cúbica definida en el intervalo $[x_0, x_m]$ el número de parámetros libres es:

$$N = \left(\sum_{i=0}^m r_i \right) - 4 \quad (\text{C.12})$$

Considérese el spline cúbico de interpolación en los nodos $x_0 < x_1 < \dots < x_m$, definido como de clase C^2 en x_1, \dots, x_{m-1} , por lo que $\sigma_1 = \sigma_2 = \dots = \sigma_{m-1} = 2$ y al no especificarse en x_0 y x_m , $\sigma_0 = \sigma_m = -1$, con lo cual:

$$\begin{aligned} r_0 &= 4 \\ r_1 &= r_2 = \dots = r_{m-1} = 1 \\ r_m &= 4 \end{aligned}$$

Por tanto:

$$N = \left[4 + \left(\sum_{i=0}^{m-1} 1 \right) + 4 \right] - 4 = m + 3$$

que es el número de parámetros libres para definir este tipo de spline cúbico.

Para concluir, una vez especificada las multiplicidades en los nodos, se puede hallar la dimensión del espacio de los splines definidos sobre el soporte y con correspondientes multiplicidades asociadas.

C.2. Splines generales

Considere un conjunto de abscisas distintas $x_0 < x_1 < \dots < x_h$ con multiplicidad asociadas r_0, r_1, \dots, r_h ($r_i \geq 1$). El soporte de definición de un spline lo describiremos:

$$S = \{u_0, \dots, u_m\} = \underbrace{\{x_0, \dots, x_0\}}_{r_0}, \underbrace{\{x_1, \dots, x_1\}}_{r_1}, \dots, \underbrace{\{x_h, \dots, x_h\}}_{r_h} \quad (\text{C.13})$$

cada abscisa aparecerá repetida tantas veces como indique su multiplicidad.

las abscisas (distintas) x_i se denominan *puntos de ruptura* y van a determinar los distintos subintervalos en $[x_0, x_m]$ que corresponden a tramos sucesivos en una curva spline.

Los puntos u_0, \dots, u_m seguirán llamándose *nodos*, pero ahora varios nodos podrán ser iguales, es decir $u_0 \leq u_1, \dots, u_m$. Y el número de puntos en el soporte es $m + 1 = r_0 + r_1 + \dots + r_h$.

Un spline de grado p definido sobre el soporte es entonces una curva $C(x)$ de clase C^{p-r_i} en cada punto de ruptura x_i y tal que $C(x)$ restringida al intervalo $[x_i, x_{i+1}]$ es un polinomio de grado menor o igual que p .

En los puntos extremos, lo más frecuente es no imponer restricciones a $P_0(x)$ y $P_{h-1}(x)$, con lo cual se prescinde totalmente de los polinomios $P_{-1}(x)$ y $P_h(x)$. Esto equivale a decir que el spline $C(x)$ es de clase C^{-1} en x_0 y x_h (discontinuo), o bien que $r_i = p + 1$ (máxima multiplicidad en los puntos x_0 y x_h). Este tipo de spline se conoce en la literatura como “que tiene en los extremos condiciones tipo *camped*”.

Para un caso en el que los splines están definidos en $(-\infty, \infty)$, el número de parámetros libres para definir un spline de grado p sobre el conjunto de nodos (puntos de ruptura) $x_0 < x_1, \dots, < x_h$, es:

$$\bar{N} = \left(\sum_{i=0}^h r_i \right) + (p + 1) = (m + 1) + (p + 1) = m + p + 2$$

Si sólo se define en $[x_0, x_h]$, entonces:

$$N = \left(\sum_{i=0}^h r_i \right) - (p + 1) = (m + 1) - (p + 1) = m - p \quad (\text{C.14})$$

En este capítulo, la tarea será encontrar bases constituidas por $N = m - p$ splines.

C.2.1. Bases de potencias truncadas

Dado un nodo x_i con multiplicidad r_i , en el intervalo $[x_i, x_{i+1})$ y para polinomios de grado p , el spline se puede representar:

$$C(x) = P_i(x) = P_{i-1}(x) + c_{i1}(x - x_i)_+^p + \dots + c_{ir_i}(x - x_i)_+^{p-r_i+1} \quad (\text{C.15})$$

es decir, que en su expresión aparecerán r_i potencias truncadas. Cada nodo x_i aporta r_i potencias truncadas, de modo que:

$$\begin{aligned}\bar{N} &= (p+1) + r_0 + r_1 + \dots + r_h = \left(\sum_{i=0}^h r_i \right) + (p+1) \\ N &= m - p = \left(\sum_{i=0}^h r_i \right) - (p+1)\end{aligned}$$

lo cual corresponde con la dimensión del espacio de los splines asociado al soporte dado, junto con las multiplicidades de los nodos.

C.2.2. Base de B-Splines

Dado un soporte

$$S = \{u_0, \dots, u_m\} = \underbrace{\{x_0, \dots, x_0\}}_{r_0}, \underbrace{\{x_1, \dots, x_1\}}_{r_1}, \dots, \underbrace{\{x_k, \dots, x_k\}}_{r_k}$$

donde $m = \left(\sum_{i=0}^k r_i \right) - 1$, un B-Spline $B_j(x)$ de grado p es una función definida en $(-\infty, \infty)$, de clase C^{p-r_i} en cada punto de ruptura x_i , y que es nula excepto en $p+1$ subintervalos consecutivos $[u_j, u_{j+1}), [u_{j+1}, u_{j+2}), \dots, [u_{j+p}, u_{j+p+1})$ en los cuales es un polinomio de grado p . Las multiplicidades r_i han de verificar:

$$r_i \leq p+1 \tag{C.16}$$

Un subintervalo $[u_{j+h}, u_{j+h+1})$ puede ser vacío y esto ocurre cuando $u_{j+h} = u_{j+h+1}$. Por tanto, el número de subintervalos en los que el spline es no nulo puede ser menor.

Cada nodo u_i con $i = 0, 1, \dots, m$ tiene asociado un B-Spline; pero el último sería no nulo en:

$$[u_{m-p-1}, u_{m-p}), [u_{m-p}, u_{m-p+1}), \dots, [u_{m-1}, u_m)$$

y este B-Spline se denota como $B_{m-p-1}(x)$. Por tanto, se tiene definidos realmente:

$$B_0(x), B_1(x), \dots, B_{m-p-1}(x)$$

que son los $m-p$ B-Splines.

Ya que la dimensión del espacio de los B-Splines restringidos al intervalo $[x_0, x_m]$ es

$m - p$ y que dos B-Splines distintos son linealmente independientes, el conjunto de los $m - p$ B-Splines definidos constituyen una base.

A continuación se verán ejemplos de B-Splines según distintas multiplicidades de los nodos y utilizando el soporte $S = \{0, 1, 2, 3, 4, 5\}$:

1. B-Splines de grado 0.

Grado $p = 0$, multiplicidades $r_i \leq p + 1 = 1$, entonces $r_i = 1 \forall i$, $k = 5$, $m = k = 5$, $m - p = 5$.

Las ordenes de continuidad asociadas a las multiplicidades son: $\sigma_i = p - r_i = -1 \forall i$.

$$S = \{u_0, u_1, \dots, u_m\} = \{0, 1, 2, 3, 4, 5\}$$

Los B-Splines correspondientes al soporte son nulos en $(-\infty, \infty)$ salvo en un subintervalo $(p + 1)$.

$$B_0(x) = \begin{cases} 0 & -\infty < x < 0 \\ 1 & 0 \leq x \leq 1 \\ 0 & 1 \leq x < \infty \end{cases}$$

$$B_1(x) = \begin{cases} 0 & -\infty < x < 1 \\ 1 & 1 \leq x \leq 2 \\ 0 & 2 \leq x < \infty \end{cases}$$

$$B_2(x) = \begin{cases} 0 & -\infty < x < 2 \\ 1 & 2 \leq x \leq 3 \\ 0 & 3 \leq x < \infty \end{cases}$$

$$B_3(x) = \begin{cases} 0 & -\infty < x < 3 \\ 1 & 3 \leq x \leq 4 \\ 0 & 4 \leq x < \infty \end{cases}$$

$$B_4(x) = \begin{cases} 0 & -\infty < x < 4 \\ 1 & 4 \leq x \leq 5 \\ 0 & 5 \leq x < \infty \end{cases}$$

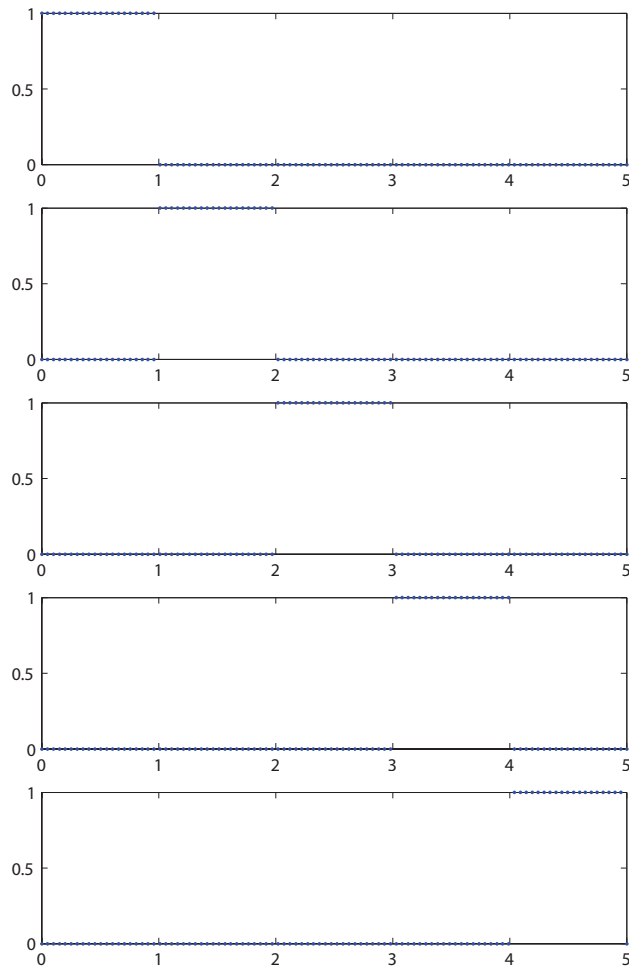


Figura C.8: B-Splines de grado 0

2. B-Spline de grado 1.

Grado $p = 1$. Las multiplicidades pueden valer 1 o 2.

a) $r_i = 1 \forall i$. $m = 5$, $m - p = 4$, $\sigma_i = p - r_i = 0 \forall i$.

$$S = \{u_0, u_1, \dots, u_m\} = \{0, 1, 2, 3, 4, 5\}$$

Los B-Splines asociados al soporte son nulos en excepto en dos subintervalos

$(p + 1)$.

$$B_0(x) = \begin{cases} 0 & -\infty < x < 0 \\ x & 0 \leq x < 1 \\ 2 - x & 1 \leq x < 2 \\ 0 & 2 \leq x < \infty \end{cases}$$

$$B_1(x) = \begin{cases} 0 & -\infty < x < 1 \\ x & 1 \leq x < 2 \\ 2 - x & 2 \leq x < 3 \\ 0 & 3 \leq x < \infty \end{cases}$$

$$B_2(x) = \begin{cases} 0 & -\infty < x < 2 \\ x & 2 \leq x < 3 \\ 2 - x & 3 \leq x < 4 \\ 0 & 4 \leq x < \infty \end{cases}$$

$$B_3(x) = \begin{cases} 0 & -\infty < x < 3 \\ x & 3 \leq x < 4 \\ 2 - x & 4 \leq x < 5 \\ 0 & 5 \leq x < \infty \end{cases}$$

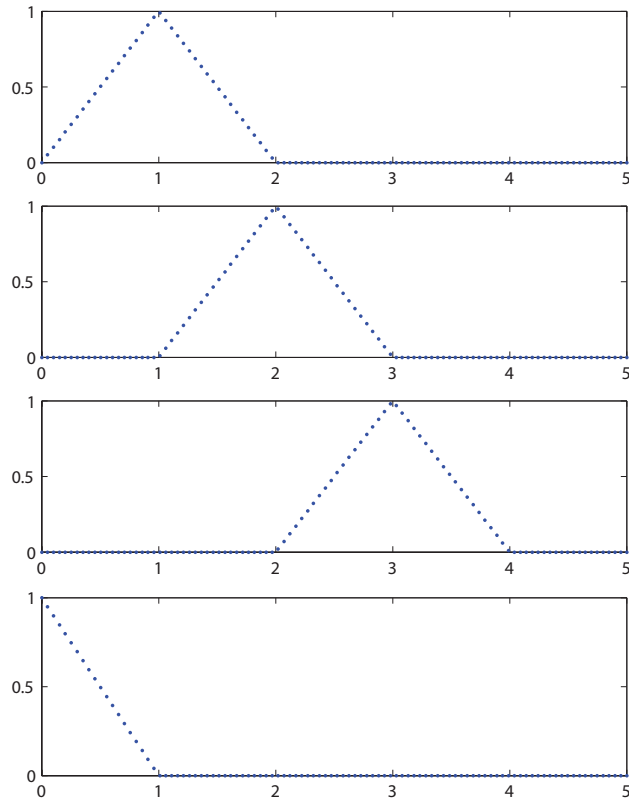


Figura C.9: B-Splines de grado 1, nodos simples

- b) Grado $p = 1$, multiplicidades $r_0 = 2, r_1 = r_2 = r_3 = r_4 = 1, r_5 = 2$.
 $m = 7, m - p = 6$.

Las órdenes de continuidad en $x = 0$ y $x = 5$ son $\sigma_i = p - r_i = -1$ y en los nodos internos $\sigma_i = 0$.

$$S = \{u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7\} = \{0, 0, 1, 2, 3, 4, 5, 5\}$$

El primer y último B-Spline son no nulos en solo un subintervalo ya que el otro es vacío.

$$B_0(x) = \begin{cases} 0 & -\infty < x < 0 \\ 1 - x & 0 \leq x < 1 \\ 0 & 1 \leq x < \infty \end{cases}$$

$$B_1(x) = \begin{cases} 0 & -\infty < x < 0 \\ x & 0 \leq x < 1 \\ 2 - x & 1 \leq x < 2 \\ 0 & 2 \leq x < \infty \end{cases}$$

$$B_2(x) = \begin{cases} 0 & -\infty < x < 1 \\ x - 1 & 1 \leq x < 2 \\ 3 - x & 2 \leq x < 3 \\ 0 & 3 \leq x < \infty \end{cases}$$

$$B_3(x) = \begin{cases} 0 & -\infty < x < 2 \\ x - 2 & 2 \leq x < 3 \\ 4 - x & 3 \leq x < 4 \\ 0 & 4 \leq x < \infty \end{cases}$$

$$B_4(x) = \begin{cases} 0 & -\infty < x < 3 \\ x - 3 & 3 \leq x < 4 \\ 5 - x & 4 \leq x < 5 \\ 0 & 5 \leq x < \infty \end{cases}$$

$$B_5(x) = \begin{cases} 0 & -\infty < x < 4 \\ x - 4 & 4 \leq x < 5 \\ 0 & 5 \leq x < \infty \end{cases}$$

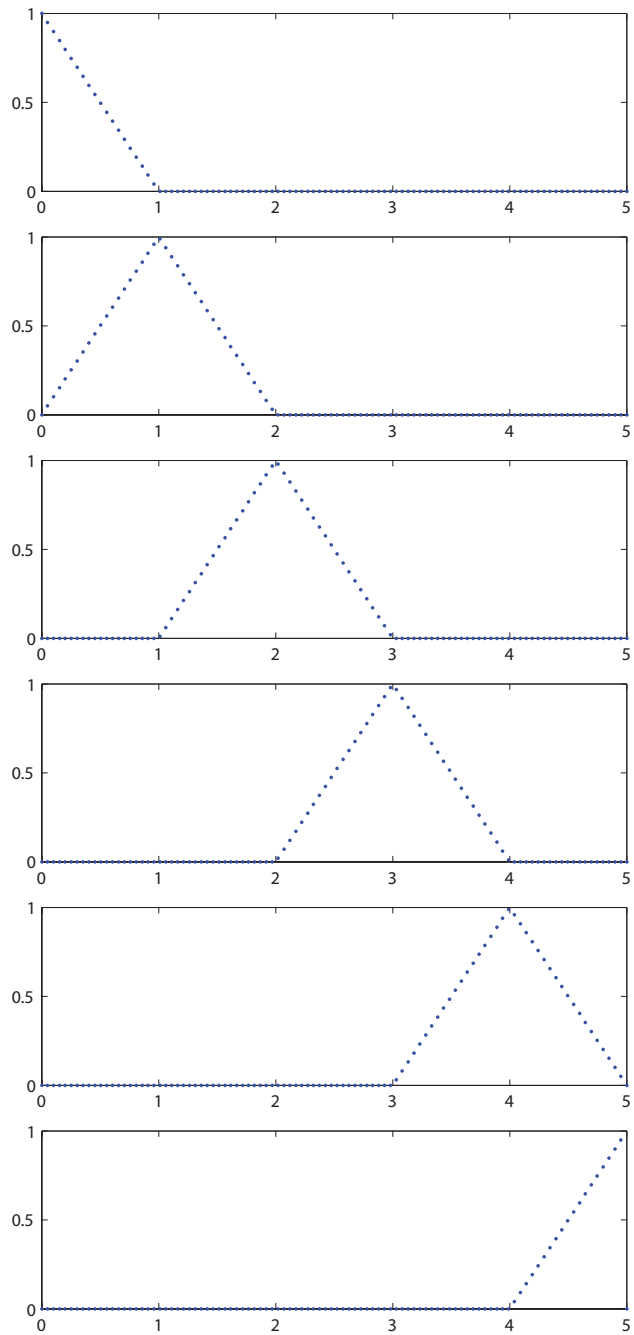


Figura C.10: B-Splines de grado 1, multiplicidad $p + 1$ en los extremos

- a) Grado $p = 1$, multiplicidades $r_0 = r_1 = r_2 = 1, r_3 = 2, r_4 = r_5 = 1$.
 $m = 6, m - p = 5$.

Las órdenes de continuidad son $\sigma_i = 0$ y en el nodo $x = 3$ es $\sigma_i = -1$.

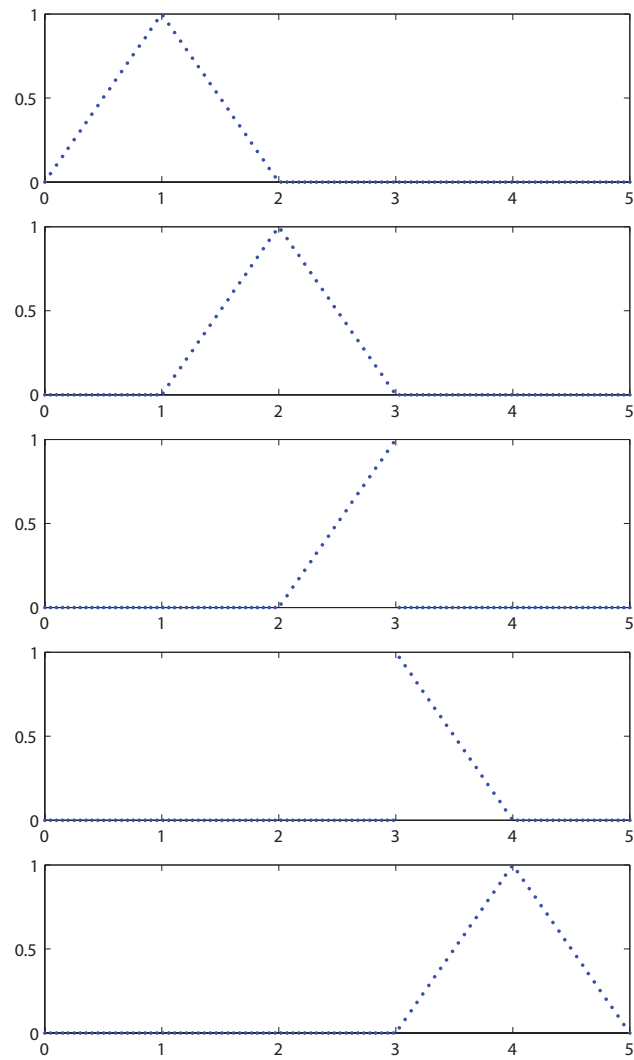


Figura C.11: B-Splines grado 1, multiplicidad $p + 1$ en un nodo interno

3. B-Spline de grado 2.

a) Grado $p = 2$, $r_1 = 1 \forall i$, $m = 5$, $m - p = 3$. Órdenes de continuidad $\sigma_i = 1$.

$$B_0(x) = \begin{cases} 0 & -\infty < x < 0 \\ \frac{x^2}{2} & 0 \leq x < 1 \\ -x^2 + 3x - \frac{3}{2} & 1 \leq x < 2 \\ \frac{(x-3)^2}{2} & 2 \leq x < 3 \\ 0 & 3 \leq x < \infty \end{cases}$$

$$B_1(x) = \begin{cases} 0 & -\infty < x < 1 \\ \frac{(x-1)^2}{2} & 1 \leq x < 2 \\ -x^2 + 5x - \frac{11}{2} & 2 \leq x < 3 \\ \frac{(x-4)^2}{2} & 3 \leq x < 4 \\ 0 & 4 \leq x < \infty \end{cases}$$

$$B_2(x) = \begin{cases} 0 & -\infty < x < 2 \\ \frac{(x-2)^2}{2} & 2 \leq x < 3 \\ -x^2 + 7x - \frac{23}{2} & 3 \leq x < 4 \\ \frac{(x-5)^2}{2} & 4 \leq x < 5 \\ 0 & 5 \leq x < \infty \end{cases}$$

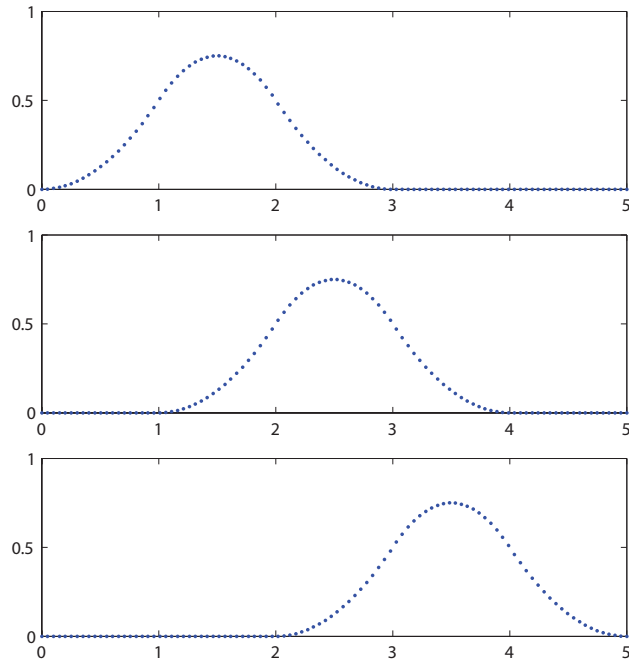


Figura C.12: B-Spline grado 2, nodos simples

b) Grado $p = 2$, multiplicidades $r_0 = 2, r_1 = r_2 = r_3 = r_4 = 1, r_5 = 2$,
 $m = 7, m - p = 5$.

Las órdenes de continuidad en $x = 0$ y $x = 5$ son $\sigma_i = p - r_i = 0$ y en los nodos internos $\sigma_i = 1$.

$$S = \{0, 0, 1, 2, 3, 4, 5, 5\}$$

Los B-Spline son:

$$B_0(x) = \begin{cases} 0 & -\infty < x < 0 \\ \frac{x(4-3x)}{2} & 0 \leq x < 1 \\ \frac{(2-x)^2}{2} & 1 \leq x < 2 \\ 0 & 2 \leq x < \infty \end{cases}$$

$$B_1(x) = \begin{cases} 0 & -\infty < x < 0 \\ \frac{x^2}{2} & 0 \leq x < 1 \\ -x^2 + 3x - \frac{3}{2} & 1 \leq x < 2 \\ \frac{(x-3)^2}{2} & 2 \leq x < 3 \\ 0 & 3 \leq x < \infty \end{cases}$$

$$B_2(x) = \begin{cases} 0 & -\infty < x < 1 \\ \frac{(x-1)^2}{2} & 1 \leq x < 2 \\ -x^2 + 5x - \frac{11}{2} & 2 \leq x < 3 \\ \frac{(x-4)^2}{2} & 3 \leq x < 4 \\ 0 & 4 \leq x < \infty \end{cases}$$

$$B_3(x) = \begin{cases} 0 & -\infty < x < 2 \\ \frac{(x-2)^2}{2} & 2 \leq x < 3 \\ -x^2 + 7x - \frac{23}{2} & 3 \leq x < 4 \\ \frac{(x-5)^2}{2} & 4 \leq x < 5 \\ 0 & 5 \leq x < \infty \end{cases}$$

$$B_4(x) = \begin{cases} 0 & -\infty < x < 3 \\ \frac{(x-3)^2}{2} & 3 \leq x < 4 \\ \frac{(5-x)(3x-11)}{2} & 4 \leq x < 5 \\ 0 & 5 \leq x < \infty \end{cases}$$

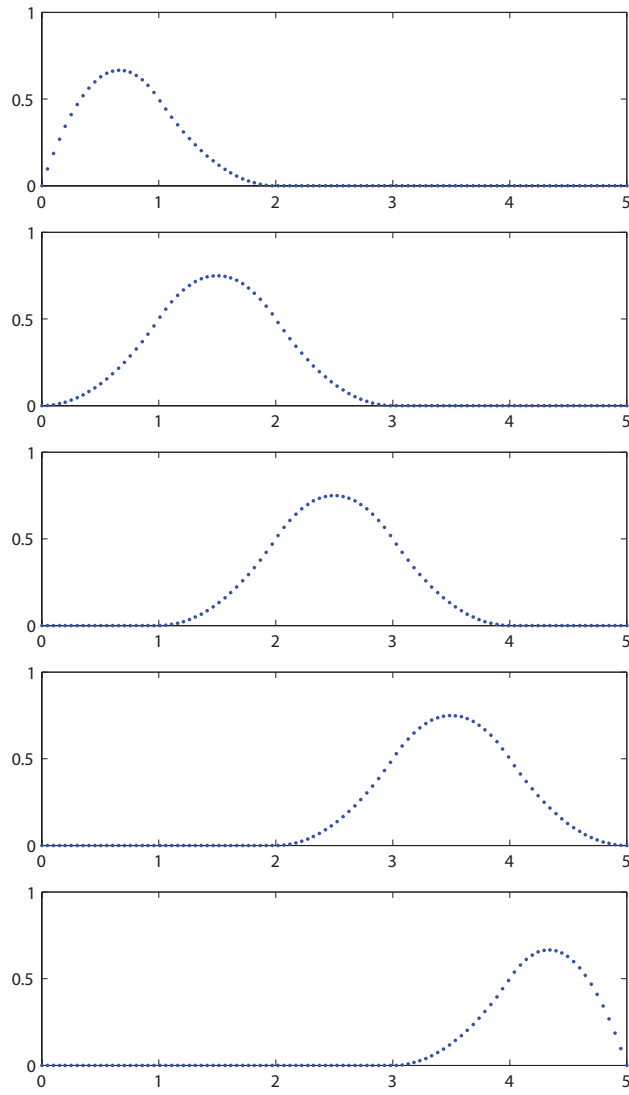


Figura C.13: B-Splines grado 2, multiplicidad p en los extremos

- a) Grado $p = 2$, multiplicidades $r_0 = 3, r_1 = r_2 = r_3 = r_4 = 1, r_5 = 3$,
 $m = 9, m - p = 7$.

Las órdenes de continuidad en $x = 0$ y $x = 5$ son $\sigma_i = p - r_i = -1$ y en los nodos internos $\sigma_i = 1$.

$$S = \{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$$

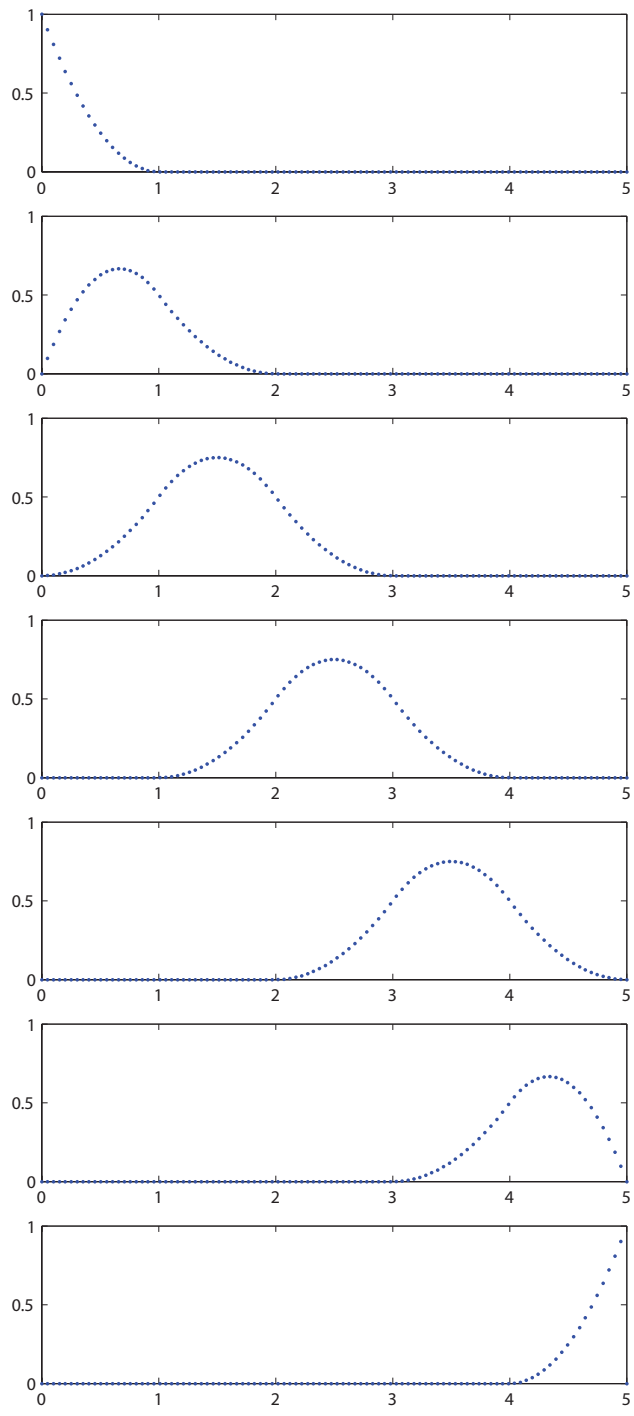


Figura C.14: B-Spline grado 2, multiplicidad $p + 1$ en los extremos

C.2.3. Relación de recurrencia para B-Splines

Para la teoría y desarrollo de los B-Splines, tres autores, L. Hansfield, C. De Boor y H. Cox, descubrieron en 1972 de forma independiente una importante relación de recurrencia que verifican B-Splines de grados sucesivos definidos sobre un mismo soporte.

Considerando el soporte:

$$S = \{u_0, u_1, \dots, u_m\} = \{\underbrace{x_0, \dots, x_0}_{r_0}, \underbrace{x_1, \dots, x_1}_{r_1}, \dots, \underbrace{x_k, \dots, x_k}_{r_k}\}$$

y cada punto de ruptura x_i aparece repetido tantas veces como indica su multiplicidad.

Sea:

$$B_i^0(x) = \begin{cases} 1 & u_i \leq x \leq u_{i+1} \\ 0 & \text{en otro caso} \end{cases} \quad 0 \leq i \leq m-1 \quad (\text{C.17})$$

Se define:

$$B_i^p(x) = \frac{x - u_i}{u_{i+p} - u_i} B_i^{p-1}(x) + \frac{u_{i+p+1} - x}{u_{i+p+1} - u_{i+1}} B_{i+1}^{p-1}(x) \quad p = 1, 2, 3, \dots \quad (\text{C.18})$$

como la relación de recurrencia para los B-Splines de grado p .

Entonces, para cada j , las funciones $B_0^j(x), B_1^j(x), \dots, B_{m-j-1}^j(x)$ son los B-Splines de grado j asociados al soporte.

Para que la recurrencia sea operativa incluso los B-Splines nulos que pertenecen a intervalos vacíos son necesarios, y el cociente $\frac{0}{0}$ debe ser interpretado como 0 para todo x .

La relación de recurrencia genera exactamente $m - p$ B-Splines no nulos de grado p . Es decir, para un grado p , la relación de recurrencia produce tantos B-Splines no nulos $B_i^p(x)$ como la dimensión del espacio de todos los splines asociados al soporte.

Con un ejemplo se mostrará el proceso operativo para obtener mediante la recurrencia los B-Splines de segundo grado con el siguiente soporte:

$$S = \{u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7\} = \{0, 0, 1, 2, 3, 4, 5, 5\}$$

con lo cual $x = 0$ y $x = 5$ son nodos dobles y $x = 1, 2, 3, 4$ simples y $m = 7$.

1. B-Splines de grado 0.

$$B_0^0(x) = 0 \quad x \in (-\infty, \infty)$$

$$B_1^0(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_2^0(x) = \begin{cases} 1 & 1 \leq x < 2 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_3^0(x) = \begin{cases} 1 & 2 \leq x < 3 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_4^0(x) = \begin{cases} 1 & 3 \leq x < 4 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_5^0(x) = \begin{cases} 1 & 4 \leq x < 5 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_6^0(x) = 0 \quad x \in (-\infty, \infty)$$

Los nodos dobles producen dos B-Splines idénticamente nulos. Dado que $p = 0$ estos nodos deben ser considerados como simples en esta etapa intermedia el calculo de los B-Splines de grado 1.

