

ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

TESIS DE GRADO

"PROGRAMAS DIGITALES PARA LA INTERPOLACION
EN TRES DIMENSIONES"

Por: GUILLERMO NIETO RIOS

TESIS PREVIA A LA OBTENCION DEL TITULO DE
INGENIERO ELECTRICO EN LA ESPECIALIZACION
DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES.

Quito, Agosto de 1978

Certifico que el presente
trabajo ha sido elaborado
en su totalidad por el se
ñor Guillermo Nieto Ríos.

A handwritten signature in blue ink, reading "Efraín Del Pino V.", is written over a light blue rectangular background.

ING. EFRAIN DEL PINO V.
Director de Tesis

DEDICATORIA:

Al amor de una Madre
y al cariño de un Hermano

AGRADECIMIENTO:

A todos los profesores y compañeros de la Escuela Politécnica Nacional, que de una u otra manera influyeron en la culminación de mi carrera. En especial al Ing. EFRAIN DEL PINO que tanto en la vida estudiantil como durante la elaboración de este trabajo demostró ser un buen maestro.

C O N T E N I D O

CAPITULO I.

1.1	Generalidades	1
1.2	Aproximación	3
1.3	Errores Computacionales	4

CAPITULO II.

2.1	Introducción	6
2.2	Funciones Aproximantes	9
2.3	Aproximación Polinomial	11
2.3.1	Polinomios de Colocación	13
2.4	Polinomio de Colocación por Diferencias Finitas Divididas de Newton	17
2.5	Polinomio de Colocación de Lagrange	27
	Interpolación Tridimensional Lineal	30
	Programa para Interpolación Tridimensional Directa	33
2.6	Polinomios de Colocación Spline Cúbicos	50
	Programa para Interpolación Tridimensional Directa e Inversa	56

CAPITULO III.

	Discusión de Resultados y Conclusiones	105
--	--	-----

APENDICE 1	(Programa para Interpolación Bidimensional)	108
APENDICE 2	(Características de las Diferencias Finitas Divididas)	121
APENDICE 3	(Subprograma para evaluar el Polinomio de Colocación de Lagrange)	125
APENDICE 4	(Desarrollo Matemático de los Polinomios de Colocación Spline Cúbicos)	130
APENDICE 5	(Detalle de los Argumentos de los Subprogramas)	138
BIBLIOGRAFIA		154

C A P I T U L O I

INTRODUCCION

1.1.- GENERALIDADES:

Gracias al advenimiento de las computadoras electrónicas, hoy día se puede realizar cálculos más rápidos; esto ha involucrado a su vez un desarrollo de métodos numéricos más perfeccionados, que estén de acuerdo con los avances y con las necesidades de la época.. La intención de este trabajo es afianzar el vínculo que existe entre el análisis numérico y la ingeniería; nexos, que por imposibilidades pretéritas estaba un tanto descoyuntado. De ahí que el enfoque que se da al presente trabajo está orientado en este sentido, por lo que se obviará el análisis matemático exhaustivo, tratando de ser lo más conciso que sea posible, sin desligarse del tema propuesto que consiste en brindar a la ingeniería y en particular a la ingeniería electrónica, una herramienta útil en el campo de la interpolación -capítulo del análisis numérico muy usado por todas las ramas de la ingeniería sin excepción alguna-.

La ayuda que se ofrece consiste en la implementación de un programa en lenguaje FORTRAN IV para computadora digital y que, a más de lograr resultados más exactos

evita los cálculos tediosos que se realizan manualmente en ingeniería, cuando es necesario obtener valores de tablas ó de gráficos, mediante interpolación. El programa de interpolación es en realidad una subrutina que trabaja con tres variables y que puede encontrar cualquiera de ellas, dando como dato las otras dos variables, suponiendo además que también son datos las muestras tomadas de las tres variables, de las cuales dos son variables independientes y de cuyos valores es función la otra variable, llamada dependiente.

El análisis teórico empleado en el presente trabajo se refiere específicamente a tres técnicas de aproximación polinomial de funciones bidimensionales, $y = f(x)$; siendo "x" la variable independiente y "y" la variable dependiente o función de "x", que son conocidas de una manera cuantitativa o discreta. La técnica de aproximación en dos dimensiones es proyectada luego a la aproximación tridimensional, mediante una forma peculiar de muestreo. (Ver tabla 2.4); por este motivo se da mayor énfasis a la aproximación con dos variables.

En lo referente a programación se sobreentiende un conocimiento del lenguaje FORTRAN IV; mencionando únicamente que se utiliza doble precisión con el objeto de reducir el error de aproximación y que todos los pro-

gramas efectuados fueron corridos en la computadora IBM 370/125 de la Escuela Politécnica Nacional.

1.2.- APROXIMACION.

La necesidad de aproximar es imperiosa ya que todo modelo matemático de procesos físicos contiene algunos errores inherentes, causados por falta de una completa concepción de los fenómenos naturales; también son debidos a la naturaleza probabilística de muchos procesos y a la incertidumbre de las medidas experimentales. Con frecuencia un modelo matemático sólo toma en cuenta los efectos más sobresalientes de un proceso físico relegando los efectos secundarios; es decir todo modelo matemático es una abstracción de un fenómeno natural y como tal posee errores.

Aún más, suponiendo que es posible desarrollar un modelo matemático libre de errores, este no puede ser resuelto exactamente en una computadora digital. Una computadora digital es capaz de ejecutar un número limitado de operaciones aritméticas sencillas - suma, resta, multiplicación, división - y finitas en el campo de los números racionales. Toda computadora tiene una capacidad de memoria limitada; únicamente un subconjunto discreto de los números reales y racionales puede ser

generado, manipulado y almacenado. De ahí que números infinitesimalmente pequeños o infinitamente grandes no pueden ser representados y peor aún un continuo de los números reales en un intervalo finito.

Refiriéndose específicamente al tema que atañe al presente trabajo, los valores tabulados de la función como de la variable independiente pueden ser por sí mismos aproximaciones de valores reales, especialmente, cuando las tablas de las cuales se parte se las conforma con resultados experimentales.

1.3.- ERRORES COMPUTACIONALES.

Antes de referirse al error, es conveniente definir lo que es un algoritmo y se conoce como tal al método de cálculo empleado para obtener resultados a partir de ciertos datos. Cuando los algoritmos emplean operaciones aritméticas y ciertas comparaciones lógicas tales como comparaciones algébricas, son llamados METODOS NUMERICOS. Con frecuencia se dispondrá de varios algoritmos para obtener el resultado requerido, debiendo escoger aquel que emplee menor tiempo de cálculo y proporcione mayor exactitud. La obtención de exactitud demandará mayor esfuerzo por parte del analista y pondrá de manifiesto un aspecto muy importante: LA PRESEN

CIA DE ERROR. A más de los errores propios de los datos de entrada, que rara vez no los hay, existen dos tipos de errores:

ERROR POR TRUNCAMIENTO.-

Este error es introducido por el algoritmo utilizado en la solución aproximante de un problema matemático.

ERROR POR REDONDIAMIENTO.-

Se produce por el redondiamiento de los resultados de una operación aritmética individual, ya que sólo un número finito de dígitos puede ser retenido después de cada operación. Este tipo de error difiere de una computadora a otra aún cuando se utilice el mismo algoritmo.

El procedimiento que se usa en el presente trabajo para evaluar los algoritmos, consiste en aplicarlos a problemas de los cuales se conoce una solución más exacta; permitiendo esto, obtener los elementos de juicios necesarios para decidirse por los algoritmos. De esta manera se evita en cierto modo hacer análisis del error, del cual sólo se puede obtener estimaciones.

C A P I T U L O I I

TECNICAS DE INTERPOLACION Y OBTENCION DE FUNCIONES APROXIMANTES PARA UN CONJUNTO DE DATOS DISCRETOS.

2.1.- INTRODUCCION.

Existen dos razones principales por las que es necesario reemplazar una función $f(x)$ por otra función $g(x)$ más conveniente.

La primera es substituir una función $f(x)$ que es difícil de manipular o evaluar por otra función $g(x)$, la cual nos permite un análisis más simple. Así por ejemplo, la operación de integración suele presentar serias dificultades y su solución solamente está restringida a casos particulares que son estudiados en los cursos respectivos de matemáticas; algo similar se presenta en la operación de diferenciación. "Las funciones trascendentales ($\ln x$, $\text{sen } x$, $\text{cos } x$, $\text{erf } (x)$) son ejemplos de funciones que no pueden ser evaluadas por operaciones estrictamente aritméticas, sin primero encontrar una función aproximante, tal como una serie finita de potencias". (1). La segunda razón, cuando $f(x)$ es conocida cuantitativamente por medio de tablas o de gráficos que es lo que sucede con frecuencia en ingeniería, se presenta la necesidad de interpolar.

En general $f(x)$ es conocida dentro de un intervalo (normalmente pequeño) para valores discretos de la variable independiente x , los mismos que se los va a tabular como sigue:

x	$f(x)$
x_0	$f(x_0)$
x_1	$f(x_1)$
x_2	$f(x_2)$
.	
.	
.	
x_{n-1}	$f(x_{n-1})$
x_n	$f(x_n)$

Los valores de la variable independiente x_0, x_1, \dots, x_n , para los cuales es conocida la variable dependiente $f(x)$, son llamados PUNTOS BASES.

El propósito, es pues, generar una función aproximante que permita una estimación de la función $f(x)$, para un valor de x distinto de x_i ; $i = 0, 1, \dots, n$.

"La síntesis de una nueva función analítica $g(x)$ que se aproxima a la función original $f(x)$ depende de muchos factores, tales como el conocimiento de la función, la exactitud y la fuente de donde provienen los

valores tabulados de la función, el tipo de función $g(x)$ usada como aproximante, y la precisión requerida para la aproximación" (1). Mientras más información haya acerca de $f(x)$, se estará en la posibilidad de encontrar la función $g(x)$ más adecuada. Por ejemplo, si un modelo teórico sugiere que $f(x)$ tenga una dependencia cúbica con respecto a la variable independiente x , entonces probablemente se comenzaría ensayando la información tabulada con un polinomio de tercer grado. Tampoco se puede esperar, que las estimaciones que proporciona $g(x)$ de $f(x)$ sean exactas en cuatro cifras significativas, si los valores discretos usados de $f(x)$ para la determinación de $g(x)$, tan sólo son exactos en dos cifras significativas. Supóngase que la integral de una cierta función $f(x)$ es difícil de evaluar y asumiendo que sean posibles las formulaciones de alternativas para $g(x)$; lo natural sería escoger una función que sea fácil de integrar. Matemáticamente

$$\int_a^b f(x) dx \approx \int_a^b g(x) dx$$

Cuando $f(x)$ no es conocida, no existe manera de evaluar el error que se comete al reemplazar $f(x)$ por $g(x)$ pero se puede hacer una estimación del orden de magnitud, haciendo algunas asunciones sobre $f(x)$ (por ejem

plo que $f(x)$ es suave, monotónica, que sus derivadas de orden superior son pequeñas, etc). En cambio cuando $f(x)$ es conocida analíticamente es posible establecer un límite superior para el error.

2.2.- FUNCIONES APROXIMANTES

"Las funciones aproximantes más comunes $g(x)$ son aquellas que involucran combinaciones lineales de funciones simples, extraídas de una clase de funciones $\{g_i(x)\}$ de la forma" (1)

$$g(x) = a_0 g_0(x) + a_1 g_1(x) + \dots + a_n g_n(x)$$

De este tipo de funciones las más frecuentes son: Los monomios $\{x^i\}$; $i = 0, 1, 2, \dots, n$. Las funciones de Fourier $\{\text{Sen } kx, \text{Cos } kx\}$; $k = 0, 1, 2, \dots, n$. Y las exponenciales $\{e^{b_i x}\}$; $i = 0, 1, 2, \dots, n$.

Los polinómicos de grado n , $P_n(x)$, son el resultado de la combinación lineal de monomios.

$$f(x) = a_0 + a_1 x_1 + a_2 x_2 + \dots + a_n x^n$$

$$6 \quad g(x) = \sum_{i=0}^n a_i x^i$$

Las series de Fourier provienen de la combinación li-

neal de las funciones de Fourier.

$$f(x) \approx a_0 + a_1 \cos x + a_2 \cos 2x + \dots + a_n \cos nx \\ + b_1 \sin x + b_2 \sin 2x + \dots + b_n \sin nx$$

$$\text{ó } g(x) = a_0 + \sum_{k=1}^n (a_k \cos kx + b_k \sin kx)$$

Aproximaciones que usan exponenciales son usualmente de la forma

$$f(x) \approx g(x) = \sum_{i=0}^n a_i e^{b_i x}$$

Las aproximaciones racionales

$$f(x) \approx g(x) = \frac{a_0 + a_1 x^1 + a_2 x^2 + \dots + a_n x^n}{b_0 + b_1 x^1 + b_2 x^2 + \dots + b_m x^m} = \frac{P_n(x)}{P_m(x)}$$

son usadas, aunque con menor frecuencia que los polinomios y series de Fourier.

De las funciones aproximantes mencionadas, la teoría de la aproximación polinomial es la más desarrollada y su ejecución es relativamente fácil; se la usa con mayor frecuencia porque, además de existir una buena justificación analítica, los polinomios poseen propiedades importantes. La suma, la resta, la integración y

la diferenciación de polinomios son también polinomios; las operaciones de integración y diferenciación no son difíciles de realizar. Si se efectúa un desplazamiento del sistema de coordenadas o si la escala de la variable independiente se cambia, la función transformada es también un polinomio; esto es, si $P_n(x)$ es un polinomio, lo es también $P_n(x+a)$ y $P_n(ax)$. Las otras funciones aproximantes gozan también de algunas propiedades; pero casi invariablemente, su evaluación se la hace en términos de polinomios o de relación de polinomios. Por lo mencionado, se analizará sólo la aproximación polinomial.

Sin embargo, debe tenerse presente que funciones periódicas suelen ser aproximadas eficientemente con series de Fourier; funciones con un carácter exponencial, serán descritas más adecuadamente con sumas de exponenciales, etc.

2.3.- APROXIMACION POLINOMIAL.

La idea central de todas las aproximaciones, en particular la aproximación polinomial, es mantener el valor absoluto del error $|f(x) - g(x)|$ lo más pequeño. La aproximación polinomial en la que $g(x)$, como ya se estableció es de la forma

$$g(x) = P_n(x) = \sum_{i=0}^n a_i x^i \quad (2.1)$$

se acerca a esta meta de varias maneras, entre las cuales se tiene:

1. colocación; 2. osculación; 3. cuadrados mínimos;
4. mín-máx ; 5. series de potencia.

Las distintas maneras de interpolación polinomial difieren entre sí en el criterio de encajar los datos discretos que se posee tanto de la función $f(x)$ como de la variable independiente x ; es decir en el proceso de cálculo de los coeficientes a_i .

Únicamente se va a referir en el presente trabajo a los polinomios de colocación y dentro de éstos, específicamente, a tres de ellos.

También será necesario utilizar interpolación inversa, o sea conocido $f(x)$ encontrar x ; suponiendo también conocidos los pares de valores $(x_i; f(x_i))$; $i = 0, 1, \dots, n$. El método de interpolación inversa, que se va a emplear consiste en intercambiar entre sí los papeles de la variable independiente y los de la variable dependiente. Una vez realizado este intercambio se procede a encontrar una función aproximante, utilizando cualquiera de las técnicas de aproximación; siendo los puntos bases

los $f(x_i)$ y las muestras de la función los x_i .

2.3.1.- POLINOMIOS DE COLOCACION.

Estos polinomios coinciden (se colocan) con $f(x)$ en ciertos puntos especificados. Según esto, dado los pares de valores $(x_i, f(x_i))$, $i = 0, 1, 2, \dots, n$ se debe cumplir que

$$P_n(x_i) = f(x_i) \quad i = 0, 1, 2, \dots, n \quad (2.2)$$

De esta manera, el polinomio de n ésimo grado $P_n(x)$ reproduce exactamente a $f(x)$ en los $n+1$ argumentos $x = x_i$. Este criterio tiene importancia porque según el teorema de existencia y unicidad (del álgebra elemental) hay uno y únicamente un polinomio de grado n o menos que asume todos los valores especificados para los $n+1$ argumentos distintos. El polinomio de colocación para $n = 3$ se ilustra en la figura 2.1; aquí se puede observar que el cumplimiento del criterio (2.2) no garantiza una aproximación exacta de $f(x)$ para un argumento $x \neq x_i$, salvo el caso que $f(x)$ sea un polinomio de grado menor o igual a n .

El hacer que se cumpla el criterio (2.2) para los $n+1$ pares de valores $(x_0, f(x_0))$, $(x_1, f(x_1))$, ..., $(x_n, f(x_n))$ da un conjunto de $n+1$ ecuaciones lineales

les simultáneas, de las cuales se puede obtener los

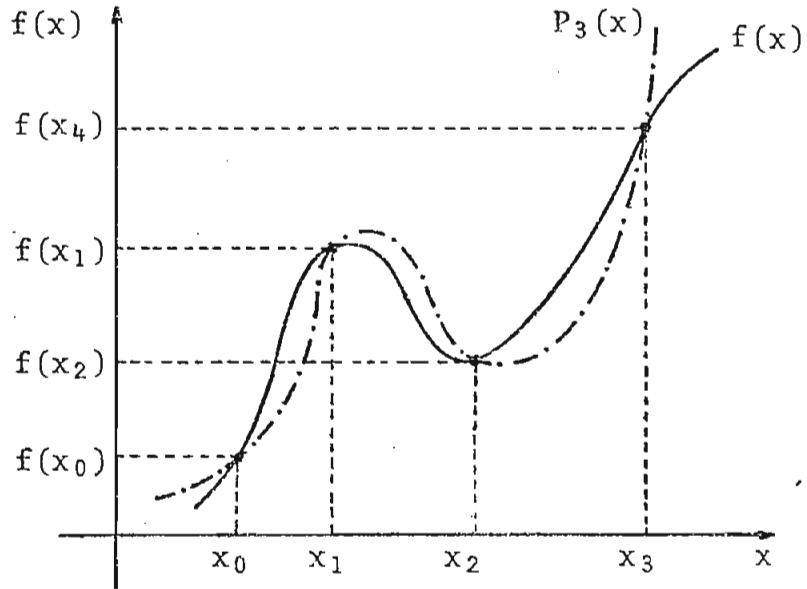


FIGURA 2.1 INTERPOLACION POLINOMIAL

coeficientes a_i del polinomio de colocación

$$a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n = P_n(x_0) = f(x_0)$$

$$a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n = P_n(x_1) = f(x_1)$$

⋮

⋮

$$a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n = P_n(x_n) = f(x_n)$$

Escribiendo en forma matricial se tiene

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & & & & \vdots \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ \vdots \\ f(x_n) \end{bmatrix}$$

o de una manera compacta $[X] \cdot [A] = [F]$

El determinante de la matriz $[X]$ se lo denomina determinante de VANDERMONDE y es diferente de cero si $x_i \neq x_j$ para $i \neq j$ (ecuaciones linealmente independientes). Esto ratifica la existencia y unicidad del polinomio de colocación $P_n(x)$ que reproduce con exactitud a $f(x)$ para los puntos bases.

Sin embargo el sistema de ecuaciones no es utilizado para la determinación de los coeficientes a_i , sino que más bien se utilizan fórmulas de interpolación. La razón radica en que la resolución de un sistema de ecuaciones en general no es una tarea fácil, especialmente cuando se le realiza manualmente. Además el término de error que produce una interpolación por fórmulas, amenudo se lo puede expresar como un producto; aunque no se puede saber exactamente la

magnitud de dicho error, existe la posibilidad de obtener una estimación del mismo.

Las fórmulas existentes para interpolación polinomial se pueden clasificar en dos grupos: Un grupo utilizado para puntos bases espaciados arbitrariamente; y el otro aplicable para puntos bases espaciados igualmente, es decir puntos bases de la forma

$$\begin{aligned} x_0 & \\ x_1 &= x_0 + h \\ x_2 &= x_0 + 2h \\ &\cdot \\ &\cdot \\ &\cdot \\ x_i &= x_0 + ih \\ &\cdot \\ &\cdot \\ &\cdot \\ x_n &= x_0 + nh \end{aligned}$$

donde h representa el espaciamiento que hay entre dos puntos bases consecutivos ($h = x_{i+1} - x_i$; $i = 0, 1, 2, \dots, n-1$). En lo posterior, se verá únicamente tres de las más comunes fórmulas de interpolación para puntos bases arbitrariamente espaciados: EL POLINOMIO DE COLOCACION POR DIFERENCIAS DIVIDIDAS DE NEWTON; EL POLINOMIO DE COLOCACION DE LAGRANGE; LOS POLINOMIOS DE COLOCACION SPLINE CUBICOS.

Necesariamente, para que el determinante de Vandermonde sea distinto de cero, los puntos bases deben ser diferentes entre sí.

Las fórmulas para intervalos iguales en la actualidad han perdido su importancia, ya que la necesidad por la que surgieron, la interpolación en tablas en las cuales los puntos bases en general son igualmente espaciados, ya no existe debido al uso de las computadoras digitales de alta velocidad, pues los valores funcionales necesarios para los cálculos, digitales son generados casi siempre por subrutinas de un programa biblioteca. "Todas estas fórmulas pueden ser derivadas ya sea del polinomio de Lagrange o del polinomio de diferencias divididas de Newton". (1). Por esto y por lo que es más importante para este trabajo, que se dispondrá de puntos bases arbitrariamente espaciados en la mayoría de los casos, se hará caso omiso de las fórmulas para puntos bases igualmente espaciados.

2.4.- POLINOMIO DE COLOCACION POR DIFERENCIAS DIVIDIDAS DE NEWTON

Este polinomio de colocación de grado n , $P_n(x)$, se expresa de la forma

$$P_n(x) = f[x_0] + (x-x_0) f[x_1, x_0] +$$

$$\begin{aligned}
 &+ (x-x_0)(x-x_1) f [x_2, x_1, x_0] + \dots + \\
 &+ (x-x_0)(x-x_1) \dots (x-x_{n-1}) f [x_n, x_{n-1}, \dots, x_0]
 \end{aligned}
 \tag{2.3}$$

en donde x_0, x_1, \dots, x_n son los $n+1$ puntos bases para los cuales la función $f(x)$ es conocida y $f [x_0]$, $f [x_1, x_0]$, \dots , $f [x_n, x_{n-1}, \dots, x_0]$ son las diferencias finitas divididas de orden 0, 1, etc., hasta n ; cuyas definiciones se indican en la tabla 2.1, en la que se puede ver de una manera explícita la depen-

TABLA 2.1 DEFINICIONES DE LAS DIFERENCIAS FINITAS DIVIDIDAS.

ORDEN	NOTACION	DEFINICION
0	$f [x_i]$	$f(x_i) ; i = 0, 1, \dots, n$
1	$f [x_{i+1}, x_i]$	$\frac{f [x_{i+1}] - f [x_i]}{x_{i+1} - x_i} ; i = 0, 1, \dots, n-1$
2	$f [x_{i+2}, x_{i+1}, x_i]$	$\frac{f [x_{i+2}, x_{i+1}] - f [x_{i+1}, x_i]}{x_{i+2} - x_i} ; i = 0, 1, \dots, n-2$
3	$f [x_{i+3}, x_{i+2}, x_{i+1}, x_i]$	$\frac{f [x_{i+3}, x_{i+2}, x_{i+1}] - f [x_{i+2}, x_{i+1}, x_i]}{x_{i+3} - x_i} ; i = 0, 1, \dots, n-3$
⋮		⋮
n	$f [x_n, x_{n-1}, \dots, x_0]$	$\frac{f [x_n, x_{n-1}, \dots, x_1] - f [x_{n-1}, x_{n-2}, \dots, x_0]}{x_n - x_0} ; i = 0$

dencia del polinomio de colocación de los $n+1$ valores conocidos de $f(x)$; $f(x_0), f(x_1), \dots, f(x_n)$.

El denominador en cada diferencia finita dividida está constituido por la diferencia de los argumentos que no son comunes en las dos diferencias finitas del numerador. En general si m es el orden de la diferencia finita y $n+1$ el número de pares de valores $(x_i, f(x_i))$; $i = 0, 1, 2, \dots, n$, se pueden obtener $n-m+1$ diferencias finitas diferentes de orden m . Así por ejemplo para $n = 4$ hay la posibilidad de 4 diferencias finitas de orden uno; 3 de orden dos; 2 de orden tres y 1 de orden cuatro. De esta manera se puede desarrollar una tabla de diferencias finitas divididas, como la que se indica en la tabla 2.2 para cuatro valores muestreados de la función $f(x) = x^3 - 2x^2 + 7x - 5$. Las mismas

TABLA 2.2 TABLA DE DIFERENCIAS FINITAS DIVIDIDAS

i	x_i	$f(x_i)$	$f_1[]^*$	$f_2[]$	$f_3[]$
0	0,	-5,	6,		
1	1,	1,	12,	2,	1,
2	3,	25,	30,	6,	
3	4,	55,			

* $f_1[]$ es una abreviación de la primera diferencia finita dividida, $f_2[]$ de la segunda, etc.

que fueron obtenidas a partir de las definiciones dadas en la tabla 2.1 de la siguiente manera:

$$f [x_1, x_0] = \frac{f [x_1] - f [x_0]}{x_1 - x_0} = \frac{1 - (-5)}{1 - 0} = 6$$

$$f [x_2, x_1] = \frac{f [x_2] - f [x_1]}{x_2 - x_1} = \frac{25 - 1}{2 - 1} = 12$$

$$f [x_3, x_2] = \frac{f [x_3] - f [x_2]}{x_3 - x_2} = \frac{55 - 25}{4 - 3} = 30$$

$$f [x_2, x_1, x_0] = \frac{f [x_2, x_1] - f [x_1, x_0]}{x_2 - x_0} = \frac{12 - 6}{3 - 0} = 2$$

$$f [x_3, x_2, x_1] = \frac{f [x_3, x_2] - f [x_2, x_1]}{x_3 - x_1} = \frac{30 - 12}{4 - 1} = 6$$

$$f [x_3, x_2, x_1, x_0] = \frac{f [x_3, x_2, x_1] - f [x_2, x_1, x_0]}{x_3 - x_0} = \frac{6 - 2}{4 - 0} = 1$$

Utilizando los valores de la tabla 2.2 se va a estimar el valor de la función $f(x)$ para $x = 0,5$ por medio de los polinomios de colocación de grado uno, dos y tres.

$$P_1 (x) = f [x_0] + (x-x_0) f [x_1, x_0]$$

$$P_1 (x) = -5 + (x-0) 6 = 6x - 5$$

$$P_1 (0,5) = 6 (0,5) - 5 = -2$$

$$P_2 (x) = f [x_0] + (x-x_0) f [x_1, x_0] + (x-x_0)(x-x_1) f [x_2, x_1, x_0]$$

$$\begin{aligned}P_2(x) &= -5 + (x-0) 6 + (x-0)(x-1) 2 \\ &= -5 + 6x + 2x^2 - 2x\end{aligned}$$

$$P_2(x) = 2x^2 + 4x - 5$$

$$P_2(0,5) = 2(0,25) + 4(0,5) - 5 = -2,5$$

$$\begin{aligned}P_3(x) &= f[x_0] + (x-x_0) f[x_1, x_0] + \\ &\quad + (x-x_0)(x-x_1) f[x_2, x_1, x_0] + \\ &\quad + (x-x_0)(x-x_1)(x-x_2) f[x_3, x_2, x_1, x_0]\end{aligned}$$

$$\begin{aligned}P_3(x) &= -5 + (x-0) 6 + (x-0)(x-1) 2 + \\ &\quad + (x-0)(x-1)(x-3) 1\end{aligned}$$

$$P_3(x) = -5 + 6x + 2x^2 - 2x + x^3 - 4x^2 + 3x$$

$$P_3(x) = x^3 - 2x^2 + 7x - 5$$

$$P_3(0,5) = 0,125 - 2(0,25) + 7(0,5) - 5 = -1,725$$

Dado que para este caso particular $f(x)$ es un polinomio de tercer grado se cumple que $f(x) = P_3(x)$. También se nota en este ejemplo que no siempre un polinomio de mayor grado da la mejor estimación, pues $P_1(x)$ genera menor error que $P_2(x)$ en la estimación hecha de $f(x)$ para $x = 0,5$ cuyo valor real está dado por $P_3(x)$.

A la expresión

$$f(x) = P_n(x) + R_n(x) \tag{2.4}$$

se la conoce con el nombre de FORMULA FUNDAMENTAL DE NEWTON. En donde $P_n(x)$ es el polinomio de colocación por diferencias finitas de Newton dado por (2.3).

$R_n(x)$ es el término restante o el error en el que se incurre al substituir $f(x)$ por $P_n(x)$ y está dado por

$$R_n(x) = (x-x_0)(x-x_1)\dots(x-x_{n-1})(x-x_n) f [x, x_n, \dots, x_0] \quad (2.4 a)$$

o; en forma más compacta por

$$R_n(x) = \left\{ \prod_{i=0}^n (x-x_i) \right\} \cdot f [x, x_n, \dots, x_0] \quad (2.4 b)$$

x en $[x_0, x_1, \dots, x_n]$

Suponiendo que es conocido otro par adicional $(x_{n+1}, f(x_{n+1}))$ y asumiendo que la diferencia finita dividida de orden $n+1$ permanece casi constante se puede escribir

$$f [x, x_n, \dots, x_0] \approx f [x_{n+1}, x_n, \dots, x_0]$$

entonces $R_n(x)$ quedaría

$$R_n(x) \approx \left\{ \prod_{i=0}^n (x-x_i) \right\} \cdot f [x_{n+1}, x_n, \dots, x_0] \quad (2.5)$$

Lastimosamente (2.5) permite tan sólo tener una estimación de $R_n(x)$ y no un límite superior del mismo. Para

el caso que $f(x)$ sea conocida analíticamente, la expresión (2.4) tiene su importancia, pues permite establecer un límite superior para el error $R_n(x)$. Según (1) se tiene que

$$f [x, x_n, x_{n-1}, \dots, x_0] = \frac{f^{(n+1)}(\epsilon)}{(n+1)!} \quad (2.6)$$
$$\epsilon \text{ en } [x_n, x_{n-1}, \dots, x_0]$$

Sustituyendo (2.6) en (2.4) se obtiene

$$R_n(x) = \left\{ \prod_{i=0}^n (x-x_i) \right\} \cdot \frac{f^{(n+1)}(\epsilon)}{(n+1)!} \quad (2.7)$$

Al escoger ϵ de tal manera que la derivada de orden $n+1$ de $f(x)$ evaluada para dicho argumento, $f^{(n+1)}(\epsilon)$, sea máxima se obtiene el límite superior para $R_n(x)$.

Cuando se ha preparado una tabla de diferencias finitas con $n+1$ puntos bases, se puede ensayar cualquier polinomio de colocación de grado m , siendo m menor o a lo mucho igual a n . Si no se usa toda la tabla ($m < n$), se debe escoger los puntos bases más cercanos al argumento para el cual se desea obtener el valor de la función por interpolación, para reducir el error.

Esto se ve claramente si se examina la ecuación (2.7),

dado que $f^{(n+1)}(\epsilon)$ es desconocida en la mayoría de los casos, hay que hacer que el valor absoluto del término dado por

$$\left| \prod_{i=0}^n (x-x_i) \right| \quad (2.8)$$

tenga el valor más pequeño posible. En general (2.8) tiene un valor mayor cuando x es próximo a los dos extremos x_0 y x_n que cuando x es próximo a $x_n/2$. Entonces para disminuir el error de aproximación se debe escoger los puntos bases de tal manera que el argumento x esté centrado en lo posible entre dichos puntos bases, cuyo número depende del grado del polinomio m que se haya escogido previamente. Todo presupone que los puntos bases están ordenados de menor a mayor; es decir que $x_0 < x_1 < x_2 < \dots < x_n$.

En la tabla 2.3 se indica un conjunto de valores de la función $f(x) = \text{Cos } x$. De los cuales, la tabla de diferencias finitas divididas y los valores interpolados se obtuvieron mediante cálculo utilizando la computadora digital IBM 370 y el programa dado en (1) que se desarrolla basándose en los criterios expuestos. (Ver pormenores en el Apéndice 1).

