



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE CIENCIAS

Un módulo para el suavizamiento de
contornos usando spline en tensión para el
sistema Editboundary

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Matemática

PRESENTA:

Leticia Angelica Ramírez Galicia

DIRECTOR DE TESIS:

Pablo Barrera Sánchez



2012



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Quiero agradecer a Dios por permitirme culminar esta parte de mi vida. A mis Padres, Hermanas Abuelitos y niños "capulin, oso, pinta", por el amor que me han dado o que en su momento recibí, son la inspiración de mis triunfos.

A toda mi familia, en especial a mi abuelita Catalina y mi tía Margarita Eliazar, porque he cumplido con lo que seguramente ellas hubiesen querido ver.

A las personas y animalitos que no están ya conmigo, porque me hicieron tener esos momentos imborrables en mi vida. Garfield, gracias por haberte quedado conmigo por las noches para trabajar en este anhelo alcanzado.

Dr. Pablo Barrera es mi modelo a seguir, por sus consejos que me han ayudado para formarme como persona y profesionista, por creer en mí.

M. en C. Guilmer González por el apoyo a este trabajo, correcciones, explicaciones que no pueden ser más entendibles, por darme tu apoyo y confianza.

Dr. Francisco Domínguez por los comentarios y el apoyo moral que recibí para poder culminar este trabajo.

Dra. Ursula Iturrarán por el tiempo brindado para revisar este trabajo, por darme tan valiosas observaciones.

M. en C. Luis Carlos Velázquez por las correcciones, por darme animos ya que cada vez que nos vemos me dices: todo va bien.

Javier Cortés por tu gran ayuda al dar una última revisión a este trabajo y así asegurar de que todas las correcciones de los sinodales estuvieran hechas. Eres parte importante en mi vida.

A mis compañeros del laboratorio de cómputo científico: César, Jorge, Javier, Gustavo, Francisco, Luis Carlos, por aprender de ustedes.

A mis amigos, por su amistad incondicional.

A SEP-PROMEP: Aplicaciones de la optimización numérica a la solución de diversos problemas de cómputo científico, por el apoyo económico que recibí para llevar a cabo este trabajo.

ANIDE: Academia Nacional de investigación y desarrollo A.C., por el apoyo económico que me fue entregado por medio del proyecto denominado: Desarrollo de un Modelo Matemático- Numérico- Computacional para la simulación de flujo multifásico reactivo en medio poroso fracturado asociado con procesos de combustión insitu.

Gracias.

Prefacio

En este trabajo estamos interesados en el tratamiento de contornos planos, es decir, simplificación, suavizamiento y parametrización. Recientemente se ha encontrado que la familia de funciones splines en tensión son de gran utilidad en diversas aplicaciones por lo que en nuestro trabajo se explora la posibilidad de usarlas en el tratamiento de contornos.

El enfoque que seguiremos con énfasis es construir spline en tensión que interpolen los contornos de interés y en especial a la elección del parámetro de tensión.

En esta presentación mencionamos los aspectos prácticos de la construcción de los interpolantes, para la fundamentación teórica se puede ver la bibliografía que se menciona en el texto.

En el capítulo 1 resolvemos el problema de construir una curva que pase por un conjunto de puntos predeterminados, es decir se resuelve el problema de interpolación usando splines lineales y cúbicos.

El capítulo 2 es exclusivo de los spline en tensión por lo prácticos que han resultado para interpolar datos. En el mismo se hace énfasis en la descripción del Algoritmo 716 y de las rutinas TSPACKGUI para Matlab, de Renka.

El capítulo 3 contiene el tema de interpolación paramétrica usando splines, se explica la importancia de la elección de la parametrización para el suavizamiento de la curva. Dentro de este también describimos como se aplica el spline en tensión para contornos cerrados.

El capítulo 4 lo hemos dedicado al sistema Editboundary propio del grupo UNAMALLA, se describe el diseño y funcionamiento del sistema para el tratamiento de contornos.

En el último capítulo se muestran las conclusiones y el trabajo futuro.

Todos los capítulos (excepto el capítulo 5) contienen figuras que hacen que el texto sea más entendible.

Esperemos que sea de alguna utilidad para el lector y sobre todo que lo motive a estudiar el tema más a fondo.

Índice general

| | |
|---|-----------|
| Prefacio | 1 |
| Prefacio | 3 |
| 1. Interpolación spline | 7 |
| 1.1. Interpolación | 7 |
| 1.1.1. Interpolación lineal por tramos | 8 |
| 1.2. Splines | 11 |
| 1.2.1. Interpolación con spline cúbico de Hermite | 14 |
| 1.2.2. Spline cúbico | 19 |
| 2. Spline en tensión | 25 |
| 2.1. Solución de la ecuación diferencial de Cline | 27 |
| 2.2. Normalizando el parámetro en tensión y condiciones de frontera | 31 |
| 2.3. Descripción del Algoritmo 716 | 43 |
| 2.4. Rutinas de TSPACKGUI para Matlab | 50 |
| 3. Interpolación paramétrica usando splines | 57 |
| 3.1. Elección del parámetro t de interpolación | 60 |
| 3.2. Spline en tensión para contornos cerrados | 64 |
| 4. Editboundary: Sistema para el tratamiento de contornos | 77 |
| 4.1. Diseño e interfase de Usuario | 77 |
| 4.1.1. Interfase con el usuario | 78 |
| 4.2. Estructura modular | 79 |
| 4.3. Módulo de contornos | 80 |
| 4.3.1. Construir un contorno en pantalla. | 80 |
| 4.3.2. Abrir un archivo de datos <code>Open</code> | 82 |

| | |
|--|------------|
| 4.3.3. Guardar información Save | 82 |
| 4.3.4. Tratamiento por reducción de puntos | 83 |
| 4.3.5. Tratamiento del contorno utilizando spline en tensión | 84 |
| 5. Conclusiones | 87 |
| 5.1. Conclusiones | 87 |
| A. Reducción de puntos | 89 |
| A.1. Criterios de colinealidad | 91 |
| A.1.1. Criterio de Error Máximo | 91 |
| A.1.2. Criterio de Longitud | 92 |
| A.1.3. Criterio de Cruce | 93 |
| A.1.4. Cómo se implementa el algoritmo de eliminación de puntos usando los criterios de colinealidad | 94 |
| Bibliografía | 111 |

Capítulo 1

Interpolación spline

1.1. Interpolación

La interpolación consiste en construir una función que pase por un conjunto de puntos prefijados.

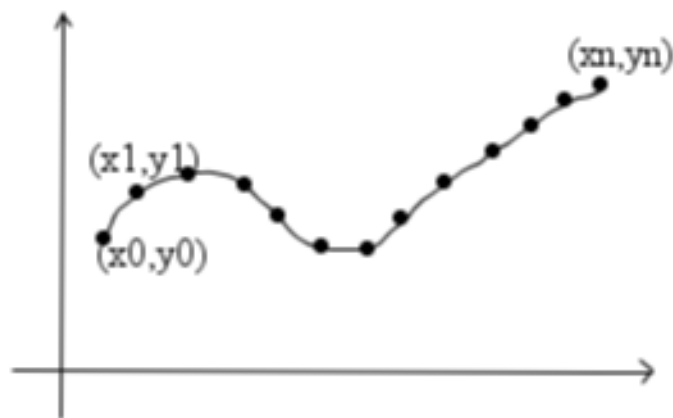


Figura 1.1: Interpolación

La interpolación es una herramienta usada en aplicaciones como en economía, medicina, biología, entre otros. En muchas ocasiones, profesionales de estas áreas se enfrentan al problema de disponer de una tabla de datos como la siguiente:

| | | | | | | |
|-----|-------|-------|-------|-------|---------|-------|
| x | x_0 | x_1 | x_2 | x_3 | \dots | x_n |
| y | y_0 | y_1 | y_2 | y_4 | \dots | y_n |

Uno de los objetivos es obtener información que no esta registrada en dicha tabla pero que se encuentra entre esos valores. La interpolación en la mayoría de los casos es la solución.

En la tabla anterior se describe la dependencia de la magnitud y con respecto a la magnitud x . Bajo esta idea la interpolación consiste en obtener el valor de la magnitud y para un x que no aparece en la tabla, entre dos valores x_i y x_{i+1} . Una forma de realizar dicho proceso es construir una función $f(x)$ que pase por los puntos (x_i, y_i) con $i = 1, \dots, n$, esa función $f(x)$ de cierta manera va a aproximar a los datos de la tabla.

La pregunta que nos resta es de qué forma será esa aproximación. Usualmente se eligen polinomios, spline, para nuestro caso será el spline en tensión. Para esto debemos definir una familia \mathcal{G} donde se encuentre la función $f(x)$ que aproximar los datos.

Es importante señalar, que si pedimos que $f(x)$ pase por los puntos (x_i, y_i) (es decir que f interpola a los puntos o que es un interpolante), solo tendremos control sobre ello, no lo que ocurra con $f(x)$ para x en (x_i, x_{i+1}) . Para el caso de que \mathcal{G} sea una familia de polinomios, podría haber oscilaciones [4], nosotros utilizaremos polinomios de grado pequeño en cada intervalo. A continuación describiremos las ideas.

1.1.1. Interpolación lineal por tramos

La solución inmediata es unir dos puntos (x_i, y_i) con (x_{i+1}, y_{i+1}) por un segmento de línea, esto es, generar una función f compuesta por tramos de recta, lo que se conoce como un spline de interpolación lineal [1].

Dados $n + 1$ puntos $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, la interpolación lineal por tramos consiste en hallar una función de grado menor o igual a uno que pase por cada dos nodos consecutivos. Con $n + 1$ puntos existen n intervalos y por tanto, $2n$ incógnitas por evaluar, como condición adicional queremos que las rectas que son generadas por las funciones de grado menor o igual a uno en cada intervalo se unan una con otra.

La gráfica 1.2 muestra el spline lineal de interpolación, notemos que los puntos se unen mediante segmentos de recta obteniendo de este modo una función lineal por tramos que podemos evaluar en cualquier punto intermedio. Usualmente a los puntos que unen los segmentos de línea se les llama “puntos de ruptura”.

Nuestro interés es el intervalo $[x_0, x_n]$, sobre este, contamos con $n + 1$ puntos (x_i, y_i) desde $i = 0, \dots, n$, donde los y_i son los valores de x_i de cierta función desconocida $g(x)$. Sobre cada segmento podemos definir una recta de la forma:

$$y = P_i(x) = a_i x + b_i$$

La idea es aproximar la gráfica de $g(x)$ en forma continua; es decir, que cada recta en los puntos de ruptura se unan, en notación:

$$P_i(x_{i+1}) = P_{i+1}(x_{i+1})$$

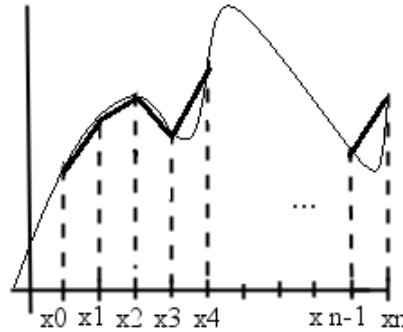


Figura 1.2: Interpolación lineal

Por otra parte cada $P_i(x)$ interpola a x_i en y_i , esto es

$$P_i(x_i) = a_i x + b_i = y_i,$$

es necesario que ningún par de x_i se repita. Lo que se está haciendo es aproximar a $g(x)$ por una función $f(x)$ lineal por tramos.

La ecuación de una recta que pasa por dos puntos (x_i, y_i) y (x_{i+1}, y_{i+1}) es de la forma:

$$(y - y_i) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} (x - x_i)$$

despejando a y obtenemos

$$y = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i} (x - x_i)$$

Por lo tanto

$$P_i(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i} (x - x_i) \quad \text{para } x_i \leq x \leq x_{i+1}$$

Con esto ya tenemos una expresión analítica para cada uno de los polinomios lineales $P_i(x)$ que constituyen el spline lineal f . En muchas aplicaciones es suficiente usar el spline lineal por ser sencillo y económico figura 1.3.

En la figura 1.4 observamos que puede no ser suficiente usar un spline lineal ya que pierde información en los cambios de curvatura.

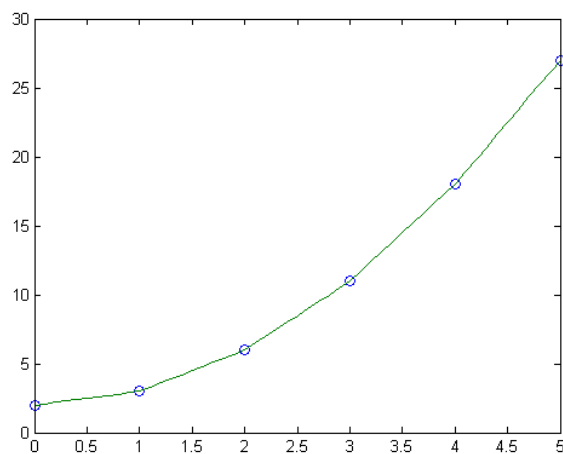


Figura 1.3: Ejemplo de interpolación lineal

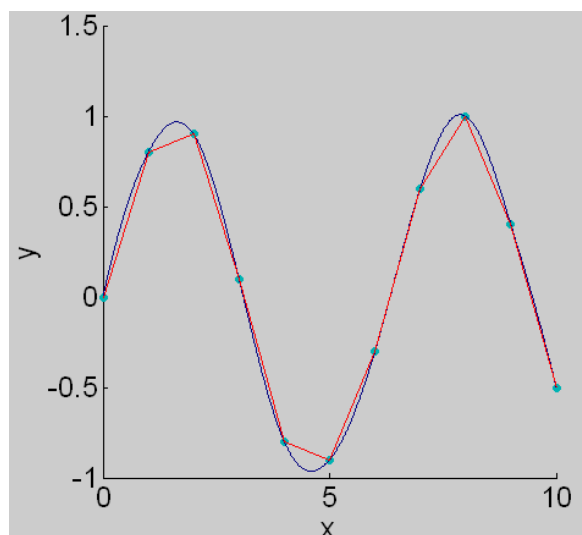


Figura 1.4: Se pierde información en los cambios de curvatura

En la siguiente sección nos abocaremos a un spline de orden mayor, particularmente aquellos que tienen grado menor o igual a tres.

1.2. Splines

En general, para $n + 1$ puntos $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, la interpolación polinomial por tramos consiste en hallar una función de grado menor o igual a k que pase por cada dos puntos consecutivos. Bajo esta idea un spline es el conjunto de polinomios definidos en cada intervalo.

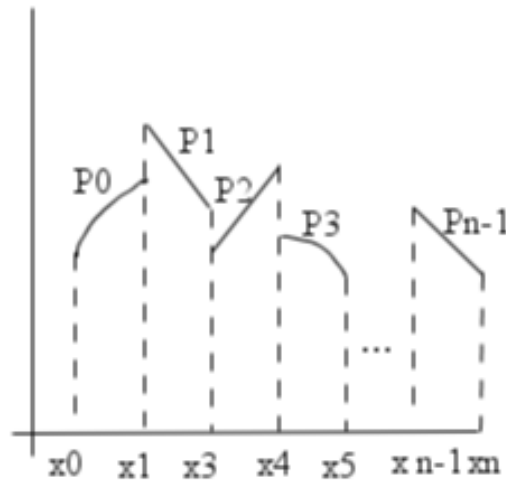


Figura 1.5: Representación gráfica de un spline cúbico

Estamos interesados en un cierto tipo de spline, aquellos polinomios que se “peguen adecuadamente”, es decir, que tengan continuidad y continuidad diferenciable.

Trabajaremos sobre $n + 1$ puntos: $a = x_0 < x_1 < \dots < x_n = b$, estos definen una partición \mathcal{P} en la cual se encontrara el spline, formando subintervalos $[x_i, x_{i+1})$ desde $i = 0, \dots, n$, sobre cada uno de ellos definimos un polinomio P_i de grado menor o igual a k .

Los spline más utilizados tienen las siguientes características :

1. f es continua en $[x_i, x_{i+1})$
2. f tiene derivada continua en $[x_i, x_{i+1}]$.
3. f tiene segunda derivada continua en $[x_i, x_{i+1}]$.

esto último representa que los polinomios se peguen suavemente entre uno y otro.

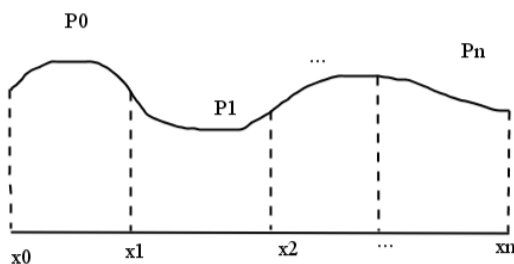


Figura 1.6: spline cúbico

Definimos el conjunto $F(\mathcal{P}, k)$ de funciones polinomiales por cachos sobre la partición \mathcal{P} y de grado menor o igual a k como:

$$F(\mathcal{P}, k) = \{f : [x_0, x_n] \longrightarrow \mathbb{R} \mid f(x)|_{[x_i, x_{i+1}]} \text{ y el grado de } P_i \leq k\} = P_i(x), i = 1, \dots, n$$

esta familia es un espacio vectorial, es fácil ver que su dimensión es:

$$\dim(F(\mathcal{P}, k)) = n * (k + 1)$$

donde n es el número de intervalos y $k + 1$ es el número de coeficientes de cada polinomio. Bajo esta condición $F(\mathcal{P}, k)$ engloba a todos los spline posibles.

Ahora la subfamilia de F , con la característica de que sea continua, la escribimos:

$$F(\mathcal{P}, k, C^0) = \{f(x) \in F(\mathcal{P}, k) \mid f(x) \in C[a, b]\}$$

esto implica que:

$$P_i(x_i) = P_{i+1}(x_i) \quad i = 1, \dots, n$$

por eso, la dimensión de este espacio es:

$$\dim(F(\mathcal{P}, k, C^0)) = n(k + 1) - (n - 1) = kn + 1$$

ya que al elegir la continuidad se pierde un grado de libertad.

Ahora, si pedimos que F sea continuamente diferenciable, el espacio del spline F se escribe

$$F(\mathcal{P}, k, C') = \{f(x) \in F(\mathcal{P}, k, C^0) \mid f(x) \in C'[a, b]\}$$

esto implica que:

$$P'_i(x_i) = P'_{i+1}(x_i) \quad i = 1, \dots, n$$

Es fácil ver, que la dimensión de este espacio es:

$$\dim(F(\mathcal{P}, k, C')) = n(k + 1) - 2(n - 1) = kn - n + 2$$

ya que al elegir derivada continua se pierde un grado más de libertad.

Nuevamente, si pedimos que F tenga segunda derivada continua, es decir

$$P_i''(x_i) = P_{i+1}''(x_i) \quad i = 1, \dots, n$$

El espacio donde habita el spline se escribe

$$F(\mathcal{P}, k, C^2) = \{f(x) \in F(\mathcal{P}, k, C^0) | f(x) \in C^2[a, b]\}$$

Es fácil ver que la dimensión de este espacio es:

$$\dim(F(\mathcal{P}, k, C^2)) = n(k + 1) - 3(n - 1) = kn - 2n + 3$$

ya que al elegir segunda derivada continua, nuevamente se pierde un grado más de libertad.

Ahora bien, si a F le imponemos que tenga tercera derivada continua en los puntos de ruptura, es decir

$$P_i'''(x_i) = P_{i+1}'''(x_i), \quad i = 1, \dots, n - 1$$

el espacio donde habita este spline se escribe

$$F(\mathcal{P}, k, C^3) = \{f(x) \in F(\mathcal{P}, k, C^2) | f(x) \in C^3[a, b]\}$$

y es fácil ver que

$$\dim(F(\mathcal{P}, k, C^3)) = n(k + 1) - 4(n - 1) = kn - 3n + 4$$

Análogamente a los casos anteriores se pierde un grado más de libertad.

A continuación mostraremos la siguiente tabla, en ella se indica la dimensión de la función spline con distintos grados y de sus subfamilias de funciones continuas y continuas diferenciables:

| Grado de f | $\dim(F(\mathcal{P}, k))$ | $\dim(F(\mathcal{P}, k, C^0))$ | $\dim(F(\mathcal{P}, k, C^1))$ | $\dim(F(\mathcal{P}, k, C^2))$ | $\dim(F(\mathcal{P}, k, C^3))$ |
|--------------|---------------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|
| $k = 0$ | n | 1 | | | |
| $k = 1$ | $2n$ | $n + 1$ | 2 | | |
| $k = 2$ | $3n$ | $2n + 1$ | $n + 2$ | 3 | |
| $k = 3$ | $4n$ | $3n + 1$ | $2n + 2$ | $n + 3$ | 4 |

Tabla 1.1: Grados de libertad para polinomios de grado ≤ 3 .

De acuerdo a la tabla 1.1, si consideramos el spline de grado $k = 3$, tendremos $4n$ grados de libertad para determinarlo; sin embargo, al exigirle mayor suavidad, en sus derivadas el número de grado de libertad es cada vez menor:

- (a) $3n + 1$ si pedimos que sea continua.
- (b) $2n + 2$ si pedimos primera derivada continua es decir si se le pide suavidad.
- (c) $n + 3$ si pedimos segunda derivada continua (aún más suavidad).
- (d) 4 si pedimos tercera derivada continua (extrema suavidad).

Este último caso no sirve de mucho para resolver el problema que nos interesa porque perdemos flexibilidad para definir la familia, ya que solo tendríamos 4 grados de libertad, es decir el spline se reduce a un polinomio cúbico.

1.2.1. Interpolación con spline cúbico de Hermite

El problema es simple, queremos una función f que interpole los puntos y que en los puntos tenga cierta derivada:

| | | | | | | |
|------|-------|-------|-------|-------|---------|-------|
| x | x_0 | x_1 | x_2 | x_3 | \dots | x_n |
| g | g_0 | g_1 | g_2 | g_3 | \dots | g_n |
| g' | s_0 | s_1 | s_2 | s_3 | \dots | s_n |

Se tiene $n + 1$ datos que son el resultado de evaluar los puntos x_i en la función g y $n + 1$ datos de la evaluación en g' , y con ellos queremos un spline cúbico.

La dimensión (grados de libertad) de la función f con $k = 3$ es $\dim(f(\mathcal{P}, k)) = 4n$, pero recordamos que una de las condiciones del problema es que f coincida con g , es decir que f sea continua de aquí la $\dim(f(\mathcal{P}, k, C^0)) = 3n + 1$, ya que al cumplir esta condición perdemos un grado de libertad, así mismo como se pide que f coincida con g' se pierde nuevamente otro grado de libertad, entonces la $\dim(f(\mathcal{P}, k, C^1)) = 2n + 2$, lo cual nos lleva a concluir que como se tienen $2(n + 1)$ datos del problema y la $\dim(f(\mathcal{P}, k, C^1)) = 2n + 2$, entonces el problema tiene solución única.

En la interpolación cúbica de Hermite se cuenta con cada punto y su valor en la función, al igual que su derivada, veremos que es fácil calcular el polinomio $P_i(x)$, que forma el spline interpolante. Por conveniencia, se realiza el siguiente cambio de variable:

$$y(x) = \frac{x - x_i}{x_{i+1} - x_i}, \quad x_i \leq x \leq x_{i+1}$$

y comprobamos, si $x = x_i$, tenemos:

$$y(x_i) = \frac{x_i - x_i}{x_{i+1} - x_i} = 0$$

si $x=x_{i+1}$, tenemos:

$$y(x_{i+1}) = \frac{x_{i+1} - x_i}{x_{i+1} - x_i} = 1$$

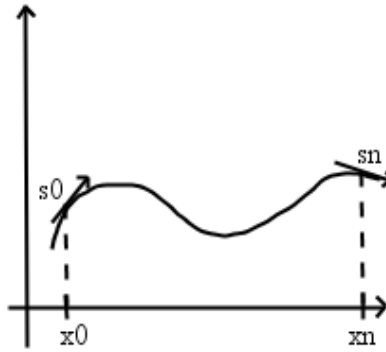


Figura 1.7: spline cúbico

La idea es hacer calculos sobre el intervalo $[0,1]$ y posteriormente generalizar al intervalo $[x_i, x_{i+1}]$.

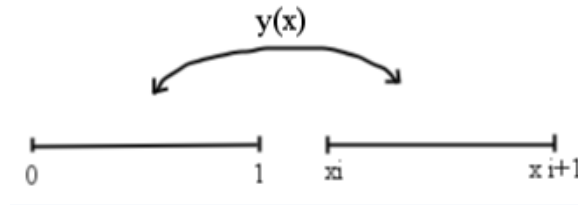


Figura 1.8: Spline

El problema que se plantea ahora es, cómo determinar un polinomio cúbico $P(y)$ tal que $P(0) = P_0, P(1) = P_1, P'(0) = P'_0$ y $P'(1) = P'_1$.

El polinomio $P(y)$ se puede escribir de la forma

$$P(y) = P_0q_1(y) + P_1q_2(y) + P'_0q_3(y) + P'_1q_4(y)$$

donde q_1, q_2, q_3 y q_4 son polinomios cúbicos:

$$\begin{aligned} q_1(0) &= 1, q_1'(0) = 0, q_1(1) = 0, q_1'(1) = 0 \\ q_2(0) &= 0, q_2'(0) = 1, q_2(1) = 0, q_2'(1) = 0 \\ q_3(0) &= 0, q_3'(0) = 0, q_3(1) = 1, q_3'(1) = 0 \\ q_4(0) &= 0, q_4'(0) = 0, q_4(1) = 0, q_4'(1) = 1 \end{aligned}$$

sea

$$\begin{aligned} q_1(y) &= ay^3 + by^2 + cy + d \\ q_1'(y) &= 3ay^2 + 2by + c \end{aligned}$$

evaluando con $y = 0$ y $y = 1$ obtenemos:

$$\begin{aligned} q_1(0) &= d = 1 \\ q_1(1) &= a + b + c + d = 0 \\ q_1'(0) &= c = 0 \\ q_1'(1) &= 3a + 2b + c = 0 \end{aligned}$$

resolviendo el sistema obtenemos que $a = 2, b = -3, c = 0$ y $d = 1$, entonces $q_1(y)$ es de la forma:

$$q_1(y) = 2y^3 - 3y^2 + 1 = (1 + 2y)(y^2 - 2y + 1) = (1 + 2y)(y - 1)^2.$$

Ahora, $q_2(y)$ se puede escribir

$$\begin{aligned} q_2(y) &= ay^3 + by^2 + cy + d \\ q_2'(y) &= 3ay^2 + 2by + c \end{aligned}$$

evaluando con $y = 0$ y $y = 1$ obtenemos:

$$\begin{aligned} q_2(0) &= d = 0 \\ q_2(1) &= a + b + c + d = 0 \\ q_2'(0) &= c = 1 \\ q_2'(1) &= 3a + 2b + c = 0 \end{aligned}$$

resolviendo el sistema obtenemos que $a = 1, b = -2$ y $c = 1$, entonces $q_2(y)$ es de la forma:

$$q_2(y) = y^3 - 2y^2 + y = (y - 1)^2 y$$

Ahora, $q_3(y)$ se puede escribir en la forma:

$$\begin{aligned} q_3(y) &= ay^3 + by^2 + cy + d \\ q_3'(y) &= 3ay^2 + 2by + c \end{aligned}$$

evaluando con $y = 0$ y $y = 1$ obtenemos:

$$\begin{aligned} q_3(0) &= d = 0 \\ q_3(1) &= a + b + c + d = 1 \\ q_3'(0) &= c = 0 \\ q_3'(1) &= 3a + 2b + c = 0 \end{aligned}$$

resolviendo el sistema obtenemos que $a = -2, b = 3$, entonces $q_3(y)$ es de la forma:

$$q_3(y) = -2y^3 + 3y^2 = 3y^2 - 2y^3 = y^2(3 - 2y)$$

Por último $q_4(y)$

$$\begin{aligned} q_4(y) &= ay^3 + by^2 + cy + d \\ q_4'(y) &= 3ay^2 + 2by + c \end{aligned}$$

evaluando con $y = 0$ y $y = 1$ obtenemos:

$$\begin{aligned} q_4(0) &= d = 0 \\ q_4(1) &= a + b + c + d = 0 \\ q_4'(0) &= c = 0 \\ q_4'(1) &= 3a + 2b + c = 1 \end{aligned}$$

resolviendo el sistema obtenemos que $a = 1, b = -1$, entonces $q_4(y)$ es de la forma:

$$q_4(y) = y^3 - y^2 = y^2(y - 1)$$

Así $P(y)$ queda de la forma:

$$P(y) = P_0(1 + 2y)(y - 1)^2 + P_1 y^2(3 - 2y) + P_0'(y - 1)^2 y + P_1' y^2(y - 1)$$

hemos encontrado el polinomio cúbico $P(y)$ para el intervalo $[0,1]$. Ahora bien tengamos en cuenta que las condiciones de interpolación cúbica de Hermite a trozos son las siguientes:

$$\begin{aligned} P_i(x_i) &= g_i \\ P'_i(x_i) &= s_i \\ P_i(x_{i+1}) &= g_{i+1} \\ P'_i(x_{i+1}) &= s_{i+1} \end{aligned}$$

El problema de interpolación cúbica de Hermite nos pide construir una función f tal que es un conjunto de polinomios P_i , es decir $f(x)|_{[x_i, x_{i+1}]}$. Lo que se trata ahora es obtener la expresión de $P_i(x)$ a partir de lo que se obtuvo de $P(y)$.