2. B-Splines de grado 1.

$$B_i^1(x) = \frac{x - u_i}{u_{i+1} - u_i} B_i^0(x) + \frac{u_{i+2} - x}{u_{i+2} - u_{i+1}} B_{i+1}^0(x)$$

$$B_0^1(x) = \frac{x}{0} B_0^0(x) + \frac{1-x}{1} B_1^0(x) = \begin{cases} 1-x & 0 \leq x < 1 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_1^1(x) = \frac{x}{1} B_1^0(x) + \frac{2-x}{1} B_2^0(x) = \begin{cases} x & 0 \leq x < 1 \\ 2-x & 1 \leq x < 2 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_2^1(x) = \frac{x-1}{1} B_2^0(x) + \frac{3-x}{1} B_3^0(x) = \begin{cases} x-1 & 1 \leq x < 2 \\ 3-x & 2 \leq x < 3 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_3^1(x) = \frac{x-2}{1} B_3^0(x) + \frac{4-x}{1} B_4^0(x) = \begin{cases} x-2 & 2 \leq x < 3 \\ 4-x & 3 \leq x < 4 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_4^1(x) = \frac{x-3}{1} B_4^0(x) + \frac{5-x}{1} B_5^0(x) = \begin{cases} x-3 & 3 \leq x < 4 \\ 5-x & 4 \leq x < 5 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_5^1(x) = \frac{x-4}{0} B_5^0(x) + \frac{5-x}{1} B_6^0(x) = \begin{cases} x-4 & 4 \leq x < 5 \\ 0 & \text{en otro caso} \end{cases}$$

3. B-Spline de grado 2.

$$B_i^2(x) = \frac{x - u_i}{u_{i+2} - u_i} B_i^1(x) + \frac{u_{i+3} - x}{u_{i+3} - u_{i+1}} B_{i+1}^1(x)$$

$$B_0^2(x) = \frac{x}{1} B_0^1(x) + \frac{2-x}{2} B_1^1(x) = \begin{cases} \frac{x(4-3x)}{2} & 0 \leq x < 1 \\ (2-x)^2 & 1 \leq x < 2 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_1^2(x) = \frac{x}{2} B_1^1(x) + \frac{3-x}{2} B_2^1(x) = \begin{cases} \frac{x^2}{2} & 0 \leq x < 1 \\ -x^2 + 3x - \frac{3}{2} & 1 \leq x < 2 \\ \frac{(3-x)^2}{2} & 2 \leq x < 3 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_2^2(x) = \frac{x-1}{2} B_2^1(x) + \frac{4-x}{2} B_3^1(x) = \begin{cases} \frac{(x-1)^2}{2} & 1 \leq x < 2 \\ -x^2 + 5x - \frac{11}{2} & 2 \leq x < 3 \\ \frac{(4-x)^2}{2} & 3 \leq x < 4 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_3^2(x) = \frac{x-2}{2} B_3^1(x) + \frac{5-x}{2} B_4^1(x) = \begin{cases} \frac{(x-2)^2}{2} & 2 \leq x < 3 \\ -x^2 + 7x - \frac{23}{2} & 3 \leq x < 4 \\ \frac{(5-x)^2}{2} & 4 \leq x < 5 \\ 0 & \text{en otro caso} \end{cases}$$

$$B_4^2(x) = \frac{x-3}{2} B_4^1(x) + \frac{5-x}{2} B_5^1(x) = \begin{cases} \frac{(x-3)^2}{2} & 3 \leq x < 4 \\ \frac{(5-x)(3x-11)}{2} & 4 \leq x < 5 \\ 0 & \text{en otro caso} \end{cases}$$

Algoritmo 4.04. Relación de recurrencia.

```

clear all
close all
clc
S=[0 0 0 1 2 3 4 5 5 5];           % EL SOPORTE
grado=3;                            % ¿cual es el grado q queremos?
n=length(S);

```

```

count=0;
% contador para el # de ecuaciones,
% 1 por c/intervalo de c/curva incluye 0's
orden=0;
% un contador para el orden
for i=1:n-1
% CUAL INTERVALO ES NULO?
if S(i+1)==S(i)
u(i)=0;
else
u(i)=1;
end
end

%===== SE LLENA MATRIZ 'B' CON LOS COEFICIENTES
%===== SE LLENA MATRIZ 'PARA' CON EL SOPORTE QUE LE CONTIENE

for i=1:n-1
for j=1:n-1
para((n-1)*(i-1)+j,1)=orden; % "B^i"
para((n-1)*(i-1)+j,2)=i; % # de la curva "B_i"
para((n-1)*(i-1)+j,3)=S(j); % COTA INFERIOR
para((n-1)*(i-1)+j,4)=S(j+1); % COTA SUPERIOR
if i==j
B((n-1)*(i-1)+j,1)=u(j); % ceros si son nulos
end
count=count+1; % cuenta cada "B_i"
end
end
curvas(1,1)=1;

%las curvas por c/grado de donde a donde
curvas(1,2)=count;

%===== para grado>0 =====
for orden=1:grado % por c/ grado de la curva esa
curvas(orden+1,1)=count+1;

```

```

for i=1:n-1-orden                                % c/curva para c/ n-1 intervalos
                                                se determina a/b c/d

%=====                                       % denominadores
e=-S(i);                                         % ya le doy el signo
f=S(i+orden+1);
b=S(i+orden)+e;                                 %= u_(i+p)-u_i
d=f-S(i+1);

% ===== con c/ b,d calculo a,b con c/intervalo
for j=1:n-1
count=count+1;
para(count,1)=orden;
para(count,2)=i;
para(count,3)=S(j);
para(count,4)=S(j+1);

%==== para calcular B ===== me debo devolver --->
%n-1-orden-->(Bi^p-1)=count-(n-orden)*(n-1)
%& n-orden-->(Bi+1^p-1)=count-(n-orden+1)*(n-1)

m=B(count-(n-orden)*(n-1),:);                 % tomo los coeficientes de B_i^(p-1)
bi=m(orden:-1:1);                             % lo invierto para usarlo y guardarlo
                                                x^0, x^1, x^2 etc.

bi(orden+1)=0;                                 % equivale a multiplicar por x
m(orden+1)=0;                                  % le añado un x^n por
                                                cero que luego se invierte

m=m(orden+1:-1:1);                             % invierto el q tiene cero de 1er
                                                para poder sumar vectores

m=m*e;                                         % multiplica la cte
a=bi+m;

%-----
m=B(count-(n-orden-1)*(n-1),:);
% tomo los coeficientes de B_(i+1)^(p-1)
bi=-m(orden:-1:1);                             % por -1

```

```

bi(orden+1)=0; % por "X"
m(orden+1)=0;
% le añado un x^n por cero que luego se invierte
m=m(orden+1:-1:1);
% invierto el q tiene cero de 1er
    para poder sumar vectores
m=m*f; % multiplica la cte
c=bi+m;
if b==0 % ya que 0/0=0.
a=0;
b=1;
elseif d==0
c=0;
d=1;
end
Bi=a/b+c/d; % calculo el nuevo "polinomio"
Bi=Bi(orden+1:-1:1); % lo invierto para guardar
for k=1:orden+1 % SE GUARDA EN "B" 1x1
B(count,k)=Bi(k);
end
end
end
curvas(orden+1,2)=count; % hasta donde van en este grado
end

% ===== graficas =====
x=linspace(S(1),S(n)); % 100 por defecto
xo=length(x);
y(xo)=0; % por si acaso
for k=0:count/(n-1)-1 % por c/intervalo <===
for j=n:-1:2
m=B(k*(n-1)+j-1,:); % se evalúa en su correspondiente ecs
m=m(orden+1:-1:1);
for i=xo:-1:1 % se va por todo el soporte <====
if x(i)<S(j)

```

```

y(i)=polyval(m,x(i));           % y evalúa el que corresponde
end
end
end
plot(x,y, 'r')
axis([S(1),S(n),0,1])
xy=ginput(1);
end

```

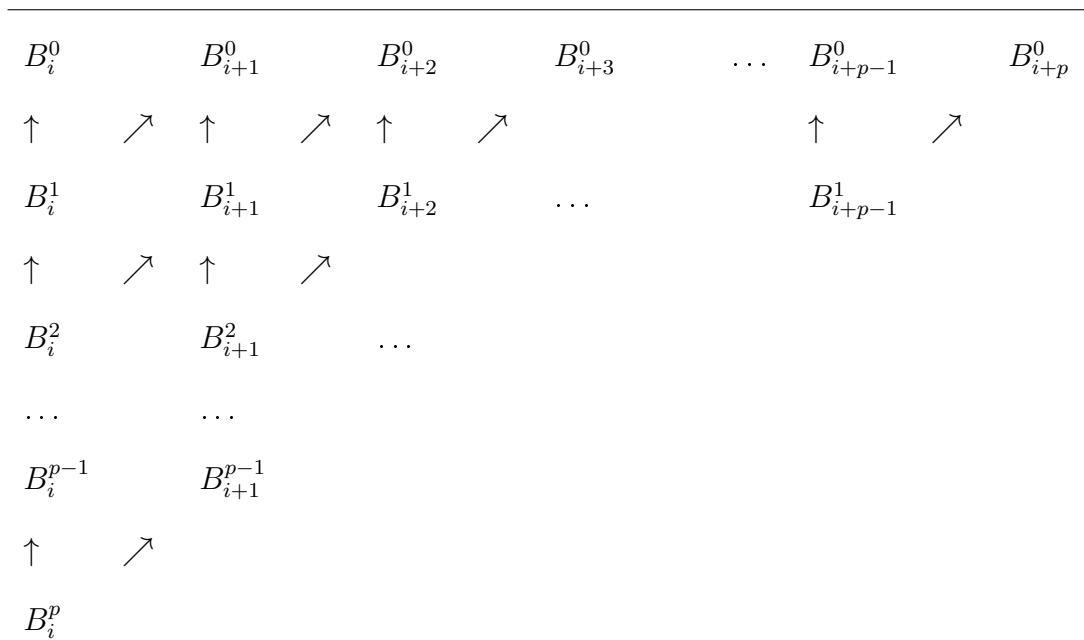
C.2.4. Evaluación de B-Splines por recurrencia

En la práctica se busca la evaluación de los B-Splines en un valor $x = \bar{x}$ del parámetro o en todo el intervalo $[x_0, x_k]$ con objeto de dibujar una curva expresada como combinación de B-Splines.

$B_i^p(x)$ requiere para su cálculo los B-Splines de grado 0:

$$B_i^0(x), B_{i+1}^0, \dots, B_{i+p}^0(x)$$

La relación de recurrencia puede representarse de forma esquemática:



Lo usual será evaluar todos los B-Splines de la base en el punto $x = \bar{x}$. Para ello, si

$\bar{x} \in [x_k, x_{k+1})$, solamente $B_{k-p}^p(\bar{x}), \dots, B_k^p(\bar{x})$ son no nulos, con lo cual se tendrá que:

$B_{k-p}^p(\bar{x})$	<i>depende de</i>	$B_{k-p}^0(\bar{x}), \dots, B_k^0(\bar{x})$
$B_{k-p+1}^p(\bar{x})$	<i>depende de</i>	$B_{k-p+1}^0(\bar{x}), \dots, B_{k+1}^0(\bar{x})$
\dots	\dots	\dots
$B_k^p(\bar{x})$	<i>depende de</i>	$B_k^0(\bar{x}), \dots, B_{k+p}^0(\bar{x})$

y en definitiva, son necesarios $B_{k-p}^0(\bar{x}), \dots, B_k^0(\bar{x})$, aunque únicamente $B_k^0(\bar{x})$ es distinto de 0 (y vale 1). El esquema para el cálculo de los B-Splines de grado p no nulos en $x = \bar{x}$ es:

			...	0	0	1	...
			↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗
			0	0	B_{k-1}^1	B_k^1	0
			↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗
			0	B_{k-2}^2	B_{k-1}^2	B_k^2	0
			↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗
			B_{k-3}^3	B_{k-2}^3	B_{k-1}^3	B_k^3	
...
0	B_{k-p+1}^{p-1}	...	B_{k-3}^{p-1}	B_{k-2}^{p-1}	B_{k-1}^{p-1}	B_k^{p-1}	0
↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗	↑ ↗
B_{k-p}^p	B_{k-p+1}^p	...	B_{k-3}^p	B_{k-2}^p	B_{k-1}^p	B_k^p	

Para un determinado grado j , el primero de los B-Splines no nulos, $B_{k-j}^j(\bar{x})$, sólo requiere el segundo término en la relación de recurrencia, mientras que el último no nulo, $B_k^j(\bar{x})$ sólo necesita el primer término de la recurrencia.

Finalmente, se puede decir que el proceso comienza por determinar el intervalo $[u_k, u_{k+1})$ con $u_k < u_{k+1}$ al que pertenece \bar{x} , pues el subíndice k establece el único B-Spline no nulo en \bar{x} .

A continuación se presenta un algoritmo que calcula los B-Splines no nulos en un punto x para un soporte no periódico tipo `camped` y se supone implícitamente que $p \leq k \leq m - p - 1$.

De manera general $B_i^p(x)$ es de clase C^{p-r_i} en el nodo u_i con $r_i \leq p + 1$.

Algoritmo 4.05. Evaluación de un B-Spline por recurrencia.

```
% Correr después del Algoritmo 4.04

clc
a=input('ingresa punto en X: ')
cont=0;
for i=1:count % por todo el contenido
if a>para(i,3) & a<para(i,4) % si "a" esta en un intervalo no nulo
if i<=n-1 % solo para hallar "k" en el
% primer intervalo

k=i;
end
bi=B(i,:);
bi=bi(ordem+1:-1:1); % se invierte para calcular
Y=polyval(bi,a);
if Y~=0
cont=cont+1;
rta(cont,1)=i; % cual es la ubicación en B
rta(cont,2)=Y;
end
end
end
```

C.2.5. Propiedades de los B-Splines

Las propiedades de los B-Splines más importantes se dan a continuación. Estas pueden ser demostradas a partir de la relación de recurrencia:

1. Soporte local.

$$B_i^p(x) = 0 \quad \forall x \notin [u_i, u_{i+p+1}) \quad (\text{C.19})$$

2. No negatividad.

$$B_i^p(x) > 0 \quad \forall x \in [u_i, u_{i+p+1}) \quad (\text{C.20})$$

3. Base local.

Si $x \in [u_k, u_{k+1})$, los únicos B-Splines (posiblemente) no nulos son:

$$B_{k-p}^p(x), B_{k-p+1}^p(x), \dots, B_k^p(x) \quad (p+1 \text{ B-Splines})$$

donde se entiende que si $k \leq p$ entonces los B-Splines no nulos son:

$$B_0^p(x), B_1^p(x), \dots, B_k^p(x)$$

4. Partición de la unidad.

En un intervalo no vacío $[u_k, u_{k+1})$ con $p \leq k < m - p - 1$, los $p+1$ B-Splines que son no nulos en él verificarán:

$$\sum_{i=k-p}^k B_i^p(x) = 1 \quad (\text{C.21})$$

Si $k < p$, entonces hay menos de $p+1$ B-Splines no nulos en $[u_k, u_{k+1})$ a saber $B_0^p(x), B_1^p(x), \dots, B_k^p(x)$, y su suma es:

$$\sum_{i=k-p}^k B_i^p(x) \leq 1$$

5. Derivadas

$$[B_i^p(x)]' = p \left[\frac{B_i^{p-1}(x)}{u_{i+p} - u_i} - \frac{B_{i+1}^{p-1}(x)}{u_{i+p+1} - u_{i+1}} \right] \quad (\text{C.22})$$

La derivada de un B-Spline es combinación de dos B-Splines de grado inferior.

C.2.6. Polinomios de Bernstein como funciones B-Splines

Considerando un caso particular de soporte, con sólo dos nodos y ambos de multiplicidad $p+1$:

$$S = \{\underbrace{0, \dots, 0}_{p+1}, \underbrace{1, \dots, 1}_{p+1}\}$$

Dado que sólo hay un subintervalo no vacío, los B-Splines de grado p están constituidos por un único polinomio en $[0, 1]$. En general, se tiene:

$$\begin{aligned} B_{p-k}^k(x) &= (1-x)^k \\ B_{p-k+j}^k(x) &= \binom{k}{j} x^j (1-x)^{k-j} \quad j = 1, \dots, k-1 \\ B_p^k(x) &= x^k \end{aligned} \quad (\text{C.23})$$

Por tanto $B_{p-k}^k(x), B_{p-k+1}^k, \dots, B_p^k(x)$ son los $k+1$ polinomios de Bernstein de grado k .

C.3. Curvas B-Spline

Dado un soporte

$$S = \{u_0, u_1, \dots, u_m\} = \{\underbrace{x_0, \dots, x_0}_{r_0}, \dots, \underbrace{x_k, \dots, x_k}_{r_k}\}$$

una curva B-Spline no paramétrica de grado p es cualquier combinación lineal de los B-Spline de grado p de la base asociada al soporte, es decir, un elemento del espacio vectorial generado por dichos B-Splines:

$$C(x) = \sum_{i=0}^n P_i B_i^p(x) \quad x \in [u_0, u_m] \quad (\text{C.24})$$

donde el número de coeficientes, $n+1$, es el número de funciones de la base, y:

$$n+1 = m-p$$

En general, interesa considerar curvas paramétricas, para lo cual, se escribirá:

$$C(x) = \sum_{i=0}^n P_i B_i(u) \quad x \in [u_0, u_m]$$

donde los coeficientes P_i tienen dos o tres componentes. A estos puntos (coeficientes) P_0, P_1, \dots, P_n se les denomina *puntos de control* de la curva y a la poligonal que los une,

poligonal de control o polígono de De Boor.

Las curvas de Bézier y B-Splines no son tipos distintos de curvas, sino reparametrizaciones distintas para curvas constituidas por tramos polinomiales y con ciertos órdenes de continuidad en la unión de dos tramos.

C.3.1. Propiedades de las curvas B-Spline

Supongamos que el primero de los nodos, u_0 , es de multiplicidad $p + 1$, siendo p el grado de los B-Splines. Entonces:

$$\begin{aligned} B_0^k(u_0) &= \dots = B_{p-k-1}^k(u_0) = 0 \\ B_{p-k}^k(u_0) &= 1 \\ B_{p-k+1}^k(u_0) &= \dots = B_{m-k-1}^k(u_0) = 0 \end{aligned} \tag{C.25}$$

Y para los B-Splines de grado p :

$$\begin{aligned} B_0^p(u_0) &= 1 \\ B_1^p(u_0) &= \dots = B_{m-p-1}^p(u_0) = 0 \end{aligned}$$

Es decir, que todos los B-Splines de grado p , excepto $B_0^p(x)$, se anulan en u_0 .

Si $C(u)$ es una curva B-Spline de grado p definida sobre el soporte con u_0 de multiplicidad $p + 1$:

$$C(x) = \sum_{i=0}^n P_i B_i(u)$$

por tanto

$$C(u_0) = P_0$$

y entonces la curva interpola al punto de control P_0 .

Análogamente, si u_m es de multiplicidad $p + 1$, todos los B-Spline excepto $B_{m-p-1}^p(x)$, se anulan en u_m , y

$$C(u_m) = P_m$$

Ahora se pueden enunciar las propiedades de las curvas B-Spline:

1. Interpolación de los puntos de control extremos.

Si el soporte $S = \{u_0, \dots, u_m\}$ es no periódico (campled):

$$S = \underbrace{\{x_0, \dots, x_0\}}_{p+1}, \dots, x_{h-1}, \underbrace{\{x_h, \dots, x_h\}}_{p+1}$$

y cualquier curva B-Spline verifica:

$$\begin{aligned} C(u_0) &= P_0 \\ C(u_m) &= P_m \end{aligned}$$

Los soportes campled hace referencia al hecho de que la curva está atada y son los únicos que se utilizan en la práctica.

2. Derivadas en los puntos extremos.

Para una curva B-Spline se tiene que

$$C'(u_0) = \sum_{i=0}^n P_i [B_i^p(u_0)]' = \frac{p}{u_{p+1} - u_1} (P_1 - P_0) \quad (\text{C.26})$$

lo que indica que el vector tangente a la curva en el punto P_0 tiene la dirección de $P_1 - P_0$. Análogamente:

$$C'(u_m) = \sum_{i=0}^n P_i [B_i^p(u_m)]' = \frac{p}{u_{m-1} - u_{m-p-1}} (P_n - P_{n-1}) \quad (\text{C.27})$$

y la tangente a la curva en P_n tiene la dirección de $P_n - P_{n-1}$.

3. Control local.

Una curva B-Spline con puntos de control P_0, P_1, \dots, P_n :

$$C(x) = \sum_{i=0}^n P_i B_i^p(u)$$

y sea $\bar{C}(x)$ la curva obtenida cambiando un único punto de control P_i por \bar{P}_i :

$$\bar{C}(x) = \sum_{i=0}^n \bar{P}_i B_i(u)$$

La diferencia entre ambas curvas para el punto k será:

$$\bar{C}(x) - C(x) = \sum_{i=0}^n (\bar{P}_k - P_k) B_k^p(u) \quad (\text{C.28})$$

como $B_k^p(u)$ es nulo fuera del intervalo $[u_k, u_{k+p+1})$, entonces:

$$\bar{C}(x) - C(x) = 0 \quad \forall x \notin [u_k, u_{k+p+1})$$

lo que significa que ambas curvas son idénticas, excepto, posiblemente, en $[u_k, u_{k+p+1})$ y la modificación de un punto de control afecta solo a $p + 1$ tramos de la curva.

4. Restricción a la envolvente convexa.

Para un valor concreto del parámetro $u = \bar{u}$, una curva B-Spline puede considerarse una combinación lineal de los puntos de control:

$$C(\bar{u}) = \sum_{i=0}^n P_i B_i^p(\bar{u})$$

pero si $\bar{u} \in [u_k, u_{k+1})$, entonces sólo $B_{k-p}^p(\bar{u}), \dots, B_k^p(\bar{u})$ son distintos de 0, luego:

$$C(\bar{u}) = \sum_{i=k-p}^k P_i B_i^p(\bar{u}) \quad (\text{C.29})$$

Si $k \geq p$, entonces:

$$\sum_{i=k-p}^k B_i^p(\bar{u}) = 1$$

por tanto, al ser $B_i^p(\bar{u}) \geq 0$ es una combinación convexa de los puntos P_i . En consecuencia $C(\bar{u})$ está en la envolvente convexa de los $p + 1$ puntos P_{k-p}, \dots, P_k . Si $k < p$, entonces sólo se puede asegurar que será $B_i^p(\bar{u}) \leq 1$ pero seguirá siendo que $C(\bar{u})$ está en la envolvente convexa.

5. Disminución de la variación.

Una recta arbitraria no intersecta a una curva B-Spline en más puntos de los que intersecta a la poligonal de control.

6. Invarianza afín.

Si el soporte es no periódico, entonces:

$$\sum_{i=0}^{m-p-1} B_i^p(x) = 1 \quad x \in [u_0, u_m)$$

y la curva B-Spline definida sobre el soporte es invariante por transformaciones afines, queriendo decir que para hallar la curva transformada basta con transformar los puntos de control y generar una nueva curva a partir de ellos.

7. Nodos con multiplicidad $p + 1$ y p .

Supóngase que un nodo tiene multiplicidad $p + 1$:

$$u_{i-1} < u_i = u_{i+1} = \dots u_{i+p} < u_{i+p+1}$$

Si ahora $C(u)$ es una curva B-Spline

$$C(u) = \sum_{i=0}^n P_i B_i^p(x)$$

entonces:

$$\begin{aligned} C(u_i^-) &= P_{i-1} \\ C(u_i^+) &= P_i \end{aligned}$$

y la curva es discontinua en $u = u_i$ si $P_{i-1} \neq P_i$.

Ahora, si el nodo u_i es de multiplicidad p , entonces todos los B-Splines son continuos en $u = u_i$, $B_i(u_i) = 1$ y en consecuencia $B_k(u_i) = 0$ para $k \neq i$, con lo cual $C(u_i) = P_i$ y la curva en este caso interpola al punto de control u_i , siendo continua en él.

8. Puntos de control múltiples.

Si un punto de control es de multiplicidad $p + 1$, es decir, $P_i = P_{i+1} = \dots = P_{i+p}$, entonces en el intervalo $[u_{i+p}, u_{i+p+1})$ los B-Spline no nulos son:

$$B_i^p(x), B_{i+1}^p(x), \dots, B_{i+p}^p(x)$$

y la curva será:

$$C(u) = \sum_{k=0}^n P_k B_k^p(u) = \sum_{k=i}^{i+p} P_k B_k^p(u) = P_i \left[\sum_{k=i}^{i+p} B_k^p(u) \right] = P_i$$

luego la curva se reduce al punto de control P_i para $[u_{i+p}, u_{i+p+1})$

Si el punto de control P_i es de multiplicidad p , $P_i = P_{i+1} = \dots = P_{i+p-1}$, entonces el intervalo $[u_{i+p-1}, u_{i+p})$ los B-Splines no nulos son:

$$B_{i-1}^p(x), \dots, B_{i+p-1}^p(x)$$

y la curva es

$$C(u) = \sum_{k=i-1}^{i+p-1} P_k B_k^p(u) = P_{i-1} B_{i-1}^p(u) + P_i [1 - B_{i-1}^p(u)]$$

para $u \in [u_{i+p-1}, u_{i+p})$, $C(u)$ es un tramo recto contenido en el segmento $\overline{P_{i-1}P_i}$.

Estas propiedades hacen muy eficiente la manipulación de curvas B-Splines, particularmente si el soporte es no periódico.

El siguiente es un algoritmo que calcula las curvas B-Splines en su soporte y enseguida se muestra una gráfica obtenida con este.

Algoritmo 4.06. Cálculo de una Curva B-Spline.

```
clear all
close all
clc
S=[0 0 0 0 1 2 3 4 4 4 5 6 7 7 7 7];
grado=3; m=length(S);
count=0; orden=0;
for i=1:m-1
    if S(i+1)==S(i)
        u(i)=0;
    else
        u(i)=1;
    end
end
end
```

```

for i=1:m-1
for j=1:m-1
para((m-1)*(i-1)+j,1)=orden; para((m-1)*(i-1)+j,2)=i;
para((m-1)*(i-1)+j,3)=S(j); para((m-1)*(i-1)+j,4)=S(j+1);
if i==j
B((m-1)*(i-1)+j,1)=u(j);
end
count=count+1;
end
end
curvas(1,1)=1; curvas(1,2)=count;
for orden=1:grado
curvas(orden+1,1)=count+1;
for i=1:m-1-orden
e=-S(i); f=S(i+orden+1);
b=S(i+orden)+e; d=f-S(i+1);
for j=1:m-1
count=count+1;
para(count,1)=orden; para(count,2)=i;
para(count,3)=S(j); para(count,4)=S(j+1);
miu=B(count-(m-orden)*(m-1),:);
bi=miu(orden:-1:1); bi(orden+1)=0;
miu(orden+1)=0; miu=miu(orden+1:-1:1);
miu=miu*e; a=bi+miu;
miu=B(count-(m-orden-1)*(m-1),:);
bi=-miu(orden:-1:1); bi(orden+1)=0;
miu(orden+1)=0; miu=miu(orden+1:-1:1);
miu=miu*f; c=bi+miu;
if b==0
a=0; b=1;
elseif d==0
c=0; d=1;
end
Bi=a/b+c/d; Bi=Bi(orden+1:-1:1);
for k=1:orden+1

```

```

B(count,k)=Bi(k);
end
end
end
curvas(orden+1,2)=count;
end

% HASTA ACA B-SPLINES POR RECURRENCIA

conter=0;
for j=curvas(grado+1,1):(m-1):count-1%desde la 1ra fila del ultimo grado
for i=1+grado:m-1-grado

% elimina los nullos de c/grado inicial & final

conter=conter+1;
bi=B(j+i-1,:);

% en la posicion j incrementa i-1 (por c/intervalo)
bi=bi(grado+1:-1:1);

% se invierte y se guarda asi
for k=1:grado+1          % SE GUARDA EN "BI" 1x1
BI(conter,k)=bi(k);
end
par(conter,1)=para(j+i-1,1); par(conter,2)=para(j+i-1,2);
par(conter,3)=para(j+i-1,3); par(conter,4)=para(j+i-1,4);
end
end
% get control points
axis([-2,2,-2,2])
hold on
for i=1:m-1-grado

%(descontando los 4 "dos*grado al inicio y al final")

```

```

a=ginput(1);
plot(a(1),a(2),'o')           % lo dibuja con una "o"
xy(i,1)=a(1);                 % guarda cada componente del "click"
xy(i,2)=a(2);                 % 1ra columna para X's 2da--> Y's
end
% each grafic de "B"          % sum(u)=5
Cx=zeros(m-1-2*grado,orden+1);Cy=Cx;

% asumimos que no hay mas ptos del soporte repetidos "solo" (1ro y n)

for j=0:par(conter,2)-1       %hasta la ultima grafica
for i=1:m-1-2*grado           % .n-1- "los 2grado mencionados antes"
cx=BI(j*(m-1-2*grado)+i,:)*xy(j+1,1);
cy=BI(j*(m-1-2*grado)+i,:)*xy(j+1,2);
for k=1:orden+1               % SE GUARDA EN "Cx,Cy" 1x1
Cx(i,k)=Cx(i,k)+cx(k);
Cy(i,k)=Cy(i,k)+cy(k);
end
end
end

% ===== graficas =====

t=linspace(S(1),S(m),500);    % 100 por defecto
to=length(t);
for j=1+grado:m-1-grado       % por c/intervalo supuestamente no nulo
cx=Cx(j-grado,:);
cy=Cy(j-grado,:);
for i=1:to
if t(i)>=S(j)
X(i)=polyval(cx,t(i));
Y(i)=polyval(cy,t(i));
end
end
end

```

```

end
plot(xy(:,1),xy(:,2))
plot(X,Y, 'r')

```

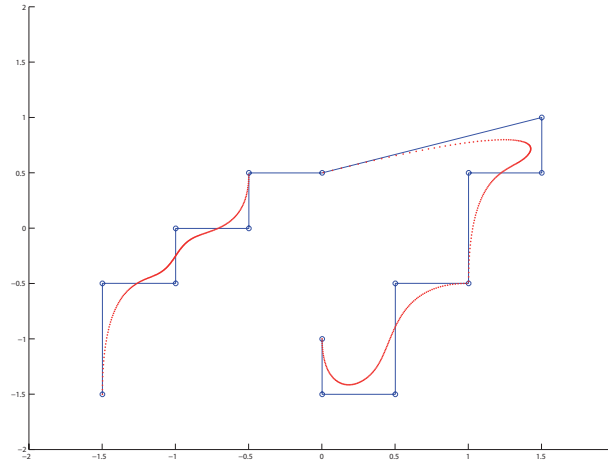


Figura C.15: Curva B-Spline

C.3.2. Inserción de nodos. Algoritmo de Boehm

En el contexto de los B-Splines se trata de expresar una misma curva en función de un número mayor de puntos de control. Esta técnica puede ser utilizada con el fin de subdividir la curva.

La idea fundamental es insertar un nuevo nodo en el soporte dado

$$S = \{u_0, u_1, \dots, u_k, u_{k+1}, \dots, u_m\}$$

para obtener

$$\bar{S} = \{u_0, u_1, \dots, u_k, \bar{u}, u_{k+1}, \dots, u_m\}$$

donde $\bar{u} \in [u_k, u_{k+1})$ es el nuevo nodo que va a ser insertado.

Dado que el número de puntos de control, $n+1$, y el de nodos están relacionados por:

$$n = m - p - 1$$

si la curva originalmente es

$$C(u) = \sum_{i=0}^n P_i B_i^p(u) \quad (\text{C.30})$$

con el nuevo soporte habrá $n + 2$ puntos de control:

$$C(u) = \sum_{i=0}^{n+1} \bar{P}_i \bar{B}_i^p(u) \quad (\text{C.31})$$

Se pretende que estas dos curvas sean idénticas, entonces:

$$\sum_{i=0}^n P_i B_i^p(u) = \sum_{i=0}^{n+1} \bar{P}_i \bar{B}_i^p(u)$$

Para los puntos de control, existe un procedimiento denominado algoritmo de Boehm que se basa en observar que los puntos P_0, \dots, P_{k-p} y P_k, \dots, P_n permanecen invariables y solamente hay que generar p nuevos puntos de control.

Naturalmente, puede insertarse un nodo ya existente, con lo cual su multiplicidad aumentaría en una unidad. En general, si un nodo u_k de multiplicidad r_k se reinserta, adquiere multiplicidad r_{k+1} y se tiene:

$$\begin{aligned} \bar{P}_i &= P_i & i &= 0, \dots, k-p \\ \bar{P}_i &= (1 - \alpha_i) P_{i-1} + \alpha_i P_i & i &= k-p+1, \dots, k-r_k \\ \bar{P}_i &= P_{i-1} & i &= k-r_k+1, \dots, n+1 \end{aligned} \quad (\text{C.32})$$

$$\alpha_i = \frac{\bar{u} - u_i}{u_{i+p} - u_i} \quad (\text{C.33})$$

con lo cual P_0, \dots, P_{k-p} permanecen incambiables y pasan a ser $\bar{P}_0, \dots, \bar{P}_{k-p}, P_{k-r_k}, \dots, P_n$ quedan así mismo iguales, pasando a ser $\bar{P}_{k-r_k}, \dots, \bar{P}_n$ y los puntos de control nuevos son $P_{k-p+1}, \dots, P_{k-r_k}$ (en total $p - r_k$).

- Si $r_k = p - 1$ entonces todos los puntos de control P_0, \dots, P_n permanecen (pero renombrados) y solo uno nuevo se calcula: \bar{P}_{k-p+1} .
- Si $r_k = p$ entonces todos los puntos de control P_0, \dots, P_n permanecen y no se calcula ninguno más nuevo, y \bar{P}_{k-p} aparece duplicado.

- Si $r_k = p + 1$ entonces todos los puntos de control P_0, \dots, P_n permanecen y no se calcula ninguno más nuevo, y \bar{P}_{k-p} aparece duplicado.

Ejemplo:

Sea el soporte:

$$S = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$$

se inserta un nodo que ya existe $u_i = 1$, $k = 4$:

$$\begin{aligned} \bar{P}_i &= P_i & i &= 0, 1 \\ \bar{P}_i &= (1 - \alpha_i) P_{i-1} + \alpha_i P_i & i &= 2, 3 \\ \bar{P}_i &= P_{i-1} & i &= 4, 5, 6, 7 \end{aligned}$$

entonces sólo dos nuevos puntos de control son generados:

$$\begin{aligned} \bar{P}_0 &= (1, 0), \bar{P}_1 = (0, 4), \bar{P}_2 = \left(\frac{3}{2}, 6\right), \bar{P}_3 = \left(\frac{14}{3}, \frac{23}{3}\right) \\ \bar{P}_4 &= (8, 7), \bar{P}_5 = (11, 0), \bar{P}_6 = (15, -1), \bar{P}_7 = (20, 5) \end{aligned}$$

y el soporte es:

$$S = \{0, 0, 0, 0, 1, 1, 2, 3, 4, 4, 4, 4\}$$

Se inserta, de nuevo, el nodo $u_i = 1$ (de multiplicidad $p - 1 = 2$), y ahora $k = 5$:

$$\begin{aligned} \bar{P}_i &= P_i & i &= 0, 1, 2 \\ \bar{P}_i &= (1 - \alpha_i) P_{i-1} + \alpha_i P_i & i &= 3 \\ \bar{P}_i &= P_{i-1} & i &= 4, 5, 6, 7, 8 \end{aligned}$$

y ahora sólo se calcula un nuevo punto de control:

$$\begin{aligned} \bar{P}_0 &= (1, 0), \bar{P}_1 = (0, 4), \bar{P}_2 = \left(\frac{3}{2}, 6\right), \bar{P}_3 = \left(\frac{37}{12}, \frac{41}{6}\right), \bar{P}_4 = \left(\frac{14}{3}, \frac{23}{3}\right) \\ \bar{P}_5 &= (8, 7), \bar{P}_6 = (11, 0), \bar{P}_7 = (15, -1), \bar{P}_8 = (20, 5) \end{aligned}$$

y el soporte es:

$$S = \{0, 0, 0, 0, 1, 1, 1, 2, 3, 4, 4, 4, 4\}$$

Nuevamente, se inserta el nodo $u_i = 1$ (de multiplicidad $p = 3$), y ahora $k = 6$:

$$\begin{aligned}\bar{P}_i &= P_i \quad i = 0, 1, 2, 3 \\ \bar{P}_i &= P_{i-1} \quad i = 4, 5, 6, 7, 8, 9\end{aligned}$$

por ahora no se genera ningún punto de control nuevo y sencillamente se duplica el punto P_3 , y el soporte queda:

$$S = \{0, 0, 0, 0, 1, 1, 1, 1, 2, 3, 4, 4, 4, 4\}$$

El punto P_3 está sobre la curva ya que el nodo ha alcanzado una multiplicidad igual al grado. Ahora se presenta una implementación del Algoritmo de Boehm para la inserción de un nodo.

Algoritmo 4.07. Inserción de un nodo.

```
% Se corre después de Bspline.m

rk=0;
n=m-grado-1;
uk=input('A new value by soport: ')
for i=1:m
if S(i)==uk
rk=rk+1;
end
end
if rk==orden+1
rk=rk+1;
end

% se le añade uno mas si esta en 4, ya que 3 ó menos OK

for i=1:m
if S(i)<=uk
k=i-1;
end
```

```

end
if rk<=grado+1
P=xy;
for i=k-grado+2:k-rk+1
alf(i)=(uk-S(i))/(S(i+grado)-S(i));
P(i,1)=(1-alf(i))*xy(i-1,1)+alf(i)*xy(i,1);
P(i,2)=(1-alf(i))*xy(i-1,2)+alf(i)*xy(i,2);
end
for i=k-rk+2:n+1
P(i,1)=xy(i-1,1);
P(i,2)=xy(i-1,2);
end
Snew=S;
Snew(k+2)=uk;
for i=k+3:m+1
Snew(i)=S(i-1);
end
xy=P;
S=Snew;

% ==== hasta aca insercion, el resto es volver a calcular pa dibujar
% ===== Bspline.m =====

clear B; m=length(S);
count=0; orden=0;
for i=1:m-1
if S(i+1)==S(i)
u(i)=0;
else
u(i)=1;
end
end
for i=1:m-1
for j=1:m-1
para((m-1)*(i-1)+j,1)=orden; para((m-1)*(i-1)+j,2)=i;

```

```

para((m-1)*(i-1)+j,3)=S(j); para((m-1)*(i-1)+j,4)=S(j+1);
if i==j
B((m-1)*(i-1)+j,1)=u(j);
end
count=count+1;
end
end
curvas(1,1)=1; curvas(1,2)=count;
for orden=1:grado
curvas(orden+1,1)=count+1;
for i=1:m-1-orden
e=-S(i); f=S(i+orden+1);
b=S(i+orden)+e; d=f-S(i+1);
for j=1:m-1
count=count+1;
para(count,1)=orden; para(count,2)=i;
para(count,3)=S(j); para(count,4)=S(j+1);
miu=B(count-(m-orden)*(m-1),:);
bi=miu(orden:-1:1); bi(orden+1)=0;
miu(orden+1)=0; miu=miu(orden+1:-1:1);
miu=miu*e; a=bi+miu;
miu=B(count-(m-orden-1)*(m-1),:);
bi=-miu(orden:-1:1); bi(orden+1)=0;
miu(orden+1)=0; miu=miu(orden+1:-1:1);
miu=miu*f; c=bi+miu;
if b==0
a=0; b=1;
elseif d==0
c=0; d=1;
end
Bi=a/b+c/d; Bi=Bi(orden+1:-1:1);
for k=1:orden+1
B(count,k)=Bi(k);
end
end

```

```

end
curvas(orden+1,2)=count;
end
conter=0;
for j=curvas(grado+1,1):(m-1):count-1
for i=1+grado:m-1-grado
conter=conter+1; bi=B(j+i-1,:);
bi=bi(grado+1:-1:1);
for k=1:grado+1
BI(conter,k)=bi(k);
end
par(conter,1)=para(j+i-1,1); par(conter,2)=para(j+i-1,2);
par(conter,3)=para(j+i-1,3); par(conter,4)=para(j+i-1,4);
end
end
Cx=zeros(m-1-2*grado,orden+1);Cy=Cx;
for j=0:par(conter,2)-1
for i=1:m-1-2*grado
cx=BI(j*(m-1-2*grado)+i,:)*xy(j+1,1);
cy=BI(j*(m-1-2*grado)+i,:)*xy(j+1,2);
for k=1:orden+1
Cx(i,k)=Cx(i,k)+cx(k); Cy(i,k)=Cy(i,k)+cy(k);
end
end
end
for j=1+grado:m-1-grado
cx=Cx(j-grado,:);
cy=Cy(j-grado,:);
for i=1:to
if t(i)>=S(j)
X(i)=polyval(cx,t(i));
Y(i)=polyval(cy,t(i));
end
end
end
end

```

```

plot(xy(:,1),xy(:,2),'b')
plot(xy(:,1),xy(:,2),'r*')
plot(X,Y,'y')
else
uk=input('Ese soporte no puede tener mas multiplicidad... gracias: ')
end

```

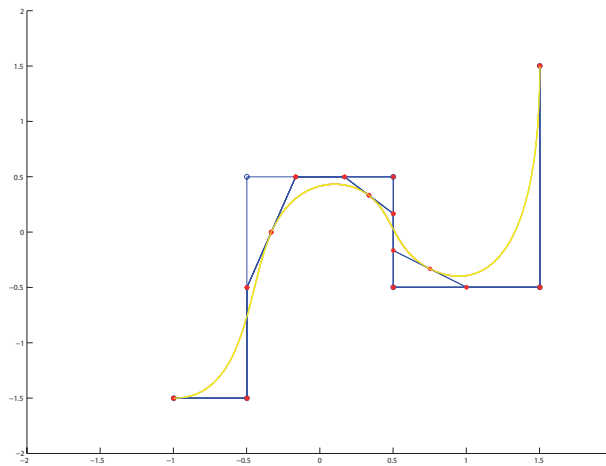
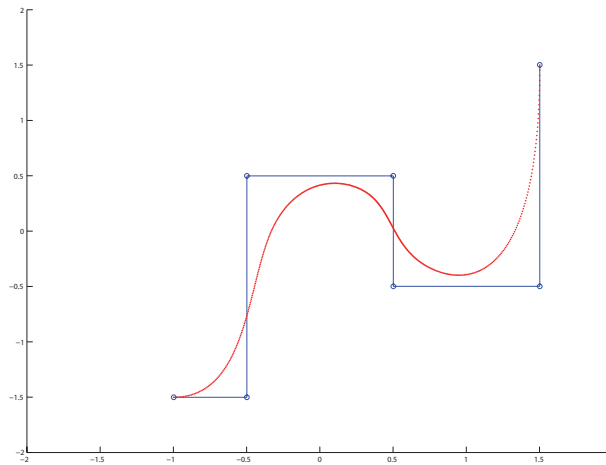


Figura C.16: Inserción de un nodo

El polígono de control obtenido tras la inserción de un nodo constituye una

aproximación a la curva mejor de lo que era el polígono inicial. De hecho, si en un soporte se inserta una sucesión de nodos densa en $[u_0, u_m]$, la sucesión de polígonos de control converge a la curva. Con una inserción apropiada de nuevos nodos, el polígono de control es suficientemente aproximado a la curva, para que pueda ser dibujado directamente.

