UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

2

FACULTAD DE CIENCIAS

UNAM

SPLINES DE AJUSTE CON NODOS LIBRES Y UN METODO PARA ESTIMACION DE PARAMETROS EN MODELOS DEFINIDOS POR ECUACIONES DIFERENCIALES ORDINARIAS

TESIS QUE PARA OBTENER EL GRADO
DE MAESTRO EN CIENCIAS (MATEMÁTICAS)
PRESENTA:

MAT. ERNESTO OLVERA SOTRES.

MÉXICO, D.F.

1986

P R O L O G O.

En este trabajo estudiamos el siguiente problema, muy frecuente en la ciencia y en la técnica: dado un conjunto de puntos, que pueden ser mediciones experimentales y que responden a una relación funcional, la cual a su vez es solución de una cierta ecuación diferencial en la que aparecen parámetros, determinar los parámetros de tal manera que la solución, con ciertas condiciones iniciales, sea la que mejor se ajuste a los datos.

Este es un problema que ha merecido la atención de la gente y diferentes soluciones se han dado. Cierto tipo de enfoques hacen énfasis en resolver la ecuación diferencial y optimizar los parámetros sobre la solución, lo cual crea muchas dificultades pues es necesario calcular la matriz Jacobiana de la solución con respecto a los parámetros como variables, y esto es un problema difícil.

El método que estudiamos es un método directo por el cual sólo es necesario resolver un problema de suma mínima de cuadrados para obtener los parámetros óptimos. La idea central es primeramente ajustar los datos por medio de un spline cúbico, dejando los nodos variables para un mejor - ajuste, y posteriormente, a través de resolver un problema de suma mínima de cuadrados, determinar los parámetros sobre la misma ecuación diferencial, sin resolverla.

Hemos tratado de abordar de manera amplia el problema de ajuste de splines con nodos libres. Por ello presentamos en el capítulo I la teoría de splines y B-splines, lo que proporciona las bases para los capítulos II y III en los que estudiamos el problema de ajuste de datos con splines cúbicos y ajuste de datos con splines cúbicos con nodos libres. Cuando se permite a los nodos ser libres y se determinan las posiciones óptimas de ellos se logran ajustes que con mucho mejoran los ajustes con nodos fijos. Damos ejemplos de esto y mencionamos algunas dificultades que se presentan.

En el capítulo IV estudiamos el método para estimación de parámetros en ecuaciones diferenciales ordinarias. Dicho método es más general que esto, pues puede ser aplicado a otros tipos de ecuaciones funcionales, por ejemplo, ecuaciones integrales.

En el capítulo V hemos recopilado los ejemplos que hemos trabajado. Hacemos algunas observaciones sobre ellos y mostramos resultados. Y en el capítulo VI exponemos algunos de los programas elaborados por nosotros.

Finalmente quiero agradecer sinceramente al Dr. Pablo Barrera su orientación generosa y desinteresada en la elaboración de este trabajo.

ERNESTO OLVERA SOTRES

Agosto de 1986.

I N D I C E

CAPITULO I

SPLINES

1.1 Definición de splines y dimensión del espacio.

En una buena parte de este trabajo vamos nosotros a tratar con las funciones conocidas con el nombre de splines. Son funciones cuya gran utilidad se ha venido reconociendo cada vez más a partir de los años 60's, aún cuando ya Schoenberg había llamado la atención sobre ellas por el año de 1946. Nosotros vámos a utilizar splines para auxiliarnos en la solución de nuestro problema central que es el de estimación de parámetros en ecuaciones diferenciales.

Antes del primer artículo de Schoenberg, algunos matemáticos habían ya mencionado los splines, aunque no con su nombre actual. Pero es el mismo Schoenberg quien más ha contribui

do al desarrollo de la teoría de los splines. Las propiedades, al mismo tiempo simples y poderosas de estas funciones hacen de ellas una herramienta muy útil para ingenieros, científicos y actuarios. Los matemáticos, en forma natural se han interesado en estas funciones y a la fecha se tiene una amplia teoría bien desarrollada. En este capítulo nos proponemos mencionar algunas definiciones y propiedades importantes de los splines que nos serán de utilidad posteriormente.

Consideremos el intervalo $[a, b]$ y un conjunto de números reales $\Pi = \{\xi_0, \xi_1, \ldots, \xi_n\}$ con $a = \xi_0 < \xi_1 < \ldots < \xi_n = b$ en este intervalo.

Definición. El espacio vectorial lineal de los splines de grado k, con nodos $\xi_1, \xi_2, \ldots, \xi_{n-1}$, es el conjunto de las funciones $s(t)$ que satisfacen las siguientes dos condiciones

1) en cada uno de los intervalos $[\xi_{i-1}, \xi_i]$, $i = 1, \ldots, n$, $s(t)$ es un polinomio de grado $\leqslant k$;

2) la función $s(t)$ y sus primeras $(k - 1)$ derivadas son continuas en $[a, b]$.

Por ejemplo, para $k = 3$, tenemos el espacio de los splines cúbicos, que son funciones de clase C^2 en $[a, b]$ y que en cada subintervalo son polinomios de grado $\leqslant 3$.

La verificación de que, en general, los splines de grado k forman un espacio vectorial lineal es inmediata. Este espacio lo designaremos con $S(k, \xi_0, \xi_1 \ldots, \xi_n)$ o simplemente $S(k, \Pi)$. Vamos a investigar su dimensión.

A) En cada intervalo $[\xi_i, \xi_{i+1}]$, $i = 0, \ldots, n - 1$, la función $s(t)$ es de la forma

$$\sum_{\ell=1}^{k+1} a_{i,\ell}\, t^{\ell-1}, \quad i = 0, \ldots, n-1 .$$

La condición para que $s(t)$ sea continua equivale a las siguientes $(n-1)$ ecuaciones:

$$\sum_{\ell=1}^{k+1} a_{i,\ell}\, \xi_{i+1}^{\ell-1} = \sum_{\ell=1}^{k+1} a_{i+1,\ell}\, \xi_{i+1}^{\ell-1}, \quad i = 0, \ldots, n-2 .$$

La j-ésima derivada del polinomio $\sum_{\ell=1}^{k+1} a_{i,\ell} t^{\ell-1}$ es

$$\sum_{\ell=1}^{k+1-j} \frac{(j+\ell-1)!}{(\ell-1)!}\, a_{i,j+\ell}\, t^{\ell-1} .$$

Por lo que la continuidad de la j-ésima derivada en cada uno de los nodos interiores $\xi_1, \xi_2, \ldots, \xi_{n-1}$ significa

$$\sum_{\ell=1}^{k+1-j} \frac{(j+\ell-1)!}{(\ell-1)!}\, a_{i,j+\ell}\, \xi_{i+1}^{\ell-1} = \sum_{\ell=1}^{k+1-j} \frac{(j+\ell-1)!}{(\ell-1)!}\, a_{i+1,j+\ell}\, \xi_{i+1}^{\ell-1},$$

$$i = 0, \ldots, n-2$$

Como j varía desde 1 hasta $(k-1)$, tenemos $(k-1)(n-1)$ ecuaciones más. Con las $(n-1)$ anteriores, hacen un total de $(n-1)k$ ecuaciones con $n(k+1)$ incógnitas.

Ahora, si escribimos las ecuaciones en un orden adecuado en la forma $Ax = 0$, A resulta una matriz escalonada con $(n-1)k$ renglones linealmente independientes y $n(k+1)$

columnas. Por ejemplo, para $k = 3$ la matriz A tiene la siguiente forma

$$
\begin{bmatrix}
1 & \xi_1 & \xi_1^2 & \xi_1^3 & -1 & -\xi_1 & -\xi_1^2 & -\xi_1^3 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
 & 1 & 2\xi_1 & 3\xi_1^2 & 0 & -1 & -2\xi_1^2 & -3\xi_1^2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
 & & 2 & 6\xi_1 & 0 & 0 & -2 & -6\xi_1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
 & & & & 1 & \xi_2 & \xi_2^2 & \xi_2^3 & -1 & -\xi_2 & -\xi_2^2 & -\xi_2^3 & 0 & \dots & 0 & 0 & 0 & 0 \\
 & & & & & 1 & 2\xi_2 & . & . & . & . & . & . & & & & & \\
 & & & & & & 2 & . & . & . & . & . & . & & & & & \\
 & & & & & & & & & 1 & . & . & & & & & & \\
 & & & & & & & & & & . & & & & & & \\
 & & & & & & & & & & & . & & & & & \\
 & & & & & & & & & & & & . & & & & & \\
 & & & & & & & & 1 & \xi_{n-1} & \xi_{n-1}^2 & \xi_{n-1}^3 & -1 & -\xi_{n-1} & -\xi_{n-1}^2 & -\xi_{n-1}^3 \\
 & & & & & & & & & 1 & 2\xi_{n-1} & 3\xi_{n-1}^2 & 0 & -1 & -2\xi_{n-1} & -3\xi_{n-1}^2 \\
 & & & & & & & & & & 2 & 6\xi_{n-1} & 0 & 0 & -2 & -6\xi_{n-1}
\end{bmatrix}
$$

En general, como el número de renglones $(n - 1)k$ es menor que el número de columnas $n(k + 1)$, el rango de la matriz es $(n - 1)k$. Y así, la dimensión de su núcleo es

$$dim\ (N(A)) = [n(k+1)] - [(n-1)k]$$

$$= n + k.$$

Lo que significa que la dimensión de $S(k, \Pi)$ es igual a $(n + k)$.

B) En esta segunda demostración de que $dim\ S(k, \Pi) = n + k$, vamos a utilizar una importante expresión mediante la cual un spline $s(t)$ puede expresarse de manera única como una suma de un polinomio de grado k y una combinación de funciones de potencia truncada. Esta expresión para $s \in S(k, \Pi)$ es la siguiente

$$s(t) = \sum_{j=0}^{k} c_j t^j + \frac{1}{k!} \sum_{j=1}^{n-1} d_j (t - \xi_j)_+^k ,$$

donde la función potencia truncada se define como

$$x_+^k = \begin{cases} x^k, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

Un poco más abajo vamos a demostrar esta fórmula para $s(t)$. Aceptándola momentáneamente, observemos que cada función spline queda caracterizada por los parámetros c_j, $j = 0,1,\ldots, k$, y d_j, $j = 1, 2,\ldots, n - 1$. Así, la dimensión es $(n + k)$. Q.E.D. !

Lo que acabamos de demostrar, lo podemos decir de otra manera:

$$dim\ S(k,\ \Pi) = (\text{núm. de nodos}) + k + 1.$$

Por ejemplo, la dimensión del espacio de splines cúbicos con nodos $\xi_1, \xi_2, \ldots, \xi_{n-1}$ es

$$dim\ S(3,\ \xi_0,\ \xi_1, \ldots, \xi_n) = n + 3.$$

Ahora vamos a demostrar que cualquier spline puede representarse de la manera que hemos dicho. Para ello vamos a necesitar el siguiente

Lema. Sea $s(t)$ una función spline de grado k y ξ_i uno de sus nodos. Suponiendo que en el intervalo (ξ_{i-1}, ξ_i) la función $s(t)$ está dada por el polinomio $p_k(t)$ y por el polinomio $q_k(t)$ en el intervalo (ξ_i, ξ_{i+1}), entonces

$$q_k(t) - p_k(t) = c(t - \xi_i)^k,$$

donde c es constante.

Demostración. Por la definición de spline los polinomios $p_k(t)$ y $q_k(t)$ tienen el mismo valor en $t = \xi_i$, e igualmente sus derivadas de órdenes $1, 2, \ldots, k - 1$. En

consecuencia, el polinomio de grado k, $q_k(t) - p_k(t)$ tiene un cero en $t = \xi_i$ de multiplicidad k; por lo tanto

$$q_k(t) - p_k(t) = c(t - \xi_i)^k,$$

donde c es una constante.

En seguida vamos a demostrar la fórmula que habíamos aceptado provisionalmente sin demostración. Dada la importancia de tal fórmula la enunciamos en forma de teorema.

Teorema. Cualquier función spline $s(t)$ en $S(k, \Pi)$ tiene una expresión de la forma

$$s(t) = \sum_{j=0}^{k} c_j t^j + \frac{1}{k!} \sum_{j=1}^{n-1} d_j (t - \xi_j)_+^k ,$$

donde los coeficientes d_j son constantes. Esta expresión es única.

Demostración. Escribamos en forma ligeramente diferente la fórmula que queremos demostrar:

$$s(t) = \sum_{j=0}^{k} c_j t^j + \sum_{j=1}^{n-1} d_j' (t - \xi_j)_+^k .$$

En el intervalo $[\xi_0, \xi_1]$ la función $(t - \xi_j)_+^k$ es cero

para $j = 1, \ldots, n-1$. En este intervalo el spline es un polinomio de grado a lo más k, este polinomio es precisamente $\sum_{j=0}^{k} c_j t^j$, la primera parte de nuestra fórmula. Apliquemos ahora el lema anterior: sea $q(t)$ el polinomio de grado k que restringido a $[\xi_1, \xi_2]$ da $s(t)$. Entonces, por el lema

$$q(t) = \sum_{j=0}^{k} c_j t^j + d_1'(t - \xi_1)^k,$$

donde d_1' es una constante. Aplicando otra vez el lema, encontramos que en el intervalo $[\xi_2, \xi_3]$ el spline $s(t)$ está dado por un polinomio de grado k de la forma

$$\sum_{j=0}^{k} c_j t^j + d_1'(t - \xi_1)_+^k + d_2'(t - \xi_2)_+^k .$$

Repitiendo este proceso llegamos a la expresión

$$s(t) = \sum_{j=0}^{k} c_j t^j + \sum_{j=1}^{n-1} d_j'(t - \xi_j)^k, \quad \xi_{n-1} \leq t \leq \xi_n$$

Pero $(t - \xi_j)_+^k = 0$ para $t \geq \xi_j$, y así se puede escribir la expresión

$$s(t) = \sum_{j=0}^{k} c_j t^j + \sum_{j=1}^{n-1} d_j'(t - \xi_j)_+^k \quad \ldots\ldots(*)$$

válida en todo el intervalo $[\xi_0, \xi_n]$.

A continuación, al mismo tiempo que demostraremos la segunda parte del teorema, la unicidad de la fórmula, vamos a encontrar los valores de los coeficientes d_j', y sustituyendo en (*) se obtiene la fórmula del teorema.

En primer lugar, observemos que para $t \leqslant \xi_1$, la fórmula (*) se reduce a $s(t) = \sum_{j=0}^{k} c_j t^j$. Por lo tanto, este polinomio coincide con el único polinomio de grado k mediante el cual está definido el spline en el intervalo $[\xi_0, \xi_1]$.

Para terminar la demostración observemos lo siguiente: la k-ésima derivada de $s(t)$ no es necesariamente continua en $[\xi_0, \xi_n]$; puede tener discontinuidades en los nodos. En el nodo ξ_j el salto de discontinuidad, llamémosle d_j, está dado por:

$$d_j = s^{(k)}(\xi_j + 0) - s^{(k)}(\xi_j - 0)$$

$$= k!(c_k + \sum_{i=1}^{j} d_i') - k!(c_k + \sum_{i=1}^{j-1} d_i')$$

$$= k!\, d_j' .$$

Los coeficientes d_j de la fórmula, repitámoslo, son los saltos de discontinuidad de la k-ésima derivada del spline

$s(t)$, lo que demuestra la unicidad. Con esto el teorema ha quedado demostrado.

1.2 B-splines

Ahora, en forma natural, nos encontramos en la necesidad de encontrar una base $\{\phi_1(t), \phi_2(t), \ldots, \phi_{n+k}(t)\}$ adecuada del espacio $S(k, \Pi)$ y poder escribir cualquier spline $s \in S(k, \Pi)$ como

$$s(t) = \sum_{j=1}^{n+k} \alpha_j \phi_j(t) \quad , \quad a \leq t \leq b.$$

Se ha demostrado que las bases

$$\{\phi_j(t) = (t - \xi_j)_+^k \ , \ a \leq t \leq b, \ j = 1, 2 \ldots, n - 1\}$$

$$\{\phi_j(t) = t^{j-n}, \ a \leq t \leq b, \ j = n, n + 1, \ldots, n + k\}$$

son computacionalmente inadecuadas.

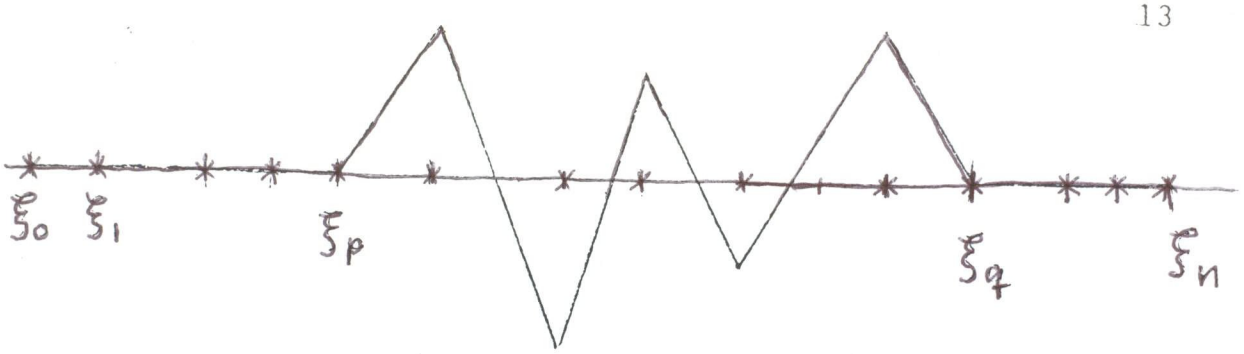
Nuestro propósito ahora es introducir los "B-splines". Una base de B-splines es particularmente adecuada, pues al hacer las computaciones se evitan cancelaciones y pérdida de exactitud, y es adecuada también desde un punto de vista teórico.

Schoenberg mostró cómo mediante una aplicación del teo

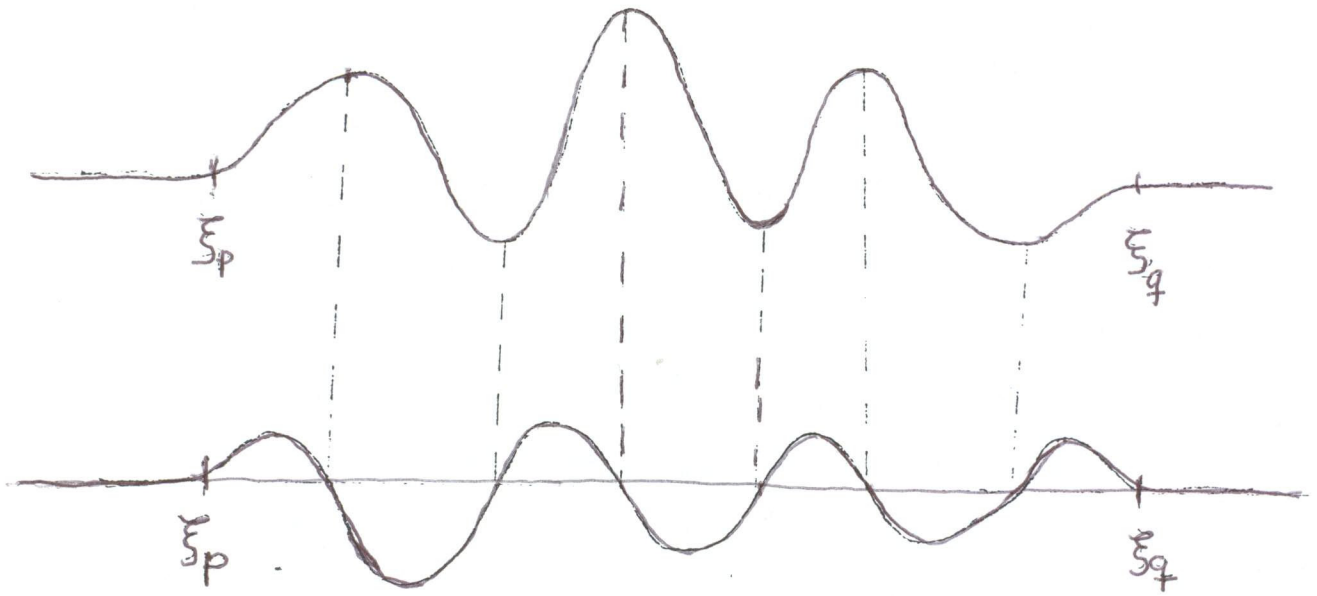
rema de Peano surgen los B-splines de una manera muy natural. Por el momento, nosotros podemos pensar que para comodidad en los cálculos una importante propiedad que deberíamos pedir a las funciones básicas que estamos buscando sería la de que fuesen funciones de soporte mínimo, es decir funciones que no se anulan solamente en un "pequeño intervalo".

Además, hagamos las siguientes consideraciones antes de proponer una definición formal. Para un spline $s(t)$, es decir un elemento de $S(k, \Pi)$ tratemos de contar el número de veces que $s(t)$ se puede anular en un intervalo (ξ_p, ξ_q) donde p y q son enteros de manera que $0 < p < q < n$. Para precisar, supongamos que $s(t)$ es idénticamente cero en $[\xi_0, \xi_p]$ y en $[\xi_q, \xi_n]$, que r es el número de ceros de $s(t)$ en (ξ_p, ξ_q) y que r es finito.

Para empezar consideremos el caso $k = 1$. Es decir, $s(t)$ está compuesto de segmentos lineales. Puesto que $s(\xi_p)$ y $s(\xi_q)$ son cero, el número máximo de ceros que $s(t)$ puede tener en el intervalo (ξ_p, ξ_q) es $(q - p - 2)$. Es decir, el número de cambios de signo que esta función spline $s(t)$ puede tener en este intervalo es $\leq (q - p - 2)$.



Ahora, como la derivada de un spline de grado k es un spline de un grado más bajo con los mismos nodos y como el spline $s(t)$ tiene r ceros en (ξ_p, ξ_q) la primera derivada deberá cambiar al menos $(r + 1)$ veces de signo.



Continuando hasta la $(k - 1)$ derivada, ésta deberá cambiar de signo al menos $(r + k - 1)$ veces. Pero esta

derivada es un spline de grado uno. Tenemos entonces la siguiente desigualdad

$$r + k - 1 \leq q - p - 2.$$

Esto nos da una cota para el número de ceros

$$r \leq q - (p + k + 1).$$

Así que, si consideramos un spline de esta forma, permitiéndole que en algún punto del intervalo $[\xi_p, \xi_{p+k+1}]$ no sea cero, no podrá anularse en ningún punto de este intervalo ya que $q = p + k + 1$, lo que obliga a r ser cero. En realidad, un spline $s(t)$ del tipo que estamos considerando, que no sea idénticamente cero debe tener, como demostraremos, al menos $(k + 2)$ nodos. Esto es lo que aclara el entrecomillado que usamos anteriormente cuando formulamos el deseo de pedir que las funciones básicas no se anularan solamente en un "pequeño intervalo".

Tomando en cuenta las anteriores afirmaciones e introduciendo ciertos coeficientes o factores de normalización convenientes, hacemos la siguiente definición de un B-spline de grado k;

$$B_p(t) = \sum_{j=p}^{p+k+1} \left[\prod_{\substack{i=p \\ i \neq j}}^{p+k+1} \frac{1}{(\xi_i - \xi_j)} \right] (t - \xi_j)_+^k , \qquad -\infty < t < \infty$$

El sub-índice p en $B_p(t)$ sirve para recordar que $B_p(t)$ no es idénticamente cero únicamente en el intervalo (ξ_p, ξ_{p+k+1}).

Efectivamente, si $t < \xi_p$ entonces $B_p(t) = 0$ porque $(t - \xi_j)_+^k = 0$ para toda j en $[p, p + k + 1]$. Si $t > \xi_{p+k+1}$ entonces el spline está dado por el polinomio

$$B_p(t) = \sum_{j=p}^{p+k+1} \left[\prod_{\substack{i=p \\ i \neq j}}^{p+k+1} \frac{1}{(\xi_i - \xi_j)} \right] (t - \xi_j)^k .$$

Si para simplificar hacemos

$$\beta_j = \prod_{\substack{i=p \\ i \neq j}}^{p+k+1} \frac{1}{(\xi_i - \xi_i)} , \qquad j = p, \ldots, p + k + 1,$$

entonces el polinomio anterior se escribe

$$B_p(t) = \sum_{j=p}^{p+k+1} \beta_j (t - \xi_j)^k , \qquad t > \xi_{p+k+1},$$

y los coeficientes β_j satisfacen las siguientes ecuaciones

$$\sum_{j=p}^{p+k+1} \beta_j \, \xi_j^r = 0 , \qquad r = 0, 1, \ldots, k. \qquad (*)$$

como puede demostrarse (1).

---

(1) Una demostración de esto la podemos hacer apelando a la fórmula de

Ahora, desarrollaremos los binomios $(t - \xi_j)^k$ y reagrupando términos nos queda

$$B_p(t) = \sum_{r=0}^{k} [(-1)^r \binom{k}{r} t^{k-r} \sum_{j=p}^{p+k+1} \beta_j \xi_j^r],$$

y por las anterioes ecuaciones (*) se obtiene

$$B_p(t) = 0,$$

para todo $t > \xi_{p+k+1}$.

La definición de B-splines que hemos dado se debe a Schoenberg y aquí estamos nosotros tratando de mostrar que es una buena definición. En primer lugar ya hicimos ver que un B-spline se anula idénticamente en los intervalos

---

interpolación de Lagrange, la cual cuando se aplica a los polinomios x^i, $i = 0, 1, \ldots, n$ da como resultado

$$\sum_{k=0}^{n} x_k^i \ell_k(x) = x^i ,$$

donde $\ell_k(x)$ son los polinomios básicos de Lagrange, $\ell_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{n} (x - x_j)/(x_k - x_j)$. Entonces, substituyendo esta definición de $\ell_k(x)$ en la fórmula de interpolación de esta nota y considerando los coeficiencients de x^n, obtenemos la identidad

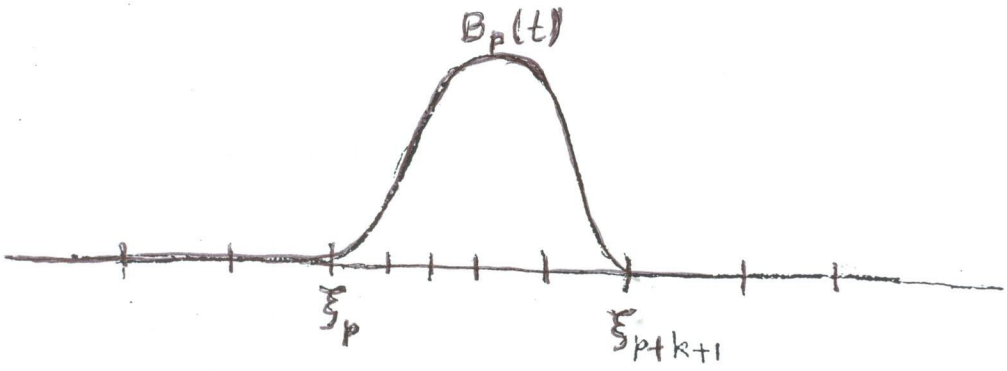
$$\sum_{k=0}^{n} \frac{x_k^i}{\prod_{j=0}^{n} (x_k - x_j)} = \delta_{in}, \qquad i = 0, 1, \ldots, n.$$

$-\infty < t \leqslant \xi_p$ y $\xi_{p+k+1} \leqslant t < \infty$. En lo que sigue vamos a mostrar otras útiles propiedades de los B-splines: 1) el B-spline es positivo en el intervalo (ξ_p, ξ_{p+k+1}); 2) un B-spline que no sea el idénticamente cero, en el intervalo donde no se anula no puede tener menos de $(k + 2)$ nodos, que son los que existen desde ξ_p hasta ξ_{p+k+1}.

Para proceder, observemos que, si $t \in (\xi_p, \xi_{p+1})$, entonces

$$B_p(t) = \frac{1}{(\xi_{p+1}-\xi_p)(\xi_{p+2}-\xi_p)\ldots(\xi_{p+k+1}-\xi_p)} (t - \xi_0)^k$$

que es un número positivo. Así pues, en el intervalo (ξ_p, ξ_{p+k+1}) el B-spline no es idénticamente cero. Sea r el número de veces que $B_p(t)$ se anula en dicho intervalo. Ya habíamos visto que r está acotado y que se cumple la desigualdad $r \leqslant q - (p + k + 1)$. Con $q = p + k + 1$, queda $r = 0$. Y así, no hay ningún cero dentro del intervalo (ξ_p, ξ_{p+k+1}). En otras palabras, el B-spline $B_p(t)$, como lo hemos definido, es siempre positivo dentro del intervalo (ξ_p, ξ_{p+k+1}). Esto es, hemos demostrado que todos los B-splines son positivos donde no se anulan. En resumen, tienen la siguiente forma



Con ingenuo humor Schoenberg los bautizó con el nombre de B-splines, porque "B-splines are bell-shaped". Nosotros nos podemos quedar con la B y decir, los B-splines son una base del espacio $S(k, \Pi)$.

Por último, ya sabemos que un B-spline de grado k no se anula en el intervalo (ξ_p, ξ_{p+k+1}); contando los extremos son $(k + 2)$ nodos. Como ya lo habíamos anunciado, aquí vamos a demostrar que no pueden ser menos. Supongamos que son r nodos: $\xi_p, \xi_{p+1}, \ldots, \xi_{p+r-1}$, y supongamos que existe un B-spline apoyado en estos nodos, el cual tiene que ser de la forma

$$\sum_{j=p}^{p+r-1} \alpha_j (t - \xi_j)_+^k$$

donde los coeficientes α_j han de satisfacer las condiciones

$$\sum_{j=p}^{p+r-1} \alpha_j \, \xi_j^r = 0, \qquad r = 0, 1, \ldots, k.$$

Designemos con H a la matriz de dimensiones $(k+1) \times r$

$$H = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \xi_p & \xi_{p+1} & \xi_{p+2} & \cdots & \xi_{p+r\ 1} \\ \xi_p^2 & \xi_{p+1}^2 & \xi_{p+2}^2 & \cdots & \xi_{p+r\ 1}^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \xi_p^k & \xi_{p+1}^k & \xi_{p+2}^k & \cdots & \xi_{p+r-1}^k \end{bmatrix}$$

Si $r \leq k+1$, H es de rango r, porque tiene un menor de orden r que es distinto de cero, el determinante cuyos renglones y columnas son los r primeros renglones y columnas de H, y que es el conocido determinante de Vandermonde de orden r, cuyo valor es

$$\prod_{i>j} (\xi_i - \xi_j).$$

Sea α el vector cuyas componentes son $\alpha_p, \ldots, \alpha_{p+r-1}$. Entonces, las condiciones sobre los coeficientes α_j pueden describirse mediante la ecuación

$$H \alpha = 0.$$

Pero, esta ecuación implica que las columnas de H son linealmente dependientes, o, en otras palabras, su rango es menor que r. Puesto que ya hemos demostrado que esto no puede ser si $r \leq k + 1$, se sigue que r es al menos $(k + 2)$.

1.3 La base de B-splines.

Ya hemos demostrado algunas de las propiedades importantes de los B-splines y a pesar de que mencionamos que constituyen una base del espacio $S(k, \Pi)$, esto último no lo hemos demostrado. Ahora lo haremos. Ya sabemos que el espacio $S(k, \Pi)$ es de dimensión $(n + k)$. Ya también mencionamos la conveniencia de que las funciones básicas sean B-splines. Pero, dados los punto $\xi_0, \xi_1, \ldots, \xi_n$, el conjunto $\{B_p : p = 0, 1, \ldots, n - k - 1\}$ de que por el momento podemos disponer consta de sólo $(n - k)$ elementos. Se requieren, al menos, $2k$ elementos más. Una manera conveniente de obtenerlos es agregando k nodos arbitrarios $\xi_{-k} < \xi_{-k+1} < \ldots < \xi_{-1}$ a la izquierda de ξ_0 y k nodos arbitrarios $\xi_{n+1} < \xi_{n+2} < \ldots < \xi_{n+k}$ a la derecha de ξ_n. Disponemos ahora del conjunto $\{B_p : p = -k, -k + 1, \ldots, n-1\}$ que consta de $(n + k)$ elementos. Todo lo que tenemos que demostrar para poder afirmar que estas funciones son una base del espacio $S(k, \Pi)$, es que son linealmente independientes. Para ello vamos a seguir el método tradicional. Es

decir, tomemos una combinación lineal

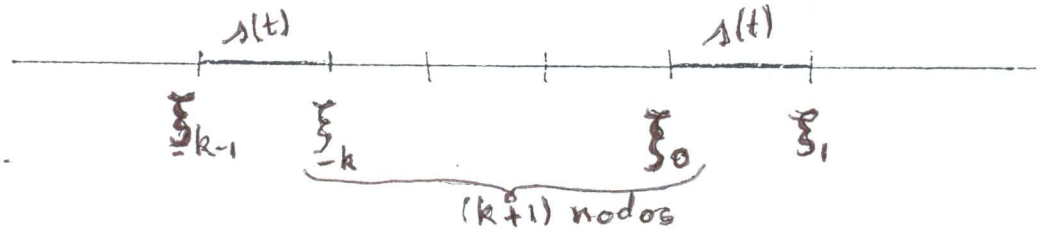
$$s(t) = \sum_{p=-k}^{n-1} \alpha_p B_p(t) ,$$

y suponiendo que este spline es idénticamente cero en $[a,b]$, vamos a demostrar que se implica que todos los coeficientes α_p son cero necesariamente.

Primero demostraremos que $s(t)$ también es cero en $[\xi_{-k}, \xi_0]$. Agreguemos un nodo auxiliar $\xi_{-k-1} < \xi_{-k}$ y consideremos el spline

$$s(t) = \sum_{p=-k}^{n-1} \alpha_p B_p(t), \quad \xi_{-k-1} \leq t \leq \xi_1$$

el cual es cero en $[\xi_{-k-1}, \xi_{-k}]$ (por ser $t \leq \xi_{-k}$) y en $[\xi_0, \xi_1]$, por la suposición que hicimos sobre $s(t)$. Veamos la figura



Desde ξ_{-k} hasta ξ_0 sólo hay $(k + 1)$ nodos, esto significa que no hay suficientes nodos para que pueda ser distinto de cero. Y así, $s(t) = 0$ en $[\xi_{-k}, b]$.

Ahora, supongamos que no todos los coeficientes

$\{\alpha_p : p = -k, -k+1, \ldots, n-1\}$ son cero. Sea q el entero más chico tal que α_q es distinto de cero. Si t es tal que $\xi_q < t < \xi_{q+1}$, entonces el spline está dado por

$$s(t) = \alpha_q B_q(t).$$

Pero esto no es cero, pues habíamos visto que el B-spline es positivo en tal intervalo, con lo que llegamos a una contradicción. En consecuencia todos los coeficientes α_p son cero y el teorema ha quedado demostrado.

1.4 Una relación de recurrencia para los B-splines.

Recordemos la definición

$$B_p^k(t) = \sum_{j=p}^{p+k+1} \left[\prod_{\substack{i=p \\ i\neq j}}^{p+k+1} \frac{1}{(\xi_i - \xi_j)} \right] (t - \xi_j)_+^k ,$$

que es la misma de antes, pero en la notación hemos agregado un super-índice k para enfatizar que el B-spline es de grado k. Para hacer cálculos, usar esta definición es erróneo computacionalmente. Algo más adecuado es usar la siguiente fórmula que nos da el valor de un B-spline de grado k a partir de los valores de B-splines de grado $k-1$:

Para $k = 1$ y $t \in [\xi_i, \xi_{i+1}]$:

$$B_j^1(t) = 0, \quad j \neq i - 1, \quad j \neq i$$

$$B_{i-1}^1(t) = (\xi_{i+1} - t) \,/\, [(\xi_{i+1} - \xi_{i-1})(\xi_{i+1} - \xi_i)]$$

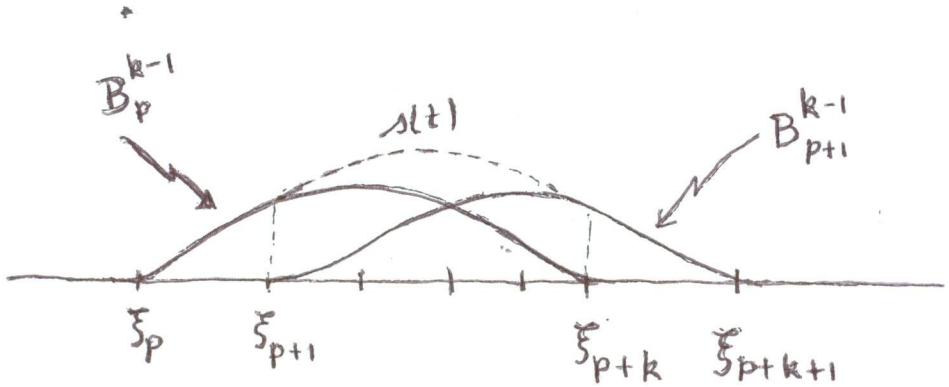
$$B_i^1(t) = (t - \xi_i)/[(\xi_{i+1} - \xi_i)(\xi_{i+2} - \xi_i)] .$$

Para $k > 1$:

$$B_p^k(t) = \frac{(t - \xi_p)\, B_p^{k-1}(t) + (\xi_{p+k+1} - t)\, B_{p+1}^{k-1}(t)}{\xi_{p+k+1} - \xi_p}$$

y todo valor de t.

Demostración. Sea $s(t)$ el segundo miembro de la última ecuación escrita. La función $s(t)$: $-\infty < t < \infty$, es polinomial por pedazos, cada polinomio de grado a lo más k y los puntos de unión son los nodos $\{\xi_j : j = p, p + 1, .., p + k + 1\}$. Por la definición de B-spline, esta función es idénticamente cero para $t \leq \xi_p$ y para $t > \xi_{p+k+1}$.



Cuando t está en el intervalo $[\xi_p, \xi_{p+1}]$ la definición de $B_p^k(t)$ implica la identidad

$$B_p^k(t) = \frac{(t - \xi_p)}{(\xi_{p+k+1} - \xi_p)} B_p^{k-1}(t).$$

Lo que demuestra que $s(t) = B_p^k(t)$ en $[\xi_p, \xi_{p+1}]$. Para demostrar que también se cumple $\{s(t) = B_p^k(t) : \xi_j \leq t \leq \xi_{j+1}\}$ para $j = p + 1, p + 2, \ldots, p + k$ será suficiente demostrar que el cambio en s en los nodos ξ_j: $j = p + 1, p + 2, \ldots, p + k$ coincide con el cambio en B_p^k. Es decir, que la modificación en ambas fórmulas es la misma al pasar al intervalo $[\xi_j, \xi_j + 1]$, $j = p + 1, \ldots, p + k$. Tomemos al entero $\ell \in [p + 1, p + k]$. Para $t \leq \xi_\ell$,

$$B_p^k(t) = \sum_{j=p}^{\ell-1} \left[\prod_{\substack{i=p \\ i \neq j}}^{p+k+1} \frac{1}{(\xi_i - \xi_j)} \right] (t - \xi_j)^k .$$

Al pasar al intervalo $[\xi_\ell, \xi_{\ell+1}]$ la fórmula se modifica con el sumando

$$\left[\prod_{\substack{i=p \\ i \neq \ell}}^{p+k+1} \frac{1}{(\xi_i - \xi_\ell)} \right] (t - \xi_\ell)^k .$$

Para investigar el cambio en $s(t)$ partimos de su definición y tendremos:

Para $t \leqslant \xi_\ell$

$$s(t) = \frac{(t - \xi_p)\, B_p^{k-1}(t) + (\xi_{p+k+1} - t) B_{p+1}^{k-1}(t)}{\xi_{p+k+1} - \xi_p}$$

$$= \frac{(t-\xi_p) \sum_{j=p}^{\ell-1} \left[\prod_{\substack{i=p \\ i\neq j}}^{p+k} \frac{1}{(\xi_i - \xi_j)} \right] (t-\xi_j)^{k-1} + (\xi_{p+k+1} - t) \sum_{p+1}^{\ell-1} \left[\prod_{\substack{i=p+1 \\ i\neq j}}^{p+k+1} \frac{1}{(\xi_i - \xi_j)} \right] (t-\xi_j)^{k-1}}{\xi_{p+k+1} - \xi_p}$$

Por lo que al pasar al intervalo $[\xi_\ell, \xi_{\ell+1}]$ la modificación es

$$\frac{(t-\xi_p) \left[\prod_{\substack{i=p \\ i\neq \ell}}^{p+k} \frac{1}{(\xi_i - \xi_\ell)} \right] (t-\xi_\ell)^{k-1} + (\xi_{p+k+1} - t) \left[\prod_{\substack{i=p+1 \\ i\neq \ell}}^{p+k+1} \frac{1}{(\xi_i - \xi_\ell)} \right] (t-\xi_\ell)^{k-1}}{\xi_{p+k+1} - \xi_p}$$

que es el polinomio $(t - \xi_\ell)^{k-1} / (\xi_{p+k+1} - \xi_p)$ multiplicado por el factor

$$(t - \xi_p) \prod_{\substack{i=p \\ i\neq\ell}}^{p+k} \frac{1}{(\xi_i - \xi_\ell)} + (\xi_{p+k+1} - t) \prod_{\substack{i=p+1 \\ i\neq\ell}}^{p+k+1} \frac{1}{(\xi_i - \xi_\ell)}$$

$$= \left[(t-\xi_p)(\xi_{p+k+1} - \xi_\ell) + (\xi_{p+k+1} - t)\ (\xi_p - \xi_\ell) \right] \prod_{\substack{i=p \\ i\neq\ell}}^{p+k+1} \frac{1}{(\xi_i - \xi_\ell)}$$

$$= (t - \xi_\ell)\ (\xi_{p+k+1} - \xi_p) \prod_{\substack{i=p \\ i\neq\ell}}^{p+k+1} \frac{1}{(\xi_i - \xi_\ell)}$$

Por lo que los cambios en $s(t)$ y $B_p^k(t)$ son los mismos al pasar al intervalo $[\xi_\ell, \xi_{\ell+1}]$ y así $s(t) = B_p^k(t)$ para todo valor real de t, con lo que hemos demostrado la fórmula de recurrencia.

1.5 Teorema de Schoenberg-Whitney.

Utilizando la desigualdad que ya anteriormente hemos demostrado

$$r \leq q - (p + k + 1),$$

que acota el número de ceros que un spline puede tener en el

intervalo (ξ_p, ξ_q) se puede demostrar el siguiente importante teorema que nos permite decidir cuándo existe una solución única en $S(k, \Pi)$ del problema de aproximación a una función $f(t)$, $t \in [a, b]$.

Teorema. (Schoenberg-Whitney)

Supongamos dados los números $\{\xi_j : j = -k, -k+1, \ldots, n+k\}$ con un orden estrictamente creciente y, para $p = -k, -k+1, \ldots, n-1$, consideremos los B-splines $\{B_p(t) : -\infty < t < \infty\}$. Sean $\{t_i : i = 1, 2, \ldots, n+k\}$ las abscisas de interpolación, también en un orden estrictamente creciente. Entonces, para cualquier función f, las ecuaciones

$$\sum_{p=-k}^{n-1} \alpha_p B_p(t_i) = f(t_i), \quad i = 1, 2, \ldots, n+k$$

tienen una única solución $\{\alpha_p : p = -k, -k+1, \ldots, n-1\}$, si y sólo si todos los números $\{B_{j-k-1}(t_j), j = 1, 2, \ldots, n+k\}$ son distintos de cero.

Demostración. Supongamos que $B_{j-k-1}(t_j)$ es cero. Entonces una de las dos siguientes desigualdades vale: $t_j \leq \xi_{j-k-1}$ ó $t_j \geq \xi_j$. En el primer caso $B_p(t)$ es cero si $p \geq j-k-1$ y $t \leq t_j$. En consecuencia las primeras j ecuaciones del teorema tienen la forma

$$\sum_{p=-k}^{j-k-2} \alpha_p B_p(t_i) = f(t_i), \quad i = 1,2,\ldots, j.$$

como estas j ecuaciones contienen $(j - 1)$ incógnitas, no existirá solución para cualesquiera segundos miembros.

Similarmente, en el segundo caso, $t_j \geqslant \xi_j$, entonces las últimas $(n + k + 1 - j)$ ecuaciones tienen la forma

$$\sum_{p=j-k}^{n-1} \alpha_p B_p(t_i) = f(t_i) \quad i = j, j + 1,\ldots, n + k,$$

y nuevamente el número de incógnitas es insuficiente. Por lo tanto, las condiciones

$$B_{j-k-1}(t_j) \neq 0, \quad j = 1, 2,\ldots, n + k,$$

son necesarias para que el sistema tenga una solución para cualquier f.