Notemos que si $P_i(x_i) = P(y_i(x_i))$ entonces:

$$P'_i(x) = \frac{d}{dx} P(y(x)) = P'(y)y' = P'(y) \frac{1}{x_{i+1} - x_i}$$

entonces

$$\begin{aligned} P_i(x_i) &= P(y_i(x_i)) = P(0) = g_i \\ P'_i(x_i) &= P'(0) \frac{1}{x_{i+1} - x_i} = s_i, \quad \text{entonces} \quad P'(0) = s_i(x_{i+1} - x_i) \\ P_i(x_{i+1}) &= P(y_i(x_{i+1})) = P(1) = g_{i+1} \\ P'_i(x_{i+1}) &= P'(1) \frac{1}{x_{i+1} - x_i} = s_{i+1}, \quad \text{entonces} \quad P'(1) = s_{i+1}(x_{i+1} - x_i) \end{aligned}$$

obtenemos

$$P_i(x) = g_i q_1(y) + g_{i+1} q_2(y) + s_i(x_{i+1} - x_i) q_3(y) + s_{i+1}(x_{i+1} - x_i) q_4(y)$$

sustituyendo se tiene:

$$\begin{aligned} P_i(x) = & g_i \frac{(x - x_{i+1})^2(2(x - x_i) + (x_{i+1} - x_i))}{(x_{i+1} - x_i)^3} + g_{i+1} \frac{(x - x_i)^2(3(x_{i+1} - x_i) + 2(x_i - x))}{(x_{i+1} - x_i)^3} + \\ & s_i \frac{(x - x_{i+1})^2(x - x_i)}{(x_{i+1} - x_i)^2} - s_{i+1} \frac{(x_i - x)^2(x_{i+1} - x)}{(x_{i+1} - x_i)^2} \end{aligned}$$

Por tanto la expresión de $P_i(x)$ anterior es la solución de la interpolación cúbica de Hermite a trozos.

En todo el proceso anterior, para construir el interpolante cúbico de Hermite, se conocen los valores de g' en los puntos de interpolación. Pero qué pasaría si no se conocieran dichos valores, entonces las tangentes s_i se pueden elegir de manera que el interpolante f tenga hasta segunda derivada continua. Si esto pasa se dice que f es un spline cúbico de interpolación.

1.2.2. Spline cúbico

Un spline cúbico es una función formada por polinomios cúbicos, los cuales se unen con la mayor suavidad posible [1]. Dados los puntos:

| | | | | | | |
|-----|-------|-------|-------|-------|---------|-------|
| x | x_0 | x_1 | x_2 | x_3 | \dots | x_n |
| g | g_0 | g_1 | g_2 | g_3 | \dots | g_n |

solo se tiene los valores de g para los puntos x_i . Con los anteriores datos se quiere construir una función $f : [a, b] \rightarrow \mathbb{R}$ que cumpla con las siguientes condiciones:

$$\begin{aligned} P_i(x_i) &= P_{i+1}(x_i) \\ P'_i(x_i) &= P'_{i+1}(x_i) \quad i = 1, \dots, n-1 \\ P''_i(x_i) &= P''_{i+1}(x_i) \end{aligned}$$

Notemos que en la tabla de valores tenemos $n+1$ datos, que son el resultado de evaluar los x_i en la función g . La dimensión (grados de libertad) de f con respecto a su grado de k y al número de subintervalos (la partición \mathcal{P}) es $\dim(f(P, k)) = 4n$, si en una de las condiciones se pide que f sea continua entonces $\dim(f(P, k, C^0)) = 3n + 1$, de la misma manera se pide que f sea continuamente diferenciable de aquí la $\dim(f(P, k, C^1)) = 2n + 2$ y por último se pide que f tenga doble derivada continua, tendremos entonces $\dim(f(P, k, C_2)) = n + 3$. Cuando se pide que sea continua, continua diferenciable y con doble derivada continua, se pierde un grado de libertad a medida que se vayan cumpliendo cada una de las condiciones anteriores. Notemos que se tiene $n+1$ datos del problema contra $\dim(f(P, k, C_2)) = n + 3$, haciendo la diferencia de estos nos percatamos que faltan dos condiciones $((n + 3) - (n + 1) = 2)$ para poder dar solución a este problema.

Recordemos que cuando obtuvimos el interpolante cúbico de Hermite llegamos a una expresión de la forma:

$$P_i(x) = g_i \frac{(x - x_{i+1})^2(2(x - x_i) + (x_{i+1} - x_i))}{(x_{i+1} - x_i)^3} + g_{i+1} \frac{(x - x_{i+1})^2(3(x_{i+1} - x_i)^2 + 2(x_i - x))}{(x_{i+1} - x_i)^3}$$

$$+s_i \frac{(x - x_{i+1})^2(x - x_i)}{(x_{i+1} - x_i)^2} + s_{i+1} \frac{(x_i - x)^2(x_{i+1} - x)}{(x_{i+1} - x_i)^2}$$

en donde se conocen los valores de x_i evaluados en g y g' , pero en el problema del spline cúbico de interpolación solo sabemos el valor de g para lo x_i y las tangentes son incógnitas, se propone trabajar con $P_i(x)$ (interpolador cúbico de Hermite), solo que ahora se considerarán a los s_i como valores no conocidos.

Derivamos dos veces al interpolante cúbico y obtenemos:

$$P_i''(x) = 6 \frac{g_{i+1} - g_i}{(x_{i+1} - x_i)^3} (x_{i+1} + x_i - 2x) - 2 \frac{s_i}{(x_{i+1} - x_i)^2} (2x_{i+1} + x_i - 3x) \\ - 2 \frac{s_{i+1}}{(x_{i+1} - x_i)^2} (2x_i + x_{i+1} - 3x)$$

Como f es una función formada por un conjunto de polinomios cúbicos, los únicos puntos posibles en los que f'' puede ser discontinua es en los x_i donde se unen los polinomios cúbicos. Entoces la continuidad de f'' es equivalente a pedir que:

$$f''(x_i^-) = f''(x_i^+)$$

donde

$$f''(x_i^+) = P_i''(x)$$

y

$$f''(x_i^-) = P_{i-1}''(x)$$

aquí es donde utilizamos la segunda derivada del interpolante cúbico de Hermite, evaluamos el x_i por la izquierda y por la derecha, obteniendo:

$$f''(x_i^+) = P_i''(x_i) = 6 \frac{g_{i+1} - g_i}{(x_{i+1} - x_i)^2} - 4 \frac{s_i}{x_{i+1} - x_i} - 2 \frac{s_{i+1}}{x_{i+1} - x_i}$$

$$f''(x_i^-) = P_{i-1}''(x_i) = -6 \frac{g_i - g_{i-1}}{(x_i - x_{i-1})^2} + 2 \frac{s_{i-1}}{x_i - x_{i-1}} + 4 \frac{s_i}{x_i - x_{i-1}}$$

hacemos $f''(x_i^-) = f''(x_i^+)$, como se dijo anteriormente esto garantiza que f'' sea continua y al mismo se obtiene la expresión:

$$2s_i + \frac{x_i - x_{i-1}}{x_i - x_{i-1} + x_{i+1} - x_i} s_{i+1} + \frac{x_{i+1} - x_i}{x_i - x_{i-1} + x_{i+1} - x_i} s_{i-1} =$$

$$= 3 \frac{(x_{i+1} - x_i)(g_i - g_i - 1)}{(x_i - x_{i-1})(x_i - x_{i-1} + x_{i+1} - x_i)} + \frac{(x_i - x_{i-1})(g_{i+1} - g_i)}{(x_{i+1} - x_i)(x_i - x_{i-1} + x_{i+1} - x_i)}$$

sea

$$d_i = 2$$

$$a_i = \frac{x_{i+1} - x_i}{(x_i - x_{i-1} + x_{i+1} - x_i)}$$

$$b_i = 3 \frac{(x_{i+1} - x_i)(g_i - g_i - 1)}{(x_i - x_{i-1})(x_i - x_{i-1} + x_{i+1} - x_i)} + \frac{(x_i - x_{i-1})(g_{i+1} - g_i)}{(x_{i+1} - x_i)(x_i - x_{i-1} + x_{i+1} - x_i)}$$

$$1 - a_i = \frac{x_i - x_{i-1}}{x_i - x_{i-1} + x_{i+1} - x_i}$$

la expresión queda reducida a:

$$d_i s_i + (1 - a_i) s_{i+1} + a_i s_{i-1} = b_i$$

o bien,

$$a_i s_{i-1} + d_i s_i + (1 - a_i) s_{i+1} = b_i, \quad \text{para } i = 0, \dots, n$$

Suponiendo que s_1 y s_n son dadas, obtenemos un sistema de $n-1$ ecuaciones lineales para calcular $n-1$ incógnitas. La matriz de este sistema es tridiagonal y diagonalmente dominante ya que $d_i=2$ y $0 \leq a_i < 1$, por tanto tiene una solución única. Los valores de s_1 y s_n depende de las condiciones de frontera que se elijan. Para el spline cúbico se hará mención de dos tipos de condiciones de frontera que dependiendo de cual se ocupe tendremos un spline cúbico completo o spline cúbico natural [1].

Condiciones de frontera para el spline cúbico

Spline cúbico completo

Por nuestra parte, si conocemos el valor de g' en x_0 y x_n podemos utilizar estos valores como condiciones de frontera del spline cúbico, así tendremos el spline cúbico completo.

$$s_0 = g'(x_0)$$

y

$$s_{1n} = g'(x_n)$$

En este caso el spline cúbico no solo interpola a la función g en los puntos x_0, \dots, x_n si no que además interpola a g' en x_0 y x_n entonces s_0, \dots, s_n se calcula como solución del sistema de ecuaciones lineales.

$$\begin{pmatrix} 1 & & & & 0 \\ a_1 & d_1 & 1 - a_1 & & \\ \cdot & & & & \\ & & \cdot & & \\ & a_{n-1} & d_{n-1} & 1 - a_{n-1} & \\ 0 & & & & 1 \end{pmatrix}$$

donde

$$s_0 = g'(x_0) = b_0$$

$$s_n = g'(x_n) = b_n$$

En la siguiente figura mostramos un ejemplo en el cual se aplica spline cúbico completo a la función $g(x) = \frac{1}{(1+x^2)}$

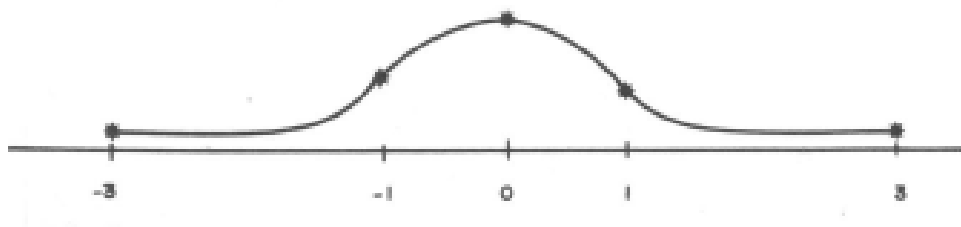


Figura 1.9: spline cúbico completo que interpola a la función $g(x) = \frac{1}{(1+x^2)}$ [1].

Spline cúbico natural

Este tipo de spline consiste en exigir que la segunda derivada del spline en los extremos sea cero

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

donde $f''(x_0)$ y $f''(x_n)$ quedan de la siguiente forma:

$$f''(x_0) = \frac{6(g_1 - g_0)^2}{(x_1 - x_0)} - \frac{4s_0}{(x_1 - x_0)} - \frac{2s_1}{(x_1 - x_0)} = 0$$

$$\frac{6(g_1 - g_0)}{(x_1 - x_0)} = 4s_0 + 2s_1$$

$$\frac{6(g_1 - g_0)}{(x_1 - x_0)} = 2(2s_0 + s_1)$$

$$\frac{3(g_1 - g_0)}{(x_1 - x_0)} = 2s_0 + s_1$$

y

$$f''(x_n) = \frac{6(g_n - g_{n-1})^2}{(x_n - x_{n-1})} - \frac{4s_n}{(x_n - x_{n-1})} - \frac{2s_{n-1}}{(x_n - x_{n-1})} = 0$$

$$\frac{6(g_n - g_{n-1})}{(x_n - x_{n-1})} = 4s_n + 2s_{n-1}$$

$$\frac{6(g_n - g_{n-1})}{(x_n - x_{n-1})} = 2(2s_n + s_{n-1})$$

$$\frac{3(g_n - g_{n-1})}{(x_n - x_{n-1})} = 2s_n + s_{n-1}$$

sumando las condiciones anteriores al sistema de ecuaciones lineales del spline cúbico de interpolación, obtenemos la siguiente matriz

$$\begin{pmatrix} 2 & 1 & & & & 0 \\ a_1 & d_1 & 1 - a_1 & & & \\ \cdot & & & & & \\ & & & & & \\ & & & & & \\ & a_{n-1} & d_{n-1} & 1 - a_{n-1} & & \\ 0 & & & & 1 & 2 \end{pmatrix}$$

La imposición arbitraria de la condición anterior puede provocar que cerca de los extremos x_1 y x_n el error aumente a menos que $g''(x_1) = g''(x_n) = 0$ [1].

En la figura 1.10 se ilustra un ejemplo en donde se aplica spline cúbico natural a la función

$$g(x) = x_+^0 = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

| | | | | |
|-------|--------|-------|-------|--------|
| x | -2 | -1 | 0 | 2 |
| g(x) | 0 | 0 | 1 | 1 |
| g'(x) | -0.304 | 0.609 | 0.869 | -0.435 |

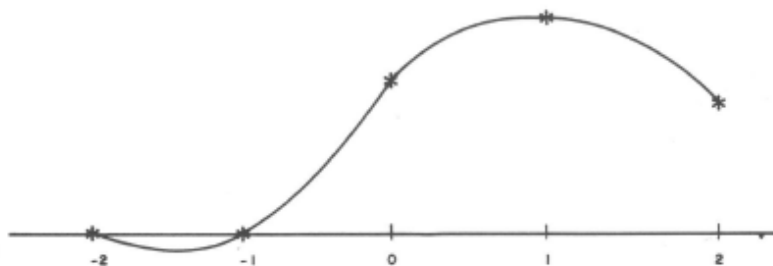


Figura 1.10: spline cúbico natural que interpola a la función $g(x) = x_+^0$ [1].

Capítulo 2

Spline en tensión

Hemos observado con anterioridad que la interpolación lineal por tramos muestra un ejemplo en donde los errores de interpolación son grandes pero aceptables en otras partes de la gráfica. Al igual en el caso cúbico hay oscilaciones en la gráfica que no se esperan, como en la figura 2.1, la función es monótona decreciente y sin embargo al aplicar spline cúbico aparecen oscilaciones “no esperadas”.

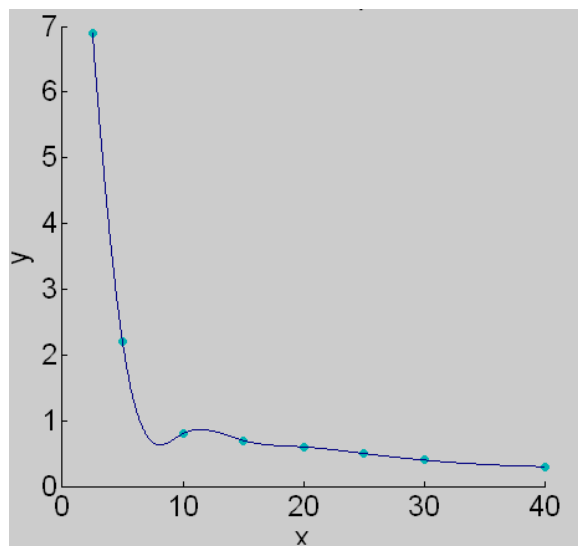


Figura 2.1: Efectividad contra costos. Esta gráfica está hecha de segmentos de funciones cúbicas pegadas una a otra [4].

Una de las primeras personas que estudia los spline en tensión es Schweikert [4], quien observa que los spline cúbicos producen oscilaciones, argumentando que esto se debe a los términos cuadráticos y cúbicos del polinomio.

Un spline cúbico está representado por una familia de funciones $\{f_i(x)\}$ de la forma:

$$f_i(x) = a_i + b_i x + c_i x^2 + d_i x^3$$

Schweikert propuso que en lugar de usar términos cuadráticos y cúbicos, se utilicen exponenciales, en la forma:

$$f_i(x) = a_i + b_i x + c_i \cosh(\sigma x) + d_i \sinh(\sigma x)$$

donde σ es un parámetro que permite controlar la forma y se le llama parámetro de tensión. La idea de esto fue reproducir la forma cuadrática y cúbica del spline pero controlando su curvatura. La manera en que se representa la forma cuadrática y cúbica es usando las funciones hiperbólicas $\sinh x$ y $\cosh x$, en la siguiente gráfica podemos apreciar el efecto σ por algunos valores de este.

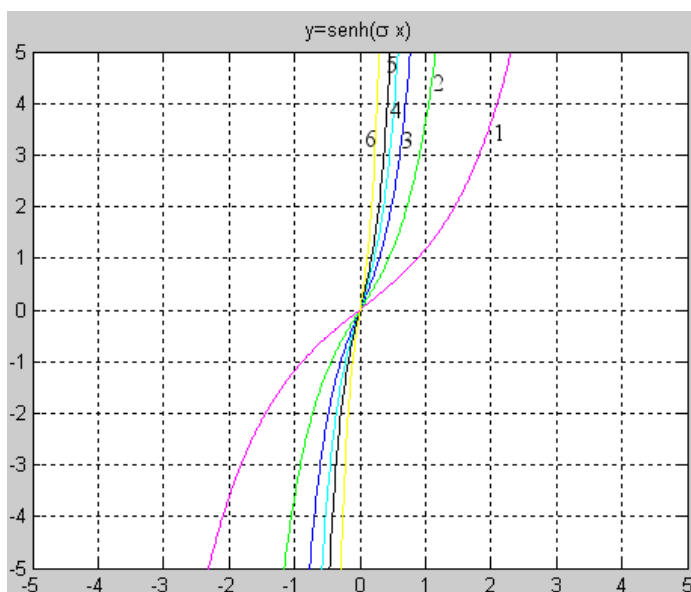


Figura 2.2: Gráfica de las funciones hiperbólicas $\sinh x$. 1) $\sigma = 1$, 2) $\sigma = 2$, 3) $\sigma = 3$, 4) $\sigma = 4$, 5) $\sigma = 5$, 6) $\sigma = 8$

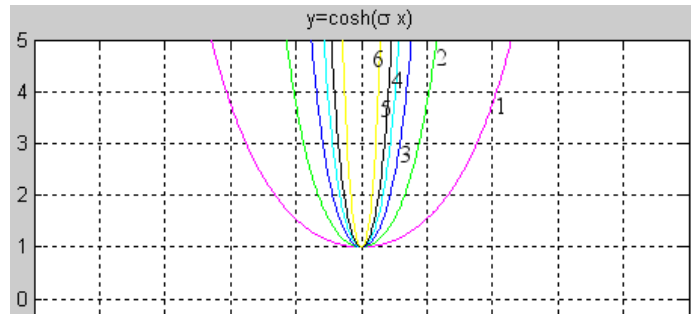


Figura 2.3: Gráficas de las funciones hiperbólicas $\cosh x$. 1) $\sigma = 1$, 2) $\sigma = 2$, 3) $\sigma = 3$, 4) $\sigma = 4$, 5) $\sigma = 5$, 6) $\sigma = 8$

Cline en 1974 propone controlar la forma del spline cúbico imponiendo condiciones a la curvatura y a la función spline [4]. Expresa al spline de tensión por medio de una ecuación diferencial:

$$f_i''(x) - \sigma^2 f_i(x) = \alpha_i x + \beta_i \quad (2.1)$$

donde α_i y β_i son desconocidas.

En adelante nos avocaremos a resolver dicha ecuación.

2.1. Solución de la ecuación diferencial de Cline

Ahora demos solución a la ecuación diferencial que origina el spline en tensión. Dada la ecuación lineal

$$f_i''(x) - \sigma^2 f_i(x) = \alpha_i x + \beta_i \quad x_i \leq x \leq x_{i+1}$$

donde α_i y β_i son desconocidas.

En los extremos $[x_i, x_{i+1}]$ tenemos que:

$$x = x_i$$

$$f_i''(x_i) - \sigma^2 f_i(x_i) = \alpha_i(x_i) + \beta_i \quad (2.2)$$

y $x = x_{i+1}$

$$f_i''(x_{i+1}) - \sigma^2 f_i(x_{i+1}) = \alpha_i(x_{i+1}) + \beta_i$$

hagamos

$$r_i = \alpha_i(x_i) + \beta_i = f_i''(x_i) - \sigma^2 f_i(x_i)$$

$$r_{i+1} = \alpha_i(x_{i+1}) + \beta_i = f_i''(x_{i+1}) - \sigma_i^2 f(x_{i+1})$$

Esto nos lleva a expresar :

$$\alpha_i = \frac{r_{i+1} - r_i}{x_{i+1} - x_i}$$

$$\beta_i = r_i - \frac{r_{i+1} - r_i}{x_{i+1} - x_i} x_i$$

hagamos

$$h_i = x_{i+1} - x_i$$

bajo esto, α_i y β_i quedan expresados como:

$$\alpha_i = \frac{r_{i+1} - r_i}{h_i}$$

y

$$\beta_i = r_i - \frac{r_{i+1} - r_i}{h_i} x_i$$

sustituyendo α_i , β_i , r_i y r_{i+1} en (2.2)

$$f_i''(x) - \sigma^2 f_i(x) = (\alpha_i(x_i) + \beta_i) \frac{x_{i+1} - x}{h_i} + (\alpha_i(x_{i+1}) + \beta_i) \frac{x - x_i}{h_i} \quad x_i \leq x \leq x_{i+1}$$

y donde α_i , β_i dependen de $f_i''(x_i)$, $f_i''(x_{i+1})$, $f_i(x_i)$, $f_i(x_{i+1})$.

La solución homogénea de esta ecuación diferencial se puede ver como combinación lineal de las funciones e^{mx} . Vamos a resolver dicha ecuación diferencial.

La solución va a estar expresada como la solución particular f_p más la solución homogénea f_h . La solución homogénea la obtenemos de la siguiente forma:

$$f_h'' = m^2 e^{mx} - \sigma^2 e^{-mx}$$

de aquí se tiene:

$$m^2 - \sigma^2 = 0$$

$$m = \pm \sigma$$

$$f_h = c_1 e^{\sigma x} + c_2 e^{-\sigma x}$$

lo cual puede ser expresado como:

$$f_h = c_1 \cosh(\sigma x) + c_2 \sinh(\sigma x)$$

para el caso que nos compete se usará de la siguiente manera:

$$f_h = c_1 \sinh\sigma(x_{i+1} - x) + c_2 \sinh\sigma(x - x_i)$$

la última expresión es la que se utilizará para futuros cálculos.

De la misma ecuación se obtiene una solución particular:

$$f_p = Ax + B$$

donde resolviendo se obtiene:

$$f_p = -\frac{\alpha_i}{\sigma^2}x - \frac{\beta_i}{\sigma^2} = -\frac{r_{i+1} - r_i}{\sigma^2 h_i}x - \left(\frac{r_i}{\sigma^2} - \frac{r_{i+1} - r_i}{\sigma^2 h_i}x_i \right)$$

por tanto la solución general es de la forma:

$$f_i(x) = c_1 \sinh\sigma(x_{i+1} - x) + c_2 \sinh\sigma(x - x_i) - \left(\frac{r_{i+1} - r_i}{\sigma^2 h_i} \right)x - \left(\frac{r_i}{\sigma^2} - \frac{r_{i+1} - r_i}{\sigma^2 h_i}x_i \right)$$

donde c_1 y c_2 son cualesquiera valores los cuales dependerán de las relaciones. Ahora, veamos que reduciendo términos se tiene:

$$f_i(x) = c_1 \sinh\sigma(x_{i+1} - x) + c_2 \sinh\sigma(x - x_i) + \frac{1}{\sigma^2} \left[r_i \left(\frac{x - x_i}{h_i} \right) + r_{i+1} \left(\frac{x - x_i}{h_i} \right) \right]$$

Sustituyendo r_i y r_{i+1} y utilizando la condición de que $f(x_i) = y_i$ y $f(x_{i+1}) = y_{i+1}$, se obtienen los valores de c_1 y c_2 :

$$c_1 = \frac{f(x_i) - \left(y_i - \frac{f''(x_i)}{\sigma^2} \right)}{\sinh\sigma h_i}$$

$$c_2 = \frac{f(x_{i+1}) - \left(y_{i+1} - \frac{f''(x_{i+1})}{\sigma^2} \right)}{\sinh\sigma h_i}$$

sustituyendo c_1 y c_2

$$f_i(x) = \frac{f(x_i) - \left(y_i - \frac{f''(x_i)}{\sigma^2} \right)}{\sinh\sigma h_i} \sinh\sigma(x_{i+1} - x) + \frac{f(x_{i+1}) - \left(y_{i+1} - \frac{f''(x_{i+1})}{\sigma^2} \right)}{\sinh\sigma h_i} \sinh\sigma(x - x_i)$$

$$+ \left(y_i - \frac{f''(x_i)}{\sigma^2} \right) \left(\frac{x - x_i}{h_i} \right) + \left(y_{i+1} - \frac{f''(x_{i+1})}{\sigma^2} \right) \left(\frac{x - x_i}{h_i} \right)$$

reduciendo términos:

$$f_i(x) = \frac{f''(x_i)}{\sigma^2} \frac{\sinh\sigma(x_{i+1} - x)}{\sinh\sigma h_i} + \left(y_i - \frac{f''(x_i)}{\sigma^2}\right) \left(\frac{x - x_i}{h_i}\right) + \frac{f''(x_{i+1})}{\sigma^2} \frac{\sinh\sigma(x - x_i)}{\sinh\sigma h_i} \\ + \left(y_{i+1} - \frac{f''(x_{i+1})}{\sigma^2}\right) \left(\frac{x - x_i}{h_i}\right)$$

$$f_i(x) = \frac{f''(x_i)}{\sigma^2} \frac{\sinh\sigma(x_{i+1} - x)}{\sinh\sigma h_i} + \left(y_i - \frac{f''(x_i)}{\sigma^2}\right) \left(\frac{x - x_i}{h_i}\right) + \frac{f''(x_{i+1})}{\sigma^2} \frac{\sinh\sigma(x - x_i)}{\sinh\sigma h_i} \\ + \left(y_{i+1} - \frac{f''(x_{i+1})}{\sigma^2}\right) \left(\frac{x - x_i}{h_i}\right)$$

cuando se va a usar la familia $f_i(x)$ para construir el spline en tensión solo contamos con $f_i(x_i)$ y entonces será necesario determinar el valor $f''(x_i)$, este se usará al final cuando pedimos que los $f_i(x)$ de la familia se peguen en forma continuamente diferenciable, y que por la forma de la expresión serán C^2 .