C.3.3. Inserción múltiple. Algoritmo de De Boor

Un punto \bar{u} puede ser insertado en un soporte $S = \{u_0, \dots, u_m\}$ produciendo un nuevo conjunto de puntos de control que definen la misma curva original. Si este proceso se repite volviéndose a insertar el nodo \bar{u} hasta que tenga multiplicidad p resulta en un polígono de control que ya no puede ser *refinado*. Este esquema iterativo se conoce como algoritmo de De Boor y lo interesante es que el punto de control P_k obtenido en la última etapa es precisamente $C(\bar{u})$, es decir, el punto de la curva B-Spline correspondiente al valor $u = \bar{u}$ y k es tal que $\bar{u} \in [u_k, u_{k+1})$. Por tanto, este algoritmo es un procedimiento para evaluar una curva B-Spline en un punto.

El método funciona dando P_0, \dots, P_n puntos de control iniciales y $\bar{u} \in [u_k, u_{k+1})$, se construyen las sucesiones:

$$P_i^0 = P_i \quad i = 0, \dots, n$$

y para $j = 1, \dots, p - r$:

$$P_i^j = (1 - t_i^j) P_{i-1}^{j-1} + t_i^j P_i^{j-1} \quad i = k - p + 1, \dots, k - r \quad (\text{C.34})$$

donde r es la multiplicidad inicial de \bar{u} en el soporte ($r = 0$ si \bar{u} no pertenecía al soporte) y

$$t_i^j = \frac{\bar{u} - u_i}{u_{i+p-j+i} - u_i} \quad (\text{C.35})$$

Entonces:

$$P_{k-r}^{p-r} = C(\bar{u})$$

C.3.4. Reducción de la multiplicidad

El objetivo es eliminar una ocurrencia de un nodo múltiple para que la multiplicidad disminuya en una unidad.

Considerando entonces una curva B-Spline con puntos de control P_0, \dots, P_n :

$$C(u) = \sum_{i=0}^n P_i B_i(u) \quad (\text{C.36})$$

definida sobre el soporte $S = \{u_0, \dots, u_m\}$, suponiendo que el nodo $u_j < u_{j+1}$ es de multiplicidad $r_j > 1$.

Si el nodo u_j es efectivamente reducible, quiere esto decir que la curva $C(u)$ puede expresarse como:

$$C(u) = \sum_{i=0}^{n-1} \bar{P}_i \bar{B}_i(u) \quad (\text{C.37})$$

Entonces, el nuevo soporte será $\bar{S} = \{u_0, \dots, u_{m-1}\}$, y:

$$\begin{aligned} \bar{u}_i &= u_i & i &= 0, \dots, j-1 \\ \bar{u}_i &= u_{i+1} & i &= j, \dots, m-1 \end{aligned}$$

Invirtiendo las expresiones halladas para la inserción de un nodo, se encuentra que:

$$\begin{aligned} \bar{P}_i &= P_i & i &= 0, \dots, j-p-1 \\ \bar{P}_i &= \beta_i P_i - (1 - \beta_i) \bar{P}_{i-1} & i &= j-p, \dots, j-r_j-1 \\ \bar{P}_i &= P_{i-1} & i &= j-r_j, \dots, n-1 \end{aligned} \quad (\text{C.38})$$

$$\beta_i = \frac{u_{i+p+1} - u_i}{u_j - u_i} \quad (\text{C.39})$$

que son las expresiones utilizadas para la eliminación de un nodo.

Algoritmo 4.08. Reducción de la multiplicidad.

```
% se supone que sabemos cual es el soporte y así eliminarlo
% se corre despues de Bspline.m ó BsplineInsercionNodos.m
```

```
ok=0;
```

```
%multiplicidad del nodo a reducir, si sigue 0 no existe
```

```
J=0;
```

```

% ubicacion del nodo to eliminate j:contador de for
% previsualiza el soporte pa'que no las caguen

uk=input('A value by soport S to delete: ')
m=length(S);
n=m-grado-1;% -1?
for i=1:m
if S(i)==uk
ok=ok+1;
end
end
for i=1:m% <--- da los valores de 1st
if S(i)==uk
J=i;
end
end
if ok==0
uk=input('no puede eliminar el nodo ya que no existe: ')
else
for i=1:J-grado-1
Pred(i,1)=xy(i,1);
Pred(i,2)=xy(i,2);
end
for i=J-grado:J-ok-1
bet(i)=(S(i+grado+1)-S(i))/(uk-S(i));%(i+grado+1)
Pred(i,1)=bet(i)*xy(i,1)-(bet(i)-1)*Pred(i-1,1);
Pred(i,2)=bet(i)*xy(i,2)-(bet(i)-1)*Pred(i-1,2);
end
for i=J-ok:n-1
Pred(i,1)=xy(i+1,1);
Pred(i,2)=xy(i+1,2);
end
xy=Pred;
for i=1:J-1
Sred(i)=S(i);

```

```

end
for i=J:m-1
Sred(i)=S(i+1);
end
S=Sred;
clear Sred;
clear Pred

% ==== hasta aca Reduccion, despues recalcula y dibuja
% ===== Bspline.m =====

clear B; m=length(S);
count=0; orden=0;
for i=1:m-1
if S(i+1)==S(i)
u(i)=0;
else
u(i)=1;
end
end
for i=1:m-1
for j=1:m-1
para((m-1)*(i-1)+j,1)=orden; para((m-1)*(i-1)+j,2)=i;
para((m-1)*(i-1)+j,3)=S(j); para((m-1)*(i-1)+j,4)=S(j+1);
if i==j
B((m-1)*(i-1)+j,1)=u(j);
end
count=count+1;
end
end
curvas(1,1)=1; curvas(1,2)=count;
for orden=1:grado
curvas(orden+1,1)=count+1;
for i=1:m-1-orden
e=-S(i); f=S(i+orden+1);

```

```

b=S(i+orden)+e; d=f-S(i+1);
for j=1:m-1
count=count+1;
para(count,1)=orden; para(count,2)=i;
para(count,3)=S(j); para(count,4)=S(j+1);
miu=B(count-(m-orden)*(m-1),:);
bi=miu(orden:-1:1); bi(orden+1)=0;
miu(orden+1)=0; miu=miu(orden+1:-1:1);
miu=miu*e; a=bi+miu;
miu=B(count-(m-orden-1)*(m-1),:);
bi=-miu(orden:-1:1); bi(orden+1)=0;
miu(orden+1)=0; miu=miu(orden+1:-1:1);
miu=miu*f; c=bi+miu;
if b==0
a=0; b=1;
elseif d==0
c=0; d=1;
end
Bi=a/b+c/d; Bi=Bi(orden+1:-1:1);
for k=1:orden+1
B(count,k)=Bi(k);
end
end
end
curvas(orden+1,2)=count;
end
conter=0;
for j=curvas(grado+1,1):(m-1):count-1
for i=1+grado:m-1-grado
conter=conter+1; bi=B(j+i-1,:);
bi=bi(grado+1:-1:1);
for k=1:grado+1
BI(conter,k)=bi(k);
end
par(conter,1)=para(j+i-1,1); par(conter,2)=para(j+i-1,2);

```

```

par(conter,3)=para(j+i-1,3); par(conter,4)=para(j+i-1,4);
end
end
Cx=zeros(m-1-2*grado,orden+1);Cy=Cx;
for j=0:par(conter,2)-1
for i=1:m-1-2*grado
cx=BI(j*(m-1-2*grado)+i,:)*xy(j+1,1);
cy=BI(j*(m-1-2*grado)+i,:)*xy(j+1,2);
for k=1:orden+1
Cx(i,k)=Cx(i,k)+cx(k); Cy(i,k)=Cy(i,k)+cy(k);
end
end
end
for j=1+grado:m-1-grado
cx=Cx(j-grado,:);
cy=Cy(j-grado,:);
for i=1:to
if t(i)>=S(j)
X(i)=polyval(cx,t(i));
Y(i)=polyval(cy,t(i));
end
end
end
plot(xy(:,1),xy(:,2),'y')
plot(xy(:,1),xy(:,2),'g.')
plot(X,Y,'g')
end

```

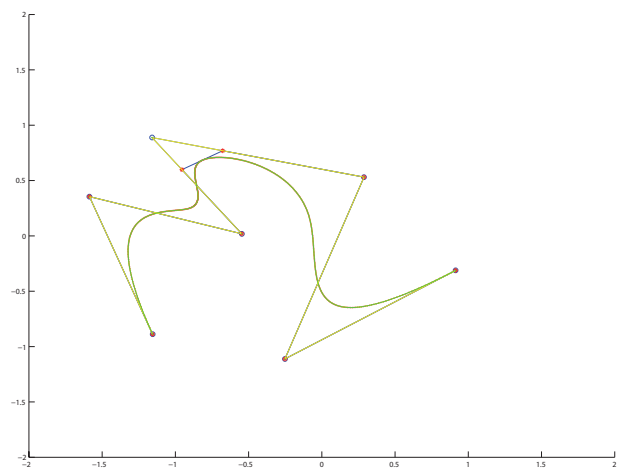
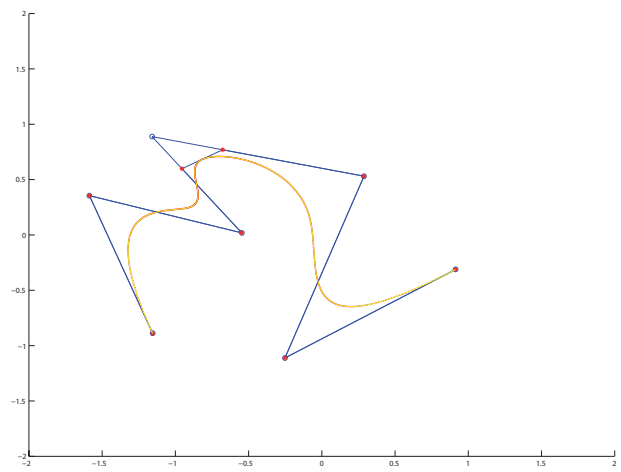
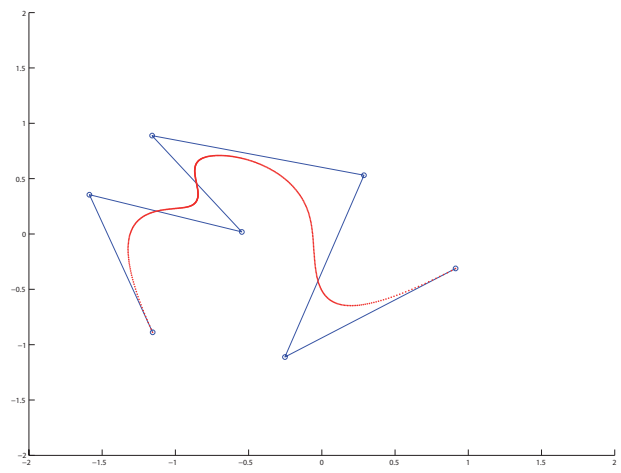


Figura C.17: Reducción de un nodo
221

C.4. Interpolación con curvas B-Spline

En el contexto de curvas paramétricas, el planteamiento es el de interpolar puntos Q_0, \dots, Q_1 en \mathbb{R}^2 o \mathbb{R}^3 en valores t_0, \dots, t_n del parámetro, es decir, obtener la curva $C(u)$ verificando:

$$C(t_k) = Q_k \quad k = 0, \dots, n$$

donde $C(u)$ consiste en dos o tres componentes siendo cada uno de ellos una función invariada de u perteneciente a un cierto espacio vectorial de funciones:

$$C(u) = \sum_{i=0}^n P_i B_i(u) \quad P_i \in \mathbb{R}^2 \text{ o } \mathbb{R}^3$$

donde $\beta = \{B_0(u), \dots, B_n(u)\}$ es una base del espacio vectorial, y la no anulación del determinante:

$$\begin{vmatrix} B_0(t_0) & B_1(t_0) & \dots & B_n(t_0) \\ B_0(t_1) & B_1(t_1) & \dots & B_n(t_1) \\ \dots & \dots & \dots & \dots \\ B_0(t_n) & B_1(t_n) & \dots & B_n(t_n) \end{vmatrix} \neq 0$$

garantiza que puede realizarse unívocamente la interpolación componente a componente.

C.4.1. Soporte de Interpolación basado en longitud de cuerda

Este soporte recoge en cierta medida la geometría de los puntos de interpolación, haciendo que los incrementos $t_i - t_{i-1}$ sean proporcionales a las longitudes de los segmentos $\overline{Q_{i-1}Q_i}$:

$$\begin{aligned} t_0 &= 0 \\ t_i &= t_{i-1} + \frac{\|Q_i - Q_{i-1}\|}{d} \quad i = 1, 2, \dots, n \\ t_n &= 1 \end{aligned} \tag{C.40}$$

donde

$$d = \|Q_1 - Q_0\| + \dots + \|Q_n - Q_{n-1}\| \tag{C.41}$$

C.4.2. Bases de B-Spline para interpolación

Se pretende interpolar $n + 1$ puntos Q_0, \dots, Q_n , entonces se debe considerar un espacio de curvas B-Spline de dimensión $n + 1$:

$$C(u) = \sum_{i=0}^n P_i B_i^p(u)$$

donde los B-Splines serán los correspondientes a un cierto soporte de nodos:

$$S = \{u_0, \dots, u_m\}$$

con $m = n + p + 1$.

Para un conjunto dados de puntos a interpolar Q_0, \dots, Q_n y un conjunto de nodos de interpolación t_0, \dots, t_n y una vez establecido el grado p , existen infinitas elecciones del soporte de nodos u_0, \dots, u_m para definir los correspondientes B-Spline y así construir la curva interpoladora.

Es común tomar $u_0 = t_0$ y $u_m = t_n$ y elegir para el soporte de nodos una distribución no periódica:

$$S = \{\underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{1, \dots, 1}_{p+1}\} \quad (\text{C.42})$$

con $0 < u_{p+1} < u_{p+2} < \dots < u_{m-p-1} < 1$. De esta forma los nodos interiores son simples y la curva es de clase C^{p-1} . Los $n - p$ nodos interiores se eligen promediando p de los nodos de interpolación:

$$u_{i+p} = \frac{t_i + t_{i+1} + \dots + t_{i+p-1}}{p} \quad i = 1, \dots, n - p \quad (\text{C.43})$$

El sistema de ecuaciones que obtiene los puntos de control es:

$$\sum_{i=0}^n P_i B_i(t_k) = Q_k \quad k = 0, \dots, n$$

para el caso de un B-Spline cúbico, se tiene por matriz:

$$\begin{bmatrix} 1 & \dots & 0 & \dots & \dots & \dots & \dots & 0 \\ B_0(t_1) & \dots & B_3(t_1) & & & & & \vdots \\ B_0(t_2) & \dots & B_3(t_2) & B_4(t_2) & & & & \vdots \\ \vdots & & & & & & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & & & & \vdots \\ \vdots & & & & & B_{n-4}(t_{n-2}) & B_{n-3}(t_{n-2}) & \dots & B_n(t_{n-2}) \\ \vdots & & & & & & B_{n-3}(t_{n-1}) & \dots & B_n(t_{n-1}) \\ 0 & \dots & \dots & \dots & \dots & 0 & \dots & 1 \end{bmatrix}$$

Aunque parezca contradictorio el predecir, en general, 5 B-Splines no nulos para t_i , cuando se sabe que sólo pueden ser a lo sumo 4 de ellos no nulos (B-Spline cúbico), realmente no ocurre así, pues o bien el primero, o bien el último en cada fila es de hecho nulo.

A continuación se presenta un algoritmo de interpolación de $n + 1$ puntos mediante una curva B-Spline cúbica.

Algoritmo 4.09. Interpolación de $n + 1$ puntos mediante una curva B-Spline Cúbica.

```
clear all
close all
clc
axis([-1,11,-1,10])
hold on
count=0; orden=0;
n=6; grado=3;
m=n+grado+1;
d=0;           % inicio contadores en cero
t=0;           % el primer dato del soporte es 0
for i=1:n
a=ginput(1);
```

```

plot(a(1),a(2),'o')
Q(i,1)=a(1);
Q(i,2)=a(2);
end
for i=2:n

%calculo la longitud total de la poligonal que une todos los nodos

d=d+((Q(i,1)-Q(i-1,1))^2+(Q(i,2)-Q(i-1,2))^2)^0.5;
end
for i=1:n-1

%desde el soporte inicial se suma la distancia hasta
el siguiente punto y lo divide en la poligonal

t(i+1)=t(i)+(((Q(i+1,1)-Q(i,1))^2+(Q(i+1,2)-Q(i,2))^2)^0.5)/d;
end
for i=5:m-4          % sin los 4 primer-ultimos
S(i)=(t(i-1)+t(i-2)+t(i-3))/3;
end
for j=i+1:m
%los primeros son 0 por defecto, se llenan los ultimos con 1

S(j)=t(n);
end

% Bsplines POR RECURRENCIA

for i=1:m-1
if S(i+1)==S(i)
u(i)=0;
else
u(i)=1;
end
end
end

```

```

for i=1:m-1
for j=1:m-1
para((m-1)*(i-1)+j,1)=orden; para((m-1)*(i-1)+j,2)=i;
para((m-1)*(i-1)+j,3)=S(j); para((m-1)*(i-1)+j,4)=S(j+1);
if i==j
B((m-1)*(i-1)+j,1)=u(j);
end
count=count+1;
end
end
curvas(1,1)=1; curvas(1,2)=count;
for orden=1:grado
curvas(orden+1,1)=count+1;
for i=1:m-1-orden
e=-S(i); f=S(i+orden+1);
b=S(i+orden)+e; d=f-S(i+1);
for j=1:m-1
count=count+1;
para(count,1)=orden; para(count,2)=i;
para(count,3)=S(j); para(count,4)=S(j+1);
miu=B(count-(m-orden)*(m-1),:);
bi=miu(orden:-1:1); bi(orden+1)=0;
miu(orden+1)=0; miu=miu(orden+1:-1:1);
miu=miu*e; a=bi+miu;
miu=B(count-(m-orden-1)*(m-1),:);
bi=-miu(orden:-1:1); bi(orden+1)=0;
miu(orden+1)=0; miu=miu(orden+1:-1:1);
miu=miu*f; c=bi+miu;
if b==0
a=0; b=1;
elseif d==0
c=0; d=1;
end
Bi=a/b+c/d; Bi=Bi(orden+1:-1:1);
for k=1:orden+1

```

```

B(count,k)=Bi(k);
end
end
end
curvas(orden+1,2)=count;
end
conter=0;
for j=curvas(grado+1,1):(m-1):count-1
for i=1+grado:m-1-grado
conter=conter+1;
bi=B(j+i-1,:);
bi=bi(grado+1:-1:1);
for k=1:grado+1
BI(conter,k)=bi(k);
end
par(conter,1)=para(j+i-1,1); par(conter,2)=para(j+i-1,2);
par(conter,3)=para(j+i-1,3); par(conter,4)=para(j+i-1,4);
end
end

% Matriz B evaluada en su soporte t

clear B
B(1,1)=1; B(n,n)=1; %el primero y el ultimo son 1
for j=2:n-1          %desde la segunda fila a la penultima
for i=1:n           %todos las curvas q conforma Bi
ba=0;
for k=1:m-7        % (-)los 3 primeros y 4 ultimos
if t(j)>=par((i-1)*(m-7)+k,3) & t(j)<par((i-1)*(m-7)+k,4)%((i-1)*(m-7)
%cada curva Bi
d=BI((i-1)*(m-7)+k,:);

% si esta en el intervalo busca ese Bi en ese intervalo
ba=ba+polyval(d,t(j)); %evalua en el tj
end

```

```

end
B(j,i)=ba;           %lo ubica en la matriz
end
end
P(:,1)=inv(B)*Q(:,1) % son los X
P(:,2)=inv(B)*Q(:,2) % son los Y
Cx=zeros(m-1-2*grado,orden+1);Cy=Cx;
for j=0:par(conter,2)-1
for i=1:m-1-2*grado
cx=BI(j*(m-1-2*grado)+i,:)*P(j+1,1);

% solo cambie xy -> P

cy=BI(j*(m-1-2*grado)+i,:)*P(j+1,2);
for k=1:orden+1
Cx(i,k)=Cx(i,k)+cx(k);
Cy(i,k)=Cy(i,k)+cy(k);
end
end
end
T=linspace(S(1),S(m),500);
to=length(T);
for j=1+grado:m-1-grado
cx=Cx(j-grado,:);
cy=Cy(j-grado,:);
for i=1:to
if T(i)>=S(j)
X(i)=polyval(cx,T(i));
Y(i)=polyval(cy,T(i));
end
end
end
plot(X,Y,'.r')
plot(P(:,1),P(:,2))

```

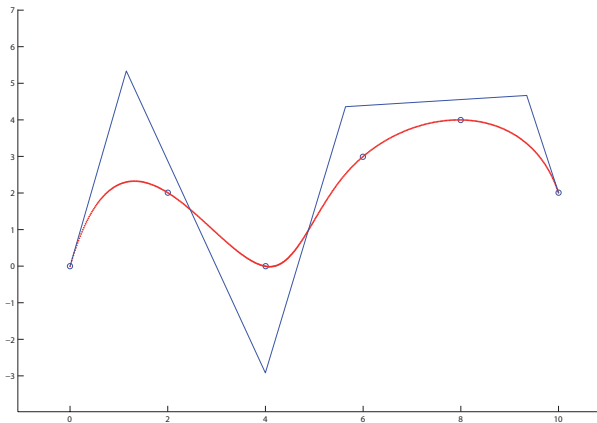


Figura C.18: Interpolación mediante B-Splines

C.4.3. Interpolación con condiciones en los extremos

Otra alternativa consiste en tomar un soporte con nodos interiores iguales a los de interpolación excepto $t_0 = 0$ y $t_n = 1$ (que ya están incluidos):

$$S = \{\underbrace{0, \dots, 0}_{p+1}, t_1, \dots, t_{n-1}, \underbrace{1, \dots, 1}_{p+1}\} \quad (\text{C.44})$$

Sin embargo, ahora hay más nodos de los necesarios. La dimensión del espacio de splines definidos sobre este soporte es: $n + p = (n + 1) + (p - 1)$; existen entonces $p - 1$ parámetros más que puntos interiores a interpolar. Estos parámetros pueden determinarse imponiendo condiciones adicionales a las de interpolación. Para $p = 3$ (spline cúbico), quedarían dos parámetros libres, lo cual se soluciona imponiendo valores para las primeras derivadas en los extremos (o las derivadas segundas en los extremos).

Un spline cúbico definido sobre un soporte del tipo:

$$S = \{0, 0, 0, 0, t_1, \dots, t_{n-1}, 1, 1, 1, 1\}$$

depende de $n + p = n + 3$ puntos de control que por conveniencia se denominan: $P_0, P_1, \dots, P_n, P_{n+1}, P_{n+2}$. Y, entonces:

$$C(u) = \sum_{i=0}^{n+2} P_i B_i^p(u) \quad (\text{C.45})$$

Imponiendo las $n + 1$ condiciones de interpolación $C(t_k) = Q_k$ para $k = 0, \dots, n$:

$$\sum_{i=0}^{n+2} P_i B_i(t_k) = Q_k \quad k = 0, \dots, n \quad (\text{C.46})$$

Pero, se sabe que para un soporte no periódico, la curva interpola los puntos de control extremos:

$$\begin{aligned} P_0 &= Q_0 \\ P_{n+2} &= Q_n \end{aligned}$$

Al imponer dos condiciones más, por ejemplo: P_1 y P_{n+1} , llegamos al sistema de $n - 1$ ecuaciones con $n - 1$ incógnitas, que muestra que para cada ecuación todos los sumandos son nulos salvo tres:

$$\begin{bmatrix} B_2(t_1) & B_3(t_1) & 0 & \dots & & & & & & 0 \\ B_2(t_2) & B_3(t_2) & B_4(t_2) & & & & & & & \\ \vdots & B_3(t_3) & B_4(t_4) & B_5(t_5) & & & & & & \\ & & & & \ddots & \ddots & \ddots & & & \\ & & & & & & & & & \vdots \\ & & & & & & & & & B_{n-2}(t_{n-2}) & B_{n-1}(t_{n-2}) & B_n(t_{n-2}) \\ 0 & & & & & \dots & 0 & B_{n-1}(t_{n-1}) & B_n(t_{n-1}) & & & \end{bmatrix} \begin{bmatrix} P_2 \\ P_3 \\ \vdots \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} = \begin{bmatrix} Q_1 - P_1 B_1(t_1) \\ Q_2 \\ \vdots \\ \vdots \\ Q_{n-2} \\ Q_{n-1} - P_{n+1} B_{n+1}(t_{n-1}) \end{bmatrix}$$

En la práctica es más frecuente dar las derivadas del spline en los extremos:

$$P_1 = Q_0 + \frac{t_1}{3}D_0$$

$$P_{n+1} = Q_n - \frac{1 - t_{n-1}}{3}D_n$$

Este método de interpolación es un procedimiento muy eficiente cuyo coste computacional es proporcional a n .

C.4.4. Interpolación local

Los anteriores métodos de interpolación pueden ser considerados como globales ya que la modificación de uno solo de los parámetros (puntos de interpolación o derivada en el extremo) afecta a toda la forma de la curva. Y esto se debe a la formulación intrínseca del problema de interpolación a través de la solución de un sistema de ecuaciones. Modificar un único punto de control se traduce en modificar todos los puntos de control, y a su vez, todos los B-Spline respecto a los cuales se expresa la curva.

Sin embargo, se vio anteriormente cómo una curva de Bézier compuesta se comportaba localmente respecto a la modificación de uno de los puntos de interpolación (solo los dos tramos que se encuentran en el punto sufren modificación). El poder expresar una curva de Bézier como una curva B-Spline es lo que da lugar a un método de interpolación local basado en B-Spline. Para ello se acude al interpolante cúbico de Hermite.

El proceso es como sigue:

1. Dados los puntos de interpolación Q_0, \dots, Q_n , determinar un soporte de interpolación basado en longitud de cuerda que una vez calculado se mantiene invariable aun cuando algunos de los Q_i se modifiquen.
2. Realizar una estimación de las derivadas en los puntos Q_i por el método de Bessel visto en el capítulo 2.
3. Obtener el interpolante cúbico de Hermite mediante las expresiones:

$$P_{3i} = Q_i \qquad i = 0, \dots, n$$

$$P_{3i-1} = Q_i - \frac{t_i - t_{i-1}}{3}D_i \qquad i = 1, \dots, n$$

$$P_{3i+1} = Q_i + \frac{t_{i+1} - t_i}{3}D_i \qquad i = 1, \dots, n - 1$$

La modificación de un punto de interpolación o una derivada afecta exclusivamente a los tramos que se encuentren en el punto en cuestión.

4. Construir el soporte no periódico:

$$S = \{0, 0, 0, 0, t_1, t_1, t_1, \dots, t_{n-1}, t_{n-1}, t_{n-1}, 1, 1, 1, 1\}$$

que tiene $m + 1 = 3n + 5$ nodos, con lo cual $m = 3n + 4$, y la dimensión del espacio es $3n + 1$ que corresponde con los $3n + 1$ puntos de control obtenidos en el numeral 3.

5. Reducir la multiplicidad de cada nodo interior para que tenga multiplicidad 2. Esta reducción es posible ya que la curva es realmente de clase C^1 . Se puede comprobar que en la reducción de cada t_j queda eliminado el punto de control $P_{3j} = Q_j$.

Una vez realizada la reducción se habrán eliminado de la lista de puntos de control iniciales los puntos de interpolación Q_i internos. Los restantes $2n + 2$ puntos de control P_0, \dots, P_{3n} definen la misma curva B-Spline sobre el soporte:

$$S = \{0, 0, 0, 0, t_1, t_1, t_2, t_2, \dots, t_{n-1}, t_{n-1}, 1, 1, 1, 1\}$$

Ejercicio 4.01: Realizar una implementación en MATLAB de este método de interpolación local mediante splines cúbicos de clase C^1 . Como datos de entrada ingrese los puntos de interpolación y el soporte de interpolación y como datos de salida obtenga los puntos de control y la gráfica de la curva.

Apéndice D

Curvas Racionales

Anteriormente se han señalado diferentes técnicas para la realización del modelado computacional, todas ellas partiendo de curvas polinomiales, o trozos de ellas, que expresadas en el espacio por sus respectivas bases, constituyen las curvas de Bézier o los B-Spline, mostrando un correcto desempeño, permitiendo desarrollar curvas suaves, sin embargo comparten una gran carencia, y es la capacidad de modelar figuras tan sencillas, primitivas y muy utilizadas como la circunferencia, ya que los polinomios tienen propiedades muy definidas, como tener a lo sumo tantas raíces como el grado, lo que coarta de sobremanera su flexibilidad.

Todos estos factores anteriormente señalados, apuntan a la conveniencia de extender la familia de curvas y dotarla con nuevos grados de libertad. para ello el paso a seguir es emplear curvas racionales, ya que los coeficientes de los polinomios de los denominadores proporcionan parámetros nuevos con los cuales se puede modificar la forma de las curvas. Es claro que una circunferencia, o un simple arco no puede representarse de forma paramétrica como un polinomio, por otra parte existe una sencilla ecuación paramétrica que expresa la circunferencia con centro en (0,0) y radio 1, como el cociente de dos polinomios cuadráticos.

$$x = \frac{1 - u^2}{1 + u^2}$$

$$-\infty < u < \infty$$

$$y = \frac{2u}{1 + u^2}$$

Así mismo la representación de todas las secciones cónicas se puede realizar mediante el cociente de dos polinomios de segundo grado, lo que llevó a la consideración de curvas racionales, para lo cual se definen las parametrizaciones racionales de grado n como aquellas cuyas componentes son cocientes de polinomios de grado n , y como casi particular las curvas polinomiales cuando el denominador se reduce a una constante.

Una extensión de las formulaciones par las curvas de Bézier y los splines, lleva a la posterior definición de curvas compuestas racionales , dando lugar a un sistema de modelado cerrado, que goza de todas las propiedades de los splines polinomiales e incluyendo a hora nuevos tipos de curvas, dentro de las que se destacan las secciones cónicas que en primera instancia evidenciaron la carencia de su definición por medio de las técnicas de modelado descritas por este libro anteriormente.

D.1. Secciones cónicas

Es evidente la ventaja de las representaciones racionales frente a las parametrizaciones polinómicas, que sólo permiten representar arcos de parábola, mientras que las parametrizaciones racionales describen arcos de todo tipo de cónicas. Así pues, a partir de una parametrización polinómica en el espacio, se ha obtenido una parametrización racional en el plano. Existen valores de u para los cuales el denominador $W(u)$ se anula, aunque son puntos legítimos del plano proyectivo, corresponden a puntos del infinito, es decir fuera del plano afín. Por ejemplo la parametrización normalizada, curva racional de grado dos, expresando los polinomios cuadráticos en la base Bernstein, para enlazar rápidamente el estudio de las cónicas con las teoría de curvas de Bézier, da como resultado una curva,

$$C(u) = \begin{cases} x = \frac{P(u)}{W(u)} = \frac{(1-u)^2 a_0 + 2u(1-u)a_1 + u^2 a_2}{(1-t)^2 + 2w_1 u(1-t) + w_2 u^2} \\ y = \frac{Q(u)}{W(u)} = \frac{(1-u)^2 b_0 + 2wu(1-u)b_1 + u^2 b_2}{(1-t)^2 + 2w_1 u(1-t) + u^2} \end{cases} \quad (D.1)$$

Para saber de que tipo de cónica se trata, basta conocer cuántos puntos tiene en el infinito. Una elipse no tiene ninguno, una parábola tiene uno y una hipérbola, dos.

El objetivo de estudio se centra por tanto, en los ceros del denominador,

$$2(1-w)u^2 - 2(1-w)u + 1 = 0$$

$$t = \frac{1-w \pm \sqrt{(w-1)(w+1)}}{2(1-w)}$$

Considerando sólo pesos positivos, para $0 < w < 1$, el discriminante es negativo, no hay puntos en el infinito, y se tiene un arco de elipse.

- Para $w = 1$ la raíz es única y tenemos un arco de parábola, puesto que todos los pesos tienen el mismo valor.
- Para $w > 1$, los polos son dos y se trata de un arco de hipérbola.

Este resultado proporciona una interpretación alternativa del peso de una cónica. Permitiendo clasificar de forma sencilla por medio de esta ecuación la clase de cónica que representa.

Como esta curva se encuentra expresada en la base de los polinomios de Bernstein, comparte las propiedades de dicha base, por ejemplo la interpolación de los puntos extremos de la curva donde $P_0 = (a_0, b_0)$ y $P_2 = (a_2, b_2)$ respectivamente para $u = 0$ y $u = 1$, propiedad que es análoga a la de los polinomios de Bernstein.

Ahora, si se analiza el caso de las derivadas en $u = 0$ y $u = 1$.

Si $x = \frac{P(u)}{W(u)}$ e $y = \frac{Q(u)}{W(u)}$, entonces

$$x' = \frac{W(u)P'(u) - W'(u)P(u)}{W^2(u)}$$

De donde

$$x' = \frac{W(0)P'(0) - W'(0)P(0)}{W^2(0)} = P'(0) - W'(0)a_0$$

Pero

$$P'(u) = -2a_0(1 - u) + 2a_1(1 - 2u) + 2a_2u \implies P'(0) = -2a_0 + 2a_1$$

$$W'(u) = -2(1 - u) + 2w(1 - 2u) + 2u \implies W'(0) = -2 + 2w$$

$$\therefore x'(0) = 2(a_1 - a_0) - 2a_0(w - 1) = 2(a_1 - wa_0) \quad (\text{D.2})$$

De manera análoga se tiene:

$$y'(0)wa_2 = 2(b_1 - wb_0) \quad (\text{D.3})$$

Para $u = 1$ se tiene:

$$x'(1) = P'(1) - W'(1)P(1) = 2(a_2 - a_1) - 2a_2(1 - w) = 2(wa_2 - a_1) \quad (\text{D.4})$$

De la misma forma:

$$y'(1) = 2(wb_2 - b_1) \quad (\text{D.5})$$

Agrupando las anteriores expresiones:

$$C'(0) = 2(Q_1 - wP_0)$$

$$C'(1) = 2(wP_2 - Q_1)$$

donde

$$P_0 = (a_0, b_0), \quad Q_1 = (a_1, b_1) \quad y \quad P_2 = (a_2, b_2)$$

La razón de representar el punto $Q_1 = (a_1, b_1)$ con una letra distinta a P , es llamar a $P_1 = \frac{1}{w}Q_1 = \frac{1}{w}(a_1, b_1)$

Esta nueva notación da lugar a:

$$\begin{aligned} C'(0) &= 2w(P_1 - P_0) \\ C'(1) &= 2w(P_2 - P_1) \end{aligned} \quad (\text{D.6})$$

Que son ecuaciones análogas a las ecuaciones para las derivadas de las curvas de Bézier polinomial de segundo grado, ahora una curva racional cuadrática expresada a través de nuevos coeficientes:

$$\begin{aligned} P_0 &= (x_0, y_0) = (a_0, b_0) \\ P_1 &= (x_1, y_1) = \frac{1}{w}(a_1, b_1) \\ P_2 &= (x_2, y_2) = (a_2, b_2) \end{aligned}$$

la curva queda representada entonces así:

$$C(u) = \begin{cases} x = \frac{x_0(1-u)^2 + 2wx_1u(1-u) + x_2u^2}{(1-u)^2 + 2wu(1-u) + u^2} \\ y = \frac{y_0(1-u)^2 + 2wy_1u(1-u) + y_2u^2}{(1-u)^2 + 2wu(1-u) + u^2} \end{cases} \quad (\text{D.7})$$

en forma vectorial

$$C(u) = \frac{P_0(1-u)^2 + 2wP_1u(1-u) + P_2u^2}{(1-u)^2 + 2wu(1-u) + u^2} \quad (\text{D.8})$$

donde por analogía a las curvas de Bézier P_0 , P_1 , P_2 se denominan *puntos de control*

Algunas propiedades importantes

1. Para $u = 0$ y $u = 1$ la curva interpola los puntos de control extremos P_0 y P_1 respectivamente.
2. La tangente de la curva para $u = 0$ tiene dirección del segmento que une P_0 con P_1 y la tangente para $u = 1$, la dirección que une P_1 con P_2

$$C'(0) = 2w(P_1 - P_0)$$

$$C'(1) = 2w(P_2 - P_1)$$

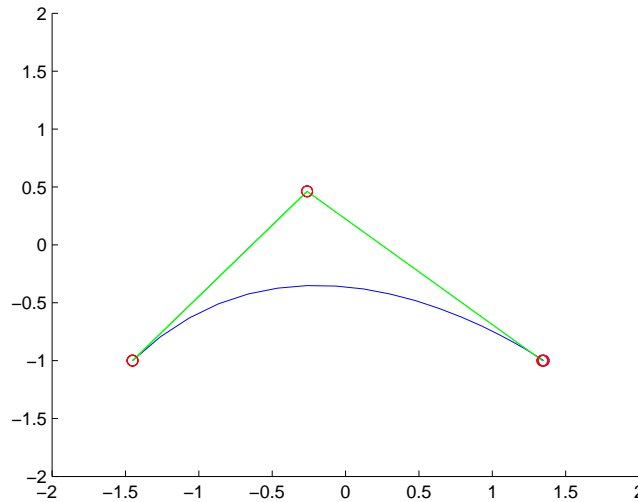


Figura D.1: puntos de control de una curva racional de Bézier

La proyección sobre el plano $Z = 1$ de la curva polinomial corresponde exactamente a la descrita por la ecuación D.8

$$\begin{aligned}
x &= x_0(1-u)^2 + 2wx_1u(1-u) + x_2u^2 \\
y &= y_0(1-u)^2 + 2wy_1u(1-u) + y_2u^2 \\
z &= (1-u)^2 + 2wu(1-u) + u^2
\end{aligned} \tag{D.9}$$

Esta es la curva de Bézier cuadrática con puntos de control $\bar{P}_0 = (x_0, y_0, 1)$, $\bar{P}_1(x_1, y_1, 1)$, , $\bar{P}_2(x_2, y_2, 1)$ y es importante constatar que lo que se ha denominado *puntos de control* de la curva racional $C(u)$ son las proyecciones de los puntos de control de la curva polinomial $\bar{C}(u)$.

