



23
2 Gen

Universidad Nacional
Autónoma de México

Facultad de Ciencias

Interpolación
y
Suavizamiento
con
Splines

T e s i s

Que para obtener el título de

ACTUARIO

presenta

Claudia Margarita Durán Sánchez

México D.F.

1994

TESIS CON
FALLA DE ORIGEN





Universidad Nacional
Autónoma de México



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 está 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.

*El premio más grande
que nos da la vida
es la oportunidad
de trabajar con constancia
para lograr lo que deseamos.*

Roosevelt.

*Ante todo le doy gracias a la VIDA,
por permitirme terminar este trabajo.*

*A mis PADRES, que me otorgaron
las bases para iniciarme en el mundo
académico, y por su continuo apoyo
hasta hoy, pude alcanzar
una meta más.*

*Y a DIOS por haberme permitido
encontrar, no tan solo a la persona
indicada, sino al AMIGO, que me
brindó su apoyo en todos los aspectos
que necesité para concluir esté
trabajo y que con especial cariño
le dedico las siguientes palabras :*

GRACIAS

DR. PABLO BARRERA S.

INDICE

INTRODUCCION.	1
Capitulo I. SPLINES LINEALES Y CUADRATICOS.	
I.1. Spline lineal de Interpolación.	3
I.2. Spline cuadrático de Interpolación.	6
I.3. Suavizamiento de un histograma mediante un spline parabólico.	14
Capitulo II. SPLINES CUBICOS DE INTERPOLACION.	
II.1. Spline cúbico Interpolante de Hermite.	23
II.2. Splines cúbicos de Interpolación.	30
II.3. Spline cúbico Paramétrico.	37
II.4. Spline cúbico Periódico.	40
Capítulo III. AJUSTE Y SUAVIZAMIENTO.	
III.1 Propiedad de norma mínima.	47
III.2 Suavizamiento de datos con ruido. Planteamiento del problema.	50
III.3 Planteamiento y solución de un problema equivalente.	53
III.4 Construcción del Spline cúbico de suavizamiento.	59
III.5 Selección de parametros de suavizamiento.	65
III.6 Algoritmo para construir el Spline cúbico de suavizamiento.	71

Capítulo IV. FAMILIAS DE SPLINES.

IV.1. Definición.	78
IV.2. Propiedades de la familia de los Splines.	79
IV.3. Dimensión de Algunas Subfamilias de Splines.	81

Capítulo V. SISTEMA DE INTERPOLACION SPLINES.

V.1. Analisis y Requerimientos Generales.	90
V.2. Diseño a Detalle.	93
V.3. Descripción detallada de los procesos.	96
V.4. Software usado para el sistema SIS	112
V.5. Requerimientos preliminares para la activación del sistema SIS.	113
V.6. Activación y Ejecución del Sistema SIS.	114
V.7. Validación del sistema SIS.	131

APENDICE .

A.1. Solución de sistemas de ecuaciones con matriz tridiagonal.	147
A.2. Solución de sistemas de ecuaciones con matriz tridiagonal cíclica.	158
A.3. Solución de sistemas de ecuaciones con matriz simétrica y definida positiva.	169

REFERENCIAS.	187
--------------	-----

Capítulo IV. FAMILIAS DE SPLINES.

IV.1. Definición.	78
IV.2. Propiedades de la familia de los Splines.	79
IV.3. Dimensión de Algunas Subfamilias de Splines.	81

Capítulo V. SISTEMA DE INTERPOLACION SPLINES.

V.1. Analisis y Requerimientos Generales.	90
V.2. Diseño a Detalle.	93
V.3. Descripción detallada de los procesos.	96
V.4. Software usado para el sistema SIS	112
V.5. Requerimientos preliminares para la activación del sistema SIS.	113
V.6. Activación y Ejecución del Sistema SIS.	114
V.7. Validación del sistema SIS.	131

APENDICE .

A.1. Solución de sistemas de ecuaciones con matriz tridiagonal.	147
A.2. Solución de sistemas de ecuaciones con matriz tridiagonal cíclica.	158
A.3. Solución de sistemas de ecuaciones con matriz simétrica y definida positiva.	169

REFERENCIAS.	187
--------------	-----

I N T R O D U C C I O N

Una de las finalidades de este trabajo fue elaborar un sistema automatizado y amigable, para aquellos lectores que tienen poca experiencia en aproximar un conjunto de datos mediante funciones Spline. Los temas principales que cubre este sistema son :

1. Interpolación Cúbica
2. Suavizamiento de Histogramas mediante Splines Cuadráticos
3. Ajuste y Suavizamiento de Datos mediante Spline Cúbico Natural.

Este trabajo comienza con describir en el Capítulo I la aplicación más sencilla de los splines lineales y cuadráticos; en el Capítulo II se muestra como usar los Splines Cúbicos; en el Capítulo III se muestra que el Spline Cúbico Natural, es el que mejor Ajusta y Suaviza un conjunto de datos; en el Capítulo IV se habla de familias de splines y sus propiedades; más interesantes; en el Capítulo V se describe el desarrollo y manejo del Sistema de interpolación con Splines "S I S" que describimos al comienzo; y por último en el Apéndice se describe los algoritmos utilizados para construir las aproximaciones mencionadas en los capítulos anteriores.

Capítulo I

Splines lineales

y

Cuadráticos

I.1. SPLINE LINEAL DE INTERPOLACION

En este capítulo presentamos varios problemas que se pueden resolver de forma relativamente simple utilizando splines. Con ellos queremos demostrarle al lector lo útiles que resultan en la práctica las funciones splines y a su vez motivarlo para el estudio de técnicas más complicadas que se utilizan para resolver problemas de mayor envergadura. En muchos problemas de ingeniería, economía, biología, medicina, etc., uno cuenta con una tabla de datos,

x_1	x_2	...	x_n
y_1	y_2	...	y_n

que describen cierta relación entre dos magnitudes x y y . Entonces el problema consiste en obtener el valor de la magnitud y para un x que no aparece en la tabla. Por ejemplo, los datos siguientes representan los valores de la masa de vapor de agua contenida en el aire saturado para diferentes temperaturas del aire

temperatura del aire ($^{\circ}$ C)	-24.4	-17.8	-10.0	-1.1	8.9	26.7
masa de vapor de agua	.584	1.229	2.231	4.488	8.755	25.31

y uno podría estar interesado en obtener el valor de masa cuando la temperatura del aire es -55 ($^{\circ}$ C). La solución más sencilla para este tipo de problema consiste en unir los datos mediante segmentos lineales, obteniéndose de este modo una función lineal por tramos que podemos evaluar en cualquier punto intermedio y que interpola los datos. Los puntos en que se unen los segmentos lineales se llaman puntos de ruptura y la función lineal por tramos y continua se conoce como spline lineal.

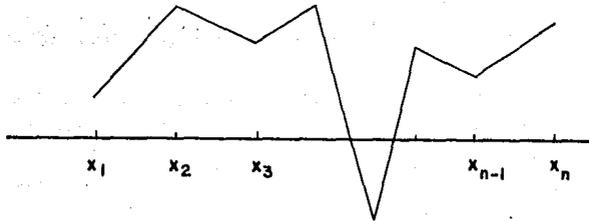


fig. 1.1 Spline lineal de interpolación.

Supongamos ahora que los y_i son los valores en los puntos x_i de cierta función g desconocida y denotemos por f la función lineal por tramos que interpola a g en dichos puntos, o sea

$$f(x) = p_i(x) \quad \text{para} \quad x_i \leq x \leq x_{i+1}$$

donde p_i es un polinomio de grado ≤ 1 y

$$f(x_i) = g(x_i), \quad i = 1, \dots, n.$$

Entonces para $x_i \leq x \leq x_{i+1}$ f es la recta que pasa por los puntos $(x_i, g(x_i))$ y $(x_{i+1}, g(x_{i+1}))$ es decir

$$f(x) = g(x_i) + \frac{g(x_{i+1}) - g(x_i)}{(x_{i+1} - x_i)}(x - x_i), \quad i = 1, \dots, n-1. \quad (1.1.1)$$

Ahora si queremos evaluar la función f en un punto x^* , basta con hallar $1 \leq i \leq n-1$ tal que $x_i \leq x^* \leq x_{i+1}$ y calcular $f(x^*)$ utilizando la expresión (1.1.1). Regresando a nuestro ejemplo, podemos decir que la masa de vapor de agua contenida en el aire a una temperatura de -55 ($^{\circ}$ C) es aproximadamente:

$$f(-5.5) = g(-10.0) + \frac{g(-1.1) - g(-10.0)}{-1.1 + 10.0}(-5.5 + 10.0) = 3.3595.$$

Aunque el spline lineal nos proporciona una solución al problema de interpolar un conjunto de datos, en la práctica tiene la desventaja de que su gráfico posee muchos "picos," mientras que frecuentemente en las aplicaciones se necesita una curva "suave". Por eso en las secciones siguientes vamos a estudiar las funciones formadas por polinomios cuadráticos y cúbicos por tramos.



I.2. SPLINE CUADRATICO DE INTERPOLACION

En la sección 1.1. construimos un spline lineal de interpolación, con la propiedad de que los segmentos de recta se unen en los puntos de interpolación. Sin embargo, esto no tiene que ser necesariamente así, es decir los puntos de ruptura de la función polinomial por tramos no tiene que coincidir con los puntos de interpolación. Incluso, en algunas ocasiones, resulta más ventajoso interpolar en otros puntos que no sean los de ruptura, como veremos a continuación :

Supongamos nuevamente que se conocen los valores de cierta función en los puntos $a = x_1 < x_2 \dots < x_n = b$ y nos interesa construir una función f formada por polinomios de grado ≤ 2 , que interpole a g en los puntos x_i , es decir tal que

$$f(x) = p_i(x) \text{ para } \xi_i \leq x \leq \xi_{i+1}, \quad i = 1, \dots, n$$

donde ξ_1, \dots, ξ_{n+1} son los puntos de ruptura de f seleccionados de modo que

$$\xi_1 \leq a = x_1 < \xi_2 < x_2 < \xi_3 < \dots < x_{n-1} < \xi_n < x_n = b \leq \xi_{n+1}$$

y p_i es un polinomio de grado ≤ 2 para cada $i = 1 \dots n$.

Por ejemplo, pudiéramos considerar que cada punto de ruptura interior es el punto medio entre dos de interpolación, es decir:

$$\xi_i = x_{i-1/2} = (x_i + x_{i-1})/2, \quad i = 2, \dots, n$$

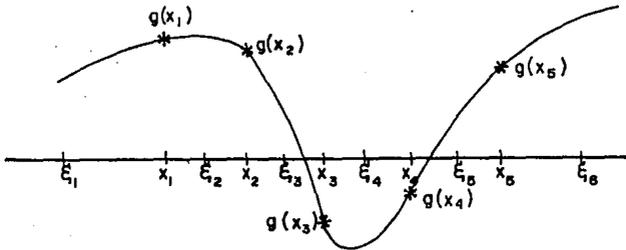


fig. 1.2 Spline parabólico. Los puntos de ruptura están situados entre los de interpolación.

Como f debe interpolar a g en los puntos x_i , entonces cada sección polinomial tiene que cumplir la condición :

$$p_i(x_i) = g(x_i), \quad i = 1, \dots, n \quad (1.2.1)$$

y a su vez como el polinomio $p_i(x)$ tiene 3 coeficientes, la condición (1.2.1) nos deja todavía dos de ellos libres. Por eso vamos a exigir además que

$$p_i(\xi_i) = v_i, \quad p_i(\xi_{i+1}) = v_{i+1}, \quad i = 1, \dots, n, \quad (1.2.2)$$

suponiendo por supuesto que $\xi_i < x_i$ y $\xi_{n+1} > x_n$. Nótese que las condiciones (1.2.1) y (1.2.2) nos garantizan que independientemente de como seleccionemos los v_i , f es una función continua que coincide con g en los puntos x_i , $i = 1, \dots, n$. Por otro lado la parábola que pasa por los puntos (ξ_i, v_i) , $(x_i, g(x_i))$ y (ξ_{i+1}, v_{i+1}) está dada por

$$p_i(x) = v_i + \alpha_i(g(x_i) - v_i)(x - \xi_i) + \left[\frac{\beta_i(v_{i+1} - g(x_i)) - \alpha_i(g(x_i) - v_i)}{\Delta\xi_i} \right] (x - \xi_i)(x - x_i) \quad (1.2.3)$$

donde

$$\alpha_i = \frac{1}{x_i - \xi_i}, \quad \beta_i = \frac{1}{\xi_{i+1} - x_i} \quad \text{y} \quad \Delta\xi_i = \xi_{i+1} - \xi_i. \quad (1.2.4)$$

Los v_i se pueden determinar entonces exigiendo que f , además de ser continua, tenga primera derivada continua. Cuando los v_i se escogen de esta manera, se dice que f es un spline parabólico (ver fig. 1.2). Por lo tanto, si queremos que f sea un spline parabólico tenemos que exigir que

$$p'_{i-1}(\xi_i) = p'_i(\xi_i), \quad i = 2, \dots, n. \quad (1.2.5)$$

Derivando en (1.2.3) obtenemos:

$$p'_i(x) = \alpha_i(g(x_i) - v_i) + \left[\frac{\beta_i(v_{i+1} - g(x_i)) - \alpha_i(g(x_i) - v_i)}{\Delta\xi_i} \right] \left[(x - \xi_i) + (x - x_i) \right],$$

luego

$$\begin{aligned} p'_i(\xi_i) &= \alpha_i(g(x_i) - v_i) + \left[\frac{\beta_i(v_{i+1} - g(x_i)) - \alpha_i(g(x_i) - v_i)}{\Delta\xi_i} \right] \left(\frac{-1}{\alpha_i} \right) \\ &= v_i \left[-\alpha_i - \frac{1}{\Delta\xi_i} \right] - \frac{\beta_i}{\alpha_i \Delta x_i} v_{i+1} + g(x_i) \left[\alpha_i + \frac{\beta_i}{\alpha_i \Delta\xi_i} + \frac{1}{\Delta\xi_i} \right], \end{aligned}$$

de manera similar se obtiene que

$$\begin{aligned} p'_{i-1}(\xi_i) &= v_{i-1} \left[-\alpha_{i-1} + \frac{\alpha_{i-1}}{\Delta\xi_{i-1}} (\Delta\xi_{i-1} + \frac{1}{\beta_{i-1}}) \right] + \\ &v_i \left[\frac{\beta_{i-1}}{\Delta\xi_{i-1}} \left(\Delta\xi_{i-1} + \frac{1}{\beta_{i-1}} \right) \right] + g(x_{i-1}) \left[\alpha_{i-1} - \left(\frac{\beta_{i-1} + \alpha_{i-1}}{\Delta\xi_{i-1}} \right) \right. \\ &\left. \left(\Delta\xi_{i-1} + \frac{1}{\beta_{i-1}} \right) \right], \end{aligned}$$

Por lo tanto, agrupando los términos semejantes, podemos expresar la condición (1.2.5) de la siguiente forma:

$$\begin{aligned} v_{i-1} \left[\frac{\alpha_{i-1}}{\beta_{i-1} \Delta\xi_{i-1}} \right] + v_i \left[\beta_{i-1} + \frac{1}{\Delta\xi_{i-1}} + \alpha_i + \frac{1}{\Delta\xi_i} \right] + v_{i+1} \left[\frac{\beta_i}{\alpha_i \Delta\xi_i} \right] \\ = -g(x_{i-1}) \left[-\beta_{i-1} - \frac{\beta_{i-1} + \alpha_{i-1}}{\Delta\xi_{i-1} \beta_{i-1}} \right] + g(x_i) \left[\alpha_i + \frac{\beta_i}{\alpha_i \Delta\xi_i} + \frac{1}{\Delta\xi_i} \right]. \end{aligned}$$

Si ahora utilizamos que :

$$\frac{\alpha_{i-1}}{\beta_{i-1} \Delta\xi_{i-1}} = \alpha_{i-1} - \frac{1}{\Delta\xi_{i-1}}, \quad \frac{\beta_i}{\alpha_i \Delta\xi_i} = \beta_i - \frac{1}{\Delta\xi_i},$$

llegamos finalmente al siguiente sistema de ecuaciones lineales

$$\begin{aligned} \left(\alpha_{i-1} - \frac{1}{\Delta\xi_{i-1}} \right) v_{i-1} + \left(\beta_{i-1} + \frac{1}{\Delta\xi_{i-1}} + \alpha_i + \frac{1}{\Delta\xi_i} \right) v_i + \left(\beta_i - \frac{1}{\Delta\xi_i} \right) v_{i+1} \\ = g(x_{i-1})(\alpha_{i-1} + \beta_{i-1}) + g(x_i)(\alpha_i + \beta_i), \quad i = 2, \dots, n. \end{aligned} \tag{1.2.6}$$

El sistema (1.2.6) consta de $n - 1$ ecuaciones en v_1, v_2, \dots, v_{n+1} , por lo tanto hace falta las condiciones adicionales para determinar unívocamente una solución. A continuación indicamos algunas de las formas más usuales.

- i) Lo más fácil, es si se asignan de antemano los valores v_1 y v_{n+1} en este caso obtenemos un sistema de $n - 1$ ecuaciones en v_2, v_3, \dots, v_n que queda de la forma:

$$\begin{pmatrix} d_1 & c_1 & 0 & \dots & \dots \\ a_2 & d_2 & c_2 & 0 & \dots \\ 0 & a_3 & d_3 & c_3 & \dots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \dots & \dots & a_{n-2} & d_{n-2} & c_{n-2} \\ \dots & \dots & \dots & a_{n-1} & d_{n-1} \end{pmatrix} = \begin{pmatrix} v_2 \\ v_3 \\ v_4 \\ \vdots \\ v_{n-1} \\ v_n \end{pmatrix} = \begin{pmatrix} \tilde{b}_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ \tilde{b}_n \end{pmatrix}$$

donde :

$$\begin{aligned} a_{i-1} &= \alpha_{i-1} - \frac{1}{\Delta\xi_{i-1}}; & c_{i-1} &= \beta_i - \frac{1}{\Delta\xi_i}; \\ d_{i-1} &= \beta_{i-1} + \frac{1}{\Delta\xi_{i-1}} + \alpha_i + \frac{1}{\Delta\xi_i}; \\ b_{i-1} &= g(x_{i-1})(\alpha_{i-1} + \beta_{i-1}) + g(x_i)(\alpha_i + \beta_i); \\ \tilde{b}_1 &= b_1 - a_1 v_1; & \tilde{b}_{n-1} &= b_{n-1} - c_{n-1} v_{n+1} \end{aligned}$$

- ii) Otra posibilidad es cuando conocemos los valores g'_1, g'_n de las derivadas en los puntos x_1 y x_n de la función $g(x)$. En este caso es natural pedir que el spline tenga los mismos valores, por lo que podemos imponer las condiciones

$$\begin{aligned} p'_1(x_1) &= g'_1, \\ p'_n(x_n) &= g'_n, \end{aligned}$$

que haciendo un poco de manipulación se transforma en :

$$\begin{aligned} \left(\frac{1}{\Delta\xi_1} - \alpha_1\right)v_1 + \left(\beta_1 - \frac{1}{\Delta\xi_1}\right)v_2 &= g'_1 + (\beta_1 - \alpha_1)g_1, \\ \left(\frac{1}{\Delta\xi_n} - \alpha_n\right)v_n + \left(\beta_n - \frac{1}{\Delta\xi_n}\right)v_{n+1} &= g'_n + (\beta_n - \alpha_n)g_n, \end{aligned}$$

Por lo que el sistema de $n + 1$ ecuaciones en $n + 1$ variables queda :

$$\begin{pmatrix} -a_0 & c_0 & \dots & \dots & \dots & \dots \\ a_1 & d_1 & c_1 & \dots & \dots & \dots \\ \dots & a_2 & d_2 & c_2 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \dots & \dots & \dots & a_{n-1} & d_{n-1} & c_{n-1} \\ \dots & \dots & \dots & \dots & -a_n & c_n \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_n \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \\ b_{n+1} \end{pmatrix}$$

donde:

$$\begin{aligned} b_1 &= g'_1 + (\beta_1 - \alpha_1)g_1, \\ b_{n+1} &= g'_n + (\beta_n - \alpha_n)g_n. \end{aligned}$$

- iii) Otra posibilidad es que se calculan g'_1 , g'_n aproximadamente por medio de interpolación polinomial, es decir se puede hacer pasar un polinomio de grado 2 por (x_1, g_1) , (x_2, g_2) , (x_3, g_3) y calcular su derivada en x_1 y usar este valor como g'_1 . Si lo hacemos obtenemos :

$$g'_1 = g[x_1, x_2] - g[x_2, x_3] + g[x_1, x_3]$$

donde

$$g[x, y] = \frac{f(x) - f(y)}{x - y}.$$

Procediendo en forma análoga con x_n obtenemos :

$$g'_n = g[x_{n-2}, x_n] - g[x_{n-2}, x_{n-1}] + g[x_{n-1}, x_n]$$

así que en el caso de que no conozcamos los valores de g'_1 y g'_n podemos usar los anteriores y resolver el sistema de ecuaciones anterior.

- iv) Otra posibilidad es pedir que el spline sea lineal en el primer y último intervalo, ésto nos da las condiciones:

$$p'' \equiv 0; \quad p_n'' \equiv 0;$$

de donde obtenemos :

$$\begin{aligned} \alpha_1 v_1 + \beta_1 v_2 &= (\alpha_1 + \beta_1)g_1, \\ \alpha_n v_n + \beta_n v_{n+1} &= (\alpha_n + \beta_n)g_n, \end{aligned}$$

el sistema de $n + 1$ ecuaciones en $n + 1$ variables nos queda de la forma :

$$\begin{pmatrix} \tilde{d}_0 & \tilde{c}_0 & \dots & \dots & \dots & \dots \\ a_1 & d_1 & c_1 & \dots & \dots & \dots \\ \dots & a_2 & d_2 & c_2 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \dots & \dots & \dots & a_{n-1} & d_{n-1} & c_{n-1} \\ \dots & \dots & \dots & \dots & \tilde{a}_n & \tilde{d}_n \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_n \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} \tilde{b}_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ \tilde{b}_n \end{pmatrix}$$

de donde

$$\begin{aligned} \tilde{d}_0 &= \alpha_1; & \tilde{c}_0 &= \beta_1; & \tilde{b}_0 &= (v_1 + \beta_1)g_1; \\ \tilde{a}_n &= \alpha_n; & \tilde{d}_n &= \beta_n; & \tilde{b}_n &= (\alpha_n + \beta_n)g_n. \end{aligned}$$

Las matrices de los sistemas son tridiagonales y en los casos i) y iv) son diagonalmente dominantes por columnas, lo que trae la agradable consecuencia de que el sistema se puede resolver fácilmente de manera confiable usando eliminación gaussiana sin pivoteo. Veamos un ejemplo de spline parabólico. Supongamos que se tiene la siguiente tabla de datos

x	2	4	8	9	12
g(x)	2	-1	-2	0	1

tabla 1.1 Datos de una función g desconocida.

y queremos construir el spline parabólico que interpola a g . Vamos a tomar cada punto de ruptura interior como el punto medio entre dos de interpolación, o sea

$$\xi_2 = 3, \quad \xi_3 = 6, \quad \xi_4 = 8.5, \quad \xi_5 = 10.5.$$

Supongamos además que $\xi_1 = 1$ y $\xi_6 = 13$. Entonces para construir la matriz del sistema (1.2.6) necesitamos calcular los valores α_i , β_i y $1/\Delta\xi_i$, $i = 1, \dots, 5$. De acuerdo con (1.2.4) tenemos :

$$\alpha_1 = \frac{1}{x_1 - \xi_1} = 1, \quad \alpha_2 = \frac{1}{x_2 - \xi_2} = 1, \quad \alpha_3 = \frac{1}{x_3 - \xi_3} = 0.5,$$

$$\alpha_4 = \frac{1}{x_4 - \xi_4} = 2, \quad \alpha_5 = \frac{1}{x_5 - \xi_5} = 0.66.$$

De manera similar se obtiene :

$$\beta_1 = 1, \quad \beta_2 = 0.5, \quad \beta_3 = 2, \quad \beta_4 = 0.66, \quad \beta_5 = 1,$$

mientras que

$$\frac{1}{\Delta\xi_1} = 0.5, \quad \frac{1}{\Delta\xi_2} = 0.33, \quad \frac{1}{\Delta\xi_3} = 0.4, \quad \frac{1}{\Delta\xi_4} = 0.5, \quad \frac{1}{\Delta\xi_5} = 0.4.$$

Por lo tanto, el sistema (1.2.6) se escribe en este caso de la siguiente forma :

$$\begin{array}{rcccccccl} 0.5v_1 & +2.83v_2 & +0.17v_3 & & & & & = 2.5 \\ & .67v_2 & +1.73v_3 & +1.6v_4 & & & & = -6.5 \\ & & 0.1v_3 & +4.9v_4 & +0.16v_5 & & & = -5 \\ & & & 1.5v_4 & +2.22v_5 & +0.6v_6 & & = 1.66 \end{array} \quad (1.2.7)$$

Para resolverlo debemos darle valores a las variables libres v_1 y v_6 . Como v_1 y v_6 son los valores del spline en ξ_1 y ξ_6 respectivamente para que el gráfico de spline quede "bonito", podemos tomar $v_1 = 1$ y $v_6 = -1$. En tal caso el sistema (1.2.7) se transforma en el siguiente sistema (expresado matricialmente).

$$\begin{pmatrix} 2.83 & 0.17 & & & 0 \\ 0.67 & 1.73 & 1.6 & & \\ & 0.1 & 4.9 & 0.16 & \\ 0 & & 1.5 & 2.22 & \end{pmatrix} \begin{pmatrix} v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix} = \begin{pmatrix} 2 \\ -6.5 \\ -5 \\ 1.06 \end{pmatrix}. \quad (1.2.8)$$

Nótese que la matriz de (1.2.8) es tridiagonal y de diagonal dominante por columnas. Utilizando el programa TRIDSIS que se ofrece en el Apéndice, se puede verificar fácilmente que la solución de este sistema (aproximando a dos cifras decimales) es

$$v_2 = 0.9; \quad v_3 = -3.19; \quad v_4 = -1; \quad v_5 = 1.15.$$

Luego de (1.2.3) se puede concluir que los polinomios que constituyen el spline parabólico de interpolación son :

$$\begin{aligned}
 p_1(x) &= -1.05x^2 + 4.15x - 2.1, \\
 p_2(x) &= 0.27x^2 - 3.78x + 9.82, \\
 p_3(x) &= 0.56x^2 - 7.27x + 20.2, \\
 p_4(x) &= -0.62x^2 + 12.86x - 65.47, \\
 p_5(x) &= -0.76x^2 + 17.01x - 93.62,
 \end{aligned}$$

donde los coeficientes de cada polinomio también se han aproximado a dos cifras decimales. La figura 1.3 muestra el gráfico del spline de este ejemplo.

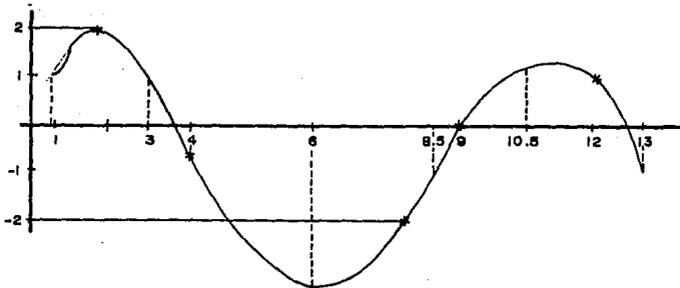


fig 1.3 Spline parabólico de interpolación
x - puntos de interpolación.

Hoja adicional para la gráfica de la figura 1.3. Valores del Spline parabólico

f.

x	f(x)	x	f(x)	x	f(x)
1	1	6.5	-3.31	10.5	1.15
1.5	1.76	7	-3.15	11	1.48
2	2	7.5	-2.72	11.5	1.43
2.5	1.71	8	-2	12	1
3	0.9	8.5	-1	12.5	0.2
4	-1	9	0	13	-1
5	-2.36	9.5	0.7		
6	-3.19	10	1.07		

I.3. SUAVIZAMIENTO DE UN HISTOGRAMA MEDIANTE UN SPLINE PARABOLICO

Los splines parabólicos no sólo son útiles para aproximar un conjunto de datos mediante el proceso de interpolación. Resulta muy frecuente que para estudiar datos de diferentes tipos se construya un histograma o diagrama de barras. Por ejemplo, en muchos estudios estadísticos se tienen diagramas donde la altura de las barras indica el crecimiento o decrecimiento de cierta magnitud por años. Sin embargo, en ocasiones resulta más ventajoso disponer de una curva suave, que en cierto sentido represente la información contenida en el histograma. Tal curva se puede obtener mediante un spline parabólico, que recibe el nombre de histospline.

Supongamos entonces que $x_1 < x_2 < \dots < x_{n+1}$ y que h_i es la altura de la barra correspondiente al intervalo (x_i, x_{i+1}) .

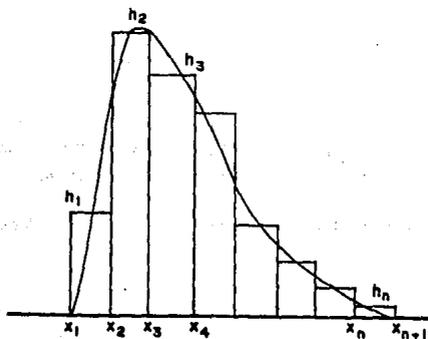


fig. 1.4 Histospline parabólico.

Como el histograma representa la distribución de cierta magnitud, usualmente se supone que el área de la i -ésima barra $h_i \Delta x_i$, es aproximadamente igual a la integral de la función de distribución sobre el intervalo (x_i, x_{i+1}) . Por eso, vamos a exigir que el spline parabólico f que suaviza el histograma, satisfaga las condiciones:

$$\int_{x_i}^{x_{i+1}} f(x) dx = h_i \Delta x_i, \quad i = 1, \dots, n, \quad (1.3.1)$$

y además vamos a tomar como puntos de ruptura del spline, los propios puntos x_i , que definen los extremos de cada barra.

Como f está formada por n secciones de polinomios cuadráticos, necesitamos calcular $3n$ coeficientes, mientras que por otro lado debemos exigir que f y f' sean continuas en los puntos x_i , $i = 2, \dots, n$, lo que representa $2(n-1)$ condiciones, que unidas a las n dadas por (1.3.1) dan un total de $3n-2$ condiciones, para calcular los $3n$ coeficientes de f . Las dos condiciones restantes, se obtienen exigiendo que en los extremos se cumpla una de las opciones siguientes:

$$\begin{aligned} \text{i) } & f(x_1) = 0, & f(x_{n+1}) = 0, \\ \text{ii) } & f'(x_1) = 0, & f'(x_{n+1}) = 0. \end{aligned} \tag{1.3.2}$$

Para calcular el histospline f se pueden utilizar diferentes métodos. Por ejemplo, cada sección $p_i(x)$ de f se puede escribir mediante la fórmula de Taylor, como:

$$p_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2;$$

donde

$$a_i = f(x_i), \quad b_i = f'(x_i); \quad \text{y} \quad c_i = \frac{f''(x_i)}{2}.$$

Entonces las condiciones de continuidad de f y f' , así como las condiciones (1.3.1) y (1.3.2), definen un sistema de $3n$ ecuaciones lineales para el cálculo de los coeficientes a_i , b_i y c_i , $i = 1, \dots, n$. Sin embargo, aunque la matriz de este sistema tiene una estructura casi diagonal por bloques, de la misma no se puede concluir si el sistema tiene o no solución única.

Por supuesto que la estructura de la matriz del sistema de ecuaciones lineales, que nos permite calcular los coeficientes del histospline, depende de la representación que hayamos utilizado para cada polinomio $p_i(x)$, de modo que el problema fundamental consiste en escoger una base $r_1(x)$, $r_2(x)$, $r_3(x)$, del espacio de polinomios de grado menor o igual a 2, tal que la representación de $p_i(x)$ en términos de esa base:

$$p_i(x) = a_i r_1(x) + b_i r_2(x) + c_i r_3(x), \quad x_i \leq x \leq x_{i+1};$$

de lugar a un sistema lineal sencillo para el cálculo de los coeficientes a_i , b_i , y c_i . Para simplificar los cálculos vamos a hacer el cambio de variable

$$y = \frac{(x - x_i)}{\Delta x_i}, \quad \text{donde} \quad \Delta x_i = x_{i+1} - x_i, \quad i = 1, \dots, n.$$

Entonces para $x_i \leq x \leq x_{i+1}$ el histospline f se escribe de la siguiente forma

$$f(x) = p_i(y) = a_i r_1(y) + b_i r_2(y) + c_i r_3(y), \quad 0 \leq y \leq 1 \quad (1.3.3)$$

y el problema se reduce a buscar tres polinomios $r_1(y)$, $r_2(y)$ y $r_3(y)$ linealmente independientes en $[0, 1]$. Consideremos los polinomios

$$r_1(y) = (1 - y)(1 - 3y);$$

$$r_2(y) = y(3y - 2);$$

$$r_3(y) = 6y(1 - y).$$

Incluso gráficamente (ver figura 1.5) es fácil verificar que estos polinomios son linealmente independientes en $[0, 1]$.

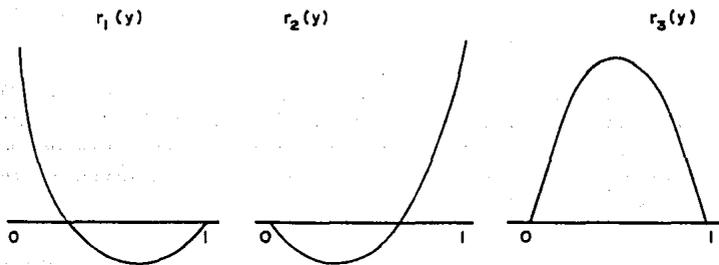


fig. 1.5 Polinomios independientes en $[0, 1]$.