TABLA 2.3 DATOS Y RESULTADOS DE LA FUNCION $f(x) = \cos x$, DETERMINADOS MEDIANTE INTERPOLACION BIDIMENSIONAL, UTILIZANDO EL POLINOMIO DE COLOCACION POR DIFERENCIAS FINITAS DIVIDIDAS DE NEWTON COMO FUNCION APROXIMANTE

VALORES UTILIZADOS PARA EL CALCULO DE LA TABLA DE DIFERENCIAS DIVIDIDAS

ARGUMENTO	FUNCION
1 0.0	1.000000
2 0.2000	0.980067
3 0.4000	0.917060
4 0.6000	0.812693
5 0.8000	0.695096
6 1.0000	0.547300
7 1.2000	0.369731
8 1.4000	0.167061
9 1.6000	0.000000
10 1.8500	0.085598

YACLA DE DIFERENCIAS DIVIDIDAS PARA $M = 4$

-0.996671115-01	..045211265D 00	0.355090472D-01	0.403039390D-01	-0.24576321D-02	-0.13351041E-02
-0.24730039D 00	-0.474983092D 00	0.51212047D-01	0.38952241D-01	-0.32406958D-02	-0.12428171E-02
-0.31927551D 00	-0.469588E1D 00	0.51212047D-01	0.37555540E-01	-0.23271555D-02	-0.11916754E-02
-0.32662343D 00	-0.45337718D 00	0.44645533D-01	0.35888672E-01	-0.15223125D-02	-0.11516754E-02
-0.45690981E 00	-0.434934165D 00	0.76142121D-01	0.32653315D-01	-0.45231256D-02	-0.11516754E-02
-0.54377414D 00	-0.40266745D 00	0.90497611E-01	0.32653315D-01	-0.45231256D-02	-0.11516754E-02
-0.62431164D 00	-0.37665775D 00	0.10403854D 00	0.31563062D-01	-0.50694466D-02	-0.11226660D-02
-0.699964319D 00	-0.34828679D 00	0.11350786D 00			
-0.73447127D 00					

ARGUMENTO	GRADO	VALOR INTERPOLADO	VALOR VERDADERO	IND
0.100000	1	0.999999724	0.99500417	0
0.100000	2	0.999999724	0.99500417	0
0.100000	3	0.999999724	0.99500417	0
0.100000	4	0.999999724	0.99500417	0
0.100000	5	0.999999724	0.99500417	0
0.100000	6	0.999999724	0.99500417	0
0.100000	7	0.999999724	0.99500417	0
0.100000	8	0.999999724	0.99500417	0
0.100000	9	0.999999724	0.99500417	0
0.100000	10	0.999999724	0.99500417	0
0.430000	1	0.900000000	0.910000000	0
0.430000	2	0.900000000	0.910000000	0
0.430000	3	0.900000000	0.910000000	0
0.430000	4	0.900000000	0.910000000	0
0.430000	5	0.900000000	0.910000000	0
0.430000	6	0.900000000	0.910000000	0
0.430000	7	0.900000000	0.910000000	0
0.430000	8	0.900000000	0.910000000	0
0.430000	9	0.900000000	0.910000000	0
0.430000	10	0.900000000	0.910000000	0
0.825000	1	0.672555956	0.662555956	0
0.825000	2	0.672555956	0.662555956	0
0.825000	3	0.672555956	0.662555956	0
0.825000	4	0.672555956	0.662555956	0
0.825000	5	0.672555956	0.662555956	0
0.825000	6	0.672555956	0.662555956	0
0.825000	7	0.672555956	0.662555956	0
0.825000	8	0.672555956	0.662555956	0
0.825000	9	0.672555956	0.662555956	0
0.825000	10	0.672555956	0.662555956	0

001818

Todos los cálculos se realizaron en doble precisión. En la tabla de diferencias finitas constan hasta las de orden $m = 6$. Los valores interpolados de la función coseno se los compara con aquellos que se obtienen a partir de la función DCOS que forma parte de la biblioteca de la computadora. Se observa que los valores estimados de la función $\text{Cos } x$ por medio de los polinomios de Newton desde un grado tres en adelante difieren del valor real a partir de la quinta cifra significativa; diferencia que de un polinomio a otro no es muy perceptible.

El polinomio de colocación por diferencias finitas divididas presenta gran versatilidad, por su estructura, al cambio del grado del polinomio. Así para cambiar de un grado j a $j+1$ sólo se requiere añadir un término adicional sin que sea necesario repetir los cálculos anteriores. Esto tiene importancia especialmente cuando el grado del polinomio es desconocido y se desea que el valor absoluto del término de error no sobrepase un límite fijado.

En resumen cuando van a ser llevadas a cabo muchas interpolaciones sobre un mismo conjunto de datos discretos, el polinomio de colocación por diferencias finitas divididas ahorra tiempo de cálculo, ya que hay un conjunto de cálculos, que sirven para determinar la ta

bla de diferencias finitas, que son realizadas solamente una vez.

Por propósitos del presente trabajo se necesita interpolación inversa; es decir dado $f(x)$ encontrar x , y tal como se ha expuesto la teoría del polinomio de colocación por diferencias finitas divididas, este proceso presentaría algunas complicaciones; las que se reducirían al considerar todos los $n+1$ pares discretos $(x_i, f(x_i))$; $i = 0, 1, 2, \dots, n$. para encontrar el valor interpolado; esto es equivalente a utilizar al polinomio de colocación de mayor grado $P_n(x)$. Pues de ser así, se necesitaría conocer sólo n diferencias finitas divididas una para cada orden (Más detalles se dan en el Apéndice 2).

2.5.- POLINOMIO DE COLOCACION DE LAGRANGE.

Asumiendo que el polinomio de colocación de enésimo grado $P_n(x)$, tiene la forma

$$\begin{aligned} P_n(x) = & a_0 (x-x_1)(x-x_2)\dots(x-x_n) + \\ & + a_1 (x-x_0)(x-x_2)\dots(x-x_n) + \\ & + a_2 (x-x_0)(x-x_1)\dots(x-x_n) + \\ & \vdots \\ & \vdots \\ & \vdots \end{aligned}$$

$$\begin{aligned}
 &+ a_i (x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n) + \\
 &\quad \vdots \\
 &\quad \vdots \\
 &+ a_n (x-x_0)(x-x_1)\dots(x-x_{n-2})(x-x_{n-1})
 \end{aligned}
 \tag{2.9}$$

Los coeficientes a_0, a_1, \dots, a_n se determinan haciendo cumplir que $f(x_i) = P_n(x_i)$, $i = 0, 1, \dots, n$. De esto resulta que

$$a_i = \frac{f(x_i)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}
 \tag{2.10}$$

Substituyendo (2.10) en (2.9) se llega al polinomio de colocación de LAGRANGE.

$$\begin{aligned}
 P_n(x) = &\frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)} f(x_0) + \\
 &+ \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)} f(x_1) + \\
 &\quad \vdots \\
 &\quad \vdots \\
 &+ \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)} f(x_i) + \\
 &\quad \vdots \\
 &\quad \vdots
 \end{aligned}
 \tag{2.11}$$

$$+ \frac{(x-x_0)(x-x_1)\dots(x-x_{n-2})(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-2})(x_n-x_{n-1})} f(x_n)$$

En forma condensada se tiene

$$P_n(x) = \sum_{i=0}^n L_i(x) f(x_i) \quad (2.12)$$

donde $L_i(x)$ es la FUNCION MULTIPLICADORA de Lagrange.

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x-x_j)}{(x_i-x_j)} \quad (2.13)$$

y tiene las propiedades

$$L_i(x_k) = 0 \text{ para } k \neq i$$

$$L_i(x_i) = 1$$

Manipulando algebraicamente el polinomio de colocación por diferencias divididas de Newton, aunque es tedioso, se puede llegar al polinomio de colocación de Lagrange (Ver (1)). Por lo tanto todo lo que se habló ya acerca del polinomio de Newton es válido para el de Lagrange.

Tome en cuenta que el polinomio de colocación de La-

grange solamente involucra los puntos bases x_i y los correspondientes valores de la función $f(x_i)$. Cuando una sola interpolación es efectuada, la cantidad de cálculos realizados tanto por la forma de Newton como por la forma de Lagrange es equivalente. Esta última forma requiere menos memoria de computadora, pues existe un ahorro en lo que respecta a la tabla de diferencias divididas que no es necesario para este caso de aproximación polinomial.

La forma de Lagrange presenta dos significantes desventajas. Por un lado está la cantidad de cálculos requeridos cuando se hace muchas interpolaciones usando el mismo conjunto de datos. La otra desventaja se presenta frente a la necesidad de incrementar el grado del polinomio, ya que se debe repetir todos los cálculos, lo que no sucede con la forma de Newton en la que sólo hay que añadir términos adicionales.

Para enfocar lo más antes posible la aproximación polinomial en tres dimensiones de una función $f = f(x,y)$, se aplicará el polinomio de colocación de Lagrange en este sentido.

Con el propósito de que el lector entienda el método de aproximación en tres dimensiones que se va a usar, se explicará la aproximación lineal (polinomio de pri-

mer grado). Suponiendo que se está utilizando un total de $m \times n$ valores de la función $f(x_i, y_j)$, que representa todas las combinaciones posibles de los m niveles de x_i ($i = 1, 2, \dots, m$) y los n niveles de y_j ($j = 1, 2, \dots, n$). Por conveniencia a los valores $f_{ij} \equiv f(x_i, y_j)$ se los arregla en una tabla de dos dimensiones, en la que las filas corresponden a $x = x_i$ y las columnas a $y = y_j$. Entonces, dado un valor arbitrario de x y otro de y encontrar por interpolación en la tabla una aproximación a $f(x, y)$. Considerando que $x_i \leq x \leq x_{i+1}$ y $y_j \leq y \leq y_{j+1}$, como se indica en la figura 2.2.

El símbolo \square señala los puntos para los cuales el valor de la función es conocido.

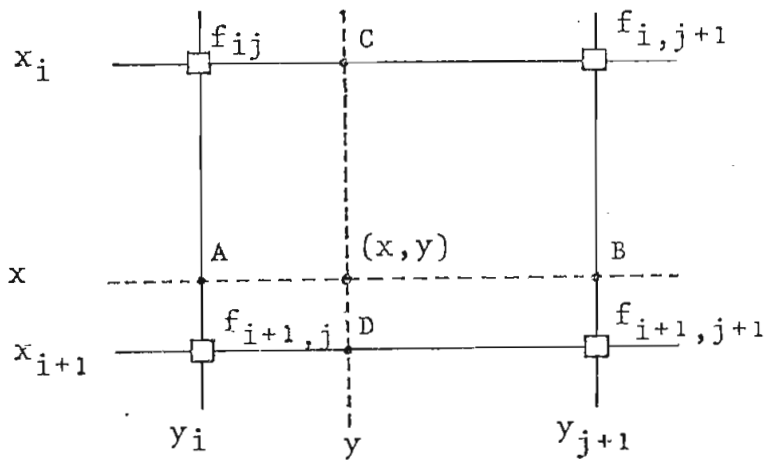


FIGURA 2.2 INTERPOLACION LINEAL EN TRES DIMENSIONES.

Interpolando linealmente entre f_{ij} y $f_{i+1,j}$; y ; $f_{i,j+1}$ y $f_{i+1,j+1}$ se obtiene las aproximaciones f_A y f_B a $f(x,y)$ de la siguiente manera:

$$\frac{x - x_i}{x_{i+1} - x_i} = \frac{f_A - f_{ij}}{f_{i+1,j} - f_{ij}} \Rightarrow$$

$$f_A = \frac{x - x_i}{x_{i+1} - x_i} \cdot (f_{i+1,j} - f_{ij}) + f_{ij}$$

$$f_A = (1-\alpha) \cdot f_{ij} + \alpha \cdot f_{i+1,j} \quad ; \quad \alpha = \frac{x - x_i}{x_{i+1} - x_i} \quad (2.14)$$

Procediendo de una manera análoga se tiene

$$f_B = (1-\alpha) \cdot f_{i,j+1} + \alpha \cdot f_{i+1,j+1} \quad (2.15)$$

Finalmente interpolando linealmente entre f_A y f_B , tomando en cuenta la variable independiente y , se obtiene la aproximación final a $f(x,y)$.

$$\frac{y - y_j}{y_{j+1} - y_j} \approx \frac{f(x,y) - f_A}{f_B - f_A} \Rightarrow$$

$$f(x,y) \approx (1-\beta) \cdot f_A + \beta \cdot f_B \quad ; \quad \beta = \frac{y - y_j}{y_{j+1} - y_j} \quad (2.16)$$

Reemplazando (2.14) y (2.15) en (2.16) se tiene

$$f(x,y) \approx (1-\alpha)(1-\beta) f_{ij} + \alpha(1-\beta) f_{i+1,j} + \\ + \beta(1-\alpha) f_{i,j+1} + \alpha\beta f_{i+1,j+1} \quad (2.17)$$

Al mismo resultado se hubiera llegado si primero se interpolaba en la dirección y (permaneciendo x constante) para determinar f_C y f_D . Luego interpolando en la dirección x (permaneciendo y constante) determinar la aproximación lineal a $f(x,y)$.

La expresión (2.17) representa un polinomio de colocación de primer grado tanto en x como en y (las dos variables independientes).

Como una extensión, en el caso que se desee encontrar una aproximación mediante un polinomio de colocación de grado dos, tres, etc., se debe trabajar con 3, 4, etc. filas y columnas del arreglo bidimensional de las muestras tomadas de la función $f(x,y)$. Es conveniente aunque no indispensable, que el número de filas escogidas sea igual al de columnas seleccionadas.

Con este método se puede utilizar cualquiera de las técnicas de aproximación. A continuación se elabora un programa para la computadora que estima el valor de la función $f(x,y) = x + \cos y \cdot \sin x$ utilizando el polinomio de colocación de Lagrange.

El programa tiene las siguientes etapas:

- 1.- Lee el número de muestras tomadas de la variable independiente x (número de filas), m ; el número de muestras de la variable independiente y (número de columnas), n ; los puntos bases o muestras de las dos variables independientes que son almacenados en dos arreglos unidimensionales uno por cada variable.
- 2.- Con los puntos bases leídos se calcula la función $f(x_i, y_j) = z_{ij} = x_i + \text{Cos } y_j \cdot \text{Sen } x_i$; $i = 1, 2, \dots, m$; $y j = 1, 2, \dots, n$ utilizando las dos funciones que existen en la biblioteca de la computadora, DCOS y DSIN. Estos valores conocidos por evaluación directa de la función $f(x,y)$ se los almacena en un arreglo de dos dimensiones. Se asume que los puntos bases, tanto de la variable x como de la variable y , están ordenados en forma ascendente (de menor a mayor).
- 3.- Escribe en forma ordenada las muestras de las variables independientes (datos de entrada) y de la variable dependiente o función (datos calculados).
- 4.- Lee el grado de los polinomios de colocación de Lagrange a utilizarse, d ; los valores de \bar{x} y \bar{y} pa-

ra los cuales se desea estimar el valor de la función, $\bar{z} = f(\bar{x}, \bar{y})$.

- 5.- Llama a la función DIRXY, que es la encargada de estimar el valor de la función, $\bar{z}(\bar{x}, \bar{y})$, mediante interpolación o extrapolación, pues el programa también permite extrapolar (\bar{x} y/o \bar{y} están fuera de los intervalos de las variables independientes, dentro de las cuales es conocida la función de una manera discreta).

DIRXY asume que las muestras están ordenadas de acuerdo a la tabla 2.4, en donde $x_1 < x_2 < \dots < x_m$, $y_1 < y_2 < \dots < y_n$.

- 6.- El valor estimado, \bar{z} , es comparado con aquel que se obtiene directamente al evaluar la función, $f(x,y)$, utilizando las dos funciones ya mencionadas que forman parte de la biblioteca de la computadora.

Para otros tres valores de d , \bar{x} , \bar{y} se repite el programa desde el numeral 4.

La FUNCION DIRXY tiene la siguiente secuencia:

- 1.- Verifica que d , el grado de los polinomios que se

TABLA 2.4 FORMA EN QUE DEBEN ESTAR ORDENADAS LAS MUESTRAS PARA INTERPOLACION TRIDIMENSIONAL

	y_1	y_2	. . .	y_n
x_1	$f(x_1, y_1)$	$f(x_1, y_2)$. . .	$f(x_1, y_n)$
x_2	$f(x_2, y_1)$	$f(x_2, y_2)$. . .	$f(x_2, y_n)$
.
.
.
x_m	$f(x_m, y_1)$	$f(x_m, y_2)$. . .	$f(x_m, y_n)$

van a usar, sea menor que m o n .

- 2.- Utiliza la FUNCION MINMON para determinar que muestras de las variables independientes y de la variable dependiente deben intervenir en la interpolación; de tal manera, que los valores \bar{x} y \bar{y} para los cuales se desea estimar el valor de la función, \bar{z} , queden lo mejor centrados; es decir, que elementos de los respectivos arreglos deben ser utilizados.
- 3.- Trabajando unicamente con los $d+1$ puntos bases seleccionados de la variable independiente x , y con

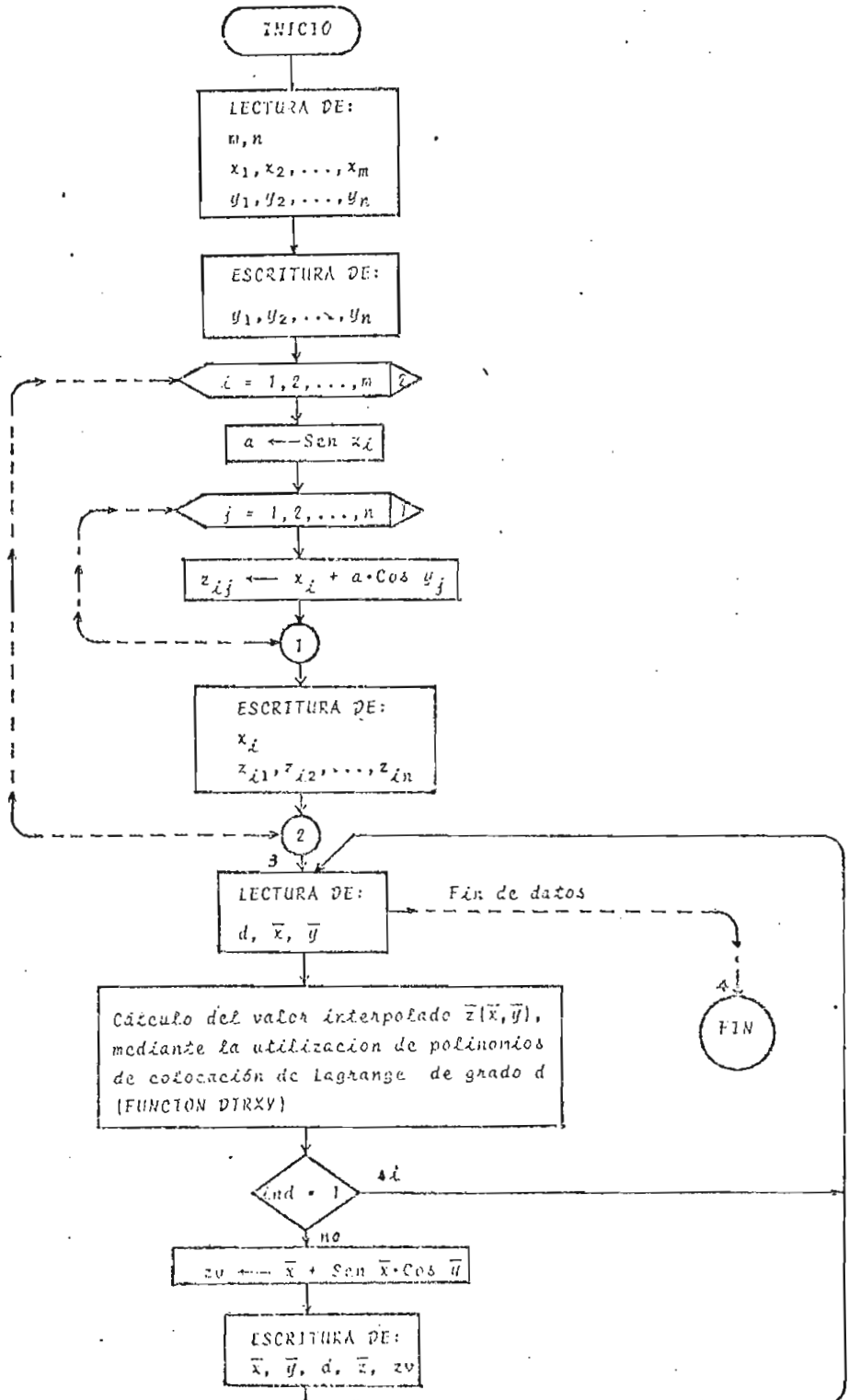
los $(d+1)^2$ muestras de la variable dependiente z se halla $d+1$ valores -uno por cada uno de los puntos bases de la variable y , que intervienen- que resultan de evaluar $d+1$ polinomios de colocación de Lagrange de grado d mediante la FUNCION DIRY. Estos valores son función tan sólo de la variable independiente y .

4.- Procediendo como interpolación en dos dimensiones se determina el valor aproximado, \bar{z} , utilizando la FUNCION DIRY cuyos detalles se dan en el Apéndice 3 por ser obtenida de (1).

El programa principal y DIRXY tienen una señalización, ind , es uno cuando se ha encontrado alguna inconsistencia y cero cuando todo es normal.

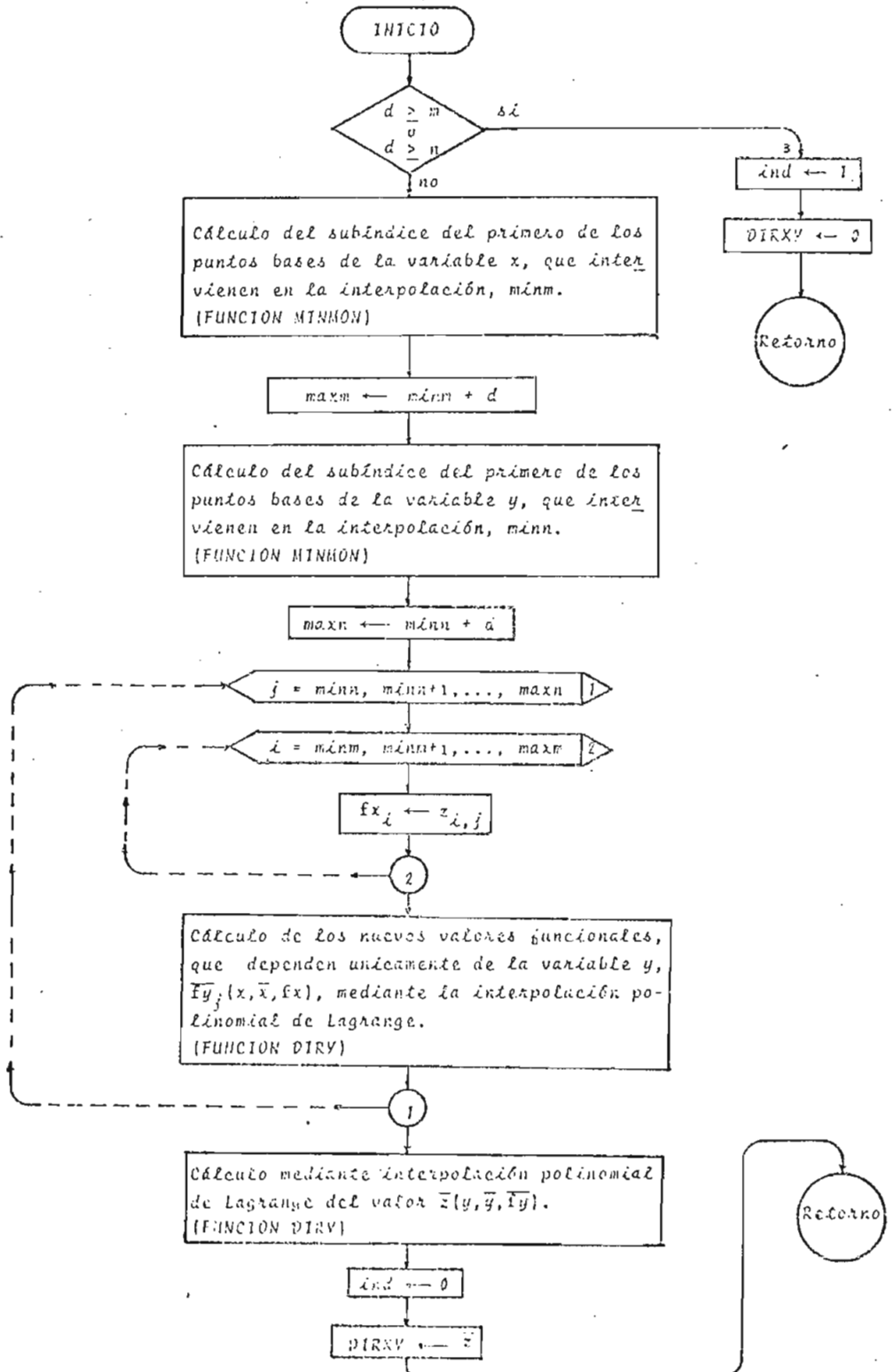
DIAGRAMAS DE FLUJO

PROGRAMA PRINCIPAL:



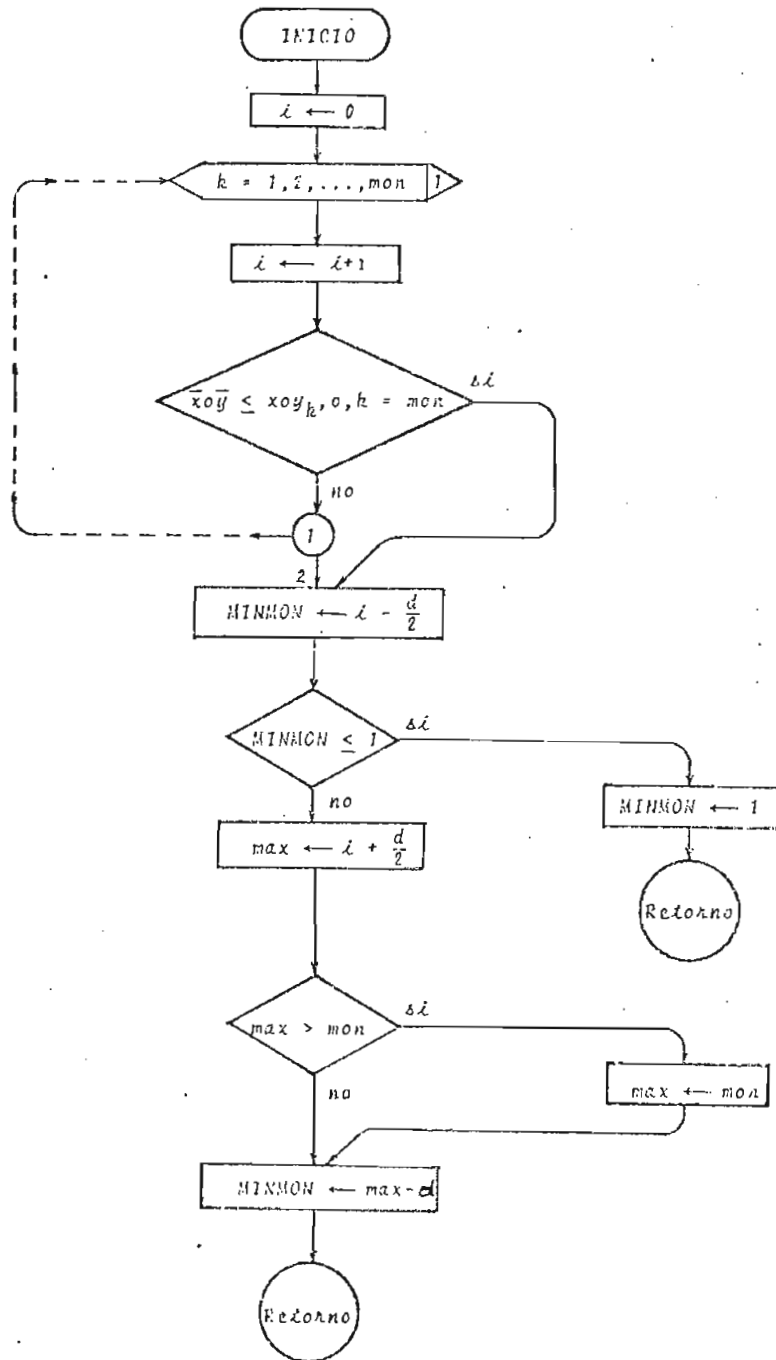
FUNCION DIRXY:

(Argumentos: $m, n, x, y, z, \bar{x}, \bar{y}, d, ind$).



FUNCION MINMON:

(Argumentos: mon, xoy, \bar{xoy} , d).



CODIFICACION FORTRAN

LISTA DE VARIABLES PRINCIPALES

Símbolos en *Definición*
los programas

(Principal)

I,J	Subíndices i, j .
NGRAD	Grado "d" de los polinomios de colocación de Lagrange.
M	Número de muestras, "m", tomadas de la variable x (número de filas).
N	Número de muestras, "n", tomadas de la variable y (número de columnas).
X	Arreglo de los puntos bases x_i ; $i = 1, 2, \dots, m$
Y	Arreglo de los puntos bases y_j ; $j = 1, 2, \dots, n$
FXY	Arreglo bidimensional de las muestras de la función, $z_{i,j} = f(x_i, y_j)$.
ARGX	Argumento de la variable x para interpolación, \bar{x} .
ARGY	Argumento de la variable y para interpolación, \bar{y} .

- FXYR Valor interpolado, $\bar{z} = f(\bar{x}, \bar{y})$.
- VERVAL Valor de $\bar{x} + \text{Sen } \bar{x} \cdot \text{Cos } \bar{y}$, calculado usando las funciones DSIN y DCOS que forman parte de la biblioteca de la computadora.
- IND Interruptor de cálculo: Es 1 si se encuentra inconsistencia y 0 cuando no la hay.

(Función DIRXY)

- MINM Subíndice del primero de los puntos bases de la variable x que intervienen en la interpolación, minm.
- MAXM minm + d.
- MINN Subíndice del primero de los puntos bases de la variable x que intervienen en la interpolación, minn.
- MAXN minn + d.
- FXI Arreglo unidimensional en el que se almacena d + 1 valores de una misma columna del arreglo z.
- FY Arreglo unidimensional en el que se almacena los d + 1 valores calculados por interpolación, uno por cada columna del arreglo z que son utilizadas.

IND Es un indicador, ind, es uno cuando existe inconsistencia y cero cuando se tiene lo contrario.

(Función MINMON)

MON Número de elementos del arreglo xoy.

XOY Nombre del arreglo en el que se almacenan los puntos bases de una variable, ordenados de menor a mayor.

ARGXOY Es el valor a compararse con los elementos del arreglo XOY.