Vamos ahora a proceder a determinar los valores de $f'(x_i)$, pidiendo que:

$$f'_i(x_i^+) = f'_{i+1}(x_i^-) \quad i = 1, \dots, n-1$$

y esto nos dará lugar a un sistema de ecuaciones diferenciables como en el caso de los spline cúbicos, en los cuales elegimos la condición a la frontera.

derivamos a $f(x)$

$$f'_i(x) = \frac{f''(x_i)}{\sigma^2 \sinh\sigma h_i} \cosh\sigma(x_{i+1} - x)(-\sigma) + \left(y_i - \frac{f''(x_i)}{\sigma^2}\right) \left(\frac{-1}{h_i}\right) \\ + \frac{f''(x_{i+1}) \cosh\sigma(x - x_i)(\sigma)}{\sigma^2 \sinh\sigma h_i} - \frac{f''(x_i)}{\sigma^2}$$

y posteriormente evaluamos x_i en $f'_i(x)$, por la izquierda y por la derecha:

$$f'_i(x_i^+) = -\sigma \frac{f''(x_i)}{\sigma^2} \frac{\cosh\sigma h_i}{\sinh\sigma h_i} - \frac{1}{h_i} \left(y_i - \frac{f''(x_i)}{\sigma^2}\right) + \frac{f''(x_{i+1})}{\sigma^2} \frac{\sigma}{\sinh\sigma h_i} + \frac{1}{h_i} \left(y_{i+1} - \frac{f''(x_{i+1})}{\sigma^2}\right)$$

$$f'_{i+1}(x_i^-) = -\sigma \frac{f''(x_{i-1})}{\sigma^2 \sinh \sigma h_{i-1}} - \frac{1}{h_{i-1}} \left(y_{i-1} - \frac{f''(x_{i-1})}{\sigma^2} \right) + \sigma \frac{f''(x_i) \cosh \sigma h_{i-1}}{\sigma^2 \sinh \sigma h_{i-1}} + \frac{1}{h_{i-1}} \left(y_i - \frac{f''(x_i)}{\sigma^2} \right)$$

Igualando $f'_i(x_i^+) = f'_{i+1}(x_i^-)$

$$-\sigma \frac{f''(x_i) \cosh \sigma h_i}{\sigma^2 \sinh \sigma h_i} - \frac{1}{h_i} \left(y_i - \frac{f''(x_i)}{\sigma^2} \right) + \frac{f''(x_{i+1})}{\sigma^2} \frac{\sigma}{\sinh \sigma h_i} + \frac{1}{h_i} \left(y_{i+1} - \frac{f''(x_{i+1})}{\sigma^2} \right) = \frac{f''(x_{i-1})}{\sigma^2}$$

$$\frac{\sigma}{\sinh \sigma h_i - 1} - \frac{1}{h_i - 1} \left(y_{i-1} - \frac{f''(x_{i-1})}{\sigma^2} \right) + \sigma \frac{f''(x_i) \cosh \sigma h_{i-1}}{\sigma^2 \sinh \sigma h_{i-1}} + \frac{1}{h_i} \left(y_i - \frac{f''(x_i)}{\sigma^2} \right)$$

Separando términos

$$-\sigma \frac{f''(x_i) \cosh \sigma h_i}{\sigma^2 \sinh \sigma h_i} + \frac{1}{h_i} \frac{f''(x_i)}{\sigma^2} + \frac{f''(x_{i+1})}{\sigma^2} \frac{\sigma}{\sinh \sigma h_i} - \frac{1}{h_i} \frac{f''(x_{i+1})}{\sigma^2} + \frac{f''(x_{i-1})}{\sigma^2} \frac{\sigma}{\sinh \sigma h_i - 1} - \frac{1}{h_i - 1}$$

$$\frac{f''(x_{i-1})}{\sigma^2} - \sigma \frac{f''(x_i) \cosh \sigma h_{i-1}}{\sigma^2 \sinh \sigma h_{i-1}} + \frac{1}{h_i - 1} + \frac{f''(x_i)}{\sigma^2} = \frac{1}{h_i} y_i - \frac{1}{h_i + 1} y_{i+1} - \frac{1}{h_i - 1} y_{i-1} + \frac{1}{h_i - 1} y_i$$

Cambio de signo en la anterior expresión

$$\sigma \frac{f''(x_i) \cosh \sigma h_i}{\sigma^2 \sinh \sigma h_i} - \frac{1}{h_i} \frac{f''(x_i)}{\sigma^2} - \frac{f''(x_{i+1})}{\sigma^2} \frac{\sigma}{\sinh \sigma h_i} + \frac{1}{h_i} \frac{f''(x_{i+1})}{\sigma^2} - \frac{f''(x_{i-1})}{\sigma^2} \frac{\sigma}{\sinh \sigma h_i - 1} + \frac{1}{h_i - 1}$$

$$\frac{f''(x_{i-1})}{\sigma^2} + \sigma \frac{f''(x_i) \cosh \sigma h_{i-1}}{\sigma^2 \sinh \sigma h_{i-1}} - \frac{1}{h_i - 1} - \frac{f''(x_i)}{\sigma^2} = -\frac{1}{h_i} y_i + \frac{1}{h_{i+1}} y_{i+1} + \frac{1}{h_{i-1}} y_{i-1} - \frac{1}{h_{i-1}} y_i$$

y como resultado se obtiene el sistema

$$\frac{f''(x_{i-1})}{\sigma^2} \left(\frac{1}{h_{i-1}} - \frac{\sigma}{\sinh \sigma h_{i-1}} \right) + \frac{f''(x_i)}{\sigma^2} \left(\sigma \frac{\cosh \sigma h_{i-1}}{\sinh \sigma h_{i-1}} - \frac{1}{h_{i-1}} + \sigma \frac{\cosh \sigma h_i}{\sinh \sigma h_i} - \frac{1}{h_i} \right) +$$

$$\frac{f''(x_{i+1})}{\sigma^2} \left(\frac{1}{h_{i+1}} - \frac{\sigma}{\sinh \sigma h_{i+1}} \right) = \frac{(y_{i+1} - y_i)}{h_i} - \frac{(y_i - y_{i-1})}{h_{i-1}} \quad i = 1, \dots, n-1 \quad (2.3)$$

2.2. Normalizando el parámetro en tensión y condiciones de frontera

En la práctica nos conviene escalar el parámetro de tensión σ , de tal forma que toma en cuenta el tamaño de la partición.

$$\gamma = \sigma \frac{x_n - x_0}{n} = \sigma h$$

donde n es el número de intervalos, x_0 y x_n son los extremos del intervalo de interés. Sustituyendo en el sistema a construir nos queda:

$$\begin{aligned} & \frac{h^2 f''_{j-1}}{\gamma^2} \left[\frac{1}{h_j} - \frac{\gamma}{h \operatorname{senh}\left(\frac{\gamma h_j}{h}\right)} \right] + \\ & \frac{h^2 f''_j}{\gamma^2} \left[\frac{\gamma}{h} \operatorname{coth}\left(\frac{\gamma h_j}{h}\right) - \frac{1}{h_j} + \frac{\gamma}{h} \operatorname{coth}\left(\frac{\gamma h_j}{h}\right) - \frac{1}{h_{j+1}} \right] + \\ & \frac{h^2 f''_{j+1}}{\gamma^2} \left[\frac{1}{h_{j+1}} - \frac{\gamma}{h \operatorname{senh}\left(\frac{\gamma h_{j+1}}{h}\right)} \right] \\ & = \frac{(y_{i+1} - y_i)}{h_i} - \frac{(y_i - y_{i-1})}{h_{i-1}} \end{aligned} \quad (2.4)$$

Procedamos a elegir las condiciones a la frontera.

Condiciones de frontera

Existe una gama de condiciones de frontera que podemos elegir para nuestro spline en tensión, nosotros en esta ocasión solo nos avocaremos a las siguientes [4]:

- Si conocemos el valor de de los vectores tangentes $P'(x_0)$ y $P'(x_n)$, corresponde al spline completo.
- Si $P''(x_0) = P''(x_n) = 0$, este tipo de de condiciones de frontera corresponden al spline natural
- Si $P'(x_0) = P'(x_n)$ y $P''(x_0) = P''(x_n)$, esto para curvas cerradas, estaremos hablando de un spline periódico (con periodo $x_n - x_0$).

Esto nos da las ecuaciones siguientes para completar nuestro sistema de ecuaciones:

$$A f'' = b$$

Enseguida mostraremos el desarrollo del sistema de ecuaciones para cada uno de ellos.

Spline completo

Aplicando la primera condición de frontera, el spline completo, se tiene:

$$f'(x_0) = y'_0$$

$$f'(x_n) = y'_n$$

sustituyendo $f'_1(x_0)$

$$f'_1(x_0) = \frac{f''_0}{\sigma^2} \left[-\frac{\sigma \cosh(\sigma(x_1 - x_0))}{\sinh(\sigma h_1)} + \frac{1}{h_1} \right] + \frac{f''_1}{\sigma^2} \left[-\frac{\sigma \cosh(\sigma(x_1 - x_0))}{\sinh(\sigma h_1)} + \frac{1}{h_1} \right] + \frac{(P_1 - P_0)}{h_1}$$

$$\frac{f''_0}{\sigma^2} = \frac{f''_1}{\sigma^2} \left[\frac{\sigma h_1 - \sinh(\sigma h_1)}{\sigma h_1 \cosh(\sigma h_1) - \sinh(\sigma h_1)} \right] + \sinh(\sigma h_1) \left[\frac{(P_1 - P_0) - h_1 y_0}{\sigma h_1 \cosh(\sigma h_1) - \sinh(\sigma h_1)} \right]$$

Ahora se sustituye $f'(x_n) = y'_n$ y se obtiene:

$$\frac{f''_n}{\sigma^2} = \frac{f''_{n-1}}{\sigma^2} \left[\frac{\sigma h_n - \sinh(\sigma h_n)}{\sigma h_n \cosh(\sigma h_n) - \sinh(\sigma h_n)} \right] + \sinh(\sigma h_n) \left[\frac{h_n y_n - (P_n - P_{n-1})}{\sigma h_n \cosh(\sigma h_n) - \sinh(\sigma h_n)} \right]$$

y los coeficientes $a_{i,j}$ de la matriz A las podemos obtener de la siguiente forma:

$$a_{i,i} = \sigma \coth(\sigma h_i) - \frac{1}{h_i} + \sigma \coth(\sigma h_{i+1}) - \frac{1}{h_{i+1}}$$

$$a_{i,i-1} = \frac{1}{h_i} - \frac{\sigma}{\sinh(\sigma h_i)}$$

$$a_{i,i+1} = \frac{1}{h_{i+1}} - \frac{\sigma}{\sinh(\sigma h_{i+1})}$$

$$f_i = \frac{f''_i}{\sigma^2}$$

$$b_i = \left[\frac{P_{i+1} - P_i}{h_{i+1}} - \frac{P_i - P_{i-1}}{h_i} \right]$$

y

$$a_{1,1} = \sigma \coth(\sigma h_1) - \frac{1}{h_1} + \sigma \coth(\sigma h_2) - \frac{1}{h_2} + \frac{\sinh(\sigma h_1) - \sigma h_1}{\sinh(\sigma h_1) - \sigma h_1 \cosh(\sigma h_1)} \left[\frac{1}{h_1} - \frac{\sigma}{\sinh(\sigma h_1)} \right]$$

$$b_1 = \frac{P_2 - P_1}{h_2} - \frac{P_1 - P_0}{h_1} \left[\frac{\sigma h_1 - \sigma h_1 \cosh(\sigma h_1)}{\sinh(\sigma h_1) - \sigma h_1 \cosh(\sigma h_1)} \right] + y_0 \left[\frac{\sinh(\sigma h_1) - \sigma h_1}{\sinh(\sigma h_1) - \sigma h_1 \cosh(\sigma h_1)} \right]$$

$$a_{n-1,n-1} = \sigma \coth(\sigma h_{n-1}) - \frac{1}{h_{n-1}} + \sigma \coth(\sigma h_n) - \frac{1}{h_n} + \left[\frac{\sinh(\sigma h_n) - \sigma h_n}{\sinh(\sigma h_n) - \sigma h_n \cosh(\sigma h_n)} \right]$$

$$b_{n-1} = \frac{P_n - P_{n-1}}{h_n} \left[\frac{\sigma h_n - \sigma h_n \cosh(\sigma h_n)}{\sinh(\sigma h_n) - \sigma h_n \cosh(\sigma h_n)} \right] - \frac{P_{n-1} - P_{n-2}}{h_{n-1}} + y_n \left[\frac{\sinh(\sigma h_n) - \sigma h_n}{\sinh(\sigma h_n) - \sigma h_n \cosh(\sigma h_n)} \right]$$

Spline natural

Utilizando condiciones de frontera $f_0''=0$ y $f_n''=0$ corresponde al spline natural:

$$a_{i,i} = \sigma \coth(\sigma h_i) - \frac{1}{h_i} + \sigma \coth(\sigma h_{i+1}) - \frac{1}{h_{i+1}}$$

$$a_{i,i-1} = \frac{1}{h_i} - \frac{\sigma}{\sinh(\sigma h_i)}$$

$$a_{i,i+1} = \frac{1}{h_{i+1}} - \frac{\sigma}{\sinh(\sigma h_{i+1})}$$

$$f_i(x) = \frac{f_i''(x)}{\sigma^2}$$

$$b_i = \left[\frac{P_{i+1} - P_i}{h_{i+1}} - \frac{P_i - P_{i-1}}{h_i} \right]$$

Spline periódico o spline cíclico

Utilizando condiciones de frontera $f_0' = f_n'$ y $f_0'' = f_n''$ obtenemos lo que se conoce como spline cíclico.

Modificando la ecuación donde se están sustituyendo las ecuaciones de frontera.

$$\frac{f_1''}{\sigma^2} \left[\frac{\sigma}{\sinh(\sigma h_1)} - \frac{1}{h_1} \right] + \frac{f_{n-1}''}{\sigma^2} \left[\frac{\sigma}{\sinh(\sigma h_n)} - \frac{1}{h_n} \right] +$$

$$\frac{f_n''}{\sigma^2} \left[-\sigma \coth(\sigma h_1) + \frac{1}{h_1} - \sigma \coth(\sigma h_n) + \frac{1}{h_n} \right] =$$

$$\frac{P_n - P_{n-1}}{h_n} - \frac{P_1 - P_0}{h_1}$$

$$a_{i,i} = \sigma \coth(\sigma h_i) - \frac{1}{h_i} + \sigma \coth(\sigma h_{i+1}) - \frac{1}{h_{i+1}}$$

$$a_{i,i-1} = \frac{1}{h_i} - \frac{\sigma}{\sinh(\sigma h_i)}$$

$$\begin{aligned}
 a_{i,i+1} &= \frac{1}{h_{i+1}} - \frac{\sigma}{\operatorname{senh}(\sigma h_{i+1})} \\
 f_i &= \frac{f_i''}{\sigma^2} \\
 b_i &= \left[\frac{P_{i+1}P_i}{h_{i+1}} - \frac{P_i - P_{i-1}}{h_i} \right] \\
 a_{1,n} &= \frac{1}{h_1} - \frac{\sigma}{\operatorname{senh}(\sigma h_1)} \\
 a_{n,1} &= \frac{1}{h_1} - \frac{\sigma}{\operatorname{senh}(\sigma h_1)} \\
 a_{n,n} &= \sigma \operatorname{coth}(\sigma h_1) - \frac{1}{h_1} + \sigma \operatorname{coth}(\sigma h_n) - \frac{1}{h_n} \\
 b_n &= \frac{P_1 - P_0}{h_1} - \frac{P_n - P_{n-1}}{h_n}
 \end{aligned}$$

En las siguientes imágenes se muestran resultados favorables, utilizando un σ adecuado para el spline en tensión.

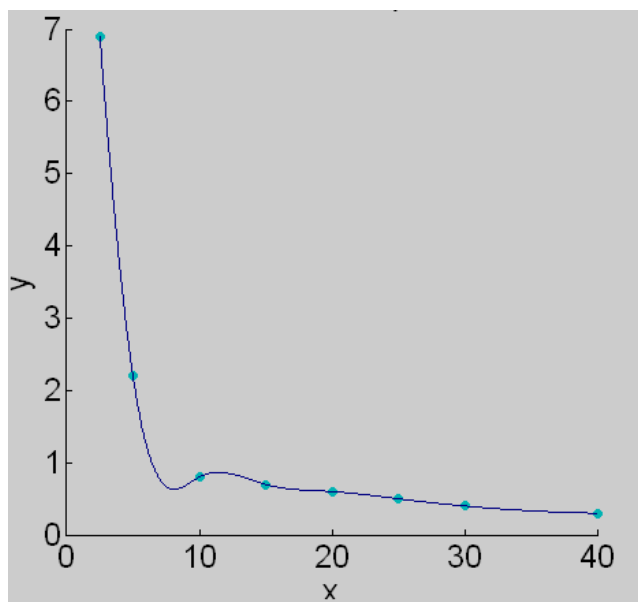


Figura 2.4: Efectividad contra costos. Esta gráfica está hecha de segmentos de funciones cúbicas pegadas una a otra [4].

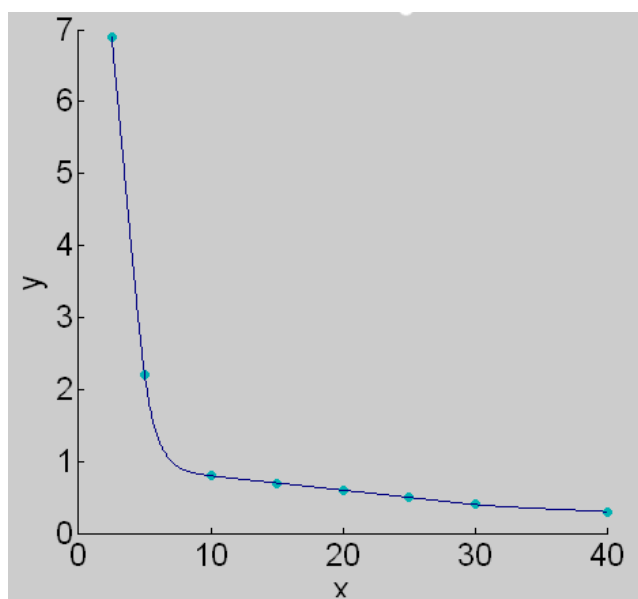


Figura 2.5: spline en tensión evita oscilaciones [4].

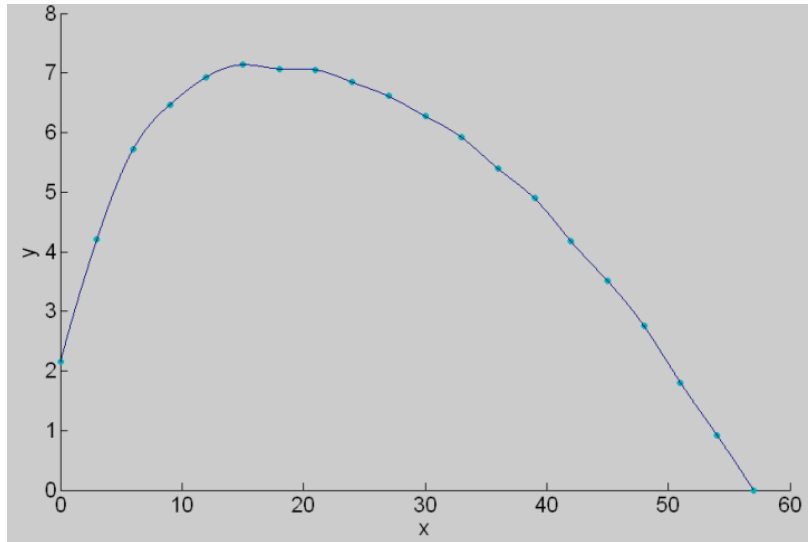


Figura 2.6: Parte de una ala de un avión en la que se observa perturbaciones.

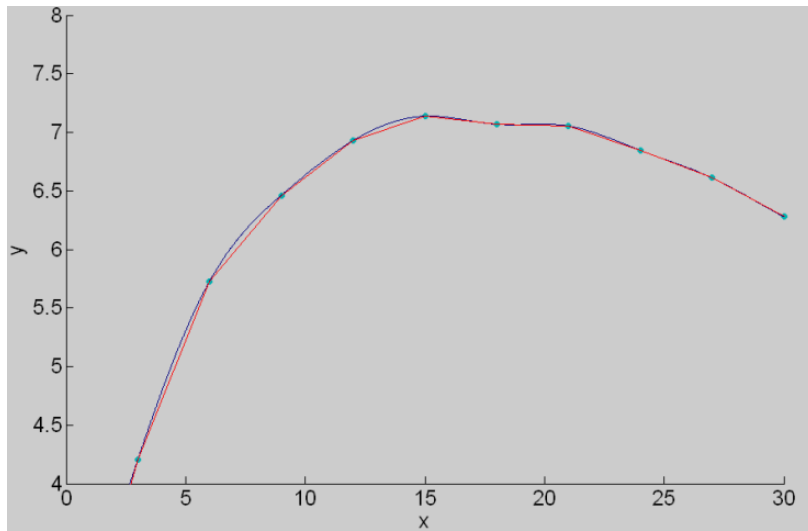
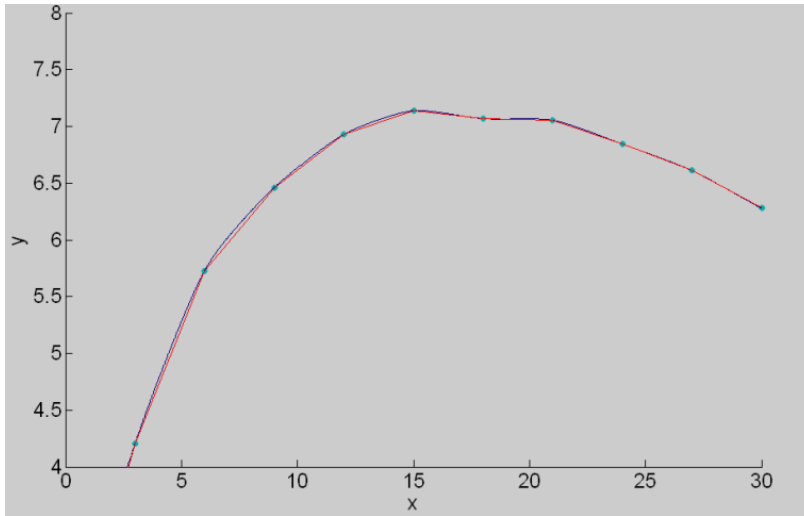
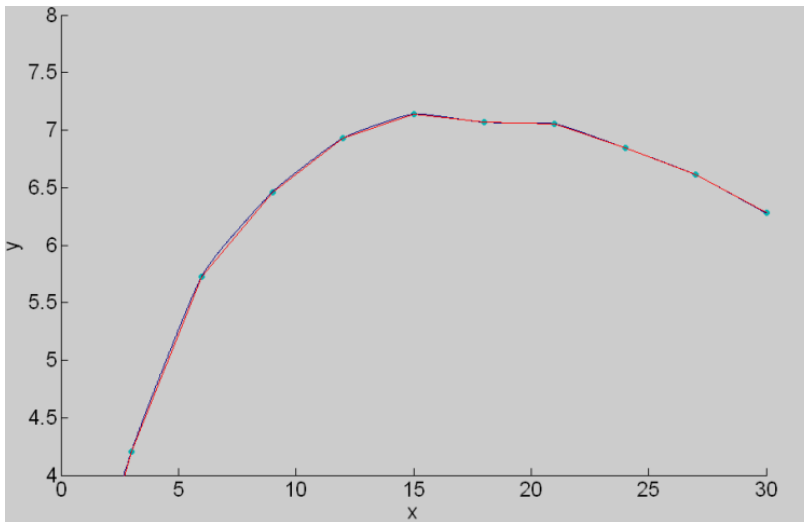


Figura 2.7: Ala de un avión anterior observada con más detenimiento.

Figura 2.8: Ala del avión con $\sigma = 5$.Figura 2.9: Ala del avión con $\sigma = 8$.

Spline paramétrico en tensión

Como información tenemos un conjunto de puntos en el plano x_i, y_i y queremos una curva tal que :

$$\vec{P}_{t_i} = (x_i, y_i)$$

$$\vec{P}_t = (P_1(t), P_2(t))$$

donde $P_1(t)$ y $P_2(t)$ serán spline en tensión.

La elección del parámetro t será la más usual:

$$t_0 = 0$$

$$t_i = t_{i-1} + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}, i = 1, \dots, n$$

$P_i(t)$ se determina escogiendo σ y obtendremos un sistema de ecuaciones conveniente, después de elegir las condiciones a la frontera que van a depender si la curva es cerrada o abierta.

A continuación mostraremos ejemplos a los cuales se les aplica spline paramétrico en tensión.

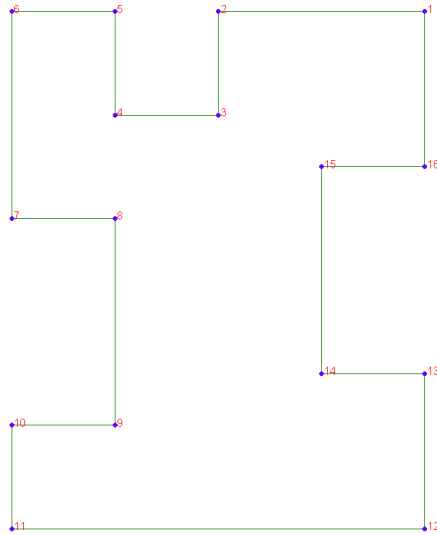
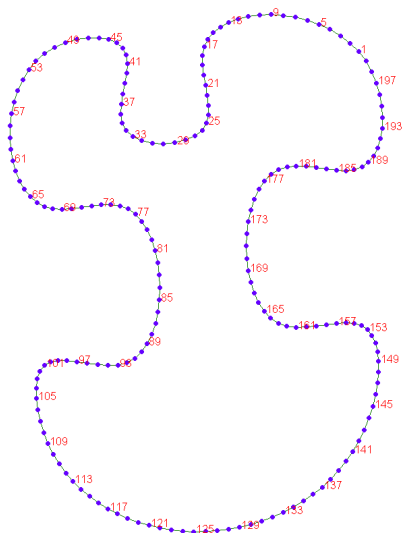
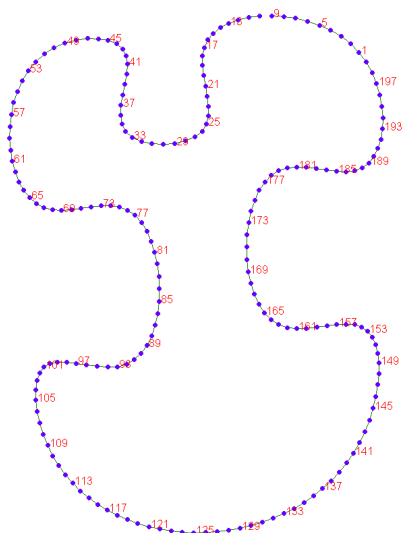


Figura 2.10: Contorno m19 original.

