1 y
10

Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

SISTEMA DE PROGRAMAS PARA LA SINTESIS DE CURVAS

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA
(AREA ELECTRICA Y ELECTRONICA)

P R E S E N T A

MIGUEL ANGEL ALVARADO RASCON

DIR: DR. JORGE ANGELES ALVAREZ

MEXICO, D. F. 1984

SISTEMA DE PROGRAMAS
PARA LA SINTESIS DE CURVAS

I N D I C E

RESUMEN

Este trabajo contiene un sistema de programas para la síntesis de funciones y de curvas planas por medio de funciones *SPLINE* que satisfacen propiedades geométricas locales (pendiente, curvatura , etc.) prescritas.

Al introducir funciones *SPLINE* con puntos de apoyo indeterminados, se tratan las coordenadas de éstos como un conjunto finito de incógnitas por determinar mediante la solución de un sistema algebraico (lineal o no lineal) de ecuaciones.

Las funciones *SPLINE* tienen gran versatilidad y encuentran aplicación en infinidad de problemas dentro de varias disciplinas de la ingeniería y las matemáticas aplicadas. Además, están ampliamente estudiados desde el punto de vista teórico y se conocen con toda precisión sus propiedades de estabilidad numérica.

Las *SPLINES* que se tratan en este trabajo son:

SPLINES periódicas no paramétricas.
SPLINES periódicas paramétricas
SPLINES naturales no paramétricas
SPLINES naturales paramétricas

Se incluyen todas las ecuaciones necesarias aunque no su obtención con todo detalle, refiriéndose para esto al lector, a la literatura correspondiente.

En un apéndice se incluyen listados de todos los programas desarrollados en FORTRAN IV, versión PDP 11/40. Todo el trabajo fue desarrollado en el Laboratorio de Cálculo Automatizado para el Diseño (CAD) de la DEPFI-UNAM.

C. Universitaria, mayo de 1984

1. INTRODUCCION

Una gran cantidad de problemas técnicos involucran en su solución la determinación de curvas que satisfagan condiciones geométricas locales prescritas; estos problemas se han resuelto con curvas o funciones algebraicas que contienen un número reducido de parámetros por determinar, esto es, tienen un grado de libertad muy bajo. Por eso, la aplicación de estas curvas es muy limitado; además no cumplen con las condiciones en todo el intervalo de interés, y cuando se requiere cambiar las condiciones de un punto se tiene que cambiar toda la curva.

Así, se nota que el problema de síntesis de curvas ha sido objeto de interés en muchos trabajos de investigación en campos de la ingeniería, el diseño industrial, el diseño gráfico, etc.

Dentro de la ingeniería se pueden enumerar varias situaciones de síntesis de curvas, como por ejemplo en ingeniería civil, al trazar una curva en carreteras, pistas, o vías de ferrocarril; en estos casos se tiene que unir dos o más puntos obtenidos de las características del campo. Esta curva debe tener una curvatura continua para que el vehículo, al pasar por la trayectoria que une estos puntos, no esté sujeto a cambios que provoquen discontinuidad en la aceleración, lo que produce a su vez discontinuidad en los esfuerzos a que esta sometida su estructura. Además, el principio y el fin de la curva deben tener curvatura cero para que sea unida a líneas rectas en sus extremos, o pueda ser un punto de inflexión.

En el diseño de recipientes a presión, el mayor problema es el diseño del cascarón que debe tener una forma tal que soporte un fluido a grandes presiones [1]. Dicho diseño debe de evitar discontinuidades en la curvatura, que provocarían cambios de signo en los esfuerzos, lo que a su vez produce falla por pandeo.

En la síntesis de mecanismos con seguidores de leva, la síntesis de perfiles de levas ha sido resuelta por polinomios o por segmentos de funciones de una familia dada [2,3] . Es de notar que la aplicación directa de polinomios de interpolación conduce a sistemas de ecuaciones mal condicionados [4] . Por otro lado, para esta síntesis se han utilizado las funciones armó

nicas y cicloidales que realizan bien este propósito, pero solo en pequeñas subintervalos de interés; introduciendo *SPLINES* se puede optimar el movimiento del seguidor.

Para el diseño de engranes el problema que se tiene es la unión de la envolvente con el círculo base, pues en realidad no se sigue ningún método para esta unión, donde aparece una concentración de esfuerzo; dicha concentración se puede evitar introduciendo una *SPLINE* que tenga, por un lado, la curvatura de la envolvente y, por el otro, la curvatura del círculo base, así como que cumpla condiciones de tangencia en la unión.

En el diseño de aberturas en elementos de máquina se sabe que éstas dan lugar a concentración de esfuerzos [5]. La forma de corregir esta concentración es determinar una forma adecuada de la abertura; para esto se han hecho intentos por obtener formas óptimas por el método del elemento finito [2,6]. *Schnack* [7], según lo establecido por *Neuber* [5] ha resuelto este problema aprovechando la relación de monotonía entre la magnitud del esfuerzo en la abertura y la curvatura en el punto correspondiente. Así, este problema se puede tratar como uno puramente geométrico, consistente en determinar una nueva abertura con una mejor distribución del esfuerzo mediante una corrección de la curvatura de la abertura, de acuerdo con la distribución conocida del esfuerzo. Mediante funciones *SPLINE* se puede calcular las coordenadas de los puntos de apoyo para que la curva obtenida posea la curvatura prescrita.

Los espejos parabólicos para la captación de energía solar contienen un contorno real distinto al contorno nominal; para determinar el real se mide la pendiente en una muestra de puntos de perfil [8]. Así, teniendo un conjunto finito de abscisas y un conjunto de valores de la pendiente en los puntos de apoyo de muestra del contorno real, se puede hallar una *SPLINE* que represente el contorno real del espejo.

En arquitectura, en el diseño de cúpulas existen fórmulas para calcular el esfuerzo tangencial [9]; utilizando también aquí la relación de *Neuber* sería posible igualmente obtener el contorno óptimo para abatir la concentración del esfuerzo para diferentes tipos de carga a que se vaya a someter la estructura.

En cuanto al análisis de esfuerzo y deformación en vigas, existe el método de doble integración para calcular la curva elástica. Este método puede realizarse eficientemente utilizando *SPLINES*, ya que existen situaciones en las que el método exige el cálculo de integrales que no existen en tablas.

Haciendo referencia al párrafo anterior, también pueden utilizarse las funciones *SPLINE* para hallar soluciones de ecuaciones diferenciales ordinarias sujetas a condiciones de frontera [10].

Es de notar que la síntesis de funciones es de gran aplicación porque existen múltiples problemas técnicos que pueden resolverse con ayuda de esta herramienta, y que sólo se ha utilizado recientemente [11], a pesar de que las funciones *SPLINE* han sido utilizadas por analistas numéricos y matemáticos por décadas, y se han dedicado libros completos tanto a su estudio teórico como a sus aplicaciones, especialmente durante la década pasada [10,12,13].

Durante los últimos 5 años ha crecido el interés por la aplicación en síntesis de curvas [14,21,22]. Sin embargo, los métodos hallados en la literatura sólamente se refieren a problemas de interpolación, dado un conjunto de puntos de apoyo conocidos. Este método permite al diseñador sólo modificar las propiedades geométricas de la curva en los puntos de interés [10,15] mediante modificaciones adecuadas en algunos parámetros de la curya de interpolación. permaneciendo los puntos de apoyo sin cambios.

En este trabajo, la síntesis es un punto de vista radicalmente diferente pues, a partir de las propiedades geométricas prescritas, permite calcular las coordenadas de los puntos de apoyo en forma algebraica. Permite, además, su optimación por medio de programación matemática.

2. INTERPOLACION

Para situar el problema en el contexto del análisis numérico es necesario ubicarlo dentro de la solución de ecuaciones [15]. Para esto, considerése la transformación siguiente:

$$T(x) = y$$

donde $x \in X$, $y \in Y$, siendo X y Y espacios lineales, no necesariamente de la misma dimensión, en tanto que T es una transformación de X a Y.

De la relación anterior se pueden reconocer tres tipos de problemas.

1. Problema directo. Dados T y x, calcular y.
2. Problema inverso. Dados T y y, calcular x.
3. Problema de identificación. Dados x y y, determinar T.

En el lenguaje de la ingeniería x, y y T representan la excitación, la respuesta y el sistema, respectivamente. Así en el problema directo se trata de determinar la respuesta generada por un sistema con excitación conocida. En el problema inverso se busca una excitación que genere una respuesta conocida. En el problema de identificación se tienen que determinar las leyes que rigen el sistema, a partir de una relación conocida entre la excitación y la respuesta.

El problema directo es relativamente fácil de tratar, mientras que el problema inverso, por sus aplicaciones importantes, ocupa un plano central en el análisis numérico.

El caso de identificación es más difícil, ya que mediante un número finito de observaciones, se tiene que hallar las leyes que gobiernan un sistema. Esto es generalmente imposible de realizar, a menos que se tenga información específica sobre la estructura del sistema.

El problema de la aproximación de funciones es un caso especial del de identificación y consiste en determinar una función que pase por un conjunto de puntos (x_i, y_i), $i = 0, 1, \ldots, n$, donde n es el número de puntos de muestra.

Uno de los métodos más conocidos de aproximación es la interpolación *po*

Linómica, la cual consiste en determinar un polinomio $P_n(t)$ (polinomio de grado n) que tome valores prescritos y_i en ciertos puntos x_i, $i=0,1,\ldots,n$.

En este método se dispone de técnicas clásicas de aproximación, como la de *Fourier* (1822), que consiste en formar una suma finita de términos

$$\tilde{f}(x) = c_0 + c_1\phi_1(x) + c_2\phi_2(x) + \ldots + c_n\phi_n(x)$$

que aproxime la función $f(x)$.

Las funciones $\phi_i(x)$ generan un espacio finito conocido, mientras que los coeficientes c_k son constantes desconocidas que hay que determinar mediante condiciones impuestas sobre $f(x)$.

Teniendo $n+1$ constantes desconocidas, se debe tener $n+1$ condiciones impuestas en $\tilde{f}(x)$.

Existe un gran número de condiciones que se pueden escoger para $\tilde{f}(x)$; entre ellas se tienen:

a) Condiciones de interpolación

$$\tilde{f}(x_i) = f(x_i), \quad i = 0, 1, 2, \ldots, n$$

donde n es el número de puntos de interpolación.

b) Condiciones de minimalidad. Las constantes c_i, $i = 0, 1, 2, \ldots, n$, se escogen de tal manera que minimicen una norma [10] del error, $\|f-\tilde{f}\|$

c) Mezcla de condiciones de interpolación y de continuidad en las derivadas.

c.1 $\tilde{f}(x_i) = f(x_i) \quad i = 0, 1, 2, \ldots, n$
c.2 $\tilde{f}(x_1) = f'(x_1)$ y $\tilde{f}'(x_k) = f'(x_k)$
c.3 $\tilde{f}(x)$ posea segundas derivadas continuas

Para el caso (a) pueden utilizarse los polinomios de *Lagrange*, lo que da lugar al problema de interpolación más sencillo [10]. Estos polinomios se pueden

determinar suponiendo valores prescritos en $n+1$ valores reales distintos x_i, $i=0,1,2,\ldots,n$; el polinomio de *Lagrange* de grado n asociado con los puntos de apoyo $[x_i, y_i]$, $i=1,2,\ldots,n$, es un polinomio $P_n(x)$ que resuelve el problema de interpolación

$$P_n(x_i) = y_i \qquad 0 \leq i \leq n$$

donde $y_i = f(x_i)$, siendo $f(x)$ como una función cualquiera.

En particular, para definir un polinomio de grado n es necesario dar

$$w(x) = (x-x_0)(x-x_1)\ldots(x-x_n)$$

De esta suerte,

$$P_n(x) = \sum_{i=0}^{n} f(x_i)\, \ell_i(x), \tag{2.1}$$

donde

$$\ell_i(x) = \frac{w(x)}{(x-x_i)\, w'(x)}$$

$$= \frac{(x-x_0)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_n)} \tag{2.2}$$

De la expresión (2.2), por simple sustitución se tiene

$$\ell_i(x_j) = \delta_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases} \qquad i,j=0,1,\ldots n$$

donde δ_{ij} es llamada la delta de *Kronecker*.

Es fácil ver que cada $\ell_i(x)$ es un polinomio de grado n y que $P_n(x)$, dado en (2.1), es un polinomio del grado requerido.

Una manera de calcular el polinomio de *Lagrange* es escogiendo una base adecuada. A cada base corresponde un método distinto para la obtención de $P_n(x)$.

El método más conocido es el siguiente:

Se desarrolla el polinomio de interpolación utilizando el espacio finito $\{1, x, \ldots, x^n\}$. Entonces se tiene que determinar un polinomio de *Lagrange* de la

forma

$$P_n(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_n x^n \qquad (2.3)$$

que es una combinación lineal de los polinomios $1, x, \ldots, x^n$ y que cumpla con las condiciones de interpolación

$$P_n(x_i) = f(x_i), \quad i=0,1,\ldots,n \qquad (2.4a)$$

dando lugar a un sistema de $n+1$ ecuaciones lineales

$$c_0 + c_1 x_i + c_2 x_i^2 + \ldots + c_n x_i^n = f(x_i), \quad i=0,1,\ldots,n \qquad (2.4b)$$

que tiene una única solución $(c_0, c_1, \ldots, c_n)$.

En forma matricial quedan las ecs (2.4b) en la forma

$$A\,c = b$$

donde A es la matriz de *Vandermonde*, no singular, de la forma

$$A = \begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ & & \vdots & & \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{bmatrix}$$

siendo b y c los siguientes vectores

$$b = [f(x_0),\ f(x_1),\ \ldots,\ f(x_n)]^T$$

y $$c = [c_0, c_1, \ldots, c_n]^T$$

Este método es poco práctico porque la matriz A es generalmente mal condicionada [10]

Otro método es el de la base *canónica*, en donde se utilizan las expresiones (2.1) y (2.2). Suponiendo que $\{\ell_0(x), \ell_1(x), \ldots, \ell_n(x)\}$ sea otra base del espacio lineal P_n, un simple desarrollo sería

$$P_n(x) = f(x_0)\,\ell_0(x) + f(x_1)\,\ell_1(x) + \ldots + f(x_n)\,\ell_n(x) \qquad (2.5)$$

donde se puede observar que el calculo de $P_n(x)$, en términos de esta base, es mucho más simple que con la base anterior.

Otra clase de interpolación es la de *Hermite* que generaliza la interpolación de *Lagrange* [10], porque además de ajustar la función $f(x)$ interpolándola en cada punto de apoyo (x_i, y_i), $i=1,2,\ldots,n$, también la interpola en un número finito de derivadas consecutivas de $f(x)$ en los puntos de apoyo y entre los tramos comprendidos en ellos. Con esta clase de interpolación el trabajo de cálculo aumenta considerablemente, obteniendo a cambio mayor derivabilidad y continuidad en las curvas.

En particular, un polinomio de interpolación de *Hermite* da un ajuste mejor a la función $f(x)$ que el obtenido mediante su correspondiente polinomio de *Lagrange*.

La construcción de un polinomio de *Hermite* es parecida a la de un polinomio de *Lagrange*, sólo que cambian las condiciones; éstas son de interpolación y de continuidad en las derivadas.

En general, se puede decir que, si se tiene un conjunto finito de números reales x_i, $i=1,2,\ldots,n$, un conjunto de enteros positivos m_i, $i=1,2,\ldots,n$, donde n es el número de puntos de apoyo, existe un polinomio $P(x)$ de grado $m_1 + m_2 + m_n - 1 = N$ o menor, el cual en cada punto de apoyo x_i, $i=1,2,\ldots,n$ resuelve el problema de interpolación

$$P^{(j)}(x_i) = f^{(j)}(x_i) \qquad (2.6)$$

donde $j=0,1,\ldots,m_i - 1$ y $f(x)$ es una función de $m_i - 1$ derivadas consecutivas en x_i, $i=1,2,\ldots,n$, quedando el polinomio buscado de la forma

$$P(x) = a_N x^N + a_{N-1} x^{N-1} + \ldots + a_0 \qquad (2.7)$$

tal que en cada x_i, $i=1,2,\ldots,n$ se cumplan las condiciones impuestas en (2.6).

El polinomio (2.7) contiene $N+1$ coeficientes desconocidos $a_N, a_{N-1}, \ldots, a_0$. El método para hallarlos es mantener fijo un punto de apoyo y aplicar las condiciones dadas en (2.6), produciendo con esto m_i ecuaciones li

neales del total de las $N+1$ ecuaciones desconocidas.

Si $i = 2$ y $m_2 = 3$, se obtienen las siguientes ecuaciones

$$\begin{aligned} p(x_2) &= f(x_2) \\ p'(x_2) &= f'(x_2) \\ p''(x_2) &= f''(x_2) \end{aligned} \qquad (2.8)$$

Sustituyendo (2.7) en (2.8), se tiene

$$\begin{aligned} & a_N x_2^N + a_{N-1} x_2^{N-1} + \ldots + a_0 = f(x_2) \\ & N a_N x_2^{N-1} + (N-1) a_{N-1} x_2^{N-2} + \ldots + a_1 = f'(x_2) \\ & (N-1) N a_N x_2^{N-2} + (N-2)(N-1) a_{N-1} x_2^{N-3} + \ldots + 2a_2 = f''(x_2) \end{aligned}$$

Así, desarrollando en cada punto x_i, $i = 1,2,\ldots,n$ el número correspondiente de igualdades m_i, $i=1,2,\ldots,n$, queda un total de $m_1 + m_2 + \ldots + m_n$ ecuaciones lineales, obteníendose el mismo número de ecuaciones que de incógnitas. Se puede demostrar que este sistema tiene una solución única [10].

En forma matricial, se obtiene el siguiente sistema

$$A\,y = b$$

donde

$$y = [a_N, a_{N-1}, \ldots, a_0]^T$$

$$b = [f(x_1), f'(x_1), \ldots, f^{(m_1-1)}(x_1), f(x_2), \ldots, f^{(m_k-1)}(x_k)]^T$$

construyéndose la matriz A como en el ejemplo anterior.

Es importante hacer notar que el grado N del polinomio de *Hermite* es considerablemente grande en la mayoría de los casos.

En particular, un polinomio cúbico de *Hermite*, para 2 puntos de apoyo, es de la forma

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

donde deben satisfacerse las siguientes condiciones

$$p(x_i) = f(x_i)$$
$$p'(x_i) = f'(x_i), \quad i = 1, 2$$

Los polinomios cúbicos construidos por tramos son de gran utilidad para eliminar las discontinuidades que provocaría un polinomio de *Lagrange* de grado 1 entre cada 2 puntos de apoyo.

Ahora bien, teniendo n puntos de apoyo (x_i, y_i), $i = 1, 2, \ldots, n$, se puede construir un polinomio cúbico por tramos que resuelva las condiciones

$$p_i(x_{i-1}) = f(x_{i-1}) \qquad p_i'(x_{i-1}) = f'(x_{i-1})$$
$$p_i(x_i) = f(x_i) \qquad p_i'(x_i) = f'(x_i)$$

Llamando $S(x)$ al polinomio cúbico por tramos correspondiente, se tiene

$$S(x) = p_i(x)$$

Es claro que, para un polinomio cúbico de *Hermite* que une sólamente 2 puntos, sólo puede tenerse $m = 0, 1$, y que sólo la primera derivada tendrá continuidad en el intervalo de interés. Para obtener una segunda derivada que tenga continuidad en el intervalo de interés se utilizan las *splines* cúbicas.

3. INTERPOLACION MEDIANTE FUNCIONES *SPLINE*

En los capítulos anteriores se han tratado problemas únicamente de interpolación, y de continuidad en la primera o en la segunda derivada;pero no en los dos simultáneamente.

En este capítulo se verá que es posible obtener continuidad tanto en la primera como en la segunda derivada simultáneamente, si se utilizan funciones *spline*.

Una función *spline* cúbica es un polinomio cúbico definido por tramos (ver figura 3.1) de la forma [16] :

$$f_k(x) = A_k\,(x-x_k)^3 + B_k\,(x-x_k)^2 + C_k\,(x-x_k) + D_k, \qquad (3.1)$$

para $k = 1, 2, \ldots, n-1, \qquad x_k \leq x \leq x_{k+1}$

donde n es el número de puntos de apoyo, en tanto que A_k, B_k, C_k y D_k son los coeficientes de la *spline*.

De aquí en adelante se define

$$y_k = f_k(x_k) \qquad y_{k+1} = f_k(x_{k+1})$$
$$y'_k = f'_k(x_k) \qquad y'_{k+1} = f'_k(x_{k+1})$$
$$y''_k = f''_k(x_k) \qquad y''_{k+1} = f''_k(x_{k+1})$$

$$\Delta x_k = x_{k+1} - x_k \qquad \Delta y'_k = y'_{k+1} - y'_k$$
$$\Delta y_k = y_{k+1} - y_k \qquad \Delta y''_k = y''_{k+1} - y''_k$$

Para determinar los coeficientes se evalúan f_k, f'_k y f''_k para x_k y x_{k+1}

$$f_k(x_k) = D_k \tag{3.2}$$

$$f_k(x_{k+1}) = A_k \Delta x_k^3 + B_k \Delta x_k^2 + C_k \Delta x_k + D_k \tag{3.3}$$

$$f'_k(x_k) = C_k \tag{3.4}$$

$$f'_k(x_{k+1}) = 3 A_k \Delta x_k^2 + 2 B_k \Delta x_k + C_k \tag{3.5}$$

$$f''_k(x_k) = 2 B_k \tag{3.6}$$

$$f''_k(x_{k+1}) = 6 A_k \Delta x_k + 2 B_k \tag{3.7}$$

Despejando los valores de los coeficientes A_k, B_k, C_k, y D_k, de (3.2), (3.3), (3.6) y (3.7) se obtiene

$$A_k = \frac{1}{6\Delta x_k} (y''_{k+1} - y''_k) = \frac{\Delta y''_k}{6 \Delta x_k} \tag{3.8}$$

$$B_k = \frac{1}{2} y''_k \tag{3.9}$$

$$C_k = \frac{\Delta y_k}{\Delta x_k} - \frac{1}{6} \Delta x_k (y''_{k+1} + 2y''_k) = y'_k \tag{3.10}$$

$$D_k = y_k \tag{3.11}$$

El valor de la primera derivada en función de (x,y,y") se halla sustituyendo (3.10) en (3.4), con lo que queda

$$y'_k = \frac{\Delta y_k}{\Delta x_k} - \frac{1}{6} \Delta x_k (y''_{k+1} + 2y''_k) \tag{3.12}$$

$k = 1, 2, \ldots n-1$

Imponiendo una condición de continuidad en la *spline* en su primera derivada, en x_k, se tiene

$$f'_{k-1}(x_k) = f'_k(x_k), \qquad k = 2, 3, \ldots n-1. \tag{3.13}$$

Combinando (3.13) con (3.4) y (3.5) se obtiene

$$3A_{k-1}\ \Delta x_{k-1} + 2\,B_{k-1}\,\Delta x_{k-1} + C_{k-1} = C_k \tag{3.14}$$

y sustituyendo los coeficientes (3.8)-(3.11), se obtiene

$$\Delta x_{k-1}\, y''_{k-1} + 2(\Delta x_{k-1} + \Delta x_k)\ y''_k + \Delta x_k\, y''_{k+1} = 6\left(\frac{\Delta y_k}{\Delta x_k} - \frac{\Delta y_{k-1}}{\Delta x_{k-1}}\right) \tag{3.15}$$

Ahora, desarrollando para $k = 2, 3, \ldots, n-1$, se obtiene un sistema de $n-2$ ecuaciones de la siguiente forma:

$$\Delta x_1\, y''_1 + 2(\Delta x_1 + \Delta x_2)\, y''_2 + \Delta x_2\, y''_3 = 6\left[\frac{y_1}{\Delta x_1} - \left(\frac{1}{\Delta x_1} + \frac{1}{\Delta x_2}\right) y_2 + \frac{y_3}{\Delta x_2}\right] \tag{3.16a}$$

$$\Delta x_2\, y''_2 + 2(\Delta x_2 + \Delta x_3)\, y''_3 + \Delta x_3\, y''_4 = 6\left[\frac{y_2}{\Delta x_2} - \left(\frac{1}{\Delta x_2} + \frac{1}{\Delta x_3}\right) y_3 + \frac{y_4}{\Delta x_3}\right] \tag{3.16b}$$

$$\Delta x_{n-2}\, y''_{n-2} + 2(\Delta x_{n-2} + \Delta x_{n-1})\, y''_{n-1} + \Delta x_{n-1}\, y''_n = 6\left[\frac{y_{n-2}}{\Delta x_{n-2}} - \left(\frac{1}{\Delta x_{n-2}} + \frac{1}{\Delta x_{n-1}}\right) y_{n-1} + \frac{y_n}{\Delta x_{n-1}}\right] \tag{3.16c}$$

Escribiendo (3.16a,b y c) en forma matricial,

$$Ay'' = 6\,Cy \tag{3.17}$$

donde

$$A = \begin{bmatrix} \Delta x_1 & 2(\Delta x_1 + \Delta x_2) & \Delta x_2 & & & \\ & \Delta x_2 & 2(\Delta x_2 + \Delta x_3) & \Delta x_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \Delta x_{n-2} & 2(\Delta x_{n-2} + \Delta x_{n-1}) & \Delta x_{n-1} \end{bmatrix} \tag{3.18}$$

$$y = [y_1, y_2, \ldots, y_n]^T \tag{3.19}$$

$$y'' = [y''_1, y''_2, \ldots, y''_n]^T \tag{3.20}$$

$$C = \begin{bmatrix} \frac{1}{\Delta x_1} & -\left(\frac{1}{\Delta x_1}+\frac{1}{\Delta x_2}\right) & \frac{1}{\Delta x_2} & & \\ & \frac{1}{\Delta x_2} & -\left(\frac{1}{\Delta x_2}+\frac{1}{\Delta x_3}\right) & \frac{1}{\Delta x_3} & \\ & & \ddots & \ddots & \ddots \\ & & \frac{1}{\Delta x_{n-2}} & -\left(\frac{1}{\Delta x_{n-2}}+\frac{1}{\Delta x_{n-1}}\right) & \frac{1}{\Delta x_{n-1}} \end{bmatrix} \qquad (3.21)$$

Es de notar que la ecuación (3.15) da la relación entre la ordenada y la segunda derivada de la función de interpolación.

Ahora, desarrollando la ecuación (3.12), se determina la relación de la ordenada con la primera derivada ;

$$y_1' = \frac{\Delta y_1}{\Delta x_1} - \frac{1}{6}\Delta x_1\,(y_2'' + 2y_1'')$$

$$y_2' = \frac{\Delta y_2}{\Delta x_2} - \frac{1}{6}\Delta x_2\,(y_3'' + 2y_2'')$$

$$\vdots$$

$$y_{n-1}' = \frac{\Delta y_{n-1}}{\Delta x_{n-1}} - \frac{1}{6}\Delta x_{n-1}\,(y_n'' + 2y_{n-1}'')$$

que se puede escribir en forma matricial como

$$y' = Fy - \frac{1}{6}Gy'' \qquad (3.22)$$

donde:

$$y' = [y_1', y_2', \ldots, y_n']^T \qquad (3.23)$$

$$F = \begin{bmatrix} -\frac{1}{\Delta x_1} & \frac{1}{\Delta x_2} & & & \\ & -\frac{1}{\Delta x_2} & \frac{1}{\Delta x_2} & & \\ & & \ddots & \ddots & \\ & & & -\frac{1}{\Delta x_{n-1}} & \frac{1}{\Delta x_{n-1}} \\ & & & -\frac{1}{\Delta x_{n-1}} & \frac{1}{\Delta x_{n-1}} \end{bmatrix} \qquad (3.24)$$

$$G = \begin{bmatrix} 2\Delta x_1 & \Delta x_1 & & & \\ & 2\Delta x_2 & \Delta x_2 & & \\ & & \ddots & \ddots & \\ & & & 2\Delta x_{n-1} & \Delta x_{n-1} \\ & & & \Delta x_{n-1} & 2\Delta x_{n-1} \end{bmatrix} \qquad (3.25)$$

habiéndose definido y y y" previamente,en (3.19) y (3.20), respectivamente.

Es necesario aclarar aquí que las matrices A,C,F y G son funciones de la - varible Δx_k, pero, para una función paramétrica es necesario definir Δt_k que es una variable que se define en seguida. Sea

$$\Delta t_k = \sqrt{\Delta x_k^2 + \Delta y_k^2} \quad , \quad k=1, 2, \ldots, n-1 \qquad (3.26)$$

y t_k , la coordenada obtenida de la forma

$$t_1 = 0 \qquad (3.27)$$

$$t_{k+1} = t_k + \Delta t_k \quad , \quad k = 1, 2, \ldots, n-1$$

Volviendo a las ecuaciones (3.17) y (3.22) se puede mencionar que la matriz A es simetrica, positiva definida y tridiagonal, lo que garantiza su estabilidad número rica.

Es positiva definida porque para toda $x \neq 0$.

$$x^T A x > 0$$

Es tridiagonal porque contiene elementos no nulos sólo en su diagonal y en las dos líneas vecinas a ella.

Las matrices C y F son singulares, es decir, su rango es menor que su orden, lo que trae como consecuencia que los productos Cy y Fy sean nulos para algunos $y \neq 0$; de hecho, su espacio nulo es de dimensión uno. Esta característica permite asignar al vector y un incremento de cualquier magnitud pero igual en todas y cada una de las ordenadas de los puntos de apoyo sin que se alteren los vectores y' y y".

La figura 3.2 permite entender el concepto anterior. En ella se nota que la curva (1) es idéntica a la (2), sólo que afectada de una traslación de cuerpo rígido.

Se pueden clasificar las funciones *spline* según las características en los puntos de apoyo como sigue:

Spline periódica no paramétrica. Para ésta, tanto el vector y como y' y y" tienen idénticos sus componentes primero y último, es decir:

$$y_1 = y_n$$
$$y'_1 = y'_n$$
$$y''_1 = y''_n$$

Así, los vectores y, y' y y'' no tienen que ser de dimensión n sino $n-1$, en el caso más general. Esta dimensión puede disminuirse más aún si se introducen condiciones de simetría.

Spline periódica paramétrica. Tiene las mismas características que la *spline* periódica no paramétrica, sólo que en este caso, en lugar de usar x como variable independiente, se utiliza el parámetro t cuya definición fue dada en (3.27).

Spline natural no paramétrica. En este caso se tiene

$$y_1'' = 0$$

$$y_n'' = 0$$

Su significado es que su curvatura en los puntos inicial y final es igual a cero; es decir, la curva comienza y termina en una línea recta.

Análogamente, la *spline* natural paramétrica tiene las mismas características, excepto que la variable independiente es t , definida en (3.27).

Otro tipo de *spline* existente es la *Spline* B [10]. Esta es una suma de polinomios cúbicos $B_i(x)$ $i = 0,1, \ldots n$, definidos de la siguiente forma :

$$B_i(x) = \begin{cases} (x-x_{i-2})^3 & \text{si } x\in[x_{i-2}, x_{i-1}] \\ h^3+3h^2(x-x_{i-1})+3h(x-x_{i-1})^2-3(x-x_{i-1})^3 & \text{si } x\in[x_{i-1}, x_i\ ] \\ h^3+3h^2(x_{i+1}-x)+3h(x_{i+1}-x)^2-3(x_{i+1}-x)^3 & \text{si } x\in[x_i\ , x_{i+1}] \\ (x_{i+2}-x)^3 & \text{si } x\in[x_{i+1}, x_{i+2}] \\ 0 & \text{para cualquier otro caso.} \end{cases} \qquad (3.28)$$

donde h es la longitud común a los cuatro intervalos que aparecen en (3.28).

Se observa por simple sustitución que cada $B_i(x)$ tiene sus 2 primeras derivadas continuas. En la gráfica de la Fig 3.3 se puede ver que

$$B_i(x_j) = \begin{cases} 4 & \text{si} \quad j = i \\ 1 & \text{si} \quad j = i-1 \text{ o } j = i = i+1 \\ 0 & \text{si} \quad j = i+2 \text{ o } j = i-2 \end{cases}$$

y que $B_i(x) \equiv 0$ para $x \geq x_{i+2}$ y $x \leq x_{i-2}$. Entonces es necesario introducir 2 intervalos adicionales, $x_{-2} < x_{-1} < x_0$, para $B_0(x)$, y otros dos, $x_{n+2} > x_{n+1} > x_n$, para $B_n(x)$, ocasionando que las *splines* B sean las únicas que no se anulen en los intervalos comprendidos entre

$$x_{-2} < x_{-1} < \ldots < x_n < x_{n+1} < x_{n+2}.$$

Entre las condiciones que puede satisfacer una *spline* B están

$$\begin{aligned} S'(x_0) &= f'(x_0) \\ S(x_i) &= f(x_i) \qquad 0 \leq i \leq n \\ S'(x_n) &= f(x_n) \end{aligned} \tag{3.29}$$

donde S(x) es la *spline* B de interpolación para f(x).

Para calcular S(x) se utiliza la tabla 3.1, donde se muestran los valores para $B_i(x)$, $B_i'(x)$ y $B_i''(x)$, omitiendo los valores nulos.

Tabla 3.1

	x_{j-2}	x_{j-1}	x_j	x_{j+1}	x_{j+2}
$B_j(x)$	0	1	4	1	0
$B_j'(x)$	0	$3/h$	0	$-3/h$	0
$B_j''(x)$	0	$6/h^2$	$-12/h^2$	$6/h^2$	0

Una función S(x), representada por una combinación lineal de *splines* B tiene la forma

$$S(x) = a_{-1} B_{-1}(x) + a_0 B_0(x) + \ldots + a_{n+1} B_{n+1}(x) \qquad (3.30)$$

sometida a las condiciones (3.28), con lo que se obtiene el siguiente sistema de ecuaciones:

$$S'(x_0) = a_{-1} B'_{-1}(x_0) + a_0 B'_0(x_0) + \ldots + a_{n+1} B'_{n+1}(x_0) = f'(x_0)$$

$$S(x_i) = a_{-1} B_{-1}(x_i) + a_0 B_0(x_i) + \ldots + a_{n+1} B_{n+1}(x_i) = f(x_i), \quad 0 \leq i \leq n$$

$$S'(x_n) = a_{-1} B'_{-1}(x_n) + a_0 B'_0(x_n) + \ldots + a_{n+1} B'_{n+1}(x_n) = f(x_n)$$

En forma matricial se puede escribir este sistema como

$$A \quad c = b$$

donde :

$$A = \begin{bmatrix} -\frac{3}{h} & 0 & \frac{3}{h} & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & 1 & 4 & 1 \\ & & & & -\frac{3}{h} & 0 & \frac{3}{h} \end{bmatrix}$$

$$c = [a_{-1}, a_0, a_1, \ldots a_n, a_{n+1}]^T$$

$$b = [f'(x_0), f(x_0), f(x_1), \ldots, f(x_n), f'(x_n)]^T$$

La matriz A es simétrica, diagonalmente dominante de orden $(n+3) \times (n+3)$ e invertible.

También se puede ver de manera análoga que existe una *spline* B única $\bar{S}(x)$ dada por (3.30) que satisface las siguientes condiciones

$$\begin{aligned} \bar{S}''(x_0) &= f''(x_0) \\ \bar{S}(x_i) &= f(x_i), \quad 0 \leq i \leq n \\ \bar{S}''(x_n) &= f''(x_n) \end{aligned} \tag{3.31}$$

donde la *spline* B, $\bar{S}(x)$, es llamada *spline* natural cúbica para $f(x)$ en el caso de las *spline* B. Es fácil determinar $\bar{S}(x)$ cambiando sólo el primero y el último renglón de A por

$$\begin{aligned} S''(x_0) &= a_{-1} B''_{-1}(x_0) + a_0 B''_0(x_0) + \ldots + a_{n+1} B''_{n+1}(x_0) = f''(x_0) \\ S''(x_n) &= a_{-1} B''_{-1}(x_0) + a_0 B''_0(x_n) + \ldots + a_{n+1} B''_{n+1}(x_n) = f''(x_n) \end{aligned} \tag{3.32}$$

respectivamente.

Adelante se describe el desarrollo a seguir para obtener las *spline* B, pudiéndose generalizar para polinomios de grado m. Para este fin se determina la ka diferencia recursiva $f(x_0)$ de la función $f(x)$ en x_0 de la siguiente forma

$$\begin{aligned} \Delta f(x_0) &= f(x_1) - f(x_0) \\ \Delta^{k-1} f(x_0) &= \Delta^k f(x_1) - \Delta^k f(x_0) \end{aligned} \tag{3.33}$$

En particular, para $k = 4$,

$$\Delta^4 f(x_0) = f(x_4) - 4f(x_3) + 6f(x_2) - 4f(x_1) + f(x_0) \qquad (3.34)$$

En la expresión anterior los coeficientes de $f(x_k)$ son los del binomio de *Newton*,

$$(-1)^k \binom{n}{k} = \frac{n!}{k!(n-k)!}(-1)^k \ , \quad k = 0, 1, \ldots, n \qquad (3.35)$$

En este caso particular hay que definir

$$F_t(x) = (x - t)_+^3$$

donde $$(x-t)_+^3 = \begin{cases} (x - t)^3 & \text{para } t \leq x \\ 0 & \text{para } t > x \end{cases}$$

cuya grafica aparece en la Fig 3.4 Se obtiene así,

$$K(t) = \Delta^4 F_t(x_0)$$

en la forma siguiente:

$$K(t) = F_t(x_4) - 4F_t(x_3) + 6F_t(x_2) - 4F_t(x_1) + F_t(x_0)$$

$$= (x_4-t)_+^3 - 4(x_3-t)^3 + 6(x_2-t)_+ - 4(x_1-t)_+ + (x_0 - t)_+^3$$

Ahora sólo es necesario decir que $\Delta^n f(t)$ anula todos los polinomios de grado $n-1$, demostrándose en [10] que $K(t) \equiv 0$ cuando $t \geq x_4$ y $t < x_0$

En particular,

$$K(t) = \begin{cases} (x_4-t)^3 & x_3 < t \leq x_4 \\ (x_4-t)^3 - 4(x_3-t)^3 & x_2 < t \leq x_3 \\ (x_4-t)^3 - 4(x_3-t)^3 + 6(x_2-t)^3 & x_1 < t \leq x_2 \\ (x_4-t)^3 - 4(x_3-t)^3 + 6(x_2-t)^3 - 4(x_1-t)^3 , & x_0 < t \leq x_1 \\ 0 & \text{para cualquier otro valor de } t \end{cases}$$

Simplificando y llamando

$$x_i = x_0 + ih$$

$$x_{ij} = x_i + jh$$

Por ejemplo,

$$(x_4-t)^3 - (4x_3-t)^3 = [(x_3-t) + h]^3 - 4(x_3-t)^3$$

$$= (x_3-t)^3 + 3(x_3-t)^2 h + 3(x_3-t) h^2 + h^3 - 4(x_3-t)^3$$

$$= h^3 + 3h^2 (x_3-t) + 3h(x_3-t)^2 - 3(x_3-t)^3.$$

se puede observar que esta expresión es la tercera línea de (3.28) valuado en $i = 2$, por lo que se concluye que

$$\frac{1}{h^3} K(t) = B_2(t) = \frac{1}{h^3} \Delta^4 F_t(x_0)$$

y además

$$B_i(x) = \frac{1}{h^3} \Delta^4 F_t(x_{i-2})$$

Generalizando esta fórmula, es necesario definir

$$F_t(x) = (x-t)_+^m$$

obteniéndose

$$K(t) = \Delta^{m+1} F_t(x) = \sum_{i=0}^{m=1} \binom{m+1}{i} (-1)^i (x_i-t)_+^m$$

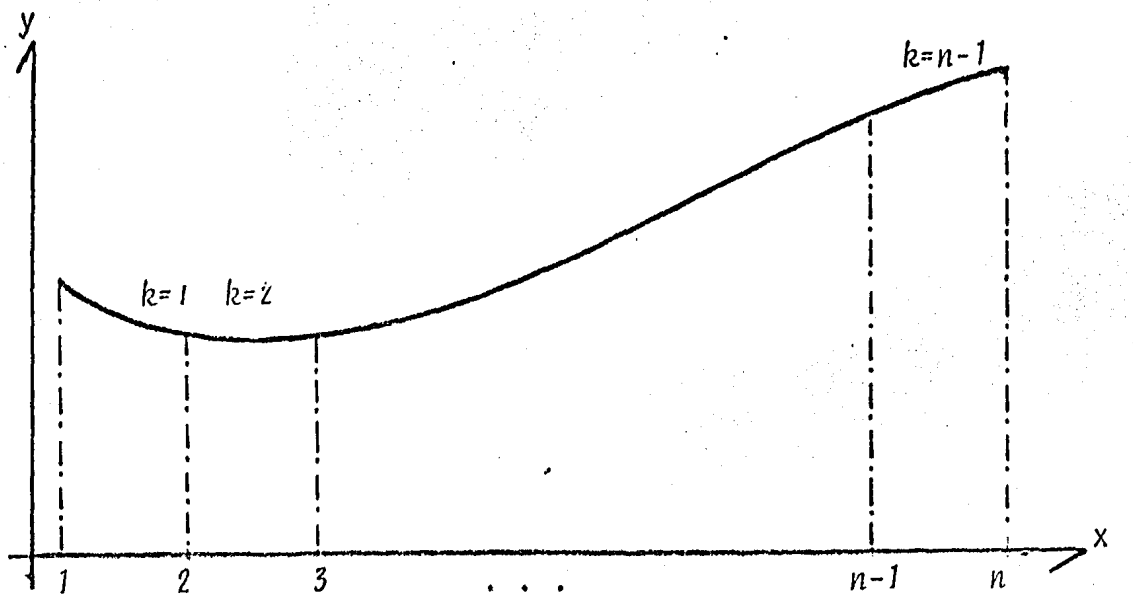


Fig 3.1 *Spline* cúbica por tramos.