Esta propiedad aplicada a las curvas de grado arbitrario se conoce como *invarianza por proyecciones*, que consiste en realizar una proyección de una determinada curva que en general es racional, para ello no se necesita de calculos y procedimientos complejos, basta con proyectar sus puntos de control y definir a partir de ellos una nueva curva, que será la proyección de la original.

Esta propiedad es de gran importancia en la fase de rendering, sobre todo para superficies, pues la proyección perspectiva para visualizar en 2D y 3D, se lleva a cabo solo transformando en las ecuaciones de la perspectiva (que siempre envuelven una división), solo un conjunto limitado de puntos, facilitando el proceso de cálculo.

D.2. Curvas racionales como proyecciones

Suponiendo la existencia de una curva polinómica que no pasa por el origen, si se deseara proyectar esta curva sobre un plano que existe en el espacio entonces esta representación seria dada por funciones racionales, Así pues, a partir de una parametrización polinómica en el espacio, se obtiene una parametrización racional en el plano, en general una curva racional 2D es aquella con ecuación paramétrica:

$$C(u) = \begin{cases} \bar{x} = \frac{P(u)}{R(u)} \\ \bar{y} = \frac{Q(u)}{S(u)} \end{cases} \quad u \in [u_0, u_1] \tag{D.10}$$

Se puede suponer que lo polinomios en el denominador son los mismos, ya que no existe restricción alguna que no lo permitan, lo que facilita ampliamente la comprensión de las propiedades que gozan este tipo de curvas, por lo tanto la nueva estructura de estas curvas racionales será

$$C(u) = \begin{cases} \bar{x} = \frac{P(u)}{W(u)} \\ \bar{y} = \frac{Q(u)}{W(u)} \end{cases} \quad u \in [u_0, u_1]$$

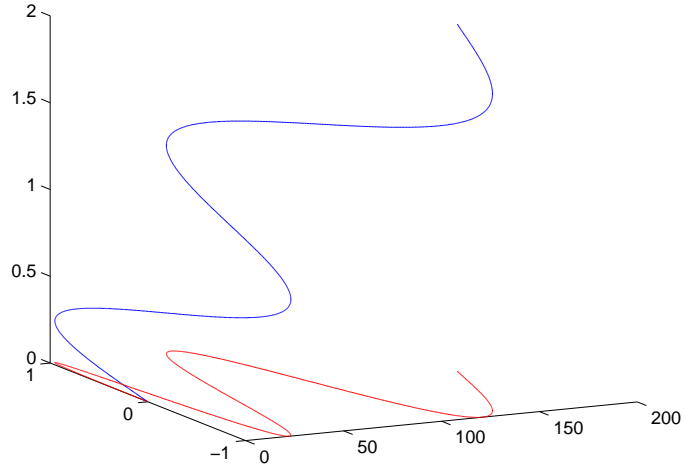


Figura D.2: Curva racional como proyección

Para este caso se parte de la suposición de que $P(u)$, $Q(u)$ y $W(u)$ son polinomios de grado menor o igual que n , tomando a n como *fijo*.

Volviendo al primer tema de discusión en donde se supone una curva 3D, cuya proyección sobre un plano que llamado $z = 1$ es una curva racional, esta representación racional permite utilizar todas las técnicas y métodos de modelado para curvas polinomiales operando en una dimensión mas, realizando una racionalización de dicha curva sobre el plano ya mencionado.

D.3. Curvas de Bézier racionales

De la misma forma que una curva de Bézier polinomial simple no es sino un polinomio expresado en la Base Bernstein, la curva de Bézier racional es el cociente de dos polinomios afectados por el efecto de unos pesos expresado en esta base:

$$C(u) = \begin{cases} x = \frac{\sum_{k=0}^n a_k B_k(u)}{\sum_{k=0}^n w_k B_k(u)} \\ y = \frac{\sum_{k=0}^n b_k B_k(u)}{\sum_{k=0}^n w_k B_k(u)} \end{cases} \quad (D.11)$$

donde B_k son los polinomios de Bernstein de grado n .

En adelante se escribirá la ecuación de una curva de Bézier racional como:

$$C(u) = \begin{cases} x = \frac{\sum_{k=0}^n w_k x_k B_k(u)}{\sum_{k=0}^n w_k B_k(u)} \\ y = \frac{\sum_{k=0}^n w_k y_k B_k(u)}{\sum_{k=0}^n w_k B_k(u)} \end{cases} \quad (\text{D.12})$$

y en forma vectorial:

$$C(u) = \frac{\sum_{k=0}^n w_k P_k B_k(u)}{\sum_{k=0}^n w_k B_k(u)} \quad (\text{D.13})$$

Los puntos de control P_k en general tendrán 2 o tres componentes, según sea una curva 2D o 3D, y se denominarán puntos de control de la curva, desempeñando un papel similar a las de las curvas polinomiales.

Por su parte los coeficientes w_k , se conocen como pesos, dado que expresan su influencia sobre cada P_k en el correspondiente polinomio de Bernstein, estos pesos se toman positivos para que tengan significado geométrico.

Se puede reescribir la curva de una manera mas compacta para lo cual:

$$C(u) = \sum_{k=0}^n P_k \frac{w_k B_k(u)}{\sum_{i=0}^n w_i B_i(u)} \quad (\text{D.14})$$

llamando

$$R_k(u) = \frac{w_k B_k(u)}{\sum_{i=0}^n w_i B_i(u)} \quad (\text{D.15})$$

se tiene

$$C(u) = \sum_{k=0}^n P_k R_k(u) \quad (\text{D.16})$$

Mostrando la curva como una combinación lineal de las funciones de base R_k y tomando los coeficientes como puntos de control, y sabiendo que sus propiedades son análogas a las de los polinomios de Bernstein, se puede destacar:

$$R(u) = \sum_{k=0}^n R_k(u) = 1 \quad (\text{D.17})$$

Asumiendo que todos los pesos son positivos se puede verificar:

$R_k(u) \geq 0$ para lo cual la curva racional es una combinación lineal convexa de los puntos de control, para ser mas claros se dice que la curva esta contenida dentro de una *envolvente convexa* de los puntos de control.

D.3.1. Propiedades de las curvas de Bézier Racionales.

1. Interpolación de los puntos de control extremos:

$$\begin{aligned} C(0) &= P_0 \\ C(t) &= P_n \end{aligned}$$

Solo se interpolan los puntos de control extremos

2. Tangencia al Polígono en los puntos de control en sus extremos

$$\begin{aligned} C'(0) &= n \frac{w_1}{w_0} (P_1 - P_0) \\ C'(t) &= n \frac{w_{n-1}}{w_n} (P_n - P_{n-1}) \end{aligned}$$

Solo el punto inicial y final son tangentes al polígono de control ya que la curva solo pasa por estos dos puntos.

3. Control seudolocal

Si dos curvas de Bézier Racionales $C(u)$ y $\bar{C}(u)$ se diferencian una de la otra en que uno de los puntos P_i es múltiplo de de $R_i(u)$, entonces:

$$C(u) - \bar{C}(u) = R_i(t)(P_i - \bar{P}_{ki})$$

Dado que $R_k(t)$ alcanza su valor máximo en $[0, 1]$, por lo tanto en valores del parámetro u lejanos de este máximo, la diferencia $C(u) - \bar{C}(u) = R_i(t)(P_i - \bar{P}_{ki})$ es pequeña, significando que el movimiento o modificación de un punto de control afecta la forma de la curva, haciendose más notorio en las cercanías del punto movido.

4. Restricción de la envolvente convexa.

Si todos los pesos son no negativos la curva , es siempre interior a la envolvente convexa de dichos puntos.

5. Disminución de la variación.

Una recta arbitraria no intercepta a una curva de Bézier racional en más puntos de los que intercepta a su polígono de control. por lo tanto el número de veces que la recta corta al polígono de control es cota superior del número de veces que corta a la curva.

6. Invarianza Afín

Para realizar una transformación afín ya sea una rotación, escalado, translación... a una curva de Bézier racional, basta con aplicarla sobre los puntos de control y con ellos transformar la curva.

7. Invarianza Projectiva

la proyectividad, también llamada homografía, es una propiedad lineal del espacio proyectivo en dos o tres dimensiones, de modo que si f es una transformación proyectiva del plano y $\{P_0, \dots, P_n\}, \{w_0, \dots, w_n\}$ son el polígono de control y se conoce la lista de pesos de una parametrización racional, entonces:

$$C(u) = \frac{\sum_{k=0}^n w_k P_k B_k(u)}{\sum_{k=0}^n w_k B_k(u)} \quad f(C(u)) = C(u) = \frac{\sum_{k=0}^n w'_k P'_k B_k(u)}{\sum_{k=0}^n w'_k B_k(u)} \quad (\text{D.18})$$

Donde los pesos y vértices de la nueva curva están relacionados con los primitivos, $P_i = (w_i, w_i P_i)$, mediante la relación $P'_i = f(P_i) = (w'_i, w'_i P_i)$, como corresponde al hecho de que \mathbf{f} es la transformación lineal asociada a la curva pero que se aplica únicamente a los puntos de control que precisamente por ser lineal cumple que,

$$C(u) = \sum_{k=0}^n P_k B_k(u), \quad f(C(u)) = \sum_{k=0}^n f(P_k) B_k,$$

de donde se infiere que el polígono de control de la curva en R^3 , $\mathbf{f}(C(u))$, es $f(P_0), \dots, f(cP)$.

De ahí se llega a D.18 por proyección sobre el plano afín , identificando las componentes de cada P'_i

Se puede concluir que la curva transformada de una curva de Bézier racional, no es más que una curva cuyos puntos de control son el producto de una transformación de los puntos de control de la curva original expresados en una nueva base.

Estas transformaciones son muy importantes para el diseño, ya que, además de las afines, incluyen los cambios de perspectiva.

8. Secciones Cónicas

Todas las secciones cónicas son curvas de Bézier racionales de segundo grado y en general todas pueden considerarse como la proyección sobre el plano $Z = 1$ de una parábola 3D. Por otra parte la propiedad anterior manifiesta que la transformada cónica por la Proyectividad es de nuevo una cónica, aunque no necesariamente la misma debido a la aparición de nuevos pesos.

9. Curvas de Bézier Polinomiales

Si todos los pesos son iguales y el denominador es una constante, la curva de Bézier racional es una curva polinomial simple, esto se debe a la propiedad de los polinomios de Bernstein de sumar la unidad, siendo las curvas de Bézier polinomiales casos particulares de las curvas de Bézier racionales.

10. Elevación del grado

Para elevar el grado de una curva polinomial, bastaba multiplicarla por $1 = (1-u) + u$ para que formalmente el grado aumente una unidad, sin alterar la gráfica ni la parametrización. En el caso de las curvas racionales, no se precisan tantas restricciones, se multiplica el numerador y denominador por una misma función $f(t)$, la parametrización permanece invariable por cancelación, aunque las expresiones de ambos se alteran notablemente. Dado que el propósito es elevar el grado en una unidad, se supone una curva racional 2D.

$$C(u) = \frac{\sum_{k=0}^n w_k P_k B_k(u)}{\sum_{k=0}^n w_k B_k(u)} \quad P_k = (x_k, y_k)$$

siendo considerada como una proyección de la curva de Bézier Polinomial 3D.

$$\bar{C}(u) = \sum_{k=0}^n \bar{P}_k B_k(u) \quad \bar{P}_k = w_k(x_k, y_k, 1)$$

Utilizando las ecuaciones del capítulo 2, usadas para elevación del grado, los nuevos

puntos de control Q_k que representan la misma curva $\bar{C}(u)$ pero respecto a los polinomios de Bernstein $gradon + 1$

$$\bar{C}(u) = \sum_{k=0}^{n+1} Q_k B_k(u) \quad Q_k = (r_k, s_k, 1)$$

Proyectando esta curva sobre el plano $Z=1$ se obtiene la misma curva, pero de grado $n + 1$.

$$C(u) = \frac{\sum_{k=0}^{n+1} \sigma_k Q_k B_k(u)}{\sum_{k=0}^{n+1} \sigma_k B_k(u)} \quad Q_k = (r_k, s_k)$$

D.4. Curvas de Bézier racionales compuestas

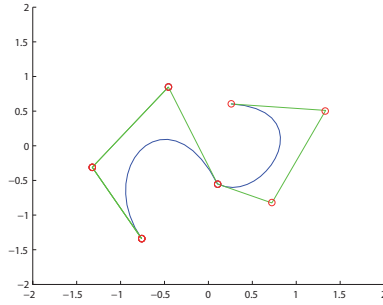
La conveniencia de componer sucesivamente varias curvas de Bézier para disponer de suficientes puntos de control sin necesidad de operar con grados demiasido altos se aplica de manera análoga al caso polinomial, ya que aprovecha la propiedad de que una curva de Bézier racional simple sea tangente en los puntos extremos.

Considerando un parámetro global u común a dos tramos consecutivos, donde para $u \in [u_0, u_1]$ se pretende generar el primer tramo y para $u \in [u_1, u_2]$, el segundo.

Las curvas vendrán definidas como:

$$C_1(u) = \frac{\sum_{k=0}^n w_k P_k B_k \left(\frac{u-u_0}{u_1-u_0} \right)}{\sum_{k=0}^n w_k B_k \left(\frac{u-u_0}{u_1-u_0} \right)} \quad u \in [u_0, u_1]$$

$$C_2(u) = \frac{\sum_{k=0}^n w_k Q_k B_k \left(\frac{u-u_1}{u_2-u_1} \right)}{\sum_{k=0}^n w_k B_k \left(\frac{u-u_1}{u_2-u_1} \right)} \quad u \in [u_1, u_2]$$



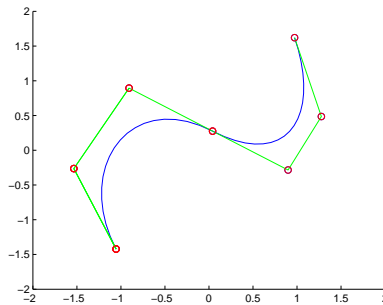
Para garantizar la continuidad el primer punto de control Q_0 de $C_2(u)$ debe coincidir con el último punto de control P_n de $C_1(u)$, teniendo en cuenta que:

$$C_1(u_1) = P_n$$

$$C_2(u_2) = Q_0$$

Ahora, para que la curva sea de clase C^1 , las derivadas de $C_1(u)$ y $C_2(u)$ deben coincidir en $u = u_1$. Se introduce el parámetro local $t = \frac{u-u_0}{u_1-u_0}$ para obtener que la condición para continuidad de clase C^1 en $u = u_1$:

$$\frac{w_{n-1}}{u_1 - u_0}(P_n - P_{n-1}) = \frac{\sigma_1}{u_2 - u_1}(Q_1 - Q_0)$$



donde σ_1 es el peso asociado a Q_1 , y puesto que $Q_0 = P_n$ es natural considerar que un mismo punto de control tiene pesos iguales. De la ecuación se implica que los puntos de control P_{n-1} , $P_n = Q_0$ y Q_1 están alineados.

Para generalizar se consideran m subintervalos para el parámetro global u , constituido por los puntos $u_0 < u_1 < \dots < u_m$. Para $u \in [u_{i-1}, u_i]$ el tramo i -ésimo de la curva está definido por puntos de control $P_0^{(i)}, P_1^{(i)}, \dots, P_n^{(i)}$ y pesos $w_0^{(i)}, \dots, w_n^{(i)}$, y:

$$C_i(u) = \frac{\sum_{k=0}^n w_k^{(i)} P_k^{(i)} B_k \left(\frac{u-u_{i-1}}{u_1-u_{i-1}} \right)}{\sum_{k=0}^n w_k^{(i)} B_k \left(\frac{u-u_{i-1}}{u_1-u_{i-1}} \right)} \quad i = 1, \dots, m$$

donde, para que la curva sea continua se debe verificar:

$$P_n^{(i)} = P_0^{(i+1)} \quad i = 1, \dots, m$$

y para que sea de clase C^1 :

$$\frac{w_{n-1}^{(i)}}{u_i - u_{i-1}} (P_n^{(i)} - P_{n-1}^{(i)}) = \frac{w_1^{(i+1)}}{u_i - u_{i-1}} (P_1^{(i+1)} - P_0^{(i+1)}) \quad i = 1, \dots, m - 1$$

En la práctica, los puntos de control y los pesos suelen ser datos. Los puntos de ruptura deben verificar:

$$\frac{u - u_{i-1}}{u_1 - u_{i-1}} = \frac{w_1^{(i+1)} \left\| P_1^{(i+1)} - P_0^{(i+1)} \right\|}{w_{n-1}^{(i)} \left\| P_n^{(i)} - P_{n-1}^{(i)} \right\|} \quad i = 1, \dots, m - 1$$

D.4.1. Secciones cónicas como curvas de Bézier racionales compuestas

Las propiedades de la curva en relación a los polinomios de Bernstein se refieren al intervalo $u \in [0, 1]$ del espacio paramétrico y, por otra parte, para $u \in [0, 1]$ sólo se dispone de un arco de la curva completa. Esto es un problema, ya que para una cónica (una elipse o una circunferencia) se requiere que $u \in [-\infty, \infty]$ para ser generada en su totalidad por una función racional cuadrática.

Gracias a las curvas de Bézier racionales compuestas se puede solucionar este problema. Por ejemplo, para una circunferencia se realizará una generalización a fin de obtener las expresiones matemáticas que permitan su generación.

En primer lugar, una curva racional cuadrática está definida en función de tres puntos de control, lo que hace que existan distintas formas de subdividir una circunferencia en arcos (cada uno controlado por tres puntos). La figura muestra la circunferencia

centrada en el origen y de radio 1 como composición de 3 arcos. El primer arco tiene por puntos de control:

$$P_0 = \left(\frac{\sqrt{3}}{2}, \frac{1}{2} \right) \quad P_1 = (0, 2) \quad P_2 = \left(-\frac{\sqrt{3}}{2}, \frac{1}{2} \right)$$

los demás arcos se obtienen rotando estos puntos 120° y 240° respectivamente para formar un triángulo equilátero. Ahora, la curva racional cuadrática correspondiente a los puntos de control anteriores, es:

$$C_1(u) = \frac{\left(\frac{\sqrt{3}}{2}, \frac{1}{2} \right) (1-u)^2 + 2w(0, 2)u(1-u) + \left(-\frac{\sqrt{3}}{2}, \frac{1}{2} \right) u^2}{(1-u)^2 + 2wu(1-u) + u^2} \quad u \in [0, 1]$$

y $w^2 < 1$ para que se trate de un arco de elipse.

Separando las coordenadas y agrupando se llega a:

$$\begin{cases} x = \frac{-\frac{\sqrt{3}}{2}(2u-1)}{2(1-w)u^2 - 2(1-w)u + 1} \\ y = \frac{(1-4w)u^2 - (1-4w)u + \frac{1}{2}}{2(1-w)u^2 - 2(1-w)u + 1} \end{cases} \quad u \in [0, 1]$$

Ahora se debe verificar cuándo estas dos expresiones conforman un arco de circunferencia:

$$x^2 + y^2 = \frac{(1-4w)^2 u^4 - 2(1-4w)^2 u^3 + 16(16w^2 - 12w + 5)u^2 + 4(w-1)u + 1}{[2(1-w)u^2 - 2(1-w)u + 1]^2}$$

Al realizar la división, se obtiene como residuo:

$$\lambda = \left(\frac{3}{4} \right) \frac{4w^2 - 1}{(1-w)^2}$$

Entonces, para que no haya residuo en la división $w = \frac{1}{2}$ y $x^2 + y^2 = 1$.

Efectivamente, la curva queda definida como:

$$C_1(u) = \frac{\left(\frac{\sqrt{3}}{2}, \frac{1}{2} \right) (1-u)^2 + (0, 2)u(1-u) + \left(-\frac{\sqrt{3}}{2}, \frac{1}{2} \right) u^2}{(1-u)^2 + u(1-u) + u^2} \quad u \in [0, 1]$$

A partir de este arco se construyen los otros dos:

$$C_2(u) = \frac{\left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right) (2-u)^2 + (-\sqrt{3}, -1) (u-1)(2-u) + (0, -1) (u-1)^2}{(2-u)^2 + (u-1)(2-u) + (u-1)^2} \quad u \in [1, 2]$$

$$C_3(u) = \frac{(0, -1) (3-u)^2 + (\sqrt{3}, -1) (u-2)(3-u) + \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right) (u-2)^2}{(3-u)^2 + (u-2)(3-u) + (u-2)^2} \quad u \in [2, 3]$$

La alternativa más utilizada en la práctica es considerar la circunferencia como composición de cuatro arcos de 90° cada uno. Los puntos de control del primer arco son:

$$P_0 = (1, 0) \quad P_1 = (1, 1) \quad P_2 = (0, 1)$$

y la expresión de la curva de Bézier racional cuadrática es:

$$C_1(u) = \frac{(1, 0) (1-u)^2 + \sqrt{2} (1, 1) u (1-u) + (0, 1) u^2}{(1-u)^2 + \sqrt{2} u (1-u) + u^2} \quad u \in [0, 1]$$

teniendo en cuenta que:

$$\lambda = \frac{2w^2 - 1}{4(1-w)^2}$$

entonces $w = \frac{\sqrt{2}}{2}$. Los otros tres arcos se obtienen con expresiones análogas utilizando sus correspondientes puntos de control.

D.5. NURBS

NURBS son las siglas en inglés de “Non-Uniform Rational B-Spline” que hacen referencia a un spline racional, definido como la proyección sobre el hiperplano $Z = 1$ de un spline polinomial en un espacio de una dimensión más, y se puede definir como:

$$C_i(u) = \frac{\sum_{k=0}^n w_k P_k B_k(u)}{\sum_{k=0}^n w_k B_k(u)} \quad (\text{D.19})$$

donde los $B_k(u)$ son los B-Spline básicos de grado p , que se obtienen por recurrencia y P_k son los puntos de control. También se recuerda que la relación entre el número de puntos de control y el grado viene dado por:

$$n = m - p - 1$$

siendo $m + 1$ el número de nodos en el soporte (no periódico):

$$S = \{\underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{1, \dots, 1}_{p+1}\}$$

La ecuación D.19 se puede escribir:

$$C(u) = \sum_{k=0}^n P_k B_k(u)$$

donde $R_k(u)$ son los B-Spline racionales básicos

$$R_k(u) = \frac{w_k B_k(u)}{\sum_{i=0}^n w_i B_i(u)}$$

que cumplen la propiedad $\sum_{k=0}^n R_k(u) = 1$. Ahora, si todos los pesos w_k son positivos entonces $R_k(u) \geq 0$.

Propiedades de los NURBS se derivan de los splines polinomiales:

1. Interpolación en los puntos extremos.

$$C(0) = P_0$$

$$C(1) = P_n$$

2. Tangencia al polígono de control en sus segmentos extremos.

$$C'(0) = \frac{p}{u_{p+1}} \left(\frac{w_1}{w_0} \right) (P_1 - P_0)$$

$$C'(1) = \frac{p}{1 - u_{m-p-1}} \left(\frac{w_{n-1}}{w_n} \right) (P_n - P_{n-1})$$

3. Control local.

Si un punto de control P_j es modificado a \bar{P}_j , la nueva curva $\bar{C}(u)$ se diferencia de la original en:

$$C(u) - \bar{C}(u) = (P_j - \bar{P}_j) R_j(u) = (P_j - \bar{P}_j) \frac{w_j B_j(u)}{\sum_{i=0}^n w_i B_i(u)}$$

Lo que muestra que ambas curvas son idénticas excepto en el intervalo $[u_j, u_{j+p+1}]$. Lo mismo es cierto si lo que se modifica es un peso w_j .

4. Restricción a la envolvente convexa.

Si $u \in [u_j, u_{j+1}]$, entonces $C(u)$ está en el interior de la envolvente convexa de los puntos de control P_{j-p}, \dots, P_j .

5. Disminución de la variación.

Una recta o plano o interseca a una curva NURBS en más puntos de los que interseca al polígono de control.

6. Invarianza afin.

Para efectuar una transformación afín a una curva NURBS, basta con transformar los puntos de control y construir a partir de éstos la nueva curva.

7. Invarianza Proyectiva.

Si:

$$C(u) = \frac{\sum_{k=0}^n w_k P_k B_k(u)}{\sum_{k=0}^n w_k B_k(u)}$$

y \bar{P} es el transformado de un punto P por una Proyectividad, entonces:

$$\bar{C}(u) = \frac{\sum_{k=0}^n \sigma_k \bar{P}_k B_k(u)}{\sum_{k=0}^n \sigma_k B_k(u)}$$

donde $\sigma_k = w_k \bar{w}_k$ y \bar{w}_k es el denominador común de la expresión de \bar{P}_k .

8. **Generalización de curvas de Bézier racionales.**

Si dos nodos $u_i < u_{i+1}$ tienen ambos multiplicidad $p + 1$, entonces $C(u)$ es en el intervalo $[u_i, u_{i+1}]$ una curva de Bézier racional simple. Si todos los nodos distintos son $p + 1$ múltiples, $C(u)$ es una curva de Bézier compuesta.

9. **Generalización de B-Splines polinomiales.**

Si todos los pesos w_k son iguales, la curva NURBS es polinomial, con lo cual la clase NURBS contiene a las curvas B-Splines polinomiales.

10. En un nodo de multiplicidad r_k , una curva NURBS es de clase C^{p-r_k} .

D.5.1. Algoritmos básicos para curvas NURBS

El algoritmo de inserción de nodos de Boehm tiene una sencilla generalización a curvas NURBS. Considerando la curva racional

$$C(u) = \frac{\sum_{k=0}^n w_k P_k B_k(u)}{\sum_{k=0}^n w_k B_k(u)}$$

que es la proyección sobre $W = 1$ de la curva polinomial

$$C(u) = \begin{cases} x = \sum_{k=0}^n w_k x_k B_k(u) \\ y = \sum_{k=0}^n w_k y_k B_k(u) \\ z = \sum_{k=0}^n w_k z_k B_k(u) \\ w = \sum_{k=0}^n w_k B_k(u) \end{cases}$$

que se puede escribir también como:

$$x = \sum_{k=0}^n (w_k x_k, w_k y_k, w_k z_k, w_k) B_k(u) = \sum_{k=0}^n \mathcal{P}_k B_k(u) \quad (\text{D.20})$$

donde $\mathcal{P} = (w_k x_k, w_k y_k, w_k z_k, w_k)$ son los puntos de control.

Utilizando el Algoritmo de Boehm, se inserta un nodo \bar{u} en el soporte obteniendo nuevos puntos de control $\bar{\mathcal{P}}_k$ de modo que la misma curva $C(u)$ se expresa en función de los $\bar{\mathcal{P}}_k$:

$$C(u) = \sum_{k=0}^{n+1} \bar{\mathcal{P}}_k \bar{B}_k(u)$$

$$\bar{\mathcal{P}}_k = (\alpha_k, \beta_k, \delta_k, \sigma_k)$$

Los $\bar{B}_k(u)$ son los B-Spline asociados al nuevo soporte (con un nodo mas)

Al proyectar sobre $W = 1$ la expresión anterior se obtiene la curva original $C(u)$ pero definida sobre el soporte nuevo:

$$C(u) = \frac{\sum_{k=0}^n \sigma_k \bar{\mathcal{P}}_k \bar{B}_k(u)}{\sum_{k=0}^n \sigma_k B_k(u)}$$

donde los nuevos puntos de control son:

$$Pk = \left(\frac{\alpha_k}{\sigma_k}, \frac{\beta_k}{\sigma_k}, \frac{\delta_k}{\sigma_k} \right)$$

Si todos los nodos interiores se reinsertan hasta tener multiplicidad p , la curva polinomial 4D D.20 es una curva de Bézier (ya que los B-Splines se reducen a polinomios de Bernstein).

D.5.2. Secciones cónicas como NURBS

Aprovechando el hecho de que una curva de Bézier compuesta es una curva B-Spline definida sobre un soporte no periódico con nodos interiores de multiplicidad p , una curva cónica podemos expresarla como un NURBS sobre el soporte no periódico con nodos interiores de multiplicidad 2:

$$S = \{0, 0, 0, v_1, v_1, v_2, v_2, \dots, v_k, v_k, 1, 1, 1\}$$

con $0 < v_i < v_{i+1} < 1$.

El numero total de nodos es: $m + 1 = 2k + 6$. El número de puntos de control es $n + 1 = 2k + 3$.

1. Circunferencia como composición de tres arcos.

Soporte:

$$S = \{0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1\}$$

Número de puntos de control:

$$n + 1 = 7$$

Puntos de control:

$$P_0 = \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right) \quad P_1 = (0, 2) \quad P_2 = \left(\frac{-\sqrt{3}}{2}, \frac{1}{2}\right) \quad P_3 = (-\sqrt{3}, -1) \\ P_4 = (0, -1) \quad P_5 = (\sqrt{3}, -1) \quad P_6 = P_0 = \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right)$$

Pesos:

$$w_0 = 1, w_1 = \frac{1}{2}, w_2 = 1, w_3 = \frac{1}{2}, w_4 = 1, w_5 = \frac{1}{2}, w_6 = 1$$

2. Circunferencia como composición de cuatro arcos.

Soporte:

$$S = \{0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1\}$$

Número de puntos de control:

$$n + 1 = 9$$

Puntos de control:

$$P_0 = (1, 0) \quad P_1 = (1, 1) \quad P_2 = (0, 1) \quad P_3 = (-1, 1) \quad P_4 = (-1, 0) \\ P_5 = (-1, -1) \quad P_6 = (0, -1) \quad P_7 = (1, -1) \quad P_8 = P_0 = (1, 0)$$

Pesos:

$$w_0 = 1, w_1 = \frac{\sqrt{2}}{2}, w_2 = 1, w_3 = \frac{\sqrt{2}}{2}, w_4 = 1, w_5 = \frac{\sqrt{2}}{2}, w_6 = 1, w_7 = \frac{\sqrt{2}}{2}, w_8 = 1$$

Algoritmo: Construcción de una circunferencia

Código

```

curvas=4;                                %número de arcos
axis([-2,2,-2,2])
hold on
center=ginput(1);                        %centro del circulo
plot(center(1),center(2),'ro')
radius=ginput(1)                          %define el radio
plot(radius(1),radius(2),'r')
radio=((center(1)-radius(1))^2+(center(2)-radius(2))^2)^0.5;
% puntos de ctrl para 4 tramos
P(1,1)=center(1)+radio;
P(1,2)=center(2);
P(2,1)=P(1,1);
P(2,2)=center(2)+radio;
P(3,1)=center(1);
P(3,2)=P(2,2);
P(4,1)=center(1)-radio;
P(4,2)=P(2,2);
P(5,1)=P(4,1);
P(5,2)=P(1,2);
P(6,1)=P(4,1);
P(6,2)=center(2)-radio;
P(7,1)=P(3,1);
P(7,2)=P(6,2);
P(8,1)=P(1,1);
P(8,2)=P(6,2);
P(9,1)=P(1,1);
P(9,2)=P(1,2);
plot(P(:,1),P(:,2),'o')
w=2^0.5/2;
t=250                                    %numero de divisiones
u=linspace(0,1,t);
for i=1:curvas
    for j=1:t                            % para X
        numx=P(2*i-1,1)*polyval([1 -2 1],u(j))
        +P(2*i,1)*w*polyval([-2 2 0],u(j))
    end
end

```

```

+P(2*i+1,1)*polyval([1 0 0],u(j));
numy=P(2*i-1,2)*polyval([1 -2 1],u(j))
+P(2*i,2)*w*polyval([-2 2 0],u(j))
+P(2*i+1,2)*polyval([1 0 0],u(j));
% y el denominador que es común a los 2
den=polyval([2 -2 1]+w*[-2 2 0],u(j));
Cx(i,j)=numx/den;
Cy(i,j)=numy/den;
end
end
for i=1:curvas
    plot(Cx(i,:),Cy(i:,:), 'r')
end
end

```

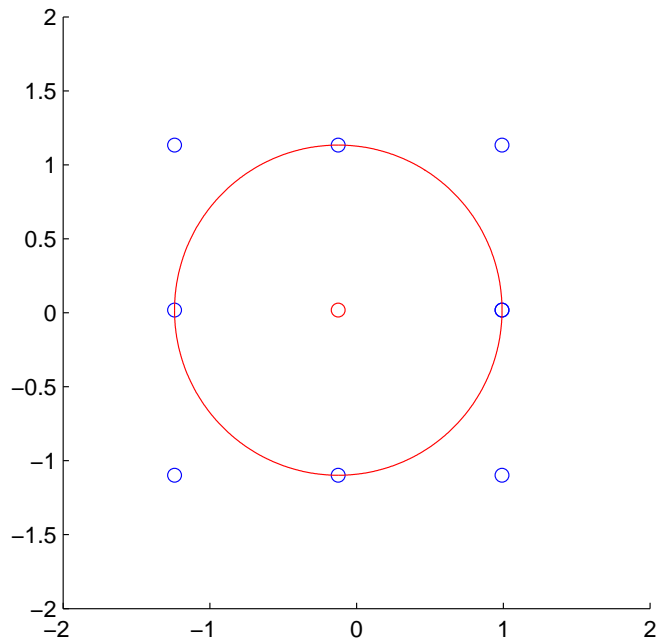


Figura D.3: Sección cónica como curva de Bézier racional

Casi todo lo que puede hacerse con una curva NURBS puede conseguirse con curvas B-Splines polinomiales y a menor coste. Es deseable que un sistema de modelado basado en NURBS opere en lo posible con curvas polinomiales y sólo convertir a formato NURBS cuando se deseen ajustes finos (modificando los pesos)

Apéndice E

Superficies 3D

E.1. Introducción

Para representar curvas planas, se empezó su estudio considerandolas como una representación de funciones reales de una variable.

$$y = f(x) \quad x \in [a, b]$$

Este es el mismo método que se va a usar para la representación paramétrica de las superficies, inicialmente esta va a estar representada por una única ecuación en la que una de sus coordenadas va a estar expresada explícitamente en función de las otras dos coordenadas.

$$z = f(x, y) \quad (x, y) \in D \tag{E.1}$$

Donde ahora el dominio puede ser cualquier subconjunto de R^2 al que se le impone la condición de ser conexo, esto quiere decir que a cualquier contorno cerrado dentro de este dominio solo deben pertenecerle puntos que estén dentro del dominio al que pertenece, dicho en otras palabras que no contenga agujeros.

Esto marca una gran diferencia con la definición del dominio de una curva en cuyo caso, este estaba representado por el intervalo de una recta real. En el caso de las superficies definir diferentes dominios incluso para una misma función $f(x, y)$ resulta complicado, ya que cada dominio puede expresarse bajo diversas estructuras algebraicas.