Por otra parte, las ecuaciones del teorema no tienen una única solución si y sólo si existen parámetros $\{\alpha_p : p = -k, -k + 1,\ldots, n - 1\}$ no todos cero tales que la función

$$s(t) = \sum_{p=-k}^{n-1} \alpha_p B_p(t), \quad -\infty < t < \infty,$$

satisface las condiciones

$$s(t_i) = 0, \quad i = 1, 2, \ldots, n + k.$$

En este caso la función $s(t)$ que se acaba de definir no es idénticamente cero. Por lo tanto, para demostrar la unicidad será suficiente demostrar que las condiciones

$$B_{j-k-1}(t_j) \neq 0, \quad j = 1, 2, \ldots, n + k,$$

$$s(t) = \sum_{p=-k}^{n-1} \alpha_p B_p(t), \quad -\infty < t < \infty,$$

$$s(t_i) = 0, \quad i = 1, 2, \ldots, n + k,$$

implican que el spline $s(t)$ es idénticamente cero.

Supongamos que todas estas condiciones valen, pero que el spline no es cero. Entonces, en algún intervalo, dentro del intervalo $[\xi_{-k}, \xi_{n+k}]$ deberá ser finito el número de ceros de $s(t)$. En consecuencia existen nodos, ξ_p y ξ_q tales que $s(t)$ es idénticamente cero en $[\xi_{p-1}, \xi_p]$ y $[\xi_q, \xi_{q+1}]$, en tanto que en el intervalo abierto $s(t)$ tiene sólo un número finito de ceros, digamos r. Quizás fuese necesario introducir dos nodos extra: $\xi_{-k-1} < \xi_{-k}$ y $\xi_{n+k+1} > \xi_{n+k}$. Ahora observemos que los B-splines $\{B_j: j = p, p + 1, \ldots, q - k - 1\}$ toman valores distintos de cero únicamente si la variable t está en el intervalo (ξ_p, ξ_q). En consecuencia, las condiciones

$$B_{j-k-1}(t_j) \neq 0, \quad j = 1, 2, \ldots, n + k,$$

implican que los puntos $\{t_{j+k+1} : j = p, p + 1, \ldots, q - k - 1\}$ están todos en el intervalo (ξ_p, ξ_q). Pero como habíamos supuesto que $s(t_i) = 0$, $i = 1, 2, \ldots, n + k$, se sigue que el número de ceros en (ξ_p, ξ_q) es al menos $(q - p - k)$, contradiciendo la desigualdad ya demostrada anteriormente: $n \leq q - (p + k + 1)$. Y así, el teorema es verdadero.

1.6 El teorema de Peano.

Debemos mencionar el teorema de Peano, porque este es una base fundamental en la teoría de aproximación. Vamos primero a recordar algunas definiciones matemáticas y a introducir algo de notación.

Definición. Sea la función f definida en $[a, b]$. Si $\mathbb{P} = \{x_0, \ldots, x_n\}$ es una partición de $[a, b]$, sea $\Delta f_i = f(x_i) - f(x_{i-1})$. Definimos la variación total de f como el $\sup \sum_{i=1}^{n} |\Delta f_i|$, sobre todas las particiones de $[a,b]$. Y decimos que la función f es de variación acotada sobre $[a, b]$ si y sólo si su variación total es finita.

Definición. Decimos que una funcional lineal $\mathcal{L}$ aniquila plinomios hasta de grado k, si cada vez que f es un polinomio de grado menor o igual a k, $\mathcal{L}(f) = 0$.

Definición. Una funcional lineal es acotada si existe una costante $||\mathcal{L}||_\infty$ tal que se cumple

$$|\mathcal{L}(f)| \leq ||\mathcal{L}||_\infty \, ||f||_\infty, \quad f \in V[a, b]$$

donde $||f||_\infty$ es la norma

$$||f||_\infty = \sup_{a \leq x \leq b} |f(x)|, \quad f \in V[a, b].$$

El símbolo $f \in V[a, b]$ significa que f está en el espacio de las funciones de variación acotada.

Pronto vamos a utilizar la siguiente función s_θ de la variable t

$$s_\theta(t) = (t - \theta)_+^k \, , \; a \leq t \leq b,$$

donde θ es un número real fijo que no necesita estar en $[a, b]$. A esta función $s_\theta(t)$ le podemos aplicar el operador $\mathcal{L}$, $\mathcal{L}(s_\theta)$, que escribiremos

$$\mathcal{L}_t[(t - \theta)_+^k]$$

para remarcar que $\mathcal{L}$ se aplica a la función $(t - \theta)_+^k$ como función de t.

El teorema de Peano es el siguiente

Teorema. Sea k cualquier entero no-negativo y sea $\mathcal{L}$ una funcional lineal acotada definida en $V[a, b]$, que aniquila polinomios hasta de grado k y tal que la función $K(\theta)$, dada por

$$K(\theta) = \frac{1}{k!}\mathcal{L}_t[(t - \theta)_+^k], \quad a \leq \theta \leq b$$

es de variación acotada. Entonces, si $f \in C^{(k+1)}[a, b]$, la funcional $\mathcal{L}(f)$ tiene el valor

$$\mathcal{L}(f) = \int_a^b K(\theta) f^{(k+1)}(\theta)\, d\theta$$

Demostración. Utilizando el teorema de Taylor con residuo, podemos escribir, para $a \leq t \leq b$

$$f(t) = \sum_{j=0}^{k} \frac{(t - a)^j}{j!} f^{(j)}(a) + \frac{1}{k!}\int_a^t (t - \theta)^k f^{(k+1)}(\theta)\, d\theta,$$

y aplicar el operador $\mathcal{L}$:

$$\mathcal{L}(f) = \frac{1}{k!}\mathcal{L}_t\left[\int_a^b (t - \theta)_+^k f^{(k+1)}(\theta)\, d\theta\right].$$

Así pues, tenemos que demostrar que en esta ecuación los operadores $\mathcal{L}_t$ y la integral, conmutan. Llamemos $\eta(t)$ a la función

$$\eta(t) = \left| \int_a^b (t - \theta)_+^k \, f^{(k+1)}(\theta) d\theta - \frac{(b-a)}{m} \sum_{\ell=1}^{m} (t-\theta_\ell)_+^k f^{(k+1)}(\theta_\ell) \right|,$$

donde $\{\theta_\ell : \ell = 1, 2, \ldots, m\}$ son puntos de una partición de $[a, b]$. De lo que queremos hablar es de aproximar a la integral por medio de las sumas de Riemann lo cual tiene sentido debido a las condiciones de variación acotada en el enunciado del teorema y también al hecho de que la variación de la función $(t - \theta)_+^k$, $a \leqslant \theta \leqslant b$ es uniformemente acotada para toda $x \in [a, b]$. Así, para cualquier $\varepsilon > 0$, existen puntos $\{\theta_\ell : \ell = 1, 2, \ldots, m\}$ en $[a, b]$ tales que

$$\eta(t) < \varepsilon, \quad \forall \quad t \in [a, b],$$

y tales que

$$\left| \int_a^b K(\theta) f^{(k+1)}(\theta) d\theta - \frac{(b-a)}{m} \sum_{\ell=1}^{m} K(\theta_\ell) f^{(k+1)}(\theta_\ell) \right| \leqslant \varepsilon .$$

Puesto que $\mathcal{L}$ es lineal

$$\mathcal{L}_t \left[\sum_{\ell=1}^{m} (t - \theta_\ell)_+^k f^{(k+1)}(\theta_\ell) \right] = \sum_{\ell=1}^{m} \mathcal{L}_t [(t - \theta_\ell)_+^k] f^{(k+1)}(\theta_\ell)$$

$$= k! \sum_{\ell=1}^{m} K(\theta_\ell) f^{(k+1)}(\theta_\ell),$$

y se sigue de la exactitud de las sumas de Riemann que, si la siguiente ecuación no es cierta

$$\mathcal{L}_t[\int_a^b (t - \theta)_+^k f^{(k+1)}(\theta) d\theta] = k! \int_a^b K(\theta) f^{(k+1)}(\theta) d\theta$$

entonces la diferencia entre sus dos miembros es acotada por el número

$$|\mathcal{L}_t[\eta(t)]| + k!\varepsilon \leq (||\mathcal{L}||_\infty + k!)\varepsilon.$$

Puesto que ε puede ser arbitrariamente pequeño, la ecuación anterior es cierta. Y puesto que

$$\mathcal{L}(f) = \frac{1}{k!} \mathcal{L}_t[\int_a^b (t - \theta)_+^k f^{(k+1)}(\theta) d\theta]$$

se sigue la afirmación del teorema.

1.7 Una relación entre diferencias divididas y B-splines

Una aplicación del teorema de Peano permite establecer una muy notable relación entre diferencias divididas y B-splines. Las diferencias divididas aparecen en la teoría de interpolación polinomial. Si $\{x_i : i = 0, 1, \ldots, n\}$ son puntos distintos de un intervalo $[a, b]$ y f es una fun-

ción definida en $[a, b]$, existe un único polinomio p de grado $\leq n$ que satisface

$$p(x_i) = f(x_i), \quad i = 0, 1, \ldots, n$$

El coeficiente de x^n en tal polinomio es la n-ésima diferencia dividida de f con respecto a los argumentos $\{x_0, x_1, \ldots, x_n\}$. Es usual representar este número mediante $[x_0, x_1, \ldots, x_n]f$. Es muy conocida y fundamental la siguiente igualdad

$$[x_0, x_1, \ldots, x_n] f = \sum_{k=0}^{n} \frac{f(x_k)}{\prod_{\substack{j=0 \\ j \neq k}}^{n} (x_k - x_j)},$$

que muestra que la diferencia dividida es lineal respecto a los valores de la función f. También se sabe que la n-ésima diferencia dividida de un polinomio de grado $(n - 1)$ es cero. Esto es, si consideramos la diferencia dividida de orden n como un operador, este es lineal y aniquila polinomios de grado $(n - 1)$.

Ahora establecemos el siguiente teorema

Teorema. Si f es de clase $C^{(k+1)}[a, b]$, y si $\{t_i : i = 0, 1, \ldots, k+1\}$ es un conjunto de puntos distintos en $[a, b]$, entonces

$$[t_0, t_1, \ldots, t_{k+1}]f = \frac{1}{k!}\int_a^b B(\theta) f^{(k+1)}(\theta)d\theta,$$

donde B es el B-spline

$$B(\theta) = \sum_{i=0}^{k+1} [(\theta - t_i)_+^k / \prod_{\substack{j=0 \\ j\neq i}}^{k+1} (t_j - t_i)], \quad a \leq \theta \leq b.$$

Demostración. Por la igualdad escrita arriba del teorema

$$[t_0, t_1, \ldots, t_{k+1}]f = \sum_{i=0}^{k+1} [f(t_i) / \prod_{\substack{j=0 \\ j\neq i}}^{k+1} (t_i - t_j)]$$

$$= \mathcal{L}(f)$$

digamos. Por lo tanto el operador lineal sobre $V[a, b]$ es un operador lineal acotado y la función

$$K(\theta) = \frac{1}{k!}\mathcal{L}_t[(t - \theta)_+^k], \quad a \leq \theta \leq b,$$

es de variación acotada. Vemos que se cumplen las condiciones del teorema de Peano, y aplicándolo obtenemos

$$\mathcal{L}(f) = \int_a^b K(\theta) f^{(k+1)}(\theta) d\theta$$

donde $K(\theta)$ está dada por

$$K(\theta) = \frac{1}{k!} \sum_{i=0}^{k+1} [(t_i - \theta)_+^k \Big/ \prod_{\substack{j=0 \\ j \neq i}}^{k+1} (t_i - t_j)], \quad a \leq \theta \leq b.$$

Y vemos entonces que la relación para la diferencia dividida es verdadera si y sólo si el B-spline $B(\theta)$ es igual a $k!\, K(\theta)$. Utilizamos la siguiente identidad

$$(t_i - \theta)_+^k = (t_i - \theta)^k + (-1)^{k+1} (\theta - t_i)_+^k$$

substituyéndola en la anterior expresión para $K(\theta)$ para $i = 0, 1, \ldots, k+1$, lo que da

$$K(\theta) = \frac{1}{k!} \{\mathcal{L}_t[(t - \theta)^k] + B(\theta)\}, \quad a \leq \theta \leq b.$$

El término $\mathcal{L}_t[(t - \theta)^k]$ es cero por ser $(t - \theta)^k$ un polinomio de grado k en t. Con esto hemos demostrado el teorema.

Y tenemos entonces lo siguiente: el teorema de Peano nos dice que si tomamos la funcional $\mathcal{L}$ como la diferencia dividida

$$\mathcal{L}(f) = [t_0, t_1, \ldots, t_{k+1}]f$$

entonces esta funcional está dada por

$$[t_0, t_1, \ldots, t_{k+1}]f = \int_a^b K(\theta) f^{(k+1)}(\theta) d\theta$$

donde el kermel de Peano es

$$K(\theta) = \frac{1}{k!}[t_0, t_1, \ldots, t_{k+1}](t - \theta)_+^k, \qquad a \leq \theta \leq b,$$

aquí, como es usual $[t_0, \ldots, t_{k+1}](t - \theta)_+^k$ representa la diferencia dividida en los puntos $\{t_i : i = 0, 1, \ldots, k + 1\}$ de la función $(t - \theta)_+^k$.

Por su parte, el último teorema que acabamos de demostrar, establece la igualdad

$$[t_0, t_1, \ldots, t_{k+1}]f = \int_a^b \frac{1}{k!} B_0(\theta) f^{(k+1)}(\theta) d\theta$$

donde $B_0(\theta)$ es el B-spline con base en los nodos $\{t_0, t_1, \ldots, t_{k+1}\}$.

Obtenemos entonces la siguiente relación:

$$\frac{1}{k!} B_0(\theta) = \frac{1}{k!}[t_0, t_1, \ldots, t_{k+1}](t - \theta)_+^k, \quad a \leq \theta \leq b$$

O sea

$$B_0(\theta) = [t_0, t_1, \ldots, t_{k+1}](t - \theta)_+^k .$$

Es decir un B-spline es una diferencia dividida.

Finalmente, para expresar la fórmula en términos de nuestra variable t, escribiremos

$$B_0(t) = [t_0, t_1, \ldots, t_{k+1}](x - t)_+^k, \quad a \leqslant t \leqslant b$$

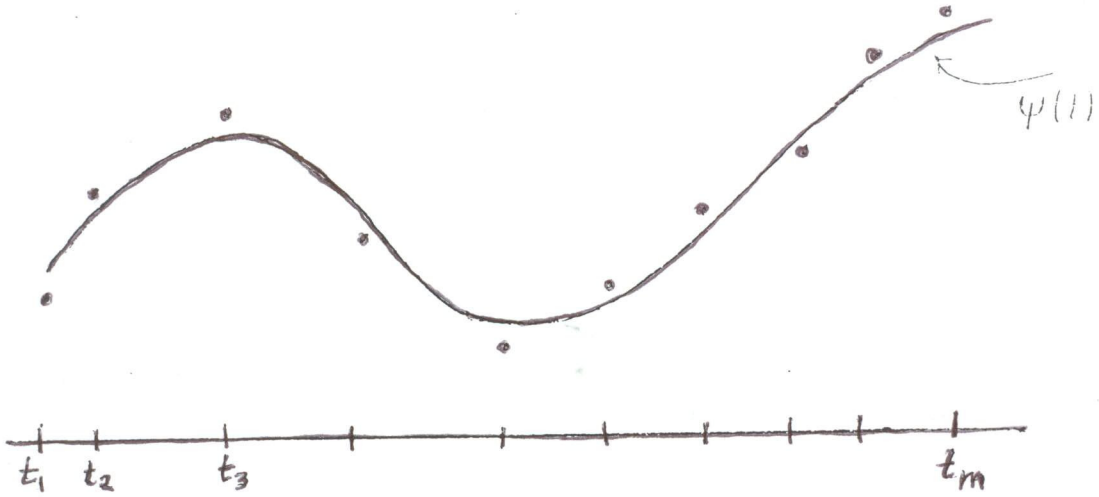
para significar que la diferencia dividida de la función de las dos variables x, t, $(x - t)_+^k$, se toma fijando t y considerando, para el cálculo de la diferencia dividida a $(x - t)_+^k$ únicamente como función de x.

CAPITULO II

AJUSTE DE DATOS CON SPLINES CUBICOS.

2.1. El problema general.

Tanto en la práctica de la investigación científica como de la técnica, es frecuente encontrar el siguiente problema: Dada una colección de puntos $t_1, t_2, \ldots, t_m$ en la recta real y datos $f_1, f_2, \ldots, f_m$ asociados a aquellos puntos, encontrar una función $\psi(t)$ perteneciente a un cierto conjunto, generalmente un espacio vectorial, que "mejor" los represente, que "mejor se ajuste" a ellos, con un cierto criterio. Es el problema de ajuste de datos.



Si la función buscada ψ pertenece a un espacio vectorial de dimensión n, digamos, y una base de tal espacio es el conjunto de funciones $\{\psi_j : j = 1,2,\ldots,n\}$, entonces ψ se puede expresar como

$$\psi(t) = \sum_{j=1}^{n} \alpha_j \psi_j(t)$$

para ciertos coeficientes $\{\alpha_j : j = 1,2,\ldots,n\}$. Así pues, el problema se transforma en encontrar los coeficientes α_j de tal manera que

$$f_i \approx \sum_{j=1}^{n} \alpha_j \psi_j(t_i), \quad i = 1,2,\ldots,m.$$

O, en otra forma, encontrar las α_j de tal manera que los residuales

$$\varepsilon_i = f_i - \sum_{j=1}^{n} \alpha_j \psi_j(t_i), \quad i = 1,2,\ldots,m$$

sean tan pequeños como sea posible, con un cierto criterio. Un criterio muy frecuentemente usado es pedir que la suma de los cuadrados de todos los residuales sea mínima. Es el criterio de *suma mínima de cuadrados*. que podemos expresar de la siguiente manera: Si ε es el vector residual $\varepsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m)^T$, entonces queremos minimizar su norma, o el cuadrado de ella, $\|\varepsilon\|_2^2$. Para expresar explícitamente los paráme-- tros α_j, vamos a definir la siguiente matriz A:

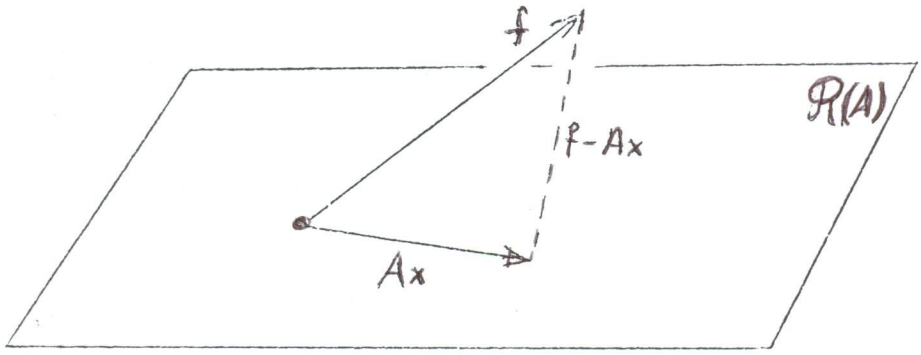
$$A = \begin{bmatrix} \psi_1(t_1) & \psi_2(t_1) & \ldots & \psi_n(t_1) \\ \psi_1(t_2) & \psi_2(t_2) & \ldots & \psi_n(t_2) \\ \psi_1(t_3) & \psi_2(t_3) & \ldots & \psi_n(t_3) \\ \ldots & \ldots & \ldots & \ldots \\ \psi_1(t_m) & \psi_2(t_m) & \ldots & \psi_n(t_m) \end{bmatrix} .$$

Si f es el vector $f = (f_1, f_2, \ldots, f_m)^T$ y el vector α es $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, entonces el problema de ajuste con el criterio de suma mínima de cuadrados puede expresarse de la siguiente manera concisa: Encontrar el vector α de tal manera que se minimice

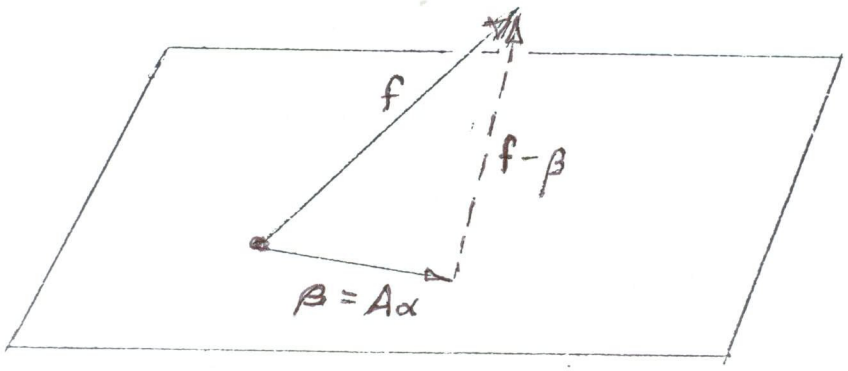
$$\|f - A\alpha\|_2^2 .$$

A veces se hace referencia este problema, sin hacer referencia al ajuste, como el problema de mínimos cuadrados,

o de cuadrados mínimos o, como nosotros preferimos, de suma mínima de cuadrados. Una manera de convencernos de que este importante problema tiene solución es la siguiente. En primer lugar observemos que A puede considerarse como la matriz que representa una transformación lineal de R^n sobre su rango $\mathcal{R}(A) \subset R^m$. Un esquema geométrico nos ayudará. (Suponemos que $m > n$).



Vemos así que lo que se requiere es encontrar un vector, digamos β, en $\mathcal{R}(A)$



de tal manera que el residual $f - \beta$ tenga longitud mínima, y, por el teorema de Pitágoras, esto se logra cuando el residual es perpendicular al plano $\mathcal{R}(A)$. El vector β es -

entonces la proyección de f sobre $\mathcal{R}(A)$. Habiendo encontrado β podemos encontrar α tal que $\beta = A\alpha$, ya que $\beta \in \mathcal{R}(A)$. Además, la solución es única si y sólo si la dimensión del espacio nulo de A es cero.

Desde un punto de vista más formal existe el siguiente teorema:

Teorema. Supongamos que A es una matriz $m \times n$ de rango k y que

$$A = HRK^T$$

donde

(a) H es una matriz ortogonal mxm.

(b) R es una matriz de la forma

$$R = \begin{bmatrix} R_{11} & 0 \\ 0 & 0 \end{bmatrix}.$$

(c) R_{11} es una matriz $k \times k$ de rango k.

(d) K es una matriz ortogonal nxn.

Definamos el vector

$$H^T f = g = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \begin{matrix} \} k \\ \} m-k \end{matrix}$$

e introduzcamos la nueva variable

$$K^T \alpha = y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \begin{matrix} \} \ k \\ \} \ n-k \end{matrix}$$

Definamos $\tilde{y}_1$ como la solución (única) de

$$R_{11}\, y_1 = g_1$$

Entonces, la solución de longitud mínima del problema de minimizar $\|f - A\alpha\|_2^2$ es

$$\tilde{\alpha} = K \begin{bmatrix} \tilde{y}_1 \\ 0 \end{bmatrix} .$$

Para una demostración de este teorema ver Lawson y Hanson [13].

2.2. Ajuste con Splines Cúbicos.

Volveremos ahora al problema de ajustar m datos $\{(t_i, f_i) : i = 1,2,\ldots,m\}$ para precisar el espacio de las funciones de ajuste que nosotros emplearemos: Tal espacio es el de los splines cúbicos. Diremos entonces que buscamos un spline cúbico $s(t)$ para ajustar los datos con el criterio de suma mínima de cuadrados.

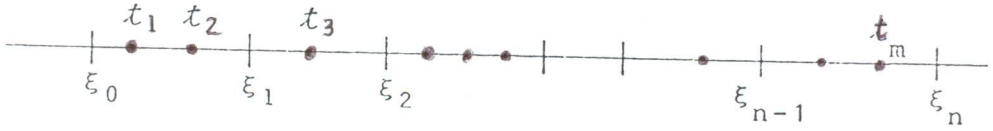
También sabemos que los B-splines cúbicos constituyen una base de tal espacio, así pues, nuestro problema se convierte en encontrar los coeficientes α_j para obtener la -

combinación lineal $\Sigma \alpha_j B_j^3(t)$ que nos proporcione tal spline $s(t)$.

Naturalmente que el espacio de splines requiere de los nodos para su definición. En el capítulo anterior habíamos supuesto que los nodos estaban dados y que ellos constituían un conjunto de $(n+1)$ puntos ordenados en un orden estrictamente creciente, $\{\xi_0, \xi_1, \ldots, \xi_n\}$. Se demostró que $dim\ S(k,\pi) = n + k$. En el caso presente, $k = 3$, la dimensión del espacio de splines cúbicos es $(n + 3)$. Dicho con otras palabras

$$dim\ S(3,\pi) = (\text{número de nodos}) + 2.$$

Por otra parte siempre vamos a suponer que el conjunto de las abscisas dato $\{t_i : i = 1,2,\ldots,m\}$ está contenido en el intervalo $[\xi_0, \xi_n]$.

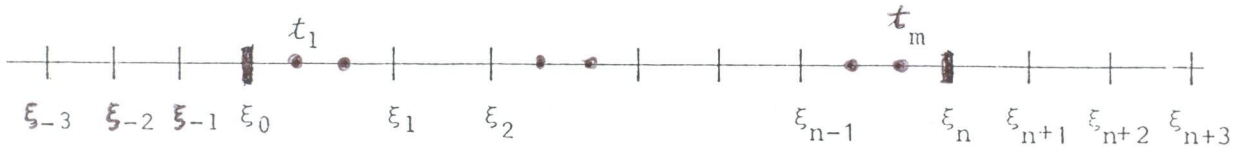


En ocasiones nos referiremos a los nodos ξ_0, ξ_n como los postes y a los nodos $\xi_1, \xi_2, \ldots, \xi_{n-1}$ como los nodos interiores. Observemos que, como consecuencia

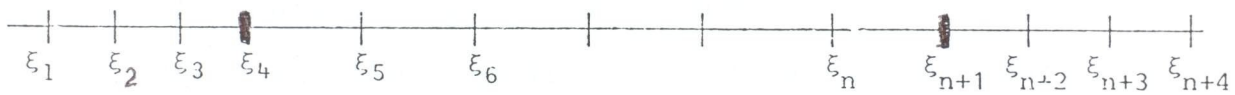
$$dim\ S(3,\pi) = (\text{número de nodos interiores}) + 4.$$

Recuérdese que también habíamos mostrado la necesidad de k, (o sea 3) nodos auxiliares a la izquierda de ξ_0 y k, (o sea 3)

nodos auxiliares a la derecha de ξ_n.



Con miras a simplificar el trabajo de programación, vamos a cambiar la notación. Tal cambio lo mostramos en el siguiente diagrama



Los postes pasan a ser designados con ξ_4 y ξ_{n+1}. Es sólo un corrimiento en los índices lo que hemos hecho, y aún cuando usamos la misma letra n, no tiene el mismo significado que antes. La razón de hacer este simple cambio es en la notación es para designar los B-splines de la base que genera el espacio - con índices desde 1 hasta n : $\{B_1(t), B_2(t), \ldots, B_n(t)\}$. Y así *en esta nueva notación la dimensión del espacio es* n. pero los conceptos y relaciones no cambian; por ejemplo la relación fundamental,

(número de nodos) + 2 = dimensión del espacio,

sigue siendo válida. También, (pero es la misma relación)

(número de nodos interiores) + 4 = dimensión del espacio.

Repetimos: el espacio $S(3,\xi_4,\xi_5,\ldots,\xi_{n+1})$ de los splines cúbicos con nodos interiores $\xi_5,\ldots,\xi_n$ y postes ξ_4,ξ_{n+1} es generado por la base de B-splines de grado 3 $\{B_1(t),\ldots, B_n(t)\}$. De manera que, en consecuencia, cualquier spline cúbico $s(t)$ lo podemos expresar como una combinación lineal de los elementos básicos con coeficientes constantes $\{\alpha_j : j = 1,\ldots,n\}$ así

$$s(t) = \sum_{j=1}^{n} \alpha_j B_j(t).$$

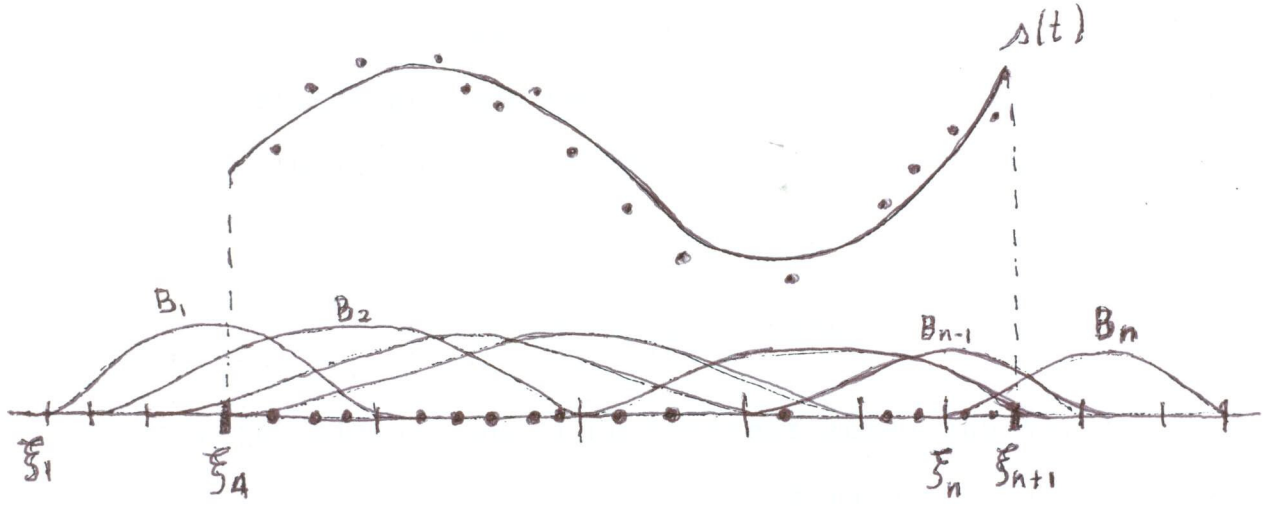
Una vez que hemos adoptado un espacio vectorial y con él una base que lo genera, podemos reformular nuestro problema de suma mínima de cuadrados: minimizar

$$\| f - A\alpha \|_2^2$$

donde A es la matriz $m \times n$

$$A = \begin{bmatrix} B_1(t_1) & B_2(t_1) & \ldots & B_n(t_1) \\ B_1(t_2) & B_2(t_2) & \ldots & B_n(t_2) \\ \vdots & & & \\ B_1(t_m) & B_2(t_m) & \ldots & B_n(t_m) \end{bmatrix}.$$

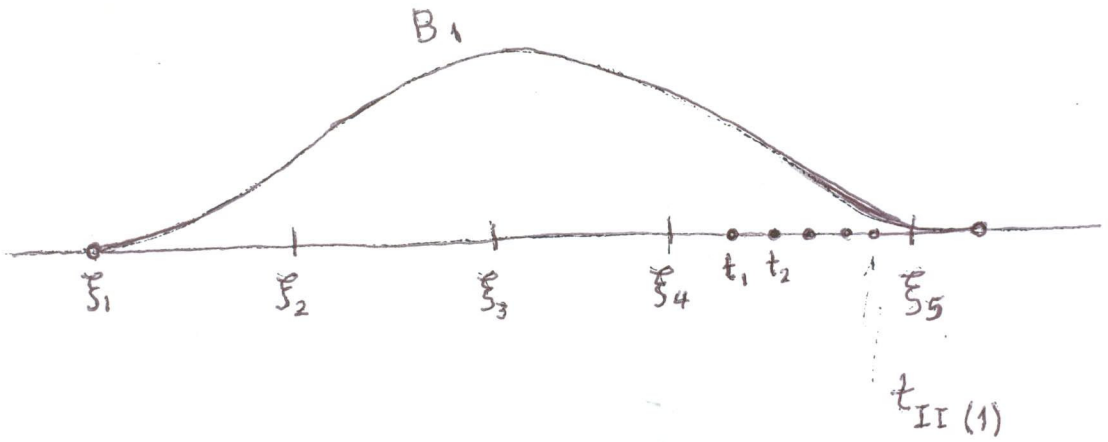
La solución $\alpha = (\alpha_1, \ldots, \alpha_n)^T$ de este problema nos determina el spline $s(t) = \sum_{j=1}^{n} \alpha_j B_j(t)$ que ajusta los datos



Pero la matriz **A** tiene una estructura interesante que ahora vamos a investigar antes de seguir adelante. Observemos la primera columna:

$$\begin{bmatrix} B_1(t_1) \\ B_1(t_2) \\ \vdots \\ B_1(t_m) \end{bmatrix};$$

esta columna en realidad no es llena, puesto que a partir de un cierto índice para las t_i's, $B_1(t_i)$ es cero.



El índice $II(1)$ indicará que a partir de él todos los elementos de la primera columna son cero. Esta primera columna tendrá el siguiente aspecto.

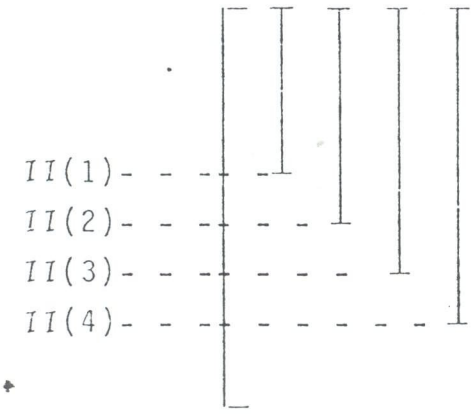
$$\begin{bmatrix} B_1(t_1) \\ B_1(t_2) \\ \vdots \\ B_1(t_{II(1)}) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad ; \quad \begin{bmatrix} * \\ * \\ \vdots \\ * \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \longleftarrow II(1)$$

Análogamente para las siguientes tres columnas:

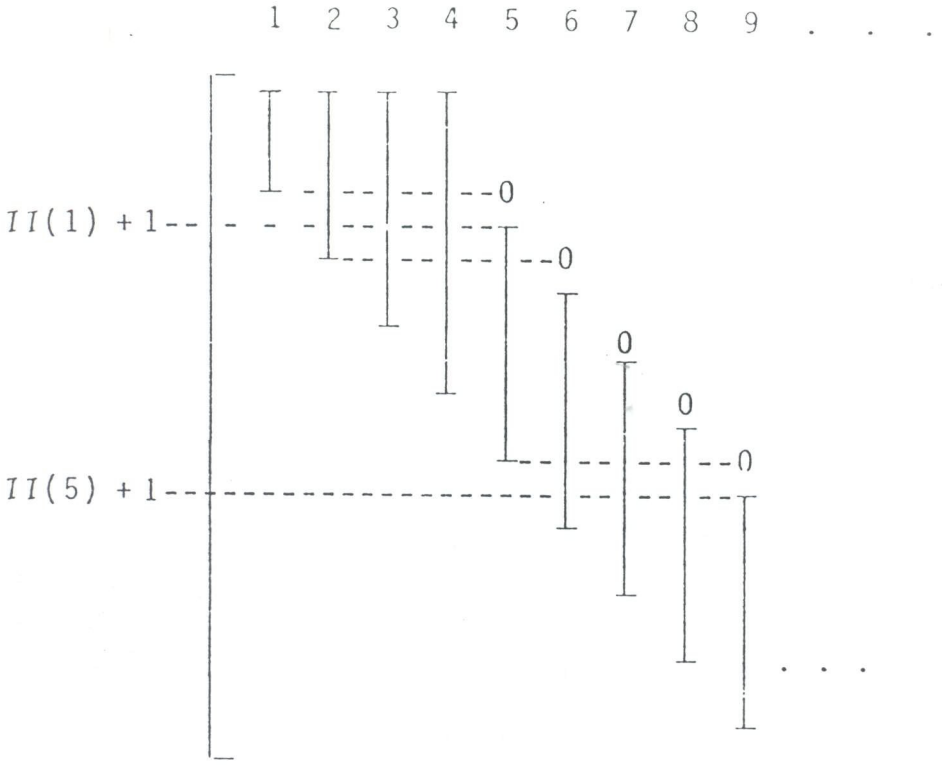
$$\begin{bmatrix} B_1(t_1) \\ B_2(t_2) \\ \vdots \\ B_2(t_{II(2)}) \\ 0 \\ \vdots \\ 0 \end{bmatrix} ; \quad \begin{bmatrix} B_3(t_1) \\ B_3(t_2) \\ \vdots \\ B_3(t_{II(3)}) \\ 0 \\ \vdots \\ 0 \end{bmatrix} ; \quad \begin{bmatrix} B_4(t_1) \\ B_4(t_2) \\ \vdots \\ B_4(t_{II(4)}) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

El subíndice $II(j)$ también indica el mayor subíndice de t_i, tal que $t_{II(j)}$ no excede a ξ_{j+4}.

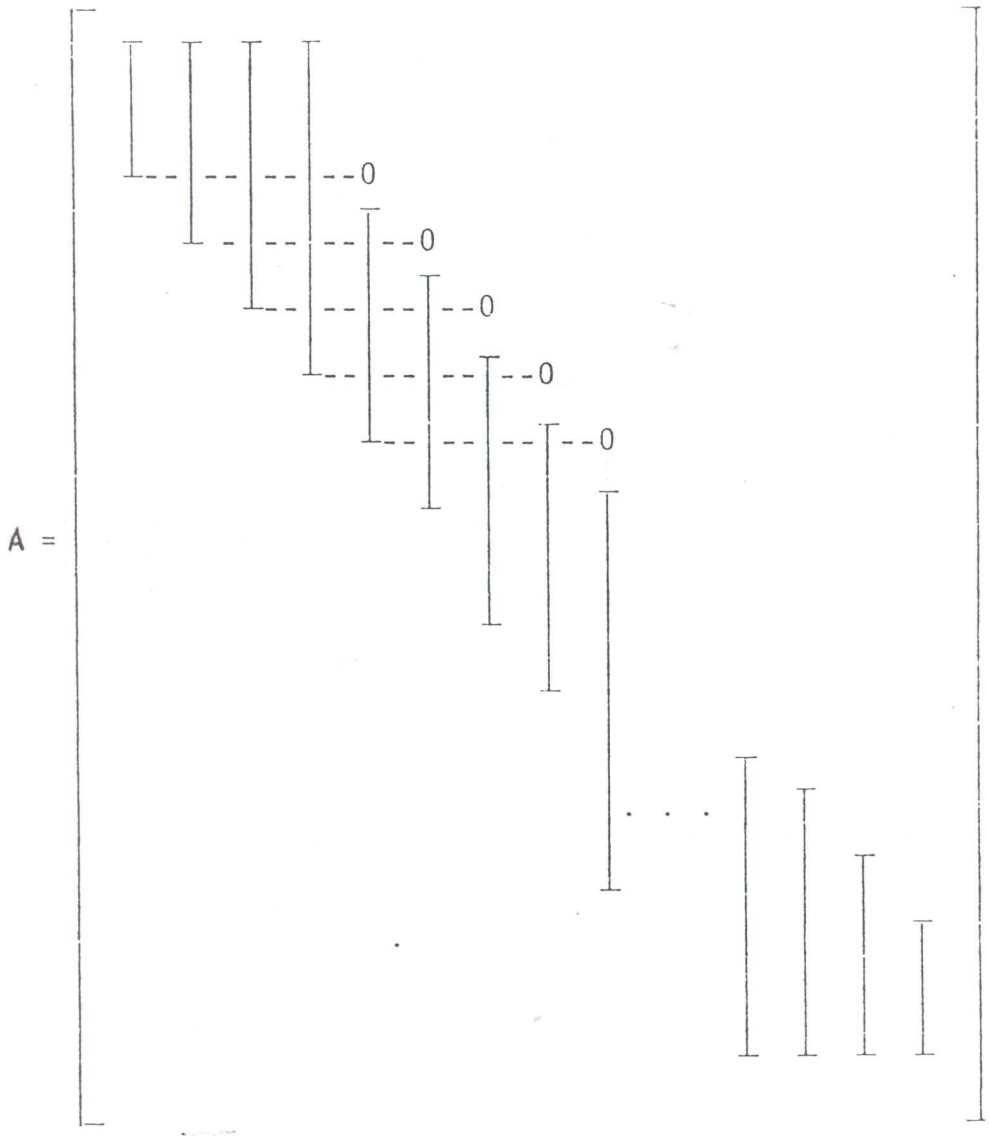
Así pues las primeras cuatro columnas de la matriz A se ven de la siguiente manera:



Las siguientes columnas, hasta la $(n\text{-}4)$, adquieren el siguientes aspecto.



Finalmente, al agregar las últimas cuatro columnas, la matriz A queda con la siguiente estructura



La subrutina que construye esta matriz es AMATR.

Volvemos ahora a nuestro problema

$$\min \| \mathbf{f} - A\alpha \|_2^2 \quad ,$$

donde A es la matriz $m \times n$ que se acaba de describir, $f = (f_1, f_2, \ldots, f_m)^T$, $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$. Vamos a aplicar el teorema que enunciamos al final de la primera sección de este capítulo. Haciendo la matriz K del teorema igual a

la identidad, nos queda

$$A = HR$$

donde H la haremos igual al producto de las transformaciones ortogonales que vamos a describir un poco más abajo; R es de la forma

$$R = \begin{bmatrix} R_{11} \\ 0 \end{bmatrix},$$

y R_{11} es de rango n, y además tiene la forma

$$R_{11} = \begin{bmatrix} & & & 0 \\ & \diagdown & & \\ & & \diagdown & \\ 0 & & & \diagdown \end{bmatrix}.$$

Habiendo logrado esta factorización definimos el vector $H^T f = g$, y de acuerdo con el teorema, resolvemos el sistema

$$R_{11}\, y_1 = g_1 ,$$

donde

$$g = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \begin{matrix} \} n \\ \} m-n \end{matrix} .$$

Al resolverlo obtenemos la solución de longitud mínima del problema de minimizar $\|f - A\alpha\|_2^2$, es decir el vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$, y así el spline que buscábamos.

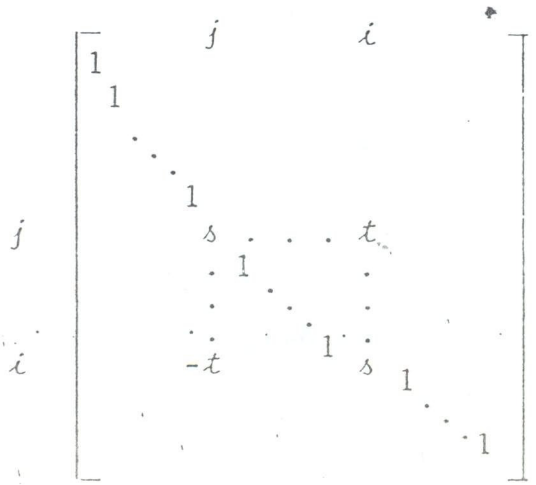
En ocasiones, este vector α, solución del problema de suma mínima de cuadrados, se representa de la siguiente manera

$$\alpha = A^{+} f ,$$

y se dice que A^{+} es la "inversa generalizada" o "pseudoinversa" de la matriz A.

Todavía nos falta decir cómo llevar a cabo la factorización de A para obtener la matriz R_{11}. Para esto fue que investigamos la estructura de la matriz A. Esta estructura se debe a haber elegido la base de B-splines y es conveniente porque la matriz no es llena y la longitud no-cero de cada columna puede conocerse. Entonces vemos que lo que más nos conviene es aplicar rotaciones de Givens para hacer ceros abajo de la diagonal principal. Pero no necesitamos afectar toda la columna (en ese caso sería preferible usar reflexiones de Householder) puesto que ya hay ceros a partir de un cierto índice. Por ello elegimos las rotaciones de Givens, pues ellas hacen un cero en cada aplicación.

Una rotación de Givens



hace un cero en el elemento a_{ij} de la matriz A al aplicarle tal rotación por la izquierda a A. Así pues para hacer ceros en la columna j debajo de su elemento a_{jj}, aplicaríamos estas rotaciones a los elementos siguientes, pero sólo hasta el de índice $II(j)$, que, recordamos, es el último elemento distinto de cero en la columna j. Nos conviene, para no tener que almacenar por separado la información, hacer simultáneamente las transformaciones correspondientes al segundo miembro de la ecuación. Las subrutinas que reúnen las ideas expuestas se encuentran en nuestra rutina SPLINE/CUBICO. Posteriormente hablaremos de la evaluación de los B-splines cúbicos, es decir, las entradas de la matriz A. Por el momento mencionamos el algoritmo para realizar las transformaciones de Givens y que se encuentra incluído en la misma rutina como subrutina GIVENS.