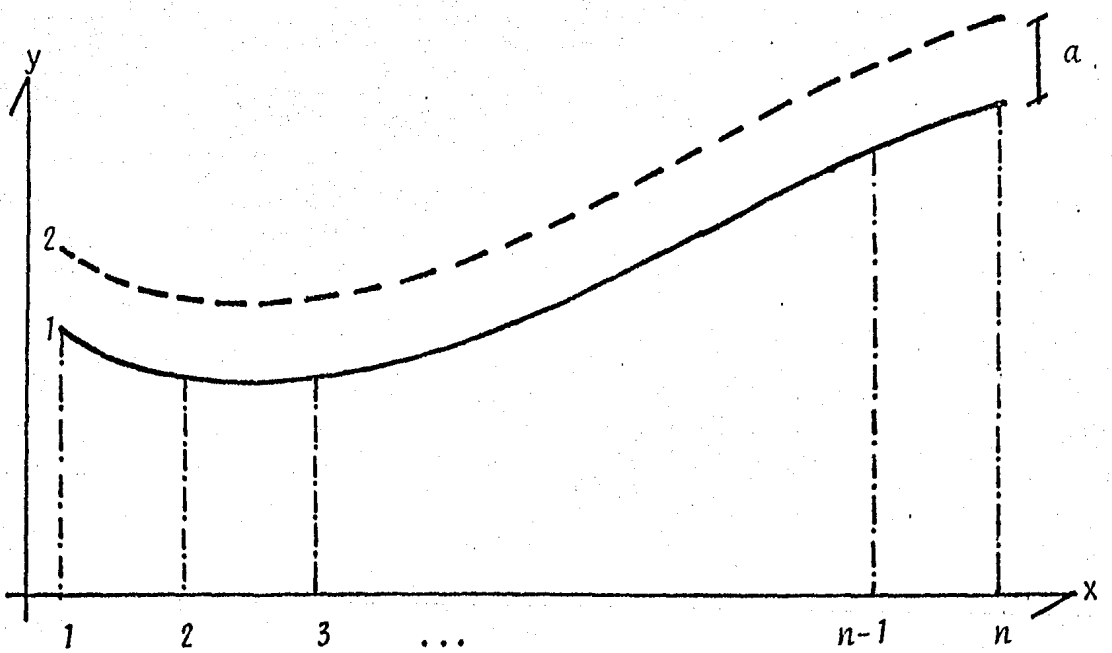


Fig 3.2 Desplazamiento de cuerpo rigído
de la *spline* de la Fig 3.1

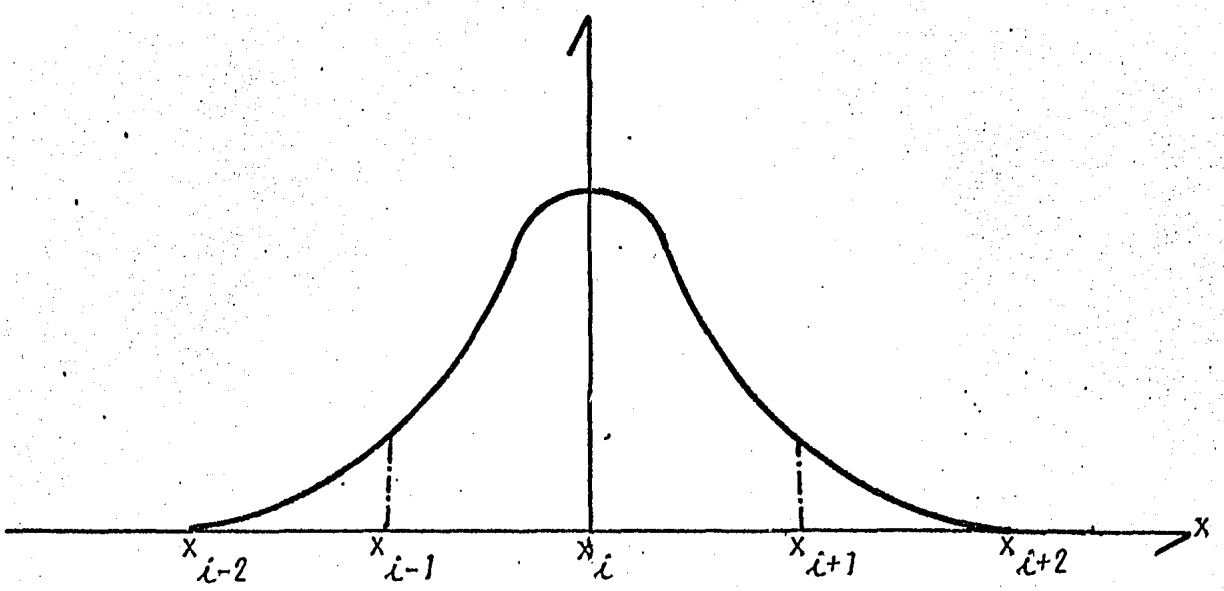


Fig 3.3 Gráfica de $B_i(x)$

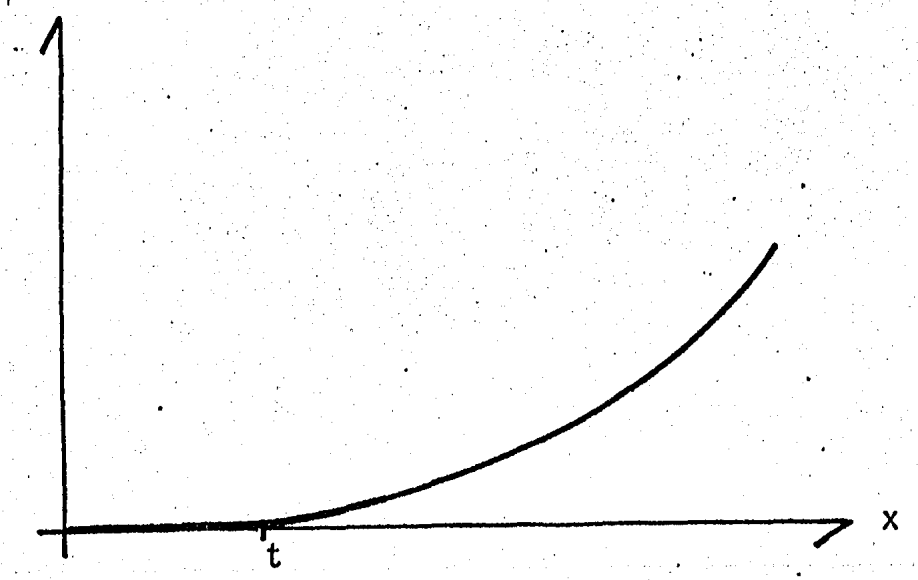


Fig 3.4 Gráfica de $(x-t)_+^3$

4. SINTESIS DE FUNCIONES MEDIANTE *SPLINES* NATURALES Y PERIODICAS

Dada una función *spline* que aproxima una función y=y (x), siendo x_i, $i = 1, 2, \ldots, n$ el conjunto de abscisas de los puntos de apoyo de la *spline*, se define el vector y como aquél que contiene las ordenadas correspondientes a las abscisas x_i en los puntos de apoyo. Análogamente se definen los vectores y' y y".

Como ya se mencionó, en una *spline* natural $y''_1 = y''_n = 0$. Es por esto que la dimensión del vector y" es $n-2$.

Como ya se vio, la relación entre y y y" está dada por la expresión (3.17).

Por las condiciones de una *spline* natural la matriz A queda ahora de la forma

$$A = \begin{bmatrix} 2(\Delta x_1 + \Delta x_2) & \Delta x_2 & & & \\ \Delta x_2 & 2(\Delta x_2 + \Delta x_3) & \Delta x_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \Delta x_{n''} & 2(\Delta x_{n''} + \Delta x_{n'}) & \end{bmatrix}, \qquad (4.1)$$

con $n' = n-1$, $n'' = n-2$, en tanto que el vector y" queda como:

$$y'' = [y''_2, y''_3, \cdots, y''_{n'}]^T \qquad (4.2)$$

mientras y y C quedan como en las expresiones (3.21) y (3.19).

Por otro lado, la relación entre y y y' puede expresarse como la ecuación (3.22), donde los únicos cambios son en y", definida en (4.2), y en G, definida como:

$$G = \begin{bmatrix} \Delta x_1 & & & & \\ 2\Delta x_2 & \Delta x_2 & & & \\ & 2\Delta x_3 & \Delta x_3 & & \\ & & \ddots & \ddots & \\ & & & 2\Delta x_{n''} & \Delta x_{n''} \\ & & & & 2\Delta x_{n'} \\ & & & & -\Delta x_{n'} \end{bmatrix} \tag{4.3}$$

Las matrices A y G son de $nx(n-2)$, mientras que las matrices C y F, de nxn.

Si y'' es una condición prescrita, y no puede despejarse de la ecuación (3.17) por que la matriz C es singular. De hecho, C es de rango $n-1$, lo cual significa, como ya se ha explicado en capítulos anteriores, que la dimensión de su espacio nulo es igual a 1[17].

Para resolver el sistema (3.17) debe introducirse una condición extra sobre el vector y. Esta condición puede ser, por ejemplo, que el valor medio de y sea cero, es decir:

$$y_1 + y_2 + y_3 + \ldots y_n = 0 \tag{4.4}$$

Esta ecuación tambien puede expresarse como

$$d^T y = 0 \tag{4.5}$$

donde d es un vector de dimensión n definido como

$$d = [1, 1, 1, \ldots, 1]^T \tag{4.6}$$

Introduciendo la condición (4.4) en la ecuación (3.17) el sistema resulta

$$\begin{bmatrix} C_1 \\ \hline d^T \end{bmatrix} y = \frac{1}{6} \begin{bmatrix} A_1 \\ \hline 0^T \end{bmatrix} y'' \tag{4.7}$$

donde A_1 y C_1 son los primeros $n-1$ renglones linealmente independientes de A y de C, respectivamente. Además, 0 es el vector cero de dimensión $n-2$.

El sistema (4.7) es no singular, por lo que y puede despejarse de ahí, en la siguiente forma:

$$y = \frac{1}{6}\begin{bmatrix} C_1 \\ d^T \end{bmatrix}^{-1}\begin{bmatrix} A_1 \\ 0^T \end{bmatrix} y'' \tag{4.8}$$

La pendiente en los puntos de apoyo de la *spline* se puede obtener de la ecuación (3.22), junto con la ecuación (3.17), para hallar

$$y' = (F - GA^{-1}C)\, y \tag{4.9}$$

Este sistema también es singular, de rango $n-1$. De hecho, las matrices F y C tienen el mismo espacio nulo, a saber, el espacio de vectores múltiplos del vector de dimensión n cuyos componentes son todos idénticos. Así, una posibilidad para poder despejar y de la ecuación (4.9), dado y', es introducir la ecuación (4.4).

En una función *spline* periódica, $y_n = y_1$, $y'_n = y'_1$ y $y''_n = y''_1$, por lo que no es necesario que la dimensión de los vectores y y y" sea n. Esta, como ya se vio, puede suponerse de $n-1$, en tanto que las matices A, C, F, y G, de las ecuaciones (3.17) y (3.22), son de $(n-1) \times (n-1)$ y están dadas a continuación:

$$A = \begin{bmatrix} 2(\Delta x_n+\Delta x_1) & \Delta x_1 & 0 & \Delta x_{n'} \\ \Delta x_1 & 2(\Delta x_1+\Delta x_2) & \Delta x_2 & \\ & \ddots & \ddots & \ddots \\ & & \Delta x_{n''} & \Delta x_{n''} \\ \Delta x_{n'} & & \Delta x_{n''} & 2(\Delta x_{n''}+\Delta x_{n'}) \end{bmatrix} \tag{4.10}$$

$$C = \begin{bmatrix} -\left(\frac{1}{\Delta x_{n'}} + \frac{1}{\Delta x_1}\right) & \frac{1}{\Delta x_1} & & & \frac{1}{\Delta x_{n'}} \\ \frac{1}{\Delta x_1} & -\left(\frac{1}{\Delta x_1} + \frac{1}{\Delta x_2}\right) & \frac{1}{\Delta x_2} & & \\ & \ddots & \ddots & \ddots & \\ & & & & \frac{1}{\Delta x_{n''}} \\ \frac{1}{\Delta x_{n'}} & & & \frac{1}{\Delta x_{n''}} & -\left(\frac{1}{\Delta x_{n''}} + \frac{1}{\Delta x_{n'}}\right) \end{bmatrix} \qquad (4.11)$$

$$y = [y_1, y_2, \ldots, y_{n'}]^T \qquad (4.12)$$

$$y' = [y'_1, y'_2, \ldots, y'_{n'}]^T \qquad (4.13)$$

$$y'' = [y''_1, y''_2, \ldots, y''_{n'}]^T \qquad (4.14)$$

$$F = \begin{bmatrix} -\frac{1}{\Delta x_1} & \frac{1}{\Delta x_1} & & & \\ & -\frac{1}{\Delta x_2} & \frac{1}{\Delta x_2} & & \\ & & \ddots & \ddots & \\ & & & & \frac{1}{\Delta x_{n''}} \\ \frac{1}{\Delta x_{n'}} & & & & -\frac{1}{\Delta x_{n'}} \end{bmatrix} \qquad (4.15)$$

$$G = \begin{bmatrix} 2\Delta x_1 & \Delta x_1 & & & \\ & 2\Delta x_2 & \Delta x_2 & & \\ & & \ddots & \ddots & \\ & & & \Delta x_{n''} & \\ \Delta x_{n'} & & & 2\Delta x_{n'} & \end{bmatrix} \qquad (4.16)$$

Cuando se prescribe y" en las funciones *spline* periódicas, el sistema (3.17) se puede resolver por medio de la condición (4.4) o, por otro lado, dando $m < n$ variables independientes z_i, $i = 1, 2, \ldots, m$, que definan las $n-1$ variables y_i. Esto se puede lograr imponiendo simetrías en la *spline* buscada. En este caso se tendrá

$$y = Wz \qquad (4.17)$$

donde W es una matriz dada de dimensión $(n-1)\ xm$. La siguiente relación también puede introducirse

$$y'' = Vz''$$

donde V es también una matriz dada de dimensión $(n-1)x\ m$. Introduciendo estas condiciones en (3.17) se tendrá

$$AVz'' = 6\ CWz$$

el cual es un sistema de $n-1$ ecuaciones en las m variables $z_i, i = 1, 2, \ldots, m$. Sin embargo, m de las $n-1$ ecuaciones son linealmente independientes; por esta razón se puede despejar z de ese sistema.

Análogamente, si se especifica la pendiente y' en los puntos de apoyo de la *spline*, se utiliza la ecuación (3.22) con una condición (4.4) o con la condición (4.17)

$$y' = Uz'$$

donde U es una matriz dada de dimensión $(n-1)\ x\ m$.

El problema se reduce entonces, a resolver un sistema lineal de ecuaciones, existiendo muchos programas que ejecutan la solución del sistema eficientemente [4,18-19]

Para ilustrar la síntesis de una función *spline* natural se introduce el siguiente ejemplo.

En el Instituto de Ingeniería [8] se tomaron medidas de la pendiente de un espeso parabólico y'_{Ri}, $i = 1, 2, \ldots, 32$, en una muestra de abscisas x_i, $i = 1, 2, \ldots, 32$, que se almacenan en un vector x.

Estos valores se compraron con los correspondientes y'_{Ni}, $i = 1,2,\ldots,32$, (Véase Fig. 4.1) obtenidos de la ecuación

$$y'_{Ni} = \frac{x_i}{2f}, \quad \text{con } f = 625 \text{ mm.} \tag{4.18}$$

Los valores del error de la pendiente y'_{Ei} se calculan como

$$y'_{Ei} = y'_{Ni} - y'_{Ri} \qquad i = 1, 2, \ldots, 32$$

y se almacenan en un vector y'_E. Los valores de x y y'_E se muestran en la tabla 4.1.

Tabla 4.1 Puntos muestra del espejo parabólico

x	y'_E
-86.2400	000.0046
-81.1600	000.0055
-76.0800	000.0081
-71.0000	000.0000
-65.9200	-00.0045
-60.8400	000.0008
-55.7600	000.0098
-50.6800	000.0100
-45.6000	000.0100
-40.5200	000.0100
-35.4400	000.0100
-30.3600	000.0100
-25.2800	000.0100
-20.2000	000.0057
-15.1200	000.0008
-10.0400	000.0100
006.2500	-00.0120
011.3300	000.0000
016.4100	000.0100
021.4900	000.0100
026.5700	000.0060
031.6500	000.0100
036.7300	000.0084

041.8100	000.0072
046.8900	000.0013
051.9700	000.0000
057.0500	000.0000
062.1300	000.0049
067.2100	000.0062
072.2900	000.0078
077.3700	000.0082
082.4500	000.0095

Conociendo el error en la pendiente y'_E , el error en la ordenada y_E , se puede obtener de la ecuación (4.9).

El error y_E calculado se encuentra en la gráfica de la Fig. 4.2 y en la tabla 4.2

Tabla 4.2 . Error y_E en las ordenadas del espejo parabolico.

x	y_E
-86.2400	-0.4571
-81.1600	-0.4322
-76.0800	-0.3983
-71.0000	-0.3680
-65.9200	-0.3922
-60.8400	-0.3971
-55.7600	-0.3778
-50.6800	-0.3125
-45.6000	-0.2766
-40.5200	-0.2109
-35.4400	-0.1750
-30.3600	-0.1093
-25.2800	-0.0734
-20.2000	-0.0149
-15.1200	-0.0165
-10.0400	0.0171
6.2500	0.0779
11.3300	0.0240
16.4100	0.0745
21.4900	0.1086
26.5700	0.1694
31.6500	0.1831
36.7300	0.2615
41.8100	0.2692
46.8900	0.3267
51.9700	0.2902
57.0500	0.3289
62.1300	0.2985
67.2100	0.3726
72.2900	0.3620
77.3700	0.4498
82.4500	0.3978

Es necesario hacer notar que

$$\frac{\partial y''}{\partial y} = 6A^{-1}C$$

$$\frac{\partial y'}{\partial y} = F - G\,A^{-1}C$$

por lo que en los programas se calculan las derivadas parciales de y' y de y'' con respecto a y.

A continuación se incluye un segundo ejemplo.

Se pretende determinar el desplazamiento del seguidor de una leva a partir de la forma dada en la Fig. 4.3, mediante una *spline* periódica $y = y(x)$.

En esa figura, y corresponde al movimiento del seguidor, mientras que x, al ángulo de rotación de la leva. Se requiere que la curva R_1 conecte D_1 con D_2, siendo tangente a estas rectas en $x=x_a$ y $x=x_b$, respectivamente. Esto equivale a decir que la velocidad del seguidor sea continua en estos puntos. Además, siendo la aceleración del seguidor una función lineal de $y'(x)$, para asegurar la continuidad de la aceleración en $x=x_a$ y $x=x_b$, $y''(x)$ debe de anularse en estos puntos. También se desea que la aceleración cambie continuamente entre x_a y x_b ; por esta razón se especifica un conjunto de valores de $y''(x)$ en el conjunto de puntos x_i, $i = 1, 2, \ldots, n$, donde $x_a = x_1 < x_2 < x_3 < \ldots < x_{n-1} < x_n = x_b$.

Entre los procedimientos empleados para determinar R_1 se pueden citar los basados en funciones armónicas, cicloidales y polinómicas [2] . Todas éstas presentan el inconveniente de que cuentan con un número muy reducido de parámetros por determinar. Además, la determinación de éstas se hace en base a un sistema lineal de ecuaciones, normalmente mal condicionado, esto es, que está dado por una matriz que posee un número de condición [4] muy alto.

Introduciendo una *spline* con doble simetría, el problema se puede resolver en forma computacionalmente muy eficiente, como se describe a continuación:

Considérese la curva de la Fig. 4.4. Se supone que la curva de esta figura posee las siguientes simetrías: dentro de los intervalos $[x_1, x_{2m-1}]$ y $[x_{2m-1}, x_{4m-3}]$, hay una simetría con respecto a L_1 y L_3, respectivamente.

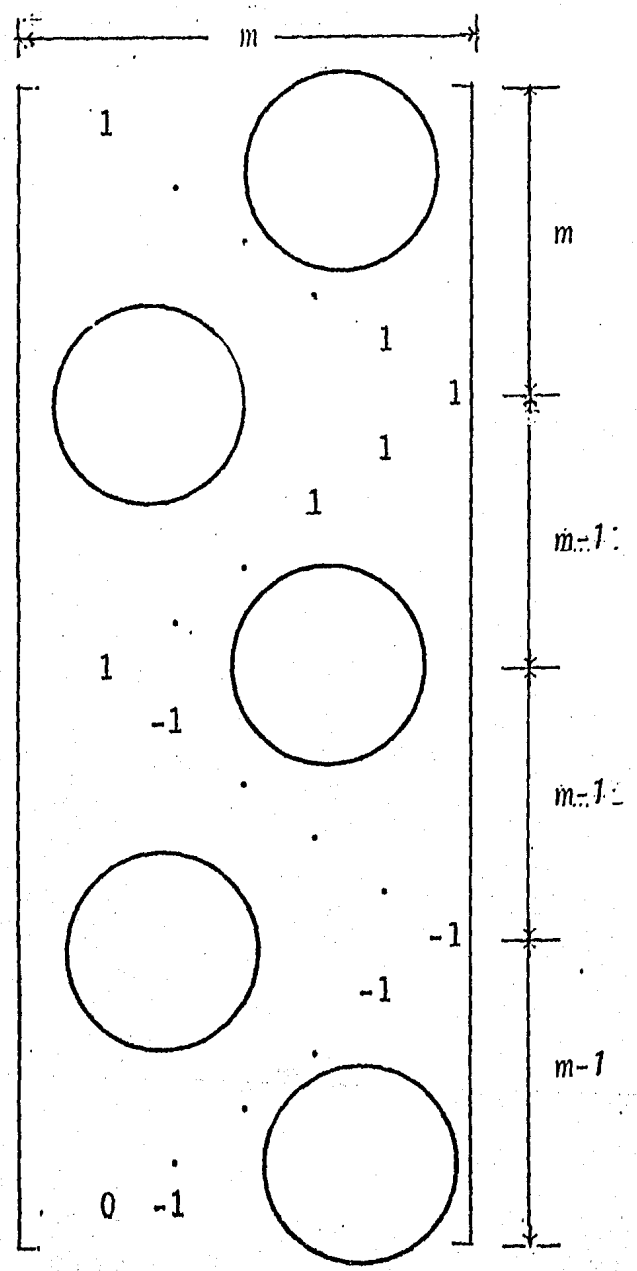
Además, dentro del intervalo $[x_m, x_{3m-2}]$, hay una antimetría con respecto a L_2. De aquí, las ordenadas de los puntos de apoyo, $y_i, i = 1, 2, \ldots, n$ dependen sólo de m variables, z_i, $i = 1, 2, \ldots, m$, donde $n=4m-3$. Además, la relación entre z_i y y_i esta definida por

$$y = Wz \quad , \quad y'' = Vz''$$

donde

$$y = [y_1, y_2, y_3, \ldots, y_n,]^T \; ; \qquad y'' = [y_1'', y_2'', y_3'', \ldots, y_n'',]^T$$
$$z = [z_1, z_2, z_3, \ldots, z_m\ ]^T \; , \qquad z'' = [z_1'', z_2'', z_3'', \ldots, z_m''\ ]^T$$

$W = y =$



Ahora, ya que los valores de z''_i, $i = 1, 2, \ldots, m$ se especifican de manera tal que se anulen en x_1 y en x_m, también y' se anula en x_m por la simetría impuesta en $[x_1, x_{2m-1}]$. Así, una curva adecuada para conectar D_1 y D_2 es una sección de la *spline* periódica comprendida entre los puntos A y B de la Fig 4.4. Además los tramos D_2 y D_3 de la Fig 4.3, se puede unir con R_2, que es una imagen de espejo de R_1, escalada adecuadamente.

Los subintervalos $[x_i, x_{i+1}]$, $i = 1, 2, \ldots, m-1$ pueden suponerse de cualquier longitud Δx_i. Entonces, supóngase

$$\Delta x_i = (x_m - x_1) / (m-1) \qquad i = 1, 2, \ldots, n'$$

$$z''_i = \text{sen} \frac{\pi (i-1) \Delta x_i}{x_m}$$

Se resolvió el problema para $m=11$, $x_m = 0.25$, $x_1 = 0$, con lo que se obtuvo la gráfica de la Fig. 4.5, y la tabla 4.3.

Tabla 4.3 m puntos de apoyo de la *spline* periódica.

	x	y
1	0.0000	0.0000
2	0.0250	-0.0039
3	0.0500	-0.0076
4	0.0750	-0.0110
5	0.1000	-0.0139
6	0.1250	-0.0161
7	0.1500	-0.0178
8	0.1750	-0.0189
9	0.2000	-0.0195
10	0.2250	-0.0197
11	0.2500	-0.0197

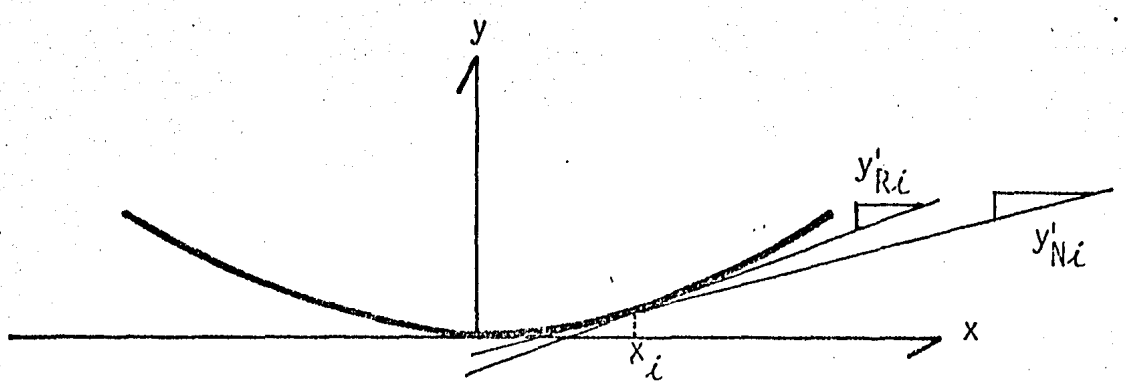


Fig 4.1 Representación de los puntos muestra del espejo parabólico y medición del error en la pendiente.

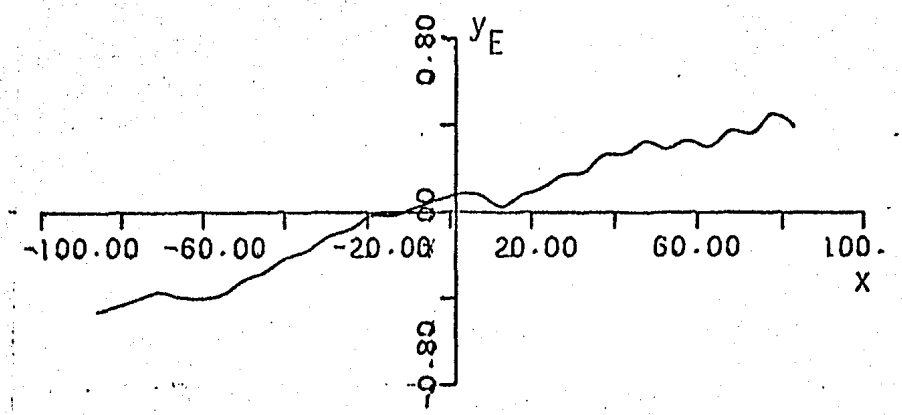


Fig 4.2 Error y_E en las ordenadas del espejo parabolico.

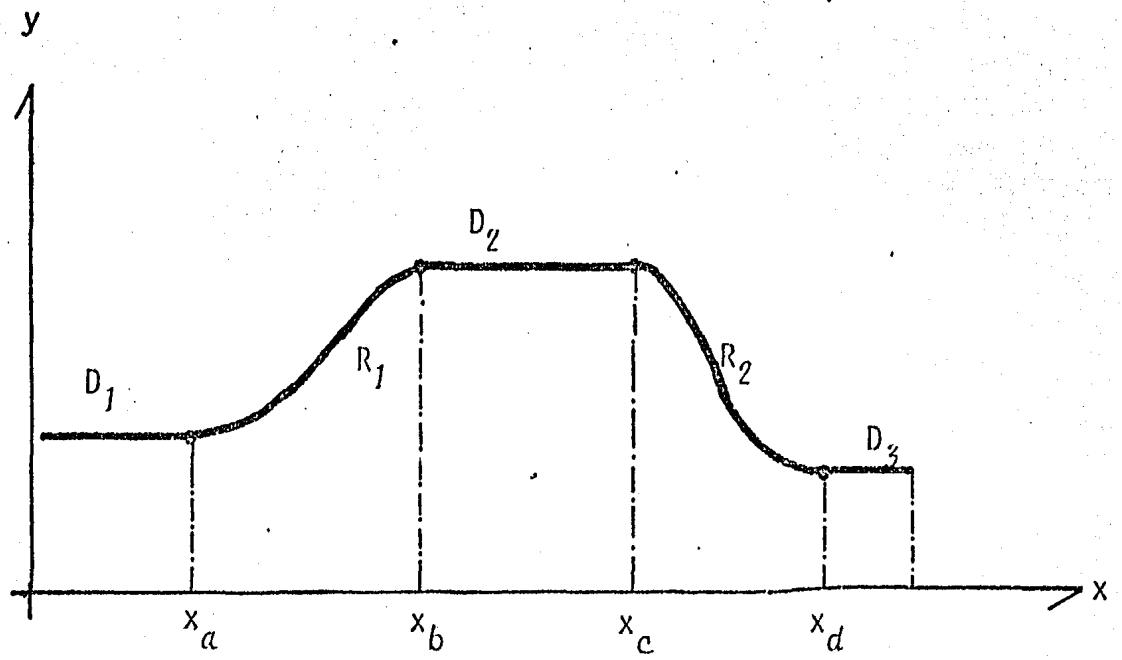


Fig 4.3 Programa de desplazamiento del seguidor de una leva

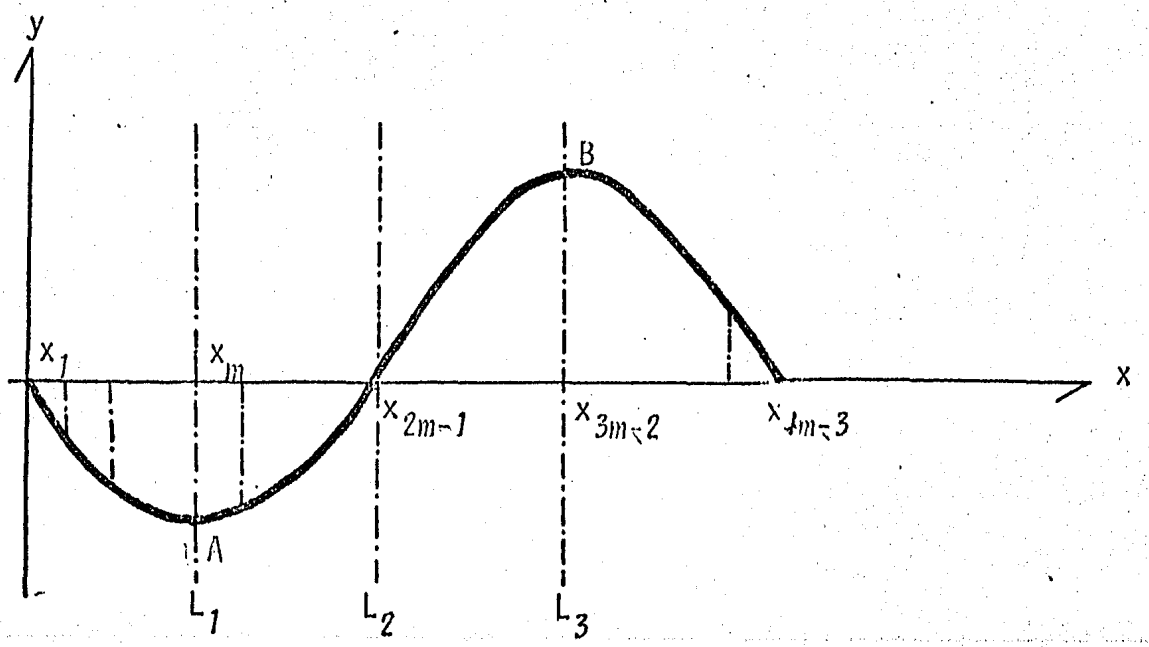


Fig 4.4 Función periódica propuesta con doble simetría.

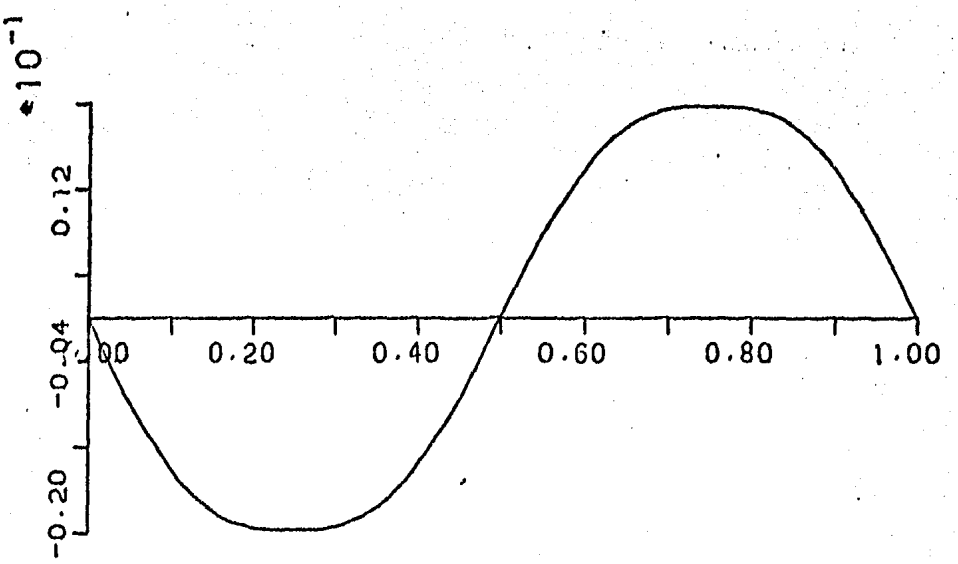


Fig 4.5 *Spline* periódica obtenida con doble simetría

5. SINTESIS DE CURVAS MEDIANTE *SPLINES* PARAMETRICAS, NATURALES Y PERIODICAS.

En este capítulo se resuelve el problema de síntesis de curvas geométricas, a diferencia del Cap. 4, en el que se resuelve el de síntesis de funciones. Las curvas geométricas pueden cerrarse, cruzarse, tener picos, etc., no así las que representan funciones continuas y diferenciables. Se supone, en este capítulo, que las coordenadas cartesianas (x,y) de un punto P de la curva de interés, están dadas en forma paramétrica por;

$$x = x(t), \qquad y = y(t) \tag{5.1}$$

donde t es un parámetro real. En el caso de las curvas abiertas, x y y son aproximadas por *splines* naturales, en tanto que, en el de curvas cerradas, por *splines* periódicas. Las funciones *spline*, tanto naturales como periódicas, ya se estudiaron en el Cap. 4.

En el caso de las *splines* naturales paramétricas, hay que determinar una curva dada en forma paramétrica por (5.1), donde x y y son funcionales de t, que satisfacen condiciones prescritas de sus propiedades geométricas locales como pendiente, curvatura, etc.

Para explicar las relaciones entre x, $\dot{x}$ y $\ddot{x}$ y entre y, $\dot{y}$ y $\ddot{y}$ es necesario definir

$$x = [x_1, x_2, \ldots, x_n]^T \qquad y = [y_1, y_2, \ldots, y_n]^T \tag{5.2a}$$

$$\dot{x} = [\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n]^T \qquad \dot{y} = [\dot{y}_1, \dot{y}_2, \ldots, \dot{y}_n]^T \tag{5.2b}$$

$$\ddot{x} = [\ddot{x}_2, \ddot{x}_3, \ldots, \ddot{x}_n]^T \qquad \ddot{y} = [\ddot{y}_2, \ddot{y}_3, \ldots, \ddot{y}_n]^T \tag{5.2c}$$

donde x, $\dot{x}$, y y $\dot{y}$ son de dimensión n en tanto que $\ddot{x}$ y $\ddot{y}$, de dimensión $(n-2)$.

Las relaciones entre x y $\ddot{x}$ y entre y y $\ddot{y}$ estan dadas por la ecuación (3,17), a saber

$$A\ddot{x} = 6Cx, \qquad A\ddot{y} = 6Cy \tag{5.3}$$

Sin embargo a diferencia de la ecuación (3.17) las matrices A y C, son funciones de x_i y y_i y, para condiciones de una *spline* natural, están definidas como:

$$A = \begin{bmatrix} 2(\Delta t_1+\Delta t_2) & \Delta t_2 & & & \\ \Delta t_2 & 2(\Delta t_2+\Delta t_3) & \Delta t_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \Delta t_{n''} & 2(\Delta t_{n''}+\Delta t_{n'}) & \end{bmatrix} \tag{5.4}$$

$$C = \begin{bmatrix} \frac{1}{\Delta t_1} & -\left(\frac{1}{\Delta t_1}+\frac{1}{\Delta t_2}\right) & \frac{1}{\Delta t_2} & & \\ & \frac{1}{\Delta t_2} & -\left(\frac{1}{\Delta t_2}+\frac{1}{\Delta t_3}\right) & \frac{1}{\Delta t_3} & \\ & & \ddots & \ddots & \ddots \\ & & & \frac{1}{\Delta t_{n''}} & -\left(\frac{1}{\Delta t_{n''}}+\frac{1}{\Delta t_{n'}}\right) \quad \frac{1}{\Delta t_{n'}} \end{bmatrix} \tag{5.5}$$

cuyas dimensiones son:

$$A : (n-2) \times (n-2) \qquad C : (n-2) \times n$$

Con Δt_i definido en (3.26). Los vectores de la expresión (5.3) son (5.2a) y (5.2c).