Figura 2.11: Contorno m19, con $\sigma = 0.0$ Figura 2.12: Contorno m19, con $\sigma = 0.5$

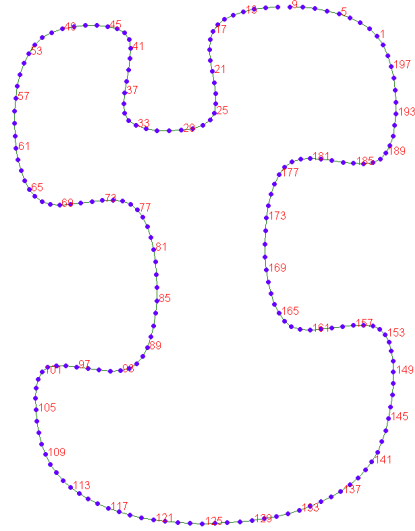


Figura 2.13: Contorno m19, con $\sigma = 3.0$

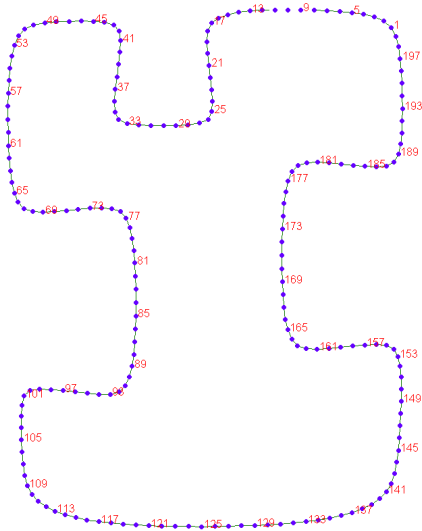


Figura 2.14: Contorno m19, con $\sigma = 7.0$

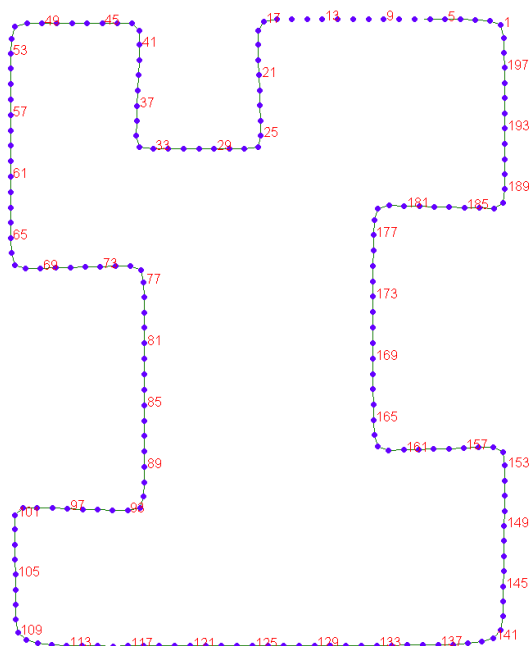


Figura 2.15: Contorno m19, con $\sigma = 25.0$

2.3. Descripción del Algoritmo 716

Renka en su trabajo [3] propone construir el spline en tensión a partir de una formulación diferente que se base en la representación del spline cúbico de Hermite modificado, esto es obtener los valores de la derivada en los puntos de interpolación y con ello se evalúa la forma del interpolante de hermite.

Las rutinas desarrolladas por Renka se conocen en la literatura como el **algoritmo 716** de la colección ACM . Las rutinas vienen descritas básicamente en dos formas: el constructor del spline y con el que evaluamos los puntos del spline.

En este trabajo nos fue más conveniente usar la versión de Renka ya que es posible usar esta biblioteca de programas para el spline en tensión que publico en los Algoritmos de ACM, en el número 716, que tiene mayor generalidad ya que permite construir el factor de tensión de un intervalo a otro; sin embargo a nosotros no nos fue necesario hacer uso de esta facilidad.

definición del spline.

La biblioteca tiene rutinas que nos permiten trabajar con curvas paramétricas planas y en el espacio. El objetivo que tiene es preservar la forma original de los datos a interpolar, para esto calcula automáticamente la primera derivada (también es posible la segunda derivada) del spline; el usuario puede elegir el factor de tensión para satisfacer las necesidades de preservar la forma de los datos o proveer el parámetro a usar.

La idea general es la siguiente: consideremos una colección de datos en el espacio (x_i, y_i, z_i) , obtener una curva $c(t) = (H1(t), H2(t), H3(t))$ que interpole a los datos con respecto al parámetro t . Las componentes $H1(t)$, $H2(t)$, y $H3(t)$ son un spline de tensión que interpolan a los componentes de los puntos discretos x_i , y_i , y z_i . Los valores de sus derivadas se calcularán y se van a denotar de la forma x_p , y_p , y z_p . En cada problema de interpolación usaremos como parámetro t a la longitud de arco.

El spline en tensión se comporta de manera similar a un spline cúbico con la diferencia de que se incluye un parámetro de tensión σ_i . Si el parámetro de tensión es cero obtendremos un spline cúbico y si el parámetro de tensión es grande (cerca de 100), se acercará a un spline lineal. La biblioteca permite al usuario establecer el parámetro de tensión para cada intervalo de definición.

Las rutinas principales se pueden describir en dos bloques, aquellas que construyen el spline y aquellas con las cuales evaluamos el spline:

TSPSI Subrutina que construye una función de interpolación que preserva la forma. Su uso es exclusivo para funciones con abscisa monótona creciente.

TSPSS Subrutina que construye una función de suavizamiento que que preserva la forma. Su uso es exclusivo para funciones con abscisa monótona creciente.

TSPSP Subrutina que construye una curva que plana o espacial paramétrica con restricciones de cota en la frontera de manera que preserve la forma. Las cotas son sobre la distancia ortogonal del segmento original de los datos con respecto a aquella deseada sobre la curva. El parámetro debe ser monótono creciente.

TSPBI Subrutina que construye una función de interpolación con restricciones de cotas en la frontera que preserve la forma. El usuario debe proveer de cotas superior e inferior en el valor de la función, así, como en la derivada y el signo de la segunda derivada. Nuevamente su uso es exclusivo para funciones con abscisa monótona creciente.

Las rutinas con cotas en la frontera no son de nuestro interés en este trabajo.

Una vez determinado el problema a resolver, sea una reconstruir una función, o como en nuestro caso, obtener una curva plana, debemos usar las herramientas adecuadas para evaluar el spline a usar. Para ello esta el siguiente bloque de rutinas

TSVAL1 Esta rutina evalua el interpolante de Hermite construido para una función; también permite evaluar su primera y segunda derivada si el usuario así lo requiere.

TSVAL2 Esta rutina regresa la pareja de valores de los interpolantes de Hermite obtenido por spline de tensión para una curva paramétrica plana. De igual manera es posible obtener la primera derivada para cada uno de los componentes.

TSVAL3 Funciona igual que TSVAL2 solo que para curvas en el espacio.

Es importante mencionar que algunas rutinas de la distribución de Renka pueden ser adecuadas o adaptadas para resolver una colección de problemas diversos, todo es cuestión de plantear adecuadamente el problema a resolver y proponer una forma de atacarlo. El paquete cuenta con otras subrutinas elementales que resuelven la construcción del spline de tensión y cuenta con otras rutinas que permite que dada una colección de datos de una curva o función, calcular la longitud de arco para poderlo usar como el parámetro de interpolación, estas son **ARCL2D** y **ARCL3D**.

El problema que deseamos resolver es construir un interpolante que preserve la forma de un contorno discreto por medio de un spline de tensión. La curva a obtener es cerrada y el parámetro de definición del spline será la longitud de arco. Para fines prácticos usaremos un sólo parámetro de tensión σ para cada segmento entre puntos

de ruptura del spline. Usaremos las rutinas **ARCL2D** , **TSPSP** y **TSVAL2**. La forma de usarlos es sencilla, calcularemos el spline en su forma Hermite usando **TSPSP** (calcularemos las derivadas en los puntos de ruptura) y luego evaluaremos el spline usando **TSVAL2**.

A continuación describiremos los parámetros de entrada y salida de cada subrutina.

```
SUBROUTINE TSPSP (N,ND,X,Y,Z,NCD,IENDC,PER,UNIFRM,
                 LWK, WK, T,XP,YP,ZP,SIGMA,IER)
```

Los parámetros de entrada son

N Es el número de nodos y datos. Por una parte debe ser mayor o igual a dos, y si se trata de una curva cerrada por lo menos mayor o igual a 3, o habrá problemas.

ND Dimensión del problema. Si la curva es plana $ND=2$ y si la curva vive en el espacio debemos usar $ND=3$.

X,Y,Z Son arreglos de longitud **N** que contienen las coordenadas cartesianas de los datos a interpolar. Si la curva está en \mathbf{R}^2 de todas maneras debe declararse el arreglo para **Z**.

NCD Es el número de derivadas continuas para los los nodos de ruptura. Para el problema de interpolación por spline en tensión usaremos $NCD=2$.

IENDC Indicador de las condiciones finales (condiciones de los extremos) para el caso $NCD = 2$ y $PER= FALSE$, se recomienda $IENDC=0$ que corresponde al spline periódico suave. Se usa $IENDC = 1$ si las primeras derivadas de $H1$ por la izquierda y por la derecha de los puntos de la frontera son condiciones de los puntos de los extremos en $XP(1)$ y $XP(N)$, respectivamente, las primeras derivadas de $H2$ son especificadas en $YP(1)$ y $YP(N)$ y si $ND=3$, las primeras derivadas para $H3$ son especificadas en $ZP(1)$ y $ZP(N)$. Si $IENDC = 2$ si las segundas derivadas de $H1$, $H2$ y $H3$ en los puntos de la frontera son usados en $XP(1)$, $XP(N)$, $YP(1)$, $YP(N)$, $ZP(1)$ y $ZP(N)$. Si $IENDC = 3$ si las condiciones de los puntos de la frontera son calculados por la subrutina **ENDSLP** y varia con $SIGMA(1)$ y $SIGMA(N-1)$.

PER Es una variable lógica, es **TRUE** si y solo si la es curva cerrada; esto es, que $H1(t)$, $H2(t)$, y $H3(t)$ son funciones periódicas con periodo $T(N)-T(1)$, donde $T(1)$ y $T(N)$ son los valores de los parámetros asociados con el primer y último punto. Es importante hacer notar que para curvas cerradas se asume que $X(N)$

= $X(1)$, $Y(N) = Y(1)$ y si $ND = 3$, $Z(N) = Z(1)$ y naturalmente se tendrá que $XP(N) = XP(1)$, $YP(N) = YP(1)$, y $ZP(N) = ZP(1)$ si $ND = 3$.

UNIFRM Es una variable lógica que es **TRUE** si y solo si la tensión es uniforme para cada segmento. El factor de tensión debe ser asignado en **SIGMA(1)**, con un rango de 0 a 85. Si **SIGMA(1)=0**, $H(t)$ es un interpolador cúbico a trozos (un spline cúbico si **NCD=2**), y como **SIGMA** aumenta, $H(t)$ se aproxima a un interpolador lineal a trozos, donde H es $H1$, $H2$ o $H3$. Si **UNIFRM = FALSE**, el factor de tensión es elegido automáticamente por **SIGS** para preservar la monotonía y convexidad de los datos.

LWK Longitud del espacio de trabajo. No es necesario el espacio de trabajo si **NCD=1**; al menos $N-1$ localizaciones se requieren si **NCD=2** por otro $N-1$ localizaciones se requieren si **PER=TRUE** y una condición adicional $ND*(N-1)$ localización requiere de convergencia si **SIGS** es llamado (**UNIFRM=FALSE**). Si **NCD=1** entonces **LWK=0**. Para el caso de curvas planas **NCD=2** los valores que se pueden usar dependen del tipo de curva, por ejemplo, usaremos **LWK GE N-1** si **PER=FALSE** y **UNIFRM=TRUE**; debemos de usar **LWK GE 2N-2**, si se tiene que **PER=TRUE**, **UNIFRM=TRUE**; otro caso es **LWK GE (ND+1)*(N-1)**; siempre que **PER= FALSE** y **UNIFRM=FALSE**; y **LWK GE (ND+2)*(N-1)** si ocurre que **PER=TRUE** y **UNIFRM=FALSE**.

Los parámetros anteriores, excepto $X(N)$, $Y(N)$ y $Z(N)$ no se ven modificados por esta rutina.

WK Longitud del arreglo de trabajo, cuya dimensión es mayor o igual a **LWK**.

T Longitud de arco de la curva. Debe ser calculado previamente usando **ARCL2D** o **ARCL3D** dependiendo de la dimensión del problema. La dimensión del arreglo es mayor o igual a N .

XP, **YP**, **ZP** En la salida, contienen la derivada del spline en los puntos de ruptura siguiendo la representación del spline de Hermite. Dependiendo de las condiciones a la frontera **IENDC** el usuario deberá asignar el valor a utilizar en los extremos del arreglo. La dimensión de esos arreglos es de N .

SIGMA Es un arreglo de dimensión $N-1$, el contiene los valores de los factores de tensión. **SIGMA(I)** es asociado con el intervalo $T(I)$, $T(I+1)$ para $I=1, \dots, N-1$. **SIGMA** no se altera si $-5 < IER < 0$, y **SIGMA** es constante (no optimo) si **IER=-6** o **IENDC** es valido. Si el usuario decide usar el mismo parámetro de tensión para cada segmento del spline, revise el parámetro **PER**.

Para el caso de la rutina

```

SUBROUTINE TSVAL2 (N,T,X,Y,XP,YP,SIGMA,IFLAG,NE,
.                  TE, VX,VY,IER)

```

los parámetros de entrada son

- N** Número de puntos del spline. N menor igual a 2.
- T** Arreglo de dimensión N que contiene los datos de la longitud de arco de la curva los cuales estan acomodados en forma creciente (valores paramétricos). Se relaciona con la subrutina ARCL2D.
- X** Arreglo de longitud N que contiene datos para la abscisa de los datos originales.
- Y** Arreglo de longitud N que contiene datos para la ordenada de los datos originales.
- XP** Arreglo de longitud N que contiene las primeras derivadas. $XP(I) = H1P(T(I))$ para $I=1, \dots, N$, donde $H1P$ denota las derivadas de $H1$ para $I=1, \dots, N-1$.
- YP** Arreglo de longitud N que contiene las primeras derivadas. $YP(I) = H2P(T(I))$ para $I=1, \dots, N$, donde $H2P$ denota las derivadas de $H2$.
- Se hace notar que $C(T(I)) = (X(I), Y(I))$ y $CP(T(I)) = (XP(I), YP(I))$ con $I=1, \dots, N$ son puntos y derivadas en forma vectorial.
- SIGMA** Es una arreglo de longitud $N-1$ que contiene los valores de los factores de tensión cuyo valor absoluto determina el balance entre el spline cúbico y el spline lineal en cada intervalo. $SIGMA(I)$ es asociado con el intervalo $(I, I+1)$ para $I=1, \dots, N-1$.
- IFLAG** Es un indicador para salir; si $IFLAG= 0$ los valores de $H1$ y $H2$ serán calculados; $IFLAG= 1$ si los vectores de las primeras derivadas son calculados. Los vectores tangentes pueden ser obtenidos normalizando estos vectores unitarios; $IFLAG= 2$ si las segundas derivadas es decir la aceleración son calculadas. Dado un vector tangente U y la segunda derivada del vector V , el vector curvatura puede ser calculado como el producto cruz de $U \times V \times U$. Si solamente desamos obtener los valores de interpolación, debemos usar $IFLAG=0$.
- NE** Número de puntos evaluados. NE debe ser mayor a cero.

TE Arreglo de longitud NE que datos sobre el parámetro de interpolación (longitud de arco) para la evaluación de los puntos. Los valores estan ordenados en forma creciente para maximizar la eficiencia. La extrapolación es realizada si los puntos no estan en el intervalo $[T(1), T(N)]$.

VX, VY Vectores de longitud de al menos NE. En la salida esos arreglos que contienen los valores de las primeras derivadas o segundas derivadas de H1 y H2 al evaluar los puntos (IER < 0). Si IER = -1, VX y VY no son modificados. Si IER = -2, VX y VY puede ser definido.

IER Indica error. IER = 0 si no han encontrado errores y no se produjo extrapolación; IER > 0, si no se han encontrado errores, pero se requiere extrapolación en los puntos de IER. Si IER = -1, N es menor a dos, IFLAG es menor a cero, IFLAG es mayor a dos, o NE es menor a uno; IER = -2, si las abscisas no estan en orden creciente. (Este error no se detecta necesariamente).

Para el caso que nos compete, la forma de interactuar con esas rutinas es la siguiente:

```

...
      call ARCL2D (m,xb,yb, t,ier)
      print*, 'longitud de arco', t(m)
      ncd = 2
      iendc = 0
      per = .TRUE.
      uniform = .false.
c
      lwk = (nd+2)*(m-1)
c
      call TSPSP (m,2,xb,yb,zb,ncd, iendc,per,uniform,
>               lwk,wk,t, xp,yp,zp,sigma,ier)
c
      print*, 'dame el numero de puntos '
      read(*,*) ne
      hstep = t(m) / dfloat(ne -1)
      do i=1,ne
          te(i) = hstep*dfloat(i-1)
      enddo

```

```
c
c * Asignamos un valor fijo del parametro de tension
c
      fsigma = 0.0d0
      print*, 'dame el parametro de tension '
      read(*,*) fsigma
      sigma(1) = fsigma
      iflag = 0
      call TSVALL2 (m,t,xb,yb,yp,yp,sigma,iflag,ne,
>                  te,vx,vy,ier)
c
c * escribimos los datos
c
      do 200 I = 1, ne
          write(4,950) vx(i), vy(i)
200    continue
```

Para la implementación dentro del sistema Editboundary, usamos una rutina de “enlace” entre Matlab y fortran. Esa rutina se conoce como API y nos permite recibir los datos desde Matlab y pasarlos a la rutina principal. Los valores de salida, que son los datos interpolados, son ahora llevados a Matlab. No describiremos esa rutina por no ser del interés de la tesis.

2.4. Rutinas de TSPACKGUI para Matlab

Recientemente Renka [3] desarrolló una versión del paquete TSPACK para Matlab 2007 o posterior, es lo que en la literatura se conoce como TSPACKGUI. El cual permite hacer uso de diversas herramientas gráficas para obtener un spline de tensión sobre una colección de datos.

Es un código desarrollado específicamente para ser usado en Matlab utilizando los GUI (Graphical User Interface) de este entorno de programación. El usuario debe introducir los datos (en 2D o 3D) en línea y llamar al sistema en la forma que se muestra en la figura 2.16.

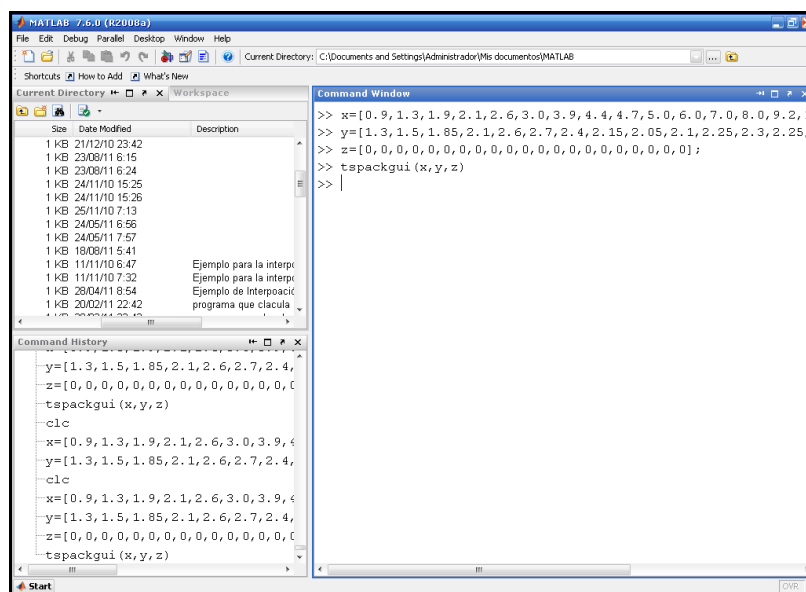


Figura 2.16: Interfase de MATLAB 2008a

El paquete TSPACKGUI cuenta con los algoritmos originalmente implementados en fortran traducidos a Matlab pero totalmente incorporados a la interface gráfica, esto no es posible usarlo independientemente de la interface, como se muestra en la figura 2.17.

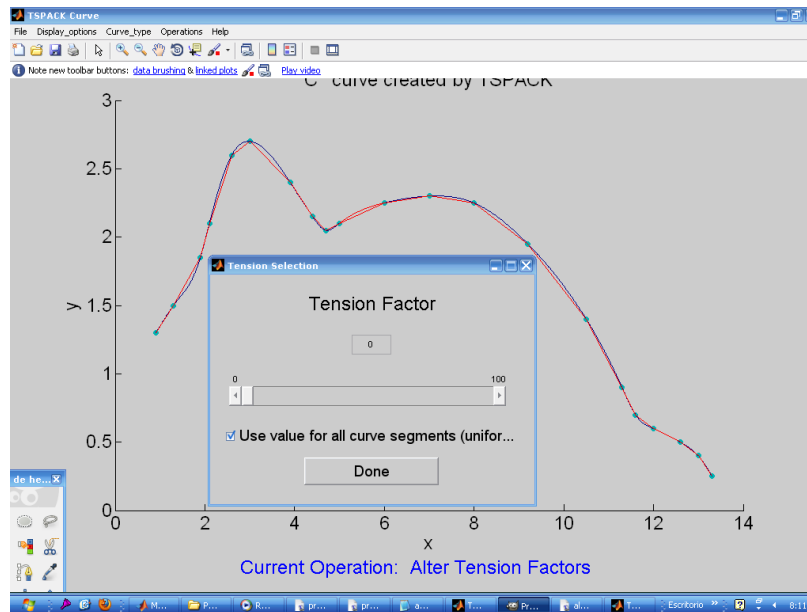


Figura 2.17: spline en tensión aplicado a los datos de datos de una función

En este programa, podemos trabajar con curvas paramétricas planas o curvas en el espacio, a las cuales se les puede aplicar interpolación o suavizamiento y de manera adecuada elegir los parámetros de tensión del spline. Como se observa en la 2.17, el usuario cuenta con opciones de menú como lo son “File”, “Display-options”, “Curve-type”, “Operations” y “Help”.

Si queremos trabajar con una función lo primero que hacemos es dar datos preliminares de esta, debemos de indicar cuales son los puntos x y y de la función. Para esto definiremos los datos en arreglos dentro del entorno de programación de MATLAB 2008a, ver figura 2.16. Una vez declarados los datos llamamos a la función `tspackgui.m` daremos suavizamiento, aplicando spline en tensión. Podemos aplicar un factor en tensión de manera uniforme o periódica con la opción de “use value for all curve segments”, la cual se despliega cuando elegimos la opción “tension factor” en “Options” y damos click sobre los datos de nuestra función o contorno, vea la figura 2.17.

Por otra parte, se tiene la libertad de elegir que tan “tensa” queremos la curvatura de los datos a nuestra función o contorno. Notemos que si el factor de tensión es cero se estará tratando del caso de un spline cúbico y si el factor de tensión es grande, entonces será prácticamente un spline lineal. En los siguientes ejemplos describiremos algunos resultados que se pueden obtener con este programa.

El primer ejemplo fue tomado de [10] que corresponde a los datos de una sección de la silueta de un pato. La figura 2.18 muestra el spline de tensión con $\sigma = 0$, con condiciones naturales a los extremos y que corresponde al caso de spline cúbico usual.

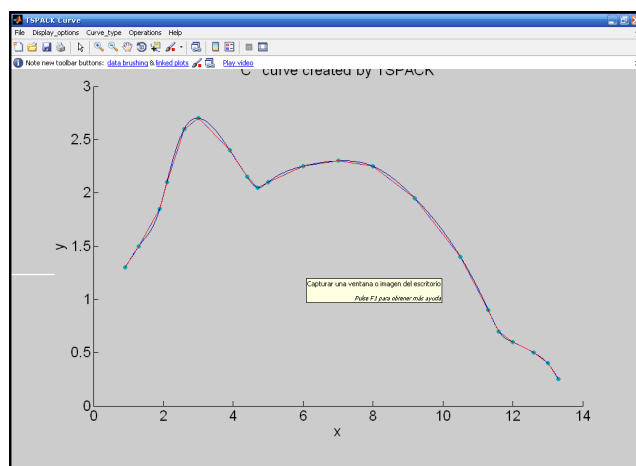


Figura 2.18: Datos anteriores con $\sigma = 0$

Luego, en la figura 2.19, hemos utilizado un valor de $\sigma = 5.0$ y en la figura 2.20 de $\sigma = 8.0$. Observe que para este valor del parámetro de tensión se aprecia que el spline entre dos puntos de interpolación es casi lineal, lo que se aprecia aún más en la figura 2.21 donde hemos usado un valor de $\sigma = 12.0$

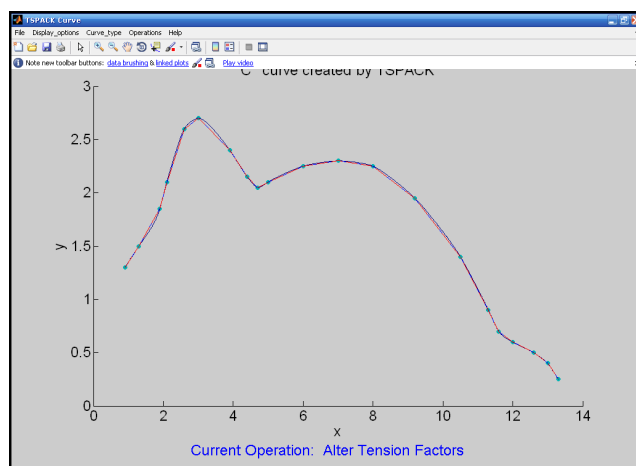
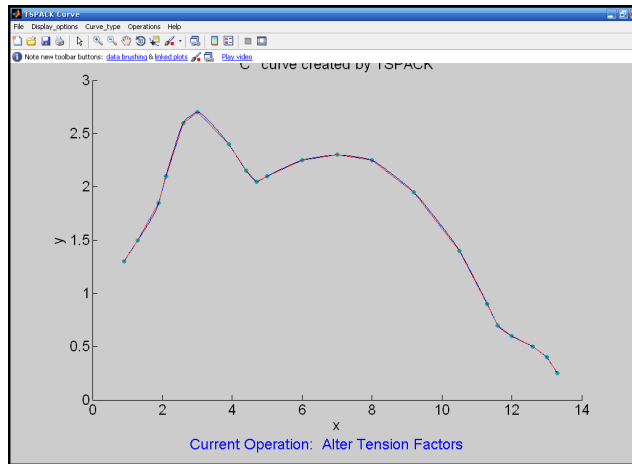
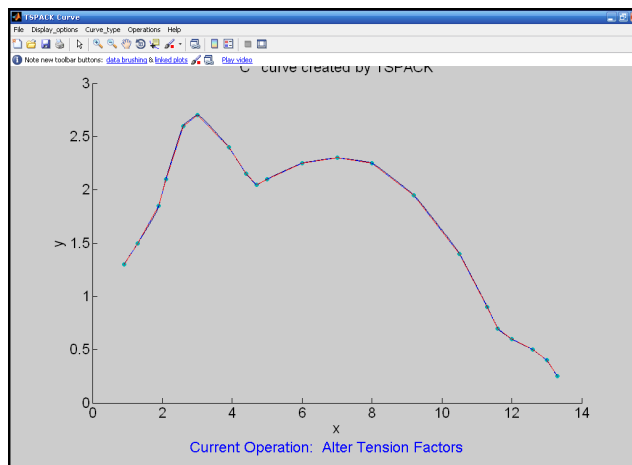


Figura 2.19: Datos anteriores con $\sigma = 5$

Figura 2.20: Datos anteriores con $\sigma = 8$ Figura 2.21: Datos anteriores con $\sigma = 12$

El segundo ejemplos que mostraremos, fue tomado de [10] el cual es muy interesante, ya que presenta diferentes cambios de curvatura al usar $\sigma = 0$, esto se aprecia en la figura 2.22

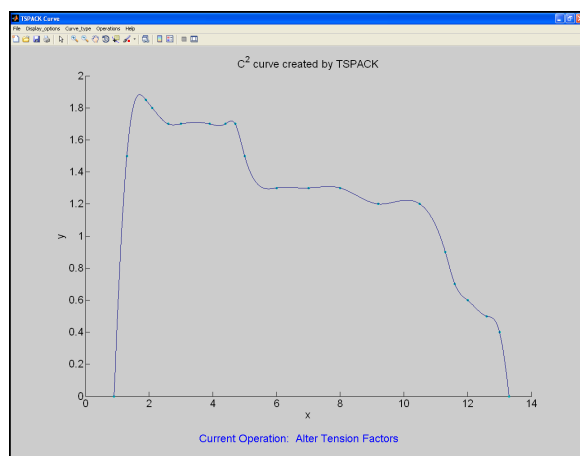


Figura 2.22: Datos de una función, presenta perturbaciones

Si no empleamos un valor $\sigma = 8$ para el parámetro de tensión, la forma del spline de tensión resultante se observa en la figura 2.23

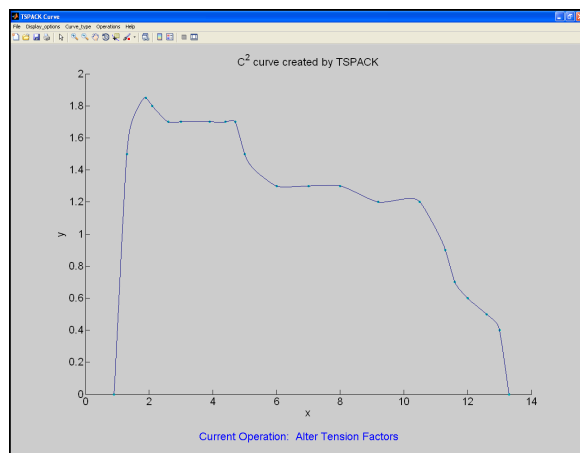


Figura 2.23: Datos anteriores con $\sigma = 8$

Ahora, si nos enfocamos a un segmento de los datos y usamos un parámetro de $\sigma = 15.0$, observamos que el spline de tensión en esta zona es prácticamente segmentos de línea, ver la figura 2.24.