Nótese además que r_1 , r_2 y r_3 cumplen las siguientes condiciones:

$$\begin{aligned}
 r_1(0) &= 1, & r_1(1) &= 0, & \int_0^1 r_1(y)dy &= 0, \\
 r_2(0) &= 0, & r_2(1) &= 1, & \int_0^1 r_2(y)dy &= 0, \\
 r_3(0) &= 0, & r_3(1) &= 0, & \int_0^1 r_3(y)dy &= 1.
 \end{aligned}
 \tag{1.3.4}$$

Por lo tanto, de (1.3.1) y (1.3.3) tenemos que:

$$\begin{aligned}
 f(x_i) &= p_i(0) = a_i, \\
 f(x_{i+1}) &= p_i(1) = b_i, \\
 \int_{x_i}^{x_{i+1}} f(x)dx &= \int_0^1 p_i(y)\Delta x_i dy = c_i \Delta x_i = h_i \Delta x_i, \quad \text{e.d. } c_i = h_i.
 \end{aligned}$$

Sustituyendo entonces en (1.3.3), podemos expresar el histospline f de la siguiente forma, para $x_i \leq x \leq x_{i+1}$, $i = 1, \dots, n$

$$f(x) = p_i(y) = f(x_i)r_1(y) + f(x_{i+1})r_2(y) + h_i r_3(y). \tag{1.3.5}$$

La expresión (1.3.5) nos demuestra además, que independientemente del valor de f en los puntos de ruptura x_i y las alturas h_i de cada barra, la función $f(x)$ es continua. En efecto,

$$\begin{aligned}
 f(x_i^-) &= p_{i-1}(1) = f(x_i), \\
 f(x_i^+) &= p_i(0) = f(x_i).
 \end{aligned}$$

Veamos entonces cómo hay que escoger los valores de f en los puntos x_i , para garantizar la continuidad de f' . Derivando en (1.3.5) obtenemos que para $x_i \leq x \leq x_{i+1}$:

$$\begin{aligned}
 f'(x) &= \frac{p_i'(y)}{\Delta x_i} = \frac{[f(x_i)(6y-4) + f(x_{i+1})(6y-2) + h_i(6-12y)]}{\Delta x_i} \\
 &= \frac{\left[f(x_i) \left(\frac{6(x-x_i)}{\Delta x_i} - 4 \right) + f(x_{i+1}) \left(\frac{6(x-x_i)}{\Delta x_i} - 2 \right) + h_i \left(6 - \frac{12(x-x_i)}{\Delta x_i} \right) \right]}{\Delta x_i}
 \end{aligned}
 \tag{1.3.6}$$

Por lo tanto, la condición de continuidad de f'

$$p'_{i-1}(x_i) = p'_i(x_i), \quad i = 2, \dots, n.$$

donde el lado derecho se construyó mediante (1.3.8). Este sistema se resolvió empleando el programa TRIDSIS que presentamos en el Apéndice, obteniéndose los siguientes resultados (redondeados a una cifra decimal) :

$$\begin{aligned} f(x_1) &= 0 \\ f(x_2) &= 15302.5 \\ f(x_3) &= 18898.9 \\ f(x_4) &= 10039.6 \\ f(x_5) &= 3738.4 \\ f(x_6) &= 1016.7 \\ f(x_7) &= 96.8 \\ f(x_8) &= 0. \end{aligned}$$

Para calcular los polinomios que constituyen el histospline debemos primero expresar los polinomios $r_1(y)$, $r_2(y)$, y $r_3(y)$ en términos de la variable x . Teniendo en cuenta que en nuestro caso $\Delta x_i = 5$ para todo i , obtenemos entonces que para $x_i \leq x \leq x_{i+1}$

$$\begin{aligned} r_1(x) &= \left(1 - \frac{(x - x_i)}{5}\right) \left(1 - \frac{3(x - x_i)}{5}\right), \\ r_2(x) &= \frac{(x - x_i)}{5} \left(\frac{3(x - x_i)}{5} - 2\right), \\ r_3(x) &= \frac{6(x - x_i)}{5} \left(1 - \frac{(x - x_i)}{5}\right). \end{aligned}$$

Por lo tanto para $x_i \leq x \leq x_{i+1}$, $i = 1, \dots, 7$ el histospline f está dada por

$$f(x) = f(x_i)r_1(x) + f(x_{i+1})r_2(x) + h_i r_3(x). \quad (1.3.12)$$

Sustituyendo en (1.3.12) los valores de $f(x_i)$ calculados y los x_i , $i = 1, \dots, 8$.

Obtenemos finalmente que los polinomios que constituyen el histospline f son :

$$\begin{aligned}
 p_1(x) &= 50.22x^2 + 1302.8x - 30841.5, \\
 p_2(x) &= -518.47x^2 + 24050.52x - 258319.1, \\
 p_3(x) &= 20.22x^2 - 2883.96x + 78360.4, \\
 p_4(x) &= 82.08x^2 - 6595.44x + 134030.8, \\
 p_5(x) &= 61.092x^2 - 5126.24x + 108319.1, \\
 p_6(x) &= 25.38x^2 - 2341.28x + 54059.9, \\
 p_7(x) &= 7.536x^2 - 735.28x + 17924.
 \end{aligned}$$

En la figura 1.6 se muestra además el gráfico del histospline f .

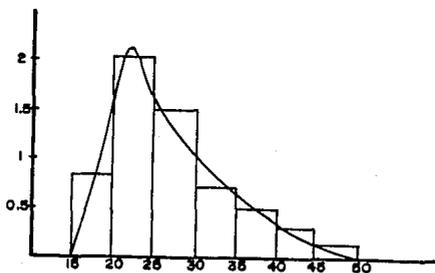


fig 1.6 Maternidad de las madres búlgaras.
Histospline que lo suaviza.

Capítulo II

Splines Cúbicos

de

Interpolación

II. SPLINES CUBICOS DE INTERPOLACION

Quizás entre todas las funciones polinomiales por tramos, las más populares debido a sus propiedades son los polinomios cúbicos por tramos y en especial los splines cúbicos. En esta sección, vamos a estudiar diferentes tipos de funciones cúbicas por tramos, que nos permitirán resolver otra vez, el problema de construir una curva que pase por un conjunto de puntos.

II.1. SPLINE CUBICO DE HERMITE

Supongamos que tenemos la siguiente tabla de valores :

x	x_1	x_2	x_3	x_4	\dots	x_n
g	g_1	g_2	g_3	g_4	\dots	g_n
g'	s_1	s_2	s_3	s_4	\dots	s_n

es decir que tenemos los valores de una función y sus derivadas en los puntos $\{x_i\}$, $i = 1, \dots, n$. Vamos a ver a continuación que es muy fácil construir un polinomio cúbico por tramos que coincida con g en sus valores y en sus derivadas en los mismos puntos.

Antes de resolver el problema anterior, vamos a resolver un caso especial, a partir del cual la expresión de la solución del problema general sera inmediata.

Problema.

Determinar un polinomio cúbico $p(y)$ tal que

$$p(0) = p_0, \quad p(1) = p_1, \quad p'(0) = p'_0, \quad p'(1) = p'_1.$$

Solución.

Vamos a expresar el polinomio $p(y)$ en la forma

$$p(y) = p_0 r_1(y) + p_1 r_2(y) + p_0' r_3(y) + p_1' r_4(y) \quad (2.1.1)$$

donde

$$r_i(y); \quad i = 1, \dots, 4$$

son polinomios cúbicos. Observemos que si

$$p_1 = p_0' = p_1' = 0 \quad \text{y} \quad p_0 = 1,$$

entonces :

$$\begin{aligned} p(y) &= r_1(y), \quad \text{es decir } r(y) \text{ debe satisfacer :} \\ r_1(0) &= 1; & r_1'(0) &= 0; \\ r_1(1) &= 0; & r_1'(1) &= 0; \end{aligned}$$

de estas condiciones es muy fácil obtener $r_1(y)$ puesto que, como $y = 1$ es una raíz doble tenemos que, debe ser de la forma

$$r_1(y) = (a + by)(y - 1)^2$$

donde a, b son coeficientes que quedan determinados por las condiciones $r_1(0) = 1$ y $r_1'(0) = 0$, obtenemos

$$\begin{aligned} a &= 1; \quad b = 2, \\ r_1(y) &= (1 + 2y)(y - 1)^2. \end{aligned}$$

Procediendo de manera similar obtenemos :

$$\begin{aligned} r_2(y) &= y^2(3 - 2y), \\ r_3(y) &= (y - 1)^2 y, \\ r_4(y) &= y^2(y - 1), \end{aligned}$$

así que entonces el polinomio $p(y)$ queda como

$$p(y) = p_0(1 + 2y)(y - 1)^2 + p_1 y^2(3 - 2y) + p_0'(y - 1)^2 y + p_1' y^2(y - 1); \quad (2.1.2)$$

veamos ahora que es muy fácil obtener el polinomio $p_i(x)$ tal que

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

consideremos el cambio de variable

$$y = \frac{x - x_i}{\Delta x_i}; \quad \Delta x_i = x_{i+1} - x_i$$

que nos transforma el intervalo $[x_i, x_{i+1}]$ en el intervalo $[0,1]$, la idea es que podemos obtener la expresión para $p_i(x)$ a partir de $p(y)$, sustituyendo y , pero ésto es posible si los términos que contienen p'_0 y p'_1 los multiplicamos por Δx_i ya que :

$$p'_i(x) = \frac{d}{dx} p(y(x)) = p'(y) y' = p'(y) \frac{1}{\Delta x_i};$$

$$\begin{aligned} p'_i(x_i) = p'(0) \frac{1}{\Delta x_i} = s_i &\implies p'_0 = p'(0) = s_i \Delta x_i; \\ p'_i(x_{i+1}) = p'(1) \frac{1}{\Delta x_i} = s_{i+1} &\implies p'_1 = p'(1) = s_{i+1} \Delta x_i; \\ p_i(x_i) = p(0) = g_i &\implies p_0 = g_i; \\ p_i(x_{i+1}) = p(1) = g_{i+1} &\implies p_1 = g_{i+1}; \end{aligned}$$

por consiguiente obtenemos :

$$\begin{aligned} p_i(x) = &g_i r_1(y) + g_{i+1} r_2(y) \\ &+ s_i \Delta x_i r_3(y) + s_{i+1} \Delta x_i r_4(y); \quad y = \frac{x - x_i}{\Delta x_i}, \end{aligned}$$

sustituyendo y obtenemos:

$$\begin{aligned} p_i(x) = &g_i \frac{(x - x_{i+1})^2 [2(x - x_i) + \Delta x_i]}{\Delta x_i^3} + g_{i+1} \frac{(x - x_i)^2 [2(x_{i+1} - x) + \Delta x_i]}{\Delta x_i^3} \\ &+ s_i \frac{(x_{i+1} - x)^2 (x - x_i)}{\Delta x_i^2} - s_{i+1} \frac{(x - x_i)^2 (x_{i+1} - x)}{\Delta x_i^2} \\ &x_i \leq x \leq x_{i+1}; \quad i = 1 \dots n - 1; \quad (2.1.3) \end{aligned}$$

así hemos obtenido finalmente una familia de polinomios $\{p_i(x)\}$ cada uno definido en el intervalo $[x_i, x_{i+1}]$ por lo que la función definida por :

$$f(x) = p_i(x), \quad \text{para } x_i \leq x \leq x_{i+1}, \quad i = 1, \dots, n - 1$$

es una función continuamente diferenciable, polinomial por tramos. Esta función es conocida por el nombre de función interpolante cúbica de Hermite. Se puede demostrar que f aproxima muy bien a la función g y mejor a medida que las longitudes de los intervalos $[x_i, x_{i+1}]$ sea más pequeña.

Ejemplos.

1) Supongamos que $g(x) = \frac{1}{(1+x^2)}$ y queremos calcular el interpolante cúbico de Hermite que coincide con g en los puntos $-3, -1, 0, 1$ y 3 . Como

$$g'(x) = \frac{-2x}{(1+x^2)^2},$$

entonces los datos que necesitamos para construir el interpolante f son los siguientes:

x	-3	-1	0	1	3
g	0.1	0.5	1	0.5	0.1
g'	0.06	0.5	0	-0.5	-0.6

tabla 2.1 Valores de la función $g = \frac{1}{1+x^2}$ y de su primera derivada.

Teniendo en cuenta que $s_i = g'(x_i)$, $i = 1, \dots, 5$ y utilizando (2.1.3) obtenemos entonces que :

$$\begin{aligned}
 p_1(x) = & s_1 \frac{(x_2 - x)^2(x - x_1)}{(\Delta x_1)^2} - s_2 \frac{(x - x_1)^2(x_2 - x)}{(\Delta x_1)^2} \\
 & + g(x_1) \frac{(x_2 - x)^2[2(x - x_1) + \Delta x_1]}{(\Delta x_1)^3} + g(x_2) \frac{(x - x_1)^2(2(x_2 - x) + \Delta x_1)}{(\Delta x_1)^3}.
 \end{aligned} \tag{2.1.4}$$

Sustituyendo cada término en (2.1.4) por su valor y simplificando llegamos a que

$$p_1(x) = \frac{(x+3)^2}{8} + \frac{(-1-x)^2}{4}(0.58 + 0.16x).$$

De esta manera similar se obtienen las siguientes expresiones para polinomios p_2, p_3 y p_4 :

$$p_2(x) = -0.5x^3 - x^2 + 1,$$

$$p_3(x) = 0.5x^3 - x^2 + 1,$$

$$p_4(x) = \frac{(3-x)^2}{8} + \frac{(x-1)^2}{4}(0.58 + 0.16x).$$

Por lo tanto, el interpolante cúbico de Hermite para la función $g(x) = \frac{1}{(1+x^2)}$ está dado por

$$f(x) = \begin{cases} \frac{(x+3)^2}{8} + \frac{(-1-x)^2}{4}(0.58 + 0.16x), & -3 \leq x \leq -1; \\ -0.5x^3 - x^2 + 1, & -1 \leq x \leq 0; \\ 0.5x^3 - x^2 + 1, & 0 \leq x \leq 1; \\ \frac{(3-x)^2}{8} + \frac{(x-1)^2}{4}(0.58 + 0.16x), & 1 \leq x \leq 3. \end{cases}$$

El gráfico de la función f aparece en la figura 2.1.

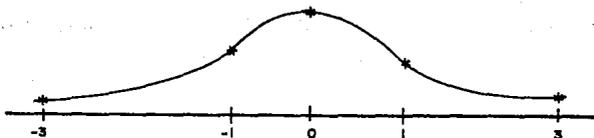


fig 2.1 Interpolante cúbico de Hermite
para $g(x) = \frac{1}{1+x^2}$
x-puntos de interpolación.

2) Consideremos ahora la función :

$$g(x) = x_+^2 = \max(x^2, 0) = \begin{cases} 1, & x \geq 0; \\ 0, & x < 0, \end{cases}$$

y construyamos el interpolante cúbico de Hermite que coincide con g en los puntos $-2, -1, 0$ y 2 . Como $g(x)$ no tiene primera derivada en 0 , vamos a experimentar con diferentes valores s_3 para ver como ésto influye en la forma del interpolante.

Organicemos nuevamente nuestros datos en una tabla

x	-2	-1	0	2
g	0	0	1	0
g'	0	0	s_3	0

tabla 2.2 Valores de la función $g = x_+^2$ y de su primera derivada.

De la expresión (2.1.3) podemos concluir que, independientemente del valor de s_3 , el polinomio p_1 que constituye el interpolante f es siempre el mismo y coincide con el polinomio nulo, ya que $s_1, s_2, g(x_1)$ y $g(x_2)$ valen 0.

Por otro lado, de (2.1.3) obtenemos que

$$p_2(x) = (x+1)^2((s_3-2)x+1),$$

$$p_3(x) = \frac{(2-x)^2}{4}((s_3+1)x+1) + \frac{x^2}{4}(3-x).$$

Por ejemplo, si tomamos $s_3 = 0$, entonces

$$f(x) = \begin{cases} 0, & -2 \leq x \leq 1, \\ (x+1)^2(-2x+1), & -1 \leq x \leq 0, \\ 1, & 0 \leq x \leq 2, \end{cases}$$

mientras que si es $s_3 = 1$, entonces

$$f(x) = \begin{cases} 0, & -2 \leq x \leq 1, \\ (x+1)^2(-x+1), & -1 \leq x \leq 0, \\ \frac{(2-x)^2}{4}(2x+1) + \frac{x^2}{4}(3-x), & 0 \leq x \leq 2, \end{cases}$$

Los gráficos de estas funciones se pueden apreciar en las figuras 2.2 y 2.3 respectivamente.

Nótese que el carácter local de la interpolación de Hermite se aprecia en las mismas, ya que las irregularidades en una vecindad de 0, no se trasladan al resto del intervalo

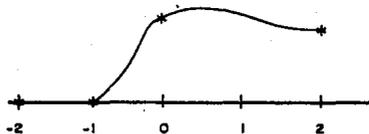
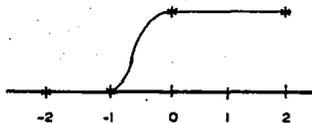


fig. 2.2 Interpolante cúbico de Hermite para x_0^+ con $s_3 = 0$.

fig. 2.3 Interpolante cúbico de Hermite para x_0^+ con $s_3 = 1$.

x-puntos de interpolación.

II.2. SPLINES CUBICOS DE INTERPOLACION

Si los valores de g' en los puntos de interpolación se desconocen, entonces las inclinaciones $(S_i)_2^{-1}$ se puede escoger de modo que el interpolante f tenga hasta segunda derivada continua. En tal caso, se dice que f es un spline cúbico de interpolación. Por lo tanto, el spline cúbico es una función formada por secciones de polinomios cúbicos, que se enlazan con la mayor suavidad posible (sin que necesariamente sea un único polinomio). Derivando dos veces la expresión (2.1.3) se obtiene :

$$p''(x) = -2s_i \frac{2x_{i+1} + x_i - 3x}{(\Delta x_i)^2} - 2s_{i+1} \frac{2x_i + x_{i+1} - 3x}{(\Delta x_i)^2} \\ + 6 \frac{g(x_{i+1}) - g(x_i)}{(\Delta x_i)^3} (x_{i+1} + x_i - 2x).$$

Por lo tanto:

$$f''(x_i^-) = p''_{i-1}(x_i) = \frac{2s_{i-1}}{\Delta x_{i-1}} + \frac{4s_i}{\Delta x_{i-1}} - 6 \frac{g(x_i) - g(x_{i-1})}{(\Delta x_{i-1})^2}, \\ f''(x_i^+) = p''_i(x_i) = \frac{-4s_i}{\Delta x_i} - \frac{2s_{i+1}}{\Delta x_i} + 6 \frac{g(x_{i+1}) - g(x_i)}{(\Delta x_i)^2} \quad (2.2.5)$$

y la condición de continuidad de f'' :

$$f''(x_i^-) = f''(x_i^+), \quad i = 2, \dots, n-1$$

se expresa mediante la ecuación :

$$\frac{1}{\Delta x_{i-1}} s_{i-1} + 2 \left(\frac{1}{\Delta x_{i-1}} + \frac{1}{\Delta x_i} \right) s_i + \frac{1}{\Delta x_i} s_{i+1} = 3 \left[\frac{g(x_i) - g(x_{i-1})}{(\Delta x_{i-1})^2} + \frac{g(x_{i+1}) - g(x_i)}{(\Delta x_i)^2} \right];$$

o en forma más conveniente como:

$$\Delta x_i s_{i-1} + 2(\Delta x_{i-1} + \Delta x_i) s_i + \Delta x_{i-1} s_{i+1} = b_i, \quad i = 2, \dots, n-1 \quad (2.2.6)$$

donde

$$b_i = 3 \left[\Delta x_i \frac{g(x_i) - g(x_{i-1})}{\Delta x_{i-1}} + \Delta x_{i-1} \frac{g(x_{i+1}) - g(x_i)}{\Delta x_i} \right].$$

Suponiendo entonces que s_1 y s_n se seleccionan de alguna forma, (2.2.6) representa un sistema de $n-2$ ecuaciones lineales para calcular las $n-2$ incógnitas s_2, \dots, s_{n-1} .

La matriz de este sistema es tridiagonal y de diagonal dominante por filas. Por lo tanto (2.2.6) tiene exactamente una solución que se puede hallar por el método de eliminación de Gauss sin pivoteo. Los parámetros s_1 y s_n se pueden escoger de diferentes formas. En la siguiente sección presentamos algunas de ellas.

SPLINE CUBICO COMPLETO DE INTERPOLACION

i) Si se conoce el valor de g' en x_1 y x_n entonces resulta muy natural tomar $s_1 = g'(x_1)$ y $s_n = g'(x_n)$. En tal caso el spline cúbico de interpolación no sólo interpola a g en los puntos x_1, \dots, x_n , sino que además interpola a g' en x_1 y x_n . Este spline se conoce como spline cúbico completo de interpolación.

Si hacemos $u_{i-1} = \frac{\Delta x_i}{\Delta x_i + \Delta x_{i-1}}$, $1 - u_{i-1} = \frac{\Delta x_{i-1}}{\Delta x_i + \Delta x_{i-1}}$ entonces el sistema se puede escribir en la forma

$$\begin{pmatrix} 2 & 1 - u_1 & 0 & \cdots & 0 \\ u_2 & 2 & 1 - u_2 & & \vdots \\ 0 & u_3 & 2 & 1 - u_3 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 - u_{n-3} \\ 0 & \cdots & 0 & u_{n-2} & 2 \end{pmatrix} \begin{pmatrix} s_2 \\ s_3 \\ s_4 \\ \vdots \\ s_{n-2} \\ s_{n-1} \end{pmatrix} = \begin{pmatrix} \tilde{\beta}_2 \\ \beta_3 \\ \beta_4 \\ \vdots \\ \beta_{n-2} \\ \beta_{n-1} \end{pmatrix}$$

donde

$$\beta_i = 3[u_{i-1}g[x_{i-1}, x_i] + (1 - u_{i-1})g[x_i, x_{i+1}]] \quad i = 2, \dots, n-1;$$

$$\tilde{\beta}_2 = \beta_2 - u_1 s_1;$$

$$\tilde{\beta}_{n-1} = \beta_{n-1} - (1 - u_{n-2})s_n.$$

SPLINE CUBICO NATURAL

ii) Otra posible selección de las condiciones en los extremos surge de exigir que:

$$f''(x_1) = f''(x_n) = 0. \quad (2.2.7)$$

Utilizando (2.1.5) la condición (2.2.6) se expresa de la siguiente forma, en término de las inclinaciones

$$2s_1 + s_2 = 3 \frac{g(x_2) - g(x_1)}{\Delta x_1};$$

$$s_{n-1} + 2s_n = 3 \frac{g(x_n) - g(x_{n-1})}{\Delta x_{n-1}}.$$

El sistema de ecuaciones queda:

$$\begin{pmatrix} 2 & 1 & 0 & \dots & 0 \\ u_1 & 2 & 1 - u_1 & \dots & \vdots \\ 0 & u_2 & 2 & 1 - u_2 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & 0 & u_{n-2} & 2 & 1 - u_{n-2} \\ 0 & & & 1 & 2 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{n-1} \\ s_n \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{pmatrix}$$

donde

$$\beta_1 = 3g[x_1, x_2]; \quad \beta_n = 3g[x_{n-1}, x_n]$$

El spline que satisface (2.2.7) se conoce como spline natural de interpolación y desde cierto punto de vista no es muy recomendable, ya que la imposición arbitraria de la condición (2.2.7) puede provocar que cerca de los extremos x_1 y x_n el error aumente (a menos que realmente $g''(x_1) = g''(x_n) = 0$). Sin embargo, más adelante mostraremos que el spline natural de interpolación tiene otras propiedades muy interesantes que justifican su uso.

SPINE CUBICO ELIMINA UN NODO

iii) Si uno no conoce nada acerca de las derivadas de g en los puntos extremos, entonces una posibilidad es escoger s_1 y s_n , de manera que p_1 coincida idénticamente con p_2 y p_{n-1} coincida con p_{n-2} . En otras palabras se trata de escoger s_1 y s_n de modo que los puntos x_2 y x_{n-1} no sean puntos de ruptura activos.

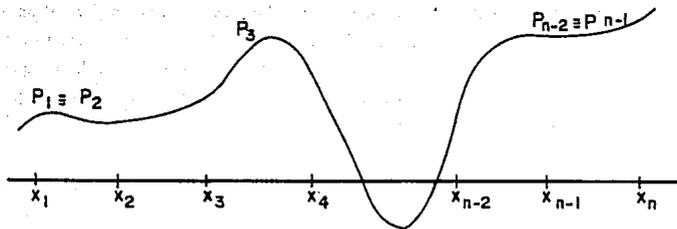


fig 2.4 Los puntos x_2 y x_{n-1} no son puntos de ruptura activos

Esta condición es equivalente a exigir f''' sea continua en x_2 y x_{n-1} y significa añadir la siguiente ecuación al inicio del sistema (2.2.6)

$$s_1 \Delta x_2 + s_2 (x_3 - x_1) = \frac{(\Delta x_1 + 2(x_3 - x_1)) \Delta x_2 (g(x_2) - g(x_1))}{\Delta x_1} + \frac{(\Delta x_1)^2 (g(x_3) - g(x_2))}{\Delta x_2} + \frac{\Delta x_2}{x_3 - x_1}$$

y la ecuación

$$s_{n-1} (x_n + x_{n-2}) + s_n \Delta x_{n-2} = \frac{(\Delta x_{n-1})^2 (g(x_{n-2}) - g(x_{n-1}))}{\Delta x_{n-2}} + \frac{2(x_n - x_{n-2}) + \Delta x_{n-1}}{\Delta x_{n-1}} \frac{\Delta x_{n-2} (g(x_n) - g(x_{n-1}))}{x_n - x_{n-2}}$$

al final del mismo. Esta condición de frontera significa que la primera y la última sección polinomial interpolan a g , en un punto adicional que no es de ruptura. En otra palabras, en vez de $n - 1$ secciones polinomiales tenemos entonces $n - 3$, de modo que f coincide con p_1 en $[x_1, x_3]$ y $p_1(x_i) = g(x_i)$, $i = 1, 2, 3$, mientras que $p_1'(x_3) = s_3$. Similarmente f coincide con p_{n-3} en $[x_{n-2}, x_n]$ y $p_{n-3}(x_i) = g(x_i)$,

$i = n - 2, n - 1, n$, mientras que $p'_{n-3}(x_{n-2}) = s_{n-2}$. Si nos planteamos la condición de frontera en estos términos, entonces el sistema lineal (2.2.6) (y la notación) varía ligeramente. Sin embargo está claro que la función f resultante es idéntica a la obtenida por el planteamiento anterior. Por otro lado, esta forma de interpretar las condiciones de frontera, nos ilustra nuevamente el hecho de que en la interpolación polinomial por tramos, los puntos de interpolación y los de ruptura no necesariamente tienen que coincidir.

En el Apéndice se puede encontrar una subrutina en FORTRAN que calcula el spline cúbico de interpolación utilizando una de las condiciones de frontera ya mencionadas.

Ejemplos.

Calculemos ahora el spline cúbico completo que interpola la función $g(x) = \frac{1}{1+x^2}$ en los puntos -3, -1, 0, 1 y 3. Utilizando los datos de la tabla 2.1 construimos el sistema de ecuaciones (2.2.6) con las condiciones de frontera i). Este sistema se expresa matricialmente de la siguiente forma:

$$\begin{pmatrix} 1 & & & & \\ 1 & 6 & 2 & & \\ & 1 & 4 & 1 & \\ & & 2 & 6 & 1 \\ & & & & 1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{pmatrix} = \begin{pmatrix} 0.06 \\ 3.6 \\ 0 \\ -1.8 \\ -0.06 \end{pmatrix}.$$

La solución del mismo se obtuvo mediante el programa TRIDSIS que aparece en el Apéndice obteniéndose que :

$$s_1 = 0.06, \quad s_2 = 0.62, \quad s_3 = -0.09, \quad s_4 = -0.26 \quad \text{y} \quad s_5 = -0.06.$$

Sustituyendo en la expresión (2.1.3) podemos calcular los cuatro polinomios que constituyen el spline cúbico f . De este modo llegamos a que :

$$f(x) = \begin{cases} 0.07x^3 + 0.56x^2 + 1.53x + 1.54, & -3 \leq x \leq -1; \\ 0.47x^3 - 1.06x^2 - 0.09x + 1, & -1 \leq x \leq 0; \\ 0.65x^3 - 1.06x^2 - 0.09x + 1, & 0 \leq x \leq 1; \\ 0.02x^3 - 0.07x^2 - 0.18x + 0.73, & -1 \leq x \leq 3. \end{cases}$$

El gráfico de esta función se aprecia en la figura 2.5.

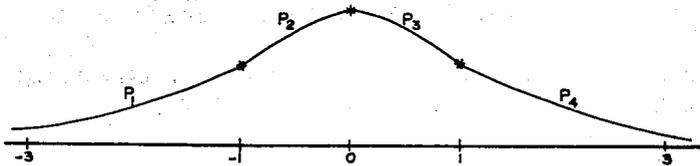


fig. 2.5 Spline cúbico que interpola a la función $g(x) = \frac{1}{1+x^2}$ en los puntos de interpolación.

Para finalizar esta sección construyamos el spline cúbico natural que interpola a la función x^2 en los puntos $-2, -1, 0$ y 2 . Utilizando esta vez los datos de la tabla 2.2 y las condiciones de frontera ii), construimos el sistema de ecuaciones (2.2.6) que nos permite calcular los coeficientes del spline:

$$\begin{pmatrix} 2 & 1 & & \\ 1 & 4 & 1 & \\ & 2 & 6 & 1 \\ & & 1 & 2 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \\ 6 \\ 0 \end{pmatrix}.$$

La solución del sistema se obtuvo otra vez utilizando el programa TRIDSIS y redondeando a dos lugares decimales resulta que:

$$s_1 = -0.3, \quad s_2 = 0.61, \quad s_3 = 0.87, \quad s_4 = 0.43.$$

Mediante la expresión (2.1.3), calculamos entonces los polinomios que forman al spline f y obtenemos finalmente

$$f(x) = \begin{cases} 0.3x^3 + 1.83x^2 + 3.35x + 1.83, & -2 \leq x \leq -1; \\ -0.52x^3 - 0.65x^2 + 0.87x + 1, & -1 \leq x \leq 0; \\ 0.11x^3 - 0.65x^2 + 0.87x + 1, & 0 \leq x \leq 2; \end{cases}$$

donde los coeficientes de cada polinomio también se aproximarán a dos lugares decimales. La figura 2.6 nos muestra en este caso el gráfico de f :

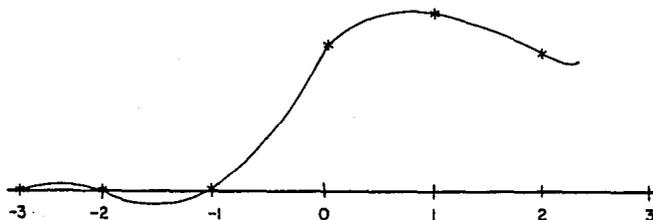


fig. 2.6 Spline cúbico que interpola a la función $g(x) = x^2$.
x-puntos de interpolación.

Observemos con detenimiento la figura 2.1, 2.2, 2.5 y 2.6. Como se puede apreciar, en general el interpolante cúbico de Hermite aproxima mejor a las funciones de prueba que el spline cúbico. Esto se debe esencialmente, a que el spline cúbico invierte, parte de sus posibilidades para aproximar datos, en ser una función suave (tiene hasta segunda derivada continua), mientras que el interpolante cúbico de Hermite, al ser una función menos suave tiene más libertad para modelar un conjunto de datos. Sin embargo, hay que tener en cuenta, que la construcción del interpolante cúbico de Hermite, requiere que conozcamos la primera derivada en los puntos de interpolación, de la función que vamos a aproximar. Pero muchas veces, en la práctica esto no se sabe, porque los datos se obtienen de forma experimental y en realidad la función g es desconocida. Por eso, en tales problemas el spline cúbico es una herramienta fundamental.

II.3. SPLINE CUBICO PARAMETRICO

Con mucha frecuencia, en problemas prácticos de diversa índole, los datos que uno necesita aproximar están dispuestos de modo que para una misma abscisa se tiene más de un valor (vea por ejemplo la figura 2.7).

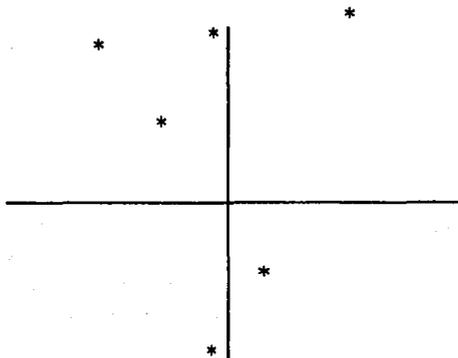


fig. 2.7 Datos de un problema hipotético donde para una abscisa se tienen dos valores diferentes.

En tal caso hay que considerar que los datos representan una curva plana y en consecuencia debemos aproximarlos mediante una curva, ya que una función no puede tomar dos valores diferentes en el mismo punto.

Supongamos entonces que se dispone de un conjunto de puntos (x_i, y_i) , $i = 1, \dots, n$, situados en el plano y se desea construir una curva suave que pase por los mismos. Los puntos (x_i, y_i) se pueden considerar como puntos de una curva, que paraméricamente se describe mediante dos funciones :

$$\begin{aligned} x &= g(t), & a \leq t \leq b, \\ y &= h(t), \end{aligned}$$

de modo que para ciertos valores

$$t_1 < t_2 < \dots < t_n \quad (2.3.8)$$

del parámetro t , se tiene que

$$\begin{aligned} g(t_i) &= x_i; \\ h(t_i) &= y_i, \quad i = 1, \dots, n. \end{aligned}$$

Una vez que se han escogido los valores t_i del parámetro t (que cumpla las condición 2.3.8), podemos entonces construir el spline cúbico f_1 que interpola las abscisas de la curva original, es decir que pasa por los puntos $(t_i, x_i) \quad i = 1, \dots, n$ y el f_2 que interpola las ordenadas, o sea el que toma el valor y_i en cada t_i , considerando en los extremos una de las condiciones que estudiamos en la sección anterior. Este spline pasa por los puntos $(x_i, y_i) \quad i = 1, \dots, n$ y se conoce como spline paramétrico, porque depende de la selección del parámetro t .

La selección de una parametrización adecuada es fundamental, para lograr que el interpolante no tenga rugosidades. En este sentido la experiencia ha demostrado que cualquier parametrización que aproxime la longitud de arco es conveniente. Sin embargo, para lograr que el parámetro sea exactamente la longitud de arco, se necesita un considerable esfuerzo de cálculo. Por eso en la práctica, se toman como parámetros

$$\begin{aligned} t_1 &= 0, \\ t_{i+1} &= t_i + d_i, \quad i = 1, \dots, n-1 \end{aligned} \quad (2.3.9)$$

donde d_i es una aproximación de la longitud de arco, que se puede escoger de una de las siguientes formas :

$$\begin{aligned}
 \text{i)} \quad d_i &= \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2}; \\
 \text{ii)} \quad d_i &= (\Delta x_i)^2 + (\Delta y_i)^2; \\
 \text{iii)} \quad d_i &= |\Delta x_i| + |\Delta y_i|; \\
 \text{iv)} \quad d_i &= \max(|\Delta x_i|, |\Delta y_i|).
 \end{aligned}
 \tag{2.3.10}$$

En particular, la selección (2.3.10) i), corresponde a la longitud de cuerda que une los puntos (x_i, y_i) y (x_{i+1}, y_{i+1}) , y en práctica se ha podido comprobar que es la más conveniente.

Si se desea tener un gráfico del spline paramétrico basta con seleccionar diferentes valores del parámetro t , digamos t_j^* , $j = 1, \dots, m$ y evaluar cada uno de los splines f_1 y f_2 que interpolan las abscisas y ordenadas respectivamente. Entonces los puntos (x_j^*, y_j^*) donde

$$\begin{aligned}
 x_j^* &= f_1(t_j^*); \\
 y_j^* &= f_2(t_j^*), \quad j = 1, \dots, m;
 \end{aligned}$$

pertenecen a la curva que representa el spline paramétrico.

II.4. SPLINE CUBICO PERIODICO

Si los datos originales corresponden a una curva cerrada, entonces resulta natural exigir que el spline cúbico paramétrico, formado por las funciones f_1 y f_2 , que interpolan las abscisas y las ordenadas respectivamente, sea una función periódica, es decir que

$$\begin{aligned} f_1^{(j)}(t_1) &= f_1^{(j)}(t_n); \\ f_2^{(j)}(t_1) &= f_2^{(j)}(t_n), \quad j = 0, 1, 2. \end{aligned} \quad (2.4.11)$$

La condición (2.4.11) para $j=0$, se logra simplemente considerando que el primer y el último punto de interpolación coinciden, o sea que $x_1 = x_n$ y $y_1 = y_n$, pues en tal caso, como

$$\begin{aligned} f_1(t_1) &= x_1; \\ f_2(t_1) &= y_1; \end{aligned}$$

entonces

$$\begin{aligned} f_1(t_1) &= x_1 = x_n = f_1(t_n); \\ f_2(t_1) &= y_1 = y_n = f_2(t_n), \quad i = 1, \dots, n. \end{aligned}$$

Para $j = 1$, la condición (2.4.11) se obtiene considerando que el sistema de ecuaciones (2.2.6), que nos permite calcular los coeficientes del spline que interpola las abscisas y las ordenadas respectivamente, se elimina la incógnita s_n , porque tomamos $s_n = s_1$.