LISTADO DEL PROGRAMA

PROGRAMA PRINCIPAL:

```

C
C PROGRAMA PARA CALCULAR LA FUNCION F(X,Y) = X + SEN(X) * COS(Y) MEDIANTE
C INTERPOLACION POLINOMIAL EN TRES DIMENSIONES UTILIZANDO EL POLINOMIO DE
C COLOCACION DE LAGRANGE.
C
C EL PROGRAMA LEE DOS CONJUNTOS DE VALORES X(I)...X(M) * Y * Y(I)...Y(N).
C CALCULA UN CONJUNTO DE VALORES FXY(I,1)...FXY(M,N) DE LA FORMA FXY(I,J) =
C X(I) + DSIN(X(I)) * DCOS(Y(J)). LEE LOS VALORES NGRAD, ARGX Y ARGY. PARA
C LOS DOS ULTIMOS VALORES SE DESEA CONOCER EL VALOR DE LA FUNCION, PXTX, POR
C INTERPOLACION. CON ESTE PROCEDIMIENTO SE LLAMA A DIRXY LA CUE SE ENCARGA DEL
C CALCULO DEL VALOR INTERPOLADO. ESTE VALOR ES COMPARADO LUEGO CON EL VALOR
C REAL, VVERVAL = ARGX + DSIN(ARGX) + DCOS(ARGY).
C
0001      IMPLICIT REAL*8 (A-H,O-Z)
0002      DIMENSION X(50),Y(50),FXY(50,50)
C
C LECTURA DE DATOS; CALCULO DE LOS VALORES FXY, Y ESCRITURA
C
0003      11 FORMAT(2I2,6X,7F10.2/(#F10.0))
0004      READ(1,11)M,N,(X(I),I=1,M),(Y(J),J=1,N)
0005      WRITE(3,22)(Y(J),J=1,N)
0006      22 FORMAT (1H1///3;#VALORES UTILIZADOS PARA LA INTERPOLACION EN TRES
C * DIMENSIONES//40X*DE LA FUNCION F(X,Y) = X + SEN(X) + COS(Y)///
C *1X*VARIABLE Y//3X*VARIABLE X [7F15.5/(15X*]7F15.5))
0007      WRITE(3,77)
0009      77 FORMAT(5X,116('---'))
0010      DO 2 I=1,M
0011      A=DSIN(X(I))
0012      DO 1 J=1,N
0013      1 FXY(I,J)=X(I)+A*DCOS(Y(J))
0014      2 WRITE(3,33)(X(I),(FXY(I,J),J=1,N)
0015      33 FORMAT(15X*]75X#9.5,1X*]7F15.10/(15X*]7F15.10))
0016      WRITE(3,66)
0017      66 FORMAT (1H1///25X*ARGUMENTO X'2X*ARGUMENTO Y'2X*GRACC'2X*VALOR *.
C *INTERPOLADO'2X*VALOR VERDADERO'/)
C NP = 0
C
C LECTURA DE NGRAD, ARGX, ARGY, LLAMADA A DIRXY PARA INTERPOLAR
C
0018      3 READ(1,44,END=4) NGRAD,ARGX,ARGY
0019      44 FORMAT(12,8X2F10.0)
0020      FXY=DIRXY(M,N,X,Y,FXY,ARGX,ARGY,NGRAD,IND)
0021      IF(1,0,FC,1) GO TO 3
0022      NP = NP + 1
0023      IF (MCD(NP,35),EQ,0) WRITE (3,66)
C
C CALCULO DEL VALOR VERDADERO ARGX + DSIN(ARGX) + DCOS(ARGY) Y ESCRITURA DE
C RESULTADOS.
C
0024      VVERVAL=ARGX+DSIN(ARGX)+DCOS(ARGY)
0025      WRITE(3,55) ARGX,ARGY,NGRAD,PXTX,VVERVAL
0026      55 FORMAT(7B3XF10.5,2XF10.5,6X,12,F10.10,F17.10)
0027      GO TO 2
0028      4 STOP
0029      END

```

FUNCION DIRXY:

```

0001      FUNCTION DIRXY (M,N,X,Y,FXY,ARGX,ARGY,NGRAD,IND)
C
C SUBPROGRAMA PARA EVALUAR MEDIANTE INTERPOLACION UNA FUNCION TRIDIMENSIONAL,
C QUE ES CONOCIDA DE UNA MANERA DISCRETA, UTILIZANDO LA TECNICA DE LAGRANGE EN
C DOS DIMENSIONES.
C
C DIRXY ASUME QUE TANTO LOS ELEMENTOS DE X COMO LOS DE Y ESTAN ORDENADOS
C DE MENOR A MAYOR. PRIMERO VERIFICA QUE EL GRADO DE LOS POLINOMIOS, NGRAD, NO
C SEA IGUAL O MAYOR QUE Y O N. LLAMA A MINON PARA DETERMINAR EL PRINCIPAL DE
C LOS ELEMENTOS DE X Y DE Y QUE DEBEN INTERVENIR. CALCULA NGRAD + 1 VACCRES.
C LOS ALMACENA EN FY UTILIZANDO DIRY CON ELEMENTOS ESPECIFICOS DE X Y FXY.
C OME SON FUNCION EXCLUSIVAMENTE DE CIERTOS ELEMENTOS CONSECUTIVOS DE Y. POR
C ULTIMO LLAMA A DIRY PARA QUE CALCULE EL VALOR INTERPOLADO, HACIENDO A Y
C VARIABLE INDEPENDIENTE Y A FY VARIABLE DEPENDIENTE.
C
0002      IMPLICIT REAL*8 (A-H,O-Z)
0003      DIMENSION X(M),Y(N),FXY(50,50),FXT(50)
C
C CONTROL DE CONSISTENCIA DE NGRAD
C
0004      IF(NGRAD.GE.N.ON.NH*0,FC,M) GO TO 3
C
C BUSQUEDA DE LOS NGRAD + 1 ELEMENTOS DE X QUE INTERVIENEN EN EL CALCULO
C
0005      MINN=MINON(M,X,ARGX,NGRAD)
0006      MAXM=MINN+NGRAD

```

FUNCION DIRXY: (Continuación).

```

C
C BUSQUEDA DE LOS NGRAD + 1 ELEMENTOS DE Y QUE INTERVIENEN EN EL CALCULO
0007      MINN=MINMON (N,Y,ARGY,NGRAD)
0008      MAXN=MINN+NGRAD
C
C CALCULO DE LOS NGRAD + 1 VALORES QUE SON FUNCION UNICAMENTE DE LOS ELEMENTOS
C SELECCIONADOS DE Y
0009      DO 1 J=MINN,MAXN
0010      DO 2 I=MINN,MAXN
0011      Z FX(I)=FX(I,J)
0012      I FX(J)=DIRY (MAXN,MINN,X,FXI,ARGX,M)
C
C CALCULO DEL VALOR FINAL INTERPOLADO
0013      DIRXY=DIRY (MAXN,MINN,Y,FY,ARGY,N)
0014      IND=0
0015      RETURN
0016      Z IND=1
0017      DIRXY=0.00
0018      RETURN
0019      END
    
```

FUNCION MINMON:

```

0001      FUNCION MINMON (MON,XCY,ARGXCY,N)
C
C SUBPROGRAMA PARA DETERMINAR EL PRIMERO DE LOS PUNTOS BASES DE CUALQUIERA DE
C LAS VARIABLES INDEPENDIENTES, QUE INTERVIENEN EN LA INTERPOLACION POLINOMIAL
C ESCORRIDA EN LOS PUNTOS BASES PARA SABER EL SUBINDICE I DEL ELEMENTO
C INMEDIATO INFERIOR A ARGXCY. BASANDOSE EN ESTE SUBINDICE CALCULA EL
C PRIMER ELEMENTO DE X QUE DEBE INTERVENIR EN LA INTERPOLACION.
C
0002      REAL*8 XCY
0003      DIMENSION XOY(MON)
0004      I=0
C
C BUSQUEDA EN EL ARREGLO XOY DEL ELEMENTO .GE. ARGXCY
0005      DO 1 K=1,MON
0006      I=I+1
0007      IF (ARGXCY.LE.XOY(K).OR.K.EQ.MON) GO TO 2
0008      1 CONTINUE
C
C CALCULO DEL SUBINDICE DEL PRIMER ELEMENTO DE LOS PUNTOS QUE INTERVIENEN
C EN LA INTERPOLACION.
0009      2 MINMON=I-N/2
0010      IF (MINMON.LE.1) MINMON=1
0011      IF (MINMON.GE.1) RETURN
0012      MAX=I+N/2
0013      IF (MAX.GT.MON) MAX=MON
0014      MINMON=MAX-N
0015      RETURN
0016      END
    
```

FUNCION DIRY:

```

0001      FUNCION DIRY (MAX,MIN,Y,FX,ARGY,N)
C
C SUBPROGRAMA PARA INTERPOLACION EN DOS DIMENSIONES UTILIZANDO EL POLINOMIO DE
C COLOCACION DE LAGRANGE.
C EL POLINOMIO A EVALUAR DE GRADO NGRAD PARA EL ARGUMENTO ARGY
C USA LOS DATOS Y(MIN) ... Y(MAX) Y FX(MIN) ... FX(MAX), DONDE
C NGR = Y(N) + NGRAD. SE ASUME QUE LOS VALORES DE Y ESTAN EN ORDEN ASCENDENTE,
C Y ES UNA VARIABLE QUE CONTIENE SUCESIVAMENTE CADA TERMINO DE LA
C FORMULA DE LAGRANGE. EL VALOR FINAL DE DIRY ES EL VALOR
C INTERPOLADO. PARA UNA DESCRIPCION DE LA VARIABLE C REFERIRSE A LA
C INFORMACION TECNICA.
C
0002      IMPLICIT REAL*8 (A-F,O-Z)
0003      DIMENSION Y(N),FX(N)
    
```

FUNCION DIRY: (Continuación).

```
C
C  CALCULO DEL VALOR C
C
0004      C=1.00
0005      JJ=MIN-1
0006      DO 1 J=MIN,MAX
0007      JJ=JJ+1
0008      IF(ARGY.EG.Y(JJ)) GO TO 4
0009      C=C*(ARGY-Y(JJ))
C
C  EVALUACION DEL POLINOMIO DE COLECCION
C
0010      FXY=C.00
0011      DO 3 I=MIN,MAX
0012      I=C*FY(I)/(ARGY-Y(I))
0013      DO 2 J=MIN,MAX
0014      IF(I.EG.J) GO TO 2
0015      I=I*(Y(I)-Y(J))
0016      2 CONTINUE
0017      3 FXY=FXY+I
0018      DIRY=FXY
0019      RETURN
0020      4 DIRY=FX(JJ)
0021      RETURN
0022      END
```

ENTRADA DE DATOS:

Los datos deben ser perforados de la siguiente manera:

- 1.- En la primera tarjeta, el valor de M (número de muestras de la variable x) en las dos primeras columnas; en las dos siguientes columnas el valor de N (número de muestras de la variable y). M y N -variables enteras- deben perforarse de tal manera que la última columna de su campo corresponda a las unidades de dichos números. A partir de la columna 11 perforar siete valores de x, abarcando cada valor 10 columnas, en una de las cuales se debe perforar el punto decimal dependiendo del valor a perforarse, sin importar en cual de ellas. (2I2,6X7F10.0).

- 2.- Desde la segunda tarjeta, perforar 8 valores de x por

tarjeta hasta terminar con todos los valores de x ; de tal manera que cada valor ocupe un campo de 10 columnas incluido el punto decimal (8F10.0).

- 3.- A continuación de los valores de x perforar los valores de y , como lo indica el numeral 2, cambiando de tarjeta cuando sea necesario; es decir, solamente cuando se hayan perforado 8 valores por tarjeta (cambio obligado de tarjeta).
- 4.- En una nueva tarjeta se deberá perforar: El grado de los polinomios, NGRAD, -variable entera-, en las columnas 1 y 2, debiendo ubicar las unidades de NGRAD en la columna 2. Entre las columnas 11 a 20 perforar el valor de ARGX incluyendo el punto decimal; en las columnas 21 a 30 el valor de ARGY incluyendo también el punto decimal. Siendo ARGX y ARGY los valores de la variable \bar{x} y \bar{y} para los cuales se desea conocer el valor de la función, $\bar{x} + \text{Sen } \bar{x} \cdot \text{Cos } \bar{y}$, mediante interpolación. (I2,8X2F10.0).
- 5.- Si se desea obtener una aproximación para otros tres valores de NGRAD, ARGX, ARGY, el cálculo se repite desde el numeral 4.

El valor máximo de M y N es 50. NGRAD siempre debe ser menor que estos valores.

RESULTADOS.

VALORES UTILIZADOS PARA LA INTERPOLACION EN TRES DIMENSIONES
DE LA FUNCION $F(X, Y, Z) = X + SEN(X) + COS(Y)$

VARIABLE X	VARIABLE Y	0.10472	0.40143	0.90484	1.57080	2.51327	3.14159	3.31225
0.01745	0.0348035259	0.0335115654	0.0209534348	0.0174593552	0.0033339122	0.0000000000	0.0000000000	0.0000000000
0.10472	0.2086760849	0.2009390075	0.1616502101	0.1097156120	0.0201447521	0.0001912542	0.0001713252	0.0001713252
0.20944	0.4162131929	0.4008230144	0.3220766779	0.2094392163	0.0412260251	0.0012276221	0.0004221620	0.0004221620
0.40143	0.7900245634	0.7611020078	0.6142188042	0.4014226547	0.0985194421	0.0100040591	0.0000000000	0.0000000000
0.50615	0.9983076990	0.9524224357	0.7701974095	0.5061432152	0.1135287574	0.0213364492	0.0028350022	0.0028350022
0.80215	1.5170643840	1.4438850723	1.1936636483	0.8021473555	0.2200879732	0.0352974442	0.0053724614	0.0053724614
0.99408	1.8949174014	1.7688400046	1.4315117826	0.9949369194	0.3100922652	0.0426161440	0.0162752449	0.0162752449
1.41372	2.3959981696	2.2228507386	1.9516519793	1.4137163720	0.6146003225	0.0426031142	0.0404122272	0.0404122272
1.57080	2.5443218698	2.4913031802	2.1154270890	1.5707963268	0.7017854250	0.0570000000	0.0583400000	0.0583400000
1.81514	2.7801200027	2.7089000050	2.3435993541	1.8151364355	1.0301521432	0.0944430000	0.0944430000	0.0944430000
2.09440	2.9556787684	2.8915766847	2.5660082168	2.0944081189	1.3537748100	1.2222770000	1.2405400000	1.2405400000
2.51327	3.0927386053	3.0543312667	2.8334014824	2.5132078409	2.0377404678	1.9224214122	1.9300255000	1.9300255000
2.80998	3.1337647357	3.1036660062	2.9829295536	2.8099708041	2.5406500000	2.4644112000	2.4445122221	2.4445122221
2.91470	3.1384185986	3.11717680280	3.0372166071	2.9146991737	2.7327314364	2.4697950000	2.4697950000	2.4697950000
3.14159	3.1415926536	3.1415926536	3.1415914932	3.1415900000	3.1415900000	3.1415900000	3.1415900000	3.1415900000

RESULTADOS (Continuación)

ARGUMENTO X	ARGUMENTO Y	GRADO	VALOR INTERPOLADO	VALOR VERDADERO
0.06321	0.25364	1	0.1236290256	0.1243566670
0.06321	0.25364	2	0.1243293461	0.1243566670
0.06321	0.25364	3	0.1244005205	0.1243566670
0.06321	0.25364	4	0.1243765858	0.1243566670
0.06321	0.25364	5	0.1243535975	0.1243566670
0.06321	0.25364	6	0.1243519502	0.1243566670
1.15413	1.26782	1	1.4106456805	1.4225662658
1.15413	1.26782	2	1.41200224050	1.4225662658
1.15413	1.26782	3	1.4201752952	1.4225662658
1.15413	1.26782	4	1.4257725365	1.4225662658
1.15413	1.26782	5	1.4269305201	1.4225662658
1.15413	1.26782	6	1.4268966025	1.4225662658
3.00000	3.21456	1	2.8604520425	2.8552555032
3.00000	3.21456	2	2.8582415910	2.8552555032
3.00000	3.21456	3	2.8592001210	2.8552555032
3.00000	3.21456	4	2.8592503970	2.8552555032
3.00000	3.21456	5	2.8592615721	2.8552555032
3.00000	3.21456	6	2.8592591242	2.8552555032
2.12597	0.25837	1	2.9454215917	2.945268797
2.12597	0.25837	2	2.9474072222	2.945268797
2.12597	0.25837	3	2.9500135652	2.945268797
2.12597	0.25837	4	2.9457750705	2.945268797
2.12597	0.25837	5	2.9494323482	2.945268797
2.12597	0.25837	6	2.9494614659	2.945268797
2.66970	1.35699	1	2.7605663555	2.7661517723
2.66970	1.35699	2	2.7594216527	2.7661517723
2.66970	1.35699	3	2.7658222172	2.7661517723
2.66970	1.35699	4	2.7656071918	2.7661517723
2.66970	1.35699	5	2.7661366651	2.7661517723
0.57596	0.80354	1	0.9213390210	0.9214855281
0.57596	0.80354	3	0.9210227662	0.9214855281
0.57596	0.80354	6	0.9215156845	0.9214855281
0.20544	1.57060	1	0.2094352303	0.2094352303
4.50751	3.76543	4	5.1601580312	5.3021776243

Los resultados obtenidos son satisfactorios; se puede observar que a partir del cuarto grado, los valores interpolados coinciden con los valores reales. También se nota que no siempre un grado mayor para los polinomios de colocación de Lagrange proporciona una mejor aproximación.

Si bien es cierto que el polinomio de colocación de Lagrange presenta facilidades para interpolación inversa; pero las dos interpolaciones -directa e inversa- necesitan conocer de antemano el grado del polinomio; de ahí la desventaja de este método. Si bien es cierto en el polinomio de colocación por diferencias finitas de Newton aparentemente se topa con el mismo problema, pero no es así, pues se puede obtener el grado más conveniente del polinomio, para el conjunto de datos, observando las variaciones de las diferencias finitas del mismo orden.

2.6.- POLINOMIOS DE COLOCACION SPLINE CUBICOS.

Dada una función $f(x)$, continua y diferenciable en el intervalo (a,b) , se desea aproximarla mediante un modelo por partes usando polinomios de colocación de bajo grado que actúan sobre subintervalos de (a,b) . Conocidos $n+1$ valores funcionales $y_i = f(x_i)$; $i = 0,1,2,\dots, n$, se quiere encontrar una función aproximante que sea continua en el intervalo (a,b) , así como también su primera y segunda derivada, para todo valor de x dentro del intervalo; además debe satisfacer $n+1$ veces el criterio (2.2). Asumiendo que tal función aproximante coincide con un polinomio de tercer grado $P_{3,i}(x_i)$ en cada subintervalo (x_i, x_{i+1}) ; $i=0,1,\dots,n-1$

(n subintervalos) esto requiere que los puntos bases sean de la forma: $a = x_0 < x_1, \dots, < x_{n-1} < x_n = b$. Así pues cada una de las n segundas derivadas $P''_{3,i}(x)$; $i = 0, 1, \dots, n-1$, puede ser escrita como una combinación lineal de $P''_{3,i}(x_i)$ y $P''_{3,i}(x_{i+1})$, ya que $P''_{3,i}(x)$ son polinomios de primer grado: (si el lector está interesado en el análisis teórico de este subcapítulo referirse al Apéndice 4).

$$P''_{3,i}(x) = \frac{(x_{i+1}-x)}{h_i} \cdot P''_{3,i}(x_i) + \frac{(x-x_i)}{h_i} \cdot P''_{3,i}(x_{i+1}) \quad (2.18)$$

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

Integrando dos veces (2.18) y satisfaciendo el criterio (2.2)

$$\left. \begin{aligned} P_{3,i}(x_i) &= y_i \\ P_{3,i}(x_{i+1}) &= y_{i+1} \end{aligned} \right\} i = 0, 1, \dots, n-1$$

se llega al polinomio de colocación spline cúbico que se expresa como:

$$P_{3,i}(x) = \frac{P''_{3,i}(x_i)}{6h_i} (x_{i+1}-x)^3 + \frac{P''_{3,i}(x_{i+1})}{6h_i} (x-x_i)^3 +$$

$$\begin{aligned}
 & + \left[\frac{y_{i+1}}{h_i} - \frac{h_i \cdot P''_{3,i}(x_{i+1})}{6} \right] \cdot (x - x_i) + \\
 & + \left[\frac{y_i}{h_i} - \frac{h_i \cdot P''_{3,i}(x_i)}{6} \right] \cdot (x_{i+1} - x) \quad (2.19)
 \end{aligned}$$

$$i = 0, 1, \dots, n-1$$

Observando (2.19), lo que se desconoce son los valores de las segundas derivadas $P''_{3,i}(x_{i+1})$ y $P''_{3,i}(x_i)$; con este fin se deriva (2.19) una sola vez y satisfaciendo el criterio de continuidad para esta derivada y para la segunda

$$\left. \begin{aligned}
 P'_{3,i}(x_i) &= P'_{3,i-1}(x_i) \\
 P''_{3,i}(x_i) &= P''_{3,i-1}(x_i)
 \end{aligned} \right\} ; i = 1, 2, \dots, n-1$$

se llega a un sistema de $n-1$ ecuaciones lineales simultáneas de la forma

$$\begin{aligned}
 & \frac{h_{i-1}}{h_i} \cdot P''_{3,i}(x_{i-1}) + \frac{2(h_i - h_{i-1})}{h_i} \cdot P''_{3,i}(x_i) + P''_{3,i}(x_{i+1}) = \\
 & = \frac{6}{h_i} \cdot \left[\frac{(y_{i+1} - y_i)}{h_i} - \frac{(y_i - y_{i-1})}{h_{i-1}} \right] \quad (2.20)
 \end{aligned}$$

$$i = 1, 2, \dots, n-1$$

con $n+1$ segundas derivadas desconocidas. $P''_{3,i}(x_i)$;
 $i = 0, 1, \dots, n$. Si se asume dos condiciones, como por
ejemplo que $P''_{3,0}(x_0) = 0$ y $P''_{3,n}(x_n) = 0$, entonces el
sistema de $n-1$ ecuaciones simultáneas lineales puede
ser resuelto; es decir se puede encontrar los valores
de las otras $n-1$ segundas derivadas. Según (1), al a-
sumir las dos condiciones ya citadas, el polinomio
spline cúbico generado posee la propiedad de CURVATURA
CUADRÁTICA MINIMA MEDIA; o sea que para todas las fun-
ciones interpolantes, $g(x)$, derivables dos veces se
cumple que

$$\int_a^b P''_{3,i}(x)^2 dx \leq \int_a^b g''(x)^2 dx$$

Una vez conocidas las $n+1$ segundas derivadas de los
 $P_{3,i}(x)$, (2.19) puede ser evaluada para todo valor de
 x contenido en el intervalo (a,b) .

Escribiendo en forma matricial el sistema de ecuacio-
nes lineales simultáneas citado y asumiendo las
dos condiciones se tiene:

$$\begin{bmatrix} b_1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_2 & b_2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & 1 & \dots & 0 & 0 & 0 \\ \vdots & & & & & & & \\ 0 & 0 & 0 & 0 & \dots & a_{n-3} & b_{n-2} & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & a_{n-2} & b_{n-1} \end{bmatrix}$$

$$\begin{bmatrix} P''_{3,1}(x_1) \\ P''_{3,1}(x_2) \\ P''_{3,2}(x_3) \\ \vdots \\ P''_{3,n-3}(x_{n-2}) \\ P''_{3,n-2}(x_{n-1}) \end{bmatrix} = \begin{bmatrix} c_1 - a_1 \cdot P''_{3,0}(x_0) \\ c_2 \\ c_3 \\ \vdots \\ c_{n-2} \\ c_{n-1} - a_{n-1} \cdot P''_{3,n-1}(x_n) \end{bmatrix}$$

donde

$$a_i = \frac{h_{i-1}}{h_i} \qquad b_i = \frac{2(h_i + h_{i-1})}{h_{i-1}}$$

$$c_i = \frac{6}{h_i} \cdot \left(\frac{(y_{i+1} - y_i)}{h_i} - \frac{(y_i - y_{i-1})}{h_{i-1}} \right)$$

$$i = 1, 2, \dots, n-2, n-1$$

(2.21)

El polinomio de colocación spline cúbico frente a los polinomios de colocación de Newton y de Lagrange ya expuestos, goza de algunas ventajas, las mismas que son:

1.- La evaluación de un polinomio de colocación se vuelve costosa y poco confiable cuando el grado del polinomio es alto. Cabe anotar que para obtener menor error de aproximación es saludable tener mayor número de muestras de una función $f(x)$ dentro de un intervalo dado; pero esto involucra al aumento del grado del polinomio. Una solución a esto, sin disminuir el número de muestras, es utilizar la técnica spline cúbica. Si bien es cierto, en los dos programas fortran mencionados ya, se puede utilizar polinomios de bajo grado en los que intervienen un número determinado de muestras que depende del grado del polinomio y del valor de la variable independiente para el cual se desea conocer el valor de la función mediante interpolación; pero esto equivale a tomar menos número de muestras, lo que está en desacuerdo con lo citado en este numeral respecto a la obtención del menor error.

2.- Con el objeto de garantizar un error de aproximación pequeño, aún para un número pequeño de muestras,

tras, se hace pequeño el intervalo sobre el cual se va a aproximar $f(x)$. Tomando en cuenta que es este intervalo, en general, es dado de antemano se opta por dividirlo en subintervalos suficientemente pequeños y aproximando $f(x)$ en cada subintervalo por medio de un polinomio adecuado. Esto es un punto a favor del polinomio de colocación spline cúbico que lleva a su utilización.

3.- La utilización del polinomio spline cúbico no implica el conocimiento anticipado del grado del polinomio, lo que es necesario para el caso del polinomio de Newton y de Lagrange. El escogitamiento del grado del polinomio más adecuado es conflictivo, pues hay que tener en cuenta que no siempre un grado más alto da una mejor aproximación; también influencia el tipo de la función, la cantidad de muestras disponibles de la misma, etc.

4.- El polinomio spline cúbico presenta una cierta facilidad para el caso de interpolación inversa.

Por las cualidades anotadas, el programa para interpolación en tres dimensiones tanto directa como inversa que se desarrollará enseguida, utiliza la técnica del polinomio de colocación spline cúbico y el método des

crito en el subcapítulo 2.5 , para interpolación tridimensional. De esta forma se logra el objetivo primordial del presente trabajo.

Los delineamientos generales del programa son:

- 1.- Lee un conjunto de datos literales que sirven como leyendas aclaratorias cuando se escribe los datos de entrada del programa. Estos datos literales son:
 - Título de la tabla que contiene las muestras tomadas de las tres variables (dos independientes y una dependiente).
 - Definición de las tres variables.
 - Unidades de las tres variables.

- 2.- Lee: El número de puntos bases, m y n , tomados de las variables independientes x y y respectivamente. Los puntos bases de dichas variables, que deben estar ordenados ascendentemente, y son almacenados en dos arreglos unidimensionales uno por cada variable.

- 3.- Lee las muestras de la variable dependiente z y las almacena en un arreglo bidimensional de m filas por n columnas; de tal manera que el valor z_{ij} corresponda a los puntos bases x_i y y_j .

- 4.- Llama a la subrutina ESCRIT, que es la encargada de escribir todos los datos de entrada en forma ordenada, conformando una tabla de valores.

- 5.- Lee tres datos, de los cuales uno es literal y los otros dos son numéricos. El dato literal sirve de información para saber cual de las tres variables va a calcularse, conociendo las otras dos variables que constituyen los datos numéricos.

- 6.- Llama a la subrutina ENITØ1 que calcula la variable desconocida, mediante interpolación spline cúbica; cuyo valor es función del conjunto de muestras disponibles de las variables x , y , z , y del valor de las dos variables para las que se desea conocer la tercera. Aquí se presentan tres casos:
 - a.- Son datos las dos variables independientes \bar{x} y \bar{y} , y la incognita, la variable dependiente \bar{z} (interpolación directa).

 - b.- Son datos la variable independiente \bar{x} , y la variable dependiente \bar{z} , y la incógnita la variable independiente \bar{y} . (interpolación inversa)

 - c.- Son datos la variable independiente \bar{y} , y la variable dependiente \bar{z} , y la incógnita, la variable independiente \bar{x} . (interpolación inversa)

7.- Escribe los tres valores de las variables, con sus respectivas unidades. Los valores de dos de estas variables son datos de entrada y el tercer valor es el calculado por la subrutina ENITØ1.

Para un nuevo trío de datos de entrada, se repite el proceso desde el numeral 5. Si se presenta alguna inconsistencia en la ejecución del programa, se utiliza dos señalizaciones, ind y ji ; cuando $ind = -1$ y $ji \neq 0$ se presenta inconsistencia con las muestras, en este caso debe suspenderse el programa, y cuando $ind = -1$ y $ji = 0$, existe inconsistencia unicamente para el trío de valores que son leídos y en este caso se va a leer otro conjunto de tres datos. Cuando todo es normal $ind = 0$ sin interesar el valor de ji .

El programa no permite extrapolar, en este caso a la variable que se calcula se le asigna el valor cero, y se escribe una leyenda que indica que se está trabajando fuera del intervalo, dentro del cual la variable de pendiente \bar{z} , es conocida en forma discreta.

Antes de llamar a la subrutina ENITØ1 por primera vez, se inicializa a una variable entera "num", con el valor cero para ir incrementando su valor en una unidad, por cada vez que se llama a la subrutina en mención.

Antes de explicar la subrutina ENITØ1, convendría referirse a las subrutinas que son utilizadas a su vez por ENITØ1, las mismas que se han desarrollado para interpolación en dos dimensiones y que gracias al método de interpolación tridimensional descrita en el subcapítulo 2.5 , que no es más en sí que una extensión de la interpolación bidimensional, permiten su utilización en la interpolación con tres variables.

SUBROUTINA ENITØ2.-

Calcula las segundas derivadas necesarias para la evaluación de los distintos polinomios de colocación spline cúbicos, con el siguiente proceso:

- 1.- Encuentra los valores de todos los coeficientes a_i , b_i , c_i , dados por (2.21) del sistema de ecuaciones lineales simultáneas. Debido a que en FORTRAN los subíndices de las variables subindicadas no pueden ser cero se hace un reordenamiento; de tal manera que si de toman "n" muestras tanto de la variable independiente "x" como de la variable dependiente "y" (interpolación en dos dimensiones) se tendrá:

$$a_i = \frac{h_{i-1}}{h_i}$$

$$b_i = \frac{2(h_i + h_{i-1})}{h_{i-1}}$$

$$c_i = \frac{6}{h_i} \cdot \left(\frac{(y_{i+1} - y_i)}{h_i} - \frac{(y_i - y_{i-1})}{h_{i-1}} \right)$$

$$i = 2, 3, \dots, n-1$$

Escribiendo b_i y c_i en función de a_i se tiene:

$$b_i = 2 \cdot (1 + a_i) ; \quad c_i = \frac{6}{h_i^2} \cdot \left((y_{i+1} - y_i) - \frac{(y_i - y_{i-1})}{a_i} \right)$$

con estas últimas expresiones son calculados los valores b_i y c_i . Los valores a_i y b_i , a excepción de a_1 y a_n , son almacenados en un arreglo bidimensional para ir conformando la matriz de coeficientes dada por (2.21). Los valores c_i se almacenan en un arreglo de una dimensión y se conforma la matriz de constantes, dada también por (2.21).

2.- Se llama a la subrutina ENITØ3 que es la encargada de resolver el sistema de $n-2$ ecuaciones simultáneas con $n-2$ incógnitas; siendo dichas incógnitas las $n-2$ segundas derivadas, $P''_{3,i}(x_i)$, $i = 2, 3, \dots, n-1$; pues como ya se mencionó antes, se asume $P''_{3,1}(x_1) = P''_{3,n-1}(x_n) = 0$. Las n segundas derivadas son almacenadas en un arreglo de una sola dimensión que constituye uno de los argumentos de la

subrutina ENITØ2.

ENITØ2 utiliza tres constantes, $\epsilon_1, \epsilon_2, \epsilon_3$. ϵ_1 se lo necesita en ENITØ3 y se lo mencionará al momento de referirse a dicha subrutina. ϵ_2 y ϵ_3 sirven para el caso en que los puntos bases consecutivos o muestras de la variable independiente, x_i y x_{i+1} , sean tan próximos (lo que se presenta en interpolación inversa) que $|h_i| = |x_{i+1} - x_i|$ sea un número tan pequeño próximo a cero y como para el cálculo de los valores a_i, b_i y c_i hay que dividir por h_i , de ahí que es necesario ir chequeando este valor; para lo cual su valor absoluto se lo compara con ϵ_3 . Si $|h_i| < \epsilon_3$ entonces a la muestra x_{i+1} se le aumenta un porcentaje igual a ϵ_2 . No existe una justificación rigurosa para esto; siendo la única, que tratándose de una aproximación toda la globalidad, no alterará en casi nada la aproximación hecha en una forma arbitraria. Los valores de ϵ_2 y ϵ_3 más adecuados se lo determinaron con ensayos; de las diferentes tentativas que se hicieron, se obtuvieron resultados satisfactorios con $\epsilon_2 = 0,01$ y $\epsilon_3 = 0,001$.

Existe una variable indicadora del buen desenvolvimiento del proceso de cálculo, ind , cuando el sistema de ecuaciones lineales simultáneas es irresoluble se le asigna a ind el valor -1 ; en caso contrario $ind = 0$.

SUBROUTINA ENITØ3.

ENITØ3 asume un sistema de ecuaciones simultáneas lineales de la forma:

$$\begin{bmatrix} a_{1,1} & 1 & 0 & \dots & 0 & 0 & 0 \\ a_{2,1} & a_{2,2} & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{n-1,n-3} & a_{n-1,n-2} & 1 \\ 0 & 0 & 0 & \dots & 0 & a_{n,n-2} & a_{n,n} \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

donde los a_{ij} , los b_i son datos y los x_i incógnitas. El método que se utiliza para resolver el sistema de ecuaciones lineales simultáneas es en base al proceso de eliminación de Gauss que consiste en ir eliminando incógnitas. Por no ser motivo de este trabajo, la

descripción del algoritmo usado para resolver el sistema de ecuaciones, no se lo hará.

El algoritmo de resolución en una de sus etapas de cálculo averigua si el sistema de ecuaciones es singular o no, mediante la comparación de los coeficientes de la diagonal principal de la matriz $[A]$ modificada con la constante ϵ_1 mencionada ya, en ENITØ2.

($|a_{ii}| < \epsilon_1; i = 1, 2, \dots, n$). El valor de $\epsilon_1 = 1 \cdot 10^{-15}$, se escogió de una forma un tanto arbitraria, pues el único punto de vista que se tomó en cuenta fue el hecho de estar trabajando en doble precisión. Si el sistema de ecuaciones simultáneas es singular (no existe solución) se le asigna a una variable indicadora, ind, el valor -1; en caso contrario ind = 0.

FUNCION ENITØ4.

ENITØ4 evalúa el polinomio de colocación spline cúbico dado por (2.19). Asume que los puntos bases están ordenados en forma ascendente y además que las segundas derivadas ya fueron calculadas por ENITØ2. Los pasos que sigue esta subrutina son:

1.- Chequea que si el valor de la variable independiente, \bar{x} , para el cual se desea evaluar el polinomio spline cúbico, $P_{3,i}(\bar{x})$, se encuentre dentro del in

tervalo de interpolación ($x_1 \leq \bar{x} \leq x_n$). Cuando \bar{x} está fuera de dicho intervalo, como no se puede extrapolar con esta técnica de aproximación polinomial, se le asigna directamente a ENITØ4 el valor cero.