Análogamente, las relaciones entre x y $\dot{x}$ y entre y y $\dot{y}$ están dadas por:

$$\dot{x} = Fx - \frac{1}{6} G\ddot{x}, \qquad \dot{y} = Fy - \frac{1}{6} G\ddot{y} \tag{5.6}$$

con F y G definidas como

$$F = \begin{bmatrix} \frac{1}{\Delta t_1} & \frac{1}{\Delta t_1} & & & & \\ & \frac{1}{\Delta t_2} & \frac{1}{\Delta t_2} & & & \\ & & \ddots & \ddots & & \\ & & & & \frac{1}{\Delta t_{n'}} & \frac{1}{\Delta t_{n'}} \\ & & & & \frac{1}{\Delta t_{n'}} & \frac{1}{\Delta t_{n'}} \end{bmatrix} \tag{5.7}$$

$$G = \begin{bmatrix} \Delta t_1 & & & & & \\ 2\Delta t_2 & \Delta t_2 & & & & \\ & 2\Delta t_3 & \Delta t_3 & & & \\ & & \ddots & \ddots & & \\ & & & 2\Delta t_{n''} & \Delta t_{n''} & \\ & & & & \Delta t_{n'} & \\ & & & & -\Delta t_{n'} & \end{bmatrix} \tag{5.8}$$

con dimensiones de nxn y $nx(n-2)$, respectivamente. Los vectores x y y y los vectores $\dot{x}$ y $\dot{y}$ ya se definieron en (5.2a) y (5.2b)

Las ecuaciones (5.3) y (5.6) son, en consecuencia, no lineales, por lo que resulta claro que en el problema de síntesis de curvas con valores prescritos de $\dot{x}$, $\dot{y}$, $\ddot{x}$ y $\ddot{y}$, o una combinación de éstos, como por ejemplo la curvatura, conduce a un sistema no lineal de ecuaciones.

Entónces, si se tienen valores prescritos de curvatura ϕ_i, $i = 1,2,\ldots,n$, se tiene un sistema de ecuaciones sobre un conjunto de funciones f_i de la forma

$$f_i = f(t_i, x_i, y_i, \dot{x}_i, \dot{y}_i, \ddot{x}_i, \ddot{y}_i) \qquad (5.9)$$

Definiendo los vectores ϕ y f de dimension (n-2) como aquellos cuyos i-ésimos componentes son ϕ_i y f_i ; $i = 1, 2, \ldots, n$, respectivamente, el problema consiste en resolver un sistema algebraico no lineal de la forma

$$f(x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}) - \phi = 0 \qquad (5.10)$$

donde el parametro t no ha sisdo incluido explícitamente ; pero aparece implícitamente en todas las variables que tiene f como argumento. El vector ϕ es, desde luego, constante.

Un medio muy eficiente para resolver el sistama no lineal (5.10) es por aplicación del método de *Newton-Raphson* [19]. Este método , sin embargo requiere de la matriz *jacobiana* del sistema (5.10).

El desarrollo algebraico que se da adelante es con el proposito de obtener la matriz *jacobiana* para las *splines* naturales paramétricas.

En primera instancia se varía (5.3), obteniendo

$$\delta A.\ddot{x} + A\delta\ddot{x} = 6(\delta C.x + C\delta x) \qquad (5.11a)$$

$$\delta A.\ddot{y} + A\delta\ddot{y} = 6(\delta C.y + C\delta y) \qquad (5.11b)$$

Para determinar el termino $\delta A.\ddot{x}$ de la expresión (5.11a) es necesario desarrollar el producto $A.\ddot{x}$ de (5.3), lo cual produce

$$A\ddot{X} = \begin{bmatrix} 2(\Delta t_1 + \Delta t_2)\ddot{X}_2 + \Delta t_2 \ddot{X}_3 \\ \Delta t_2\ddot{X}_2 + 2(\Delta t_2 + \Delta t_3)\ddot{X}_3 + \Delta t_3\ddot{X}_4 \\ \Delta t_3\ddot{X}_3 + 2(\Delta t_3 + \Delta t_4)\ddot{X}_4 + \Delta t_4\ddot{X}_5 \\ \vdots \\ \Delta t_{n''}\ddot{X}_{n''} + 2(\Delta t_{n''} + \Delta t_{n'})\ddot{X}_{n'} + \Delta t_{n'}\ddot{X}_n \end{bmatrix} \qquad (5.12)$$

Ahora se varían sólo los términos de A en (5.12)

$$\delta A.\ \ddot{X} = \begin{bmatrix} 2\ddot{X}_2[c_1\delta(x_2-x_1) + c_2\delta(x_3-x_2)] + \ddot{X}_3 c_2\delta(x_3-x_2) \\ \ddot{X}_2 c_2\delta(x_3-x_2) + 2\ddot{X}_3(x_3-x_2) + c_3\delta(x_4-x_3) + \ddot{X}_4 c_3\delta(x_4-x_3) \\ \ddot{X}_3 c_3\delta(x_4-x_3) + 2\ddot{X}_4(x_4-x_3) + c_4\delta(x_5-x_4) + \ddot{X}_5 c_4\delta(x_5-x_4) \\ \vdots \\ \ddot{X}_{n''} c_{n''}\delta(x_{n'}-x_{n''}) + 2\ddot{X}_{n'}[c_{n''}\delta(x_{n'}-x_{n''}) + c_{n'}\delta(x_n-x_{n'})] \end{bmatrix} +$$

$$+ \begin{bmatrix} 2\ddot{X}_2[s_1\delta(y_2-y_1) + s_2\delta(y_3-y_2)] + \ddot{X}_3 s_2\delta(y_3-y_2) \\ \ddot{X}_2 s_2\delta(y_3-y_2) + 2\ddot{X}_3[s_2\delta(y_3-y_2) + s_3\delta(y_4-y_3)] + \ddot{X}_4 s_3\delta(y_4-y_3) \\ \ddot{X}_3 s_3\delta(y_4-y_3) + 2\ddot{X}_4[s_3\delta(y_4-y_3) + s_4\delta(y_5-y_4)] + \ddot{X}_5 s_4\delta(y_5-y_4) \\ \vdots \\ \ddot{X}_{n''} s_{n''}\delta(y_{n'}-y_{n''}) + 2\ddot{X}_{n'}[s_{n''}\delta(y_{n'}-y_{n''}) + s_{n'}\delta(y_n-y_{n'})] \end{bmatrix} \qquad (5.13)$$

donde:

$$c_i = \frac{\Delta x_i}{\Delta t_i} \quad , \quad s_i = \frac{\Delta y_i}{\Delta t_i} \quad , \quad i = 1,2,\ldots,n \qquad (5.14)$$

Además, es necesario tomar en cuenta que

$$\delta\Delta x_i = \delta x_{i+1} - \delta x_i, \qquad \delta\Delta y_i = \delta y_{i+1} - \delta y_i, \tag{5.15}$$

$$i = 1, 2, \ldots, n$$

con lo que

$$\delta\Delta t_i = c_i\, \delta\Delta x_i + s_i\, \delta\Delta y_i, \qquad i = 1, 2, \ldots, n \tag{5.16}$$

Sustituyendo (5.15) en (5.13)

$$\delta A.\ddot{X} = \begin{bmatrix} 2\ddot{x}_2\,(c_1\delta\Delta x_1 + c_2\delta\Delta x_2) + \ddot{x}_3\, c_2\delta\Delta x_2 \\ \ddot{x}_2 c_2\delta\Delta x_2 + 2\ddot{x}_3\,(c_2\delta\Delta x_2 + c_3\delta\Delta x_3) + \ddot{x}_4 c_3\delta\Delta x_3 \\ \ddot{x}_3 c_3\delta\Delta x_3 + 2\ddot{x}_4\,(c_3\delta\Delta x_3 + c_4\delta\Delta x_4) + \ddot{x}_5 c_4\delta\Delta x_4 \\ \vdots \\ \ddot{x}_{n''} c_{n''}\delta\Delta x_{n''} + 2\ddot{x}_{n'}\,(c_{n''}\delta\Delta x_{n''} + c_{n'}\delta\Delta x_{n'}) + \ddot{x}_n c_{n'}\delta\Delta x_{n'} \end{bmatrix} +$$

$$+ \begin{bmatrix} 2\ddot{x}_2\,(s_1\delta\Delta y_1 + s_2\delta\Delta y_2) + \ddot{x}_3\, s_2\delta\Delta y_2 \\ \ddot{x}_2 s_2\delta\Delta x_2 + 2\ddot{x}_3\,(s_2\delta\Delta y_2 + s_3\delta\Delta y_3) + \ddot{x}_4 s_3\delta\Delta y_3 \\ \ddot{x}_3 s_3\delta\Delta y_3 + 2\ddot{x}_4\,(s_3\delta\Delta y_3 + s_4\delta\Delta y_4) + \ddot{x}_5 s_4 x_4\delta\Delta y_4 \\ \vdots \\ \ddot{x}_{n''} s_{n''}\delta\Delta y_{n''} + 2\ddot{x}_{n'}\,(s_{n''}\delta\Delta y_{n''} + s_{n'}\delta\Delta y_{n'}) + \ddot{x}_n s_{n'}\delta\Delta y_{n'} \end{bmatrix} \tag{5.17}$$

Factorizando $\delta\Delta x_i$, $\delta\Delta y_i$ en (5.17)

$$\delta A.\ddot{X} = \begin{bmatrix} 2\ddot{x}_2 c_1 & (2\ddot{x}_2+\ddot{x}_3)\, c_2 & & & & \\ (\ddot{x}_2+2\ddot{x}_3)\, c_2 & (2\ddot{x}_3+\ddot{x}_4)\, c_3 & & & & \\ & (\ddot{x}_3+2\ddot{x}_4)\, c_3 & (2\ddot{x}_4+\ddot{x}_5)\, c_4 & & & \\ & & \ddots & \ddots & & \\ & & & & (\ddot{x}_{n''}+2\ddot{x}_{n'})\, c_{n''} & 2\ddot{x}_{n'} c_{n'} \end{bmatrix} \begin{bmatrix} \delta\Delta x_1 \\ \delta\Delta x_2 \\ \delta\Delta x_3 \\ \vdots \\ \delta\Delta x_{n'} \end{bmatrix}$$

(5.18)

$$+ \begin{bmatrix} 2\ddot{x}_2 s_1 & (2\ddot{x}_2+\ddot{x}_3)\, s_2 & & & & \\ (\ddot{x}_2+2\ddot{x}_3)\, s_2 & (2\ddot{x}_3+\ddot{x}_4)\, s_3 & & & & \\ & (\ddot{x}_3+2\ddot{x}_4)\, s_3 & (2\ddot{x}_4+\ddot{x}_5)\, s_4 & & & \\ & & \ddots & \ddots & & \\ & & & & (\ddot{x}_{n''}+2\ddot{x}_{n'})\, s_{n''} & 2\ddot{x}_{n'} s_{n'} \end{bmatrix} \begin{bmatrix} \delta\Delta y_1 \\ \delta\Delta y_2 \\ \delta\Delta y_3 \\ \vdots \\ \delta\Delta y_{n'} \end{bmatrix}$$

Hasta el momento se puede escribir $\delta A.\ddot{X}$ como:

$$\delta A.\ddot{X} = A_{11}\ \delta\Delta x + A_{12}\ \delta\Delta y \qquad (5.19)$$

Los terminos $\delta\Delta x$ y $\delta\Delta y$ se pueden expresar como transformaciones lineales de δx y δy, representados como se indica a continuación

$$\delta\Delta x = J'.\delta x = \begin{bmatrix} x_2-x_1 \\ x_3-x_2 \\ x_4-x_3 \\ \vdots \\ x_n-x_{n'} \end{bmatrix} = \begin{bmatrix} -1 & 1 & & & \\ 0 & -1 & 1 & & \\ & & -1 & \ddots & \\ & & & \ddots & 1 \\ & & & & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = J'\delta x \qquad (5.20)$$

$$\delta\Delta y = \begin{bmatrix} y_2-y_1 \\ y_3-y_2 \\ y_4-y_5 \\ \vdots \\ y_n-y_{n'} \end{bmatrix} = \delta \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \\ & & & \ddots & \ddots \\ & & & & -1 \quad 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = J'\delta y \qquad (5.20b)$$

Ahora se define

$$A'_{11} = A_{11}.J' \quad , \quad A'_{12} = A_{12}.J' \qquad (5.21)$$

Sustituyendo (5.21) en (5.19) se obtiene

$$\delta A.\ddot{X} = A'_{11}\,\delta x + A'_{12}\,\delta y \qquad (5.22)$$

Análogamente, el término $\delta A\ddot{y}$ de la expresión (5.11b) se puede escribir como

$$\delta A\ddot{y} = A'_{21}\,\delta x + A'_{22}\,\delta \ddot{y} \qquad (5.23)$$

Las matrices A'_{11} , A'_{12} , A'_{21} , A'_{22} se muestran explícitamaente en las ecs (5.24) y (5.25)

$$A'_{11}=\begin{bmatrix} -2\ddot{x}_2c_1 & 2\ddot{x}_2c_1-(2\ddot{x}_2+\ddot{x}_3)c_2 & (2\ddot{x}_2+\ddot{x}_3)c_2 & & \\ & -(\ddot{x}_2+2\ddot{x}_3)c_2 & (\ddot{x}_2+2\ddot{x}_3)c_2-(2\ddot{x}_3+\ddot{x}_4)c_3 & (2\ddot{x}_3+\ddot{x}_4)c_3 & \\ & & \ddots & \ddots & \ddots \\ & & -(\ddot{x}_{n''}+2\ddot{x}_{n'})c_{n''} & (\ddot{x}_{n''}+\ddot{x}_{n'})c_{n''}-2\ddot{x}_{n'}c_{n'} & 2\ddot{x}_{n'} \quad c_{n'} \end{bmatrix} \tag{5.24}$$

$$A'_{12}=\begin{bmatrix} -2\ddot{x}_2s_1 & 2\ddot{x}_2s_1-(2\ddot{x}_2+\ddot{x}_3)s_2 & (2\ddot{x}_2+\ddot{x}_3)s_2 & & \\ & -(\ddot{x}_2+2\ddot{x}_3)s_2 & (\ddot{x}_2+2\ddot{x}_3)s_2-2(\ddot{x}_3+\ddot{x}_4)s_3 & (2\ddot{x}_3+\ddot{x}_4)s_3 & \\ & & \ddots & \ddots & \ddots \\ & & -(\ddot{x}_{n''}+2\ddot{x}_{n'})s_{n''} & (\ddot{x}_{n''}+2\ddot{x}_{n'})s_{n''}-2\ddot{x}_{n'}s_{n'} & 2\ddot{x}_{n'} \quad s_{n'} \end{bmatrix}$$

$$A'_{21} = \begin{bmatrix} -2\ddot{y}_2\ c_1 & 2\ddot{y}_2 c_1-(2\ddot{y}_2+\ddot{y}_3)\ c_2 & (2\ddot{y}_2+\ddot{y}_3)\ c_2 & & \\ & -(\ddot{y}_2+2\ddot{y}_3)\ c_2 & (\ddot{y}_2+2\ddot{y}_3)\ c_2-(2\ddot{y}_3+\ddot{y}_4)\ c_3 & (2\ddot{y}_3+\ddot{y}_4)\ c_3 & \\ & & \ddots & \ddots & \ddots \\ & & & -(\ddot{y}_{n''}+2\ddot{y}_{n'})\ c_{n''} & (\ddot{y}_{n''}+2\ddot{y}_{n'})\ c_{n''}-2\ddot{y}_n, c_n, \quad 2\ddot{y}_n, c_n, \end{bmatrix} \tag{5.25}$$

$$A'_{22} = \begin{bmatrix} -2\ddot{y}_2\ s_1 & 2\ddot{y}_2 s_1-(2\ddot{y}_2+\ddot{y}_3)\ s_2 & (2\ddot{y}_2+\ddot{y}_3)\ s_2 & & \\ & -(\ddot{y}_2+2\ddot{y}_3)\ s_2 & (\ddot{y}_2+2\ddot{y}_3)\ s_2-(2\ddot{y}_3+\ddot{y}_4)\ s_3 & (2\ddot{y}_3+\ddot{y}_4)\ s_3 & \\ & & \ddots & \ddots & \ddots \\ & & & -(\ddot{y}_{n''}+2\ddot{y}_{n'})\ s_{n''} & (\ddot{y}_{n''}+2\ddot{y}_{n'})\ s_{n''}-2\ddot{y}_n, s_n, \quad 2\ddot{y}_n, s_n, \end{bmatrix}$$

El término $\delta C.x$ de la expresión (5.11a) se encuentra con el producto Cx de la forma

$$Cx = \begin{bmatrix} \frac{x_1}{\Delta t_1} - \left(\frac{1}{\Delta t_1} + \frac{1}{\Delta t_2}\right) x_2 + \frac{x_2}{\Delta t_2} \\ \frac{x_2}{\Delta t_2} - \left(\frac{1}{\Delta t_2} + \frac{1}{\Delta t_3}\right) x_3 + \frac{x_4}{\Delta t_3} \\ \frac{x_3}{\Delta t_3} - \left(\frac{1}{\Delta t_3} + \frac{1}{\Delta t_4}\right) x_4 + \frac{x_5}{\Delta t_4} \\ \vdots \\ \frac{x_{n''}}{\Delta t_{n''}} - \left(\frac{1}{\Delta t_{n''}} + \frac{1}{\Delta t_{n'}}\right) x_{n'} + \frac{x_n}{\Delta t_{n'}} \end{bmatrix} = \begin{bmatrix} -\frac{\Delta x_1}{\Delta t_1} + \frac{\Delta x_2}{\Delta t_2} \\ -\frac{\Delta x_2}{\Delta t_2} + \frac{\Delta x_3}{\Delta t_3} \\ -\frac{\Delta x_3}{\Delta t_3} + \frac{\Delta x_4}{\Delta t_4} \\ \vdots \\ -\frac{\Delta x_{n''}}{\Delta t_{n''}} + \frac{\Delta x_{n'}}{\Delta t_{n'}} \end{bmatrix} \tag{5.26}$$

Ahora bien, puesto que se tiene que

$$\delta \frac{1}{\Delta t_i} = -\frac{1}{(\Delta t_i)^2} \delta \Delta t_i \tag{5.27}$$

Al obtener $\delta C.x$ de la ecuación (5.26), tomando en cuenta (5.16) y (5.27), se obtiene

$$\delta C.x = \begin{bmatrix} -\frac{1}{(\Delta t_1)^2}(c_1\delta\Delta x_1 + s_1\delta\Delta y_1)(x_1 - x_2) - \frac{1}{(\Delta t_2)^2}(c_2\delta\Delta x_2 + s_2\delta\Delta y_2)(x_3 - x_2) \\ -\frac{1}{(\Delta t_2)^2}(c_2\delta\Delta x_2 + s_2\delta\Delta y_2)(x_2 - x_3) - \frac{1}{(\Delta t_3)^2}(c_3\delta\Delta x_3 + s_3\delta\Delta y_3)(x_4 - x_3) \\ -\frac{1}{(\Delta t_3)^2}(c_3\delta\Delta x_3 + s_3\delta\Delta y_3)(x_3 - x_4) - \frac{1}{(\Delta t_4)^2}(c_4\delta\Delta x_4 + s_4\delta\Delta y_4)(x_5 - x_4) \\ \vdots \\ -\frac{1}{(\Delta t_{n''})^2}(c_{n''}\delta\Delta x_{n''} + s_{n''}\delta\Delta y_{n''})(x_{n''} - x_{n'}) - \frac{1}{(\Delta t_{n'})^2}(c_{n'}\delta\Delta x_{n'} + s_{n'}\delta\Delta y_{n'})(x_n - x_{n'}) \end{bmatrix} \tag{5.28}$$

Desarrollando los productos y acomodando términos, se tiene

$$\delta C.x = \begin{bmatrix} \frac{c_1 \Delta x_1}{(\Delta t_1)^2}\delta\Delta x_1 - \frac{c_2 \Delta x_2}{(\Delta t_2)^2}\delta\Delta x_2 \\ \frac{c_2 \Delta x_2}{(\Delta t_2)^2}\delta\Delta x_2 - \frac{c_3 \Delta x_3}{(\Delta t_3)^2}\delta\Delta x_3 \\ \vdots \\ \frac{c_{n''} \Delta x_{n''}}{(\Delta t_{n''})^2}\delta\Delta x_{n''} - \frac{c_{n'} \Delta x_{n'}}{(\Delta t_{n'})^2}\delta\Delta x_{n'} \end{bmatrix} + \begin{bmatrix} \frac{s_1 \Delta x_1}{(\Delta t_1)^2}\delta\Delta y_1 - \frac{s_2 \Delta x_2}{(\Delta t_2)^2}\delta\Delta y_2 \\ \frac{s_2 \Delta x_2}{(\Delta t_2)^2}\delta\Delta y_2 - \frac{s_3 \Delta x_3}{(\Delta t_3)^2}\delta\Delta y_3 \\ \vdots \\ \frac{s_{n''} \Delta x_{n''}}{(\Delta t_{n''})^2}\delta\Delta y_{n''} - \frac{s_{n'} \Delta x_{n'}}{(\Delta t_{n'})^2}\delta\Delta y_{n'} \end{bmatrix} \quad (5.29)$$

Sustituyendo (5.14) en (5.29)

$$\delta C.x = \begin{bmatrix} \frac{c_1^2}{\Delta t_1} & -\frac{c_2^2}{\Delta t_2} & & & \\ & \frac{c_2^2}{\Delta t_2} & -\frac{c_3^2}{\Delta t_3} & & \\ & & \ddots & \ddots & \\ & & & \frac{c_{n''}^2}{\Delta t_{n''}} & -\frac{c_{n'}^2}{\Delta t_{n'}} \end{bmatrix} \begin{bmatrix} \delta\Delta x_1 \\ \delta\Delta x_2 \\ \vdots \\ \delta\Delta x_{n'} \end{bmatrix} +$$

$$+\begin{bmatrix} \frac{c_1 s_1}{\Delta t_1} - \frac{c_2 s_2}{\Delta t_2} & & & \\ & \frac{c_2 s_2}{\Delta t_2} - \frac{c_3 s_3}{\Delta t_3} & & \\ & & \ddots & \\ & & & \frac{c_{n''} s_{n''}}{\Delta t_{n''}} - \frac{c_{n'} s_{n'}}{\Delta t_{n'}} \end{bmatrix} \begin{bmatrix} \delta\Delta y_1 \\ \delta\Delta y_2 \\ \vdots \\ \delta\Delta y_{n'} \end{bmatrix} \qquad (5.30)$$

La ecuación (5.30) se puede escribir, entonces, como

$$\delta C.x = C_{11}\ \delta\Delta x + C_{12}\delta\Delta y \qquad (5.31)$$

Sustituyendo (5.20a y b) en (5.31) y definiendo

$$C'_{11} = C'_{11}\ J \quad y \quad C'_{12} = C'_{12}\ J \qquad (5.32)$$

produce

$$\delta C.x = C'_{11}\ \delta x + C'_{12}\ \delta y \qquad (5.33)$$

donde

$$C'_{11} = \begin{bmatrix} -\frac{c_1^2}{\Delta t_1} & \frac{c_1^2}{\Delta t_1} + \frac{c_2^2}{\Delta t_2} & -\frac{c_2^2}{\Delta t_2} & & \\ & -\frac{c_2^2}{\Delta t_2} & \frac{c_2^2}{\Delta t_2} + \frac{c_3^2}{\Delta t_3} & -\frac{c_3^2}{\Delta t_3} & \\ & & \ddots & & \\ & & -\frac{c_{n''}^2}{\Delta t_{n''}} & \frac{c_{n''}^2}{\Delta t_{n''}} + \frac{c_{n'}^2}{\Delta t_{n'}} & -\frac{c_{n'}^2}{\Delta t_{n'}} \end{bmatrix} \qquad (5.34)$$

$$C'_{12} = \begin{bmatrix} -\frac{c_1 s_1}{\Delta t_1} & \frac{c_1 s_1}{\Delta t_1}+\frac{c_2 s_2}{\Delta t_2} & -\frac{c_2 s_2}{\Delta t_2} & & \\ & -\frac{c_2 s_2}{\Delta t_2} & \frac{c_2 s_2}{\Delta t_2}+\frac{c_3 s_3}{\Delta t_3} & \frac{c_3 s_3}{\Delta t_3} & \\ & & \ddots & \ddots & \ddots \\ & & & -\frac{c_{n''} s_{n''}}{\Delta t_{n''}} & \frac{c_{n''} s_{n''}}{\Delta t_{n''}}+\frac{c_{n'} s_{n'}}{\Delta t_{n'}} & -\frac{c_{n'} s_{n'}}{\Delta t_{n'}} \end{bmatrix} \qquad (5.35)$$

Análogamente, el término $\delta C.y$ de la expresión (5.11b) puede escribirse como

$$\delta C.y = C'_{21}\,\delta x + C'_{22}\,\delta \Delta y \qquad (5.36)$$

donde

$C'_{21} = C'_{12}$ habiéndose definido ya C'_{12} en (5.35) y C'_{22} es

$$C'_{22} = \begin{bmatrix} -\frac{s_1^2}{\Delta t_1} & \frac{s_1^2}{\Delta t_1}+\frac{s_2^2}{\Delta t_2} & -\frac{s_2^2}{\Delta t_2} & & \\ & -\frac{s_2^2}{\Delta t_2} & \frac{s_2^2}{\Delta t_2}+\frac{s_3^2}{\Delta t_3} & -\frac{s_3^2}{\Delta t_3} & \\ & & \ddots & \ddots & \ddots \\ & & & -\frac{s_{n''}^2}{\Delta t_{n''}} & \frac{s_{n''}^2}{\Delta t_{n''}}+\frac{s_{n'}^2}{\Delta t_{n'}} & -\frac{s_{n'}^2}{\Delta t_{n'}} \end{bmatrix} \qquad (5.37)$$

Con lo obtenido anteriormente se pueden sustituir (5.22), (5.23), (5.33) y (5.36) en las expresiones (5.11a y b) para llegar a

$$A'_{11}\,\delta x + A'_{12}\,\delta y + A\delta X = 6\,(C'_{11}\,\delta x + C'_{12}\,\delta y + C\delta x)$$

$$A'_{21}\,\delta x + A'_{22}\,\delta y + A\delta y = 6\,(C'_{21}\,\delta x + C'_{22}\,\delta y + C\delta y)$$

Despejando $\delta\ddot{x}$ y $\delta\ddot{y}$

$$\delta\ddot{X} = A^{-1}[6(C'_{11} + C) - A'_{11}]\,\delta x + (6\,C'_{12} - A'_{12})\,\delta y \qquad (5.38)$$

$$\delta\ddot{y} = A^{-1}[6\,C'_{21} - A'_{21})\,\delta x + [6(C'_{22}+C) - A'_{22}]\delta y \qquad (5.39)$$

de donde se pueden identificar de inmediato las siguientes derivadas:

$$\frac{\partial\ddot{x}}{\partial x} = A^{-1}[6(C'_{11}+C) - A'_{11}\,,\ \frac{\partial\ddot{x}}{\partial y} = A^{-1}(6C'_{12}-A'_{12}) \qquad (5.40)$$

$$\frac{\partial\ddot{y}}{\partial x} = A^{-1}(6C'_{21}-A'_{21}),\ \frac{\partial\ddot{y}}{\partial y} = A^{-1}[6(C'_{22}+C) - A'_{22}] \qquad (5.41)$$

Hasta el momento se obtuvieron las derivadas parciales de $\ddot{x}$ y $\ddot{y}$ con respecto a las coordenadas de los puntos de apoyo. También es necesario hallar las derivadas parciales de $\dot{x}$ y $\dot{y}$ con respecto a las coordenadas de los puntos de apoyo; para esto se hace variar (5.6) en la forma

$$\delta\dot{x} = \delta F.x + F\delta x - \frac{1}{6}\,\delta G.\ddot{x} - \frac{1}{6}\,G\delta\ddot{x} \qquad (5.42)$$

$$\delta\dot{y} = \delta F.y + F\delta y - \frac{1}{6}\,\delta G.\ddot{y} - \frac{1}{6}\,G\delta\ddot{y}$$

donde sólo falta hallar los terminos $\delta F.x$, $G.\ddot{x}$, $\delta F.y$ y $G.\ddot{y}$.

El termino $\delta F.x$ se halla variando el producto

$$Fx = \begin{bmatrix} (-x_1 + x_2)\,/\Delta t_1 \\ (-x_2 + x_3)\,/\Delta t_2 \\ (-x_3 + x_4)\,/\Delta t_3 \\ \vdots \\ (-x_{n'} + x_n)/\Delta t_{n'} \\ (-x_{n'} + x_n)/\Delta t_{n'} \end{bmatrix} \qquad (5.43)$$

Variando la expresión (5.43), sustituyendo la expresión (5.16) en (5.27) y ésta a su vez en (5.43), se obtiene

$$\delta F.x = \begin{bmatrix} -(-x_1 + x_2)\,(c_1\delta\Delta x_1 + s_1\delta\Delta y_1)/\Delta t_1^2 \\ -(-x_2 + x_3)\,(c_2\delta\Delta x_2 + s_2\delta\Delta y_2)/\Delta t_2^2 \\ \vdots \\ -(-x_{n'} + x_n)\,(c_{n'}\,\delta\Delta x_{n'} + s_{n'}\,\delta\Delta y_{n'})/\Delta t_{n'}^2 \\ -(-x_{n'} + x_n)\,(c_{n'}\,\delta\Delta x_{n'} + s_{n'}\,\delta\Delta y_{n'})/\Delta t_{n'}^2 \end{bmatrix} \qquad (5.44)$$

Sustituyendo (5,14) en (5,44) da

$$\delta F.x = \begin{bmatrix} -(c_1^2\,\delta\Delta x_1 + s_1\,c_1\,\delta\Delta y_1)/\Delta t_1 \\ -(c_2^2\,\delta\Delta x_2 + s_2\,c_2\,\delta\Delta y_2)/\Delta t_2 \\ \vdots \\ -(c_{n'}^2\,\delta\Delta x_{n'} + s_{n'}\,c_{n'}\,\delta\Delta y_{n'})/\Delta t_{n'} \\ -(c_{n'}^2\,\delta\Delta x_{n'} + s_{n'}\,c_{n'}\,\delta\Delta y_{n'})/\Delta t_{n'} \end{bmatrix}$$

Factorizando y sustituyendo (5,20a y b) en la expresión anterior,

$$\delta F.x = \begin{bmatrix} \frac{c_1^2}{\Delta t_1} & -\frac{c_1^2}{\Delta t_1} & & & \\ & \frac{c_2^2}{\Delta t_2} & -\frac{c_2^2}{\Delta t_2} & & \\ & & \ddots & \ddots & \\ & & & \frac{c_{n'}^2}{\Delta t_{n'}} & -\frac{c_{n'}^2}{\Delta t_{n'}} \\ & & & \frac{c_{n'}^2}{\Delta t_{n'}} & -\frac{c_{n'}^2}{\Delta t_{n'}} \end{bmatrix} \delta \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n'} \\ x_{n'} \end{bmatrix} +$$

$$+\begin{bmatrix} \frac{s_1c_1}{\Delta t_1} & -\frac{s_1c_1}{\Delta t_1} & & & \\ & \frac{s_2c_2}{\Delta t_2} & -\frac{s_2c_2}{\Delta t_2} & & \\ & & \ddots & \ddots & \\ & & & \frac{s_{n'}c_{n'}}{\Delta t_{n'}} & -\frac{s_{n'}c_{n'}}{\Delta t_{n'}} \\ & & & \frac{s_{n'}c_{n'}}{\Delta t_{n'}} & -\frac{s_{n'}c_{n'}}{\Delta t_{n'}} \end{bmatrix} \delta \begin{bmatrix} y_1 \\ y_2 \\ \\ \\ y_{n'} \\ \\ y_n \end{bmatrix} \qquad (5.45)$$

El sistema anterior se puede escribir como

$$\delta F_{\cdot}x = F_{11}\ \delta x + F_{12}\ \delta y \qquad (5.46)$$

Por analogía se puede hallar $\delta F.y$ como

$$\delta\ F.y = F_{21}\ \delta x + F_{22}\ \delta y \qquad (5.47)$$

donde:

$$F_{21} = F_{12}\ ,$$

F_{22} es semejante a F_{11} ; pero con s_i^2 cambiada por c_i^2 , ambas definidas en (5.14)

El termino $\delta G.\ddot{X}$ se halla desarrollando el producto

$$G\ddot{X} = \begin{bmatrix} \ddot{X}_2 \Delta t_1 \\ (2\ddot{X}_2 + \ddot{X}_3)\ \Delta t_2 \\ \vdots \\ (2\ddot{X}_{n''} + \ddot{X}_{n'})\ \Delta t_{n''} \\ 2\ddot{X}_{n'}\ \Delta t_{n'} \\ -\ \ddot{X}_{n'}\ \Delta t_{n'} \end{bmatrix} \qquad (5.48)$$

Variándolo y sustituyendo (5.14) y (5.16) en (5.48) se llega a

$$\delta G.\ddot{X}=\begin{bmatrix} \ddot{X}_2\,(c_1\delta\Delta x_1 + s_1\delta\Delta y_1) \\ (2\ddot{X}_2 + \ddot{X}_3)\,(c_2\delta\Delta x_2 + s_2\delta\Delta y_2) \\ (2\ddot{X}_3 + \ddot{X}_4)\,(c_3\delta\Delta x_3 + s_3\delta\Delta y_3) \\ \vdots \\ (2\ddot{X}_{n''} + \ddot{X}_{n'})\,(c_{n''}\,\delta\Delta x_{n''} + s_{n''}\,\delta\Delta y_{n''}) \\ 2\ddot{X}_{n'}\,(c_{n'}\,\delta\Delta x_{n'} + s_{n'}\,\delta\Delta y_{n'}) \\ -\ddot{X}_{n'}\,(c_{n'}\,\delta\Delta x_{n'} + s_{n'}\,\delta\Delta y_{n'}) \end{bmatrix} \qquad (5.49)$$

Factorizando y sustituyendo (5.20a y b) en (5.49), se obtiene:

$$\delta G.\ddot{X}=\begin{bmatrix} \ddot{X}_2c_1 & \ddot{X}_2c_1 & & & \\ & -(2\ddot{X}_2+\ddot{X}_3)c_2 & (2\ddot{X}_2+\ddot{X}_3)c_2 & & \\ & & & & \\ & & & -2\ddot{X}_{n'}c_{n'} & 2\ddot{X}_{n'}c_{n'} \\ & & & \ddot{X}_{n'}c_{n'} & -\ddot{X}_{n'}c_{n'} \end{bmatrix} \delta \begin{bmatrix} x_1 \\ x_2 \\ \\ \\ x_{n'} \\ x_n \end{bmatrix} +$$

$$+\begin{bmatrix} -\ddot{X}_2s_1 & \ddot{X}_2s_1 & & & \\ & -2(\ddot{X}_2+\ddot{X}_3)\,s_2 & (2\ddot{X}_2+\ddot{X}_3)\,s_2 & & \\ & & & & \\ & & & -2\ddot{X}_{n'}s_{n'} & 2\ddot{X}_{n'}s_{n'} \\ & & & \ddot{X}_{n'}s_{n'} & -\ddot{X}_{n'}s_{n'} \end{bmatrix} \delta \begin{bmatrix} y_1 \\ y_2 \\ \\ \\ y_{n'} \\ y_n \end{bmatrix} \qquad (5.50)$$

El sistema anterior puede escribirse como

$$\delta G.\ddot{X} = G_{11}\delta x + G_{12}\,\delta y \qquad (5.51)$$

Por analogía, se obtiene

$$\delta G.\ddot{y} = G_{21}\,\delta x + G_{22}\,\delta y \qquad (5.52)$$

donde G_{21} es semejante a G_{11}, pero con $\ddot{y}_i$ cambiando por $\ddot{x}_i$, $i = 2,3,\ldots n'$ y G_{22} es semejante a G , pero con $\ddot{y}_i$ cambiada por $\ddot{x}_i$, $i = 2,3,..n'$

Sustituyendo (5.38), (5.39), (5.45), (5.46), (5.51) y (5.52) en (5.42) se tiene

$$\delta\dot{x} = F_{11}\delta x + F_{12}\delta y + F\delta x - \frac{1}{6}\,[G_{11}\delta x + G_{12}\delta y + G\delta\ddot{x}]$$

$$\delta\dot{y} = F_{21}\delta x + F_{22}\delta y + F\delta y - \frac{1}{6}\,(G_{21}\delta x + G_{22}\delta y) -$$

$$- \frac{1}{6}\,GA^{-1}\,(6.C'_{21} - A'_{21})\,\delta x - \frac{1}{6}\,GA^{-1}\,[6\,C'_{22} + C) - A'_{22}]\,\delta y$$

Agrupando términos semejantes,

$$\delta\dot{x} = \{F_{11}+F - \frac{1}{6}\,G_{11} - \frac{1}{6}\,GA^{-1}\,[6(C'_{11} + C) - A'_{11}]\}\delta x +$$
$$+ [F_{12} - \frac{1}{6}\,G_{12} - \frac{1}{6}\,GA^{-1}\,(6C'_{12} - A'_{12}]\,\delta y$$

$$\delta\dot{y} = [F_{21} - \frac{1}{6}\,GA^{-1}(6C'_{21} - A'_{21}) - \frac{1}{6}\,G_{21}]\,\delta x +$$

$$+ \{F_{22} + F - \frac{1}{6}\,GA^{-1}\,[6(C'_{22} + C) - A'_{22}] - \frac{1}{6}\,G_{22}\}\delta y$$

de donde se puede identificar de inmediato las siguientes derivadas

$$\left.\begin{aligned} \frac{\partial\dot{x}}{\partial x} &= F_{11} + F - \frac{1}{6}\,G_{11} - \frac{1}{6}\,GA - [6(C_{11} + C) - A_{11}] \\ \frac{\partial\dot{x}}{\partial y} &= F_{12} - \frac{1}{6}\,G_{12} - \frac{1}{6}\,GA^{-1}\,(6C'_{12} - A'_{12}) \end{aligned}\right\} \qquad (5.53)$$

$$\left.\begin{aligned} \frac{\partial\dot{y}}{\partial x} &= F_{21} - \frac{1}{6}\,G_{21} - \frac{1}{6}\,GA^{-1}(6C'_{21} - A'_{21}) \\ \frac{\partial\dot{y}}{\partial y} &= F_{22} + F - \frac{1}{6}\,G_{22} - \frac{1}{6}\,GA^{-1}\,[6(C_{22}+C) - A'_{22}] \end{aligned}\right\} \qquad (5.54)$$

Sustituyendo (5.40) y (5.41) en (5.53) y (5.54) respectivamente se tiene

$$\left.\begin{aligned}\frac{\partial \dot{x}}{\partial x} &= F_{11} + F - \frac{1}{6} G_{11} - \frac{1}{6} G \frac{\partial \ddot{x}}{\partial x} \\ \frac{\partial \dot{x}}{\partial y} &= F_{12} - \frac{1}{6} G_{12} - \frac{1}{6} G \frac{\partial \ddot{x}}{\partial y}\end{aligned}\right\} \qquad (5.55)$$

$$\left.\begin{aligned}\frac{\partial \dot{x}}{\partial x} &= F_{21} - \frac{1}{6} G_{21} - \frac{1}{6} G \frac{\partial \ddot{y}}{\partial x} \\ \frac{\partial \dot{y}}{\partial y} &= F_{22} + F - \frac{1}{6} G_{22} - \frac{1}{6} G \frac{\partial \ddot{y}}{\partial y}\end{aligned}\right\} \qquad (5.56)$$

Ya que la matriz *Jacobiana* del sistema (5.10) requiere del cálculo de las derivadas

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial x} + \frac{\partial f}{\partial \dot{y}} \frac{\partial \dot{y}}{\partial x} + \frac{\partial f}{\partial \ddot{x}} \frac{\partial \ddot{x}}{\partial x} + \frac{\partial f}{\partial \ddot{y}} \frac{\partial \ddot{y}}{\partial x} \qquad (5.57)$$

$$\frac{df}{dy} = \frac{\partial f}{\partial y} + \frac{\partial f}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial y} + \frac{\partial f}{\partial \dot{y}} \frac{\partial \dot{y}}{\partial y} + \frac{\partial f}{\partial \ddot{x}} \frac{\partial \ddot{x}}{\partial y} + \frac{\partial f}{\partial \ddot{y}} \frac{\partial \ddot{y}}{\partial y} \qquad (5.58)$$

sólo faltaría calcular las derivadas de f que se obtienen directamente de (5.10), mientras que las restantes están dadas en (5.40), (5.41), (5.55) y (5.56)

En el cálculo de todas estas matrices se ve que son singulares, siendo su rango de $n-3$. La interpretación geométrica de este hecho es que en un movimiento diferencial de cuerpo rígido aplicado a la curva no produce ningun cambio en $\dot{x}$, $\dot{y}$, $\ddot{x}$ y $\ddot{y}$. Así el espacio nulo de estas matrices de $(n-2)\ xn$ es de dimensión 1.