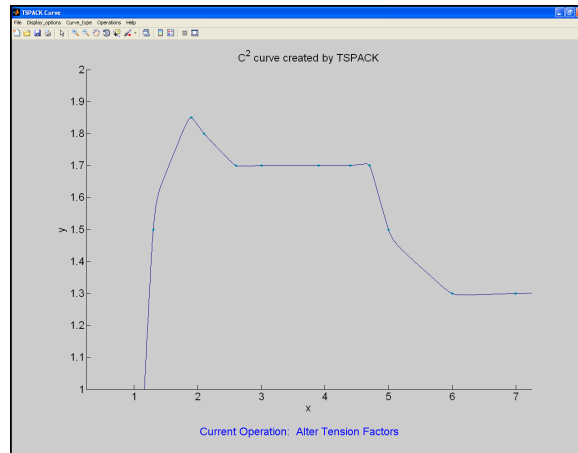


Figura 2.24: Datos anteriores con $\sigma = 15$

El paquete TSPACKGUI también permite trabajar con contorno cerrado pudiendo elegir entre varias opciones tanto de interpolación como de suavizamiento.

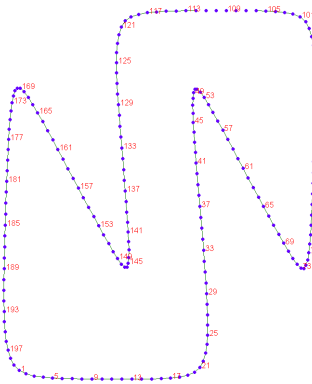


Figura 2.25: Contorno del cisne con $\sigma = 10$

El paquete cuenta con información adicional que puede ser desplegado para cuestiones de calibración del ajuste obtenido como lo es el comportamiento de la derivada

del spline de tensión, esto se logra con la opción de “First Derivative or Curvature”, la cual se despliega del menú de “Display-options”

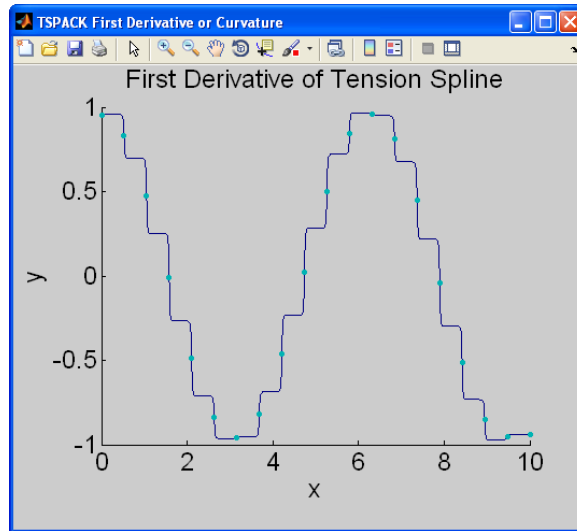


Figura 2.26: Derivadas de la función seno

Para finalizar esta sección, es importante señalar, que el paquete TSPACKGUI no fue implementado dentro del sistema Editboundary, lo que hicimos fue adaptar las rutinas en fortran TOMS93 o Algoritmo 716 [3] de 1996 de Renka y programamos una interface API de Matlab. Todo esto fue incorporado como un módulo del sistema Editboundary donde el usuario puede elegir de manera global el parámetro de tensión. En el siguiente capítulo discutiremos la forma en que incorpora el spline paramétrico y el uso del módulo desarrollado.

Capítulo 3

Interpolación paramétrica usando splines

El problema de nuestro interés, es representar adecuadamente un contorno poligonal para construir mallas 2D, donde el contorno representa a la región de estudio sobre la cual queremos representar algún fenómeno físico, por ejemplo, flujo de fluidos.

Nosotros vamos a entender como contorno a una curva plana cerrada que será la frontera de una región, es decir, una región acotada Ω , simplemente conexa (una curva sin agujeros y que no se auto-intersecte), y $\mathcal{C} = \partial\Omega$. Usualmente se tienen descripciones de la forma: explícita, paramétrica y como un conjunto de puntos.

- $\mathcal{C} = \{(x, y) | f(x, y) = 0\}$
- $\mathcal{C} = \{\bar{\mathbf{x}} | t \in [a, b], \bar{\mathbf{x}}(a) = \bar{\mathbf{x}}(b)\}$
- $\mathcal{C} = \text{Polígono } (P_1, P_2, \dots, P_n, P_{n+1} = P_1)$

En muchas aplicaciones la última representación es la más usual ya que solo se conocen algunos puntos. Las siguientes figuras muestran ejemplos de contornos.

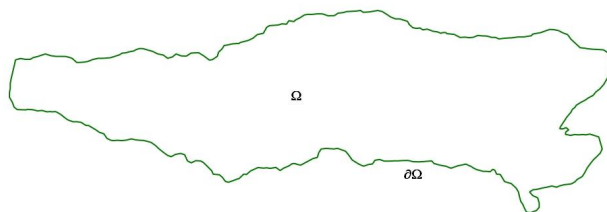


Figura 3.1: Contorno: Lago de Chapala.

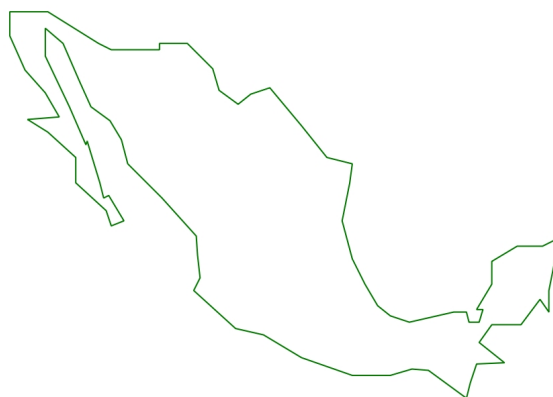


Figura 3.2: Contorno. República Mexicana

Como hemos dicho, usualmente contamos con un conjunto de puntos $P_i = (x_i, y_i)$ que van a definir a un contorno, nuestro interés es una representación en forma continua y paramétrica del mismo. Por ello es necesario elegir un parámetro $t \in [a, b]$ adecuado. El contorno se verá como un conjunto de puntos (x_i, y_i) donde al utilizar una representación para t , buscaremos funciones $f_1(t)$ y $f_2(t)$ que interpolen las coordenadas x y y .

$$x = f_1(t)$$

$$y = f_2(t)$$

donde $f_1(t)$ y $f_2(t)$ para nosotros serán splines. Esto en general, pero para curvas cerradas tendremos además que:

$$f_1(a) = f_1(b)$$

$$f_2(a) = f_2(b)$$

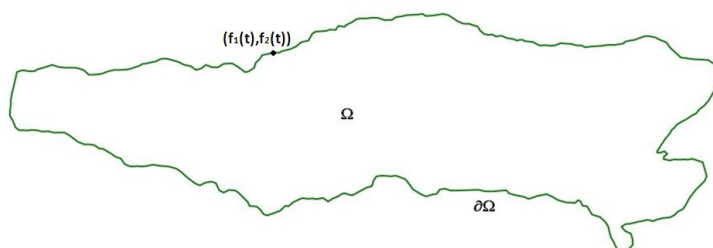


Figura 3.3: Representación paramétrica

Encontraremos una función $\mathbf{S}(t) = (f_1(t), f_2(t))$ de manera que $f_1(t)$ sea un spline que interpole:

| | | | | | | |
|-----|-------|-------|-------|-------|-----|-------|
| t | t_0 | t_1 | t_2 | t_3 | ... | t_n |
| x | x_0 | x_1 | x_2 | x_4 | ... | x_n |

y $f_2(t)$ interpole:

| | | | | | | |
|-----|-------|-------|-------|-------|-----|-------|
| t | t_0 | t_1 | t_2 | t_3 | ... | t_n |
| y | y_0 | y_1 | y_2 | y_4 | ... | y_n |

Tenemos dos problemas a resolver, uno de ellos es saber qué tipo de spline usamos y el segundo quién es el parámetro t . Lo inmediato será considerar dos spline lineales uno para cada f_i , por parámetro se usará la posición i del punto. Como ejemplo tenemos:

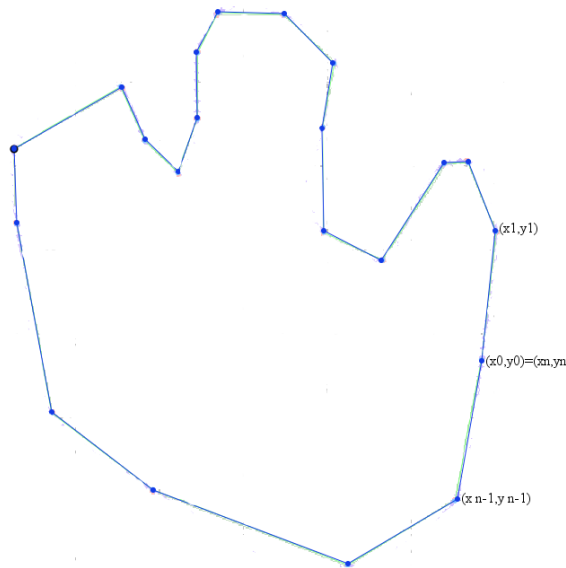
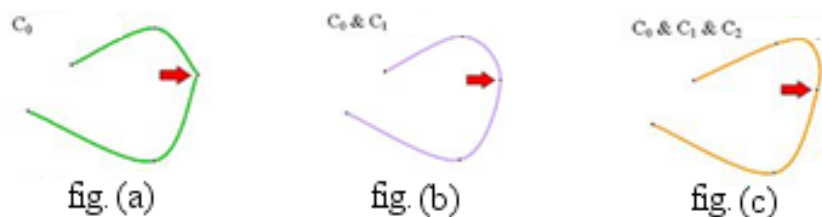


Figura 3.4: Un contorno formado por segmentos

Para algunas aplicaciones no será suficiente, ya que solamente tenemos continuidad pero no continuidad diferenciable.

Nuestro interés es que los segmentos se peguen adecuadamente. Si solamente queremos continuidad podemos obtener picos en la representación fig.(a). En cambio si pedimos diferenciability continua, esto geométricamente significa que los segmentos se peguen con la misma tangente fig.(b). Ahora bien si pedimos continuidad de segundo orden, la curva será suave en los puntos de ruptura o empate fig.(c) [5].



Por otra parte, el parámetro t elegido aparentemente no es relevante, más adelante veremos que es necesario tomarlo en cuenta. Lo que buscamos es una representación $\mathbf{S}(t)$ con características necesarias: continuidad y diferenciabilidad.

Al igual que en los polinomios debemos garantizar una transición sucesiva entre los spline y esto nos lleva a imponer condiciones de continuidad en los puntos de ruptura típicamente hay dos tipos: continuidad paramétrica y continuidad geométrica, la idea de la continuidad paramétrica es que no se rompa nuestro contorno y por otro lado la continuidad geométrica, la suavidad del contorno no garantiza la unión de los puntos.

Nosotros queremos parametrizar un polígono cerrado por una curva suave. Siguiendo la idea del capítulo anterior, se sugiere usar varios spline cúbicos de interpolación tanto para $x = f_1(t)$ como para $y = f_2(t)$. Tenemos opciones, el spline natural:

$$f_1''(t_0) = f_1''(t_n) = 0$$

$$f_2''(t_0) = f_2''(t_n) = 0$$

el spline completo

$$f_1'(t_0) = f_1'(t_n) = g_0$$

$$f_2'(t_0) = f_2'(t_n) = g_n$$

o bien el spline cíclico:

$$f_1'(t_0) = f_1'(t_n)$$

$$f_1''(t_0) = f_1''(t_n),$$

3.1. Elección del parámetro t de interpolación

Elegir la parametrización es crucial para el suavizamiento de la curva. Una elección que puede ser usada es $t_i = i$ desde $i = 0, \dots, n$, como en el siguiente ejemplo:

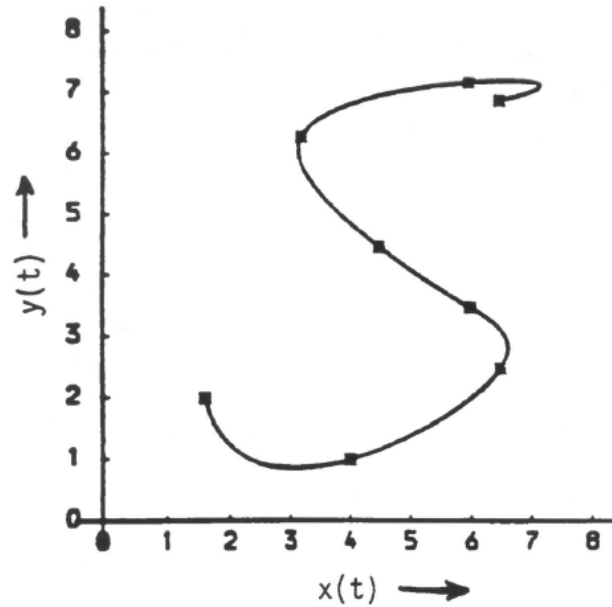


Figura 3.5: spline cúbico paramétrico usando $t_i = i$ [4]

Por otra parte, Ahlberg, Nilson y Walsh [4], definen al parámetro t como longitud de arco

$$t_0 = 0$$

$$t_i = t_{i-1} + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}, i = 1, \dots, n$$

elegir esta parametrización como longitud de arco es conveniente, incluso cuando en los puntos (x_i, y_i) y (x_{i+1}, y_{i+1}) la distancia no es la misma entre ellos ya que la longitud de arco es independiente de la parametrización. Vamos a utilizar longitud de arco como parámetro para los datos que se mostraron en anterior fig.3.5

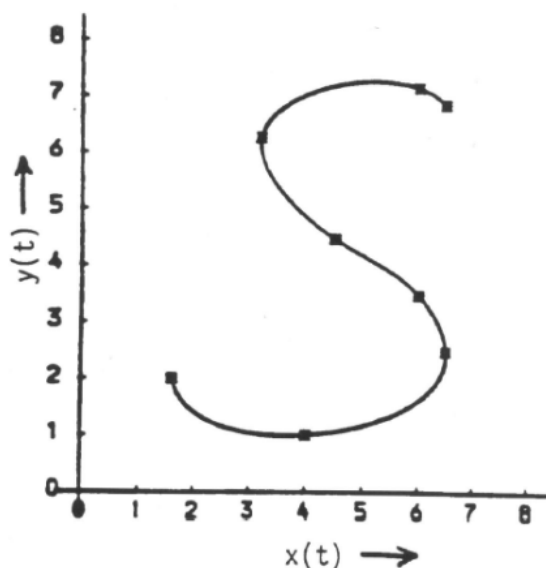


Figura 3.6: spline cúbico paramétrico usando longitud de arco como parámetro t [4]

Visualmente la fig.3.6, la curva es más suave; en adelante usaremos la longitud de arco del contorno como parámetro t para los contornos.

Como ya habíamos dicho, tenemos dos problemas, uno de ellos es saber qué tipo de spline usamos y el segundo quién es el parámetro t , para este último hemos elegido usar a la longitud de arco ya que los resultados obtenidos en las pruebas nos convino usar esta elección, ahora lo que nos resta es elegir el spline. En las observaciones de este capítulo vimos que el spline lineal no ofrece continuidad diferenciable que en algunos casos puede ser necesaria para el usuario, debido a esto vamos a considerar a los spline cúbicos cíclicos para nuestros fines de este trabajo.

En seguida tenemos contornos originales como el m19 en la fig.3.7 (contorno académico del grupo UNAMALLA), al cual se le aplica spline cúbico fig.3.8 y se observa que se pierde la forma del contorno original.

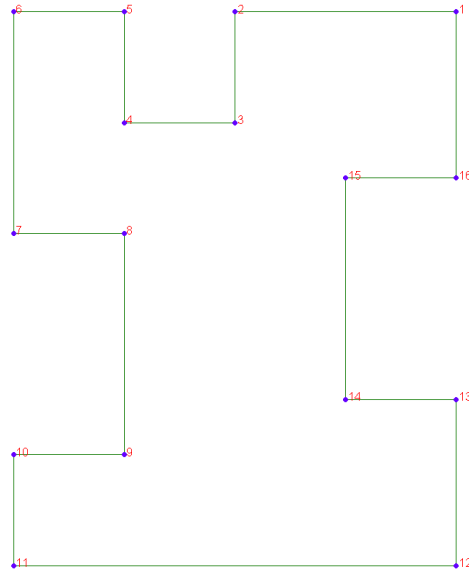


Figura 3.7: contorno original de m19.

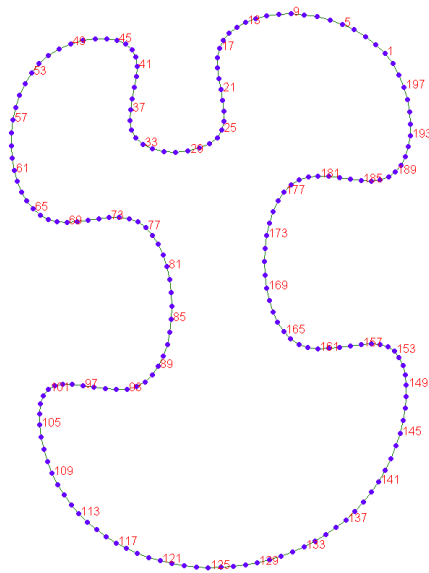


Figura 3.8: spline cúbico aplicado al contorno m19.

3.2. Spline en tensión para contornos cerrados

Debido a estos resultados vamos a elegir otra opción como spline, la óptima por sus características es el spline en tensión, el cual para este trabajo es tema central.

Al spline en tensión se le piden las siguientes condiciones para $j = 1, \dots, n - 1$:

$$\begin{aligned}\vec{P}_j(t_j) &= (x_j, y_j) \\ \vec{P}_{j+1}(t_j) &= \vec{P}_j(t_j) \\ \vec{P}'_{j+1}(t_j) &= \vec{P}'_j(t_j) \\ \vec{P}''_{j+1}(t_j) &= \vec{P}''_j(t_j) \\ \vec{P}''_{j+1}(t) - \sigma^2 \vec{P}''_j(t) &= [\vec{P}''(t_{j-1}) - \sigma^2 \vec{P}''(t_{j-1})] \left[\frac{t_j - t}{t_j - t_{j-1}} \right] \\ &+ [\vec{P}''(t_j) - \sigma^2 \vec{P}''(t_j)] \left[\frac{t - t_{j-1}}{t_j - t_{j-1}} \right]\end{aligned}$$

más una condición de frontera que puede corresponder al spline natural, completo o periódico.

Condiciones de frontera

Existe una gama de condiciones de frontera que podemos elegir para nuestro spline en tensión, nosotros en esta ocasión solo nos avocaremos a las siguientes:

- Si conocemos el valor de los vectores tangentes $\vec{P}'_1(t_0)$ y $\vec{P}'_n(t_n)$, corresponde al spline completo.
- Si $\vec{P}''_1(t_0) = \vec{P}''_n(t_n) = 0$, este tipo de condiciones de frontera corresponden al spline natural
- Si $\vec{P}'_1(t_0) = \vec{P}'_n(t_n)$ y $\vec{P}''_1(t_0) = \vec{P}''_n(t_n)$, esto para curvas cerradas, estaremos hablando de un spline periódico (con periodo $t_n - t_0$).

Aplicamos spline en tensión a los siguientes contornos, variando el factor σ de 0 a un valor donde visualmente el contorno no pierde su forma original. Recordamos que si $\sigma = 0$ estamos trabajando con el spline cúbico.

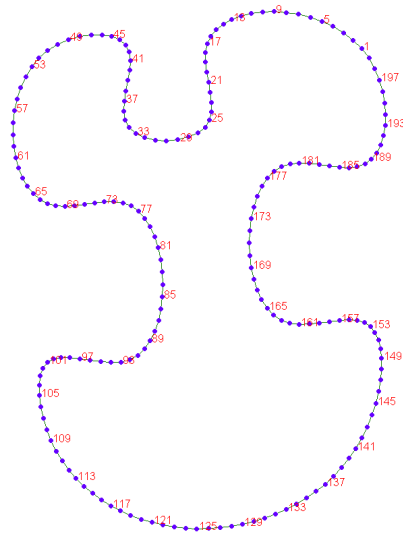


Figura 3.9: Contorno m19 con $\sigma = 0$. Spline cúbico aplicado al contorno.

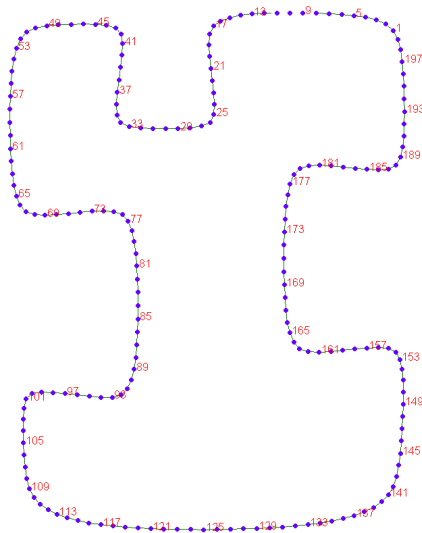


Figura 3.10: Contorno m19 con $\sigma = 7$. Visualmente preserva la forma original del contorno.

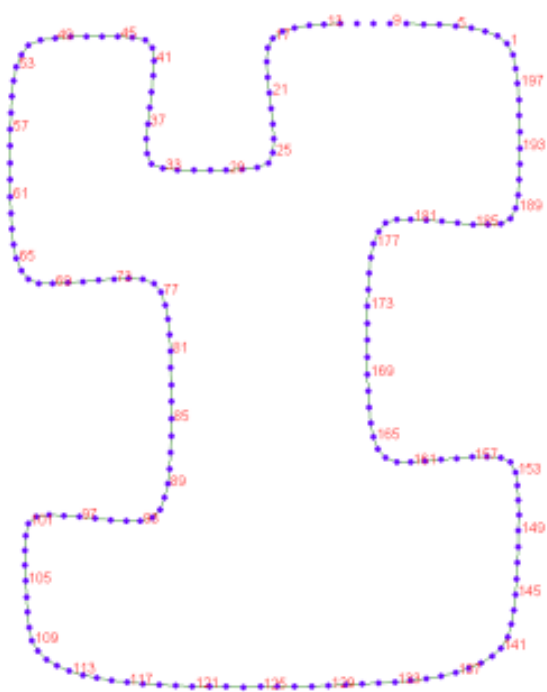


Figura 3.11: Contorno m19 con $\sigma = 8$. Visualmente se preserva aún más la forma original del contorno.

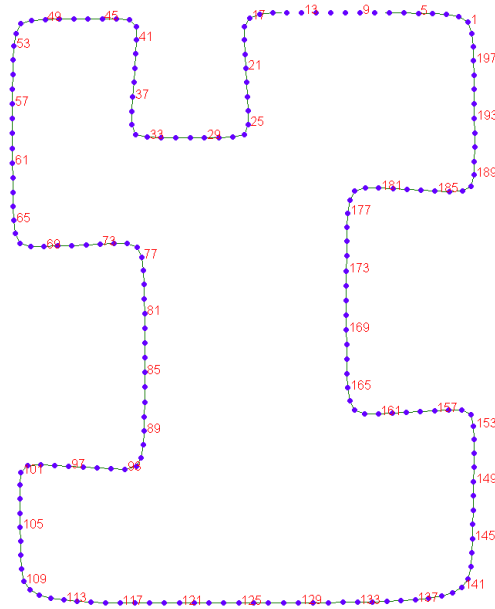


Figura 3.12: Contorno m19 con $\sigma = 15$, mejor visualización a la forma original del contorno

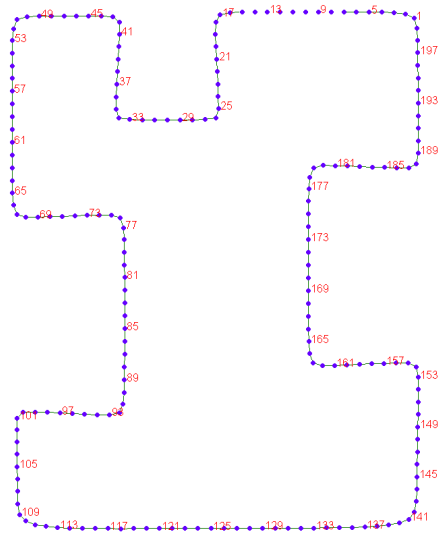


Figura 3.13: Contorno m19 con $\sigma = 20$, se preserva la forma original del contorno

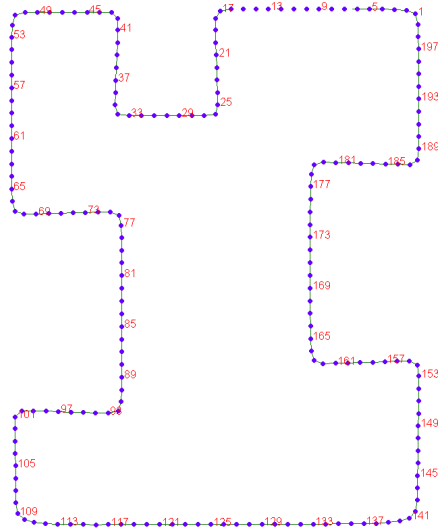


Figura 3.14: Contorno m19 con $\sigma = 25$. Visualmente es la forma original del contorno.

Así también en la fig3.15 tenemos el contorno original con nombre cisne, le hemos aplicado spline cúbico y notamos que se pierde la forma del contorno, hay demasiada suavidad que no es observada en el contorno original.

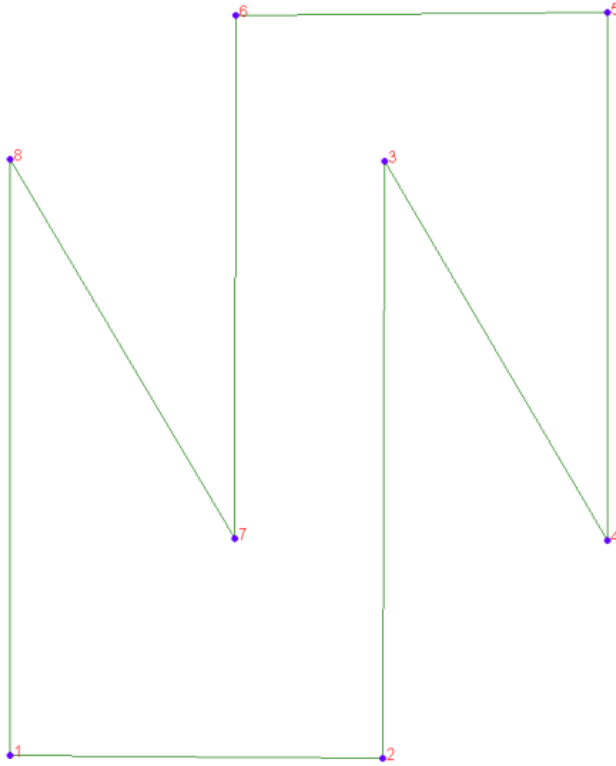


Figura 3.15: Contorno original de cisne.