- 2.- Se determina cual de los $n-1$ polinomios de colocación spline cúbicos debe utilizarse; siendo función esto de la ubicación de \bar{x} en el intervalo.
- 3.- Se evalúa el respectivo polinomio, asignando este valor a ENITØ4.

SUBROUTINA ENITØ5.-

ENITØ5 pone en orden ascendente los elementos de un arreglo. Como para cada elemento del arreglo que se va a ordenar le corresponde un elemento de otro arreglo; en el proceso de ordenamiento, para no alterar dicha correspondencia, hay que intercambiar también los elementos del otro arreglo.

SUBROUTINA ENITØ1.-

ENITØ1 asume que los puntos bases o muestras de las dos variables independientes, x y y , están ordenados de menor a mayor (orden ascendente) y que las muestras de la variable dependiente, z , están ordenadas de a

cuerdo a la siguiente tabla:

TABLA 2.5 ORDENAMIENTO DE LAS MUESTRAS PARA UTILIZAR LA SUBROUTINA ENIT01.

	y_1	y_2	. . .	y_n
--	-------	-------	-------	-------

x_1	z_{11}	z_{12}	. . .	z_{1n}
x_2	z_{21}	z_{22}	. . .	z_{2n}
.	.			.
.	.			.
.	.			.
x_m	z_{m1}	z_{m2}	. . .	z_{mn}

El modo de ordenamiento de las muestras nos permite efectuar la interpolación tridimensional como varias interpolaciones bidimensionales. Si se desea conocer para los valores \bar{x} y \bar{y} , el valor de la función $\bar{z}(\bar{x}, \bar{y})$; se encuentra un conjunto de $m-1$ polinomios spline cúbicos que son función de los puntos bases x_i ; $i = 1, 2, \dots, m$ y de la primera columna de las muestras z_{ij} ; $j = 1$, equivaliendo esto a una aproximación en dos dimensiones. De igual forma se obtiene otro conjunto de $m-1$ polinomios con las muestras de la segunda columna z_{i2} y así sucesivamente hasta obtener n conjuntos, cada

uno de $m-1$ polinomios de colocación spline cúbicos que son función de los dos conjuntos de puntos bases $x_i, i = 1, 2, \dots, m$ y $y_j, j = 1, 2, \dots, n$ y de las muestras z_{ij} . Note que estos conjuntos de polinomios se obtuvieron manteniendo "y" constante (por columnas); también se pudo obtener otros m conjuntos, cada uno de $n-1$ polinomios manteniendo "x" constante (por filas). ENITØ1 utiliza los m y los n conjuntos de polinomios ya que son necesarios los dos grupos para el caso de interpolación inversa. De lo expuesto, será necesario conocer $m \times n$ segundas derivadas con respecto a la variable x y $m \times n$ segundas derivadas con respecto a la variable y . Todo lo dicho equivale en la figura 2.3 a encontrar f_1, f_2, \dots, f_n valores calculados mediante interpolación spline cúbica bidimensional, tra

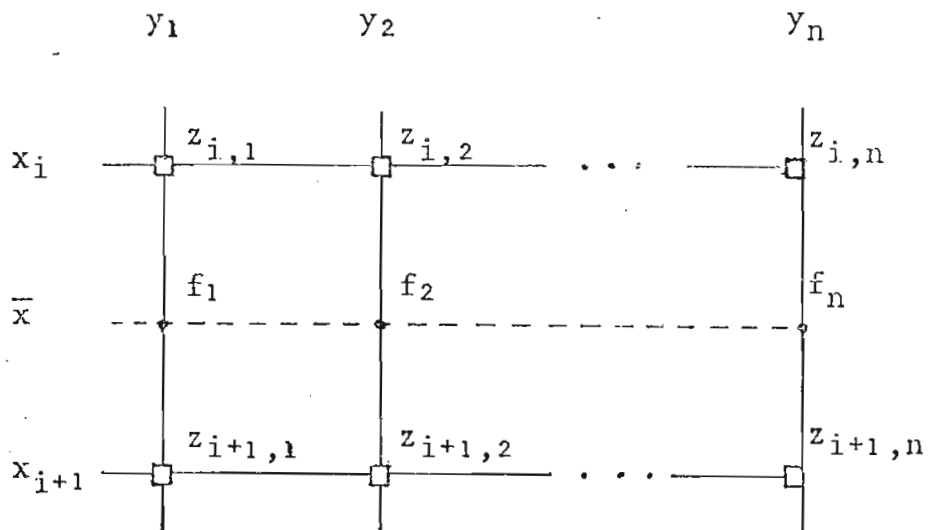


FIGURA 2.3 INTERPOLACION TRIDIMENSIONAL.

bajando con el valor \bar{x} , como puntos bases de la variable independiente los x_i ; $i = 1, 2, \dots, m$, y como muestras de la variable dependiente los z_{ij} de una determinada columna; así f_1 con la columna uno, f_2 con la columna dos y así sucesivamente hasta f_n con la columna enésima. Ahora bien f_j es solamente función de y_j ; $j = 1, 2, \dots, n$; entonces nuevamente estamos en el caso de interpolación bidimensional y toca encontrar un conjunto de $n-1$ polinomios spline cúbicos para evaluar el respectivo polinomio que es función de \bar{y} y determinar así el valor interpolado $\bar{z}(\bar{x}, \bar{y})$.

Quando se da como datos \bar{x} y \bar{z} y se quiere encontrar $\bar{y}(\bar{x}, \bar{z})$ se tiene interpolación inversa; en este caso el proceso es similar a interpolación directa hasta la determinación de f_1, f_2, \dots, f_n . De aquí en adelante como se conoce la variable dependiente \bar{z} y se desconoce la variable independiente \bar{y} , se intercambia los papeles es decir a los valores calculados f_1, f_2, \dots, f_n , luego de ser reordenados de menor a mayor, se les denomina puntos bases o muestras de la variable independiente y las respectivas muestras y_1, y_2, \dots, y_n pasan a constituir las muestras de la variable dependiente. Luego se encuentra un conjunto de $n-1$ polinomios spline cúbicos, para evaluar el correspondiente polinomio que es función de \bar{z} y determinándose de esta forma el

valor interpolado $\bar{y}(\bar{x}, \bar{z})$. Es necesario ordenar f_1, f_2, \dots, f_n , porque es requisito de la técnica del polinomio de colocación spline cúbico que los puntos bases estén en orden ascendente y como estos valores son calculados, en general están en desorden. También puede darse el caso que $f_j = f_{j+1}$; $j = 1, 2, \dots, n-1$. lo que ya está previsto por la subrutina ENITØ2 que es la que determina las segundas derivadas.

Cuando se conoce \bar{y} y \bar{z} y se quiere encontrar $\bar{x}(\bar{y}, \bar{z})$, se tiene también interpolación inversa y la forma de cálculo es similar al caso anterior, con la diferencia que hay que trabajar con los m conjuntos, cada uno de $n-1$ polinomios spline cúbicos, determinados por filas (x constante), para calcular m valores $(f'_1, f'_2, \dots, f'_m)$ que son función de los puntos bases x_i ; $i = 1, 2, \dots, m$, Después se procede con interpolación inversa bidimensional para determinar el valor interpolado $\bar{x}(\bar{y}, \bar{z})$.

El proceso de cálculo que sigue ENITØ1 es:

- 1.- Llama a la subrutina ENITØ2 n veces, una por cada columna de las muestras de la variable dependiente z , para calcular las $m \times n$ segundas derivadas con respecto a la variable independiente x ; permaneciendo la otra variable independiente, y , constante.

- 2.- Llama a la subrutina ENITØ2 m veces, una por cada fila de las muestras de la variable dependiente, z, para calcular las $m \times n$ segundas derivadas con respecto a y, permaneciendo x constante.
- 3.- Realiza el cálculo de los valores f_1, f_2, \dots, f_n o f_1, f_2, \dots, f'_m según sea el caso, llamando a la función ENITØ4 n o m veces respectivamente.
- 4.- Se llama a la subrutina ENITØ5 para ordenar los valores f_1, f_2, \dots, f_n o f'_1, f'_2, \dots, f'_m en forma ascendente y poder realizar la interpolación inversa. Este numeral no se lo realiza para interpolación directa.
- 5.- Llama a la subrutina ENITØ2 para determinar las n o m segundas derivadas. Los puntos bases para interpolación directa son los y_j ; $j = 1, 2, \dots, n$ y las muestras de la variable dependiente los f_j . Para interpolación inversa, en cambio, los puntos bases son los f_j ; $j = 1, 2, \dots, n$ o los f'_i ; $i = 1, 2, \dots, m$ y las muestras de la variable dependiente los y_j o los x_i respectivamente.
- 6.- Llama por última vez a ENITØ4 y se encuentra el valor interpolado, pudiendo ser este $\bar{z}(\bar{x}, \bar{y})$; $\bar{y}(\bar{x}, \bar{z})$ o $\bar{x}(\bar{y}, \bar{z})$.

Para saber cual de los tres casos de interpolación se tiene se utiliza una variable entera, ide. Los valores que se asigna a ide son de acuerdo al siguiente cuadro.

ide	datos	incognita	interpolación
1	\bar{x}, y	$\bar{z} (\bar{x}, \bar{y})$	directa
2	\bar{x}, \bar{z}	$\bar{y} (\bar{x}, \bar{z})$	inversa
3	\bar{y}, \bar{z}	$\bar{x} (\bar{y}, \bar{z})$	inversa

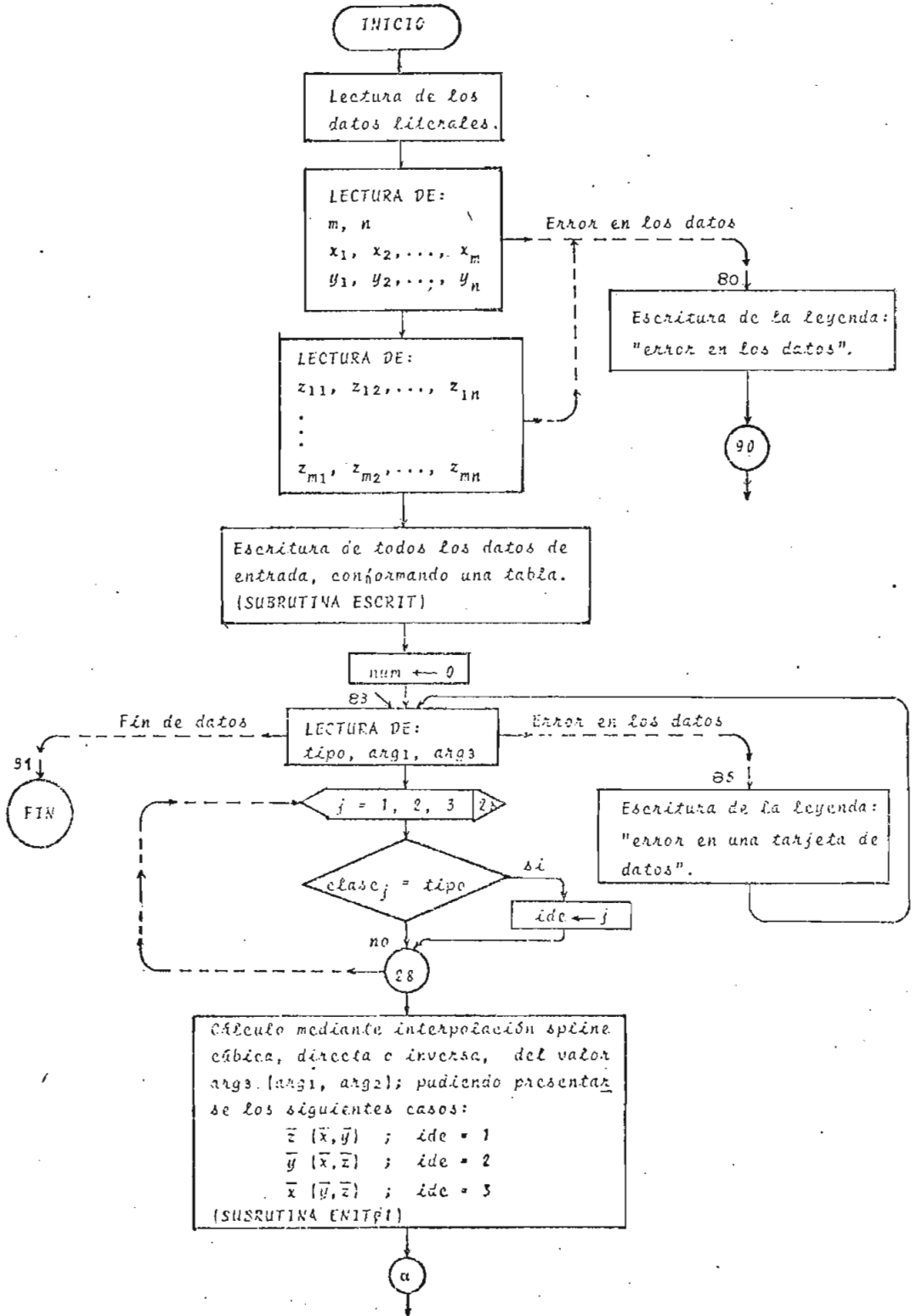
Si se llama por segunda vez o más a ENITØ1 los cálculos se realizan desde el numeral 3:

Se utiliza dos variables indicadoras, "ind" y "ji". Cuando hay algún problema en los numerales 2 y 3, ind = -1 y ji ≠ 0; en este caso la subrutina escribe una leyenda que indica que el programa que utiliza a ENITØ1 debe interrumpirse. Si se presenta problemas en el numeral 5 ind = -1 y ji = 0 se escribe el valor de "ind" y de "ji"; sin que sea imprescindible interrumpir el programa principal. Cuando todo es normal ind = 0, sin importar el valor de "ji".

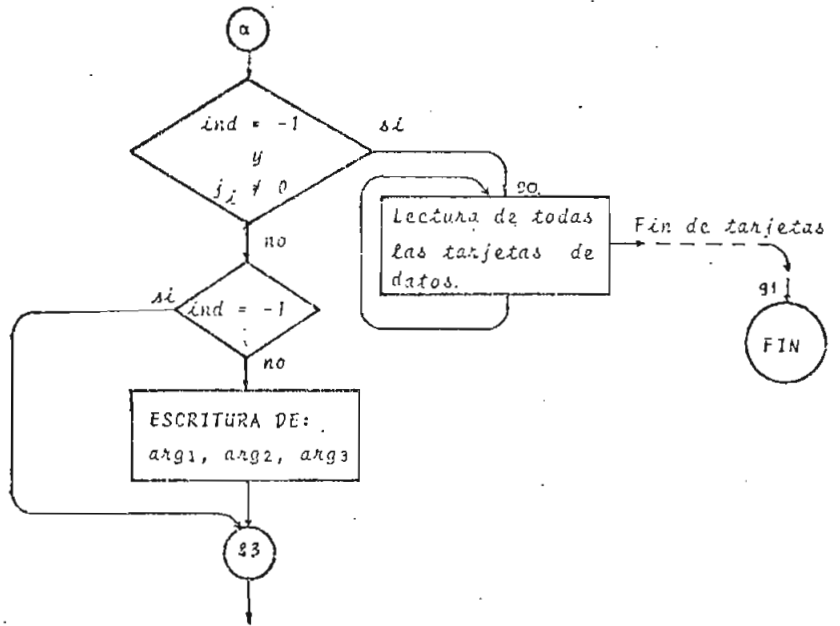
Cabe anotar que todos los subprogramas utilizados, a excepción de la subrutina ESCRIT, se hallan catalogados en la computadora de la Escuela Politécnica Nacional.

DIAGRAMAS DE FLUJO

PROGRAMA PRINCIPAL:

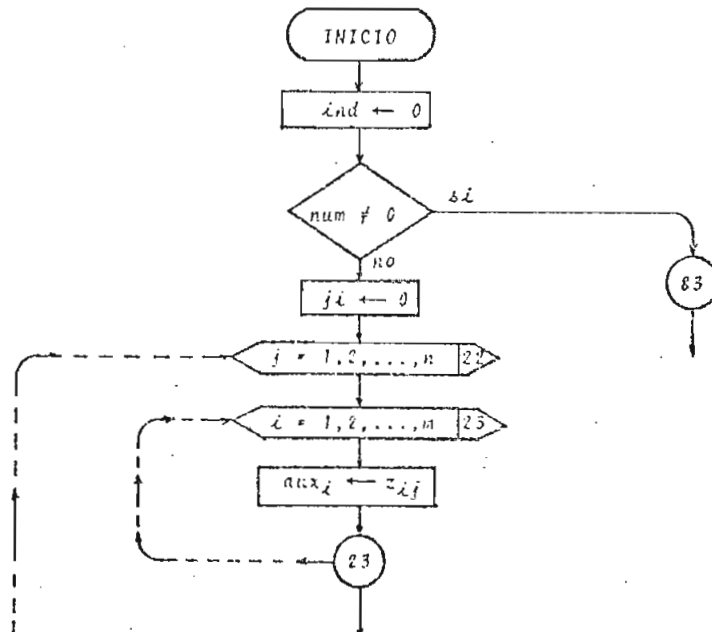


PROGRAMA PRINCIPAL: (Continuación).

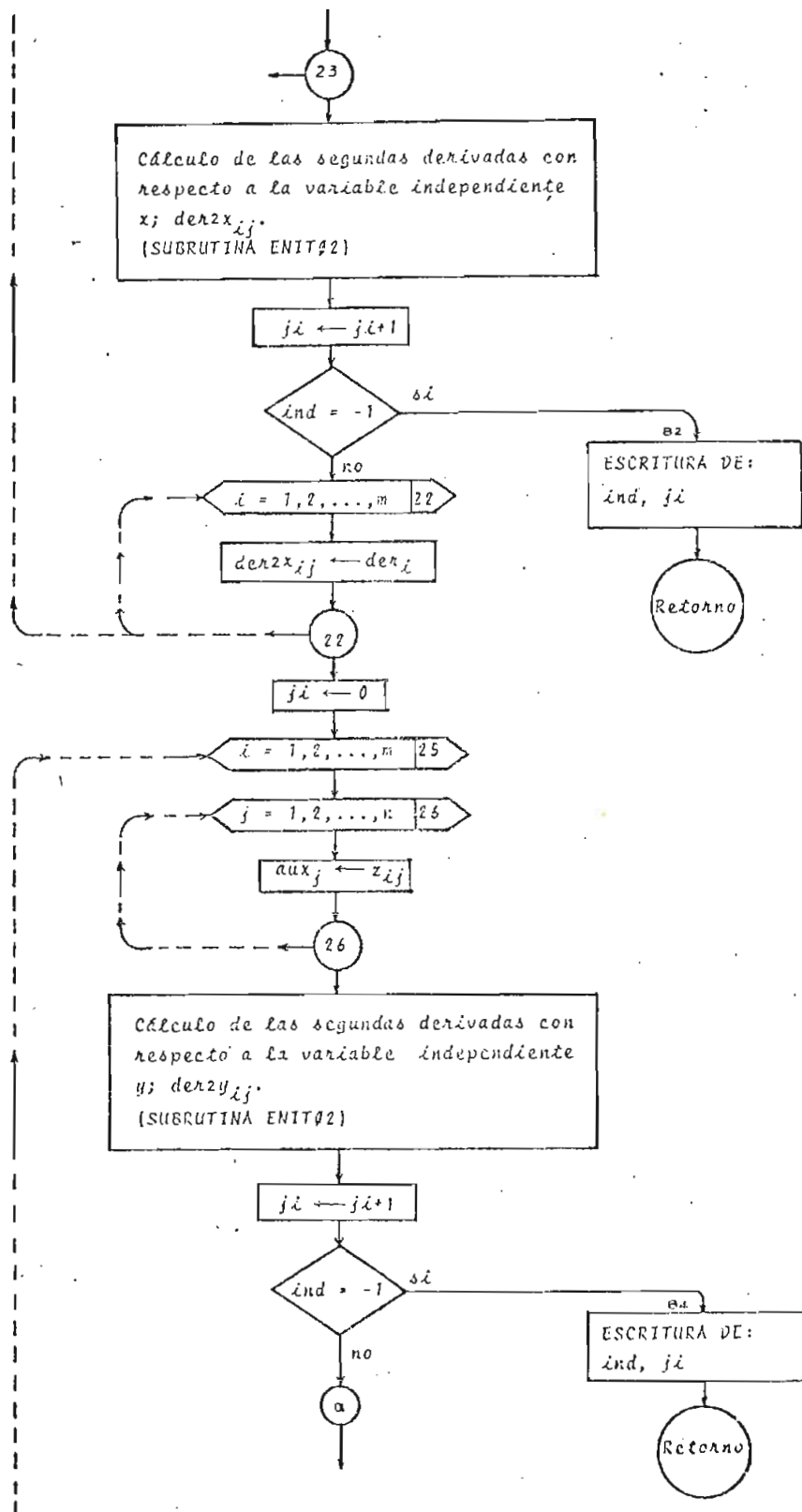


SUBROUTINA ENIT01:

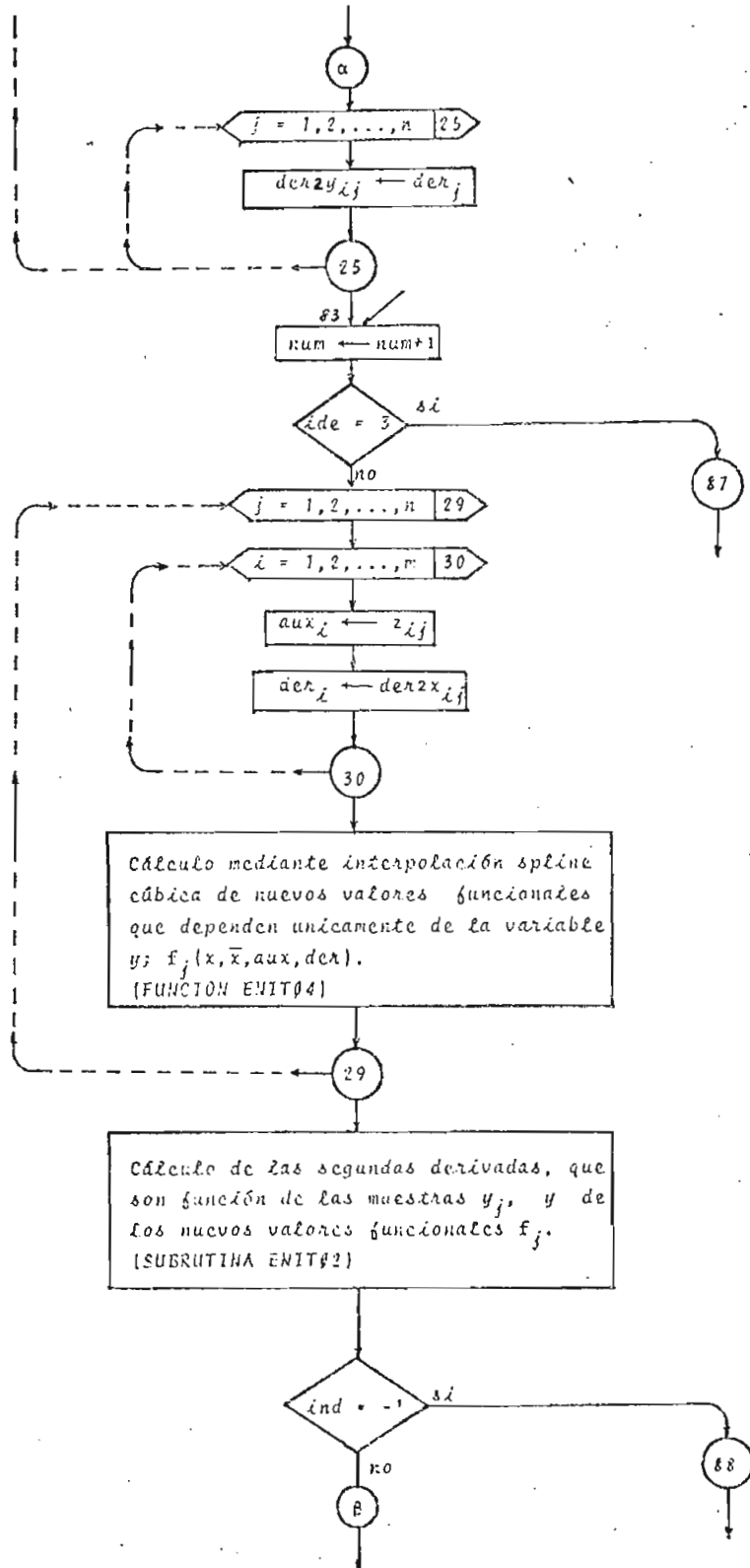
(Argumentos: $m, n, x, y, z, arg1, arg2, arg3, ide, num, ind, j_i, mm$).



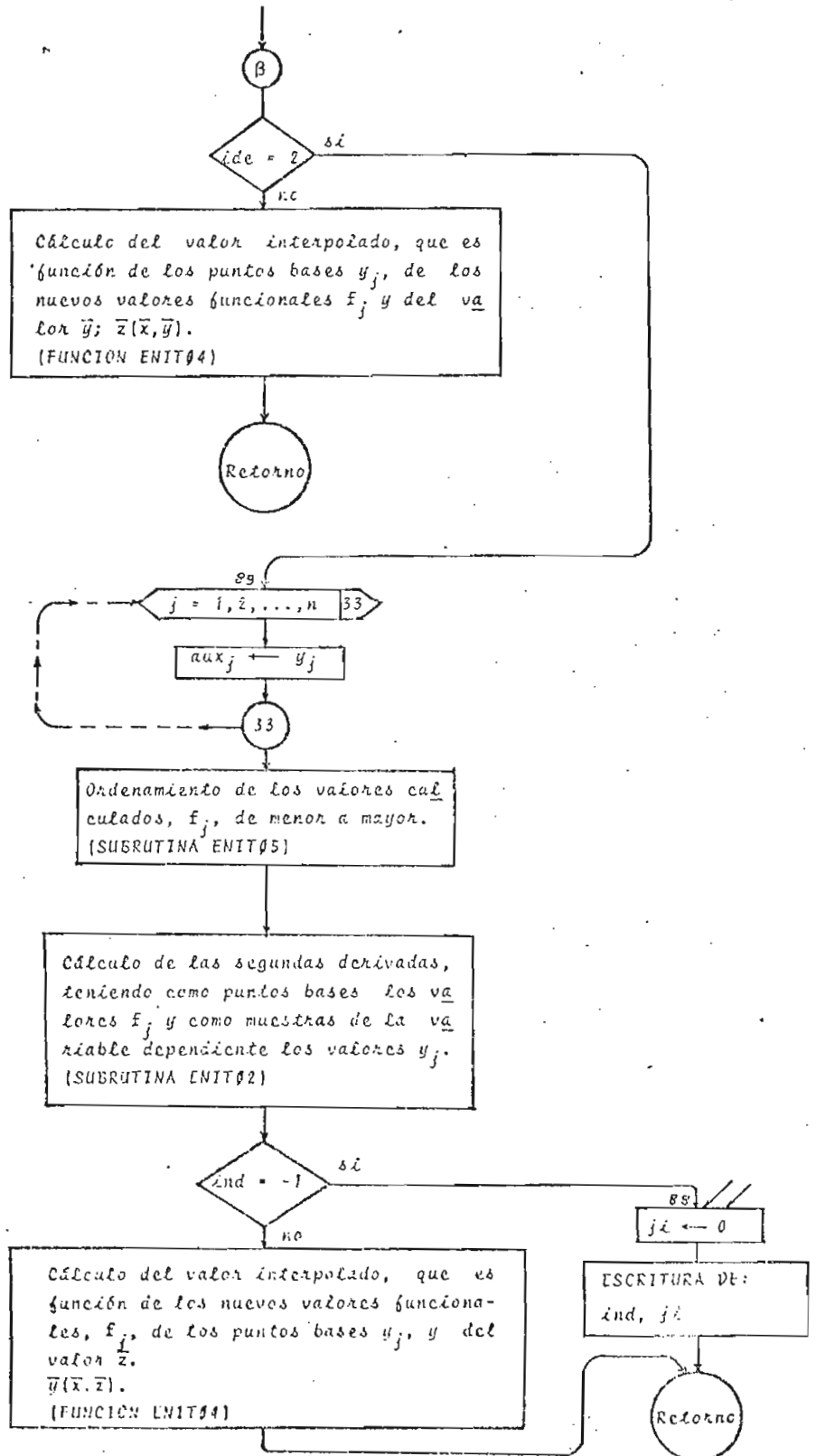
SUBROUTINA ENIT01: (Continuación).



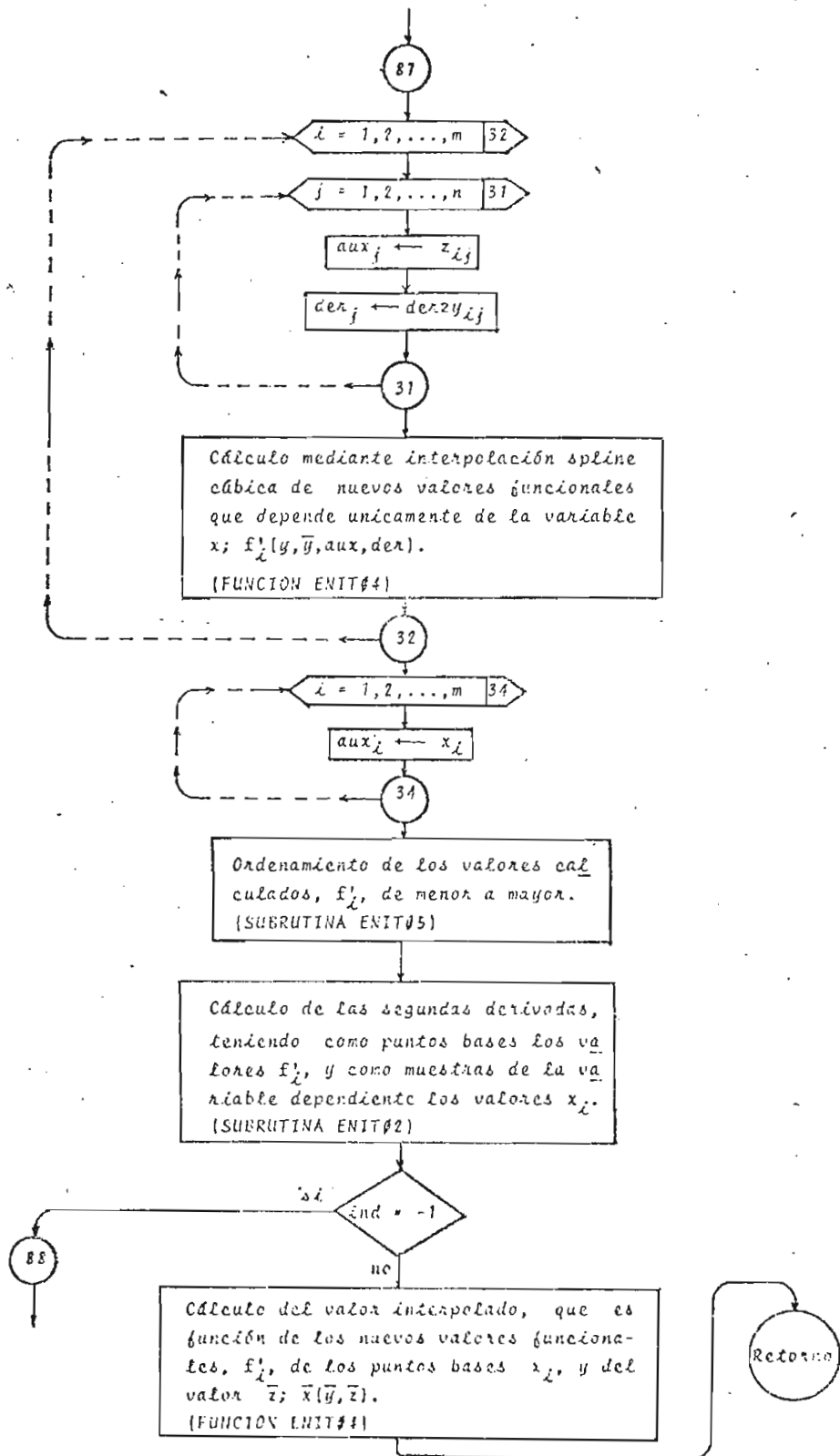
SUBROUTINA ENIT01: (Continuación).



SUBROUTINA ENIT01: (Continuación).