Por último, teniendo en cuenta que $s_1 = s_n$, de (2.2.5) obtenemos que la condición (2.4.11) para f_1 y $j = 2$, se expresa mediante la ecuación

$$2(\Delta t_1 + \Delta t_{n-1})s_1 + \Delta t_{n-1}s_2 + \Delta t_1 s_{n-1} = 3 \left[\Delta t_{n-1} \frac{x_2 - x_1}{\Delta t_1} + \Delta t_1 \frac{x_n - x_{n-1}}{\Delta t_{n-1}} \right].$$

Por lo tanto, las inclinaciones s_1, \dots, s_{n-1} del spline f_1 que interpola las abscisas de los datos, se obtienen como solución del siguiente sistema de ecuaciones lineales

$$\begin{pmatrix} 2(\Delta t_1 + \Delta t_{n-1}) & \Delta t_{n-1} & 0 \dots 0 & \Delta t_1 \\ \Delta t_2 & 2(\Delta t_1 + \Delta t_2) & \Delta t_1 & 0 \dots \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \Delta t_{n-1} \\ \Delta t_{n-2} & 0 \dots 0 & \Delta t_{n-1} & 2(\Delta t_{n-2} + \Delta t_{n-1}) \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{n-1} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{pmatrix}$$

donde

$$\begin{aligned} b_1 &= 3 \left[\Delta t_{n-1} \frac{x_2 - x_1}{\Delta t_1} + \Delta t_1 \frac{x_n - x_{n-1}}{\Delta t_{n-1}} \right]; \\ b_i &= 3 \left[\Delta t_i \frac{x_i - x_{i-1}}{\Delta t_{i-1}} + \Delta t_{i-1} \frac{x_{i+1} - x_i}{\Delta t_i} \right]; \quad i = 2, \dots, n-1. \end{aligned} \quad (2.4.12)$$

Para calcular las inclinaciones s_1, \dots, s_{n-1} del spline f_2 que interpola las ordenadas de los datos, hay que resolver un sistema lineal con la misma matriz que (2.4.12) donde el término independiente b esta dado por

$$\begin{aligned} b_1 &= 3 \left[\Delta t_{n-1} \frac{y_2 - y_1}{\Delta t_1} + \Delta t_1 \frac{y_n - y_{n-1}}{\Delta t_{n-1}} \right], \\ b_i &= 3 \left[\Delta t_i \frac{y_i - y_{i-1}}{\Delta t_{i-1}} + \Delta t_{i-1} \frac{y_{i+1} - y_i}{\Delta t_i} \right], \quad i = 2, \dots, n-1. \end{aligned} \quad (2.4.13)$$

Nótese que la matriz del sistema (2.4.12) es tridiagonal cíclica y de diagonal dominante por filas. Este tipo de sistema también se puede resolver fácilmente por

el método de eliminación de Gauss sin pivoteo. En el Apéndice, el lector puede encontrar una descripción sencilla del método de Gauss para este caso.

Finalmente, utilizando (2.1.3), obtenemos que el spline f_1 que interpola las abscisas de los datos originales, esta constituido por los siguientes polinomios.

$$\begin{aligned}
 p_i(t) = & s_i \frac{(t_{i+1} - t)^2(t - t_i)}{(\Delta t_i)^2} + s_{i+1} \frac{(t - t_i)^2(t_{i+1} - t)}{(\Delta t_i)^2} \\
 & + \frac{x_i(t_{i+1} - t)^2(2(t - t_i) + \Delta t_i)}{(\Delta t_i)^3} + \frac{x_{i+1}(t - t_i)^2(2(t_{i+1} - t) + \Delta t_i)}{(\Delta t_i)^3} \\
 & t_1 \leq t \leq t_{i+1}, \quad i = 1, \dots, n-1
 \end{aligned}
 \tag{2.4.14}$$

donde los s_i se calculan como solución del sistema (2.4.12) y $s_1 = s_n$. Los polinomios que forman el spline f_2 , que interpola las ordenadas de los datos originales, se obtienen simplemente sustituyendo en (2.4.14) x_i por y_i y x_{i+1} por y_{i+1} y calculando los s_i como solución del sistema (2.4.12) con el lado derecho (2.4.13).

Ejemplo.

Los datos de la tabla 2.3 representa valores de la lemniscata

$$s = \sqrt{2\cos(2\theta)}; \quad |\theta| \leq \pi/4. \tag{2.4.15}$$

Como la lemniscata tiene dos lazos (ver fig. 2.8) es muy importante que los datos se organicen como muestra la tabla, es decir siguiendo el recorrido de la curva a partir de un punto inicial, que puede ser arbitrario (ver fig. 2.8).

ángulo	radio
θ	ρ
0	$\sqrt{2}$
30	1
45	0
210	1
180	$\sqrt{2}$
150	1
135	0
330	1
0	$\sqrt{2}$

tabla 2.3 Puntos de la Lemniscata (2.4.14).

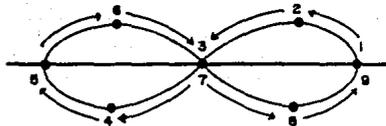


fig. 2.8 Lemniscata.

Los puntos aparecen organizados en el sentido de recorrido de la curva.

Para utilizar la parametrización (2.4.9) debemos expresar los datos de la tabla 2.3 en coordenadas cartesianas, mediante las ecuaciones de transformación de coordenadas

$$x = \rho \cos \theta,$$

$$y = \rho \operatorname{sen} \theta.$$

De este modo obtenemos la tabla 2.4. Si ahora utilizamos la condición (2.4.10) i), resulta entonces, que por ejemplo (utilizando dos dígitos decimales)

$$d_1 = \sqrt{(\Delta x_1)^2 + (\Delta y_1)^2} = 0.74.$$

luego

$$t_2 = t_1 + d_1 = 0.74.$$

Continuando de esta forma calculamos el resto de los d_i y los t_i , $i = 2, \dots, 9$.

x	$\sqrt{2}$	$\sqrt{3}/2$	0	$-\sqrt{3}/2$	$-\sqrt{2}$	$-\sqrt{3}/2$	0	$\sqrt{3}/2$	$\sqrt{2}$
y	0	1/2	0	-1/2	0	1/2	0	-1/2	0

tabla 2.4 Puntos de la Lemniscata (2.4.15) dados en coordenadas cartesianas.

Por lo tanto, a partir de (2.4.14) se pueden calcular los polinomios $p_i(t)$, $i = 1, \dots, 8$ que constituyen el spline f_1 .

Por otro lado, los coeficientes del spline f_2 se obtienen resolviendo el sistema (2.4.16), donde el lado derecho se sustituye por el vector

$$b = (3, 0.91, -3, 0.91, 3, 0.91, -3, 0.91)^t.$$

Mediante el programa TRIDCSIS se obtiene entonces que los coeficientes del spline f_2 que interpolan las ordenadas son:

$$\begin{aligned} s_1 &= 0.93, & s_2 &= 0.17, & s_3 &= -0.84, & s_4 &= 0.17, \\ s_5 &= 0.93, & s_6 &= 0.17, & s_7 &= -0.84, & s_8 &= 0.17. \end{aligned}$$

Los polinomios que constituyen el spline f_2 se calculan sustituyendo en (2.4.14) estos valores y x_i por y_i , $i = 1, \dots, 9$.

Para construir un gráfico del spline de interpolación podemos tomar, por ejemplo, 50 puntos equidistantes del intervalo $[t_1, t_9] = [0, 6.97]$

$$t_j^* = t_1 + (j-1)h, \quad j = 1, \dots, 50 \text{ con } h = \frac{t_9 - t_1}{49} = 0.142$$

y evaluar los splines f_1 y f_2 en estos puntos, es decir calcular:

$$\begin{aligned} x_j^* &= f_1(t_j^*), \\ y_j^* &= f_2(t_j^*), \quad j = 1, \dots, 50. \end{aligned}$$

Capítulo III

Ajuste

y

Suavizamiento

III. SUAVIZAMIENTO Y AJUTE

Con mucha frecuencia, en problemas de la ciencia y la técnica, los especialistas disponen de datos que no pueden considerarse precisos. Estos datos se obtienen por lo general como resultado de algún experimento o medición, y están afectados con "ruidos" debido a la imprecisión de los equipos o técnicas que se utilizaron para obtenerlos.

Esta claro entonces que los métodos de interpolación que hemos estudiado hasta aquí, no nos servirían para aproximar tales datos, puesto que no tiene sentido construir una función que pase por los mismos. Por eso, en lo que resta de este capítulo nos vamos a dedicar a la construcción de otro tipo de aproximación que sea capaz de reflejar la tendencia o el comportamiento de los datos ignorando el ruido de los mismos.

III.1. PROPIEDAD DE NORMA MINIMA DEL SPLINE CUBICO NATURAL

En esta sección vamos a probar que entre todas las funciones $g \in C^2$ que interpolan un conjunto de datos (x_i, y_i) $i = 1, \dots, n$, el spline cúbico natural que definimos en II.2, es la función más "suave" en el sentido de que minimiza el funcional

$$I(g) = \int_{x_1}^{x_n} [g''(x)]^2 dx \quad (3.1.1)$$

precisamente debido a esta propiedad las funciones spline recibieron tal nombre, ya que durante mucho tiempo para dibujar una curva suave que pasara por un conjunto de puntos prefijados los ingenieros utilizaban un instrumento llamado spline.

Este instrumento mecánico estaba formado por una varilla flexible de acero a la cual se le sujetaban ciertos pesos (ver fig 3.1), que hacían que la varilla se flexionara y la curva resultante f pasara por los puntos.

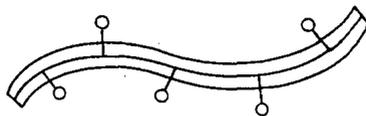


fig 3.1 Spline mecánico.

De este modo, el spline mecánico f minimizaba la energía de tensión que aproximadamente es proporcional a (3.1.1). De ahí que por analogía, la función que minimiza $I(g)$ recibiera el nombre de spline.

Teorema 3.1.

Sea $f(x)$ el único spline cúbico natural que interpola los datos (x_i, y_i) , $i = 1, \dots, n$ ($n \geq 2$), y sea $g(x)$ cualquier función de clase C^2 tal que

$$g(x_i) = y_i, \quad i = 1, \dots, n.$$

entonces:

$$I(g) \geq I(f) \tag{3.1.2}$$

y la igualdad en (3.1.2) se cumple si y sólo si $g(x) \equiv f(x)$.

Demostración.

Como evidentemente

$$g''(x) = f''(x) + [g''(x) - f''(x)],$$

entonces :

$$\int_{x_1}^{x_n} [g''(x)]^2 dx = \int_{x_1}^{x_n} [f''(x)]^2 dx + \int_{x_1}^{x_n} [g''(x) - f''(x)]^2 dx \\ + 2 \int_{x_1}^{x_n} f''(x) [g''(x) - f''(x)] dx, \quad (3.1.3)$$

calculemos la última integral del lado derecho de (3.1.3) que denotaremos por I .

Integrando por partes obtenemos :

$$I = f''(x) [g'(x) - f'(x)] \Big|_{x_1}^{x_n} - \int_{x_1}^{x_n} f'''(x) [g'(x) - f'(x)] dx, \quad (3.1.4)$$

como f es un spline cúbico natural, entonces $f''(x_1) = f''(x_n) = 0$. Por lo tanto, en (3.1.4) nos queda

$$I = - \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} f'''(x) [g'(x) - f'(x)] dx,$$

pero $f'''(x)$ es una función constante por tramos, cuyo valor en el intervalo $[x_i, x_{i+1}]$ denotaremos por h_i . Luego

$$I = - \sum_{i=1}^{n-1} h_i \left[(g(x_{i+1}) - f(x_{i+1})) - (g(x_i) - f(x_i)) \right]$$

y como f y g interpolan los mismos datos (x_i, y_i) , $i = 1, \dots, n$, entonces $g(x_i) - f(x_i) = 0$ para toda i . Por lo tanto, $I = 0$ y de (3.1.4) llegamos a que

$$\int_{x_1}^{x_n} [g''(x)]^2 dx = \int_{x_1}^{x_n} [f''(x)]^2 dx + \int_{x_1}^{x_n} [g''(x) - f''(x)]^2 dx \quad (3.1.5)$$

de aquí resulta inmediatamente (3.1.2), ya que la última integral en (3.1.5) es no negativa. Por otro lado, la igualdad en (3.1.2) se obtiene si y sólo si $g''(x) - f''(x)$ es idénticamente nula, lo cual implica que $g - f$ es un polinomio de grado ≤ 1 . Sin embargo, de acuerdo con las condiciones de interpolación, la función $g - f$ se anula en n puntos ($n \geq 2$) y por lo tanto $g - f$ tiene que ser el polinomio nulo o en otras palabras $g \equiv f$, lo cual completa la demostración. ■

Como veremos, en la siguiente sección, la propiedad de norma mínima del spline cúbico natural es muy útil cuando estamos interesados en construir una curva "suave" que aproxime un conjunto de datos, sin que necesariamente los interpole.

III.2. SUAVIZAMIENTO DE DATOS CON "RUIDO"

Planteamiento del problema.

Como ya mencionamos al inicio de este capítulo, el problema de suavizar un conjunto de datos, está vinculado a la necesidad de construir una aproximación que refleje la tendencia de datos imprecisos, lo cual evidentemente no se logra interpolando los mismos. Las características más importantes de este problema se pueden resumir en los siguientes aspectos:

1. El especialista necesita poca precisión en los resultados (a lo sumo 3 dígitos correctos).
2. Los datos son sustancialmente imprecisos, de modo que el ruido que los afecta usualmente es aleatorio pero se desconoce su distribución.
3. No se dispone de suficiente información especializada sobre el modelo de comportamiento de los datos.
4. Los datos son discretos y resulta muy difícil o muy costoso obtener información adicional.

Supongamos que g es una función suave (usualmente desconocida) y que

$$y_i = g(x_i) + e_i, \quad i = 1, \dots, n$$

es el valor de g en el punto x_i afectado con el ruido e_i . El problema de suavizar los datos (x_i, y_i) se puede formular entonces, como el de construir una función "suave" que aproxime a los datos (x_i, y_i) , $i = 1, \dots, n$, de modo que se conserve la información sobre g , contenida en los y_i y se elimine el ruido e_i .

En otras palabras, se trata de encontrar la función más "suave", cuya distancia a los datos se mantenga acotada por una magnitud que usualmente conoce el especialista a partir del problema práctico que representan los mismos.

Ahora para que esta formulación del problema sea precisa debemos aclarar los términos de función "suave" y de "distancia".

Como medida de la distancia de una función f a los datos vamos a utilizar el fun-

cional

$$S(f) = \sum_{i=1}^n \left[\frac{f(x_i) - y_i}{\delta y_i} \right]^2 \quad (3.2.1)$$

donde δy_i es una estimación de la varianza de los y_i . Por ejemplo, en ciertos problemas técnicos el experimento que nos permite obtener los y_i se repite varias veces (réplicas) de modo que en vez de uno sólo y_i , para cada x_i tenemos un conjunto de $y_i^{(j)}$, $j = 1, \dots, m$, donde m es el número de repeticiones. En tal caso es aconsejable tomar como y_i en (3.2.1), la media de los $y_i^{(j)}$, es decir

$$y_i = \left(\sum_{j=1}^m y_i^{(j)} \right) / m$$

y como δy_i la varianza de cada y_i , o sea

$$\delta y_i = \left(\sum_{j=1}^m \left(y_i^{(j)} - y_i \right)^2 / (m - 1) \right)^{\frac{1}{2}}.$$

Otras veces el experimento sólo se realiza una vez, pero se conoce que el equipo de medición que se utilizó para obtener los y_i tiene cierta precisión. Por ejemplo, si el equipo nos permitiera medir los y_i con un 1 % de error, entonces es razonable tomar en (3.2.1) $\delta y_i = 0.01$ para todo i .

Por otra parte, como medida de la suavidad de una función f , tomaremos el funcional I definido por (3.1.1), pues es conocido que $I(f)$ nos da una aproximación de la curvatura de f en el intervalo $[x_1, x_n]$.

Esto significa que en términos matemáticos, el problema de suavizamiento de los datos se puede formular, como el de hallar una función que minimice $I(f)$ entre todas aquellas para las cuales $S(f)$ está acotado, es decir

$$\min I(f) \quad \text{con } f \in C^2[x_1, x_n] \quad \text{sujeto a } S(f) \leq \sigma \quad (I)$$

donde σ es la distancia máxima entre la aproximación, y los datos que el especialista desea.

Teorema 3.2.

Si el problema I tiene solución entonces una solución del mismo, es un spline cúbico natural y es única.

Demostración.

Supongamos que f^* es una solución de I . De los resultados del capítulo II sabemos que existe un único spline cúbico natural f , que interpola a f^* en los puntos x_1, \dots, x_n , es decir tal que

$$f^*(x_i) = f(x_i), \quad i = 1, \dots, n,$$

pero entonces de (3.2.1) tenemos que

$$S(f) = S(f^*) \leq \sigma$$

es decir, f es una solución factible de I . Más aún, del Teorema 3.1 podemos asegurar que

$$I(f) \leq I(f^*)$$

y que la desigualdad es estricta a menos que $f^* = f$. Por lo tanto, f es la única solución óptima de I . ■

III.3. PLANTEAMIENTO Y SOLUCION DE UN PROBLEMA EQUIVALENTE

Aunque el teorema 3.2 nos asegura que la solución del problema I es un spline cúbico natural, su demostración no nos ofrece un método para construir el mismo. Por eso, en la práctica, en vez de resolver el problema I vamos a considerar el siguiente problema

$$\min \Delta_p(f) \quad \text{con} \quad f \in C^2[x_1, x_n]$$

donde

$$\Delta_p(f) = pS(f) + (1-p)I(f) \quad \text{y} \quad p \in (0, 1). \quad II$$

Al final de esta misma sección, probaremos que para cierto valor del parámetro p en el intervalo $(0,1)$, la función f_p que resuelve el problema II , es la solución óptima de I .

Pero, supongamos inicialmente que el parámetro p aún está libre en $(0,1)$. El primer término de la suma en Δ_p nos proporciona una medida de la distancia de la aproximación a los datos, mientras que el segundo constituye una estimación de la suavidad de f , la minimización de Δ_p establece una especie de compromiso entre éstos dos aspectos, de modo que, la selección del parámetro p nos da la posibilidad de escoger a cual de ellos le damos la mayor importancia. Por ejemplo, si p está cerca de 1, entonces la minimización de Δ_p requiere sobre todo minimizar la sumatoria y eso significa, que estamos más interesados en ajustar los datos que en obtener una aproximación suave. Si por el contrario, tomamos p cerca de 0 entonces para minimizar Δ_p hay que lograr que la integral sea pequeña y por lo tanto, en ese caso, le damos más peso al hecho de que la curva de aproximación sea suave.

El siguiente teorema, nos demuestra que la solución del problema II también es un spline cúbico natural.

Teorema 3.3

Sea $f_p \in C^2[x_1, x_n]$, la función que resuelve el problema II para un p fijo en el intervalo $(0,1)$. Entonces f_p es un spline cúbico natural, y satisface adicionalmente

las siguientes condiciones:

$$\begin{aligned} \text{i)} \quad & -f_p'''(x_1^+) = \frac{p}{1-p} \frac{f(x_1) - y_1}{(\delta y_1)^2}, \\ \text{ii)} \quad & f_p'''(x_i^-) - f_p'''(x_i^+) = \frac{p}{1-p} \frac{f(x_i) - y_i}{(\delta y_i)^2}, \\ \text{iii)} \quad & f_p'''(x_n^-) = \frac{p}{1-p} \frac{f(x_n) - y_n}{(\delta y_n)^2}. \end{aligned}$$

Demostración.

El hecho, de que f_p es un spline cúbico natural, se obtiene de manera muy similar a como hicimos en el teorema 3.2. En efecto, si f^* fuese la solución óptima de II, entonces el único spline cúbico natural f_p , que interpola a f^* en los puntos x_1, \dots, x_n , satisface que

$$S(f_p) = S(f^*), \quad (3.3.1)$$

pero de acuerdo con el teorema 3.1

$$I(f_p) \leq I(f^*) \quad (3.3.2)$$

por lo tanto, combinando (3.3.1) y (3.3.2) obtenemos que

$$pS(f_p) + (1-p)I(f_p) \leq pS(f^*) + (1-p)I(f^*) \quad (3.3.3)$$

y la desigualdad en (3.3.3) es estricta, a menos que $f^* \equiv f_p$. En otras palabras

$$\Delta_p(f_p) \leq \Delta_p(f^*).$$

Veamos ahora que f_p satisface las condiciones i), ii) y iii). Del cálculo variacional es conocido que si f_p es la solución del problema II, entonces la variación del funcional Δ_p debe anularse para toda función $h \in C^2[x_1, x_n]$, es decir

$$\delta[\Delta_p(f_p); h] := \left. \frac{d}{de} \Delta_p(f_p + eh) \right|_{e=0} = 0 \quad \text{para toda } h \in C^2[x_1, x_n].$$

Pero

$$\delta[\Delta_p(f_p); h] = p\delta[S(f_p); h] + (1-p)\delta[I(f_p); h]. \quad (3.3.4)$$

Por eso vamos a calcular a continuación la variación de los funcionales $S(f_p)$ e $I(f_p)$. De (3.2.1) tenemos que :

$$\begin{aligned} \delta S[f_p; h] &= \frac{d}{de} \sum_{i=1}^n \left[\frac{f_p(x_i) + eh(x_i) - y_i}{\delta y_i} \right]^2 \Big|_{e=0} \\ &= 2 \sum_{i=1}^n \left[\frac{f_p(x_i) + eh(x_i) - y_i}{\delta y_i} \right] \frac{h(x_i)}{\delta y_i} \Big|_{e=0} \\ &= 2 \sum_{i=1}^n \left[\frac{f_p(x_i) - y_i}{\delta y_i} \right] \frac{h(x_i)}{\delta y_i}. \end{aligned} \quad (3.3.5)$$

Por otro lado, de (3.1.1) resulta que :

$$\begin{aligned} \delta I[f_p; h] &= \frac{d}{de} \int_{x_1}^{x_n} [f_p''(x) - eh''(x)]^2 dx \Big|_{e=0} \\ &= 2 \int_{x_1}^{x_n} [f_p''(x) + eh''(x)] h'' dx \Big|_{e=0} \\ &= 2 \int_{x_1}^{x_n} f_p''(x) h''(x) dx \\ &= 2 \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} f_p''(x) h''(x) dx. \end{aligned}$$

Integrando dos veces por partes se obtiene entonces que :

$$\begin{aligned} \delta I[f_p; h] &= 2 \left[-f_p''(x_1^+) h'(x_1) + \sum_{i=2}^{n-1} \left(f_p''(x_i^-) - f_p''(x_i^+) \right) h'(x_i) \right. \\ &\quad + f_p''(x_n^-) h'(x_n) + f_p'''(x_1^+) h(x_1) - \sum_{i=2}^{n-1} \left(f_p'''(x_i^-) - f_p'''(x_i^+) \right) h(x_i) \\ &\quad \left. - f_p'''(x_n^-) h(x_n) + \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} f_p^{iv}(x) h(x) dx \right]. \end{aligned} \quad (3.3.6)$$

Ahora como $f_p(x)$ es un spline cúbico natural entonces $f_p''(x_1^+) = f_p''(x_n^-) = 0$ y $f_p^{iv}(x) \equiv 0$. Además $f_p \in C^2[x_1, x_n]$, luego $f_p''(x_i^-) = f_p''(x_i^+)$, $i = 2, \dots, n-1$.

Por lo tanto (3.3.6) se reduce a

$$\delta[f_p; h] = 2 \left[f_p'''(x_1^+) h(x_1) - \sum_{i=2}^{n-1} \left(f_p'''(x_i^-) - f_p'''(x_i^+) \right) h(x_i) - f_p'''(x_n^-) h(x_n) \right]. \quad (3.3.7)$$

Sustituyendo (3.3.5) y (3.3.7) en (3.3.4) obtenemos que si $f_p \in C^2[x_1, x_n]$ es la función que minimiza Δ_p entonces para toda $h \in C^2[x_1, x_n]$ se debe cumplir que

$$2p \left[\left(\frac{f_p(x_1) - y_1}{(\delta y_1)^2} + \frac{1-p}{p} f_p'''(x_1^+) \right) h(x_1) + \sum_{i=2}^{n-1} \left(\frac{f_p(x_i) - y_i}{(\delta y_i)^2} - \frac{1-p}{p} \left(f_p'''(x_i^-) - f_p'''(x_i^+) \right) \right) h(x_i) + \left(\frac{f_p(x_n) - y_n}{(\delta y_n)^2} - \frac{1-p}{p} f_p'''(x_n^-) \right) h(x_n) \right] = 0. \quad (3.3.8)$$

Finalmente, de (3.3.8) resulta evidente que f_p debe satisfacer entonces, las condiciones i), ii) y iii) con lo cual concluye la demostración del teorema ■.

A diferencia del teorema 3.2, el teorema 3.3 no sólo nos dice que la solución del problema II es un spline cúbico natural, si no que además nos informa que este spline satisface las condiciones i), ii) y iii) gracias a ésto la construcción explícita de tal spline se puede llevar a cabo sin ninguna dificultad, como probaremos en la siguiente sección.

Por otra parte, este spline natural no necesariamente interpola los datos (como el que construimos en el problema I). En particular, si $p = 0^+$ la solución óptima del problema II es la recta de ajuste mínimo cuadrada, o sea la aproximación mínimo cuadrada para los datos desde el espacio de los polinomios de grado ≤ 1 . En efecto, cuando $p = 0^+$ la recta f_p que se halla más cerca de los datos (en el sentido de que minimiza $S(f)$) es la solución óptima del problema II, ya que $I(f_{0^+}) = 0$ mientras que $I(f_p) \geq 0$ para todo p . Nótese que una recta también es un spline cúbico natural, pues está formada por "secciones" de polinomios de grado ≤ 3 , tiene hasta segunda derivada continua y en los extremos vale 0.

Sin embargo, si $p = 1^-$ la solución óptima del problema *II* es el spline cúbico natural que interpola los datos (ver sección 1) ya que esta función anula $S(f_p)$ y entre todas las funciones que interpolan los datos (y en consecuencia hacen 0 la sumatoria) es la que minimiza $I(f)$ (teorema 3.1).

De todo lo anterior podemos concluir que la solución del problema *II* es un spline cúbico natural, que (en la medida en que el parámetro p varía entre 0^+ y 1^-) oscila entre la recta que mejor ajusta los datos y el spline cúbico natural que los interpola.

Para finalizar esta sección veamos entonces cuál es la relación que existe entre el problema *I* y *II*.

Teorema 3.4

Supongamos que para $p = p_\sigma$ el spline cúbico natural f_{p_σ} que resuelve el problema *II* satisface que:

$$S(f_{p_\sigma}) = \sigma, \quad (3.3.9)$$

entonces f_{p_σ} es la solución óptima del problema *I*.

Demostración.

Supongamos que f^* es la solución óptima del problema *I* y que $f^* \neq f_{p_\sigma}$. Como f_{p_σ} satisface (3.3.9) es una solución factible del problema *I* y por lo tanto debía cumplirse que

$$I(f^*) < I(f_{p_\sigma}), \quad (3.3.10)$$

pero

$$S(f^*) \leq \sigma = S(f_{p_\sigma}). \quad (3.3.11)$$

Luego de (3.3.10) y (3.3.11) obtendríamos que

$$p_\sigma S(f^*) + (1 - p_\sigma) I(f^*) < p_\sigma S(f_{p_\sigma}) + (1 - p_\sigma) I(f_{p_\sigma})$$

y ésto es imposible puesto que f_{p_σ} es la solución del problema *II* para $p = p_\sigma$ ■

Como consecuencia inmediata de este teorema podemos concluir que para resolver el problema *I*, basta con hallar la raíz p_σ de la ecuación (3.3.9) y resolver el problema *II* con $p = p_\sigma$. Por eso en la siguiente sección vamos a construir explícitamente el spline cúbico natural que resuelve el problema *II* para un p fijo.

III.4. CONSTRUCCION DEL SPLINE CUBICO DE SUAVIZAMIENTO

Con los resultados de las secciones anteriores ya estamos en condiciones de obtener explícitamente el spline cúbico natural que resuelve el problema II para un p fijo en $(0,1)$. Como $f_p(x)$ es un polinomio de grado ≤ 3 para $x \in [x_i, x_{i+1}]$, $i = 1, \dots, n-1$, entonces su segunda derivada en ese intervalo es la recta

$$f_p''(x) = f_p''(x_{i+1}) \frac{x - x_i}{x_{i+1} - x_i} - f_p''(x_i) \frac{x - x_{i+1}}{x_{i+1} - x_i}.$$

Sea $c_i = f_p''(x_i)/2$, entonces f_p'' se puede escribir como

$$f_p''(x) = 2c_{i+1} \frac{x - x_i}{\Delta x_i} - 2c_i \frac{x - x_{i+1}}{\Delta x_i} \quad (3.4.1)$$

donde $\Delta x_i = x_{i+1} - x_i$.

Integrando 2 veces en (3.4.1) obtenemos que

$$f_p(x) = c_{i+1} \frac{(x - x_i)^3}{3\Delta x_i} - c_i \frac{(x - x_{i+1})^3}{3\Delta x_i} + b_i x + d_i \quad (3.4.2)$$

donde b_i y d_i son constantes a determinar.

Denotemos por $a_j = f_p(x_j)$, $j = i, i+1$, entonces de (3.4.2) está claro que $b_i x + d_i$ es la recta que pasa por los puntos $(x_i, a_i - \frac{c_i}{3} \Delta x_i^2)$ y $(x_{i+1}, a_{i+1} - \frac{c_{i+1}}{3} \Delta x_i^2)$.

luego

$$b_i x + d_i = (a_{i+1} - \frac{c_{i+1}}{3} \Delta x_i^2) \frac{x - x_i}{\Delta x_i} - (a_i - \frac{c_i}{3} \Delta x_i^2) \frac{x - x_{i+1}}{\Delta x_i}$$

y sustituyendo en (3.4.2) obtenemos una expresión para $f_p(x)$ que sólo depende de los parámetros c_i y a_i , $i = 1, \dots, n-1$,

$$f_p(x) = c_{i+1} \frac{(x - x_i)^3}{3\Delta x_i} - c_i \frac{(x - x_{i+1})^3}{3\Delta x_i} + (a_{i+1} - \frac{c_{i+1}}{3} \Delta x_i^2) \frac{x - x_i}{\Delta x_i} - (a_i - \frac{c_i}{3} \Delta x_i^2) \frac{x - x_{i+1}}{\Delta x_i}. \quad (3.4.3)$$

Ahora de (3.4.1) y (3.4.3) está claro que $f_p(x)$ y $f_p''(x)$ son continuas, ya que :

$$\begin{aligned} f_p(x_i^-) &= a_i = f_p(x_i^+), \\ f_p''(x_i^-) &= 2c_i = f_p''(x_i^+) \end{aligned} \quad i = 1, \dots, n-1$$

luego, para garantizar que $f_p(x)$ sea un spline cúbico natural sólo nos resta exigir que

$$f_p''(x_1^-) = 0, \quad (3.4.4)$$

$$f_p'(x_i^-) = f_p'(x_i^+), \quad i = 2, \dots, n-1 \quad (3.4.5)$$

$$f_p''(x_n^-) = 0, \quad (3.4.6)$$

Derivando en (3.4.3) obtenemos que :

$$f_p'(x) = c_{i+1} \frac{(x-x_i)^2}{\Delta x_i} - c_i \frac{(x-x_{i+1})^2}{\Delta x_i} + \left(\frac{a_{i+1}}{\Delta x_i} - \frac{c_{i+1}}{3} \Delta x_i \right) - \left(\frac{a_i}{\Delta x_i} - \frac{c_i}{3} \Delta x_i \right)$$

por lo tanto, la condición (3.4.5) se expresa mediante la ecuación

$$\begin{aligned} -c_i \Delta x_i + \frac{a_{i+1}}{\Delta x_i} - \frac{c_{i+1}}{3} \Delta x_i - \frac{a_i}{\Delta x_i} + \frac{c_i}{3} \Delta x_i &= c_i \Delta x_{i-1} + \frac{a_i}{\Delta x_{i-1}} - \frac{c_i}{3} \Delta x_{i-1} \\ &\quad - \frac{a_{i-1}}{\Delta x_{i-1}} + \frac{c_{i-1}}{3} \Delta x_{i-1}, \\ i &= 2, \dots, n-1. \end{aligned}$$

Agrupando ahora los términos de c_i a la izquierda y los de a_i a la derecha obtenemos finalmente que f_p' es continua si los parámetros a_i y c_i , $i = 1, \dots, n$, satisfacen las ecuaciones

$$c_{i-1} \Delta x_{i-1} + 2c_i (\Delta x_{i-1} + \Delta x_i) + c_{i+1} \Delta x_i = 3 \left(\frac{\Delta a_i}{\Delta x_i} - \frac{\Delta a_{i-1}}{\Delta x_{i-1}} \right), \quad i = 2, \dots, n-1 \quad (3.4.7)$$

donde $\Delta a_i = a_{i+1} - a_i$.

Añadiendo a las ecuaciones (3.4.7) las condiciones (3.4.4) y (3.4.6) $c_1 = c_n = 0$ obtenemos un sistema de n ecuaciones lineales para el cálculo de las c_i en términos de las a_i . Denotemos por c el vector de $n-2$ componentes $c = (c_2, \dots, c_{n-1})^t$ y por

Nótese que el vector c no incluye las componentes c_1 y c_n ya que $c_1 = c_n = 0$. Pero hasta aquí, sólo hemos utilizado el hecho de que $f_p(x)$ es un spline cúbico natural. Sin embargo, en el teorema 3.3 demostramos que la solución del problema II no es cualquier spline cúbico natural, sino que además satisface las condiciones i), ii) y iii). Derivando en (3.4.1) obtenemos directamente que

$$f_p'''(x) = 2 \frac{\Delta c_i}{\Delta x_i} \quad \text{para toda } x \in (x_i, x_{i+1}) \quad (3.4.9)$$

donde $\Delta c_i = c_{i+1} - c_i$.

Pero como $f_p'''(x)$ es una función constante por tramos, entonces de (3.4.9) tenemos que

$$\begin{aligned} f_p'''(x_1^+) &= 2 \frac{\Delta c_1}{\Delta x_1}, \\ f_p'''(x_i^+) &= 2 \frac{\Delta c_i}{\Delta x_i}, \quad f_p'''(x_i^-) = 2 \frac{\Delta c_{i-1}}{\Delta x_{i-1}}, \quad i = 2, \dots, n-1 \\ f_p'''(x_n^-) &= 2 \frac{\Delta c_{n-1}}{\Delta x_{n-1}}. \end{aligned}$$

Por lo tanto, en términos de los c_i y los a_i las condiciones i), ii) y iii) del teorema 3.3 se escriben como

$$\begin{aligned} \frac{a_1 - y_1}{(\delta y_1)^2} &= 2 \frac{1-p}{p} \frac{c_1 - c_2}{\Delta x_1} = -2 \frac{(1-p)}{p} \frac{c_2}{\Delta x_1}, \\ \frac{a_i - y_i}{(\delta y_i)^2} &= 2 \frac{1-p}{p} \left[\frac{c_i - c_{i-1}}{\Delta x_{i-1}} - \frac{c_{i+1} - c_i}{\Delta x_i} \right], \\ &= -2 \frac{1-p}{p} \left[\frac{1}{\Delta x_{i-1}} c_{i-1} - \left(\frac{1}{\Delta x_{i-1}} + \frac{1}{\Delta x_i} \right) c_i + \frac{1}{\Delta x_i} c_{i+1} \right], \quad (3.4.10) \\ &\quad i = 2, \dots, n-1, \\ \frac{a_n - y_n}{(\delta y_n)^2} &= 2 \frac{1-p}{p} \frac{c_n - c_{n-1}}{\Delta x_{n-1}} = -2 \frac{(1-p)}{p} \frac{c_{n-1}}{\Delta x_{n-1}}. \end{aligned}$$

En forma matricial las ecuaciones (3.4.10) se pueden escribir como

$$D^{-2}(a - y) = -2 \frac{(1-p)}{p} Qc \quad (3.4.11)$$

donde D^{-2} es una matriz diagonal de orden n cuyos elementos de la diagonal son $(1/(\delta y_1)^2, \dots, 1/(\delta y_n)^2)$ y $(a - y) = (a_1 - y_1, a_2 - y_2, \dots, a_n - y_n)^t$.