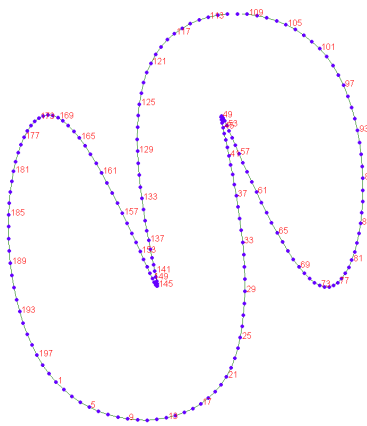


Figura 3.16: Spline cúbico aplicado al contorno del cisne.

El tratamiento que le damos a este contorno para no perder la forma del original es la aplicación del spline en tensión, que enseguida se mostrara en las siguientes imágenes y veremos notablemente como mejora su forma del cisne.

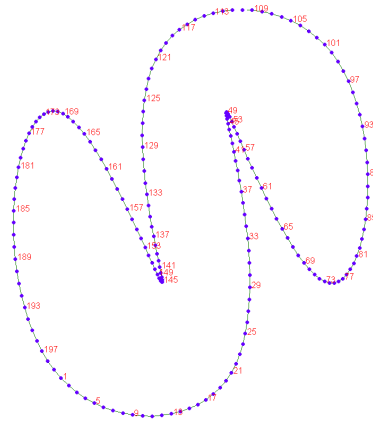


Figura 3.17: Contorno cisne con $\sigma = 0$, se observa el spline cúbico aplicado al contorno.

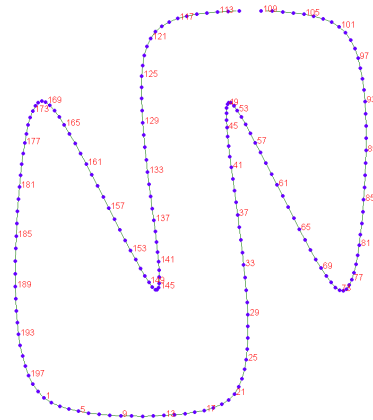


Figura 3.18: Contorno cisne con $\sigma = 5$, va preservando la forma original del contorno.

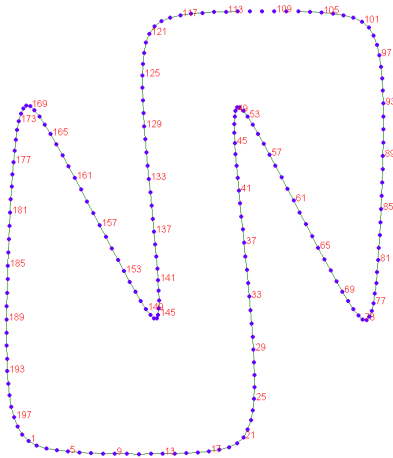


Figura 3.19: Contorno cisne con $\sigma = 8$, mejor visualización a la forma original del contorno.

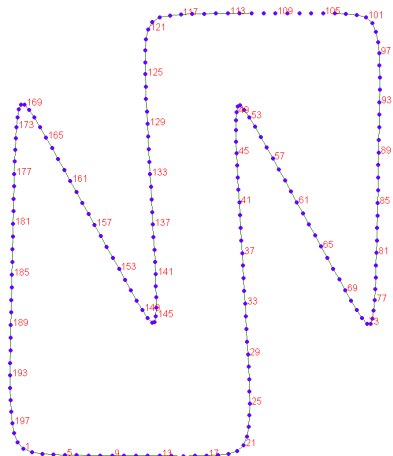


Figura 3.20: Contorno cisne con $\sigma = 15$, mejor visualización a la forma original del contorno.

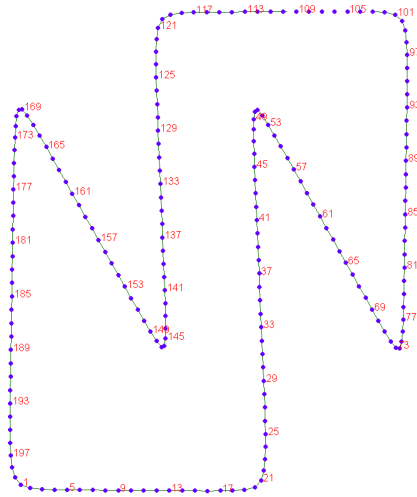


Figura 3.21: Contorno cisne con $\sigma = 20$, se preserva la forma original del contorno.

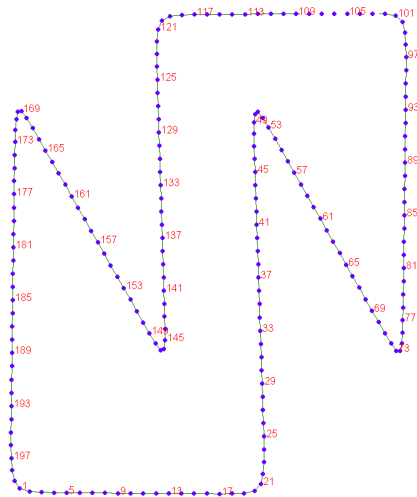


Figura 3.22: Contorno cisne con $\sigma = 25$, se preserva aún mejor la forma original del contorno.

De la misma forma al contorno de aplicación: contorno del lago de chapala figura 3.23, le aplicamos spline cúbico figura 3.24, notemos que suaviza las partes que se encuentran en pico en el contorno original. Hay que mencionar que para este tipo de contornos en donde se observa que tienen un exceso de información o bien tiene un exceso de puntos debemos de reducir ese número de puntos para un mejor funcionamiento del spline, esto lo podemos hacer al momento de aplicar spline en tensión.

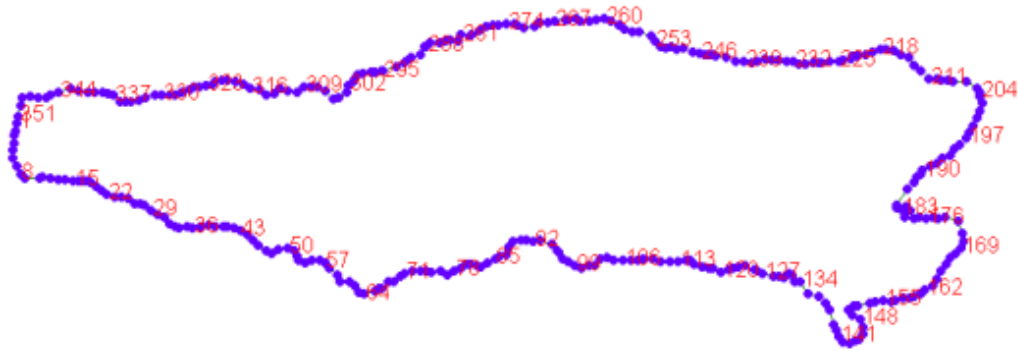


Figura 3.23: contorno original del lago de chapala.

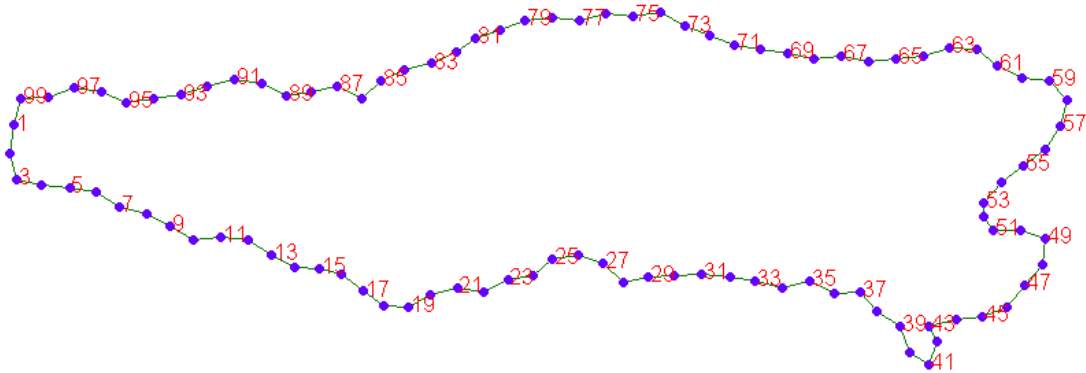


Figura 3.24: spline cúbico y reducción de puntos aplicados al contorno del lago de chapala.

Observemos como se preserva la forma aplicando spline en tensión .

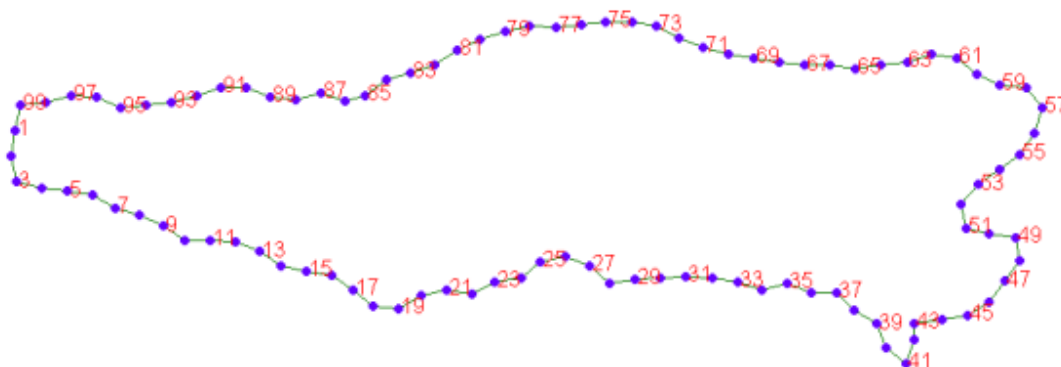


Figura 3.25: Contorno del lago de chapala con $\sigma = 5$, se observa el spline cúbico aplicado al contorno.

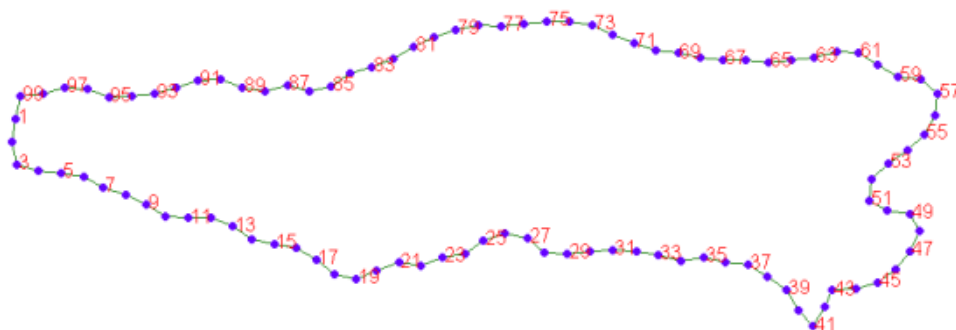


Figura 3.26: Contorno del lago de chapala con $\sigma = 15$. Se preserva la forma original del contorno.

Podemos decir que el spline cúbico es práctico, ya que la curva es una representación de polinomios cúbicos tanto para x como para y . Sin embargo como pudimos observar que en los contornos m19 y cisne, el contorno ha perdido su forma, hay aparentemente curvaturas y oscilaciones que los datos no presentan.

El siguiente ejemplo es una aplicación en el ala de un avión:

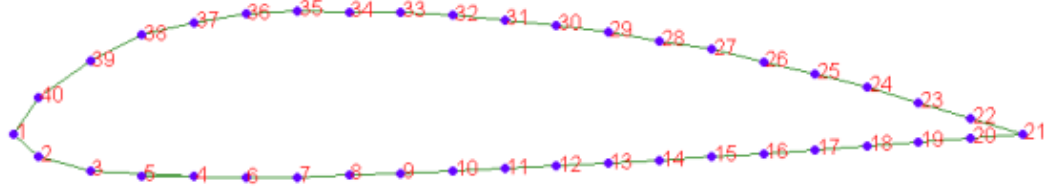


Figura 3.27: Corte transversal del ala de un avión.

El contorno que representa un corte transversal del ala de un avión parametrizado por un spline cúbico natural de interpolación, presenta oscilaciones, las cuales para medir el problema de fricción del aire, pueden ocasionar errores no controlables en su simulación. Se le dio un tratamiento aplicando spline en tensión, los resultados fueron los siguientes.

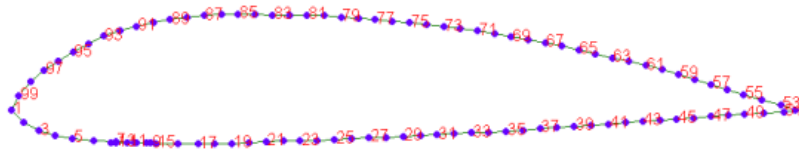


Figura 3.28: Ala de un avión con $\sigma = 5$. Preserva la forma original del contorno.

Los resultados son favorables cuando se prueba en el simulador de aire, ya no se observan esas turbulencias de aire que se formaban con esas oscilaciones, el problema de fricción con las corrientes de viento se reduce.

Capítulo 4

Editboundary: Sistema para el tratamiento de contornos

El objetivo de este trabajo fue explicar de una manera sencilla las bases teóricas que permiten que el spline en tensión sea una buena herramienta de suavizamiento de los contornos. Las pruebas hechas a los contornos que a lo largo del trabajo se han mostrado, fueron realizados en el sistema editboundary del grupo UNAMALLA. En este se le hace la implementación del método de spline en tensión como herramienta de suavizamiento.

4.1. Diseño e interfase de Usuario

El sistema Editboundary, es un paquete computacional que nos permite resolver de manera eficiente el problema de suavizamiento de contornos sobre regiones irregulares planas. El sistema cuenta con una interfase gráfico muy sencillo de usar. El sistema se ha diseñado para que sea semi - automático bajo la idea de que con muy pocas opciones de tecla sea posible obtener un tratamiento adecuado sobre la región. El sistema ha sido diseñado como un ambiente de ventanas ejecutable sobre MATLAB 2008 - 2010. El sistema consta de siete Menús de aplicaciones. Las opciones del sistema están escritas en inglés: `Main`, `Contour`, `View`, `reduction/Insert`, `smooth`, `Tools` `Info` y `Help`, el uso de cada menú y sus opciones o submenús estarán disponibles siempre que el proceso realizado lo permita. Por ejemplo, si se cuenta con un contorno en memoria podremos operar con la opción de suavizamiento o reducción, y mientras no se realice este post-procesamiento no estará disponible la información referente a los resultados del método aplicado. En las secciones siguientes haremos una descripción

de los módulos **Contour** e **Information** y describiremos cada una de las opciones del menú, cómo y cuándo es posible interactuar con cada una de ellas.

4.1.1. Interfase con el usuario

La interfase con el usuario se encuentra en la barra superior de la ventana de despliegue gráfico y para algunos módulos será desplegado algún sub-panel según la opción elegida. Los paneles consisten en un conjunto de botones, campos de entrada, controles deslizadores y cajas de diálogo.

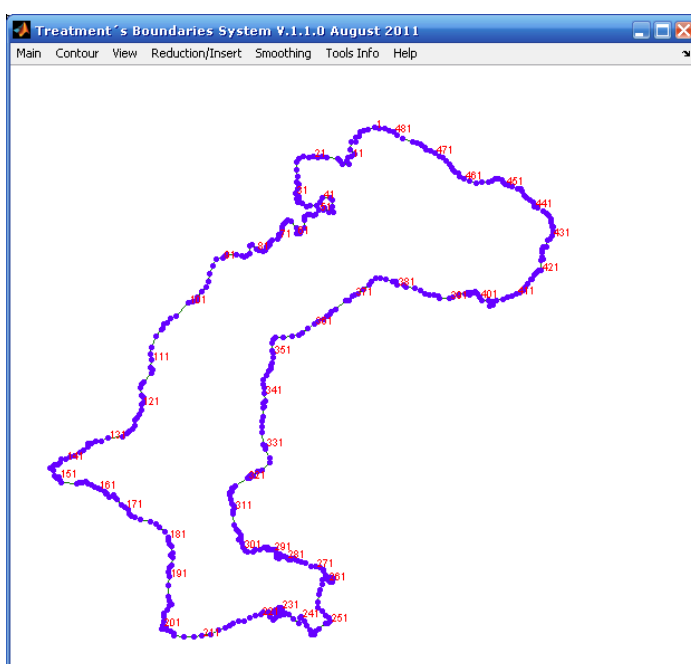


Figura 4.1: Ventana principal de editboundary, con un contorno inicial en pantalla.

De acuerdo con el diseño del sistema, el panel de control principal va cediendo el control a otros paneles de acuerdo al trabajo que se desea realizar. Los paneles posteriores son activados para su uso pulsando el área que los describe en el panel principal.

El usuario interactúa con el sistema a través de posicionarse y elegir la opción por dar *pointing/clicking* de los botones del mouse.

A continuación listamos una serie de operaciones que podemos realizar sobre los paneles.

- Para pulsar un botón u opción del panel, mueva el cursor sobre el botón y pulse uno de los botones del mouse.
- Cuando el indicador del mouse se encuentra sobre un botón del panel se desplegará una caja de títulos tipo `text` indicando un mensaje relativo a la opción donde se encuentra el indicador.

La ventana principal puede ser modificada en tamaño y mientras no se trabaje en alguna acción se pueden realizar acercamientos con la acción `Zoom On/Off` en el menú `View`.

4.2. Estructura modular

El sistema es modular, esto es, integra una colección de tareas que permiten interactuar entre ellas de manera que el usuario pueda experimentar con los elementos del sistema: la construcción del contorno y suavizamiento de contorno.

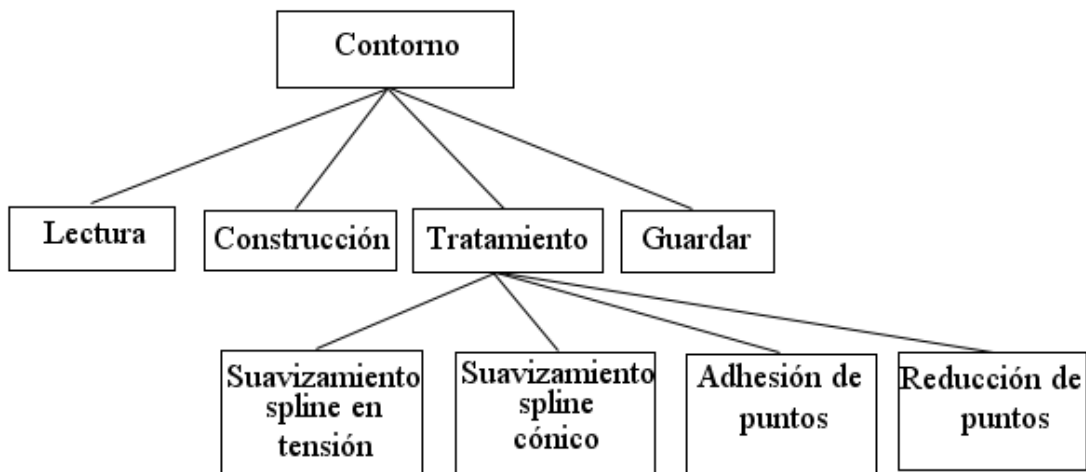


Figura 4.2: Estructura del sistema y dependencia de los módulos.

Como se describe en el diagrama esquemático de los módulos del sistema, la operación que puede realizarse con el sistema es: suavizamiento de contorno. Uno puede partir desde la información del contorno de la región previamente obtenida, para realizar algún suavizamiento. De igual manera puede considerar la información de un contorno, en archivo o en memoria y sobre ella operar el suavizamiento.

Nosotros en esta ocasión desarrollaremos el suavizamiento de contornos usando spline de tensión.

En los siguientes apartados describiremos los módulos principales del sistema y diremos la manera de interactuar con ellos.

4.3. Módulo de contornos

El conjunto de operaciones que podemos realizar con un contorno son: obtener la información del contorno a partir de un archivo de datos (con el formato convenido), construir un contorno poligonal en pantalla, guardar el contorno en uso y dar tratamiento al contorno ya sea por medio de un suavizamiento en tensión, por una adhesión uniforme de nodos o por una reducción de nodos por criterios de colinealidad.

El sistema contempla que en cada una de las opciones antes señaladas se cuente con un panel apropiado, cada uno cuenta con su propia ventana de visualización y sólo se visualizará el nuevo contorno en la ventana principal hasta ser concluido el tratamiento.

Al solicitar la opción **Contour** del menú principal aparecerá en pantalla el submenú con las opciones descritas.

Las opciones de **Open** y **New Poligon** son independientes de que exista o no algún contorno en memoria de ahí que estas opciones estén siempre disponibles.

Describamos cada una de las opciones disponibles del módulo **Contour**.

4.3.1. Construir un contorno en pantalla.

Bajo esta opción el usuario puede construir un contorno poligonal para ser usado durante el resto de su trabajo en el sistema. Esta opción abre una ventana de control con un espacio de graficación en el cual podrá capturar el contorno poligonal a partir de la colocación de vértices pulsando el botón izquierdo del mouse y pulsando el botón derecho del mouse para cerrar el contorno. Describamos cada una de las opciones integradas en éste panel.

New Con esta opción podemos volver a iniciar la captura de un contorno. Se eliminará de la memoria cualquier trabajo previo realizado en el área de trabajo.

- Remove Point** Con esta opción el usuario podrá deshacerse del último vértice colocado en el panel actual siempre que el contorno no este ya cerrado.
- Accept** Una vez obtenido el contorno cerrado esta opción finaliza la captura de los datos y nos regresa el módulo principal. Al pulsar la opción se llevará a memoria los datos del contorno cerrado definido y sobre éste se trabajará a partir de ese momento. Si el contorno no fue orientado positivamente el sistema invertirá el orden para que sea positivo.
- Help** Es una pequeña ayuda en línea, se abre el browser de Matlab y desde ahí se asesora en la creación de un contorno.
- Quit** Esta opción cancela todo trabajo hecho con este módulo y nos lleva de vuelta al módulo principal, se debe tener cuidado porque al salir no se conservarán los datos de un contorno previo en memoria ya que fueron borrados al optar por la opción **New Polygon**.

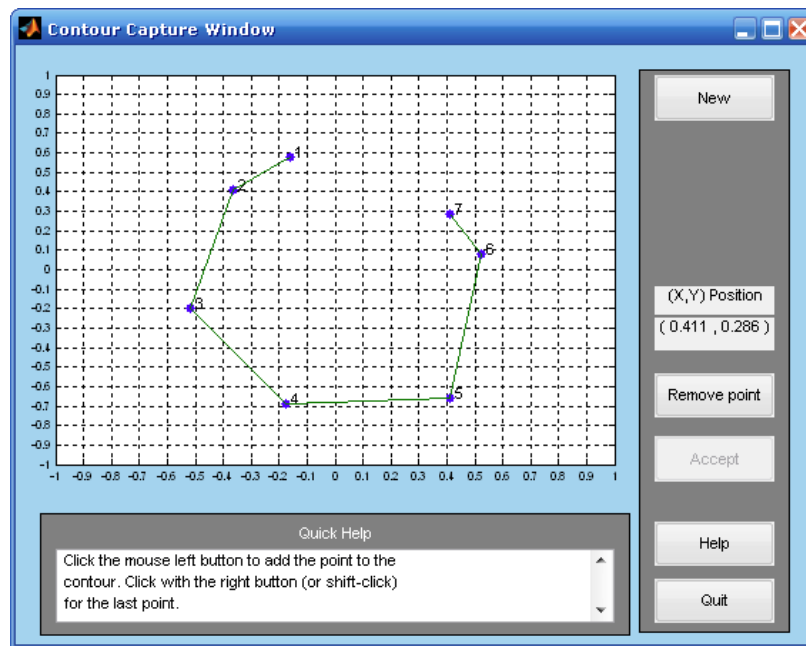


Figura 4.3: Ventana de captura de contorno.

Dentro de este panel puede observarse una caja de diálogo. En ella se desplegará una serie de breves señalamientos sobre el estado que guardan las acciones llevadas a cabo, ya sea que no se haya cerrado el contorno, que hagan falta puntos para

ello, que exista un contorno en memoria o bien que la ejecución haya sido satisfactoria y se cuente con el contorno en memoria.

Por otra parte se observa una caja donde se despliega la posición del mouse dentro del área de trabajo, esto con la finalidad de conocer la posición del mouse y del punto a elegir del contorno dentro del sistema coordenado.

4.3.2. Abrir un archivo de datos `Open`

La información de un contorno puede estar contenida en un archivo de documentos tipo ASCII en formato convenido y podemos tomar esta información para trabajar con ella a lo largo del sistema.

Estos archivos tienen una extensión de archivo `.con` y a través de un **browser** podemos tomar su información desde algún directorio. Al pulsar la opción `Open` del menú de contornos será desplegada la ventana desde la cual podrá el usuario localizar el archivo del contorno deseado. Esta caja de diálogo es la que se conoce como **browser** y consta de botones propios del sistema para aceptar el archivo elegido, reescanear el directorio, y presentar información de la localización del mismo.

Cada vez que el archivo sea elegido los datos pasarán a la memoria del sistema, es decir será posible trabajar en la región y esta será graficada en la ventana principal.

4.3.3. Guardar información `Save`

Una vez que el sistema cuente con un contorno en memoria es posible almacenar su información en un nuevo archivo o remplazar un archivo ya existente. Estando en el módulo contorno podemos pulsar la opción `Save` con lo cual aparecerá en pantalla un **browser** bajo el cual podemos indicar la ruta por búsqueda de directorios donde se hospedarán los archivos de datos del contorno. En la línea de entrada deberemos escribir el nombre del archivo que contendrá dichos datos. Si en el nombre del archivo omitimos la extensión de archivo `.con`, el sistema lo integra automáticamente.

Tanto la opción del **browser** como en la de `Open`, permanece en primer plano mientras que el panel principal se encuentra inactivo hasta que la operación haya sido concluida. La información del contorno en memoria es guardada en el formato convenido `.con`. En la sección correspondiente a los formatos de archivos se detalla dicho formato.

4.3.4. Tratamiento por reducción de puntos

El módulo de tratamiento por adhesión de puntos es activado una vez que se tiene cargado un contorno, y se mantiene activo a partir de ese momento. Daremos enseguida una explicación de los que representa cada parámetro en en el interfase de reducción de puntos.

K Número de puntos a tomar para aplicar los criterios de colinealidad.

R0 Parámetro 1 del criterio de longitud, para la manipulación de los parámetros utilice las barras deslizables.

R1 Parámetro 2 del criterio de longitud, recuerde que $1 < R0 < R1$.

E0 Parámetro 1 del criterio de error máximo.

E1 Parámetro 2 del criterio de error máximo, recuerde que $1 < R0 < R1$.

Ahora explicaremos lo que permiten hacer cada boton que aparece en el interfase de reducción de puntos.

Apply Una vez establecido el parámetro con este botón realiza la adhesión de puntos, se activan los botones posteriores.

Aceptar Este botón permite escoger el contorno construido, se cerrará la ventana de tratamiento por adhesión de puntos y con ello se cederá el control al panel principal.

Cancel Esta opción cancela todo trabajo hecho con este módulo y nos lleva de vuelta al módulo principal. Al salir no se conservarán los datos del tratamiento.