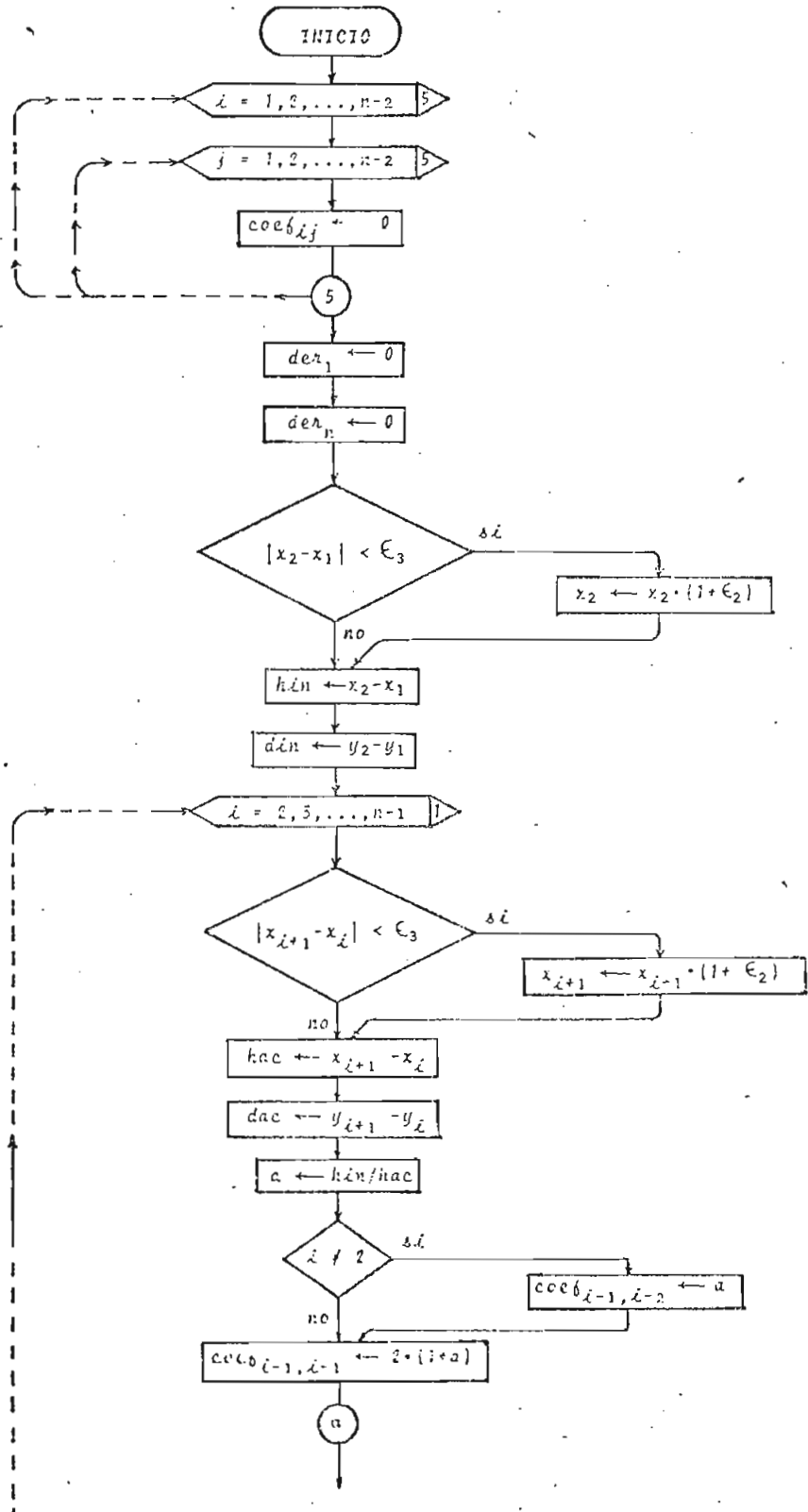


SUBROUTINA ENIT01: (Continuación).

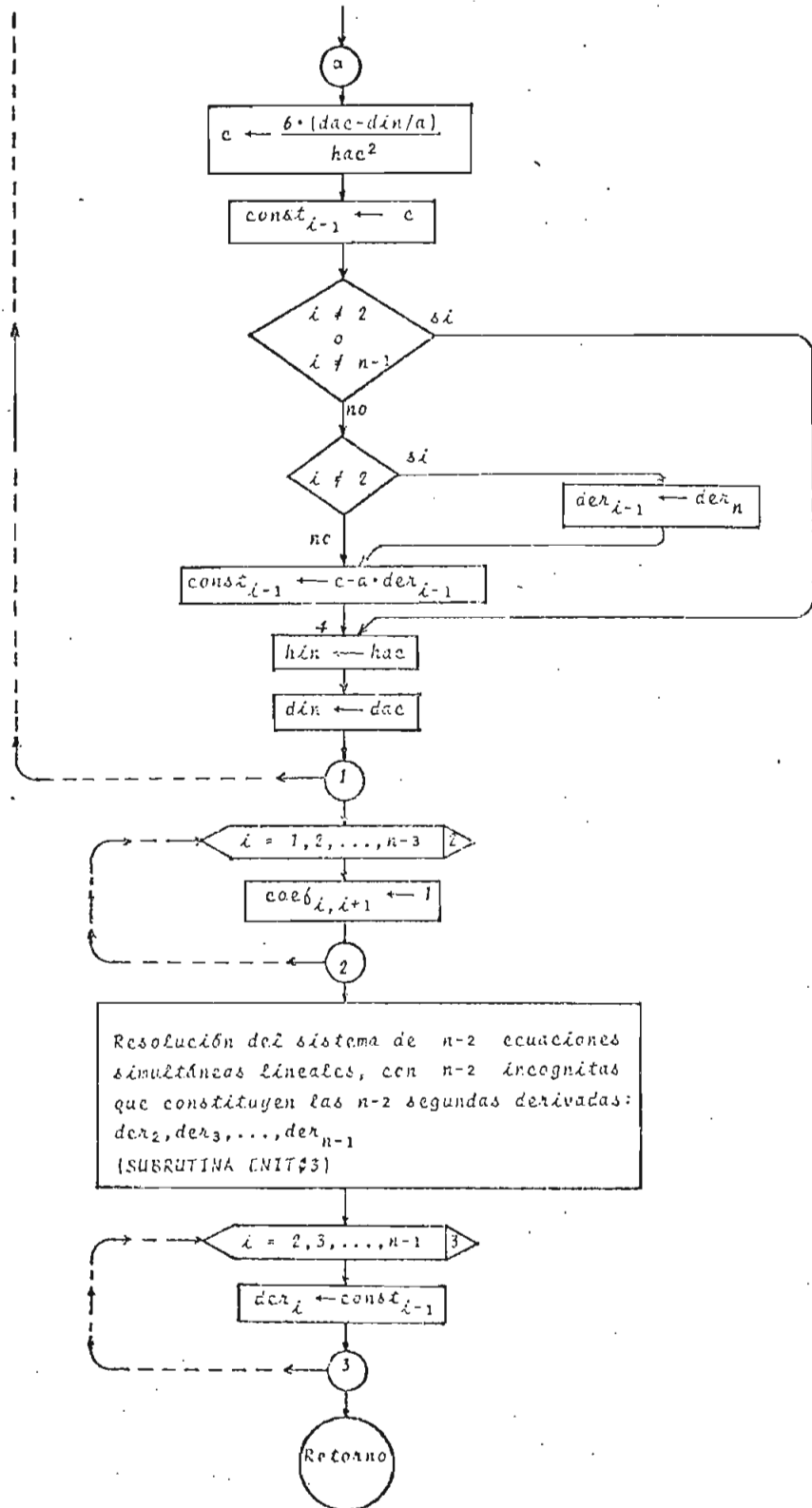


SUBROUTINA ENIT02:

{Argumentos: n, x, y, der, ind}.

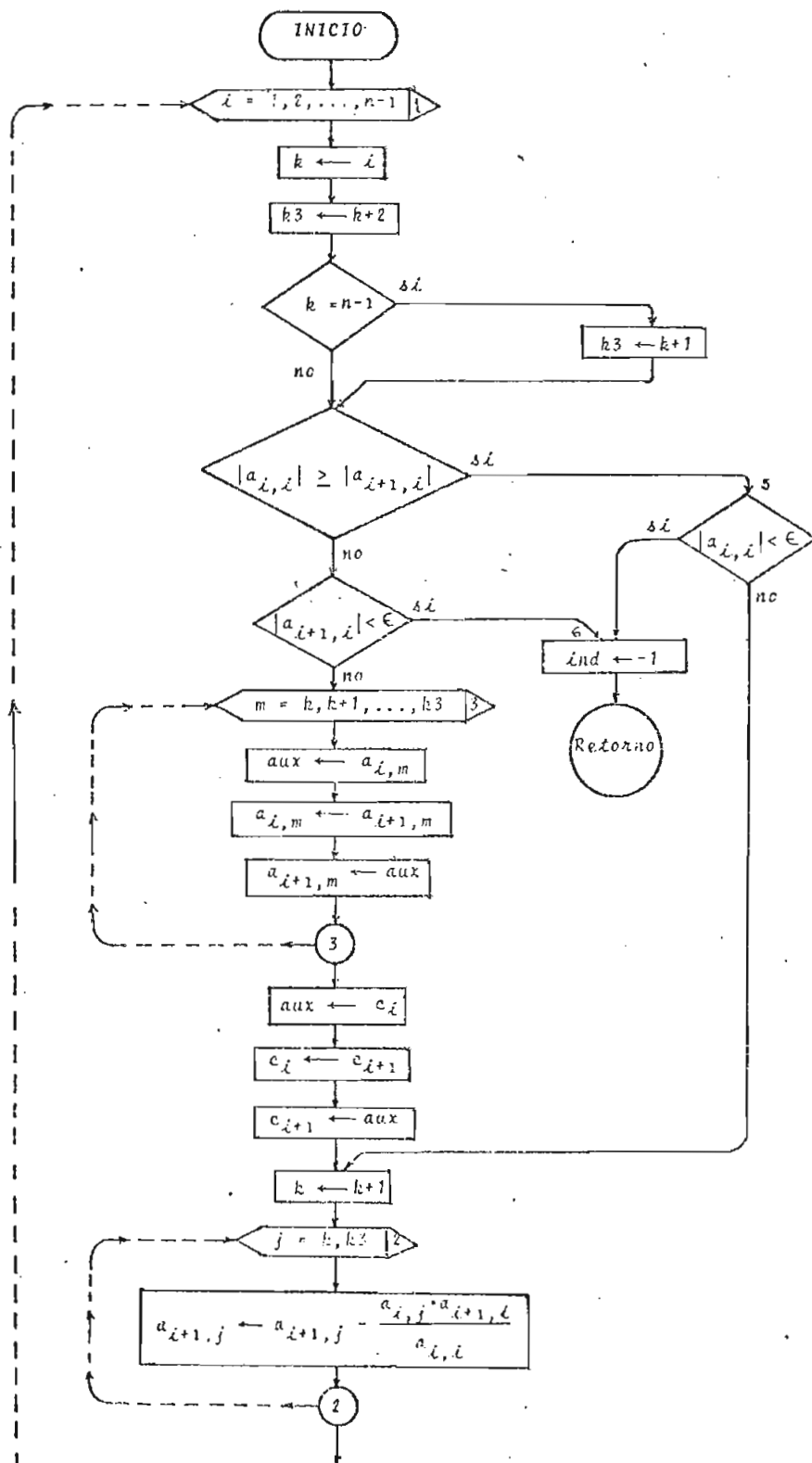


SUBROUTINA ENIT02: (Continuación).

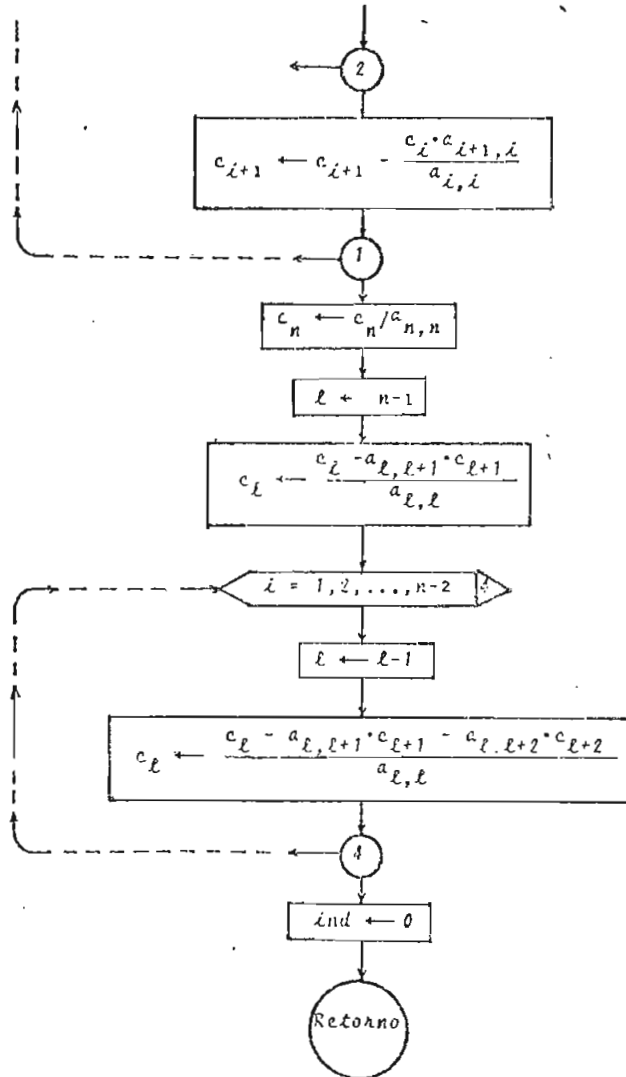


SUBROUTINA ENIT03:

(Argumentos: n, a, c, eps, ind, mm).

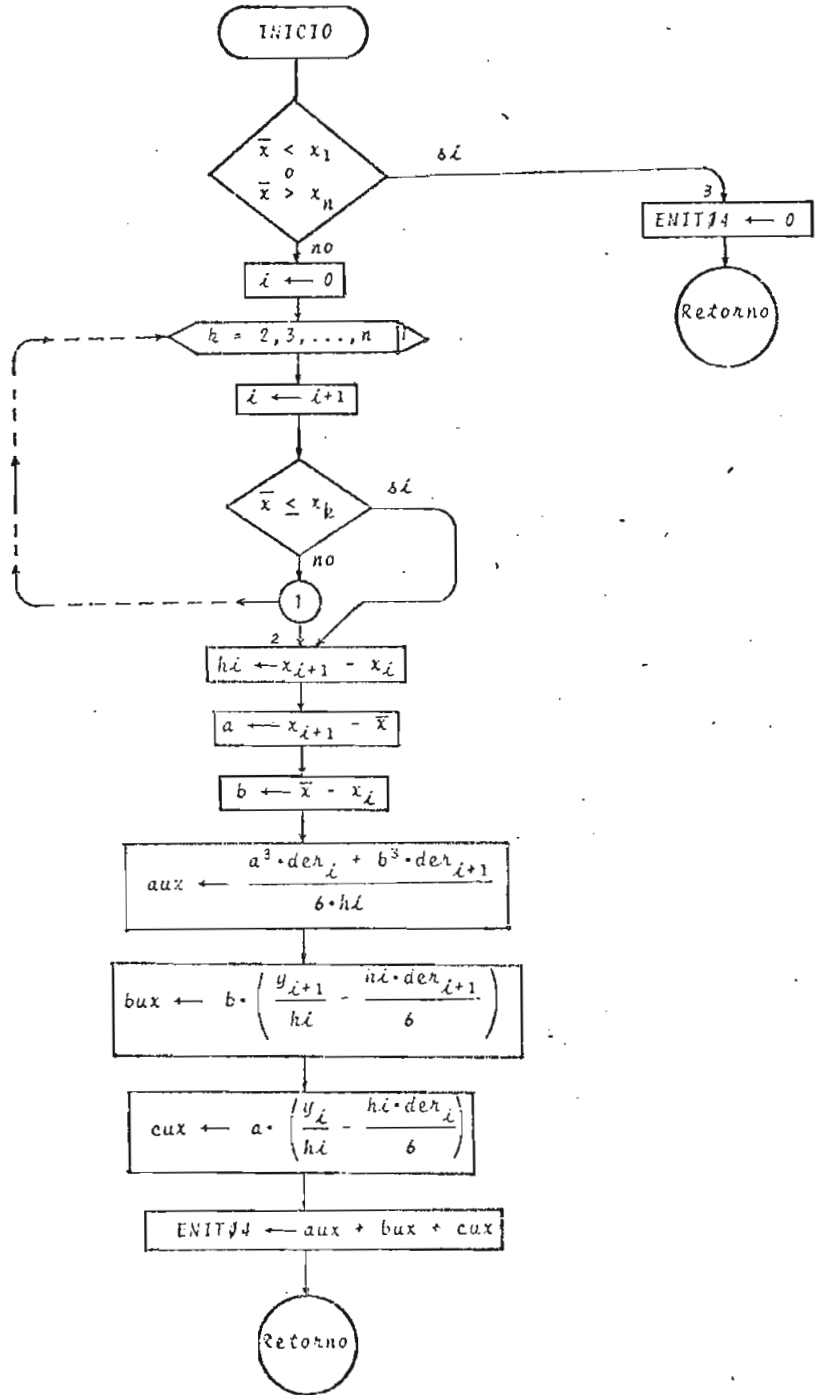


SUBROUTINA ENIT03: (Continuación).



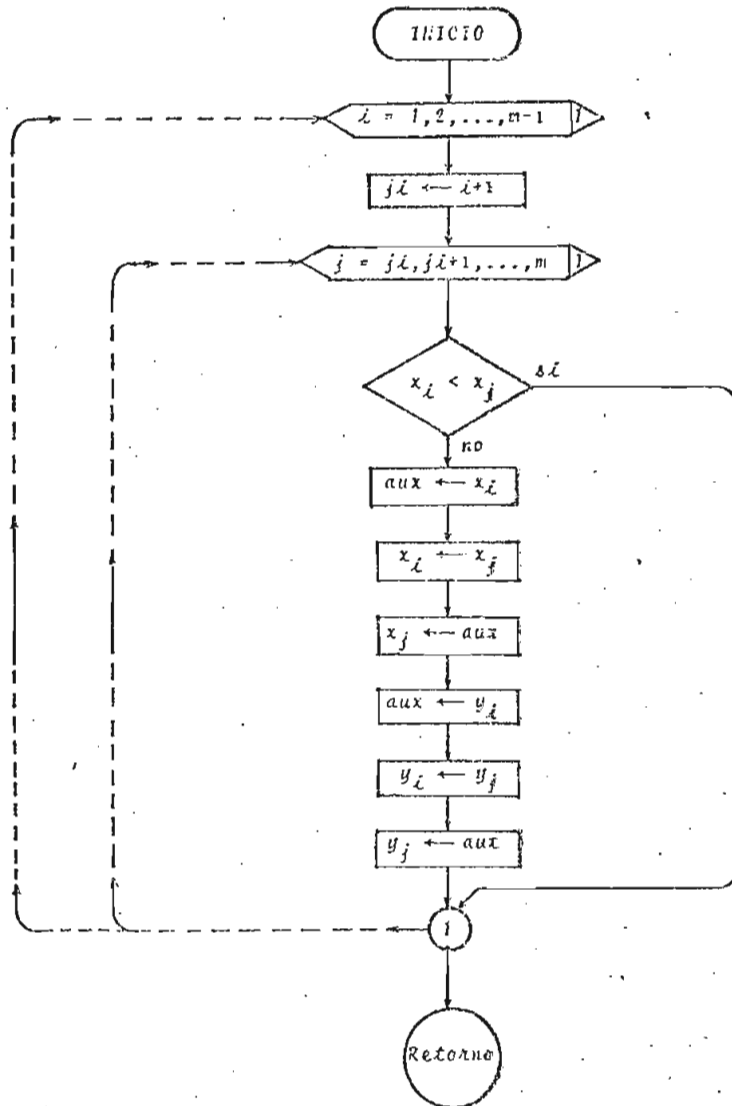
FUNCION ENIT04:

(Argumentos: n, x, y, argx, der).

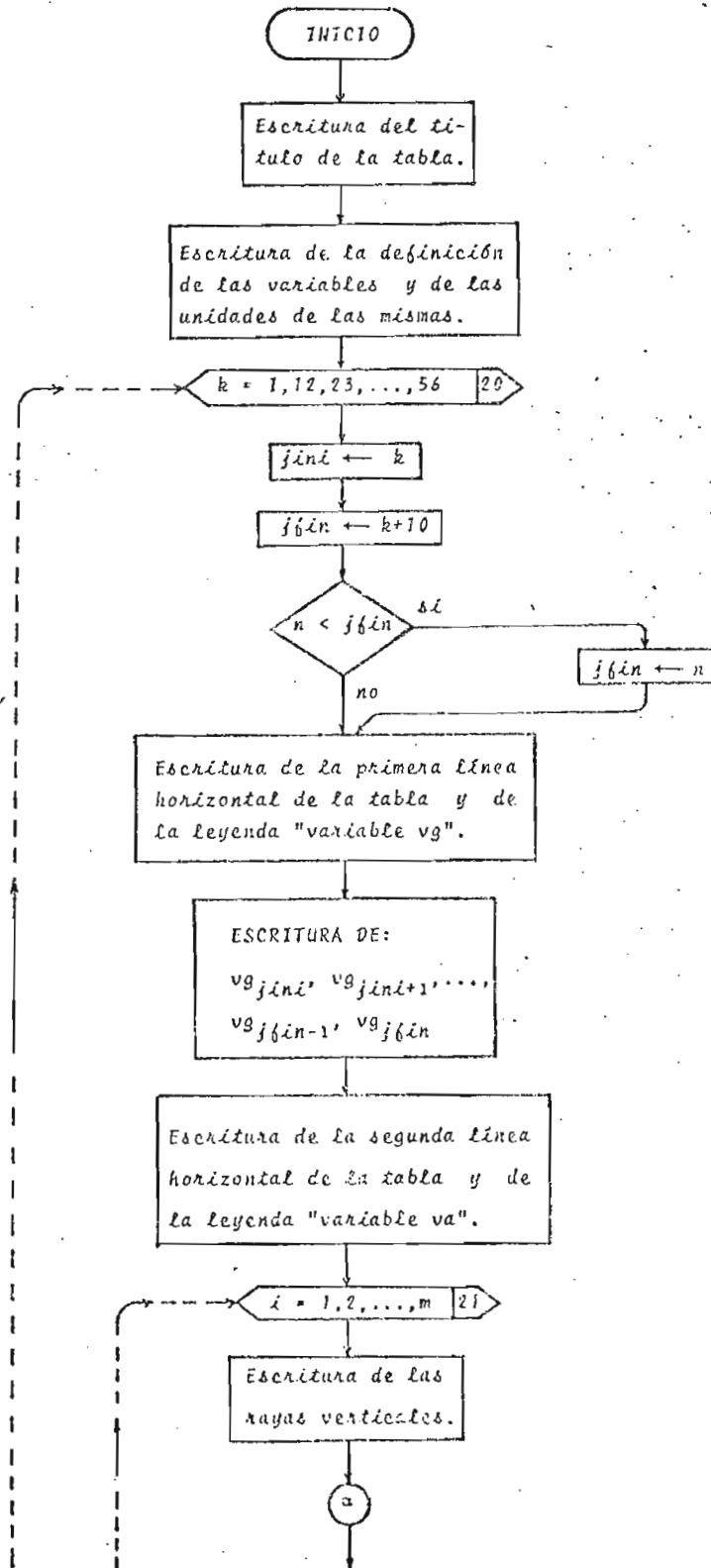


SUBROUTINA ENITØ5:

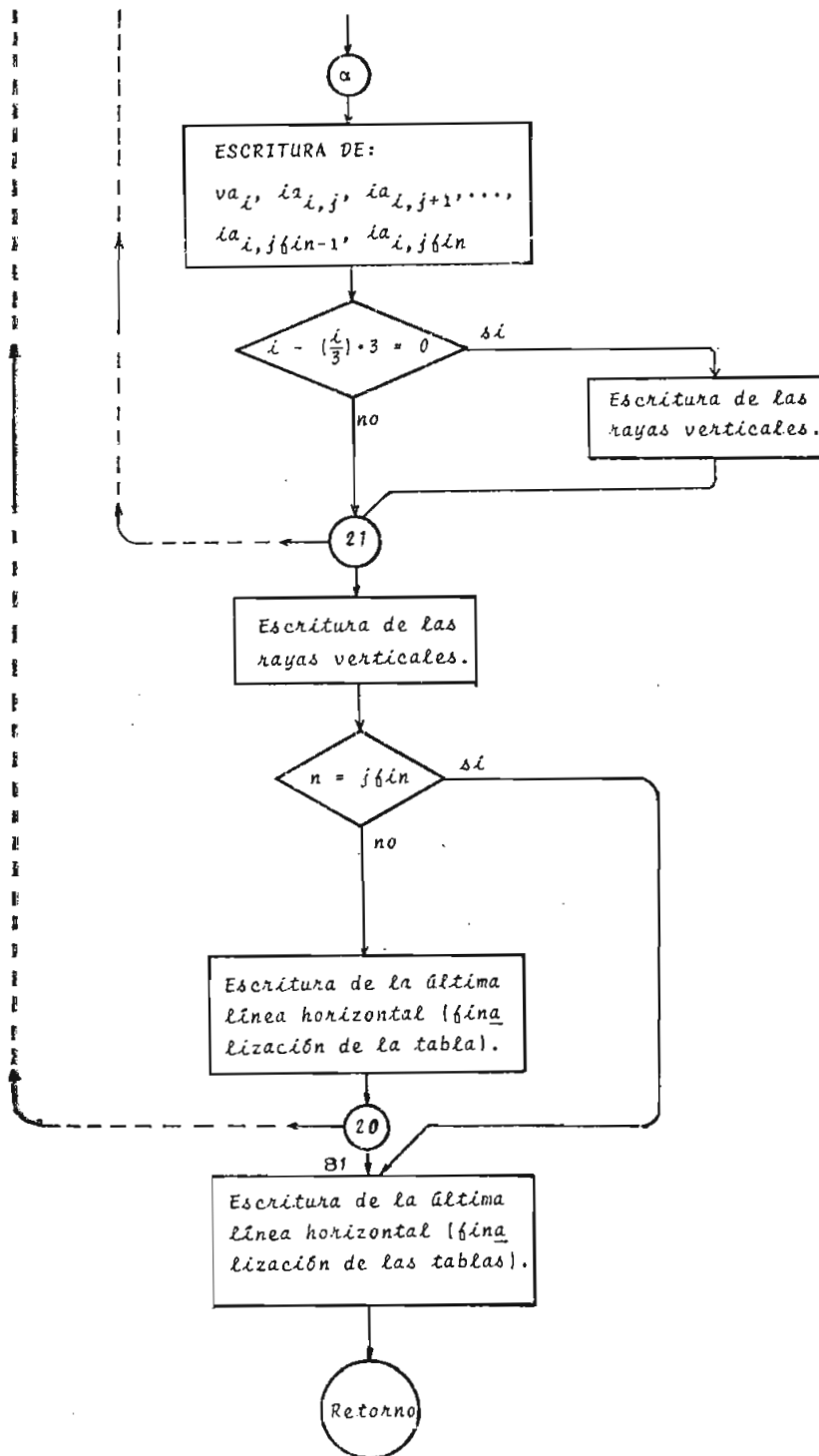
(Argumentos: m , x , y).



SUBROUTINA ESCRIT:



SUBROUTINA SCRIT: (Continuación).



CODIFICACION FORTRAN

LISTA DE VARIABLES PRINCIPALES

Símbolos en Definición
Los programas

(Principal)

M	Número de puntos bases tomados de la variable independiente x , m .
N	Número de puntos bases tomados de la variable independiente y , n .
VA	Arreglo que contiene los puntos bases de x , x_i ; $i = 1, 2, \dots, m$.
VG	Arreglo que contiene los puntos bases de y , y_j ; $j = 1, 2, \dots, n$.
IA	Arreglo que contiene las $m \times n$ muestras tomadas de la variable dependiente z , z_{ij} ; $i = 1, 2, \dots, m$ y $j = 1, 2, \dots, n$.
ELMT	Arreglo en el que se almacena datos literales (título de la tabla que imprime ESCRIT)
DVARVA	Arreglo que contiene como dato literal, la definición de la variable x .
DVARVG	Arreglo que contiene, como dato literal, la definición de la variable y .

DVARIA	Arreglo que contiene, como dato literal, la definición de la variable z.
UVA	Arreglo que contiene las unidades de la variable x.
UVG	Arreglo que contiene las unidades de la variable y.
UIA	Arreglo que contiene las unidades de la variable z.
TIPO	Variable en la que se almacena la información, que sirve para saber cual de las tres variables va a ser determinada por interpolación conociendo las otras dos.
ARG1, ARG2	Contienen los valores de las dos variables para las cuales se desea conocer la tercera variable por interpolación, \bar{x} y \bar{y} ; \bar{x} y \bar{z} o \bar{y} y \bar{z} .
IDE	Variable entera que se utiliza para informar a la subrutina ENITØ1, cuál es la incognita a determinarse por interpolación.
NUM	Variable en la que se almacena el número de veces que se llama a ENITØ1.
ARG3	Variable que contiene el valor de la incognita, determinado por interpolación, \bar{z} , \bar{y} o \bar{x} .

IND, JJ Variables indicadoras del buen desenvolvimiento del programa.

(Subrutina ENITØ1)

DER2VA Arreglo bidimensional en el que se almacenan las $m \times n$ segundas derivadas con respecto a la variable independiente x , $der2x_{ij}$;
 $i = 1, 2, \dots, m$
 $j = 1, 2, \dots, n$

DER2VG Arreglo bidimensional que contiene las $m \times n$ segundas derivadas con respecto a la variable dependiente y , $der2y_{ij}$; $i = 1, 2, \dots, m$
 $j = 1, 2, \dots, n$

FVAOG Arreglo en el que se almacenan los nuevos valores funcionales, que dependen sólo de una de las variables independientes, ya sean estos los f_j ; $j = 1, 2, \dots, n$ o los f'_i ; $i = 1, 2, \dots, m$.

(Subrutina ENITØ2)

N Número de muestras, n , tomadas de x y de $f(x)$.

X Arreglo en el que se almacena los n puntos bases conocidos de x , x_i .
 $i = 1, 2, \dots, n$.

- Y Arreglo que contiene las n muestras de $f(x)$ y_i ; $i = 1, 2, \dots, n$.
- DER Arreglo en el que se almacenan las n segundas derivadas, $P''_{3,i}(x)$, determinadas por cálculo.
- COEF Arreglo bidimensional en el que se almacenan los coeficientes del sistema de $n-2$ ecuaciones simultáneas lineales.
- CONST Arreglo en el que se almacena las constantes del mismo sistema de ecuaciones.

(Subrutina ENITØ3)

- N Número de ecuaciones lineales simultáneas, n .
- A Arreglo bidimensional que contiene los elementos de la matriz de coeficientes del sistema de ecuaciones.
- C Arreglo que un principio contiene la matriz de constantes del sistema de ecuaciones. Luego en este arreglo se almacena las incógnitas, después de ser calculadas.

(Función ENITØ4)

- N Número de muestras tomadas de x y de $f(x)$, n .

- X Arreglo que contiene los n puntos bases conocidos de x , x_i ; $i = 1, 2, \dots, n$.
- Y Arreglo que contiene las n muestras de $f(x)$. y_i ; $i = 1, 2, \dots, n$.
- DER. Arreglo que contiene las n segundas derivadas, $P_{3,i}''(x)$.
- ARGX Valor para el cual se desea conocer el valor de la función por interpolación, \bar{x} .

(Subrutina ENIT05)

- X Arreglo, cuyos elementos van a ser ordenados en forma ascendente.
- Y Arreglo, cuyos elementos corresponde, uno a uno a los elementos del arreglo X.
- M Número de elementos del arreglo X y del arreglo Y.