Ahora de (3.4.11) tenemos que

$$a = y - 2 \frac{1-p}{p} D^2 Q c \quad \text{donde} \quad D^2 = [D^{-2}]^{-1}. \quad (3.4.12)$$

Multiplicando por $3Q^t$ y utilizando entonces (3.4.8) llegamos a que

$$3Q^t a = 3Q^t y - 6 \frac{1-p}{p} Q^t D^2 Q c = R c.$$

Luego

$$\left(6 \frac{(1-p)}{p} Q^t D^2 Q + R \right) c = 3Q^t y, \quad (3.4.13)$$

o

$$\left(6(1-p)Q^t D^2 Q + pR \right) u = Q^t y \quad (3.4.14)$$

donde $u = c/3p$.

Por otro lado, en términos de la matriz D^{-2} y el vector $(y - a)$, $S(f_p)$ se expresa como

$$S(f_p) = (y - a)^t D^{-2} (y - a)$$

y de (3.4.11) obtenemos entonces que

$$\begin{aligned} S(f_p) &= (y - a)^t 2 \frac{1-p}{p} Q c \\ &= 2 \frac{1-p}{p} (D^2 Q c)^t 2 \frac{1-p}{p} Q c \\ &= \left[2 \frac{1-p}{p} \right]^2 \|DQc\|_2^2, \end{aligned} \quad (3.4.15)$$

donde $\|DQc\|_2^2 = \sum_1^n [(DQc)_i]^2$ y $(DQc)_i$ es la i ésima componente del vector DQc . La matriz del sistema de ecuaciones (3.4.13) es pentadiagonal, simétrica y definida positiva. Por lo tanto es no singular y el sistema se puede resolver de manera eficiente por el método de Cholesky que se describe brevemente en el Anexo.

En resumen, dado un valor fijo de $p \in (0, 1)$ el spline cúbico natural f_p que minimiza el funcional Δ_p se construye resolviendo el sistema (3.4.13) para obtener el vector u y sustituyendo en la expresión (3.4.3) del spline, los coeficientes a_i y c_i obtenidos mediante (3.4.12) y (3.4.14) respectivamente .

III.5. SELECCION DE PARAMETROS DE SUAVIZAMIENTO

Una vez que ya sabemos resolver el problema *II* para un p fijo, de acuerdo con el teorema 3.4 para hallar la solución del problema *I*, solo nos resta calcular una raíz p_σ de la ecuación (3.3.9) y resolver el problema *II* con $p = p_\sigma$. En esta sección vamos a probar que $S(f_p)$ es una función estrictamente decreciente de parámetro p . Por lo tanto, en relación con la solución de la ecuación (3.3.9) solo hay dos posibilidades, o $S(f_p)$ corta la recta $y = \sigma$ en un solo punto $S(f_{p_\sigma})$ o $S(f_p) \leq \sigma$ para todo p .

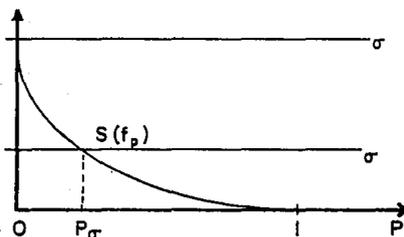


fig 3.2 $S(f_p)$ corta la recta $y = \sigma$
en un solo punto o no la corta.

En el primer caso el teorema 3.4 nos asegura que la solución óptima del *II* con $p = p_\sigma$ es la solución óptima de *I*. En el segundo caso resulta evidente que la solución del problema *I* es la recta f_0 que mejor ajusta los datos (o sea, la solución del problema *II* para $p = 0$), ya que $I(f_0) = 0$ mientras que $I(f) \geq 0$ para toda f .

Teorema 3.5

Sea $S(f_p)$ definido como en (3.2.1) donde f_p es la solución óptima del problema *II*. Entonces, como función de p , $S(f_p)$ es estrictamente decreciente para $p \in (0, 1)$, de modo que $S(f_{1-}) = 0$.

Demostración.

Vamos a probar que $\frac{d}{d_p} S(f_p) < 0$ para todo $p \in (0, 1)$. De (3.2.1) se obtiene directamente que

$$\frac{d}{d_p} S(f_p) = 2 \sum_1^n \left(\frac{f_p(x_i) - y_i}{\delta y_i} \right) \frac{\frac{d}{d_p} f_p(x_i)}{\delta y_i}. \quad 3.5.1)$$

Por otro lado, vamos a calcular $\int_{x_1}^{x_n} \left[\frac{d}{d_p} f_p''(x) \right]^2 dx$ integrando 2 veces por partes

Como $f_p''(x_1^+) = f_p''(x_n^-) = 0$ de la primera integración obtenemos que

$$\begin{aligned} \int_{x_1}^{x_n} \left(\frac{d}{d_p} f_p''(x) \right)^2 dx &= - \int_{x_1}^{x_n} \frac{d}{d_p} f_p'(x) \frac{d}{d_p} f_p'''(x) dx \\ &= - \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} \frac{d}{d_p} f_p'(x) \frac{d}{d_p} f_p'''(x) dx, \end{aligned}$$

Teniendo en cuenta además que $f_p^{IV} \equiv 0$ de la segunda integración llegamos a que

$$\begin{aligned} \int_{x_1}^{x_n} \left(\frac{d}{d_p} f_p''(x) \right)^2 dx &= - \sum_{i=1}^{n-1} \frac{d}{d_p} f_p(x) \frac{d}{d_p} f_p'''(x) \Big|_{x_i}^{x_{i+1}} \\ &= - \left[\left(\frac{d}{d_p} f_p(x_2) \frac{d}{d_p} f_p'''(x_2^-) - \frac{d}{d_p} f_p(x_1) \frac{d}{d_p} f_p'''(x_1^+) \right) \right. \\ &\quad + \left(\frac{d}{d_p} f_p(x_3) \frac{d}{d_p} f_p'''(x_3^-) - \frac{d}{d_p} f_p(x_2) \frac{d}{d_p} f_p'''(x_2^+) \right) \\ &\quad + \dots + \left. \left(\frac{d}{d_p} f_p(x_n) \frac{d}{d_p} f_p'''(x_n^-) - \frac{d}{d_p} f_p(x_{n-1}) \frac{d}{d_p} f_p'''(x_{n-1}^+) \right) \right] \\ &= - \left[- \frac{d}{d_p} f_p(x_1) \frac{d}{d_p} f_p'''(x_1^+) + \frac{d}{d_p} f_p(x_2) \left[\frac{d}{d_p} f_p'''(x_2^-) \right. \right. \\ &\quad \left. \left. - \frac{d}{d_p} f_p'''(x_2^+) \right] + \dots + \frac{d}{d_p} f_p(x_{n-1}) \left[\frac{d}{d_p} f_p'''(x_{n-1}^-) \right. \right. \\ &\quad \left. \left. - \frac{d}{d_p} f_p'''(x_{n-1}^+) \right] + \frac{d}{d_p} f_p(x_n) \frac{d}{d_p} f_p'''(x_n^-) \right], \end{aligned}$$

luego

$$\int_{x_1}^{x_n} \left(\frac{d}{d_p} f_p''(x) \right)^2 dx = - \left[- \frac{d}{d_p} f_p(x_1) \frac{d}{d_p} f_p'''(x_1^+) \right. \\ \left. + \sum_{i=2}^{n-1} \frac{d}{d_p} f_p(x_i) \left(\frac{d}{d_p} f_p'''(x_i^-) - f_p'''(x_i^+) \right) \right. \\ \left. + \frac{d}{d_p} f_p(x_n) \frac{d}{d_p} f_p'''(x_n^-) \right]. \quad (3.5.2)$$

Pero del teorema 3.3 sabemos cuanto valen los saltos de f_p''' en los puntos x_i , de modo que derivando en la condición ii) obtenemos que

$$\frac{d}{d_p} \left(\frac{d}{d_p} f_p'''(x_i^-) - f_p'''(x_i^+) \right) = \frac{1}{(1-p)^2} \frac{f_p(x_i) - y_i}{(\delta y_i)^2} + \frac{p}{1-p} \frac{\frac{d}{d_x} f_p(x_i)}{(\delta y_i)^2} \quad i = 2, \dots, n-1 \quad (3.5.3)$$

Similarmente, de las condiciones i) y ii) llegamos a que

$$-\frac{d}{d_p} f_p'''(x_1^+) = \frac{1}{(1-p)^2} \frac{f_p(x_1) - y_1}{(\delta y_1)^2} + \frac{p}{1-p} \frac{\frac{d}{d_x} f_p(x_1)}{(\delta y_1)^2}, \\ \frac{d}{d_p} f_p'''(x_n^-) = \frac{1}{(1-p)^2} \frac{f_p(x_n) - y_n}{(\delta y_n)^2} + \frac{p}{1-p} \frac{\frac{d}{d_x} f_p(x_n)}{(\delta y_n)^2}. \quad (3.5.4)$$

Sustituyendo entonces (3.5.3) y (3.5.4) en (3.5.2) resulta que

$$\int_{x_1}^{x_n} \left(\frac{d}{d_p} f_p''(x) \right)^2 dx = - \sum_1^n \frac{d}{d_p} f_p(x_i) \left[\frac{1}{(1-p)^2} \frac{f_p(x_i) - y_i}{(\delta y_i)^2} + \frac{p}{1-p} \frac{\frac{d}{d_x} f_p(x_i)}{(\delta y_i)^2} \right] \\ = - \frac{1}{(1-p)^2} \sum_1^n \frac{d}{d_p} f_p(x_i) \left[\frac{f_p(x_i) - y_i}{(\delta y_i)^2} \right] \\ - \frac{p}{1-p} \sum_{i=1}^n \left[\frac{\frac{d}{d_x} f_p(x_i)}{\delta y_i} \right]^2. \quad (3.5.5)$$

Teniendo en cuenta además (3.5.1), de (3.5.5) se obtiene que

$$\int_{x_1}^{x_n} \left(\frac{d}{d_p} f_p''(x) \right)^2 dx = - \frac{1}{2(1-p)^2} \frac{d}{d_p} S(f_p) - \frac{p}{(1-p)} \sum_{i=1}^n \left[\frac{\frac{d}{d_x} f_p(x_i)}{\delta y_i} \right]^2.$$

Por lo tanto

$$-\frac{1}{2(1-p)^2} \frac{d}{dp} S(f_p) = \int_{x_1}^{x_n} \left(\frac{d}{dp} f_p''(x) \right)^2 dx + \frac{p}{(1-p)} \sum_{i=1}^n \left[\frac{d}{dp} \frac{f_p(x_i)}{\delta y_i} \right]^2 > 0,$$

luego $\frac{1}{2(1-p)^2} \frac{d}{dp} S(f_p) \leq 0$, es decir $\frac{d}{dp} S(f_p) \leq 0$.

Por último $S(f_1^-) = 0$ pues como ya habíamos señalado si $p = 1$ la solución del problema II es el spline cúbico natural que interpola los datos. ■

El hecho de que $S(f_p)$ sea una función decreciente del parámetro p solo nos asegura que si la ecuación (3.3.9) tiene solución entonces es única, sino que nos garantiza la convergencia del método de las secantes que se puede utilizar para resolverlae.

Sea $G(p) := S(f_p) - \sigma$. Entonces la ecuación (3.3.9) se puede escribir como

$$G(p) = 0$$

partiendo de $p_0 = 0$ el método de las secantes calcula p_{i+1} como el punto donde la recta tangente a $G(p)$ en p_i se anula. En particular

$$p_1 = p_0 - \frac{G(p_0)}{G'(p_0)} \quad (3.5.6)$$

En otras palabras, a partir de p_0 y p_1 el método de las secantes calcula p_{i+1} como el punto donde la secante a $S(f_p)$ en p_{i-1} y p_i toma el valor σ . Luego

$$p_{i+1} = p_i - \left(\frac{p_i - p_{i-1}}{S(f_{p_i}) - S(f_{p_{i-1}})} \right) (S(f_{p_i}) - \sigma) \quad (3.5.7)$$

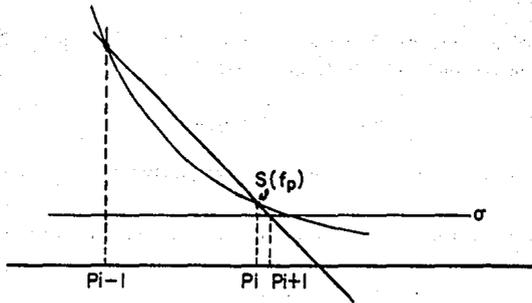


fig 3.3 Método de las secantes para hallar la raíz de $S(f_p)$.

En vez de (3.5.6) se puede tomar

$$p_1 = \frac{\sigma - S(f_0)}{Df_0} \quad (3.5.8)$$

donde $Df_0 = -24U^tRu$ y u es la solución de (2.4.13) con $p = 0$

Nótese que Df_0 es una buena aproximación de

$$\left. \frac{d}{dp} S(f_p) \right|_{p=0} = -24u^tRu + 60(DQu)^t(DQu).$$

Como $S(f_p)$ es una función decreciente de p , entonces $p_{i+1} > p_i$ y en consecuencia

$$S(f_{p_{i+1}}) < S(f_{p_i}),$$

de modo que en un número finito de pasos se llega a un i^* tal que

$$S(f_{p_{i^*}}) \leq \sigma.$$

Si en algún paso i ocurriese que $p_{i+1} \geq 1$, entonces se recomienda aproximar $S(f_{p_i})$ por la parábola

$$y(p) = S(f_{p_i}) \left(\frac{1-p}{1-p_i} \right)^2,$$

que al igual que $S(f_p)$ vale 0 cuando $p = 1$ y vale $S(f_{p_i})$ cuando $p = p_i$. En tal caso p_{i+1} es el punto donde la parábola toma el valor σ , o sea

$$p_{i+1} = 1 - \sqrt{1/S(f_{p_i})(1 - p_i)}$$

y consideramos el signo negativo en la raíz para garantizar que $p_{i+1} < 1$.

Una vez que ya sabemos como resolver aproximadamente la ecuación (4.3.9) estamos en condiciones de resumir los pasos que se deben dar para construir el spline cúbico que suaviza los datos (x_i, y_i) , $i = 1, \dots, n$.

III.6. ALGORITMO PARA CONSTRUIR EL SPLINE CUBICO DE SUAVIZAMIENTO

Se supone conocidos los puntos (x_i, y_i) , $i = 1, \dots, n$, las estimaciones δy_i de la varianza de cada y_i y la máxima "distancia" σ (medida en términos de $S(f)$) entre el spline y los datos que el especialista desea.

Paso 1. Verificar si $\sigma = 0$. En caso afirmativo la solución del problema I es la función que minimiza Δp en el problema II con $p = 1$, es decir el spline cúbico natural que interpola los datos. En otro caso ir al paso 2.

Paso 2. Calcular la solución f_0 del problema II con $p = 0$, verificar si $S(f_0) \leq \sigma$. En caso afirmativo la solución del problema I es la recta f_0 que mejor ajusta los datos. De lo contrario ir al paso 3.

Paso 3. Para $i = 1 \dots$ comenzando con p_i dado por (3.5.8).

3.1. Calcular el spline cúbico natural que minimiza Δp , resolviendo el sistema (3.4.13) con $p = p_i$ y sustituyendo en (3.4.3) los coeficientes a_i y c_i del spline obtenido mediante (3.4.12) y (3.4.14) respectivamente.

3.2. Verificar si $S(f_{p_i}) \leq \sigma$.

En caso afirmativo terminamos, es decir f_{p_i} es la función más "suave" entre todas las que están a una "distancia" menor que σ de los datos.

De lo contrario calcular p_{i+1} por el método de las secantes (3.5.7) y regresar a 3.1 con $i = i + 1$.

Por último, es importante señalar que si el especialista no dispone de información para seleccionar la cota σ entonces existen varias opciones. Si los δy_i constituyen una buena estimación de la varianza de los y_i entonces σ se puede tomar entre $\sqrt{2n}$ y n como propone Reinch, Craven y Wahba consideran una técnica mucho más sofisticada para la selección de σ , que se basa en una estimación del ruido de los datos, obtenido mediante un proceso conocido como validación cruzada.

Además en el Apéndice el lector interesado puede encontrar una subrutina llamada SMOOTH que construye el spline cúbico de suavizamiento, a partir de un σ dado.

Ejemplo 2.1.

Supongamos que en el intervalo $[0,1]$ se toman 21 puntos equidistantes, es

decir

$$x_i = (i - 1)/20, \quad i = 1, \dots, 21$$

y consideramos las parejas de datos (x_i, y_i) , $i = 1, \dots, 21$ donde $y_i = 0$ para toda i , excepto para $i = 14$, es decir la pareja $(x_{14}, y_{14}) = (0.65, 1)$.

Supongamos además que los datos tienen un error del 10 % y que por lo tanto

$$\delta y_i = 0.1 \quad \text{para toda } i.$$

Con este ejemplo queremos destacar el papel que juega la selección de la cota σ , en la forma de spline cúbico que suaviza un conjunto de datos. Como en nuestro ejemplo $\delta y_i = 0.1$ para toda i

$$S(f) = 1/0.001 \sum_{i=1}^{21} (f(x_i) - y_i)^2.$$

Por lo tanto, la restricción $S(f) \leq \sigma$ se expresa en este caso de la siguiente forma:

$$\sum_{i=1}^{21} (f(x_i) - y_i)^2 \leq 0.01\sigma.$$

Construyamos el spline cúbico que suaviza los datos con $\sigma = 100$.

Como $\sigma > 0$, de acuerdo con nuestro algoritmo debemos calcular la solución del problema *II* con $p = 0$. Esto significa que tenemos que resolver el sistema (3.4.13) con $p = 0$. La matriz de este sistema es de orden 19. La solución del mismo se halla en el Apéndice y se obtuvo utilizando el programa SBANDP. Además de (3.4.15) llegamos a que

$$S(f_0) = 93.1.$$

Como $S(f_0) < 100$, entonces f_0 es una solución factible del problema *I* y evidentemente la óptima puesto que

$$I(f_0) = 0.$$

Sin embargo, si tomamos $\sigma = 90$, entonces f_0 ya no es solución factible y debemos ir al paso 3 del algoritmo. En este caso se requieren 2 iteraciones para obtener la solución del problema *I* que es la función que minimiza Δ_p con $p = 0.127$.

Utilizando la función SMOOTH que aparece en el capítulo V se calcularon además los splines de suavizamiento para $\sigma = 80, 70$, obteniéndose respectivamente que $p = 0.807$ y $p = 0.9999$.

Como era de esperar, en la medida que σ disminuye, el valor de p aumenta, ya que para un σ grande incluso la recta de ajuste, que corresponde al caso $p = 0$ en el problema II, es una solución factible del problema I y en consecuencia es la óptima. Sin embargo, cuando σ es "pequeño", una solución factible del problema I debe casi interpolar los datos, lo cual corresponde a un valor de p cerca de 1 en el problema II. La figura 3.4 representa los splines de suavizamiento obtenidos para los distintos valores de la cota σ .

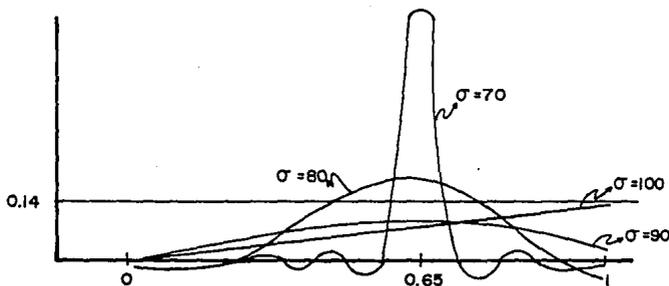


fig 3.4 Spline cúbico de suavizamiento para diferentes valores de σ .

Capítulo 4

Familias

de

Splines

IV. FAMILIAS DE SPLINES

Una pregunta que cualquiera se haría, al terminar de hojear este trabajo, sería, ¿Cómo se puede generalizar el concepto de Spline?, la respuesta se puede dar, tomando en cuenta, los distintos aspectos considerados anteriormente y su posible generalización. Como los splines tratados hasta ahora son funciones formadas por polinomios de cierto grado, uno a continuación de otro, una generalización posible sería la que enseguida mostramos en una gráfica, donde p_1, p_2 , hasta p_n son polinomios de cualquier grado, que forman la función Spline.

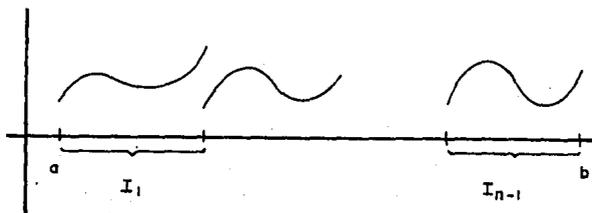


fig. 4.1

Una idea que se refleja en esta gráfica; es que una función Spline: es una función a trozos, en donde cada trozo es un polinomio p_i , y cada polinomio p_i esta evaluado sobre un subintervalo del intervalo $[a, b]$, que denotamos como I_i . Así pues, podemos ver que el polinomio p_1 , esta evaluado sobre el subintervalo I_1 , p_2 sobre el subintervalo I_2 , y así sucesivamente.

Esta función Spline queda determinada por su dominio, contradominio y regla de correspondencia; podemos hablar del concepto de función de clase C^σ , donde σ es un entero no negativo. Si σ toma el valor 0, decimos que la función Spline es Continua; si σ toma el valor 1, estamos hablando de que la función spline es continuamente derivable; si σ es igual a 2 estamos hablando de segunda derivada continua y así sucesivamente. A continuación presentamos varios ejemplos, donde destacamos algunas características importantes de las funciones spline.

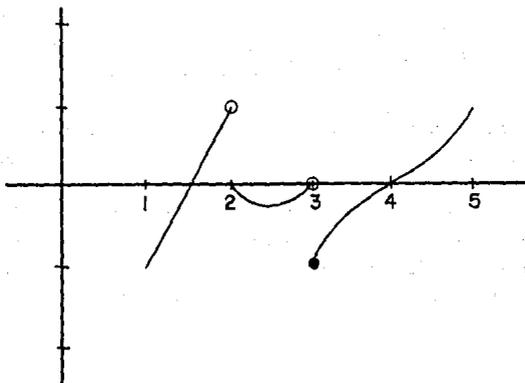


fig. 4.2

Ejemplo 1

Esta es una función Spline, formada por los polinomios

$$s(x) = \begin{cases} p_1(x) = 2x - 3, & \text{si } x \in I_1 = [1, 2) \\ p_2(x) = (x - 2)(x - 3), & \text{si } x \in I_2 = [2, 3) \\ p_3(x) = (x - 4)^3, & \text{si } x \in I_3 = [3, 5]. \end{cases}$$

La figura 4.2 muestra la gráfica del Spline.

Observaciones :

1. I_1 y I_2 son subintervalos cerrados por la izquierda y abiertos por la derecha, y I_3 el último subintervalo es cerrado.
2. El polinomio $p_1(x)$ es lineal, $p_2(x)$ es cuadrático y $p_3(x)$ es cúbico,
3. La función $s(x)$ presenta dos discontinuidades en los puntos de unión 2 y 3.

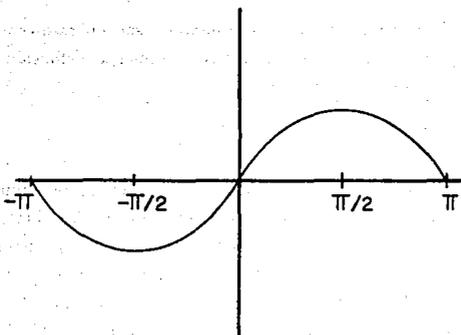


fig. 4.3

Ejemplo 2.

$$s(x) = \begin{cases} -x(x + \pi), & \text{si } x \in [-\pi, 0) \\ x(x - \pi), & \text{si } x \in [0, \pi] \end{cases}$$

La función $s(x)$ es continuamente diferenciable, en su dominio. La continuidad se puede verificar en general, calculando la derivada por la derecha y por la izquierda, y verificando que son iguales. La figura 4.3, muestra la gráfica del spline y obsérvese que es suave en el origen.

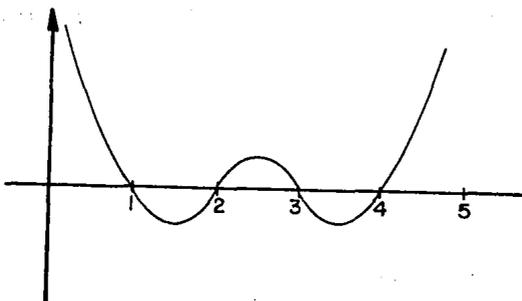


fig. 4.4

Ejemplo 3

$$s(x) = (x-1)(x-2)(x-3)(x-4), \quad x \in [0, 5].$$

Este ejemplo nos muestra que los polinomios son un caso especial de splines (ver figura 4.4), con base a lo anterior, reformularemos, la definición de spline, para poder analizar su estructura algebraica.

IV.1 Definición:

Un spline s es una función formada por una familia p_i de polinomios de grado menor o igual que un entero positivo g , que esta definida sobre el intervalo $I = [a, b]$. El intervalo $I = [a, b]$ es la unión de varios subintervalos I_i , estos subintervalos I_i quedan determinados por una partición del intervalo $[a, b]$, donde los elementos de la partición de $I = [a, b]$, son $\xi_1, \xi_2, \dots, \xi_n$ y $a \leq \xi_1 < \xi_2 < \dots < \xi_n \leq b$; de manera que los subintervalos I_i , estan formados como $I_1 = [\xi_1, \xi_2)$, $I_2 = [\xi_2, \xi_3)$ y así sucesivamente, hasta $I_{n-2} = [\xi_{n-2}, \xi_{n-1})$ y $I_{n-1} = [\xi_{n-1}, \xi_n]$. La familia p_i de polinomios de grado menor igual a g , estan definidos sobre el conjunto de subintervalos I_i . Así que, el spline s es una función que

$$s : I \rightarrow R$$

$$s(x) = p_i(x), \text{ si } x \in I_i, i = 1 \dots n-1$$

y la denotamos como $s = \{I_i, p_i\}$, que indica que es una función, que depende de una familia de polinomios p_i y de un conjunto de subintervalos I_i . También mencionaremos que a los elementos de la partición, reciben el nombre de nodos o puntos de ruptura o puntos de unión de los subintervalos.

IV.2 PROPIEDADES DE LA FAMILIA DE LOS SPLINES

Sea $S = \{s \mid s = \{I_i, p_i\}\}$, la familia de todas las funciones splines y para analizar la estructura algebraica que tiene este conjunto; tomemos dos elementos de S , sean s_1 y s_2 , preguntemos, si la suma $s_1 + s_2$ tiene sentido y si pertenece a S . La suma tiene sentido, sólo si sus intervalos de definición se intersectan, pero aún en este caso, la regla para obtener el spline suma, no es muy fácil, pues la partición del intervalo de definición de s_1 no necesariamente es igual a la partición de s_2 y se vuelve complicado describir la suma. Por lo anterior nos reducimos al caso en que la suma está bien definida y se obtiga facilmente.

Para ello, nos restringimos a aquellos splines, en que la partición $\xi_1, \xi_2, \dots, \xi_n$, del intervalo $I = [a, b]$, se mantiene fija; y llamamos a esta subfamilia de S como

$$S(\vec{\xi}) = \left\{ s \in S \mid s : [\xi_1, \xi_n] \rightarrow R \mid \begin{aligned} s(x) &= p_i(x), \quad x \in [\xi_i, \xi_{i+1}) \quad i = 1, \dots, n-2 \\ & \text{y } s(x) = p_{n-1}(x), \quad x \in [\xi_{n-1}, \xi_n] \end{aligned} \right\}$$

donde $\vec{\xi} = (\xi_1, \xi_2, \dots, \xi_n)$.

Si escogemos dos elementos distintos, Si $s_1, s_2 \in S(\vec{\xi})$; con $s_1 = \{I_i, p_i(x)\}$ y $s_2 = \{I_i, \tilde{p}_i(x)\}$ entonces

1. $s_1 + s_2 = \{I_i, p_i(x) + \tilde{p}_i(x)\}$
2. $\alpha s_1 = \{I_i, \alpha p_i(x)\}$

también pertenecen a $S(\vec{\xi})$. Se puede checar facilmente, que $S(\vec{\xi})$ forma un espacio lineal.

Como $S(\vec{\xi})$ es un espacio lineal muy grande, es decir tiene dimensión infinita debido entre otras razones a que el grado de los polinomios que forman los splines no esta acotado, consideramos a la subfamilia de $S(\vec{\xi})$, formados por los polinomios de grado menor o igual que g , que denotaremos como

$$S(\vec{\xi}, g) = \left\{ s = \{I_i, p_i(x)\} \mid s \in S(\vec{\xi}) \mid p_i \text{ es polinomio de grado menor o igual a } g, \right. \\ \left. i = 1, \dots, n-1 \right\}.$$

Si deseamos que el spline tenga cierto grado de suavidad podemos restringir, aún más éste conjunto, y pedir que las funciones spline sean de clase C^σ en el intervalo I .

A esta subfamilia de $S(\vec{\xi}, g)$ la denotamos como

$$S(\vec{\xi}, g, \sigma) = \left\{ s \in S(\vec{\xi}, g) \mid s \text{ es de clase } C^\sigma \text{ en } \right\}.$$

A continuación presentamos algunos ejemplos de estos subespacios y calcularemos adicionalmente dimensión. Obsérvese que el número de argumentos de $S(\vec{\xi}, g, \sigma)$, se refiere el primero a la partición, el segundo al grado máximo de los polinomios y el último al grado de suavidad. Cuando no aparece algunos de los argumentos, quiere decir que no hay restricciones sobre lo no indicado.

IV.3 DIMENSION DE ALGUNOS ESPACIOS DE SPLINES

Familia de Splines lineales no Continuos

Consideremos, a la subfamilia de $S(\vec{\xi})$ de todos los splines, formados por polinomios de grado menor o igual a 1, y llamemosle

$$S(\vec{\xi}, 1) = \left\{ s = \{I_i, p_i(x)\} \in S(\vec{\xi}) \mid p_i \text{ es polinomio de grado menor o igual a 1,} \right. \\ \left. i = 1, \dots, n-1 \right\}.$$

Enseguida mostraremos un ejemplo gráfico de una función spline $s(x)$ de este conjunto.

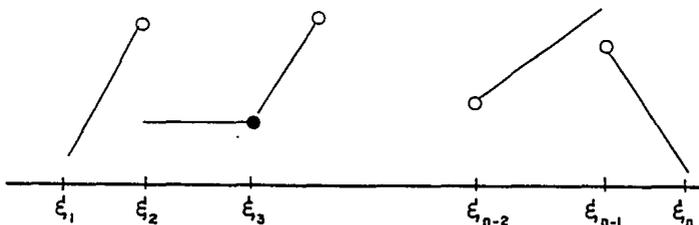


fig. 4.5

Así que en cada subintervalo I_i , se define un polinomio lineal $p_i(x)$. La regla de correspondencia de $s(x)$ es :

$$\begin{aligned} p_1(x) &= a_1x + b_1; & \text{si } x \in I_1 \\ p_2(x) &= a_2x + b_2; & \text{si } x \in I_2 \\ & \vdots \\ p_{n-1}(x) &= a_{n-1}x + b_{n-1}; & \text{si } x \in I_{n-1}. \end{aligned}$$

Veamos ahora, como se puede determinar la dimensión de este subespacio. Dada una función spline $s \in S(\vec{\xi}, 1)$, s esta formada de $n-1$ polinomios de grado menor o

igual a 1, y cada polinomio está determinado, por dos coeficientes. Por lo tanto, para que quede determinado de forma única un elemento $s \in S(\vec{\xi}, 1)$, se necesitan elegir $2(n-1)$ coeficientes, por tanto la $\mathcal{DTM}(S(\vec{\xi}, 1))$ es igual al número de parámetros libres. Así que, la

$$\mathcal{DTM}(S(\vec{\xi}, 1)) = 2(n-1).$$

Ahora consideremos, a las funciones Spline $s \in S(\vec{\xi}, 1)$, que sean continuos en los nodos o puntos de unión de los subintervalos,

$$S(\vec{\xi}, 1, 0) = \left\{ s = \{I_i, p_i(x)\} \in S(\vec{\xi}, 1) \mid s \text{ es continua en } I \right\}.$$

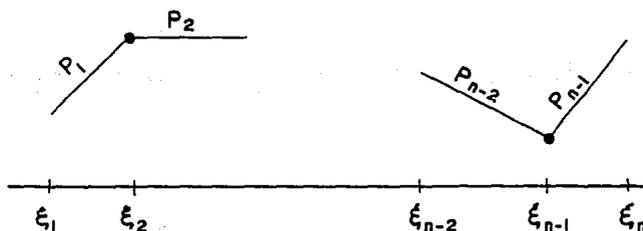


fig. 4.6

Dado un elemento s en $S(\vec{\xi}, 1, 0)$; s es de la forma :

$$s(x) = p_i(x) = a_i x + b_i \quad \text{si } x \in I_i, \quad i = 1, \dots, n-1.$$

Si s es continua en los puntos de unión, s satisface las siguientes condiciones, en los puntos ξ_2, ξ_3 , hasta ξ_{n-1} :

$$a_1 \xi_2 + b_1 = a_2 \xi_2 + b_2$$

$$a_2 \xi_3 + b_2 = a_3 \xi_3 + b_3$$

.....

$$a_{n-2} \xi_{n-1} + b_{n-2} = a_{n-1} \xi_{n-1} + b_{n-1}.$$

Que representa $(n-2)$ condiciones, que s cumple, para que pertenezca a la familia $S(\vec{\xi}, 1, 0)$; así que la dimensión de $S(\vec{\xi}, 1, 0)$ se puede ver como la dimensión de $S(\vec{\xi}, 1, 1)$ menos el número de condiciones, esto se puede formular como :

$$\begin{aligned} \text{DIM}(S(\vec{\xi}, 1, 0)) &= 2(n-1) - (n-2) \\ &= n. \end{aligned}$$

Familia de Splines Cuadráticos

Ahora consideramos a la familia de splines $s \in S(\vec{\xi})$ formados por polinomios de grado menor o igual a 2, y la denotamos

$$S(\bar{\xi}, 2) = \left\{ s_1 = \{I_i, p_i(x)\} \in S(\bar{\xi}) \mid p_i \text{ es polinomio de grado menor o igual a } 2, \right. \\ \left. i = 1, \dots, n-1 \right\}.$$

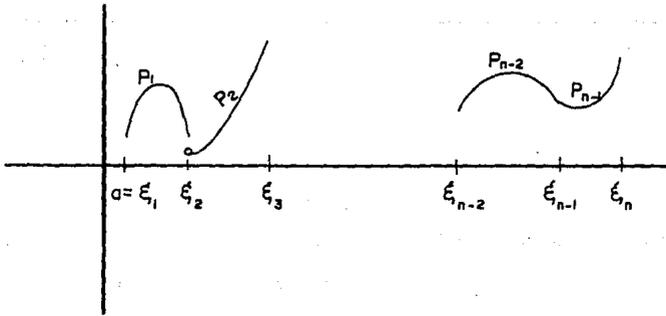


fig. 4.7

Ahora tomemos $s \in S(\bar{\xi}, 2)$; s es de la forma:

$$s(x) = p_i(x) = a_i x^2 + b_i x + c_i, \quad \text{si } x \in I_i$$

para cada $i = 1, \dots, n-1$, para determinar un elemento $s \in S(\bar{\xi}, 2)$ de forma única, se necesita para cada polinomio 3 coeficientes; como son $(n-1)$ polinomios, en total son $3(n-1)$ parámetros libres, la $\text{DIM}(S(\bar{\xi}, 2))$ igual $3(n-1)$.