Algoritmo (GIVENS). Conocida la matriz A y su estructura y el vector f de ordenadas dato, este algoritmo aplica las rotaciones de Givens, por columnas, a cada uno de los elementos no-cero debajo de la diagonal principal. La matriz triangular superior R_{11} de la descomposición queda almacenada en A. Al mismo tiempo se hace la transformación correspondiente al vector f de datos.

1) Para $j = 1, \ldots, n$

  1) Para $i = j + 1, \ldots, II(j)$

    1) Si $|a_{ij}| < |a_{jj}|$ pasa a 3)

    2) $t \longleftarrow a_{jj}/a_{ij}$

      $s \longleftarrow 1/\sqrt{1 + t^2}$

      $c \longleftarrow t \cdot s$

      Pasa a 4)

    3) $t \longleftarrow a_{ij}/a_{jj}$

      $c \longleftarrow 1/\sqrt{1 + t^2}$

      $s \longleftarrow t \cdot c$

    4) $a_{jj} \longleftarrow c \cdot a_{jj} + s \cdot a_{ij}$

      $a_{ij} \longleftarrow 0$

      $f_j \longleftarrow c \cdot f_j + s \cdot f_i$

      $f_i \longleftarrow - s \cdot f_j + c \cdot f_i$

    5) Para $j \neq n$

      Para $k = j + 1, \ldots, \min(j + 3, n)$

      $a_{jk} \longleftarrow c \cdot a_{jk} + s\, a_{ik}$

      $a_{ik} \longleftarrow -s \cdot a_{jk} + c \cdot a_{ik}$

FIN

2.3 B-splines otra vez.

Antes de terminar este capítulo hemos de mencionar algo sobre los B-splines que tiene por objetivo hacerlos normalizados para mayor estabilidad computacional. Sabemos que es posible expresar cualquier spline $s(t)$ como una combinación de B-splines. Sin embargo, desde el punto de vista computacional, es conveniente introducir cierto factor de normalización en los B-splines que hemos estado considerando. Para ello vamos a definir la siguiente función

$$N_p^k(t) = (\xi_{p+k+1} - \xi_p)\, B_p^k(t),$$

que no es más que el B-spline B_p^k multiplicado por el factor que se indica. Esto significa que los splines $N_p^k(t)$ son base del espacio y, como los B-splines, únicamente no son cero en el intervalo (ξ_p, ξ_{p+k+1}), donde son estrictamente positivos. En consecuencia, tienen la misma forma que los B-splines. El factor positivo $(\xi_{p+k+1} - \xi_p)$ que hemos introducido es con el objeto de que estas funciones tengan la siguiente propiedad:

Teorema. Las funciones $\{N_p^k(t) : p = -k, -k+1, \ldots, n-1\}$ (con la notación del capítulo I) satisfacen

$$\sum_{p=-k}^{n-1} N_p^k(t) = 1, \qquad a \leq t \leq b.$$

Demostración: Recordemos la fórmula de recurrencia que de-- mostramos en el capítulo I:

$$B_p^k(t) = \frac{(t - \xi_p)B_p^{k-1}(t) + (\xi_{p+k+1} - t)\, B_{p+1}^{k-1}(t)}{\xi_{p+k+1} - \xi_p}.$$

Multipliquemos por $(\xi_{p+k+1} - \xi_p)$:

$$(\xi_{p+k+1} - \xi_p)B_p^k(t) = (t - \xi_p)\, B_p^{k-1}(t) + (\xi_{p+k+1} - t)B_{p+1}^{k-1}(t).$$

Por la definición de $N_p^k(t)$, nos queda

$$N_p^k(t) = \frac{(t - \xi_p)}{(\xi_{p+k} - \xi_p)}\, N_p^{k-1}(t) + \frac{(\xi_{p+k+1} - t)}{(\xi_{p+k+1} - \xi_{p+1})} N_{p+1}^{k-1}(t),$$

$p = -k, -k+1, \ldots, n-1.$

En seguida, para hacer la demostración por inducción, vamos a sumar sobre el índice p:

$$\sum_{p=-k}^{n-1} N_p^k(t) = \sum_{p=-k}^{n-1} \frac{(t - \xi_p)}{(\xi_{p+k} - \xi_p)}\, N_p^{k-1}(t) + \sum_{p=-k}^{n-1} \frac{(\xi_{p+k+1} - t)}{(\xi_{p+k+1} - \xi_{p+1})} N_{p+1}^{k-1}(t),$$

cambiando índices en el segundo sumando

$$\sum_{p=-k}^{k-1} N_p^k(t) = \sum_{p=-k}^{n-1} \frac{(t - \xi_p)}{(\xi_{p+k} - \xi_p)}\, N_p^{k-1}(t) + \sum_{p=-k+1}^{n} \frac{(\xi_{p+k} - t)}{(\xi_{p+k} - \xi_p)} N_p^{k-1}(t).$$

Pero, $N_{-k}^{k-1}(t) = 0$ y $N_n^{k-1}(t) = 0$, $a \leq t \leq b$, por lo que podemos escribir

$$\sum_{p=-k}^{n-1} N_p^k(t) = \sum_{p=-k+1}^{n-1} \frac{(t-\xi_p)}{(\xi_{p+k}-\xi_p)} N_p^{k-1}(t) + \sum_{p=-k+1}^{n-1} \frac{(\xi_{p+k}-t)}{(\xi_{p+k}-\xi_p)} N_p^{k-1}(t).$$

$$= \sum_{p=-k+1}^{n-1} N_p^{k-1}(t), \qquad a \leq t \leq b.$$

Por lo tanto, si la fórmula del teorema vale para $k=1$, habremos completado la inducción. Pero esto es ciertamente así, como podemos fácilmente checar al observar la definición de $B_p^1(t)$. Con lo cual queda demostrado el teorema.

Ahora, vamos a utilizar la primera ecuación que escribimos en esta última demostración, cambiando ligeramente la notación y particularizando para $k=3$:

$$N_j^3(t) = \frac{(t-\xi_j)}{(\xi_{j+3}-\xi_j)} N_j^2(t) + \frac{(\xi_{j+4}-t)}{(\xi_{j+4}-\xi_{j+1})} N_{j+1}^2(t)$$

y volviendo a hacer el cambio en la numeración de los nodos, j recorrerá los valores $j=1,2,\ldots,n$.

Como ya dijimos, estas funciones N_j^3 son un factor constante por los básicos B-splines. Por lo tanto, pueden ser tomados como una base; y eso es lo que haremos. Nada nos impide, aunque es desde luego un abuso de notación, denotar a estas funciones N_j^3 con la misma letra B_j^3:

$$B_j^3(t) = \frac{(t-\xi_j)}{(\xi_{j+3}-\xi_j)} B_j^2(t) + \frac{(\xi_{j+4}-t)}{(\xi_{j+4}-\xi_{j+1})} B_{j+1}^2(t)$$

Y abusando otra vez del lenguaje, son estas funciones a las que llamaremos, de aquí en adelante, B-splines. Nuestros actuales B-splines pueden definirse como lo hemos hecho, o bien, la anterior fórmula de recurrencia puede tomarse como la definición, partiendo de la siguiente fórmula para los B-splines de grado cero que debemos adoptar para ser consecuentes

$$B_j^0(t) = \begin{cases} 1 & \text{si} \quad \xi_j \leq t \leq \xi_{j+1} \\ 0 & \text{si} \quad t \notin [\xi_j, \xi_{j+1}] \end{cases}$$

Todo lo que dijimos sobre el problema de ajuste con splines cúbicos tiene validez sin modificaciones cuando la base usada es esta nueva base de B-splines y un spline cúbico $s(t)$ es expresado como

$$s(t) = \sum_{j=1}^{n} \alpha_j B_j^3(t).$$

2.4 Evaluación de un spline cúbico.

En la práctica, para evaluar un spline $s(t)$ cuando se conoce su expresión en la forma

$$s(t) = \sum_{j=1}^{n} \alpha_j B_j^3(t)$$

lo que se requiere es evaluar los B-splines en t y hacer la combinación lineal ya que los coeficientes α_j's son conocidos.

Dos útiles observaciones están a la orden. La primera de ellas es que si t está en el intervalo $[\xi_j, \xi_{j+1})$, digamos, entonces únicamente no son cero los B-splines $\{B_p(t): p = j-3, j-2, j-1, j\}$ y así

$$s(t) = \sum_{p=j-3}^{j} \alpha_p B_p(t)$$

únicamente.

La segunda observación permite evaluar cualquier B-spline con un algoritmo sencillo que calcule una diferencia dividida y un producto, pues recuérdese, por ejemplo, que

$$B_p^3(t) = (\xi_{p+4} - \xi_p)[\xi_p, \ldots, \xi_{p+4}](x-t)_+^3 .$$

Entonces, dado t en $[a,b]$, podríamos hacer un corrimiento provisional de índices y proceder con el siguiente algoritmo (subrutinas MOVI y BASICO)

Algoritmo (MOVI).

1) Para $i = 0,1,\ldots,4$

$\xi_{1+i} \leftarrow \xi_{p+i}$

FIN

Algoritmo (BASICO). Se calcula, evaluado en t, el B-spline cúbico $B_1(t)$ con nodos $\{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5\}$. Esto es el

producto del factor $(\xi_5 - \xi_1)$ multiplicado por la diferencia dividida $[\xi_1, \xi_2, \xi_3, \xi_4, \xi_5](x - t)_+^3$ de la función $(x - t)_+^3$ como función de x. El valor obtenido se guarda en BASICO.

1) Si

$(t \leq \xi_1)$ ó $(t > \xi_5)$

BASICO $\longleftarrow$ 0

Salida

2) Si $(t \leq \xi_2)$ pasa a 3)

Si $(t \leq \xi_3)$ pasa a 4)

Si $(t \leq \xi_4)$ pasa a 5)

Si $(t \leq \xi_5)$ pasa a 6)

3) Para $i = 2, \ldots 5$

$G_i \longleftarrow (\xi_i - t)^3$

Pasa a 7)

4) Para $i = 3, \ldots, 5$

$G_i \longleftarrow (\xi_i - t)^3$

Pasa a 7)

5) Para $i = 4, 5$

$G_i \longleftarrow (\xi_i - t)^3$

Pasa a 7)

6) $G_5 \longleftarrow (\xi_5 - t)^3$

7) Para $j = 1, \ldots, 4$

    1) Para $i = 1, 5 - j$

        $G_i \longleftarrow (G_i - G_{i+1})/(\xi_i - \xi_{i+j})$

BASICO $\longleftarrow (\xi_5 - \xi_1)/G_1$

FIN

2.5 Ejemplo.

Incluímos aquí un ejemplo para ilustrar las ideas de este capítulo. Posteriormente vamos a presentar más ejemplos y otros resultados experimentales.

Para construir este ejemplo se tomó la función

$$f(t) = t^2 \text{ sen } t, \quad -\pi \leq t \leq 2\pi.$$

Tomamos (arbitrariamente) los siguientes 5 nodos interiores

$$\xi_5 = -2.2222222$$
$$\xi_6 = -0.6666666$$
$$\xi_7 = 0.9333333$$
$$\xi_8 = 2.2666666$$
$$\xi_9 = 5.2000000$$

Los postes los elegimos como

$$\xi_4 = -3.1416$$

$$\xi_{10} = 6.2832$$

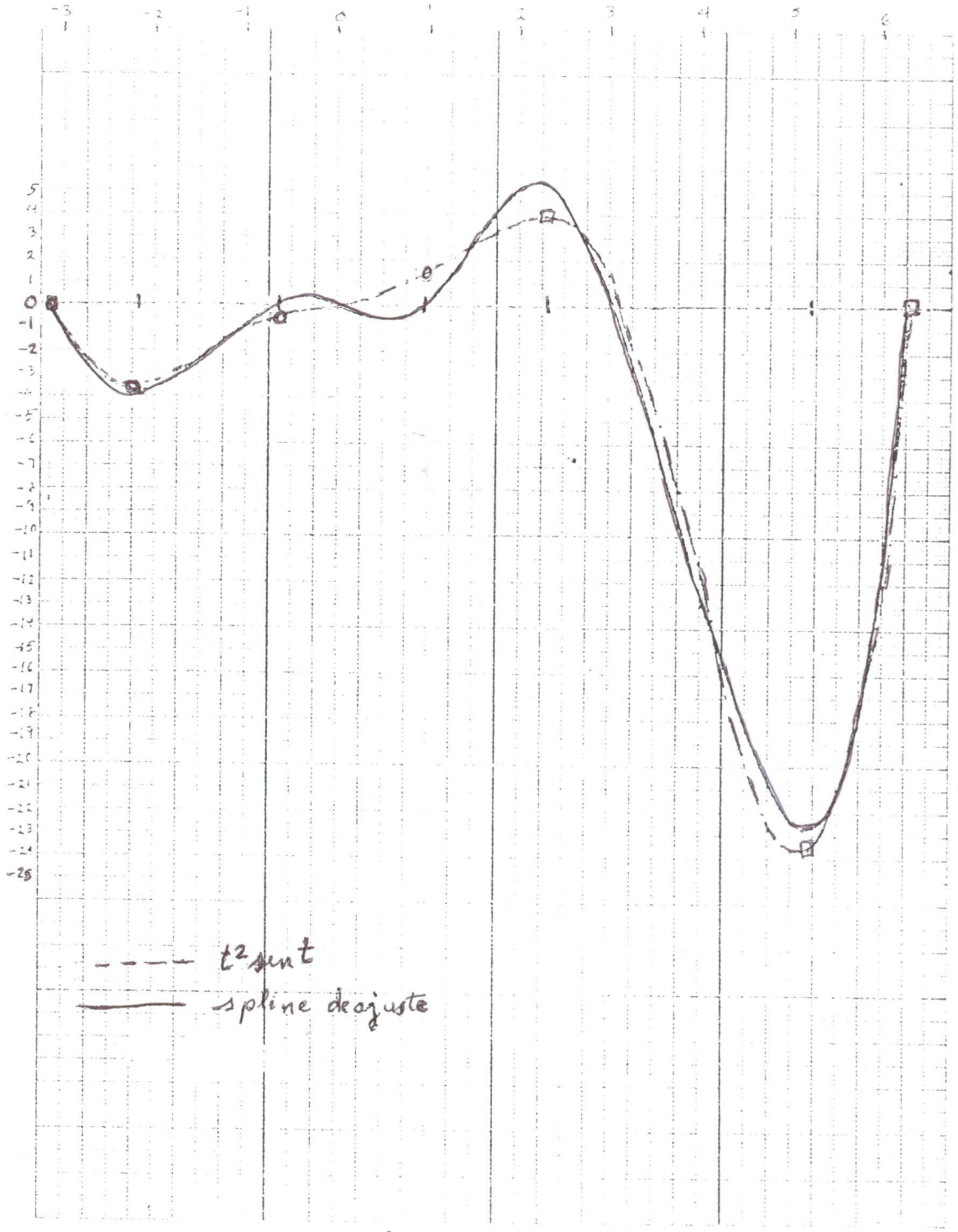
Como abscisas muestrales se tomaron 50 puntos equidistantes en el intervalo $[-\pi, 2\pi]$. Al resolver el problema

$$\min \| f - A\alpha \|_2^2$$

se encontró el siguiente vector α:

11.67500
-1.27740
-6.31312
3.54634
-4.23494
16.24514
-32.88860
-15.20042
82.97364

y la gráfica del spline cúbico de ajuste es la siguiente


- - - - $t^2 \operatorname{sen} t$
——— spline de ajuste

CAPITULO III

NODOS LIBRES

3.1 Aproximación spline con nodos libres.

En el capítulo II hemos resuelto el problema lineal de suma mínima de cuadrados

$$\min \; ||f - A\,\alpha||_2^2 \;,$$

donde la incógnita es el vector α, cuyas n componentes son los n coeficientes del spline

$$s(t) = \sum_{j=1}^{n} \alpha_j B_j^3(t)$$

que es el mejor spline cúbico, fijados los nodos, que ajusta

los datos con el criterio de suma mínima de cuadrados. Si a la función $||f - A\,\alpha||_2^2$ que queremos minimizar la denotamos con $F(\alpha)$, es decir

$$F(\alpha) = ||f - A\,\alpha||_2^2, \qquad \alpha \in \mathbb{R}^n,$$

entonces, el mismo problema lo podemos expresar como

$$\min_{\alpha \in \mathbb{R}^n} F(\alpha).$$

Pero recordemos que en aquel entonces hicimos énfasis en determinar o fijar el espacio de funciones en el cual se encontraría la función $s(t)$ de ajuste. Tal espacio fue $S(3, \Pi)$, el espacio de splines cúbicos con nodos $\{\xi_4, \xi_5, \ldots, \xi_{n+1}\}$, con nuestra notación modificada actual.

En lo que queremos hacer énfasis en este momento es que se consideró a los nodos dados fijos. Pero si cambiamos adecuadamente la posición de los nodos, manteniendo fijo el número de nodos, se dispone de otro espacio de funciones y, probablemente, en este nuevo espacio existe una mejor función de ajuste que sea mejor que la que se encontró en el espacio anterior.

La intuición hace sentir que un mejor ajuste se logra si se colocan los nodos en los lugares donde "está la acción",

de acuerdo con los datos del problema.

Sin embargo, podemos dar un gran paso adelante si permitimos que los nodos varíen continuamente dentro del intervalo $[a, b]$ que contiene a las abscisas dato $\{t_i : i=1,\ldots, m\}$.

Para enunciar matemáticamente el nuevo problema, denotemos con x al vector cuyas componentes son los nodos libres: $x = (\xi_5, \xi_6,\ldots, \xi_n)^T$, y con α el vector de coeficientes: $\alpha = (\alpha_1, \alpha_2,\ldots, \alpha_n)^T$. Otra vez, la función que queremos minimizar es

$$||f - A\alpha||_2^2,$$

pero ahora consideramos a los nodos variables tanto como al vector α, que de hecho depende de los nodos, es decir del vector x. Así pues, el problema completo lo podemos enunciar como

$$\min_{\substack{\alpha\in\mathbb{R}^n \\ x\in\Delta_N}} F(x, \alpha) = \min_{\substack{\alpha\in\mathbb{R}^n \\ x\in\Delta_N}} ||f - A(x)\,\alpha(x)||_2^2$$

donde Δ_N es el simplejo

$$\Delta_N = \{x \in \mathbb{R}^{n-4}: \quad a < \xi_5 < \ldots < \xi_n < b\}$$

Hemos introducido la notación $A(x)$ y $\alpha(x)$ para mostrar que tanto la matriz A como el vector α dependen de x, y la función por minimizar $F(x, \alpha)$ va a ser minimizada tanto con respecto a x como con respecto a α.

Este es un nuevo problema de suma mínima de cuadrados. Es no-lineal, pues la matriz A tiene como entrada (i, j) el valor

$$B_j^3(t_i),$$

que es el B-spline $B_j(t)$ evaluado en t_i. Una manera de escribir este número es

$$B_j^3(t_i) = \sum_{\ell=j}^{j+4} \left[\prod_{\substack{p=j \\ p\neq\ell}}^{j+4} \frac{1}{(\xi_p - \xi_\ell)} \right] (t_i - \xi_\ell)_+^3 ,$$

que muestra que el B-spline es una función racional de los nodos; muestra también que es una función continua y diferenciable, pues aquí no consideramos el caso de nodos coincidentes. Esta diferenciabilidad de los B-splines con respecto a los nodos implica que podemos hablar también de la derivada parcial con respecto a los nodos de la matriz $A(x)$. Esto último nos interesa porque al intentar resolver el problema de suma mínima de cuadrados no-lineal por alguno de

los métodos usuales, vamos a necesitar las derivadas parciales de los B-splines.

3.2 Variables que se separan.

Ahora bien, una observación que debemos hacer en este momento es que en la expresión

$$F(x, \alpha) = ||f - A(x)\ \alpha(x)||_2^2$$

los parámetros α aparecen linealmente, mientras que los parámetros x aparecen en forma no-lineal. Se puede demostrar que el problema se puede resolver separadamente, para la parte lineal y para la parte no-lineal por separado. Es el tipo de problemas en el que "las variables se separan". Es este un enfoque que a nosotros en este trabajo no nos ha sido posible abordar. (El lector interesado puede ver Jupp [10] y Golub y Pereyra[8]. Sin embargo haremos algunas observaciones.

Para la parte no-lineal es que se necesita el gradiente de la función

$$\tilde{F}(x) = ||f - A(x)\ \alpha||_2^2 ,$$

el cual es un vector cuya j-ésima componente puede calcular

se recordando la fórmula para la derivada del producto escalar de dos funciones

$$\frac{d}{dx}\left\langle u(x),\, v(x)\right\rangle = \left\langle u(x),\, \frac{d}{dx}v(x)\right\rangle + \left\langle \frac{d}{dx}u(x),\, v(x)\right\rangle .$$

Si u y v son iguales

$$\frac{d}{dx}\left\langle u(x),\, u(x)\right\rangle = 2\left\langle u(x),\, \frac{d}{dx}u(x)\right\rangle .$$

Para aplicar esta fórmula, escribimos $\widetilde{F}(x)$ en la siguiente forma

$$\widetilde{F}(x) = \left\langle f - A(x)\alpha,\, f - A(x)\alpha\right\rangle ,$$

por lo tanto

$$(\operatorname{grad}\widetilde{F})_j = \frac{\partial \widetilde{F}}{\partial \xi_j} = 2\left\langle f - A(x)\alpha,\, \frac{\partial}{\partial \xi_j}(f - A(x)\alpha)\right\rangle$$

$$= 2\left\langle f - A(x)\alpha,\, -\frac{\partial A}{\partial \xi_j}\alpha\right\rangle$$

Haciendo $\varepsilon = f - A(x)\alpha$, queda

$$(\operatorname{grad}\widetilde{F})_j = -2\left\langle \varepsilon,\, \frac{\partial A}{\partial \xi_j}\alpha\right\rangle$$

$$= -2\alpha^T \frac{\partial A^T}{\partial \xi_j}\varepsilon .$$

Antes de que pasemos a describir el algoritmo que hemos elaborado para construir el gradiente de $\widetilde{F}$ utilizando esta fórmula, observemos que ella implica tener una fórmula para la derivada de un spline con respecto a los nodos, ya que

$$a_{ij} = B_j(t_i) = B_j(x;\ t_i),$$

donde x ha sido introducida en la notación para enfatizar la dependencia de los B-splines del conjunto de nodos. Y así

$$\frac{\partial a_{ik}}{\partial \xi_j} = \frac{\partial}{\partial \xi_j} B_k(x;\ t_i)$$

Algoritmo (GRADF). Dado el vector x de nodos, la matriz $A = A(x)$, el vector f de datos y el vector α de los coeficientes, esta función construye la j-ésima componente del vector $(\text{grad } \widetilde{F})_j = -2\alpha^T \frac{\partial A^T}{\partial \xi_j} \varepsilon$. La función $\widetilde{F}$ es $\widetilde{F}(x) = ||f - A(x)\alpha||_2^2$. Se llama a la subrutina PARCIA que construye la matriz $\frac{\partial A}{\partial \xi_j}$. El vector ε es el residual $\varepsilon = f - A\alpha$.

1) $\varepsilon = f - A\alpha$

2) Llamar a PARCIA que proporciona la matriz $\frac{\partial A}{\partial \xi_j}$

3) GRADF $\longleftarrow -2\,\alpha^T \frac{\partial A}{\partial \xi_j}\,\varepsilon$

FIN

Algoritmo. (PARCIA). Dado el vector x y el conjunto de puntos $\{t_i : i = 1, \ldots m\}$, este algoritmo construye la matriz $\frac{\partial A}{\partial \xi_j}$, derivada parcial de la matriz A con respecto al nodo ξ_j. La dimensión de ambas matrices es $m \times n$. Se llama a las subrutinas CORR y DERIB. La salida, que es $\frac{\partial A}{\partial \xi_j}$ queda almacenada en la matriz APAR.

1) Para $k = 1, \ldots, n$

   1) Llamar a CORR (k, j, x, np)

      1) Para $i = 1, \ldots, m$

         APAR$(i,j) \longleftarrow$ DERIB (np, x, t_i)

FIN

La subrutina CORR, al igual que MOVI, es un ingenioso artificio para hacer un corrimiento de índices y así, disponiendo de una sola subrutina para evaluar la derivada de un B-spline, utilizarla para calcular la derivada parcial de un spline cualquiera B_i con respecto un nodo cualquiera ξ_j. En el caso de CORR, si se desea evaluar en t el B-spline

B_i con nodos $\xi_i, \xi_{i+1}, \ldots, \xi_{i+4}$, o su derivada con respecto al nodo ξ_j, se aplica el siguiente

Algoritmo. (CORR) Esta subrutina hace el corrimiento de índices $\xi_1 \leftarrow \xi_i$, $\xi_2 \leftarrow \xi_{i+1}, \ldots, \xi_5 \leftarrow \xi_{i+4}$, con el fin de llamar a BASICO o a DERIB. El índice np es el corrimiento del índice j correspondiente al corrimiento realizado del índice i.

1) Para $k = 0, 1, \ldots, 4$

   $\xi_{1+k} \leftarrow \xi_{i+k}$

2) $np \leftarrow j - i + 1$

3.3 Subrutina DERIB.

3.3.1. Diferencias divididas con argumentos coincidentes.

Utilizando la subrutina DERIB, que vamos a presentar en esta sección, vamos a evaluar las derivadas de un B-spline con respecto a los nodos. Recordemos en primer lugar que hemos dicho que para evaluar B-splines lo mejor que podemos hacer, desde el punto de vista computacional es utilizar su expresión como una diferencia dividida. A partir de esta expresión podemos obtener también elegantes fórmulas para las

derivadas parciales de un B-spline con respecto a los nodos. Estas fórmulas incluyen diferencias divididas con nodos coincidentes; por ello nos es necesario hacer una generalización de las diferencias divididas que incluya el caso de abscisas coincidentes.

Antes de hacer una generalización de las diferencias divididas que incluya el caso de argumentos coincidentes, queremos recordar que dos propiedades fundamentales de las diferencias divididas usuales, son:

(i) $[x_1, x_2, \ldots, x_n]f$ es una función simétrica de sus argumentos, es decir, depende únicamente de los valores x_1, $x_2, \ldots, x_n$, pero no del orden en que aparecen.

(ii) Si $f \in C^{(n-1)}$ entonces $[x_1, x_2, \ldots, x_n]f$ es una función continua de sus n argumentos.

Las diferencias divididas pueden generalizarse de la siguiente manera:

Definición. Sean $\{x_1, x_2, \ldots, x_n\}$ cualesquiera números reales.

$$[x_1, x_2, \ldots, x_n]f =$$

$$= \begin{cases} \dfrac{f^{(n-1)}(x_1)}{(n-1)!} & \text{si } x_1 = x_2 = \ldots = x_n \quad \text{y } f \in C^{(n-1)}; \\ \dfrac{[x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n]f - [x_1, \ldots, x_{s-1}, x_{s+1}, \ldots, x_n]f}{x_s - x_r} & \\ & \text{si } x_r \neq x_s \end{cases}$$

Teorema. Se cumple la propiedad simétrica de las diferencias divididas. Es decir, si $\{i_1, i_2, \ldots, i_n\}$ es cualquier arreglo de $\{1,2,\ldots, n\}$ entonces

$$[x_{i_1}, x_{i_2}, \ldots, x_{i_n}]f = [x_1, x_2, \ldots, x_n]f$$

Demostración. Por inducción sobre n. Para $k = 1$, no hay nada que demostrar. Para $k = 2$; si $x_1 = x_2$, es obvio por la definición. Si $x_1 \neq x_2$, entonces $[x_1, x_2]f$, por la definición, tiene cualquiera de los siguientes valores

$$\frac{[x_1]f - [x_2]f}{x_1 - x_2} \quad , \quad 0 \quad , \quad \frac{[x_2]f - [x_1]f}{x_2 - x_1}$$

que son iguales, y además son los mismos que puede tomar $[x_2, x_1]f$. Es decir

$$[x_1, x_2]f = [x_2, x_1]f.$$

Supongamos que vale para k, entonces, para el paso de inducción tenemos dos casos: (i) si los $(k+1)$ argumentos son iguales y $f \in C^{(k)}$, entonces

$$[x_1, \ldots, x_{k+1}]f = [x_{i_1} \ldots, x_{i_{k+1}}]f$$

$$= \frac{f^{(k)}(x_1)}{k!}$$

para cualquier ordenamiento de los argumentos. (ii) Si hay dos distintos, $x_p \neq x_q$, digamos y si $\{x_1, i_2, \ldots, i_{k+1}\}$ es un arreglo de $\{1, 2, \ldots, k+1\}$, donde $i_p = p$ e $i_q = q$, entonces podemos hacer

$$[x_{i_1}, x_{i_2}, \ldots, x_{i_{k+1}}]f =$$

$$= \frac{[x_{i_1}, \ldots, x_{i_p-1}, x_{i_p+1}, \ldots, x_{i_{k+1}}]f - [x_{i_1}, \ldots, x_{i_q-1}, x_{i_q+1}, \ldots, x_{i_{k+1}}]f}{x_{i_q} - x_{i_p}}$$

$$= \frac{[x_{i_1},\ldots,x_{i_p-1},x_{i_p+1},\ldots x_{i_{k+1}}]f - [x_{i_1},\ldots,x_{i_q-1},x_{i_q+1},\ldots,x_{i_{k+1}}]f}{x_q - x_p}$$

y por la hipótesis de inducción

$$= \frac{[x_1,\ldots,x_{p-1},x_{p+1},\ldots,x_{k+1}]f - [x_1,\ldots,x_{q-1},x_{q+1},\ldots,x_{k+1}]f}{x_q - x_p}$$

$$= [x_1, x_2,\ldots, x_{k+1}]f.$$

Con lo cual se completa la demostración.

Observaciones. Cuando los puntos $\{x_i\}$ son todos distintos, la definición anterior coincide con las diferencias divididas usuales y por lo tanto son los coeficientes en el polinomio de interpolación de Newton, cumpliéndose la propiedad de continuidad con respecto a los argumentos, etc. Cuando los puntos no son todos distintos, las diferencias divididas que se acaban de definir aparecen y se estudian en interpolación osculatoria y en ese contexto se puede demostrar las propiedades, por ejemplo, de simetría, continuidad, diferenciabilidad, etc.

3.3.2 Derivada de una diferencia dividida.

Ahora vamos a obtener la derivada de una diferencia dividida de una función f con respecto a uno de sus argumentos.

$$\frac{\partial}{\partial x_i}[x_1, x_2, \ldots, x_n]f =$$

$$= \lim_{h \to 0} \frac{[x_1, \ldots, x_i + h, \ldots, x_n]f - [x_1, \ldots, x_i, \ldots, x_n]f}{h}$$

$$= \lim_{h \to 0} \frac{[x_1, \ldots, x_i + h, \ldots, x_n]f - [x_1, \ldots, x_i, \ldots, x_n]f}{(x_i + h) - x_i}$$

$$= \lim_{h \to 0} [x_1, \ldots, x_i, x_i + h, \ldots, x_n]f,$$

por la definición de diferencia dividida. Y finalmente, por la continuidad con respecto a los argumentos:

$$\frac{\partial}{\partial x_i}[x_1, \ldots, x_n]f = [x_1, \ldots, x_i, x_i, \ldots, x_n]f.$$

Esta es la fórmula que buscábamos. Nos dice que la derivada parcial con respecto a la variable x_i, es una diferencia dividida de un orden mayor en la que aparece repetida la variable x_i.

3.3.3 Derivada de un B-spline con respecto a los nodos

Por lo que dijimos cuando hablamos de la subrutina CORR, únicamente necesitamos las derivadas del B-spline $B_1(t)$. Recordemos su expresión

$$B_1(t) = (\xi_5 - \xi_1)[\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5]\,(x-t)_+^3.$$

Derivemos con respecto al primer nodo:

$$\frac{\partial B_1}{\partial \xi_1} = (\xi_5 - \xi_1)\frac{\partial}{\partial \xi_1}[\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5](x-t)_+^3 - [\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5](x-t)_+^3$$

$$= (\xi_5 - \xi_1)[\xi_1\ \xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5](x-t)_+^3 - [\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5](x-t)_+^3$$

$$= (\xi_5 - \xi_1)\frac{[\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5](x-t)_+^3 - [\xi_1\ \xi_1\ \xi_2\ \xi_3\ \xi_4](x-t)_+^3}{\xi_5 - \xi_1} -$$

$$- [\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5](x-t)_+^3$$

$$= -[\xi_1\ \xi_1\ \xi_2\ \xi_3\ \xi_4](x-t)_+^3 .$$

Las derivadas con respecto a ξ_2, ξ_3 ξ_4 son más sencillas:

$$\frac{\partial B_1}{\partial \xi_2} = (\xi_5 - \xi_1)\frac{\partial}{\partial \xi_2}[\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5](x-t)_+^3$$

$$= (\xi_5 - \xi_1)[\xi_1\ \xi_2\ \xi_2\ \xi_3\ \xi_4\ \xi_5](x-t)_+^3 .$$

$$\frac{\partial B_1}{\partial \xi_3} = (\xi_5 - \xi_1)[\,\xi_1\ \xi_2\ \xi_3\ \xi_3\ \xi_4\ \xi_5\,]\,(x-t)_+^3 .$$

$$\frac{\partial B_1}{\partial \xi_4} = (\xi_5 - \xi_1)[\,\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_4\ \xi_5\,](x-t)_+^3 .$$

Y, finalmente

$$\frac{\partial B_1}{\partial \xi_5} = (\xi_5 - \xi_1)\,\frac{\partial}{\partial \xi_5}[\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5](x-t)_+^3 + [\,\xi_1\ \xi_2\ \xi_3\ \xi_4\xi_5\ (x-t)_+^3$$

$$= (\xi_5 - \xi_1)[\,\xi_1\ \xi_2\ \xi_3\xi_4\xi_5\ \xi_5](x-t)_+^3 + [\,\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5\,]\,(x-t)_+^3$$

$$= (\xi_5 - \xi_1)\,\frac{[\,\xi_2\ \xi_3\ \xi_4\ \xi_5\ \xi_5\,]\,(x-t)_+^3 - [\,\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5\,]\,(x-t)_+^3}{\xi_5 - \xi_1}$$

$$+ [\,\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5\,]\,(x-t)_+^3$$

$$= [\,\xi_2\ \xi_3\ \xi_4\ \xi_5\ \xi_5\,]\,(x-t)_+^3 .$$

Estas fórmulas son realizadas en el siguiente algoritmo.

Algoritmo. (DERIB). Evalúa en t la derivada parcial del B-spline cúbico con nodos $\xi_1, \xi_2, \xi_3, \xi_4, \xi_5$ con respecto al nodo ξ_{np}.

1) Si $(np > 5)$ DERIB $\leftarrow$ 0; Salida

2) Si $(np < 1)$ DERIB $\leftarrow$ 0; Salida

3) Si $(t > \xi_5)$ DERIB $\leftarrow$ 0; Salida

4) Si $(t < \xi_1)$ DERIB $\leftarrow$ 0; Salida

5) Para $i = 2, \ldots, 5$

   $G_i \leftarrow (\xi_i - t)_+^3$

6) Para $i = 1, \ldots, 4$

   $G_i \leftarrow (G_i - G_{i+1})/(\xi_i - \xi_{i+1})$

7) Para $i = 5, 4, \ldots, np + 1$

   $G_i \leftarrow G_{i-1}$

8) Si $((\xi_{np} - t) > 0)$ $G_{np} \leftarrow 3(\xi_{np} - t)^2$

   Si $((\xi_{np} - t) \leq 0)$ $G_{np} \leftarrow 0$

9) Para $i = 6, 5, \ldots, np + 1$

   $\xi_i \leftarrow \xi_{i-1}$

```
10) | Para j = 2,..., 5
    |
    |   | Para i = 1,..., 6 - j
    |   |
    |   | G_i ← (G_i - G_{i+1})/(ξ_i - ξ_{i+j})
    |
    | Si (np = 1  y  j = 4) pasa a 11)
    |
    | Si (np = 5  y  j = 4) pasa a 12)
    |
    DERIB ← (ξ_6 - ξ_1). G_1
    Salida
11) DERIB ← - G_1
    Salida
12) DERIB ← G_2
```

3.4 El problema completo.

En la sección anterior, cuando hablamos del ajuste de datos con splines con nodos libres (es decir, variables), mencionamos que el problema no-lineal de suma mínima de cuadrados

$$\min_{\substack{\alpha\in\mathbb{R}^n \\ x\in\Delta_N}} ||f - A(x)\ \alpha\ (x)||_2^2 = \min_{\substack{\alpha\in\mathbb{R}^n \\ x\in\Delta_N}} F(x, \alpha)$$

tiene la interesante propiedad de que las variables se separan. Sin embargo, dijimos que no podíamos explorar esa posi

bilidad en este trabajo. Nuestro enfoque, por ahora, es considerar el problema

$$\min_{\substack{\alpha \in \mathbb{R}^n \\ x \in \Omega_N}} F(x, \alpha)$$

como el problema de encontrar el mínimo de la función $F(x, \alpha)$ de las variables x y α, con la restricción $x \in \Omega_N$.

Para encontrar este mínimo vamos a aplicar uno de los métodos usuales de optimización; uno de los que ha demostrado ventajas es el método Levenberg-Marquardt que será el que aplicaremos y del que tendremos ocasión de hablar un poco más adelante. En su uso, como veremos, en alguna etapa será necesario calcular la matriz Jacobiana de la función objetivo $f - A(x)\alpha$.

Sea $\varepsilon = f - A(x)\alpha$ la función vectorial cuyas componentes son $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$. Podemos escribir

$$\varepsilon = \begin{bmatrix} f_1 - \sum_{i=1}^{n} \alpha_i B_i(t_1) \\ f_2 - \sum_{i=1}^{n} \alpha_i B_i(t_2) \\ \vdots \\ f_m - \sum_{i=1}^{n} \alpha_i B_i(t_m) \end{bmatrix}$$

así que la matriz Jacobiana de la función ε es

$$J = \begin{bmatrix} \frac{\partial \varepsilon_1}{\partial \alpha_1} & \frac{\partial \varepsilon_1}{\partial \alpha_2} & \cdots & \frac{\partial \varepsilon_1}{\partial \alpha_n} & \frac{\partial \varepsilon_1}{\partial \xi_1} & \frac{\partial \varepsilon_1}{\partial \xi_2} & \cdots & \frac{\partial \varepsilon_1}{\partial \xi_{n-4}} \\ \frac{\partial \varepsilon_2}{\partial \alpha_1} & \frac{\partial \varepsilon_2}{\partial \alpha_2} & \cdots & \frac{\partial \varepsilon_2}{\partial \alpha_n} & \frac{\partial \varepsilon_2}{\partial \xi_1} & \frac{\partial \varepsilon_2}{\partial \xi_2} & \cdots & \frac{\partial \varepsilon_2}{\partial \xi_{n-4}} \\ & & & & & & & \\ \frac{\partial \varepsilon_m}{\partial \alpha_1} & \frac{\partial \varepsilon_m}{\partial \alpha_2} & \cdots & \frac{\partial \varepsilon_m}{\partial \alpha_n} & \frac{\partial \varepsilon_m}{\partial \xi_1} & \frac{\partial \varepsilon_m}{\partial \xi_2} & \cdots & \frac{\partial \varepsilon_m}{\partial \xi_{n-4}} \end{bmatrix}$$

$$= -\begin{bmatrix} B_1(t_1) \cdots B_n(t_1) & \sum_{i=1}^{n} \alpha_i \frac{\partial B_i}{\partial \xi_1}(t_1) \cdots & \sum_{i=1}^{n} \alpha_i \frac{\partial B_i}{\partial \xi_{n-4}}(t_1) \\ B_1(t_2) \cdots B_n(t_2) & \sum_{i=1}^{n} \alpha_i \frac{\partial B_i}{\partial \xi_1}(t_2) \cdots & \sum_{i=1}^{n} \alpha_i \frac{\partial B_i}{\partial \xi_{n-4}}(t_2) \\ \cdot \quad \cdot & \cdot \quad \cdot & \cdot \quad \cdot \\ \cdot \quad \cdot & \cdot \quad \cdot & \cdot \quad \cdot \\ \cdot \quad \cdot & \cdot \quad \cdot & \cdot \quad \cdot \\ B_1(t_m) \cdots B_n(tm) & \sum_{i=1}^{n} \alpha_i \frac{\partial B_i}{\partial \xi_1}(tm) \cdots & \sum_{i=1}^{n} \alpha_i \frac{\partial B_i}{\partial \xi_{n-4}}(tm) \end{bmatrix}$$

Hemos nosotros elaborado el siguiente algoritmo para construir la matriz Jacobiana

Algoritmo. (JAC) Se construye la matriz Jacobiana FJAC de dimensiones $m \times (2n-4)$ de la función $\varepsilon = f - A(x)\alpha$. $(n-4)$ es el número de nodos interiores componentes del vector $x = (\xi_5, \xi_6, \ldots, \xi_n)^T$. El vector $\alpha = (\alpha_1, \ldots, \alpha_n)^T$ tiene n componentes. La variable entera n es la dimensión del espacio.

1) Para $j = 1, 2, \ldots, n$

   1) Si $(j \leqslant n)$ pasa a 3

   2) Para $i = 1, \ldots, m$

      $$FJAC(i,j) \leftarrow - \sum_{k=1}^{n} \alpha_k \frac{\partial B_k}{\partial \xi_{j-(n-4)}} (t_i)$$

      Salida

   3) Para $i = 1, \ldots, m$

      $$FJAC(i,j) \leftarrow - B_j(t_i)$$

3.5 El método Levenberg-Marquardt.

Vamos a incluir aquí un breve resumen de las ideas que conducen al altoritmo de Levenberg-Marquardt, en especial de la modificación hecha por Moré [14] al algoritmo general, ya

que es el algoritmo así modificado, el que nosotros utilizamos para resolver el problema de suma mínima de cuadrados no lineal que hemos planteado más arriba.

Sea $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$, una función continuamente diferenciable en un subconjunto compacto Ω de $\mathbb{R}^n$. Se requiere encontrar $w^* \in \Omega$ tal que

$$||F(w^*)|| = \min_{w \in \Omega} ||F(w)||.$$

Para encontrar tal w^* podemos considerar el siguiente proceso: generar una sucesión $\{w_k\}$ tal que

$$||F(w_{k+1})|| < ||F(w_k)||.$$

Si $w_0 \in \Omega$, entonces la aproximación lineal a $F(w)$ es

$$F(w) \approx F(w_0) + F'(w_0)(w - w_0)$$

donde $F'(w_0)$ es la matriz Jacobiana. De acuerdo a esto, el problema es aproximado en la forma

$$\min_{w \in \Omega} ||F(w)|| \approx \min_{w \in \Omega} ||F(w_0) + F'(w_0)(w - w_0)||$$

y si hacemos $p = w - w_0$, queda

$$\min_{p} \; ||F'(w_0)p + F(w_0)||.$$

Esto es un problema lineal de suma mínima de cuadrados. Sin embargo, la solución pudiera quedar fuera de la vecindad de w_0 donde es válida la aproximación lineal y donde la Jacobiana es positiva definida. La familia de métodos de paso restringido, a la cual pertenece el Levenberg-Marquardt, somete al nuevo valor w_1 a estar restringido a una vecindad de w_0:

$$B = \{w: \; ||w - w_0|| \leq \alpha\},$$

donde α es un valor que puede determinarse por la experiencia y la función $F(w)$, de manera que la aproximación lineal sea válida en B. Así, el problema se ha convertido en

$$\min_{p} \; ||F'(w_0)p + F(w_0)||, \text{ con } \; ||p|| \leq \alpha,$$

o bien, usando la notación $A_0 = F'(w_0)$, $b_0 = -F(w_0)$:

$$\min_{p} \; ||A_0 p - b_0||, \text{ con } ||p|| \leq \alpha$$

Introduzcamos el parámetro z tal que

$$||p||^2 = \alpha^2 - z^2.$$

Así pues, tenemos el problema

$$\min_{p} \; ||A_0 p - b_0||^2, \quad \text{con} \quad ||p||^2 = \alpha^2 - z^2,$$

equivalente al anterior. Aplicando el método de los multiplicadores de Lagrange, tal problema se convierte en el de resolver para p el siguiente sistema de ecuaciones

$$(A_0^T A_0 + \lambda I)p = - A_0^T b_0$$

$$||p||^2 = \alpha^2 - z^2,$$

donde λ es la constante de Lagrange.

Existen dos casos:

i) Si $z \neq 0$, $\lambda = 0$

Se tiene entonces

$$(A_0^T A_0)p = A_0^T b_0$$

que son las ecuaciones normales para el problema de suma mínima de cuadrados sin restricciones. En este caso $||p|| < \alpha$ y la solución p^* de norma mínima viene dada por el proceso al límite

$$p^* = \lim_{\lambda \to 0} (A_0^T A_0 + \lambda I)^{-1} A_0^T b_0$$

ii) Si $\lambda \neq 0$, $z = 0$

Se tiene entonces el problema

$$\min_p ||A_0 p - b_0||, \quad \text{con} \quad ||p||^2 = \alpha^2,$$

ya descrito.