Este sistema puede convertirse a uno no singular si los vectores x y y se hacen dependientes del vector z de dimensión $m<n$, esto es, si se prescriben simetrías a la curva en estudio. Por ejemplo, la Fig 5.1 tiene una curva con eje de simetría AA' , $m = \frac{n+1}{2}$ con n non.

La dependencia entre x, y y z puede establecerse como

$$x = Vz \ , \ y = Wz \qquad (5.59)$$

donde V y W son matrices constantes dadas de dimensión $n \times m$, que describen la simetría en cuestión. Entonces se tiene

$$\frac{df}{dz} = \frac{df}{dx} V + \frac{df}{dy} W \qquad (5.60)$$

la cual es, por lo general una matriz de rango m, si el parametro z se escoge de manera adecuada. Además, tomando en cuenta la simetría introducida, f resulta dependiente de un vector g de dimensión m como sigue

$$f = Tg \tag{5.61}$$

donde T es una matriz constante dada de dimensión $(n-2) \times m$.

Así

$$\frac{df}{dz} = T \frac{dg}{dz}, \quad \frac{df}{dx} = T \frac{dg}{dx}, \quad \frac{df}{dy} = T \frac{dg}{dy} \tag{5.62}$$

y la ecuación (5.60) toma la forma

$$T \frac{dg}{dz} = T \left(\frac{dg}{dx} V + \frac{dg}{dy} W\right) \tag{5.63}$$

De aquí que, si se supone que T sea de rango m, la ecuación (5.63) conduce a

$$\frac{dg}{dz} = \frac{dg}{dx} V + \frac{dg}{dy} W \tag{5.64}$$

que es una matriz de $m \times m$ no singular, por lo que permite el calculo de z para valores dados de g, usando el método de *Newton - Raphson*, que, como ya se explicó es un metodo muy eficiente para resolver sistemas de ecuaciones no lineales.

Pasando al caso de las *splines* periódicas paramétricas, el problema consiste ahora en determinar una curva dada en forma paramétrica por (5.1) donde x (t) y y (t) son funciones periódicas de t.

Este problema se resuelve aproximando la curva con funciones *spline* periódicas x(t), y(t) con puntos de apoyo de coordenadas x_i, y_i, $i = 1, 2, \ldots n$, almacenados en vectores x y y, respectivamente, de dimensión $n-1$.

Las relaciones entre x y $\ddot{x}$ y entre y y $\ddot{y}$ están dadas por la ecuación (5.3), sólo que las matrices están ahora definidas como

$$A = \begin{bmatrix} 2(\Delta t_{n'} + \Delta t_1) & \Delta t_1 & & & \Delta t_{n'} \\ \Delta t_1 & 2(\Delta t_1 + \Delta t_2) & \Delta t_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \Delta t_{n''} \\ \Delta t_{n'} & & & \Delta t_{n''} & 2(\Delta t_{n''} + \Delta t_{n'}) \end{bmatrix} \qquad (5.65)$$

$$C = \begin{bmatrix} -\left(\frac{1}{\Delta t_{n'}} + \frac{1}{\Delta t_1}\right) & \frac{1}{\Delta t_1} & & & \frac{1}{\Delta t_{n'}} \\ \frac{1}{\Delta t_1} & -\left(\frac{1}{\Delta t_1} + \frac{1}{\Delta t_2}\right) & \frac{1}{\Delta t_2} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{1}{\Delta t_{n''}} \\ \frac{1}{\Delta t_{n'}} & & & \frac{1}{\Delta t_{n''}} & -\left(\frac{1}{\Delta t_{n''}} + \frac{1}{\Delta t_{n'}}\right) \end{bmatrix} \qquad (5.66)$$

con Δt_i definido en (3.26)

Análogamente, la relación entre x y $\dot{x}$, así como aquélla entre y y $\dot{y}$ se expresan como (5.6).

Sin embargo, en el caso de *splines* periódicas, las matrices F y G son de dimensión $(n-1) \times (n-1)$ y de la forma

$$F = \begin{bmatrix} -\frac{1}{\Delta t_1} & \frac{1}{\Delta t_1} & & & & \\ & -\frac{1}{\Delta t_2} & \frac{1}{\Delta t_2} & & & \\ & & \ddots & \ddots & & \\ & & & & \frac{1}{\Delta t_{n''}} & \\ \frac{1}{\Delta t_{n'}} & & & & -\frac{1}{\Delta t_{n'}} & \end{bmatrix} \qquad (5.67)$$

y

$$G = \begin{bmatrix} 2\Delta t_1 & \Delta t_1 & & & & \\ & 2\Delta t_2 & \Delta t_2 & & & \\ & & \ddots & \ddots & & \\ & & & & \Delta t_{n''} & \\ \Delta t_{n'} & & & & 2\Delta t_{n'} & \end{bmatrix} \qquad (5.68)$$

Para este caso, si se prescribe la curvatura, el método de solución es exactamente el mismo que se vio para *splines* naturales paramétricas

Los vectores ϕ y f tienen ahora dimensión $(n-1)$ para valores prescritos ϕ_i, f_i, $i = 1, 2, \ldots, n'$.

Las derivadas parciales de las primeras y segundas derivadas con respecto a los puntos de apoyo ahora toman la forma

$$\frac{\partial \ddot{x}}{\partial x} = 6\left[\frac{\partial A^{-1}}{\partial x} Cx + A^{-1}\frac{\partial C}{\partial x} + A^{-1} C\right] \qquad (5.69)$$

$$\frac{\partial \ddot{x}}{\partial y} = 6\left[\frac{\partial A^{-1}}{\partial y} Cx + A^{-1} \frac{\partial C}{\partial y} x\right] \tag{5.70}$$

$$\frac{\partial \ddot{y}}{\partial x} = 6\left[\frac{\partial A^{-1}}{\partial x} Cy + A^{-1} \frac{\partial C}{\partial x} y\right] \tag{5.71}$$

$$\frac{\partial \ddot{y}}{\partial y} = 6\left[\frac{\partial A^{-1}}{\partial y} Cy + A^{-1} \frac{\partial C}{\partial y} + A^{-1} C\right] \tag{5.72}$$

y

$$\frac{\partial \dot{x}}{\partial x} = \frac{\partial F}{\partial x} x + F - \frac{1}{6}\left[\frac{\partial G}{\partial x} \ddot{x} + G \frac{\partial \ddot{x}}{\partial x}\right] \tag{5.73}$$

$$\frac{\partial \dot{x}}{\partial y} = \frac{\partial F}{\partial y} x - \frac{1}{6}\left[\frac{\partial G}{\partial y} \ddot{x} + G \frac{\partial \ddot{x}}{\partial y}\right] \tag{5.74}$$

$$\frac{\partial \dot{y}}{\partial x} = \frac{\partial F}{\partial x} y - \frac{1}{6}\left[\frac{\partial G}{\partial x} \ddot{y} + G \frac{\partial \ddot{y}}{\partial x}\right] \tag{5.75}$$

$$\frac{\partial \dot{y}}{\partial y} = \frac{\partial F}{\partial y} y + F - \frac{1}{6}\left[\frac{\partial G}{\partial y} \ddot{y} + G \frac{\partial \ddot{y}}{\partial y}\right] \tag{5.76}$$

El calculo de las derivadas que aparecen en las ecs (5.69) - (5.76) se describe con detalle en [1,24]

La única diferencia que es importante señalar es que aquí los sistemas (5.65) y (5.66) son de $(n-1)$ ecuaciones, singulares de rango $(n-2)$. La simetría impuesta para hacer que los valores x y y sean dependientes del vector z de dimensión $m<n$, es tal que conduce a un sistema de m ecuaciones con m incógnitas no singular. Por ejemplo, para una curva cerrada con un eje de simetría, $m = \frac{n+1}{2}$, en tanto que si la curva tiene dos ejes de simetría $m = \frac{n+3}{4}$.

Como ejemplo para ilustrar la aplicación de funciones *spline* naturales paramétricas se propone diseñar los filetes para las 4 esquinas de una abertura en forma de trapecio, que se va a maquinar en una placa sometida a esfuerzos. La abertura se muestra en la Fig 5.2.

Se sabe [23] que si se dejan las esquinas terminadas en el cruce de 2 líneas rectas se provoca una concentración de esfuerzos en las esquinas.

Para evitar la concentración de esfuerzos se recurre invariablemente a diseñar los filetes en cuestión en forma de arcos de círculo, lo que produce una discontinuidad en el radio de curvatura en el punto de tangencia del arco con los tramos rectos. Esto produce igualmente concentración de esfuerzo.

Aquí se propone esos filetes como curvas definidas por *splines* naturales paramétricas.

Ahora el problema para la esquina izquierda inferior se reduce a unir las líneas rectas a y b que están separadas un angulo α. Por lo tanto, la función *spline* que se va a introducir entre los puntos a' y b' tiene que terminar en líneas rectas con pendientes de las líneas a y b respectivamente, lo cual conduce a imponer 2 condiciones, a saber, una de curvatura y una de pendiente.

Para la formulación del problema se introduce un sistema cartesiano x-y, con el eje x coincidente con la recta a. El eje y se ubica arbitrariamente, de manera que sea perpendicular a x, a una distancia adecuada del punto de intersección de las rectas con el objeto de no introducir un mal condicionamiento del sistema de ecuaciones resultante.

Las condiciones de la *spline* en los puntos extremos se expresan ahora como

$$\dot{x}_{a'} = 0 \tag{5.77}$$

y

$$\dot{x}_{b'} + \dot{y}_{b'} \tan \alpha = 0 \tag{5.78}$$

Ya que se obtiene un sistema de ecuaciones no lineales, se escoge, para su solución, el método *Newton- Raphson*.

Es necesario entonces calcular, la matriz *jacobiana* del sistema. Para lograr esto, sólo hay que definir $P_1 = P_a$, y $P_n = P_b$, y obtener, de (5.77) y (5.78)

$$\frac{\partial \dot{x}_1}{\partial z_i} \quad \text{para} \quad i = 1, 2, \ldots, n \tag{5.79}$$

$$\frac{\partial \dot{x}_n}{\partial z_i} + \frac{\partial \dot{y}_n}{\partial z_i} \tan \alpha \quad \text{para} \quad i = 1, 2, \ldots, n \tag{5.80}$$

Entonces hay que introducir $\frac{\partial \dot{x}}{\partial z}$ y $\frac{\partial \dot{y}}{\partial z}$ de (5.53) y (5.54), junto con la dependencia de z ontenida de (5.5a), como

$$\frac{\partial \dot{x}}{\partial z} = \frac{\partial \dot{x}}{\partial x} V + \frac{\partial \dot{x}}{\partial y} W \tag{5.81}$$

$$\frac{\partial \dot{y}}{\partial z} = \frac{\partial \dot{y}}{\partial x} V + \frac{\partial \dot{y}}{\partial x} W \tag{5.82}$$

y sólo utilizar el primer renglón de (5.81) para (5.79), y el n-esimo renglón de (5.81) y (5.82) para (5.80).

Ahora bien, la curvatura en P , está dada por

$$\kappa_i = \frac{\ddot{x}_i \dot{y}_i - \dot{x}_i \ddot{y}_i}{(\dot{x}_i^2 + \dot{y}_i^2)^{3/2}} \qquad i = 1, 2, \ldots, n \tag{5.83}$$

y se prescribe una distribución continua de curvatura como sigue

$$\phi_i = \frac{k \operatorname{sen}^2 (\pi \theta_i)}{0_n} \qquad i = 1, 2, \ldots, n \tag{5.84}$$

Entonces las derivadas faltantes en (5.57) y (5.58) están dadas por las matrices diagonales definidas como

$$\frac{\partial \kappa}{\partial \dot{x}} = \text{diag} \left(D_1 , D_2 , \ldots , D_n \right) \tag{5.85a}$$

$$\frac{\partial \kappa}{\partial \dot{y}} = \text{diag} \left(E_1 , E_2 , \ldots , E_n \right) \tag{5.85b}$$

$$\frac{\partial \kappa}{\partial \ddot{x}} = \text{diag} \left(F_1 , F_2 , \ldots , F_n \right) \tag{5.85c}$$

$$\frac{\partial \kappa}{\partial \ddot{y}} = \text{diag} \left(G_1 , G_2 , \ldots , G_n \right) \tag{5.85d}$$

con

$$D_i = \left[(\dot{y}_i^2 - 2\dot{x}_i^2) \ddot{y}_i + 3 \dot{x}_i \dot{y}_i \ddot{x}_i \right] / \left(\dot{x}_i^2 + \dot{y}_i^2 \right)^{5/2} \tag{5.86a}$$

$$E_i = -\left[(\dot{x}_i^2 - 2\ddot{y}_i^2) \ddot{x}_i + 3 \dot{x}_i \dot{y}_i \ddot{y}_i \right] / \left(\dot{x}_i^2 + \dot{y}_i^2 \right)^{5/2} \tag{5.86b}$$

$$F_i = - y_i / \left(x_i^2 + y_i^2 \right)^{3/2} \tag{5.86c}$$

$$G_i = x_i / \left(x_i^2 + y_i^2 \right)^{3/2} \tag{5.86d}$$

válidas para $i = 2, 3, \ldots , n'$

y

$$D_1 = D_n = E_1 = E_n = F_1 = F_n = G_1 = G_n = 0$$

debido a las condiciones de una función *spline* natural.

Entonces el sistema a resolver es

$$f(z) = 0 \tag{5.87a}$$

$$\begin{aligned} f_i &= \kappa_i (\dot{x}_i, \dot{y}_i, \ddot{x}_i \ddot{y}_i) - \phi_i = 0 \qquad i = 2, 3, \ldots , n' \\ f_1 &= \dot{x}_1 = 0 \\ f_n &= \dot{x}_n + \dot{y}_n \tan \alpha = 0 \end{aligned} \tag{5.87b}$$

Las coordenadas cartesianas de los puntos de apoyo se almacenan en los vectores x y y de dimensión n y se introducen coordenadas polares z_i, θ_i,

$i = 1, 2, \ldots, n$, fijando θ_i.

Así x y y dependen de las variables z_i, $i = 1, 2, \ldots, n$. Siguiendo las relaciones (5,59) las matrices V y W quedan como :

$$V = \begin{bmatrix} c_1 & & & \\ & c_2 & & \\ & & \ddots & \\ & & & c_n \end{bmatrix} \qquad W = \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_n \end{bmatrix}$$

donde

$$c_i = \cos \theta_i, \qquad s_i = \text{sen } \theta_i \qquad i = 1, 2, \ldots, n.$$

Entonces el problema a resolver es un istema no lineal de n ecuaciones con n incógnitas. Este sistema se resolvió con el método de *Newton-Raphson* amortiguado, realizado en la subrutina NRDAMP [20]

El resultado grático para k = 1, n = 7, α = 60° y β = 120°se muestran en la Fig 5.3, con resultados numéricos dados en la tabla 5.1.

Tabla 5.1

Resultados numericos de la síntesis de filetes

para α = 60°

```
ITERACION NO.    4
LA NORMA DE LA FUNCION ES          0.235438E-04
LA FUNCION FUE EVALUADA    6  VECES
EL PROCEDIMIENTO CONVERGIO Y LA SOLUCION FUE :

X(  1)=          0.172353E+01
X(  2)=          0.181674E+01
X(  3)=          0.202555E+01
X(  4)=          0.213647E+01
X(  5)=          0.202555E+01
X(  6)=          0.181674E+01
X(  7)=          0.172353E+01
```

para $\alpha = 120°$

```
ITERACION NO.    2
LA NORMA DE LA FUNCION ES          0.381470E-05
LA FUNCION FUE EVALUADA   4  VECES
EL PROCEDIMIENTO CONVERGIO Y LA SOLUCION FUE :

X(  1)=          0.193016E+01
X(  2)=          0.195504E+01
X(  3)=          0.200616E+01
X(  4)=          0.203210E+01
X(  5)=          0.200616E+01
X(  6)=          0.195504E+01
X(  7)=          0.193016E+01
```

Si la curva deseada debe pasar por puntos dados de las rectas, el número de incógnitas se reduce a $n-2$; pero el de ecuaciones sigue siendo n, por lo que se tiene ahora un sistema no lineal sobredeterminado. En estas condiciones se busca un vector z, z_0, que aporoxime el sistema de ecuaciones (5.87a) con el mínimo error posible. Una forma adecuada de definir este error es mediante la norma *Euclidiana* lo que da lugar a un problema de mínimos cuadrados.

Debe observarse que no todas las ecuaciones (5.87b) tienen la misma importancia en la síntesis de la curva deseada. En efecto, las condiciones de curvatura pueden violarse sin mayores consecuencias, pues la distribución que se especifique será, en cierta forma, arbitraria. Sin embargo las condiciones de pendiente en los puntos de unión con las rectas deben satisfacerse con la mayor precisión posible. En estas condiciones, es necesario introducir factores de ponderación del error, ω_i , que den más importancia a las ecuaciones primera y n-ésima. Lo más cómodo es especificar $\omega_1 = \omega_n = \omega$ y $\omega_i = \omega'$, $i = 2, 3, \ldots, n$, con $\omega > \omega' > 0$. Ahora bien , con el objeto de evitar la aparición de errores de redondeo significativos, se impone la condicion $\omega \leq 1$, por lo que, finalmente, se selecciona

$$\omega = 1, \qquad 0 < \omega' < 1$$

El sistema de ecuaciones (5.7a) adopta ahora la forma :

$$\Omega f(z) = 0 \tag{5.88}$$

con

$$\Omega = \text{diag}\,(\omega_1, \omega_2, \ldots, \omega_n) \qquad 0 \leq \omega_i \leq 1$$

El sistema (5.88) se resolvio mediante el método de *Newton-Raphson*, aplicado a sistemas sobredeterminados. Para esto, se recurrio a la subrutina NLLS [20].

Se obtuvieron resultados para k = 1, n = 7, α = 60° y para valores de ω_i = 1., 0.1, 0.01. Estos resultados se muestran en forma gráfica en la Fig 5.4 y, en forma numérica en la tabla 5.2.

En esta gráfica se nota poca diferencia entre ω_i = 0.1 y ω_i = 0.01.

Tabla 5.2

Resultados numéricos para síntesis de filetes con los puntos *a'* y *b'* fijos.

ω_i = 1.0

```
EN LA ITERACION NO.    6
LA NORMA DE LA  FUNCIONES          0.1303785E+01
EL PROCEDIMIENTO CONVERGIO, LA SOLUCION ES :

X( 1)=          0.57735
X( 2)=          0.45922
X( 3)=          0.41147
X( 4)=          0.39924
X( 5)=          0.41418
X( 6)=          0.46330
X( 7)=          0.57735
```

error en la pendiente en el punto 1 : -0.6694

error en la pendiente en el punto n : -1.3038

$\omega_i = 0.1$

```
EN LA ITERACION NO.    7
LA NORMA DE LA  FUNCIONES          0.1585231E+00
EL PROCEDIMIENTO CONVERGIO, LA SOLUCION ES :

X( 1)=          0.57735
X( 2)=          0.60122
X( 3)=          0.65025
X( 4)=          0.67557
X( 5)=          0.65268
X( 6)=          0.60367
X( 7)=          0.57735
```

error en la pendiente en el punto 1 : -0.0199

error en la pendiente en el punto n : -0.0120

$\omega_i = 0.01$

```
EN LA ITERACION NO.    6
LA NORMA DE LA  FUNCIONES          0.1603620E-01
EL PROCEDIMIENTO CONVERGIO, LA SOLUCION ES :

X( 1)=          0.57735
X( 2)=          0.60519
X( 3)=          0.65712
X( 4)=          0.68252
X( 5)=          0.65715
X( 6)=          0.60522
X( 7)=          0.57735
```

error en la pendiente en el punto 1 : -0.0002

error en la pendiente en el punto n : -0.0001

En la tabla 5.2 se observa el abatimiento del error en la pendiente en los puntos extremos al decrecer ω_i.

Como ejemplo de aplicación de las curvas *spline* periódicas paramétricas, se propone ahora la síntesis de un cascarón para un recipiente a presión de simetría axial.

Como se muestra en la Fig 5,5 el tramo curvo ABC es tangente a las líneas D y E en los puntos A y C. La transición del tramo recto al curvo debe realizarse con cambios continuos de curvatura para evitar cambios de signo en el esfuerzo meridional

El problema del diseño del cascarón se reduce ahora al de la síntesis de la curva generatriz ABC del tramo curvo. Esto se puede obtener como una de las mitades de una curva doblemente simétrica, con ejes de simetría perpendiculares entre sí, como se muestra en la Fig 5.6. Esta curva debe de tener una distribución continua de curvatura de manera tal que alcance un valor nulo en el punto de transición con el tramo recto, punto a, y llegue al punto b con un valor finito prescrito, alcanzando un máximo en un punto correspondiente a θ_0 del ángulo θ de la Fig 5.6.

Las coordenadas cartesianas de los puntos de apoyo de la *spline* que sintetiza la curva deseada se almacenan en los vectores x y y de dimensión $(n-1)$ Se introducen coordenadas polares z_i , θ_i , $i = 1,2, \ldots , m$, fijando θ_i.

Así,x y y dependen del vector

$$z = [z_1 , z_2 , \ldots , z_m]^T$$

y siguiendo las relaciones (5,59) las matrices V y W son de la forma

$$V = \begin{bmatrix} c_1 \\ c_2 \\ \\ c_m \\ c_{m+1} \\ \\ c_{2m-1} \\ c_{2m} \\ \\ \\ c_{3m-2} \\ c_{3m-1} \\ 0 \quad c_{4m-4} \end{bmatrix} \qquad W = \begin{bmatrix} s_1 \\ s_2 \\ \\ s_m \\ s_{m+1} \\ \\ s_{2m-1} \\ s_{2m} \\ \\ \\ s_{3m-2} \\ s_{3m-1} \\ 0 \quad s_{4m-4} \end{bmatrix}$$

Siendo de nuevo f_i , $i = 1, 2, \ldots, n$ la ecuación (5.83), y ϕ_i especificado tal que la curvatura mínima sea $k_{mín}$, en el punto b y la curvatura máxima $k_{máx}$, en el punto $j = \frac{m+1}{2}$ para m non, con la siguiente distribución

$$\phi_i = \frac{k_{máx}}{2} \left(1 - \cos\left[\frac{\pi\, \theta_i}{\theta_j}\right]\right) + k_{mín} \qquad i = 1, 2, \ldots, \frac{m+1}{2}$$

$$\phi_i = \frac{k_{máx}}{2} \left(1 + \cos\left[\frac{\pi(\theta_i - \theta_j)}{\pi - \theta_j}\right]\right) \qquad i = \frac{m+1}{2} + 1, \ldots, m$$

Aquí el problema se reduce a resolver un sistema algebraico no lineal de la forma (5.87a) , siendo z el vector de variables independientes.

Este se resolvio por el método de *Newton-Raphson* para

$m = 5$

$k_{max} = 1.$

$k_{min} = 0,2$

$\theta_j = 15°, 30°, 45°, 75°$

Los resultados numéricos se encuentran en la tabla 5.3 y los gráficos en la Fig 5.7

Tabla 5.3

Resultados numericos en la síntesis de cascarones.

$\theta_j = 15°$

```
ITERACION NO.    3
LA NORMA DE LA FUNCION ES          0.584126E-04
LA FUNCION FUE EVALUADA    5   VECES
EL PROCEDIMIENTO CONVERGIO Y LA SOLUCION FUE :

X(   1)=          0.223493E+01
X(   2)=          0.223975E+01
X(   3)=          0.223177E+01
X(   4)=          0.183133E+01
X(   5)=          0.156838E+01
```

$\theta_j = 30°$

```
ITERACION NO.    3
LA NORMA DE LA FUNCION ES          0.413060E-04
LA FUNCION FUE EVALUADA    5   VECES
EL PROCEDIMIENTO CONVERGIO Y LA SOLUCION FUE :

X(   1)=          0.200365E+01
X(   2)=          0.202601E+01
X(   3)=          0.202140E+01
X(   4)=          0.177155E+01
X(   5)=          0.160444E+01
```

$\theta_j = 45°$

```
ITERACION NO.    3
LA NORMA DE LA FUNCION ES          0.343323E-04
LA FUNCION FUE EVALUADA    5  VECES
EL PROCEDIMIENTO CONVERGIO Y LA SOLUCION FUE :

X(  1)=          0.183189E+01
X(  2)=          0.188575E+01
X(  3)=          0.191116E+01
X(  4)=          0.177734E+01
X(  5)=          0.168226E+01
```

$\theta_j = 60°$

```
ITERACION NO.    3
LA NORMA DE LA FUNCION ES          0.301600E-04
LA FUNCION FUE EVALUADA    5  VECES
EL PROCEDIMIENTO CONVERGIO Y LA SOLUCION FUE :

X(  1)=          0.169695E+01
X(  2)=          0.179636E+01
X(  3)=          0.188164E+01
X(  4)=          0.182591E+01
X(  5)=          0.178268E+01
```

$\theta_j = 75°$

```
ITERACION NO.    3
LA NORMA DE LA FUNCION ES          0.423118E-04
LA FUNCION FUE EVALUADA    5  VECES
EL PROCEDIMIENTO CONVERGIO Y LA SOLUCION FUE :

X(  1)=          0.159144E+01
X(  2)=          0.175053E+01
X(  3)=          0.192505E+01
X(  4)=          0.191237E+01
X(  5)=          0.190123E+01
```

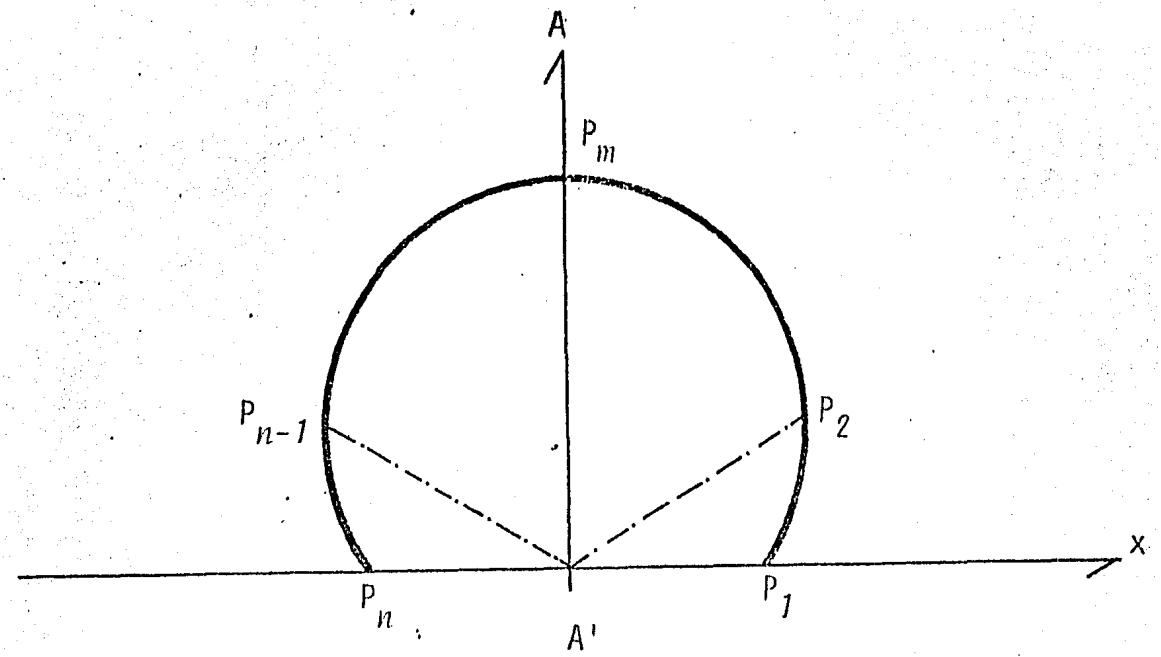


Fig 5.1 *Spline* natural paramétrica con una simetria.

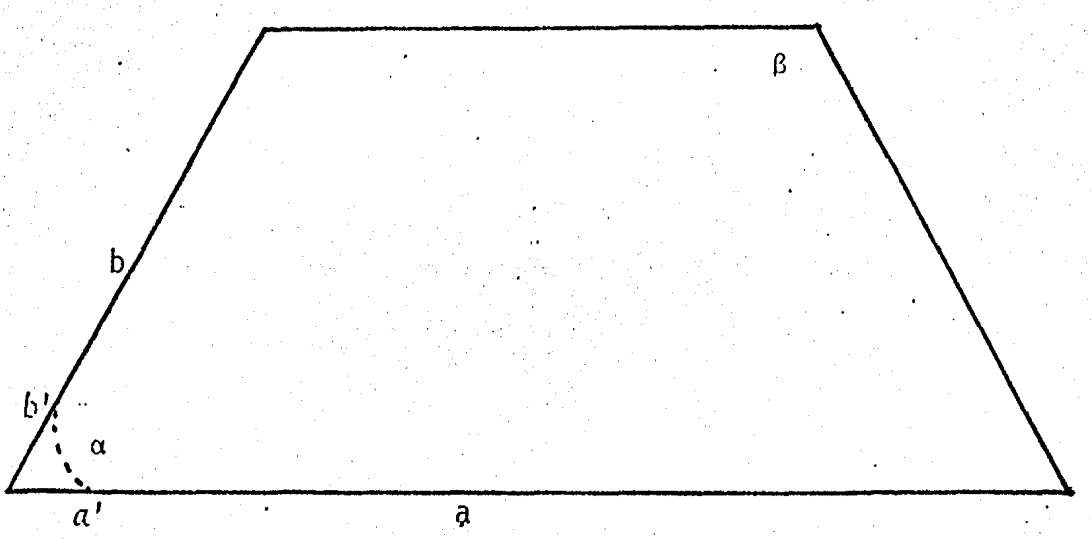


Fig 5.2 Abertura en forma de trapecio.

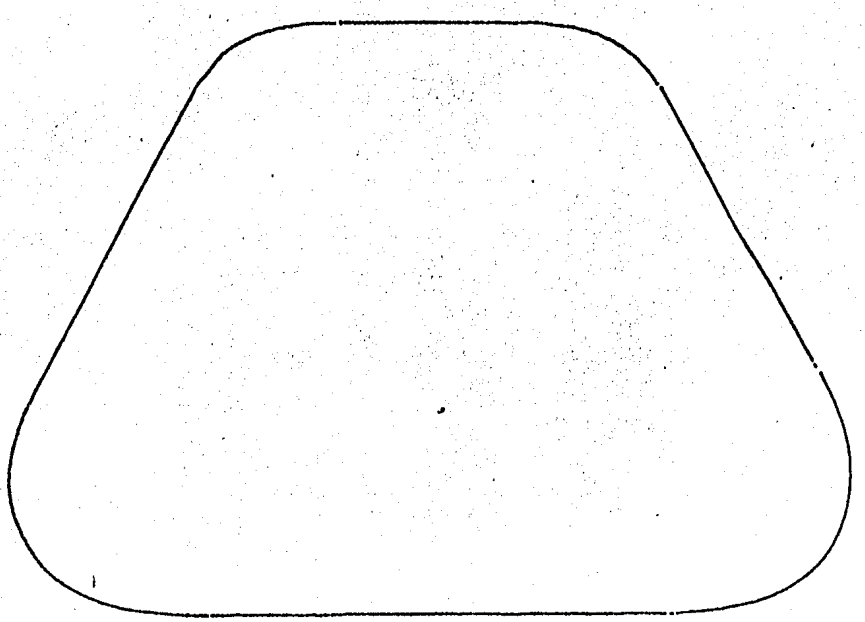


Fig 5.3 Abertura con esquinas terminadas en filetes sintetizados por *splines* naturales paramétricas.

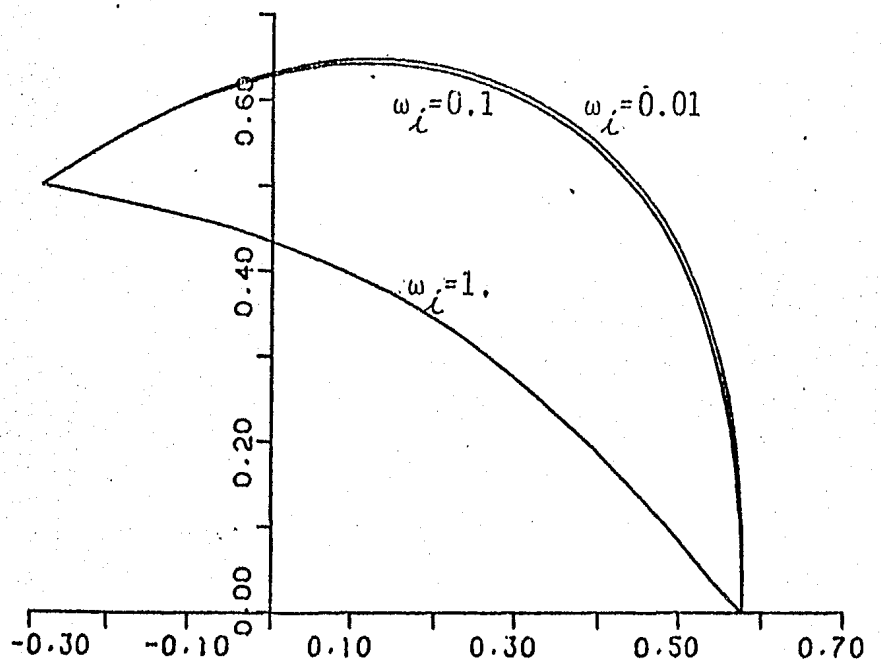


Fig 5.4 *Spline* natural paramétrica que sintetiza a un filete con abertura de 60°. ω_i= factor de ponderización.

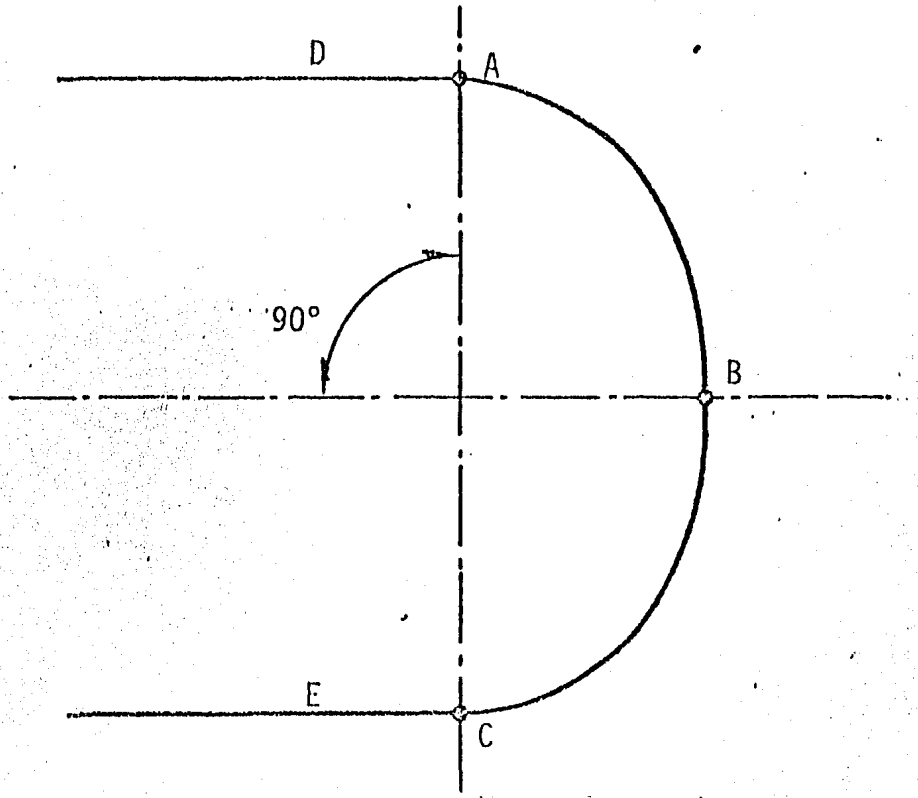


Fig 5.5 Recipiente de presión cilíndrico con simetría axial

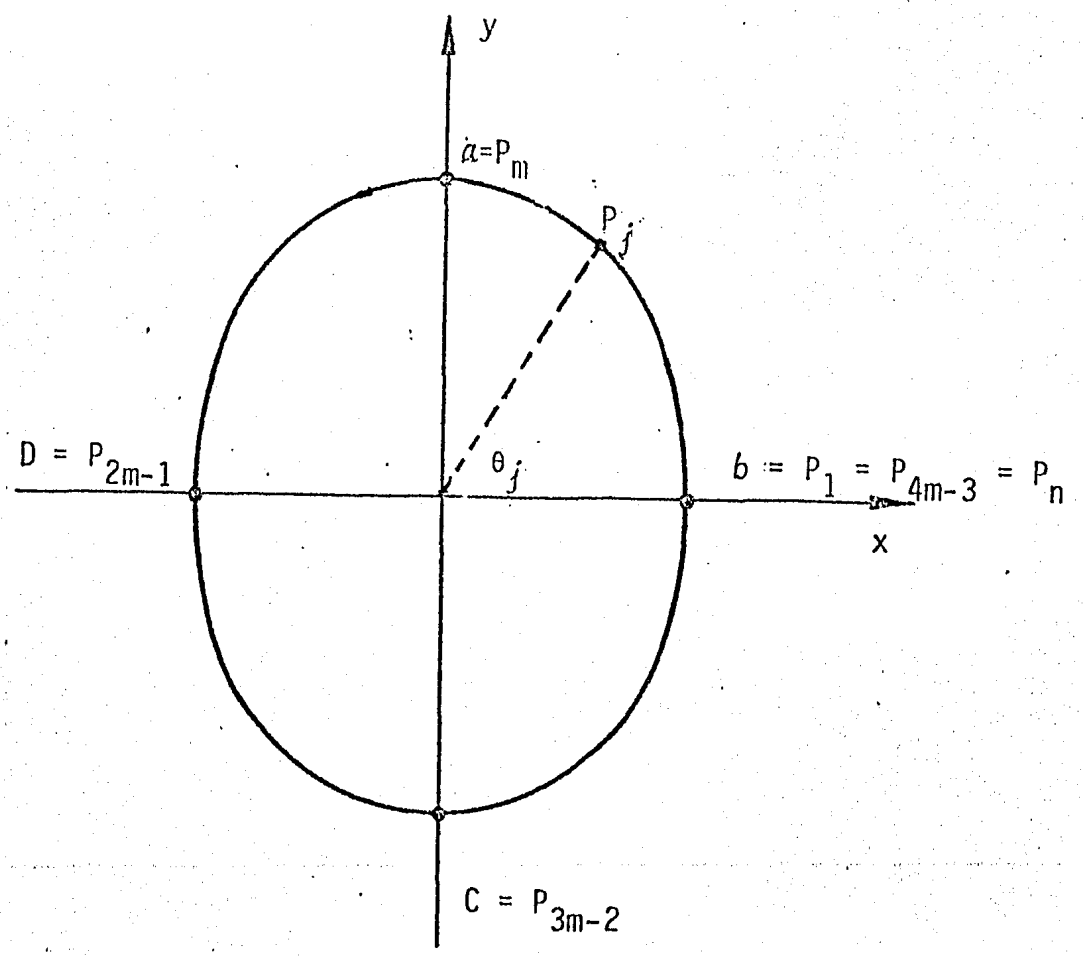


Fig 5.6 Curva con doble simetría aproximada por *splines* periódicas paramétricas

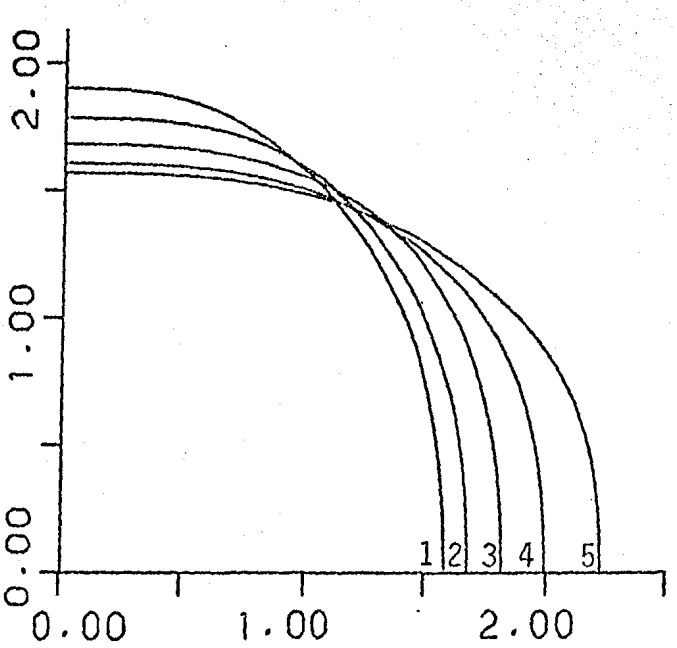


Fig 5.7 Síntesis de cubiertas para recipientes a presión con diferentes valores de " rebaje " $f(f \equiv x_{máx}/y_{máx})$.