Ahora consideremos a las funciones spline $s \in S(\bar{\xi}, 2)$ continuas en su intervalo de definición $I = [a, b]$,

$$S(\bar{\xi}, 2, 0) = \left\{ s \in S(\bar{\xi}, 2, 0) \mid s \text{ es continua en } I \right\}.$$

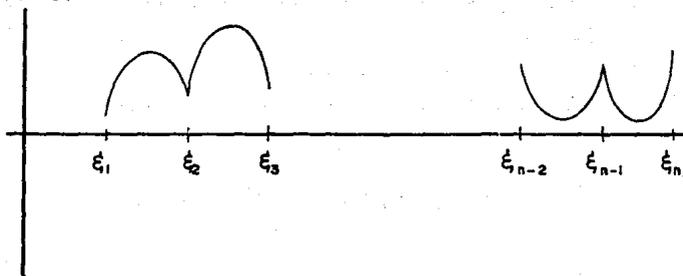


fig. 4.8

como $s \in S(\bar{\xi}, 2)$ es continua en $[a, b] = [\xi_1, \xi_n]$, s satisface:

$$s(x) = p_i(x) = a_i x^2 + b_i x + c_i, \quad \text{si } x \in I_i, \quad i = 1, \dots, n-1$$

y además:

$$a_1 \xi_2^2 + b_1 \xi_2 + c_1 = a_2 \xi_2^2 + b_2 \xi_2 + c_2$$

$$a_2 \xi_3^2 + b_2 \xi_3 + c_2 = a_3 \xi_3^2 + b_3 \xi_3 + c_3$$

.....

$$a_{n-2} \xi_{n-1}^2 + b_{n-2} \xi_{n-1} + c_{n-2} = a_{n-1} \xi_{n-1}^2 + b_{n-1} \xi_{n-1} + c_{n-1}$$

que son en total $n-2$ condiciones, así que la

$$\text{DIM}(S(\bar{\xi}, 2, 0)) = \text{DIM}(S(\bar{\xi}, 2)) - (n-2) = 3(n-1) - (n-2).$$

Aún podemos restringir más a la familia de Splines, $s \in S(\bar{\xi}, 2, 0)$, pidiendo, que s sea derivable en $i = \xi_2, \dots, \xi_{n-1}$ puntos de unión o nodos, y así, s pertenece a la familia $S(\bar{\xi}, 2, 1)$, que es

$$S(\bar{\xi}, 2, 1) = \left\{ s \in S(\bar{\xi}, 2, 0) \mid s \text{ es de clase } C^1, \quad i = 1, \dots, n-1 \right\}.$$

Sea $s \in S(\bar{\xi}, 2, 1)$, s satisfice :

$$1) s(x) = p_i(x) = a_i x^2 + b_i x + c_i, \text{ si } x \in I_i, \quad i = 1, \dots, n-1$$

$$2) p_i(\xi_{i+1}) = p_{i+1}(\xi_{i+1}), \quad i = 1, \dots, n-2$$

$$3) p'_i(\xi_{i+1}) = p'_{i+1}(\xi_{i+1}), \quad i = 1, \dots, n-2$$

que contando las condiciones del inciso 2 y 3, hacen un total de $2(n-2)$.

Así que

$$\begin{aligned} DIM(S(\bar{\xi}, 2, 1)) &= DIM(S(\bar{\xi}, 2) - 2(n-2)) \\ &= 3(n-1) - 2(n-2). \end{aligned}$$

Ahora, si restringimos más, y nos fijamos en la familia de Splines $s \in S(\bar{\xi}, 2, 1)$, tienen segunda derivada;

$$S(\bar{\xi}, 2, 2) = \left\{ s \in S(\bar{\xi}, 2, 1) \mid s \text{ sea de clase } C^2 \text{ en } I \right\}.$$

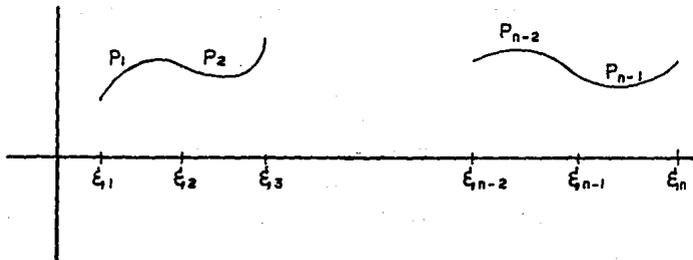


fig. 4.9

Tomemos un elemento $s \in S(\bar{\xi}, 2, 2)$, s debe satisfacer :

$$1) s(x) = p_i(x) = a_i x^2 + b_i x + c_i, \text{ si } x \in I_i, \quad i = 1, \dots, n-1$$

$$2) p_i(\xi_{i+1}) = p_{i+1}(\xi_{i+1}), \quad i = 1, \dots, n-2$$

$$3) p'_i(\xi_{i+1}) = p'_{i+1}(\xi_{i+1}), \quad i = 1, \dots, n-2$$

$$4) p''_i(\xi_{i+1}) = p''_{i+1}(\xi_{i+1}), \quad i = 1, \dots, n-2$$

Todas estas condiciones del inciso 2,3 y 4 suman un total de $3(n-2)$ condiciones. Así

que,

$$\begin{aligned} \mathcal{DLM}(S(\vec{\xi}, 2, 2)) &= \mathcal{DLM}(S(\vec{\xi}, 2)) - 3(n-2) \\ &= 3. \end{aligned}$$

Este resultado refleja, que si, se piden demasiadas condiciones al spline cuadrático en los nodos, resulta que el spline formado por $n-1$ polinomios cuadráticos diferentes, se convierte en un solo polinomio de grado menor o igual a 2.

Familia de Splines formados por polinomios de grado menor o igual a g

Ahora, consideremos, a la familia de Splines formados por polinomios de grado menor o igual que g ,

$$S(\vec{\xi}, g) = \left\{ s = \{I_i, p_i(x)\} \in S(\vec{\xi}) \mid p_i \text{ es polinomio de grado menor o igual a } g, \right. \\ \left. i = 1, \dots, n-1 \right\}.$$

Si tomamos un elemento s en $S(\vec{\xi}, g)$, la regla de correspondencia de s es:

$$s(x) = p_i(x) = a_1^i + a_2^i x + \dots + a_{g+1}^i x^g, \quad \text{si } x \in I_i, \quad i = 1, \dots, n-1$$

Esta expresión representa el polinomio p_i de grado menor o igual que g , evaluado en el intervalo I_i . La dimensión de $S(\vec{\xi}, g)$ es el número de parámetros libres, que se expresa así:

$$\mathcal{DLM}(S(\vec{\xi}, g)) = (g+1)(n-1).$$

Ahora buscamos una expresión adecuada, para generalizar, el caso de la familia de funciones splines formados por polinomios de grado menor o igual a g , y que son de clase C^σ en el intervalo I .

$$S(\vec{\xi}, g, \sigma) = \left\{ s \in S(\vec{\xi}, g) \mid s \text{ es de clase } C^\sigma \text{ en } I \right\}.$$

donde σ es un entero no negativo. Así pues, si σ toma el valor 0, estaremos refiriéndonos a la familia de splines, formados por polinomios de grado menor o

igual a g y que son continuos en el intervalo I . Si σ toma el valor de 1, estaremos refiriéndonos a splines derivables en I , y así sucesivamente. Si σ toma el valor de k , estaremos refiriéndonos a los splines que son k veces diferenciables en I .

En otros términos, si tomamos un elemento s en $S(\vec{\xi}, g, \sigma)$, s satisface las condiciones de continuidad, derivabilidad, y ésto ocurre, en sus puntos de ruptura $i = \xi_2, \dots, \xi_{n-1}$ de s . Estas condiciones de s se expresan matemáticamente como:

$$p_i^K(\xi_{i+1}) = p_{i+1}^K(\xi_{i+1}), \quad i = 1, \dots, n-2, \quad k = 0 \dots \sigma.$$

Si calculamos la suma de las condiciones, tendremos que $(n-2)(\sigma+1)$. Teniendo ésto, podemos saber la dimensión del espacio $S(\vec{\xi}, g, \sigma)$, que está dada por el número de parámetros libres menos el número de condiciones. Así que la

$$DIM(S(\vec{\xi}, g, \sigma)) = (g+1)(n-1) - (\sigma+1)(n-2).$$

Capítulo V

Sistema de Interpolación Splines

< SIS >

V.1. ANALISIS Y DETERMINACION DE REQUERIMIENTOS

Problema a Resolver. Construir tres tipos de aproximación :

- 1) Interplación Cúbica.
- 2) Suavizamiento de Histogramas.
- 3) Suavizamiento y Ajuste.

Objetivo del Problema. Realizar un sistema automatizado que determine las funciones splines dado un conjunto de datos.

Las funciones de aproximación que se construye son :

1) Spline Cúbico de Interpolación para un conjunto de datos $(x_i, y_i) \quad i = 1, \dots, n$ de una función o curva representados en coordenadas polares o cartesianas

2) Función Histospline parabólico para un conjunto de datos que provienen de un histograma de frecuencias o diagramas de barras.

3) Función Spline Cúbico de Ajuste y Suavizamiento para un conjunto de datos $(x_i, y_i) \quad i = 1 \dots n$, para diferentes cotas (' σ '), donde las cotas representan la distancia máxima que se permite entre la aproximación y los datos.

Limitaciones.

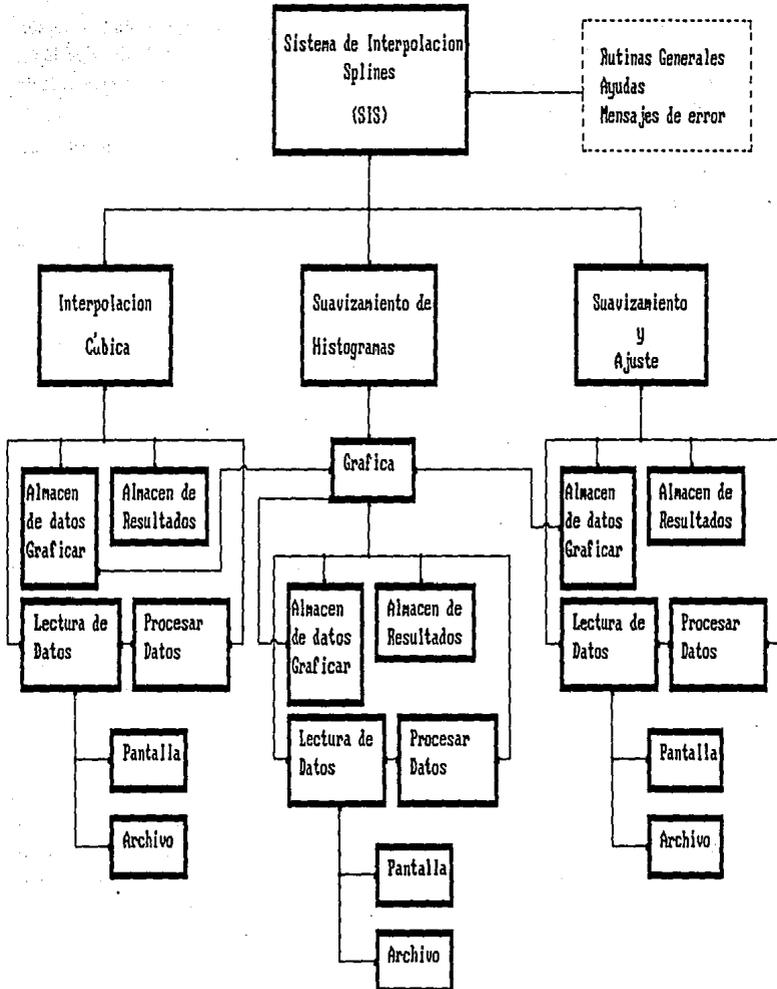
Este sistema carece de un subsistema que realice el ordenamiento de los datos, tanto en su representación cartesiana como polar.

Requerimientos Generales. Se requiere de funciones que realicen:

- 1) La lectura de los datos de entrada, por pantalla o por archivo.
- 2) Construcción analítica de las funciones splines
- 3) Graficación de las funciones splines y los datos
- 4) Almacenar los datos de entrada y resultados
- 5) Crear un subsistema de ayuda para que guíe al usuario de cómo estructurar sus datos de entrada, así como la interpretación de sus resultados.
- 6) Contar con mensajes suficientes para que el usuario detecte fácilmente algún error en la lectura de la información.
- 7) Tener ejemplos de corridas disponibles para facilitar al usuario el uso del sistema.

Dados los requerimientos anteriores, como primer diseño se puede crear la siguiente estructura del sistema.

Modelo Conceptual del Sistema.



V.2. DISEÑO A DETALLE

El primer nivel del Modelo Conceptual del Sistema de Interpolación Splines son programas de control, es decir son programas que generan los menús de selección y llaman a las rutinas necesarias según la opción elegida. El programa principal de este sistema se llama SIS.

SIS muestra unas pantallas de presentación y enseguida un menú de selección llamado Menú Principal que contiene cuatro tipos de opciones:

Opción 1. INFORMACION GENERAL.

En esta opción se da Información General sobre los tópicos que puede resolver el sistema SIS.

Opción 2. TIPO DE APROXIMACION.

Esta opción nos activa otro menú de selección llamado Tipo de Aproximación que nos da cinco opciones (ver el Tipo de Aproximación).

Opción A. AYUDA.

Esta opción nos lleva a un menú de selección llamado Ayuda. Ayuda presenta 10 opciones, cada una presenta una serie de pantallas en donde se orienta al usuario del manejo de sus datos según el tipo de aproximación que valla a calcular.

Opcion S. SALIRSE.

Esta opción es para salirse del sistema de una forma adecuada.

TIPO DE APROXIMACION.

Este menú de selección cuenta con cinco opciones, las tres primeras nos activan módulos de Operación; la opción A nos activa un menú de selección Ayuda y la opción S nos regresa al menú principal.

Los tres módulos de operación son:

Módulo 1. INTERPOLACION CUBICA.

En este módulo construye el Spline Cúbico de Interpolación (parámetrico y/o periódico) de a lo sumo 100 y mínimo 2 datos de una función (periódica o no periódica) o curva (cerrada o abierta); representados en coordenadas polares o cartesianas permitiendo seleccionar una de las siguientes condiciones en la frontera

1) No se proporciona información. (El sistema calcula la información en los extremos).

2) Si se conocen los valores de la primera derivada de g , es decir $g'(x_1) = a$ y $g'(x_n) = b$, pedir que $f'(x_1) = a$ y $f'(x_n) = b$, donde f es la función spline (Se construye Spline Completo de Interpolación y en caso de una curva abierta el Paramétrico).

3) Pedir que en los valores extremos del intervalo $f''(x_1) = 0$ y $f''(x_n) = 0$ (se construye el Spline Natural de Interpolación y en caso de una curva abierta el paramétrico)

4) Cuando los datos se comporten o provengan de una función periódica o curva cerrada se construye el Spline Cúbico Paramétrico y/o Periódico.

Módulo 2. SUAVIZAMIENTO DE HISTOGRAMAS.

En este módulo se construye el histospline parabólico f que suaviza un histograma de n sumo 50 barras. Apartir de la frecuencia absoluta observada en la i -ésima clase, Fa_i , el programa calcula la altura h_i , de esta barra de modo que su area corresponda a la frecuencia relativa de esa clase, es decir:

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

donde $x_1 \leq x_2 \leq \dots \leq x_{n+1}$ son los extremos de cada barra.

Como el histograma de inicio que se construye es un histograma de frecuencias relativas, el histospline que lo suaviza aproxima la función densidad de la variable aleatoria, cuya distribución de frecuencias conoce el usuario. El programa permite seleccionar una de las dos condiciones de frontera siguientes:

- a) El histospline f se anula en los extremos.
- b) La primera derivada de f se anula en los extremos.

Módulo 3. SUAVIZAMIENTO Y AJUSTE DE DATOS.

Este módulo aproxima un conjunto de datos (x_i, y_i) de una función (desconocida) mediante un spline cúbico de suavizamiento. Esta aproximación es recomendable cuando los valores y_i con que se dispone son imprecisos o están afectados por un error δ_i , que es un estimador de la varianza y_i . El spline cúbico de suavizamiento satisface :

a) Entre todas las funciones que se mantienen a una distancia σ de los datos es decir:

$$\sum_{i=1}^n \left[\frac{f(x_i) - y_i}{\delta y_i} \right]^2 \leq \sigma$$

b) Es la más suave en el sentido de que minimiza

$$\int_{x_1}^{x_n} [f''(x)]^2 dx$$

Este módulo ofrece la opción de seleccionar tres valores diferentes de la cota σ , de esta manera se puede comparar (gráficamente) las curvas obtenidas para diferentes niveles de suavidad. Si se elige $\sigma = 0$, se construye el spline cúbico natural; si no se tiene idea de como elegir la cota el programa propone como cota al punto medio del intervalo de Reinsh $(n - \sqrt{2n}, n + \sqrt{2n})$.

En general estos tres módulos cumplen con los requerimientos generales establecidos en la sección anterior.

V.3. DESCRIPCION DETALLADA DE LOS PROCESOS

1. Módulo ejecutivo *SIS*.

Es el programa principal, hace pantallas de presentación, pantallas de Ayuda al usuario, menús primarios y secundarios del sistema, y en general coordina la ejecución de las tareas, llama a los siguientes módulos.

Módulo *UNIDAD*. En este módulo se definen estructuras que sirven de apoyo para el funcionamiento de los menús.

Módulo ejecutivo *CONVIERTE*.

Es un módulo independiente de *SIS*, pero es necesario para la actualización del Help que controla *SIS*.

Módulo ejecutivo *SPLCINT*. Este módulo se encarga de coordinar las tareas de los submódulos para obtener el cálculo de los splines cúbicos; entre las tareas que coordina están : lectura de datos, construcción analítica y gráfica etc.; llama a los siguientes módulos:

1.1 Módulo *LECTUSC*.

Este módulo se encarga de hacer la lectura por pantalla o por archivo de los datos, dados en coordenadas cartesianas o polares, pertenecientes a una función o curva, función (periódica o no periódica), curva (abierta o cerrada), hace la transformación de coordenadas polares a cartesianas.

1.2 Módulo *SPLCUMSC*.

Este módulo se encarga de construir el Spline cúbico de Interpolación, para datos que pertenecen a una función no periódica o curva abierta;

1.3 Módulo *SPLCUPSC*.

Este módulo se encarga de construir el spline cúbico paramétrico y/o periódico para datos que pertenecen a una función periódica o curva cerrada.

1.4 Módulos *PPVALSC* y *INTERVSC*.

Ppvalsc es una función que llama al módulo *Intervsc* para identificar los intervalos a los cuales pertenecen los puntos donde se va a evaluar la función interpolante y realiza la evaluación.

1.5 Módulo GRAVASC.

Este módulo almacena los datos, tanto originales como los que se construyen al evaluar la función spline.

DIAGRAMA 1

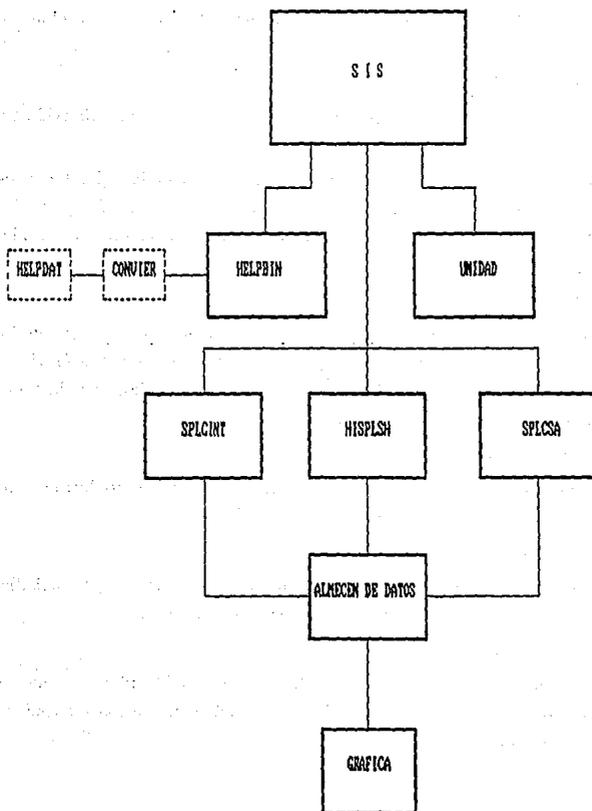
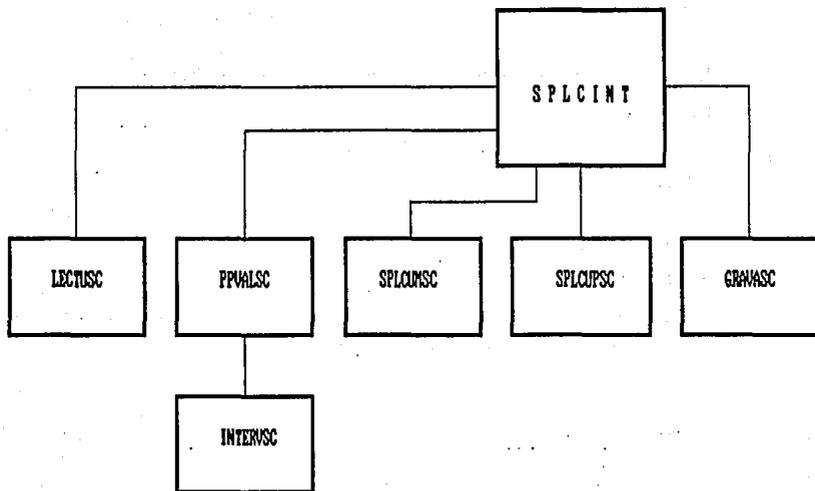


DIAGRAMA 2



2. Módulo ejecutivo *HISPLSH*.

Este módulo coordina tareas de submódulos para la construcción analítica y gráfica del histospline que suaviza un histograma; entre las tareas que realiza ésta la de controlar la lectura de los datos (archivo o pantalla), chequeo contra posible captura de datos errónea etc.

2.1. Módulo *TRIDSH*.

Este módulo resuelve el sistema de ecuaciones con matriz tridiagonal.

2.2. Módulo *PPVALSH*.

Es una función que evalúa el histospline f en un punto x

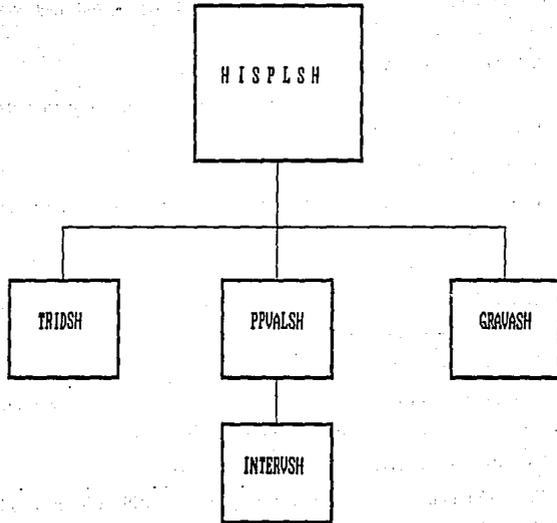
2.3. Módulo *INTERVSH*.

Este módulo es llamado por *Ppvalsh* para calcular los intervalos donde se evaluará la función Histospline.

2.4. Módulo *GRAVASH*.

Este módulo almacena los datos con que se construye el histograma de frecuencias y el histospline respectivamente.

DIAGRAMA 3



3. Módulo ejecutivo *SPLCSA*.

Coordina las tareas de submódulos para la construcción analítica y gráfica del spline de suavizamiento y ajuste para datos de una función o curva, entre las tareas que hace ésta la del control de lectura de datos, parametrización de curvas, preparación de archivos etc.

3.1. Módulo *VARRERSA*.

Este módulo permite seleccionar una variante para la lectura de los errores de los datos (pantalla o archivo).

3.2. Módulo *CORPOLSA*.

Este módulo realiza la lectura por pantalla, de los datos representados en coordenadas polares pertenecientes a una función o curva.

3.3. Módulo *CORCARSA*.

Este módulo realiza la lectura por pantalla de los datos representados en coordenadas cartesianas, pertenecientes a una función o curva.

3.4. Módulo *SMOOTHSA*.

Este módulo construye el spline que Ajusta y Suaviza al conjunto de datos, y coordina la tarea de los módulos *setupqsa* y *choldsa*.

3.5. Módulo *PPVALSA*.

Este módulo llama al módulo *Intervsa* para identificar los intervalos a los cuales pertenecen los puntos donde se va a evaluar la función de Suavizamiento y Ajuste y realiza la evaluación.

3.6. Módulo *SMTERRSA*.

Este módulo calcula el error máximo que hay del spline cúbico de suavizamiento y los datos.

3.7. Módulo *SETUPQSA*.

Este módulo calcula la matriz del sistema de ecuaciones lineales que se forma de definir una función spline cúbica que suavice los puntos $(x_i, y_i) \quad i = 1, \dots, n$.

3.8. Módulo *CHOLISA*.

Este módulo realiza la factorización de Cholesky de la matriz obtenida en *setupqsa* y resuelve el sistema de ecuaciones.

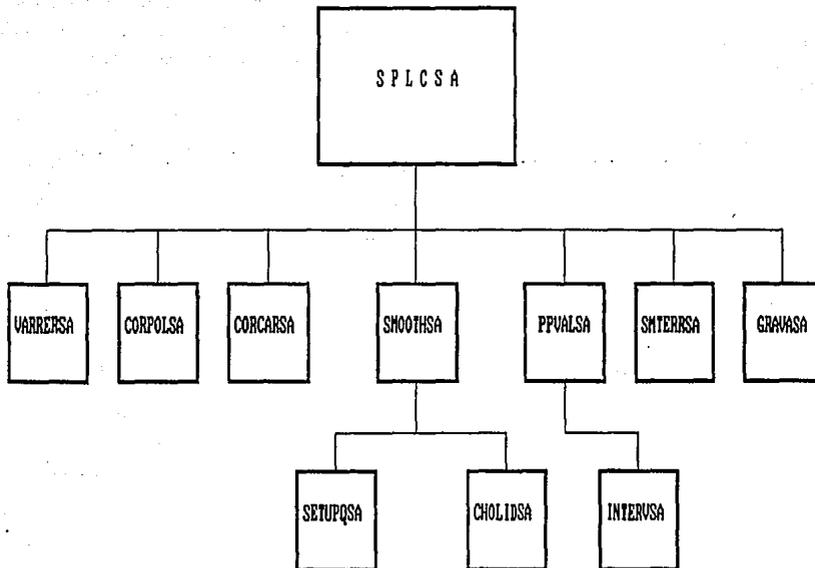
3.9. Módulo *INTERUSA*.

Este módulo recibe de *ppvalsa* los puntos donde se quiere evaluar el spline de suavizamiento e identifica el intervalo al cual pertenece, regresando este dato para sus evaluación.

3.10. Módulo *GRAVASA*.

Este módulo almacena los datos de entrada y los datos que se construyen para evaluar el spline cúbico de suavizamiento (un conjunto para cada cota requerida).

DIAGRAMA 4



Módulo *GRAFICA*.

Este módulo contiene un conjunto de funciones cuya finalidad es graficar funciones, y es llamado por *SPLCINT*, *HISPL*, *SPLCSA*.

La función con que fué realizado este módulo, es para reproducir gráficamente los resultados que en un archivo de datos llamado *DATOS* resulta de correr *SIS* de acuerdo con la especificación de uso que el usuario desee, ésto es, para Interpolación, Ajuste de Datos o el Histospline. *GRAFICA* lee estos datos en un cierto formato y de acuerdo a éste se tiene la especificación correcta, el grafo que desea ver representado en la pantalla. Este módulo consta de las siguientes rutinas *MULTPLT* e *HISTPL*, que a continuación describiremos.

[*MULTPLT*] Esta rutina toma los datos de las coordenadas de los puntos x y y de cada gráfica a representar en la pantalla y de acuerdo al tipo de datos se tiene el resultado para Interpolación y el de Ajuste de datos.

[*HISTPL*] Esta rutina toma los datos de las coordenadas de los puntos x y y de cada gráfica a representar en la pantalla y de acuerdo al tipo de datos se tiene el resultado para representar el Histograma y el Histospline respectivo.