Una vez que hemos obtenido p^*, hacemos

$$w_1 = w_0 + p^*,$$

que esperamos esté más cerca del mínimo buscado. En general, a partir de w_k, obtenemos

$$w_{k+1} = w_k + p_k^*.$$

Pero, como hemos dicho, α es un valor que puede depender de w_k, así que puede ser necesario cambiarlo de una iteración a otra. Esto nos permite, en caso de haber buena concordancia, aumentar α para el siguiente paso y así avanzar más rápido; si la concordancia es mala se requerirá disminuir α.

Estamos ahora en condiciones de esbozar el algoritmo de Levenberg-Marquardt.

Algoritmo (Levenberg-Marquardt).

Dados w_0, α_0

Para $k = 0, 1, 2, \ldots$

1) Determinar $\lambda_k \geq 0$ tal que

$$(A_k^T A_k + \lambda_k I) p_k = - A_k^T b_k$$

$$||p_k|| \leq \alpha_k$$

y si $||p_k|| < \alpha_k$ entonces $\lambda_k = 0$

2) Si $||F(w_k + p_k)|| < ||F(w_k)||$, $w_{k+1} = w_k + p_k$

si no $w_{k+1} = w_k$

3) Calcular α_{k+1}.

FIN

Cálculo de λ.

Teniendo en cuenta lo anterior, podemos escribir

$$p(\lambda) = (A^T A + \lambda I)^{-1} A^T b$$

y entonces quisieramos obtener el mínimo del valor absoluto de la función

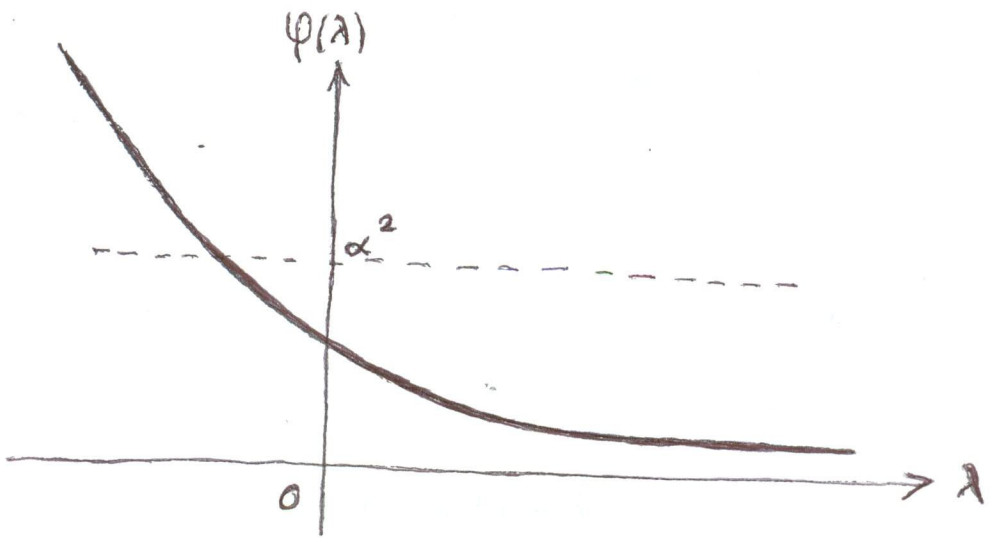
$$\phi(\lambda) = \left|\left|(A^T A + \lambda I)^{-1} A^T b\right|\right|^2 - \alpha^2.$$

Si usamos la descomposición singular de A, $A = U \Sigma V^T$, entonces

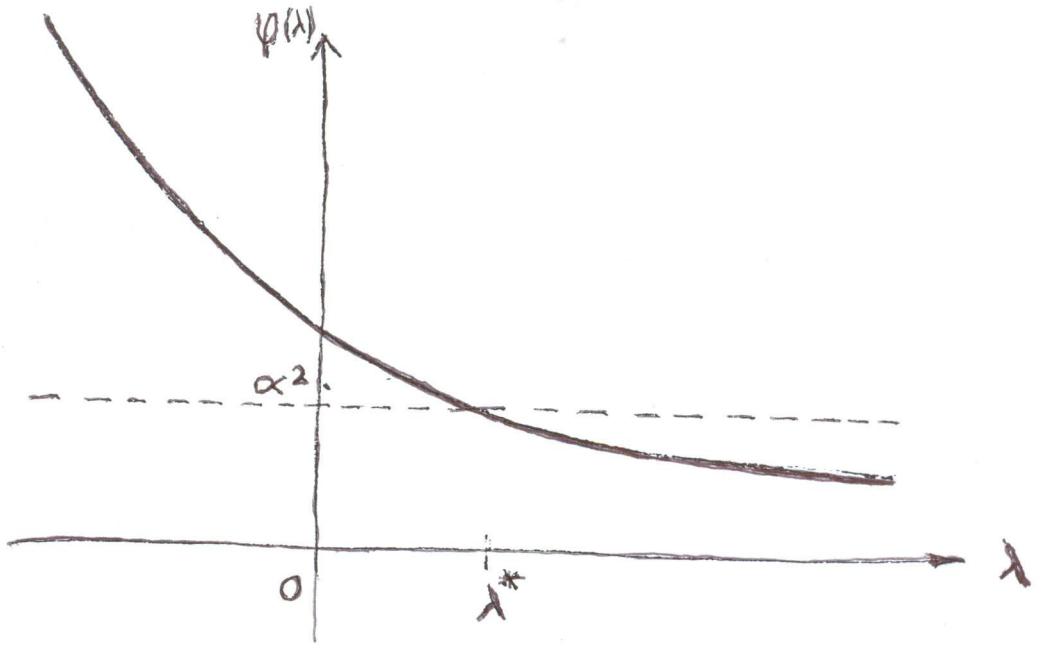
$$\phi(\lambda) = \sum_{i=1}^{n} \left(\frac{s_i c_i}{s_i^2 + \lambda}\right)^2 - \alpha^2$$

donde $s_1, \ldots, s_n$ son los valores singulares de A y $A^T b = (c_1, \ldots, c_n)^T$.

Sea $$\varphi(\lambda) = \sum_{i=1}^{n} \left(\frac{s_i c_i}{s_i^2 + \lambda}\right)^2$$



Es claro que si $\varphi(0) < \alpha^2$, entonces $\lambda = 0$ es la solución buscada ya que $\varphi(\lambda)$ es decreciente. Si $\varphi(0) > \alpha^2$, entonces, como $\varphi(\lambda) \to 0$ cuando $\lambda \to \infty$, se tendrá una única solución positiva λ^*.



Además, como $\phi(\lambda)$ es convexa en el intervalo $(0, \infty)$ se pensaría en usar el método de Newton para encontrar la solución λ^*. Pero Moré sugiere usar otra estrategia para aprovechar mejor la estructura del problema. Tal es suponer que

$$\psi(\lambda) \equiv [\phi(\lambda)]^{1/2}$$

puede ser aproximada por

$$\hat{\phi}(\lambda) = \frac{a}{b+\lambda}$$

donde a, b son constantes tales que

$$\hat{\phi}(\lambda_k) = \psi(\lambda_k) \quad y \quad \hat{\phi}'(\lambda_k) = \psi'(\lambda_k).$$

De acuerdo a esto se obtiene

$$a = - \frac{[\psi(\lambda_k)]^2}{\psi'(\lambda_k)} ; \qquad b = - \lambda_k - \frac{\psi(\lambda_k)}{\psi'(\lambda_k)} ,$$

y para encontrar la raíz de $\hat{\phi}(\lambda) - \alpha = 0$ aplicamos la siguiente iteración

$$\lambda_{k+1} = \lambda_k - \frac{\psi(\lambda_k)}{\psi'(\lambda_k)} \left[\frac{\alpha + \psi(\lambda_k)}{\alpha} \right] .$$

Pero es necesario protegerla para realmente obtener convergencia. Con este propósito se usan en cada iteración cotas inferiores y superiores ℓ_k y u_k.

Con el fin de obtener u_0 observemos que, como

$$||p^*|| = ||(A^T A + \lambda^* I)^{-1} A^T b|| = \alpha$$

entonces

$$\alpha \leq ||(A^T A + \lambda^* I)^{-1}|| \; ||A^T b||$$

pero

$$||(A^TA + \lambda^*I)^{-1}|| = ||(\Sigma^T\Sigma + \lambda^*I)^{-1}||$$

$$= ||(diag(s_1^2 + \lambda^*, s_2^2 + \lambda^*, \ldots, s_n^2 + \lambda^*)^{-1}||$$

$$= || diag((s_1^2 + \lambda^*)^{-1}, (s_2^2 + \lambda^*)^{-1}, \ldots, (s_n^2 + \lambda^*)^{-1})||$$

$$= (s_n^2 + \lambda^*)^{-1}$$

$$\leqslant \lambda^{*-1};$$

por lo tanto

$$\lambda^* \leqslant ||A^T b||/\alpha;$$

y la cota superior adecuada es

$$u_0 = ||A^T b||/\alpha$$

Para calcular ℓ_0, notemos que, como $\hat{\phi}$ es convexa, si $\psi'(0) \neq 0$ tenemos que el valor obtenido al realizar una iteración con el método de Newton y $\lambda_0 = 0$, es menor o igual que λ^*, entonces

$$\lambda^* \geqslant -\frac{\psi(0)}{\psi'(0)},$$

así que

$$\ell_0 = -\frac{\psi(0)}{\psi'(0)};$$

en caso que $\psi'(0) = 0$, $\ell_0 = 0$.

Con estas observaciones, λ^* se puede calcular con el siguiente algoritmo

Algoritmo. Dado λ_0,

Para $k = 0, 1, 2, \ldots$

1. Evaluar $\psi(\lambda_k)$, $\psi'(\lambda_k)$

2. $u_{k+1} = \lambda_k$ si $\psi(\lambda_k) < \alpha$
   $u_{k+1} = u_k$ en otro caso

3. $\ell_{k+1} = \max \{\ell_k, \lambda_k - \psi(\lambda_k)/\psi'(\lambda_k)\}$

4. Obtener λ_{k+1} de la fórmula

$$\lambda_{k+1} = \lambda_k - \frac{\psi(\lambda_k)}{\psi'(\lambda_k)}\left[\frac{\alpha - \psi(\lambda_k)}{\alpha}\right]$$

En la práctica se acepta λ_k como una buena aproximación a λ^* si

$$\left|\frac{\hat{\phi}(\lambda_k) - \alpha}{\alpha}\right| \leq \Delta$$

con Δ cierta tolerancia que comunmente no es muy pequeña ya que, de cualquier manera, esta aproximación forma parte de

otra aproximación (la linealización) que pudiera ser muy burda. Una elección usual es $\Delta = 0.1$.

Actualización de α.

Como ya mencionamos, si la concordancia en una iteración del algoritmo de Levenberg-Marquardt es buena, en la siguiente podemos aumentar el valor de α; en caso contrario α debe permancer igual o disminuir. La medida de esta concordancia está dada por

$$\rho(p) = \frac{||F(x)||^2 - ||F(x+p)||^2}{||F(x)||^2 - ||F(x)+Ap||^2}$$

Si ρ es pequeña ($\rho \leq 1/4$), la concordancia es mala y se debe disminuir α; si ρ se acerca a 1 ($\rho \geq 3/4$), la concordancia es buena, se puede aumentar α; si $1/4 < \rho < 3/4$ la concordancia se puede considerar aceptable y dejar α sin cambio. Estos criterios provienen de una larga experiencia de trabajo con este algoritmo en una gran variedad de problemas, [14], [6].

Hagamos la siguiente simplificación obvia en la escritura, con el fin de determinar una manera estable de calcular ρ:

$$\rho = \frac{||f||^2 - ||f_+||^2}{||f||^2 - ||f+Ap||^2}$$

como

$$(A^TA + \lambda I)p = - A^Tf$$

entonces

$$||Ap||^2 + \lambda ||p||^2 = - p^TA^Tf$$

por lo cual

$$||f||^2 - ||f + Ap||^2 = ||f||^2 - ||f||^2 - 2p^TA^Tf - ||Ap||^2$$

$$= -2p^T A^Tf - ||A p||^2$$

$$= 2||A p||^2 + 2\lambda ||p||^2 - ||A p||^2$$

$$= ||A p||^2 + 2\lambda ||p||^2 \qquad ...(1)$$

y, por tanto

$$\rho = \frac{||f||^2 - ||f_+||^2}{||A p||^2 + 2\lambda||p||^2} = \frac{1 - (\frac{||f_+||}{||f||})^2}{(\frac{||A p||}{||f||})^2 + 2(\lambda^{1/2} \frac{||p||}{||f||})^2}$$

De (1) vemos que

$$||A p|| \leq ||f|| \quad y \quad \lambda^{1/2}||p|| \leq ||f||,$$

por lo cual el cálculo del denominador es muy estable. En el numerador, si $||f_+|| \gg ||f||$ se puede generar un nú-

mero demasiado grande para nuestro instrumento de cálculo, pero en ese caso $\rho < 0$, y como sólo nos interesan valores positivos, ponemos $\rho = 0$.

El algoritmo es detenido cuando

$$||w_{k+1} - w_k|| \leq ||w_k|| 10^{-t}$$

donde t es el número de dígitos de la aritmética de la máquina. Presentamos a continuación el algoritmo global

Algoritmo de Levenberg-Marquardt.

Entrada w_0, α_0, Δ, t

1. Para $k = 0, 1, 2, \ldots,$

   1. Evaluar A_k, b_k

   2. Calcular SVD de A_k

   3. Resolver

      $$\min ||A_k p_k - b_k||^2$$

      si $||p_k|| \leq \alpha_k$ entonces $\lambda_k = 0$ y hacer 1.7

   4. $\lambda_0' = \lambda_k$, $u_0 = ||A_k^T b_k||/\alpha_k$

      $\ell_0 = -\psi(0)/\psi'(0)$ si $\psi'(0) \neq 0$, $\ell_0 = 0$ en otro caso.

5. Para $j = 0, 1, 2, \ldots$

   1. Evaluar $\psi(\lambda'_j)$, $\psi'(\psi'_j)$

   2. $u_{j+1} = \lambda'_j$ si $\psi(\lambda'_j) < \alpha_k$

      $u_{j+1} = u_j$ en otro caso

   3. $\ell_{j+1} = \max\{\ell_j, \lambda'_j - \psi(\lambda'_j)/\psi'(\lambda'_j)\}$

   4. $\lambda'_{j+1} = \lambda'_j - \dfrac{\psi(\lambda'_j)}{\psi'(\lambda'_j)} \quad [\dfrac{\alpha_k - \psi(\lambda'_j)}{\alpha_k}]$

   5. Si $|\hat{\phi}(\lambda'_j) - \alpha_k| \leq \alpha_k \Delta$

      entonces $\lambda_k = \lambda_j$ y hacer 1.6, de otra manera incrementar j y hacer 1.5

6. Calcular $p_k = (A_k^T A_k + \lambda_k I)^{-1} A_k^T b_k$

7. Si $||F(\omega_k + p_k)|| < ||F(\omega_k)||$

   entonces $\omega_{k+1} = \omega_k + p_k$ de otro modo $\omega_{k+1} = \omega_k$

8. Si $||F(\omega_k + p_k)|| \geq ||F(\omega_k)||$, $\rho = 0$ de otro modo

$$\rho = \frac{1 - \left[\dfrac{||F(\omega_k + p_k||}{||F(\omega_k)||}\right]^2}{\left[\dfrac{||A p_k||}{||F(\omega_k)||}\right]^2 + 2\left[\lambda_k^{1/2} \dfrac{||p_k||}{||F(\omega_k)||}\right]^2}$$

9. Hacer α_{k+1} de acuerdo al valor de ρ.

10. Si $||w_{j+1} - w_j|| \leq 10^{-t} ||w_j||$

$w^* = w_{k+1}$ y FIN

De otra manera incrementar k y repetir 1.

FIN

3.6 La transformación σ.

Ahora debemos enfrentar la siguiente dificultad: el método Levenberg-Marquardt es un método de optimización sin restricciones y el problema que queremos resolver

$$\min_{\substack{\alpha \in R^n \\ x \in \Delta_N}} F(x, \alpha)$$

tiene la restricción $x \in \Delta_N$. Hay dos soluciones, o usamos un método de optimización con restricciones o eliminamos las restricciones.

Efectivamente, en una serie de experimentos que realizamos en la B-7800 nos encontramos con el desagradable resultado que en algunos casos al llevarse a cabo el proceso de optimización de los nodos libres con el paquete de Moré,

los nodos, como una parte de las variables de la función objetivo quedaban desordenados, es decir, en ocasiones se alteraba la relación $\xi_4 < \xi_5 < \ldots < \xi_{n+1}$, y aún en ocasiones algunos nodos interiores quedaban fuera del intervalo limitado por los dos postes fijos ξ_4 y ξ_{n+1}. Esto es inaceptable en nuestro contexto, es una solución sin sentido para nosotros.

El siguiente paso dado por Jupp [12] es una elegante solución a esta dificultad. El propone un cambio de variables, una transformación, que convierte al problema con restricciones en una formulación sin restricciones. Tal transformación es la siguiente: definamos las variables

$$\sigma_i = \ell n \frac{h_{i+1}}{h_i}, \quad i = 1, 2, \ldots, N$$

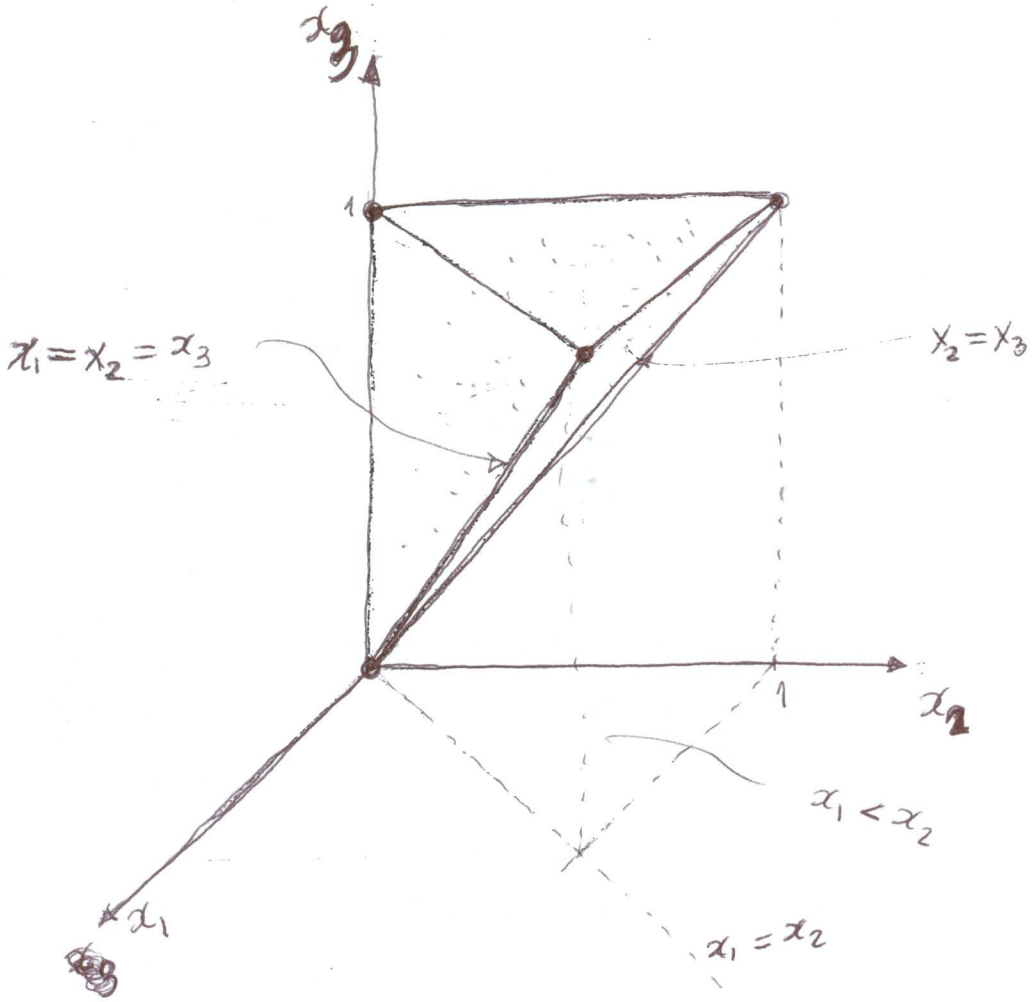
donde N es el número de nodos interiores,

y

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

$$x_i = \xi_{i+4} .$$

Antes de continuar, veamos como ejemplo $S_3[0, 1] = \{x = (x_1, x_2, x_3): 0 < x_1 < x_2 < x_3 < 1\}$, cuya cerradura es el simplejo en R^3 que se ve en la siguiente figura



En general, como aquí, las caras y lados de $\overline{s}_N[a, b]$ contienen aquellas subdivisiones de $[a, b]$ que corresponden a nodos coincidentes. El simplejo $\overline{s}_N$ queda también caracterizado por el sistema de $(N+1)$ desigualdades lineales siguiente

$$x_p - x_{p-1} \geqslant 0, \quad p = 1, 2, \ldots, N+1.$$

Los parámetros $(\sigma_1, \sigma_2, \ldots, \sigma_N)$ son independientes de un cambio lineal en la escala de los nodos originales, lo

cual proporciona un escalamiento automático en los métodos iterativos usados para localizar los nodos óptimos.

Para encontrar x a partir de una σ arbitraria, hacemos

$$p_i = e^{\sigma_i} \quad , \quad i = 1, 2, \ldots, N$$

y

$$Z = 1 + p_1 + p_1 p_2 + \ldots + p_1 p_2 \ldots p_N .$$

Entonces

$$h_1 = \frac{b - a}{Z}$$

y

$$h_{i+1} = p_i \, h_i , \quad i = 1, 2, \ldots, N$$

por lo siguiente:

$$\frac{b - a}{Z} = \frac{x_{N+1} - x_0}{Z} = \frac{x_{N+1} - x_0}{1 + p_1 + p_1 p_2 + \ldots + p_1 p_2 \ldots p_N}$$

$$= \frac{x_{N+1} - x_0}{1 + e^{\sigma_1} + e^{\sigma_1} e^{\sigma_2} + \ldots + e^{\sigma_1} e^{\sigma_2} \ldots e^{\sigma_N}}$$

Pero

$$e^{\sigma_i} = e^{\ln(h_{i+1}/h_i)} = \frac{h_{i+1}}{h_i} ,$$

Por lo que

$$\frac{b-a}{2} = \frac{x_{N+1} - x_0}{1 + \frac{h_2}{h_1} + \frac{h_2}{h_1}\frac{h_3}{h_2} + \ldots + \frac{h_2}{h_1}\frac{h_3}{h_2}\cdots\frac{h_{N+1}}{h_N}}$$

$$= \frac{x_{N+1} - x_0}{1 + \frac{h_2}{h_1} + \frac{h_3}{h_1} + \frac{h_4}{h_1} + \ldots + \frac{h_{N+1}}{h_1}}$$

$$= \frac{h_1 (x_{N+1} - x_0)}{h_1 + h_2 + h_3 + \ldots + h_{N+1}}$$

$$= \frac{h_1 (x_{N+1} - x_0)}{(x_1 - x_0) + (x_2 - x_1) + \ldots + (x_{N+1} - x_N)}$$

$$= h_1$$

Además,

$$p_i h_i = e^{\sigma_i} h_i = \frac{h_{i+1}}{h_i} h_i = h_{i+1}$$

conviene, naturalmente, para evaluar el polinomio Z, utilizar el esquema de Horner y doble precisión.

Ahora bien, la transformación σ, definida por $\ell n(h_{i+1}/h_i)$ es continua y continuamente diferenciable, los lados y las caras de $\overline{S}_N$ donde se da igualdad de nodos se mapean en infinito, y la transformación es sobre todo R^N, como se ve de la transformación inversa que acabamos de escribir, por lo que quedan borradas las restricciones.

Para transformar el gradiente o la Jacobiana a las nuevas variables, podemos usar la regla de la cadena

$$\frac{\partial \tilde{F}}{\partial x_i} = \sum_{j=1}^{N} \frac{\partial \tilde{F}}{\partial \sigma_j} \frac{\partial \sigma_j}{\partial x_i}$$

Puesto que

$$\frac{\partial \sigma_j}{\partial x_i} = \begin{cases} 0 & \text{si } j \neq i-1,\ i,\ i+1 \\ 1/h_i & \text{si } j = i-1 \\ -(1/h_i + 1/h_{i+1}) & \text{si } j = i \\ 1/h_{i+1} & \text{si } j = i+1 \end{cases}$$

entonces $\nabla_\sigma \tilde{F}(\sigma)$ es la solución del sistema de ecuaciones

$$G\, \nabla_\sigma \tilde{F}(\sigma) = \nabla_x \tilde{F}(x),$$

donde G es simétrica y tridiagonal, como se ve a continuación

$$\begin{bmatrix} \frac{\partial \sigma_1}{\partial x_1} & \frac{\partial \sigma_2}{\partial x_1} & \frac{\partial \sigma_3}{\partial x_1} & \cdots & \frac{\partial \sigma_N}{\partial x_1} \\ \frac{\partial \sigma_1}{\partial x_2} & \frac{\partial \sigma_2}{\partial x_2} & \frac{\partial \sigma_3}{\partial x_2} & & \frac{\partial \sigma_N}{\partial x_2} \\ \frac{\partial \sigma_1}{\partial x_3} & \frac{\partial \sigma_2}{\partial x_3} & \frac{\partial \sigma_3}{\partial x_3} & & \frac{\partial \sigma_N}{\partial x_3} \\ \vdots & & & & \\ \frac{\partial \sigma_1}{\partial x_N} & \frac{\partial \sigma_2}{\partial x_N} & \frac{\partial \sigma_3}{\partial x_N} & & \frac{\partial \sigma_N}{\partial x_N} \end{bmatrix} \begin{bmatrix} \frac{\partial F}{\partial \sigma_1} \\ \frac{\partial F}{\partial \sigma_2} \\ \frac{\partial F}{\partial \sigma_3} \\ \\ \frac{\partial F}{\partial \sigma_N} \end{bmatrix} = \begin{bmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \\ \frac{\partial F}{\partial x_3} \\ \\ \frac{\partial F}{\partial x_N} \end{bmatrix}$$

$$G = \begin{bmatrix} -(1/h_1 + 1/h_2) & 1/h_2 & 0 & 0 & \cdots & 0 \\ 1/h_2 & -(1/h_2+1/h_3) & 1/h_3 & 0 & \cdots & 0 \\ 0 & 1/h_3 & -(1/h_3+1/h_4) & 1/h_4 & \cdots & 0 \\ & & \ddots & \ddots & \ddots & \\ & & & & & 1/h_N \\ & & & 1/h_N & & -(1/h_N+1/h_{N+1}) \end{bmatrix}$$

Así pues, tenemos que resolver un sistema de la forma

$$G\,u = k.$$

Para resolver este sistema Jupp sugiere utilizar un algoritmo debido a Rose [16]. Para seguir a Rose, hacemos

$$B = \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & 1 & -1 & & \\ & & & \ddots & \ddots & \\ & & & & 1 & -1 \end{bmatrix} \qquad (N \times (N+1))$$

$$H = \begin{bmatrix} h_1 & & & & \\ & h_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & h_{N+1} \end{bmatrix} \qquad ((N+1) \times (N+1))$$

Se tiene

$$G = BH^{-1}B^T.$$

El altamente eficiente algoritmo propuesto por Rose para resolver el sistema

$$Gu = k$$

es el siguiente:

Algoritmo. (ROSE)

Hacer

$$\theta_j = \sum_{\ell=1}^{j} h_\ell$$

Paso 1. Calcular el escalar s:

$$s = -(\theta_{N+1})^{-1} \sum_{j=1}^{N} \theta_j k_j, \quad (k = (k_1, k_2, \ldots, k_N)^T)$$

Paso 2. Calcular el N-vector v:

$$v_N = k_N + s$$

$$v_j = k_j + v_{j+1}, \quad j = N - 1, \ldots, 1$$

Paso 3. Calcular el N-vector u:

$$u_1 = h_1 v_1,$$

$$u_j = h_j v_j + u_{j-1}, \quad j = 2, \ldots, N.$$

Paso 4. Cambiar de signo

$$u_j = - u_j \quad , \quad j = 1, \ldots, N.$$

Después de que introdujimos esto en el paquete de Moré, se eliminó la dificultad que señalamos anteriormente y que consistía en que al optimizar los nodos, algunos de estos se salían del intervalo o se desordenaban.

Contamos, entonces, con el paquete debido a Moré que realiza el método Levenberg-Marquardt de optimización, adaptado a nuestro problema de splines y modificado con la subrutina SIGMA que hace el cambio de variable que hemos reseñado. Pero, ¿para nuestras necesidades futuras, qué precisión necesitamos? y relacionado con esto, nos planteamos la cuestión de reducir el tiempo de cómputo, medido en términos del número de evaluaciones de la función (NFEV) y del número de evaluaciones de la matriz Jacobiana (NJEV).

Se hicieron gráficas y una serie de experimentos con el fin de determinar, en un número de problemas, la sensibilidad del spline con respecto a los nodos. Se llegó a la conclusión de que tres cifras son suficientes para determinar un spline. Por otra parte, con dicha precisión ya determinada, se logró reducir notablemente el tiempo de cómputo de la manera que se muestra en el Capítulo V.

Las subrutinas que implementan la transformación σ de la que nos hemos estado ocupando y su inversa son las siguientes:

Algoritmo (XSIGMA). Construye la nueva variable $\sigma = (\sigma_1, \ldots, \sigma_N)^T$ a partir de los nodos interiores $x = (\xi_5, \xi_6, \ldots, \xi_{N+4})^T$

1) Para $i = 1, \ldots, N+1$

   $h_i \leftarrow \xi_{i+4} - \xi_{i+3}$

2) Para $i = 1, \ldots, N$

   $\sigma_i \leftarrow \ell n\ (h_{i+1}/h_i)$

Algoritmo (SIGMAX). Construye el arrego x a partir del arreglo σ.

1) Para $i = 1, 2, \ldots, N$

   $p_i \leftarrow \exp(\sigma_i)$

2) $z \leftarrow 1 + p_1 + p_1 p_2 + \ldots + p_1 p_2 \ldots p_N$

3) $h_1 \leftarrow (b-a)/z$

1) | Para $i = 1, 2, \ldots, N$

$h_{i+1} \leftarrow p_i \, h_i$

4) | Para $i = 1, 2, \ldots, N$

$\xi_{i+4} \leftarrow h_i + \xi_{i+3}$

3.7 Ejemplo.

Finalizamos este capítulo con un ejemplo; un ejemplo que ha sido utilizado por de Boor [4] y Jupp [12] a causa del interés que tiene por ser un problema difícil para aproximación polinomial debido al pico abrupto y por contener errores experimentales. Se trata de 49 datos, mostrados en la figura siguiente, y que expresan una propiedad del titanio como función de la temperatura.

Un punto de inicio utilizado por Jupp es

$$x_0 = (724.984, 849.976, 910.008, 976.184, 1042.360)$$

y encuentra después de 20 iteraciones

$$(835.967, 876.402, 898.146, 916.315, 973.908)$$

con un residual de 0.01226

Con el mismo punto inicial, nuestros resultados fueron: con el método Levenberg-Marquardt sin la transformación σ, los nodos quedan desordenados en el punto final (15 iteraciones):

(820.283,903.174,888.211,903.594,990.3153).

Después que agregamos la transformación σ al Levenberg-Marquardt nuestro resultado es notablemente mejorado, eliminándose totalmente la dificultad de los nodos desordenados. Se obtuvo el siguiente punto final (37 iteraciones, residual = 0.08):

(835.457,876.506,898.167,916.280,974.017)

El spline con estos nodos aparece en la figura siguiente.

Nota: Se observa una discrepancia fuerte entre los dos residuales, pero ésta es sólo aparente. Jupp [12] p.339, utiliza la siguiente fórmula para el error:

$$\left(\frac{1}{m-1} \sum_{i=1}^{m} w_i |\varepsilon_i|^2\right)^{1/2}$$

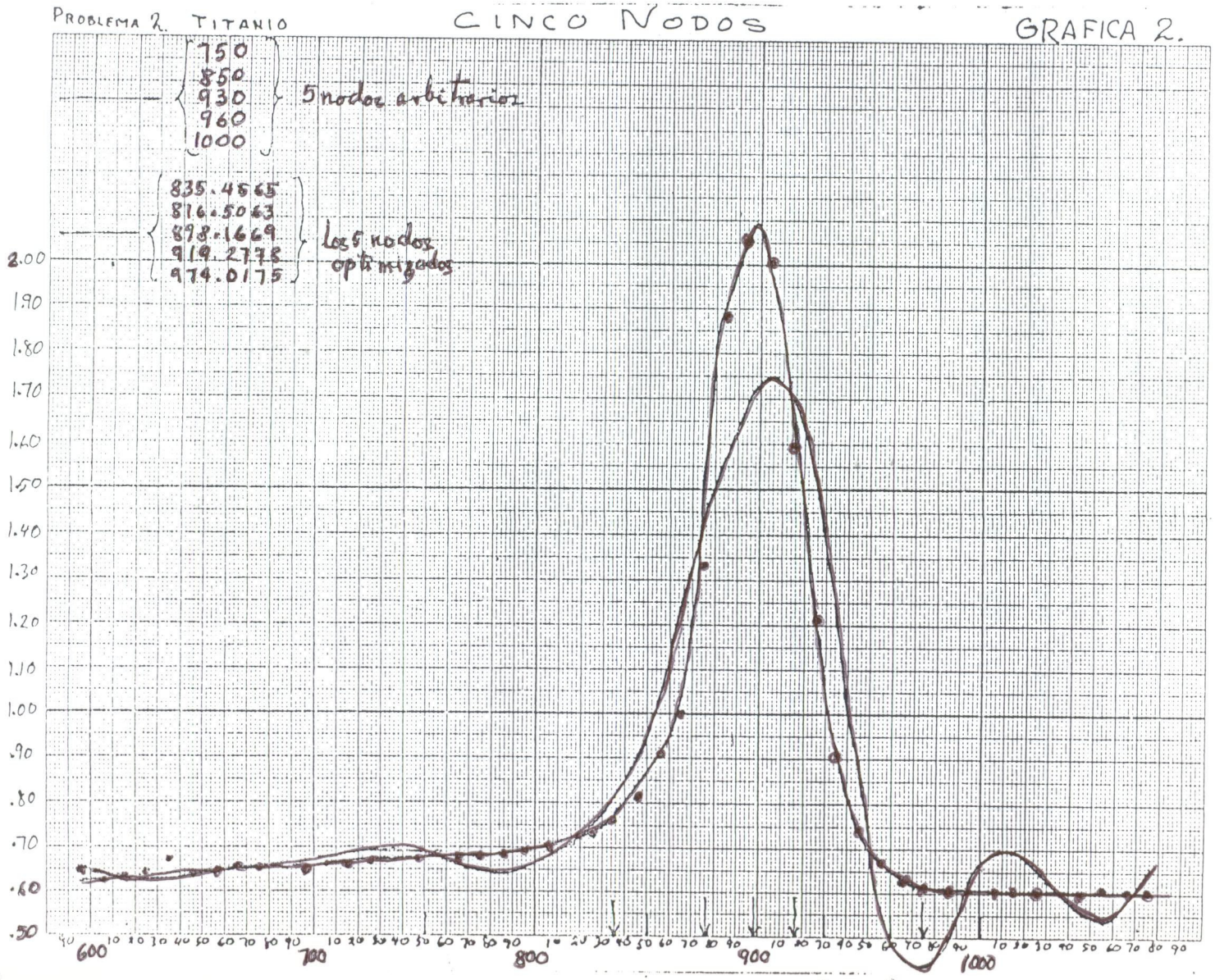
donde $w_1 = w_m = \frac{1}{2}$, y $w_i = 1$ de otro modo, mientras que nosotros utilizamos

$$\left(\sum_{i=1}^{m} |\varepsilon_i|^2\right)^{1/2}$$

Así que para hacer una mejor comparación multiplicamos por el factor $1/(\sqrt{49 - 1})$, lo que nos da como residual

0.0115470

que ya se compara favorablemente con el de Jupp.


PROBLEMA 2. TITANIO
CINCO NODOS
GRAFICA 2.
750
850
930
960
1000
5 nodos arbitrarios
835.4565
816.5063
898.1669
919.2778
974.0175
Los 5 nodos optimizados
2.00
1.90
1.80
1.70
1.60
1.50
1.40
1.30
1.20
1.10
1.00
.90
.80
.70
.60
.50
600
700
800
900
1000

CAPÍTULO IV

ESTIMACIÓN DE PARÁMETROS EN ECUACIONES DIFERENCIALES ORDINARIAS

4.1 El problema matemático.

El problema matemático lo podemos concebir de la siguiente manera:

Supongamos que de algún proceso se han hecho mediciones y se dispone, en consecuencia, de un conjunto de datos. Supongamos también que se sabe que dicho proceso está modelado

por un cierto tipo de ecuación diferencial, la cual se conoce hasta ciertos parámetros. Es decir, estamos suponiendo que los datos de que disponemos $\{(t_i, y_i) : i = 1, \ldots, m\}$ corresponden a una relación funcional $y(t)$ la cual es a su vez solución de una ecuación diferencial que se sabe es de cierto tipo, excepto por ciertos parámetros desconocidos que en ella aparecen:

$$y' = f(t, y, c_1, \ldots, c_n),$$

donde $c_1, \ldots, c_n$ son los parámetros.

Nuestro problema es: determinar valores de los parámetros en la ecuación diferencial de tal manera que, con ciertas condiciones iniciales, la solución de dicha ecuación diferencial mejor ajuste a los datos.

Digamos el enunciado del problema en una forma un poco más general, en el sentido de que la ecuación diferencial puede ser un sistema: supongamos dado un sistema de ecuaciones diferenciales ordinarias

$$y_1' = f_1(t, y, \gamma)$$

$$\vdots$$

$$y_p' = f_p(t, y, \gamma)$$

donde las componentes del vector $\gamma = (c_1, \ldots, c_n)^T$ son parámetros cuyos valores numéricos se desconocen. Suponemos

también que la solución es conocida en puntos $\{t_i : i = 1, \ldots, m\}$; son los datos. El problema es encontrar valores de $c_1, \ldots, c_n$ de tal manera que la solución, con ciertas condiciones iniciales, mejor se ajuste a los datos.

4.2 Antecedentes.

Una solución de este problema, descrita por Bard [1] (1974), van Domselaar y Hemker [5] (1975), y Benson [3] (1979), ha sido usada y sus ideas centrales trataremos de resumir: Para ello usaremos aquí una notación que remarca la dependencia del parámetro γ. Representemos al sistema diferencial en notación vectorial y escribámoslo con sus condiciones iniciales, que suponemos dadas

$$\begin{cases} y'(t,\gamma) = f(t,y,\gamma) \\ y(t_0,\gamma) = y_0(p) \, . \end{cases}$$

Denotemos el vector de observaciones por

$$y = (y_1, \ldots, y_m)^T$$

donde y_i es un valor observado de una componente de y para el valor de t_i. Definamos la función $\tilde{y}$ dada por

$$\tilde{y}(\gamma) = (\tilde{y}(t_1,\gamma), \ldots, \tilde{y}(t_m,\gamma))^T \, ,$$

donde $\tilde{y}(t_i,\gamma)$ es aquella componente de $y(t_i,\gamma)$ que corresponde al valor observado y_i, $1 \leq i \leq k$.

La función residual depende del parámetro γ:

$$\varepsilon(\gamma) = \tilde{y}(\gamma) - y \ .$$

Y entonces el problema consiste en determinar el vector γ para el cual se minimiza la suma de cuadrados

$$\| \varepsilon(\gamma) \|_2^2 = \| \tilde{y}(\gamma) - y \|_2^2 \ .$$

Notemos que para obtener $\tilde{y}(\gamma)$ hay que integrar el sistema de ecuaciones diferenciales. También observemos que, en el proceso de minimización de la suma de cuadrados, hay que calcular la matriz Jacobiana de la función $\varepsilon(\gamma)$:

$$J_{ij}(\gamma) = \frac{\partial \varepsilon_i(\gamma)}{\partial c_j} = \frac{\partial \tilde{y}_i(\gamma)}{\partial c_j} \ .$$

Esta matriz Jacobiana se obtiene de una familia de sistemas de ecuaciones diferenciales que se obtienen para cada componente de γ. Esto da un conjunto grande de ecuaciones diferenciales:

$$y'(t,\gamma) = f(t,y,\gamma)$$

$$y'_{c_1}(t,\gamma) = f_{c_1}(t,y,\gamma) + f_y(t,y,\gamma)\ y_{c_1}(t,y,\gamma)$$

$$\vdots$$

$$y'_{c_n}(t,\gamma) = f_{c_n}(t,y,\gamma) + f_y(t,y,\gamma) y_{c_n}(t,y,\gamma).$$

Sistema este que debe resolverse con algún método numérico.

Objeciones.

a) La solución de la ecuación diferencial puede ser muy sensible a las condiciones iniciales, lo que hace difícil la integración de dicha ecuación.

b) Durante el proceso se hace necesario integrar varias veces todas las ecuaciones diferenciales, posiblemente con un método especial (por ejemplo si el sistema es stiff).

c) No existe simplificación alguna cuando los parámetros aparecen linealmente.

4.3 El Método de Varah.

J. M. Varah (1980) [18] ha propuesto un método directo y simple que supera todas las dificultades anteriores. Nosotros nos hemos propuesto hacer una exposición de tal método y someterlo a prueba con algunos ejemplos.

El método es el siguiente:

(1) Ajustar los datos con un spline cúbico. Esto es, para cada componente $\{y_j : j = 1,\ldots,p\}$, construir un spline cúbico $s_j(t)$, con nodos $\{\xi_k : k = 1,\ldots,q\}$, escogiendo tal spline de manera que se minimice la suma de los cuadrados de las desviaciones en los puntos dato. Sin embargo, los nodos deben escogerse muy cuidadosamente, buscando que el spline de ajuste también describa la tendencia de los datos.

(2) Cuando este spline ha sido encontrado, se determinarán los parámetros de manera que se minimice la suma de cuadrados de las desviaciones en la ecuación diferencial - evaluada en un conjunto de puntos muestrales $\{\hat{t}_i : i = 1,\ldots,M\}$. Esto es, se determina γ tal que

$$\min_{\gamma} \sum_{j=1}^{p} \sum_{i=1}^{M} [s'_j(\hat{t}_i) - f_j(\hat{t}_i, s, \gamma)]^2$$

Nótese que si los parámetros γ aparecen linealmente, lo que resulta es un problema lineal de suma mínima de cuadrados, el cual puede resolverse por una factorización QR o usando las ecuaciones normales. Además, nótese que ninguna integración de las ecuaciones diferenciales se ha hecho y - ningunas condiciones iniciales se han usado.

Comentarios.

Después que los datos han sido ajustados por un spline cúbico, es muy importante revisar su gráfica. No basta obtener un residual pequeño; debe manifestarse la tendencia de los datos; debe tenerse presente que no es sólo un spline de ajuste, es una curva que aproxima la solución de una ecuación diferencial.

De la misma manera la elección de los puntos muestrales es conveniente hacerla también interactivamente. Es algo arbitraria y conviene hacerla de tal manera que el comporta-

miento de la solución quede adecuadamente representado. De la misma manera que para los nodos, es necesario desarrollar un "sentimiento" de cuántos puntos son suficientes y dónde colocarlos.

Una vez que el parámetro γ ha sido determinado, hay que checar su validez integrando el sistema diferencial con tal valor del parámetro.

Una nota final que debemos hacer, hace referencia a una situación que es posible encontrar en la práctica. En ocasiones ocurre que teniendo como modelo un sistema de ecuaciones diferenciales, sólo se conocen datos para algunas, pero no todas, las funciones componentes del sistema. Algunas veces es posible eliminar esta dificultad convirtiendo al sistema dado en uno de orden mayor en el que sólo aparecen las incógnitas de las que se conocen datos. Naturalmente que esto puede hacerse únicamente cuando aquellas otras variables pueden resolverse explícitamente a partir del sistema dado. Esto es lo que se hace en uno de los ejemplos que presentamos en el siguiente capítulo. Y como aumenta el orden de la ecuación diferencial, se requiere la segunda derivada del spline $S(t)$.