Curva	Angulo θ_j
1	70°
2	60°
3	45°
4	30°
5	15°

6. CONCLUSIONES Y RECOMENDACIONES

Dado que las ecuaciones que aparecen en el cálculo de funciones *spline* son bien condicionados, y además, como el cálculo de sus derivadas con respecto a los parámetros independientes se puede realizar con un error de redondeo bajo,el cálculo de las *splines* contiene un error de redondeo igualmente bajo, y se hace en un tiempo reducido, con ayuda de métodos iterativos, lográndose así la solución en un número limitado de iteraciones.

Se mostró que, mediante el uso de funciones *spline*, algunos problemas geométricos pueden transformarse en problemas algebraicos cuya solución se logra mediante métodos ampliamente conocidos. Además, esta solución es muy eficiente y fácil de hallar, dada la cantidad relativamente baja de parámetros independientes que se utiliza.

Es posible que en otra área de interés se pueda requerir de la aplicación de otro tipo de *splines* como la *spline B*. La extensión de este método para dicho tipo de *spline* puede realizarse usando las bases desarrolladas en este trabajo.

Otras áreas en la que este trabajo puede encontrar aplicaciones son la solución de problemas del cálculo variacional y del control óptimo. Efectivamente este tio de problemas se formula en un espacio de *Hilbert*, por lo que su solución debe buscarse en este espacio. Usando funciones *spline* y las técnicas desarrolladas en este trabajo, su solución puede obtenerse mediante técnicas de programación matemática, lo cual se propone aquí como un proyecto de investigación, para desarrollarse a mediano plazo.

Cabe señalar que los métodos presentados aquí, y los programas de computadora que los realizan, ya se han aplicado con éxito en algunos problemas de diseño y análisis de sistemas mecánicos [25,26,27,28,29] .

RECONOCIMIENTOS

Agradezco muy profundamente al Dr. Jorge Angeles A. por su ayuda y apoyo que me brindó durante la elaboracion de este trabajo.

También agradezco a la DEPFI por haberme facilitado las instalaciones del Laboradorio de CAD y, desde luego, a la Facultad de Ingeniería de la UNAM, por la formación que me brindó y que espero desarrollar de la mejor manera posible.

REFERENCIAS

1 Flugge W. , *Statik und Dynamick der Schalen*, Springer-Verlag, Berlín, 1930.

2 Rothbart H. A., *Cams*, John Willey , N. Y. , 1956.

3 Tesar D. y Matthew G. K. , *The Dynamics Synthesis, Analysis, and Design of Modeled Cam Systhems*, Lexington Books, Lexington, Massachusets, 1976.

4 Forsythe G and Moler C B , *Computer Solution of Linear Algegbraic System*, Prentice-Hall, Englewood Cliffs, NJ, 1967 , pp 80-86

5 Neuber H, *Kerbspannungslehre. Grundlage fur genaue Festigkeitsberechnungen*, 2a edición, Springer-Verlag, Berlín, 1958.

6 Kristensen E S y Mandsen N F, " *On the optimun shape of fillets in plates subjected to multiple in-plane loading cases*" , Int J Num Meth Engng, Vol 10, 1976, pp 1007-1019

7 Schnack E, " *Optimierung von Zuglaschen*", Konstruktion, Vol 30, No 7, 1978, pp 27-28

8 Almanza R, Valdéz A, Montes E., *Sistema Generador Solar*, Inf 1er semestre 1980, Instituto de Ingeniería, UNAM, pp 1-7

9 Olvera L, *Análisis, Cálculo y Diseño de las Bovedas de Cáscara*, 2da edición, Editorial Continental, 1969.

10 Prenter P M, *Splines and Variational Methods*, John Wiley and Sons, Inc, NJ, 1975

11 Angeles J, *Síntesis de Curvas Planas Cerradas usando Funciones Spline Parametricas y Periodicas*, Informe Interno DEPFI- UNAM

12 Shultz M H, *Spline Analisis*, Prentice-Hall, Inc. Englewood, Cliffs, N J, 1973

13 Böhmer K, *Spline Analisis*, B G Teubner, Stuttgart, 1974

14 Schechter A , *Synthesis of 2D curves by blending piecewise linear curvature profiles"*, Computer Aided Design, Vol 10, No 1, 1978, pp 8-18

15 Linz P, *Theorical Numerical Analysis* , John Willey and Sons, Inc, N J, 1979, pp 23-63

16 Späth H, *Spline-Algorithmen zur Konstruktion glatter Kurven und Flächen*, 2a edicion, R Olaerburg , Munich, 1978, pp 27-28

17 Halmos P R, *Finite-Dimensional Vector Space*, Springer-Verlay , N J, 1970, pp 88

18 Wilkinson J H and Reinsch C , *Handbook for Automatic Computation*, Vol II , *Linear Algebra*, Springer-Verlag, Berlin, 1982, pp 52-83

19 Björk Å and Dahlquist G, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N J, 1974,pp 248-254

20 Angeles J, *Spatial Kinematic Chains. Analysis, Synthesis, Optimization*, Springer-Verlag, Berlin, 1982,pp 47, 306-307

21 Pal T K ,*"Intrinsic spline curve with local control"*, Computer-Aided Design, Vol 10, No 1, 1978, pp 30-34

22 Pal T K, *"Mean tangent rotational angles and curvature integration"*, Computer-Aided Desing, Vol 10, No 1,1978, pp 30-34

23 Jhnson R C, *Optimun Desing of Mechanical Elementes*, John Wiley and Sons, Inc, N J, 1980,

24 Angeles J.,"*Synthesis of plane curves with prescribed local geometric properties using periodic splines*" Computer-Aided Design, Vol. 15, No. 3, mayo 1983.

25 Angeles J. and Lopez C.C.,"*Optimal synthesis of oscillating roller-follower cam mechanisms with prescribed functional constraints*", aceptado para su presentación en: 1944 ASME International Computers in Engineering Conference and Exhibit a celebrarse del 12 al 16 de agosto de 1984, en Las Vegas, Nevada (EUA).

26 Angeles J. y Lopez C.C., "*Optimal synthesis of traslating-follower cam mechanisms with prescribed functional constraints*", aceptado para su presentación en: International Symposium on Design and Synthesis, a celebrarse el 12 de julio de 1984, en Tokio (Japón).

27 Angeles J.,"*Diseño automatizado de mecanismos de leva de disco con seguidor traslacional de cara plana*", Memorias del IX Congreso de la Academia Nacional de Ingeniería, A.C., septiembre de 1983.

28 Angeles J., "*Analysis of general seven-link revolute-coupled kinematic chains*", Informe Interno DEPFI-UNAM, 1984.

29 Angeles J.,"*Guidance of axially-simetric rigid bodies using fifth-degree-of-freedom-revolute-coupled manipulators*," Informe Interno, DEPFI-UNAM, 1984.