DIAGRAMA 5

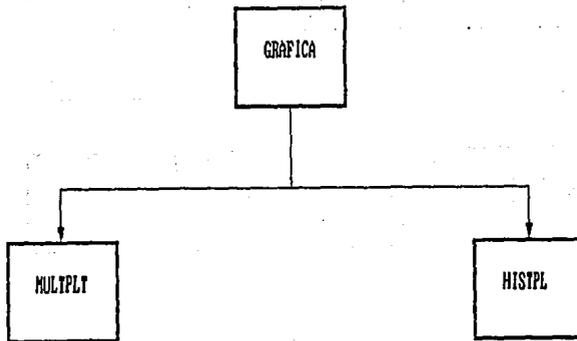


DIAGRAMA 7

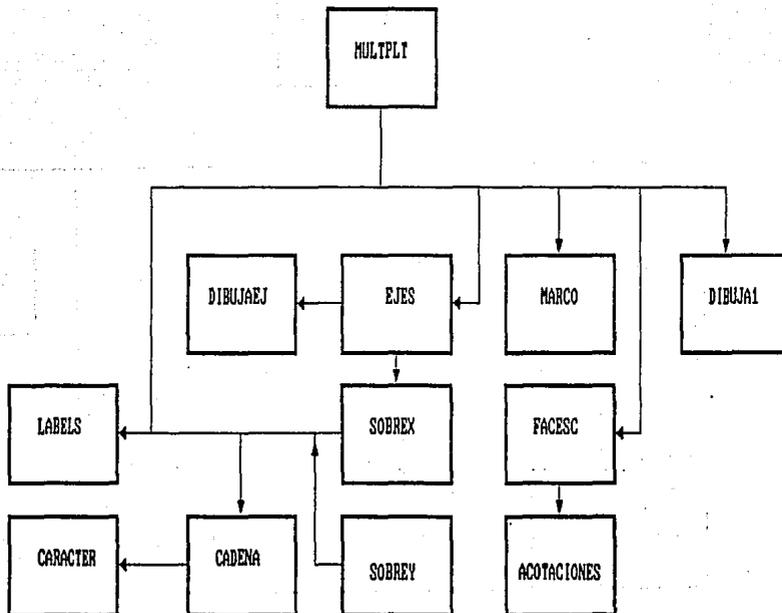


Diagrama de Warnier que apoya al Sistema SIS

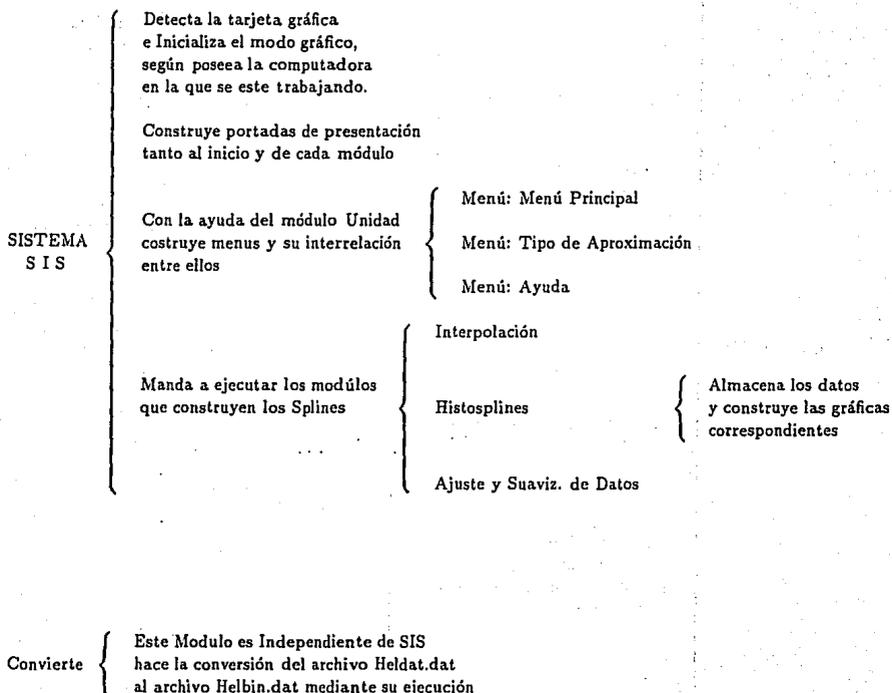


Diagrama de Warnier que apoya al módulo SPLCINT

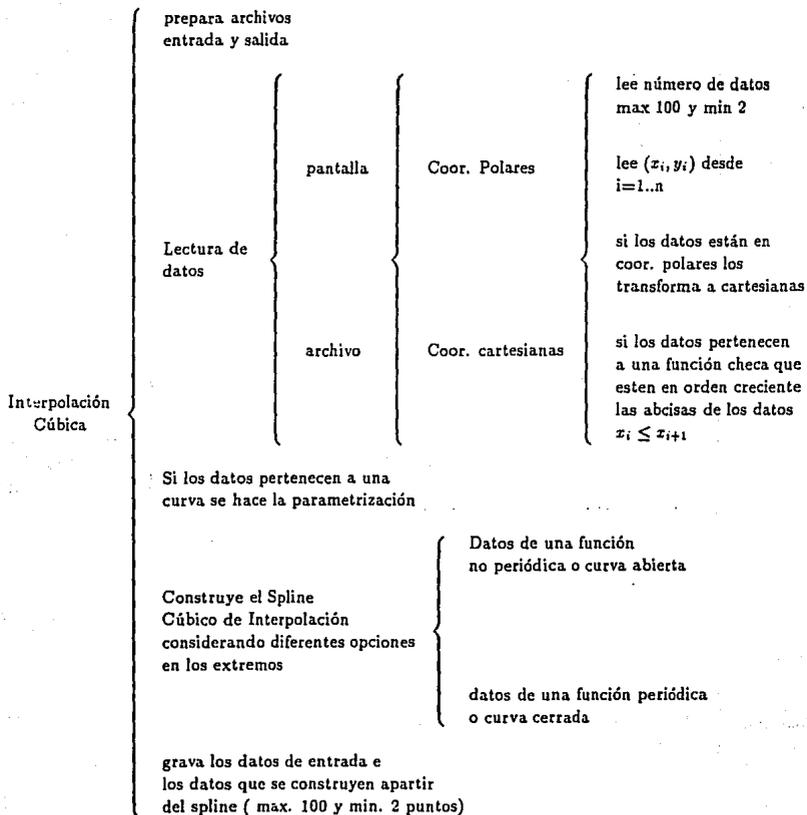


Diagrama de Warnier que apoya al módulo HISPL

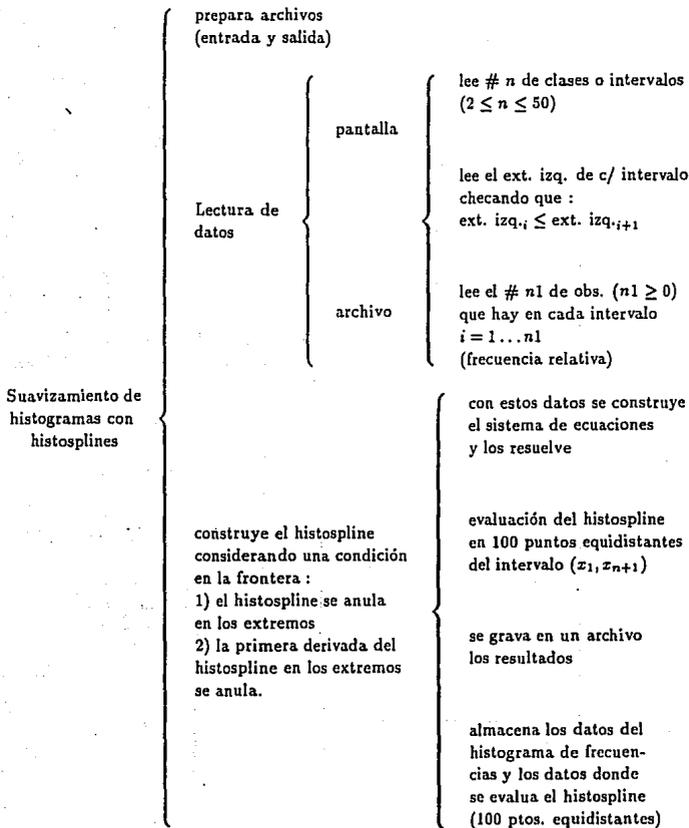


Diagrama de Warnier que apoya al módulo SPLCSA

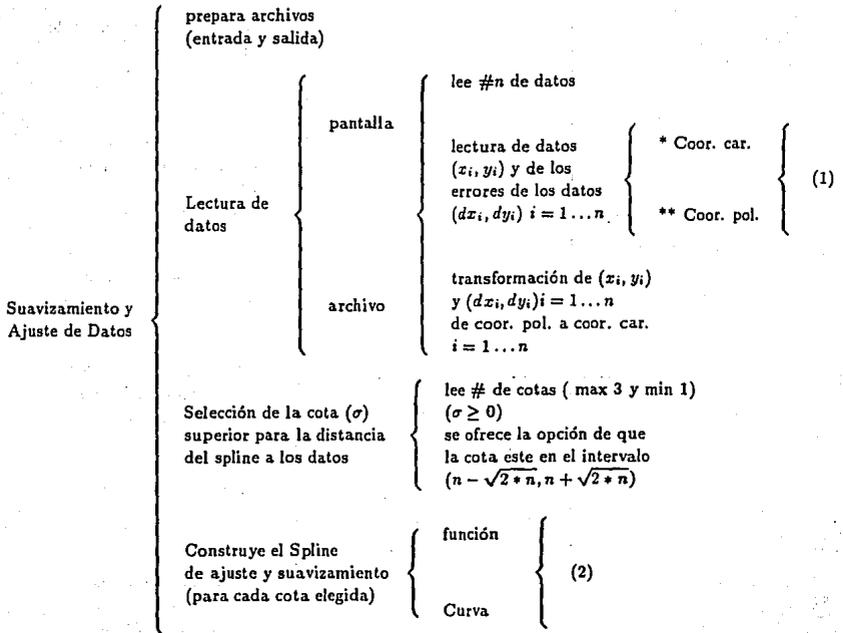


Diagrama de Warnier que apoya al módulo SPLCSA

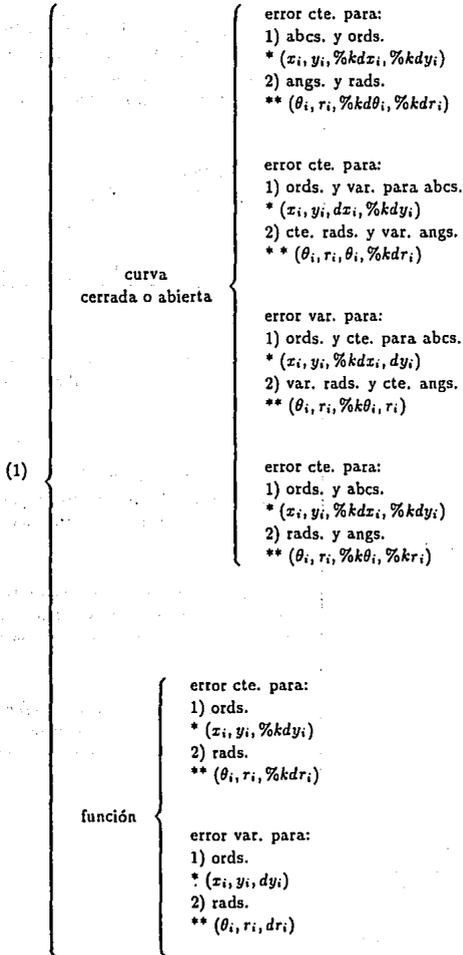
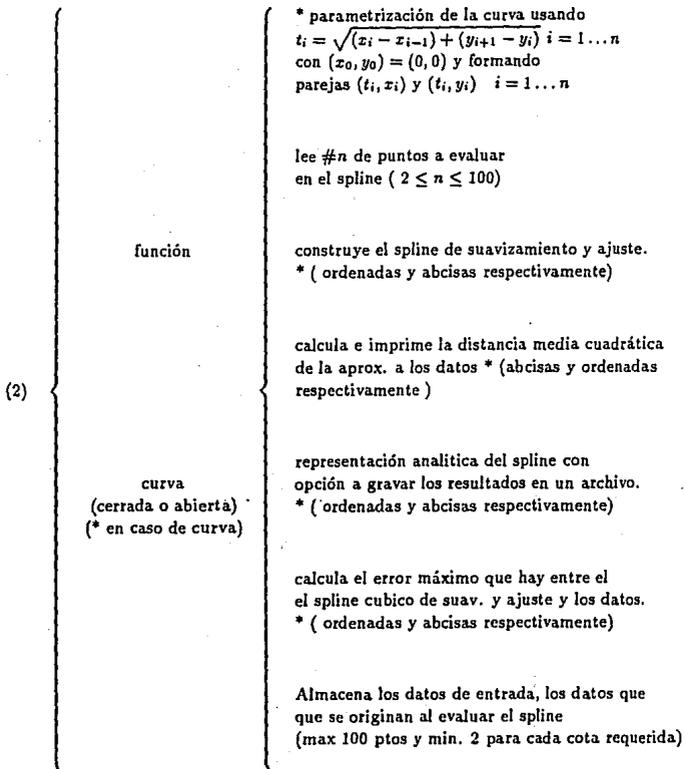


Diagrama de Warnier que apoya al módulo SPLCSA



V.4. SOFTWARE USADO PARA EL DESARROLLO DEL SISTEMA SIS

Considerando la facilidad que proporciona TURBO PASCAL (ver. 6.0) para acceder archivos texto, crear unidades, diseñar pantallas; usando colores, tipos y tamaños de letra etc; en general se usaron las rutinas de su modo gráfico.

Como también este lenguaje tiene la facilidad para llamar programas ejecutables de otros lenguajes, se uso como coordinador de la ejecución de los tres módulos principales del sistema SIS (Interpolación, Suavizamiento de Histogramas, Ajuste y Suavizamiento de Datos), los programas que apoyan a estos estan hechos en FORTRAN 77 y fueron compilados usando el compilador microsoft (ver. 5.0); estos programas corresponden a la implatación de las técnicas y algoritmos presentados en los capítulos I, II, III, IV, y Apéndice.

Para la presentación de las gráficas de las curvas se uso las rutinas de graficación LAHEY GRAPHIC LIBRARY. Los programas fuentes que apoyan al sistema SIS son los siguientes

SIS.PAS	SPLCINT.FOR	HISPLSH.FOR	SPLCSA.FOR
UNIDAD.INC	SPLCUPSC.FOR	GRAVASH.FOR	CHOLIDSA.FOR
CONVIER.PAS	SPLCUMSC.FOR	TRIDSH.FOR	CORCARSA.FOR
HELPTX.BAT	PPVALSC.FOR	INTERVSH.FOR	CORPOLSA.FOR
HELPBIN.DAT	INTERVSC.FOR	PPVALSH.FOR	GRAVASA.FOR
GRAFICA.FOR	GRAVASC.FOR		PPVALSA.FOR
			INTERVSA.FOR
			SETPQSA.FOR
			SMOOTHSA.FOR
			SMTERRSA.FOR
			VARERRSA.FOR

V.5. REQUERIMIENTOS PRELIMINARES PARA LA ACTIVACION DE SISTEMA SIS

El sistema SIS puede correr en cualquier microcomputadora IBM o Compatible con o sin coprocesador 8087 y con cualquier tarjeta gráfica mencionadas más adelante.

Es indispensable disponer de 512k de memoria interna para su ejecución, además es necesario que en el directorio donde se aloje el sistema SIS, se encuentren los siguientes archivos:

GOTH.CHR

LITT.CHR

SANS.CHR

TRIP.CHR

(Los utilizan las rutinas gráficas de Pascal, para elaborar y definir las diferentes tipos y tamaños de letras.)

EGAVGA.BGI, CGA., MCGA, EGA, EGAMONO, HERC, y VGA

(Lo utilizan tanto las rutinas de graficación de Pascal como la biblioteca gráfica de Fortran.)

HELPTEX.DAT

HELPPBIN.DAT

(Archivos que contienen los textos que acompañan al menú AYUDA.)

CONVIER.EXE

(Módulo ejecutable para hacer la conversión del archivo HELP- TEX.DAT (archivo tipo texto) a archivo HELPPBIN.DAT (archivo tipo binario))

SPLCINT.EXE

HISPLSH.EXE

SPLCSA.EXE

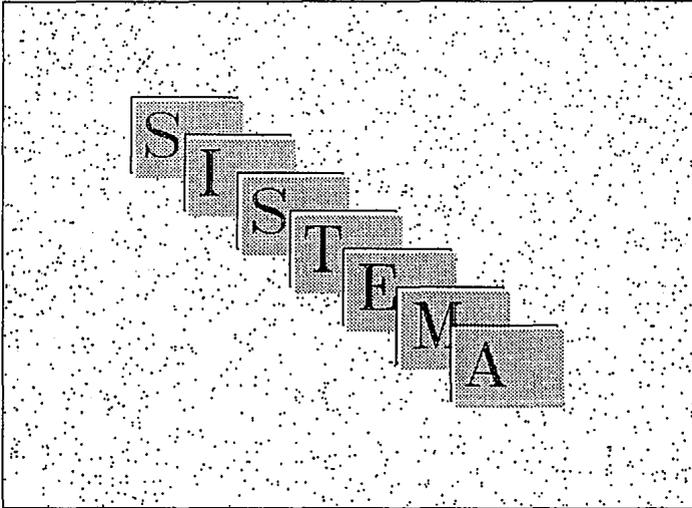
(Archivos llamados por SIS.EXE para el cálculo de los Splines.)

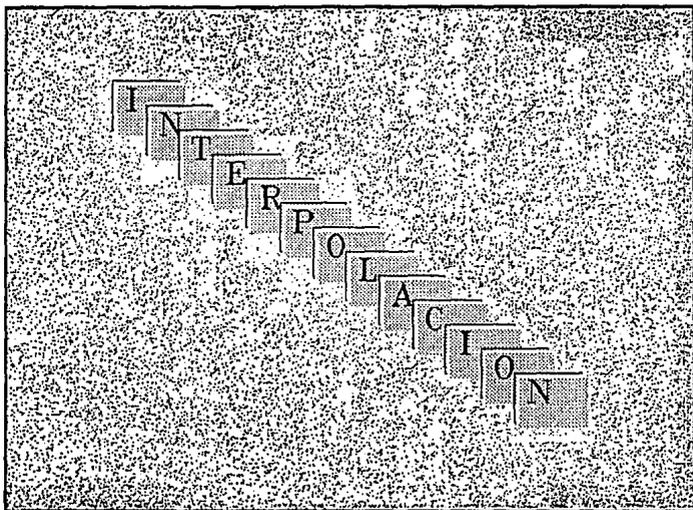
Observación : Cada vez que se modifique HELPTEX.DAT será necesario correr el programa CONVIER.EXE para hacer la conversión.

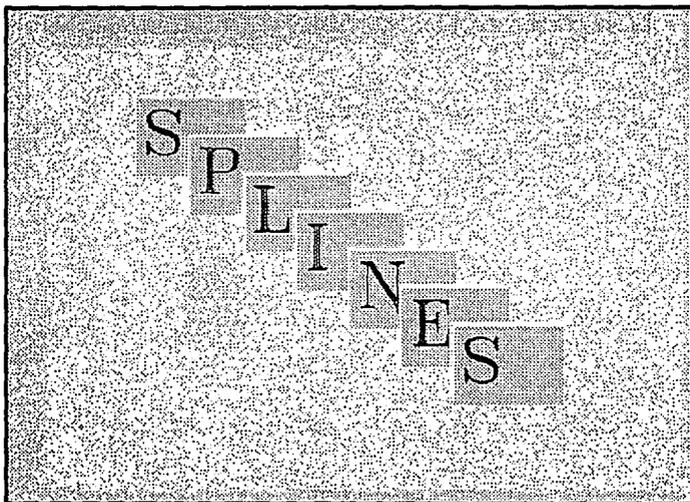
V.6. ACTIVACION Y EJECUCION DEL SISTEMA SIS

El sistema splines es activado por el usuario, cuando este escribe en la pantalla
> SIS

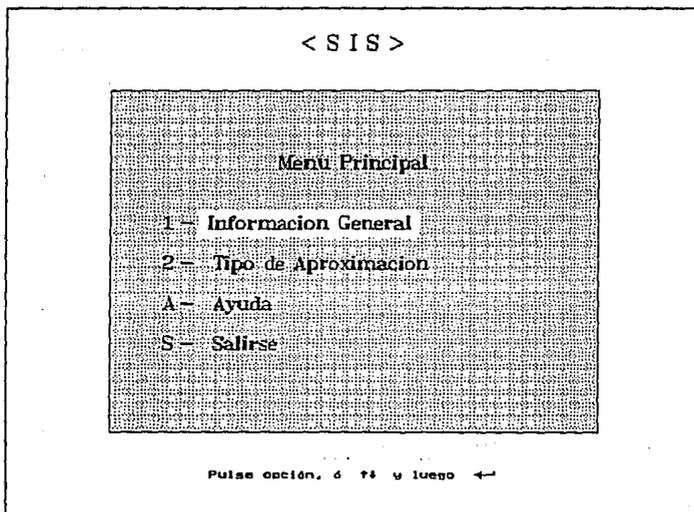
Acción que corresponde a la inicialización del programa ejecutable SIS.EXE. En respuesta a esta activación, se presenta:





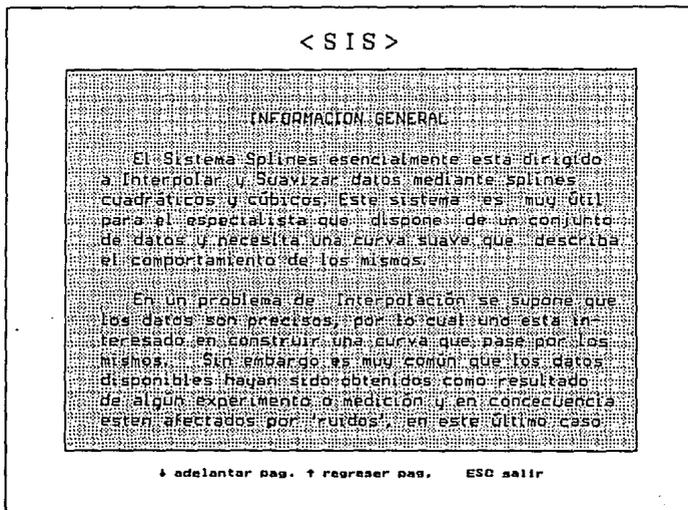


Hasta que en la pantalla aparece el siguiente menú



El usuario puede elegir cualquiera de las opciones que ahí aparece, colocando el cursor en la línea deseada y dando un < ENTER > o bien oprimiendo la tecla que le corresponda a la opción elegida.

Menú Principal. Opción 1. La respuesta a la elección de esta opción es



El usuario puede avanzar a la siguiente pág. apretando la tecla ↓, o regresar con ↑ y para regresar al menú principal apretar ESC.

Menú Principal. Opción 2. La respuesta a la elección de esta opción es el siguiente menú

< S I S >

Tipo de Aproximacion

1 - Interpolacion Cubica

2 - Suavizamiento de Histogramas

3 - Ajuste y Suavizamiento de Datos

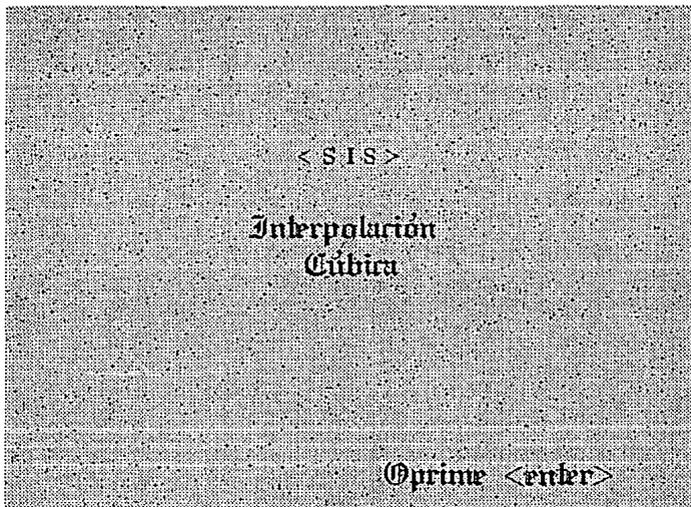
A - Ayuda

S - Menú Principal

Pulse opción, ↓ ↑ y luego ←

El usuario puede elegir cualquiera de las opciones que ahí aparece, colocando el cursor en la línea deseada y dando un `< ENTER >` o bien oprimiendo la tecla que le corresponda a la opción elegida.

Tipo de Aproximación. Opción 1. La respuesta a la elección de esta opción es



En este momento el usuario esta accedando el módulo de Splines Cúbicos de Interpolación, teclear < ENTER > y la respuesta a esta acción es:

.....
 . Cálculo del spline cúbico de interpolación para
 datos de una curva o función.

Desea la entrada de datos por
 Archivo o por pantalla ? [a/p]

a

Diga el nombre del archivo de datos
 dico.dat

Los datos corresponden a una
 función o a una curva ? [f/c]

f

Los datos están en coord. polares
 o cartesianas ? [p/c]

c

Diga el nombre del archivo de resultados
 res

Su curva es cerrada o abierta ? [c/a]

a

En cuanto más desea evaluar el spline?
 (max=100)

100

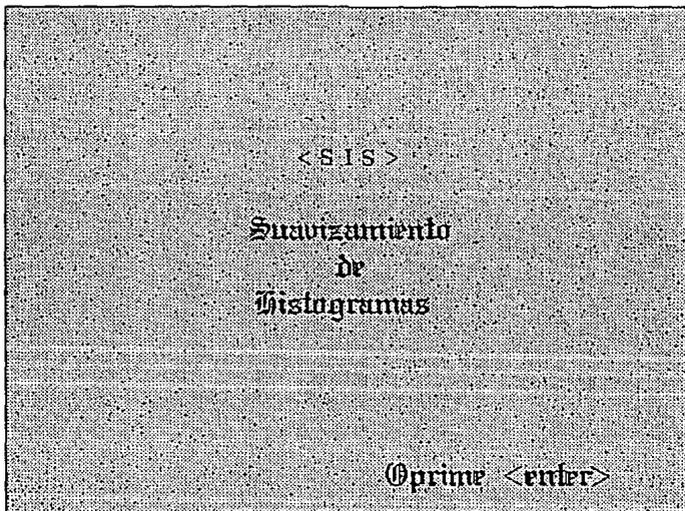
.....
 desea volver a correr el modulo

En el Menú Ayuda que presenta SIS, se explica en forma breve cómo se deberá meter los datos o bien en la sección de Validación y Pruebas del Sistema SIS se dará la corrida de algunos ejemplos.

Para salir de este módulo el usuario deberá teclear < Ctrl + C > para abortar el programa de este módulo, o bien teclear N.

Y enseguida volver a teclear < ENTER >; la respuesta a esta acción es regresar a la pantalla de presentación de Splines Cúbicos de Interpolación, enseguida volver a teclear < ENTER > para regresar al menú Tipo de Aproximación.

Tipo de Aproximación. Opción 2. La respuesta a la elección de esta opción es



En este momento el usuario esta accediendo el módulo de Suavizamiento de Histogramas, teclear < ENTER > y la respuesta a esta acción es:

```
.....  
  .  Calculo del histospline parabolico que  .  
  .  suaviza un histograma                 .  
.....
```

Desde la entrada de datos por
Archivo o por pantalla ? (a/p)

A

Diga el nombre del archivo de datos
NOT.DAT

Escoja una de las sgtes condiciones
en la frontera

1. El histospline se anula en los extremos
2. La 1ra derivada del histospline se anula en los extremos

1

Diga el nombre del archivo de resultados
RES.DAT

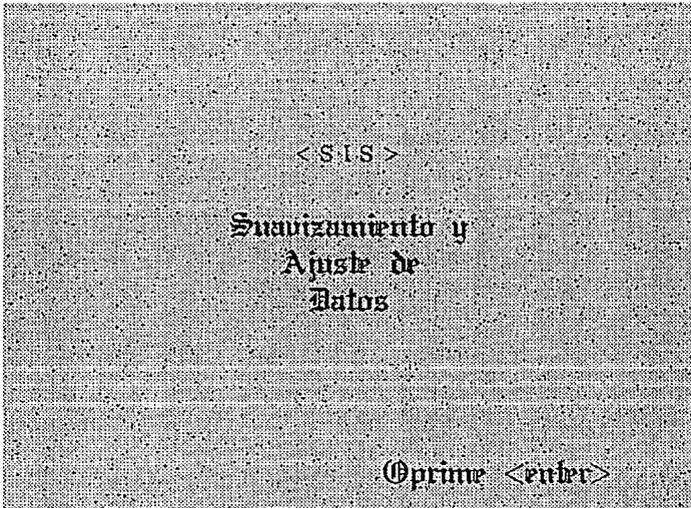
deseas volver a correr el modulo

En el Menú Ayuda que presenta SIS, se explica en forma breve cómo se deberá meter los datos o bien en la sección de Validación y Pruebas del Sistema SIS se dará la corrida de algunos ejemplos.

Para salir de este módulo el usuario deberá teclear < Ctrl + C > para abortar el programa de este módulo, o bien teclear N.

y enseguida volver a teclear < ENTER >; la respuesta a esta acción es regresar a la pantalla de presentación de Suavizamiento de Histogramas, enseguida volver a teclear < ENTER > para regresar al menú Tipo de Aproximación.

Tipo de Aproximación. Opción 3. La respuesta a la elección de esta opción es



En este momento el usuario esta accedendo el módulo de Suavizamiento y Ajuste de datos, teclear < ENTER > y la respuesta a esta acción es:

```

.....
. Suavizamiento y Ajuste .
. de Datos .
.....

```

Desee la entrada de datos por
Archivo o por pantalla ? (a/p)

A

Diga el nombre del archivo de datos
SIN.DAT

Los datos corresponden a una
funcion o a una curva ? (f/c)

F

Los datos están en coord. polares
o cartesianas ? (p/c)

Error :

1. Por fichero.
 2. Por pantalla.
- 2

Escoja una de las sgtes variantes:

1. error variable para las ordenadas
 2. error cte para las ordenadas
- 2

% de error en las
ordenadas (>0)

.001

diga el nombre del fichero de resultados
RES

Información sobre la cota superior para la distancia
media cuadrática de la aprox. a los datos

1. Cuantas cotas va dar (% 3) ?

3

Se recomienda tomar las cotas en
el intervalo [13.675, 26.325]

2. Valores de las cotas ($>= 0$)

13

20

26

En cuantos puntos desea evaluar el spline? (max=100)

100

deseas volver a correr el modulo

En el Menú Ayuda que presenta SIS, se explica en forma breve cómo se deberá meter los datos o bien en la sección de Validación y Pruebas del Sistema SIS se dará la corrida de algunos ejemplos de este modulo.

Para salir de este módulo el usuario deberá teclear < Ctrl + C > para abortar el programa de este módulo, o bien teclear N.

y enseguida volver a teclear enter; la respuesta a esta acción es regresar a la pantalla de presentación de Ajuste y Suavizamiento de Datos , enseguida volver a teclear < ENTER > para regresar al menú Tipo de Aproximación.

Tipo de Aproximación. Opción A. Si el usuario elige esta opción se desplegará el siguiente menú Ayuda :

< S I S >

AYUDA			
	Sátines Cúbicos de Int.	Suaviz. de un Histog.	Ajuste u Suaviz. de datos
Breve Información.	1	2	3
Lectura de Datos.	4	5	6
Resultados Obtenidos.	7	8	9
Regresar	S		

Pulse opción, ó ↑, →, y luego ←

El usuario puede elegir cualquiera de las opciones que ahí aparece, colocando el cursor en la línea deseada y dando un < ENTER > o bien oprimiendo la tecla que le corresponda a la opción elegida. por ejemplo si elige la opción 4 se presentará :

< S I S >			
AYUDA			
	Solines Cubicos de Int.	Suaviz. de Un Histon.	Ajuste u suaviz. de datos
Breve Información:	1	2	3
Lectura de Datos:	4	5	6
Resultados Obtenidos:	7	8	9
Regresar:	S		

Pulse opción, ó ↑↓, ←→, y luego ←

< SIS >

UNIDADES DE LECTURA

Los datos deberán leerse desde pantalla o archivo:

si es por pantalla teclear la letra D

si es por archivo teclear la letra F

si los datos vienen en archivo teclear su nombre correcto y enseguida su extensión:

<nomarch>.dat

TIPO DE DATOS

Teclear el número de datos (máximo 100). Los

↑ adelantar pag. ↓ regresar pag. ESC salir

El usuario puede avanzar a la siguiente pág. apretando la tecla ↓, o regresar con ↑ y para regresar al menú anterior apretar ESC. Este menú tiene como finalidad apoyar y facilitar al usuario del manejo de sus datos dependiendo del tipo de aproximación. Este menú se puede acceder desde el Menú Principal. Opción A.

Tipo de Aproximación. Opción S. La respuesta a esta opción es desplegar el menú principal.

Menú Principal. Opción A. si se elige esta opción se desplegará el menú Ayuda que mencionamos en Tipo de Aproximación. Opción A.

Menú Principal. Opción S. Es para salirse del sistema SIS de una forma adecuada.

V.7. VALIDACION Y PRUEBAS DEL SISTEMA S I S

El objetivo de esta sección tiene doble propósito

- 1) La validación de SIS
- 2) Mostrar al usuario el manejo de SIS

La validación correspondiente a la manipulación de las pantallas de SIS se mostraron en la sección anterior. Esta parte esta enfocada especialmente a los 3 módulos :

- 1) Interpolación Cúbica.
- 2) Suavizamiento de Histogramas.
- 3) Suavizamiento y Ajuste de Datos.

Se probaron todas las posibles opciones y se muestra en los siguientes ejemplos algunas de ellas, se anexa a este sistema un disket que contiene archivos de datos que se utilizaron para hacer las pruebas del sistema.

INTERPOLACION CUBICA

CORRIDA 1

-
 . Cálculo del spline cúbico de interpolación para .
 . datos de una curva o función.

Desea la entrada de datos por

Archivo o por pantalla ? (a/p)

a

Diga el nombre del archivo de datos

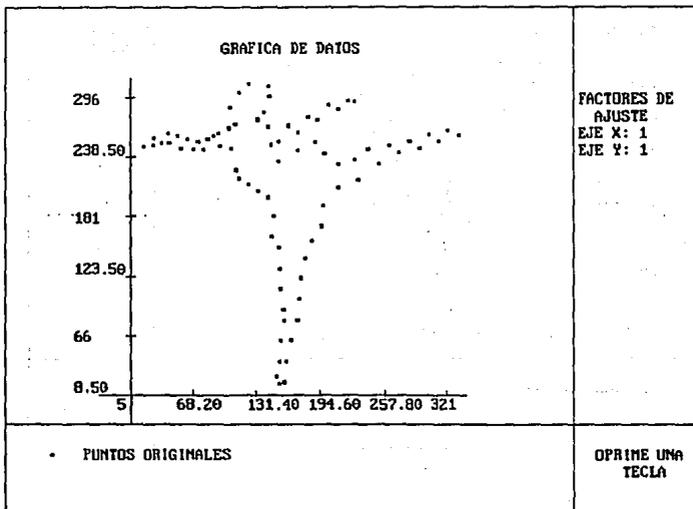
ballarin.dat

Los datos corresponden a una
 funcion o a una curva ? (f/c)

c

Los datos están en coord. polares
 o cartesianas ? (p/c)

c



CONTINUACION.

digite el nombre del archivo de resultados
baillar.in.res

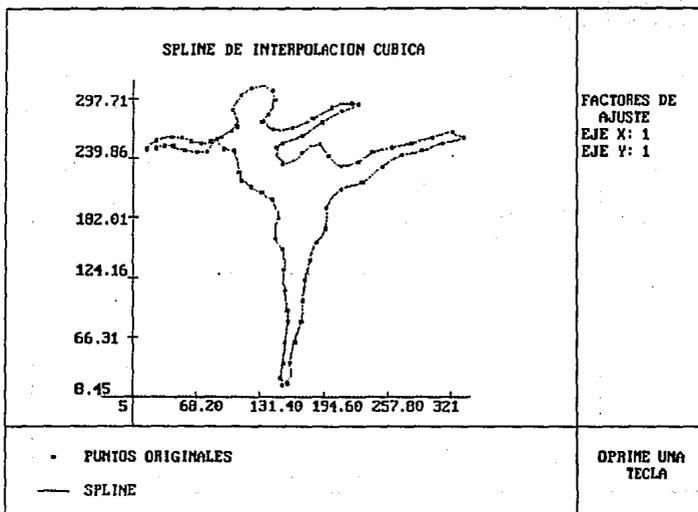
Su curva es cerrada o abierta ?(c/a)

c

En cuantos pto. desea evaluar el spline?

(max=100)

100



CORRIDA 2

.....
 . Cálculo del spline cúbico de Interpolación para .
 . datos de una curva o función.

Desee la entrada de datos por

Archivo o por pantalla ? [a/p]

a

Diga el nombre del archivo de datos

sin.dat

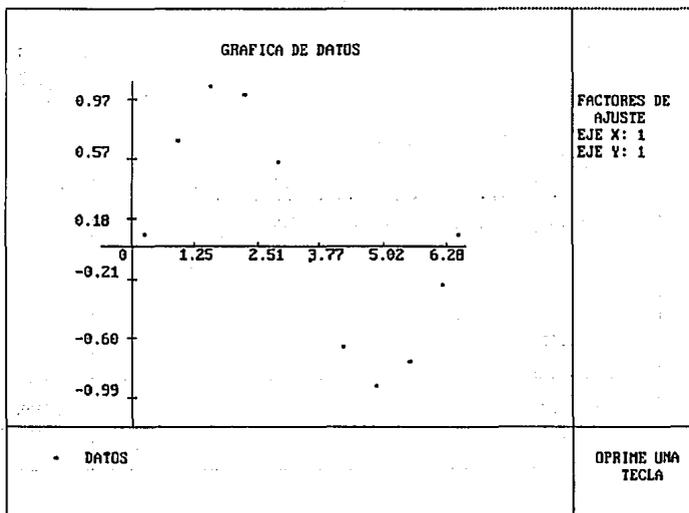
Los datos corresponden a una
 función o a una curva ? [f/c]

f

Los datos están en coord. polares

o cartesianas ? [p/c]

c



CONTINUACION

diga el nombre del archivo de resultados
sin.res

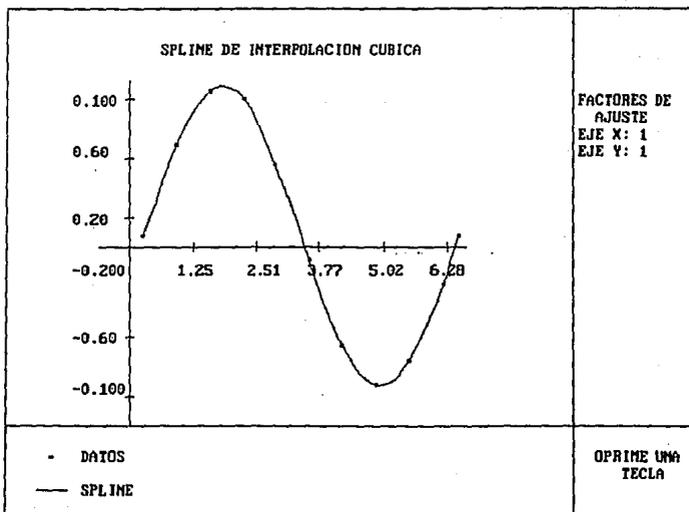
escoja una de las variantes siguientes :

- 1-No hay condiciones en los extremos.
- 2-La 2da derivada en los extremos es 0
- 3-Se conoce la 1ra derivada en los extremos
- 4-La función es periódica

4

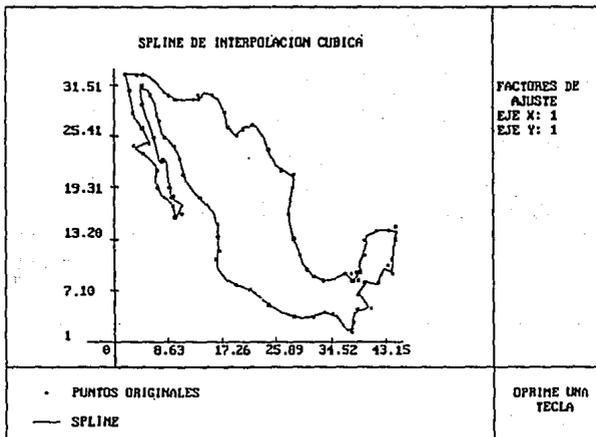
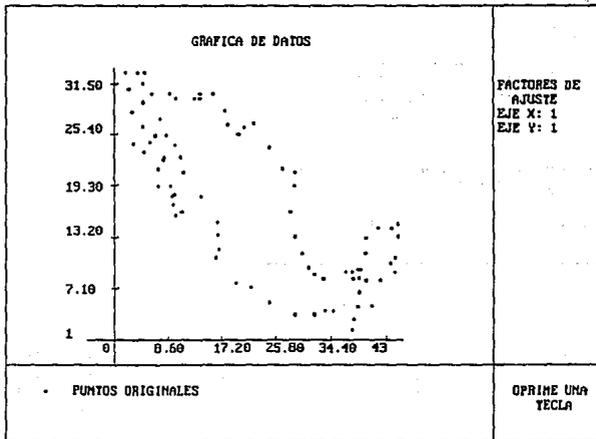
En cuantos pto desea evaluar el spline?
(max=100)

100



CORRIDA 3

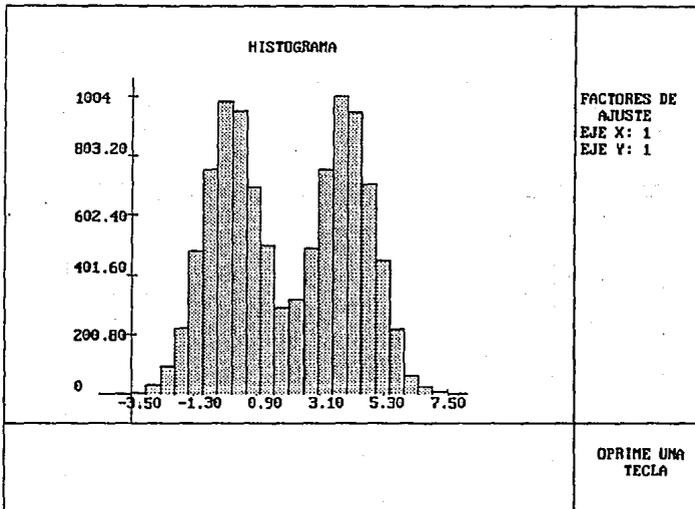
archivo de datos mapa.det
 archivo de resultados mapa.res



SUAVIZAMIENTO DE HISTOGRAMAS
CORRIDA 4

.....
 . Calculo del histospline parabolico que .
 suaviza un histograma

Desea la entrada de datos por
 Archivo o por pantalla ? (a/p)
 A
 Diga el nombre del archivo de datos
 N0141.DAT