4.4 La Subrutina AMGEAR.

Una de las subrutinas que en la práctica se ha probado ser más efectivas para la solución de sistemas de ecuaciones

(1971)

diferenciales, es la creada por Gear [7], tanto para sistemas no-stiff, como stiff. Podemos decir, a grosso modo, que un sistema es stiff si contiene tanto componentes que varían muy rápido como componentes lentas, ambas con un comportamiento de decaimiento. En esta subrutina la que nosotros hemos utilizado para integrar los sistemas que aparecen más adelante.

La subrutina AMGEAR incorpora dos métodos de solución de orden y longitud de paso variable; un método Adams de órdenes 1 a 7 para ecuaciones no stiff y un método de Gear de órdenes 1 a 6 para ecuaciones stiff. Ambos métodos utilizan una fórmula predictor-corrector, con iteración del corrector para la convergencia. Consisten en discretizar el intervalo $[x_0, x_F]$ en una malla de $(n+1)$ puntos determinados automáticamente por la subrutina. Una fórmula de orden uno es usada para empezar, pero es incrementado conforme la integración avanza. La longitud de paso es controlada por la subrutina para controlar (si es posible) el error estimado dentro de una tolerancia especificada.

4.5 Derivadas de un Spline Cúbico.

Puesto que las funciones $y_i(t)$ que aparecen en el sistema diferencial van a ser aproximadas por un spline $s(t)$, es obvio que necesitamos un algoritmo para evaluar la primera derivada del spline $s(t)$ con respecto a la variable in-

dependiente t; y por lo que dijimos arriba, también vamos a necesitar evaluar la segunda derivada $\frac{d^2 s(t)}{dt^2}$.

Una fórmula para la derivada de un B-spline es la siguiente

$$\frac{d}{dt} B_j^k(t) = \left(\frac{k}{\xi_{j+k} - \xi_j}\right) B_j^{k-1}(t) - \left(\frac{k}{\xi_{j+k+1} - \xi_{j+1}}\right) B_{j+1}^{k-1}(t)$$

que puede probarse por inducción y utilizando la fórmula de recursión

$$B_j^k(t) = \left(\frac{t - \xi_j}{\xi_{j+k} - \xi_j}\right) B_j^{k-1}(t) + \left(\frac{\xi_{j+k+1} - t}{\xi_{j+k+1} - \xi_{j+1}}\right) B_{j+1}^{k-1}(t)$$

del capítulo II.

Para $k = 3$:

$$\frac{d}{dt} B_j^3(t) = \left(\frac{3}{\xi_{j+3} - \xi_j}\right) B_j^2(t) + \left(\frac{3}{\xi_{j+4} - \xi_{j+1}}\right) B_{j+1}^2(t)$$

Por otra parte, recordemos que habíamos mostrado, al final del capítulo II, que si $\xi_i \leq t \leq \xi_{i+1}$, el spline cúbico $s(t)$ está dado por

$$s(t) = \sum_{j=i-3}^{i} \alpha_j B_j^3(t).$$

Por lo tanto, tendremos

$$\frac{d}{dt} s(t) = \alpha_{i-3} \frac{d}{dt} B_{i-3}^3(t) + \alpha_{i-2} \frac{d}{dt} B_{i-2}^3(t) + \alpha_{i-1} \frac{d}{dt} B_{j-1}^3(t) + \alpha_i \frac{d}{dt} B_i^3(t).$$

$$= \alpha_{i-3}\left[\frac{3}{\xi_i - \xi_{i-3}} B^2_{i-3}(t) - \frac{3}{\xi_{i+1} - \xi_{i-2}} B^2_{i-2}(t)\right]$$

$$+ \alpha_{i-2}\left[\frac{3}{\xi_{i+1} - \xi_{i-2}} B^2_{i-2}(t) - \frac{3}{\xi_{i+2} - \xi_{i-1}} B^2_{i-1}(t)\right]$$

$$+ \alpha_{i-1}\left[\frac{3}{\xi_{i+2} - \xi_{i-1}} B^2_{i-1}(t) - \frac{3}{\xi_{i+3} - \xi_i} B^2_i(t)\right]$$

$$+ \alpha_i \left[\frac{3}{\xi_{i+3} - \xi_i} B^2_i(t) - \frac{3}{\xi_{i+4} - \xi_{i+1}} B^2_{i+1}(t)\right]$$

$$= 3\,\frac{\alpha_{i-3}}{\xi_i - \xi_{i-3}} B^2_{i-3}(t) + 3\,\frac{\alpha_{i-2} - \alpha_{i-3}}{\xi_{i+1} - \xi_{i-2}} B^2_{i-2}(t)$$

$$+ 3\,\frac{\alpha_{i-1} - \alpha_{i-2}}{\xi_{i+2} - \xi_{i-1}} B^2_{i-1}(t) + 3\,\frac{\alpha_i - \alpha_{i-1}}{\xi_{i+3} - \xi_i} B^2_i(t)$$

$$- 3\,\frac{\alpha_i}{\xi_{i+4} - \xi_{i+1}} B^2_{i+1}(t)$$

$$= 3\,\frac{\alpha_{i-2} - \alpha_{i-3}}{\xi_{i+1} - \xi_{i-2}} B^2_{i-2}(t) + 3\,\frac{\alpha_{i-1} - \alpha_{i-2}}{\xi_{i+2} - \xi_{i-1}} B^2_{i-1}(t)$$

$$+ 3\,\frac{\alpha_i - \alpha_{i-1}}{\xi_{i+3} - \xi_i} B^2_i(t)$$

Por otra parte, puede demostrarse directamente que

$$B_i^2(t)=\begin{cases}\dfrac{(t-\xi_i)^2}{(\xi_{i+1}-\xi_i)(\xi_{i+2}-\xi_i)}, & \xi_i \leq t \leq \xi_{i+1}\\[2ex] \dfrac{(t-\xi_i)(\xi_{i+2}-t)}{(\xi_{i+2}-\xi_i)(\xi_{i+2}-\xi_{i+1})}+\dfrac{(\xi_{i+3}-t)(t-\xi_{i+1})}{(\xi_{i+3}-\xi_{i+1})(\xi_{j+2}-\xi_{i+1})}, & \xi_{i+1} \leq t \leq \xi_{i+2}\\[2ex] \dfrac{(\xi_{i+3}-t)^2}{(\xi_{i+3}-\xi_{i+1})(\xi_{i+3}-\xi_{i+2})}, & \xi_{i+2} \leq t \leq \xi_{i+3}\end{cases}$$

Los algoritmos para la primera y segunda derivada.

Algoritmo. (DERIVA). Conociendo los coeficientes $\{\alpha_i : i = 1,\ldots,n\}$, los nodos $\{\xi_i : i = 1,\ldots,n+4\}$, el real t, y el entero k tal que $\xi_k \leq t \leq \xi_{k+1}$, con este algoritmo se encuentra la derivada de un spline cúbico evaluada en t.

1) $$B^2_{k-2} \leftarrow \frac{(\xi_{k+1}-t)^2}{(\xi_{k+1}-\xi_{k-1})(\xi_{k+1}-\xi_k)}$$

2) $$B^2_{k-1} \leftarrow \frac{(t-\xi_{k-1})(\xi_{k+1}-t)}{(\xi_{k+1}-\xi_{k-1})(\xi_{k+1}-\xi_k)}+\frac{(\xi_{k+2}-t)(t-\xi_k)}{(\xi_{k+2}-\xi_k)(\xi_{k+1}-\xi_k)}$$

3) $$B^2_k \leftarrow \frac{(t-\xi_k)}{(\xi_{k+1}-\xi_k)(\xi_{k+2}-\xi_k)}$$

4) $$\frac{ds(t)}{dt} \leftarrow 3\,\frac{\alpha_{k-2}-\alpha_{k-3}}{\xi_{k+1}-\xi_{k-2}}\,B^2_{k-2} + 3\,\frac{\alpha_{k-1}-\alpha_{k-2}}{\xi_{k+2}-\xi_{k-1}}\,B^2_{k-1} + 3\,\frac{\alpha_k-\alpha_{k-1}}{\xi_{k+3}-\xi_k}\,B^2_k$$

FIN

Para la segunda derivada de un spline de grado k tenemos la siguiente fórmula:

Si $\xi_i \leq t \leq \xi_{i+1}$, entonces

$$\frac{d^2}{dt^2}\sum_{j=i-k}^{i}\alpha_j B^k_j =$$

$$k(k-1)\sum_{j=i-k}^{i}\left[\frac{\alpha_j-\alpha_{j-1}}{(\xi_{j+k}-\xi_j)(\xi_{j+k-1}-\xi_j)} - \frac{\alpha_{j-1}-\alpha_{j-2}}{(\xi_{j+k-1}-\xi_{j-1})(\xi_{j+k-1}-\xi_j)}\right]B^{k-2}_j .$$

Para $k=3$, únicamente queda

$$\frac{d^2}{dt^2}\sum_{j=i-3}^{i}\alpha_j B^3_j =$$

$$6\left[\frac{\alpha_{i-1}-\alpha_{i-2}}{(\xi_{i+2}-\xi_{i-1})(\xi_{i+1}-\xi_{i-1})} - \frac{\alpha_{i-2}-\alpha_{i-3}}{(\xi_{i+1}-\xi_{i-2})(\xi_{i+1}-\xi_{i-1})}\right]B^1_{i-1}$$

$$+ 6\left[\frac{\alpha_i-\alpha_{i-1}}{(\xi_{i+3}-\xi_i)(\xi_{i+2}-\xi_i)} - \frac{\alpha_{i-1}-\alpha_{i-2}}{(\xi_{i+2}-\xi_{i-1})(\xi_{i+2}-\xi_i)}\right]B^1_i$$

$$= 6\left[\frac{\alpha_{i-1}-\alpha_{i-2}}{(\xi_{i+2}-\xi_{i-1})(\xi_{i+1}-\xi_{i-1})} - \frac{\alpha_{i-2}-\alpha_{i-3}}{(\xi_{i+1}-\xi_{i-2})(\xi_{i+1}-\xi_{i-1})}\right]\frac{\xi_{i+1}-t}{\xi_{i+1}-\xi_i}$$

$$+ 6\left[\frac{\alpha_{i}-\alpha_{i-1}}{(\xi_{i+3}-\xi_{i})(\xi_{i+2}-\xi_{i})} - \frac{\alpha_{i-1}-\alpha_{i-2}}{(\xi_{i+2}-\xi_{i-1})(\xi_{i+2}-\xi_{i})}\right]\frac{t-\xi_i}{\xi_{i+1}-\xi_i}\ .$$

Esta fórmula la implementamos en la subrutina DERSEG para evaluar la segunda derivada de un spline $s(t)$ con respecto a la variable independiente t.

Las Subrutinas KLOCAT y PUNTOS.

Hemos visto que en la aplicación de nuestro método para estimación de parámetros, frecuentemente se requiere, dado un valor real t, determinar el entero k tal que

$$\xi_k \leq t < \xi_{k+1}.$$

Este trabajo lo realiza el siguiente algoritmo.

Algoritmo. (KLOCAT). Dados los nodos $\xi_4, \xi_5, \ldots, \xi_{n+1}$, y el punto t, este algoritmo encuentra k tal que $\xi_k \leq t < \xi_{k+1}$, con $k \geq 4$ y $k < (n+1)$.

1) Si ($t < \xi_4$ ó $t > \xi_{n+1}$) escribir "t está fuera de intervalo" y salir.

2) Para $j = 4, .., n$

   1) Si ($t < \xi_{j+1}$) pasa a 3)

3) KLOCAT = j

4) Si ($t = \xi_{n+1}$) KLOCAT = n

FIN

La construcción que realiza la subrutina PUNTOS es muy útil en el paso (2) de nuestro método de estimación de parámetros, pues construye un conjunto de puntos nuestrales $\{\hat{t}_i : i = 1,\ldots,M\}$, con una libertad suficientemente amplia.

Algoritmo. (PUNTOS). La salida es un arreglo unidimensional $TM(\ )$ que contiene al conjunto de puntos muestrales y un entero NM que es el total de puntos muestrales construídos.

1) Se leen los extremos del intervalo en el que var. a estar contenidos los puntos muestrales.

2) Se lee el número de subintervalos en que se quiere dividir el intervalo total.

3) Se leen los extremos de tales subintervalos.

4) Se lee para cada uno de los subintervalos el número de puntos equidistantes que se desea tenga dicho subintervalo.

5) Se construyen los puntos muestrales dentro de cada subintervalo y en el arreglo $TM(\ )$ quedan almacenados todos los puntos muestrales, que incluyen a los extremos de todos los subintervalos.

4.6 Ejemplos.

Aquí sólo vamos a incluir los resultados finales y mejores, obtenidos por nosotros en algunos problemas típicos y significativos que se encuentran en la literatura. Una descripción más detallada la dejamos para el siguiente capítulo.

EJEMPLO A. El problema de Bellman [2].

$$y' = c_1(126.2 - y)(91.9-y)^2 - c_2 y^2$$

t	1	2	3	4	5	6	7	8
y	0.0	1.4	6.3	10.4	14.2	17.6	21.4	23.0
t	10	12	15	20	25	30	40	
y	27.0	30.4	34.4	38.8	41.6	43.5	45.3	

Estos datos se obtuvieron de una cierta reacción química. La ecuación diferencial incluye dos parámetros que aparecen linealmente.

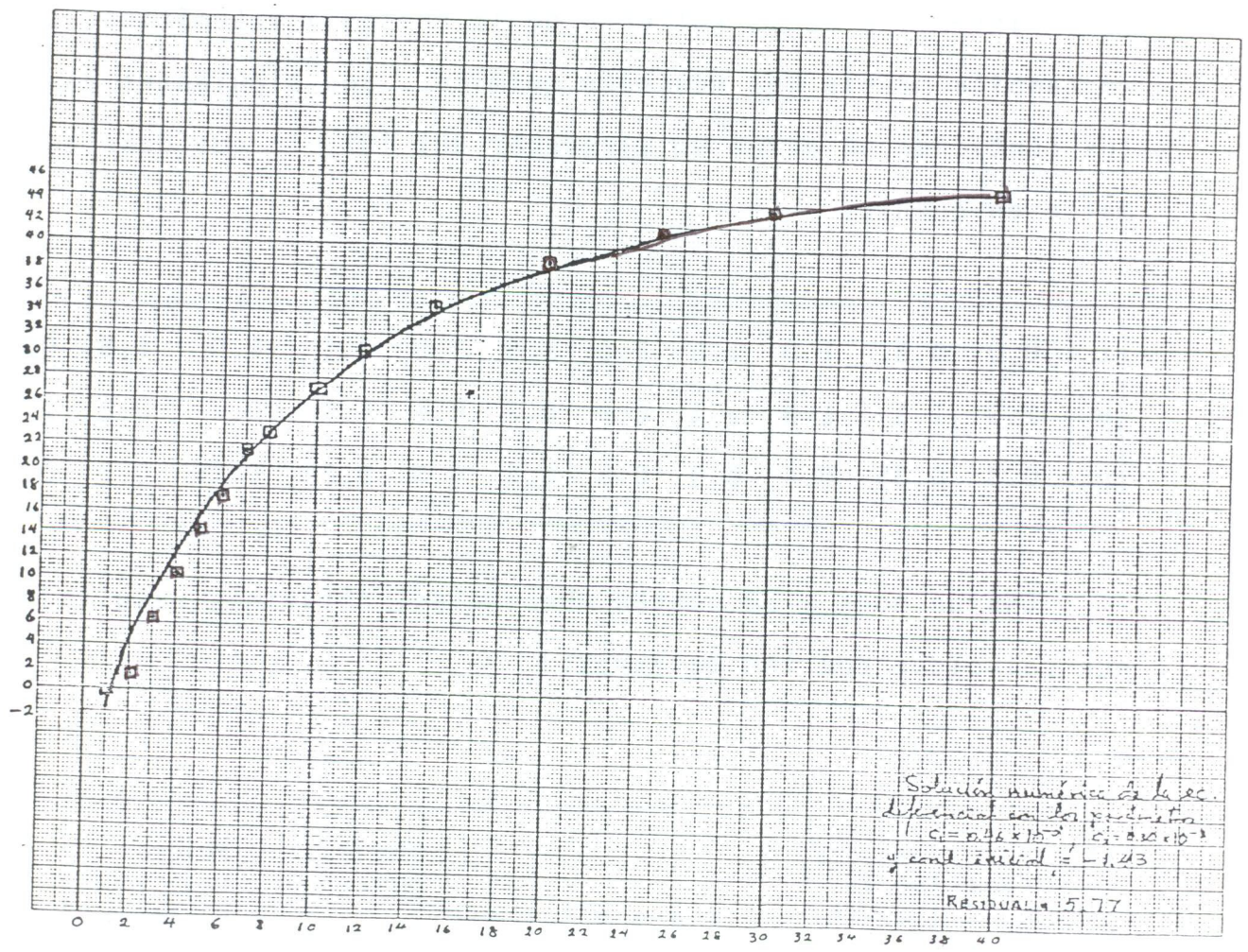
Con un nodo y 40 puntos muestrales, obtuvimos

Nodo	Residual en el spline	Residual en la Ec. Dif.	Parámetros	Residual Integrado	Mejor cond. inic.
20.22	2.660	0.618	0.46×10^{-5}; 0.30×10^{-3}	5.77	-1.43

Varah obtiene (también con un nodo y 40 puntos muestrales)

20.2	2.7	0.98	$0.47 \times 10^{-5}; 0.31 \times 10^{-3}$	3.7	-1.49

La curva solución de la ecuación diferencial y los datos del problema pueden verse en la siguiente gráfica.



Solución numérica de la ec. dif. con los parámetros 0.46×10^{-5}; 0.30×10^{-3}; y condición inicial = -1.43

Residual = 5.77.

EJEMPLO B. El problema de Barnes, Ref. [5].

El sistema con dos ecuaciones

$$y_1' = c_1 y_1 - c_2 y_1 y_2$$

$$y_2' = c_2 y_1 y_2 - c_3 y_2$$

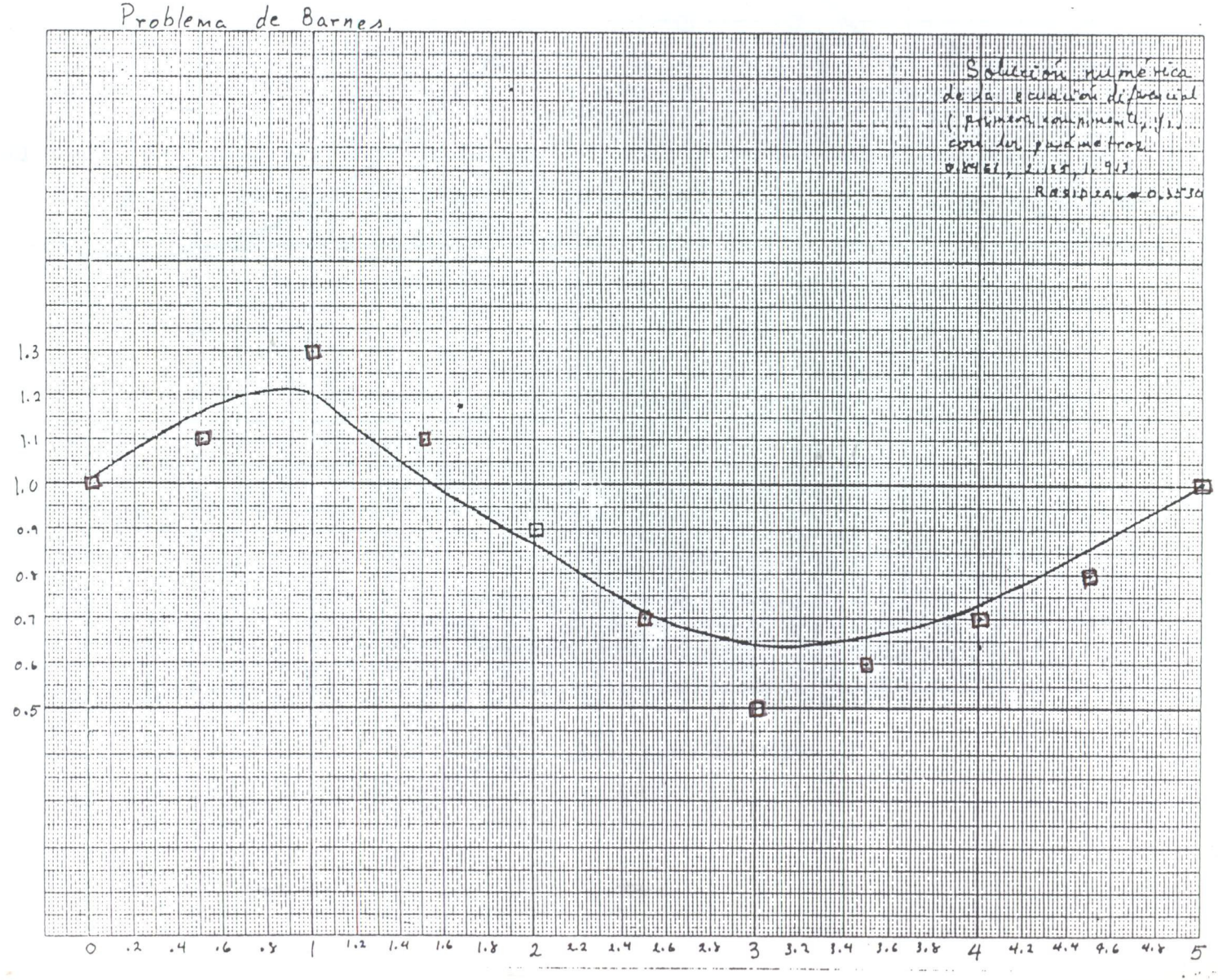
en el que aparecen 3 parámetros linealmente, es el bien conocido sistema de Lotka-Volterra de la ecología matemática, pero también modela una reacción química. Los siguientes datos contienen un error de alrededor del 10%, lo que debe tomarse en cuenta.

t	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
y_1	1.0	1.1	1.3	1.1	0.9	0.7	0.5	0.6	0.7	0.8	1.0
y_2	0.3	0.35	0.4	0.5	0.5	0.4	0.3	0.25	0.25	0.3	0.35

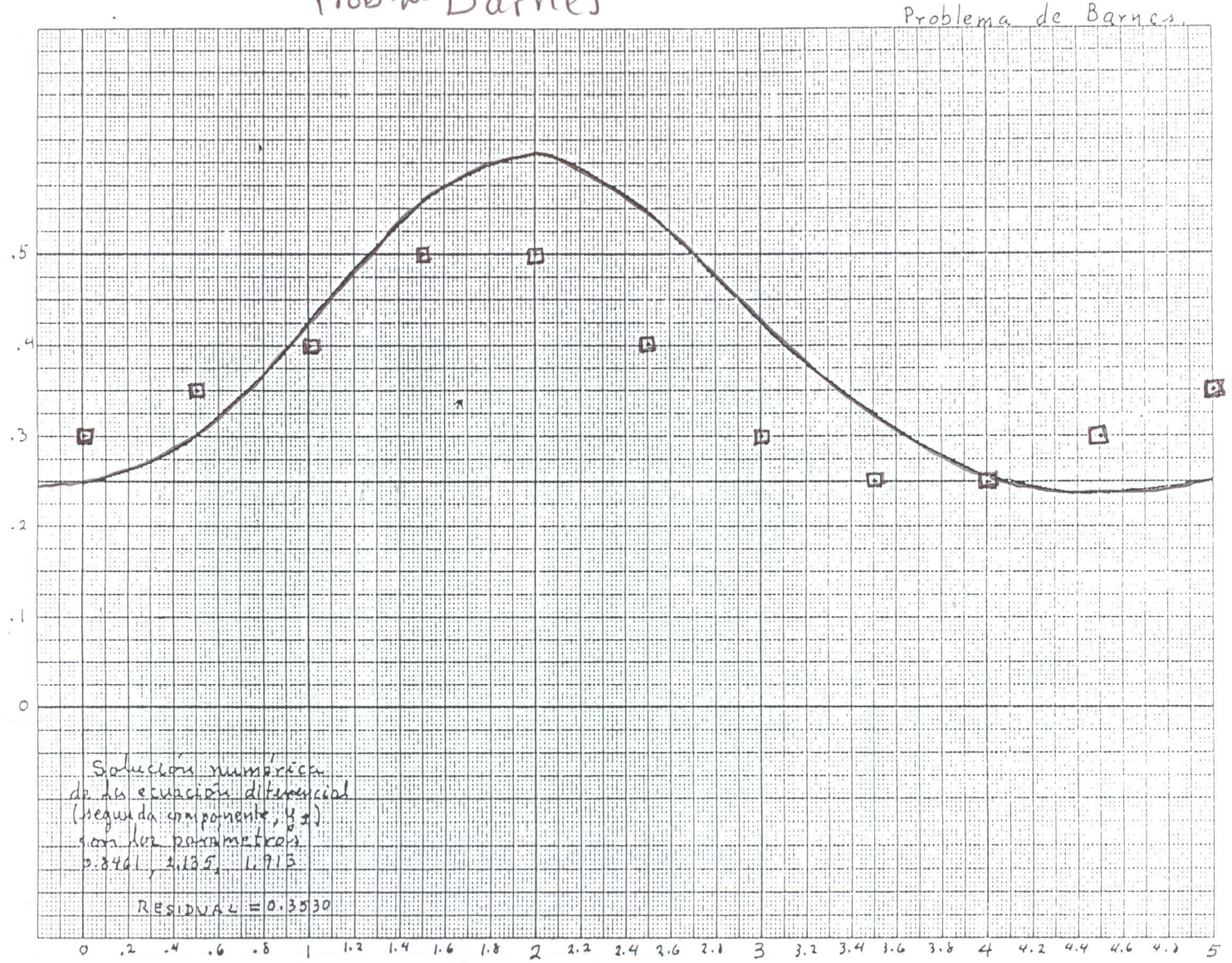
Con un nodo y 20 puntos muestrales, obtuvimos resultados que coinciden con los de Varah:

Nodo	Residual en la ec.dif.	P a r á m e t r o s	Residual integrado	Mejores condiciones iniciales
3.0	1.260	0.8461;2.135;1.913	0.3530	1.02;0.25

Solución numérica de la ecuación diferencial (primera componente (t, y_1) con los parámetros 0.8461 ; 2.135 ; 1.913 con residual 0.3530.



Prob. de Barnes



2ª componente y_2.

EJEMPLO C. El problema del enzima. Ref. [18].

$$y_1' = c_1(27.8 - y_1) + \frac{c_4}{2.6}(y_2 - y_1) + \frac{4991}{t\sqrt{2\pi}} \exp[-0.5[\frac{\ln t - c_2}{c_3}]^2]$$

$$y_2' = \frac{c_4}{2.7}(y_1 - y_2)$$

t	0.1	2.5	3.8	7.0	10.9	15.0	18.2	21.3
y_1	27.8	20.0	23.5	63.6	267.5	427.8	339.7	331.9
t	22.9	24.9	26.8	30.1	34.1	37.8	42.4	44.4
y_1	243.5	212.0	164.1	112.7	88.1	76.2	62.3	58.7
t	47.9	53.1	59.0	65.1	73.1	81.1	91.2	101.9
y_1	41.9	40.2	31.3	30.0	30.6	23.5	24.8	26.1
t	115.4	138.7	163.2	186.7				
y_1	33.3	17.8	16.8	16.8				

La ecuación diferencial es un modelo que representa la concentración del enzima en la sangre, dentro y fuera del corazón, en un período de tiempo. Existe aquí la dificultad de que sólo se tienen los datos de las observaciones correspondientes a sólo una de las variables, y_1. Esta dificultad se superó, como se indicó antes, resolviendo para y_2 en la primera ecuación, diferenciando y substituyendo

en la segunda ecuación, para obtener una ecuación de segundo orden en y_1 únicamente, [ver un capítulo más adelante].

Este problema es más difícil que los dos anteriores a causa de que los parámetros aparecen no-linealmente y porque los datos son difíciles de ser ajustados a causa del brusco ascenso y descenso que presentan.

Con 4 nodos y 40 puntos muestrales obtuvimos los siguientes resultados

Nodos	Residual en la Ec. Dif.	P a r á m e t r o s	Residual integrado
8,11,23,43	6.66	0.239, 2.634, 0.368, 0.297	109

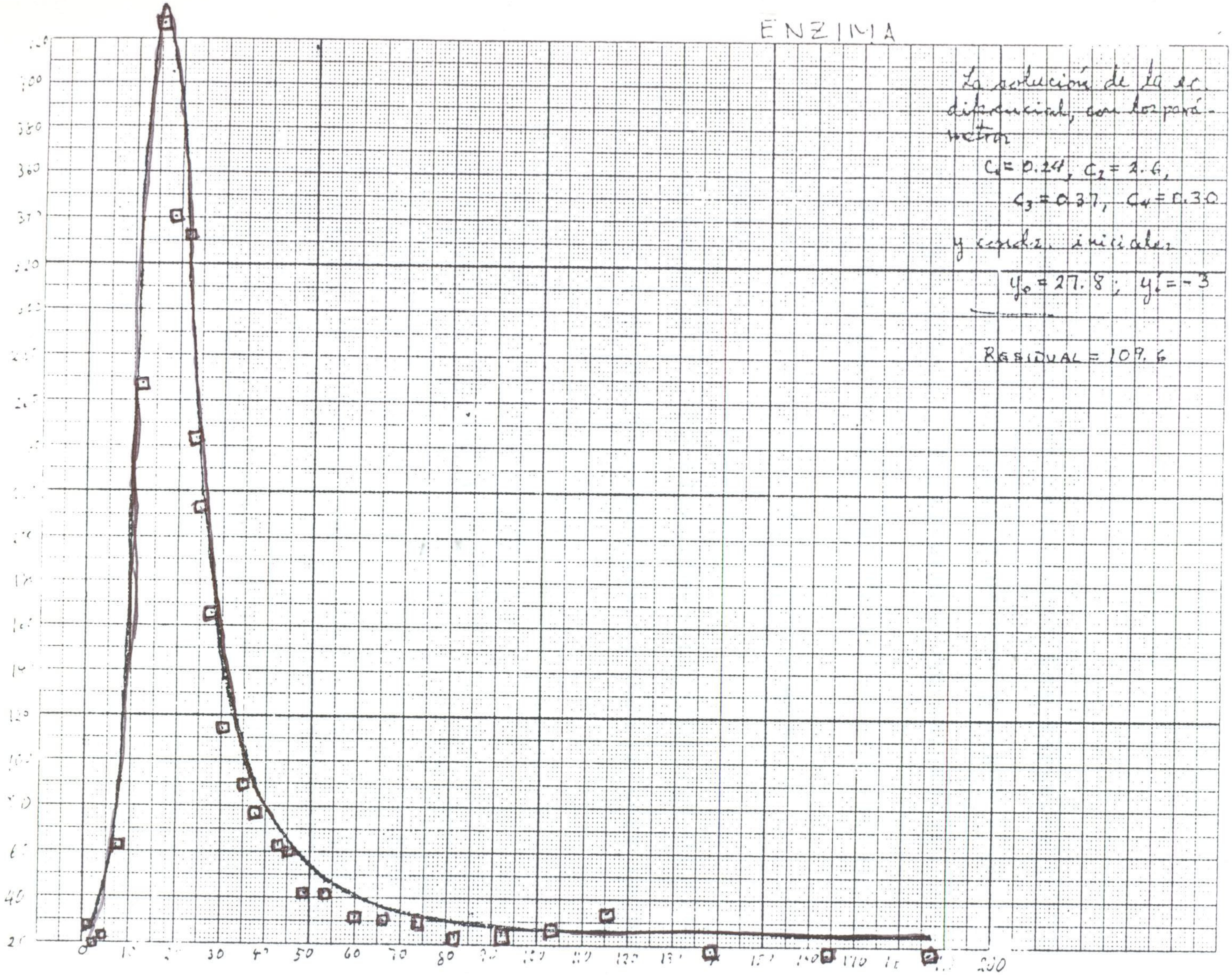
que pueden compararse satisfactoriamente con los de Varah:

8,11,23,43	15.5	0.257, 2.620,0,364, 0.290	70

La gráfica de la siguiente página muestra la solución de la ecuación diferencial con nuestros parámetros.

EJEMPLO D. La logística.

$$y' = c_1 y(c_2 - y) , \qquad c_1 , c_2 > 0.$$


ENZIMA
La solución de la ec. diferencial, con los parámetros
$C_1 = 0.24$, $C_2 = 2.6$,
$C_3 = 0.37$, $C_4 = 0.30$
y cond. iniciales
$y_0 = 27.8$; $y_0' = -3$
RESIDUAL = 107.6

ENZIMA

t	0	4	7.5	25	31	48.75	52	58.5
y	8	6	6	7	8	10	13	18
t	72.7	78	95	96	108	112	133	136.75
y	33	38	76	78	164	175	280	300
t	143	156.5	166.7	181				
y	320	405	385	450				

Los datos corresponden el crecimiento de una población de bacterias. Si escribimos la ecuación diferencial en la forma

$$y' = k_1 y - k_2 y^2$$

vemos que los parámetros aparecen linealmente.

Aplicando nuestro método de estimación de parámetros en ecuaciones diferenciales con 3 nodos y 40 puntos muestrales, obtuvimos los resultados que aparecen en la siguiente tabla.

Nodos	Parámetros	Residual en la Ec. Difirencial
25,100,140	0.04608, 0.00009570	3.729

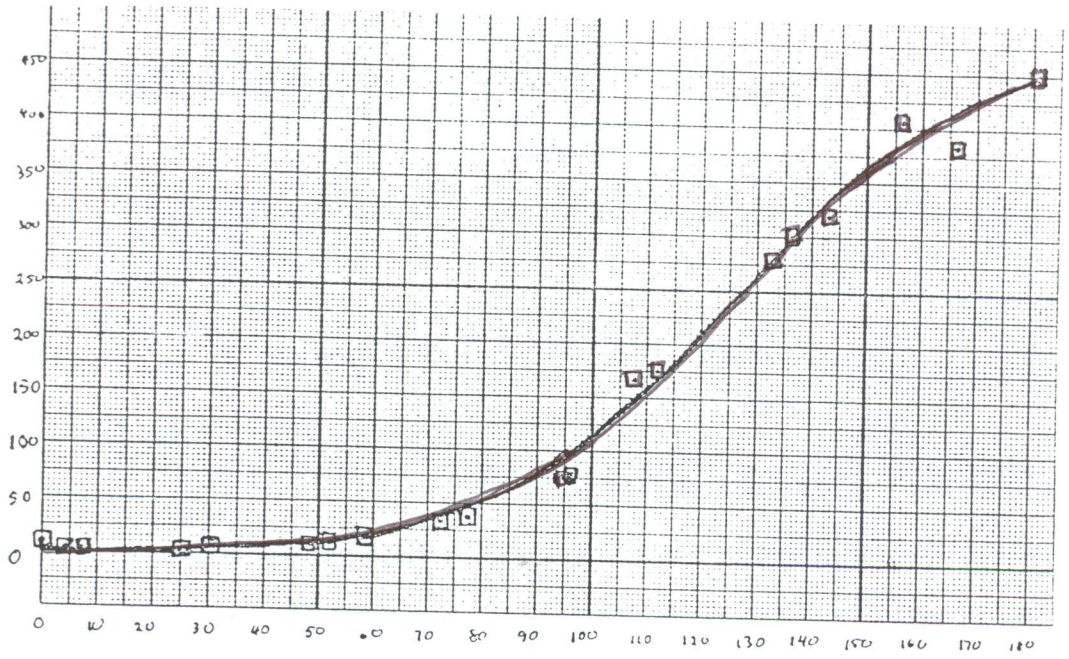
Así, pues, hemos determinado los parámetros k_1 y k_2. La ecuación queda

$$y' = 0.04608\, y - 0.0009570\, y^2$$

o bien

$$y' = 0.0009570\, y\, (481.5 - y).$$

Integrando esta ecuación, con la condición inicial $y_0 = 1.5$, nos da la surva de la siguiente figura



CAPITULO V

CONCLUSIONES Y EJEMPLOS

5.1 Observaciones y Conclusiones.

Con los ejemplos siguientes tratamos de mostrar de manera concreta la efectividad de nuestros métodos, en primer lugar del uso de splines cúbicos con nodos libres en la aproximación y después, de la estimación de parámetros en ecuaciones diferenciales con el método indicado en el capítulo IV. Pero también consideramos necesario hacer algunas observaciones sobre las dificultades que se presentan.

Un invariante que aparece en todos los ejemplos, es la notable mejoría que se obtiene en el ajuste de datos cuando se considera a los nodos como variables. Además, puesto que

encontrar los nodos óptimos significa resolver el problema

$$\min F(x, \alpha) = \min \|f - A(x)\alpha\|^2,$$

resulta que, si el método Levenberg-Marquardt converge a un mínimo, esto es bastante satisfactorio, puesto que sabemos que el método Levenberg-Marquardt es un método robusto, confiable, seguro y económico.

Sin embargo, bajo ciertas condiciones se produce el desafortunado fenómeno, intrínseco al problema de splines con nodos libres e independiente de los datos, de que la función $F(x,\alpha)$ varía muy lentamente con x. Esto conduce a una doble dificultad. Por un lado una convergencia sumamente lenta del Levenberg-Marquardt. Y por el otro lado, la existencia de numerosos puntos estacionarios, no necesariamente mínimos, que de hecho son soluciones indeseables del problema y a las que puede conducir una aplicación del método Levenber - Marquardt.

Una solución a esto la proporciona la transformación σ de la que hablamos en el capítulo III.

Aún más. El método Levenberg-Marquardt está diseñado para aplicarse a problemas en que las variables independientes no tienen restricciones, sin embargo en nuestro caso, el conjunto de nodos debe satisfacer, como afirmamos al inicio del capítulo III, las condiciones

$$a < \xi_5 < \xi_6 < \ldots < \xi_n < b.$$

Por ello, en ocasiones, cuando aplicamos el método Levenberg-Marquardt en estas condiciones al problema de splines con nodos libres, tal método proporciona soluciones inaceptables: soluciones en que los nodos han perdido la relación de orden que deben cumplir. Otra vez la transformación σ, al liberar a las variables de las restricciones, proporciona soluciones factibles y se obtiene ciertamente un conjunto de nodos correspondiente a un mínimo (local) de la función.

Esto es, el método Levenberg-Marquardt dotado de la transformación σ, converge a una solución, eliminando el problema del aletargamiento y la existencia de soluciones indeseables. Naturalmente, hay que pagar un precio por ello, y esto equivale a un mayor tiempo de cómputo: entre un 30% y 50% mayor, porque hay que hacer más operaciones, entre ellas un mayor número de evaluaciones de la función y de la matriz Jacobiana.

Emerge aquí un problema importante. ¿Cómo detener la ejecución del problema? Se está realizando un proceso iterativo en el que se construye una sucesión $\{x_k\}$ tal, que esperamos que x_k converja a un mínimo x^* de una función $\|F(x)\|$. Así, en forma natural existen dos criterios fundamentales de detención del programa. Si llamamos FTOL y XTOL a dos constantes positivas, podemos detener el programa si el error relativo en la suma de los cuadrados es a lo más FTOL o si el error relativo entre dos iteraciones consecuti-

vas de x es a lo más XTOL. Resulta necesario hacer una adecuada elección de estas tolerancias en el error. Una exigencia demasiado grande puede llevar a un trabajo demasiado grande e inútil. Por el contrario, pedir poco, puede ser poco provechoso y conducirnos a aproximaciones demasiado malas. Nosotros, después de numerosos ensayos logramos determinar que para nuestros propósitos de determinar el spline óptimo, una tolerancia en ambos casos de 10^{-5} es suficiente y útil.

Por otra parte, determinar el número de nodos a usar es un problema delicado. Un número escaso de nodos puede ser insuficiente, un número demasiado abundante de nodos, puede ser excedido. La práctica y el análisis de cada problema particular pueden ayudar a determinar el número de nodos más adecuado. En todo caso es aconsejable proceder interactivamente, haciendo gráficas, variando el número de nodos y haciendo comparaciones con los residuales respectivos.

Especial importancia tiene esto cuando se está construyendo el spline que ajusta los datos en un problema de estimación de parámetros en ecuaciones diferenciales. Es frecuente en este caso que los datos contengan errores experimentales, y por ello debemos buscar que el spline refleje la tendencia de los datos, ya que este spline ha de imitar la solución de una ecuación diferencial, precisamente aquella solución que mejor se ajuste a los datos.

Así pues, una importante enseñanza adquirida, es la certeza de que para obtener una aproximación spline razonable, es necesario proceder interactivamente haciendo gráficas, variando el número de nodos y buscando representar la tendencia de los datos. Pocos nodos nos pueden dar una mejor representación aunque el residual no sea tan pequeño, pues una curva spline que se apegue demasiado a los datos puede ser inconveniente desde el punto de vista de la tendencia que los datos manifiestan; esto puede verse claramente en los ejemplos 4 y 5 que aparecen más adelante.

Usar precisamente splines cúbicos para ajustar los datos en el problema de estimación de parámetros no deja de ser importante, porque la derivada $s'(t)$ tiene que ser usada para aproximar la derivada de la función $y(t)$, y son conocidas las dificultades de este problema con otros medios, en tanto que el uso de splines cúbicos ha mostrado sus ventajas.

Pero la principal ventaja, a nuestro modo de ver, de este método de estimación de parámetros es que no se requiere integrar las ecuaciones hasta determinar los parámetros. Esto significa menor trabajo computacional y menor complejidad en los programas. Tampoco se requiere usar condiciones iniciales, y cuando hay libertad de hacerlo estas pueden elegirse al final de manera más o menos libre, con los parámetros ya determinados.

Análogamente, si los parámetros aparecen linealmente en la ecuación diferencial, es posible utilizar esto para resolver sólo un problema de suma mínima de cuadrados lineal, lo cual significa, otra vez, menor trabajo computacional, así como mayor precisión en los cálculos.

Un problema que queda es el de la elección de los puntos muestrales $\{\hat{t}_i\}$ que mencionamos en el capítulo IV. No hay un método y nuestra experiencia es que no hay necesidad de usar una gran abundancia de ellos y frecuentemente el colocarlos igualmente distribuídos en el intervalo da buenos resultados. Pero más experimentos y ensayos con nuevos problemas creemos que son necesarios.

5.2 Ejemplos de Ajuste con Splines con Nodos Libres.

EJEMPLO 1. *La función* $f(t) = t^2 \operatorname{sen} t$. Es un ejemplo sencillo el que hemos construído con esta función. Los puntos los elegimos tomando 50 abscisas equitativamente distribuidas en el intervalo $[-\pi, 2\pi]$.

En el caso de dos nodos ya se palpa de manera fuerte el mucho mejor ajuste que se logra optimizando los nodos; el residual ha sido reducido de 67.53 a 4.45. Con cinco nodos casi no hay error. Sin optimizar los nodos el ajuste es ya bueno, porque los coeficientes del spline se toman, como en todos los casos, buscando minimizar las diferencias

en los datos y se manifiesta también la gran virtud de aproximación a las curvas que poseen los splines cúbicos.

Con el método Levenberg-Marquardt en ocasiones se obtienen nodos desordenados. Después de introducir la transformación sigma no se observa esta dificultad.

Las figuras 1 y 2 muestran los splines correspondientes a algunos de los resultados obtenidos.

En la segunda columna expresamos el método utilizado. x_0 es el conjunto de nodos iniciales.

x_0	Método	Nodos Optimizados.	Residual	Grad	NFEV	NJEV	
-0.666 2.333	LM	2.066 3.000	4.45	10^{-8}	26	21	
-2.5 -0.5 1.0	LM	-1.027 1.020 3.159	1.26	10^{-10}	21	14	
-2 0 3 5	LM	-0.987 0.907 3.277 5.508	0.43	10^{-10}	17	10	
-2.5 -0.4 0.0 2.0 5.3	LM	-2.892 0.320 -2.383 3.398 4.873			58	36	[no hubo orden]

x_0	Método	Nodos Optimizados	Resi-- dual	Grad	NFEV	NJEV
-1.5 -0.4 1.5 3.0 4.0	LM	-1.168236 -0.485310 0.862764 3.281149 5.505520	0.42	10^{-9}	19	14
,,	LM/ SIGMA	-1.168235 -0.485352 0.862755 3.281160 5.505396	0.42	10^{-5}	26	19

Tabla 1.