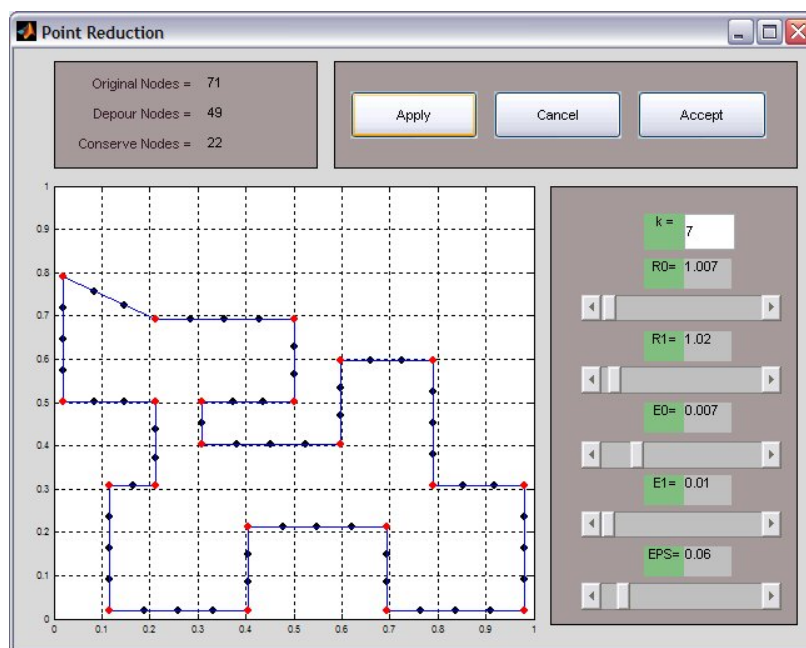


Figura 4.4: Ventana de tratamiento por reducción de puntos.

4.3.5. Tratamiento del contorno utilizando spline en tensión

Cuando se abre o se crea un contorno, uno de los tratamientos que puede recibir este contorno es el suavizamiento ya sea aplicando el spline cónico o bien el spline en tensión el cual para este trabajo es de nuestro interés. Para aplicar spline en tensión se desplegará una pantalla que permitirá al usuario elegir el factor en tensión y a la vez el número de puntos que se quieren considerar en el contorno.

El coeficiente en tensión tendrá un valor máximo de 85. o un mínimo de 0.0, si el usuario introduce un valor mayor a 85, el factor en tensión regresará al valor de 8 que esta predeterminado. Cuando elegimos aplicar spline en tensión para el suavizamiento del contorno, uno de los botones que aparece en el interfase es el Histogram, en el cual al pulsarlo podemos consultar el área y perímetro del contorno.

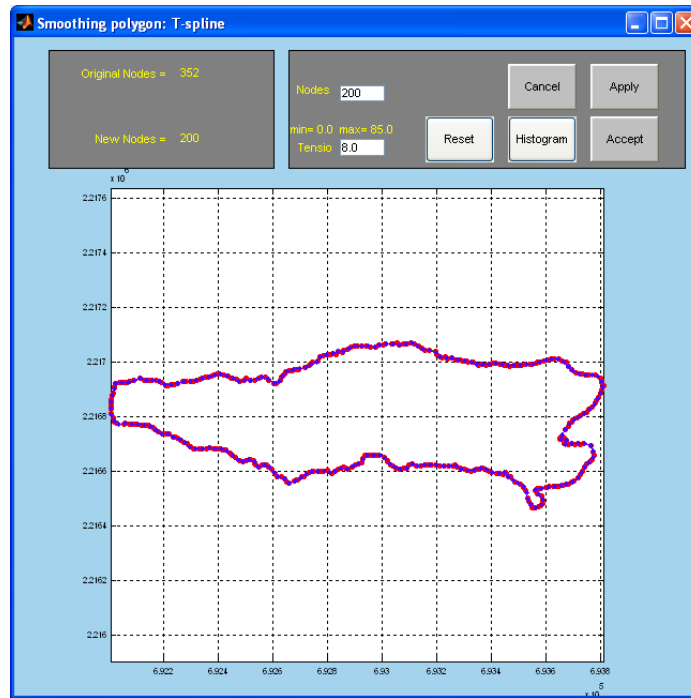


Figura 4.5: Suavizamiento con spline en tensión

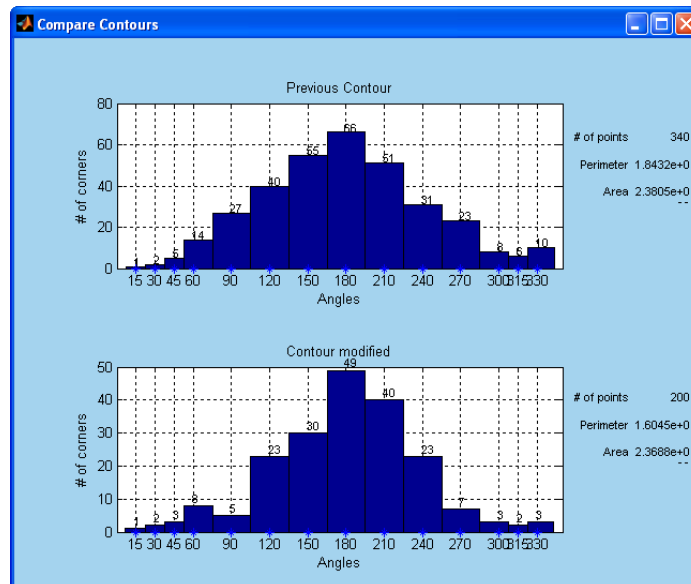


Figura 4.6: Histograma que nos proporciona el sistema de editboundary

Los paneles consisten en un conjunto de botones, campos de entrada y cajas de dialogo.

Botones.

| | |
|------------------|--|
| Apply | Una vez establecido el parámetro de tensión y el número de puntos en el contorno, con este botón se activan los botones posteriores. |
| Aceptar | Este botón permite escoger el contorno construido, se cerrará la ventana suavizamiento de contorno con spline de tensión y con ello se cederá el control al panel principal. |
| Cancel | Esta opción cancela todo trabajo hecho con este módulo y nos lleva de vuelta al módulo principal. Al salir no se conservarán los datos del tratamiento. |
| Histogram | Este botón permite consultar el área y perímetro del contorno original y modificado. De igual manera nos permite saber el número de ángulos y su valor en el contorno original o modificado. |
| Reset | Este botón nos permite borrar los datos elegidos hasta el momento y regresar a los datos iniciales del contorno (el contorno no tiene aplicado spline en tensión), cuando se escoje esta opción el parámetro regresa al valor de 8.0 que se tiene por default. |

Capítulo 5

Conclusiones

5.1. Conclusiones

En este trabajo fue abordado el problema de interpolación por spline en tensión sobre contornos cerrados o abiertos. Desarrollamos un módulo para ser usado en el sistema Editboundary del grupo UNAMALLA empleando las rutinas de Renka (algoritmo 716). Describimos con detalle los fundamentos básicos de la teoría de spline en tensión. En resumen:

- El uso de los spline en tensión evita oscilaciones que se presentan en los spline cúbicos.
- La variación del factor de tensión nos permite pasar de un spline cúbico a un spline lineal
- Para el sistema Editboundary del grupo UNAMALLA, el spline en tensión es una herramienta útil y no costosa para los problemas de aplicación y académicos para los cuales el sistema se ha usado.

Trabajo Futuro

Quedan varios caminos de trabajo que pueden seguirse a partir de lo que hemos desarrollado, en particular estamos interesados en lo siguiente:

- Que el usuario pueda elegir el parámetro de tensión de manera automática.
- Incorporar la técnica de suavizamiento de datos por spline en tensión.

- Incorporar al sistema Editboundary otras técnicas de suavizamiento.

Apéndice A

Reducción de puntos

Representemos una región en el plano, como la siguiente figura A.1.

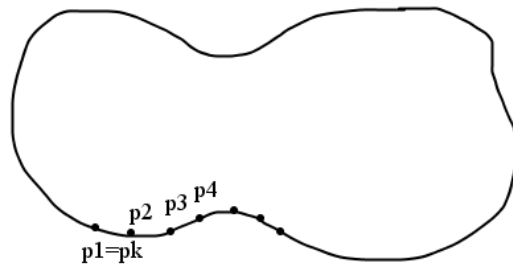


Figura A.1: region

Sea $\Gamma = \{p_1, \dots, p_m\}$ una región acotada, simplemente conexa y con frontera, formada por un conjunto de puntos con una orientación. Estos puntos definirán a un polígono cerrado es decir $p_1 = p_m$.

Como objetivo se quiere una representación parecida de Γ . Esta representación consiste en construir un polígono que cumpla con aproximar al contorno original y respetando la forma de la curva, reduciendo el número de puntos ya que en ocasiones Γ puede tener demasiada información, es decir que tenga puntos alineados o representativos de la curva Figura A.2.

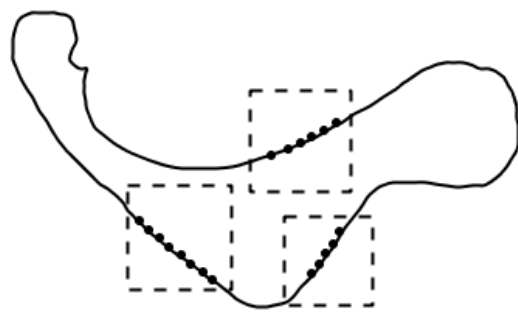


Figura A.2: Alineados.

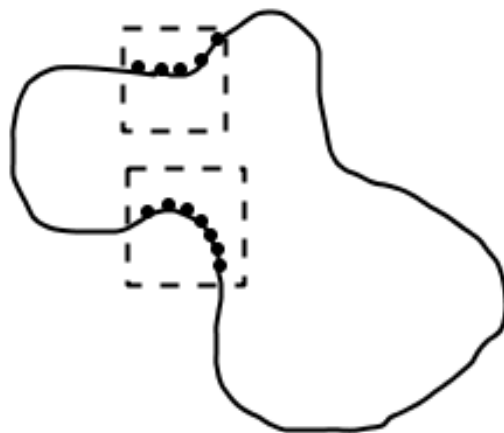


Figura A.3: Representativos de la curva.

Cuando se tienen estas particularidades hay que darle un tratamiento a la región. En especial se trabajara con puntos alineados.

El tratamiento consistira en eliminar puntos que esten *casi alineados*, este es un proceso en el que para poder efectuarlo necesitamos de ciertos criterios.

Se explica a continuación en que consiste cada criterio, son tres y se les llamara criterios de colinealidad [9].

A.1. Criterios de colinealidad

A.1.1. Criterio de Error Máximo

La idea de este criterio es encerrar en una banda una colección de k puntos, donde los extremos de estos formaran el segmento $\overline{p_1 p_k}$ y el resto serán los $k-2$ puntos interiores. Denotemos $d_{\text{máx}}$ como la distancia máxima que existe del punto más alejado p_j , con respecto al segmento $\overline{p_1 p_k}$

$$d_{\text{máx}} = \max_{j=2, \dots, k-1} d(p_j, \overline{p_1 p_k})$$

Se hará uso de una cota E_0 , que será la longitud de la mitad de la banda, esta nos servirá de parámetro para decir si el p_j se encuentra dentro de la banda completa de longitud $2E_0$ Figura A.4.

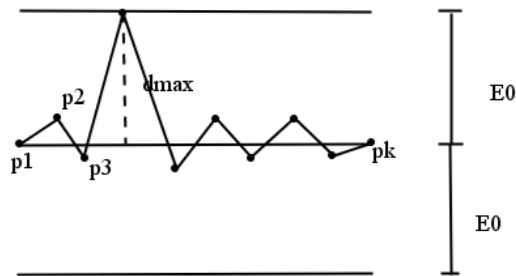


Figura A.4: Criterio de error máximo

Si $d_{\text{máx}}$ es menor que E_0 todos los k puntos están en la banda y los $k-2$ puntos se pueden eliminar.

Esta condición puede ser muy estricta porque si E_0 es muy grande puede ocurrir que existan puntos muy pequeños (muy cercanos al segmento $\overline{p_1 p_k}$) no se detecten Figura A.5.

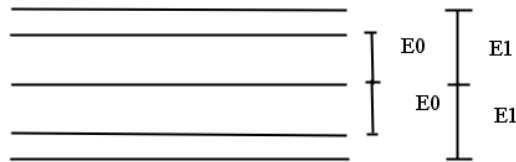


Figura A.5: Cotas E_1 y E_0

Para que el criterio siga satisfaciendo al caso anterior, se ve en la necesidad de utilizar un parámetro más, al cual llamaremos E_1 , de modo que E_0 sea menor a E_1 .

Con estos parámetros se pueden presentar los siguientes casos:

Si $d_{\text{máx}}$ es menor o igual que $E0$, se aceptara la colinealidad de los puntos.

Si $d_{\text{máx}}$ es mayor o igual que $E1$, se rechaza la colinealidad de los puntos.

O en el último de los casos

Si $d_{\text{máx}}$ se encuentra entre $E0$ y $E1$ no se puede afirmar nada sobre la colinealidad de los puntos y se recurre a otro criterio [9]. El valor de la cota de $E0$, va a depender de las características que tenga el problema a resolver.

A.1.2. Criterio de Longitud

Los extremos de una colección de *puntos alineados* definirán un segmento $\overline{p_1 p_k}$. Puede pasar que los k puntos, sumadas sus longitudes que existe entre cada par de los k puntos definidos formen un segmento que no diste del segmento $\overline{p_1 p_k}$. Así que se definirá lo siguiente:

$$S = d(p_1, p_k)$$

$$D = \sum_{i=1}^{k-1} d(p_i, p_{i+1})$$

y se compara S y D con el fin de verificar que tan *parecido* es el segmento S con respecto a D . Como lo muestra la siguiente figura FigureA.6.

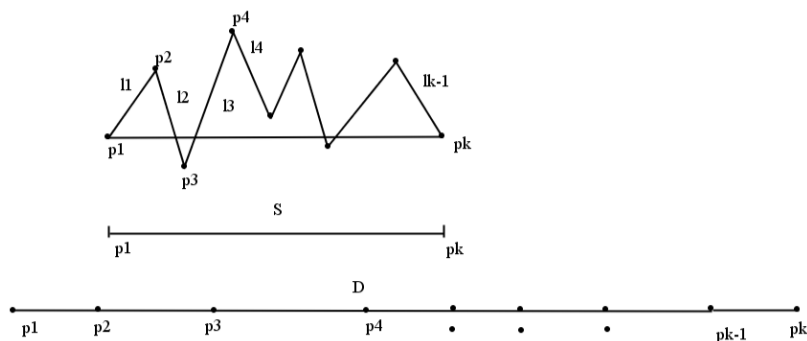


Figura A.6: Criterio de longitud

Para la comparación del segmento S y D se hace uso del cociente de estos dos y diremos que S se parece a D siempre que $\frac{S}{D}$ sea cercano a uno, lo cual es una condición que no se puede cumplir fácilmente ya que en realidad D siempre será más grande que S . Por tal motivo el criterio se ve obligado a considerar otras cotas mayores a uno y así no ser tan estricto con la condición, es decir conciderar unas cotas mayores para que S se parezca a D . Las cotas de las que se habla son $R0$ y $R1$, donde $R0$ es mayor que uno y $R1$ es mayor que $R0$.

Si el cociente es menor a $R0$ entonces los puntos son colineales y bien se pueden eliminar los $k-2$ puntos.

Si el cociente es mayor o igual que $R1$, los puntos no son colineales.

Si el cociente se encuentra entre $R0$ y $R1$, se aplica otro criterio de colinealidad.

El valor de la cota de $R0$ y $R1$, van a depender de las características que se tengan que cumplir en el problema que se quiera resolver.

A.1.3. Criterio de Cruce

Este criterio toma en cuenta el número de veces que la curva atravieza el segmento de ajuste. Si la curva atraviesa al segmento de ajuste una sola vez, entonces $d_{\text{máx}}$ es $d_{\text{máx}}/2$. Esta decisión se toma ya que puede darse el caso en que $d_{\text{máx}}$ es muy grande y hay puntos sobre la banda que son muy pequeños, en el sentido de su distancia con respecto al segmento $\overline{p_1 p_k}$, entonces se les ayuda haciendo a $E0$ más pequeño como la siguiente figura lo muestra A.7.

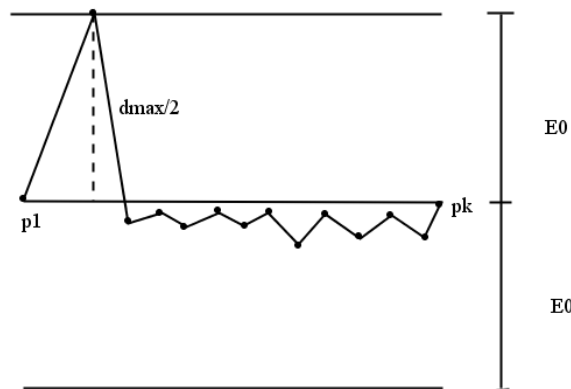


Figura A.7: La curva atraviesa el segmento de ajuste una sola vez

Si pasa lo contrario entonces $d_{\text{m}\acute{\text{a}}\text{x}}$ ser\acute{a} $d_{\text{m}\acute{\text{a}}\text{x}}/4$ A.8.

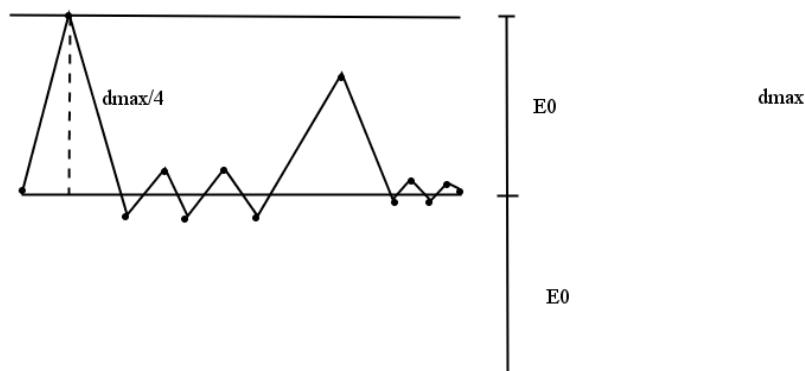


Figura A.8: La curva atraviesa el segmento de ajuste solo una sola vez

Estos nuevos valores de $d_{\text{m}\acute{\text{a}}\text{x}}$ se compararan con $E0$ que es la cota que se hab\iacute;a manejado en el criterio de Error m\acute{a}ximo [9].

Se puede decir que el criterio de cruce y el criterio de error m\acute{a}ximo comparten el par\ametro $E0$ pero con distinta $d_{\text{m}\acute{\text{a}}\text{x}}$.

A.1.4. C\omodo se implementa el algoritmo de eliminaci\on de puntos usando los criterios de colinealidad

Con el uso de los criterios de colinealidad se sabe si para un conjunto de k puntos que definen un segmento $\overline{p_1 p_k}$, los $k - 2$ intermedios se pueden eliminar. Esto es solo para un segmento $\overline{p_1 p_k}$ pero sobre el contorno se formar\an varios de estos, a los cuales a uno por uno se le estudiara y se determinara si se acepta o bien se rechaza por no cumplir con las caracter\xedsticas que pida alguno de los criterios de colinealidad.

Cuando el usuario determina el n\umero de puntos que conformaran a cada uno de los segmentos $\overline{p_1 p_k}$, el algoritmo proceder\a estudiar el segmento i -\acute{e}simo. Si este satisface alg\un criterio de colinealidad, se acepta y se sigue con el estudio del siguiente segmento. Si es rechazado se procede a elegir un nuevo segmento, el cual se determinara con el punto inicial p_i del segmento anterior rechazado y el punto p_j que satisface a $d_{\text{m}\acute{\text{a}}\text{x}}$; si ese segmento nuevo a estudiar no es aceptado nuevamente, se repite el paso anterior y as\i sucesivamente las veces que se requiera, se puede llegar al caso de que solo queden dos puntos. La finalidad de este proceso es poder aceptar el segmento y as poder seguir el estudio del siguiente, como lo muestra el siguiente esquema A.9.

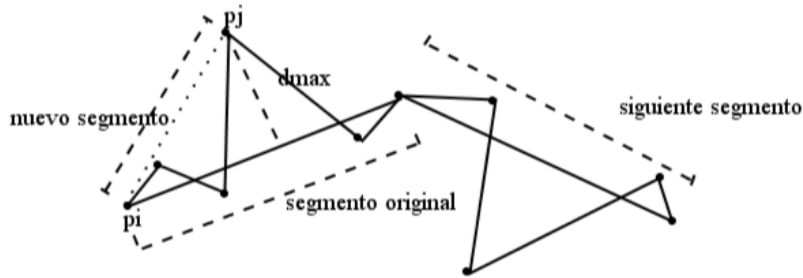


Figura A.9: algoritmo de eliminación de puntos

En el proceso anterior no podemos descartar la idea de que se tengan dos segmentos consecutivos $\overline{p_i p_k}$ que formen un ángulo muy cercano a 180 grados. Para evitar esto dentro del algoritmo se conserva el punto inicial p_i del segmento i -ésimo y si el segmento $i+1$ es aceptado, antes de ser incorporado al nuevo contorno, se compara con el segmento anterior, si ambos segmentos forman un ángulo que se acerque a 180 grados, se elimina el punto p_m que une a estos dos segmentos y el nuevo segmento que se incorpora al nuevo contorno tiene como puntos extremos p_i y p_f (punto final del segmento $i+1$), como la siguiente figura A.10.

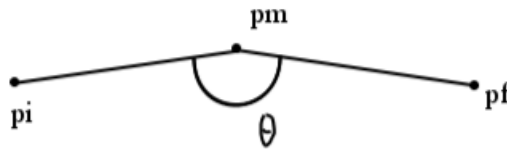


Figura A.10: ángulo entre dos segmentos, cercano a 180 grados

Para esta comparación de θ de que si este muy cercano a 180 grados o bien es muy pequeño, se lleva a cabo un procedimiento en el cual se incorpora un nuevo parámetro llamado EPS el cual tendrá un valor muy cercano a cero ya que será usado de la siguiente forma, si el valor absoluto de $\cos\theta - 1$ es menor igual a EPS entonces se eliminara al punto que une los segmentos $i, i + 1$ y no es considerado el segmento $i + 1$ como parte del nuevo contorno, quedando así el nuevo contorno $\overline{p_i p_f}$, como lo muestra la siguiente representación A.11.

El procedimiento para determinar si los puntos de un segmento $\overline{p_i p_k}$ son colineales es el siguiente: el usuario introduce el número de puntos que definira al segmento $\overline{p_i p_k}$, a la vez que intrduce el valor de las cotas $E0, E1, R1$ y $R0$, las dos primeras del criterio de longitud y las restantes del criterio de error máximo. Si $\frac{S}{D}$ del criterio de longitud se encuentra en el intervalo de incertidumbre pasamos al criterio de Error

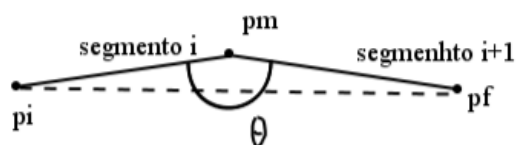


Figura A.11: Segmento $\overline{p_i p_f}$

máximo y si d_{max} se encuentra en el intervalo de incertidumbre, se aplicará el criterio de cruce y aquí se define si el segmento $\overline{p_1 p_k}$, se acepta o se rechaza.

El siguiente digrama de flujo representa los pasos anteriores A.12.

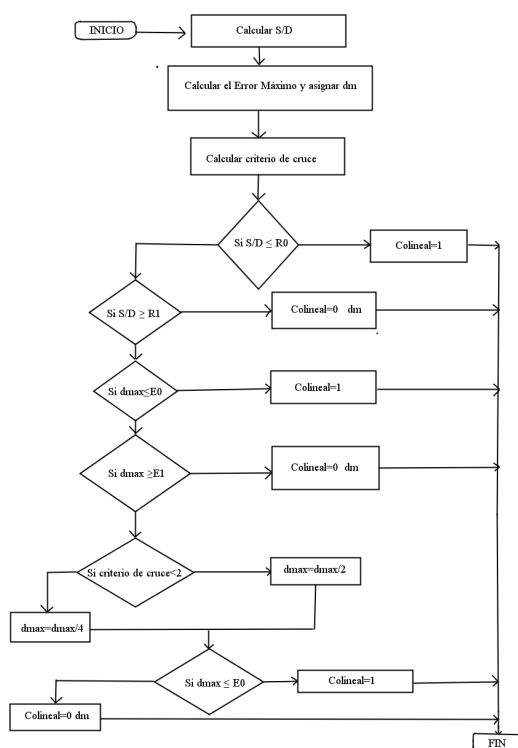


Figura A.12: Diagrama de flujo [9].

Se mostrara a continuación el contorno del Lago Tuba al cual se eliminaron puntos utilizando los criterios de colinealidad, de la misma forma se presenta las observaciones que se obtuvieron A.13.

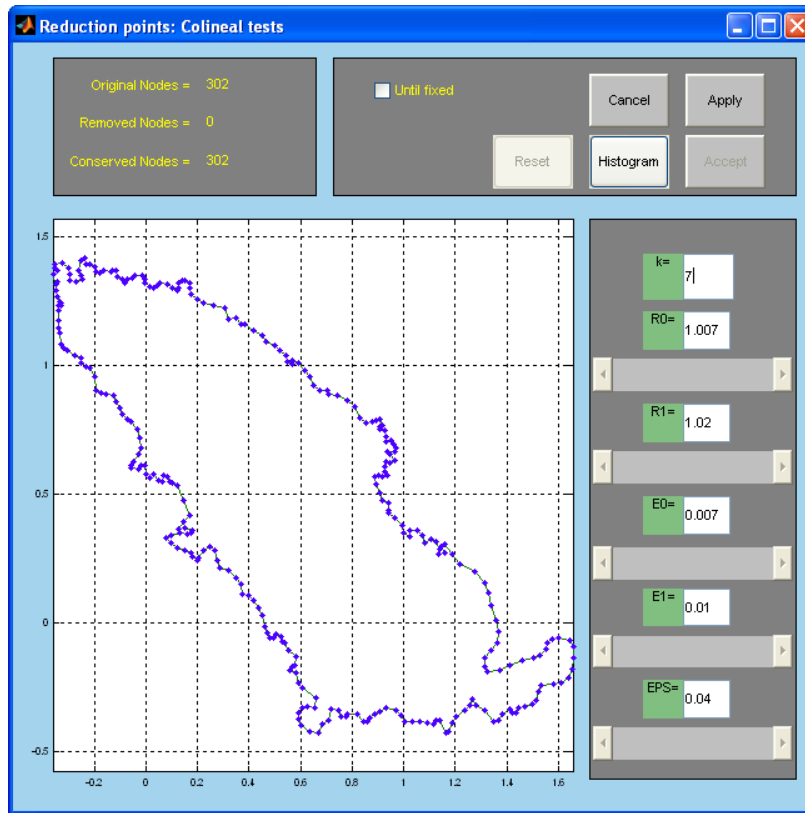


Figura A.13: Lago Tuba.

Observamos que no hay puntos eliminados y se tienen 302 puntos originales. Al efectuar un cambio en el número de puntos elegidos para el segmento $\overline{p_1 p_{11}}$ con $k=11$ obtenemos lo siguiente A.14.