A P E N D I C E

```
C    ************************************************************
C    *                                                          *
C    *    ESTE PROGRAMA SINTETIZA  UNA FUNCION SPLINE NATURAL Y=Y(X)   *
C    *    A PARTIR DE LA PENDIENTE PRESCRITA                    *
C    *                                                          *
C    *    PROGRAMO Y REALIZO : MIGUEL A. ALVARADO RASCON        *
C    *    INSTALACION : PDP 11/40 LABORATORIO CAD-DEPFI-UNAM    *
C    *    PROYECTO : PAQUETE DE SINTESIS DE CURVAS              *
C    *    FECHA DE REVISION: 24/MARZO/1984                      *
C    *                                                          *
C    ************************************************************
C
C
          REAL X(32),Y1D(32),DY2DY(30,32),DY1DY(32,32),Y(32),A(32,32)
          REAL B(32),DELTAX(32)
          INTEGER IP(32)
C
C    LLAMANDO AL ARCHIVO DE LECTURA
C
          TYPE 10
          CALL ASSIGN (1,'RK1:PEND.MAR',-1)
C
C    LECTURA DEL NUMERO DE VARIABLES
C
          READ(1,14)N
C
C    LECTURA DE LAS VARIABLES
C
          READ(1,15) (X(I),Y1D(I),I=1,N)
          TYPE 14,N
          TYPE 15,(X(I),Y1D(I),I=1,N)
          NN=N-1
          N2=N-2
          CALL RELAC(DELTAX,A,B,X,Y1D,DY2DY,DY1DY,Y,
     $    IP,N,NN,N2)
C
C    ZONA DE FORMATOS
C
   10     FORMAT (3X,'ARCHIVO DE LECTURA ?  ',$)
   14     FORMAT (I4)
   15     FORMAT (2F8.4)
          CALL EXIT
          END
C    ************************************************************
          SUBROUTINE RELAC (DELTAX,A,B,X,Y1D,DY2DY,DY1DY,Y,
     $    IP,N,NN,N2)
C
C    TEXTO:
C    DADO EL VECTOR PRESCRITO Y' DE DIMENSION N, CUYOS ELEMENTOS REPRE-
C    SENTAN LA DERIVADA DE Y CON RESPECTO A X , SE OBTIENE Y=Y(X)  QUE
C    ES UNA SPLINE CUBICA NATURAL PARA LOS VALORES PRESCRITOS DE Y',EL
C    VECTOR Y SE ALMACENA EN EL ARREGLO Y DE DIMENSION N
C
C    ENTRADA:
C    N...NUMERO DE PUNTOS DE APOYO
C    X...VECTOR DE DIMENSION N QUE CONTIENE LAS ABSCISAS DE LA SPLINE
```

```
C    Y1D...VECTOR DE DIMENSION N QUE CONTIENE LA PRIMERA DERIVADA DE Y
C    CON RESPECTO A X
C
C    SALIDA:
C    Y...VECTOR DE DIMENSION N QUE CONTIENE LAS ORDENADAS DE LA SPLINE
C    BUSCADA
C
C    DECLARACION DE VARIBLES
C

         REAL DELTAX(NN),A(N2,N2),B(N2),Y(N)
         REAL X(N),Y1D(N),DY2DY(N2,N),DY1DY(N,N)
         INTEGER IP(N)
C
C    CALCULO DE LA DERIVADA DE Y'' CON RESPECTO A Y
C
         CALL SEGNAT (N,X,DY2DY,N2,NN,DELTAX,A,B,IP)
C
C    CALCULO DE LA DERIVADA DE Y' CON RESPECTO A Y
C
         CALL PRINAT (N,X,DY2DY,DY1DY,N2,NN,DELTAX)
C
C    SOLUCION DEL SISTEMA LINEAL AY"=-6CY
C
         CALL DECOMP (N,N,DY1DY,IP)
         CALL SOLVE  (N,N,DY1DY,Y1D,IP)
C
C    LLAMANDO AL ARCHIVO PEND1.MAR PARA GUARDAR RESULTADOS
C
         CALL ASSIGN (2,'RK1:PEND1.MAR')
         WRITE(2,14)N
         WRITE (2,16) (X(I),Y1D(I),I=1,N)
   16    FORMAT (2F8.4)
   14    FORMAT(I4)
         RETURN
         END
C    ***************************************************************
         SUBROUTINE SEGNAT (N,X,DY2DY,N2,NN,DELTAX,A,B,IP)
C
C    TEXTO:
C    DADA UNA SPLINE CUBICA NATURAL Y= Y(X) CON N PUNTOS  DE APOYO,
C    CUYAS ORDENADAS  (EN LOS PUNTOS DE APOYO -PA-) SE ALMACENAN EN
C    EL VECTOR Y DE DIMENSION N, SE DEFINE EL VECTOR Y''  DE DIMEN-
C    SION N-2, QUE CONTIENE COMO  ELEMENTOS LOS VALORES DE LA SEGUN-
C    DA DERIVADA,  DE Y CON RESPECTO A X.  ESTE SUBPROGRAMA CALCULA
C    LA MATRIZ DE LAS DERIVADAS DEL VECTOR Y''  CON RESPECTO A Y DE
C    DIMENSION (N-2)XN
C
C    DATOS:
C    N...NUMERO DE PUNTOS DE APOYO DE LA SPLINE
C    X...VECTOR DE DIMENSION N,CUYOS ELEMENTOS SON LAS ABSCISAS  DE
C    LOS PUNTOS DE APOYO DE LA SPLINE
C
C    SALIDA:
C    DY2DY...MATRIZ DE (N-2)XN, QUE CONTIENE LAS DERIVADAS DE Y'' CON
```

```
C   RESPECTO A Y, MULTIPLICADAS POR EL FACTOR  1./6.
C
C   VARIABLES AUXILIARES:
C   DELTAX...VECTOR DE DIMENSION N-1, CUYOS ELEMENTOS SON LAS DIFE-
C   RENCIAS X(I+1)-X(I)
C   A...MATRIZ DE (N-2)X(N-2) QUE RELACIONA Y'' CON Y.
C   NN...=N-1
C   N2...=N-2
C
C   DECLARACION DE VARIABLES:
C
      INTEGER N,N2,NN,IP(N2)
      REAL X(N),DELTAX(NN),A(N2,N2),DY2DY(N2,N),B(N2)
C
C   CALCULO DE DELTAX
C
      NM1= N-1
      NM2= N-2
      DO 10 I= 1,NM1
10    DELTAX(I)= X(I+1)-X(I)
C
C   CALCULO DE LA MATRIZ A
C
      DO 5 K=1,N2
         DO 5 J=1,N2
5     A(J,K)=0.
      DO 50 I= 1,NM2
         INDEX= (I-1)*(I-NM2)
         IF(INDEX.EQ.0) GOTO 30
         A(I-1,I)= DELTAX(I)
         A(I+1,I)= DELTAX(I+1)
         A(I,I)= A(I-1,I)+A(I+1,I)
         GOTO 50
30       IF (I.EQ.NM2) GOTO 40
         A(I+1,I)= DELTAX(I+1)
         A(I,I)= A(I+1,I)+DELTAX(I)
         GOTO 50
40       A(I-1,I)= DELTAX(I)
         A(I,I)= A(I-1,I)+DELTAX(I+1)
50    A(I,I)= A(I,I)+A(I,I)
C
C   FACTORIZA A EN EL PRODUCTO LU:
C
      CALL DECOMP (N2,N2,A,IP)
C
      DO 100 I= 1,N
         DO 60 J= 1,NM2
            B(J)= 0.
60       IF (J.EQ.(I-1)) B(J)= VAL
C
C   CALCULO DE LAS COLUMNAS DE LA MATRIZ C
C
         IF (I.GE.(N-1)) GOTO 90
         B(I)= 1./DELTAX(I)
         VAL = B(I)
         IF (I.EQ.1) GOTO 70
         IF (I.EQ.2) GOTO 80
```

```
            B(I-2)= B(I-1)
 80         B(I-1)= -B(I-1)-B(I)
            GOTO 70
 90         B1= 1./DELTAX(NM1)
            VAL= B1
            IF (I.EQ.N) GOTO 95
            B(I-2)= B(I-1)
            B(I-1)= -B1-B(I-1)
            GOTO 70
 95         B(I-2)=B1
C
C   SUSTITUCION REGRESIVA
C
 70         CALL SOLVE (N2,N2,A,B,IP)
            DO 85 J= 1,NM2
 85         DY2DY(J,I)= B(J)
 100    CONTINUE
        RETURN
        END
C   ***********************************************************************
        SUBROUTINE PRINAT (N,X,DY2DY,DY1DY,N2,NN,DELTAX)
C
C   TEXTO:
C   DADA  UNA SPLINE CUBICA NATURAL  Y= Y(X) CON N PUNTOS DE APOYO
C   (PA) CUYAS ORDENADAS  SE ALMACENAN EN EL VECTOR Y DE DIMENSION
C   N, SE DEFINE EL VECTOR  Y', DE DIMENSION N,CUYOS ELEMENTOS SON
C   LOS VALORES  DE Y'(X) EN LOS PUNTOS DE APOYO. ESTE SUBPROGRAMA
C   CALCULA LAS  DERIVADAS PARCIALES DEL VECTOR Y' CON RESPECTO AL
C   VECTOR Y. ESTE ARREGLO ES UNA MATRIZ DE NXN.
C
C   DATOS:
C   N...NUMERO DE PUNTOS DE APOYO DE LA SPLINE
C   X...VECTOR DE DIMENSION N, CUYOS ELEMENTOS SON LAS ABSCISAS  DE
C   LOS PUNTOS DE APOYO DE LA SPLINE
C   DY2DY...MATRIZ DE (N-2)XN, CALCULADA EN LA SUBROUTINA SEGNAT
C
C   SALIDA:
C   DY1DY...MATRIZ DE  NXN: DERIVADA DEL VECTOR Y' CON RES-
C   PECTO A Y.
C
C   VARIABLES AUXILIARES:
C   DELTAX...VECTOR DE DIMENSION N-1 CUYOS ELEMENTOS SON LAS DIFE-
C   RENCIAS X(I+1)-X(I).
C   F Y G...MATRICES DE NXN Y DE NX(N-2)
C
C   DECLARACION DE VARIABLES:
C
        INTEGER N,NN,N2
        REAL X(N),DELTAX(NN),DY2DY(N2,N),DY1DY(N,N)
C
C   CALCULO DE DELTAX
C
        NM1= N-1
        NM2= N-2
        DO 10 I= 1,NM1
 10     DELTAX(I)= X(I+1)-X(I)
C
C   CALCULO DE LA MATRIZ -G*DY2DY=-DA**(-1)*C. SE ALMACENA EN DY1DY.
```

```
C
      DO 11 I= 1,N
         DY1DY(1,I)= (DY2DY(1,I)*DELTAX(1))*-1.
         DY1DY(NM1,I)= DY2DY(NM2,I)*DELTAX(NM1)
         DY1DY(NM1,I)=-DY1DY(NM1,I)-DY1DY(NM1,I)
         DY1DY(N,I)= DY2DY(NM2,I)*DELTAX(NM1)
         DO 11 J= 2,NM2
11    DY1DY(J,I)= (-2.*DY2DY(J-1,I)-DY2DY(J,I))*DELTAX(J)
C
C   SUMA DE LA MATRIZ F CON -GA**(-1)*C
C
      DO 20 I= 1,NM1
         DY1DY(I,I)= DY1DY(I,I)-1./DELTAX(I)
20    DY1DY(I,I+1)= DY1DY(I,I+1)+1./DELTAX(I)
C
C   IMPONIENDO LA CONDICION PARA QUITAR LA SINGULARIDAD
C
      DO 30 I= 1,N
30    DY1DY(N,I)= 1.
      RETURN
      END
C   ***********************************************************************
      SUBROUTINE SOLVE(N,NDIM,A,B,IP)
C
C   TEXTO:
C   SOLUCION DEL SISTEMA LINEAL A*X=B...(L)
C
C   ENTRADA:
C   N,NDIM,IP...EXPLICADAS EN DECOMP
C   A...MATRIZ TRIANGULAR OBTENIDA EN DECOMP
C   B...VECTOR DE DIMENSION N, MIEMBRO DERECHO DE LA ECUACION DADA
C   (L)
C
C   SALIDA:
C   B...VECTOR SOLUCION, X
C
C   MOLER C.B. ,"ALGORITHM 423. LINEAR EQUATION  SOLVER I (F 4)"
C   COMMUNICATIONS OF THE ACM, VOL 15 , NO 4, ABRIL 1973, P 274.
C
C  DECLARACION DE VARIABLES
C
      REAL A(NDIM,NDIM)
      REAL B(NDIM),TT
      INTEGER IP(NDIM)
      IF(N.EQ.1) GO TO 90
      NM1=N-1
      DO 70 K=1,NM1
         KP1=K+1
         M=IP(K)
         TT=B(M)
         B(M)=B(K)
         B(K)=TT
            DO 70 I=KP1,N
70  B(I)=B(I)+A(I,K)*TT
```

```
      DO 80 KB=1,NM1
           KM1=N-KB
           K=KM1+1
           IF(A(K,K).EQ.0) STOP2
           B(K)=B(K)/A(K,K)
           TT=-B(K)
                DO 80 I=1,KM1
   80 B(I)=B(I)+A(I,K)*TT
   90 IF(A(1,1).EQ.0) STOP3
      B(1)=B(1)/A(1,1)
      RETURN
      END
C     ***************************************************************
      SUBROUTINE DECOMP(N,NDIM,A,IP)
C
C     TEXTO:
C     SE FACTORIZA LA MATRIZ A POR ELIMINACION GAUSSIANA
C
C     ENTRADA:
C     N...ORDEN DE LA MATRIZ
C     NDIM...DIMENSION DE LA MATRIZ A EN EL PROGRAMA PRINCIPAL
C     A...MATRIZ QUE SE VA A FACTORIZAR
C
C     SALIDA:
C     A(I,J)...I<J=FACTOR TRIANGULAR SUPERIOR DE LA MATRIZ : U
C     A(I,J)...I>J=MULTIPLICADORES= FACTOR TRIANGULAR INFERIOR DE LA
C     MATRIZ A : I-L
C     IP(K)...K<N=INDICE DEL K-ESIMO RENGLON
C     IP(N)...=(-1)**(NUMERO DE INTERCAMBIOS) O 0.
C     DETER(A)...=IP(N)*A(1,1)*...*(A(N,N)
C
C
C     MOLER C.B. ,"ALGORITHM 423, LINEAR EQUATION SOLVER (F 4)"
C     COMMUNICATIONS OF THE ACM, VOL 15, NO 4, ABRIL 1973, P 274
C
C     DECLARACION DE VARIABLES

      REAL A(NDIM,NDIM)
      INTEGER IP(NDIM)
      IP(N)=1
      DO 60 K=1,N
           IF(K.EQ.N) GOTO 50
           KP1=K+1
           M=K
                DO 10 I=KP1,N
                     IF(ABS(A(I,K)).GT.ABS(A(M,K))) M=I
   10           CONTINUE
           IP(K)=M
           IF(M.NE.K) IP(N)=-IP(N)
           TT=A(M,K)
           IF(TT.EQ.0.)STOP1
           A(M,K)=A(K,K)
           A(K,K)=TT
           IF(TT.EQ.0.)GO TO 50
```

```
                DO 20 I=KP1,N
20              A(I,K)=-A(I,K)/TT
                DO 40 J=KP1,N
                  TT=A(M,J)
                  A(M,J)=A(K,J)
                  A(K,J)=TT
                  IF(TT.EQ.0.) GOTO 40
                      DO 30 I=KP1,N
30              A(I,J)=A(I,J)+A(I,K)*TT
40 CONTINUE
50 IF(A(K,K).EQ.0.) IF(N)=0
60 CONTINUE
   RETURN
   END
```

```
C    ***********************************************************************
C    *                                                                     *
C    *    ESTE PROGRAMA SINTETIZA LA FUNCION SPLINE PERIODICA  Y=Y(X)     *
C    *    A PARTIR  DE LA PRESCRIPCION DE LA SEGUNDA DERIVADA  DE  Y      *
C    *    CON RESPECTO A X.                                                *
C    *                                                                     *
C    *    PROGRAMO Y REALIZO : MIGUEL ANGEL ALVARADO RASCON                *
C    *    INSTALACION : PDP 11/40 LAB CAD-DEPFI-UNAM                       *
C    *    PROYECTO : PAQUETE DE SINTESIS DE CURVAS                         *
C    *    FECHA DE REVISION : 1/ABRIL/1984                                 *
C    *                                                                     *
C    ***********************************************************************
C
C    DECLARACION DE VARIABLES
C
      REAL XD(41),ZD(41),DEL(40),CW1(40,11),A1(40,40)
      REAL C1(40,40),X(11),Z(11),Z2D(11)
      REAL AVZ(40),AC(40,11),Y2D(41)
      REAL IP(40)
C
C    PREGUNTA POR EL NUMERO DE VARIABLES M
C
      TYPE 14
      ACCEPT 15,M
C
C    CALCULO DE N
C
      N=M*4-3
      NN=N-1
      NN2=2*NN
C
C
C    LLAMANDO A LA SUBRUTINA PRINCIPAL
C
      CALL NIPAR(XD,ZD,DEL,AC,CW1,A1,C1,X,Z,Z2D,AVZ,Y2D,M,N,NN,IP)
C
C    ZONA DE FORMATOS
C
14    FORMAT(3X,'NUMERO DE PUNTOS M= ',$)
15    FORMAT(I4)
      CALL EXIT
      END
C    ***********************************************************************
      SUBROUTINE NIPAR(XD,ZD,DEL,AV1,CW1,A1,C1,X,Z,Z2D,AVZ,Y2D,M,N,
     $  NN,IP)
C
C    TEXTO:
C    DADO EL VECTOR PRESCRITO Y'' DE DIMENSION (N-1), CUYOS ELEMENTOS
C    REPRESENTAN LA SEGUNDA DERIVADA DE Y CON RESPECTO A X,  EVALUADA
C    EN LOS PUNTOS DE APOYO, SE OBTIENE Y=Y(X),  QUE  ES UNA  SPLINE
C    CUBICA PERIODICA PARA LOS VALORES PRESCRITOS DE Y''. EL VECTOR Y
C    DEPENDE DEL VECTOR Z  QUE SE ALMACENA  EN EL ARREGLO Z DE DIMEN-
C    SION M.LT.(N-1)
C
C    ENTRADA:
C    M...NUMERO DE VARIABLES Z2D
C    Z2D...=VECTOR DE DIMENSION M QUE ALMACENA LA SEGUNDA DERIVADA  DE
C    Z CON RESPECTO A X
C
```

```
C    X...VECTOR DE DIMENSION N=(M*4-3) CUYOS ELEMENTOS SON   LAS ABS-
C    CISAS DE LOS PUNTOS DE APOYO DE LA SPLINE
C
C    SALIDA:
C    Z... VECTOR  DE  DIMENSION  M. CONTIENE LAS M ORDENADAS INDEPEN-
C    DIENTES DE LA SPLINE PERIODICA
C
C
C    DECLARACION DE VARIABLES
C
          REAL XD(N),ZD(N),DEL(NN),AV1(NN,M),CW1(NN,M),A1(NN,NN)
          REAL X(M),Z(M),Z2D(M),AVZ(NN),Y2D(N),IP(NN),C1(NN,NN)
          INTEGER M,N,NM1,NM2
C
C    SE ABRE EL ARCHIVO PARA LA LECTURA
C
          N=M*4-3
          NM1=N-1
C
C    ABSCISA DEL PUNTO M
C
          TYPE 21
          ACCEPT 22, AMAX
          X(M)=AMAX
          X(1)=0.
          Z1=0.
          PI=3.1416
          DIV=FLOAT(M-1)
          DK=AMAX/DIV
C
C    CALCULANDO LAS ABSCISAS X(I) IGUALMENTE ESPACIADAS
C
          DO 12 I=2,NM1
   12     X(I)=X(I-1)+DK
C
C    ESPECIFICANDO LA SEGUNDA DERIVADA EN LOS M PUNTOS.  Y''= Z2D
C
          DO 28 I=1,M
             Z2D(I)=SIN((PI*DK*Z1)/(X(M)))
   28     Z1=Z1+1.
          TYPE 23
          TYPE 24,(I,X(I),Z2D(I),I=1,M)

C
C    LLAMANDO A LA SUBRUTINA PARA EFECTUAR LOS CALCULOS
C
          CALL PERNO (M,N,NM1,Z2D,X,Z,DEL,AV1,CW1,A1,C1,AVZ,IP)
C
C    OBTENIENDO EL VECTOR Y=ZD A PARTIR DE Z
C
          CALL DUPLIF (M,M,N,X,Z,XD,ZD)
C
C    OBTENCION DE Y2D A PARTIR DE Z2D
C
          CALL DUPLIF(M,M,N,X,Z2D,XD,Y2D)
C
C    GUARDANDO REULTADOS EN EL ARCHIVO PUNTOS.MAR
C
          CALL ASSIGN (2,'RK1:PUNTOS.MAR')
```

```
      WRITE(2,88)( XD(I),ZD(I),I=1,N)
   88 FORMAT (2F8.4)
      TYPE 69
      TYPE 70, (I,XD(I),ZD(I),Y2D(I),I=1,N)
C
C   ZONA DE FORMATOS
C
   21 FORMAT(3X,'ABSCMAX EN M =' ,$)
   22 FORMAT(F8.4)
   23 FORMAT(3X,'   I    X(I)    Z2D(I)')
   24 FORMAT(3X,I4,2F8.4)
   69 FORMAT (3X,'   I    XD(I)    ZD(I)    Y2D(I)')
   70 FORMAT(3X,I4,3F8.4)
      CALL EXIT
      END
C  ************************************************************
      SUBROUTINE PERNO (M,N,NM1,Z2D,X,Z,DELTAX,AV,CW,A,C,AVZ2D,IP)
C
C   TEXTO:
C   DADA UNA SPLINE PERIODICA Y=Y(X) CON N PUNTOS DE APOYO CUYAS
C   ORDENADAS INDEPENDIENTES EN LOS PUNTOS DE APOYO SE ALMACENAN
C   EN EL VECTOR Z DE DIMENSION M. SE DEFINE EL ARREGLO Z2D CUYOS
C   ELEMENTOS SON LAS SEGUNDAS DERIVADAS DE Y CON RESPECTO A X E-
C   VALUADA EN LOS PUNTOS DE APOYO. ESTE PROGRAMA CALCULA LA ORDE-
C   NADA A PARTIR DE LA SEGUNDA DERIVADA PRESCRITA.
C
C   ENTRADA :
C   M...NUMERO ENTERO DE VARIABLES INDEPENDIENTES Z2D
C   Z2D...VECTOR DE M VARIABLES QUE ALMACENA LA SEGUNDA DERIVADA
C   X...VECTOR DE DIMENSION N=(MX4)-3 CUYOS ELEMENTOS SON LAS ABS-
C   CISAS DE LOS PUNTOS DE APOYO DE LA SPLINE
C
C   VARIABLES AUXILIARES:
C   N...NUMERO DE PUNTOS DE APOYO DE LA SPLINE N=(MX4)-3
C   DELTAX...VECTOR DE DIMENSION (N-1) CUYOS ELEMENTOS SON LAS DI-
C   FERENCIAS X(I+1)-X(I)
C   A...MATRIZ DE DIMENSION MX(N-1)
C   AV...MATRIZ DE DIMENSION MXM QUE RELACIONA Z2D CON Z
C   CW...MATRIZ DE DIMENSION MXM QUE RELACIONA Z CON Z2D
C   C...MATRIZ DE DIMENSION MXN-1
C   AVZ2D...MULTIPLICACION DE AV X Z2D CUYA DIMENSION ES M
C   Z... VECTOR DE DIMENSION M CUYOS ELEMENTOS SON LAS ORDENADAS
C   INDEPENDIENTES DE LA SPLINE BUSCADA
C   HECOMP Y HOLVE...SUBRUTINAS QUE RESUELVEN UN SISTEMA SOBREDE-
C   TERMINADO AX=B
C
C
C   DECLARACION DE VARIABLES
C
      REAL Z2D(M),X(M),DELTAX(NM1),AV(NM1,M),CW(NM1,M),Z(M)
      REAL A(NM1,NM1),C(NM1,NM1),AVZ2D(NM1),IP(NM1)
      INTEGER M,N,NM1
C
C   CALCULO DE DELTAX
C
      NM2=M-1
```

```
      DO 10 I=1,NM2
           DELTAX(I)=X(I+1)-X(I)
           MO=M*2-1-I
           M1=M*2-2+I
           M2=M*4-3-I
           DELTAX(MO)=DELTAX(I)
           DELTAX(M1)=DELTAX(I)
           DELTAX(M2)=DELTAX(I)
10    CONTINUE
C
C   CALCULO DE LA MATRIZ A DE LA RELACION AY''=6CY
C
      DO 20 J=1,NM1
           DO 25 I=1,NM1
                A(I,J)=0.
                C(I,J)=0.
25         CONTINUE
20    CONTINUE
      DO 30 I=1,NM1
           IM1=I-1
           IP1=I+1
           IF (I.EQ.1) IM1=NM1
           IF (I.EQ.NM1) IP1=1
           A(I,IM1)=DELTAX(IM1)
           A(I,IP1)=DELTAX(I)
           A(I,I)=2*(A(I,IM1)+A(I,IP1))
C
C   CALCULO DE LA MATRIZ C DE LA RELACION AY''=6CY
C
           C(I,IM1)=1/DELTAX(IM1)
           C(I,IP1)=1/DELTAX(I)
           C(I,I)=-(C(I,IM1)+C(I,IP1))
30    CONTINUE
C
C   MULTIPLICACION POR LAS MATRICES DE SIMETRIA
C
      CALL MULVWF(A,C,AV,CW,NM1,M)
C
C   MULTIPLICACION DE AV*Z2D=AVY''
C
      DO 66 J=1,NM1
66    AVZ2D(J)=0.
      DO 70 J=1,NM1
           DO 80 I=1,M
80         AVZ2D(J)=AVZ2D(J)+AV(J,I)*Z2D(I)
70    CONTINUE
C
C  APLICACION DE LA REFLEXION DE HOUSEHOLDER A LA MATRIZ CW
C
      CALL HECOMP (NM1,NM1,M,CW,IP)
C
C   SOLUCION DEL SISTEMA Z=CW**(-1)*AV*Z2D POR SUSTITUCIONES REGRESIVAS
C
      CALL HOLVE(NM1,NM1,M,CW,IP,AVZ2D)
      DO 90 I=1,M
90     Z(I)=AVZ2D(I)/6.
      RETURN
      END
```

```
C    ***********************************************************************
     SUBROUTINE DUPLIF (M,NDIM1,NDIM2,X,Z,X1,Z1)
C
C    ESTA SUBRUTINA  OBTIENE LAS  N VARIABLES X Y Y A PARTIR DE
C    LAS M VARIABLES X Y Y INDEPENDIENTES PARA UNA FUNCION
C    SPLINE PERIODICA
C
C    ENTRADA:
C    X...VECTOR QUE CONTIENE LAS M ABSCISAS DE LOS PUNTOS DE APOYO
C    Z...VECTOR QUE CONTIENE LAS M ORDENADAS DE LOS PUNTOS DE APOYO
C
C    SALIDA:
C    X1...VECTOR QUE CONTIENE LAS M ABSCISAS DE LOS PUNTOS DE APOYO
C    Z1...VECTOR QUE CONTIENE LAS N ORDENADAS DE LOS PUNTOS DE APOYO
C
C    DECLARACION DE VARIABLES
C
          REAL X1(NDIM2),Z1(NDIM2)
          REAL X(NDIM1),Z(NDIM1)
          DO 10 I= 1,M
   10     Z1(I)=Z(I)
          L5=M*4-3
          Z1(L5)=-Z(1)
          L0=M*2-1
          Z1(L0)=Z(1)
          L1=M*3-2
          Z1(L1)=-Z(M)
          N1=M-1
          DO 21 I=2,N1
               L0=M*2-I
               L1=M*2-2+I
               L2=M*4-2-I
               Z1(L0)=Z(I)
               Z1(L1)=-Z(I)
               Z1(L2)=-Z(I)
   21     CONTINUE
          DO 11 I=1,M
   11     X1(I)=X(I)
          XA=X(M)-X(1)
          M1=M*4-3
          M2=M*3-2
          M3=M*2-1
          X1(M1)=4*XA+X(1)
          X1(M2)=3*XA+X(1)
          X1(M3)=2*XA+X(1)
          M2=M-2
          DO 22 I=1,M2
               K=I+1
               XB=X(K)-X(1)
               M4=M3-I
               M5=M3+I
               M6=M1-I
               X1(M4)=X1(M3)-XB
               X1(M5)=X1(M3)+XB
               X1(M6)=X1(M1)-XB
```

```
   22     CONTINUE
          NM2=M-1
          GO TO 24
   24     RETURN
          END
C  ***********************************************************************
          SUBROUTINE HECOMP(MDIM,M,N,A,U)
C
C    DECLARACION DE VARIABLES
C
          INTEGER MDIM,N,M
          REAL A(MDIM,N),U(M)
          REAL ALPHA,BETA,GAMMA,SQRT
C
C    TEXTO:
C     REDUCCION DE HOUSEHOLDER DE LA MATRIZ RECTANGULAR A CUYOS VA-
C     LORES SE ALMACENAN  EN LA  TRIANGULAR  SUPERIOR .  SE UTILIZA
C     HOLVE PARA RESOLVER EL SISTEMA POR MINIMOS  CUADRADOS POR SER
C     UN SISTEMA SOBREDETERMINADO
C
C    MDIM= DIMENSION DECLARADA DE LOS RENGLONES DE A
C    M= NUMERO DE RENGLONES DE A
C    N= NUMERO DE COLUMNAS DE A
C    A= (MXN) MATRIZ CON M>N
C
C    ENTRADA:
C                MATRIZ QUE VA A SER REDUCIDA
C       SALIDA:
C                  MATRIZ REDUCIDA E INFORMACION DE LA REDUCCION
C    U= VECTOR(M)
C         ENTRADA:
C                HACER CASO OMISO
C         SALIDA:
C               INFORMACION DE LA REDUCCION
C
C    MOLER C.B., MATRIX EINGENVALUE AND LEAST SQUARE COMPUTATIONS,
C    COMPUTER SCIENCE DEPARTAMENT,  STANFORD UNIVERSITY, STANFORD,
C    CALIFORNIA,1973, PP 4.1,- 4.15
C
C    ENCUENTRA LA REFLEXION CON CEROS DE A(I,K),I=K+1...M

          DO 6 K= 1,N
              ALPHA= 0.
              DO 1 I= K,M
                  U(I)= A(I,K)
                  ALPHA= ALPHA+U(I)*U(I)
    1         CONTINUE
              ALPHA= SQRT(ALPHA)
              IF(U(K).LT.0.) ALPHA= -ALPHA
              U(K)= U(K)+ALPHA
              BETA= ALPHA*U(K)
              A(K,K)= -ALPHA
              IF(BETA.EQ.0.0.OR.K.EQ.N) GOTO 6
C
C    SE APLICA LA REFLEXION A COLUMNAS RESTANTES DE A
C
```

```
            KP1= K+1
            DO 4 J= KP1,N
                GAMMA= 0.
                DO 2 I= K,M
                    GAMMA= GAMMA+U(I)*A(I,J)
    2           CONTINUE
                GAMMA= GAMMA/BETA
                DO 3 I= K,M
                    A(I,J)= A(I,J)-GAMMA*U(I)
    3           CONTINUE
    4       CONTINUE
    6   CONTINUE
        RETURN
C
C   EL FACTOR TRIANGULAR SE ALMACENA EN A(I,J) I.LE.J
C   LOS VECTORES QUE DEFINEN LA REFLEXION SE ALMACENAN EN U Y EN EL
C   RESTO DE A                          ,
        END
C*************************************************************************
        SUBROUTINE HOLVE(MDIM,M,N,A,U,B)
C
C   DECLARACION DE VARIABLES
C
C
        INTEGER MDIM,M,N
        REAL A(MDIM,N),U(M),B(M)
        REAL BETA,GAMMA,T
C
C   TEXTO:
C
C   SOLUCION  POR MINIMOS CUADRADOS DE UN SISTEMA SOBREDETERMINADO
C   ENCUENTRA X QUE MINIMIZA LA NORMA (A*X-B)
C
C   MDIM,M,N,A,U ...REULTADOS DE HECOMP
C   B= M-VECTOR
C
C   ENTRADA:
C         MIEMBRO DEL LADO DERECHO
C   SALIDA:
C   LOS PRIMEROS N COMPONENTES = LA SOLUCION ,X
C   ULTIMOS M-N COMPONENTES= TRANSFORMACION RESIDUAL
C   LA DIVISION POR CERO IMPLICA UNA MATRIZ A DE RANGO NO COMPLETO
C
C   MOLER C. B., MATRIX EINGENVALUE AND LEAST SQUARE COMPUTATIONS,
C   COMPUTER SCIENCE DEPARTAMAMENT, STANFORD UNIVERSITY, STANFORD,
C   CALIFORNIA, 1973, PP 4.1-4.15
C
C   APLICA REFLEXION A B
C
        DO 3 K= 1,N
            T= A(K,K)
            BETA= -U(K)*A(K,K)
            A(K,K)= U(K)
            GAMMA= 0.0
            DO 1 I= K,M
                GAMMA= GAMMA+A(I,K)*B(I)
            CONTINUE
    1
```

```
            GAMMA =GAMMA/BETA
            DO 2 I= K,M
                B(I)= B(I)-GAMMA*A(I,K)
2           CONTINUE
            A(K,K)= T
3       CONTINUE
C
C     SUSTITUCION REGRESIVA
C
        DO 5 KB= 1,N
            K= N+1-KB
            B(K)= B(K)/A(K,K)
            IF(K.EQ.1) GOTO 5
            KM1= K-1
            DO 4 I= 1,KM1
                B(I)=B(I)-A(I,K)*B(K)
4           CONTINUE
5       CONTINUE
        RETURN
        END
C     ************************************************************
        SUBROUTINE MULVWF(A,C,AV,CW,NM1,M)
        REAL A(NM1,NM1),C(NM1,NM1),AV(NM1,M),CW(NM1,M)
C
C     SUBRUTINA QUE MULTIPLICA A*V Y C*W SIENDO V Y W LAS SIME-
C     TRIAS PARA UNA SPLINE PERIODICA
C
C     LAS VARIABLES SON EXPLICADAS EN PERNO
C
C     CALCULO DE AV
C
30      CONTINUE
        DO 35 J=1,NM1
            DO 40 I=1,M
                K=M*2-I
                L=M*2-2+I
                L1=M*4-I-2
                AV(J,I)=A(J,I)+A(J,K)-A(J,L)-A(J,L1)
                IF (I.EQ.1)AV(J,I)=A(J,I)+A(J,K)
                L3=M*3-2
                IF (I.EQ.M) AV(J,I)=A(J,I)-A(J,L3)
40          CONTINUE
35      CONTINUE
C
C     CALCULO DE LA MATRIZ CW
C
        DO 50 J=1,NM1
            DO 60 I=1,M
                L0=M*2-I
                L1=M*2-2+I
                L2=M*4-I-2
                L3=M*3-2
                CW(J,I)=C(J,I)+C(J,L0)-C(J,L1)-C(J,L2)
```

```
                IF(I.EQ.1) CW(J,I)=C(J,I)+C(J,L0)
                IF(I.EQ.M) CW(J,I)=C(J,I)-C(J,L3)
60          CONTINUE
50      CONTINUE
        RETURN
        END
```

```
C    ***********************************************************************
C    *                                                                     *
C    *    ESTE PROGRAMA OBTIENE LA SEGUNDA DERIVADA DE Y CON RES-          *
C    *    PECTO A X, DE UNA SPLINE PERIODICA Y=Y(X) DADA.                  *
C    *                                                                     *
C    *    PROGRAMO Y REALIZO: MIGUEL A. ALVARADO RASCON.                   *
C    *    INSTALACION : PDP 11/40 LAB CAD-DEPFI-UNAM                       *
C    *    PROYECTO : PAQUETE DE SINTESIS DE CURVAS                         *
C    *    FECHA DE REVISION : 30/MARZO/1984.                               *
C    *                                                                     *
C    ***********************************************************************
C
C   DECLARACION DE VARIABLES
C
      REAL X(57),Y(57),Z2D(57),DELTAX(56),G(112),A(56),B(56)
      REAL C(56),A1(56,56),B1(56)
C
C   PREGUNTA POR LA CANTIDAD DE PUNTOS M
C
      TYPE 14
      ACCEPT 15,M
      N= M*4-3
      NN= N-1
      NN2= NN*2
      CALL COSCU (N,NN,NN2,X,Y,Z2D,DELTAX,G,A,B,C,A1,B1)
C
C   ZONA DE FORMATOS
C
14    FORMAT (3X,'NUMERO DE PUNTOS M =  ',$)
15    FORMAT (I4)
      CALL EXIT
      END
C   ***********************************************************************
      SUBROUTINE COSCU (N,NN,NN2,X,Y,Z2D,DELTAX,G,A,B,C,A1,B1)
C   TEXTO:
C   DADA UNA SPLINE PERIODICA Y=Y(X) , LOS  VALORES DE Y EVALUADA
C   EN LOS PUNTOS DE APOYO SE ALMACENAN  EN  EL  ARREGLO Y, Y LOS
C   VALORES DE  LAS ORDENADAS DE LOS PUNTOS DE APOYO SE ALMACENAN
C   EN EL ARREGLO X ; ESTE  PROGRAMA  OBTIENE LA SEGUNDA DERIVADA
C   DE Y CON RESPECTO A X EN LOS PUNTOS DE APOYO ALMACENANDO ESTOS
C   EN EL ARREGLO Z2D
C
C   ENTRADA:
C   X...ABSCISAS DE APOYO DE LA SPLINE. DIMENSION N
C   Y...ORDENADAS DE APOYO DE LA SPLINE. DIMENSION N
C   N...NUMERO DE PUNTOS DE APOYO
C
C   SALIDA:
C   Z2D...VECTOR QUE CONTIENE LA SEGUNDA DERIVADA DE Y CON RESPECTO
C   A X. DIMENSION N
C
C   VARIABLES AUXILIARES:
C   A1,B1...MATRIZ Y VECTOR DEL SISTEMA AY''=6CY. DONDE, A1=A Y
C   B1=6CY
C   A(K),B(K),C(K),D(K)---(K=1...N-1)...COEFICIENTES DE LOS POLI--
C
```

```
C  NOMIOS CUBICOS DE LAS SPLINES
C
C    DECLARACION DE VARIABLES
C
          REAL XAU(100),F1(100),F1S(100),F2S(100)
          REAL X(N),Y(N),Z2D(N),DELTAX(NN),G(NN2)
          REAL A(NN),B(NN),C(NN),A1(NN,NN),B1(NN)
          NN=N-1
          N2=NN+NN
C
C    LLAMANDO AL ARCHIVO PUNTOS.MAR PARA LA LECTURA DE DATOS
C
          CALL ASSIGN (2,'RK1:PUNTOS.MAR')
          READ (2,5) (X(I),Y(I),I=1,N)
          TYPE 14
          TYPE 15 ,(I,X(I),Y(I),I=1,N)
C
C    CALCULANDO LA MATRIZ A1 Y EL VECTOR B1
C
          CALL MATAF (N,NN,X,Y,B1,A1,DELTAX)
C
C    DESCOMPOSICION DE LA MATRIZ A1 POR EL METODO DE  CHOLESKY
C    (GILBERT STRANG, ALGEBRA LINEAL Y SUS APLICACIONES, 1982,
C    FONDO EDUCATIVO INTERAMERICANO,MEXICO , PP30-40)
C
          CALL ACAG (NN,NN2,A1,G)
C
C    SOLUCION DEL SISTEMA A1*Z2D=B1 POR SUSTITUCION REGRESIVA
C
          CALL SOLU (N,NN,NN2,G,Z2D,B1)
C
C    IMPRIMIENDO EL RESULTADO
C
          TYPE 16
          TYPE 17, (I,Z2D(I),I=1,N)
C
C    OBTENCION DE LOS COEFICIENTES A(K),B(K),C(K),K=1,1,..NN
C
          CALL COEFF (N,NN,X,Y,Z2D,A,B,C)
C
C    INTERPOLANDO
C
          CALL CASPLI (N,NN,X,A,B,C,Y,XAU,F1,F1S,F2S)
C
C    ZONA DE FORMATOS
C
    5     FORMAT (2F8.8)
   14     FORMAT(3X,'    I     X(I)     Y(I)',)
   15     FORMAT (3X,I4,2F8.4)
   16     FORMAT(3X,'    I   Z2D(I)')
   17     FORMAT (3X,I4,F8.4,3X,)
          CALL EXIT
          END
C   ****************
```

```
      $
C     ************************************************************************
          SUBROUTINE SOLU (N,NN,NN2,G,X,B)
C
C     TEXTO:
C     SOLUCION DEL SISTEMA AX=B.........(L)
C
C     ENTRADA:
C     G...VECTOR QUE CONTIENE LOS ELEMENTOS DE LA DESCOMPOSICION  DE
C     CHOLESKY DE LA MATRIZ A DEL SISTEMA (L). DIMENSION NN2
C     B...VECTOR DE DIMENSIN NN. MIEMBRO DERECHO DE LA ECUACION L
C
C     SALIDA:
C     X...VECTOR SOLUCION DE DIMENSION NN
C
C     DECLARACION DE VARIABLES
C
          REAL G(NN2),X(N),B(NN)
C
          X(1)=B(1)/G(1)
          DO 10 I= 2,NN
               J= I-1
               J1= J+NN
               J2= I+NN
               G1= 0.
               IF (I.EQ.NN) G1= G(J2)
               X(I)= (B(I)-X(J)*G(J1)-X(1)*G1)/G(I)
   10     CONTINUE
          X(NN)= X(NN)/G(NN)
          J= NN-1
          DO 20 I= 2,NN
               K= J+1
               J1= J+NN
               G1=0.
               IF (I.EQ.NN) G1= G(NN2)
               X(J)= (X(J)-X(K)*G(J1)-X(NN)*G1)/G(J)
               J= J-1
   20     CONTINUE
          X(N)=X(1)
          RETURN
          END
C************************************************************************
          SUBROUTINE ACAG (NN,NN2,A,G)
C
C     TEXTO:
C     DADA LA MATRIZ A DE ORDEN NN*NN (MATRIZ NO SINGULAR ,P.D.
C     (XT*A*X>0),TRIDIAGONAL SE OBTIENEN LOS VECTORES ALFA Y GE
C     POR MEDIO DE LA DESCOMPOSICION DE CHOLESKY (VER SUB. COSCU)
C
C     ENTRADA:
C     A...MATRIZ DE DIMENSION (NN)X(NN)
C
C     SALIDA:
C     G...VECTOR QUE  CONTIENE  LAS  ALFAS Y  LAS GES DE LA DES-
C
```

```
C    COMPOSICION  DE CHOLESKY.  DONDE, LAS  ALFAS  SON LOS PRI-
C    MEROS NN ELEMENTOS Y  LAS  GES SON LOS SEGUNDOS ELEMENTOS
C    NN, QUEDANDO SU DIMENSION DE NN2=NN*2.
C
C    DECLARACION DE VARIABLES
C
          REAL A(NN,NN),G(NN2)
C
          G(1)=SQRT(A(1,1))
          G(NN2)=A(1,NN)/G(1)
          DO 10 I= 2,NN
               J= I-1
               J2= J+NN
               G1= 0.
               J1= J2+1
               K= I+1
               G(J2)= A(J,I)/G(J)
               IF (I.EQ.NN) G1=G(J1)
               G(I)= SQRT(A(I,I)-G(J2)**2-G1**2)
   10     CONTINUE
          RETURN
          END
C    ***************************************************************
          SUBROUTINE MATAF (N,NN,X,Y,B,A,DELTAX)
C
C    TEXTO:
C    SPLINE PERIODICA Y=Y(X).
C    ESTE PROGRAMA OBTIENE LA MATRIZ A DE LA RELACION AY''=6CY
C    Y EL VECTOR B=6CY.

C    ENTRADA:
C    X,Y...COORDENADAS DE LOS PUNTOS DE APOYO DE LA SPLINE. DIMEN-
C    SION N
C
C    SALIDA:
C    A...MATRIZ DE LA ECUACION AX=6CY. DIMENSION (N-1)X(N-1)
C    B... VECTOR SOLUCION QUE CORRESPONDE A SER B=6CY. DIMENSION
C    DE N-1
C
C    DECLARACION DE VARIABLES
          REAL DELTAX(NN),X(N),Y(N),B(NN),A(NN,NN)
C
C    CALCULO DE DELTAX
C
          NM1= N-1
          NM2= N-2
          DO 10 I= 1,NM1
   10     DELTAX(I)= X(I+1)-X(I)
C
C    CALCULO DE LA MATRIZ A
C
          DO 20 I= 1,NM1
               DO 20 J= 1,NM1
   20     A(I,J)= 0.
          DO 30 I= 1,NM1
```

```
            IM1= I-1
            IP1= I+1
            IF (I.EQ.1) IM1=NM1
            IF (I.EQ.NM1) IP1= 1
            A(I,IM1)= DELTAX(IM1)
            A(I,IP1)= DELTAX(I)
            A(I,I)= A(I,IM1)+A(I,IP1)
 30     A(I,I)= A(I,I)+A(I,I)
C
C   CALCULO DEL VECTOR B
C
        DO 50 I= 1,NM1
            J= I+1
            K= I-1
            IF (I.EQ.1) K=NM1
            K1= K+1
            B(I)= 0.
            AV1= (Y(J)-Y(I))/DELTAX(I)
            AV2= (Y(K1)-Y(K))/DELTAX(K)
            B(I)= 6.*(AV1-AV2)
 50     CONTINUE
        RETURN
        END
C   ***********************************************************
        SUBROUTINE SPLINE (N,NN,T,A,B,C,D,ARG,F,FS,FSS)
C
C   TEXTO:
C   ESTA SUBRUTINA INTERPOLA UN ARGUMENTO ENTRE 2 PUNTOS DE APO-
C   YO DE UNA SPLINE
C
C   ENTRADA:
C   T...PARAMETRO INDEPENDIENTE DE LOS PUNTOS DE APOYO DE LA SPLINE,
C   DE DIMENSION N
C   A,B,C...COEFICIENTES DE LOS POLINOMIOS CUBICOS DE LA SPLINE,
C   DE DIMENSION NN
C   D...TERMINO INDEPENDIENTE DE LOS POLINOMIOS CUBICOS ; A SU
C   VEZ SON LAS ORDENADAS DE LOS PUNTOS DE APOYO DE LA SPLINE.
C   DIMENSION N
C   ARG...ES EL ARGUMENTO QUE SE PRETENDE INTERPOLAR
C
C   SALIDA:
C   F... VALOR INTERPOLADO PARA EL ARGUMENTO DADO
C   FS...PRIMERA DERIVADA
C   FSS...SEGUNDA DERIVADA
C
C   DECLARACION DE VARIABLES
        REAL A(NN),B(NN),C(NN),D(N),T(N)
        EPS= 1.E-6
        X= (ARG-T(1)+EPS)*(ARG-T(N)-EPS)
        IF (X.GT.0.0) GOTO 30
        DO 20 K= 1,NN
            IF (ABS(ARG-T(K)).GT.EPS) GOTO 10
            F= D(K)
            FS= C(K)
```

```
            FSS= B(K)+B(K)
            RETURN
   10       CONTINUE
            IF ((T(K)-EPS).GT.ARG) GOTO 20
            IF ((T(K+1)-EPS).LT.ARG) GOTO 20
            X= ARG-T(K)
            X2= X*X