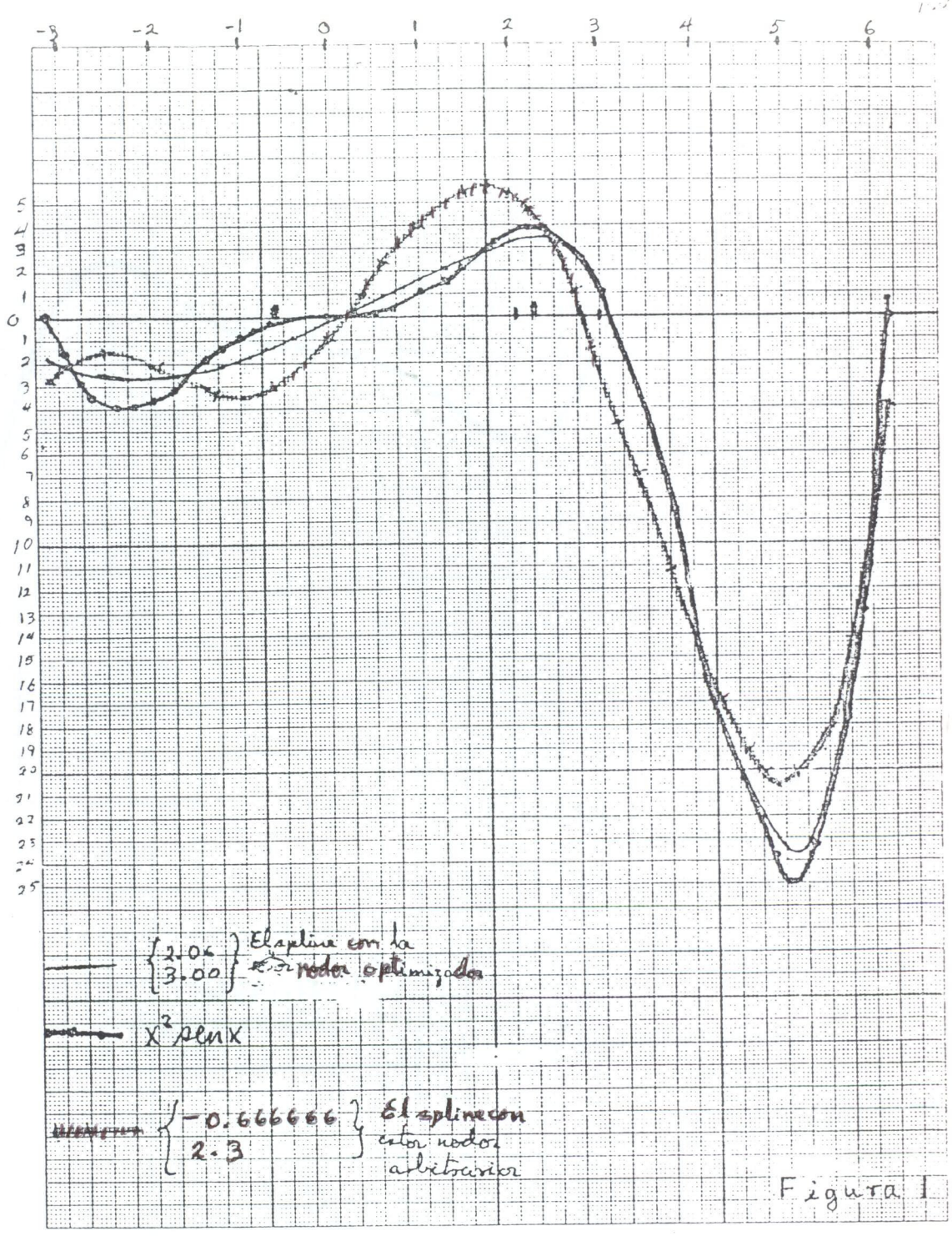

-3 -2 -1 0 1 2 3 4 5 6
{2.0x
{3.00
El spline con los nodos optimizados
X² spline
{-0.666666
{2.3
El spline con estos nodos arbitrarios

Figura 1

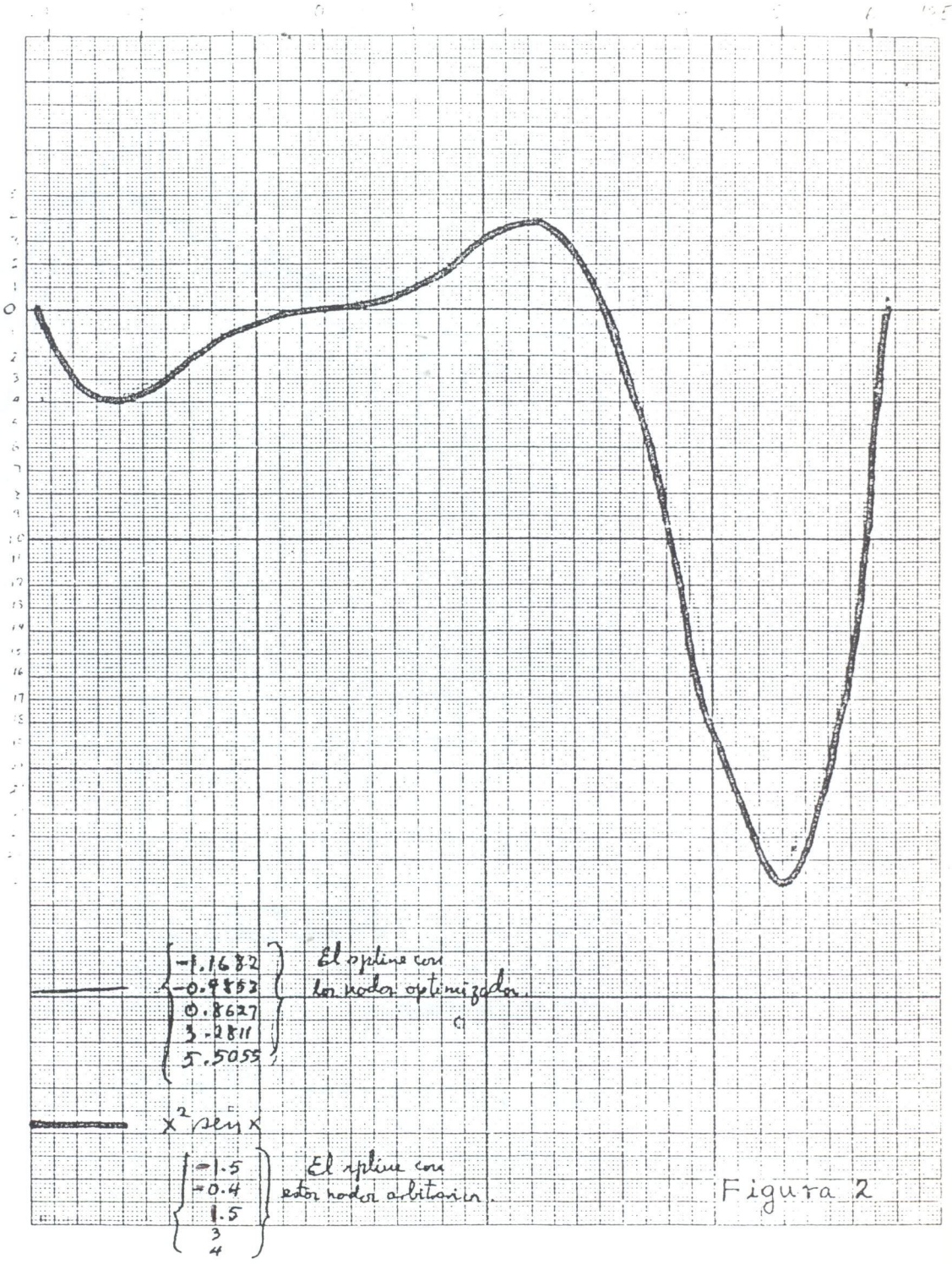

-1.1682
-0.9853
0.8627
3
5.5055
El spline con los nodos optimizados.
$x^2 \operatorname{sen} x$
-1.5
-0.4
1.5
3
4
El spline con estos nodos arbitrarios.

Figura 2

EJEMPLO 2. *El Problema del Titanio.*

Este es un interesante y difícil problema que ha sido usado por de Boor [4] y por Jupp [12] por las dificultades que tiene la aproximación a los datos, digamos, por polinomios. Sin embargo, la técnica de ajuste con nodos libres - muestra su excelencia.

Jupp encuentra para el caso de 5 nodos que existe una disposición óptima, la cual es

(835.967, 876.402, 898.146, 916.315, 973.908).

Nosotros hemos obtenido los resultados que aparecen en la siguiente tabla para 4,5 y 6 nodos, tomando inicialmente un conjunto más o menos arbitrario. Los splines correspondientes se muestran en las figuras 3,4 y 5.

x_0	Método	Nodos Optimizados	Residual	Grad	NFEV	NJEV
800 900 930 980	LM	No hubo orden				
"	LM/ SIGMA	898.75 904.94 910.75 974.76	0.64	10^{-7}	39	22
750 850 930 960 1000	LM	No hubo orden				
,,	LM/ SIGMA	835.45 876.53 898.13 916.31 974.00	0.09	10^{-6}	11	11
840 900 905 910 920 1000	LM	835.461 876.302 899.097 914.019 935.673 970.718	0.08	10^{-7}	42	24

Tabla 2

Problema del Titanio (4 nodos)

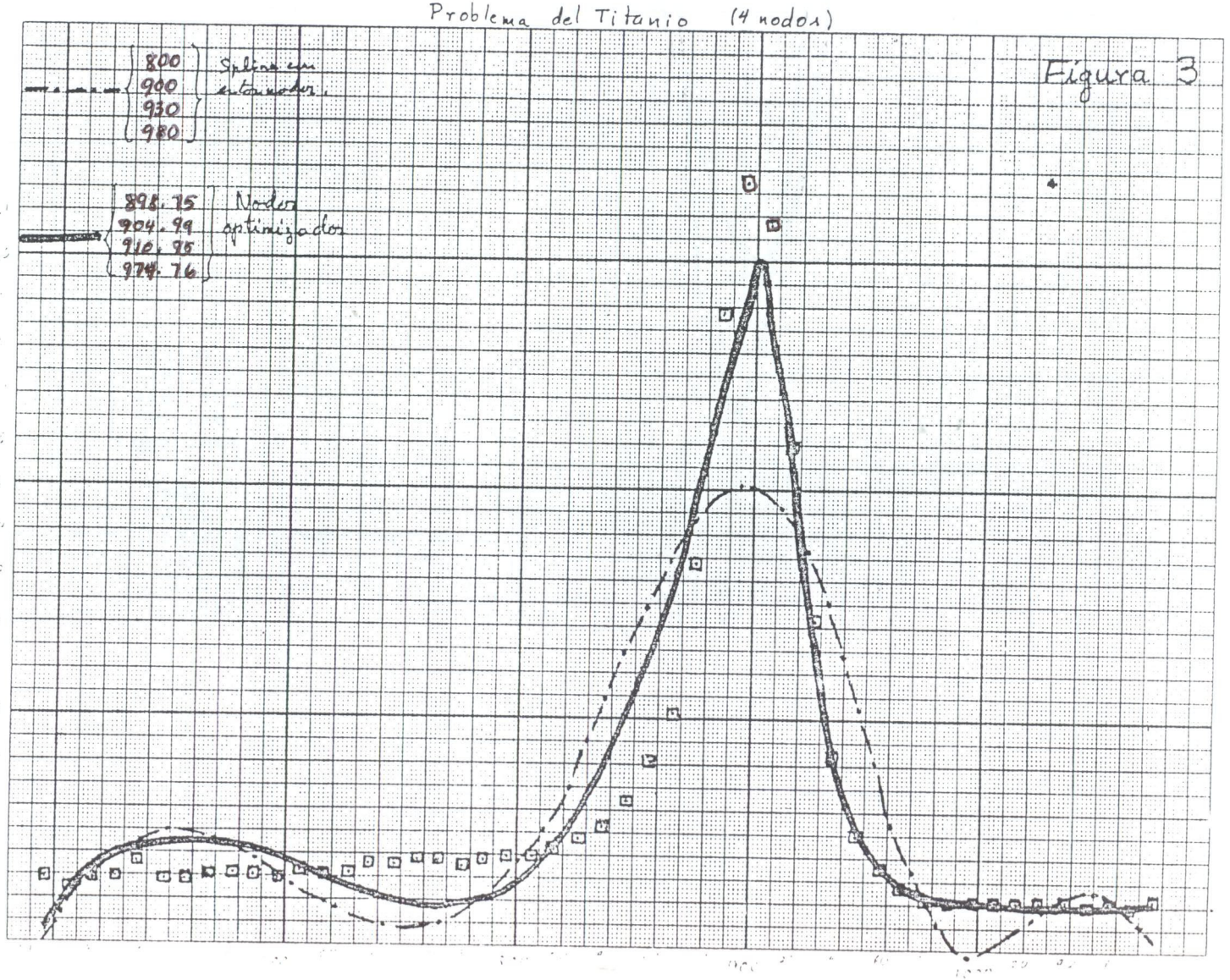
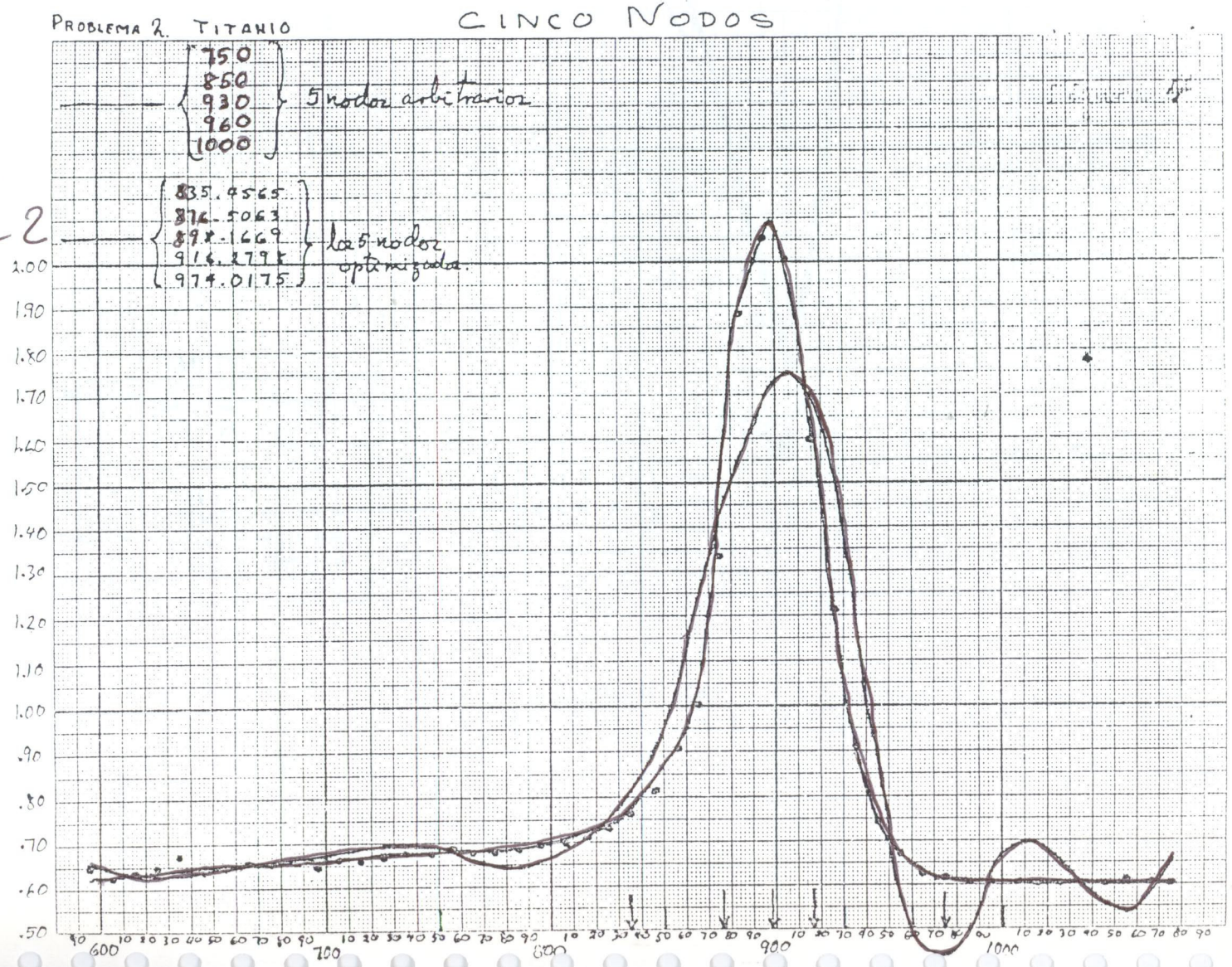
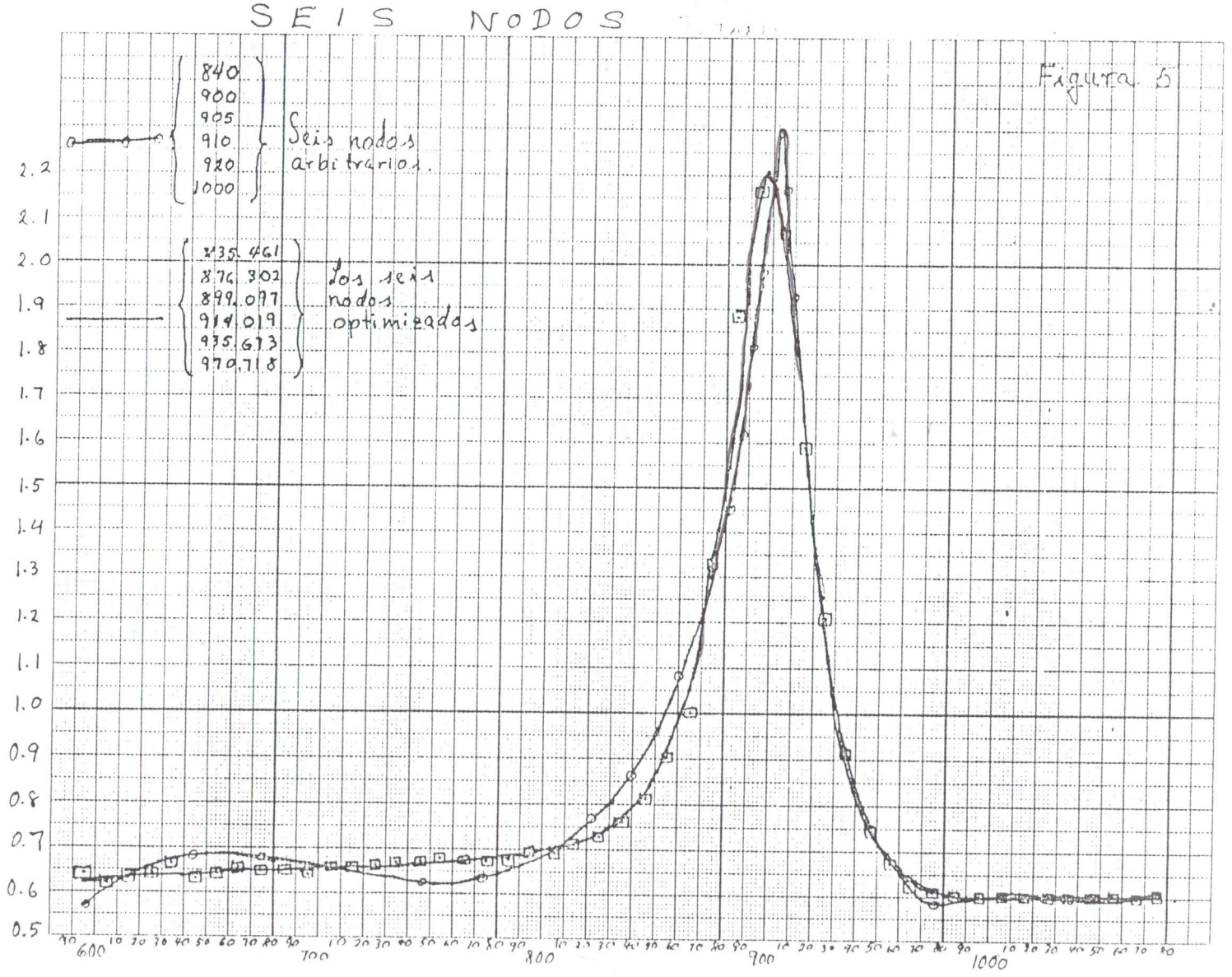


Figura 3

Ver → Tabla 2


PROBLEMA 2. TITANIO
CINCO NODOS
750
850
930
960
1000
5 nodos arbitrarios
835.4565
876.5063
894.1669
916.2794
974.0175
los 5 nodos optimizados.
2.00
1.90
1.80
1.70
1.60
1.50
1.40
1.30
1.20
1.10
1.00
.90
.80
.70
.60
.50
600
700
800
900
1000


SEIS NODOS
Figura 5
840
900
905
910
920
1000
Seis nodos arbitrarios.
835.461
876.302
899.097
914.019
975.613
979.718
Los seis nodos optimizados
2.2
2.1
2.0
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
600
700
800
900
1000

EJEMPLO 3. *La Logística.* Como se dijo en el capítulo IV, los datos experimentales corresponden al crecimiento de una población de bacterias. Al hacer el ajuste mediante splines con nodos libres, encontramos que dos nodos no son suficientes pero tres sí lo son. Para el método Levenberg-Marquardt encontramos dos nodos que se colapsan.

x_0	Meto-do	Nodos Optimizados	Residual	Grad	NFEV	NJEV	
105 155	LM/ SIGMA	97.3 169.8	40.40	10^{-5}	36	25	[Figura 6]
25 100 140	LM	78.21906 78.21960 166.68125	38.46	10^{-1}	38	18	[Figura 7]
,,	LM/ SIGMA	-0.0900 0.7610 85.9322	49.78	10^{-2}	24	19	[Figura 8]

Tabla 3

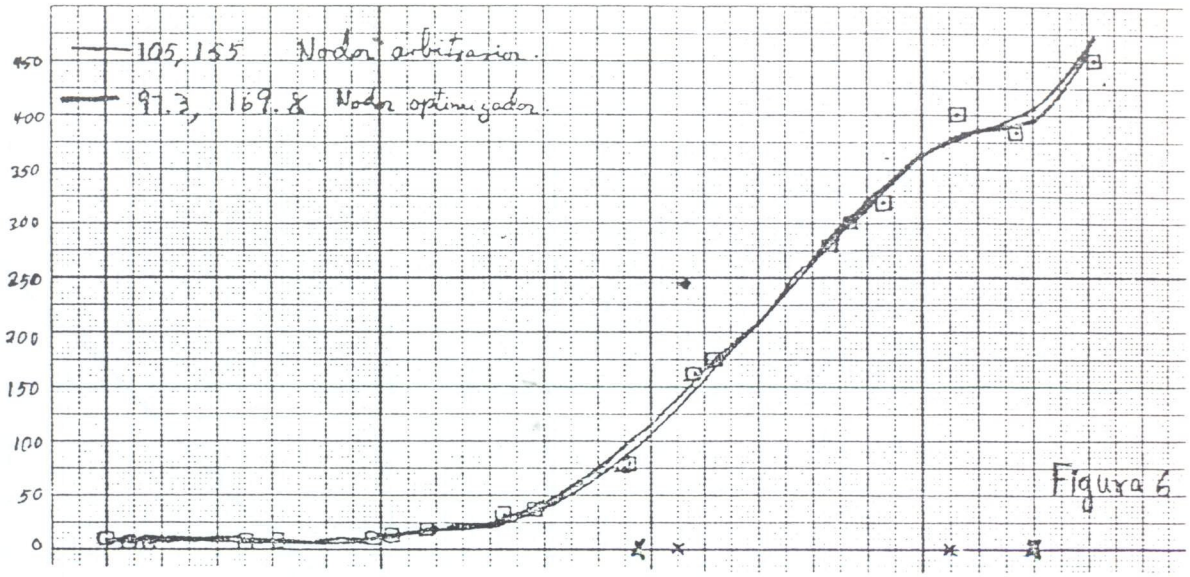

105, 155 Nodo arbitrario
97.3, 169.8 Nodo optimizado
450
400
350
300
250
200
150
100
50
0

Figura 6

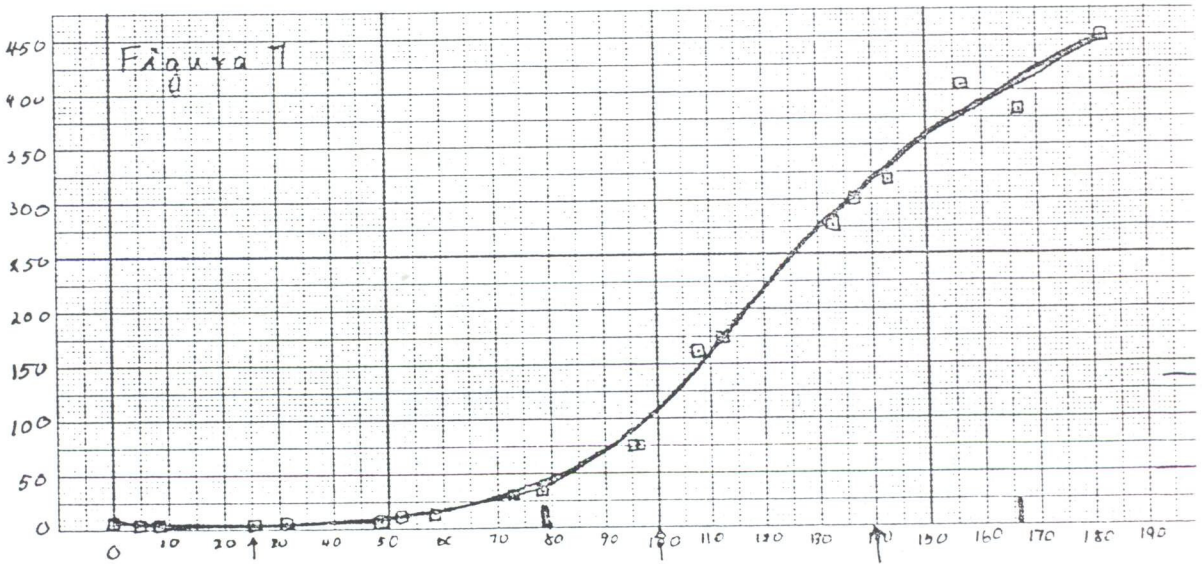
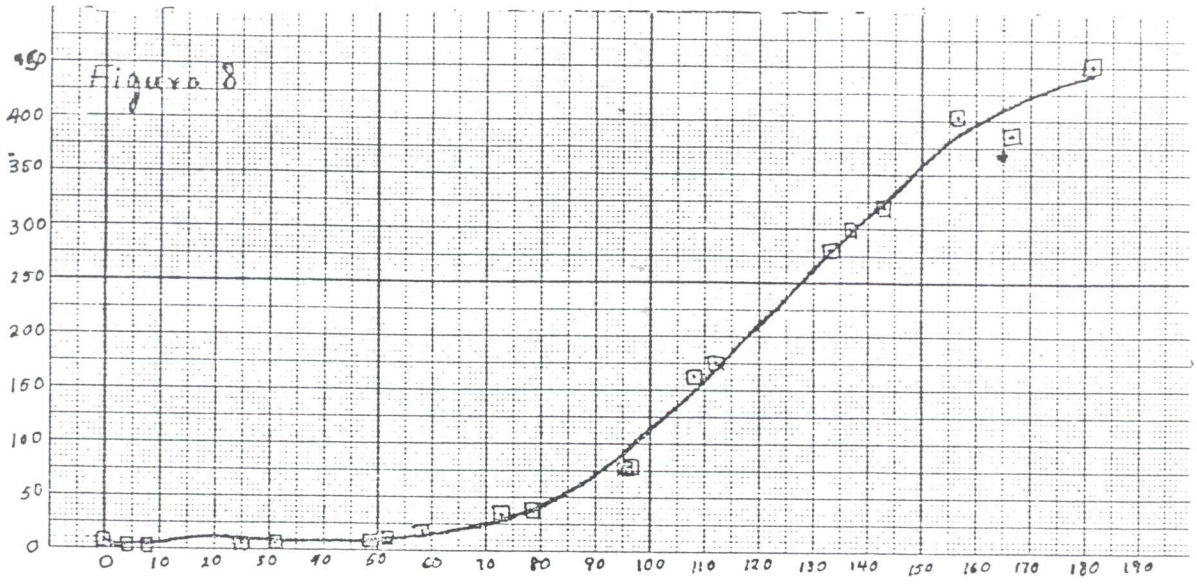


Figura 7
450
400
350
300
250
200
150
100
50
0
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190



EJEMPLO 4. El Problema de Barnes. Los datos de este problema ya han sido presentados en el capítulo IV. Este es un ejemplo de cómo una aproximación muy buena es inconveniente para nuestros propósitos de mantener la tendencia de los datos y de imitar la solución de una ecuación diferencial. Lo que se muestra en este problema es que con sólo un nodo se obtiene un spline mucho mejor en ese sentido, que con 3 no-

dos, porque en este caso el spline de ajuste sigue demasiado a los datos y aunque es menor el residual se pierde lo que hemos llamado la tendencia de los datos.

Empecemos con tres nodos y los datos de y_1. Hacemos notar además otro aspecto de la cuestión. La notable reducción de tiempo de cómputo que se alcanza en el método Levenberg-Marquardt con la transformación sigma. Esto se observa en las columnas NFEV y NJEV que, como ya dijimos, son el número de evaluaciones de la función y de la Jacobiana, respectivamente.

x_0	Método	Nodos Optimizados	Residual	Grad.	NFEV	NJEV
0.9 2.1 2.6	LM	0.831 1.000 2.917	0.08	10^{-10}	88	67
"	LM/ SIGMA	0.597 1.001 2.917	0.08	10^{-5}	22	16

Tabla 4

Veamos la gráfica del spline con los tres nodos optimizados utilizando σ, (figura 9). Este es el spline cúbico con tres nodos que mejor ajusta los datos. Pero observemos su mal comportamiento en el intervalo [0, 0.5]. Pero si reducimos a uno el número de nodos obtenemos los resultados de la tabla y el spline de la figura 10.

x_0	Método	Nodo Optimizado	Residual	Grad	NFEV	NJEV
1.0	LM	3.048	0.16	10^{-7}	28	19
3.0	"	"	"	10^{-8}	22	10

Tabla 5

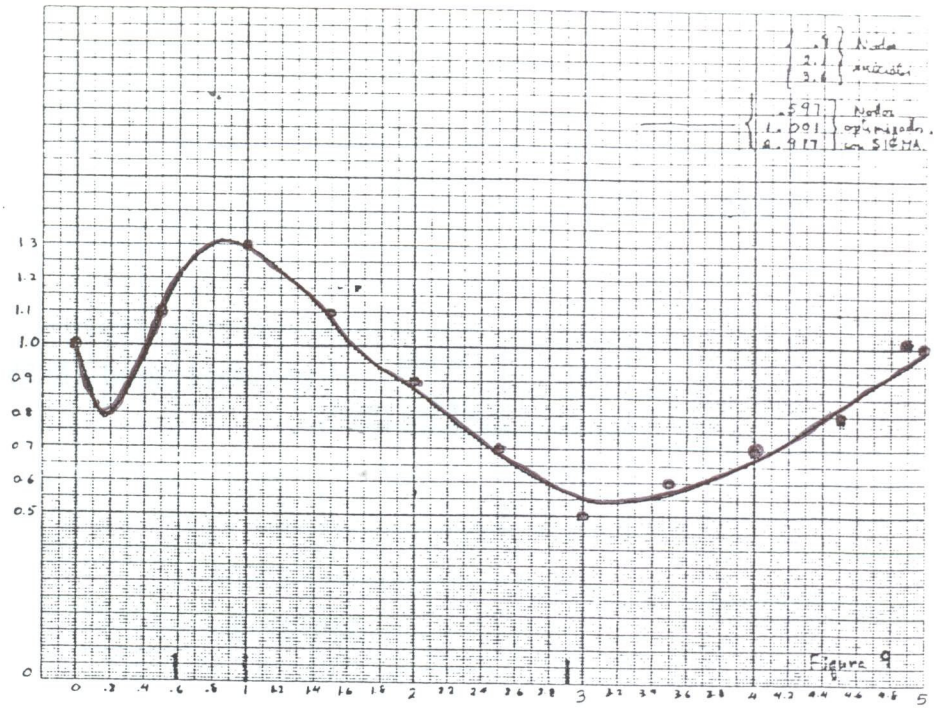


Figura 9

PROBLEMA DE BARNES ∇_1 . UN NOD

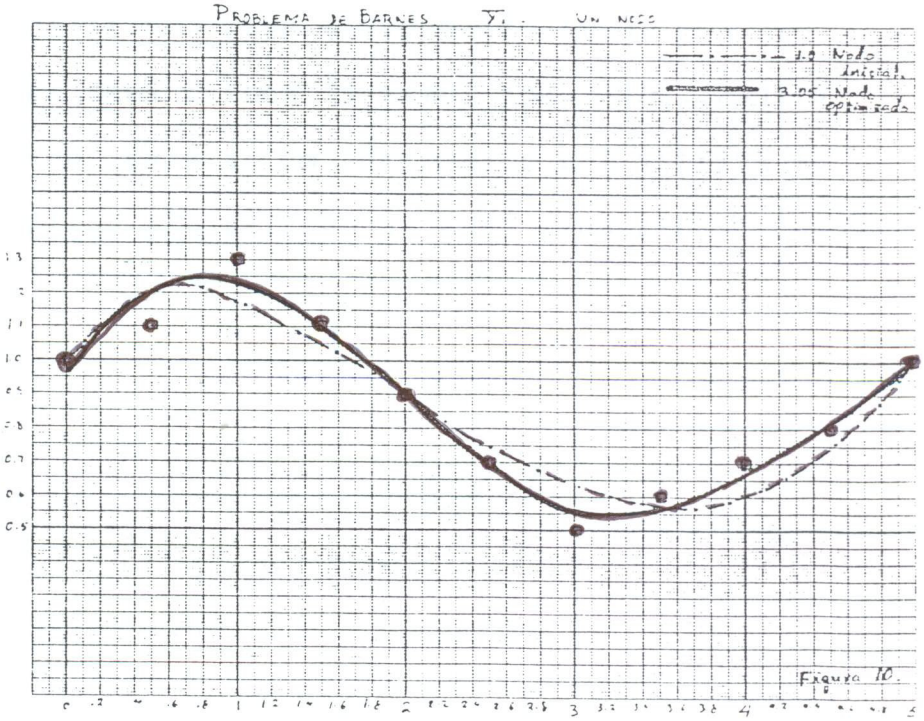


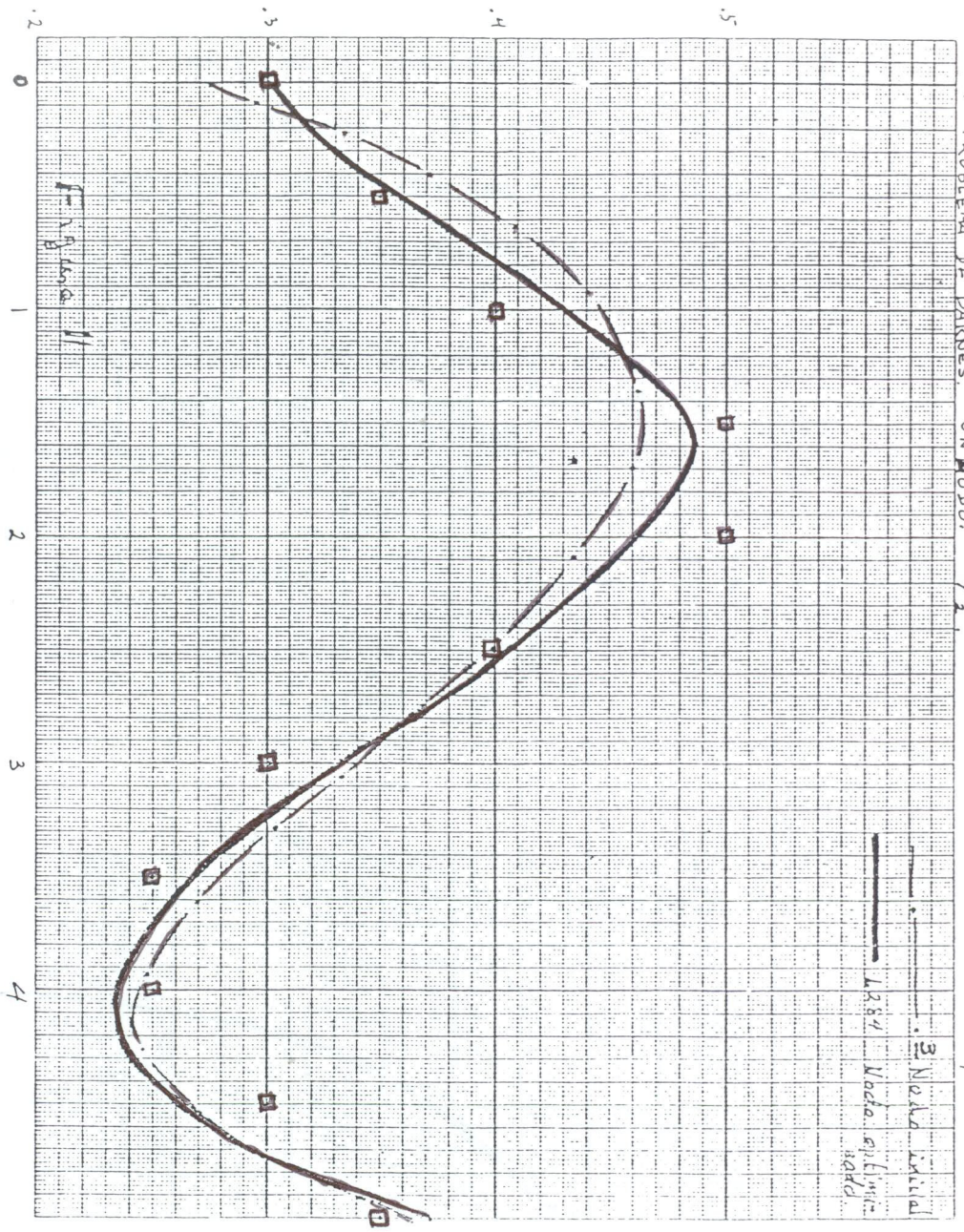
Figura 10.

Así pues, con un nodo, tenemos la posición óptima en 3.048. Por otra parte, para y_2:

x_0	Método	Nodo Optimizado	Residual	Grad	NFEV	NJEV
3	LM	1.287	0.08	10^{-9}	29	17

Tabla 6

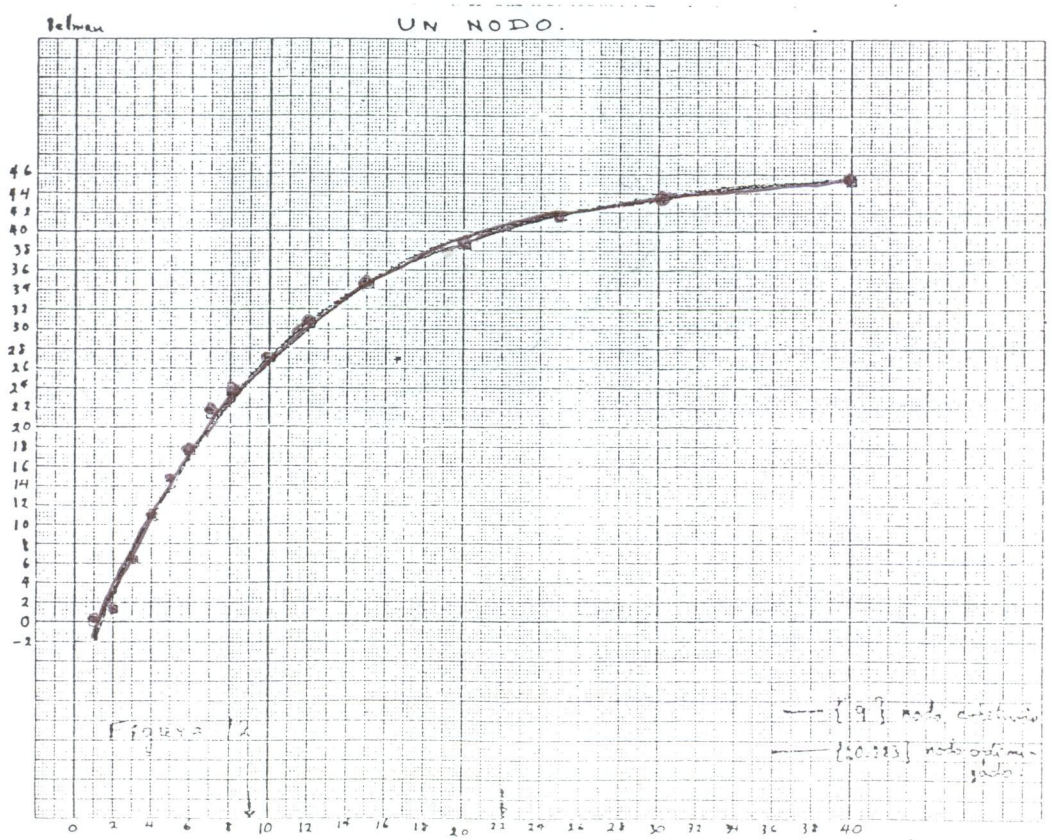
Y la sigura 11 muestra que, conciliando para las dos funciones, podríamos tomar el nodo 3 como adecuado. Ciertamente, al estimar los parámetros en la ecuación diferencial, da buenos resultados, como se muestra más adelante.