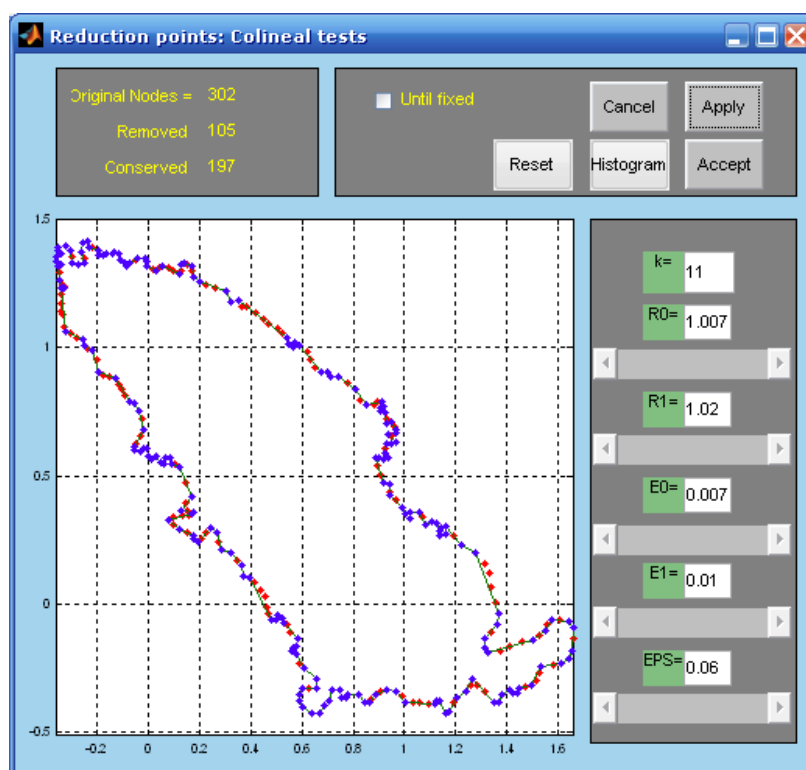


Figura A.14: Lago Tuba con $k=11$

Hay 105 puntos removidos con $EPS=0.06$.

EPS va a medir que tan cerca esta el coseno del ángulo de la unidad, si EPS es muy pequeño, la aproximación a los picos de la curva será mejor y si esto pasara no se eliminaran muchos puntos. Por pruebas que se han realizado, encontraron que $EPS=0.06$ ayuda a tener una buena aproximación ya que se conservan los segmentos $\overline{p_1 p_{11}}$ que forman un ángulo no mayor a 160 grados. La cota EPS determina la colinealidad de los puntos que definen dos segmentos $\overline{p_1 p_{11}}$ consecutivos, es decir se reemplazara dos segmentos $\overline{p_1 p_{11}}$ por uno solo.

De la prueba anterior se observo que hay puntos que se eliminaron y en consecuencia cambio de forma mínima algunas partes del contorno original, sustituyendolas por una segmento $\overline{p_1 p_{11}}$. Recordemos que los criterios de colinealidad al usarlos para reducir puntos pretenden conservar lo más que se pueda al contorno original.

Se propone ahora no solo modificar el número de puntos de puntos que conforman al segmento $\overline{p_1 p_{11}}$, se dara un valor de $E0=0.02$ y a $E1=0.04$, es decir se incremento la longitud de la mitad de la banda que se trato en el criterio de error máximo.A.15.

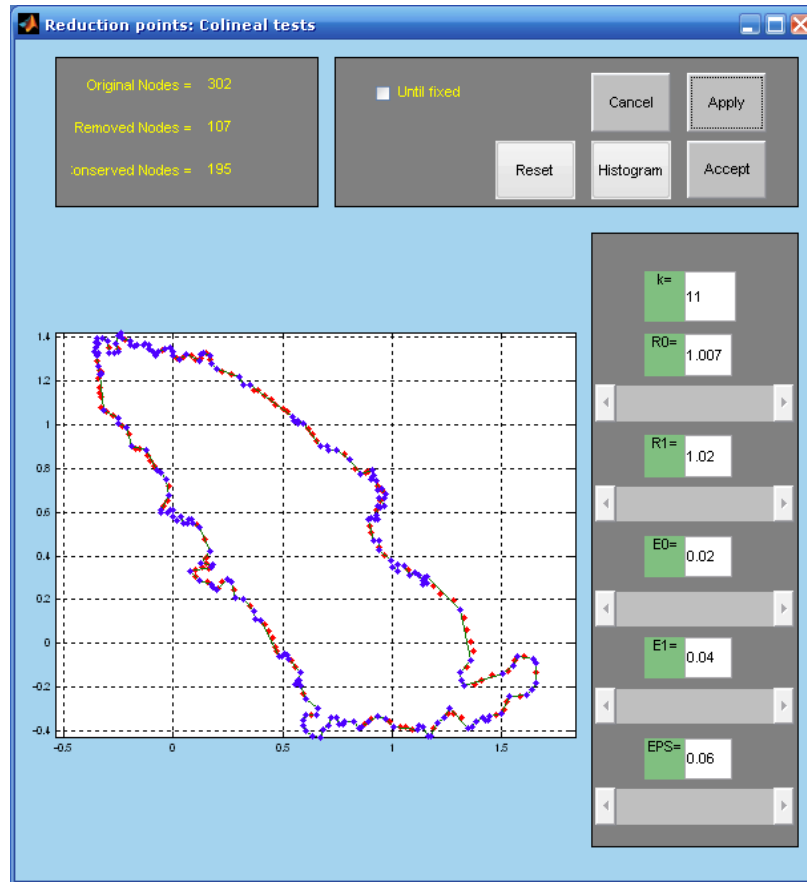


Figura A.15: Lago Tuba con $k=11$, $E_0=0.02$ y $E_1=0.04$

Notemos que no se modifica de una manera significativa el contorno pero aumento en dos el número de puntos eliminados, esto lo podemos ver en algunos trozos de la curva donde hay puntos que se eliminaron y ese tramo de la curva se ve ahora como un segmento $\overline{p_1 p_{11}}$.

Ahora bien el valor de $E_0=0.007$ a $E_0=0.02$ y de $E_1=0.01$ a $E_1=0.04$ son valores muy pequeños porque estos dependerán de la escala con la cual se este trabajando, aquí por ejemplo la escala de este contorno con el cual se esta trabajando es de -0.5 a 1.5 y de -0.4 a 1.4, realmente son valores pequeños, entonces E_0 y E_1 son proporcionales a la escala del contorno. Como resultado de la prueba anterior que se venia haciendo es la siguiente: A.16.

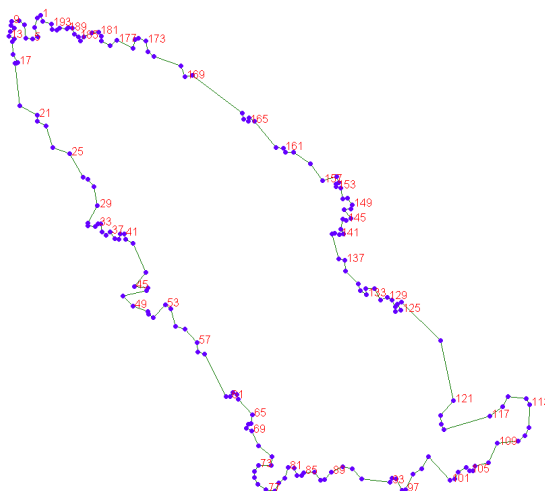


Figura A.16: Lago Tuba con $k=11, E_0=0.02$ y $E_1=0.04$

Se han cambiado los valores a K , E_0 , E_1 y se ven cambios mínimos con respecto al contorno original, pero también no se debe pasar desapercibido que $R_0=1.007$ y $R_1=1.02$ están activados (estas cotas recordemos que corresponden al criterio de longitud) por tanto no podemos afirmar con seguridad si al modificar a E_0 y E_1 , solo se está utilizando el criterio de error máximo porque R_0 y R_1 del criterio de longitud están activados.

En la siguiente prueba se modificara a $R_0=1.001$ y $R_1=1.01$, E_0 y E_1 se dejan con los valores anteriores.A.17.

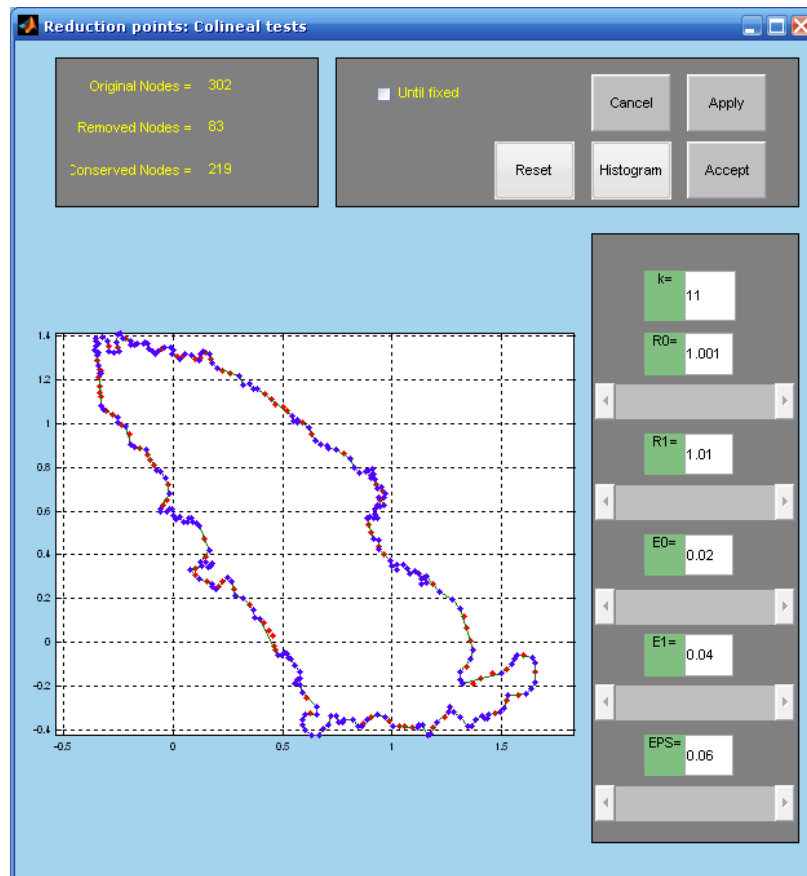


Figura A.17: Lago Tuba con $k=11$, $E0=0.02$, $E1=0.04$, $R0=1.001$ y $R1=1.01$

Se han removido 83 puntos menos que cuando solo se había modificado $E0$ y $E1$, aunque un trozo mínimo del contorno se sigue cambiando por el segmento $\overline{p_1 p_{11}}$. Hagamos otro cambio $R0=1.01$ y $R1=1.02$ como se muestra a continuación en la figura A.18.

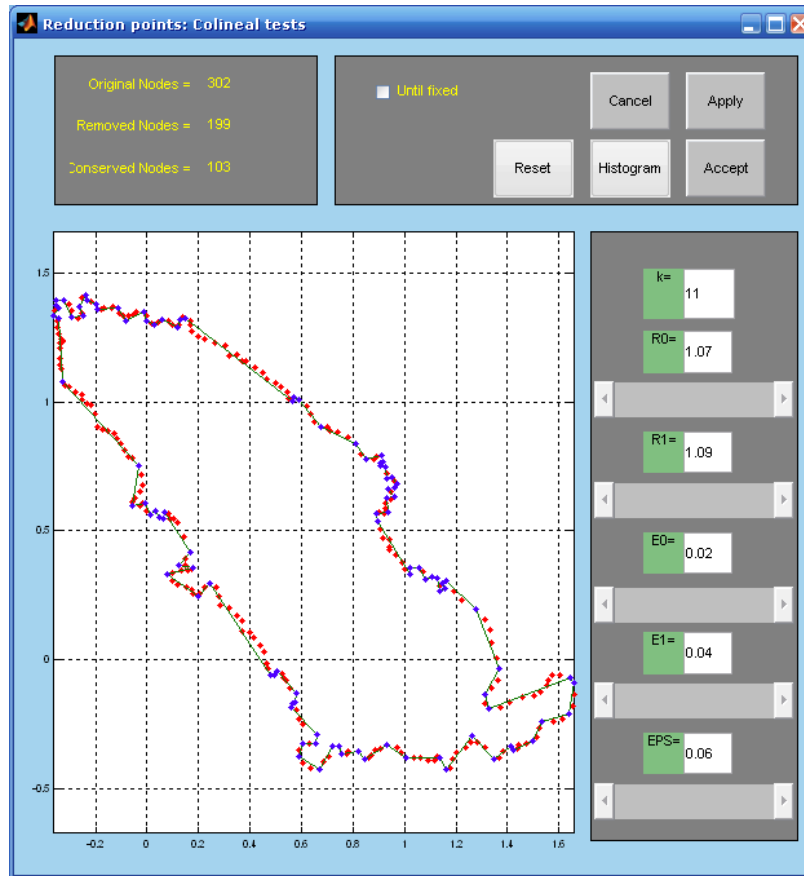


Figura A.18: Lago Tuba con $k=11$, $E_0=0.02$, $E_1=0.04$, $R_0=1.01$ y $R_1=1.02$

No fue buena propuesta, se eliminaron 199 puntos y el contorno original ha tenido un mayor número de cambios.

Otro cambio $R_0=1.007$ y $R_1=1.009$ aparece nuevamente el pedazo de curva que es cambiado por el segmento $\overline{p_1 p_{11}}$. Se han removido 83 puntos A.19.

Con $EPS=0.01$ se han removido 66 puntos menor a lo que se había reducido anteriormente.

Con $EPS=0.03$ se han removido 69 puntos.

Con $EPS=0.05$ se han removido 74 puntos.

Con $EPS=0.06$ se han removido 83 puntos.

Con $EPS=0.09$ se han removido 110 puntos.

Con este último resultado no se ve bien la aproximación a la curva A.20.

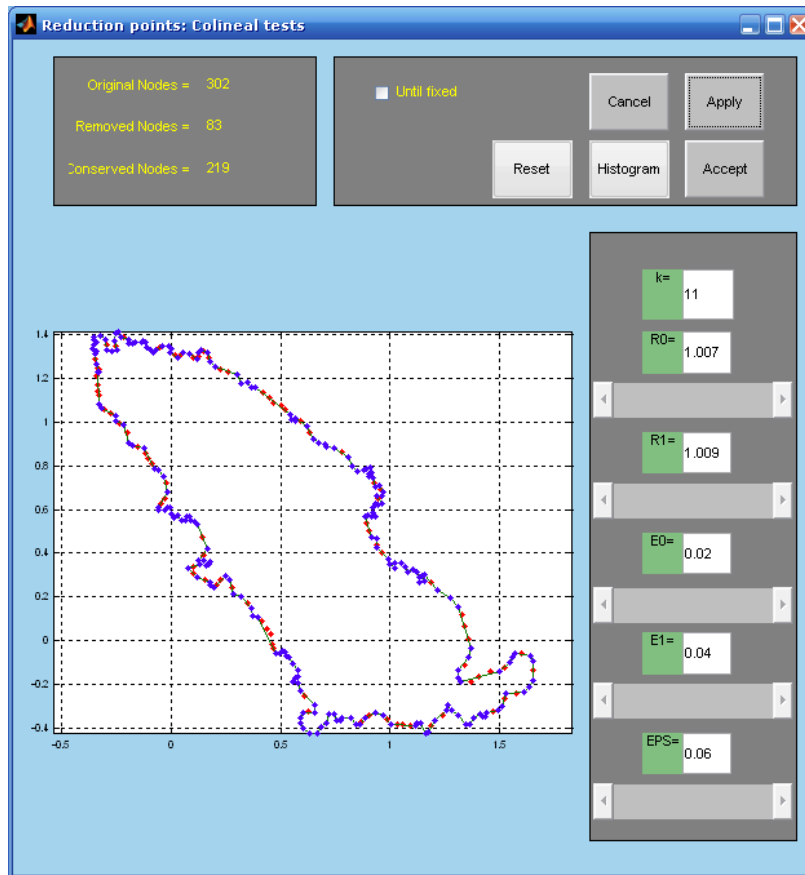


Figura A.19: Lago Tuba con $k=11$, $E_0=0.02$, $E_1=0.04$, $R_0=1.007$ y $R_1=1.009$

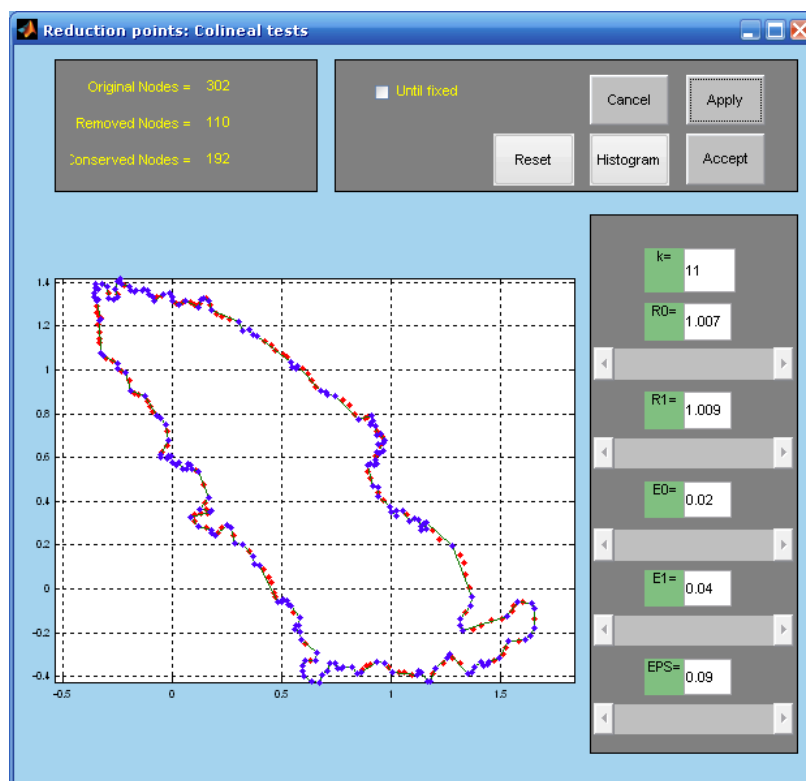


Figura A.20: Lago Tuba con $k=11$, $E_0=0.02$, $E_1=0.04$, $R_0=1.007$, $R_1=1.009$ y $EPS=0.09$

Con $EPS=0.08$ se han removido 109 puntos, se observa mejor que la prueba anterior A.21.

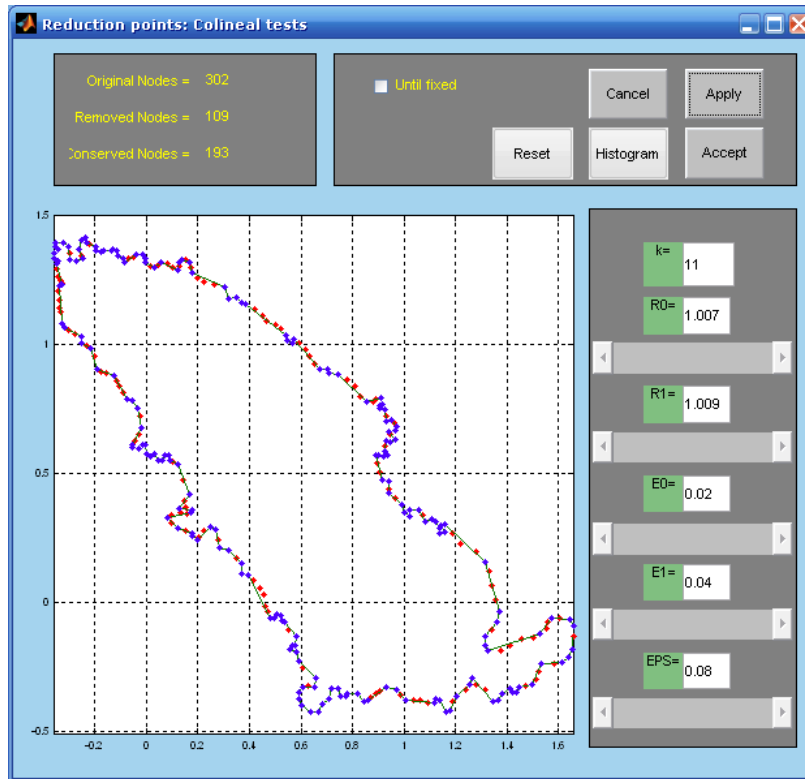


Figura A.21: Lago Tuba con $k=11$, $E_0=0.02$, $E_1=0.04$, $R_0=1.007$, $R_1=1.009$ y $EPS=0.08$

Con $EPS=0.07$ se han removido 100 puntos, aunque en apariencia el contorno quedo como la prueba anterior A.22.



Figura A.22: Lago Tuba con $k=11$, $E0=0.02$, $E1=0.04$, $R0=1.007$, $R1=1.009$ y $EPS=0.07$

Se queda el valor de $EPS=0.08$, se elimina un número adecuado de puntos (109 puntos) y es visiblemente es una buena aproximación a la curva A.23.

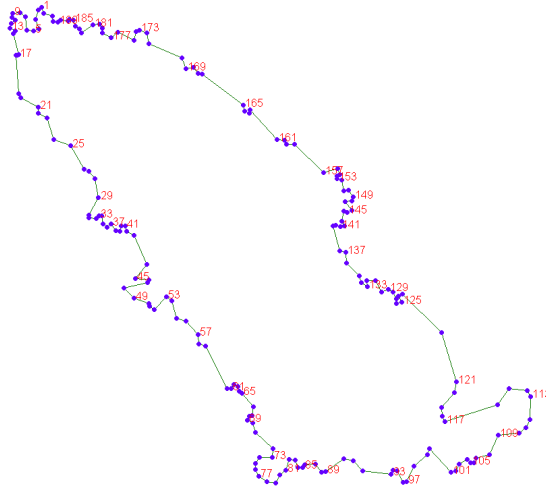


Figura A.23: Lago Tuba con $k=11$, $E_0=0.02$, $E_1=0.04$, $R_0=1.007$, $R_1=1.009$ y $EPS=0.08$

Si nos fuésemos a los extremos con $EPS=0.6$, definitivamente no sería buena elección A.24.

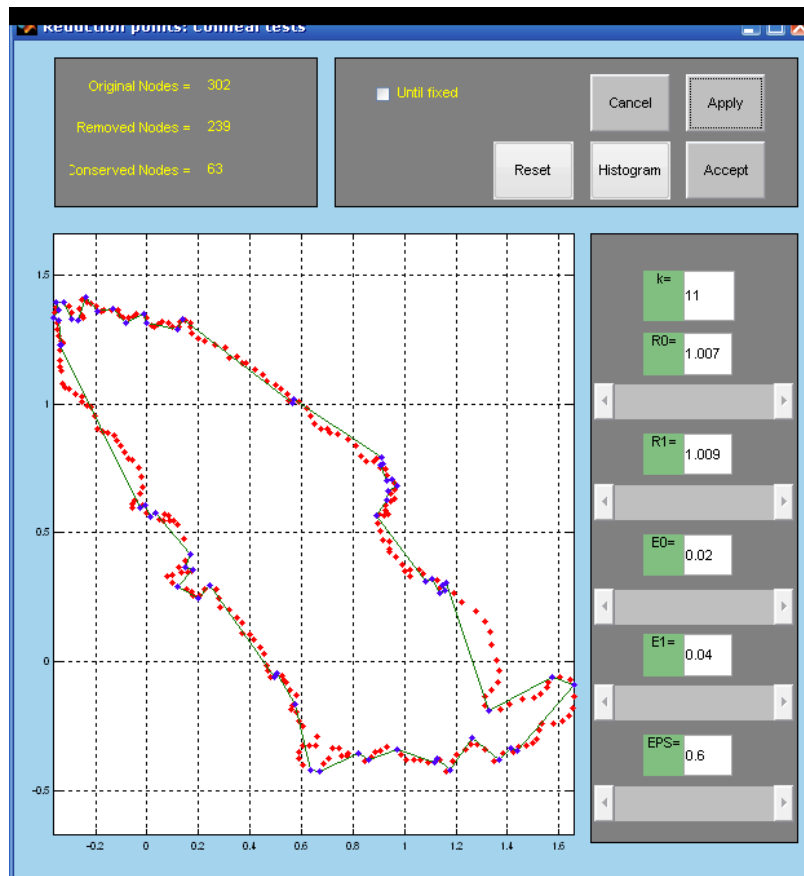


Figura A.24: Lago Tuba con $k=11$, $E0=0.02$, $E1=0.04$, $R0=1.007$, $R1=1.009$ y $EPS=0.6$

De la opción elegida se puede comparar el perímetro y área del contorno original contra el resultante A.25.

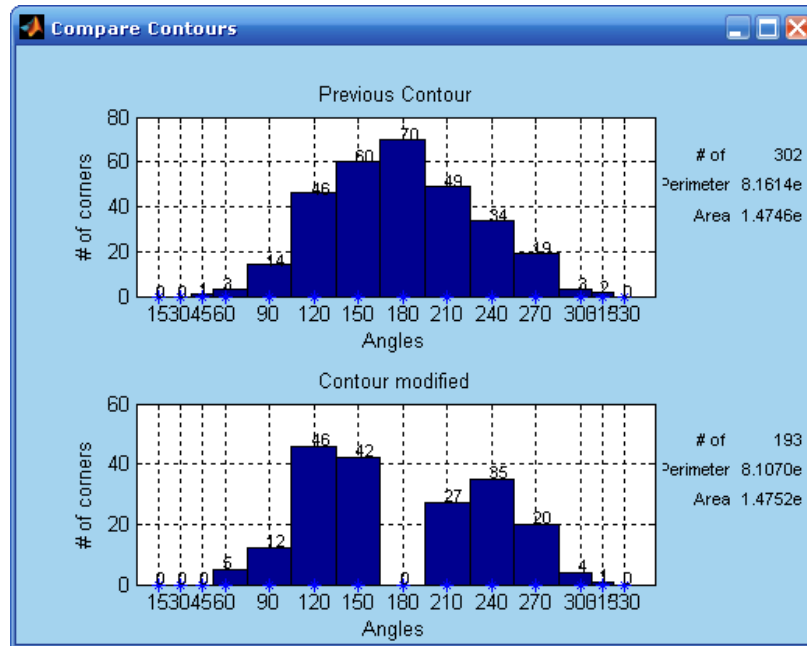


Figura A.25: Lago Tuba con $k=11, E_0=0.02, E_1=0.04, R_0=1.007, R_1=1.009$ y $EPS=0.08$

La diferencia de área es casi nada y eso es buena señal porque entonces el contorno solo se modificó muy poco, en cambio hay una disminución de perímetro (es poca la cantidad) pero era de esperarse ya que se eliminaron puntos.

Bibliografía

- [1] Barrera P., Hernández V. y Durán C., *El ABC de los splines*, Sociedad Matemática Mexicana, 1996.
- [2] R.J. Renka, *TSPACK: Tension Spline Curve Fitting Package*, Department of Computer Sciences, University of North TEXAS, 1991.
- [3] R.J. Renka, *Algorithm 716, Tspack: Tension Spline, Curve - Fitting Package*, Department of Computer Sciences, University of North TEXAS, 1993.
- [4] Surendra K. Gupta, *Parametric Splines in Tension* Degree of master of science, Department of Computer Science, Rochester Institute of Technology, Rochester, New York, July 1989.
- [5] Jee Lee, *Splines*, Seoul National University.
- [6] Fabio Pellacini, *parametric spline curves*, 2008.
- [7] UNAM Malla Workgroup, *Editboundary 1.0. Un sistema para el suavizado de regiones poligonales planas*, Facultad de Ciencias, UNAM, México, 2010.
- [8] Luis Carlos Velázquez Guerrero, *Un sistema experimental para generar mallas no estructuradas en regiones planas irregulares* Tesis de maestría, Posgrado en Ciencias Matemáticas-UNAM, México.
- [9] Rivera, Hernández A., *Un punto de vista sobre las cónicas y su uso en el suavizado de poligonos* Tesis de licenciatura, Facultad de Ciencias, UNAM, México.
- [10] Richard L. Burden - J. Douglas Faires, *Análisis Numérico*, International Thomson Editores, 1998.

-
- [11] Romualdo Mariano, *Detección de puntos dominantes de un contorno. Una técnica para la reducción de puntos*, presentado en el seminario del Laboratorio de Computo Científico, Facultad de Ciencias, UNAM.
- [12] J.M. Sanz - Serna, *Diez Lecciones de Cálculo Numérico*, Secretariado de Publicaciones e intercambio científico, Universidad de Valladolid, 1998.
- [13] Félix García Merayo - Antonio Nevot Luna, *Ejercicios resueltos de Cálculo Numérico*, Universidad Pontificia, Madrid, 2009.