Una de las mayores preocupaciones cuando se trabaja con superficies es la elección de un apropiado espacio vectorial, sobre el cual plantear y resolver dichos problemas. Para

ilustrar un poco mas esta posición se toma la existencia de un dominio en R^2 cuyos $n + 1$ puntos son $(x_0, y_0), (x_1, y_1), (x_2, y_2)$ y sean $z_0, z_1, z_2, \dots, z_n$ los valores a interpolar, para este caso se esta buscando una función polinomial de modo que:

$$f(x_i, y_i) = z_i \quad i = 0, 1, 2, \dots, n \quad (\text{E.2})$$

Si la función $f(x, y)$ pertenece al espacio vectorial de dimensión $n + 1$, se puede elegir una base tal que:

$$B_0(x, y), B_1(x, y), \dots, B_n(x, y)$$

De forma tal que $f(x, y)$ depende de $n + 1$ parámetros respecto a la base

$$f(x, y) = \sum_{k=0}^n a_k B_k(x, y)$$

Imponiendo condiciones de interpolación:

$$\sum_{k=0}^n a_k B_k(x_i, y_i) = z_i \quad i = 0, 1, \dots, n$$

Lo que constituye un sistema lineal de $n+1$ ecuaciones en las $n+1$ incógnitas a_0, a_1, \dots, a_n que proporcionarán una solución única si cumplen con la condición de no anular el determinante.

$$|G| = \begin{bmatrix} B_0(x_0, y_0) & B_1(x_0, y_0) & \dots & B_n(x_0, y_0) \\ B_0(x_1, y_1) & B_1(x_1, y_1) & \dots & B_n(x_1, y_1) \\ \vdots & \vdots & \vdots & \vdots \\ B_0(x_n, y_n) & B_1(x_n, y_n) & \dots & B_n(x_n, y_n) \end{bmatrix} \neq 0$$

Este planteamiento es el mismo anteriormente propuesto para interpolar en una sola variable, para resolverlo es necesario definir un espacio vectorial de dimensión 4 y para ello se busca una base de tipo polinomial compuesta por cuatro funciones, por su simetría se va a tomar:

$$\mathcal{B} = \{1, x, y, xy\} \quad (\text{E.3})$$

El polinomio expresado en esta base queda:

$$P(x, y) = a_0 + a_1x + a_2y + a_3xy \quad (\text{E.4})$$

La solución a esta interpolación da lugar a una superficie que en este contexto será llamada *Superficie Bilineal*. Ahora bien considerando un polinomio arbitrario, por ejemplo:

$$P(x, y) = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2 \quad (\text{E.5})$$

Que es una combinación de la base

$$\mathcal{B} = \{1, x, y, xy, x^2, y^2\} \quad (\text{E.6})$$

En este último caso el espacio vectorial de los polinomios es 6, y se empieza a hacer evidente que la elección de la base es uno de los factores fundamentales a la hora de solucionar el problema de interpolación, por ello encontrar bases que resulten elecciones *razonables* es una tarea difícil, ya que existen infinitos polinomios que satisfacen y resuelven una misma postulación de puntos a interpolar. Desde luego existen bases polinomiales compuestas por cinco funciones, que no contienen funciones cruzadas de x e y por ejemplo:

$$\mathcal{B} = \{1, x, y, x^2, y^2\} \quad (\text{E.7})$$

Pero se sigue aumentando el número de puntos a interpolar, y con ello las dificultades en el proceso de interpolación, lo que debe llevar a entender la gran diferencia que existe cuando se trata de una base expresada en función de una o dos variables, y así mismo comprender que el planteamiento de determinados problemas conlleva serias restricciones sobre los datos del mismo. Para ello es claro que no es bueno partir de un número arbitrario de puntos, por el contrario genera mejores resultados hacer uso de bases como las dadas en los casos E.6 y E.3 para las que se van a plantear diferentes restricciones.

E.1.1. Silueta de una Base

Por razones de simetría solo se van a considerar las bases del tipo E.3 y E.6, para estos casos se utiliza en método del *producto tensorial* de dos bases de potencias, de grados m y n respectivamente en las variable x e y

$$\mathcal{B} = [1, x, x^2, \dots, x^n] \otimes [1, y, y^2, \dots, y^m] \quad (\text{E.8})$$

$$= \{1, x, x^2, \dots, x^n, xy, x^2y, \dots, x^ny, y^2,$$

$$xy^2, x^2y^2, \dots, x^ny^2, \dots, y^m, xy^m, \dots, x^ny^m\}$$

En este caso las funciones son todos los posibles productos entre una potencia x^k de la primera base y otra y^i de la segunda base, de esta base se dice que posee una *silueta rectangular* con ello se pueden agrupar todos sus elementos en una estructura rectangular:

$$\begin{array}{cccccc} 1 & x & x^2 & \dots & x^n & \\ y & xy & x^2y & \dots & x^ny & \\ y^2 & xy^2 & x^2y^2 & \dots & x^ny^2 & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ y^m & xy^m & x^2y^m & \dots & x^ny^m & \end{array} \quad (\text{E.9})$$

Construida por $N = (m + 1)(n + 1)$ funciones y esta es la dimensión del espacio que generan. Este es el tipo de base considerada en 3 con $m = n = 1$ y corresponde al producto tensorial de dos bases univariadas de grado 1:

$$[1 \ x] \otimes [1 \ y] = [1 \ x \ y \ xy]$$

Para esta primera generalización cabe mencionar que de una base en una variable el espacio que genera no es de los polinomios de grado menor o igual que $n + m$. Como es posible apreciar en el caso en que $n = m = 1$ cuyo resultado no es un polinomio arbitrario de segundo grado.

La segunda generalización a tener en cuenta es la considerada en 5.5 verdaderos polinomios de grado menor o igual que n :

$$\mathcal{B} = \{1, x, x^2, \dots, x^n, y, xy, x^2y, \dots, x^{n-1}y, \dots, y^{n-1}, xy^{n-1}, y^n\} \quad (\text{E.10})$$

Entonces las funciones de las bases son:

$$x^i y^j \quad \text{con} \quad i + j \leq n$$

Cuya base rectangular es:

$$\begin{array}{ccccccc}
 1 & x & x^2 & \dots & x^{n-2} & x^{n-1} & x^n \\
 y & xy & x^2y & \dots & x^{n-2}y & x^{n-1}y & \\
 y^2 & xy^2 & x^2y^2 & \dots & x^{n-2}y^2 & & \\
 \vdots & \vdots & \vdots & & & & \\
 \vdots & \vdots & \vdots & & & & \\
 y^{n-1} & xy^{n-1} & & & & & \\
 y^n & & & & & &
 \end{array} \tag{E.11}$$

Esta es la base triangular posee al menos la mitad de funciones de una base rectangular, en este caso $N = \frac{(n+1)(n+1)}{2}$ elementos, siendo esta por lo tanto la dimensión que generan.

E.2. Superficies producto tensorial

Se considera una base rectangular y una función expresada como la combinación lineal de sus elementos que tendrá la siguiente forma:

$$P(x, y) = \sum_{i=0}^n \sum_{j=0}^m a_{ij} x^i y^j \tag{E.12}$$

Esta es la denominada superficie *producto tensorial* y puede ser descrita como el conjunto de puntos generados cuando una curva *barre* el espacio a la vez que se *deforma*.

$$P(x, y) = \sum_{i=0}^n \left(\sum_{j=0}^m a_{ij} y^j \right) x^i = \sum_{i=0}^n c_i x^i$$

La superficie puede ser considerada como una curva polinomial en la variable x , donde los coeficientes están *controlados* por otra curva polinomial, esta vez en la variable y . Se eligen bases de potencias en las dos variables que pueden ser diferentes en sus principios de estructura bien sea rectangular o triangular.

si se consideran dos bases arbitrarias en x e y :

$$B_x = \{B_0(x), \dots, B_n(x)\} \quad B_y = \{C_0(y), \dots, C_n(y)\}$$

Por lo tanto cualquier potencia $x^k y^j$ puede expresarse:

$$x^k y^j = \left[\sum_{i=0}^n \alpha_{ki} B_i(x) \right] \left[\sum_{j=0}^m \beta_{hj} C_j(y) \right]$$

reordenando:

$$P(x, y) = \sum_{i=0}^n \sum_{j=0}^m b_{ij} B_i(x) C_j(y) \quad (\text{E.13})$$

Recuerde siempre que una superficie de producto tensorial es aquella que constituye una combinación lineal de todos los productos posibles entre las funciones de dos bases arbitrarias, cada una en una de las variables x e y .

Expresando esta superficie en forma matricial:

$$P(x, y) = [B_0(x) B_1(x) \dots B_n(x)] \begin{bmatrix} b_{00} & b_{01} & \dots & b_{0m} \\ b_{10} & b_{11} & \dots & b_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ b_{n0} & b_{n1} & \dots & b_{nn} \end{bmatrix} \begin{bmatrix} C_0(y) \\ C_1(y) \\ \vdots \\ C_m(y) \end{bmatrix}$$

E.2.1. Interpolación con superficies producto tensorial

La dimensión del espacio de funciones de la forma 5.1.2 ó 5.1.3 es $N = (n + 1)(m + 1)$ por lo que este debe ser el número de puntos a interpolar para que el problema de interpolación quede bien planteado. Es conveniente que los puntos posean una característica en su estructura que sea similar a la de la base, es decir que puedan agruparse en un esquema rectangular.

$$\begin{array}{cccccc} (x_0, y_0, z_{00}) & (x_1, y_0, z_{10}) & (x_2, y_0, z_{20}) & \dots & (x_n, y_0, z_{n0}) \\ (x_0, y_1, z_{01}) & (x_1, y_1, z_{11}) & (x_2, y_1, z_{21}) & \dots & (x_n, y_1, z_{n1}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (x_0, y_m, z_{0m}) & (x_1, y_m, z_{1m}) & (x_2, y_m, z_{2m}) & \dots & (x_n, y_m, z_{nm}) \end{array}$$

De forma tal que los argumentos (x_i, y_i) forman los nodos de una malla, producto cartesiano de dos soportes, en x e y respectivamente.

$$S = \{x_0, x_1, \dots, x_n\} \otimes \{y_0, y_1, \dots, y_m\}$$

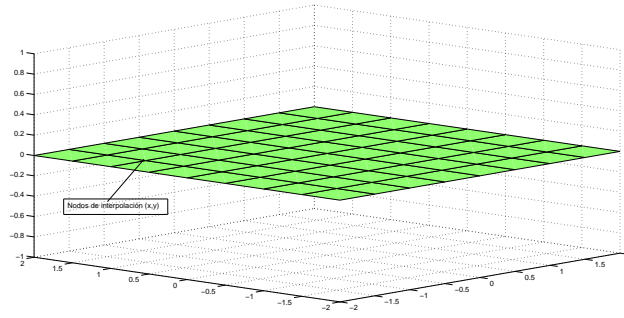


Figura E.1: Nodos formados por los soportes en u y v

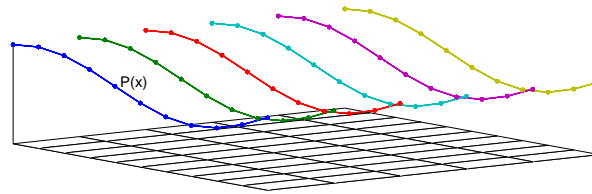


Figura E.2: Interpolación de las alturas

Donde a cada punto $(x_i, y_j,)$ se le tiene asignada una *altura en* z_{ij} tal que:

$$P(x, y) = z_{ij} \quad 0 \leq i \leq n \quad 0 \leq j \leq m \quad (\text{E.14})$$

La ventaja que presentan las superficies de producto tensorial, es la facilidad de resolver los problemas en dos pasos, primero en la variable x manteniendo constante el valor de y , algo así como trabajar en un plano de $y=cte$ sobre el cual se interpolan unos puntos de coordenadas (x, z) , por medio de un polinomio $P(x)$ tal que $z = P(x)$ la y a continuación llevar a cabo el mismo procedimiento en la variable y .

Considerando los puntos $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_0)$, las alturas $z_{00}, z_{10}, \dots, z_{n0}$ pueden ser interpoladas con cualquier técnica en una variable, de tal forma que se obtiene una curva:

$$z_{(i,0)} = P_0(x) = \sum_{i=0}^n a_{i0} B_i(x) \quad \text{con } y = y_0$$

$B(x)$ es una base de los polinomios de grado menor o igual que n , para

calcular los polinomios $P_1(x)$ puede repetirse el procedimiento con los puntos $(x_0, y_1), (x_1, y_1), \dots, (x_n, y_1)$, y se interpola a $z_{01}, z_{11}, \dots, z_{n1}$, repitiendo este método se obtienen los $(n + 1)$ polinomios para una variable¹.

$$\begin{aligned}
 P_0(x) &= \sum_{i=0}^n a_{i0} B_i(x) & \text{verifica} & & P_0(x_k) &= z_{k0} & \cdots & & k &= 0, \dots, n \\
 P_1(x) &= \sum_{i=0}^n a_{i1} B_i(x) & \text{verifica} & & P_1(x_k) &= z_{k1} & \cdots & & k &= 0, \dots, n \\
 & \vdots & & & \vdots & & \ddots & & \vdots & \\
 P_m(x) &= \sum_{i=0}^n a_{im} B_i(x) & \text{verifica} & & P_m(x_k) &= z_{km} & \cdots & & k &= 0, \dots, n
 \end{aligned} \tag{E.15}$$

Una vez que se termina la interpolación en la variable x , se procede a realizar la interpolación en la variable y con respecto al soporte y_0, y_1, \dots, y_m y se utiliza la base $C(y)$ de los polinomios de grado menor o igual a m . Pero esta segunda serie de interpolaciones ya no es la de los puntos a_{ij} sino la de los coeficientes de los polinomios que se hallaron en la primera serie de interpolaciones cuando se tomaba a $y = \text{constante}$, nuevamente se considera el caso para el que $x = x_0$ considerando el polinomio:

$$Q_0(y) = \sum_{j=0}^m b_{0j} C_j(y) \quad \text{tal que } Q_0(y_k) = a_{0k} \quad k=0,1,\dots,m$$

este es el polinomio que interpola al primer coeficiente de cada una de las sumas en la variable x descrita por la ecuación 5.1.5, análogamente se repite el procedimiento para $x = x_1$ se tiene el polinomio $Q_1(y)$ interpolando al segundo coeficiente de cada suma.

$$Q_1(y) = \sum_{j=0}^m b_{1j} C_j(y) \quad \text{con } Q_1(y_k) = a_{1k} \quad k=0,1,\dots,m$$

Y de esta manera se construyen $(n + 1)$ polinomios interpolando todos los coeficientes.

$$Q_i(y) = \sum_{j=0}^m b_{ij} C_j(y) \quad \text{tal que } Q_i(y_k) = a_{ik} \quad k=0,1,\dots,m \tag{E.16}$$

Ahora se considera el polinomio en dos variables :

¹No necesariamente los polinomios interpoladores son iguales

$$P(x, y) = \sum_{i=1}^n Q_i(y) B_i(x) \quad (\text{E.17})$$

para dos puntos cuales quiera (x_k, y_h) entonces:

$$P(x_k, y_h) = \sum_{i=0}^n Q_i(y_h) B_i(x_k) = \sum_{i=0}^n a_{ih} B_i(x_k) = P_h(x_k) = z_{kh}$$

Por lo tanto el polinomio E.15 interpola los puntos (x_k, y_h, z_{kh}) para $0 \leq k \leq n$, $0 \leq h \leq m$.

Nótese que $P(x, y)$ se puede escribir así:

$$P(x, y) = \sum_{i=0}^n \left[\sum_{j=0}^m b_{ij} C_j(y) \right] B_i(x) = \left[\sum_{i=0}^n r_i B_i(x) \right] \left[\sum_{j=0}^m s_j C_j(y) \right]$$

Que es lo que se conoce como un producto tensorial, este procedimiento será mas sencillo, si para la interpolación en la segunda variable se toman polinomios de Lagrange, ya que para este caso se sabe que los coeficientes del polinomios respecto a la base son los propios valores a interpolar. Por lo tanto, estos polinomios se pueden escribir así:

$$Q_i(y) = \sum_{j=0}^m a_{ij} L_j(y)$$

Y el polinomio $P(x, y)$

$$\begin{aligned} P(x, y) &= \sum_{i=0}^n Q_i(y) B_i(x) = \sum_{i=0}^n \left[\sum_{j=0}^m a_{ij} L_j(y) \right] B_i(x) = \\ &= \sum_{j=0}^m \left[\sum_{i=0}^n a_{ij} B_i(x) \right] L_j(y) = \sum_{j=0}^m P_j(x) L_j(y) \end{aligned} \quad (\text{E.18})$$

Esta expresión muestra al polinomio $P(x, y)$ al que se le puede considerar interpolando en una variable y las curvas $P_j(x)$, que son polinomios interpoladores, pero a su vez pueden ser considerados como curvas cualesquiera. ahora bien por las propiedades de los polinomios de Lagrange se tendrá entonces:

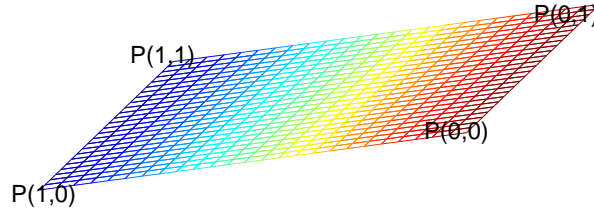


Figura E.3: Superficie Bilineal

$$P(x, y_k) = \sum_{j=0}^m P_j(x) L_j(y) = P_x(y)$$

sabiendo que:

$$P_j(x) = \sum_{i=0}^n z_{ij} \bar{L}_i(x) \quad \text{se tiene entonces :}$$

$$\begin{aligned}
 P(x, y) &= \sum_{j=0}^m P_j(x) L_j(y) = \sum_{j=0}^m \left[\left[\sum_{i=0}^n z_{ij} \bar{L}_i(x) \right] L_j(y) \right] = \\
 &\sum_{i=0}^n \sum_{j=0}^m z_{ij} \bar{L}_i(x) L_j(y)
 \end{aligned} \tag{E.19}$$

En esta expresión aparecen explícitamente los valores z_{ij} a interpolar.

E.2.2. Superficie Bilineal

Esta es la conocida superficie para la que se interpolan cuatro puntos $(x_0, y_0, z_{00}), (x_0, y_1, z_{01}), (x_1, y_0, z_{10}), (x_1, y_1, z_{11})$, la superficie tensorial correspondiente al caso $n = m = 1$, esta es la superficie mas sencilla pasando por cuatro puntos en R^3 .

Para $y = y_0$, se trata de interpolar los puntos $(x_0, z_{00}), (x_1, z_{10})$, haciendo uso de la base de potencias

$$P_0(x) = \frac{x_1 z_{00} - x_0 z_{10}}{x_1 - x_0} + \frac{z_{10} - z_{00}}{x_1 - x_0} x = a_{00} + a_{10} x \tag{E.20}$$

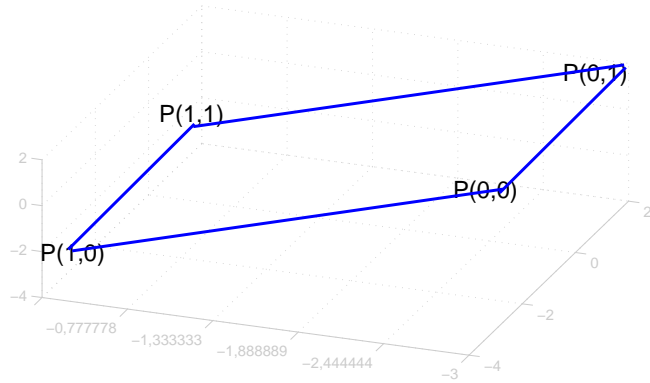


Figura E.4: Líneas que interceptan por los puntos $P(0, 0)$, $P(0, 1)$, $P(1, 0)$, $P(1, 1)$

Para $y = y_1$

$$P_1(x) = \frac{x_1 z_{01} - x_0 z_{11}}{x_1 - x_0} + \frac{z_{11} - z_{01}}{x_1 - x_0} x = a_{01} + a_{11} x \quad (\text{E.21})$$

En la variable y , para $x = x_0$:

$$Q_0(y) = \frac{y_1 a_{00} - y_0 a_{01}}{y_1 - y_0} + \frac{a_{01} - a_{00}}{y_1 - y_0} y$$

Para $y = y_1$

$$Q_1(y) = \frac{y_1 a_{10} - y_0 a_{11}}{y_1 - y_0} + \frac{a_{11} - a_{10}}{y_1 - y_0} y$$

Para lo cuál la superficie Bilineal viene dada por:

$$\begin{aligned} P(x, y) &= Q_0(y) + Q_1(y)x = \\ &= \frac{y_1 a_{00} - y_0 a_{01}}{y_1 - y_0} + \frac{a_{01} - a_{00}}{y_1 - y_0} y + \frac{y_1 a_{10} - y_0 a_{11}}{y_1 - y_0} + \frac{a_{11} - a_{10}}{y_1 - y_0} y x = \\ &= \frac{y_1 a_{00} - y_0 a_{01}}{y_1 - y_0} + \frac{y_1 a_{10} - y_0 a_{11}}{y_1 - y_0} x + \frac{a_{01} - a_{00}}{y_1 - y_0} y + \frac{a_{11} - a_{10}}{y_1 - y_0} x y \end{aligned}$$

Se sustituyendo finalmente las expresiones para a_{ij} :

$$\begin{aligned} P(x, y) &= \frac{x_1 y_1 z_{00} - x_0 y_1 z_{10} - x_1 y_0 z_{01} + x_0 y_0 z_{11}}{(x_1 - x_0)(y_1 - y_0)} + \\ &\quad \frac{y_1(x_1 z_{00} - x_0 z_{10}) - y_0(x_1 z_{01} + x_0 z_{11})}{(x_1 - x_0)(y_1 - y_0)} x \\ &\quad + \frac{x_1(y_1 z_{00} - y_0 z_{01}) - x_0(y_1 z_{10} + y_0 z_{11})}{(x_1 - x_0)(y_1 - y_0)} y + \end{aligned} \quad (\text{E.22})$$

$$+ \frac{z_{00} - z_{10} - z_{01} + z_{11}}{(x_1 - x_0)(y_1 - y_0)} xy$$

Agrupando términos:

$$P(x, y) = z_{00} + \frac{z_{10} - z_{00}}{x_1 - x_0}(x - x_0) + \frac{z_{01} - z_{00}}{y_1 - y_0}(y - y_0) + \frac{z_{00} - z_{01} - z_{10} + z_{11}}{(x_1 - x_0)(y_1 - y_0)} \quad (\text{E.23})$$

si se utiliza la Base Bernstein

$$\left. \begin{aligned} B_0(u) &= 1 - u = 1 - \frac{x - x_0}{x_1 - x_0} \\ B_1(u) &= u = \frac{x - x_0}{x_1 - x_0} \end{aligned} \right| \begin{aligned} B_0(v) &= 1 - v = 1 - \frac{y - y_0}{y_1 - y_0} \\ B_1(v) &= v = \frac{y - y_0}{y_1 - y_0} \end{aligned}$$

Por lo tanto la superficie Bilineal esta dada finalmente por:

$$P(x, y) = z_{00}(1 - u)(1 - v) + z_{01}(1 - u)v + z_{10}u(1 - v) + z_{11}uv \quad (\text{E.24})$$

E.3. Superficies paramétricas

Después de realizar la elección de las bases de la superficies de la forma $f(x, y)$, se debe pasar a la forma paramétrica:

$$S(u, v) \equiv \begin{cases} x = f(u, v) \\ y = g(u, v) \\ z = h(u, v) \end{cases} \quad (\text{E.25})$$

para los cual h , f y g son funciones de un espacio vectorial, con una base producto tensorial. Modificando los valores de los parámetros se recorre la superficie, obteniéndose las coordenadas de los puntos que están sobre ella.

$$S(u, v) \equiv \begin{cases} x = \sum_{i=0}^n \sum_{j=0}^m a_{ij} B_i(u) C_j(v) \\ y = \sum_{i=0}^n \sum_{j=0}^m b_{ij} B_i(u) C_j(v) \\ z = \sum_{i=0}^n \sum_{j=0}^m c_{ij} B_i(u) C_j(v) \end{cases}$$

Las expresiones paramétricas presentan la ventaja de ser flexibles, aunque no son orientables. Por otra parte, cualquier expresión explícita se puede poner, trivialmente, en forma paramétrica.

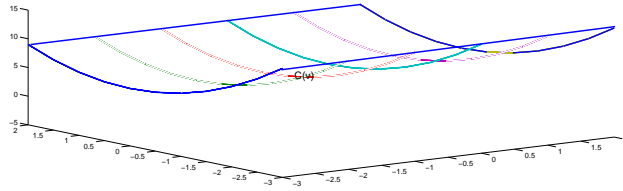


Figura E.5: Curva isoparamétrica

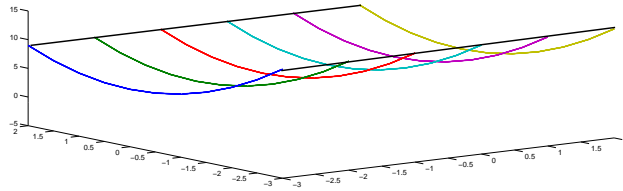


Figura E.6: Familia de curvas

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} B_i(u) C_j(v) \quad (\text{E.26})$$

Las funciones base, se han supuesto polinómicas, pero pueden ser B-splines, o Curvas racionales, así mismo $P_{ij} = (a_{ij}, b_{ij}, c_{ij})$ son coeficientes que pueden ser considerados como puntos en R^3 . que controlan las propiedades de la superficie.

E.3.1. Curvas Isoparamétricas

Al considerar un valor concreto para el parámetro $u = \bar{u}$ de una superficie paramétrica cualquiera $S(u, v)$, se obtiene una curva que esta contenida en la superficie, a las que se le denomina curva isoparamétrica, que generalmente se usa para describir la forma de la superficie, tomando una valor discreto de parámetros $u = u_0, u_1, \dots, u_p$, que se visualiza mas claramente dibujando una familia de curvas contenidas dentro de la superficie, obtenidas a partir de fijar un valor constante para uno de los parámetros.

Ahora bien, si se asume un valor constante para la variable v , se obtienen curvas isoparamétricas en la superficie: $C(u) = S(u, \bar{v})$. La representación simultanea de curvas isoparamétricas donde $u = constante$ y $v = constante$, es un procedimiento eficiente para la visualización de la forma de una superficie y sus propiedades geométricas.

Considerando una superficie Bilineal:

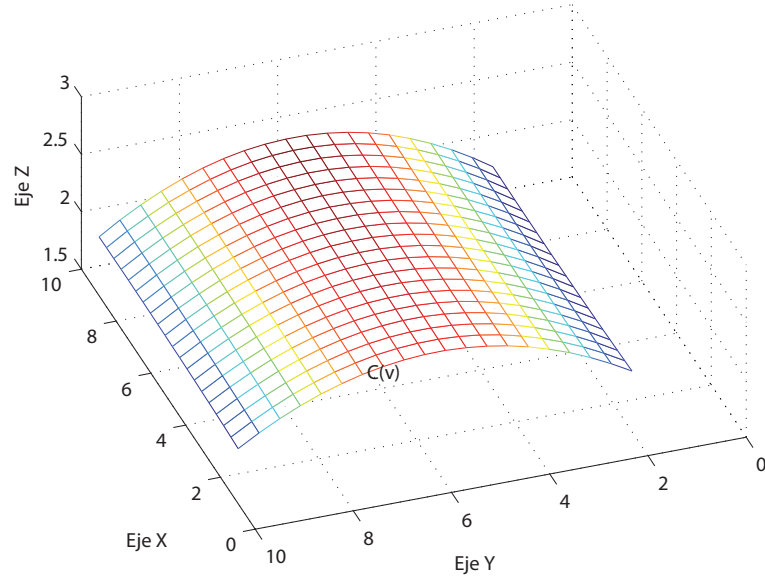


Figura E.7: Superficie isoparamétrica

$$S(u, v) = P_{00}(1 - u)(1 - v) + P_{10}(1 - v)u + P_{01}(1 - u)v + P_{11}uv$$

Donde $P_{00}, P_{10}, P_{01}, P_{11}$ son puntos de R^3 que son interpolados respectivamente para $(u, v) = (0, 0), (1, 0), (0, 1), (1, 1)$ para un valor de $u = \bar{u}$, $S(u, v)$, es una función lineal en la variable v . Por lo tanto la curva isoparamétrica $C(u) = S(\bar{u}, v)$ es una recta que contiene a los puntos Q_0 y Q_1 , que a su vez se encuentran ubicados sobre las rectas que unen los puntos $P_{00}, P_{10}, P_{01}, P_{11}$ respectivamente. De la misma manera estas rectas corresponden a las curvas isoparamétricas para v constante.

E.3.2. Superficies Regladas

Una superficie reglada se puede definir como el resultado de interpolar linealmente puntos correspondientes a un mismo valor del parámetro $u = \bar{u}$ situados sobre dos curvas $C_1(u)$ y $C_2(u)$, que corresponden a una simple generalización de la superficie bilineal, cuya mayor aplicación se encuentra dentro de la industria aeronaval ya que las alas de los aviones son generalmente de este tipo, en efecto se tiene:

$$S(u, v) = (1 - v)C_1(u) + vC_2(u) \tag{E.27}$$

Para lo cuál $C_1(u)$ y $C_2(u)$, pueden ser cualesquiera curvas 3D, es evidente que las curvas isoparamétricas para u =constante son rectas, mientras que para v constante son

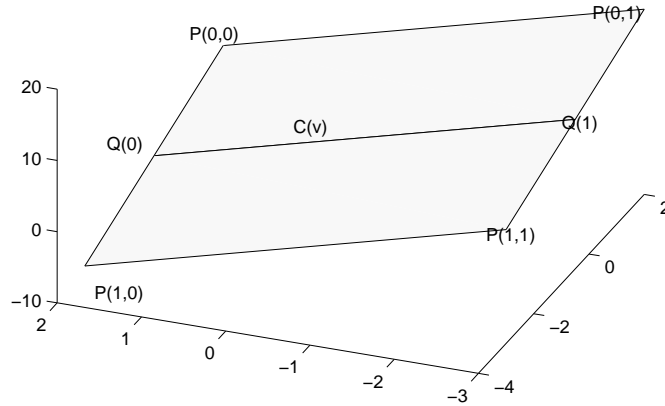


Figura E.8: Superficie doblemente reglada

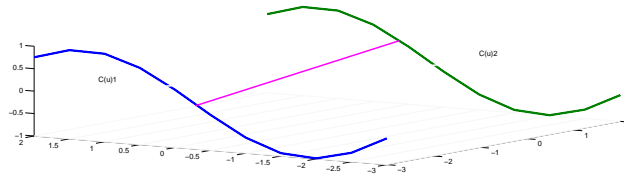


Figura E.9: Superficie reglada

una combinación convexa de las curvas $C_1(u)$ y $C_2(u)$.

E.3.3. Superficies Cilíndricas

Uno de los casos particulares de las superficies regladas son los *cilindros generalizados* que se obtienen al desplazar una curva $C(u)$ a través de un espacio siguiendo una trayectoria recta.

Para este caso particular se llama \mathcal{D} al vector que indica la dirección del desplazamiento, dando como resultado para la ecuación de la superficie:

$$S(u, v) = (1 - v)C(u) + v[C(u) + \mathcal{D}] = C(u) + v\mathcal{D} \quad (\text{E.28})$$

Generalmente \mathcal{D} se elige en la dirección perpendicular al plano que contiene la curva $C(u)$. Una de las mejores características de este tipo de superficies es que son *desarrollables* es decir que pueden ser deformadas en un plano sin llegar a variar

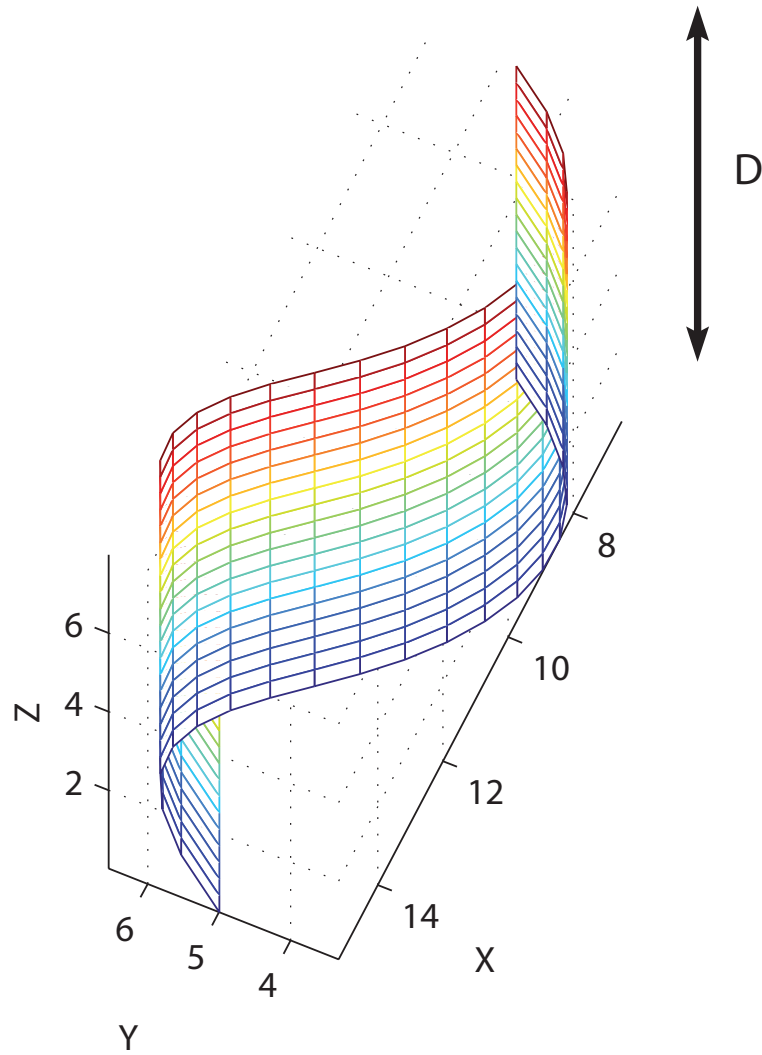


Figura E.10: Superficie cilíndrica

la distancia entre cualquiera de sus puntos, siendo de gran utilidad en la industria metalúrgica ya que puede ser rotada infinitesimalmente sin estriarla o contraerla.

E.3.4. Superficies de revolución

Estas son la superficies que se obtienen al rotar una curva al rededor de un eje, ahora bien para simplificar su formulación se supone que el eje de rotación coincide con el eje z , considerando este caso, la curva que sirve de *generatriz* se supone plana y contenida en el plano XZ , se tiene entonces para la curva $C(u)$ representada paraméricamente:

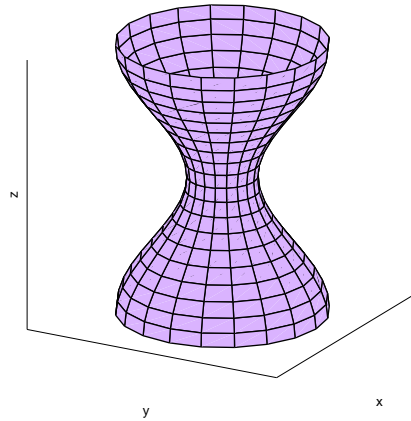


Figura E.11: Superficie de revolución

$$C(u) = \begin{cases} x = f(u) \\ +0 \\ z = g(u) \end{cases}$$

El ángulo de rotación φ respecto viene dado por la transformación:

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\text{sen}\varphi & 0 \\ \text{sen}\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \text{Para ello la curva transformada es:}$$

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\text{sen}\varphi & 0 \\ \text{sen}\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f(u) \\ 0 \\ g(u) \end{bmatrix} = \begin{bmatrix} f(u)\cos\varphi \\ f(u)\text{sen}\varphi \\ g(u) \end{bmatrix}$$

Que se puede escribir:

$$C(u) = \begin{cases} \bar{x} = f(u)\cos\varphi \\ \bar{y} = f(u)\text{sen}\varphi \\ \bar{z} = g(u) \end{cases}$$

La superficie de revolución se obtiene tomando un valor arbitrario entre 0 y 2π para φ , llamando $v=\varphi$, al segundo parámetro.

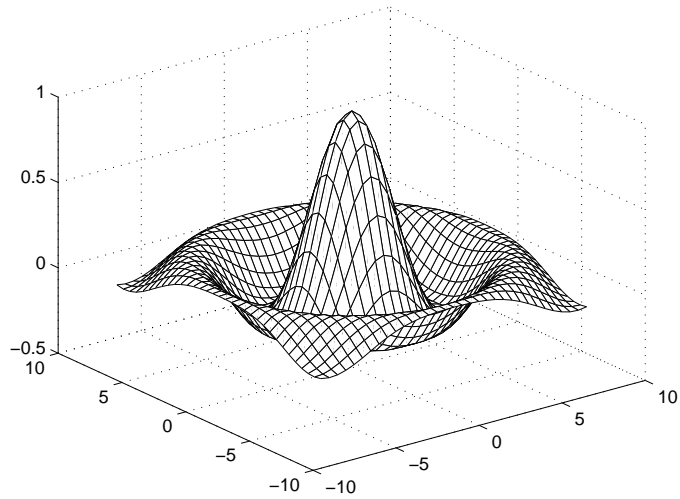


Figura E.12: Rotación alrededor del eje z

$$C(u) = \begin{cases} x = f(u)\cos\varphi \\ y = f(u)\text{sen}\varphi \\ z = g(u) \end{cases}$$

De esta forma haciendo uso de una malla de curvas isoparamétricas, la superficie de revolución obtenida al rotar la curva:

$$C(u) = \begin{cases} x = u \\ y = 0 \\ z = \frac{\text{sen } u}{u} \end{cases}$$

Cabe resaltar que para esta superficie, las variables están separadas como en el caso del producto tensorial, ya que la variable u se encuentra las funciones que definen la curva generatriz $C(u)$ mientras que en la variable v se tiene:

$$x = \cos v$$

$$y = \text{sen } v$$

que corresponden a la circunferencia cuyo centro es el origen y cuyo radio es 1,

suponiendo $z=0$.