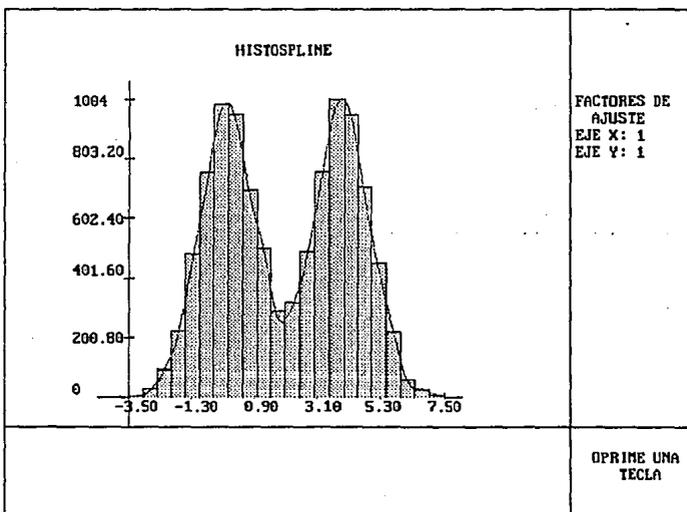
CONTINUACION

Escoja una de las sgtes condiciones
en la frontera

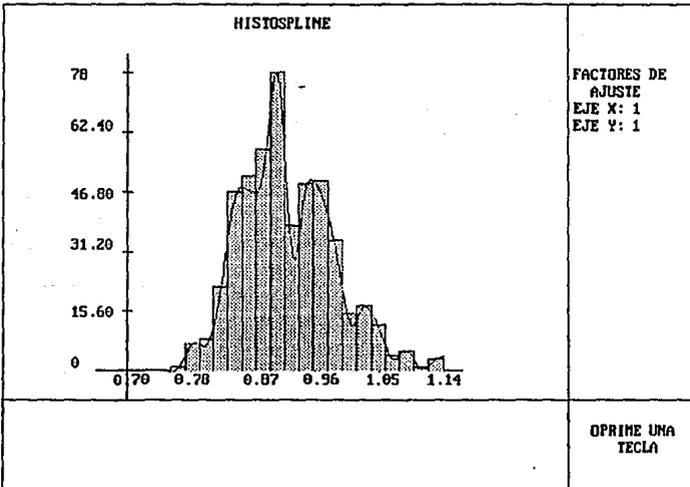
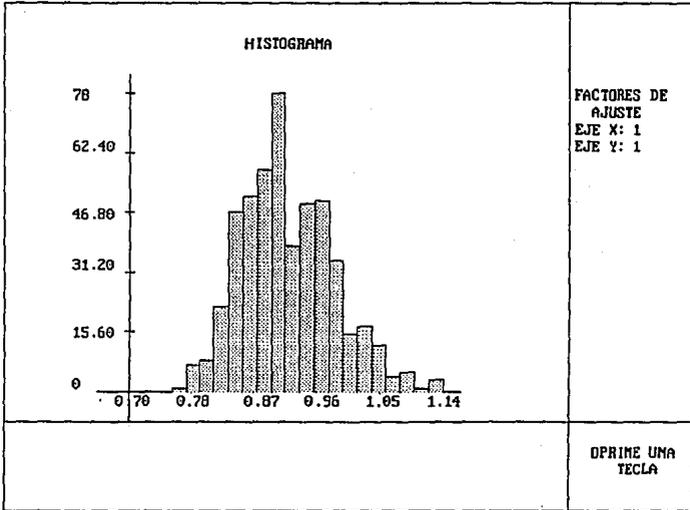
1. El histospline se anula en los extremos
2. La 1ra derivada del histospline se anula en los extremos

2

Diga el nombre del archivo de resultados
N0141.RES

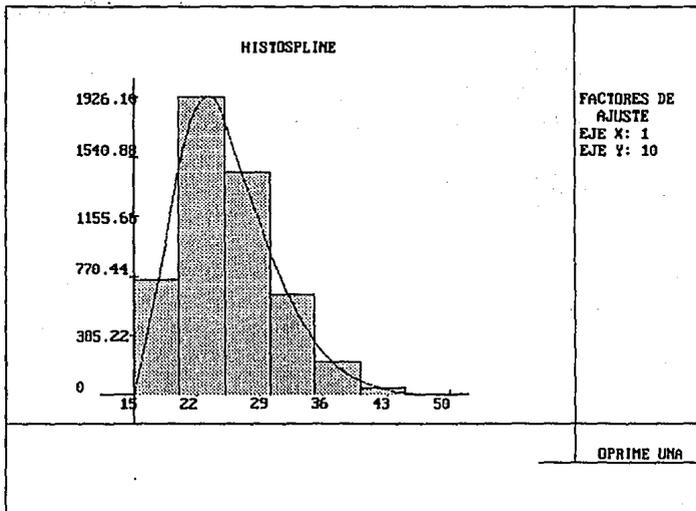
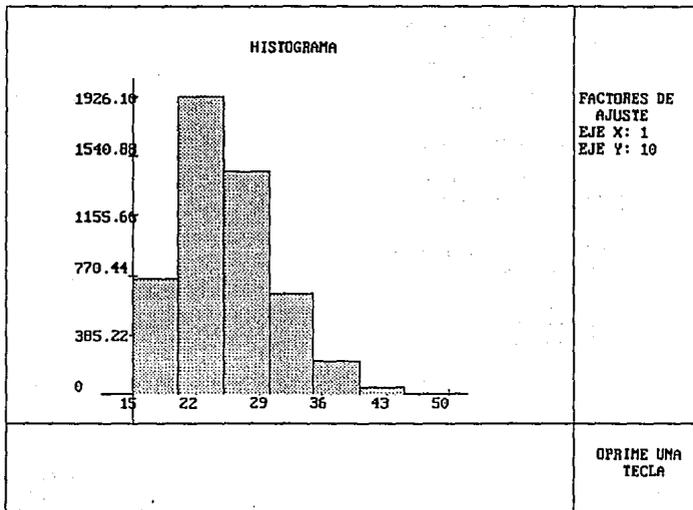


CORRIDA 5
 archivo de datos N10KA-1.DAT



CORRIDA 6

archivo de datos MADREBULDAT



AJUSTE Y SUAVIZAMIENTO CORRIDA 7

.....
Suavizamiento y Ajuste
 de Datos

Desea la entrada de datos por
 Archivo o por pantalla ? (a/p)

A

Diga el nombre del archivo de datos
 DFSIN.DAT

Los datos corresponden a una
 función o a una curva ? (f/c)

F

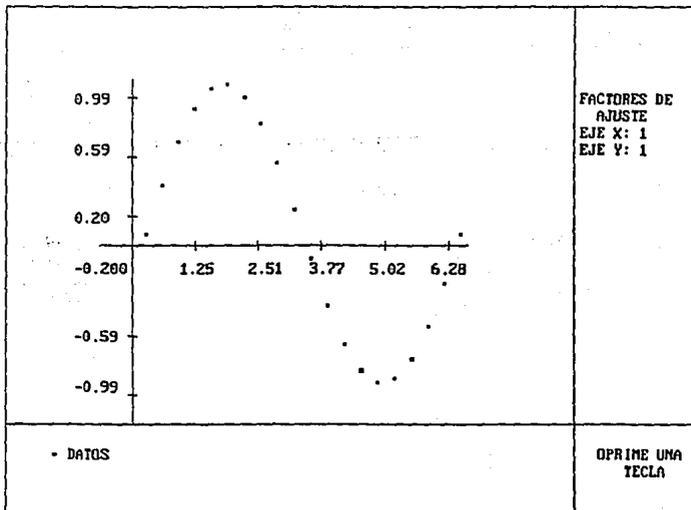
Los datos están en coord. polares
 o cartesianas ? (p/c)

C

Error :

1. Por archivo.
2. Por pantalla.

1



CONTINUACION

diga el nombre del archivo de resultados

SIN.RES

Informacion sobre la cota superior para la distancia
media cuadratica de la aprox. a los datos
Cuantas cotas mín 1 (max 3) ?

3

Se recomienda tomar las cotas en
el intervalo [13.675, 26.325]

Valores de las cotas (>= 0)

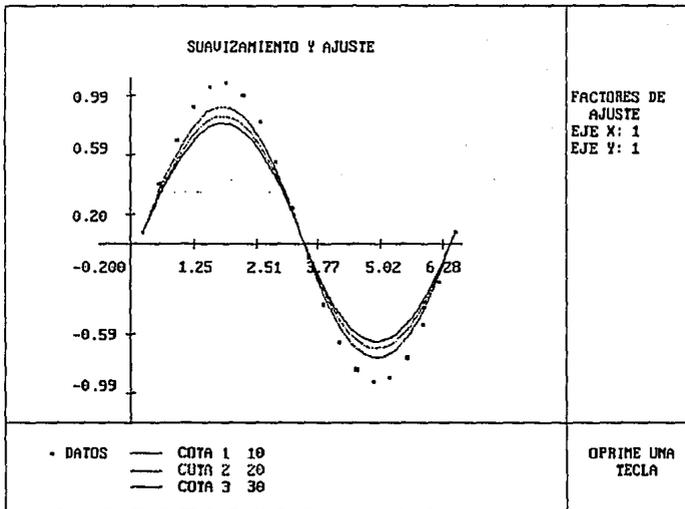
10

20

30

En cuantos puntos desea evaluar el spline ?(max=100)

100



APENDICE

Algebra Lineal

Básica

A. Álgebra Lineal Básica

La construcción de una gran variedad de funciones splines, requiere la solución de sistemas de ecuaciones lineales que nos permiten calcular sus coeficientes. Las matrices de tales sistemas tienen propiedades especiales. En particular, en el caso de spline cúbico de suavizamiento la matriz es simétrica y definida positiva, mientras que en el resto de los casos que vimos en el capítulo II, las matrices son tridiagonales o tridiagonales cíclicas, con diagonal dominante por filas o por columnas.

En esta sección vamos a demostrar que tales matrices son no singulares y en consecuencia los sistemas de ecuaciones tienen solución única. Además discutiremos de forma muy breve, los métodos computacionales más utilizados para resolver los mismos. Comencemos recordando que una matriz $A = (a_{i,j})_{i,j=1,\dots,n}$ se dice de diagonal dominante (por filas) si :

$$|a_{i,i}| \geq \sum_{j=1, j \neq i}^n |a_{i,j}|,$$

y la desigualdad es estricta al menos para un i . Por otro lado, la matriz A es tridiagonal si

$$a_{i,j} = 0 \quad \text{para } |i - j| > 1,$$

y tridiagonal cíclica si

$$a_{i,j} = 0 \quad \text{para } 1 < |i - j| < n - 1.$$

En particular las matrices tridiagonales o tridiagonales cíclicas, que aparecen en los sistemas de ecuaciones para el cálculo de los splines ya mencionados, tienen todos los elementos de la subdiagonal y supradiagonal diferente de cero. El siguiente teorema nos asegura, que tales matrices son no singulares:

Teorema A.1.

Sea $A = (a_{i,j})_{i,j=1,\dots,n}$ una matriz tridiagonal o tridiagonal cíclica de diagonal dominante por filas y con elementos no nulos en subdiagonal y la supradiagonal. Entonces A es no singular.

Demostración.

Supongamos que A fuese singular, entonces existe un vector $x = (x_1, x_2, \dots, x_n)^t$ no nulo tal que

$$Ax = 0. \quad (A.1)$$

Sea $|x_k| = \max_{1 \leq j \leq n} |x_j|$.

Como x no es el vector nulo $|x_k| > 0$. Por otro lado, de (A.1) tenemos que

$$\sum_{j=1}^n a_{k,j} x_j = 0. \quad (A.2)$$

Pero como A es tridiagonal $a_{k,j} = 0$ para $|k-j| > 1$ y de (A.2) obtenemos que

$$a_{k,k-1} x_{k-1} + a_{k,k} x_k + a_{k,k+1} x_{k+1} = 0,$$

es decir

$$-a_{k,k} x_k = a_{k,k-1} x_{k-1} + a_{k,k+1} x_{k+1},$$

luego

$$|a_{k,k}| \leq |a_{k,k-1}| \frac{|x_{k-1}|}{|x_k|} + |a_{k,k+1}| \frac{|x_{k+1}|}{|x_k|}. \quad (A.3)$$

Si $|x_k| > |x_j|$ para $j = k-1$ o $j = k+1$, entonces como $a_{k,k-1}$ y $a_{k,k+1}$ son no nulos por hipótesis, de (A.3) obtenemos que :

$$|a_{k,k}| < |a_{k,k-1}| + |a_{k,k+1}|,$$

lo cual contradice que A es de diagonal dominante por filas .

Por otro lado, si $|x_k| = |x_{k-1}| = |x_{k+1}|$, entonces podríamos repetir la demostración anterior con la ecuación $k-1$ o $k+1$, si se cumpliese una de las alternativas siguientes

- i) $|x_{k-1}| \geq |x_j|$ para $j = k-2$ o $j = k$.
- ii) $|x_{k+1}| \geq |x_j|$ para $j = k$ o $j = k+2$.

Sin embargo, si para ningún k se cumple que

$$|x_k| \geq |x_j| \quad \text{para } j = k - 1 \quad \text{o} \quad j = k + 1$$

entonces esto significa que $|x_1| = |x_2| = \dots = |x_n|$ y en tal caso de (A.3) podemos concluir que

$$|a_{k,k}| \leq |a_{k,k-1}| + |a_{k,k+1}| \quad \text{para todo } k,$$

lo cual contradice que A sea diagonal dominante. ■

La demostración para el caso tridiagonal cíclico es muy similar y queda de ejercicio para el lector. Los sistemas de ecuaciones lineales con matriz tridiagonal se pueden resolver de manera eficiente por el método de eliminación de Gauss, que describiremos brevemente a continuación para este caso. Por otro lado, se puede demostrar que si la matriz del sistema tiene todos los elementos de la subdiagonal y supradiagonal diferentes de cero (es irreducible) entonces no es necesario utilizar el pivoteo en el método de Gauss. Una de las ventajas inmediatas que tiene la solución de sistemas de ecuaciones lineales con matriz tridiagonal, es el hecho de que se puede aprovechar su estructura de banda y guardar solamente los elementos no nulos de la matriz en tres vectores, lo cual consume $3n$ localizaciones de memoria, en lugar del total de n^2 que se necesitan para guardar una matriz A densa.

**A.1. Solución de sistemas de ecuaciones
con matriz tridiagonal**

Supongamos que la matriz A del sistema

$$Ax = b \quad (A.4)$$

se escribe de la forma

$$\begin{pmatrix} d_1 & c_1 & & 0 \\ a_2 & d_2 & c_2 & \\ \ddots & \ddots & \ddots & \ddots \\ 0 & a_{n-1} & d_{n-1} & c_{n-1} \\ & 0 & a_n & d_n \end{pmatrix}.$$

La idea esencial del método de Gauss consiste en transformar el sistema (A.4) en otro sistema, cuya matriz sea triangular superior y que a su vez tenga la misma solución que (A.1). Esto se debe a que un sistema con matriz triangular superior se puede resolver fácilmente. En efecto, supongamos que se tiene el sistema

$$Ax = b,$$

donde A es una matriz de la forma

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ & a_{22} & \dots & a_{2n} \\ & & \ddots & \vdots \\ & & & a_{nn} \end{pmatrix} \quad (A.5)$$

con elementos no nulos en la diagonal.

Entonces, de la última ecuación se obtiene inmediatamente que como $a_{n,n} \neq 0$

$$x_n = \frac{b_n}{a_{nn}}.$$

Una vez conocida x_n podemos sustituir en la penúltima ecuación y como $a_{n-1,n-1}$ es diferente de cero, obtenemos

$$x_{n-1} = \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}}.$$

En general, una vez que se han calculado $x_{k+1}, x_{k+2}, \dots, x_n$, como $a_{k,k} \neq 0$, de la k -ésima ecuación llegamos a que

$$x_k = \frac{b_k - \sum_{j=k+1}^n a_{k,j} x_j}{a_{k,k}}.$$

Este proceso se conoce como sustitución hacia atrás ya que las incógnitas se calculan en orden x_n, x_{n-1}, \dots, x_1 . Si la matriz del sistema de ecuaciones no es triangular superior, entonces se debe transformar inicialmente mediante el método de eliminación de Gauss. El método se basa en el hecho de que la solución de un sistema de ecuaciones lineales, no cambia si se realiza una de las siguientes operaciones elementales:

- i) Multiplicación de una ecuación por una constante no nula.
- ii) Adición de un múltiplo de una ecuación a otra ecuación.
- iii) Intercambio de dos ecuaciones.

En particular, si la matriz A del sistema de ecuaciones es tridiagonal, entonces resulta muy fácil transformarla en una triangular superior, mediante operaciones elementales, ya que sólo habría que "hacer cero" los elementos de la subdiagonal. Como en la primera ecuación no hay ningún elemento en la subdiagonal, entonces podemos empezar con la segunda. Una forma de eliminar la incógnita x_1 de esta ecuación (o lo que es equivalente, poner un cero en lugar de a_2) sería sustraer a_2/d_1 veces la primera ecuación de la segunda. La nueva segunda ecuación sería entonces

$$d'_2 x_2 + c_2 x_3 = b'_2,$$

donde

$$d'_2 = d_2 - \frac{a_2}{d_1} c_1, \quad (A.6)$$

y

$$b'_2 = b_2 - \frac{a_2}{d_1} b_1.$$

Nótese que en particular el coeficiente de x_3 en la segunda ecuación, es decir c_2 , no se altera por esta transformación, debido a que el coeficiente de x_3 en la primera ecuación es cero.

Después de este primer paso de la eliminación el sistema original se ha transformado en otro equivalente (con la misma solución) de la forma

$$A^{(1)}x = b^{(1)},$$

donde

$$A^{(1)} = \begin{pmatrix} d'_1 & c_1 & & & 0 \\ & d'_2 & c_2 & & \\ & a_3 & c_3 & & \\ & & \ddots & \ddots & \\ & & & a_{n-1} & c_{n-1} \\ 0 & & & a_n & d_n \end{pmatrix} \quad b^{(1)} = \begin{pmatrix} b'_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}$$

$$\text{y } b'_1 = b_1, \quad d'_1 = d_1.$$

Suponiendo entonces que d'_2 es diferente de cero podemos utilizar la ecuación (A.6) para eliminar la incógnita x_2 de la tercera ecuación. En efecto, si a esta ecuación le restamos a_3/d'_2 veces la segunda ecuación obtenemos :

$$d'_3 x_3 + c_3 x_4 = b'_3, \quad (A.7)$$

donde

$$d'_3 = d_3 - \frac{a_3}{d'_2} c_2,$$

y

$$b'_3 = b_3 - \frac{a_3}{d'_2} b'_2.$$

De este modo, al concluir el segundo paso del proceso de eliminación tenemos el siguiente sistema equivalente a (A.4)

$$A^{(2)}x = b^{(2)},$$

donde

$$A^{(2)} = \begin{pmatrix} d'_1 & c'_1 & & & \\ & d'_2 & c_2 & & \\ & & d'_3 & c_3 & \\ & & a_4 & d_4 & c_4 \\ & & & \ddots & \ddots \\ & & & & a_n & d_n \end{pmatrix}, \quad b^{(2)} = \begin{pmatrix} b'_1 \\ b'_2 \\ b'_3 \\ b_4 \\ \vdots \\ b_n \end{pmatrix}.$$

Continuando de esta manera, en el k -ésimo paso podemos eliminar x_k de la ecuación $k+1$ (suponiendo que $d'_k \neq 0$), lo cual da lugar a la nueva ecuación

$$d'_{k+1}x_{k+1} + c_{k+1}x_{k+2} = b'_{k+1},$$

donde

$$d'_{k+1} = d_{k+1} - \frac{a_{k+1}}{d'_k}c_k \quad \text{para } k = 1, 2, \dots, n-1,$$

y

$$b'_{k+1} = b_{k+1} - \frac{a_{k+1}}{d'_k}b'_k.$$

Al concluir este proceso hemos transformado el sistema original (A.4) en un sistema equivalente

$$A^{(n-1)}x = \delta^{(n-1)}, \quad (\text{A.8})$$

donde $A^{(n-1)}$ es una matriz triangular superior de la forma

$$A^{(n-1)} = \begin{pmatrix} d'_1 & c_1 & & & 0 \\ a_1 & d'_2 & c_2 & & \\ & & \ddots & & \\ & & & d'_{n-1} & c_{n-1} \\ 0 & & & & d'_n \end{pmatrix}, \quad \delta^{(n-1)} = \begin{pmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_{n-1} \\ b'_n \end{pmatrix}.$$

El sistema triangular (A.8) se puede resolver incluso más rápido que (A.5), ya que en cada ecuación del mismo sólo aparecen involucradas dos incógnitas. En efecto, de la última ecuación se obtiene que

$$x_n = \frac{b'_n}{d'_n}.$$

Sustituyendo en la penúltima resulta que

$$x_{n-1} = \frac{b'_{n-1} - c_{n-1}x_n}{d'_{n-1}},$$

y en general para $k = n-1, \dots, 1$, se tiene que

$$x_k = \frac{b'_k - c_k x_{k+1}}{d'_k}.$$

A.I.1. Algoritmo

El algoritmo para resolver sistemas de ecuaciones con matriz tridiagonal se puede resumir de la siguiente forma :

Dados los coeficientes a_i , d_i , c_i y el lado derecho b_i del sistema tridiagonal

$$a_i x_{i-1} + d_i x_i + c_i x_{i+1} = b_i, \quad i = 1 \dots n \quad (\text{con } a_1 = c_n = 0).$$

Repetir para $k = 2, \dots, n$

Si $d_{k-1} = 0$ detener el proceso (la matriz es singular para la precisión con que estamos trabajando)

En otro caso

Calcular

$$m = a_k / d_{k-1}$$

$$d_k = d_k - m \cdot c_{k-1}$$

$$b_k = b_k - m \cdot b_{k-1}$$

Si $d_n = 0$ detener el proceso (la matriz es singular para la precisión con que estamos trabajando)

En otro caso

$$x_n = b_n / d_n$$

Repetir para $k = n - 1, \dots, 1$

$$x_k = (b_k - c_k x_{k+1}) / d_k.$$

A.1.2. Subrutina TRID

El algoritmo mencionado en la sección anterior aparece programado en la subrutina TRID escrita en FORTRAN 77. Los parámetros de entrada de la subrutina son el orden n del sistema tridiagonal

$$\text{Sub}(i)x_{i-1} + \text{diag}(i)x_i + \text{sup}(i)x_{i+1} = b(i) \quad (\text{A.9})$$

así como los vectores de dimensión n , sub, diag, sup y b, que contienen los coeficientes del sistema (A.9) y el lado derecho del mismo. Al finalizar la subrutina la solución del sistema se almacena sobre el vector b .

```

Subroutine Trid (sub, diag, sup, b, n)
Integer n, x
real b(n), diag(n), sub(n), sup(n)
if (n .le. 1) then
    b(1) = b(1)/diag(1)
    return
endif
do 11 i= 2, n
    sub(i) = sub(i) / diag(i-1)
    diag(i) = diag(i) - sub(i) sup(i-1)
11 b(i) = b(i) - sub(i) b(i-1)
    b(n) = b(n) / diag(n)
do 12 i = n-1, 1
12    b(i) = (b(i) - sup(i) b(i+1)) /diag(i)
return
end

```

Ejemplo.

Resolver el sistema de ecuaciones lineales

$$\begin{aligned}
 +2x_1 - x_2 &= 1 \\
 -x_{i-1} + 2x_i - x_{i+1} &= 0 \quad i = 2, \dots, n-1 \\
 -x_{n-1} + 2x_n &= 0,
 \end{aligned} \tag{A.10}$$

donde $n = 10$.

Como la matriz del sistema (A.10) es tridiagonal y de diagonal dominante, para resolver el mismo basta con escribir un programa principal que llame a la subrutina TRID. La solución exacta de este sistema es

$$x_i = \frac{n+1-i}{n+1}, \quad i = 1, \dots, n.$$

El programa que ofrecemos a continuación compara la solución exacta x_i , con la solución aproximada \tilde{x}_i , e imprime el valor relativo en cada punto dado por :

$$e(i) = \left| \frac{x_i - \tilde{x}_i}{x_i} \right|, \quad i = 1, \dots, n.$$

Además calcula la norma uniforme del error relativo, es decir

$$\|e\| = \max_{1 \leq i \leq n} \{e(i)\}$$

- C Programa para calcular la solución
 C del sistema lineal (A.10)
 C
 parameter (n=10)
 integer i
 real a(n), b(n), c(n), d(n), errmax
 C Formación de la matriz del sistema (2.2.6)
 do 10 i=1, n
 a(i) = -1
 d(i) = 2
 c(i) = -1
 10 b(i) = 0
 b(1) = 1
 C Solución del sistema (2.2.6)

```

call Trid (a, d, c, b, n)
print *, 'la solución del sistema es '
write(*,20) (i, b(i), i=1, n)
20 format (i5, e15.7)
errmax = 0
C
Calculo del error relativo
do 30 i=1, n
    a(i) = (float (n+1 -i) ) / float (n+1)
    c(i) = abs( (a(i) - b(i)) / a(i))
    if (c(i) .gt. errmax) errmax = c(i)
30 continue
print *, '
print *, 'El error relativo '
write (*,20) (i, c(i), i=1,n)
print *, 'Norma del error relativo
write (*,40) errmax
40 format (e15.7)
stop
end

```

La solución del sistema (A.10), utilizando este programa en una IBM AT, se muestran a continuación.

Solución del sistema (A.10)

1	0.9090070	E + 00
2	0.8181810	E + 00
3	0.7272720	E + 00
4	0.6363628	E + 00
5	0.5454537	E + 00
6	0.4545446	E + 00
7	0.3636356	E + 00
8	0.2727266	E + 00
9	0.1818177	E + 00
10	0.9090886	E - 01.

La norma del error relativo que se obtuvo, fue $\|e\| = 0.2622604 E - 05$, lo cual significa que el error de la solución aproximada aparece en la quinta cifra decimal.

A.1.3. Programa TRIDSIS

En general para resolver un sistema de ecuaciones lineales con matriz tridiagonal y diagonal dominante, se puede utilizar el siguiente programa

C Programa TRIDSIS para resolver un sistema de ecuaciones lineales
 C con matriz tridiagonal y de diagonal dominante. El sistema se resuelve
 C por el método de eliminación de Gauss sin pivoteo, utilizando la
 C subrutina TRID. La matriz se lee por diagonales y la solución del
 C sistema se guarda en el vector b que contiene al inicio el lado
 C derecho del sistema.
 C

```

parameter (nmax=50)
integer i
real a(nmax), b(nmax), c(nmax), d(nmax)
print *, 'Solución de un sistema lineal'
print *, ' con matriz tridiagonal'
print *, '
print *, ' orden del sistema'
read(*,*) n
print *, 'Lectura de la matriz por diagonales'
print *, '
print *, 'diagonal'
do 10 i=1, n
    write(*,5) i
10    read(*,*) d(i)
5    format (5x, 'd(', i2, ')=' )
    print *, 'subdiagonal'
    do 20 i=2, n
        write(*,15) i
20    read(*,*) a(i)
15    format (5x, 'a(', i2, ')=' )
    print *, 'supradiagonal'
    do 30 i=1, n-1

```

```
        write(x,25) i
30      read (*,*) c(i)
25     format (5x, 'c(', i2, ')=' )
       print *, 'lado derecho'
       do 40 i=1, n
           write(x,35) i
30      read (*,*) b(i)
35     format (5x, 'b(', i2, ')=' )
       call Trid(a, d, c, b, n)
       print *, 'La solución del sistema es'
       write(*,50) (i, b(i), i=1, n)
50     format (1x, e15.7)
       stop
       end
```

La subrutina SGTSL que se halla en el LINPACK , resuelve de manera aún más eficiente sistemas de ecuaciones con matriz tridiagonal, aunque en este caso se utiliza pivoteo parcial, porque la subrutina también es válida para matrices tridiagonales que no tengan diagonal dominante.

**A.2. Solución de sistemas de ecuaciones
con matriz tridiagonal cíclica**

Vemos ahora como se modifica el método de eliminación de Gauss que acabamos de describir, si la matriz es tridiagonal cíclica, como la que aparece en el caso de spline cúbico periódico .

Consideremos el sistema lineal

$$Ax = b \quad (A.11)$$

donde A es una matriz tridiagonal cíclica dada por

$$\begin{pmatrix} d_1 & c_1 & & & a_1 \\ a_2 & d_2 & c_2 & & 0 \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & d_{n-1} & c_{n-1} \\ c_n & & & a_n & d_n \end{pmatrix}$$

Según el método de Gauss para cada $k = 2, \dots, n-2$ debemos transformar las ecuaciones k y n , a partir de la ecuación $k-1$, con el objetivo de eliminar la incógnita x_k de las ecuaciones k a la n .

Vamos a denotar por r_2, \dots, r_{n-2} las componentes del vector r que se va formando en la última columna de la matriz A , como consecuencia de los pasos $1, \dots, n-2$ de la eliminación de Gauss. Entonces en el primer paso, debemos eliminar x_1 de la segunda y la última ecuación, restándole a la segunda a_2/d_1 veces la primera ecuación y a la última, c_n/d_1 veces la primera. En otras palabras debemos calcular

$$\begin{aligned} d'_2 &= d_2 - \frac{a_2}{d_1} c_1, & c_n &= -\frac{c_n}{d_1} c_1, \\ b'_2 &= b_2 - \frac{a_2}{d_1} b_1, & d'_n &= d_n - \frac{c_n}{d_1} a_1, \\ r_2 &= -\frac{a_2}{d_1} a_1, & b'_n &= b_n - \frac{c_n}{d_1} b_1. \end{aligned}$$

Al concluir este paso tenemos el sistema equivalente

$$\begin{pmatrix} d'_1 & c_1 & & & r_1 \\ & d_2 & c_2 & & r_2 \\ & a_3 & d_3 & c_3 & 0 \\ & \ddots & \ddots & \ddots & \vdots \\ & & a_{n-1} & d_{n-1} & c_{n-1} \\ 0 & c_n & 0 & a_n & d_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ b_3 \\ \vdots \\ b'_{n-1} \\ b'_n \end{pmatrix}$$

donde $r_1 = a_1$, $d'_1 = d_1$, $b'_1 = b_1$.

En general, para $k = 2, \dots, n-2$, eliminamos la incógnita x_{k-1} de la k -ésima ecuación restándole a esta a_k/d_{k-1} veces la ecuación $k-1$. Esto significa que debemos calcular

$$\begin{aligned} d'_k &= d_k - \frac{a_k}{d_{k-1}} c_{k-1}, \\ r_k &= -\frac{a_k}{d_{k-1}} r_{k-1}, \\ b'_k &= b_k - \frac{a_k}{d_{k-1}} b'_{k-1} \end{aligned}$$

Además eliminamos la incógnita x_{k-1} de la n -ésima ecuación, restándole a la misma c_n/d_{k-1} veces la ecuación $k-1$. Para ello debemos calcular

$$c_n = -\frac{c_n}{d_{k-1}} c_{k-1}, \quad d'_n = d'_n - \frac{c_n}{d_{k-1}} r_{k-1}, \quad b'_n = b'_n - \frac{c_n}{d_{k-1}} b_k$$

Al cabo de estos $n-3$ pasos del método de Gauss, el sistema original se ha transformado en el sistema equivalente

$$\begin{pmatrix} d'_1 & c_1 & & & r_1 \\ & d_2 & c_2 & & r_2 \\ & & \ddots & & \vdots \\ & & & d'_{n-2} & c_{n-2} & r_{n-2} \\ & & & a_{n-1} & d_{n-1} & c_{n-1} \\ 0 & & & c_n & a_n & d_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_{n-2} \\ b'_{n-1} \\ b'_n \end{pmatrix} \quad (A.12)$$

Para concluir el proceso debemos entonces eliminar de la ecuación x_{n-2} de la ecuación $n-1$ y x_{n-2} y x_{n-1} de la n -ésima ecuación. Esto se logra restándole

a la ecuación $n-1$, a_{n-1}/d'_{n-2} veces la ecuación $n-2$ y a la ecuación n , c_n/d'_{n-2} veces la ecuación $n-2$. De este modo obtenemos

$$\begin{aligned}d'_{n-1} &= d_{n-1} - \frac{a_{n-1}}{d'_{n-2}} c_{n-2} & c_n &= a_n - \frac{c_n}{d'_{n-2}} c_{n-2} \\r_{n-1} &= c_{n-1} - \frac{a_{n-1}}{d'_{n-2}} r_{n-2} & d'_n &= d'_n - \frac{c_n}{d'_{n-2}} r_{n-2} \\b'_{n-1} &= b'_{n-1} - \frac{a_{n-1}}{d'_{n-2}} b'_{n-2} & b'_n &= b'_n - \frac{c_n}{d'_{n-2}} b'_{n-2},\end{aligned}$$

y el sistema (A.12) se transforma en el sistema equivalente

$$\begin{pmatrix} d'_1 & c_1 & & & r_1 \\ & d'_2 & c_2 & & r_2 \\ & & \ddots & & \vdots \\ & & & d'_{n-2} & c_{n-2} & r_{n-2} \\ & & & & d'_{n-1} & r_{n-1} \\ 0 & & & & & c_n & d'_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_{n-2} \\ b'_{n-1} \\ b'_n \end{pmatrix}$$

Por último, para eliminar x_{n-1} de la ecuación n , le restamos a ésta $\frac{c_n}{d'_{n-1}}$ veces la ecuación $n-1$ y calculamos entonces:

$$\begin{aligned}d'_n &= d'_n - \frac{c_n}{d'_{n-1}} r_{n-1}, \\ b'_n &= b'_n - \frac{c_n}{d'_{n-1}} b'_{n-1}.\end{aligned}$$

Esto nos conduce finalmente a un sistema triangular equivalente al original dado por

$$\begin{pmatrix} d'_1 & c_1 & & & r_1 \\ & d'_2 & c_2 & & r_2 \\ & & \ddots & & \vdots \\ & & & d'_{n-2} & c_{n-2} & r_{n-2} \\ & & & & d'_{n-1} & r_{n-1} \\ 0 & & & & & c_n & d'_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_{n-2} \\ b'_{n-1} \\ b'_n \end{pmatrix} \quad (\text{A.13})$$

La solución de (A.11) se obtiene entonces mediante sustitución hacia atrás, a partir de (A.13). En efecto, de la última ecuación obtenemos inmediatamente que

$$x_n = b'_n/d'_n.$$

Sustituyendo x_n en la ecuación $n - 1$ resulta entonces que

$$x_{n-1} = (b'_{n-1} - r_{n-1}x_n)/d'_{n-1}.$$

En general, para $k = n - 2, \dots, 1$ se calcula x_k como

$$x_k = \frac{b'_k - c_k x_{k+1} - r_k x_n}{d'_k}.$$

A.2.1. Algoritmo

El algoritmo para resolver sistemas de ecuaciones con matriz tridiagonal cíclica y de diagonal dominante se puede resumir entonces de la siguiente forma.