SUBROUTINA SCRIT:

```

0001      SUBROUTINE ESCRIT
C
C      SUBPROGRAMA PARA ESCRIBIR UN CONJUNTO DE VALORES, CONFORMANDO UNA
C      TABLA, QUE SON FUNCION DE DOS VARIABLES.
C
C      ESCRIT ESCRIBE EL TITULO DE LA TABLA, LA DEFINICION DE LAS VARIABLES
C      USADAS CON SUS RESPECTIVAS UNIDADES. LA TABLA CONSTA (CNC MAXINC CE
C      11 COLUMNAS DE VALORES FUNCIONALES. EL NUMERO DE FILAS DEPENDE DE LA
C      CANTIDAD DE VALORES QUE SE DISPONEN DE LA VARIABLE INDEPENDIENTE, VA.
C      CUANDO EL NUMERO DE VALORES DE LA VARIABLE INDEPENDIENTE, VG, ES
C      MAYOR QUE 11, ENTONCES SE ELABORA DOS O MAS TABLAS ( MAXINC 5 TABLAS ).
C
0002      REAL * 8 VA, VG, IA
0003      DIMENSION ELMT(40), DVARVA(6), DVAVG(6), DVARIA(6), UVA(3),
0004      *UVG(3), UIA(3), VA(20), VG(15), IA(20,15)
      COMMON VA, VG, IA, ELMT, DVARVA, DVAVG, DVARIA, UVA, UVG, UIA, N, N
C
C      ESCRITURA DEL TITULO DE LA TABLA; DE LA DEFINICION DE LAS VARIABLES
C      Y UNIDADES DE LAS MISMAS.
C
0005      WRITE (3,44) ELMT
0006      44 FORMAT (1H1///(21X20A4))
0007      WRITE (3,62) DVARVA, UVA, DVAVG, DVARIA, UIA
0008      62 FORMAT (//11X*SIMBOLOGIA USADA*,4X,15X*UNIDADES*//11X*VA = *,
      *6A4,6X3A4//11X*VG = *6A4,6X3A4//11X*A = *6A4,6X3A4)
C
C      ESCRITURA DE LA VARIABLE VG.
C
0009      DO 20 K=1,56,11
0010      JINI=K
0011      JFIN=K+10
0012      IF(N.LT.JFIN) JFIN=N
0013      WRITE(3,45)
0014      45 FORMAT(//16X* *S(---)* *1(6X*) VARIABLE VG*86X
      *1//15X*1*98X*//15X*11(8X*1))
0015      WRITE(3,46) (VG(J),J=JINI,JFIN)
0016      46 FORMAT(*1//16X11(F7.2,2X))
0017      WRITE(3,47)
0018      47 FORMAT(6X* *S(---)* *11(R(---)*1)/6X*) VARIABLE *
      *98X*//6X* VA *1//6X*//6X*//9(---)* *S(---)* *1)
0019      WRITE(3,48)
C
C      ESCRITURA DE LAS VARIABLES VA E IA.
C
0020      DO 21 I=1,M
0021      WRITE(3,48)
0022      48 FORMAT(6X*11(8X*1))
0023      WRITE(3,49) VA(1), (IA(1,J),J=JINI,JFIN)
0024      49 FORMAT(*1//6X*F8.2,11F5.2)
0025      IF(MOD(I,3).EQ.0) WRITE(3,48)
0026      21 CONTINUE
0027      WRITE(3,48)
0028      IF(N.EQ.JFIN) GO TO 21
0029      20 WRITE (3,50)
0030      50 WRITE(3,50)
0031      50 FORMAT(5X* *10E(---)* *1)
0032      RETURN
0033      END

```

SUBROUTINA ENIT01:

```

C
C      SUBROUTINE ENIT01 (M,N,VA, VG, IA, ARG1, ARG2, ARG3, IDE, NUM, IND, JI, NP)
C
C      SUBPROGRAMA PARA INTERPOLAR DIRECTA E INVERSAMENTE UNA FUNCION
C      TRIDIMENSIONAL, QUE ES CONOCIDA DE UNA MANERA DISCRETA, UTILIZANDO
C      POLINOMIOS DE COLOCACION SPLINE CUBICOS EN DOS DIMENSIONES.
C
C      ENIT01 ASUME QUE LOS ELEMENTOS DE VA Y VG ESTAN EN ORDEN ASCENDENTE.
C      CALCULA LAS M*N SEGUNDAS DERIVADAS CON RESPECTO A VA (VG CONSTANTE)
C      Y A VG (VA CONSTANTE); REALIZANDO ESTE CALCULO SOLO EN LA PRIMERA
C      LLAMADA A ENIT01 (NUM = 0). DEPENDIENDO DEL VALOR DE IDE = 1,2,3
C      CALCULA POR INTERPOLACION EL VALOR DE ARG3 = 1, VA, VG, VA RESPECTIVAMENTE
C
      REAL * 8 VA, VG, IA, AUX, DER, DERVA, DERVVG, ARG1, ARG2, ARG3, FVACC, SPLINE
      DIMENSION VA(M), VG(N), IA(M,N), DER(50), DER2VA(50,50),
      *DER2VG(50,50), FVAUG(50), AUX(60)
      IND = 0
      IF (NUM.NE.0) GO TO 63

```

SUBROUTINA ENIT01: (Continuación).

```
C
C CALCULO DE LAS M*N SEGUNDAS DERIVADAS CON RESPECTO A VA. DER2VA.
C
      JI=0
      DO 22 J=1,N
      DO 23 I=1,M
23   AUX(I)=IA(I,J)
      CALL ENIT02 (N,VA,AUX,DER,IND)
      JI=JI+1
      IF (IND.EQ.-1) GO TO 82
      DO 22 I=1,M
22   DER2VA(I,J)=DER(I)
C
C CALCULO DE LAS M*N SEGUNDAS DERIVADAS CON RESPECTO A VG. DER2VG.
C
      JI=0
      DO 25 I=1,M
      DO 26 J=1,N
26   AUX(J)=IA(I,J)
      CALL ENIT02 (N,VG,AUX,DER,IND)
      JI=JI+1
      IF (IND.EQ.-1) GO TO 84
      DO 25 J=1,N
25   DER2VG(I,J)=DER(J)
83   NUM = NUM + 1
      IF (IDE.EQ.3) GO TO 87
C
C CALCULO DE LOS NUEVOS VALORES FUNCIONALES. QUE DEPENDEN UNICAMENTE DE
C LA VARIABLE VG.
C
      DO 29 J=1,N
      DO 30 I=1,M
      AUX(I)=IA(I,J)
30   DER(I)=DER2VA(I,J)
29   FVADG(J)=ENIT04 (N,VA,AUX,ARG1,DER)
C
C CALCULO DE LA VARIABLE IA. CONOCIDO LAS VARIABLES VA Y VG. (INTERPOLACION
C DIRECTA).
C
      CALL ENIT02 (N,VG,FVADG,DER,IND)
      IF (IND.EQ.-1) GO TO 86
      IF (ICE.EQ.2) GO TO 89
      ARG3=ENIT04 (N,VG,FVADG,ARG2,DER)
      RETURN
C
C CALCULO DE LA VARIABLE VG. CONOCIDO LAS VARIABLES VA E IA (INTERPOLACION
C INVERSA).
C
89   DO 31 J=1,N
31   AUX(J)=VG(J)
      CALL ENIT05(N,FVADG,AUX)
      CALL ENIT02 (N,FVADG,AUX,DER,IND)
      IF (IND.EQ.-1) GO TO 88
      ARG3=ENIT04 (N,FVADG,AUX,ARG2,DER)
      RETURN
C
C CALCULO DE LOS NUEVOS VALORES FUNCIONALES QUE DEPENDEN UNICAMENTE DE
C LA VARIABLE VA.
C
87   DO 32 I=1,M
      DO 31 J=1,N
      AUX(J)=IA(I,J)
31   DER(J)=DER2VG(I,J)
32   FVADG(I)=ENIT04 (N,VG,AUX,ARG1,DER)
C
C CALCULO DE LA VARIABLE VA. CONOCIDO LAS VARIABLES VG E IA (INTERPOLACION
C INVERSA).
C
      DO 34 I=1,M
34   AUX(I)=VA(I)
      CALL ENIT05(N,FVADG,AUX)
      CALL ENIT02 (N,FVADG,AUX,DER,IND)
      IF (IND.EQ.-1) GO TO 88
      ARG3=ENIT04 (N,FVADG,AUX,ARG2,DER)
      RETURN
C
C ESCRITURA DE IND Y JI EN EL CASO QUE HAYA INCONSISTENCIA EN EL PROGRAMA.
C
82   WRITE(3,58) IND,JI
58   FORMAT(/'11X'PROGRAMA INTERRUMPIDO POR OBTENERSE UN IND.=*
      *12.* EN LA COLUMNA *12)
      RETURN
84   WRITE(3,59) IND,JI
59   FORMAT(/'11X'PROGRAMA INTERRUMPIDO POR OBTENERSE UN IND.=*
      *12.* EN LA FILA *12)
      RETURN
88   JI=0
      WRITE(3,60) IND,JI
60   FORMAT(/'11X'IND.=*12,2X*JI=*12)
      RETURN
      END
```

SUBROUTINE ENIT02:

```

SUBROUTINE ENIT02 (N,X,Y,DER,IND)
C
C SUBPROGRAMA PARA EL CALCULO DE LAS SEGUNDAS DERIVADAS NECESARIAS PARA
C LA EVALUACION DEL POLINOMIO DE COLOCACION SPLINE CUBICO EN DOS DIMENSIONES
C
C
C ENIT02 ASUME QUE LOS PUNTOS BASES X(I); I = 1,2,...,N ESTAN EN ORDEN
C ASCENDENTE. INICIALIZA LOS ELEMENTOS DEL ARREGLO COEF CON EL VALOR
C CERO. ASIGNA A DER(1) Y A DER(N) EL VALOR CERO. CALCULA LOS ELEMENTOS
C DE LA MATRIZ DE COEFICIENTES, COMO TAMBIEN LOS ELEMENTOS DE LA MATRIZ
C DE CONSTANTES DEL SISTEMA SIMULTANEO DE N-2 ECUACIONES LINEALES.
C LLAMA A LA SUBROUTINA ENIT03 PARA RESOLVER DICHO SISTEMA Y CE ESTA
C MANEJA SE ENCUENTRA LAS OTRAS SEGUNDAS DERIVADAS DE LOS POLINOMIOS
C SPLINE CUBICOS DER(2)...DER(N-1)
C
      IMPLICIT REAL*8 (A-D,F-H,O-Z)
      DIMENSION X(N),Y(N),DER(N),COEF(48,48),CONST(48)
      DATA EPS/1.E-15/,EPS2/1.E-2/,EPS3/1.E-3/
      NN = 48
      NME2=N-2
C
C INICIALIZACION DE LOS ELEMENTOS DEL ARREGLO COEF.
C
      DO 5 I=1,NME2
      DO 5 J=1,NME2
      5 COEF(I,J)=0.00
C
C ASIGNACION DEL VALOR CERO A LAS SEGUNDAS DERIVADAS DER(1) Y DER(N).
C
      DER(1)=0.00
      DER(N)=0.00
C
C CALCULO DE LOS ELEMENTOS DE LOS ARREGLOS COEF Y CONST.
C
      IF(DABS(X(2)-X(1)).LT.EPS3) X(2)=X(2) * (1. + EPS2)
      H1=X(2)-X(1)
      D1=Y(2)-Y(1)
      NMF1=N-1
      DO 1 I=2,NMF1
      IF(DABS(X(I+1)-X(I)).LT.EPS3) X(I+1)= X(I+1) * (1. + EPS2)
      HAC=X(I+1)-X(I)
      DAC=Y(I+1)-Y(I)
      A=H1/HAC
      IF(1.NE.2) COEF(I-1,I-2)=A
      COEF(I-1,I-1)=2.00*(1.00+A)
      C=5.00*(DAC-D1/A)/HAC**2
      CONST(I-1)=C
      IF(1.NE.2) OR(1.NE.NMF1) GO TO 4
      IF(1.NE.2) DER(I-1)=DER(N)
      CONST(I-1)=C-DER(I-1)*A
      4 H1=HAC
      1 D1=DAC
      NME3=N-3
      DO 2 I=1,NME3
      2 COEF(I,I+1)=1.00
C
C CALCULO DE LAS N-2 SEGUNDAS DERIVADAS RESTANTES.
C
      CALL ENIT03 (NME2,COEF,CONST,EPS,IND,NN)
      DO 3 I=2,NMF1
      3 DER(I)=CONST(I-1)
      RETURN
      END

```

SUBROUTINE ENIT03:

```

SUBROUTINE ENIT03 (N,A,C,EPS,IND,NN)
C
C SUBPROGRAMA PARA RESOLVER UN SISTEMA ESPECIFICO DE ECUACIONES
C SIMULTANEAS LINEALES, QUE SE PRESENTA EN LA TECNICA DE INTERPOLACION
C SPLINE CUBICA, UTILIZANDO EL METODO DE ELIMINACION DE GAUSS.
C
C
C ENIT03 INICIA HACIENDO CEROS LOS ELEMENTOS DEL ARREGLO A, QUE ESTAN
C BAJO LA DIAGONAL PRINCIPAL. EN ESTE PROCESO SE VA COMPARANDO EL
C ELEMENTO A(I,I) DE LA FILA I QUE SE TOMA COMO REFERENCIA CON EL
C ELEMENTO A(I+1,I) ; EN EL CASO QUE ESTE ULTIMO SEA MAYOR EN VALOR
C ABSOLUTO SE TOMA COMO REFERENCIA LA FILA I+1. UNA VEZ CONSEGUIDOS
C LOS CEROS SE ENCUENTRA LAS INCOGNITAS DESPESANDO AGRESIVAMENTE;

```

SUBROUTINA ENITØ3: (Continuación).

```

C ES DECIR PRIMERØ X(N), LUEGO X(N-1), HASTA X(1). LOS VALORES DE LAS
C INCOGNITAS SON ALMACENADOS EN EL ARRREGLO C, QUE EN EL MOMENTO QUE
C ENITØ3 ES LLAMADA CONTIENE LAS CONSTANTES DEL SISTEMA DE ECUACIONES.
C TAMBIEN SE VA VERIFICANDO SI ES UN SISTEMA SINGULAR O NO.
C
      DEFLAG A,C,AUX
      DIMENSION A(M,N),C(N)
C
C OBTENCION DE CPDS EN LOS ELEMENTOS DE A, QUE ESTAN BAJO LA DIAGONAL
C PRINCIPAL.
      NME1=N-1
      DO 1 I=1,NME1
        K=1
        Y3=K+2
        IF(K.EQ.NME1)K3=K+1
        IF(DABS(A(I,1)).GE.DABS(A(I+1,1))) GO TO 5
C
C CHEQUEO DE SINGULARIDAD EN EL SISTEMA DE ECUACIONES
      IF(DABS(A(I+1,1)).LT.EPS) GO TO 6
      DO 3 M=K,K2
        AUX=A(I,M)
        A(I,M)=A(I+1,M)
      3 A(I+1,M)=AUX
        AUX=C(I)
        C(I)=C(I+1)
        C(I+1)=AUX
      5 K=K+1
      IF(DABS(A(I,1)).LT.EPS) GO TO 6
      DO 2 J=K+1,N
        A(I+1,J)=A(I+1,J)-A(I,J)*A(I+1,1)/A(I,1)
      2 C(I+1)=C(I+1)-C(I)*A(I+1,1)/A(I,1)
C
C DESPEJE REGRESIVO DE LAS INCOGNITAS.
      C(N)=C(N)/A(N,N)
      L=N-1
      C(L)=(C(L)-A(L,L+1)*C(L+1))/A(L,L)
      NME2=N-2
      DO 4 I=1,NME2
        L=L-1
      4 C(L)=(C(L)-(A(L,L+1)*C(L+1)+A(L,L+2)*C(L+2)))/A(L,L)
      IND=0
      RETURN
      6 IND=-1
      RETURN
      END

```

FUNCION ENITØ4:

```

      FUNCTION ENITØ4 (N,X,Y,ARGX,DER)
C
C SUBPROGRAMA PARA EVALUAR LOS POLINOMIOS DE COLOCACION SPLINE CUBICOS
C EN DOS DIMENSIONES.
C
C ENITØ4 ASUME QUE LAS SEGUNDAS DERIVADAS, DER(1) ... DER(N), FUERON
C YA CALCULADAS CON ANTERIORIDAD. TAMBIEN ASUME QUE LOS PUNTOS BASES
C X(1) ... X(N) ESTAN ORDENADOS DE MENOR A MAYOR. VERIFICA SI EL VALOR
C ARGX, PARA EL CUAL SE DESEA CONOCER EL VALOR DE LA FUNCION POR
C INTERPOLACION, ESTA DENTRO DEL INTERVALO DE INTERPOLACION ( X(1),LE.
C ARGX,LE,X(N) ). EN EL CASO QUE ARGX ESTE FUERA DEL INTERVALO SE
C ASIGNA A ENITØ4 EL VALOR CERO. LUEGO SE SELECCIONA EL POLINOMIO
C SPLINE CUBICO A EVALUARSE, DETERMINANDO ESTO DE LA POSICION QUE TIENE
C ARGX EN EL INTERVALO DE INTERPOLACION. EL VALOR EVALUADO SE LE
C ASIGNA A ENITØ4.
C
      IMPLICIT REAL*8 (A-H,C-Z)
      DIMENSION X(N),Y(N),DER(N)
C
C VERIFICACION SI ARGX ESTA DENTRO DEL INTERVALO DE INTERPOLACION.
      IF(ARGX.LE.X(1).OR.ARGX.GT.X(N)) GO TO 3
C
C SELECCIONAMIENTO DEL POLINOMIO SPLINE CUBICO RESPECTIVO.
      I=0
      DO 1 K=2,N
        I=I+1
        IF(ARGX.LE.X(K)) GO TO 2
      1 CONTINUE

```

FUNCION ENITØ4: (Continuación).

```
C
C EVALUACION DEL POLINOMIO SPLINE CUBICO SELECCIONADO.
C
2 H1=X(I+1)-X(I)
  A=Y(I+1)-A*Y(I)
  B=ANGX-X(I)
  AUX=A**3*(1/6.00*H1)+B**3*DER(I+1)/6.00*H1
  BUX=A*Y(I+1)/H1-H1*Y(I)*DER(I+1)/6.00
  CUX=A*Y(I+1)**2-H1*Y(I)*DER(I+1)/6.00
  ENITØ4=AUX+BUX+CUX
  RETURN
3 ENITØ4=0.00
  RETURN
  END
```

SUBROUTINA ENITØ5:

```
      SUBROUTINE ENITØ5(M,X,Y)
C
C SUBPROGRAMA PARA ORDENAR UN ARREGLO DE MENOR A MAYOR (ORDEN ASCENDENTE)
C
C EL ARREGLO QUE SE VA A ORDENAR ES X; A CADA ELEMENTO X(I) (I=1,2,...,M)
C LE CORRESPONDE UN VALOR Y(I) (I=1,2,...,M) DE DICHO ARREGLO Y EN EL
C ORDENAMIENTO NO SE ALTERA DICHA CORRESPONDENCIA.
C
  REAL*4 X,Y,AUX
  DIMENSION X(M),Y(M)
  MM=1
  DO 1 I=1,MM-1
    J=I+1
    DO 1 J=J,M
      IF(X(I).LT.X(J)) GO TO 1
      AUX=X(I)
      X(I)=X(J)
      X(J)=AUX
      AUX=Y(I)
      Y(I)=Y(J)
      Y(J)=AUX
1 CONTINUE
  RETURN
  END
```

ENTRADA DE DATOS:

La forma de perforar los datos es:

- 1.- En la primera y segunda tarjeta se puede perforar en cualquier columna cualquier carácter. Esto se lo utiliza para el título de la tabla de valores que se imprime.
- 2.- En la tercera tarjeta perforar desde la columna 3 hasta

la columna 26 la definición de la variable x. Desde la columna 29 hasta la columna 52 la definición de la variable y, y desde la columna 55 hasta la 78 la definición de la variable z. (3(2X6A4)).

- 3.- En la cuarta tarjeta se debe perforar las unidades de las variables x, y, z. En las 12 primeras columnas las unidades de x. A partir de la columna 21 hasta la 32 las unidades de y. Desde la columna 41 hasta la 52 las unidades de z. (3A4, 8X3A4, 8X3A4).
- 4.- En la quinta tarjeta se perfora: En las columnas 1 y 2 el número de muestras de la variable x, m. En las columnas 6 y 7 el número de muestras de la variable y, n. A partir de la columna 11 perforar siete valores de x, abarcando cada valor 10 columnas, en una de las cuales se debe perforar el punto decimal, dependiendo del valor a perforarse, sin importar en cual de ellas. (I2, 3XI2, 3X7F10.0). Los valores de m y n deben perforarse de tal manera que la última columna de su campo corresponda a las unidades de dichos números.
- 5.- Desde la sexta tarjeta, perforar 8 valores de x por tarjeta hasta terminar con todas las muestras de x; de tal manera que cada valor ocupe un campo de 10 columnas incluido el punto decimal. (8F10.0).

- 6.- A continuación de las muestras de x, perforar las muestras de y como lo indica el numeral 5, cambiando de tarjeta cuando sea menester; es decir solamente cuando se hayan perforado 8 valores por tarjeta (cambio obligado de tarjeta).
- 7.- En una nueva tarjeta perforar 7 muestras de z, abarcando cada valor un campo de 10 columnas incluido el punto decimal. De igual forma se deben perforar las siguientes tarjetas hasta terminar con las muestras de z. Estas muestras se deben ir perforando por filas (ver tabla 2.5). (10X7F10.0).
- 8.- En una tarjeta perforar: En la columna 11 hasta la 14 el valor de la variable. TIPO, Entre las columnas 31 y 40 el valor de ARG1. Desde la columna 41 a 50 el valor de ARG2. En las columnas correspondientes a ARG1 y ARG2 debe perforarse el respectivo punto decimal, sin importar su ubicación dentro del campo. Del valor de TIPO depende la definición de ARG1 y ARG2; presentándose tres casos: (10XA4,16X,2F10.0)

TIPO	ARG1	ARG2
PIA	\bar{x}	\bar{y}
PVG	\bar{x}	\bar{z}
PVA	\bar{y}	\bar{z}

El numeral 8 se debe repetirse tantas veces como conjuntos de tres valores haya (TIPO, ARG1, ARG2).

El valor máximo de m es 20 y el de n 15.

RESULTADOS: (Continuación).

Nº.	DATOS :	INCOGNITA :
1	VA = 15.00 VOLTIOS IA = 100.00 MILLIAMPEROS	VG = -12.68 VOLTIOS
2	VA = 50.00 VOLTIOS IA = 200.00 MILLIAMPEROS	VG = -6.68 VOLTIOS
3	VA = 150.00 VOLTIOS IA = 300.00 MILLIAMPEROS	VG = -4.35 VOLTIOS
4	VG = -17.50 VOLTIOS IA = 70.00 MILLIAMPEROS	VA = 65.00 VOLTIOS
5	VG = -12.50 VOLTIOS IA = 140.00 MILLIAMPEROS	VA = 99.00 VOLTIOS
6	VG = -10.00 VOLTIOS IA = 60.00 MILLIAMPEROS	VA = 6.73 VOLTIOS
7	VA = 75.00 VOLTIOS VG = -17.50 VOLTIOS	IA = 73.16 MILLIAMPEROS
8	VA = 100.00 VOLTIOS VG = -17.50 VOLTIOS	IA = 66.57 MILLIAMPEROS
9	VA = 100.00 VOLTIOS VG = -5.00 VOLTIOS	IA = 258.57 MILLIAMPEROS
10	VA = 125.00 VOLTIOS VG = -20.00 VOLTIOS	IA = 71.52 MILLIAMPEROS
11	VA = 175.00 VOLTIOS VG = -10.00 VOLTIOS	IA = 203.64 MILLIAMPEROS
12	VA = 10.00 VOLTIOS IA = 100.00 MILLIAMPEROS	VG = 6.0 VOLTIOS FUERA DEL INTERVALO
13	VA = 75.00 VOLTIOS IA = 53.00 MILLIAMPEROS	VG = -19.55 VOLTIOS
14	VA = 80.00 VOLTIOS IA = 100.00 MILLIAMPEROS	VG = -15.09 VOLTIOS
15	VA = 100.00 VOLTIOS IA = 100.00 MILLIAMPEROS	VG = -18.08 VOLTIOS
16	VA = 130.00 VOLTIOS IA = 100.00 MILLIAMPEROS	VG = -17.50 VOLTIOS
17	VG = -7.50 VOLTIOS IA = 120.00 MILLIAMPEROS	VA = 10.21 VOLTIOS
18	VG = -7.50 VOLTIOS IA = 200.00 MILLIAMPEROS	VA = 79.48 VOLTIOS
19	VA = 50.00 VOLTIOS VG = -8.75 VOLTIOS	IA = 167.10 MILLIAMPEROS
20	VA = 75.00 VOLTIOS VG = -6.75 VOLTIOS	IA = 211.00 MILLIAMPEROS
21	VA = 180.00 VOLTIOS VG = -22.50 VOLTIOS	IA = 64.74 MILLIAMPEROS

RESULTADOS: (Continuación).

NO.	DATOS :		INCÓGNITA :	
22	VC = -27.50 IA = 20.00	VOLTIOS MILIAMP*5	VA = 79.23	VOLTIOS
23	VG = -22.50 IA = 40.00	VOLTIOS MILIAMP*5	VA = 75.13	VOLTIOS
24	VG = -11.75 IA = 140.00	VOLTIOS MILIAMP*5	VA = 86.28	VOLTIOS
25	VA = 25.00 IA = 100.00	VOLTIOS MILIAMP*5	VG = -13.64	VOLTIOS
26	VA = 75.00 IA = 250.00	VOLTIOS MILIAMP*5	VG = -4.64	VOLTIOS
27	VA = 75.00 IA = 50.00	VOLTIOS MILIAMP*5	VG = -20.28	VOLTIOS

Los resultados pertenecen al pentodo PL81 (válvula utilizada para recepción). Se escogió esta válvula por tener las características de placa bastante irregulares y son las que se indican en la figura 2.4. De estas características se obtuvieron las 220 muestras de la corriente de placa que se indican en la tabla de los resultados del programa y que son función del voltaje de placa (20 puntos bases) y del voltaje de grilla (11 puntos bases).

Los valores interpolados son satisfactorios y bastante alagadores. La precisión está dentro de los rangos razonables; pues sería ilógico pretender una precisión más allá de la que permiten las muestras. En la tabla 2.6 se compara algunos de los valores interpolados con a-

quellos leídos directamente de la Figura 2.4.

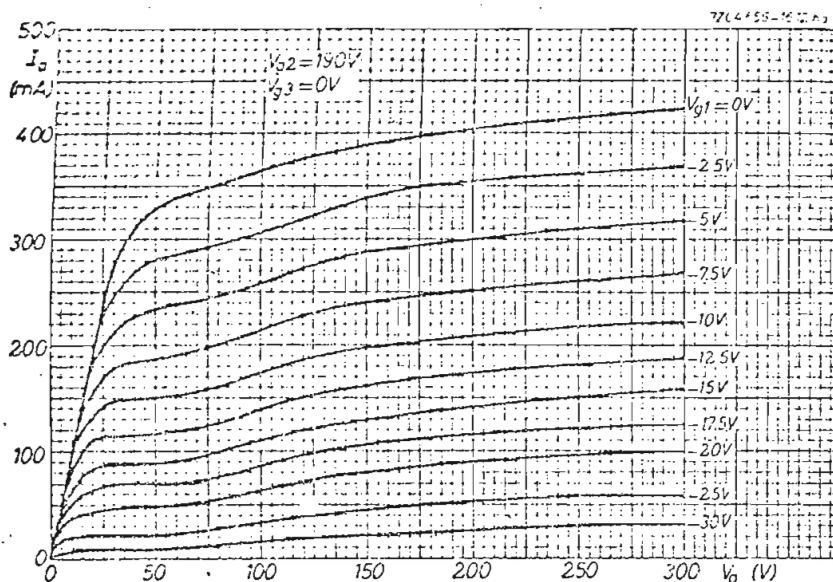


FIGURA 2.4 CARACTERISTICAS DE PLACA DEL PENTODO PL81

Hay que tomar en cuenta, en los valores leídos, que el error de lectura puede ser apreciable dada la escala en la que está el gráfico.

Cabe anotar que al realizar interpolación inversa, por ser una función multivaluada, no se puede predecir cual de los valores funcionales se calculará por estar fuera de todo control. Así para los resultados del ejemplo N° 4, los datos $V_G = -17,5 V$ e $I_A = 70 mA$, que constituyen muestras tomadas, a las que corresponden 4 valores de $V_A = 35 V, 45 V, 50 V, 65 V$; el programa entrega el valor de $V_A = 65 V$.

TABLA 2.6 TABLA COMPARATIVA DE VALORES INTERPOLADOS
CON LOS REALES.

DATOS	VALOR DETERMINADO POR :	
	LECTURA DE LA FIGURA 2.4.	INTERPOLACION
VA = 75,0 V VG = -17,5 V	IA = 74,0 mA	IA = 73,15 mA
VA = 15,0 V IA = 100,0 mA	VG = -12,5 V	VG = -12,68 V
VG = -7,5 V IA = 200,0 mA	VA = 80,0 V	VA = 79,48 V
VA = 130,0 V IA = 100,0 mA	VG = -18,0 V	VG = -17,50 V

C A P I T U L O I I I

DISCUSION DE RESULTADOS Y CONCLUSIONES.

En el capítulo anterior y después de cada programa desarrollado, se realizó ya la discusión pormenorizada de los resultados obtenidos.. En el presente capítulo basándose en la discusión particular anotada, se quiere dar a la discusión de resultados un carácter más general y unitario con el objeto primordial de dar mayor énfasis a las conclusiones que implícitamente son consecuencia, a nivel de recomendación, de la discusión de los resultados.

Si bien es cierto que los programas de interpolación se orientaron para tres dimensiones, no hay inconveniente para interpolar en dos dimensiones; puesto que, con la utilización adecuada de los distintos subprogramas se puede lograr tal propósito. Los requerimientos de capacidad de memoria para estos subprogramas no presentan ningún inconveniente, en relación con la capacidad total disponible de la computadora; por ejemplo el subprograma que mayor capacidad de memoria utiliza es la SUBROUTINA ENITØ1 y es 67374 bytes.

En lo referente a la forma de muestrear una función -de dos o tres dimensiones- se menciona algunas indicaciones:

- 1.- Cuando más se conoce acerca de una determinada función, el error de aproximación en el que se incurre será menor. Por tal motivo, es conveniente obtener un número considerable de muestras de dicha función.
- 2.- Debe obtenerse mayor información de la función a aproximarse, en las regiones en donde presenta mayores variaciones. Claro está, siempre y cuando sea posible.
- 3.- Puede presentarse situaciones, en la utilización de los programas de interpolación en tres dimensiones, ya que es necesario elaborar una tabla con las muestras que se obtienen de una función (ver tabla 2.4), en las que será más conveniente o a veces imprescindible el cambio a otro sistema de coordenadas, con el propósito de completar la tabla indicada. Esto sucede especialmente cuando la fuente de información proviene de un gráfico.

La aplicación de los diferentes programas de interpolación desarrollados en este trabajo es tan vasta, que por tal motivo se los elaboró como subrutinas y subprogramas de función, pues su estructura brinda una gran versatilidad y pudiendo ser utilizados, sin ningún inconveniente, por cualquier programa principal que los requiera. Concretamente la subrutina ENITØ1, que es la más completa, tuvo ya aplicación en la Tesis de Grado "Optimización de alumbrado Público". En la cual, se quería conocer el porcentaje de iluminación en un

punto dado de una calzada, iluminación debido a dos lámparas del mismo tipo y que es igual a la suma de las contribuciones de todas y cada una de las lámparas. Hubo la necesidad de interpolar en tres dimensiones para conocer dicha iluminación. La información provenía de un gráfico de características de iluminación de una lámpara en el plano horizontal, dada la configuración de estas características fue imposible completar la tabla de muestras trabajando en el sistema de coordenadas rectangulares y se optó por el sistema de coordenadas polares. Obteniéndose resultados bastante alagadores; si se desea más detalles a este respecto referirse a (5).

Los programas desarrollados, no obstante de estar lejos de ser los más óptimos proporcionan resultados satisfactorios. Los errores cometidos están dentro del rango de tolerancia que se acepta en ingeniería. Cabe anotar al respecto que la falta de bibliografía en nuestro medio, sobre interpolación tridimensional constituyó una limitación al mejor desenvolvimiento del presente trabajo.

Desligándose del tema, sería conveniente que en los planes de estudio de ingeniería se introduzca como materia obligatoria "Métodos Numéricos" y se la dicte a la par con "Programación", con el objeto de brindar al estudiante un mayor campo de acción.

A P E N D I C E S

A P E N D I C E 1

PLANTEAMIENTO

Se trata de escribir una subrutina TABLAD que calcule todas las diferencias divididas de orden m o menos para los n pares de valores (x_i, y_i) , $i = 1, 2, 3, \dots, n$; siendo $m < n$. Los elementos de la tabla de diferencias finitas deben ser asignados a las primeras m columnas de las primeras $n-1$ filas de la matriz triangular inferior T , de tal manera que

$$T_{ij} = f [x_{i+1}, x_i, \dots, x_{i+1-j}] \quad \begin{cases} i = 1, 2, \dots, n-1 \\ j = 1, 2, \dots, m \end{cases} \quad (A1.1)$$

donde $f(x_i) = y_i$

Asumiendo que los valores x_i están ordenados en forma ascendente y arbitrariamente espaciados; escribir una función llamada EVAPOL para evaluar el polinomio de colocación por diferencias finitas divididas de Newton de grado d para interpolar el argumento \bar{x} , usando las apropiadas diferencias divididas de la matriz T de (A1.1). La función debe determinar de los n puntos bases del arreglo $X (x_1, x_2, \dots, x_n)$ los $d+1$ más convenientes, de tal manera que, \bar{x} quede lo mejor centrado posible entre dichos puntos bases. \bar{x} puede ser menor que x_1 y mayor que x_n , en éstos casos se utilizan los primeros o los últimos $d+1$ puntos bases respectivamente y el poli

nomio de colocación será usado para extrapolar, ya que \bar{x} está fuera del intervalo de los puntos bases utilizados. Con el propósito de probar los dos subprogramas, escribir un programa principal que lea los valores de m , n y x_i , $i = 1, 2, \dots, n$, calcule los elementos del arreglo Y mediante $y_i = \text{Cos } x_i$, y llame a la subrutina `TABLAD` para calcular las diferencias divididas. El programa debe entonces leer los valores \bar{x} y d , luego llamar a la función `EVAPOL` para evaluar el apropiado polinomio de colocación, y por último compare el valor interpolado, $\bar{y}(\bar{x})$, con el valor que se obtiene de $\text{Cos } \bar{x}$ utilizando la función de la biblioteca de la computadora `DCOS`.