Para este caso se puede suponer que la curva $C(u)$ se ha multiplicado por la circunferencia para obtener la superficie de revolución:

$$\begin{aligned} x &= c(v) \\ y &= s(v) \end{aligned} \quad 0 \leq v \leq v_1 \quad (\text{E.29})$$

Ahora, si el producto de la curva por la circunferencia, es la misma superficie de revolución, con una parametrización distinta es:

$$T(u, v) = \begin{cases} x = f(u)c(v) \\ y = f(u)s(v) \\ z = g(u) \end{cases} \quad (\text{E.30})$$

Tomando secciones de $z=\text{constante}$, que vienen dadas por valores de $u=\text{constante}$, se tiene:

$$T(\bar{u}, v) = F(v) = \begin{cases} x = f(\bar{u})c(v) \\ y = f(\bar{u})s(v) \\ z = g(\bar{u}) \end{cases}$$

Que es la ecuación de la circunferencia con radio $f(\bar{u})$ y centrada en el punto $(0, 0, g(\bar{u}))$, ahora si se considera $v=\text{constante}$.

$$T(u, \bar{v}) = G(u) = \begin{cases} x = f(u)c(\bar{v}) \\ y = f(u)s(\bar{v}) \\ z = g(u) \end{cases}$$

De este modo se obtiene una curva que mediante la afinidad

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \begin{bmatrix} c(\bar{v}) & s(\bar{v}) & 0 \\ -s(\bar{v}) & c(\bar{v}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

que se transforma en:

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \begin{bmatrix} [c^2(\bar{v}) + s^2(\bar{v})]f(u) \\ 0 \\ g(u) \end{bmatrix} = \begin{bmatrix} f(u) \\ 0 \\ g(u) \end{bmatrix} = C(u)$$

Teniendo en cuenta que la afinidad es una rotación, resulta que las curvas de $v=constante$ son rotaciones al rededor del eje z de la curva generatriz $C(u)$. Para efectos prácticos es útil considerar la ecuación de una circunferencia expresada en la misma base de definición de la curva $C(u)$. Haciendo uso de funciones racionales para la circunferencia, se cuenta con sencillas ecuaciones:

$$\begin{aligned} x &= \frac{1-v^2}{1+v^2} & -\infty < v < \infty \\ y &= \frac{2v}{1+v^2} \end{aligned}$$

con lo que se puede construir la superficie de revolución así:

$$T(u, \bar{v}) = \begin{cases} x = f(u) \frac{1-v^2}{1+v^2} \\ y = f(u) \frac{2v}{1+v^2} \\ z = g(u) \end{cases}$$

De esta forma si se tienen $f(u)$ y $g(u)$ expresadas como curvas racionales, la superficie de revolución también estará expresada como una función racional. Mas interesante aún sería considerara, la circunferencia como una curva de Bézier o un B-Spline racional (NURBS), en este caso si la curva generatriz es del mismo tipo la superficie de revolución quedará definida por la misma base de funciones empleada para definir la curva.

E.3.5. Superficies Swing

Son una generalización de las superficies de revolución, que hacen uso de dos funciones arbitrarias que no necesariamente deben ser una circunferencia:

$$S(u, v) = \begin{cases} x = f(u)c(v) \\ y = f(u)s(v) \\ z = g(u) \end{cases} \quad (\text{E.31})$$

Se considera como el producto de dos curvas contenidas en el plano XY y XY :

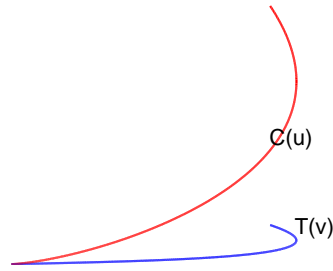


Figura E.13: Curvas planas contenidas en planos perpendiculares

$$C(u) = \begin{cases} x = f(u) \\ y = 0 \\ z = g(u) \end{cases} \quad T(v) = \begin{cases} x = c(v) \\ y = s(v) \\ z = 0 \end{cases} \quad (\text{E.32})$$

Por analogía con el procedimiento constructivo para las superficies de revolución.

$$S(u, v) = \begin{bmatrix} c(v) & -s(v) & 0 \\ s(v) & c(v) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f(u) \\ 0 \\ g(u) \end{bmatrix}$$

Donde la matriz ahora no debe ser ortogonal ya que $c^2(v) + s^2(v) \neq 1$ sin embargo la matriz se puede factorizar tal que:

$$S(u, v) = \begin{bmatrix} r(v) & 0 & 0 \\ 0 & r(v) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{c(v)}{r(v)} & -\frac{s(v)}{r(v)} & 0 \\ \frac{s(v)}{r(v)} & \frac{c(v)}{r(v)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f(u) \\ 0 \\ g(u) \end{bmatrix} \quad (\text{E.33})$$

siendo $r(v) = \sqrt{c^2(v) + s^2(v)}$.

De manera ilustrativa, se puede pensar en la generación de esta superficie como, si la curva $C(u)$ es dibujada en un plano vertical desde $z=0$ que gira alrededor del eje z recorriendo en un sentido u otro la trayectoria marcada por la función $T(v)$

Lo que da lugar a la representación de la curva *swing* obtenida como el *producto* de la curva perfil.

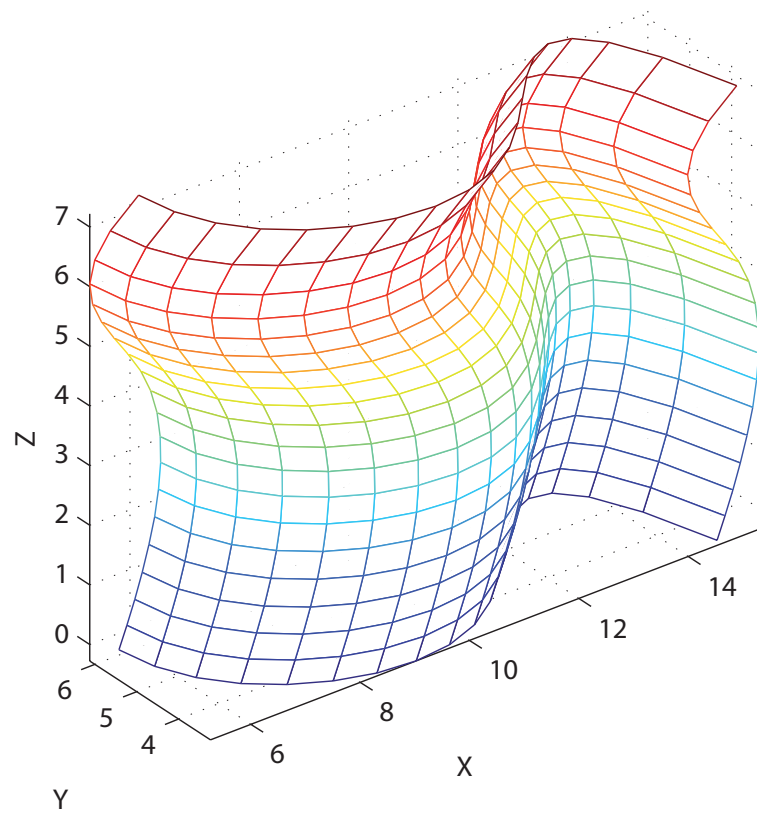


Figura E.14: Superfície swing

$$C(u) = \begin{cases} x = u \\ y = 0 \\ z = -u^2 + 3 \end{cases}$$

Con la curva trayectoria:

$$T(v) = \begin{cases} x = -2v^2(v^2 - 1) + v + 1 \\ y = -2v^2(v^2 - 1) - v + 1 \\ z = 0 \end{cases}$$

E.3.6. Superficies generadas por Barrido

Cabe resaltar que las técnicas para construcción de las superficies anteriormente vistas son buenas herramientas, pero son sólo casos particulares de una técnica de generación de superficies que se conoce como barrido *osweeping*.

Las técnicas anteriormente mostradas consisten en desplazar una curva a través del espacio y obligarla a seguir una trayectoria dada por otra función, pero es más general permitir que la curva sea rotada y escalonada a la vez que se mueve.

$$S(u, v) = A(v)C(u) + T(v) \tag{E.34}$$

donde

$$\begin{bmatrix} a_{11}(u) & a_{12}(u) & a_{13}(u) \\ a_{21}(u) & a_{22}(u) & a_{23}(u) \\ a_{31}(u) & a_{32}(u) & a_{33}(u) \end{bmatrix}$$

es una combinación Afín de rotación y escalado al mismo tiempo. Se debe observar que la curva $C(u)$ y $T(v)$ no están obligados a tener un punto en común, generalmente, la superficie de revolución no contendrá a la curva de trayectoria.

Un caso particularmente interesante es el de las superficies de barrido translacional, que corresponde a tomar $A(v) = I$ matriz de identidad: $S(u, v) = C(u) + T(v)$

Aquí la curva $C(u)$ simplemente se traslada en el espacio, siguiendo la trayectoria $T(v)$ sin sufrir ningún tipo de deformación o escalado. por ejemplo una curva definida en el plano XZ , que se traslada siempre permaneciendo un plano vertical paralelo al XZ ,

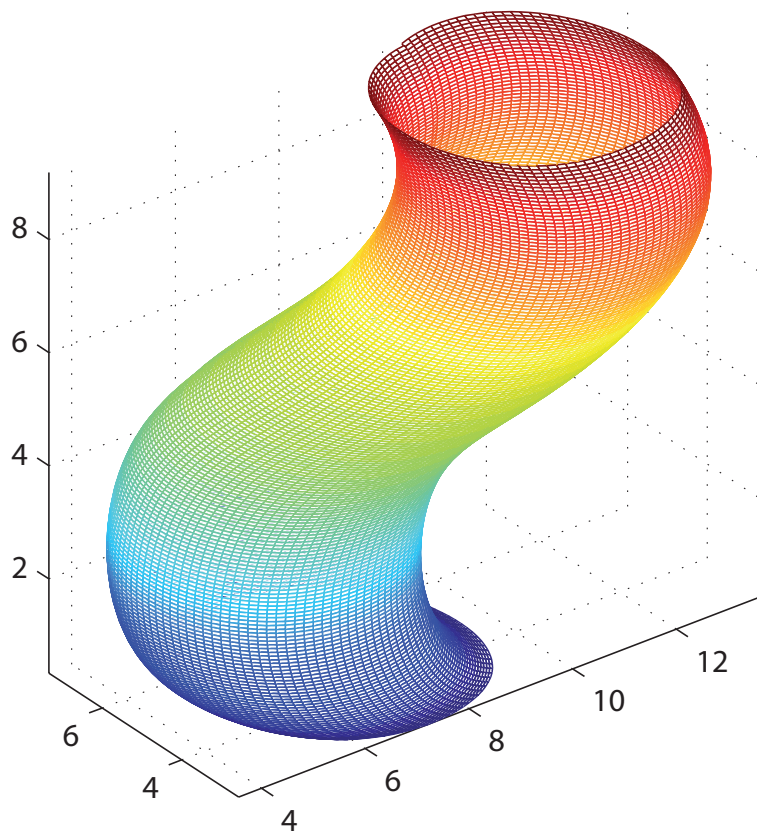


Figura E.15: Superficie de barrido siguiendo una trayectoria

según una trayectoria definida en el plano YZ .

Cuando se utiliza una matriz de deformación $A(v)$, frecuentemente se le aplica a la curva un escalado o rotado, aunque no resulta poco probable la combinación de estas dos transformaciones ya que predecir de manera exacta la forma de la superficie resultante se hace complicado.

Cuando se le aplica un escalado a una curva contenida en el plano XZ y su vez se desplaza a lo largo del eje Y , resulta:

$$A(v) = \begin{bmatrix} f(v) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f(v) \end{bmatrix} \quad \text{Para lo que : } f(v) = \frac{1}{2} \left(\frac{v^2}{4} + 1 \right)$$

siendo $f(2) = f(-2) = 1$ y $f(0) = \frac{1}{2}$, lo que corresponde a las pretensiones de trasladar la curva, para $v = 2$ y -2 empezando una contracción en la medida que aumenta el valor de v hasta llegar a $v = 0$ en donde se contrae a la mitad, sigue su recorrido sobre la trayectoria en Y hasta recuperar su tamaño normal en $v = 2$.

las ecuaciones que representan esta transformación son:

$$S(u, v) = \begin{bmatrix} f(v) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f(v) \end{bmatrix} \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} + \begin{bmatrix} 0 \\ v \\ 0 \end{bmatrix} \quad -2 \leq v \leq 2$$

Ahora en el caso de la rotación son particularmente dos casos, cuando la curva rota sobre el plano que la contiene al mismo tiempo que se desplaza, y el segundo cuando el que rota es el plano para mantener el vector normal al plano paralelo al vector tangente a la trayectoria.

El primer caso supone que la curva se encuentra sobre el plano XZ

$$C(u) = \begin{cases} x = f(u) \\ 0 \\ z = g(u) \end{cases}$$

Entonces la matriz de rotación alrededor del eje Y es

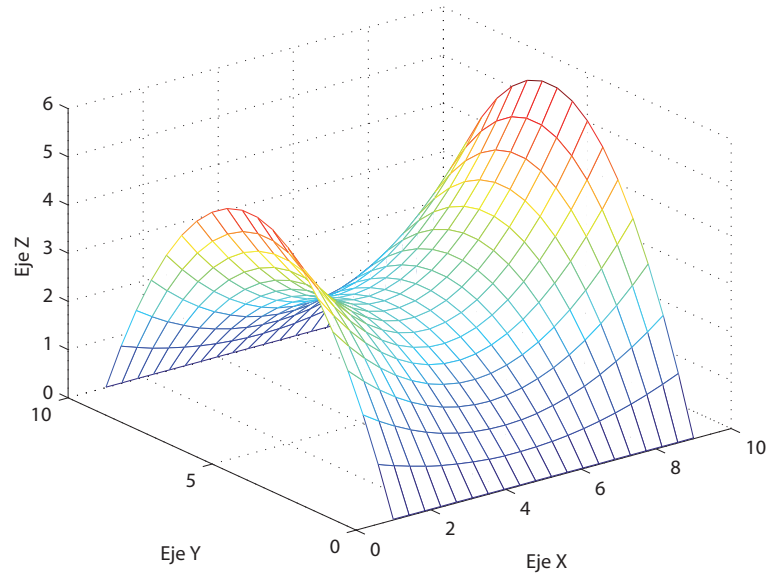


Figura E.16: Superficie de barrido deformada

$$A(v) = \begin{bmatrix} \cos(\varphi(v)) & 0 & \text{sen}(\varphi(v)) \\ 0 & 1 & 0 \\ -\text{sen}(\varphi(v)) & 0 & \cos(\varphi(v)) \end{bmatrix}$$

$\varphi(v)$ Es el ángulo de rotación en función de v

Es decir que el ángulo varía entre 0 y α de modo que la curva va rotando sobre el eje y al mismo tiempo que rota hasta alcanzar el valor del ángulo α en $v=2$

$$S(u, v) = \begin{bmatrix} c(v) & 0 & s(v) \\ 0 & 1 & 0 \\ -s(v) & 0 & c(v) \end{bmatrix} \begin{bmatrix} f(u) \\ 0 \\ g(u) \end{bmatrix} + \begin{bmatrix} 0 \\ v \\ 0 \end{bmatrix} \quad -2 \leq v \leq 2$$

donde

$$c(v) = \cos\left(\frac{v+2}{4}\alpha\right)$$

$$s(v) = \text{sen}\left(\frac{v+2}{4}\alpha\right)$$

El segundo caso corresponde a la condición de que el vector normal al plano que contiene la curva $C(u)$ sea paralelo al vector tangente de la trayectoria $T(v)$.

Este caso será mas claro con un ejemplo:

El vector tangente y normalizado a la trayectoria es:

$$\bar{e}_1(v) = \frac{T'(v)}{\|T'(v)\|} \quad (\text{E.35})$$

Ahora se debe definir un vector perpendicular a este vector tangente al plano que contiene la curva, para ello:

$$\bar{e}_1(v)\bar{e}_3(v) = 0$$

$$\|\bar{e}_3(v)\| = 1$$

Entonces el producto vectorial da como resultado:

$$\bar{e}_2(v) = \bar{e}_1(v) \times \bar{e}_3(v) \quad (\text{E.36})$$

Esto da como resultado un sistema ortogonal para cada punto de la trayectoria $T(v)$, por lo tanto estos vectores imponen las nuevas condiciones de interpolación de la curvas y así se obtiene la matriz de transformación afín que se necesita para generar la superficie producto del barrido de la curva sobre la trayectoria, haciendo que el plano que contiene la curva sea siempre perpendicular al vector tangente de la trayectoria.

Ahora, si la curva $C(u)$ esta definida sobre el plano XZ el vector normal de ese plano hacia afuera esta dado por: $-e_2 = [0 \ -1 \ 0]$, por su parte el vector vertical del plano esta dado por $e_3 = [0 \ 0 \ 1]$

Tomando las curvas

$$C(u) = \begin{cases} x = u \\ y = 0 \\ z = u^3 \end{cases} \quad T(u) = \begin{cases} x = v \\ y = y^2 \\ z = 0 \end{cases}$$

Para lo cual el vector tangente a la trayectoria esta dado por:

$$\bar{e}_1(v) = \frac{T'(v)}{\|T'(v)\|} = \frac{[1, 2v, 0]}{\sqrt{1 + 4v^2}}$$

Cuadro E.1: Clasificación de las superficies 3D

$$\bar{e}_2(v) = \bar{e}_3(v) \times \bar{e}_1(v) = \frac{1}{\sqrt{1+4v^2}} [1, 2v, 0] \times [0 \ 0 \ 1]$$

$$A(v) = [\bar{e}_2(v) \mid -\bar{e}_1(v) \mid \bar{e}_3(v)] = \begin{bmatrix} \frac{-2v}{\sqrt{1+4v^2}} & \frac{-1}{\sqrt{1+4v^2}} & 0 \\ \frac{1}{\sqrt{1+4v^2}} & \frac{-2v}{\sqrt{1+4v^2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S(u, v) = \begin{bmatrix} \frac{-2v}{\sqrt{1+4v^2}} & \frac{-1}{\sqrt{1+4v^2}} & 0 \\ \frac{1}{\sqrt{1+4v^2}} & \frac{-2v}{\sqrt{1+4v^2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ 0 \\ u^3 \end{bmatrix} + \begin{bmatrix} v \\ v^2 \\ 0 \end{bmatrix}$$

Generalmente la curva $C(u)$ esta contenida en el plano XZ y la trayectoria en el plano YZ, que es el caso mas común y el que se clarifico anteriormente, para la cual la superficie resultante será:

$$si \quad A(u) = \begin{bmatrix} -\bar{e}_{1y}(v) & -\bar{e}_{1x}(v) & 0 \\ \bar{e}_{1x}(v) & -\bar{e}_{1y}(v) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S(u, v) = \begin{bmatrix} -e_y(v)f(u) + T_x(v) \\ e_x(v)f(u) + T_y(v) \\ g(u) + T_z(v) \end{bmatrix}$$

En caso de que la curva trayectoria $T(v)$ no sea plana, es decir que no este contenida en un plano perpendicular al de la curva $C(u)$, se debe tomar el vector $\bar{e}_3(v)$ como un vector binomial a $T(v)$ normalizado.

$$\bar{e}_3(v) = \bar{e}_1(v) = \frac{T'(v) \times T'(v)}{\|T'(v) \times T'(v)\|}$$

SUPERFICIES 3D	
<p>Superficies Regladas</p> <p>Interpolación de 2 puntos para $u = \bar{u}$ situados sobre 2 curvas C_1, C_2</p>	<ul style="list-style-type: none"> ■ Isoparamétricas: Representación simultánea con $v = constante$ y $u = constante$ ■ Bilineal: Solo se interpolan 4 puntos y $n = m = 1$ ■ Cilíndrica: Se desplaza $C(v)$ en el espacio siguiendo una trayectoria recta
<p>Superficies de Barrido</p> <p>Desplazar una curva $C(u)$ (generatriz) en el espacio siguiendo una trayectoria $T(v)$ al mismo tiempo que se deforma.</p>	<ul style="list-style-type: none"> ■ Caso I: Rotación siguiendo una trayectoria <ul style="list-style-type: none"> · $\left. \begin{array}{l} Revolución \\ Swing \end{array} \right\}$ ■ Caso II: Escalado siguiendo una trayectoria. ■ Caso III: El plano que contiene $C(u)$ rota para permanecer siempre perpendicular a la tangente de $T(v)$.
<p>Matriz afín de transformación</p>	<p>Sus elementos transforman de manera afín $C(v)$ de modo que satisfacen la forma deseada en la superficie resultante</p>
<p>Nota: $C(u)$ y $T(v)$ deben estar contenidas en planos perpendiculares</p>	
<p>Superficies paramétricas</p>	<p>Una vez definidas las bases que componen la superficie bajo su forma funcional se debe pasar a la representación paramétrica</p>

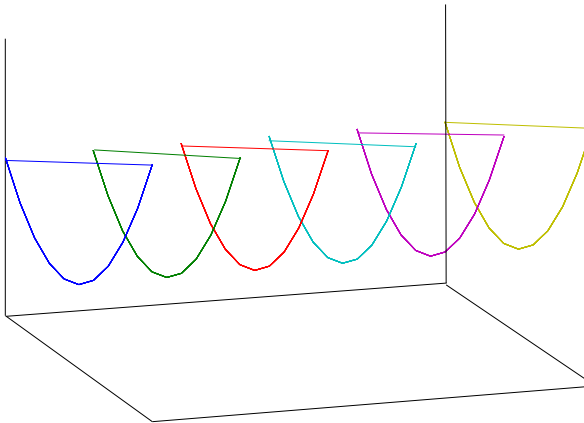


Figura E.17: Curvas planas en un entorno tridimensional

E.4. Lofting

Las superficies regladas utilizan una técnica de generación llamada interpolación de dos curvas dadas. La generalización de esta idea da lugar a dos métodos de generación de superficies que globalmente se conocen por *recubrimiento* o *rellenado* y que en la literatura inglesa se denomina *lofting*, termino que no debería traducirse. El *Loft* es la plataforma sobre la que antiguamente se construían los barcos a partir de secciones que posteriormente se recubrían para obtener la forma del casco.

La técnica se basa en posicionar curvas planas en un entorno tridimensional para a continuación recubrirlas generando así una superficie, y recibió el nombre de *lofting*.

En modelamiento geométrico, a la esta técnica se le conoce como *v-lofting* indicando con ello que los datos de entrada son un conjunto de curvas isoparamétricas, todas definidas según el mismo parámetro u y variando en el mismo intervalo $u_0 \leq u \leq u_1$. Se trata entonces de interpolar estas curvas en la variable v para generar una superficie que contenga a las curvas dadas como secciones. A estos métodos también se les conoce como *interpolación transinfinita*, ya que resuelven el problema de interpolación en un número infinito de puntos.

E.4.1. V-Lofting

Sean $C_0(u), C_1(u), \dots, C_n(u)$ $n + 1$ curvas definidas sobre el mismo intervalo $[u_0, u_1]$. Se entiende por v-lofting la construcción de una superficie $S(u, v)$ tal que para un soporte

de interpolación

$$v_0 < v_1 < v_n$$

se verifique:

$$S(u, v_i) = C_i(u) \quad i = 0, 1, 2, \dots, n$$

Sean $L_0(v), L_1(v), \dots, L_n(v)$ un conjunto de $n+1$ funciones cardinales respecto al soporte de interpolación, es decir:

$$L_k(v_i) = 0 \text{ si } i \neq k$$

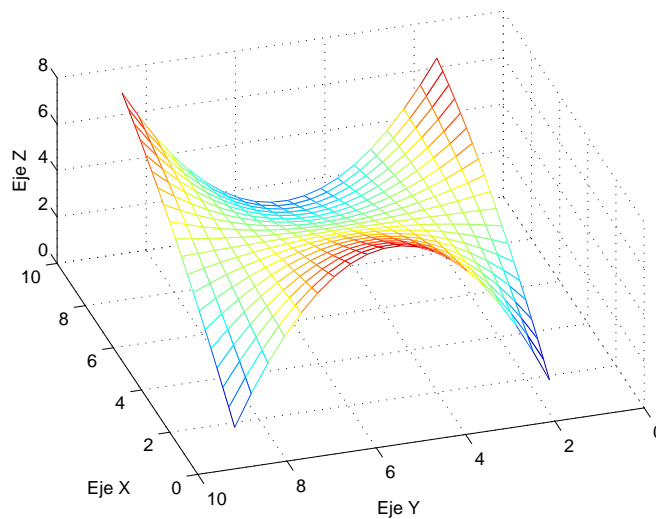
$$L_k(v_k) = 1$$

entonces:

$$S(u, v) = \sum_{k=0}^n C_k(u) L_k(v)$$

verifica:

$$S(u, v_i) = \sum_{k=0}^n C_k(u) L_k(v_i) = C_i(u)$$



con lo que $S(u, v)$ interpola a las curvas $C_i(u)$ y las contiene como curvas isoparamétricas para $v = v_0, \dots, v_n$.

Las funciones $L_k(x)$ pueden ser polinomios de Lagrange, lo que permite que las curvas $C_i(u)$ aparezcan explícitamente en la formulación de la superficie. Pero el problema de

interpolación puede resolverse utilizando cualquier otra base de polinomios.

Por ejemplo, si $B_0(v), B_1(v), \dots, B_n(v)$ constituyen una base arbitraria de $n + 1$ polinomios, entonces $S(u, v)$ puede escribirse:

$$S(u, v) = \sum_{k=0}^n a_k B_k(v)$$

Imponiendo las condiciones de interpolación:

$$S(u, v_i) = \sum_{k=0}^n a_k B_k(v_i) = C_i(u) \quad i = 0, 1, \dots, n$$

obtenemos un sistema de $n + 1$ ecuaciones con $n + 1$ incógnitas, que son las curvas $a_k(u)$:

$$\begin{bmatrix} B_0(v_0) & B_1(v_0) & \dots & B_n(v_0) \\ B_0(v_1) & B_1(v_1) & \dots & B_n(v_1) \\ \dots & \dots & \dots & \dots \\ B_0(v_n) & B_1(v_n) & \dots & B_n(v_n) \end{bmatrix} \begin{bmatrix} a_0(u) \\ a_1(u) \\ \dots \\ a_n(u) \end{bmatrix} = \begin{bmatrix} C_0(u) \\ C_1(u) \\ \dots \\ C_n(u) \end{bmatrix}$$

Sean $C_0(u), C_1(u), C_2(u)$ tres curvas a interpolar definidas por:

$$\begin{aligned} C_0(u) &\equiv (\cos(u), 0, \operatorname{sen}(u)) \\ C_1(u) &\equiv \left(\frac{\sqrt{2}}{2} \cos(u), -\frac{\sqrt{2}}{2} \cos(u) + 3, \operatorname{sen}(u) \right) 0 \leq u \leq 2\pi \\ C_2(u) &\equiv (3, -\cos(u) + 3, \operatorname{sen}(u)) \end{aligned}$$

Como soporte de interpolación en la variable v se puede elegir: $S = \{0, 1, 2\}$.

Los polinomios de Lagrange asociados al soporte son:

$$\begin{aligned} L_0(v) &= \frac{1}{2}v^2 - \frac{3}{2}v + 1 \\ L_1(v) &= -v^2 + 2v \\ L_2(v) &= \frac{1}{2}v^2 - \frac{1}{2}v \end{aligned}$$

y la superficie v-lifting:

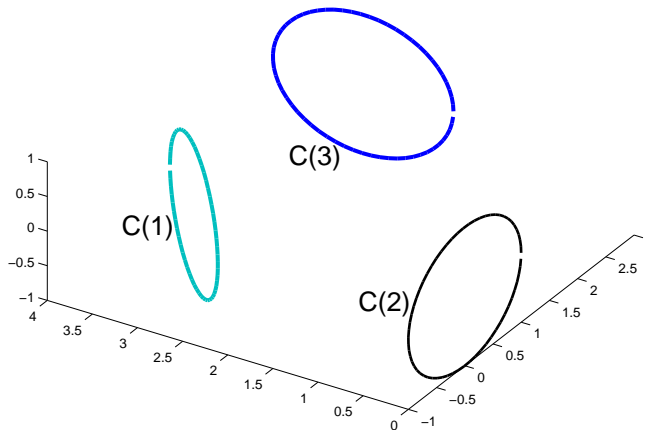


Figura E.18: Curvas a interpolar

$$\begin{aligned}
 S(u, v) &= C_0(u)L_0(v) + C_1(u)L_1(v) + C_2(u)L_2(v) \\
 &= \left(\frac{1}{2}v^2 - \frac{3}{2}v + 1\right) (\cos(u), 0, \text{sen}(u)) \\
 &\quad + (-v^2 + 2v) \left(\frac{\sqrt{2}}{2}\cos(u), -\frac{\sqrt{2}}{2}\cos(u) + 3, \text{sen}(u)\right) \\
 &\quad + \left(\frac{1}{2}v^2 - \frac{1}{2}v\right) (3, -\cos(u) + 3, \text{sen}(u))
 \end{aligned}$$

También se puede utilizar la base de potencias para generar la superficie. Por ejemplo:

$$S(u, v) = a_0(u) + a_1(u)v + a_2(u)v^2$$

y las condiciones de interpolación son:

$$\begin{aligned}
 v = 0 & \qquad \qquad \qquad a_0(u) = C_0(u) \\
 v = 1 & \quad a_0(u) + a_1(u) + a_2(u) = C_1(u) \\
 v = 2 & \quad a_0(u) + 2a_1(u) + 4a_2(u) = C_2(u)
 \end{aligned}$$

El sistema resultante es:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} a_0(u) \\ a_1(u) \\ a_2(u) \end{bmatrix} = \begin{bmatrix} C_0(u) \\ C_1(u) \\ C_2(u) \end{bmatrix}$$

Invirtiendo la matriz:

$$\begin{bmatrix} a_0(u) \\ a_1(u) \\ a_2(u) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 0 & 0 \\ -3 & 4 & -1 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} C_0(u) \\ C_1(u) \\ C_2(u) \end{bmatrix}$$

Obteniendo:

$$\begin{aligned} a_0(u) &\equiv (\cos(u), 0, \operatorname{sen}(u)) \\ a_1(u) &\equiv \frac{1}{2} \left((2\sqrt{2} - 3) \cos(u) - 3, (1 - 2\sqrt{2}) \cos(u) + 9, 0 \right) \\ a_2(u) &\equiv \frac{1}{2} \left((1 - 2\sqrt{2}) \cos(u) + 3, (\sqrt{2} - 1) \cos(u) - 3, 0 \right) \end{aligned}$$

Al obtener la superficie se observa que es exactamente la misma que la obtenida a través de los polinomios de Lagrange.

$$\begin{aligned} S(u, v) &= (\cos(u), 0, \operatorname{sen}(u)) \\ &+ v \left(\frac{2\sqrt{2} - 3}{2} \cos(u) - \frac{3}{2}, -\frac{1 - 2\sqrt{2}}{2} \cos(u) + \frac{9}{2}, 0 \right) \\ &+ v^2 \left(\left(\frac{1 - \sqrt{2}}{2} \right) \cos(u) + \frac{3}{2}, \left(\frac{\sqrt{2} - 1}{2} \right) \cos(u) - \frac{3}{2}, 0 \right) \end{aligned}$$

Un tercer procedimiento para construir una superficie v-lofting (el más empleado en la práctica) es el llamado *interpolación con superficie producto tensorial*.

Se supone ahora que las curvas $C_0(u), C_1(u), \dots, C_n(u)$ se expresan como combinación lineal de una base de $m + 1$ funciones $R_0(u), R_1(u), \dots, R_m(u)$:

$$C_i(u) = \sum_{j=0}^m a_{ij} R_j(u) \quad i = 0, 1, \dots, n$$

donde a_{ij} son coeficientes 3D, $a_{ij} = (x_{ij}, y_{ij}, z_{ij})$

Se agrupan los a_{ij} en una estructura rectangular:

$$\begin{array}{rcccccc}
 C_0(u) & \rightarrow & a_{00} & a_{01} & a_{02} & \dots & a_{0m} \\
 C_1(u) & \rightarrow & a_{10} & a_{11} & a_{12} & \dots & a_{1m} \\
 \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\
 C_n(u) & \rightarrow & a_{n0} & a_{n1} & a_{n2} & \dots & a_{nm}
 \end{array}$$

y ahora el método consiste en realizar interpolaciones por columnas para hallar las $m+1$ curvas $D_k(v)$ en la variable v , cada una interpolando una columna de coeficientes, pero sobre el mismo soporte v_0, v_1, \dots, v_n :

$$\begin{array}{rcccccc}
 D_0(v) & \textit{interpola} & a_{00}, a_{10}, a_{n0} & : & D_0(v_j) = a_{i0} & i = 0, 1, \dots, n \\
 D_1(v) & \textit{interpola} & a_{01}, a_{11}, a_{n1} & : & D_1(v_j) = a_{i1} & i = 0, 1, \dots, n \\
 \dots & \dots & \dots & & \dots & \dots \\
 D_n(v) & \textit{interpola} & a_{0n}, a_{1n}, a_{nn} & : & D_n(v_j) = a_{in} & i = 0, 1, \dots, n
 \end{array}$$

La superficie v-loft es entonces:

$$S(u, v) = \sum_{j=0}^m D_j(v) R_j(u)$$

Efectivamente:

$$S(u, v) = \sum_{j=0}^m D_j(v) R_j(u) = \sum_{j=0}^m a_{ij} R_j(u) = C_i(u) \quad i = 0, \dots, n$$

o en función de la base:

$$S(u, v) = \sum_{k=0}^n \sum_{j=0}^m b_{kj} B_k(v) R_j(u)$$

Considerando nuevamente el ejemplo anterior:

$$\begin{aligned}
C_0(u) &\equiv (\cos(u), 0, \operatorname{sen}(u)) \\
C_1(u) &\equiv \left(\frac{\sqrt{2}}{2} \cos(u), -\frac{\sqrt{2}}{2} \cos(u) + 3, \operatorname{sen}(u) \right) \\
C_2(u) &\equiv (3, -\cos(u) + 3, \operatorname{sen}(u))
\end{aligned}$$

se elige como base en la variable u , la constituida por las funciones:

$$\{R_0(u), R_1(u), R_2(u)\} \equiv \{1, \operatorname{sen}(u), \cos(u)\}$$

de forma que las curvas $C_i(u)$ se expresan:

$$\begin{aligned}
C_0(u) &\equiv (0, 0, 0) R_0(u) + (0, 0, 1) R_1(u) + (1, 0, 0) R_2(u) \\
C_1(u) &\equiv (0, 3, 0) R_0(u) + (0, 0, 1) R_1(u) + \left(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, 0 \right) R_2(u) \\
C_2(u) &\equiv (3, 3, 0) R_0(u) + (0, 0, 1) R_1(u) + (0, -1, 0) R_2(u)
\end{aligned}$$

Se realizan tres interpolaciones:

$$\begin{aligned}
D_0(v) \text{ interpolando } & (0, 0, 0), (0, 0, 1), (1, 0, 0) \\
D_1(v) \text{ interpolando } & (0, 3, 0), (0, 0, 1), \left(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, 0 \right) \\
D_n(v) \text{ interpolando } & (3, 3, 0), (0, 0, 1), (0, -1, 0)
\end{aligned}$$

con esto se obtiene:

$$\begin{aligned}
D_0(v) &= \left(\frac{3}{2}, -\frac{3}{2}, 0 \right) v^2 + \left(-\frac{3}{2}, \frac{9}{2}, 0 \right) v \\
D_1(v) &= (0, 0, 1) \\
D_2(v) &= \left(\frac{1 - \sqrt{2}}{2}, \frac{\sqrt{2} - 1}{2}, 0 \right) v^2 + \left(\frac{2\sqrt{2} - 3}{2}, \frac{1 - 2\sqrt{2}}{2}, 0 \right) v + (1, 0, 0)
\end{aligned}$$

expresadas en la base de potencias. Entonces, la superficie lofting en función de la base producto tensorial de las bases definidas en las variables u y v , es:

Algoritmo: Superficie producto tensorial

```

clear all
close all
clc n=3;
axis([-1,11,-1,10])
er=[-1 10];
hold on
for i=1:n
    ay=ginput(1);
    plot(ay(1),ay(2),'ro')
    Q(i,1)=ay(1);
    Q(i,2)=ay(2);
end
% creamos lineas a Ycte para saber por donde pasara Z
wl=[Q(1,1) Q(1,1)];
wc=[Q(2,1) Q(2,1)];
wr=[Q(3,1) Q(3,1)];
for j=1:n
    for i=1:n
        m(j,i)=Q(j,1)^(i-1);
    end
end
for i=1:n
    ord(i)=Q(i,2);
end
au=inv(m)*ord';
z=au';z=z(n:-1:1);a=z;
y=linspace(Q(1,1),Q(n,1),100);
x=linspace(1,9,10);
abs=[1 5 9];
% la distribución de los planos X
zo=polyval(z,y);
plot(y,zo)
ay=ginput(1);
% hasta aca creamos la 1ª curva y definimos
% los Y iguales para las otras curvas

```

```

for h=2:n
    close all
    axis([-1,11,-1,10])
    hold on
    plot(wl,er,'g')
    plot(wc,er,'g')
    plot(wr,er,'g')
    xlabel('click en los Z ==>')
    for i=1:n
        ay=ginput(1);
        plot(Q(i,1),ay(2),'ro')
        Q(3*(h-1)+i,1)=Q(i,1);
        Q(3*(h-1)+i,2)=ay(2);
    end
    for i=1:n
        ord(i)=Q(3*(h-1)+i,2);
    end
    au=inv(m)*ord';
    z=au';z=z(n:-1:1);
    a(h,:)=z; %ubico los coef en la siguiente fila
    zo=polyval(z,y);
    plot(y,zo)
    ay=ginput(1);
    % pausar el calculo para ver la curva
    end
    close all
    for j=1:n
        for i=1:n
            m(j,i)=abs(j)^(i-1);
        end
    end
    m=inv(m);
    %ya que es la misma base para todos
    for j=1:n
        ord=a(:,j);
    end

```

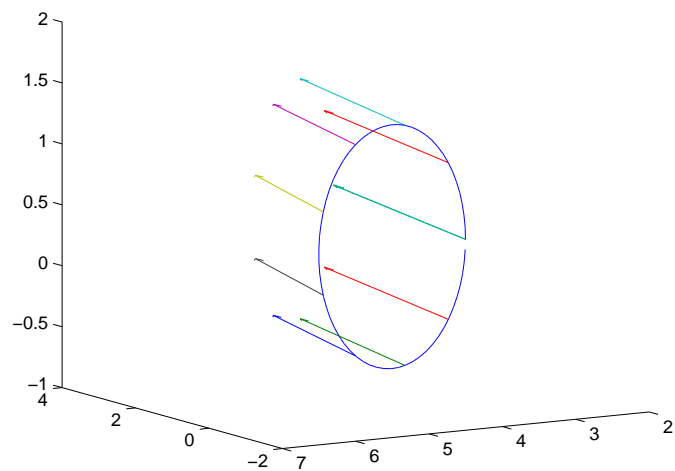
```

    % c/columna de coef's
    bi=m*ord;
    z=bi';z=z(n:-1:1);
    for i=1:n
        b(i,j)=z(i);
    end
end
[X,Y]=meshgrid(x,y);
Z=polyval(b(:,1),X).*(Y.^2)+polyval(b(:,2),X).*Y+polyval(b(:,3),X);
mesh(X,Y,Z)
xlabel('Eje X')
ylabel('Eje Y')
zlabel('Eje Z')
title('superficie')
axis equal

```

E.4.2. V-Lofting con especificación de derivadas.