PROBLEMA DE BARNES. UN MODO.
3. Modo inicial
Modo elíptico
.2
.3
.4
.5
0
1
2
3
4

EJEMPLO 5 : El Problema de Bellman. Los datos están incluídos en el capítulo IV. Tenemos en este ejemplo, con sólo una función, un comportamiento parecido al del ejemplo anterior: un nodo es mejor que 2:

x_0	Método	Nodos Optimizados	Residual	Grad.	NFEV	NJEV
9.0	LM	20.22	2.66	10^{-10}	13	11
10	LM	2.68	0.896	10^{-8}	23	13
25		12.13				

Tabla 7



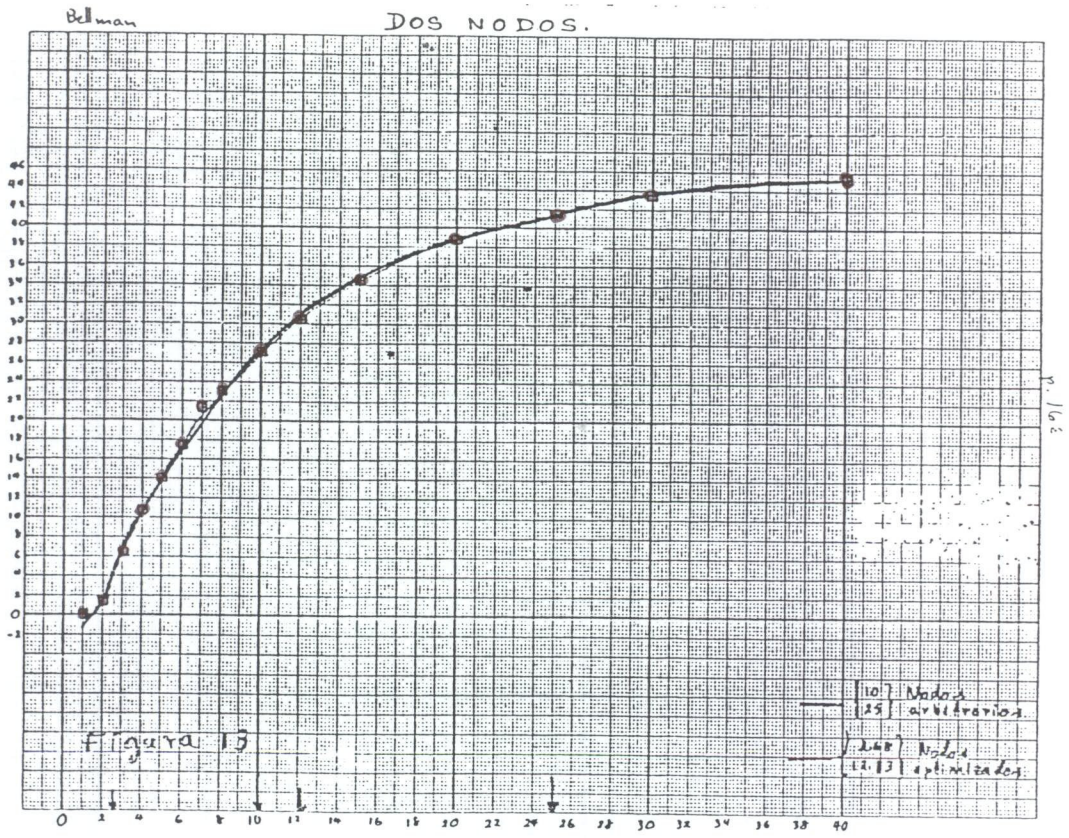
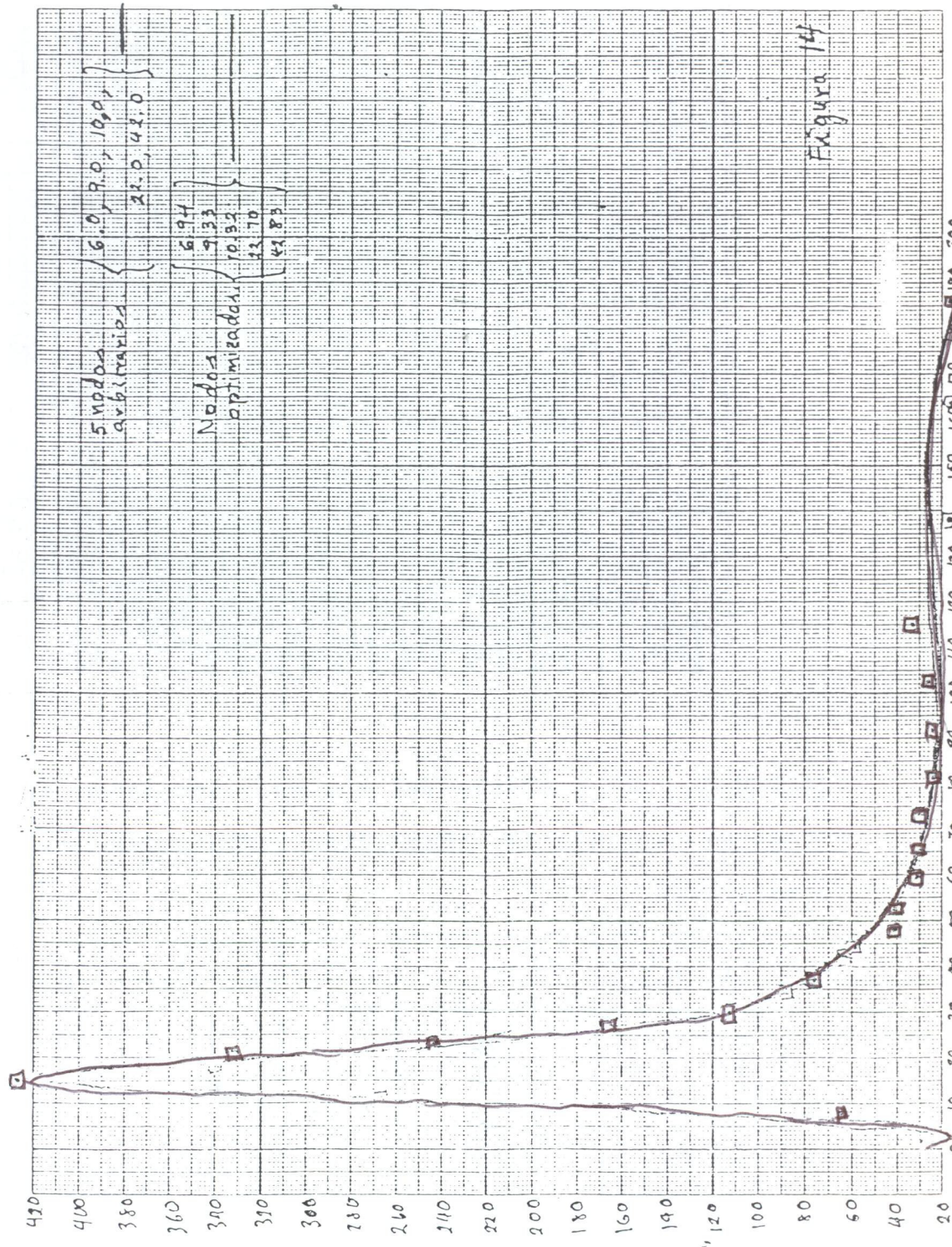


Figura 15

EJEMPLO 6. El problema del enzima. Los datos están incluídos en el capítulo IV. Con cinco nodos obtenemos el siguiente resultado.

x_0	Método	Nodos Optimizados	Residual	Grad.	NFEV	NJEV
6.0 9.0 10.0 22.0 42.0	LM	6.94 9.33 10.22 22.70 42.83	52.02	10^{-4}	24	14
"	LM/SIGMA	6.96 9.45 10.21 22.70 42.83	"	10^{-3}	15	9

Tabla 8


5 nodos arbitrarios
6.0, 8.0, 10.0, 21.0, 42.0
Nodos optimizados
6.94
9.33
10.32
42.70
41.83
Figura 14

EJEMPLO 7. El Problema de Brunhilda. Los datos de la siguiente tabla se refieren a la distribución de un sulfato radiactivo inyectado en la sangre de una mandril llamada Brunhilda, Jennrich y Bright [10].

t	2	4	6	8	10	15	20	25
y	151117	113601	97652	90935	84820	76891	73342	70593
t	30	40	50	60	70	80	90	110
y	67041	64313	61554	59946	57698	56440	53915	50938
t	130	150	160	170	180			
y	48717	45996	44968	43602	42668			

Tabla 9

Para ajustar estos datos con un spline utilizamos dos nodos, 15 y 80. Intentamos optimizar estos nodos con el método Levenberg-Marquardt y éste no nos dá solución aceptable. Al utilizar el método Levenberg-Marquardt con la transformación σ, obtenemos el primer nodo en 11.71 y el segundo colapsado con el poste derecho (180).

x_0	Método	Nodos Optimizados	Residual	Grad.	NFEV	NJEV
15 80	LM	11.996 827.54 (Fuera del intervalo)				
,,	LM/ SIGMA	11.71 179.57	8116	10^4	32	24

Tabla 10

En realidad no sabemos interpretar estos resultados, y más aún viendo la siguiente figura que representa los dos - splines, los cuales, aparentemente son un ajuste muy bueno.

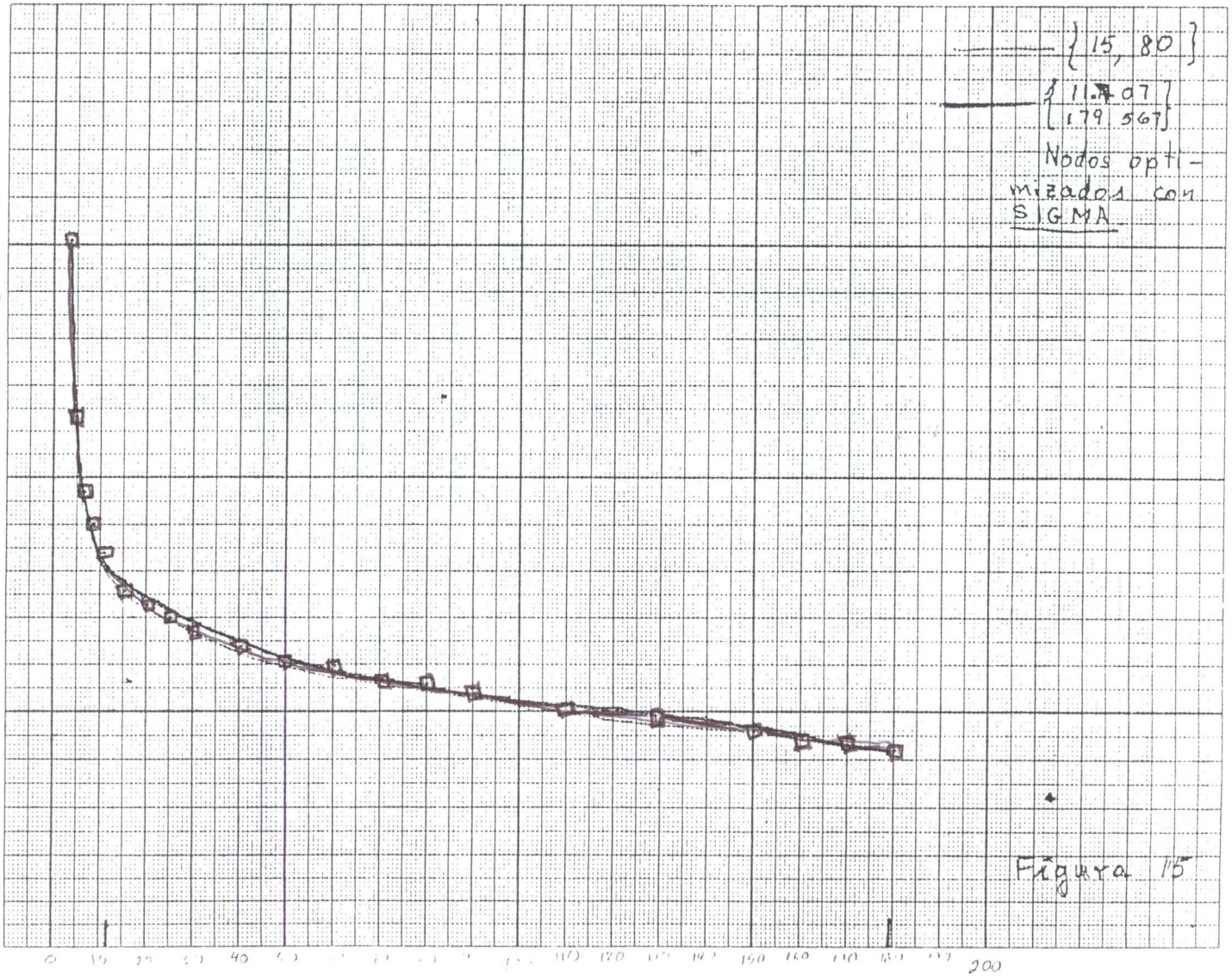


Figura 15

PROBLEMA 8. Los siguientes datos expresan los cambios en los precios mundiales del azúcar en un período de 30 años.

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13
y	7	3	1	3	0	4	6	10	15	18	15	15	35	44
t	14	15	16	17	18	19	20	21	22	23	24	25	26	27
y	19	22	74	50	38	37	29	16	7	3	10	13	10	8
t	28	29	30											
y	10	6	5											

Tabla 11

Jupp [12] estudia este problema para ajuste con splines con nodos libres y para el caso de 7 nodos determina un óptimo

$$x^* = (7.4,\ \underline{10.14},\ \underline{10.40},\ 13.23,\ \underline{15.27},\underline{15.62},\underline{15.97})$$

que incluye una confluencia doble y una confluencia triple.

La siguiente tabla muestra que cuando nosotros aplicamos el método Levenberg-Marquardt, sin la transformación σ no obtuvimos resultado; en cambio con ella, obtenemos el óptimo, incluyendo las dos confluencias, lo que muestra que la transformación no excluye soluciones con nodos múltiples - cuando realmente son mínimos (o mínimos locales).

x_0	Método	Nodos Optimizados	Residual	Grad.	NFEV	NJEV
7.0 10.0 10.5 13.2 15.2 15.6 16.0	LM	No hubo orden				
"	LM/ SIGMA	7.454 10.065 10.481 13.216 15.257 15.630 15.994	15.6	10^{-1}	10	6

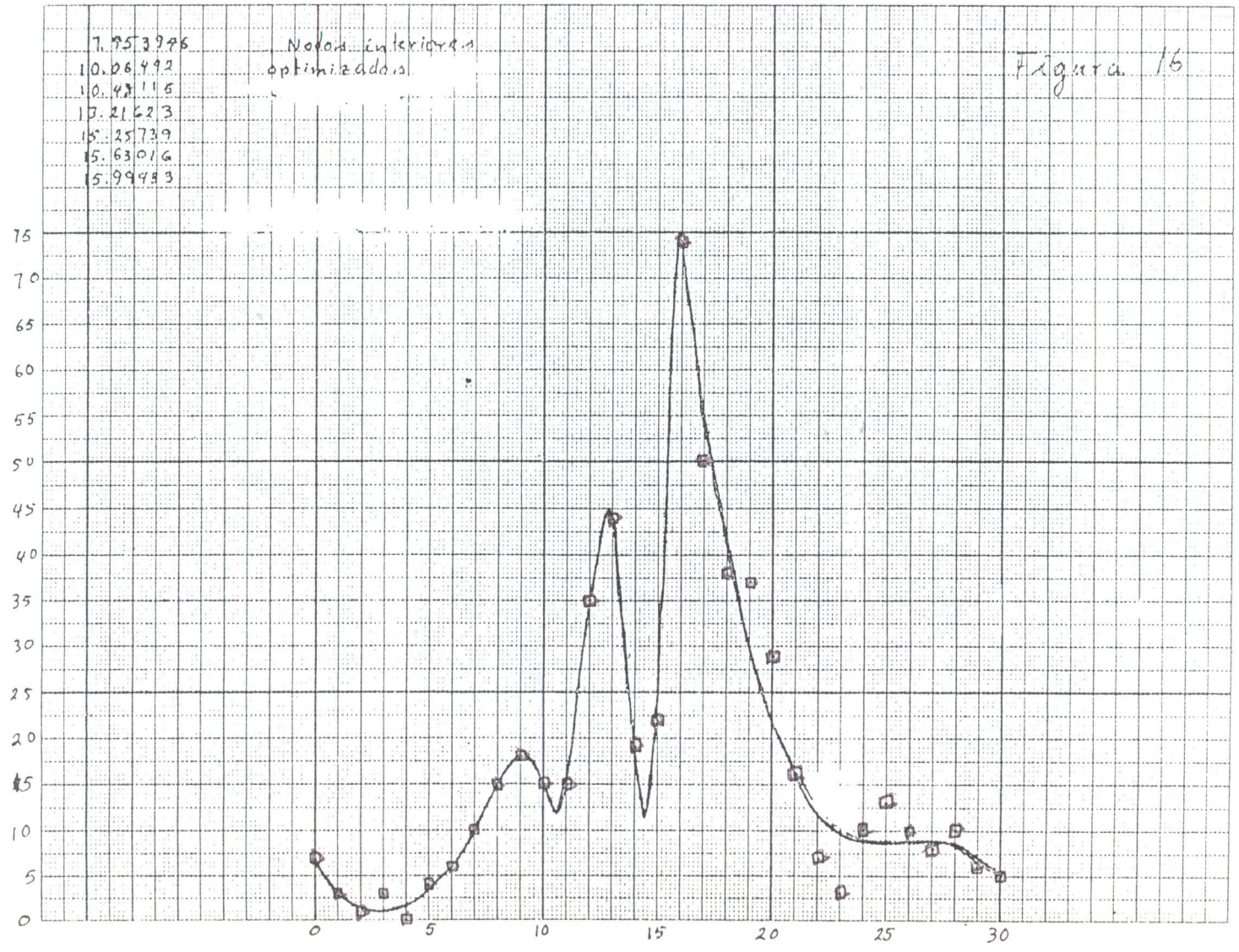
Tabla 12

(7 NODOS LIBRES)

7.953996
10.06492
10.43116
10.21623
15.25739
15.63016
15.99993

Nodos interiores optimizados

Figura 15



5.3 Ejemplos de Estimación de Parámetros.

EJEMPLO A. El Problema de Bellman

$$y' = c_1(126.2 - y)(91.9 - y)^2 - c_2 y^2$$

Una vez construído el spline, se eligen los puntos muestrales $\{\hat{t}_i : i = 1,\ldots,M\}$ y se minimiza para $\gamma = (c_1, c_2)$

$$\sum_{i=1}^{M} [s'(\hat{t}_i) - f(\hat{t}_i, s(\hat{t}_i), \gamma)]^2$$

donde

$$f(t,s,\gamma) = c_1(126.2 - s(t))(91.9 - s(t))^2 - c_2 s^2(t)$$

Es un problema lineal, y si hacemos

$$\phi_1(t) = (126.2 - s(t))(91.9 - s(t)^2)$$

$$\phi_2(t) = -s^2(t),$$

y la matriz

$$A = \begin{bmatrix} \phi_1(t_1) & \phi_2(t_1) \\ \phi_1(t_2) & \phi_2(t_2) \\ \vdots & \\ \phi_1(t_M) & \phi_2(t_M) \end{bmatrix}$$

el problema lo podemos expresar en la forma

$$\min_{\gamma} \| s'(t_i) - A\gamma \|^2.$$

La siguiente tabla muestra que con un nodo se obtiene mejores resultados que con dos nodos.

Nodos	Residual en el spline	Residual en la ec. diferencial. (No. de pts. Muestrales)	Parámetros	Residual integrado.	Mejor cond.inic.
20.22	2.66	0.448 (15)	$0.45 \times 10^{-5}; 0.29 \times 10^{-3}$	5.632	-1.678
20.22	2.66	0.618 (40)	$0.46 \times 10^{-5}; 0.30 \times 10^{-3}$	6.106	-0.980
5.0;15.0		3.01 (20)	$0.40 \times 10^{-5}; 0.17 \times 10^{-3}$	5.76	0.89
5.0;15.0		3.09 (40)	$0.41 \times 10^{-5}; 0.23 \times 10^{-3}$	6.37	0.33

Tabla 13

Figura 17

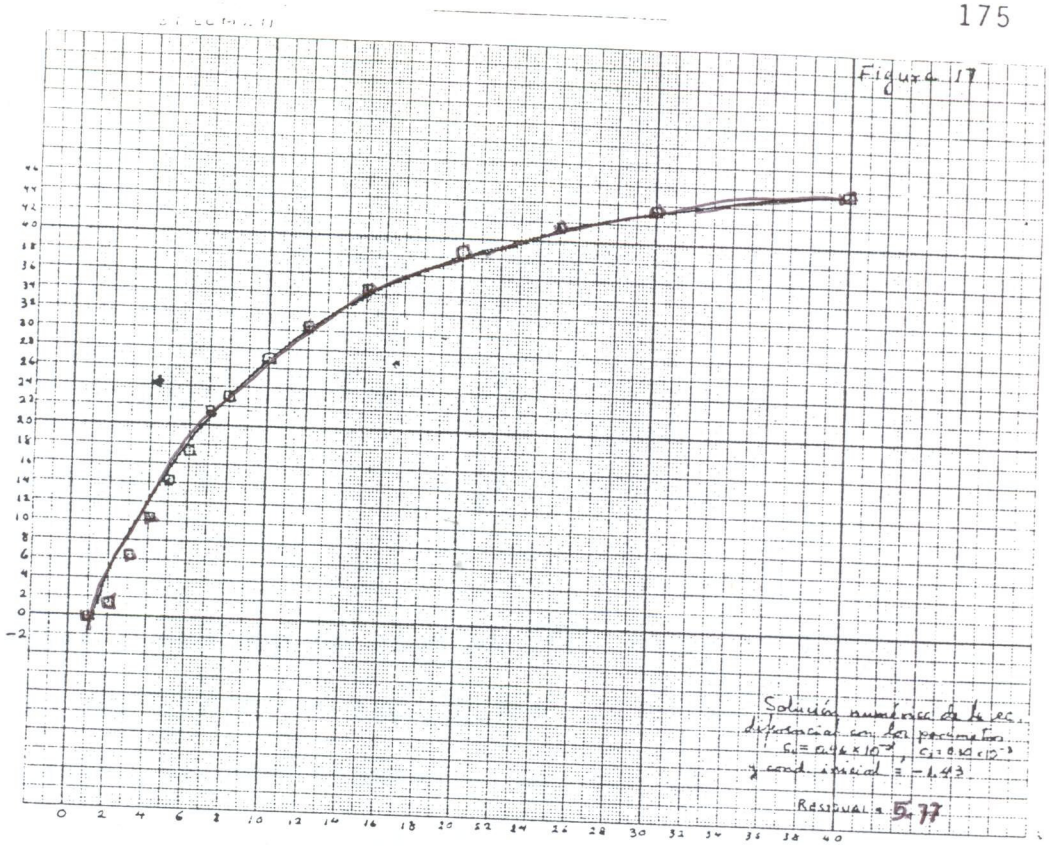
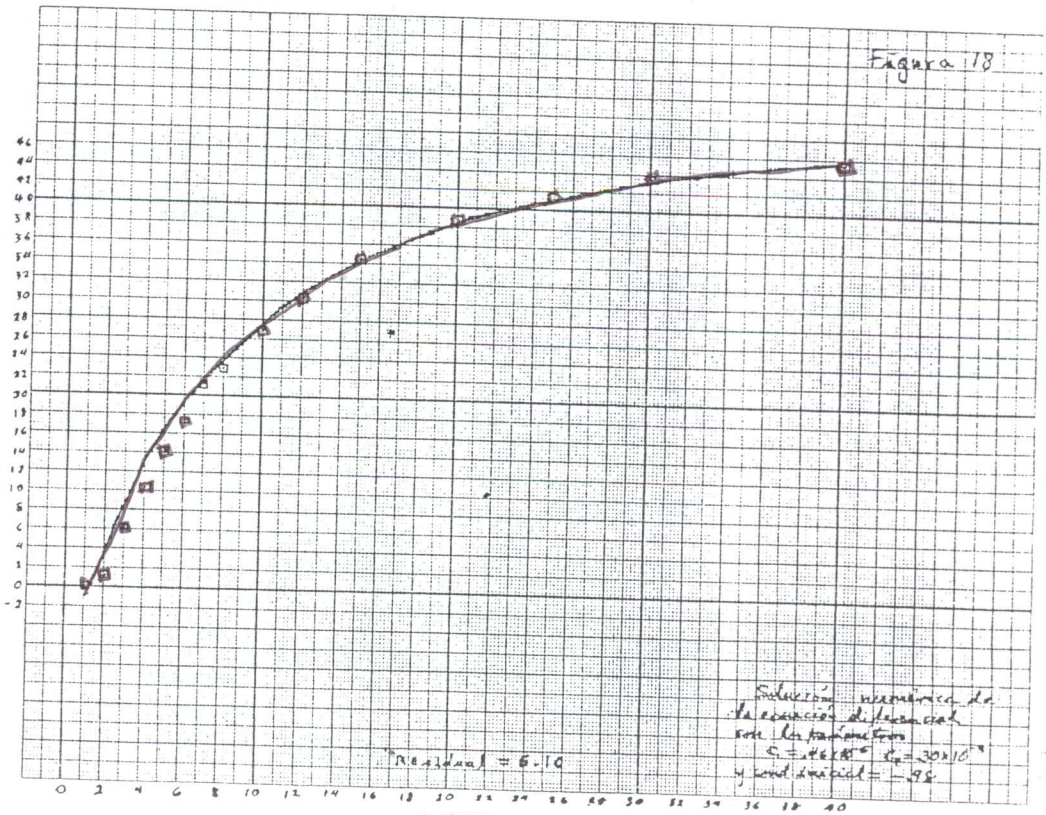


Figura 18



EJEMPLO B. *El ejemplo de Barnes*

$$y_1' = c_1 y_1 - c_2 y_1 y_2$$

$$y_2' = c_2 y_1 y_2 - c_3 y_2$$

Recordemos que, después de que hemos encontrado los splines $s_1(t)$ y $s_2(t)$ que aproximan los datos correspondientes a y_1 y y_2, respectivamente, debemos minimizar la expresión del capítulo IV

$$\sum_{j=1}^{p} \sum_{i=1}^{M} [s_j'(\hat{t}_i) - f_j(\hat{t}_i, s(\hat{t}_i), \gamma)]^2 .$$

Como los parámetros aparecen linealmente, podemos darle una estructura al problema utilizando este hecho, teniendo en cuenta que $p = 2$, y

$$f_1(\hat{t}_i, s(\hat{t}_i), \gamma) = c_1 s_1(\hat{t}_i) - c_2 s_1(\hat{t}_i) s_2(\hat{t}_i)$$

$$f_2(\hat{t}_i, s(\hat{t}_i), \gamma) = c_2 s_1(\hat{t}_i) s_2(\hat{t}_i) - c_3 s_2(\hat{t}_i),$$

podemos definir

$$\overline{s}_1' = \begin{bmatrix} s_1'(\hat{t}_1) \\ s_1'(\hat{t}_2) \\ \vdots \\ s_1'(\hat{t}_M) \end{bmatrix}, \qquad \overline{s}_2' = \begin{bmatrix} s_2'(\hat{t}_1) \\ s_2'(\hat{t}_2) \\ \vdots \\ s_2'(\hat{t}_M) \end{bmatrix}$$

$$A_1 = \begin{bmatrix} s_1(\hat{t}_1) & -s_1(\hat{t}_1) & s_2(\hat{t}_1) & 0 \\ s_1(\hat{t}_2) & -s_1(\hat{t}_2) & s_2(\hat{t}_2) & 0 \\ & & \vdots & \\ s_1(\hat{t}_M) & -s_1(\hat{t}_M) & s_2(\hat{t}_M) & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & s_1(\hat{t}_1) & s_2(\hat{t}_1) & -s_2(\hat{t}_1) \\ 0 & s_1(\hat{t}_2) & s_2(\hat{t}_2) & -s_2(\hat{t}_2) \\ & & \vdots & \\ 0 & s_1(\hat{t}_M) & s_2(\hat{t}_M) & -s_2(\hat{t}_M) \end{bmatrix}$$

Y si llamamos $g(\gamma)$ a la función que queremos minimizar, tenemos

$$g(\gamma) = ||\bar{\delta}'_1 - A_1\gamma||^2 + ||\bar{\delta}'_2 - A_2\gamma||^2 ,$$

o sea

$$g(\gamma) = ||\bar{\delta}'_1||^2 - 2\bar{\delta}'^{T}_1 A_1\gamma + \gamma^T A_1^T A_1\gamma$$

$$+ ||\bar{\delta}'_2||^2 - 2\bar{\delta}'^{T}_2 A_2\gamma + \gamma^T A_2^T A_2\gamma$$

Ahora procedemos como sigue

$$g'(\gamma) = -2A_1^T\bar{\delta}'_1 + 2A_1^T A_1\gamma - 2A_2^T\bar{\delta}'_2 + 2A_2^T A_2\gamma .$$

Para obtener el mínimo, igualamos a cero

$$(A_1^T A_1 + A_2^T A_2)\gamma = b$$

donde el segundo miembro del sistema es

$$b = A_1^T\bar{\delta}'_1 + A_2^T\bar{\delta}'_2 .$$

Una simplificación es posible de la siguiente manera:

Hagamos

$$\tilde{A}_1 = \begin{bmatrix} s_1(\hat{t}_1) & -s_1(\hat{t}_1) & s_2(\hat{t}_1) \\ s_1(\hat{t}_2) & -s_1(\hat{t}_2) & s_2(\hat{t}_2) \\ \vdots & & \\ s_1(\hat{t}_M) & -s_1(\hat{t}_M) & s_2(\hat{t}_M) \end{bmatrix}, \quad \tilde{A}_2 = \begin{bmatrix} s_1(\hat{t}_1) & s_2(\hat{t}_1) & -s_2(\hat{t}_1) \\ s_1(\hat{t}_2) & s_2(\hat{t}_2) & -s_2(\hat{t}_2) \\ & \vdots & \\ s_1(\hat{t}_M) & s_2(\hat{t}_M) & -s_2(\hat{t}_M) \end{bmatrix}$$

y al hacer la descomposición QR de estas dos matrices obtenemos

$$\tilde{A}_1 = \tilde{Q}_1 \tilde{R}_1, \quad \tilde{A}_2 = \tilde{Q}_2 \tilde{R}_2 ,$$

donde $\tilde{Q}_1$, $\tilde{Q}_2$ son matrices ortogonales $M \times M$ y $\tilde{R}_1$ y $\tilde{R}_2$ son triangulares superiores $M \times 2$. Haciendo $Q_1 = \tilde{Q}_1$ y $Q_2 = \tilde{Q}_2$, obtenemos

$$A_1 = Q_1 R_1 , \quad A_2 = Q_2 R_2$$

donde

$$R_1 = \begin{bmatrix} \begin{bmatrix} \tilde{R}_1 \end{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{y} \quad R_2 = \begin{bmatrix} 0 \begin{bmatrix} \tilde{R}_2 \end{bmatrix} \\ 0 \end{bmatrix}$$

tienen dimensiones $M \times 3$.

Así pues, el sistema

$$(A_1^T A_1 + A_2^T A_2)\gamma = b$$

se convierte en

$$(R_1^T R_1 + R_2^T R_2)\gamma = b$$

$$R\gamma = b$$

donde $R = R_1^T R_1 + R_2^T R_2$ es una matriz tridiagonal 3×3, simétrica. Aplicando el algoritmo de Cholesky obtenemos la solución para γ.

Resultados.

Nodos	El residual en la ec.dif. (No. de pts. muestrales)	Parámetros	El residual integrado	Condiciones iniciales
3.0	1.260 (20)	0.8461; 2.135; 1.913	0.3530	1.02;0.25
3.0	1.724 (40)	0.8040; 2.056; 1.857	0.3603	1.05;0.26
1.5, 3.0	1.047 (20)	0.8500; 2.197; 2.036	0.3814	1.02;0.24
1.5, 3.0	1.465 (40)	0.8315; 2.168, 2.007	0.3742	1.04;0.24

Tabla 14

Barnes

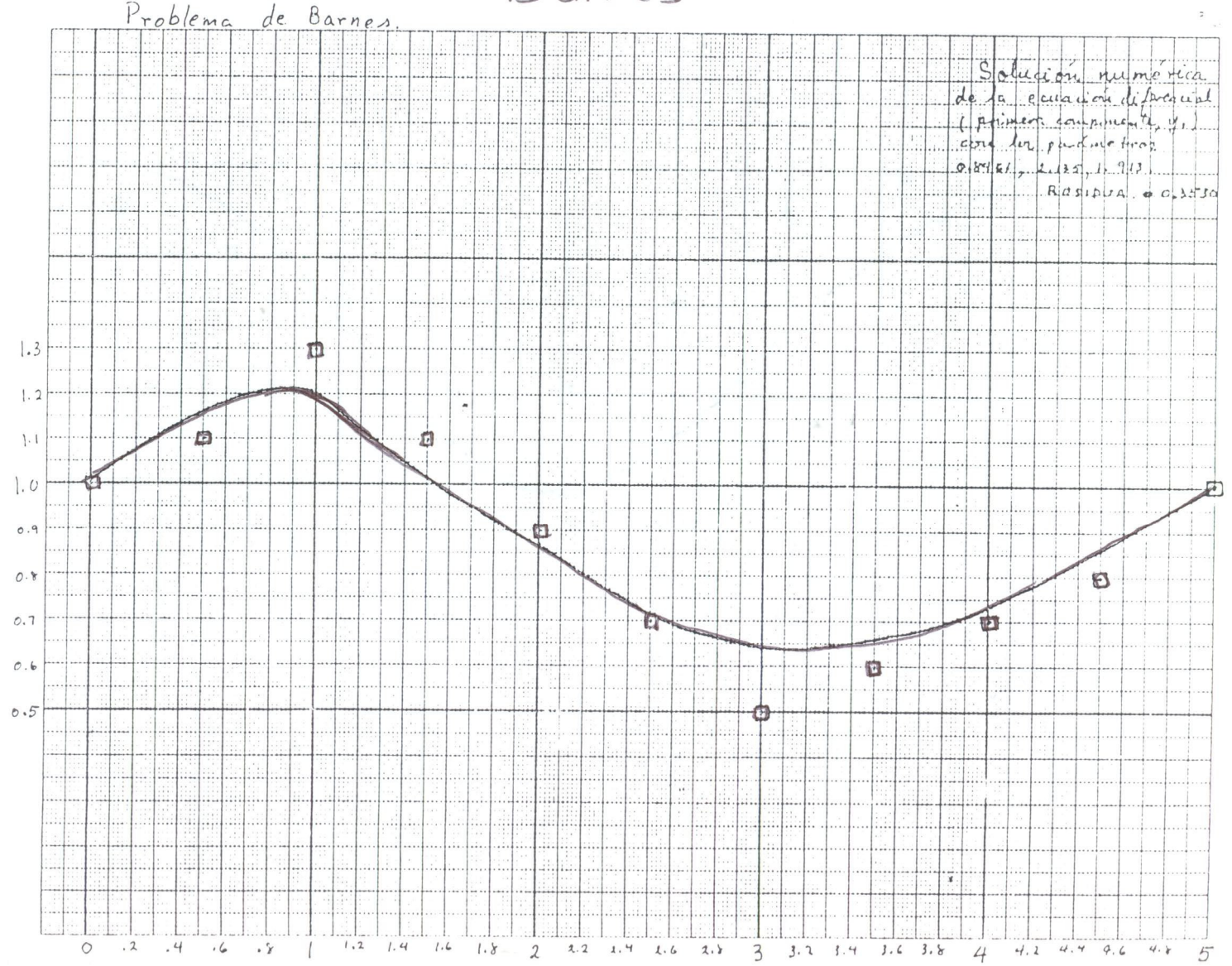


Figura 19

Barnes

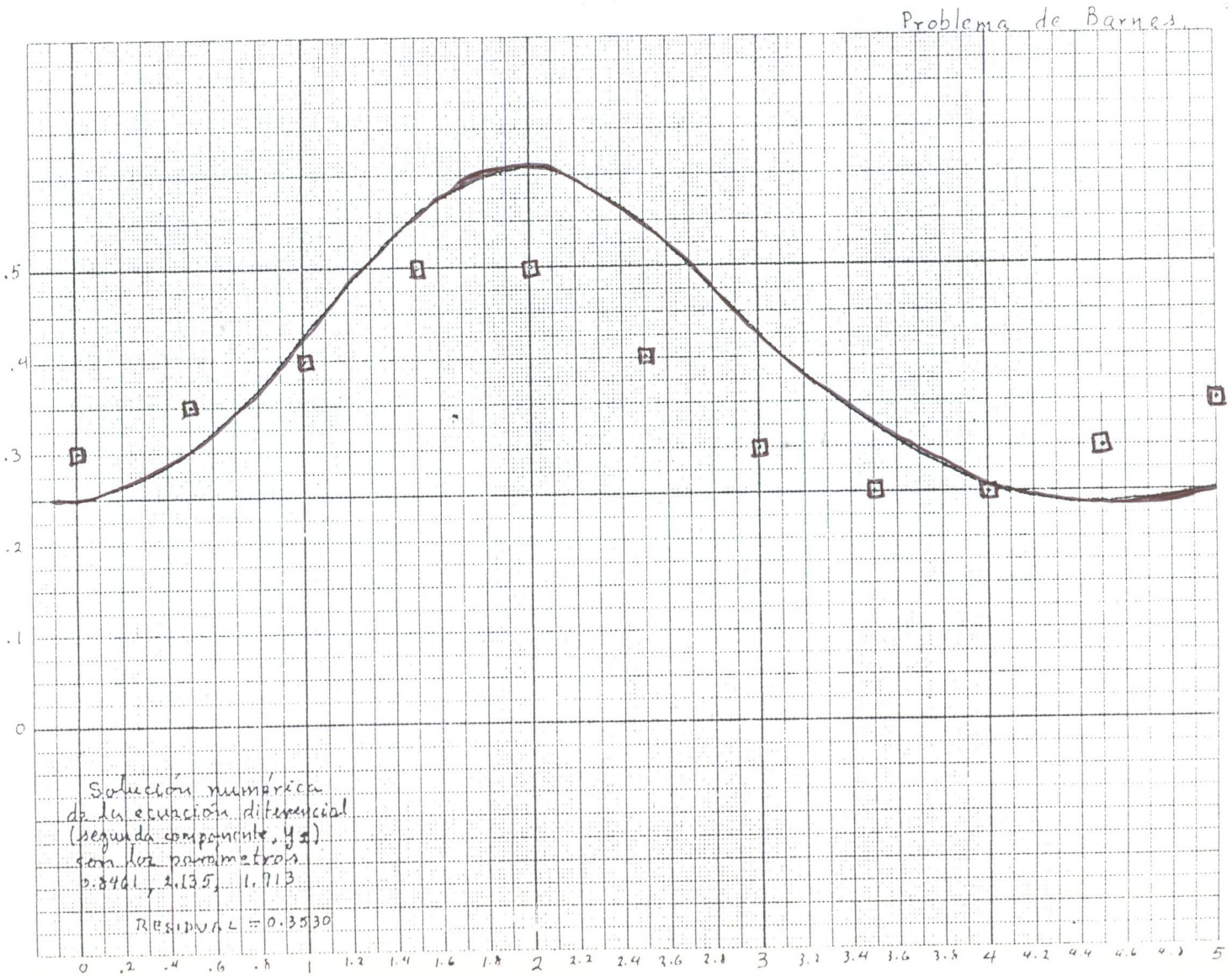


Figura 20

EJEMPLO C: El problema del enzima

$$y_1' = c_1(27.8 - y_1) + \frac{c_4}{2.6}(y_2 - y_1) + \frac{4991}{t\sqrt{2\pi}} \exp[-0.5[\frac{\ln t - c_2}{c_3}]]$$

$$y_2' = \frac{c_4}{2.7}(y_1 - y_2)$$

Los datos disponibles, que hemos mostrado ya en el capítulo IV, corresponden únicamente a la variable y_1. Esta dificultad la resolvemos despejando y_2 de la primera ecuación, diferenciando y substituyendo en la segunda ecuación, para obtener una ecuación de segundo orden en y_1 únicamente. He aquí los detalles:

Primero, hacemos

$$A = 27.8,\ B = \frac{1}{2.6},\ C = \frac{4991}{\sqrt{2\pi}},\ D = -0.5,\ E = \frac{1}{2.7}$$

Y así, el sistema lo podemos escribir como

$$y_1' = c_1(A - y_1) + Bc_4(y_2 - y_1) + \frac{C}{t}\exp[D[\frac{\ln t - c_2}{c_3}]^2] \quad \ldots(1)$$

$$y_2' = E\, c_4(y_1 - y_2) \quad (2)$$

De la primera ecuación obtenemos

$$Bc_4\, y_2 = y_1' - c_1(A - y_1) + Bc_4\, y_1 - \frac{C}{t}\, exp\,[\,D\,[\,\frac{\ell n\, t - c_2}{c_3}\,]^2\,]\,.$$

Hagamos

$$g(t) = -\frac{C}{t}\, exp\,[\,Dh^2(t)\,]\;;\quad h(t) = \frac{\ell n\, t - c_2}{c_3}$$

y así

$$Bc_4\, y_2 = y_1' - c_1(A - y_1) + Bc_4\, y_1 + g(t) \quad ...(3)$$

Derivando:

$$Bc_4\, y_2' = y_1'' + c_1\, y_1' + Bc_4\, y_1' + g'(t) \quad ..... \quad (4)$$

Por (2):

$$Bc_4\, y_2' = BE\, c_4^2(y_1 - y_2) = BE\, c_4^2\, y_1 - Ec_4\, Bc_4\, y_2 \quad ...(5)$$

Por (3), (4) y (5).

$$y_1'' + (c_1 + Bc_4 + Ec_4)y_1' + Ec_1\, c_4\, y_1 + Ec_4\, g(t)$$

$$+ g'(t) - AEc_1\, c_4 = 0 \quad ...(6)$$

y además

$$g'(t) = -\frac{g(t)}{t}\,(1 + \frac{h(t)}{c_3})$$

La ecuación (6) es la ecuación diferencial que buscábamos. Es una ecuación diferencial de 2º orden en la que sólo

aparece y_1 como función incógnita. Dejando y por y_1, podemos escribir

$$y'' = -(c_1 + Bc_4 + Ec_4)y' - Ec_1c_4\, y - Ec_4\, g(t) - g'(t) + AEc_1c_4$$

Es una ecuación diferencial del tipo

$$y'' = f(t, y, y', \gamma)$$

donde el vector de parámetros es $\gamma = (c_1, c_2, c_3, c_4)^T$. Para determinar estos parámetros según la técnica del capítulo IV, después de elegir un conjunto de puntos muestrales debemos minimizar

$$\sum_{i=1}^{M} [s''(\hat{t}_i) - f(\hat{t}_i, s, s', \gamma)]^2 \qquad \ldots (7)$$

con respecto a γ. Es un problema no-lineal de suma mínima de cuadrados. Para usar el método Levenberg-Marquardt necesitamos la matriz Jacobiana

$$J_{ij} = \left[\frac{\partial f}{\partial c_j}(t_i)\right], \quad i = 1, \ldots, M; \; j = 1, \ldots, 4.$$

Vamos a obtener estas parciales. Hagamos

$$F(\gamma) = A_1(\gamma)y' + A_2(\gamma)y + Ec_4\ G(\gamma) + G'(\gamma) - AEc_1c_4$$

donde

$$A_1(\gamma) = c_1 + Bc_4 + Ec_4$$

$$A_2(\gamma) = Ec_1c_4$$

$$G(\gamma) = -\frac{c}{t}\ exp\ [D\ H^2(\gamma)]$$

$$H(\gamma) = \frac{\ln t - c_2}{c_3}$$

$$G'(\gamma) = -\frac{G(\gamma)}{t}\left(1 + \frac{H(\gamma)}{c_3}\right)$$

Entonces

$$\frac{\partial F}{\partial c_1} = y' + Ec_4y - AEc_4 = y' + Ep_4(y - A)$$

Ahora:

$$\frac{\partial G}{\partial c_2} = \frac{H(\gamma)\ G(\gamma)}{c_3}$$

$$\frac{\partial G'}{\partial c_2} = \frac{G(\gamma)}{c_3 t}\left[\frac{1}{c_3} - H(\gamma)\left(1 + \frac{H(\gamma)}{c_3}\right)\right]$$

Y como

$$\frac{\partial F}{\partial c_2} = E c_4 \frac{\partial G(\gamma)}{\partial c_2} + \frac{\partial G'(\gamma)}{\partial c_2} ,$$

queda

$$\frac{\partial F}{\partial c_2} = \frac{G(\gamma)}{c_3} [E H(\gamma) c_4 + \frac{1}{t} (\frac{1}{c_3} - H(\gamma)(1 + \frac{H(\gamma)}{c_3})].$$

Por otro lado:

$$\frac{\partial F}{\partial c_3} = E c_4 \frac{\partial G}{\partial c_3} + \frac{\partial G'}{\partial c_3} .$$

Necesitamos

$$\frac{\partial G}{\partial c_3} = - GH \frac{\partial H}{\partial c_3} , \quad \text{pero} \quad \frac{\partial H}{\partial c_3} = - \frac{H}{c_3} , \text{ y así}$$

$$\frac{\partial G}{\partial c_3} = \frac{G H^2}{c_3} .$$

$$\frac{\partial G'}{\partial c_3} = \frac{G H}{c_3 t} [\frac{1}{c_3} - H(1 + \frac{H}{c_3})] + \frac{G}{c_3^3 t}$$

y así

$$\frac{\partial F}{\partial c_3} = Ec_4 \frac{GH^2}{c^3} + \frac{GH}{c_3 t}\left[\frac{1}{c_3} - H\left(i + \frac{H}{c_3}\right]\right. + \frac{G}{c_3^3 t}$$

$$= \frac{GH}{c_3}\left[E H c_4 + \frac{1}{t}\left[\frac{1}{c_3} - H\left(1 + \frac{H}{c_3}\right)\right]\right\} + \frac{G}{c_3^3 t}$$

Finalmente

$$\frac{\partial F}{\partial c_4} = \frac{\partial A1}{\partial c_4} y' + \frac{\partial A2}{\partial c_4} y + E\,G(\gamma) - A\,E\,c_1$$

$$= (B + E)y' + Ec_1 y + E\,G(\gamma) - A\,E\,c_1$$

$$= (B + E)y' + Ec_1(y - A) + E\,G(\gamma).$$

Resumiendo: para obtener el mínimo de la expresión (7) utilizamos el método Levenberg-Marquardt con la matriz Jacobiana construída con las fórmulas para las derivadas parciales que se acaban de obtener. Los resultados obtenidos son los siguientes

Nodos	Residual en la ec. [Nº de pts. muestrales]	Parámetros	Residual integrado	Condic. iniciales
8, 11, 23, 43	6.6 [40]	0.24, 2.6 0.37, 0.30	109	27.8,-3
	6.6 [28]	0.250, 2.63 0.354, 0.324	82	27, -4.1

Tabla 15

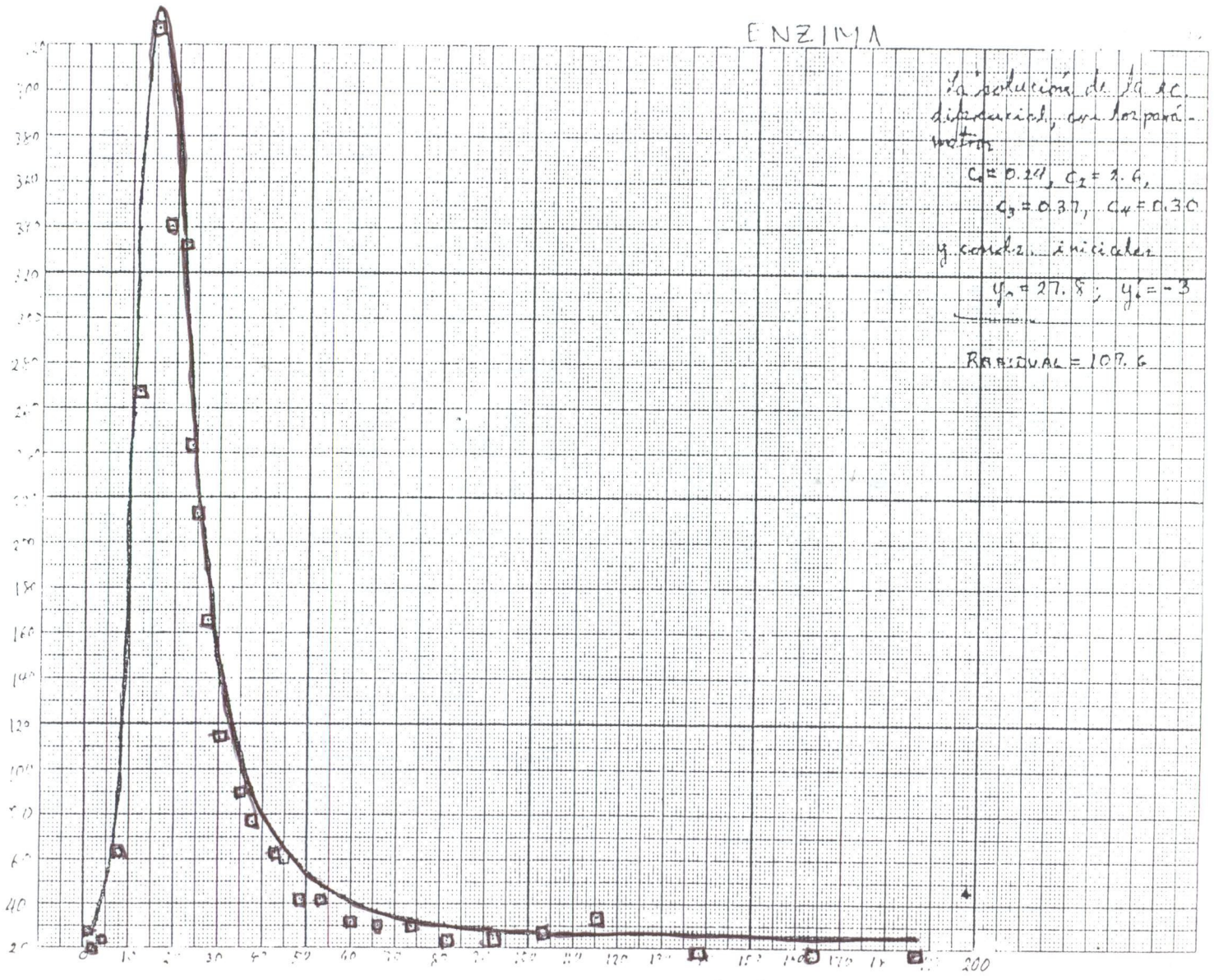

ENZIMA
La solución de la E.C. diferencial, con Runge-Kutta
$C_1 = 0.24$, $C_2 = 1.6$,
$C_3 = 0.37$, $C_4 = 0.30$
y condic. iniciales
$y_1 = 27.8$; $y_1' = -3$
RESIDUAL = 107.6

Figura 2.