METODO DE SOLUCION.-

Usando (A1.1), la tabla de diferencias divididas tendrá la siguiente forma:

$$\begin{array}{rcll}
 x_1 & y_1 & T_{1,1} & = f [x_2, x_1] \\
 x_2 & y_2 & T_{2,1} & = f [x_3, x_2] & T_{2,2} & = f [x_3, x_2, x_1] \\
 x_3 & y_3 & T_{3,1} & = f [x_4, x_3] & T_{3,2} & = f [x_4, x_3, x_2] \\
 \vdots & \vdots & \vdots & & & \\
 x_m & y_m & T_{m,1} & = f [x_{m+1}, x_m] & T_{m,2} & = f [x_{m+1}, x_m, x_{m-1}] & T_{m,m} & = f [x_{m+1}, \dots, x_1] \\
 \vdots & \vdots & \vdots & & & & & \\
 x_{n-1} & y_{n-1} & T_{n-1,1} & = f [x_n, x_{n-1}] & T_{n-1,2} & = f [x_n, x_{n-1}, x_{n-2}] & T_{n-1,m} & = f [x_n, \dots, x_{n-m}] \\
 x_n & y_n & & & & & &
 \end{array}$$

Los elementos de la matriz T, T_{ij} para $j > i$, $j > m$, e $i \geq n$ no se usan. La subrutina TABLAD asume que T es una matriz cuadrada k por k, y se asegura que $m < n$.

La función EVAPOL busca en el arreglo de los puntos bases el valor de i para el cual se cumple $\bar{x} \leq x_i$ (valor más próximo); si $\bar{x} > x_n$, a i se le asigna el valor de n. Los puntos bases usados para determinar el polinomio de colocación son normalmente $x_{\max-d}, \dots, x_{\max}$; siendo $\max = i + d/2$ para d par y $\max = i + (d-1)/2$ para d impar. Cuando max es más pequeño que $d+1$ o más grande que n, entonces se reasigna a max el valor $d+1$ o n respectivamente. Esto asegura que unicamente los puntos bases dados son utilizados para encontrar el polinomio.

En términos de las diferencias divididas, el valor interpolado, $\bar{y}(\bar{x})$, está descrito por el polinomio de grado d:

$$\begin{aligned} \bar{y}(\bar{x}) = & y_{\max-d} + (\bar{x}-x_{\max-d}) f [x_{\max-d+1}, x_{\max-d}] + \\ & + (\bar{x}-x_{\max-d+1}) \cdot (\bar{x}-x_{\max-d}) \times \\ & \times f [x_{\max-d+2}, x_{\max-d+1}, x_{\max-d}] + \\ & + \dots + (\bar{x}-x_{\max-1}) \dots (\bar{x}-x_{\max-d}) \times \\ & \times f [x_{\max}, \dots, x_{\max-d}] \end{aligned} \tag{A1.2}$$

(A1.2) puede ser reescrita de la siguiente forma:

$$\begin{aligned}
 \bar{y}(\bar{x}) = & \{ \{ \dots \{ f [x_{\max}, \dots, x_{\max-d}] \cdot (\bar{x} - x_{\max-1}) + \\
 & + f [x_{\max-1}, \dots, x_{\max-d}] \} \cdot (\bar{x} - x_{\max-2}) + \\
 & + f [x_{\max-2}, \dots, x_{\max-d}] \} \cdot (\bar{x} - x_{\max-3}) + \\
 & + \dots + f [x_{\max-d+1}, x_{\max-d}] \} \cdot (\bar{x} - x_{\max-d}) + \\
 & + y_{\max-d}
 \end{aligned}
 \tag{A1.3}$$

o según (A1.1)

$$\begin{aligned}
 \bar{y}(\bar{x}) = & \{ \{ \dots \{ \{ \dots \{ T_{\max-1,d} \cdot (\bar{x} - x_{\max-1}) + T_{\max-2,d-1} \} \cdot \\
 & \cdot (\bar{x} - x_{\max-2}) + T_{\max-3,d-2} \} \cdot (\bar{x} - x_{\max-3}) + \dots + \\
 & + T_{\max-d,1} \} \cdot (\bar{x} - x_{\max-d}) + y_{\max-d}
 \end{aligned}
 \tag{A1.4}$$

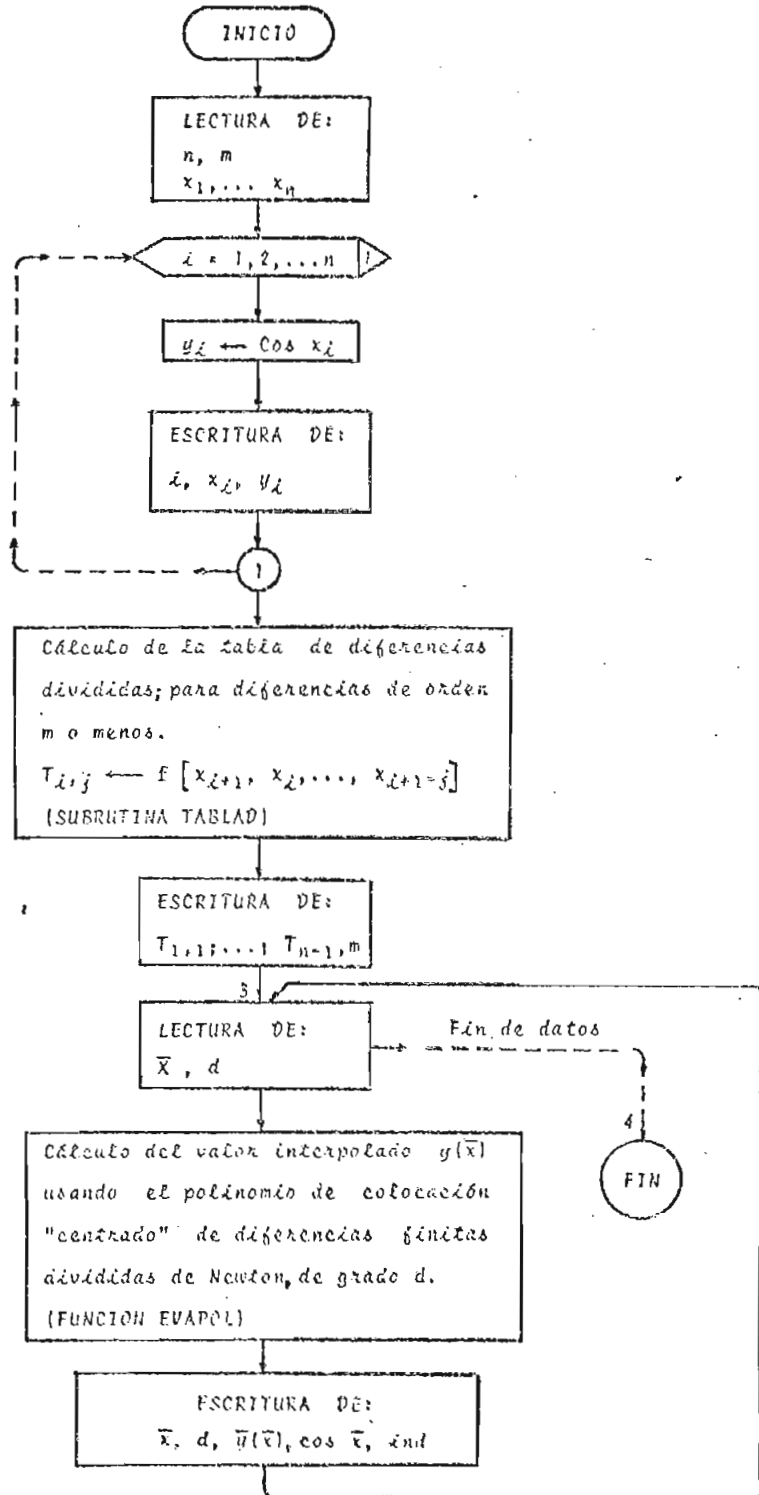
EVAPOL usa la forma (A1.4) para evaluar $\bar{y}(\bar{x})$. Si hubiera inconsistencia en los argumentos de la función, esta es, si $d > m$, el valor que se asigna a EVAPOL es cero; de otra manera el valor asignado es $\bar{y}(\bar{x})$.

Tanto TABLAD como EVAPOL tienen una bandera, ind, es uno cuando se ha encontrado alguna inconsistencia y cuando todo es normal ind es cero. EVAPOL no chequea que los puntos

bases x_1, x_2, \dots, x_n estén en orden ascendente.

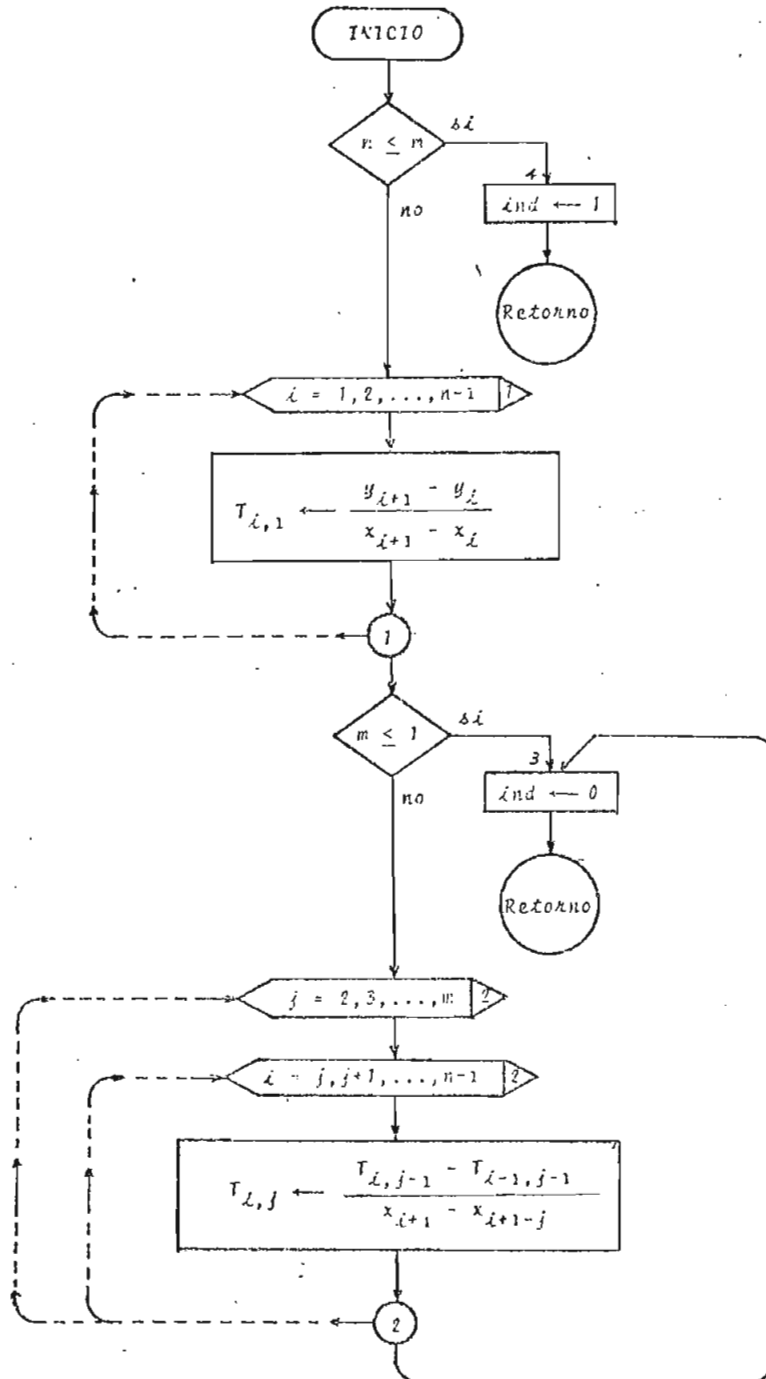
DIAGRAMAS DE FLUJO

PROGRAMA PRINCIPAL:



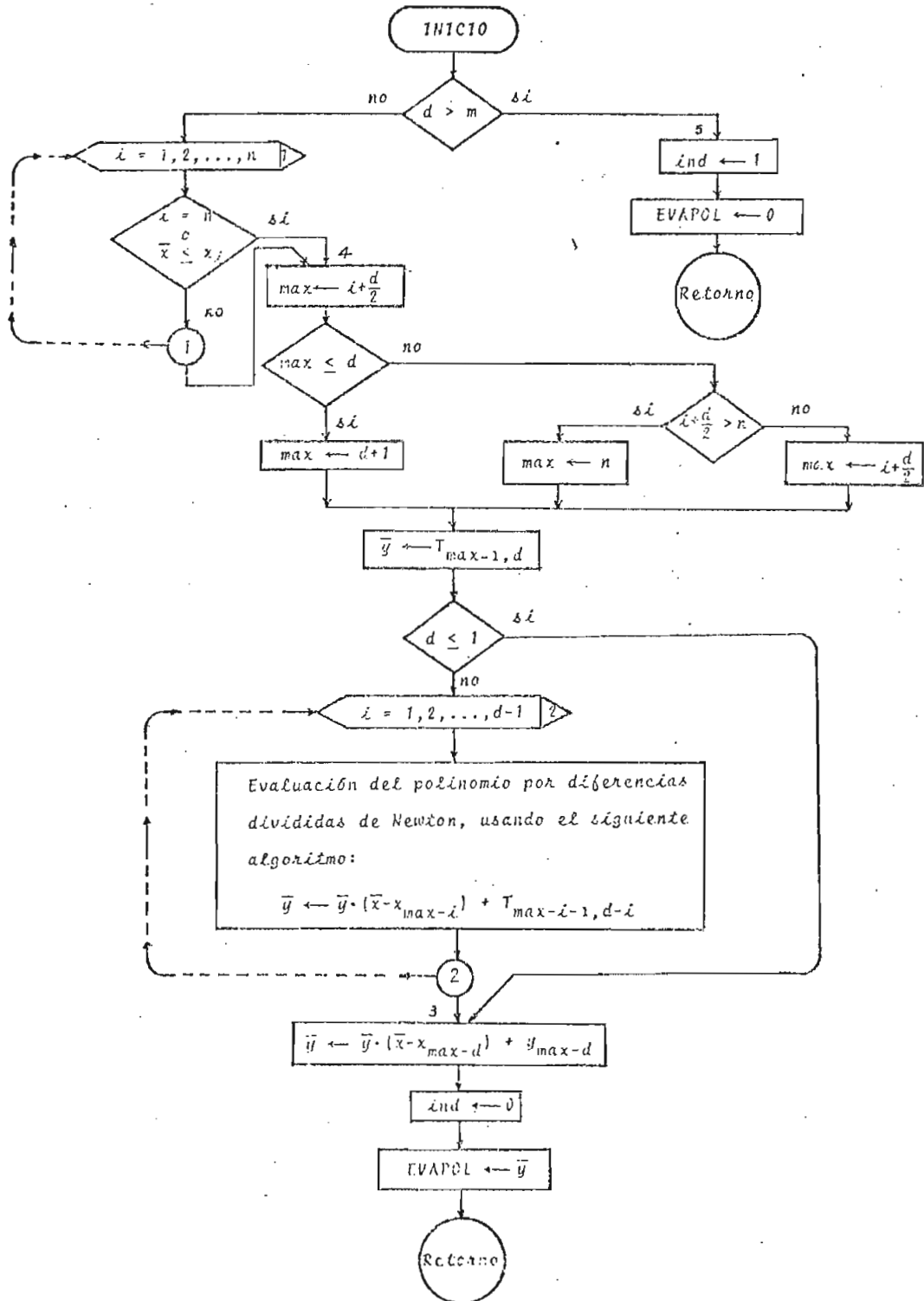
SUBROUTINA TABLAD:

(Argumentos: n, m, x, y, T, ind, k).



FUNCION EVAPOL:

(Argumentos: $m, n, d, \bar{x}, x, y, T, ind, k$).



CODIFICACION FORTRAN

LISTA DE VARIABLES PRINCIPALES

*Símbolos en
los Programas*

Definición

(Principal)

I,J,L	Subíndices i,j,l
NGRAD	Grado "d" del polinomio de colocación de Newton
N	Número de muestras tomadas de la variable x.
M	m, diferencia dividida de más alto orden que es calculada por TABLAD.
NM1	n-1
TABLA	Matriz de las diferencias divididas, T_{ij}
IND	Interruptor de cálculo: Es 1 si se encuentra inconsistencia en un argumento, de no ser así es 0.
VERVAL	Valor de $\cos \bar{x}$ calculado usando la función DCOS que forma parte de la biblioteca de la computadora.
X	Arreglo de los puntos bases, x_i ; $i = 1,2,$

...,n.

ARGX Argumento para interpolación, \bar{x} .

Y Arreglo de los valores funcionales, $y_i = f(x_i)$.

VALIN Valor interpolado, $\bar{y}(\bar{x})$.

(Subrutina TABLAD)

K Dimensión de las filas y columnas de la matriz T,k.

(Función EVAPOL)

K1 d-1

MAX Subíndice del último de los puntos bases que se usa para determinar el polinomio de colocación, max.

YR Variable usada en la evaluación del polinomio de colocación.

LISTADO DEL PROGRAMA

PROGRAMA PRINCIPAL:

```

C
C PROGRAMAMA PARA CALCULAR LA FUNCION F(X) = COS(X) MEDIANTE INTERPOLACION
C USANDO EL METODO DE DIFERENCIAS DIVIDIDAS DE NEWTON
C
C EL PROGRAMA LEE UN CONJUNTO DE N VALORES X(1)...X(N), CALCULA UN CONJUNTO
C DE VALORES Y(1)...Y(N) DE LA FORMA Y(I) = COS(X(I)), Y LLEGA LLAMA
C A LA SUBROUTINA TABLAD PARA QUE CALCULE TODAS LAS DIFERENCIAS DIVIDIDAS DE
C ORDEN N O MENOS. CON LAS DIFERENCIAS ALMACENADAS EN LA MATRIZ TABLA, EL
C PROGRAMA LEE VALORES PARA ARGX, ARGUMENTO DE INTERPOLACION, Y ARGF, EL
C GRADO DEL POLINOMIO DE COLOCACION QUE SE EVALLA CON LA FUNCION EVAPOL.
C EVAPOL CALCULA EL VALOR INTERPOLADO, VALIN, QUE SE LO COMPARA CON EL VALOR
C VERDADERO, VVERVAL = DCOS(ARGX).
0001      IMPLICIT REAL*8 (A-H,O-Z)
0002      DIMENSION X(20),Y(20),TABLA(20,20)
C
C LECTURA DE DATOS, CALCULO DE LOS VALORES Y ESCRITURA
C
0003      READ(1,11)N,M,(X(I),I=1,N)
0004      11 FORMAT(2I5,7F10.0/(8F10.0))
0005      GO 1 I=1,N
0006      1 Y(I)=DCOS(X(I))
0007      WRITE(3,22) (I,X(I),Y(I),I=1,N)
0008      22 FORMAT (1H1//1X*VALORES UTILIZADOS PARA EL CALCULO DE LA*,
C *Y TABLA DE DIFERENCIAS DIVIDIDAS**//11X*ARGUMENTO*2X*FUNCION*
C *//8X12,F8.4,F15.5)
C
C CALCULO Y ESCRITURA DE LAS DIFERENCIAS DIVIDIDAS
C
0009      CALL TABLAD(N,M,X,Y,TABLA,IND,20)
0010      IF(IND.EQ.1) STOP
0011      WRITE(3,33)M
0012      33 FORMAT (//1X*TABLA DE DIFERENCIAS DIVIDIDAS PARA P=*12)
0013      NM1=N-1
0014      DO 2 I=1,NM1
0015      L=I
0016      IF(I.GT.NM1)M
0017      2 WRITE(3,44)(TABLA(I,J),J=1,L)
0018      44 FORMAT (/1X(6(8X15.6)))
0019      WRITE(3,77)
0020      77 FORMAT (//1X*ARGUMENTO*2X*GRADO*2X*VALOR INTERPOLADO*2X*
C *VALOR VERDADERO*2X*IND/)
C
C LECTURA DE NGRAD Y ARGX, LLAMADA A EVAPOL PARA INTERPOLAR
C
0021      3 READ(1,55)END=4 INGRAD,ARGX
0022      55 FORMAT(12,F10.0)
0023      KDIM=20
0024      VALIN=EVAPOL(M,N,NGRAD,ARGX,X,Y,TABLA,IND,KDIM)
C
C CALCULO DEL VALOR VERDADERO DCOS(ARGX) Y ESCRITURA DE RESULTADOS
C
0025      VVERVAL=DCOS(ARGX)
0026      WRITE(3,66)ARGX,NGRAD,VALIN,VVERVAL,IND
0027      66 FORMAT(12X10.6,15.6X10.8,9X10.8,I7)
0028      GO TO 3
0029      4 STOP
0030      END

```

FUNCION EVAPOL:

```

0001      FUNCTION EVAPOL(M,N,NGRAD,ARGX,X,Y,TABLA,IND,K)
C
C SUBPROGRAMA PARA EVALUAR EL POLINOMIO DE COLOCACION POR DIFERENCIAS
C FINITAS DIVIDIDAS DE NEWTON.
C
C EVAPOL ASUME QUE X(1)...Y(N) ESTAN EN ORDEN ASCENDENTE, PRIMERO ESCRIBIEN
C EN EL ARGUMENTO X PARA DETERMINAR QUE ELEMENTO ES MAS PROXIMO (AQUEL) AL ARGU
C MENTO DE INTERPOLACION, ARG.
C
C LOS NGRAD + 1 PUNTOS MAS NECESARIOS PARA LA EVALUACION DEL POLINOMIO DE
C COLOCACION DE GRADO NGRAD SON CONTADOS ALREDEDOR DEL ELEMENTO ESCOGIDO CON
C EL MAS GRANDE SUPLENDO QUE SE OBTIENE, MAX. SE ASUME QUE LAS DIFERENCIAS
C DIVIDIDAS FUERON CALCULADAS YA POR LA SUBROUTINA TABLAD.
C
C MAX ES CONTADO PARA ASEGURARSE QUE TODOS LOS PUNTOS BASES SON UTILIZADOS
C CUANDO EL POLINOMIO DE COLOCACION HA SIDO EVALUADO, ESTE VALOR REFIERNA CUEN
C EL VALOR DE LA FUNCION. SI HAY INCONSISTENCIA EN LOS ARGUMENTOS IND = 1.
C
C DE NO SER ASI IND = 0 EN LA SALIDA
C
0002      IMPLICIT REAL*8 (A-H,C-Z)
0003      DIMENSION TABLA(M,K),X(N),Y(N)

```

FUNCION EVAPOL: (Continuación).

```

C
C CONTROL DE INCONSISTENCIA DEL ARGUMENTO
0004 IF(NGRAD.GT.M) GO TO 5
C
C ELEGIR EN EL ARREGLO X DEL ELEMENTO .GE. ARGX
0005 DO 1 I=1,N
0006 IF(ARGX.LT.X((I).CR.I.EC.N)) GO TO 4
0007 1 CONTINUE
C
C ADECUARSE QUE TODAS LAS DIFERENCIAS DIVIDIDAS REQUERIDAS ESTAN EN LA MATRIZ
C TABLA
0008 4 MAX=I+NGRAD/2
0009 IF(MAX.LE.NGRAD) MAX=NGRAD+1
0010 IF(MAX.GT.N) MAX=N
C
C CALCULO DEL VALOR INTERPOLADO
0011 YR=TABLA(MAX-1,NGRAD)
0012 IF(NGRAD.LE.1) GO TO 3
0013 XI = NGRAD - 1
0014 IC 2 I=1,K1
0015 2 YR=YR+(ARGX-X(MAX-I))*TABLA(MAX-I-1,NGRAD-I)
0016 3 YR=YR+(ARGX-X(MAX-NGRAD))*Y(MAX-NGRAD)
0017 INC=C
0018 EVAPOL=YR
0019 RETURN
0020 5 INC=1
0021 EVAPOL=0.
0022 RETURN
0023 END

```

SUBROUTINE TABLAD:

```

0001 SUBROUTINE TABLAD(N,M,X,Y,TABLA,INC,K)
C
C SUBPROGRAMA PARA CALCULAR LA TABLA DE DIFERENCIAS DIVIDIDAS
C
C LAS DIFERENCIAS DIVIDIDAS SON ALMACENADAS EN LA PORCION TRIANGULAR INFERIOR
C DE LAS M PRIMERAS COLUMNAS DE LAS N-1 FILAS DE LA MATRIZ TABLA. POR
C INCONSISTENCIA DE LOS ARGUMENTOS IND = 1. DE NO SER ASI INC = 0 EN LA SALIDA
0002 IMPLICIT REAL*8 (A-H,C-Z)
0003 DIMENSION TABLA(K,K),X(N),Y(N)
C
C CONTROL DE CONSISTENCIA DEL ARGUMENTO
0004 IF(N.LE.M) GO TO 4
C
C CALCULO DE LA PRIMERA DIFERENCIA
0005 NM1=N-1
0006 DO 1 I=1,NM1
0007 1 TABLA(I,1)=(Y(I+1)-Y(I))/(X(I+1)-X(I))
C
C CALCULO DE LAS DIFERENCIAS DE MAYOR ORDEN
0008 IF(M.LE.1) GO TO 3
0009 DO 2 J=2,M
0010 DO 2 I=J,NM1
0011 2 TABLA(I,J)=(TABLA(I,J-1)-TABLA(I-1,J-1))/(X(I+1)-X(I+1-J))
0012 3 INC=C
0013 RETURN
0014 4 INC=1
0015 RETURN
0016 END

```

ENTRADA DE DATOS:

Las tarjetas deben perforarse como sigue:

- 1.- En la primera tarjeta, el valor de M desde la columna 1 hasta la 5; el valor de N desde la columna 6 hasta la 10; -Estos dos valores deben ser perforados de tal manera que la última columna de su respectivo campo vaya las unidades del número, en la penúltima las decenas, etc. Tanto M como N son variables enteras.- En las 70 columnas restantes perforar los siete primeros valores de x, abarcando cada valor un campo de 10 columnas, en el cual se debe perforar el respectivo número incluido el punto decimal, (215,7F10.0).
- 2.- Para el caso que N sea mayor que 7, pues representa el número de valores que tiene el arreglo x, habrá una segunda y hasta una tercera tarjeta en las que se perfora dichos valores. En cada tarjeta se puede perforar máximo 8 valores de x; ocupando cada valor 10 columnas incluyendo de igual forma que en el numeral 1 el punto decimal ya que se debe perforarlo. (8F10.0).
- 3.- Una vez que se ha perforado todos los valores del arreglo x, en una nueva tarjeta perforar el grado del polinomio, NGRAD, en las columnas 1 y 2, -variable entera- de tal manera que las unidades estén en la columna 2; y en

las columnas 3 a 12 el valor para el cual se desea estimar el valor de la función coseno por interpolación, ARGX. (I2,F10.0).

- 4.- En general se va a interpolar varias veces; entonces para otros pares de valores, NGRAD y ARGX, repetir desde el numeral 3.

M representa el orden más alto de la o de las diferencias divididas que se desean obtener.

N es el número de puntos bases tomados de la variable x que deben estar ordenados de menor a mayor. El valor máximo de N es 20 y debe ser siempre mayor que M.

A P E N D I C E 2

Aquí mencionaremos algunas características de las diferencias finitas divididas:

- Las diferencias divididas están relacionadas con la derivada del orden correspondiente por

$$f [x, x_{n-1}, \dots, x_0] = \frac{f^{(n)}(\epsilon)}{n!} \quad \epsilon \text{ en } (x, x_{n-1}, \dots, x_0) \quad (A2.1)$$

- "La propiedad de SIMETRIA de las diferencias divididas establece que tales diferencias son invariantes bajo todas las permutaciones de los argumentos x_i , previsto que los valores $f(x_i)$ sean permutados de la misma manera" (2).

Así por ejemplo:

$$f [x_1, x_0] = f [x_0, x_1]$$

Mediante manipulación algébrica de las diferencias de más bajo orden y por inducción se llega a una forma de representación de la diferencia de enésimo orden llamada SIMETRICA

$$f [x_n, x_{n-1}, \dots, x_0] = \frac{f(x_n)}{(x_n - x_{n-1})(x_n - x_{n-2}) \dots (x_n - x_0)} + \\ + \frac{f(x_{n-1})}{(x_{n-1} - x_n)(x_{n-1} - x_{n-2}) \dots (x_{n-1} - x_0)} +$$

$$+ \dots$$

$$+ \frac{f(x_0)}{(x_0-x_n)(x_0-x_{n-1})\dots(x_0-x_1)}$$

La representación simétrica puede ser escrita de una forma más compacta.

$$f [x_n, x_{n-1}, \dots, x_0] = \sum_{i=0}^n \frac{f(x_i)}{\prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j)} \quad (A2.2)$$

Una consecuencia de la propiedad de simetría es que en la tabla de diferencias divididas el orden de los subíndices como el de los puntos bases x_i no importa. Las tablas A2.1 y A2.2 que se indican a continuación se obtuvieron al cambiar el orden de los subíndices y de los puntos bases x_i en la tabla 2.2 respectivamente, dada en el capítulo II.

TABLA A2.1 TABLA DE DIFERENCIAS FINITAS DIVIDIDAS CON LOS SUBINDICES DE LOS PUNTOS BASES INTERCAMBIADOS.

i	x_i	$f(x_i)$	$f_1 [\]$	$f_2 [\]$	$f_3 [\]$
3	0	-5,			
1	1	1,	6,		
0	3	25,	12,	2,	
2	4	55,	30,	6,	1,

El polinomio de colocación de tercer grado que se obtiene de la tabla A2.1 está dado por:

$$P_3(x) = f[x_0] + (x-x_0) f[x_1, x_0] + (x-x_0)(x-x_1) f[x_2, x_1, x_0] + (x-x_0)(x-x_1)(x-x_2) f[x_3, x_2, x_1, x_0]$$

ó (siguiendo el camino señalado en la tabla)

$$P_3(x) = 25 + (x-3) 12 + (x-3)(x-1) 6 + (x-3)(x-1)(x-4) = x^3 - 2x^2 + 7x - 5$$

TABLA A2.2 TABLA DE DIFERENCIAS FINITAS DIVIDIDAS CON LOS VALORES FUNCIONALES EN ORDEN ARBITRARIO

i	x_i	$f(x_i)$	$f_1[\]$	$f_2[\]$	$f_3[\]$
0	3,	25,			
1	0,	-5,	10,		
2	4,	55,	15,	5,	
3	1,	1,	18,	3,	1,

Siguiendo la ruta indicada en la tabla A2.2 se obtiene el polinomio de grado tres $P_3(x)$ que se indica a continuación:

$$P_3(x) = 25 + (x-3) 10 + (x-3)(x-0) 5 + \\ + (x-3)(x-0)(x-4) 1 = x^3 - 2x^2 + 7x - 5$$

Como se puede apreciar este polinomio es el mismo que se obtuvo tanto de la tabla A2.1 como de la tabla A2.2 .

Cuando se usan todos los $n+1$ pares de valores $(x_i, f(x_i))$ es indistinta la ruta que se siga en la tabla de diferencias divididas ya que existe uno y solamente un polinomio de grado n o menos que asume los $n+1$ valores de la función. Por lo que no es necesario conocer todas las diferencias divididas, sino que basta las n diferencias (una por cada orden) de cualquier ruta. Por simplicidad se suele escoger la ruta que se indica en la tabla A2.2. Si bien es cierto que para conocer estas diferencias se debe conocer las restantes, no obstante para el caso de una estimación por medio de una computadora digital resulta más cómodo trabajar con n diferencias en lugar que con $n(n+1)/2$.

A P E N D I C E 3

PLANTEAMIENTO

Escribir una función llamada DIRY que evalúe para el valor \bar{y} el polinomio de colocación de Lagrange de grado d que pasa por los puntos $(y_{\min}, z_{\min}), (y_{\min+1}, z_{\min+1}), \dots, (y_{\max}, z_{\max})$; $d = \max - \min$, siendo "y" la variable independiente y "z" la variable dependiente. A DIRY se le adjudica el valor interpolado $\bar{z}(\bar{y})$.

METODO DE SOLUCION.