En la figura se muestra una serie de problemas que se ponen de manifiesto en la superficie lofting calculada con un simple esquema interpolatorio. Aunque $S(u, v)$ efectivamente interpola a las curvas dadas, no lo hace en la forma esperada, pues las tangentes a las curvas isoparamétricas de $u = constante$ no son perpendiculares a los planos que contienen a $C_0(u)$ y $C_2(u)$, como se quiere.



Es posible especificar para cada nodo de interpolación v_i , no sólo la curva $C_i(u)$ a interpolar, sino las derivadas parciales $S(u, v)$ con respecto a v , es decir, un vector tangente a la superficie para $v = v_i$ y para cada u .

En la práctica las curvas se suponen planas y entonces estos vectores tangentes se tomarán siempre perpendiculares a los planos que contienen las curvas.

Ahora, siguiendo el método de Hermite, si se pretende interpolar $n + 1$ curvas $C_0(u), \dots, C_n(u)$ de forma que la superficie tenga derivadas parciales respecto a v en los nodos de interpolación $T_0(u), \dots, T_n(u)$:

$$S(u, v) = \sum_{k=0}^{2n+2} a_k(u) B_k(v)$$

Entonces:

$$S(u, v) = \sum_{k=0}^{2n+2} a_k(u) B_k(v_i) = C_i(u) \quad i = 0, \dots, n$$

$$\frac{\partial S(u, v)}{\partial v} \Big|_{v=v_i} = \sum_{k=0}^{2n+1} a_k(u) B'_k(v_i) = T_i(u) \quad i = 0, \dots, n$$

lo que constituye un sistema lineal de $2n + 2$ ecuaciones en las $2n + 2$ incógnitas $a_0(u), \dots, a_{2n+1}(u)$.

Ejemplo:

Suponga dadas las tres curvas $C_0(u), C_1(u), C_2(u)$ y vectores tangentes a lo largo de su contorno $T_0(u), T_1(u), T_2(u)$

Puesto que $n = 2$ utilizaremos una base de seis funciones en la variable v , por ejemplo, la base de potencias:

$$S(u, v) = a_0(u) + a_1(u)v + a_2(u)v^2 + a_3(u)v^3 + a_4(u)v^4 + a_5(u)v^5$$

Las condiciones de interpolación para el soporte $\mathcal{S} = \{0, 1, 2\}$:

$$S(u, 0) = a_0(u) = C_0(u)$$

$$S(u, 1) = a_0(u) + a_1(u) + a_2(u) + a_3(u) + a_4(u) + a_5(u) = C_1(u)$$

$$S(u, 2) = a_0(u) + 2a_1(u) + 4a_2(u) + 8a_3(u) + 16a_4(u) + 32a_5(u) = C_2(u)$$

Las derivadas parciales de $S(u, v)$ con respecto a v :

$$S'_v(u, v) = \frac{\partial S(u, v)}{\partial v} = a_1(u)v + 2a_2(u)v + 3a_3(u)v^2 + 4a_4(u)v^3 + 5a_5(u)v^4$$

y en los nodos del soporte:

$$S'_v(u, 0) = a_1(u) = T_0(u)$$

$$S'_v(u, 1) = a_1(u) + 2a_2(u) + 3a_3(u) + 4a_4(u) + 5a_5(u) = T_1(u)$$

$$S'_v(u, 2) = a_1(u) + 4a_2(u) + 12a_3(u) + 32a_4(u) + 80a_5(u) = T_2(u)$$

con lo cual tenemos un sistema de seis ecuaciones con seis incógnitas:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 & 32 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 4 & 12 & 32 & 80 \end{bmatrix} \begin{bmatrix} a_0(u) \\ a_1(u) \\ a_2(u) \\ a_3(u) \\ a_4(u) \\ a_5(u) \end{bmatrix} = \begin{bmatrix} C_0(u) \\ C_1(u) \\ C_2(u) \\ T_1(u) \\ T_2(u) \\ T_3(u) \end{bmatrix}$$

Los coeficientes $a_k(u)$ se obtienen invirtiendo la matriz:

$$\begin{bmatrix} a_0(u) \\ a_1(u) \\ a_2(u) \\ a_3(u) \\ a_4(u) \\ a_5(u) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ -23 & 16 & 7 & -12 & -16 & -2 \\ 33 & -16 & -17 & 13 & 32 & 5 \\ -17 & 4 & 13 & -6 & -20 & -4 \\ 3 & 0 & -3 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} C_0(u) \\ C_1(u) \\ C_2(u) \\ T_1(u) \\ T_2(u) \\ T_3(u) \end{bmatrix}$$

Si las curvas $C_i(u)$ son las del ejemplo anterior, pueden tomarse $T_i(u)$ perpendiculares

a los planos que las contienen:

$$\begin{aligned} T_0(u) &= r_0(u) [0, 1, 0] \\ T_1(u) &= r_1(u) \left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right] \\ T_2(u) &= r_2(u) [1, 0, 0] \end{aligned}$$

Para cada punto de la curva $C_i(u)$ se puede aún dar el módulo del vector tangente.

En la práctica, si las curvas $C_i(u)$ son planas y se expresan en una base

$$R_0(u), \dots, R_m(u)$$

verificando $\sum_{j=0}^n R_j(u) = 1$, entonces

$$C_i(u) = \sum_{j=0}^m a_{ij} R_j(u) \quad i = 0, 1, \dots, n$$

y los coeficientes $a_{ij} = (x_{ij}, y_{ij}, z_{ij})$ son para i fijo, y están en el mismo plano de definición que $C_i(u)$. Por tanto se puede especificar un módulo para los vectores de las derivadas parciales, asociado a cada coeficiente a_{ij} con $j = 0, \dots, m$.

Esta simplificación permite obtener la superficie lofting mediante un procedimiento donde para j fijo se interpolan los coeficientes $a_{0j}, a_{1j}, \dots, a_{nj}$ con las condiciones adicionales de que las tangentes sean las establecidas.

Como ejemplo, supóngase que las curvas $C_0(u), \dots, C_n(u)$ se expresan en la base de los polinomios de Lagrange, según un soporte u_0, \dots, u_n .

$$C_i(u) = \sum_{j=0}^m a_{ij} L_j(u) \quad i = 0, 1, \dots, n$$

Dado que $\sum_{j=0}^n L_j(u) = 1$ los coeficientes a_{ij} están en el mismo plano que $C_i(u)$ y son puntos por los que pasa $C_i(u)$. El método consiste en especificar vectores tangentes asociados a cada coeficiente, que tendrán la misma dirección y sentido que el vector normal al plano $C_i(u)$ pero con módulos en principio variables:

$$a_{ij} \rightarrow T_{ij} = r_{ij} N_i$$

Para obtener la superficie lofting basta entonces con realizar $m + 1$ interpolaciones de Hermite:

$$\begin{array}{rcl}
 D_0(v) & \text{tal que} & D_0(v_i) = a_{i0}, \quad D'_0(v_i) = r_{i0}N_i \quad i = 0, 1, \dots, n \\
 D_1(v) & \text{tal que} & D_1(v_i) = a_{i1}, \quad D'_1(v_i) = r_{i1}N_i \quad i = 0, 1, \dots, n \\
 \dots & \dots & \dots \\
 D_n(v) & \text{tal que} & D_n(v_i) = a_{in}, \quad D'_n(v_i) = r_{in}N_i \quad i = 0, 1, \dots, n
 \end{array}$$

E.4.3. Posicionamiento de curvas de sección

Con frecuencia las curvas $C_i(u)$ se modelan en un entorno 2D y posteriormente se posicionan en el lugar y con la orientación correctas. Esta tarea de colocación de cada curva no es sencillo ni intuitivo.

Lo que se hace entonces es modelar una de estas secciones longitudinales para que sirva de ayuda en el posicionamiento de las curvas $C_i(u)$.

Una curva $T(v)$ en la dirección v se denomina trayectoria (o espina dorsal). De este modo dada la curva $T(v)$ y un conjunto de $n + 1$ valores del parámetro $\bar{v}_0 < \bar{v}_1 < \dots < \bar{v}_n$, cada $C_i(u)$ se engancha a la trayectoria en el correspondiente punto $T(\bar{v}_i)$ y lo hace en un valor concreto \bar{u}_i del intervalo de definición de su parámetro.

Una vez que la trayectoria $T(v)$ se ha utilizado para el posicionamiento y orientación de los $C_i(u)$, es desechada y no interviene en el proceso de construcción de la superficie.

La figura muestra un ejemplo en el que las curvas $C_i(u)$ son todas circunferencias de igual radio que se han posicionado a lo largo de una hélice:

$$T(v) = \begin{cases} x = \cos(v) \\ y = \text{sen}(v) \\ z = 5v \end{cases}$$

E.4.4. Superficies translacionales

Las superficies translacionales están definidas a partir de sólo dos curvas $C_1(u)$ y $C_2(v)$, la primera definida en la variable u y la segunda en la variable v , y ambas se intersectan en un extremo $P_0 = C_1(0) = C_2(0)$.

La expresión para esta superficie es:

$$S(u, v) = C_1(u) + C_2(v) - P_0$$

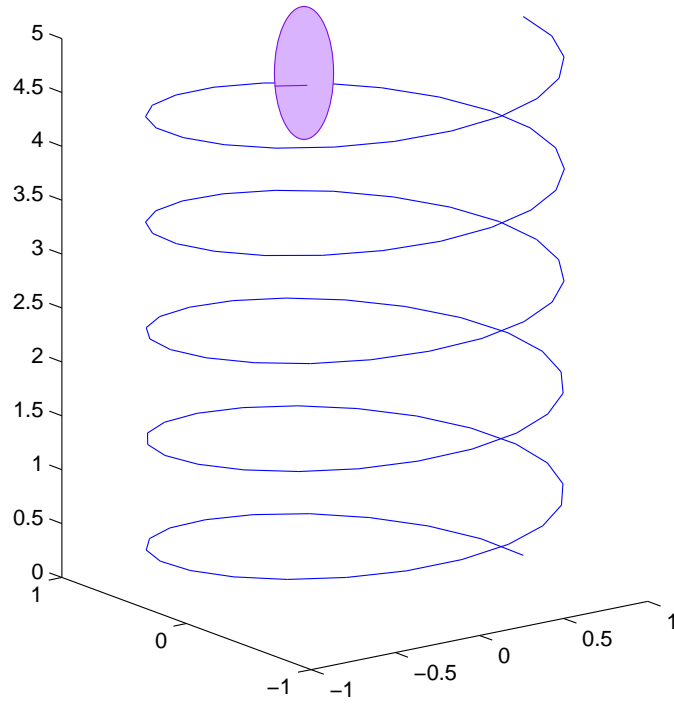


Figura E.19: Curvas de sección

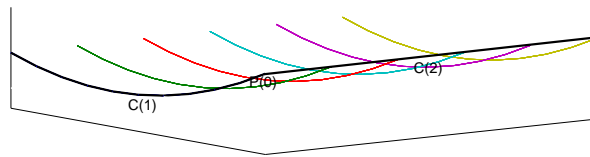
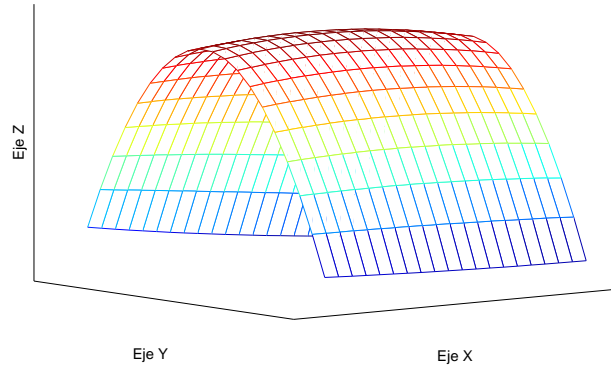


Figura E.20: Las dos curvas comparten un vértice

E.4.5. Superficies de Coons

Se trata de una superficie que interpola dos pares de curvas definidas respectivamente en las variables u y v .



Para obtener la superficie se realizan dos interpolaciones lineales independientes entre curvas opuestas, obteniendo dos superficies regladas:

$$R_1(u, v) = (1 - v)C_1(u) + vC_2(u) \quad 0 \leq u, v \leq 1$$

$$R_2(u, v) = (1 - u)C_3(u) + uC_4(u) \quad 0 \leq u, v \leq 1$$

Los puntos de intersección de la curva son:

$$P_{00} = C_1(0) = C_3(0)$$

$$P_{01} = C_2(0) = C_3(0)$$

$$P_{10} = C_1(1) = C_4(1)$$

$$P_{11} = C_4(1) = C_2(1)$$

Sumando:

$$\begin{aligned} R(u, v) &= R_1(u, v) + R_2(u, v) \\ &= (1 - v)C_1(u) + vC_2(u) + (1 - u)C_3(u) + uC_4(u) \end{aligned}$$

Se obtiene:

$$R(0, v) = (1 - v)P_{00} + vP_{01} + C_3(v)$$

$$R(1, v) = (1 - v)P_{10} + vP_{11} + C_4(v)$$

$$R(u, 0) = C_1(u) + (1 - u)P_{00} + uP_{10}$$

$$R(u, 1) = C_2(u) + (1 - u)P_{01} + uP_{11}$$

Si a $R(u, v)$ se le sustrae la superficie bilineal interpolando los cuatro puntos:

$$B(u, v) = (1 - u)(1 - v)P_{00} + (1 - u)vP_{01} + u(1 - v)P_{10} + uvP_{11}$$

se obtiene:

$$\begin{aligned} S(u, v) &= R_1(u, v) + R_2(u, v) - B(u, v) \\ &= (1 - v)C_1(u) + vC_2(u) + (1 - u)C_3(u) + uC_4(u) \\ &\quad - [(1 - u)(1 - v)P_{00} + (1 - u)vP_{01} + u(1 - v)P_{10} + uvP_{11}] \end{aligned}$$

Se puede comprobar que:

$$S(0, v) = C_3(v)$$

$$S(1, v) = C_4(v)$$

$$S(u, 0) = C_1(v)$$

$$S(u, 1) = C_2(v)$$

La figura muestra una superficie Coons interpolando las curvas:

$$C_1(u) = (2, u, -5u^2)$$

$$C_2(u) = (u, 2, -5u^2)$$

$$C_3(u) = (6v^2, -9v + 2, 6v^2 - 3v - 1, -1)$$

$$C_4(u) = (2v^2 - 3v + 2, 2v^2 - v + 2, 6v^2 - 3v + 1, -1)$$

E.4.6. Superficies de Gordon. UV-Lofting

Sean $C_0(u), C_1(u), \dots, C_n(u)$ las curvas en la dirección u y $D_0(u), D_1(u), \dots, D_m(u)$ las curvas en la dirección v , de forma que ambos conjuntos intersectan en los puntos:

$$P_{ij} = C_i(u_j) = D_j(v_i) \quad i = 0, \dots, n \quad j = 0, \dots, m$$

que corresponden a valores de los parámetros u, v en los soportes:

$$\mathcal{S}_u = \{u_0, u_1, \dots, u_m\}, \quad \mathcal{S}_v = \{v_0, v_1, \dots, v_n\}$$

Sean entonces $R_1(u, v)$, $R_2(u, v)$ superficies de v-lofting y u-lofting interpolando respectivamente (y de forma independiente una respecto a la otra) las curvas $C_i(u)$ y $D_j(v)$. Es decir,

$$\begin{aligned} R_1(u, v) &= C_i(u) i = 0, \dots, n \\ R_2(u, v) &= D_j(v) j = 0, \dots, m \end{aligned}$$

Los esquemas interpolatorios seguidos en ambas direcciones son similares, basados en funciones cardinales $L_0(v), \dots, L_n(v)$ para v-lofting y $M_0(u), \dots, M_m(u)$ para u-lofting, verificando:

$$\begin{aligned} L_k(v_i) &= \delta_{ki} & k, i &= 0, \dots, n \\ M_h(u_j) &= \delta_{hj} & h, j &= 0, \dots, m \end{aligned}$$

donde δ_{pq} es la *delta de Kronecker*:

$$\delta_{pq} = \begin{cases} 0 & \text{si } p \neq q \\ 1 & \text{si } p = q \end{cases}$$

Ahora, las superficies intermedias $R_1(u, v)$ y $R_2(u, v)$ pueden escribirse:

$$R_1(u, v) = \sum_{k=0}^n C_k(u) L_k(v)$$

$$R_2(u, v) = \sum_{h=0}^m D_h(v) M_h(u)$$

Sea ahora $B(u, v)$ la superficie interpolando la malla de puntos de intersección P_{ij} :

$$B(u_i, v_j) = P_{ij} \quad i = 0, \dots, n \quad j = 0, \dots, m$$

de esta manera $B(u, v)$ está construida como producto tensorial de las bases $L_k(u)$ y $M_h(v)$:

$$B(u, v) = \sum_{k=0}^n \sum_{h=0}^m P_{kh} L_k(u) M_h(v)$$

Finalmente, la superficie de Gordon o uv-lofting se construye de manera análoga a la de Coons:

$$S(u, v) = R_1(u, v) + R_2(u, v) - B(u, v)$$

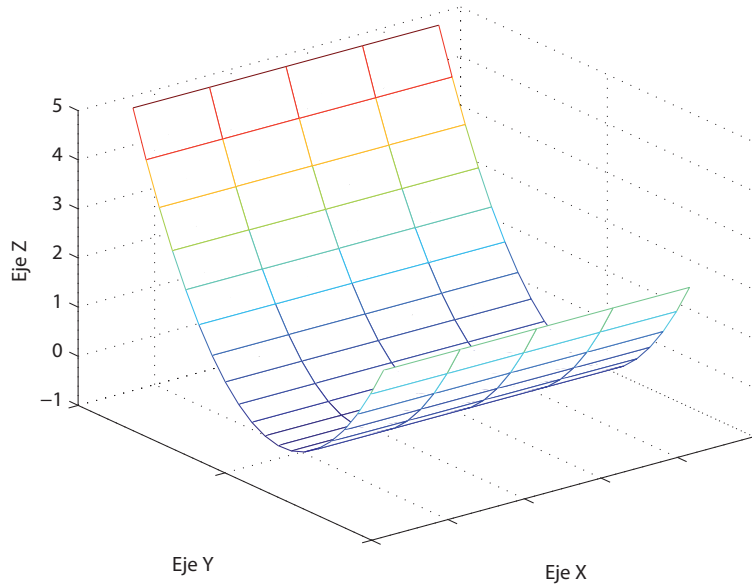
Ejemplo:

Considere la malla de curvas:

$$\begin{cases} C_0(u) = (1, u, 0) \\ C_1(u) = \left(0, u, \frac{u^4+79}{80}\right) \\ C_2(u) = (-1, u, 0) \end{cases} \quad -3 \leq u \leq 3$$

$$\begin{cases} D_0(v) = (v, -3, 2 - 2v^4) \\ D_1(v) = (v, -1, 1 - v^4) \\ D_2(v) = (v, 1, 1 - v^4) \\ D_3(v) = (v, 3, 2 - 2v^4) \end{cases} \quad -1 \leq v \leq 1$$

que corresponde a $n = 2$ y $m = 3$.



Primero se construye la superficie de interpolación de las curvas $C_0(u)$, $C_1(u)$, $C_2(u)$ respecto a la variable v .

Tomando un soporte uniforme de tres puntos:

$$\mathcal{S}_v = \{-1, 0, 1\}$$

Los polinomios de Lagrange son:

$$L_0(v) = \frac{v(v-1)}{2}, \quad L_1(v) = 1 - v^2, \quad L_2(v) = \frac{v(v+1)}{2};$$

La superficie intermedia $R_1(u, v)$ es:

$$R_1(u, v) = \left(-v, u, \frac{(1-v^2)(u^4 + 79)}{80} \right)$$

La segunda superficie es la de interpolación en la variable u de las curvas $D_0(v)$, $D_1(v)$, $D_2(v)$ y $D_3(v)$ para el soporte uniforme:

$$\mathcal{S}_u = \{-3, -1, 1, 3\}$$

Los polinomios de Lagrange son:

$$M_0(u) = \frac{(u^2 - 1)(u - 3)}{-48}, \quad M_1(u) = \frac{(u - 1)(u^2 - 9)}{16},$$

$$M_2(v) = \frac{(u + 1)(u^2 - 9)}{-16}, \quad M_3(v) = \frac{(u^2 - 1)(u + 3)}{48};$$

con lo cual $R_2(u, v)$ es:

$$R_2(u, v) = \left(v, u, \frac{(u^2 + 7)(1 - v^4)}{8} \right)$$

Ahora, se necesitan las superficies producto tensorial que interpolan los doce puntos de intersección entre las curvas:

$$\begin{aligned} &(1, -3, 0) \quad (1, -1, 0) \quad (1, 1, 1) \quad (1, 3, 0) \\ &(0, -3, 2) \quad (0, -1, 1) \quad (0, 1, 1) \quad (0, 3, 2) \\ &(-1, -3, 0) \quad (-1, -1, 0) \quad (-1, 1, 0) \quad (-1, 3, 0) \end{aligned}$$

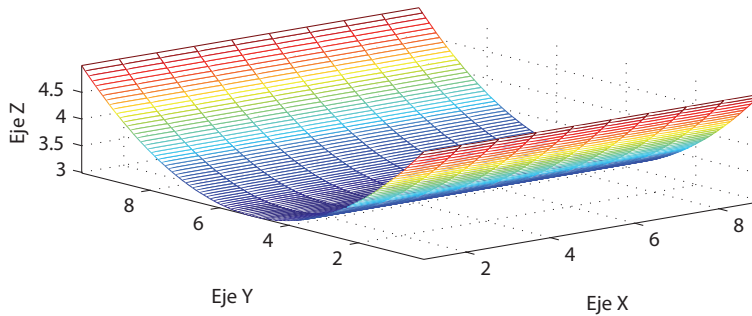
En este paso se utiliza el producto tensorial de los polinomios de Lagrange en u y v se tendrá:

$$B(u, v) = \left(-v, u, \frac{(u^2 + 7)(1 - v^2)}{8} \right)$$

La superficie uv-lofting de las curvas consideradas es:

$$S(u, v) = \left(v, u, \frac{(1 - v^2)(u^4 + 10u^2v^2 + 70v^2 + 79)}{80} \right)$$

la cual interpola las curvas dadas.



Apéndice F

Superficies rectangulares

Uno de los hechos más notables del modelado de superficies es que se pueden aplicar los modelos de interpolación y aproximación de curvas de una manera casi directa. En vez de contar con una secuencia de puntos de control, se tiene un arreglo bidimensional o matriz. Los modelos de interpolación y aproximación que se vieron en el capítulo anterior (exceptuando Lagrange) producen una base funcional parametrizada que indica la importancia de cada punto de control en el resultado final de la curva. Por lo tanto, es matemáticamente sencillo formular la existencia de dos familias de bases, cada una con su parámetro (por fila y por columna). El producto cartesiano de dichas bases produce una base biparamétrica, la cual, en función del valor de ambos parámetros, determina la importancia de cada uno en el control del arreglo bidimensional de la superficie resultante.

Una superficie paramétrica producto tensorial fue definida en el capítulo anterior como una combinación lineal de funciones que son el producto tensorial de dos bases univariadas:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} B_i(u) C_j(v) \quad (\text{F.1})$$

Donde P_{ij} son puntos distribuidos según una estructura rectangular teniendo mayor o menor significado geométrico dependiendo de las bases seleccionadas para su representación.

F.1. Superficies de Bézier simples

$$B_i(u) = B_i^n(u) = \binom{n}{i} (1-u)^{n-i} u^i$$

$$C_j(v) = C_j^m(v) = \binom{m}{j} (1-v)^{m-j} v^j$$

De manera análoga al caso de una variable estas bases son los polinomios de Bernstein definidos en $[0, 1]$ que combinados por medio del producto tensorial, se denominan *superficies de Bézier* o *parches de Bézier*.

De manera general una superficie de Bézier se expresará:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} B_i^n(u) B_j^m(v) \quad (\text{F.2})$$

donde los puntos P_{ij} son los ya conocidos puntos de control de la superficie, que constituyen una malla en 3D, denominada *malla de control*, relacionados con la superficie de manera similar a como se hacía con las curvas de Bézier.

A partir de las propiedades de los polinomios de Bernstein se puede inferir de las superficies de Bézier:

- $B_0^n(0) = 0 \quad i = 1, \dots, n \quad B_0^n(0) = 1$
 $B_i^n(1) = 0 \quad i = 0, \dots, n-1 \text{ y } B_n^n(1) = 1$
- $B_i^n(u) \geq 0 \quad \forall u \in [0, 1]$
- $\sum_{i=0}^n B_i^n(u) = 1$
- $[B_k^n(u)]' = n[B_k^{n-1}(u) - B_k^n(u)]$
- $B_i^n(u) = (1-u)B_i^{n-1}(u) + uB_{i-1}^{n-1}(u)$

F.1.1. Propiedades

1. Interpolación de los vértices de la malla de control

- $S(0, 0) = P_{00}$

- $S(0, 1) = P_{0m}$
- $S(1, 0) = P_{n0}$
- $S(1, 1) = P_{nm}$

2. Derivadas Parciales.

$$\frac{\partial S(u, v)}{\partial(u)} = \sum_{i=0}^n \sum_{j=0}^m P_{ij} \left(\frac{dB_i^n(u)}{du} B_j^m(v) \right)$$

Siendo de especial interés los valores que tiene estas derivadas parciales en los 4 bordes de la superficie.

$$\frac{\partial S(0, 0)}{\partial(u)} = n(P_{10} - P_{00})$$

$$\frac{\partial S(1, 0)}{\partial(u)} = n(P_{n0} - P_{n-1,0})$$

3. Restricción de la envolvente convexa

Cualquier punto de la superficie es una combinación lineal convexa de los puntos P_{ij}

4. Invarianza Afín

Para transformar una superficies de Bézier, ya sea que se desee realizar una rotación, translación o escalado, solo se deben transformar los puntos de control, y construir la respectiva base que le representa.

5. Curvas Isoparamétricas

$$C(u) = S(u, \bar{v}) = \sum_{i=0}^n \left[\sum_{j=0}^m P_{ij} B_j^m(\bar{v}) \right] B_i^n(u)$$

$$Q_i = \sum_{j=0}^m P_{ij} B_j^m(\bar{v}) \quad i = 0, \dots, n$$

Control seudolocal

La modificación de un punto de control afecta toda la superficie, siendo más evidente en las cercanías al punto modificado.

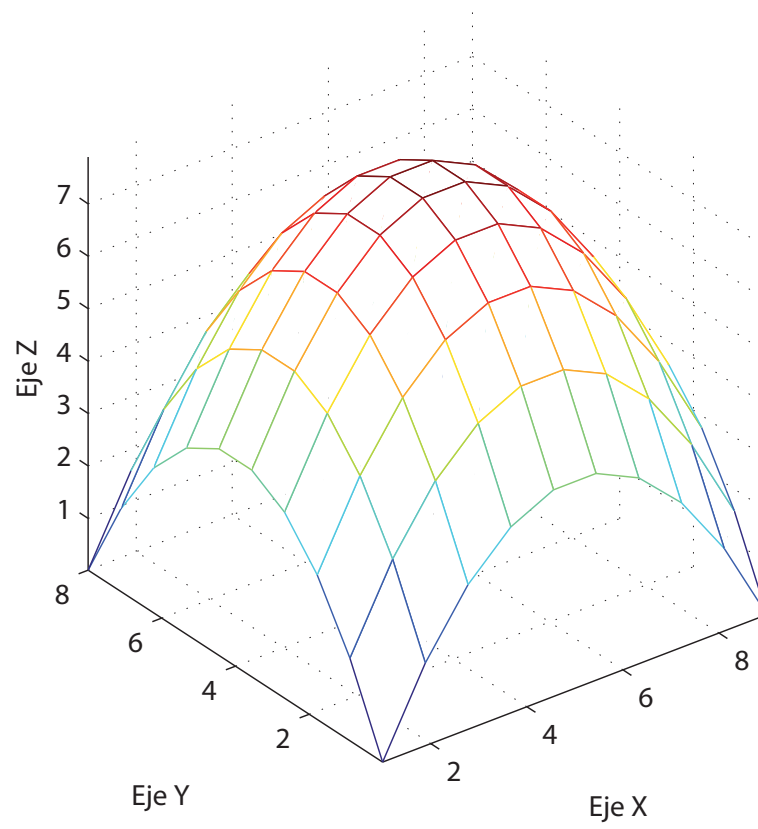


Figura F.1: Patch de Bézier

F.2. Superficies de Bézier compuestas

Aquí se tienen en cuenta las mismas consideraciones de las curvas de Bézier compuestas, de manera análoga no es efectivo trabajar con solo un *patch de Bézier* ya que para lograr mayores exactitudes se tendría que trabajar con grados elevados en los polinomios de Bernstein, por ello se usan varios *patch de Bézier*, es decir superficies a trozos, a los que se les impone continuidad en las fronteras de unión y continuidad de las derivadas parciales a lo largo de dichas fronteras, formando superficies compuestas de clase C^1 .

F.2.1. Composición de dos superficies simples

Si $P_{ij} = \{i = 0, \dots, n\}, \{j = 0, \dots, m\}$ puntos de control para la superficie $S(u, v)$ definida en el rectángulo $[u_0, u_1] \times [v_0, v_1]$:

Y $t = \frac{u-u_0}{u_1-u_0}, s = \frac{v-v_0}{v_1-v_0}$ Parámetros locales.

$$S_1(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} B_i^n(t) B_j^m(s)$$

La segunda superficies se supone del los mismos grados m y n y definida sobre el rectángulo adyacente a S_1

$$S_2(u, v) = \sum_{i=0}^n \sum_{j=0}^m Q_{ij} B_i^n(r) B_j^m(s) \quad r = \frac{u - u_2}{u_2 - u_1}, s = \frac{v - v_0}{v_1 - v_0}$$

Siendo curvas isoparamétricas están definidas dentro del intervalo $[v_0, v_1]$ esto será cierto si coinciden sus puntos de control

$$P_{nj} = Q_{0j} \quad \text{para } j = 0, \dots, m \quad (\text{F.3})$$

lo que evidencia continuidad C^0 .

La superficie será de clase C^1 si cumple ciertas condiciones.

$$\frac{1}{u_1 - u_0} (P_{nj} - P_{n-1,j}) = \frac{1}{u_2 - u_1} (Q_{1j} - Q_{0j}) \quad j = 0, \dots, m \quad (\text{F.4})$$

lo que implica que sus puntos extremos están alineados y manteniendo distancias proporcionales.

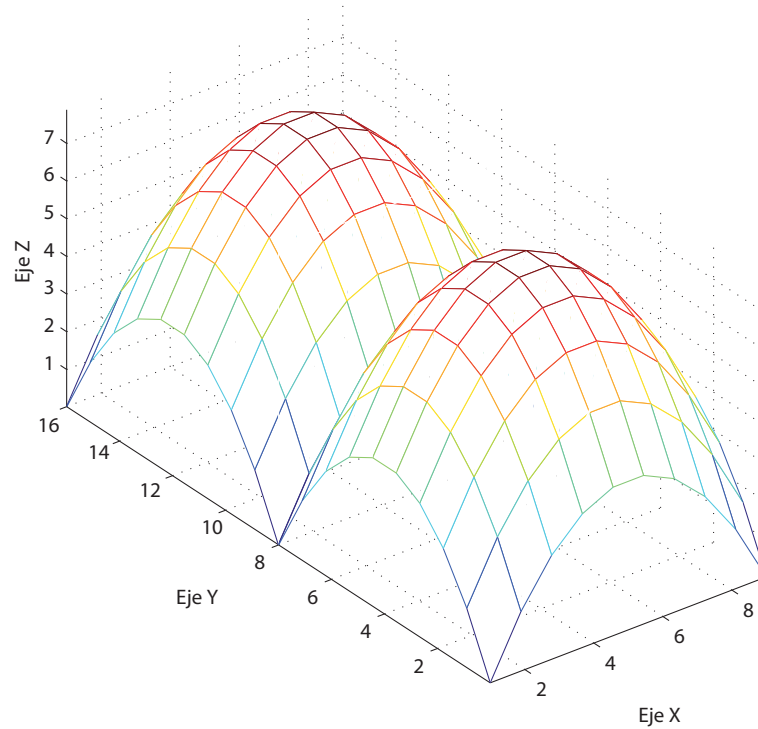
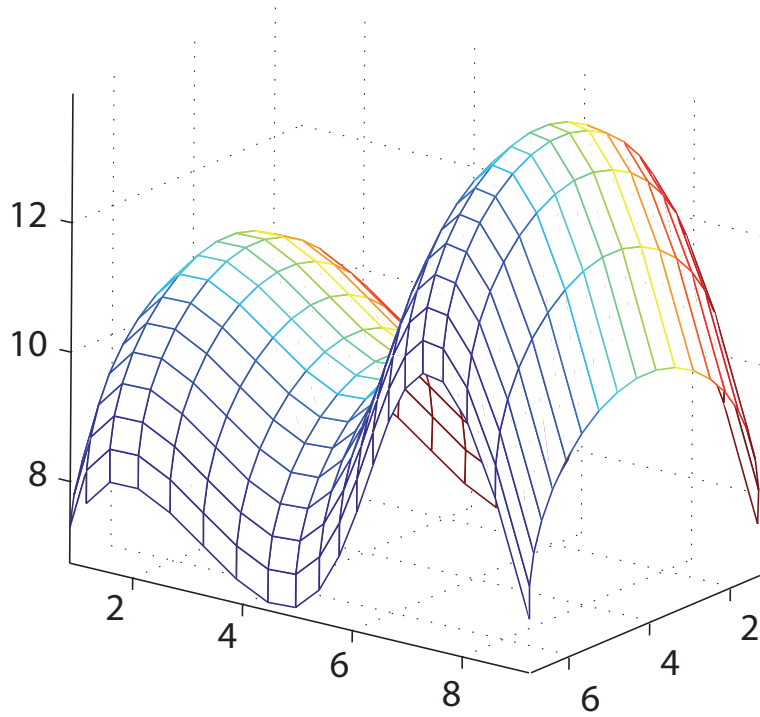


Figura F.2: Unión de dos superficies de Bézier

A pesar de que las condiciones para lograr la continuidad C^1 están dadas a partir de lo sucedido en una variable, se deben tener en cuenta ciertas restricciones impuestas a la malla de control, por ejemplo los puntos extremos no solo deben estar alineados sino que deben permanecer a unas distancias proporcionales entre sí, ahora bien en caso de que los puntos no estén a distancias proporcionales, el problema no podrá solucionarse reparametrizando, y la superficie será discontinua en este punto.



F.2.2. Superficies compuestas rectangulares

Sean considerados los soportes

$$S_u = \{u_0, \dots, u_p\} \quad S_v = \{v_0, \dots, v_q\}$$

lo que incluye una malla rectangular en el espacio paramétrico de puntos (u_k, v_h) , en cada rectángulo se puede definir un patch de Bézier de grado m en v y n en u .

$$S_{kh}(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{kn+i, hm+j} B_i^n(t) B_j^m(s) \quad (\text{F.5})$$

Donde $t = \frac{u-u_k}{u_{k+1}-u_k}$, $s = \frac{v-v_h}{v_{h+1}-v_h}$ y $P_{kn+i, hm+j}$ son los puntos de control,

Los puntos de control a lo largo de un borde son compartidos por el patch adyacente.

Por lo tanto en el patch $S_{kh}(u, v)$ están implicados los puntos:

$$\begin{array}{cccc} P_{kn, hm} & P_{kn, hm+1} & \cdots & P_{kn, (h+1)m} \\ P_{kn+1, hm} & P_{kn+1, hm+1} & \cdots & P_{kn+1, (h+1)m} \\ \vdots & \vdots & \vdots & \vdots \\ P_{(k+1)n, hm} & P_{(k+1)n, hm+1} & \cdots & P_{(k+1)n, (h+1)m} \end{array}$$

Las condiciones de interpolación para este patch son:

$$\frac{1}{u_{k+1} - u_k}(P_{knj} - P_{kn-1,j}) = \frac{1}{u_{k+1} - u_k}(P_{kn+1j} - P_{kn,j}) \quad \begin{cases} j = 0, \dots, qm \\ k = 0, \dots, p-1 \end{cases} \quad (\text{F.6})$$

en v

$$\frac{1}{v_{h+1} - v_h}(P_{ihm} - P_{ihm-1}) = \frac{1}{v_{h+1} - v_h}(P_{i,hm+1j} - P_{ihm}) \quad \begin{cases} j = 0, \dots, pn \\ h = 0, \dots, q-1 \end{cases} \quad (\text{F.7})$$

F.2.3. Interpolación con superficies de Bézier C^1

Sean un conjunto de $(p+1)(q+1)$ puntos Q_{ij} que se deben interpolar por una superficie de Bézier compuesta en el soporte $S_u \times S_v$

La primera fase consiste en realizar $(q+1)$ interpolaciones de modo que:

$$\begin{array}{lll} Q_{00}, Q_{10}, \dots, Q_{p0} & \implies & \text{genera puntos de control } R_{00}, R_{10}, \dots, R_{pn,0} \\ Q_{01}, Q_{11}, \dots, Q_{p1} & \implies & \text{genera puntos de control } R_{01}, R_{11}, \dots, R_{pn,1} \\ \vdots & \vdots & \vdots \\ Q_{0q}, Q_{1q}, \dots, Q_{pq} & \implies & \text{genera puntos de control } R_{0q}, R_{1q}, \dots, R_{pn,q} \end{array}$$

A continuación, los puntos de control obtenidos se interpolan por columnas

$$\begin{array}{lll} R_{00}, R_{10}, \dots, R_{pn,0} & \implies & \text{genera puntos de control } P_{00}, P_{10}, \dots, P_{0,qm} \\ R_{01}, R_{11}, \dots, R_{pn,1} & \implies & \text{genera puntos de control } P_{01}, P_{11}, \dots, P_{1,qm} \\ \vdots & \vdots & \vdots \\ R_{0q}, R_{1q}, \dots, R_{pn,q} & \implies & \text{genera puntos de control } P_{0q}, P_{1q}, \dots, P_{pn,qm} \end{array}$$

Estos puntos de control componen la malla de Bézier y definen la superficie de Bézier. Ahora para lograr una notable simplificación de los cálculos, se hacen $n = m = 3$, para lo que se pueden utilizar los métodos para estimación de derivadas en cada Q_{ij} en la correspondiente dirección, y después se obtienen los puntos de control intermedios usando el método de Hermite, este procedimiento se repite a continuación en la dirección faltante.