            X3= X2*X
            F= A(K)*X3+B(K)*X2+C(K)*X+D(K)
            FS= 3.*A(K)*X2+2.*B(K)*X+C(K)
            FSS= 6.*A(K)*X+2.*B(K)
            RETURN
   20       CONTINUE
            F= D(N)
            FS= C(1)
            FSS= B(1)+B(1)
            RETURN
   30       TYPE 14
   14       FORMAT (3X,'ARGUMENTO INVALIDO')
            END
C   ***********************************************************************
            SUBROUTINE CASPLI (N,NN,X,A,B,C,Y,XAU,F1,F1S,F2S)
C
C   TEXTO:
C   SUBRUTINA QUE AYUDA A INTERPOLAR UNA SPLINE DADA, GENERAN-
C   DO UN  NUMERO FINITO  DE  ARGUMENTOS IGUALMENTE ESPACIADOS
C   DENTRO DE UN INTERVALO DE INTERES
C
C   ENTRADA:
C   A,B,C,D=Y...COEFICIENTES DE LA SPLINE
C   X... PARAMETRO INDEPENDIENTE DE LA SPLINE
C   Y...PARAMETRO DEPENDIENTE DE LA SPLINE
C   SALIDA:
C   XAU...VECTOR QUE CONTIENE LOS ARGUMENTOS GENERADOS
C   F1...VECTOR QUE CONTIENE LAS ORDENADAS
C   F1S...VECTOR QUE CONTIENE LA PRIMERA DERIVADA
C   F2S...VECTOR QUE CONTIENE LA SEGUNDA DERIVADA
C
C   DECLARACION DE VARIABLES
C
            REAL XAU(1),F1(1),F1S(1),F2S(1),A(NN),B(NN),C(NN)
            REAL X(N),Y(N)
            DOUBLE PRECISION ALF,CONT,NO,ITER
            DATA NO/'NO'/
            DATA ITER/'ITER'/
C
C   PREGUNTAR QUIEN GENERA LOS ARGUMENTOS
C
            TYPE 195
            ACCEPT 19,ALF
            IF (ALF.EQ.ITER) GOTO 21
   26       TYPE 225
C
C   PIDIENDO ARGUMENTO AL USUARIO
C
            ACCEPT 22,ARG
C
```

```
C
C    LLAMANDO A LA SUBRUTINA QUE INTERPOLA
C
          CALL SPLINE (N,NN,X,A,B,C,Y,ARG,F,FS,FSS)
C
C    IMPRIMIENDO RESULTADOS
C
          TYPE 29,F
C
C    PREGUNTA SI EL USUARIO CONTINUA
C
          TYPE 245
          ACCEPT 19,CONT
          IF (CONT.NE.NO) GOTO 26
          GOTO 27
C
C    PREGUNTANDO CUANTOS PUNTOS SE VAN A GENERAR
C
   21     TYPE 285
          ACCEPT 28,IN
          DIV= FLOAT(IN)-1.
          ARG= X(1)
          R=0.
C
C    GENERANDO LOS ARGUMENTOS Y OBTENIENDO RESULTADOS
C
          DIF= X(N)-X(1)
          TYPE 32
          DO 23 I= 1,IN
               IF (I.EQ.1) GOTO 34
               R= R+1.
               ARG= DIF*R/DIV+X(1)
C
C    LLAMANDO A LA SUBRUTINA QUE INTERPOLA
C
   34          CALL SPLINE (N,NN,X,A,B,C,Y,ARG,F,FS,FSS,IO,IO)
               XAU(I)= ARG
               F1(I)= F
               F1S(I)= FS
               F2S(I)= FSS
               TYPE 33,I,F,FS,FSS,ARG
   23     CONTINUE
C
C    GUARDANDO RESULTADOS EN EL ARCHIVO POLAR.MAR
C
          CALL ASSIGN (3,'RK1:POLAR.MAR')
          WRITE (3,28) IN
          WRITE (3,71) (XAU(I),F1(I),F2S(I),I=1,IN)
C
C    ZONA DE FORMATOS
C
   19     FORMAT (A8)
   22     FORMAT(F8.4)
   28     FORMAT(I4)
```

```
 29     FORMAT(3X,'ABSCISA CORRESPONDIENTE = ',F8.4)
 32     FORMAT(3X,'   I     F(I)    FS(I)    FSS(I)   ARG(I)')
 33     FORMAT(3X,I4,4F8.4,3X,)
 71     FORMAT (3F8.4)
 195    FORMAT (3X,'FORMA DE LECTURA = (ITER,TELE)',3X,$)
 225    FORMAT (3X,'ARGUMENTO = ',$)
 245    FORMAT(3X,'CONTINUAS (? (NO/RET)--',$)
 285    FORMAT (3X,'NUMERO DE PUNTOS = ',$)
 27     RETURN
        END
C   ***********************************************************************
        SUBROUTINE COEFF(N,NN,X,Y,Z2D,A,B,C)
C
C   TEXTO:
C   ESTE PROGRAMA CALCULA LOS COEFICIENTES DE LOS POLINOMIOS CU-
C   BICOS DE LA SPLINE DADA
C
C   ENTRADA:
C   N...NUMERO DE PUNTOS DE APOYO
C   X...ABSCISAS DE LOS PUNTOS DE APOYO, DE DIMENSION N
C   Y...ORDENADAS DE LOS PUNTOS DE APOYO, DE DIMENSION N
C
C   SALIDA:
C   A,B,C...COEFICIENTES DE LOS POLINOMIOS CUBICOS DE LA SPLINE.
C   SU DINMENSION ES  NN=N-1
C
C   DECLARACION DE VARIABLES
C
        REAL X(N),Y(N),Z2D(N),A(NN),B(NN),C(NN)
C
        DO 10 I= 1,NN
 10     B(I)=Z2D(I)
        NM1= N-1
        K1= 2
        DO 5 K= 1,NM1
C
C   SOLO PARA SPLINE PEROIODICA
C
            IF (K.NE.NM1) GOTO 4
            Y(K1)= Y(1)
  4         A(K)= Z2D(K1)-B(K)
            A(K)= A(K)/(6.*(X(K1)-X(K)))
            B(K)= B(K)/2.
            C(K)= Z2D(K1)/2.+B(K)+B(K)
            C(K)= (C(K)+C(K))*(X(K1)-X(K))/6.
            C(K)= (Y(K1)-Y(K))/(X(K1)-X(K))-C(K)
  5     K1= K1+1
        RETURN
        END
```

```
C    ************************************************************
C    *                                                          *
C    *    ESTE PROGRAMA SINTETIZA UNA SPLINE NATURAL PARAMETRICA*
C    *                                                          *
C    *    PROGRAMO Y REALIZO : MIGUEL A. ALVARADO RASCON        *
C    *    INSTALACION :PDP 11/40 LAB CAD-DEPFI-UNAM             *
C    *    PROYECTO : PAQUETE DE SINTESIS DE CURVAS              *
C    *                                                          *
C    *                                                          *
C    ************************************************************
C
C    TEXTO:
C    DADO EL VECTOR PRESCRITO DE CURVATURA CU(I),I=1,2,...,N, SE
C    SINTETIZA UNA SPLINE CUBICA NATURAL PARAMETRICA X=X(T),
C    Y=Y(T) PARA LOS VALORES PRESCRITOS DE CURVATURA
C
C    PARAMETROS:
C    N...NUMERO DE PUNTOS DE APOYO
C    NN=N-1
C    N2=N-2
C    P...VECTOR QUE CONTIENE LA SIGUIENTE INFORMACION
C    X(1),Y(1),DELX,DELY,DELT,X1D,Y1D,X2D,Y2D,TE,CU
C    DE LA FORMA : 1,1,NN,NN,NN,NN,N,N,N2,N2,N,N2
C    DECLARACION DE VARIABLES:
C    DELX(I)...=X(I+1)-X(I),I=1,2,...,NN. DIMENSION NN
C    DELY(I)...=Y(I+1)-Y(I),I=1,2,...,NN. DIMENSION NN
C    DELT(I)...SQRT(DELT(I)**2+DELY(I)**2). DIMENSION NN
C    TE(I)...ANGULO DE LOS RADIOVECTORES DE LOS PUNTOS DE APOYO
C    CU(I)...DATO PRESCRITO. EN ESTE CASO ES LA CURVATURA
C    RAD1...RADIOVECTOR INICIAL PARA EL PUNTO 1
C    RAD2...RADIOVECTOR INICIAL PARA EL PUNTO N
C    PES...FACTOR DE PONDERACION
C    CO... FACTOR DE CURVATURA
C
C    DECLARACION DE VARIABLES
C
      EXTERNAL FUN ,DFDX
      REAL TE(11),RAD(11),CU(9),DELX(11),DELY(11),DELT(11),P(96)
      NDIM1=11
      NDIM2=NDIM1-1
      NDIM3=NDIM2-1
      NP=NDIM1*9-3
      TYPE 14
      ACCEPT 24,N
      TYPE 34
      ACCEPT 20,RAD1,RAD2,TE(N),PES,CO
      NN=N-1
      N2=NN-1
C
C    TRASLACION DE EJES
C
      PIEN4=ATAN(1.)
      TEAUX=TE(N)-90.
      TE(N)=180.-TE(N)
```

```
      TEAUX=TEAUX*PIEN4*4./180.
      TE(N)=TE(N)*PIEN4*4./180.
      TE(1)=0.
C
C     CALCULO DE LOS RADIOVECTORES INICIALES
C
      RAD(N)=(RAD1+RAD2*SIN(TEAUX))/SIN (TE(N))
      RAD(1)=RAD(N)*COS(TE(N))+RAD2*COS(TEAUX)
      DEL=(RAD(N)-RAD(1))/NN
      DO 2 I=2,NN
    2 RAD(I)=RAD(I-1)+DEL
      X2= RAD(1)*COS(TE(1))
      Y2=RAD(1)*SIN(TE(1))
      XP=X2
      YP=Y2
C
C     PARA ANGULOS IGUALMENTE ESPACIADOS EN EL INTERVALO DE INTERES
C
C
      DO 3 I=2,NN
    3 TE(I)=((TE(N)-TE(1))*FLOAT(I-1)/(N-1))+TE(1)
C
C     CALCULO DE LA CURVATURA PRESCRITA
C
      DO 30 I=2,NN
   30 CU(I-1)=CO*(SIN(PIEN4*4.*TE(I)/TE(N)))**2
C
C     CALCULO DE DELX,DELY, Y DELT
C
      DO 10 I=1,NN
         X1=X2
         Y1=Y2
         X2=RAD(I+1)*COS(TE(I+1))
         Y2=RAD(I+1)*SIN(TE(I+1))
         IF (I.EQ.N)GOTO 10
         DELX(I)=X2-X1
         DELY(I)=Y2-Y1
   10 DELT(I)=SQRT(DELX(I)*DELX(I)+DELY(I)*DELY(I))
C
      P(1)=XP
      P(2)=YP
      IN=2
      IND=IN+3*NN
      INDI=IND+N+N
      INDA=INDI+N2+N2
      NN2=NN+NN
C
C     LOS DATOS SE ALMACENAN EN P
C
      DO 25 I=1,N
         P(INDA+I+N2)=TE(I)
         IF (I.GT.N2)GOTO 35
         P(INDA+I)=CU(I)
   35    IF(I.EQ.N) GOTO 25
```

```
            IN =IN+1
            P(IN)=DELX(I)
            P(IN+NN)=DELY(I)
            P(IN+NN2)=DELT(I)
   25     CONTINUE
C
C    DEFINICION DE PARAMETROS PARA UTILIZAR LA SUBRUTINA NRDAMP O
C    NLLS
        K=INDA+NN+N
        NDIM=11
        P(K)=PES
        TOLX=1.E-5
        TOLF=1.E-4
        DAMP=.5
        MAX=20
        KMAX=20
C
C    SE LLAMA A NRDAMP SI LOS RADIOVECTORES 1 Y N  NO SE ESPECIFICAN
C
C        CALL NRDAMP(RAD,FUN,DFDX,P,TOLX,TOLF,DAMP,N,NDIM,ITER,MAX
C      $  KMAX)
C
C
C    SE LLAMA A NLLS SI SE ESPECIFICAN LOS RADIOVECTORES 1 Y N
C
C    NLLS ES UNA SUBRUTINA QUE RESUELVE  UN SISTEMA SOBREDETERMI-
C    NADO NO LINEAL, MINIMIZANDO LA NORMA EUCLIDIANA, DEL ERROR,
C    MEDIANTE EL METODO DE NEWTON RAPHSON
C    TOMADA DE: ANGELES J,SPATIAL KINEMATIC CHAINS,
C    SPRINGER-VERLAG, BERLIN, 1982, PP 306-307
C
        CALL NLLS(RAD,FUN,DFDX,P,TOLX,N,N2,ITER,MAX)
        BAND=1.
        CALL FUN (RAD,F,P,N,BAND)
C
C    ZONA DE FORMATOS
C
   14   FORMAT(3X,'NUMERO DE PUNTOS N=  ',$)
   20   FORMAT (5F8.4)
   24   FORMAT(I4)
   34   FORMAT(3X,'RADIO1,RADIO2,ANGULO FINAL,PESO,CURVATURA ',/)
        CALL EXIT
        END
C    ************************************************************
        SUBROUTINE FUN(X,F,P,N,BAND)
C
C    TEXTO:
C
C    ESTA SUBRUTINA EVALUA LA FUNCION F(Z) EN EL VALOR ACTUAL DE
C    Z, DONDE  F(Z)  REPRESENTA LA DIFERENCIA ENTRE LA CURVATURA
C    GENERADA EN LOS PUNTOS DE APOYO Y LA PRESCRITA
C
C    ENTRADA:
C    P...VER PROGRAMA PRINCIPAL
C
C    SALIDA:
C    F...VECTOR IE DIMENSION N, CUYOS VALORES SON LAS DIFERENCIAS
C
C
```

```
C    VARIABLES AUXILIARES:
C    X1D,Y1D,..PRIMERAS DERIVADAS, DIMENSION N
C    X2D,Y2D,..SEGUNDAS DERIVADAS, DIMENDON N2
C
      REAL X2D(9),Y2D(9),P(96),DELX(11),DELY(11),A(9,9),DELT(11)
      REAL X1D(11,1),Y1D(11,1),CU(11),F(11),X(11),AX(10),BX(10)
      REAL CX(10),AY(10),BY(10),CY(10),F2(11),G(11),H(11),TAU(100)
      REAL F1(100),F1S(100)
      INTEGER IP(9)
      NDIM1=11
      NDIM2=NDIM1-1
      NDIM3=NDIM2-1
      NP=NDIM1*9-3
      NN=N-1
      N2=NN-1
C
C   SE ACTUALIZA EL VECTOR P
C
      CALL ACTU(X,P,N,NN,N2,NDIM1,NP)
      IN =2
      IND=IN+3*NN
      INDI=IND+N+N
      INDA=INDI+N2+N2
      NN2=NN+NN
      K=INDA+N+NN
      FES=P(K)
      DO 10 I=1,NN
         IF(I.GT.N2)GOTO 15
         CU(I)=P(INDA+I)
15       IN =IN+1
         DELX(I)=P(IN)
         DELY(I)=P(IN+NN)
10       DELT(I)=P(IN+NN2)
      XP=P(1)
      YP=P(2)
C
C   CALCULO DE LA MATRIZ A
C
      CALL MATA(A,DELT,N2,NDIM2,NDIM3)
C
C   FACTORIZA A EN EL PRODUCTO LU
C
      CALL DECOMP(N2,NDIM3,A,IP)
C
C   CALCULO DE X2D Y Y2D
C
      CALL CALW2D(X2D,DELX,DELT,A,IP,N,NN,N2,NDIM2,NDIM3)
      CALL CALW2D(Y2D,DELY,DELT,A,IP,N,NN,N2,NDIM2,NDIM3)
C
C   CALCULO DE X1D Y Y1D
C
      CALL CALW1D(X1D,X2D,DELT,DELX,N,NN,N2,NDIM1,NDIM2,NDIM3)
      CALL CALW1D(Y1D,Y2D,DELT,DELY,N,NN,N2,NDIM1,NDIM2,NDIM3)
C
C   CALCULO DE F CON CURVATURA PRESCRITA
C
C   USAR FES=1 PARA CUANDO LOS RADIOVECTORES 1 Y N SON LIBRES
C
```

```
C
      DO 20 I=2,NN
         F11=(X1D(I,1)*X1D(I,1)+Y1D(I,1)*Y1D(I,1))**1.5
      F(I-1)=((X1D(I,1)*Y2D(I-1))-(X2D(I-1)*Y1D(I,1)))/F11
20    F(I-1)=(F(I-1)-CU(I-1))*PES
C
C   CALCULO DE F EN LA PENDIENTE PRESCRITA
C
      INTE=8*N-7
      TAN=SIN(P(INTE+N))/COS(P(INTE+N))
      F(NN)=X1D(1,1)
      F(N)=(X1D(N,1)+Y1D(N,1)*TAN)
C
C   ACTUALIZACION DE P
C
      DO 30 I=1,N
         IND=IND+1
         INDI=INDI+1
         P(IND)=X1D(I,1)
         P(IND+N)=Y1D(I,1)
         IF(I.GT.N2) GOTO 30
         P(INDI)=X2D(I)
         P(INDI+N2)=Y2D(I)
30    CONTINUE
      IF (BAND.EQ.0.)GO TO 25
C
C   CALCULO DE LOS COEFICIENTES DE LA SPLINE
C   PARA ESTO LOS VECTORES DELX,DELY,DELT CAMBIAN SUS VALORES
C
      IN=1
      DELX(1)=P(1)
      DELY(1)=P(2)
      DELT(1)=0.
      DO 35 I=2,N
         DELX(I)=DELX(I-1)+P(IN+I)
         DELY(I)=DELY(I-1)+P(IN+NN+I)
35    DELT(I)=DELT(I-1)+P(IN+NN2+I)
      CALL COEF(N,NN,DELT,DELX,X2D,AX,BX,CX)
      CALL COEF(N,NN,DELT,DELY,Y2D,AY,BY,CY)
      CALL COSPLI(N,NN,DELT,DELX,DELY,AX,BX,CX,DELX,AY,BY,CY,DELY,TAU,
     $  F1,F1S)
25    RETURN
      END
C   ************************************************************
      SUBROUTINE DFDX(X,X1DX,P,N)
C
C   TEXTO:
C   ESTA SUBRUTINA CALCULA LA MATRIZ JACOBIANA DEL VECTOR F CON
C   RESPECTO A LOS RADIOVECTORES DE LOS PUNTOS DE APOYO
C
C   ENTRADA:
C   P...VER PROGRAMA PRINCIPAL
C
C   SALIDA:
C   X1DX...MATRIZ QUE CONTIENE LA MATRIZ JACOBIANA
C
```

```
      $
C     X2DX,X2DY,Y2DX,Y2DY,..,VER V2DW
C     X1DX,X1DY,Y1DX,Y1DY,..,MATRICES QUE CONTIENEN LAS PARCIALES
C     DE LAS PRIMERAS DERIVADAS CON RESPECTO A LOS PUNTOS DE APOYO
C
      REAL CS(9,11),DELX(11),DELY(11),DELT(11),A(9,9),Y2DY(11,11)
      REAL X2DX(11,11),X1D(11),Y1D(11),X2DY(11,11),Y2DX(11,11)
      REAL X1DX(11,11),X1DY(11,11),Y1DX(11,11),Y1DY(11,11),X(11)
      REAL Y2D(9),X2D(9),P(96)
      INTEGER IP(9)
      NDIM1=11
      NDIM2=NDIM1-1
      NDIM3=NDIM2-1
      NP=NDIM1*9-3
      NN=N-1
      N2=NN-1
      IN =2
      IN0=IN+3*NN
      IND1=IN0+N+N
      INDA=IND1+N2+N2
      NN2=NN+NN
      K=INDA+N+NN
      PES=P(K)
      XP=P(1)
      YP=P(2)
      DO 10 I=1,N
         X1D(I)=P(IN0+I)
         Y1D(I)=P(IN0+I+N)
         IF(I.GT.NN) GOTO 15
         DELX(I)=P(IN+I)
         DELY(I)=P(IN+NN+I)
         DELT(I)=P(IN+NN2+I)
15       IF(I.GT.N2) GO TO 10
         X2D(I)=P(IND1+I)
         Y2D(I)=P(IND1+I+N2)
10    CONTINUE
C
C     CALCULO DE LA MATRIZ A
C
      CALL MATA(A,DELT,N2,NDIM2,NDIM3)
C
C     FACTORIZA LA MATRIZ A EN EL PRODUCTO LU
C
      CALL DECOMP(N2,NDIM3,A,IP)
C
C     CALCULO DE LA PARCIAL DE LA SEGUNDA DERIVADA DE X CON RESPEC-
C     TO A X. X2DX
C
      BAND=0.
      CALL V2DW(CS,DELT,X2DX,DELX,DELX,BAND,X2D,A,IP,N,NN,N2,NDIM1,
     $  NDIM2,NDIM3)
C
C     CALCULO DE LA PARCIAL DE LA SEGUNDA DERIVADA DE Y CON RESPEC-
C     TO A Y. Y2DY
C
5     CALL V2DW(CS,DELT,Y2DY,DELY,DELY,BAND,Y2D,A,IP,N,NN,N2,NDIM1,
```

```
     $   NDIM2,NDIM3)
C
C    CALCULO DE LA PARCIAL DE LA SEGUNDA DERIVADA DE X CON RESPEC-
C    TO A Y. X2DY
C
          BAND=1.
          CALL V2DW(CS,DELT,X2DY,DELX,DELY,BAND,X2D,A,IP,N,NN,N2,NDIM1,
     $    NDIM2,NDIM3)
C
C    CALCULO DE LA PARCIAL DE LA SEGUNDA DERIVADA DE Y CON RESPEC-
C    TO A X. Y2DX
C
          CALL V2DW(CS,DELT,Y2DX,DELY,DELX,BAND,Y2D,A,IP,N,NN,N2,NDIM1,
     $    NDIM2,NDIM3)
C
          DO 20 I=1,N2
               DO 20 J=1,N
                    X1DX(I,J)=X2DX(I,J)
                    X1DY(I,J)=X2DY(I,J)
                    Y1DX(I,J)=Y2DX(I,J)
                    Y1DY(I,J)=Y2DY(I,J)
   20     CONTINUE
C
C    MULTIPLICACION DE (-G/6)*(X2DX,X2DY,Y2DX,Y2DY)
C
          NES=NDIM1
          CALL MULG(X1DX,DELT,N,N,NN,N2,NDIM1,NDIM2,NES)
          CALL MULG(Y1DY,DELT,N,N,NN,N2,NDIM1,NDIM2,NES)
          CALL MULG(X1DY,DELT,N,N,NN,N2,NDIM1,NDIM2,NES)
          CALL MULG(Y1DX,DELT,N,N,NN,N2,NDIM1,NDIM2,NES)
C
C    RESTA CON (G11,G12,G21,G22)*1/6
C
          CALL RESTGS(X2D,DELX,DELT,X1DX,N,NN,N2,NDIM1,
     $    NDIM2,NDIM3)
          CALL RESTGS(X2D,DELY,DELT,X1DY,N,NN,N2,NDIM1,NDIM2,
     $    NDIM3)
          CALL RESTGS(Y2D,DELX,DELT,Y1DX,N,NN,N2,NDIM1,NDIM2,
     $    NDIM3)
          CALL RESTGS(Y2D,DELY,DELT,Y1DY,N,NN,N2,NDIM1,NDIM2,
     $    NDIM3)
C
C    SUMA CON LAS MATRICES F,F11,F12,F221,F22 PARA OBTENER X1DX,
C    X1DY,Y1DX,Y1DY
C
          BANF=1.
          CALL SUMF(X1DX,DELT,DELX,DELX,BANF,N,NN,NDIM1,NDIM2)
          CALL SUMF(Y1DY,DELT,DELY,DELY,BANF,N,NN,NDIM1,NDIM2)
          BANF=0.
          CALL SUMF(X1DX,DELT,DELX,DELX,BANF,N,NN,NDIM1,NDIM2)
          CALL SUMF(Y1DY,DELT,DELY,DELY,BANF,N,NN,NDIM1,NDIM2)
          CALL SUMF(X1DY,DELT,DELY,DELX,BANF,N,NN,NDIM1,NDIM2)
          CALL SUMF(Y1DX,DELT,DELY,DELX,BANF,N,NN,NDIM1,NDIM2)
C
```

```
C    DE AQUI EN ADELANTE SE UTILIZAN LOS VECTORES DELX,DEY Y DELT PA-
C    RA ALMACENAR OTROS DATOS.
C
          INTE=8*N-7
          TAN=SIN(P(INTE+N))/COS(P(INTE+N))
          DO 40 I=1,N
               CO=COS(P(INTE+I))
               SE=SIN(P(INTE+I))
               DELX(I)=X1DX(1,I)*CO+X1DY(1,I)*SE
               DELY(I)=X1DX(N,I)*CO+X1DY(N,I)*SE
               AUX=Y1DX(N,I)*CO+Y1DY(N,I)*SE
   40     DELY(I)=DELY(I)+AUX*TAN
C         DO 40 I=2,NN
C              CO=COS(P(INTE+I))
C              SE=SIN(P(INTE+I))
C              DELX(I-1)=X2DX(1,I)*CO+X2DY(1,I)*SE
C              DELY(I-1)=X2DX(N2,I)*CO+X2DY(N2,I)*SE
C   40    DELT(I-1)=Y2DX(N2,I)*CO+Y2DY(N2,I)*SE
C
C    MULTIPLICACION POR LAS MATRICES DIAGONALES SI LA CURVATURA
C    ES PRESCRITA
C
          BAND=0.
          CALL MULPRI(X1D,Y1D,X2D,Y2D,X1DX,BAND,N,NN,N2,NDIM1,NDIM3)
          CALL MULPRI(X1D,Y1D,X2D,Y2D,X1DY,BAND,N,NN,N2,NDIM1,NDIM3)
          CALL MULSEG(X1D,Y1D,Y2DX,BAND,N,NN,NDIM1)
          CALL MULSEG(X1D,Y1D,Y2DY,BAND,N,NN,NDIM1)
          BAND=1.
          CALL MULPRI(Y1D,X1D,Y2D,X2D,Y1DX,BAND,N,NN,N2,NDIM1,NDIM3)
          CALL MULPRI(Y1D,X1D,Y2D,X2D,Y1DY,BAND,N,NN,N2,NDIM1,NDIM3)
          CALL MULSEG(Y1D,X1D,X2DX,BAND,N,NN,NDIM1)
          CALL MULSEG(Y1D,X1D,X2DY,BAND,N,NN,NDIM1)
C
C    OBTENCION DE LA PARCIAL DE F CON RESPECTO A X
C
          BAND=0.
          CALL SUM4(X1DX,Y1DX,X2DX,Y2DX,BAND,N,N2,NDIM1)
C
C    OBTENCION DE LA PARCIAL DE F CON RESPECTO A Y
          CALL SUM4(X1DY,Y1DY,X2DY,Y2DY,BAND,N,N2,NDIM1)
C
C
C    MULTIPLICACION CON LAS MATRICES V Y W
          CALL MULVW(X1DX,X1DY,P,N,NDIM1,NP)
C
C
C    OBTENCION DEL JACOBIANO
C
          BAND=1.
          CALL SUM4(X1DX,X1DY,X2DY,Y2DX,BAND,N,N2,NDIM1)
C
C    CALCULO DE LOS ULTIMOS RENGLONES DE LA MATRIZ JACOBIANA
C
C    PARA CUANDO LOS RADIOVECRORES 1 Y N SON FIJOS
C
```

```
      DO 50 I=1,N2
         DO 50 J=1,N2
50    X1DX(I,J)=X1DX(I,J+1)*PES
      DO 60 I=2,NN
         X1DX(NN,I-1)=DELX(I)
60    X1DX(N,I-1)=DELY(I)
C
C   PARA CUANDO LOS RADIOVECTORES 1 Y N SON VARIABLES
C
C     DO 50 I=1,N
C        DO 50 J=1,N2
C 50  X1DX(J,I)=X1DX(J,I)*PES
C     DO 60 I=1,N
C        X1DX(NN,I)=DELX(I)
C 60  X1DX(N,I)=DELY(I)
30    RETURN
      END
C   ***********************************************************************
      SUBROUTINE V2DW(CS,DELT,AS,DELXY1,DELXY2,BAND,S2D,ATRI,
     $  IP,N,NN,N2,NDIM1,NDIM2,NDIM3)

C
C   TEXTO:
C OBTIENE:
C
C   PARCIAL DE LA SEGUNDA DERIVADA DE X CON RESPECTO A X SI:
C   X2DX
C        DELXY1=DELX
C        DELXY2= DELX
C        BAND=0.
C        S2D=X2D
C
C   PARCIAL  DE LA SEGUNDA DERIVADA DE Y CON RESPECTO A Y SI:
C   Y2DY
C        DELXY1=DELY
C        DELXY2=DELY
C        BAND=0. O DIF. DE 1.
C        S2D=Y2D
C
C   LA PARCIAL DE LA SEGUNDA DERIVADA DE X CON RESPECTO A Y SI:
C   X2DY
C        DELXY1=DELX
C        DELXY2=DELY
C        BAND=1.
C        S2D=X2D
C
C   LA PARCIAL DE  LA SEGUNDA DERIVADA DE Y CON RESPECTO A X SI:
C   Y2DX
C
C        DELXY1=DELY
C        DELXY2=DELX
C        BAND=1.
C        S2D=Y2D
C
C   ENTRADA:
C
C
```

```
C   ATRI...ES LA MATRIZ A YA DESCOMPUESTA EN LOS FACTORES LU.
C    DELT,DELXY1,DELXY2,BAND,S2D...VARIABLES QUE DEBEN DE ENTRAR EN EL
C    MODO QUE SE EXPLICO ANTERIORMENTE
C    N,NN,N2...ENTEROS QUE SE NECESITAN PARA DIMENSIONAR LAS MATRICES Y
C    HACER OPERACIONES.
C
C    SALIDA:
C    CS...MATRIZ AUXILIAR DE DIMENSION (N2XN)
C    AS...MATRIZ BUSCADA DE SALIDA QUE CONTIENE LA DERIVADA PARCIAL
C
C    DECLARACION DE VARIABLES:
C
          REAL CS(NDIM3,NDIM1),AS(NDIM1,NDIM1),ATRI(NDIM3,NDIM3)
          REAL DELT(NDIM2),DELXY1(NDIM2)
          REAL DELXY2(NDIM2),S2D(NDIM3)
          INTEGER IP(NDIM3),N,NN,N2
C
C    CALCULO DE CS
C
          CALL CES(CS,DELT,DELXY1,DELXY2,N,NN,N2,NDIM1,NDIM2,NDIM3)
          IF(BAND.EQ.1.)GOTO 5
C
C    CALCULO DE CS+C
C
          CALL SUMC(CS,DELT,N,NN,N2,NDIM1,NDIM2,NDIM3)
C
C    MULTIPLICACION POR 6.
C
  5       DO 10 I=1,N2
              DO 10 J=1,N
  10      CS(I,J)=6.*CS(I,J)
C
C    CALCULO DE LA MATRIZ AS
C
          CALL MATAS(DELT,DELXY2,AS,S2D,N,NN,N2,NDIM1,NDIM2,NDIM3)
C
C    RESTA DE LAS MATRICES
C
          DO 20 I=1,N2
              DO 20 J=1,N
  20      AS(I,J)=CS(I,J)-AS(I,J)
C
C    SUSTITUCION REGRESIVA DE ATRI CON AS
C
          CALL INV(ATRI,AS,IP,N,N2,NDIM1,NDIM3)
          RETURN
          END
C    ***************************************************************
          SUBROUTINE MATA(A,DELT,N2,NDIM2,NDIM3)
;
;  SUBRUTINA QUE CALCULA LA MATRIZ A DE LAS SPLINE NATURALES PA-
;   RAMETRICAS
;
;    ENTRADA:
;    DELT...VER PROGRAMA PRINCIPAL
;
;    SALIDA:
;    A...MATRIZ A DE LAS SPLINE PARAMETRICAS
```

```
      REAL A(NDIM3,NDIM3),DELT(NDIM3)
      DO 10 I= 1,N2
         DO 10 J=1,N2
10    A(I,J)=0.
C   CALCULO DE LA MATRIZ A
      DO 20 I=1,N2
         IF(I.EQ.1) GO TO 30
         A(I,I-1)=DELT (I)
         IF (I.EQ.N2) GO TO 40
30    A(I,I+1)= DELT(I+1)
40    A(I,I)=DELT(I)+DELT(I+1)
20    A(I,I)= A(I,I)+A(I,I)
      RETURN
      END
C  ***********************************************************************
      SUBROUTINE INV(IN,R,IP,NR,N,NDIMB,NDIMA)
C
C
C   SUBRUTINA QUE HACE LAS SUSTITUCIONES REGRESIVAS DE LA MATRIZ
C   IN CON LA MATRIZ R.
C   SE DEJA AL USUARIO LA DESCOMPOSICION DE IN EN LOS FACTORES LU
C
C   DATOS:
C   IN...DIMENSION NDIMB X NDIMB.
C   R...DIMENSION NDIMA XNDIMA
C   N...ORDEN DE LA MATRIZ IN Y NUMERO DE RENGLONES DE LA MATRIZ R
C   NR... NUMERO DE COLUMNAS DE LA MATRIZ R
C
C   SALIDA:
C   R...MATRIZ RESULTANTE
C
      REAL IN(NDIMA,NDIMA),AUX1(20),R(NDIMB,NDIMB)
      INTEGER IP(NDIMA)
      DO 10 I=1,NR
         DO 20 J=1,N
20       AUX1(J)=R(J,I)
         CALL SOLVE(N,NDIMA,IN,AUX1,IP)
         DO 10 J=1,N
10    R(J,I)=AUX1(J)
      RETURN
      END
C  ***********************************************************************
C
      SUBROUTINE SUMC(INC,DELT,N,NN,N2,NDIM1,NDIM2,NDIM3)
C
C   SUBRUTINA QUE SUMA UNA MATRIZ CUALQUIERA QUE SEA COMPATIBLE CON EL
C   ORDEN DE LA MATRIZ C
C
C   ENTRADA:
C   INC...MATRIZ QUE SE DESEA SUMAR A LA MATRIZ C
C   N,N2...ENTEROS QUE NOS DAN EL ORDEN DE LAS MATRICES
C
C   SALIDA:
C   INC...MATRIZ QUE SE OBTIENE DE LA SUMA INC+C
```

```
C
C    NOTA:
C    PARA OBTENER LA MATRIZ C SE SUMA UNA MATRIZ NULA
C
C
      REAL INC(NDIM3,NDIM1),DELT(NDIM2)
      DO 20 I= 1,N2
           INC(I,I)= INC(I,I)+1./DELT(I)
           INC(I,I+1)= INC(I,I+1)-1./DELT(I)-1./DELT(I+1)
20    INC(I,I+2)=INC(I,I+2)+1./DELT(I+1)
      RETURN
      END
C*************************************************************************
      SUBROUTINE MATAS(DELT,DELA,AS,S2D,N,NN,N2,NDIM1,NDIM2,NDIM3)
C
C    SUBRUTINA QUE CALCULA LAS MATRICES:
C
C    A11...SI DELA= DELX Y S2D= X2D
C    A12...SI DELA= DELY Y S2D= X2D
C    A21...SI DELA= DELY Y S2D= Y2D
C    A22...SI DELA= DELY Y S2D= Y2D
C
C    ENTRADA:
C    DELT,DELX,DELY...INCREMENTOS EXPLICADOS EN LA PROGRAMA PRINCIPAL
C    S2D...QUE PUEDE SER X2D O Y2D QUE SON SEGUNDAS DERIVADAS DE X Y Y
C    DELA...QUE PUEDE SER DELX O DELY
C
C    SALIDA:
C    AS...MATRIZ QUE REPRESENTA A11,A12,A21,A22,DEPENDIENDO DE LOS DATOS
C    DE ENTRADA
C
      REAL AS(NDIM1,NDIM1),DELA(NDIM2),DELT(NDIM2),S2D(NDIM3)
      DO 5 I=1,N2
           DO 5 J=1,N
5     AS(I,J)=0.
           N3=N2-1
           DO 10 I=2,N3
           C1=DELA(I)/DELT(I)
           C2=DELA(I+1)/DELT(I+1)
           AS(I,I)=(-S2D(I-1)-2.*S2D(I))*C1
           AS(I,I+2)=(2.*S2D(I)+S2D(I+1))*C2
10    AS(I,I+1)=-AS(I,I)-AS(I,I+2)
C
C    PARA LOS PRIMEROS RENGLONES
C
      C1=DELA(1)/DELT(1)
      C2=DELA(2)/DELT(2)
      AS(1,1)=-2.*S2D(1)*C1
      AS(1,3)=(2.*S2D(1)+S2D(2))*C2
      AS(1,2)=-AS(1,1)-AS(1,3)
C
C    PARA EL ULTIMO RENGLON
C
      C1=DELA(N2)/DELT(N2)
```

```
      C2=DELA(NN)/DELT(NN)
      AS(N2,N2)=(-S2D(N3)-2.*S2D(N2))*C1
      AS(N2,N)=2.*S2D(N2)*C2
      AS(N2,NN)=-AS(N2,N2)-AS(N2,N)
      RETURN
      END
C     ***************************************************************
      SUBROUTINE CES(CS,DELT,DELXY1,DELXY2,N,NN,N2,NDIM1,NDIM2,NDIM3)
C
C     SUBRUTINA QUE CALCULA LAS MATRICES:
C
C     C11...SI: DELXY1= DELX Y DELXY2= DELX
C     C22...SI: DELXY1= DELY Y DELXY2= DELY
C     C12=C21...SI: DELXY1=DELX Y DELXY2= DELY O VICEVERSA
C
C     ENTRADA:
C     DELT...PARAMETRO EXPLICADO EN LA SUBRUTINA PRINCIPAL DE DIMENSION NN
C     DELX,DELY...COMO DELXY1 O DELXY2,O VISCEVERSA, DE DIMENSION NN.ESTOS
C     PARAMETROS SE EXPLICAN EN LA PROGRAMA PRINCIPAL.
C     N,NN,N2...VARIABLES ENTERAS EXPLICADAS EN LA PROGRAMA PRINCIPAL
C
C     SALIDA:
C     CS...MATRIZ DE DIMENSION N2XN QUE REPRESENTA C11,C12,C21,C22. DEPEN-
C     DIENDO DE LOS PARAMETROS DE ENTRADA
C
C
      REAL CS(NDIM3,NDIM1),DELT(NDIM2),DELXY1(NDIM2),DELXY2(NDIM2)
C
C
      DO 10 I=1,N2
         DO 10 J=1,N
10    CS(I,J)=0.
      DO 20 I= 1,N2
         C1= DELXY1(I)/DELT(I)
         C11=DELXY2(I)/DELT(I)
         C2=DELXY1(I+1)/DELT(I+1)
         C21=DELXY2(I+1)/DELT(I+1)
         CS(I,I)=-C1*C11/DELT(I)
         CS(I,I+2)=-C2*C21/DELT(I+1)
20    CS(I,I+1)=-CS(I,I)-CS(I,I+2)
      RETURN
      END
C***************************************************************
      SUBROUTINE CALW2D(W2D,DELXY,DELT,ATRI,IP,N,NN,N2,NDIM2,NDIM3)
C
C     SUBRUTINA QUE CALCULA LAS SEGUNDAS DERIVADAS EN LOS PUNTOS DE APOYO
C
C     ENTRADA:
C     ATRI...MATRIZ FACTORIZADA EN LU
C
C     SALIDA:
C     W2D...SEGUNDA DERIVADA EN LOS PUNTOS DE APOYO .
C     SI DELXY=DELX   SE  OBTIENE X2D
```

```
C    SI DELXY=DELY     SE OBTIENE Y2D
C
C    DECLARACION DE VARIABLES
C
          REAL W2D(NDIM3),ATRI(NDIM3,NDIM3),DELXY(NDIM2),DELT(NDIM2)
          INTEGER IP(NDIM3)
C
C    MULTIPLICACION DE C POR LOS PUNTOS DE APOYO
C
          DO 10 I=1,N2
   10          W2D(I)=6.*(DELXY(I+1)/DELT(I+1)-DELXY(I)/DELT(I))
          CALL SOLVE (N2,NDIM3,ATRI,W2D,IP)
          RETURN
          END
C*************************************************************************
          SUBROUTINE CALW1D(W1D,W2D,DELT,DELXY,N,NN,N2,NDIM1,NDIM2,
      *   NDIM3)
C
C    SUBRUTINA QUE OBTIENE LAS PRIMERAS DERIVADAS DE X Y DE Y CON
C    RESPECTO A T EVALUADAS EN LOS PUNTOS DE A POYO
C    ENTRADA:
C    WP...PRIMER PUNTO DE APOYO
C    DELT ,DELXY..INCEMENTOS EN LOS PUNTOS DE APOYO EXPLICADAS ANTERI-
C    ORMENTE
C    SALIDA:
C    W1D...VECTOR DE DIMENSION N QUE CONTIEE LOS RESULTADOS ESPERADOS
C
C    DECLARACION DE VARIABLES
C
          REAL W1D(NDIM1,1),W2D(NDIM3),DELT(NDIM2),DELXY(NDIM2)
C
C    MULTIPLICACION POR -G/6
C
          NR=1
          NES=NR
          DO 5 I=1,N2
    5     W1D(I,1)=W2D(I)
          CALL MULG(W1D,DELT,N,NR,NN,N2,NDIM1,NDIM2,NES)
C
C    CALCULO DE FY-G/6Y
C
          DO 10 I=1,NN
   10     W1D(I,1)=DELXY(I)/DELT(I)+W1D(I,1)
          W1D(N,1)=DELXY(NN)/DELT(NN)+W1D(N,1)
          RETURN
          END
C*************************************************************************
          FUNCTION FNORM(F,N)
C
C    FUNCION QUE ENCUENTRA EL MAYOR VALOR ABSOLUTO DE UN ARREGLO
          REAL F(11)
          FNORM=0.
          DO 10 I=1,N
   10     IF (FNORM.LT.ABS(F(I))) FNORM= ABS(F(I))
```

```
C       TYPE 14, FNORM
C  14    FORMAT (3X,'FNORM',3X,F8.4)
       RETURN
       END

C  ***********************************************************************
       SUBROUTINE NRDAMP (X,FUN,DFDX,P,TOLX,TOLF,DAMP,N,NDIM
     $  ,ITER,MAX,KMAX)
       REAL X(5),P(5),DF(5,5),DELTA(5),F(5)
       INTEGER IP(5)
C
C   ESTA SUBRUTINA DETERMINA LAS RAICES DE UN SISTEMA ALGEBRAI-
C   CO  NO  LINEAL  DE ORDEN N, POR MEDIO DEL METODO DE NEWTON-
C   RAPHSON (ISSACSON E. AND KELLER H.B. ANALISIS  OF NUMERICAL
C   METHODS, JHON WHILEY AND SONS, INC. NEW YORK 1966,PP,85-123)
C   CON AMORTIGUAMIENTO.
C
C   PARAMETROS:
C   X...VECTOR DE N VARIABLES DESCONOCIDAS
C   FUN...SUBRUTINA EXTERNA QUE CALCULA EL VECTOR F,QUE CONTIE-
C   NE LA FUNCION CUYAS RAICES SE VAN A CALCULAR
C   DFDX...SUBRUTINA EXTERNA QUE CALCULA LA MATRIZ JACOBIANA DEL
C   VECTOR F CON RESPECTO A X
C   P...VECTOR AUXILIAR DE DIMENSION APROPIADA,CONTIENE PARAME-
C   TROS DIFERENTES PARA CADA PROBLEMA.
C   TOLX...ESCALAR POSITIVO, TOLERANCIA IMPUESTA EN LA APROXIMA-
C   CION DE X
C   TOLF...ESCALAR POSITIVO,TOLERANCIA IMPUESTA EN LA APROXIMA-
C   CION DE F
C   DAMP...VALOR DEL AMORTIGUAMIENTO, PROVISTO  POR  EL USUARIO:
C   0<DAMP<1
C   ITER...NUMERO DE ITERACIONES EJECUTADAS
C   MAX...NUMERO MAXIMO PERMITIDO DE ITERACIONES
C   KMAX...NUMERO MAXIMO PERMITIDO DE AMORTIGUAMIENTOS POR ITE-
C   RACION
C   FUN Y DFDX SON DADAS POR EL USUARIO
C   LAS SUBRUTINAS DECOMP Y SOLVE RESUELVEN EL SISTEMA ALGE-
C   BRAICO LINEAL DF(X)*DELTA= - F(X).
C   DELTA ES EL DE CORRECCION PARA LA K-ESIMA ITERACION, EL ME-
C   TODO USADO  ES LA DESCOMPOSICION LU, (MOLER C.B. MATRIX COM-
C   PUTATIONS WHITH FORTRAN AND PAGING, COMUNICATIONS  OF THE
C   A.C.M, VOLUME 15 APRIL 1972 )
C
C   TOMADA DE   : ANGELES J, SPATIAL KINEMATIC CHAINS,
C   SPRINGER-VERLAG, BERLIN, 1982, PP 47
C
       IO=7
       KONT=1
       ITER=0
       BAND=0.
       CALL FUN (X,F,P,N,BAND)
       FNOR1= FNORM (F,N)
       IF (FNOR1.LE.TOLF) GO TO 4
   1   CALL DFDX (X,DF,P,N)
C
       CALL DECOMP (N,NDIM,DF,IP)
```

```
      K=0
C
C   SI LA MATRIZ DF ES SINGULAR, LA SUBRUTINA REGRESA AL PROGRA-
C   MA PRINCIPAL. DE OTRA MANERA, EL PROCEDIMIENTO SIGUE
C
      IF (IP(N).EQ.0) GOTO 14
      CALL SOLVE  (N,NDIM,DF,F,IP)
      DO 2 I=1,N
          DELTA(I)= F(I)
    2 DELNOR= FNORM (DELTA,N)
      IF (DELNOR.LT.TOLX) GOTO 4
      DO 3 I= 1,N
    3 X(I)= X(I)-DELTA(I)
      GO TO 5
    4 FNOR2= FNOR1
      GO TO 6
    5 CALL FUN (X,F,P,N,BAND)
      KONT=KONT+1
      FNOR2=FNORM(F,N)
    6 IF(FNOR2.LE.TOLF)GO TO 11
C
C   PROBANDO LA NORMA DE LA FUNCION. SI ESTA NO DISMINUYE SE IN-
C   TRODUCE AMORTIGUAMIENTO
C
      IF (FNOR2.LT.FNOR1)GO TO 10
      IF(K.EQ.KMAX)GO TO 16
      K=K+1
      DO 8 I=1,N
      IF (K.GE.2)GO TO 7
      DELTA(I)=(DAMP-1.)*DELTA(I)
      GO TO 8
    7 DELTA(I)=DAMP*DELTA(I)
    8 CONTINUE
      DELNOR = FNORM(DELTA,N)
      IF(DELNOR.LE.TOLX) GO TO 16
      DO 9 I=1,N
    9 X(I)=X(I)-DELTA(I)
      GO TO 5
   10 IF(ITER.GT.MAX)GO TO 16
      ITER=ITER+1
      FNOR1=FNOR2
      GO TO 1
   11 WRITE(IO,110) ITER,FNOR2,KONT
   12 DO 13 I=1,N
   13 WRITE(IO,120) I,X(I)
      RETURN
   14 WRITE(IO,130) ITER,KONT
      GO TO 12
   16 WRITE(IO,140)ITER,FNOR2,KONT
      GO TO 12
  110 FORMAT(5X,'ITERACION NO. ',I3,/,5X,'LA NORMA DE LA FUNC'
     $ ,'ION ES',E20.6,/,5X,'LA FUNCION FUE EVALUADA ',I3,'  VE'
     $ ,'CES',/,5X,'EL PROCEDIMIENTO CONVERGIO Y LA SOLUCION FUE'
     $ ,' :',/)
  120 FORMAT(5X,'X(',I3,')=',E20.6)
  130 FORMAT(5X,'EN LA ITERACION NO. ',I3,' LA MATRIZ'
```

```
     $  ' JACOBIANA ES SINGULAR',/,5X,'LA FUNCION FUE EVALUADA ',I3,
     $' VECES',/,
     $  5X,'EL VALOR DE X ES  ',/)
 140 FORMAT(10X,'NO CONVERGE EN ITERACION NO. ',I3,
     $  /,10X,'LA NORMA DE LA FUNCION ES ',E20.6,/,10X,
     $  'LA FUNCION FUE EVALUADA ',I3,' VECES',/,10X,
     $  'EL VALOR DE X ES: ',/)
        END

$
C    **********************************************************************
      SUBROUTINE NLLS(X,FUN,DFDX,P,TOL,N,N2,ITER,MAX)
      REAL X(11),F(11),DF(11,11),P(96),U(11),DELTA(11)
C
C    ESTE PROGRAMA OBTIENE  LA SOLUCION DE MINIMOS CUADRADOS  DE
C    UN SISTEMA NO LINEAL F(X)=0,DONDE F Y X SON VECTORES DE DI-
C    MENSION N Y  N2 , SIENDO N MAYOR QUE N2.EL PROCEDIMIENTO ES
C    ITERATIVO  Y EN CADA ITERACION CALCULA LA SOLUCION DE MINI-
C    MOS  CUADRADOS DEL SISTEMA LINEAL DF*DELTA= -F, DONDE DF ES
C    LA MATRIZ JACOBIANA DE DIMENSION  NXN2 DEL SISTEMA ORIGINAL,
C    CALCULADA  CON  EL VALOR PRESENTE  DE X.SE CALCULA LA SOLU-
C    CION LINEAL DE MINIMOS  CUADRADOS  EN CADA  ITERACION POR
C    MEDIO DE LAS REFLEXIONES DE HOUSEHOLDER ( BJOERCK A. AND G.
C    DAHLQUIST,  NUMERICAL  METHODS, PRENTICE  HALL, ENGLEWOOD
C    CLIFFS, N.J. ,1974, PP. 201-206,443,-444).
C
C    PARAMETROS:
C
C    X...=VECTOR DE INCOGNITAS DE DIMENSION N
C    F...=VECTOR  DE LA FUNCION  CUYA LA NORMA  EUCLIDIANA
C    VA A SER MINIMIZADA (DIMENSION N2)
C    DF...MATRIZ JACOBIANA DE F CON RESPECTO A X (DIMENSION MXN)
C    TOL...=VALOR REAL PEQUENO QUE DENOTA LA TOLERANCIA.
C    P...=VECTOR DE PARAMETROS QUE APARECEN EN FUN Y/O DFDX. SU
C    DIMENSION VARIA DE PROBLEMA A PROBLEMA.
C    ITER...=VARIABLE ENTERA QUE EXPRESA EL NUMERO DE ITERACIO-
C    NES.
C    MAX...=VARIABLE ENTERA QUE DA EL LIMITE DE ITERACIONES.
C
C    SUBRUTINAS AUXILIARES:
C
C    HECOMP= TRIANGULARIZA DE UNA MATRIZ RECTANGULAR
C    POR REFLEXIONES DE HOUSEHOLDER (MOLER C.B.,MATRIX EINGENVA-
C    LUE AND LEAST-SQUARE COMPUTATIONS,COMPUTER SCIENCE DEPARTA-
C    MENT, STANFORD UNIVERSITY, MARCH, 1973)
C
C    HOLVE...=RESUELVE EL SISTEMA TRIANGULAR POR SUSTITUCION RE-
C    GRESIVA (MOLER C. B.,OP. CIT.)
C    FUN...=CALCULA F
C    DFDX...=CALCULA DF
C    FNORM...CALCULA LA  NORMA MAXIMA DE UN VECTOR
C
C    TOMADA DE: ANGELES J, SPATIAL KINEMATIC CHAINS,