EJEMPLO D. *La logística.* En el capítulo IV mostramos datos que corresponden al crecimiento de una población de bacterias. La ecuación diferencial es

$$y' = k_1 y - k_2 y^2 ,$$

en la que los parámetros aparecen linealmente. Con 3 nodos y 40 puntos muestrales obtuvimos los resultados que aparecen en la siguiente tabla

Nodos	Residual en la ec. dif.	parámetros	Mejor cond. inic.
25,100,140	3.729	0.04608, 0.00009570	1.5

Tabla 16

En la siguiente figura, en la que aparece la curva integral de la ecuación diferencial con los parámetros 0.04608, 0.00009570, muestra el excelente resultado obtenido.

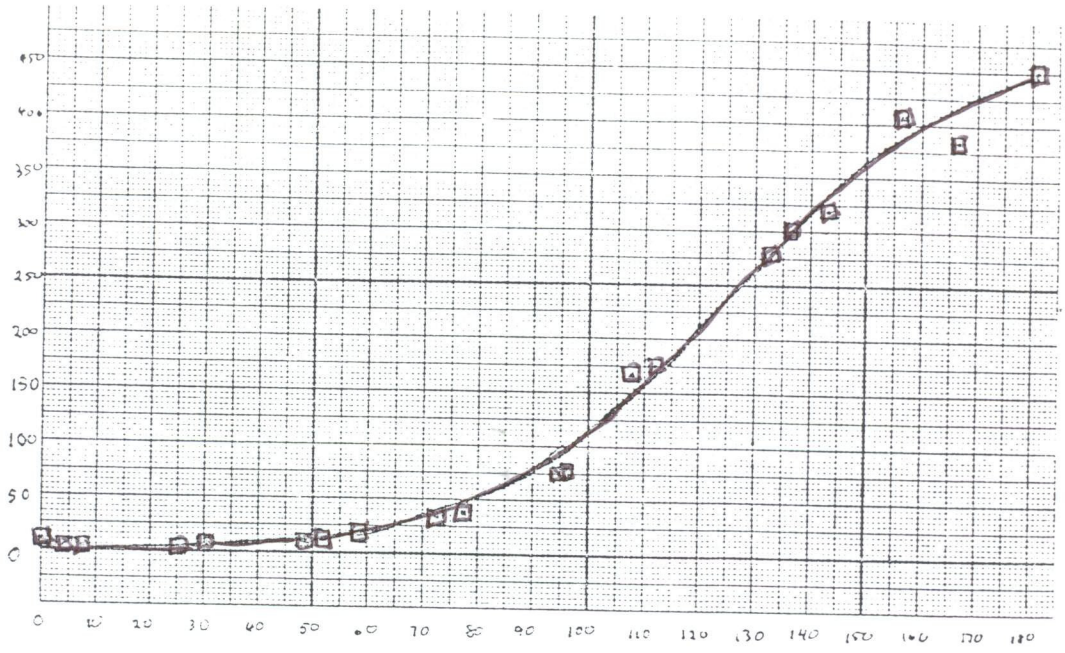

450
400
350
300
250
200
150
100
50
0
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180

Figura 22

CAPITULO VI

P R O G R A M A S

```
C*    PROGRAMA SPLINE/CUBICO/PROBS.
C*    CONSTRUYE EL SPLINE QUE AJUSTA M PUNTOS.
C*
      DIMENSION A(100,100),T(100),B(100),F(100),XSI(30)
       DIMENSION GRAD(20),APAR(100,20)
         INTEGER II(20)
      MDIM=100
C*    M = # DE OBSERVACIONES
C*      = # DE PUNTOS DATO (T(),F()), DONDE
C*    T() ES UN VECTOR DE DIMENSION M QUE CONTIENE LAS ABSCISAS DATO
C*    F() ES UN VECTOR DE DIMENSION M QUE CONTIENE LAS ORDENADAS DATO.
C*
C*    NIXSI = # DE NODOS INTERIORES LIBRES. ESTE VALOR DEBE SER
C*                PROPORCIONADO POR EL USUARIO.
C*    NXSI = # DE NODOS = NIXSI+2. A LOS ANTERIORES SE AGREGAN
C*          LOS NODOS EXTREMOS.
C*    LOS NODOS INTERIORES,ASI COMO LOS NODOS EXTREMOS, DEBEN SER
C*    PROPORCIONADOS POR EL USUARIO.
C*    N = DIMENSION DEL ESPACIO DE SPLINES CUBICOS
C*      = NXSI+2 = NIXSI+4.
C*
C*    NTXSI = # TOTAL DE NODOS NECESARIOS PARA GENERAR LOS
C*            B-SPLINES = NXSI+2*3. RESULTAN DE AGREGAR TRES NO-
C*            DOS ARBITRARIOS A LA IZQUIERDA DEL NODO EXTREMO IZ-
C*            QUIERDO Y TRES NODOS ARBITRARIOS A LA DERECHA DEL
C*            NODO EXTREMO DERECHO.
C*
      DIMENSION XSIO(20),X(5),BS(4)
C*
C*    LEE M, NIXSI, LOS POSTES Y  LOS NODOS INTERIORES:
                        WRITE(6,10)
 10                     FORMAT(/5X,'M=')
                        READ(5,/)M
C*
                        WRITE(6,20)
 20                     FORMAT(/5X,'# DE NODOS INTERIORES=')
                        READ(5,/)NIXSI
                       WRITE(6,30)
 30                   FORMAT(/5X,'LOS NODOS INTERIORES+LOS DOS POSTES:')
                        DO 40 I=1,NIXSI+2
 40                     READ(5,/)XSIO(I)
C*
      NXSI=NIXSI+2
      N=NXSI+2
      NTXSI=NXSI+6
C*
C*    SE AGREGAN LOS NODOS AUXILIARES:
C*
      DO 50 I=1,NXSI
 50   XSI(I+3)=XSIO(I)
      XSI(1)=XSIO(1)-3.
      XSI(2)=XSIO(1)-2.
      XSI(3)=XSIO(1)-1.
      XSI(NXSI+4)=XSIO(NXSI)+1.
      XSI(NXSI+5)=XSIO(NXSI)+2.
      XSI(NXSI+6)=XSIO(NXSI)+3.
C*
C*    SE LLAMA A LA SUBRUTINA PARA OBTENER
C*    LOS DATOS.
C*
      CALL BARNES(T,F,M)
C*
        WRITE(6,65)
   65   FORMAT(/10X,'SI SE DESEA QUE SE ESCRIBA LA INFORMACION'/
       *10X,'RECIBIDA, HAGASE DATOS=1.EN CASO CONTRARIO, DATOS=0.')
        WRITE(6,70)
   70   FORMAT(/10X,'DATOS=')
        READ(5,/)DATOS
        IF(DATOS.EQ.0.)GO TO 170
        WRITE(6,80)M
   80   FORMAT(/10X,'M = ',I4,/)
        WRITE(6,90)
   90   FORMAT(/,13X,'LAS ABSCISAS', 17X,'LAS ORDENADAS',/)
        DO 100 I=1,M
  100   WRITE(6,110)T(I),F(I)
  110   FORMAT(7X,F12.5,12X,F12.5)
        WRITE(6,120)
  120   FORMAT(//,6X,'TODOS LOS NODOS',/)
        WRITE(6,130)NTXSI
  130   FORMAT(22X,'# TOTAL DE NODOS=',I3,/)
        DO 140 I=1,NTXSI
  140   WRITE(6,150)I,XSI(I)
```

```
150   FORMAT(9X,'XSI(',I2,')=',F13.7)
      WRITE(6,160)N
160   FORMAT(//,3X,'DIMENSION DEL ESPACIO DE SPLINES=',I3,/)
170   CONTINUE
C*
C*    SE GUARDA EL VECTOR DE ORDENADAS F() EN EL VECTOR B():
                        DO 180 I=1,M
180                              B(I)=F(I)
C*
C*    LLAMA A AMATR QUE CONSTRUYE LA MATRIZ A:
C*
      CALL AMATR(MDIM,M,N,A,XSI,T,II)
C*
      DO 190 J=1,N
        DO 200 I=1,M
           APAR(I,J)=A(I,J)
 200          CONTINUE
 190    CONTINUE
      CALL GIVENS(MDIM,M,N,APAR,B,II)
C*
C*
      CALL SOLVE(MDIM,N,APAR,B)
C*    EL VECTOR ALFA* HA QUEDADO ALMACENADO EN LAS
C*    PRIMERAS N COMPONENTES DEL VECTOR B() .
C*
      WRITE(6,202)
 202  FORMAT(//,8X,'EL VECTOR ALFA*',//)
      DO 203 I=1,N
 203  WRITE(6,204)B(I)
 204  FORMAT(9X,E12.5,/)
C*
      H=(T(M)-T(1))/200.
      I=1
      WRITE(6,210)
 210  FORMAT(//,5X,'T',18X,'EL SPLINE')
 302  TT=T(1)+FLOAT(I-1)*H
      DO 303 MM=4,N
      IF(XSI(MM) .LE. TT .AND. TT .LT. XSI(MM+1))GO TO 310
 303  CONTINUE
 310  CONTINUE
C*
       DO 320 IN=1,4
       IB=4-IN
       CALL MOVI(MM-IB,XSI,X)
       BS(IN)=BASICO(X,TT)
  320  CONTINUE
       AA=B(MM-3)*BS(1)+B(MM-2)*BS(2)+B(MM-1)*BS(3)+B(MM)*BS(4)
       WRITE(6,311)TT,AA
  311  FORMAT(2X,F11.4,4X,F11.4)
  350  I=I+1
       IF(I .LT. 202)GO TO 302
       WRITE(6,250)
  250  FORMAT(2X)
 C*
       STOP
       END
 C*
 C*      * * * * * * * * * * * * * * * * * *
 C*
       SUBROUTINE MOVI(I,XSI,X)
 C*
 C*    SI SE VA A EVALUAR EN T EL SPLINE BASICO B(I) CON
 C*    NODOS XSI(I),...,XSI(I+4), ESTA SUBRUTINA HACE EL
 C*    MOVIMIENTO DE INDICES X(1):=XSI(I),...,X(5):=XSI(I+4)
 C*    CON EL FIN DE LLAMAR A BASICO.
 C*
       DIMENSION XSI(1), X(1)
       DO 10 K=1,5
       X(K)=XSI(I+K-1)
  10   CONTINUE
       RETURN
       END
 C*
 C*    * * * * * * * * * * * * * * *
 C*
       REAL FUNCTION BASICO(X,T)
       DIMENSION X(1),G(5)
 C*
 C*    EVALUA EL B-SPLINE CUBICO CON NODOS X(1),...,X(5), DISTINTOS,
 C*    EN EL PUNTO T PERTENECIENTE AL INTERVALO, ES DECIR,
 C*    X(1) .LE. T .LE. X(5).
 C*                              ESTA EVALUACION ES EL VALOR DE
 C*     (X(5)-X(1))*[X(1),...X(5)](X-T)+**3, HACIENDO DIRECTAMENTE
 C*    EL ARREGLO TRIANGULAR DE DIFERENCIAS DIVIDIDAS.
 C*
```

```
C*
      IF(T .LE. X(1)) GO TO 810
      IF(T .GT. X(5)) GO TO 820
      DO 10 I=1,5
 10   G(I)=0.
C*
      IF(T .LE. X(2))GO TO 200
      IF(T .LE. X(3))GO TO 300
      IF(T .LE. X(4))GO TO 400
      IF(T .LE. X(5))GO TO 500
C*
 200  DO 210 I=2,5
 210  G(I)=CUBO(X(I),T)
      GO TO 600
 300  DO 310 I=3,5
 310  G(I)=CUBO(X(I),T)
      GO TO 600
 400  DO 410 I=4,5
 410  G(I)=CUBO(X(I),T)
      GO TO 600
 500  G(5)=CUBO(X(5),T)
C*
C*    CALCULO DE LAS DIFERENCIAS DIVIDIDAS.
C*
 600  CONTINUE
      DO 700 J=1,4
      DO 710 I=1,5-J
        G(I)=(G(I)-G(I+1))/(X(I)-X(I+J))
 710  CONTINUE
 700   CONTINUE
C*
      BASICO=(X(5)-X(1))*G(1)
      GO TO 800
 810  BASICO=0.
      GO TO 800
 820  BASICO=0.
 800  RETURN
      END
C*
C*    * * * * * * * * ** * * * * * * * * * *
C*
      SUBROUTINE AMATR(MDIM,M,N,A,XSI,T,II)
C*
C*    CONSTRUYE LA MATRIZ A(J,I)=B-SPLINE(I)(T(J))
C*
      DIMENSION A(MDIM,1),XSI(1),T(1)
      INTEGER  II(1)
      DIMENSION X(6)
      DO 10 I=1,N
          DO 20 J=1,M
          CALL MOVI(I,XSI,X)
          A(J,I)=BASICO(X,T(J))
 20       CONTINUE
 10   CONTINUE
C*
      DO 30 I=1,N-3
          DO 40 J=1,M
          IF(T(J).LT.XSI(I+4))II(I)=J
          IF(T(J).GE.XSI(I+4))GO TO 30
 40       CONTINUE
 30   CONTINUE
      II(N-2)=M
      II(N-1)=M
      II(N)=M
      RETURN
      END
C*
C*     * * * * * * * * * * * * * * * * * * *
C*
      SUBROUTINE GIVENS(MDIM,M,N,A,B,II)
      DIMENSION A(MDIM,1),B(1)
      INTEGER II(1)
      DIMENSION VAUXI(20),VAUXJ(20)
      DO 1 J=1,N
        DO 2 I=J+1,II(J)
        IF(ABS(A(I,J)).GT.0.)GO TO 3
        C=1.
        S=0.
        GO TO 5
C*
 3      IF(ABS(A(I,J)).LE.ABS(A(J,J)))GO TO 4
        TE=A(J,J)/A(I,J)
        S=1./SQRT(1.+TE*TE)
        C=TE*S
        GO TO 5
```

```
C*
 4      TE=A(I,J)/A(J,J)
        C=1./SQRT(1.+TE*TE)
        S=TE*C
C*
 5      DO 8 K=J,N
        VAUXJ(K)=A(J,K)
        VAUXI(K)=A(I,K)
 8      CONTINUE
        A(J,J)=C*VAUXJ(J)+S*VAUXI(J)
        BT=C*B(J)+S*B(I)
        BS=-S*B(J)+C*B(I)
         B(J)=BT
         B(I)=BS
       IF(J.EQ.N)GO TO 2
       DO 6 JJ=J,N
 6     A(J,JJ)=C*VAUXJ(JJ)+S*VAUXI(JJ)
       IF(J.EQ.N)GO TO 2
       DO 7 JJ=J,N
 7     A(I,JJ)=-S*VAUXJ(JJ)+C*VAUXI(JJ)
 2       CONTINUE
 1     CONTINUE
       RETURN
        END
C*
C*     * * * * * * * * * * * * * * * * * * * * * * *
C*
       SUBROUTINE SOLVE(MDIM,N,A,B)
C*
C*     RESUELVE EL SISTEMA  R*X=B,  DONDE  R  ES UNA MATRIZ  N*N
C*     CON ELEMENTOS NO CERO SOBRE LA DIAGONAL PRINCIPAL Y LAS
C*     TRES SOBREDIAGONALES INMEDIATAS.
C*
       DIMENSION A(MDIM,1), B(1)
                            DO 10 I=1,N
                            IF(A(I,I).EQ.0.)GO TO 5
  10                        CONTINUE
       B(N)=B(N)/A(N,N)
       B(N-1)=(B(N-1)-A(N-1,N)*B(N))/A(N-1,N-1)
       B(N-2)=(B(N-2)-A(N-2,N-1)*B(N-1)-A(N-2,N)*B(N))/
      *                                        A(N-2,N-2)
       IF(N-2.EQ.1)GO TO 2
       DO 1 J=1,N-3
       I=N-2-J
       B(I)=(B(I)-A(I,I+1)*B(I+1)-A(I,I+2)*B(I+2)-A(I,I+3)*
      *                                        B(I+3))/A(I,I)
 1     CONTINUE
 2     RETURN
 5                          WRITE(6,6)
 6                  FORMAT(//8X,'ELEMENTO CERO EN LA DIAGONAL',///)
            STOP
       END
C*
C*     * * * * * * * * * * * * * * * * * * * * * * * *
C*
       REAL FUNCTION CUBO(AX,AT)
       REAL AX,AT,D
       D=AX-AT
       CUBO=D*D*D
       RETURN
       END
C*
C*     * * * * * * * * * * * * * * * * * * * * * * * *
       SUBROUTINE TITAN(T,FY,M)
 C*    ESTOS DATOS REPRESENTAN UNA PROPIEDAD DEL TITANIO
 C*    COMO UNA FUNCION DE LA TEMPERATURA. HAN SIDO MUY USADOS
 C*    COMO UN BUEN EJEMPLO PARA APROXIMACION SPLINE CON NODOS LIBRES.
 C*
       INTEGER M,I
       REAL FY(1),T(1),GTITAN(49)
       DATA GTITAN /.644,.622,.638,.649,.652,.639,.646,.657,.652
      *      ,.655,.644,.663,.663,.668,.676,.676,.686,.679,.678
      *      ,.683,.694,.699,.710,.730,.763,.812,.907,1.044,1.336,
      *      1.581,2.169,2.075,1.598,1.211,.916,.746,.672,.627,
      *      .615,.607,.606,.609,.603,.601,.603,.601,.611,.601,
      *      .608/
       DO 10 I=1,M
           T(I)=585.+10.*FLOAT(I)
  10   FY(I)=GTITAN(I)
                            RETURN
       END
C*
C*     * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
      SUBROUTINE BARNES(T,FY,M)
C*
C*    NPROB=4, M=11, POSTE IZQ.=-0.1, POSTE DER.=5.5
C*
C*    REF.  VARAH.
C*
      INTEGER M,I
      REAL FY(1),T(1),VFY(11)
      DATA VFY/1.0,1.1,1.3,1.1,0.9,0.7,0.5,0.6,0.7,0.8,1.0/
      DO 10 I=1,M
       T(I)=0.5*DFLOAT(I-1)
 10    FY(I)=VFY(I)
                  RETURN
      END
C*
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C*
      SUBROUTINE BARNE2(T,FY,M)
C*
C*    NPROB=5, M=11, POSTE IZQ.=-0.1, POSTE DER.=5.5
C*
C*    REF.  VARAH.
C*
      INTEGER M,I
      REAL FY(1),T(1),VFY(11)
      DATA VFY/0.3,0.35,0.4,0.5,0.5,0.4,0.3,0.25,0.25,0.3,0.35/
      DO 10 I=1,M
       T(I)=0.5*DFLOAT(I-1)
 10    FY(I)=VFY(I)
                  RETURN
      END
C*
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C*
      SUBROUTINE BELLMAN(T,FY,M)
C*
C*    NPROB=5, M=15, POSTE IZQ.=1, POSTE DER.=40
C*
C*    REF. VARAH.
C*
      INTEGER M,I
      REAL FY(1),T(1),VT(15),VFY(15)
      DATA VT /1,2,3,4,5,6,7,8,10,12,15,20,25,30,40 /
      DATA VFY /0.0,1.4,6.3,10.4,14.2,17.6,21.4,23.0,27.0,
     *          30.4,34.4,38.8,41.6,43.5,45.3 /
      DO 10 I=1,M
         T(I)=VT(I)
 10      FY(I)=VFY(I)
                  RETURN
      END
C*
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C*
      SUBROUTINE ENZIMA(T,FY,M)
C*
C*    NPROB=7, M=28, POSTE IZQ.=0, POSTE DER.=187
C*
C*    REF. VARAH.
C*
      INTEGER M, I
      REAL FY(1),T(1),VT(28),VFY(28)
      DATA VT /0.1,2.5,3.8,7.0,10.9,15.0,18.2,21.3,22.9,24.9,
     *         26.8,30.1,34.1,37.8,42.4,44.4,47.9,53.1,59.0,65.1,
     *         73.1,81.1,91.2,101.9,115.4,133.7,163.2,186.7 /
      DATA VFY/27.8,20.0,23.5,63.6,267.5,427.8,339.7,331.9,243.5,
     *         212.0,164.1,112.7,88.1,76.2,62.3,58.7,41.9,40.2,
     *         31.3,30.0,30.6,23.5,24.8,26.1,33.3,17.8,16.8,16.8 /
      DO 10 I=1,M
         T(I)=VT(I)
 10      FY(I)=VFY(I)
                  RETURN
      END
C*
```

```
      SUBROUTINE SOBI(I,J,XSI,X,NP)
C*
C*    SI SE VA A EVALUAR EN T EL SPLINE B(I) CON NODOS XSI(I),...
C*    XSI(I+4), O SI DERIVADA CON RESPECTO AL NODO XSI(J), ESTA
C*    SUBRUTINA HACE EL CORRIMIENTO DEL INDICE X(1)I=XSI(I),...
C*    X(5):=XSI(I+4), CON EL FIN DE LLAMAR A BASICO(X,T,G) O A DE-
C*    RBASI(NP,X,T,G). DONDE NP ES EL CORRIMIENTO DEL INDICE J CO-
C*    RRESPONDIENTE AL CORRIMIENTO REALIZADO DEL INDICE I.
C*    .
      DOUBLE PRECISION XSI(1), X(1)
      DO 10 K=1,5
      X(K)=XSI(I+K-1)
 10   CONTINUE
      NP=J-I+1
      RETURN
      END
C*
C*    ***********************************
      REAL FUNCTION DERBI(NP,X,T)
      DOUBLE PRECISION X(1), G(5), XT(6),T
C*
C*    EVALUA, EN T, LA DERIVADA PARCIAL DEL B-SPLINE CUBICO
C*    CON RESPECTO AL NODO XSI(NP), HACIENDO RESPECTIVAMENTE EL
C*    ARREGLO TRIANGULAR DE DIFERENCIAS DIVIDIDAS.
C*
            IF(NP .GT. 5)GO TO 320
            IF(NP .LT. 1)GO TO 320
                        DO 5 I=1,5
                         XT(I)=X(I)
 5                            CONTINUE
C*
      IF(T .LE. XT(1))GO TO 320
      IF(T .GT. XT(5))GO TO 320
C*
      DO 10 I=1,5
 10   G(I)=0.
C*
      IF(T .LE. XT(2))GO TO 200
      IF(T .LE. XT(3))GO TO 300
      IF(T .LE. XT(4))GO TO 400
      IF(T .LE. XT(5))GO TO 500
C*
 200  DO 210 I=2,5
 210  G(I)=CUBO(XT(I),T)
      GO TO 600
 300  DO 310 I=3,5
 310  G(I)=CUBO(XT(I),T)
      GO TO 600
 400  DO 410 I=4,5
 410  G(I)=CUBO(XT(I),T)
      GO TO 600
 500  G(5)=CUBO(XT(5),T)
C*
```

```
  600  CONTINUE
C*
      DO 610 I=1,4
  610 G(I)=(G(I)-G(I+1))/(XT(I)-XT(I+1))
C*    -
      DO 620 I=NP+1,5
      II=6+NP-I
  620 G(II)=G(II-1)
      IF(XT(NP)-T.GT.0.) G(NP)=G.*(XT(NP)-T)*(XT(NP)-T)
      IF(XT(NP)-T.GT.0.) GO TO 635
      G(NP)=0.
  635  CONTINUE
C*
      DO 640 I=NP+1,5
      II=7+NP-I
  640 XT(II)=XT(II-1)
C*
      DO 700 J=2,5
      DO 710 I=1,6-J
       G(I)=(G(I)-G(I+1))/(XT(I)-XT(I+J))
  710 CONTINUE
                    IF(NP.EQ.1.AND.J.EQ.4)GO TO 720
                    IF(NP.EQ.5.AND.J.EQ.4)GO TO 730
  700  CONTINUE
C*
      DERIB=(XT(6)-XT(1))*G(1)
      GO TO 307
  720 DERIB=G(1)
      GO TO 300
  730 DERIB=G(2)
C*
      GO TO 300
  320 DERIB=0.
C*
  300 RETURN
      END
C*
C*    * * * * * * * * * * * * * * * * * * * * * * * *
C*
C*     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE BLOOD(T,FY,M)
C*
C*      NPROB=8, M=21, POSTE IZQ.=2, POSTE DER.=180
C*
C*      REF. R.I.JENNRICH AND P.B.BRIGHT
C*
       INTEGER M,I
       REAL FY(1),T(1),VT(21),VFY(21)
      DATA VT/2,4,6,8,10,15,20,25,30,40,50,60,70,80,90,110,130,
     *        150,160,170,180 /
      DATA VFY /151117,113601,97652,90935,84829,76891,73342,70593,
     *          67041,64313,61554,59940,57698,56440,53915,50938,
     *          48717,45996,44968,43602,42668 /
      DO 10 I=1,M
         T(I)=VT(I)
10       FY(I)=VFY(I)
                    RETURN
                           END
```

```
      SUBROUTINE SIGMA(N,X,SIGMA)
C     CONSTRUYE LA NUEVA VARIABLE VECTOR SIGMA( ) A PARTIR DE
C     LOS NODOS INTERIORES X( ).
C
      DOUBLE PRECISION X(1),SIGMA(1),H(40),XSIG(1)
      INTEGER N,NIXSI,I
C
C     NIXSI ES EL NUMERO DE NODOS INTERIORES.
C
      COMMON /INT/ NIXSI
      COMMON /OLVER1/ XSIG,XSI,T,FY
C
C     EL PRIMER NODO LIBRE ES X(NIXSI+1). EL POSTE IZQUIERDO ES XSIG(1).
C     POR COMODIDAD HACEMOS COINCIDIR EL INDICE DE H( ) CON EL DE X( ).
C     EL POSTE DERECHO ES XSIG(NIXSI+1).
C
      H(NIXSI+1)=X(NIXSI+1)-XSIG(1)
      DO 10 I=2,NIXSI
      H(NIXSI+I)=X(NIXSI+4+I)-X(NIXSI+I-1)
   10 CONTINUE
      H(2+NIXSI+1)=XSIG(N-NIXSI+1)-X(2+NIXSI+4)
      DO 20 I=1,NIXSI
      SIGMA(NIXSI+4+I)=DLOG(H(NIXSI+1+I)/H(NIXSI+4+I))
   20 CONTINUE
C
      DO 30 I=1,NIXSI+4
      SIGMA(I)=X(I)
   30 CONTINUE
C
      RETURN
      END
C
C     * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE SIGMAX(N,SIGMA,X)
C
C     A PARTIR DEL VECTOR VARIABLE SIGMA( ) SE CONSTRUYE EL VECTOR X( )
C     QUE CONTIENE A LOS NODOS LIBRES.
C
      DOUBLE PRECISION SIGMA(1),X(1),H(40),P(40),XSIG(1),T
      INTEGER N,NIXSI,I,J
C
C     NIXSI ES EL NUMERO DE NODOS INTERIORES.
      COMMON /INT/ NIXSI
      COMMON /OLVER2/ XSIG,XSI,T,FY
C
      DO 10 I=1,NIXSI
      P(I)=DEXP(SIGMA(NIXSI+4+I))
   10 CONTINUE
C
C     EVALUACION DEL POLINOMIO Z UTILIZANDO HORNER:
C
```

```
      Z=1.+P(NIXSI)
      DO 30 J=1,NIXSI-1
      I=NIXSI-J
      Z=1.+P(I)*Z
   30 CONTINUE
C
C     LOS INCREMENTOS H(I)=X(I)-X(I-1), I=1,...,NIXSI+1; SIN
C     EMBARGO, POR CONSIGUIENTE SE CORREN LOS INDICES AGREGANDO NIXSI+4,
C     YA QUE EL PRIMER NODO INTERIOR ES X(NIXSI+5).
C
      H(NIXSI+3)=DSIGN(NIXSI+3)-XSIG(1))/Z
      DO 31 I=2,NIXSI+1
      H(NIXSI+4+I)=P(I-1)*H(NIXSI+3+I)
   31 CONTINUE
C
C     LOS NODOS LIBRES
C
      X(NIXSI+5)=H(NIXSI+5)+XSIG(1)
      DO 40 I=2,NIXSI
      X(NIXSI+4+I)=H(NIXSI+4+I)+X(NIXSI+3+I)
   40 CONTINUE
C
      DO 50 I=1,NIXSI+4
      X(I)=SIGMA(I)
   50 CONTINUE
      RETURN
      END
C     * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE SYSJAC(M,N,FJAC,SJAC,LDFJAC,Y)
C
C     EST. SUBRUTINA, PUES EL ALGORITMO ES BASE PARA RESOLVER EL SISTEMA
C                 G*J(SIGMA)(TRANSP)=J(Y)(TRANSP),
C     DONDE
C
C                 G=3*H(NV)*B(TRANSP)
C                 H=DIAG(H( ),...,H( ) )
C                H(1)=X(I)-Y(I-1)
C     REF.: POSE, "AN ALGORITHM FOR SOLVING A SPECIAL CLASS OF
C                 TRIDIAGONAL SYSTEMS."
C
      DOUBLE PRECISION X(N),FJAC(LDFJAC,N),SJAC(LDFJAC,N),H(40),
     *       TET(40),V(40),XSIG(1),S
      INTEGER M,N,LDFJAC,NIXSI,I,J,JJ
      COMMON /SOLVER/ XSIG,XSI,I,FY
      COMMON /INIT/ NIXSI
C
C     LOS INCREMENTOS H(I)=X(I)-X(I-1),I=1,...,NIXSI+1; SIN
C     EMBARGO, POR COMODIDAD SE CORREN LOS INDICES AGREGANDO NIXSI+4
C     YA QUE EL PRIMER NODO INTERIOR ES X(NIXSI+5)
C
```

```
      H(NIXSI+1)=W(NIXSI+5)-XSI**(1)
      DO 10 I=1,NIXSI
      H(NIXSI+4+I)=W(NIXSI+4+I)-X(NIXSI+1+I)
   10 CONTINUE
      H(2*NIXSI+1)=XSI**(NIXSI+1)-X(1+NIXSI+4)
C
C
      TETA(1)=H(NIXSI+1)
      DO 20 J=2,NIXSI+1
      TETA(J)=TETA(J-1)+H(NIXSI+4+J)
   20 CONTINUE
C
      DO 70 I=1,N
           S=TETA(1)*FJAC(I,NIXSI+1)
           DO 30 J=2,NIXSI
           S=S+TETA(J)*FJAC(I,NIXSI+4+J)
   30      CONTINUE
           S=-S/TETA(NIXSI+1)
C
           V(N)=FJAC(I,N)+S
           DO 40 JJ=1,NIXSI-1
           J=NIXSI-JJ
           V(NIXSI+4+J)=FJAC(I,NIXSI+4+J)+V(NIXSI+5+J)
   40      CONTINUE
C
           SJAC(I,NIXSI+1)=H(NIXSI+5)*V(NIXSI+5)
           DO 50 J=2,NIXSI
           SJAC(I,NIXSI+4+J)=H(NIXSI+4+J)*V(NIXSI+4+J)+
     *                       SJAC(I,NIXSI+1+J)
   50      CONTINUE
C
           DO 60 J=1,NIXSI
           SJAC(I,NIXSI+4+J)=-SJAC(I,NIXSI+4+J)
   60      CONTINUE
   70 CONTINUE
C
      DO 80 I=1,N
      DO 80 J=NIXSI+1,N
      FJAC(I,J)=SJAC(I,J)
   80 CONTINUE
C
      RETURN
      END
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * *


      INTEGER FUNCTION KLOCAT(XSI,T,N,)
C     DADOS LOS NODOS XSI(1),XSI(2),...,XSI(N+1) Y EL PUNTO T
C     ESTA FUNCION ENCUENTRA K TAL QUE EL PUNTO T SE LOCALIZA
C     EN EL INTERVALO (XSI(K),XSI(K+1) ), CON K MAYOR O IGUAL A
C     1 Y MENOR QUE (N+1).
C
```

```
      DOUBLE PRECISION T
      DOUBLE PRECISION XSI(4).
      INTEGER N,K
      IF(T .LT. XSI(4))WRITE(6,90)
      IF(T .GT. XSI(N+1))WRITE(6,90)
      IF(T .LT. XSI(4))GOTO 95
      IF(T .GT.XSI(N+1))GOTO 95
      DO 10 J = 4,N

         IF(T .LT. XSI(J+1))GO TO 20
   10 CONTINUE
   20 KLOCAT = J
C
         IF(T .EQ. XSI(N+1))KLOCAT=N
C
   90 FORMAT(5X,"T ESTA FUERA DEL INTERVALO",///)
   95 RETURN
      END
C
C      * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE PUNTOS(TM,LM)
      DOUBLE PRECISION AS(20),TM(1)
      INTEGER NPS(20),NS,LM,I,J,KI
C
C     NS   ES EL NUMERO DE SUBINTERVALOS EN EL QUE SE DESEA SUBDIVI-
C          DIR TODO EL INTERVALO.
C     AS   ES UN ARREGLO QUE CONTIENE LOS EXTREMOS DE TALES SUBINTER-
C          VALOS.
C     NPS  ES UN ARREGLO TAL QUE NPS(I) ES EL NUMERO DE PUNTOS EQUI-
C          DISTANTES EN EL INTERIOR DEL I-ESIMO SUBINTERVALO.
C     LM   ES EL NUMERO TOTAL DE PUNTOS MUESTRALES.
C     TM   ES UN ARREGLO QUE CONTIENE A LOS PUNTOS MUESTRALES.
C
C     SE DETERMINAN LOS PUNTOS MUESTRALES:
C
C     PRIMERO SE LEE EL NUMERO DE SUBINTERVALOS NS EN QUE SE DESEE
C     DIVIDIR EL INTERVALO TOTAL, ES DECIR EL INTERVALO [T(1),T(N)].
C
      WRITE(6,10)
   10 FORMAT(5X,"NUMERO DE SUBINTERVALOS, NS=")
      READ(5,*)NS
C
C     SE ASIGNAN EN EL ARREGLO  AS  LOS EXTREMOS DE TALES SUBINTER-
C     VALOS.
      WRITE(6,15)
   15 FORMAT(/5X,"LOS EXTREMOS DE LOS SUBINTERVALOS:",/)
      DO 20 I=1,NS+1
         WRITE(6,30)I
         READ(5,*)AS(I)
   20 CONTINUE
   30 FORMAT(/5X,"AS(",I2,")=")
C
```

```
C     SE ASIGNA EN EL ARREGLO NPS(I) EL NUMERO DE PUNTOS EQUIDISTAN-
C     TES EN EL INTERIOR DE CADA SUBINTERVALO.
C
      DO 40 I=1,NS
        WRITE(6,45)I
        READ(5,*)NPS(I)
 40   CONTINUE
 45   FORMAT(//5X,"NUM. DE PTOS MUESTRALES DENTRO DEL INTERVALO",
     1 I4,": ")
C
      NN = 1
      DO 50 I=1,NS
        NN = NN + NPS(I) + 1
 50   CONTINUE
C
      WRITE(6,55)NN
 55   FORMAT(///5X,"NUMERO DE PUNTOS MUESTRALES=",I4,/)
C
C     SE CONSTRUYEN LOS PUNTOS MUESTRALES DENTRO DE CADA SUBINTERVALO
C     Y EN EL ARREGLO TH QUEDAN ALMACENADOS TODOS LOS PUNTOS MUES-
C     TRALES.
C
      TH(1)=AS(1)
      K=1
      DO 70 I=1,NS
        H=(AS(I+1)-AS(I))/(FLOAT(NPS(I)+1.))
        DO 60 KI=1,NPS(I)
        TH(K+KI)=TH(K)+FLOAT(KI)*H
 60     CONTINUE
        K=K+NPS(I)+1
        TH(K)=AS(I+1)
 70   CONTINUE
      RETURN
      END
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      FUNCTION EVASPL(XSI,F,T,N)
      DOUBLE PRECISION XSI(11),F(11),T(10),A(10),BS(10)
      DOUBLE PRECISION T
      INTEGER K
      DO 10 IN=1,4
        IE=4*IN
        CALL MOVI(K-IE,XSI,X)
        BS(IN)=BASICO(X,T)
 10   CONTINUE
C
      EVASPL=F(K-3)*BS(1)+(H-3)*BS(2)+F(K-2)*BS(3)+F(K-1)*BS(4)
C
      RETURN
      END
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      FUNCTION DERIVA(XSI,ALFA,I,T)
      DOUBLE PRECISION XSI(11),ALFA(1)
      DOUBLE PRECISION T
      INTEGER K
C     ENCUENTRA LA DERIVADA DE UN SPLINE CUBICO EVALUADA EN
```

```
C      PUNTO T QUE SE SABE QUE ESTA EN EL INTERVALO (XSI(K),
C      XSI(K+1)), CON K ENTRE 4 Y N.
       D1 = 3.*(ALFA(K-2)-ALFA(K-3))/(XSI(K+1)-XSI(K-3))
       D2 = 3.*(ALFA(K-1)-ALFA(K-2))/(XSI(K+2)-XSI(K-2))
       D3 = 3.*(ALFA(K)  -ALFA(K-1))/(XSI(K+3)-XSI(K-1))
       A1 = (XSI(K+1)-T)*(XSI(K+1)-T)
       B1 = D1/((XSI(K+1)-XSI(K-1))*(XSI(K+1)-XSI(K)))
       AA1 = (T-XSI(K-1))*(XSI(K+1)-T)
       AB1 = (XSI(K+1)-XSI(K-1))*(XSI(K+1)-XSI(K))
       AA2 = (XSI(K+2)-T)*(T-XSI(K))
       AB2 = (XSI(K+2)-XSI(K))*(XSI(K+1)-XSI(K))
       A = AA1/AB1
       B = AA2/AB2
       C = A+B
       D=(T-XSI(K))
       B=C*D
       B3=B3/((XSI(K+1)-XSI(K))*(XSI(K+2)-XSI(K)))
       DERIVA=B1*B1+C2*B2+D3*B3
           RETURN
           END
C
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       REAL FUNCTION DERSEG(XSI,ALFA,I,T)
C
C      DADOS LOS NODOS XSI(1),...,XSI(N+4) Y LOS COEFICIENTES
C      ALFA(1),...,ALFA(N), QUE DEFINEN UN SPLINE CUBICO, Y
C      DADO I TAL QUE 4 .LE. I .LE. N
C               XSI(I) .LE. T .LT. XSI(I+1)
C      ESTA SUBRUTINA CALCULA LA SEGUNDA DERIVADA DEL SPLINE
C      EVALUADA EN T.
C
       REAL XSI(1), ALFA(1), T
       INTEGER I
C
       A1=(ALFA(I-1)-ALFA(I-2))
       A2=ALFA(I-2) - ALFA(I-3)
       A3=ALFA(I) - ALFA(I-1)
       A4=(ALFA(I-2) - ALFA(I-2)
       B1=(XSI(I+2)-XSI(I-1))*(XSI(I+1)-XSI(I-1))
       B2=(XSI(I+1)-XSI(I-2))*(XSI(I+1)-XSI(I-1))
       B3=(XSI(I+2)-XSI(I))*(XSI(I+2)-XSI(I))
       B4=(XSI(I+2)-XSI(I-1))*(XSI(I+2)-XSI(I))
C
       A=A2/B1
       B=A3/B2
       C=A3/B3
       D=A4/B4
       E=(XSI(I+2)-T)/(XSI(I+1)-XSI(I))
       F=(T-XSI(I))/(XSI(I+2)-XSI(I))
C
       DERSEG=2.*(A-B)+6.*(C-D)*F
C
       RETURN
       END
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

B I B L I O G R A F I A

[1] Bard, Y., (1974). Nonlinear parameter estimation. Academic Press, New York.

[2] Bellman et al, (1967), "Quasilinearization and the estimation of chemical rate constants from raw kenetic data". Math. Biosc. 1, pp. 71-76.

[3] Benson, M., (1979). "Parameter fiting in dynamic models". Ecol. Mod. 6. pp. 97-115.

[4] De Boor, C., (1978). A practical guide to splines. Springer-Verlag.

[5] van Domselaar y Hemker, (1975). "Nonlinear parameter estimation in initial value problems". Report NW 18/75, Mathematical Centrum, Amsterdam.

[6] Fletcher, R., (1980). Practical methods of optimization. v.1. John Wiley.

[7] Gear, G.W., (1971). Numerical initial value problems in ordinary differential equations. Prentice-Hall. New Jersey.

[8] Golub, G.H. y Pereyra, V., (1973). "The differentiation of pseudoinverses and nonlinear least squares problems whose variables separate". J. Siam. Num. Anal. 10, pp. 413-432.

[9] Greville, T.W.E., Spline functions and applications. Mathematics Research Center, United States Army. The University of Wisconsin. Orientation Lecture Series No. 8

[10] Jennrich, R.I. y Bright, P.B., (1974). "Fitting systems of linear differential equations using computer generated exact derivatives". Technical Report No. 10, Health Sciences Computing Facility. University of California, Los Angeles.

[11] Jupp, D.L.B., (1975). "The 'lethargy' theorem-a property of approximation by γ-polinomials". J. of Approximation Theory. 14, pp. 204-217.

[12] Jupp, D.L.B. (1978). "Approximation to data by splines with free knots". SIAM J. Numer. Anal. v. 15, n. 2, pp. 328-343.

[13] Lawson, C.L. y Hanson, R.J., (1974) Solving least squares problems. Prentice-Hall, Englewood Cliffs, N.J.

[14] Moré, J.J. (1978). "The Levenberg-Marquardt algorithm; implementation and Theory". Gölda & Eckman B. (Eds.), Num. Anal. Proc. Biennial Conf. at Dundee (Jun. 28-Jul. 1, 1977). Springer-Verlag.

[15] Powell, M.J.D. (1981). Approximation Theory and methods. Cambridge University Press.

[16] Rose, D.J., (1969). "An algorithm for solving a special class of tridiagonal systems of linear equations", Comm. ACM, 12. pp. 234-236.

[17] Stewart, G.W., (1973). Introduction to matriz computations. Academic Press.

[18] Varah, J.M. (1980). "A spline least squares method for numerical parameter estimation in differential equations". Computer Science Department. University of British Columbia. 32 pags.

[19] Varah, J.M. (1982) "A spline least squares method for numerical parameter estimation in differential equations". SIAM J. on Scientific and Statistical Computing. v. e, n. 1, pp 28-46.