F.3. Superficies B-Spline

Las curvas de Bézier, presentan un gran inconveniente cuando se establecen las condiciones de continuidad en los puntos extremos, ya que para lograrlo se deben imponer serias restricciones en la estructura de los puntos de control. Esta es la principal razón para desarrollar las bases B-spline, que haciendo uso del concepto de nodo múltiples, que establece ordenes de continuidad y mantiene los puntos de control totalmente ajenos a esta cuestión, liberándolos de las restricciones presentadas en las curvas de Bézier.

Considerar bases B-Spline producto tensorial proporciona una herramienta flexible de modelado de superficies, ya que permite independizar las condiciones de continuidad de la estructura de los puntos de control.

Considerando dos soportes de nodos no periódicos definidos en $[0, 1]$ en las variables u y v

$$S_u = \{u_0, \dots, u_r\} \quad S_v = \{v_0, \dots, v_s\}$$

$$u_0 = u_1 = \dots = u_p = 0 \quad u_{r-p} = u_{r-p+1} = \dots = u_r = 1$$

$$v_0 = v_1 = \dots = v_q = 0 \quad v_{s-q} = v_{s-q+1} = \dots = v_s = 1$$

Siendo p y q el correspondiente grado de los B-spline, y generan un espacio $n + 1 = r - p, m + 1 = s - q$, una superficie B-spline entonces está representada de la siguiente manera:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} B_i^p(u) B_j^q(v) \quad (\text{F.8})$$

Donde $P_{ij} = (x_{ij}, y_{ij}, z_{ij})$ que constituyen la malla de control, aquí las constantes m y n tienen un significado muy distinto al caso de las superficies de Bézier en donde representan el grado de los polinomios.

F.3.1. Propiedades de las superficies B-spline

1. Interpolación de las esquinas de la malla de control.

- $S(0, 0) = P_{00}$

- $S(0, 1) = P_{0m}$
- $S(1, 0) = P_{n0}$
- $S(1, 1) = P_{nm}$

2. Derivadas parciales en las esquinas de la malla de control

$$\frac{\partial S(u, v)}{\partial(u)} = \sum_{i=0}^n \sum_{j=0}^m P_{ij} \left(\frac{dB_i^p(u)}{du} B_j^q(v) \right) = \sum_{i=0}^n \left[\sum_{j=0}^m P_{ij} \left(\frac{dB_i^p(u)}{du} \right) B_j^q(v) \right]$$

$$\frac{\partial S(0, v)}{\partial(u)} = \frac{p}{u_{p+1}} \sum_{j=0}^m (P_{1j} - P_{0j}) B_j^q(v)$$

$$\frac{\partial S(1, v)}{\partial(u)} = \frac{p}{1 - u_n} \sum_{j=0}^m (P_{nj} - P_{n-1j}) B_j^q(v)$$

Las derivadas parciales respecto a las 4 esquinas

$$\frac{\partial S(0, 0)}{\partial(u)} = \frac{p}{u_{p+1}} (P_{10} - P_{00})$$

$$\frac{\partial S(0, 1)}{\partial(u)} = \frac{p}{u_{p+1}} (P_{1m} - P_{0m})$$

$$\frac{\partial S(1, 0)}{\partial(u)} = \frac{p}{1 - u_n} \sum_{j=0}^m (P_{n0} - P_{n-1j})$$

$$\frac{\partial S(1, 1)}{\partial(u)} = \frac{p}{1 - u_n} (P_{nm} - P_{n-1m})$$

Las curvas isoparamétricas $v = 0$ y $u = 0$ son tangentes al primer y último segmento de los polígonos $P_{00}, P_{10}, \dots, P_{0n}$ y $P_{1m}, P_{1m}, \dots, P_{nm}$ respectivamente.

3. Orden de continuidad

En un punto (u_k, v_h) donde u_k y v_h es de multiplicidad v , $S(u, v)$ tiene derivadas parciales de orden $p - u$ con respecto a u y $q - v$ con respecto a v

4. Control local

Si un punto de control es modificado, entonces la nueva superficie se diferencia de la inicial en:

$$S(u, v) - \bar{S}(u, v) = (P_{kh} - \bar{P}_{kh})B_h^q(v)B_k^p(u)$$

diferenciándose solo en el rectángulo paramétrico:

$$[u_k, u_{k+p+1}] \times [v_h, v_{h+q+1}]$$

5. Restricción de la envolvente convexa

Los B-Splines constituyen en cada variable una partición de la unidad con la base del producto tensorial.

$$\sum_{i=0}^n \sum_{j=0}^m B_i^p(u)B_j^q(v) = 1$$

Un punto en el espacio paramétrico es una combinación convexa de los puntos de control:

$$S(\bar{u}, \bar{v}) = \sum_{i=0}^n \sum_{j=0}^m P_{ij}a_{ij}$$

Donde a_{ij} y $\sum_{i=0}^n \sum_{j=0}^m a_{ij} = 1$

los únicos B-spline no nulos son:

$$B_{k-p}^p(u), B_{k+p+1}^p(u), \dots, B_k^p(u)$$

y

$$B_{h-q}^q(v), B_{h+q+1}^q(v), \dots, B_h^q(v)$$

dando lugar a

$$S(\bar{u}, \bar{v}) = \sum_{i=k-p}^k \sum_{j=h-q}^h P_{ij}a_{ij} \quad \sum_{i=k-p}^k \sum_{j=h-q}^h a_{ij} = 0$$

implicando que la superficie compuesta por la combinación convexa de los puntos es interior a la envolvente convexa.

6. Invarianza Afín

La transformación se aplica a lo puntos de control y a continuación se encuentra las funciones bases para obtener la superficie transformada.

7. Superficies de Bézier como B-spline

Si los soportes no contienen nodos interiores:

$$S_u = \{\underbrace{0, \dots, 0}_r, \underbrace{1, \dots, 1}_p\} \quad S_v = \{\underbrace{0, \dots, 0}_m, \underbrace{1, \dots, 1}_q\}$$

y la superficie B-spline es un patch de Bézier

F.3.2. Interpolación con superficies B-Spline

Es conveniente que el número de puntos a interpolar así como su estructura coincida con el número de funciones bases, lo que determina el número de nodos correspondiente a los soportes de definición de B-splines.

si Q_{ij} $0 \leq i \leq n$, $0 \leq j \leq m$ son los puntos a interpolar, se busca una superficie B-spline de la forma $S(u, v) = P_{ij} B_i^p(u) B_j^q(v)$ de grado p en u y q en v , con soporte $S_u = \{u_0, \dots, u_{n+p+1}\}$ que contiene $r + 1 = n + p + 2$ nodos y $S_v = \{v_0, \dots, v_{m+q+1}\}$ con $s + 1 = m + q + 2$ nodos.

Se debe especificar el soporte de interpolación en cada variable, a continuación se lleva a cabo el proceso de interpolación de la misma forma en que se realizó con las superficies de Bézier, comenzando con una de las variables y a continuación con la siguiente, dependiendo del orden elegido para interpolar:

$$\begin{array}{lll} Q_{00}, Q_{10}, \dots, Q_{p0} & \implies & \text{se interpola con } C_0(u) = \sum_{i=0}^n R_{i0} B_i^p(u) \\ Q_{01}, Q_{11}, \dots, Q_{p1} & \implies & \text{se interpola con } C_1(u) = \sum_{i=0}^n R_{i1} B_i^p(u) \\ \vdots & \vdots & \vdots \\ Q_{0q}, Q_{1q}, \dots, Q_{pq} & \implies & \text{se interpola con } C_m(u) = \sum_{i=0}^n R_{im} B_i^p(u) \end{array}$$

Ahora se interpolan en la variable v los coeficientes R_{ij} por columnas

$$\begin{array}{lll} R_{00}, R_{10}, \dots, R_{pn,0} & \implies & \text{se interpola con } D_0(v) = \sum_{j=0}^m P_{i0} B_j^q(v) \\ R_{01}, R_{11}, \dots, R_{pn,1} & \implies & \text{se interpola con } D_1(v) = \sum_{j=0}^m P_{i1} B_j^q(v) \\ \vdots & \vdots & \vdots \\ R_{0q}, R_{1q}, \dots, R_{pn,q} & \implies & \text{se interpola con } D_m(v) = \sum_{j=0}^m P_{im} B_j^q(v) \end{array}$$

Donde los coeficientes P_{ij} son los puntos de control de la superficie B-Spline.

F.4. Superficies racionales

Como ya se había mencionado antes una superficie racional se caracteriza por que la representación paramétrica de las tres variables x, y, z están dadas por el cociente de dos polinomios, para este caso dos superficies polinomiales, de modo que

$$C(u) = \begin{cases} \bar{x} = \frac{P(u)}{W(u)} \\ \bar{y} = \frac{Q(u)}{W(u)} \\ \bar{z} = \frac{R(u)}{W(u)} \end{cases} \quad (\text{F.9})$$

Representada en una base, fruto del producto tensorial de dos bases univariadas, se tiene:

$$W(u, v) = \sum_{i=0}^n \sum_{j=0}^m w_{ij} B_i(u) C_j(v)$$

cuyos coeficientes se representan así:

$$a_{ij} = w_{ij} x_{ij}$$

$$b_{ij} = w_{ij} y_{ij}$$

$$c_{ij} = w_{ij} z_{ij}$$

y la superficie queda representada por

$$S(u, v) = \begin{cases} x = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} x_{ij} B_i(u) C_j(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_i(u) C_j(v)} \\ y = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} y_{ij} B_i(u) C_j(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_i(u) C_j(v)} \\ z = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} z_{ij} B_i(u) C_j(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_i(u) C_j(v)} \end{cases}$$

de forma vectorial

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} P_{ij} B_i(u) C_j(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_i(u) C_j(v)} \quad (\text{F.10})$$

Donde son los pesos w_{ij} y P_{ij} son los puntos de control de la superficie, ó podran tener

otro significado, dependiendo de la base elegida

La curva racional puede considerarse una proyección sobre el hiperplano $W = 1$, de una superficies polinomial 4D

$$\begin{aligned}x &= \sum_{i=0}^n \sum_{j=0}^m w_{ij} x_{ij} B_i(u) C_j(v) \\y &= \sum_{i=0}^n \sum_{j=0}^m w_{ij} y_{ij} B_i(u) C_j(v) \\z &= \sum_{i=0}^n \sum_{j=0}^m w_{ij} z_{ij} B_i(u) C_j(v) \\w &= \sum_{i=0}^n \sum_{j=0}^m w_{ij} w_{ij} B_i(u) C_j(v)\end{aligned}$$

y los puntos de control son $P_{ij} = (w_{ij} x_{ij}, w_{ij} y_{ij}, w_{ij} z_{ij}, w_{ij})$

el problema de interpolación se resuelve operando en una dimensión más haciendo uso de funciones polinomiales y proyectando sobre un plano en R^3

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} \left[\frac{w_{ij} B_i(u) C_j(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_i(u) C_j(v)} \right] \quad (\text{F.11})$$

llamando

$$R(u, v) = \frac{w_{ij} B_i(u) C_j(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_i(u) C_j(v)} \quad (\text{F.12})$$

la superficie queda definida como:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} R_{ij}(u, v) \quad (\text{F.13})$$

Una superficie racional no es propiamente una superficie producto tensorial, sino más bien una proyección de una hiper superficie producto tensorial de \mathbb{R}^4

siendo posible comprobar que:

$$\sum_{i=0}^n \sum_{j=0}^m R_{ij}(u, v) = 1 \quad (\text{F.14})$$

Y

$$B_i(u) \geq 0, C_j(v) \geq 0$$

$R_{ij}(u, v) \geq 0$ y todo punto de la superficie es una combinación convexa de los coeficientes P_{ij}

De manera análoga con las curvas racionales, una superficie producto tensorial es solo un caso particular para el que $w_{ij} = 1$

F.4.1. Patches de Bézier Racionales

Teniendo una superficie de Bézier simple:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} P_{ij} B_i^n(u) C_j^m(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_i^n(u) C_j^m(v)} \quad u, v \in [0, 1]$$

donde las funciones $B_i(u)$, $C_j(v)$ son los polinomios de Bernstein de grado n y m respectivamente. Además se cumple que $\sum_{i=0}^n B_i^n(u) = \sum_{j=0}^m C_j^m(v) = 1$ y $B_i^n(u) \geq 0$, $C_j^m(v) \geq 0$ (invarianza afín y restricción en la envolvente convexa de los puntos de control)

Se cumple también la propiedad de interpolación de las cuatro esquinas $P_{00}, P_{0m}, P_{n0}, P_{nm}$.

Entonces, existe una cierta invarianza proyectiva fundamental para la fase de rendering. Es decir, en lugar de calcular la perspectiva, con profundidad, basta con transformar por perspectiva los puntos de control, recalculando los pesos y reconstruir una nueva superficie racional.

Dada la proyectividad:

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \\ \bar{w} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & h_4 \\ a_{21} & a_{22} & a_{23} & h_4 \\ a_{31} & a_{32} & a_{33} & h_4 \\ p_3 & p_3 & p_3 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \quad (\text{F.15})$$

La superficie transformada es:

$$\Pi(S(u, v)) = \frac{\sum_{i=0}^n \sum_{j=0}^m \sigma_{ij} \Pi(P_{ij}) B_i^n(u) C_j^m(v)}{\sum_{i=0}^n \sum_{j=0}^m \sigma_{ij} B_i^n(u) C_j^m(v)}$$

Los $\Pi(P_{ij})$ son los transformados de los P_{ij} por la proyectividad.

Lo que usualmente se conoce como perspectiva es en realidad una proyectividad como F.15, donde se suprime la coordenada z para obtener puntos (x, y) .

F.4.2. Cuádricas

Teniendo la expresión para la esfera centrada en el origen y de radio 1 puede expresarse:

$$S(u, v) = \begin{cases} x = \frac{1-u^2}{1+u^2} \frac{1-v^2}{1+v^2} \\ y = \frac{1-u^2}{1+u^2} \frac{2v}{1+v^2} \\ z = \frac{2u}{1+u^2} \end{cases}$$

Se recurre a superficies de Bézier racionales compuestas para representarla gráficamente. Restringiendo u y v al intervalo $[0, 1]$ se obtiene un octante de la esfera. Al obtener los puntos de control $P_{00} = P_{01} = P_{02} = (0, 0, 1)$ y el patch representa a un octante de la esfera.

Repitiendo de forma simétrica estos puntos de control se genera la esfera completa como superficie de Bézier racional compuesta por ocho Patches.

F.4.3. Superficies B-Spline racionales (NURBS)

Una superficie NURBS está expresada de la forma F.10 donde ahora $B_i(u)$ y $C_j(v)$ se toman como los B-Splines $B_i^p(u)$ y $B_j^q(v)$ definidos sobre dos soportes de nodos (no periódicos):

$$S_u = \{u_0, \dots, u_r\} \quad S_v = \{v_0, \dots, v_s\}$$

El número de funciones de cada base son $n + 1$ y $m + 1$ respectivamente. El número de puntos de cada soporte es: $r = n + p + 1$ y $s = m + q + 1$, siendo p y q los grados de los B-Splines $B_i^p(u)$ y $B_j^q(v)$. Entonces:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} P_{ij} B_i^p(u) B_j^q(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_i^p(u) B_j^q(v)}$$

Las propiedades de las superficies B-Splines polinomiales son compartidas por las racionales. Las superficies NURBS son generalizaciones de las superficies racionales y no racionales, así como de los B-Splines polinomiales.

Estas superficies constituyen el esquema de modelado de superficies más general posible entre los descritos. Sin embargo, algunos problemas prácticos no necesitan la generalidad y flexibilidad de las superficies NURBS, ya que la interpolación discreta y la transinfinita se resuelven de forma más eficiente con superficies o curvas no racionales.

F.5. Técnicas constructivas

La herramienta fundamental en la construcción de superficies es la interpolación de un conjunto discreto de puntos Q_{ij} distribuidos en una estructura rectangular.

F.5.1. Superficies regladas

Dadas dos curvas NURBS del mismo grado p y definidas en el soporte $S_u = \{u_0, \dots, u_n\}$:

$$C_1(u) = \frac{\sum_{i=0}^n w_i^{(1)} P_i^{(1)} B_i^p(u)}{\sum_{i=0}^n w_i^{(1)} B_i^p(u)}$$

$$C_2(u) = \frac{\sum_{i=0}^n w_i^{(2)} P_i^{(2)} B_i^p(u)}{\sum_{i=0}^n w_i^{(2)} B_i^p(u)}$$

Ahora lo que interesa es expresar las funciones de base en la variable v como curvas NURBS.

Dado que $B_1(v) = 1 - v$, $B_2(v) = v$ son los polinomios de Bernstein de grado 1, pueden considerarse a su vez B-Splines de primer grado sobre el soporte no periódico $S_v = 0, 0, 1, 1$. Por tanto:

$$S(u, v) = (1 - v) C_1(u) + v C_2(u)$$

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} \frac{\sum_{i=0}^n w_i^{(1)} B_i^p(u)}{\sum_{i=0}^n w_i^{(1)} B_i^p(u)} \quad (\text{F.16})$$

A pesar de su apariencia F.16 no es una superficie NURBS tal y como se definió en la sección anterior (a menos que los pesos en ambas curvas sean iguales). Para construir una superficie NURBS es necesario efectuar la interpolación lineal en coordenadas homogéneas, es decir, también deben ser interpolados los pesos, y entonces se obtiene:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^1 w_i^{(j)} P_i^{(j)} B_i^p(u) B_j^1(v)}{\sum_{k=0}^n \sum_{h=0}^1 w_k^{(h)} B_k^p(u) B_h^1(v)}$$

F.5.2. Superficies cilíndricas

Dada una curva NURBS sobre el soporte $S_u = \{u_0, \dots, u_n\}$

$$C(u) = \frac{\sum_{i=0}^n w_i P_i B_i^p(u)}{\sum_{i=0}^n w_i B_i^p(u)}$$

la superficie definida por $C(u)$ y un vector constante D es:

$$S(u, v) = (1 - v) C(u) + v [C(u) + D] = C(u) + vD$$

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^1 w_i P_{ij} B_i^p(u) B_j^1(v)}{\sum_{k=0}^n \sum_{h=0}^1 w_k B_k^p(u) B_h^1(v)}$$

el soporte de nodos en la variable v es: $S_v = \{0, 0, 1, 1\}$

F.5.3. Superficies de revolución

Dada una curva NURBS en el plano XZ :

$$S(u, v) = \begin{cases} C_x(u) = \frac{\sum_{i=0}^n w_i x_i B_i^p(u)}{\sum_{i=0}^n w_i B_i^p(u)} \\ C_y(u) = 0 \\ C_z(u) = C_x(u) = \frac{\sum_{i=0}^n w_i z_i B_i^p(u)}{\sum_{i=0}^n w_i B_i^p(u)} \end{cases}$$

La superficie generada al rotar la curva $C(u)$ alrededor del eje Z viene dada por:

$$S(u, v) = \begin{cases} x = C_x(u) c(v) \\ y = C_z(u) s(v) \\ z = C_z(u) = \frac{\sum_{i=0}^n w_i z_i B_i^p(u)}{\sum_{i=0}^n w_i B_i^p(u)} \end{cases}$$

donde $c(v)$, $s(v)$ son las componentes x e y de una circunferencia paramétrica en el plano XY :

$$T(v) = \begin{cases} T_x(v) = c(v) \\ T_y(v) = s(v) \\ T_z(v) = 0 \end{cases}$$

Representando la circunferencia como una curva NURBS, tomando como puntos de

control:

$$R_0 = (1, 0) \quad R_1 = (1, 0) \quad R_2 = (1, 0) \quad R_3 = (1, 0) \quad R_4 = (1, 0)$$

$$R_5 = (1, 0) \quad R_6 = (1, 0) \quad R_7 = (1, 0) \quad R_8 = (1, 0)$$

con pesos:

$$\sigma_0 = \sigma_2 = \sigma_4 = \sigma_6 = \sigma_8 = 1 \quad \sigma_1 = \sigma_3 = \sigma_5 = \sigma_7 = \sigma_8 = \frac{\sqrt{2}}{2}$$

entonces:

$$T(v) = \frac{\sum_{i=0}^8 \sigma_j R_j B_j^2(v)}{\sum_{i=0}^8 \sigma_j B_j^2(v)}$$

siendo $B_j^2(v)$ los B-Splines de segundo grado definidos con respecto al soporte:

$$S_v = \left\{ 0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1 \right\}$$

Por tanto, la superficie queda en su forma vectorial:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^8 (w_i \sigma_j) (x_i R_{jx}, x_i R_{jy}, z_i) B_i^p(u) B_j^2(v)}{\sum_{i=0}^n \sum_{j=0}^8 (w_i \sigma_j) B_i^p(u) B_j^2(v)}$$

F.5.4. Superficies de barrido

La ecuación definida para una superficie obtenida por barrido es:

$$S(u, v) = A(u)C(u) + T(v)$$

donde $C(u)$ es la curva base, $T(v)$ la trayectoria y $A(v)$ la afinidad que transforma a $C(u)$

En el caso que $A(u) = I$, la curva $C(u)$ simplemente se traslada si deformación:

$$S(u, v) = C(u) + T(v)$$

Si la curva generatriz es:

$$C(u) = \frac{\sum_{i=0}^n w_i P_i B_i^p(u)}{\sum_{i=0}^n w_i B_i^p(u)}$$

y la trayectoria:

$$T(v) = \frac{\sum_{j=0}^m \sigma_j Q_j B_j^q(v)}{\sum_{j=0}^m \sigma_j B_j^q(v)}$$

Entonces, la superficie obtenida es:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m (w_i \sigma_j) (P_i + Q_j) B_i^p(u) B_j^q(v)}{\sum_{i=0}^n \sum_{j=0}^m (w_i \sigma_j) B_i^p(u) B_j^q(v)}$$

Algoritmo: Superficie de barrido

```
% curva X-Z
clear all
close all
clc
axis([-1,11,-1,10])
hold on
ylabel('altura Y')
xlabel('curva generatriz')
count=0; orden=0;
n=7; grado=3;
m=n+grado+1;
d=0;
t=0;
for i=1:n
    a=ginput(1);
    plot(a(1),a(2),'o')
    Q(i,1)=a(1);
    Q(i,2)=a(2);
end
w=Q(:,1)./Q(:,2);
```

```

for i=1:n
    if w(i)>1
        w(i)=inv(w(i));
    end
end
for i=2:n
    d=d+((Q(i,1)-Q(i-1,1))^2+(Q(i,2)-Q(i-1,2))^2)^0.5;
end
for i=1:n-1
    t(i+1)=t(i)+(((Q(i+1,1)-Q(i,1))^2+(Q(i+1,2)-Q(i,2))^2)^0.5)/d;
end
for i=5:m-4
    S(i)=(t(i-1)+t(i-2)+t(i-3))/3;
end
for j=i+1:m
    S(j)=t(n);
end
for i=1:m-1
    if S(i+1)==S(i)
        u(i)=0;
    else
        u(i)=1;
    end
end
for i=1:m-1
    for j=1:m-1
        para((m-1)*(i-1)+j,1)=orden;        para((m-1)*(i-1)+j,2)=i;
        para((m-1)*(i-1)+j,3)=S(j);        para((m-1)*(i-1)+j,4)=S(j+1);
        if i==j
            B((m-1)*(i-1)+j,1)=u(j);
        end
        count=count+1;
    end
end
curvas(1,1)=1;  curvas(1,2)=count;

```

```

for orden=1:grado
    curvas(orden+1,1)=count+1;
    for i=1:m-1-orden
        e=-S(i);          f=S(i+orden+1);
        b=S(i+orden)+e;  d=f-S(i+1);
        for j=1:m-1
            count=count+1;
            para(count,1)=orden;      para(count,2)=i;
            para(count,3)=S(j);      para(count,4)=S(j+1);
            miu=B(count-(m-orden)*(m-1),:);
            bi=miu(orden:-1:1);      bi(orden+1)=0;
            miu(orden+1)=0;          miu=miu(orden+1:-1:1);
            miu=miu*e;                a=bi+miu;
            miu=B(count-(m-orden-1)*(m-1),:);
            bi=-miu(orden:-1:1);     bi(orden+1)=0;
            miu(orden+1)=0;          miu=miu(orden+1:-1:1);
            miu=miu*f;                c=bi+miu;
            if b==0
                a=0;      b=1;
            elseif d==0
                c=0;      d=1;
            end
            Bi=a/b+c/d;          Bi=Bi(orden+1:-1:1);
            for k=1:orden+1
                B(count,k)=Bi(k);
            end
        end
    end
    curvas(orden+1,2)=count;
end
conter=0;
for j=curvas(grado+1,1):(m-1):count-1
    for i=1+grado:m-1-grado
        conter=conter+1;
        bi=B(j+i-1,:);
    end
end

```

```

        bi=bi(grado+1:-1:1);
        for k=1:grado+1
            BI(conter,k)=bi(k);
        end
        par(conter,1)=para(j+i-1,1);        par(conter,2)=para(j+i-1,2);
        par(conter,3)=para(j+i-1,3);        par(conter,4)=para(j+i-1,4);
    end
end
clear B
B(1,1)=1;    B(n,n)=1;
for j=2:n-1
    for i=1:n
        ba=0;
        for k=1:m-7
            if t(j)>=par((i-1)*(m-7)+k,3) && t(j)<par((i-1)*(m-7)+k,4)
                d=BI((i-1)*(m-7)+k,:);
                ba=ba+polyval(d,t(j));
            end
        end
        B(j,i)=ba;
    end
end
end
P(:,1)=inv(B)*Q(:,1);
P(:,2)=inv(B)*Q(:,2);
Cx=zeros(m-1-2*grado,orden+1);Cy=Cx;Dxy=Cx;
for j=0:par(conter,2)-1
    for i=1:m-1-2*grado
        ncx=BI(j*(m-1-2*grado)+i,:)*P(j+1,1)*w(par(conter,2)-1);
        %numerador cx sum_i=0{wi*Pi*Bi(u)}
        ncy=BI(j*(m-1-2*grado)+i,:)*P(j+1,2)*w(par(conter,2)-1);
        dc=BI(j*(m-1-2*grado)+i,:)*w(par(conter,2)-1);
        %denominador sum_i=0->n{wi*Bi} el mismo pa'X,Y
        %cx=BI(j*(m-1-2*grado)+i,:)*P(j+1,1);
        %cy=BI(j*(m-1-2*grado)+i,:)*P(j+1,2);
        for k=1:orden+1

```

```

        Cx(i,k)=Cx(i,k)+ncx(k);
        Cy(i,k)=Cy(i,k)+ncy(k);
        Dxy(i,k)=Dxy(i,k)+dc(k);
    end
end
end
T=linspace(S(1),S(m),100);
to=length(T);
for j=1+grado:m-1-grado
    cx=Cx(j-grado,:);
    cy=Cy(j-grado,:);
    dxy=Dxy(j-grado,:);
    for i=1:to
        if T(i)>=S(j)
            %X(i)=polyval(cx,T(i));
            %Y(i)=polyval(cy,T(i));
            X(i)=polyval(cx,T(i))/polyval(dxy,T(i));
            Y(i)=polyval(cy,T(i))/polyval(dxy,T(i));
        end
    end
end
end
plot(X,Y,'.r')
plot(P(:,1),P(:,2))
a=ginput(1);
% curva X-Y
close all
clc
axis([-1,11,-1,10])
hold on
ylabel('altura Z')
xlabel('curva trayectoria')
countq=0;    ordenq=0;
nq=7;        gradoq=3;
mq=nq+gradoq+1;
dq=0;

```

```

tq=0;
for i=1:n
    a=ginput(1);
    plot(a(1),a(2),'o')
    Qq(i,1)=a(1);
    Qq(i,2)=a(2);
end
wq=Qq(:,1)./Qq(:,2);
for i=1:nq
    if wq(i)>1
        wq(i)=inv(wq(i));
    end
end
for i=2:nq
    dq=dq+((Qq(i,1)-Qq(i-1,1))^2+(Qq(i,2)-Qq(i-1,2))^2)^0.5;
end
for i=1:nq-1
    tq(i+1)=tq(i)+(((Qq(i+1,1)-Qq(i,1))^2+(Qq(i+1,2)-Qq(i,2))^2)^0.5)/dq;
end
for i=5:mq-4
    Sq(i)=(tq(i-1)+tq(i-2)+tq(i-3))/3;
end
for j=i+1:mq
    Sq(j)=tq(nq);
end
for i=1:mq-1
    if Sq(i+1)==Sq(i)
        uq(i)=0;
    else
        uq(i)=1;
    end
end
for i=1:mq-1
    for j=1:mq-1
        paraq((mq-1)*(i-1)+j,1)=ordenq; paraq((mq-1)*(i-1)+j,2)=i;
    end
end

```

```

paraq((mq-1)*(i-1)+j,3)=Sq(j); paraq((mq-1)*(i-1)+j,4)=Sq(j+1);
    if i==j
        Bq((mq-1)*(i-1)+j,1)=uq(j);
    end
    countq=countq+1;
end
end
curvasq(1,1)=1; curvasq(1,2)=countq;
for ordenq=1:gradoq
    curvasq(ordenq+1,1)=countq+1;
    for i=1:mq-1-ordenq
        eq=-Sq(i);          fq=Sq(i+ordenq+1);
        bq=Sq(i+ordenq)+eq; dq=fq-Sq(i+1);
        for j=1:mq-1
            countq=countq+1;
            paraq(countq,1)=ordenq;          paraq(countq,2)=i;
            paraq(countq,3)=Sq(j);          paraq(countq,4)=Sq(j+1);
            miuq=Bq(countq-(mq-ordenq)*(mq-1),:);
            biq=miuq(ordenq:-1:1);          biq(ordenq+1)=0;
            miuq(ordenq+1)=0;          miuq=miuq(ordenq+1:-1:1);
            miuq=miuq*eq;          aq=biq+miuq;
            miuq=Bq(countq-(mq-ordenq-1)*(mq-1),:);
            biq=-miuq(ordenq:-1:1);          biq(ordenq+1)=0;
            miuq(ordenq+1)=0;          miuq=miuq(ordenq+1:-1:1);
            miuq=miuq*fq;          cq=biq+miuq;
            if bq==0
                aq=0;          bq=1;
            elseif dq==0
                cq=0;          dq=1;
            end
            Biq=aq/bq+cq/dq;          Biq=Biq(ordenq+1:-1:1);
            for k=1:ordenq+1
                Bq(countq,k)=Biq(k);
            end
        end
    end
end

```

```

        end
        curvasq(ordenaq+1,2)=countq;
    end
    conteraq=0;
    for j=curvasq(gradoq+1,1):(mq-1):countq-1
        for i=1+gradoq:mq-1-gradoq
            conteraq=conteraq+1;
            biq=Bq(j+i-1,:);
            biq=biq(gradoq+1:-1:1);
            for k=1:gradoq+1
                BIq(conteraq,k)=biq(k);
            end
            paraq(conteraq,1)=paraq(j+i-1,1);           paraq(conteraq,2)=paraq(j+i-1,2);
            paraq(conteraq,3)=paraq(j+i-1,3);           paraq(conteraq,4)=paraq(j+i-1,4);
        end
    end
    clear Bq
    Bq(1,1)=1;    Bq(nq,nq)=1;
    for j=2:nq-1
        for i=1:n
            baq=0;
            for k=1:mq-7
                if tq(j)>=paraq((i-1)*(mq-7)+k,3) && tq(j)<paraq((i-1)*(mq-7)+k,4)
                    dq=BIq((i-1)*(mq-7)+k,:);
                    baq=baq+polyval(dq,tq(j));
                end
            end
            Bq(j,i)=baq;
        end
    end
    end
    Pq(:,1)=inv(Bq)*Qq(:,1);
    Pq(:,2)=inv(Bq)*Qq(:,2);
    Cxq=zeros(mq-1-2*gradoq,ordenaq+1);Cyq=Cxq;Dxyq=Cxq;
    for j=0:paraq(conteraq,2)-1
        for i=1:mq-1-2*gradoq

```

```

ncxq=BIq(j*(mq-1-2*gradoq)+i,:)*Pq(j+1,1)*wq(parq(conterq,2)-1);
ncyq=BIq(j*(mq-1-2*gradoq)+i,:)*Pq(j+1,2)*wq(parq(conterq,2)-1);
dcq=BIq(j*(mq-1-2*gradoq)+i,:)*wq(parq(conterq,2)-1);
for k=1:ordenq+1
    Cxq(i,k)=Cxq(i,k)+ncxq(k);
    Cyq(i,k)=Cyq(i,k)+ncyq(k);
    Dxyq(i,k)=Dxyq(i,k)+dcq(k);
end
end
end
Tq=linspace(Sq(1),Sq(m),200);
toq=length(Tq);
for j=1+gradoq:mq-1-gradoq
    cxq=Cxq(j-gradoq,:);
    cyq=Cyq(j-gradoq,:);
    dxyq=Dxyq(j-gradoq,:);
    for i=1:toq
        if Tq(i)>=Sq(j)
            Xq(i)=polyval(cxq,Tq(i))/polyval(dxyq,Tq(i));
            Yq(i)=polyval(cyq,Tq(i))/polyval(dxyq,Tq(i));
        end
    end
end
end
plot(Xq,Yq,'.r')
plot(Pq(:,1),Pq(:,2))
a=ginput(1);
close all
[wZ,wY]=meshgrid(Y,Yq); %mesheo para z,y
[wX,wXq]=meshgrid(X,Xq); %mesheo para X, Xq
wXw=wX+wXq;% sumo los valores d X en (v), y en (u)
mesh(wXw,wZ,wY)
axis equal

```

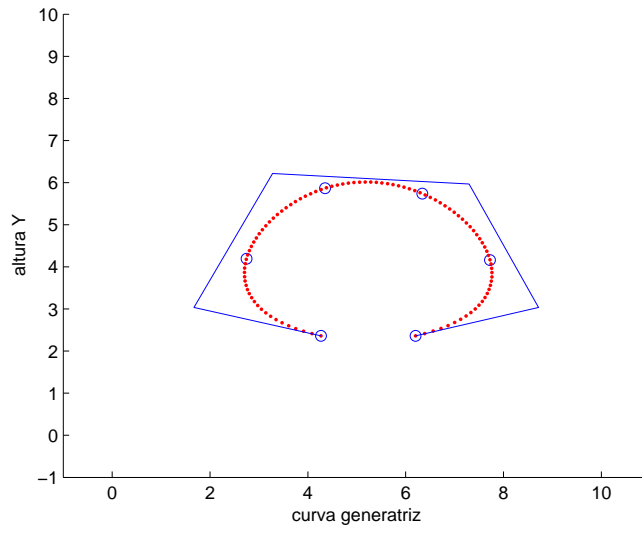


Figura F.3: Generatriz

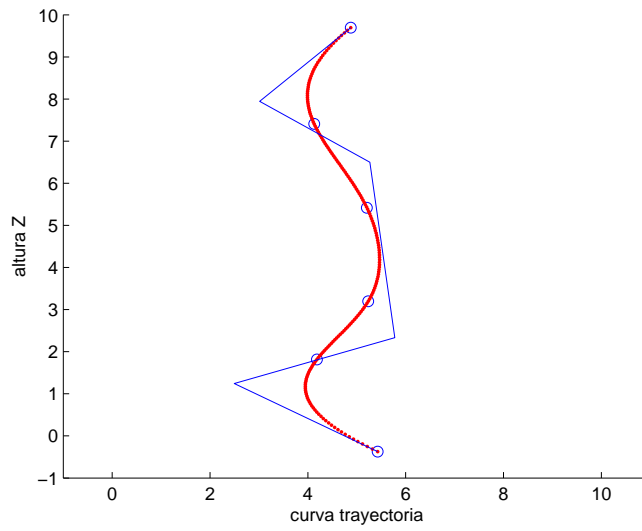


Figura F.4: Trayectoria

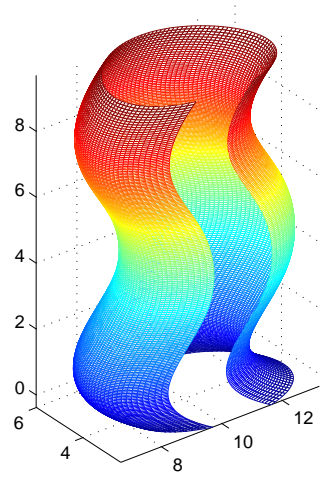


Figura F.5: Superficie de barrido