```

```
C     SPRINGER-VERLAG, BERLIN, 1982,PP 306,307
C
C
      BAND=0.
      NDIM1=11
      NDIM2=NDIM1-1
      NDIM3=NDIM2-1
      NP=NDIM1*9-3
      NN=N-1
      ITER=0
    1 ITER=ITER+1
      CALL FUN (X,F,P,N,BAND)
      FNOR=FNORM(F,N)
      IF(ITER.GT.MAX) GO TO 5
C
C     FORMANDO EL PROBLEMA LINEAL DE MINIMOS CUADRADOS
C
      CALL DFDX(X,DF,P,N)
      CALL HECOMP(NDIM1,N,N2,DF,U)
      CALL HOLVE(NDIM1,N,N2,DF,U,F)
C
C     CALCULO DE A CORRECCION ENTRE DOS ITERACCIONES SUCESIVAS
C
      DO 2 I=1,N2
      DELTA(I)=F(I)
    2 CONTINUE
      DELNOR=FNORM(DELTA,N2)
      IF (DELNOR.LT.TOL)GO TO 6
C
C     SI DELNOR ES GRANDE, REALIZA CORRECCIONES AL VECTOR X
C
    3 DO 4 I=1,N2
        X(I+1)=X(I+1)-DELTA(I)
    4 CONTINUE
      GO TO 1
    5 WRITE(7,101) ITER,FNOR
      GO TO 7
    6 WRITE (7,102) ITER,FNOR
    7 DO 8 I=1,N
    8 WRITE(7,103) I,X(I)
      RETURN
  101 FORMAT(5X,'EL PROCEDIMIENTO DIVERGE EN LA ITERACION NO.',I3/5X,
     $        'LA NORMA DE LA FUNCION ES  : ',E20.7/5X
     $        'EL VALOR ACTUAL DE X ES : '/)
  102 FORMAT(5X,'EN LA ITERACION NO. ',I3/5X,'LA NORMA DE LA  FUNCION'
     $      ,' ES ',E20.7/5X,'EL PROCEDIMIENTO CONVERGIO, LA SOLUCION'
     $      ,' ES :'/)
  103 FORMAT(5X,2HX(I2,3H)= F15.5)
      END
C     ***********************************************************
      SUBROUTINE MULVW(KDX,KDY,P,N,NDIM1,NP)
C
C     ENTRADA:
C     KKDX,KDY...MATRICES QUE CONTIENEN LAS DERIVADAS PARCIALES DE LAS
C     SEGUNDAS DERIVADAS CON RESPECTO ALAS COORDENADAS DE LOS PUNTOS
C     DE APOYO
C
C     SALIDA
C     KDX,KDY...MATRICES OBTENIDAS DE LAS MULTIPLICACIONES EN
```

```
C    ANTERIORMENTE
C
          REAL KDX(NDIM1,NDIM1),KDY(NDIM1,NDIM1),P(NP)
          INDA=8*N-7
          N2=N-2
          DO 10 J=1,N2
              DO 10 I=1,N
          KDX(J,I)=KDX(J,I)*COS(P(INDA+I))
   10     KDY(J,I)=KDY(J,I)*SIN(P(INDA+I))
          RETURN
          END


C    ***********************************************************************
          SUBROUTINE COSPLI (N,NN,T,X,Y,AX,BX,CX,DX,AY,BY,CY,DY,
         $  TAU,F1,F1S)
C
C    TEXTO:
C    SUBRUTINA QUE AYUDA A INTERPOLAR UNA SPLINE PARAMETRICA DA-
C    DA. GENERA  UN NUMERO FINITO DE ARGUMENTOS IGUALMENTE ESPA-
C    CIADOS DENTRO DE UN INTERVALO DE INTERES
C
C    ENTRADA:
C    T...PARAMETRO INDEPENDIENTE I. DIMENSION N
C    X,Y...VECTORES  CON LA ABSCISA Y ORDENADA DE LA SPLINE. DI-
C    MENSION N
C    AX,BX,CX,AY,BY,CY...COEFICIENTES  DE  LA SPLINE. DIMENSION
C    (N-1)
C
C    VARIABLES AUXILIARES
C    TAU...VECTOR QUE CONTIENE LOS ARGUMENTOS GENERADOS
C    F1...VECTOR QUE CONTIENE LAS ABSCISAS PARA LOS ARGUMENTOS
C    GENERADOS
C    F1S...VECTOR QUE CONTIENE LAS ORDENDAS PARA LOS ARGUMENTOS
C    GENERADOS
C
C    DECLARACION DE VARIABLES
          REAL T(1),X(1),Y(1),AX(1),BX(1),CX(1),DX(1),AY(1),BY(1)
          REAL CY(1),DY(1),TAU(1),F1(1),F1S(1)
C
          M= (N+3)/4
C
C    PREGUNTAR CUANTOS PUNTOS SE QUIEREN INTERPOLAR
C
          TYPE 285
          ACCEPT 28,IN
          DIV= FLOAT(IN)-1.
          ARG= T(1)
          R=0.
C
C    PARA INTERPOLAR EN LOS N PUNTOS
C
          DIF= T(N)-T(1)
C
C
C    PARA INTERPOLAR EN LOS M PUNTOS
C
C         DIF = T(M)-T(1)
```

```
C
C
C   COMIENZA LA INTERPOLACION
C
        DO 23 I= 1,IN
           IF (I.EQ.1) GOTO 34
           R= R+1.
           ARG= DIF*R/DIV+T(1)
C
C    INTERPOLACION EN LAS ABSCISAS
C
  34       CALL SPLINE (N,NN,T,AX,BX,CX,X,ARG,F,FS,FSS)
           TAU(I)= ARG
           F1(I)= F
C
C    INTERPOLACION EN LAS ORDENADAS
C
           CALL SPLINE (N,NN,T,AY,BY,CY,Y,ARG,F,FS,FSS)
           F1S(I)= F
  23    CONTINUE
        TYPE 6
        TYPE 7, (F1(I),F1S(I),I=1,IN)
C
C    GUARDANDO LOS RESULTADOS EN EL ARCHIVO POLAR.MAR
C
        CALL ASSIGN (2,'RK1:POLAR.MAR')
        READ(2,5) IN2,NUMVEC
        IF (IN2.EQ.0) GOTO 42
        READ (2,10) (F1(I+IN),F1S(I+IN),I=1,IN2)
  42    IN= IN+IN2
        NUMVEC=NUMVEC+1
        CALL CLOSE(2)
        CALL ASSIGN (2,'RK1:POLAR.MAR')
        WRITE (2,5) IN,NUMVEC
        WRITE (2,10) (F1(I),F1S(I),I=1,IN)
        CALL CLOSE (2)
C
C    ZONA DE FORMATOS
C
  5     FORMAT (2I4)
  6     FORMAT (5X,'ABSCISA',3X,'ORDENADA')
  7     FORMAT (3X,F8.4,3X,F8.4)
  10    FORMAT (2F8.4)
  28    FORMAT (I4)
  285   FORMAT(3X,'NUMERO DE PUNTOS= ',$)
        RETURN
        END

C   ************************************************************
      SUBROUTINE SUM4(A,B,C,D,BAND,N,N2,NDIM1)
C
C   SI BAND=1,SUMA LAS MATRICES (A+B) ;SI NO, SUMA (A+B+C+D)

C
C   SALIDA DE LA SUMA LA MATRIZ A,LAS DEMAS MATRICES NO SON AFECTADAS
C
```

```
      REAL A(NDIM1,NDIM1),B(NDIM1,NDIM1),C(NDIM1,NDIM1),D(NDIM1,NDIM1)
      DO 10 I=1,N2
         DO 10 J=1,N
            A(I,J)=A(I,J)+B(I,J)
            IF (BAND.EQ.1.)GOTO 10
            A(I,J)=A(I,J)+C(I,J)+D(I,J)
10    CONTINUE
      RETURN
      END
C*************************************************************************
      SUBROUTINE MULPRI(S1D1,S1D2,S2D1,S2D2,Z1DZ,BAND,N,NN,N2,
     $  NDIM1,NDIM3)
C
C    TEXTO:
C    SUBRUTINA  QUE MULTIPLICA LA MATRIZ DE LAS DERIVADAS DE LA
C    CURVATURA CON RESPECTO A LA PRIMERA DERIVADA
C
C    SI S1D1=X1D  S1D2=Y1D S2D1=X2D S2S2=Y2D Y BAND=0. OBTIENE
C    DIAGONAL D
C    SI S1D1=Y1D S1D2= X1D S2D1=Y2D S2D2=X2D Y BAND=1. OBTIENE
C    DIAGONAL E
C
C
C    LA SALIDA ES Z1DZ,QUE TAMBIEN ES ENTRADA
C    DECARACION DE LAS VARIABLES
      REAL S1D1(NDIM1),S1D2(NDIM1),S2D1(NDIM3),S2D2(NDIM3)
      REAL  Z1DZ(NDIM1,NDIM1)
      K=2
      J=NN
            DO 10 I=K,J
         DIA=0.
         A=S1D1(I)*S1D1(I)
         B=S1D2(I)*S1D2(I)
         IF(I.EQ.1.OR.I.EQ.N) GO TO 25
         DIA=(B-2.*A)*S2D2(I-1)+3.*S1D1(I)*S1D2(I)*S2D1(I-1)
         DIA=DIA/((A+B)**2.5)
25       IF (BAND.EQ.1.) DIA =-DIA
         DO 10 JJ=1,N
         M=I-1
10    Z1DZ(M,JJ)=Z1DZ(I,JJ)*DIA
C     SE NOTA QUE EL PRIMERO Y EL ULTIMO RENGLON SON NULOS. POR ESO
C    SE RECORREN
      RETURN
      END
C*************************************************************************
      SUBROUTINE MULSEG(S1D1,S1D2,Z2DZ,BAND,N,NN,NDIM1)
C
C    SUBRUTINA QUE MULTIPLICA LA MATRIZ DE DERIVADAS  DE LA CURVA-
C    TURA CON RESPECTO A LAS SEGUNDAS DERIVADAS
C
C    SI S1D1=X1D S1D2=Y1D  Y BAND=0. SE OBTIENE DIAGONAL G
C    SI S1D1=Y1D S1D2=X1D Y BAND=1. SE OBTIENE  DIAGONAL F
C    SALIDA Y ENTRADA Z2D
C
C    DECLARACION DE VARIABLES
C
```

```
      REAL S1D1(NDIM1),S1D2(NDIM1),Z2DZ(NDIM1,NDIM1)
C
C   NOTA:COMO EL PRIMER Y EL ULTIMO RENGLONES SON CEROS LOS RECORREMOS
C
      N2=NN-1
      DO 10 I=2,NN
         DIA=0.
         A=S1D1(I)*S1D1(I)
         B=S1D2(I)*S1D2(I)
         DIA=S1D1(I)/((A+B)**1.5)
         IF (BAND.EQ.1.) DIA =-DIA
            DO 10 J=1,N
            M=I-1
10    Z2DZ(M,J)=Z2DZ(I-1,J)*DIA
      RETURN
      END
C**********************************************************************
      SUBROUTINE SUMF(SF,DELT,DELXY1,DELXY2,BANF,N,NN,NDIM1,NDIM2)
C
C   ESTA SUBRUTINA SUMA LA MATRIZ F A LA SF
C   LA SALIDA ES LA MATRIZ SF, QUE TAMBIEN FUNCIONA COMO ENTRADA
C   SI BANF=1. SOLO SE SUMA LA MATRIZ F. NO IMPORTAN LAS ENTRA-
C   DAS DELXY1 Y DELXY2
C   SI BANF=0.
C   DELXY1= DELX DELXY2=DELX SE OBTIENE  F11
C   DELXY1=DELX DELXY2=DELY O VS, SE OBTIENE F21=F12
C   DELXY1=DELY DELXY2=DELY SE OBTIENE F22
C
C   DECLARACION DE VARIABLES
C
      REAL SF(NDIM1,NDIM1),DELT(NDIM2),DELXY1(NDIM2),DELXY2(NDIM2)
C
      DO 10 I=1,NN
         A=-1.
         IF(BANF.EQ.1.) GO TO 5
         A=(DELXY1(I)*DELXY2(I))/(DELT(I)*DELT(I))
5        SF(I,I)=SF(I,I)+(A/DELT(I))
10    SF(I,I+1)=SF(I,I+1)-(A/DELT(I))
C
C   PARA EL ULTIMO RENGLON N
C
      A=-1.
      IF (BANF.EQ.1.)GO TO 15
      A=(DELXY1(NN)*DELXY2(NN))/(DELT(NN)*DELT(NN))
15    SF(N,NN)=SF(N,NN)+(A/DELT(NN))
      SF(N,N)=SF(N,N)-(A/DELT(NN))
      RETURN
      END
C   **********************************************************************
      SUBROUTINE RESTGS(S2D,DELXY,DELT,GS,N,NN,N2,NDIM1,NDIM2
     $  ,NDIM3)
C
C   ENTRADA:
C   S2D... PUEDE SER Y2D O X2D
C   GS...MATRIZ QUE SE RESTA A(G11/6,G12/6,G21/6,G22/6)
```

```
C     SALIDA
C     GS.. DEPENDIENDO DE
C     DELXY=DELX S2D=X2D SE RESTA G11
C     DELXY=DELX S2D=Y2D SE RESTA G21
C     DELXY=DELY S2D=X2D SE RESTA G12
C     DELXY=DELY S2D=Y2D SE RESTA G22
C
C     DECLARACION DE LAS VARIABLES
C
      REAL S2D(NDIM3),DELXY(NDIM2),DELT(NDIM2),GS(NDIM1,NDIM1)
C
      DO 10 I=2,N2
        A=(-2.*S2D(I-1)-S2D(I))*DELXY(I)/(DELT(I)*6.)
        GS(I,I)=GS(I,I)-A
10    GS(I,I+1)=GS(I,I+1)+A
C
C     PARA EL PRIMER RENGLON
      A=-S2D(1)*DELXY(1)/(DELT(1)*6.)
      GS(1,1)=GS(1,1)-A
      GS(1,2)=GS(1,2)+A
C
C     PARA EL (N-1)-ESIMO RENGLON
C
      A= -2.*S2D(N2)*DELXY(NN)/(DELT(NN)*6.)
      GS(NN,NN)=GS(NN,NN)-A
      GS(NN,N)=GS(NN,N)+A
C
C     PARA EL N-ESIMO RENGLON
C
      A=S2D(N2)*DELXY(NN)/(DELT(NN)*6.)
      GS(N,NN)=GS(N,NN)-A
      GS(N,N)=GS(N,N)+A
      RETURN
      END
C**********************************************************************
      SUBROUTINE MULG(Z2DR,DELT,N,NR,NN,N2,NDIM1,NDIM2,NES)
C
C     SUBRUTINA QUE MULTIPLCA POR (G/6)
C
C     ENTRADA:
C     Z2DR...=X2DX,X2DY,Y2DX,Y2DY... MATRIZ QUE SE VA A MULTIPLICAR
C     POR (G/6)
C     SALIDA:
C     Z2DR...MATRIZ RESULTADO DE LA MULTIPLICACION
C
C     NR...NUMERO DE COLUMNAS DE LA MATRIZ Z2DR
C     DECLARACION DE VARIABLES
      REAL Z2DR(NDIM1,NES),DELT(NDIM2)
      DO 10 I=1,NR
        A=Z2DR(1,I)*DELT(1)
        B=Z2DR(N2,I)*2.*DELT(NN)
        C=Z2DR(N2,I)*(-DELT(NN))
        K=N2
          DO 20 J=2,N2
```

```
                    Z2DR(K,I)=Z2DR(K-1,I)*2.*DELT(K)+Z2DR(K,I)*DELT(K)
   20             K=K-1
                Z2DR(1,I)=A
                Z2DR(NN,I)=B
   10       Z2DR(N,I)=C
C
C    DIVISION ENTRE 6.  NEGATIVO
         DO 30 I=1,N
            DO 30 J=1,NR
   30    Z2DR(I,J)=Z2DR(I,J)/(-6.)
         RETURN
         END
C    ***************************************************************
         SUBROUTINE COEF (N,NN,X,Y,Z2D,A,B,C)
C
C    DEFINICION DE VARIABLES
C    SE CALCULAN LOS COEFICIENTES DE LOS POLINOMIOS
C
C    ENTRADA:
C    N...NUMERO DE PUNTOS DE APOYO
C    X...ABSCISAS DE LOS PUNTOS DE APOYO. DIMENSION N
C    Y...ORDENADAS DE LOS PUNTOS DE APOYO. DIMENSION N
C
C    SALIDA:
C    A,B,C...COEFICIENTES DE LOS POLINOMIOS CUBICOS DE LA SPLINE
C    SU DIMENSION ES DE NN=N-1
C
C    DECLARACION DE VARIABLES
C
         REAL X(11),Y(11),Z2D(9),A(10),B(10),C(10)
C
         N2=NN-1
C
C    SOLO PARA SPLINES NATURALES
C
         B(1)=0.
         B(N)=0.
         DO 10 I= 1,N2
   10    B(I+1)=Z2D(I)
         K1= 2
         DO 5 K= 1,NN
            A(K)= B(K1)-B(K)
            A(K)= A(K)/(6.*(X(K1)-X(K)))
            B(K)= B(K)/2.
            C(K)= B(K1)/2.+B(K)+B(K)
            C(K)= (C(K)+C(K))*(X(K1)-X(K))/6.
            C(K)= (Y(K1)-Y(K))/(X(K1)-X(K))-C(K)
    5    K1= K1+1
         RETURN
         END
C    ***************************************************************
         SUBROUTINE ACTU(RAD,P,N,NN,N2,NDIM1,NP)
C
C    SUBRUTINA QUE ACTUALIZA EL VECTOR P
C    ENTRADA:
C    RAD...VECTOR DE RADIOVECTORES
C    P...VER PROGRAMA PRINCIPAL
C
```

```
C    SALIDA
C    P...ACTUALIZADO
C
      REAL RAD(NDIM1),P(NP)
      NN2=NN+NN
      IN =2
      IND=IN+3*NN
      INDI=IND+N+N
      INDA=INDI+N2+N2+N2
      P(1)=RAD(1)*COS(P(INDA+1))
      P(2)=RAD(1)*SIN(P(INDA+1))
      X2=P(1)
      Y2=P(2)
      DO 10 I=1,NN
         J=I+1
         X1=X2
         Y1=Y2
         X2=RAD(J)*COS(P(INDA+J))
         Y2=RAD(J)*SIN(P(INDA+J))
         P(IN+I)=X2-X1
         P(IN+NN+I)=Y2-Y1
10    P(IN+NN2+I)=SQRT((X2-X1)*(X2-X1)+(Y2-Y1)*(Y2-Y1))
      RETURN
      END
```

```
C    ************************************************************
C    *                                                          *
C    *   ESTE PROGRAMA SINTETIZA UNA SPLINE PERIODICA PARAMETRICA *
C    *                                                          *
C    *   PROGRAMO Y REALIZO: MIGUEL ANGEL ALVARADO  RASCON      *
C    *   INSTALACION : PDP 11/40 LAB CAD-DEPFI-UNAM             *
C    *   PROYECTO : PAQUETE DE SINTESIS DE CURVAS               *
C    *   FECHA DE REVISION : 01/ABRIL/1984                      *
C    *                                                          *
C    ************************************************************
C
C    TEXTO:
C    DADO EL VECTOR PRESCRITO DE CURVATURA CU(I),I=1,2,...,M, SE
C    OBTIENE  UNA  FUNCION  SPLINE CUBICA PERIODICA PARAMETRICA,
C    X=X(T), Y=Y(T) PARA LOS VALORES PRESCRITOS DE CU.
C
C    PARAMETROS:
C    M...NUMERO DE PUNTOS DE APOYO  INDEPENDIENTES
C    N...=M*4-3 NUMERO TOTAL DE PUNTOS DE APOYO
C    NN...=N-1
C    N2...=NN*2
C    Z...(1-M) VECTOR DE DIMENSION M QUE CONTIENE LOS RADIOVECTO-
C    RES DE LA FUNCION
C    TE...(1-M) VECTOR DE DIMENSION M QUE CONTIENE LOS ANGULOS DE
C    LOS RADIOVECTORES Z
C    P...VECTOR  QUE  CONTIENE   LOS VECTORES A0,A1,A2,V,W,CU,T Y
C    TE DE LAS SIGUIENTES DIMENSIONES:N2,N2,N2,M,M,M,N Y M N;P ES
C    UNA DIMENSION (6*NN+4*M+N)
C    A0...=X,Y.VECTOR (X,Y) DE LOS PUNTOS DE APOYO
C    A1,A2... VER ZWEITE
C    T...VER CYCLI
C
C    DECLARACION DE VARIABLES
C
          EXTERNAL FUNP ,DFDXP
          REAL Z(5),X(17),P(133),TE(5),Y(17),CU(5),ZW1(5)
          REAL TOLX,TOLF,PIEN4
          INTEGER M,I,N,NN,N2,IND,INDI,IN,MAX,KMAX
          TYPE 12
          ACCEPT 13,M
          PIEN4= ATAN(1.)
          TYPE 41
          ACCEPT 51,RAD,ANG1
          ANG=ANG1*PIEN4*4./180.
          DO 33 I=1,M
C
C    PARA QUE TODOS LOS RADIOVECTORES INICIALES SEAN IGUALES
C
               Z(I)= RAD
C
C    PARA ANGULOS IGUALMENTE ESPACIADOS
C
C              TE(I)= (I-1)*PIEN4/(M-1)
C   33         TE(I)= TE(I)+TE(I)
    33    CONTINUE
C
C    ANGULOS PARA 5 PUNTOS DONDE SE DA EL T(3) Y BISECANDO,
```

```
C    SE DETERMINAN LOS OTROS 4
          TE(3)= ANG
          TE(1)= 0.
          TE(2)= (TE(3)-TE(1))/2.
          TE(5)= 2.*PIEN4
          TE(4)= (TE(5)-TE(3))/2.+TE(3)
          CUMAX=1.
          CUMIN =.2
          DELCU= CUMAX-CUMIN
          DO 52 I=1,M
C
C    PARA UNA DISTRIBUCION COSENOIDAL CON CURVATURA MINIMA DE
C    CUMIN=.2 Y CURVATURA MAXIMA DE CUMAX=1.
C
          CU(I)= CUMAX-COS (PIEN4%4.*TE(I)/TE(3))
          CU(I)=CUMIN+DELCU%CU(I)/2.
   52     IF (TE(I).GT.ANG) CU(I)= CUMAX/2.*(1.+
     $    1.*(COS(PIEN4%4.*(TE(I)-TE(3))/(PIEN4%2.-TE(3)))))
C
C    DEFINICION DE INDICES PARA EL VECTOR P
C
          N=M%4-3
          NN=N-1
          N2=NN+NN
          IND=N2+N2+N2
          INDI=IND+M
          IN=INDI+M
          INTE=IN+N+M
          DO 10 I=1,M
               P(INTE+I)=TE(I)
               P(IND+I)= COS(TE(I))
               P(INDI+I)= SIN (TE(I))
               X(I)= Z(I)*COS(TE(I))
               Y(I)= Z(I)*SIN(TE(I))
   10     P(IN+I)=CU(I)
C
C    OBTENCION DE N PUNTOS X  Y Y A PARTIR DE LOS M PUNTOS
C
          CALL DUPLI (M,N,N,X,Y)
C
C    ALMACENANDO AO EN P
C
          DO 20 I=1,NN
               P(I)=X(I)
   20     P(I+NN)= Y(I)
C
C    DEFINICION DE PARAMETROS PARA UTILIZAR LA SUBRUTINA NRDAMP
C
          NDIM=5
          TOLX= 1.E-4
          TOLF= 1.E-4
          DAMP= .5
          MAX= 20
          KMAX= 20
C
C    LLAMANDO A LA SUBRUTINA QUE APLICA NEWTON-RAPHSON
C
```

```
      CALL NRDAMP (Z,FUNP,DFDXP,P,TOLX,TOLF,DAMP,M,NDIM,ITER
     $  ,MAX,KMAX)
      BAND= 1.
C
C   EL LLAMADO DE FUNP AQUI ES PARA PODER INTERPOLAR Y GRAFICAR
C
      CALL FUNP (Z,F,P,M,BAND)
C
C   ZONA DE FORMATOS
C
   12 FORMAT (3X,'NUMERO DE PUNTOS M= ',$,I4)
   13 FORMAT (I4)
   14 FORMAT (3X,'Z,TE,I',3X,2F8.4,I4)
   41 FORMAT(3X,'RADIO,ANGULO  ',$)
   51 FORMAT(2F8.4)
      CALL EXIT
      END
C  ***********************************************************************
      SUBROUTINE ERSTE (NN,N2,A2,CUS,DELT,ZWE,ERS)
C
C   TEXTO:
C   ESTA  SUBROUTINA CALCULA LAS DERIVADAS PARCIALES  DEL CAMPO
C   A1 CON RESPECTO A LAS COORDENADAS CARTESIANAS DE LOS PUNTOS
C   DE APOYO DE UNA SPLINE PERIODICA PARAMETRICA QUE APROXIMA
C   UNA CURVA PLANA CERRADA .
C
C   ENTRADA :
C   N1...N-1,N=NUMERO TOTAL DE PUNTOS DE APOYO
C   N2...=2*(N-1)
C   A2,CUS,DELT...VER ZWEITE
C   ZWE.. SALIDA DE ZWEITE
C
C   SALIDA:
C   ERS(I,J)...DERIVADA DE A1(I) CON RESPECTO A AO(J), DE DIMEN-
C   SION N2XN2
C
C   DECLARACION DE VARIABLES
C      REAL A2(N2),ZWE(N2,N2),DELT(NN),ERS(N2,N2),CUS(N2)
C
      REAL A2(32),ZWE(32,32),DELT(16),ERS(32,32),CUS(32)
C
C   CALCULO DE ERS
C
      SG=-1.
      DO 50 J=1,2
         SG= -SG
         DO 50 I= 1,2
            SG=-SG
            DO 20 K= 1,NN
               KK= (I-1)*NN+K
               DO 20 L=1,NN
                  LL= L+(J-1)*NN
```

```
                    LLP1= LL+1
                    IF (L.EQ.NN) LLP1= LLP1-NN
                    ERS(LL,KK)= ZWE(LL,KK)+ZWE(LL,KK)
                    ERS(LL,KK)= ERS(LL,KK)+ZWE(LLP1,KK)
20              ERS(LL,KK)= -ERS(LL,KK)*DELT(L)/6.
                DO 50 K= 1,NN
                    KI1= K+(I-1)*NN
                    KI1P1= KI1+1
                    IF (K.EQ.NN) KI1P1=KI1P1-NN
                    KI2= K+(2-I)*NN
                    KJ1= K+(J-1)*NN
                    KJ2= K+(2-J)*NN
                    KJ1P1= KJ1+1
                    IF (K.EQ.NN) KJ1P1= KJ1P1-NN
                    ADD= CUS(KJ2)*CUS(KI2)*SG/DELT(K)
                    ADD= ADD+(A2(KJ1)+A2(KJ1)+A2(KJ1P1))*CUS(KI1)/6.
                    ERS(KJ1,KI1)= ERS(KJ1,KI1)+ADD
50      ERS(KJ1,KI1P1)= ERS(KJ1,KI1P1)-ADD
        END
C   ***********************************************************************
        SUBROUTINE DERKRU (NN,N2,A1,A2,DKDA1,DKDA2)

C
C   TEXTO:
C   ESTA  SUBRUTINA  CALCULA  LAS DERIVADAS PARCIALES DE LA CURVA-
C   TURA DE LOS PUNTOS DE APOYO DE LA SPLINE PARAMATRICA PERIODICA
C   QUE APROXIMA UNA CURVA PLANA CERRADA CON RESPECTO A LAS VARIA-
C   BLES  A1 Y A2 DEFINIDAS EN ZWEITE. ESTA DERIVADAS SE ALMACENAN
C   EN DKDA1 Y DKDA2
C
C   ENTRADA:
C   NN,N2,A1,A2...DEFINIDAS EN ZWEITE
C
C   SALIDA:
C   DKDA1...DERIVADA  DE LA CURVATURA EN EL I-ESIMO. PUNTO DE APOYO
C   CON RESPECTO A A1(I),DE  DIMENSION (2*(N-1))
C   DKDA2...DERIVADA  DE LA CURVATURA EN EL I-ESIMO. PUNTO DE APOYO
C   CON RESPECTO A A2(I), DE  DIMENSION (2*(N-1))
C
C
C   DECLARACION DE VARIABLES
C        REAL A1(N2),A2(N2),DKDA1(N2),DKDA2(N2)
C
        REAL A1(32),A2(32),DKDA1(32),DKDA2(32)
        DO 10 I= 1,2
            DO 10 K= 1,NN
                K1= K+(I-1)*NN
                K2= K+(2-I)*NN
                G= A1(K1)*A1(K1)+A1(K2)*A1(K2)
                EINS= -A1(K1)*A1(K1)
                EINS= EINS+EINS+A1(K2)*A1(K2)
                EINS= EINS*A2(K2)
                ZWEI= 3.*A1(K1)*A1(K2)*A2(K1)
                DKDA1(K1)= (EINS+ZWEI)/(G**2.5)
```

```
      $
                  IF (I.EQ.2) DKDA1(K1)= -DKDA1(K1)
                  DKDA2(K1)= A1(K2)/(G**1.5)
   10     IF (I.EQ.1) DKDA2(K1)= -DKDA2(K1)
          RETURN
          END
C     ***************************************************************
          SUBROUTINE DUPLI(M,NDIM1,NDIM2,X,Z)
C
C     TEXTO:
C     DADOS LOS M PUNTOS INDEPENDIENTES DE APOYO (X(I),Y(I),) I=
C     1,2,..,M, DE UNA SPLINE PARAMETRICA PERIODICA QUE APROXIME
C     UNA CURVA CERRADA, SE OBTIENEN LOS N PUNTOS DE APOYO.
C
C     X...ENTRADA Y SALIDA DE LA ABSCISA
C     Z...ENTRADA Y SALIDA DE LA ORDENADA
C
C     DECLARACION DE VARIABLES
C          REAL X(NDIM1),Z(NDIM2)
          REAL X(17),Z(17)
          L5= M*4-3
          Z(L5)= -Z(1)
          X(L5)= X(1)
          L0= M+M-1
          Z(L0)= Z(1)
          X(L0)= -X(1)
          L1= L0+M-1
          Z(L1)= -Z(M)
          X(L1)= X(M)
          N1= M-1
          DO 21 I= 2,N1
               L0= M+M-I
               L1= M+M-2+I
               L2= L0+M+M-2
               X(L0)= -X(I)
               X(L1)= -X(I)
               X(L2)= X(I)
               Z(L0)= Z(I)
               Z(L1)= -Z(I)
   21     Z(L2)= -Z(I)
          RETURN
          END
C     ***************************************************************
          SUBROUTINE DFDXP (Z,DF,P,M)
C
C     TEXTO:;
C     ESTA SUBRUTINA CALCULA LA MATRIZ JACOBIANA DEL VECTOR F CON
C     RESPECTO A Z
C
C          ENTRADA:
C     P...VECTOR QUE DEBE TENER LA SIGUIENTE INFORMACION
C     ZWE,T,A0,A1,A,BB,DELT,CUS...DEFINIDAS EN ZWEITE
C     ERS...DEFINIDA EN ERSTE
C     F...DEFINIDA EN DFDX
C     DKDA0...DERIVADA DE LA CURVATURA EN EL PUNTO I-ESIMO, DE APO-
C     YO     CON RESPECTO A A0(I), DIMENSION (MXN2)
C
```

```
C    SALIDA:
C    DF...MATRIZ JACOBIANA DEL VECTOR F CON RESPECTO A Z
C
C    DECLARACION DE VARIABLES
C
      REAL ZWE(32,32),ERS(32,32),DKDA0(5,32),P(133),A0(32)
      REAL A1(32),A2(32),DKDA1(32),DKDA2(32),A(16,16)
      REAL BB(16),DF(5,5),Z(5),T(17),DELT(16),CUS(32)
      INTEGER IP(16)
      N= M*4-3
      NN= N-1
      N2= NN+NN
      INDI= N2+N2+N2+M+M+M
      DO 5 I= 1,N
    5 T(I)= P(INDI+I)
      DO 10 I= 1,N2
         A0(I)= P(I)
         A1(I)= P(N2+I)
   10 A2(I)= P(N2+N2+I)
C
C    OBTENIENDO LOS VECTORES DKDA1Y DKDA2, DEFINIDOS EN DERKRU
C
      CALL DERKRU (NN,N2,A1,A2,DKDA1,DKDA2)
C
C    OBTENCION  DE  LAS  DERIVADAS DE A2 CON RESPECTO A A0
C
      CALL ZWEITE (N,NN,N2,T,A0,A1,A2,ZWE,DELT,CUS,A,BB,IP)
C
C    OBTENCION DE LAS DERIVADA DE A1 CON RESPECTO A A0
C
      CALL ERSTE (NN,N2,A2,CUS,DELT,ZWE,ERS)
C
C    OBTENCION DE DKDA0
C
      DO 20 I= 1,M
      DO 20 J= 1,NN
         DKDA0(I,J)=DKDA1(I)*ERS(I,J)+DKDA1(I+NN)*ERS(I+NN,J)+
     $   DKDA2(I)*ZWE(I,J)+DKDA2(I+NN)*ZWE(I+NN,J)
         DKDA0(I,J+NN)= DKDA1(I)*ERS(I,J+NN)+DKDA1(I+NN)*ERS(I+NN,J+
     $ NN)+ DKDA2(I)*ZWE(I,J+NN)+DKDA2(I+NN)*ZWE(I+NN,J+NN)
   20 CONTINUE
C
C    OBTENCION DE DF
C
      DO 30 I=1,M
         IND = N2+N2+N2+1
         INDI= IND+M
         L0= M+M-1
         L1=NN+1
         L2= L1+L0-1
         DF(I,1)= (DKDA0(I,1)-DKDA0(I,L0))*P(IND)+
     $   (DKDA0(I,L1)+DKDA0(I,L2))*P(INDI)
         IND= IND+M-1
         INDI= IND+M
         L0= L0-1+M
```

```
            L1= L1-1+M
            L2= L2-1+M
   30     DF(I,M)= (DKDAO(I,M)-DKDAO(I,L0))*P(IND)+
     $    (DKDAO(I,L1)-DKDAO(I,L2))*P(INDI)
          IF(M.EQ.2)RETURN
C
C    PARA LAS COLUMNAS INTERMEDIOS
C
          MM1= M-1
          DO 40 I= 2,MM1
               DO 40 J= 1,M
                    IND =N2+N2+N2+I
                    L0= M+M-I
                    L1= M+M-2+I
                    L2= L0+M+M-2
                    DF(J,I)= (DKDAO(J,I)-DKDAO(J,L0)-DKDAO(J,L1)+
     $              DKDAO(J,L2))*P(IND)
                    INDI= IND+M
                    L3= I+NN
                    L0= L0+NN
                    L1= L1+NN
                    L2= L2+NN
   40     DF(J,I)= DF(J,I)+(DKDAO(J,L3)+DKDAO(J,L0)-DKDAO(J,L1)-
     $    DKDAO(J,L2))*P(INDI)
          RETURN
          END
C    ***********************************************************************
          SUBROUTINE FUNP (Z,FF,P,M,BAND)
C
C    TEXTO:
C    ESTA SUBRUTINA EVALUA LA FUNCION F(Z) EN EL VALOR ACTUAL DE
C    Z   DONDE F(Z)  REPRESENTA LA DIFERENCIA ENTRE LA CURVATURA
C    GENERADA EN LOS PUNTOS DE APOYO Y LA PRESCRITA.
C
C    ENTRADA:
C    P...VER PROGRAMA PRINCIPAL
C
C    SALIDA:
C    FF...VECTOR DE DIMENSION M CUYOS ELEMENTOS CONTIENEN LAS DI-
C    FERENCIAS BUSCADAS
C
C   DECLARACION DE VARIABLES
C
          REAL Z(5),FF(5),P(133),F(17),G(17),H(17),T(17),X(17)
          REAL Y(17),X2(17),Y2(17),AX(16),BX(16),CX(16),DX(16)
          REAL AY(16),BY(16),CY(16),DY(16)
          REAL F1(220),F1S(220),TAU(220)
          N= M*4-3
          NN= N-1
C
C    ACTUALIZANDO EL VECTOR P
C
          CALL ACTUF (Z,X,Y,P,M,NN,N)
C
```

```
C    CALCULANDO LOS COEFICIENTES DE LA SPLINE
C
          CALL CYCLIC (N,NN,F,G,H,T,X,Y,X2,Y2,AX,BX,CX,DX,AY,BY,CY,DY)
C
C    BAND=1. PARA INTERPOLAR Y GRAFICAR LOS RESULTADOS
C
          IF (BAND.EQ.0.) GOTO 11
          CALL COSPLI (N,NN,T,X,Y,AX,BX,CX,DX,AY,BY,CY,DY,TAU,F1,F1S)
C
          GOTO  12
   11     IN= NN+NN
          IND= IN+IN
          INDI= IND+IN+3*M
C
C     ALMACENANDO A1 Y A2 EN P. VER ZWEITE
C
          DO 20 I= 1,NN
               P(IN+I)= CX(I)
               P(IN+I+NN)= CY(I)
               P(INDI+I)= T(I)
               P(IND+I)= BX(I)+BX(I)
   20     P(IND+NN+I)= BY(I)+BY(I)
          P(INDI+N)= T(N)
          INDI= INDI-M
          DO 30 I= 1,M
C
C    CALCULO DE FF
C
               FF(I)= P(IN+I)*P(IND+NN+I)-P(IND+I)*P(IN+NN+I)
               DEN= P(IN+I)*P(IN+I)+P(IN+NN+I)*P(IN+NN+I)
               DEN= DEN **1.5
   30     FF(I)= FF(I)/DEN-P(INDI+I)
   12     RETURN
          END

C    ***********************************************************************
       SUBROUTINE CUBICF (N,NN,F,G,H,X,Y,Y2,A,B,C,D)
C
C    TEXTO:
C    ESTA  SUBRUTINA  CALCULA  LOS  COEFICIENTES  A,B,C,D  DE LA
C    SPLINE
C
C      Y=A(K)*(X-X(C))**3+B(K)*(X-X(K))**2+
C        +C(K)*(X-X(K))+D(K),  X(K)<X<X(K+1)
C
C    DATOS:
C    N,NN(=N-1),X(K),Y(K),K=1,...,N
C                WOBET Y(N)=Y(1)
C
C      RESULTADOS:A(K),B(K),C(K),D(K),K=1,...,N-1
C
C    VERSION  MODIFICADA  DE  LA SUBRUTINA DEL MISMO NOMBRE, DE
C    SPAETH H., SPLINE ALGORITMEN ZUR KONSTRUKTION, GLATTER KUR-
C    VEN UND FLAECHEN,R. OLDENBURG VERLAG,MUNICH-VIENA,1973,P.46
C
C
C       REAL X(N),Y(N),Y2(N),F(N),G(N),H(N)
C       REAL A(NN),B(NN),C(NN),D(NN)
```

```
      REAL X(17),Y(17),Y2(17),F(17),G(17),H(17)
      REAL A(16),B(16),C(16),D(16)
    N2=NN-1
    J1=1
    G(1)=0.
      F(1)=0.
      H(1)=-1.
    H1=X(N)-X(NN)
    W=H1
    H2=X(NN)-X(N2)
    U=2.*(H1+H2)
    R1=(Y(N)-Y(NN))/H1
    R2=(Y(NN)-Y(N2))/H2
    V=6.*(R1-R2)
    DO 2 K=1,N2
         J2=K+1
         H2=X(J2)-X(K)
         R2=(Y(J2)-Y(K))/H2
         IF (K.EQ.1) GO TO 1
         U=U-W*H(J1)
         V=V-W*F(J1)
         W=-G(J1)*W
1        Z=1./(2.*(H1+H2)-H1*G(J1))
         G(K)=Z*H2
         H(K)=-Z*H(J1)*H1
         F(K)=Z*(6.*(R2-R1)-H1*F(J1))
         J1=K
         H1=H2
         R1=R2
2   CONTINUE
    H2=W+H1
    H1=(V-H2*F(N2))/(U-H2*(G(N2)+H(N2)))
    Y2(NN)=H1
    DO 3 J1=2,NN
         K=N-J1
         Y2(K)=F(K)-G(K)*Y2(K+1)-H(K)*H1
3   CONTINUE
     Y2(N)=Y2(1)
    K1=2
    DO 5 K=1,NN
         IF(K.NE.NN) GO TO 4
         Y(K1)=Y(1)
4        A(K)=Y2(K1)-Y2(K)
         A(K)=A(K)/(6.*(X(K1)-X(K)))
         B(K)=Y2(K)/2.
         C(K)=(Y(K1)-Y(K))/(X(K1)-X(K))
         C(K)=C(K)-(X(K1)-X(K))*(Y2(K1)+Y2(K)+Y2(K))/6.
         D(K)=Y(K)
5   K1=K1+1
      RETURN
      END
*************************************************************
      SUBROUTINE CYCLIC(N,NN,F,G,H,T,X,Y,X2,Y2,AX,BX,CX,DX,AY,BY,
     $  CY,DY)
```

```
C
C     N...NUMERO DE PUNTOS DE APOYO
C     F,G,H...VECTORES DE DIMENSION N CALCULADOS
C     EN LA SUBRUTINA CUBICP
C     T...VECTOR DE DIMENSION N CUYOS COMPONENTES SE DEFINEN COMO
C     SIGUE:
C
C       T(1)=0.,T(K+1)=T(K)+DELTA(K),K=1,....,N-1
C     DONDE DELTA(K)=SQRT((X(K+1)-X(K))**2+((Y(K+1)-Y(K))**2), SE
C     CALCULA EN CYCLIC
C
C     X,Y...VALORES CONTENIDOS EN A0. VER ZWEITE
C     X2,Y2...VALORES CONTENIDOS EN A2. VER ZWEITE
C     AX,BX,CX,DX...COEFICIENTES DE LA SPLINE
C         AX(K)*(X-(X(K))**3+BX(K)*(X-X(K)**2+CX(K)*(X-X(K))+DX(K)
C     AY,BY,CY,DY:COEFICIENTES DE LA SPLINE
C         AY(K)*(Y-(Y(K))**3+BY(K)*(Y-Y(K)**2+CY(K)*(Y-Y(K))+DY(K)
C        REAL T(N),X(N),Y(N),X2(N),Y2(N),F(N),G(N),H(N),AX(NN),
C       1   BX(NN),CX(NN),DX(NN),AY(NN),BY(NN),CY(NN),DY(NN)
C
          REALT(17),X(17),Y(17),X2(17),Y2(17),F(17),G(17),H(17),AX(16),
       $   BX(16),CX(16),DX(16),AY(16),BY(16),CY(16),DY(16)
          T(1)=0.
          J1=1
          DO 1 K=2,N
               U=X(K)-X(J1)
               V=Y(K)-Y(J1)
              D=SQRT(U*U+V*V)
               T(K)=T(J1)+D
     1    J1=K
C
C     COEFICIENTES DE LA SPLINE
C
          CALL CUBICP (N,NN,F,G,H,T,X,X2,AX,BX,CX,DX)
          CALL CUBICP (N,NN,F,G,H,T,Y,Y2,AY,BY,CY,DY)
          RETURN
          END
C     ************************************************************
          SUBROUTINE ZWEITE(N,NN,N2,T,A0,A1,A2,ZWE,DELT,CUS,
         1A,BB,IP)
C     TEXTO:
C     ESTA SUBRUTINA CALCULA LAS DERIVADAS PARCIALES DEL ARREGLO
C     A2 CON RESPECTO A LAS COORDENADAS CARTESIANAS DE LOS PUN-
C     TOS DE APOYO DE UNA SPLINE PARAMETRICA,  QUE APROXIMA UNA
C     CURVA CERRADA
C
C     PARAMETROS:
C     NN...=N-1 SIENDO N EL NUMERO TOTAL DE PUNTOS DE APOYO,
C     INCLUYENDO EL ULTIMO
C     N2...=2*(N-1)
C     T...PARAMETRO DE LAS FUNCIONES SPLINE (VER SUB. CYCLIC)
C     A0(I),A0(I+NN)...COORDENADAS X,Y, RESPECTIVAMAENTE DEL .
C     A1(I),A1(N1+I)... PRIMERAS  DERIVADAS  DE LAS COORDENADAS
C     ANTERIORES CON RESPECTO A T, QUE SE CALCULA EN CYCLIC. ES
```

```
C    UN CAMPO DE DIMENSION 2*(N-1)
C    A2(I),A2(N1,I),.. SEGUNDAS  DERIVADAS DE  LAS COORDENADAS
C    CON RESPECTO A T , SE CALCULA EN CYCLIC.
C    CAMPO DE DIMENSION 2*(N-1)
C    ZWE(I,J),.. DERIVADA PARCIAL DE A2(I) CON RESPECTO A A0(J),
C    DELT(K),...SQRT((X(K+1)-X(K))**2+(Y(K+1)-Y(K))**2)
C    CUS(K),CUS(K+1),...X(K)/DELT(K) ,Y(K)/DELT(K),RESPECTIVA-
C    MENTE .
C
C    OTRAS VARIABLES DE LA SUBRUTINA:
C    A.., MATRIZ SIMETRICA  DE (N-2)X(N-2) DEFINE LA RELACION
C    ENTRE A2 Y A0
C    BB...CAMPO AUXILIAR DE DIMENSION N-1
C
C
C       REAL A1(N2),T(N),A0(N2),A2(N2),BB(NN),A(NN,NN),
C      1 ZWE(N2,N2),DELT(NN),CUS(N2)
C       INTEGER IP(NN)
        REAL A1(32),T(17),A0(32),A2(32),BB(16),A(16,16),ZWE(32,32)
        REAL DELT(16),CUS(32)
        INTEGER IP(16)
        DO 10 K= 1,NN
            KP1= K+1
            IF (K.EQ.NN) KP1=1
            CUS(K)= A0(KP1)-A0(K)
            KK= K+NN
            KPP= KP1+NN
            DELT(K)= T(K+1)-T(K)
            CUS(KK)= A0(KPP)-A0(KK)
            CUS(K)= CUS(K)/DELT(K)
10      CUS(KK)= CUS(KK)/DELT(K)
        DO 20 I=1,NN
            DO 20 J= 1,NN
20      A(I,J)=0.
        A(1,1)= DELT(NN)+DELT(1)
        A(1,1)= A(1,1)+A(1,1)
        A(1,NN)= DELT(NN)
        A(NN,1)= A(1,NN)
        DO 30 I= 2,NN
            A(I,I)= DELT(I-1)+DELT(I)
            A(I,I)= A(I,I)+A(I,I)
            A(I,I-1)=DELT(I-1)
30      A(I-1,I)= A(I,I-1)
        NDIM=16
        CALL DECOMP (NN,NDIM,A,IP)
        IF (IP(NN).EQ.0) STOP 99
        SG=1.
        DO 80 J= 1,2
            SG= -SG
            DO 80 I = 1,2
                SG= -SG
                    DO 80 K= 1,NN
                DO 40 L=1,NN
40              BB(L)= 0.
                KI1= K+(I-1)*NN
                KI2= K+(2-I)*NN
```

```
                KJ1=K+(J-1)*NN
                KJ2=K+(2-J)*NN
                KI1M1= KI1-1
                KI2M1= KI2-1
                KJ1M1= KJ1-1
                KJ2M1= KJ2-1
                KM1= K-1
                KJ1P1= KJ1+1
                KP1= K+1
                IN=3
                IF (K.EQ.1) IN=1
                IF (K.EQ.NN) IN=2
                GOTO (50,60,70) ,IN
 50             KI1M1= KI1M1 +NN
                KI2M1= KI2M1+NN
                KJ1M1= KJ1M1+NN
                KJ2M1= KJ2M1+NN
                KM1= KM1+NN
                GOTO 70
 60             KJ1P1= KJ1P1-NN
                KP1= KP1-NN
 70             BB(KM1)= 6.*SG*CUS(KJ2M1)*CUS(KI2M1)/DELT(KM1)
                BB(KP1)=6.*SG*CUS(KJ2)*CUS(KI2)/DELT(K)
                BB(K)= -(BB(KM1)+BB(KP1))
                EINS= (A2(KJ1M1)+A2(KJ1M1)+A2(KJ1))*CUS(KI1M1)
                ZWEI= (A2(KJ1M1)+A2(KJ1)+A2(KJ1))*CUS(KI1M1)
                ZWEI= ZWEI-(A2(KJ1)+A2(KJ1)+A2(KJ1P1))*CUS(KI1)
                DREI= -(A2(KJ1)+A2(KJ1P1)+A2(KJ1P1))*CUS(KI1)
                BB(KM1)= BB(KM1)-EINS
                BB(K)= BB(K)-ZWEI
                BB(KP1)= BB(KP1)-DREI
                CALL SOLVE (NN,NDIM,A,BB,IP)
                DO 80 L= 1,NN
                    LL= L+(J-1)*NN
 80     ZWE(LL,KI1)= BB(L)
        RETURN
        END
C   ***********************************************************************
        SUBROUTINE ACTUP (Z,X,Y,P,M,NN,N)
C
C   ESTA SUBRUTINA ACTUALIZA EL VECTOR P
C   DECLARACION  DE VARIABLES
C
        REAL Z(5),TE(5),X(17),Y(17),P(133)
C
        INTE=6.*NN+N+3.*M
        DO 10 I=1,M
            TE(I)=P(INTE+I)
            X(I)=Z(I)*COS(TE(I))
 10     Y(I)=Z(I)*SIN(TE(I))
        CALL DUPLI (M,N,N,X,Y)
        DO 20 I=1,NN
        P(I)=X(I)
 20     P(I+NN)=Y(I)
        RETURN
        END
```