Dados los coeficientes a_i, d_i, c_i y el lado derecho b_i del sistema tridiagonal cíclico

$$\begin{aligned}d_1 x_1 + c_1 x_2 + a_1 x_n &= b_1, \\d_i x_{i-1} + d_i x_i + c_i x_{i+1} &= b_i, & i = 2, \dots, n-1 \\c_n x_1 + a_n x_{n-1} + d_n x_n &= b_n.\end{aligned}$$

Asignar $r_1 = a_1$

Repetir para $k = 2, \dots, n-2$

Si $d_{k-1} = 0$ detener el proceso

(la matriz es singular para la
precisión con que trabajamos)

En otro caso, calcular

$$\begin{aligned}m &= a_k / d_{k-1} \\d_k &= d_k - m \cdot c_{k-1} \\r_k &= -m \cdot r_{k-1} \\l &= c_n / d_{k-1} \\c_n &= -l \cdot c_{k-1} \\d_n &= d_n - l \cdot r_{k-1} \\b_n &= b_n - l \cdot b_{k-1}\end{aligned}$$

Si $d_{n-2} = 0$ detener el proceso

(la matriz es singular para la
precisión con que trabajamos)

En otro caso, calcular

$$\begin{aligned}m &= a_{n-1} / d_{n-2} \\d_{n-1} &= d_{n-1} - m \cdot c_{n-2} \\r_{n-1} &= c_{n-1} - m \cdot r_{n-2} \\b_{n-1} &= b_{n-1} - m \cdot b_{n-2} \\l &= c_n / d_{n-2} \\c_n &= a_n - l \cdot c_{n-2}\end{aligned}$$

$$d_n = d_n - l \cdot r_{n-2}$$

$$b_n = b_n - l \cdot b_{n-2}$$

Si $d_{n-1} = 0$ detener el proceso

(la matriz es singular para la
precisión con que trabajamos)

En otro caso, calcular

$$l = c_n/d_{n-1}$$

$$d_n = d_n - l \cdot r_{n-1}$$

$$b_n = b_n - l \cdot b_{n-1}$$

Si $d_n = 0$ detener el proceso

(la matriz es singular para la
precisión con que trabajamos)

En otro caso, calcular

$$x_n = b_n/d_n$$

$$x_{n-1} = (b_{n-1} - r_{n-1}x_n)/d_{n-1}$$

Repetir para $k = n-2, \dots, 1$

$$x_k = (b_k - c_k x_{k+1} - r_k x_n)/d_k$$

A.2.2. Subrutina TRIDE

El algoritmo que acabamos de describir en la sección anterior aparece programado en la subrutina TRIDE, escrita en Fortran 77. Los parámetros de entrada de esta subrutina son el orden n del sistema tridiagonal cíclico.

$$\begin{aligned} \text{diag}(1)x_1 + \text{sup}(1)x_2 + \text{sub}(1)x_n &= b(1), \\ \text{sub}(i)x_{i-1} + \text{diag}(i)x_i + \text{sup}(i)x_{i+1} &= b(i), \quad i = 2, \dots, n-1, \quad (\text{A.14}) \\ \text{sup}(n)x_1 + \text{sub}(n)x_{n-1} + \text{diag}(n)x_n &= (b_n), \end{aligned}$$

así cómo los vectores de dimensión n , sub , diag , sup y b , que contienen los coeficientes del sistema (A.14) y al lado derecho del mismo. La solución del sistema a la salida se guarda sobre el propio vector b . El vector ULT es un arreglo de trabajo y contiene la última columna del sistema en cada paso del método de eliminación.

```

Soubroutine Tride (sub, diag, sup, ult, b, n)
integer n, i
real b(n), diag(n), sub(n), sup(n), ult(n)
if (n .le. 1) then
    b(1) = b(1) / diag(1)
    return
endif
    ult(1) = sub(1)
do 11 i= 2, n-2
C   Transformación de la ecuación i a partir de la i-1
    sub(i)= sub(i) / diag(i-1)
    diag(i) = diag(i) - sub(i) sup(i-1)
    ult(i) = -sub(i) ult(i-1)
    b(i) = b(i) - sub(i) b(i-1)
C   Transformación de la ecuación n a partir i-1
    sub(i) = sub(n) / diag(i-1)
    sup(n) = -sub(i) sup(i-1)

```

```

diag(n) = diag(n) - sub(i) ult (i-1)
b(n) = b(n) - sub(i) b(i-1)
11 continue
C Transformación de la ecuación n-1 a partir n-2
  sub(n-1) = sub(n-1) / diag(n-2)
  diag(n-1) = diag(n-1) - sub(n-1) sup (n-2)
  ult(n-1) = sup(n-1) - sub(n-1) ult(n-2)
  b(n-1) = b(n-1) - sub(n-1) b(n-2)
C Transformación de la ecuación n a partir n-2
  sub(n-1) = sup(n) / diag(n-2)
  sub(n) = sub(n) - sub(n-1) sup(n-2)
  diag(n) = diag(n) - sub(n-1) ult (n-2)
  b(n) = b(n) - sub(n-1) b(n-2)
C Transformación de la ecuación n a partir n-1
  sub(n) = sub(n) / diag(n-1)
  diag(n) = diag(n) - sub(n) ult (n-1)
  b(n) = b(n) - sub(n) b(n-1)
C Sustitución hacia atrás
  b(n) = b(n) / diag(n)
  b(n-1) = ( b(n-1) - ult (n-1) b(n) ) / diag(n-1)
do 12 i = n-2, 1, -1
  b(i) = (b(i) -sup(i) b(i+1) - ult(i) b(n) ) / diag(i)
return
end.

```

A.2.3. Programa TRIDCSIS

Para utilizar esta subrutina en la solución de un sistema arbitrario, con matriz tridiagonal cíclica y de diagonal dominante, se puede emplear el siguiente programa principal.

```

C      Programa TRIDCSIS para la solución de un sistema
C      de ecuaciones lineales con matriz tridiagonal cíclica
C      y de diagonal dominante. El sistema se resuelve por el
C      método de Gauss sin pivoteo, utilizando la subrutina
C      TRIDC.
C      La matriz se lee por diagonales y al final se leen los
C      elementos de la esquina superior derecha e inferior
C      izquierda respectivamente.
C      la solución del sistema se guarda en el vector I
integer i, n
parameter (nmax,=50 )
real a(nmax), b(nmax), c(nmax), d(nmax), trab(nmax)
print *, 'Solución de un sistema lineal con '
print *, 'matriz tridiagonal cíclica '
print *, '      '
print *, 'orden del sistema'
read (*,*) n
print *, 'Lectura de la matriz por diagonales '
print *, '      '
print *, 'diagonal'
do 10 i=1, n
    write (*, 5) i
10    read (*,*) d(i)
5    format ( 5x,'d(', i2,')=')
    print *, 'subdiagonal '
do 20 i=2, n
    write(*, 15) i
20    read (*,*) a(i)

```

```

15 format ( 5x, 'a(', i2, ')=' )
   print *, 'supradiagonal'
   do 30 i=1, n-1
       write (*, 25) i
30     read (*, *) c(i)
25 format ( 5x, 'c(', i2, ')=' )
   print *, 'esquina superior derecha '
   read (*, *) a (1)
   print *, 'esquina inferior izquierda '
   read (*, *) c(n)
   print *, 'lado derecho'
   do 40 i=1, n
       write (*, 35) i
40     read (*, *) b(i)
35 format ( 5x, 'b(', i2, ')=' )
   call Tride (a, d, c, trab, b, n )
   print *, ' La solución del sistema es '
   write (*, 50) (i, b(i), i=1, n)
50 format (i5, e15.7)
   stop
   end

```

Ejemplo.

Resolver el sistema de ecuaciones lineales

$$\begin{array}{rcccccc}
 3x_1 & -2x_2 & & & +x_5 & = & -3 \\
 x_1 & +2x_2 & -0.5x_3 & & & = & 2 \\
 & x_2 & -3x_3 & +x_4 & & = & 0 \\
 & & -2x_3 & +4x_4 & +x_5 & = & 1 \\
 x_1 & & & +x_4 & +3x_5 & = & -2
 \end{array}$$

Como la matriz del sistema es tridiagonal cíclica y de diagonal dominante por filas, para resolver el mismo basta con utilizar el programa que acabamos de presentar.

Los resultados de la corrida de este programa en una microcomputadora IBM-AT son los siguientes

```
1  -0.4918033  E+00
2  0.1126464  E+01
3  0.4777518  E+00
4  0.3067916  E+00
5  0.783372   E+00
```

A.3. Solución de Sistema de Ecuaciones con matriz simétrica y definida positiva

Consideramos la solución del sistema lineal

$$Ax = b,$$

cuando A es una matriz de orden n simétrica y definida positiva, como la matriz que aparece en sistema (3.4.13) para cálculo de los coeficientes del spline cúbico de suavizamiento.

Recordemos que una matriz A se dice definida positiva si para todo vector x , no nulo

$$x^T Ax > 0.$$

Las matrices definidas positivas tienen todos los valores propios positivos, como muestra el siguiente teorema

Teorema A.2.

Sea A una matriz de orden n simétrica y definida positiva entonces todos los valores propios de A son positivos.

Demostración.

Sea λ un valor propio de A , entonces existe un vector x^* no nulo tal que

$$Ax^* = \lambda x^*. \quad (\text{A.15})$$

Por otro lado como A es definida positiva y $x^* \neq 0$ sabemos que

$$x^{*T} Ax^* > 0, \quad (\text{A.16})$$

luego de (A.15) y (A.16) se obtiene que

$$0 < x^{*T} Ax^* = (x^{*T}) \lambda x^* = \lambda x^{*T} x^* = \lambda \|x^*\|_2^2 \quad (\text{A.17})$$

donde $\|x^*\|_2 = (\sum_{i=1}^n (x_i^*)^2)^{1/2}$,

como $x^* \neq 0$, $\|x^*\|_2^2 > 0$ y por lo tanto de (A.17) resulta que $\lambda > 0$.

Por otro lado, es conocido que el determinante de una matriz A es el producto de sus valores propios. Luego, si A es definida positiva el teorema A.3 nos permite concluir inmediatamente que A es no singular. Esto nos garantiza en particular, que el sistema de ecuaciones (3.4.13) tiene solución única y en consecuencia el spline cúbico de suavizamiento existe y es único. ■

Los sistemas de ecuaciones lineales con matriz simétrica y definida positiva se pueden resolver de manera eficiente por el método de Cholesky, que no es más que la versión del método de Gauss para este tipo de sistema y que se basa en el siguiente resultado

Teorema A.4.

Sea A una matriz real, simétrica y definida positiva, Entonces existe una única matriz triangular superior R .

$$\begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix}$$

con elementos positivos en la diagonal, tal que

$$A = R^T R. \quad (\text{A.18})$$

Desarrollando el producto $R^T R$ que aparece en (A.18), se obtiene inmediatamente que si a_{jj} está en la diagonal de A , entonces

$$a_{jj} = r_{1,j}^2 + r_{2,j}^2 + \dots + r_{j,j}^2 \quad (\text{A.19})$$

mientras que si $a_{k,j}$ es un elemento de A por encima de la diagonal, entonces

$$a_{k,j} = r_{1,k}r_{1,j} + r_{2,k}r_{2,j} + \dots + r_{k,k}r_{k,j}. \quad (\text{A.20})$$

Utilizando las expresiones (A.19) y (A.20) se pueden calcular todos los elementos del triángulo superior de R por columnas, es decir en el orden $r_{11}, r_{12}, r_{22}, \dots, r_{1n}, r_{2n}, \dots, r_{nn}$.

A.3.1. Algoritmos

Algoritmo para efectuar la descomposición $R^T R$ de una matriz simétrica y definida positiva.

Para cada $j = 1, \dots, n$ calcular

Para $k = 1, \dots, j - 1$

$$r_{k,j} = (a_{k,j} - \sum_{i=1}^{k-1} r_{i,k} r_{i,j}) / r_{k,k}$$

$$r_{j,j} = (a_{j,j} - \sum_{k=1}^{j-1} r_{k,j}^2)^{1/2}.$$

Nótese que los elementos de la matriz A por debajo de la diagonal no aparecen involucrados en el algoritmo, debido a que A es una matriz simétrica. Por otro lado gracias a la simetría de A , sólo se necesita guardar los elementos del triángulo superior, que ocupan $\frac{1}{2}n(n+1)$ localizaciones de memoria. Una vez que se calcula la matriz R , como $A = R^T R$ el sistema de ecuaciones original es equivalente a

$$R^T R x = b. \quad (A.21)$$

A su vez la solución de (A.21) se obtiene resolviendo los sistemas

$$R^T y = b, \quad (A.22)$$

$$R x = y. \quad (A.23)$$

en ese orden, con la ventaja de que ambos sistemas tienen matrices triangulares y por lo tanto sus soluciones se calculan de forma sencilla, mediante los procesos conocidos como sustitución hacia adelante y hacia atrás respectivamente. En efecto, como R^T es una matriz triangular inferior, de la primera ecuación de (A.22) resulta que

$$y_1 = b_1 / r_{1,1}.$$

Sustituyendo y_1 en la segunda ecuación podemos despejar y_2 y obtenemos entonces

$$y_2 = (b_2 - r_{1,2} y_1) / r_{2,2}$$

En general, conocemos y_1, \dots, y_{k-1} , de la K -ésima ecuación de (A.22) se obtiene

$$y_k = (b_k - \sum_{i=1}^{k-1} r_{i,k} y_i) / r_{k,k}$$

Después que se ha resuelto el sistema (A.22) disponemos del lado derecho para el sistema (A.23) y en este caso, como R es una matriz triangular superior, de la última ecuación se obtiene inmediatamente que

$$x_n = y_n / r_{nn}$$

y en general, conocidos $x_n, x_{n-1}, \dots, x_{k+1}$ de la K -ésima ecuación resulta que

$$x_k = (y_k - \sum_{i=k+1}^n r_{k,i} x_i) / r_{k,k}$$

Si se observa con detenimiento el proceso que hemos descrito, es evidente que en el primer paso la solución y del sistema (A.22) se puede guardar en el propio vector b y de manera similar, en el segundo paso, la solución x del sistema (A.23) se puede guardar sobre b , de modo que no hace falta reservar memoria adicional para los vectores x y y .

El algoritmo para calcular la solución del sistema (una vez que se tiene la factorización de Cholesky de la matriz) se puede resumir entonces de la siguiente forma :

Sustitución hacia adelante

Para $k = 1, \dots, n$ calcular

$$b_k = (b_k - \sum_{i=1}^{k-1} r_{i,k} b_i) / r_{k,k}$$

(En este momento b contiene la solución y del sistema (A.22))

Sustitución hacia atrás

Para $k = n, \dots, 1$ calcular

$$b_k = b_k / r_{k,k}$$

Para $i = 1, \dots, k-1$ calcular

$$b_i = b_i - b_k r_{i,k}$$

(En este momento b contiene la solución x del sistema (A.23))

Si la matriz A además de ser simétrica y definida positiva es una matriz banda, con m diagonales por encima de la diagonal principal (como la que aparece en el sistema (3.4.13) para el cálculo de los coeficientes del spline cúbico de suavizamiento, donde $m = 2$), entonces la matriz R de la factorización de Cholesky, sólo tiene elementos no nulos en las m diagonales por encima de la principal.

Esto nos permite ahorrar localizaciones de memoria para almacenar tanto A como R . Por ejemplo, si A es una matriz pentadiagonal simétrica de orden 7, es decir A tiene la siguiente estructura

$$A = \begin{pmatrix} 11 & 12 & 13 & & & & 0 \\ 12 & 22 & 23 & 24 & & & \\ 13 & 23 & 33 & 34 & 35 & & \\ & 24 & 34 & 44 & 45 & 46 & \\ & & 35 & 45 & 55 & 56 & 57 \\ & & & 46 & 56 & 66 & 67 \\ 0 & & & & 57 & 67 & 77 \end{pmatrix}$$

entonces los elementos del triángulo superior de A , se pueden guardar en una matriz ABD de dimensión 3×7 , de modo que las columnas de A coinciden con las de ABD y las diagonales de A se guardan en las filas de ABD , es decir

$$ABD = \begin{pmatrix} * & * & 13 & 24 & 35 & 46 & 57 \\ * & 12 & 23 & 34 & 45 & 56 & 67 \\ 11 & 22 & 33 & 44 & 55 & 66 & 77 \end{pmatrix}$$

Los elementos señalados con * en ABD nunca se utilizan porque corresponden a elementos que no existen en la matriz A .

Suponiendo que m es la cantidad de diagonales de la matriz simétrica A , por encima de la diagonal principal, el siguiente segmento de programa traslada las $m+1$

diagonales del triángulo superior de A , a las filas de la matriz ABD

```

do 20 j=1, n
  il= max 0 (i, j-m)
  do 10 i=il, j
    k=i-j + m + 1
    abd(k, j)=a(i, j)
10  continue
20  continue

```

Gracias a ese tipo de almacenamiento, si $m < n/2$ entonces las subrutinas que trabajan con matrices de banda definidas positivas, pueden utilizar menos localizaciones de memoria, que las subrutinas preparadas para matrices definidas positivas pero llenas y pueden utilizar menos tiempo si $m < n/\sqrt{3}$. Claro que los resultados numéricos no varían, aunque los elementos se van guardando en posiciones diferentes.

A.3.2. Subrutinas y Funciones

Subrutina SBANDP

La subrutina SPBFA, que presentamos a continuación pertenece al paquete LINPACK y realiza la factorización de Cholesky de una matriz simétrica, de banda y definida positiva.

Los parámetros de entrada de la subrutina SPBFA son los siguientes

<i>ABD</i>	matriz de dimensión (<i>lda</i> , <i>n</i>) que contiene en sus filas, las diagonales del triángulo superior de la matriz simétrica <i>A</i> . que se quiere factorizar
<i>lda</i>	número de filas de la matriz <i>ABD</i> . Se debe cumplir que $lda \geq m + 1$
<i>n</i>	orden de la matriz <i>A</i>
<i>m</i>	número de diagonales de <i>A</i> por encima de la diagonal principal. Se debe verificar que $0 \leq m \leq n$.

Los parámetros de salida de esta subrutina son

<i>ABD</i>	que contiene las bandas de la matriz triangular <i>R</i> , tal que $R^T R$
------------	--

La factorización de Cholesky no se realiza hasta el final si *INFO* no es cero

<i>INFO</i>	es un entero que vale 0, si <i>A</i> es definida positiva y en consecuencia la factorización de Cholesky se realiza exitosamente, y vale <i>k</i> si la submatriz principal de orden <i>k</i> de <i>A</i> no es definida positiva. En este caso la factorización de Cholesky no se realiza
-------------	---

Subrutine SPBFA

Subroutine SPBFA (ABD, lda, n, m, INFO)

```

C
C   Utiliza las subrutinas y funciones siguientes
C   BLAS SDOT
C   FORTRAN MAX0, SQRT
      real sdot, t
      real s
      integer ik, j, jk, k, mu
C
      do 30 j=1, n
         info = j
         s = 0.0e0
         ik = m+1
         jk = max0( j-m, 1 )
         mu = max0( m+2-j, 1 )
         if ( m .lt. mu ) goto 20
         do 10 k=mu, m
            t = abd(k,j)- sdot(k-mu, abd(ik, jk), 1, abd(mu, j))
            t= t / abd(m+1, jk)
            abd(k, j) = t
            s = s + t * t
            ik = ik - 1
            jk = jk + 1
10          continue
20          continue
            s = abd ( m+1, j) - s
            if (s .le. 0.0e0) goto 40
            abd(m+1, j) = sqrt(s)
30          continue
         info = 0
40          continue
      return
      end

```

Para emplear la Subrutina APBFA, en la solución del sistema lineal

$$Ax = b$$

necesitamos otra subrutina, que una vez calculada la factorización de Cholesky $R^T R$ de la matriz A , utilice la misma para hallar la solución del sistema.

Subrutina SPBSL

La subrutina SPBSL, que ofrecemos a continuación, también pertenece al paquete LINPACK y usa la factorización de Cholesky construida por la subrutina SPBFA, para resolver el sistema lineal $Ax = b$.

Los parámetros de entrada de la subrutina SPBSL son los siguientes

<i>ABD</i>	matriz de dimensión (<i>lda</i> , <i>n</i>) que contiene factorización de Cholesky de la matriz A , calculada mediante SPBFA
<i>lda</i>	número de filas de la matriz ABD
<i>n</i>	orden de la matriz A y número de elementos del vector b
<i>m</i>	número de diagonales de A por encima de la diagonal principal
<i>b</i>	arreglo de dimensión n que contiene el lado derecho del sistema $Ax = b$

A la salida de la subrutina SPBSL, el vector b contiene la solución del sistema $Ax = b$

Subroutine SPBSL (*ABD*, *lda*, *n*, *m*, *b*)
 Integer *lda*, *n*, *m*
 real *abd* (*lda*, 1), *b*(1)

C

C

Utiliza las subrutinas y funciones siguientes

```

C      BLAS saxpy, sdot
C      FORTRAN min0
      real sdot, t
      integer k, kb, la, lb, lm
C      Resolver  $\text{trans}(R) * y = b$ 
C
      do 10 k=1, n
          lm = min0(k-1, m)
          la = m+1 - lm
          lb = k - lm
          t = sdot(lm, abd(la,k), 1, b(lb), 1)
          b(k) = (b(k) - t) / abd(m+1, k)
10      continue
C
C      Resolver  $R * x = y$ 
          t = t / abd(m+1, k)
C      do 20 kb=1, n
          k = n+t - kb
          lm = min0(k-1, m)
          la = m+1 - lm
          ib = k - lm
          b(k) = b(k) / abd(m+1, k)
          T = -3(k)
          Call saxpy (lm, t, abd(la, k), 1, b(lb), 1)
20      continue
      return
      end

```

Función SDOT y Subrutina SAXPY

Tanto la subrutina SPBFA como la subrutina SPBSL utilizan la función SDOT, que pertenece a la biblioteca BLAS, también contenida en el LINPACK. Esta función calcula el producto escalar de dos vectores. Además la subrutina SPBSL utiliza la subrutina SAXPY de la biblioteca BLAS, para sumar a un vector un múltiplo de

otro vector. A continuación ofrecemos los listados de ambas funciones.

```

real function SDOT(n, sx, incx, sy, incy)
C
C      Calcula el producto escalar de dos vectores
real sx(1), sy(1), stemp
integer i, incx, incy, ix, iy, m, mp1, n
C
      stemp = 0.0e0
      sdot = 0.0e0
      if (n .le. 0) return
      if (incx .eq. 1 .and. incy .eq. 1) goto 20
C      Caso de incrementos desiguales o incrementos
C      iguales pero diferentes de 1
C
      ix = 1
      iy = 1
      if (incx .it. 0) ix = (-n+1) incx+1
      if (incy .it. 0) iy = (-n+1) incy +1
      do 10 i=1, n
          stemp = stemp + sx(ix) sy(iy)
          ix = ix + incx
          iy = iy + incy
10      continue
      sdot = stemp
      return
C
C      Caso en que ambos incrementos son iguales a 1
C
20      m = mod (n, 5)
      if (m .eq. 0) goto 40
      do 30 i=1, m
          stemp = stemp + sx(i) sy(i)
30      continue
      if (n .it. 5) goto 60

```

```

40      mp1 = m+1
      do 50 i=mp1, n, 5
          stemp = stemp + sx(i) sy(i) + sx(i+1) sy(i+1)
*          sx(i+2) sy(i+2) + sx(i+3) sy(i+3) + sx(i+4) sy(i+4)
50      continue
60      sdot = stemp
      return
      end

```

Subroutine SAXPY(n, sa, sx, incx, sy, incy)

Suma a un vector un múltiplo de otro

C
C

```

      real sx(1) , sy(1) , sa
      integer i, incx, incy, ix, iy, m, mp1, n

```

C

```

      if (n .le. 0 ) return
      if (sa .eq. 0.0 ) return
      if ( incx .eq. 1 .and. incy .eq. q ) goto 20

```

C

Caso de incrementos desiguales o incrementos
iguales pero diferentes de 1

C

C

```

      ix=1
      iy= 1
      if (incx .lt. 0 ) ix= (-n +1) incx+1
      if (incy .lt. 0 ) iy= (-n+1) incy +1
      do 10 i=1,n
          sy(iy)= sy(iy) + sa sx (ix)
          ix=ix + incx
          iy=iy + incy

```

10

```

      continue
      return

```

C

Caso en que ambos incrementos son iguales a 1

C

C

20

```

      m= mod(n, 4)

```

```

if ( m .eq. 0 ) goto 40
do 30 i=1, m
sy(i)= sy(i) + sa  sx(i)
30  continue
if ( n .lt. 4) return
40  mp1=m+1
do 50 i=mp1, n, 4
sy(i) = sy(i) + sa  sx(i)
sy(i+1)= sy(i+1) + sa  sx(i+1)
sy(i+2) = sy(i+2) + sa  sx(i+2)
sy(i+3) = sy(i+3) + sa  sy(i+3)
50  continue
return
end

```

Programa SBANDP

Finalmente presentamos al lector un programa que utiliza las subrutinas SPBFA y SPBSL para resolver el sistema lineal $Ax = b$, cuando A es una matriz simétrica, de banda y definida positiva

```

C      Programa SBANDP para resolver un sistema de
C      ecuaciones con matriz de banda definida
C      positiva
parameter (nmax=50)
real a(nmax,nmax), b(nmax), abd(nmax,nmax)
integer lda, n, m, k, c
print *, 'Matrices de banda definidas positivas '
print *, ''
print *, 'orden de la matriz'
read (*,2) n
2      format (i3)
print *, 'numero de diagonales por encima'
print *, 'de la diagonal principal '

```

```

read (*,2) m
do 5 i=1, n
    do 5 j=1, n
        a(i,j)=0
5  continue
k=m+1
do 14 i=1, n
    do 15 j=i, k
        write(*,7) i, j
        read (*,6) a(i,j)
        a(j,i)=a(i,j)
15  continue
if (k .it. n) k= k+1
14  continue
7  format (1x, 'A(', i2, ', ', i2, ')=')
6  format (f10.5)
print *, 'si deseas cambiar los datos de entrada '
print *, 'diga cuantos, de lo contrario de c=0 '
print *, '
print *, 'c='
read (*,2) c
if ( c .eq. 0 ) goto 35
do 19 k=1, c
print *, 'teclea los indices del elemento a cambiar '
write (*,24)
24  format (1x, 'i=')
read (*,2) i
write(*,26)
26  format (1x, 'j=')
read(*,2) j
write(*,7) i, j
read (*,6) a(i,j)
19  continue
35  do 120 j=1, n
    i1= max0(1, j-m)

```

```

do 130 i=i1, j
    k=i-j+m+1
    abd(k,j)= a(i,j)
130     continue
120     continue
C     LDA se utiliza para reservar memoria
    lda=50
C
C     Factorización de Cholesky de la matriz A
C
    call spbfa(abd, lda, n, info)
    if (info .ne. 0) goto 150
    print *, 'b es el vector de terminos independientes'
    do 30 i=1, n
        write(*,21) i
        read(*,6) b(i)
30     continue
21     format (1x, 'b(', i2, ')=' )
C
C     Solución del sistema a partir de la factorización
    print *, '
    call spbsl(abd, lda, n, m, b)
    print *, 'S O L U C I O N '
    print *, '
    do 40 i=1, n
        write(*,16) b(i)
40     continue
16     format (5x, f10.6)
    stop
150    print *, 'La matriz de entrada no es definida positiva'
    stop
end

```

Ejemplo.

Resolver el siguiente sistema de ecuaciones utilizando el programa SBANDP

$$\begin{pmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 23 \\ 32 \\ 33 \\ 31 \end{pmatrix} \quad (A.24)$$

Nótese que la matriz del sistema (A.24) es densa pero se puede considerar también que es de banda con ($m = 3$) bandas por encima de la diagonal principal. Además esta matriz es simétrica y definida positiva. La siguiente corrida muestra los resultados del programa obtenidos en una microcomputadora IBM-AT.

Corrida de prueba

Matrices de banda definidas positivas

Orden de la matriz 4

numero de diagonales por encima

de la diagonal principal

A(1,1)=5.0

A(1,2)=7.0

A(1,3)=6.0

A(1,4)=5.0

A(2,2)=10.

A(2,3)=8.0

A(2,4)=7.0

A(3,3)=10.

A(3,4)=9.0

A(4,4)=10.

Si desea cambiar los datos de entrada

diga cuantos, de lo contrario de c=0

c=0

b es el vector de terminos independientes

b(1)= 23.0

b(2)=32.0

b(3)=33.0

b(4)=31.0

SOLUCION

0.999902

1.000060

1.000025

0.999985

Ejemplo.

Resolver el sistema de ecuaciones lineales, que nos permite calcular los coeficientes del spline cúbico de suavizamiento, del ejemplo de la sección 3.6 con $p=0$.

Utilizando las expresiones de Q^T y D calculamos la matriz $6Q^T D^2 Q$ del sistema (3.4.13), que como ya sabemos es simétrica, pentadiagonal, definida positiva y de orden 19. La diagonal principal de esta matriz es el siguiente vector (con 4 cifras decimales):

(144.0	144.0	144.0	144.0	143.9999	144.0
144.0	144.0	144.0	143.9999	143.9999	144.0001
144.0001	143.9999	143.9999	143.9999	144.0001	144.0001
143.9998)					

La primera diagonal por encima de la principal tiene todos los elementos iguales a -96.0 (con 4 cifras decimales) salvo los elementos 12 y 17 que valen -96.00001. En la segunda diagonal por encima de la principal todos los elementos valen 24.0. El lado derecho del sistema (3.4.13), es decir $Q^t y$, es un vector que tiene todas las componentes nulas salvo la 12, 13 y 14 que valen 20.0002, -40.0002 y 20.0 respectivamente.

El sistema lineal

$$6(Q^T D^2 Q)u = Q^T y$$

se resolvió con el programa SBANDP en una microcomputadora IBM-AT, obteniéndose los siguientes resultado (con cifras decimales):

(-0.007251	-0.024993	-0.056465	-0.104906	-0.173557	-0.265656
-0.3844445	-0.533163	-0.715041	-0.933347	-1.19298	-1.839128
-1.402176	-0.689453	-0.420197	-.213362	-0.072208)	

Como vimos en la sección 3.4, el vector u nos permite calcular los coeficientes del spline cúbico de suavizamiento mediante (1.4.13), así como el valor del $S(f_0)$ a través de (3.4.15).

R E F E R E N C I A S

- [Ahl] Ahlberg, J.H., Nilson, E.N., and Walsh, J.L. [1967]. "The Theory of Splines and their Applications", New York, Academic Press .
- [And] Andersson, L. and Elfving, T. [1991]. "Interpolation and Approximation by monotone cubic splines ", J. Approx. Theory , 66 302-333.
- [Beh] Behforooz, G.H., and Papamichael, N. [1979]. " End conditions for cubic spline interpolation.", J. Inst. Maths. Appls., 23 367-372.
- [Bon] Boneva, L I., Kendall, D., and Stefanov, I., [1971]. " Spline transformation: three new diag aids for the statistical data analysis", J. of the Royal Statistical Society, Series B, 33, 1.
- [Boo1] De Boor, C. [1978]. " A practical guide to splines", Springer Verlag. New York.
- [Boo2] De Boor C., and Swartz B. [1977]. "Piecewise monotone interpolation", J. Approx. Theo., 21, 411-416.
- [Cli] Cline, A. K. [1974]. " Scalar and planar-valued curve fitting using splines under tension", Communications of the ACM, 7 , 4, 218-220. .
- [Con] Conte, S. D., and De Boor, C. [1980]. " Elementary Numerical Analysis", Mc Graw-Hill.
- [Cra] Craven, P., and Wahba, G., [1979]. " Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation", Numer. Math., 31, 377-403.
- [Dah] Dahlquist, G., and Bjorck, A. [1974]. " Numerical Methods", Pretince-Hall.
- [Dau] Dauner, H., and Reinsch, C. [1989]. " An analysis of two algorithms for shape preserving cubic spline interpolation", J. Numer. Anal. 9, 299-314.
- [Don] Dongarra, J.J., Bunch, J. R., and Stewart, G. W. [1978]. " LINPACK User Guide", SIAM Publications Philadelphia, P.A.

[For] Forsythe, G., and Moler, C.B. [1967]. "Computer solution of linear algebraic systems", Prentice-Hall, INC.

[Fri] Fritsch, F., and Carlson, R. [1980]. " Monotone piecewise cubic interpolation", SIAM J. Numer. Anal., 17, 238-246.

[Gam] Gamder, H. [1979]. " Choice of an optimal shape parameter when smoothing data", Communications in Statistics: Theory and Methods, 14 , 1425-1435.

[Gol] Golub, G., Heath, M., and Wahba, G. [1979]. " Generalized cross validation as a method for choosing a good ridge parameter", Technometrics, 21 , 215-224.

[Ign] Ignatov, M. I., and Pevnyi, A. B. [1990]. " Reinsch's method for solving the smoothing problem in several dimensions", U.R.S.S. Comput. Maths. Math. Phys., 30 , 1, 137-142.

[Kam] Kammerer, W.J., and Reddien, G.W. [1972]. " Local convergence of smooth cubic spline interpolates", SIAM J. Numer. Anal., 9 687-694.

[Luc] Lucas, T.R. [1974]. " Error bounds for interpolating cubic splines under various end conditions", SIAM J. Numer. Anal., 11, 569-584.

[Lyc] Lyche, T., and Shumaker, L. [1973]. " Computation of smoothing and interpolating natural splines via local bases", SIAM J. Numer. Anal., 10, 6, 1027-1038.

[Mc] Allister, D. Mc., Pawsow, E., and Roulier, J. A. [1977]. " Algorithms for computing shape preserving spline interpolation to data", Math. Comp., 31, 715-725.

[Nie] Nielson, G. [1974]. " Some piecewise polynomial alternatives to splines under tension", Computer Aided Design, Academic Press, New York 209-236.

[Pow] Powell, M. J. D. [1981]. " Approximation Theory and Methods", Cambridge University Press .

[Pre] Prenter, P. M. [1975]. " Splines and Variational Methods", Wiley, New York .

[Pru1] Pruess, S. [1979]. " Alternatives to the exponential spline in tension", Math. Comp. 33, 1273-1281.

- [Pru2] Pruess, S. 1989. "Splines and their applications", Comunicaciones Internas, Departamento de Matemáticas, Facultad de Ciencias, UNAM Vinculos Matemáticos 169.
- [Rei] Reinsch, C. M. [1971]. "Smoothing by spline functions", Numer. Math. 16, 451-454.
- [Ren] Renka, R. [1987]. "Interpolatory tension splines with automatic selection of tension factors", SIAM J. Sci. and Stat. Comp., 8, 393-415.
- [Sch] Schweikert, D. [1966]. "An interpolation curve using splines in tension", J. Math. Phys. 45, 312-317.
- [Shk] Shumaker, L. [1981]. "Spline functions: Basic Theory", Wiley, New York
- [Shl] Shumilov, B.M. [1990]. "Recursive Interpolation by cubic splines with additional nodes", U.R.S.S. Compt. Maths. Math. Phys., . 30, 1, 132-137.
- [Sil] Silverman, B. W. [1974]. "A fast and efficient cross-validation method for smoothing parameter choice in spline regression", J. of the Amer. Stat. Assoc., . 79, 584-589.
- [SpΞ] SpΞth, H. [1974]. "Spline Algorithms for Curves and surfaces", Utilitas Mathematical Publ. Inc., Winnipeg, Manitoba.
- [Utr] Utreras, Diaz F. [1979]. "Cross validation techniques for smoothing spline functions in one or two dimensions", "Smoothing Techniques for Curve Estimation", Berlin: Springer-Verlag, 196-232.
- [Wah1] Wahba, G. [1975]. "Smoothing noisy data with spline functions", Numer. Math. 24, 383-393.
- [Wah2] Wahba, G. [1976]. "A survey of some smoothing problems and the method of generalized cross-validation for solving them", Applications of Statistics, North Holland, 507-524.
- [Wah3] Wahba, G. [1979]. "How to smooth curves and surfaces with splines and cross-validation", Proceedings of the 24th Conference of design of experiments, U.S. Army Research Office, Report 79-2.

[Wah4] Wahba, G., and Wendelberger J. [1980]. "Some new mathematical methods for variational object analysis using splines and cross validation", *Monthly Weather Review*, 108, 36-57.

[Wah5] Wahba, G., and Wold S. [1975]. "A completely automatic french curve: Fitting spline functions by cross-validation", *Communications in Statistics*, 4, 1-17.

[Wah6] Wahba, G., and Wold S. [1975]. "Periodic splines for spectral density estimation: the use of cross-validation for determining the degree of smoothing", *Communications in Statistics*, 4, 125-141.

[Weg] Wegman, E. J., and Wright, I. W. [1983]. "Splines in Statistics", *J. of the Amer. Stat. Assoc.*, . 78, 382-395.

[Woo] Woodford, C. H. [1970]. "An algorithm for data smoothing using spline functions", *BIT* 10 505-510.