Utilizando la ecuación (2.12) para el polinomio de colocación de Lagrange se tiene:

$$\bar{z}(\bar{y}) = \sum_{i=\min}^{\max} L_i(\bar{y}) \cdot z_i \tag{A3.1}$$

donde

$$L_i(\bar{y}) = \prod_{\substack{j=\min \\ j \neq i}}^{\max} \frac{\bar{y} - y_j}{y_i - y_j} \tag{A3.2}$$

$$i = \min, \min+1, \dots, \max$$

Se ahorra algunos cálculos si (A3.2) se le escribe de la for

ma:

$$L_i(\bar{y}) = \frac{c/(\bar{y} - y_i)}{\max_{\substack{j=\min \\ j \neq i}} |y_i - y_j|} \quad (A3.3)$$

$$i = \min, \min+1, \dots, \min+d; \bar{y} \neq y_i$$

donde

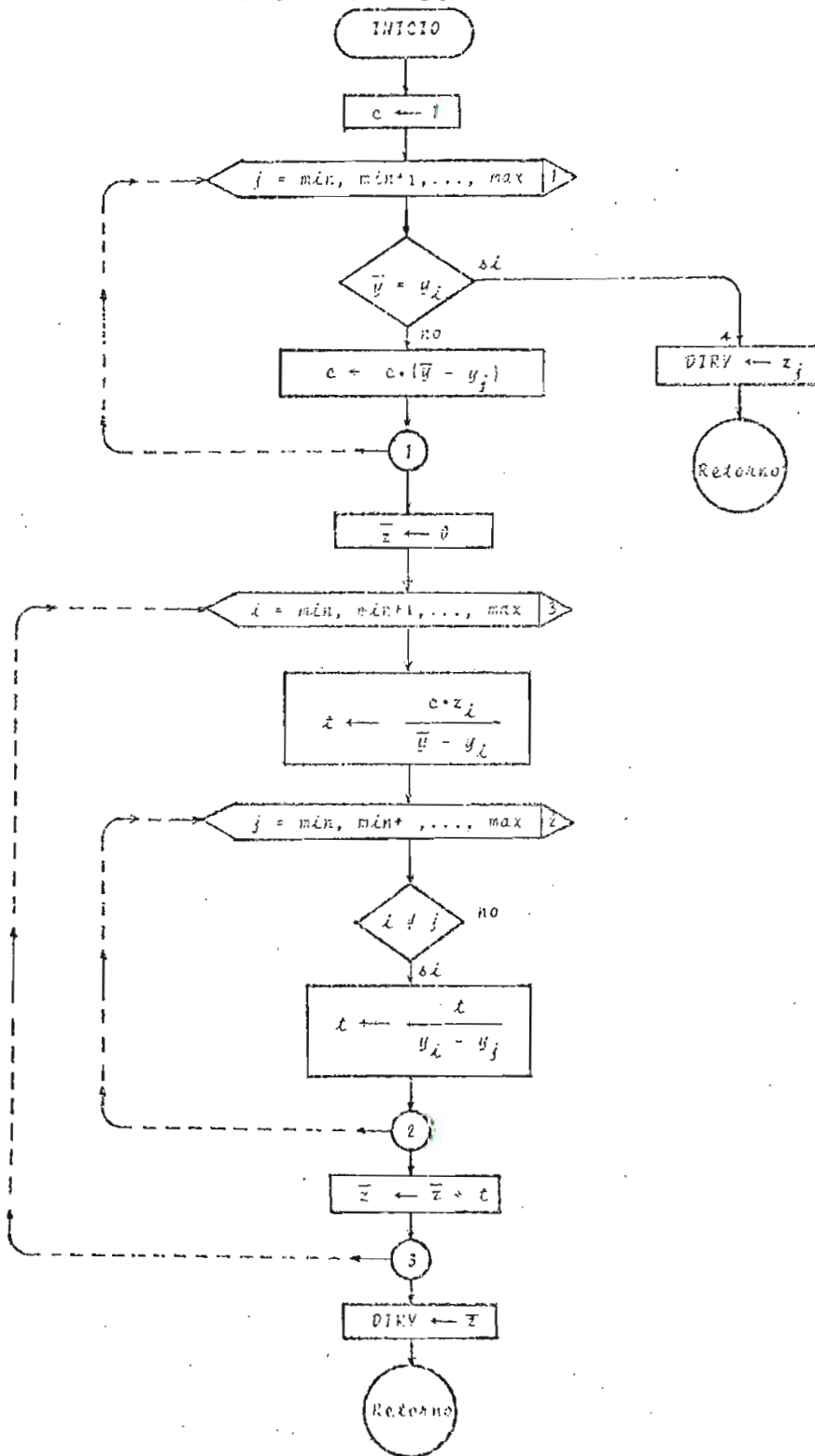
$$c = \max_{j=\min} | \bar{y} - y_j | \quad (A3.4)$$

La restricción $\bar{y} \neq y_i$ en (A3.3) no causa dificultades, puesto que si $\bar{y} = y_i$, el valor interpolado $z(\bar{y})$ se conoce y es igual a z_i y no se requiere cálculos adicionales.

DIAGRAMA DE FLUJO

FUNCION DIRV:

(Argumentos: max, min, y, z, argy, n).



CODIFICACION FORTRAN

LISTA DE VARIABLES PRINCIPALES

<i>Símbolos del Subprograma.</i>	<i>Definición</i>
I,J	Subíndices i, j
N	Número de pares de valores (y_i, z_i) ; $i = 1, 2, \dots, n$
Y	Arreglo de los puntos bases y_i
FX	Arreglo de las muestras z_i
MIN	Subíndice inferior de los puntos bases usados para determinar el polinomio de colocación.
MAX	Subíndice superior de los puntos bases usados para determinar el polinomio de colocación de grado $\text{max} - \text{min} < n$
C	El factor c (ver (A3.4)).
T	t , variable que asume sucesivamente los valores $L_i(\bar{y}) z_i$, en (A3.1).
FXY	Valor interpolado $\bar{z}(y)$.

LISTADO DEL SUBPROGRAMA

```
0001          FUNCTION DIRY (MAX,MIN,Y,FX,ARGY,N)
C
C SUBPROGRAMA PARA INTERPOLACION EN DOS DIMENSIONES UTILIZANDO EL POLINOMIO DE
C COLOCACION DE LAGRANGE.
C
C EL POLINOMIO A EVALUAR DE GRADO NGRAD PARA EL ARGUMENTO ARGY
C USA LOS DATOS Y(MIN) ... Y(MAX) Y FX(MIN) ... FX(MAX), DONDE
C MAX = MIN + NGRAD. SE ASUME QUE LOS VALORES DE Y ESTAN EN ORDEN ASCENDENTE.
C Y ES UNA VARIABLE QUE CONTIENE SUCCESIVAMENTE CADA TERMINO DE LA
C FORMULA DE LAGRANGE. EL VALOR FINAL DE FXY ES EL VALOR
C INTERPOLADO. PARA UNA DESCRIPCION DE LA VARIABLE C REFERIRSE A LA
C INFORMACION TEORICA.
C
0002          IMPLICIT REAL*8 (A-H,C-Z)
0003          DIMENSION Y(N),FX(N)
C
C CALCULO DEL VALOR C
C
0004          C=1.00
0005          JJ=MIN-1
0006          DO 1 J=MIN,MAX
0007             JJ=JJ+1
0008             IF(ARGY.EQ.Y(J)) GO TO 4
0009             C=C*(ARGY-Y(J))
C
C EVALUACION DEL POLINOMIO DE COLOCACION
C
0010          FXY=0.00
0011          DO 3 I=MIN,MAX
0012             T=C*FX(I)/(ARGY-Y(I))
0013             DO 2 J=MIN,MAX
0014                 IF(I.EQ.J) GO TO 2
0015                 T=T/(Y(I)-Y(J))
0016             2 CONTINUE
0017             3 FXY=FXY+T
0018             DIRY=FX(J)
0019             RETURN
0020         4 DIRY=FX(JJ)
0021             RETURN
0022             END
```


A P E N D I C E 4

Los polinomios de colocación spline cúbicos como se explica en 2.6. , son tales que para una función $f(x)$ continua y diferenciable, conocida en forma discreta en un intervalo (a,b) para $n+1$ valores de x ordenados en forma ascendente, es decir $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$, se tienen n polinomios de colocación de tercer grado, $P_{3,i}(x)$, uno por cada subintervalo (x_i, x_{i+1}) . Además el polinomio de colocación total aproximante de $f(x)$ debe ser continuo en su primera y segunda derivada; o sea que $P'_{3,i}(x_i) = P'_{3,i-1}(x_i)$ y $P''_{3,i}(x_i) = P''_{3,i-1}(x_i)$, cumpliéndose con esto las condiciones de continuidad en todo el intervalo (a,b) por ser polinomios de tercer grado. En la figura A4.1 se indica la función aproximante por polinomios de colocación spline cúbicos, en línea

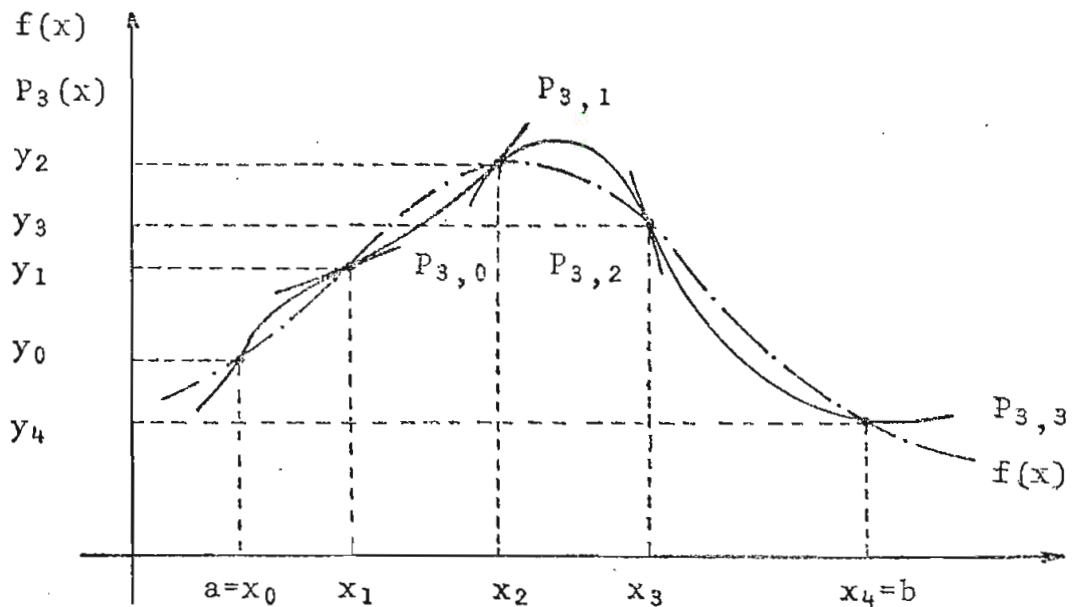


FIGURA A4.1 APROXIMACIÓN POLINOMIAL SPLINE CUBICA.

continua, para cinco valores de la variable independiente x ($n = 4$), en donde $y_i = f(x_i)$.

La segunda derivada de $P_{3,i}(x)$ es un polinomio de primer grado que pasa por los puntos $P''_{3,i}(x_i)$ y $P''_{3,i}(x_{i+1})$; por lo tanto usando la ecuación de una línea recta se tiene:

$$\frac{P''_{3,i}(x) - P''_{3,i}(x_i)}{x - x_i} = \frac{P''_{3,i}(x_{i+1}) - P''_{3,i}(x_i)}{x_{i+1} - x_i}$$

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

$$P''_{3,i}(x) = \frac{(x_{i+1} - x)}{h_i} \cdot P''_{3,i}(x_i) + \frac{(x - x_i)}{h_i} \cdot P''_{3,i}(x_{i+1}) \quad (\text{A4.1})$$

Integrando dos veces (A4.1) con respecto a x

$$P_{3,i}(x) = \frac{(x_{i+1} - x)^3}{6 h_i} P''_{3,i}(x_i) + \frac{(x - x_i)^3}{6 h_i} P''_{3,i}(x_{i+1}) + k_1 x + k_2 \quad (\text{A4.2})$$

Satisfaciendo el criterio (2.2) en (A4.2)

$$P_{3,i}(x_i) = y_i = \frac{(x_{i+1} - x_i)^3}{6 h_i} P''_{3,i}(x_i) + k_1 x_i + k_2 \quad (\text{A4.3})$$

$$P_{3,i}(x_{i+1}) = y_{i+1} = \frac{(x_{i+1} - x_i)^3}{6 h_i} P''_{3,i}(x_{i+1}) + k_1 x_{i+1} + k_2 \quad (A4.4)$$

Manipulando algebraicamente (A4.3) y (A4.4) se determina

$$k_1 = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6} \cdot \left(P''_{3,i}(x_{i+1}) - P''_{3,i}(x_i) \right) \quad (A4.5)$$

$$k_2 = \frac{1}{h_i} \cdot \left(x_{i+1} y_i + x_i y_{i+1} \right) - \frac{h_i}{6} \cdot \left(x_{i+1} P''_{3,i}(x_i) - x_i P''_{3,i}(x_{i+1}) \right) \quad (A4.6)$$

Reemplazando (A4.5) y (A4.6) en (A4.2) se llega a la forma usual del polinomio de colocación spline cúbico para el sub-intervalo (x_i, x_{i+1}) .

$$P_{3,i}(x) = \frac{(x_{i+1} - x)^3}{6 h_i} \cdot P''_{3,i}(x_i) + \frac{(x - x_i)^3}{6 h_i} \cdot P''_{3,i}(x_{i+1}) + (x - x_i) \cdot \left(\frac{y_{i+1}}{h_i} - \frac{h_i P''_{3,i}(x_{i+1})}{6} \right) + (x_{i+1} - x) \cdot \left(\frac{y_i}{h_i} - \frac{h_i P''_{3,i}(x_i)}{6} \right) \quad (A4.7)$$

$$i = 0, 1, 2, \dots, n-2, n-1$$

En (A4.7) para evaluar $P_{3,i}(x)$ para un valor x cualquiera dentro del intervalo, es necesario conocer las segundas derivadas. Con esta finalidad derivamos (A4.7) con respecto a la variable x .

$$\begin{aligned} \frac{d}{dx} (P_{3,i}(x)) &= P'_{3,i}(x) \\ P'_{3,i}(x) &= - \frac{(x_{i+1} - x)^2}{2 h_i} \cdot P''_{3,i}(x_i) + \frac{(x - x_i)^2}{2 h_i} \cdot P''_{3,i}(x_{i+1}) + \\ &+ \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6} \cdot \left(P''_{3,i}(x_{i+1}) - P''_{3,i}(x_i) \right) \end{aligned} \tag{A4.8}$$

Tomando en cuenta las primeras derivadas de dos polinomios consecutivos, $P_{3,i-1}(x)$ y $P_{3,i}(x)$; $i = 1, 2, \dots, n-1$; evaluados en el punto x_i .

$$\begin{aligned} P'_{3,i-1}(x_i) &= \frac{h_{i-1}}{2} \cdot P''_{3,i-1}(x_i) + \frac{y_i - y_{i-1}}{h_{i-1}} - \\ &- \frac{h_{i-1}}{6} \cdot \left(P''_{3,i-1}(x_i) - P''_{3,i-1}(x_{i-1}) \right) \end{aligned} \tag{A4.9}$$

$$\begin{aligned}
 P'_{3,i}(x_i) &= -\frac{h_i}{2} \cdot P''_{3,i}(x_i) + \frac{y_{i+1} - y_i}{h_i} \\
 &\quad - \frac{h_i}{6} \cdot \left(P''_{3,i}(x_{i+1}) - P''_{3,i}(x_i) \right) \quad (A4.10)
 \end{aligned}$$

Satisfaciendo la condición de continuidad, (A4.9) = (A4.10), se llega a

$$\begin{aligned}
 \frac{h_{i-1}}{h_i} P''_{3,i-1}(x_{i-1}) + \frac{2(h_i + h_{i-1})}{h_i} P''_{3,i}(x_i) + P''_{3,i}(x_{i+1}) &= \\
 &= \frac{6}{h_i} \cdot \left(\frac{(y_{i+1} - y_i)}{h_i} - \frac{(y_i - y_{i-1})}{h_{i-1}} \right) \\
 & \quad (A4.11)
 \end{aligned}$$

$$i = 1, 2, \dots, n-1$$

Habr a pues $n-1$ ecuaciones de la forma (A4.11), pero existen $n+1$ segundas derivadas, $P''_{3,i}(x_i)$; $i = 0, 1, \dots, n$. La resoluci n del sistema de ecuaciones simult neas lineales ser a im-
 posible, pues se tienen $n+1$ inc gnitas; entonces se ve en la necesidad de imponerse dos condiciones y as  se puede decir que $P''_{3,0}(x_0)$ y $P''_{3,n}(x_n)$ son valores conocidos. Escribiendo (A4.11) para $i = 1, 2, \dots, n-2, n-1$, se tiene

$$\begin{aligned}
 i = 1 & : a_1 P_{3,0}''(x_0) + b_1 P_{3,1}''(x_1) + P_{3,1}''(x_2) & = c_1 \\
 i = 2 & : a_2 P_{3,1}''(x_1) + b_2 P_{3,1}''(x_2) + P_{3,2}''(x_3) & = c_2 \\
 & \vdots & \vdots \\
 i = n-2 & : a_{n-2} P_{3,n-3}''(x_{n-3}) + b_{n-2} P_{3,n-2}''(x_{n-2}) + P_{3,n-2}''(x_{n-1}) & = c_{n-2} \\
 i = n-1 & : a_{n-1} P_{3,n-2}''(x_{n-2}) + b_{n-1} P_{3,n-1}''(x_{n-1}) + P_{3,n-1}''(x_n) & = c_{n-1}
 \end{aligned}$$

Donde

$$a_i = \frac{h_{i-1}}{h_i} \qquad b_i = \frac{2(h_i + h_{i-1})}{h_i}$$

$$c_i = \frac{6}{h_i} \cdot \left(\frac{(y_{i+1} - y_i)}{h_i} - \frac{(y_i - y_{i-1})}{h_{i-1}} \right)$$

$$i = 1, 2, \dots, n-1$$

Satisfaciendo la condición de continuidad para las segundas derivadas, $P_{3,i-1}''(x_i) = P_{3,i}''(x_i)$; $i = 1, 2, \dots, n-1$, y escribiendo en forma matricial el sistema de ecuaciones se tiene:

$$\begin{bmatrix}
 a_1 & b_1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
 0 & a_2 & b_2 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\
 \vdots & & & & & & & & & \vdots \\
 0 & 0 & 0 & 0 & 0 & \dots & a_{n-2} & b_{n-2} & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & \dots & 0 & a_{n-1} & b_{n-1} & 1
 \end{bmatrix}$$

$$\begin{bmatrix} P''_{3,0}(x_0) \\ P''_{3,1}(x_1) \\ P''_{3,1}(x_2) \\ P''_{3,2}(x_3) \\ \cdot \\ \cdot \\ \cdot \\ P''_{3,n-2}(x_{n-1}) \\ P''_{3,n-1}(x_n) \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ c_{n-2} \\ c_{n-1} \end{bmatrix}$$

Si $P''_{3,0}(x_0)$ y $P''_{3,n-1}(x_n)$ son conocidas, entonces el sistema de ecuaciones simultáneas lineales queda de la forma:

$$\begin{bmatrix} b_1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_2 & b_2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & 1 & \dots & 0 & 0 & 0 \\ \cdot & & & & & & & \cdot \\ \cdot & & & & & & & \cdot \\ \cdot & & & & & & & \cdot \\ 0 & 0 & 0 & 0 & \dots & a_{n-3} & b_{n-2} & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & a_{n-2} & b_{n-1} \end{bmatrix}$$

$$\begin{bmatrix} P''_{3,1}(x_1) \\ P''_{3,1}(x_2) \\ P''_{3,2}(x_3) \\ \cdot \\ \cdot \\ \cdot \\ P''_{3,n-3}(x_{n-2}) \\ P''_{3,n-2}(x_{n-1}) \end{bmatrix} = \begin{bmatrix} c_1 - a_1 P''_{3,0}(x_0) \\ c_2 \\ c_3 \\ \cdot \\ \cdot \\ \cdot \\ c_{n-2} \\ c_{n-1} - a_{n-1} P''_{3,n-1}(x_n) \end{bmatrix}$$

Pudiendo ser resuelto ya el sistema de ecuaciones simultáneas lineales (número de incógnitas igual al número de ecuaciones) por los métodos convencionales conocidos.

A P E N D I C E 5

Este Apéndice se refiere al propósito, a la forma de uso y a la descripción de los argumentos de los subprogramas de mayor importancia desarrollados en este trabajo.

SUBROUTINA ENITØ1.

a.- PROPOSITO.-

Calcula mediante interpolación directa o inversa el valor de una variable, conociendo las otras dos variables de una función tridimensional; siendo esta función conocida en forma discreta. Se usa como funciones aproximantes los polinomios de colocación spline cúbicos.

b.- FORMA DE USO.-

CALL ENITØ1 (M, N, VA, VG, IA, ARG1, ARG2, ARG3, IDE,
 NUM, IND, JI, MM)

c.- DESCRIPCION DE LOS ARGUMENTOS.

M: Variable entera que contiene el número de puntos bases de la variable independiente x. El mínimo valor de M es 3 y el máximo 50.

N: Variable entera que contiene el número de

puntos bases tomados de la variable independiente y. Pudiendo ser N, mínimo 3 y máximo 50.

VA: Arreglo real de 8 bytes que contiene los M puntos bases de la variable x; los mismos que deben estar ordenados en forma ascendente.

VG: Arreglo real de 8 bytes que contiene los N puntos bases de la variable y. También estos puntos bases deben estar en orden ascendente.

IA: Arreglo bidimensional real de 8 bytes, que contiene las $M \times N$ muestras de la función o variable dependiente z. Estas muestras se deben ordenar de acuerdo a la tabla 2.5.

ARG1,ARG2: Variables reales de 8 bytes, que contienen los valores de las dos variables, para las cuales se va a encontrar la tercera.

ARG3: Variable real de 8 bytes, que al retorno de la subrutina al programa principal que lo requiera, contiene el valor de la tercera variable que es determinada por interpolación. Para el caso de extrapolación, como no se puede realizar, a ARG3 se le asigna directamente el valor cero.

IDE: Variable entera que informa a la subrutina cuál de las tres variables se desea que cal

cule. En la Tabla A5.1 constan los tres valores que puede tomar IDE. Esto es de acuerdo a la variable que se desee calcular.

TABLA A5.1

IDE	ARG1	ARG2	ARG3
1	\bar{x}	\bar{y}	$\bar{z} (\bar{x}, \bar{y})$
2	\bar{x}	\bar{z}	$\bar{y} (\bar{x}, \bar{z})$
3	\bar{y}	\bar{z}	$\bar{x} (\bar{y}, \bar{z})$

IND, JI: Variables enteras que indican la existencia de alguna anomalía en la ejecución de la subrutina. No se garantiza el resultado. Cuando IND = -1 y JI \neq 0 se debe suspender el programa que está utilizando a ENITØ1 y cuando IND = -1 y JI = 0 significa que no se puede encontrar el valor interpolado (ARG3) para los valores ARG1 y ARG2 dados.

NUM: Variable entera que debe tener asignada el valor cero cuando se llama por vez primera a la subrutina, para un mismo conjunto de muestras. (La subrutina ENITØ1 va incrementando a NUM en una unidad cada vez que es llamada).

MM: Variable entera cuyo valor debe ser igual al dimensionamiento del número de filas que

en el programa principal tiene el arreglo IA.

Todos los argumentos de entrada (M, N, VA, VG, IA, ARG1, ARG2, NUM, MM), a excepción de NUM no sufren ninguna alteración en la subrutina.

SUBROUTINA ENITØ2.

a.- PROPOSITO:

Calcula las n segundas derivadas, $P_{3,i}''(x)$, de los $n-1$ polinomios de colocación spline cúbicos, $P_{3,i}(x)$; $i = 1, 2, \dots, n-1$, que son necesarias para evaluar dichos polinomios.

b.- FORMA DE USO:

CALL ENITØ2 (N, X, Y, DER, IND).

c.- DESCRIPCION DE LOS ARGUMENTOS:

N: Variable entera que contiene el número de puntos bases de la variable independiente x . El valor mínimo que puede ser N es 3.

X: Arreglo real de 8 bytes que contiene los N puntos bases de la variable independiente x , los mismos que deben estar ordenados en forma ascendente.

Y: Arreglo real de 8 bytes que contiene las muestras de la variable independiente y. Se debe tomar en cuenta que la muestra y_i es función del punto base x_i ($y_i = f(x_i)$; $i = 1, 2, \dots, n$).

DER: Arreglo real de 8 bytes, que al retorno de ENITØ2 contiene las n segundas derivadas, $P''_{3,i}(x)$; $i = 1, 2, \dots, n$, de los $n-1$ polinomios spline cúbicos.

IND: Variable entera que informa sobre el buen desenvolvimiento del subprograma. Cuando $IND = -1$ existe una anomalía en el subprograma y no se garantiza los resultados. En cambio cuando $IND = 0$ todo es normal.

El contenido de los argumentos de entrada (N, X, Y) no es alterado por la subrutina.

FUNCION ENITØ4.

a.- PROPOSITO:

Evalúa el polinomio de colocación spline cúbico, $P_{3,i}(x)$, $i = 1, 2, \dots, n$, para todo x comprendido dentro del intervalo en el cual es conocida $f(x)$ en forma discreta, función que es aproximada mediante polinomios spline cúbicos.

b.- FORMA DE USO:

YR = ENITØ4 (N, X, Y, ARGX, DER).

YR debe ser una variable real de 8 bytes, ya que ENITØ4 es de este tipo de variable.

c.- DESCRIPCION DE LOS ARGUMENTOS:

N,X,Y,DER: La descripción de estos argumentos es la misma que se hizo para los argumentos de la subrutina ENITØ2 con igual nombre.

ARGX: Variable real de 8 bytes que contiene el valor, \bar{x} , para el cual se desea conocer por aproximación spline cúbica, el valor de la función $\bar{y} = f(\bar{x})$.

ENITØ4: Nombre del subprograma de función, al que se le asigna el valor evaluado del polinomio spline cúbico, $P_{3,i}(\bar{x})$, para el argumento, \bar{x} . Cuando \bar{x} , está fuera del intervalo de interpolación se asigna a ENITØ4 el valor cero; pues no se puede extrapolar.

Ninguno de los argumentos de ENITØ4 sufre alteración alguna durante el proceso de cálculo. También este subprograma de función asume que las segundas derivadas $P_{3,i}''(x)$ fueron ya calculados por ENITØ2.

FUNCION DIRXY.

a.- PROPOSITO:

Subprograma de función que calcula, mediante interpolación directa, el valor de una variable. z , que es función de dos variables independientes, x y y . (Interpolación tridimensional). La función $f(x,y)$ es conocida de una manera discreta mediante muestras, $z_{ij} = f(x_i, y_j)$; $i = 1, 2, \dots, m$ y $j = 1, 2, \dots, n$. Se usa como función aproximante los polinomios de colocación de Lagrange.

b.- FORMA DE USO:

ZR = DIRXY (M, N, X, Y, FXY, ARGX, ARGY, NGRAD, IND).

ZR variable real de doble precisión, pues DIRXY es de do
ble precisión.

c.- DESCRIPCION DE LOS ARGUMENTOS:

M: Variable entera que contiene el número de puntos bases tomados de la variable independiente x . N como máximo puede ser 50 y como mínimo 2.

N: Variable entera que contiene el número de puntos bases tomados de la variable independiente y . Así mismo el valor máximo es 50 y el valor mí
nimo 2.

X: Arreglo de 8 bytes que contiene los M puntos bases que se posee de la variable independiente x. Deben estar ordenados de tal forma que: $X(1) < X(2) < \dots < X(M)$ (Orden ascendente).

Y: Arreglo de 8 bytes que contiene los N puntos bases de la variable independiente y. También deben estar ordenados en forma ascendente.

FXY: Arreglo bidimensional de 8 bytes que contiene las $M \times N$ muestras de la función o de la variable dependiente z. Debiendo ordenarse estas muestras de acuerdo a la Tabla 2.4.

ARGX, ARGY: Variables reales de 8 bytes que contienen los valores de las variables \bar{x} y \bar{y} respectivamente, para los cuales se desea conocer el valor de la función $\bar{z} = f(\bar{x}, \bar{y})$ por interpolación.

NGRAD: Grado de los polinomios de Lagrange a evaluarse. NGRAD debe ser menor que M o N.

IND: Variable entera utilizada como control del buen desenvolvimiento del subprograma. Cuando existe inconsistencia con los argumentos M, N y NGRAD $IND = -1$; en caso contrario $IND = 0$.

DIRXY: Variable real de 8 bytes que corresponde al nombre de la función y a la que se le asig-

na el valor determinado de la variable dependiente, mediante la evaluación de los polinomios de Lagrange, $\bar{z}(\bar{x}, \bar{y})$. El valor interpolado es función de ARGX, ARGY, NGRAD y del conjunto de muestras. Para el caso que NGRAD sea mayor o igual que M o N se asigna directamente a DIRXY el valor cero.

Solamente el argumento IND sufre alteración durante el proceso de cálculo del subprograma de función, DIRXY permite extrapolar, pudiendo ser el valor calculado bastante aproximado o no al valor real.

FUNCION DIRY:

a.- PROPOSITO:

Evalúa el polinomio de colocación de Lagrange

$$P_d(x) = \sum_{i=\min}^{\max} L_i(x) f(x_i); ; \left\{ \begin{array}{l} \text{máx} - \text{min} = d + 1 \\ \text{min} \geq 1 \\ \text{máx} \leq n \end{array} \right\} \quad d < n$$

d = grado del polinomio.

n = número de muestras tomadas.

que se utiliza como función aproximante de una función $f(x)$, conocida en forma discreta para n valores x_i ; $i = 1, 2, \dots, n$, de la variable independiente x.

b.- FORMA DE USO:

YR = DIRY (MAX, MIN, Y, FX, ARGY, N).

YR debe ser una variable real de 8 bytes ya que DIRY es también de 8 bytes.

c.- DESCRIPCION DE LOS ARGUMENTOS:

MAX: Variable entera que contiene el subíndice máximo del par de valores $(x_i; f(x_i))$ que interviene en la evaluación del polinomio de Lagrange.

MIN: Variable entera que contiene el subíndice mínimo del par de valores $(x_i; f(x_i))$ que interviene en la evaluación del polinomio de Lagrange.

Y: Arreglo real de 8 bytes que contiene los puntos bases $x_i; i = 1, 2, \dots, n$, de la variable independiente x , ordenados en forma ascendente.

FX: Arreglo real de 8 bytes que contiene las muestras $f(x_i)$, tomadas de la variable dependiente o función $f(x)$.

ARGY: Variable real de 8 bytes que contiene el valor \bar{x} , para el cual se desea obtener una estimación de la función, $f(x)$, mediante interpolación.

N: Variable entera que contiene el número de

pares de valores $(x_i; f(x_i))$; $i = 1, 2, \dots, n$
que se han tomado como muestras.

DIRY: Nombre del subprograma de función, es una
variable de 8 bytes al que se le asigna el
valor evaluado del polinomio de Lagrange.

Ninguno de los argumentos sufre alteración durante el
proceso de cálculo de DIRY.

FUNCION MINMON.

a.- PROPOSITO:

Calcula el subíndice del primer par ordenado $(x_{\min}, f(x_{\min}))$ que debe intervenir en la evaluación del polinomio de colocación de Lagrange de grado d , $P_d(x)$.

b.- FORMA DE USO:

MIN = MINMON (MON, XOY, ARGXOY, N).

MIN debe ser variable entera.

c.- DESCRIPCION DE LOS ARGUMENTOS:

MON: Variable entera que contiene el número de
puntos bases de la variable independiente
x.

XOY: Arreglo real de 8 bytes que contiene los
MON puntos bases x_i ; $i = 1, 2, \dots, \text{MON}$ de la
variable independiente; los mismos que deben estar ordena
dos de menor a mayor.

ARGXOY: Variable real de 8 bytes, que contiene el
valor \bar{x} , de la variable independiente x , pa
ra el cual se desea encontrar el subíndice del menor de
los puntos bases que debe intervenir en la evaluación
del polinomio de Lagrange.

N: Grado del polinomio a evaluarse.

MINMON: Variable entera, que corresponde al nombre
del subprograma de función, y se le asigna
el valor del subíndice que corresponde al menor de los
puntos bases, x_{\min} , que debe intervenir en la evaluación
del polinomio.

Este subprograma de función se le debe utilizar antes de
llamar al subprograma de función DIRY.

SUBROUTINA TABLAD.

a.- PROPOSITO:

Encontrar la tabla de diferencias divididas, necesarias
para la evaluación del polinomio de colocación por dife-
rencias divididas finitas de Newton.

b.- FORMA DE USO:

CALL TABLAD (N, M, X, Y, TABLA, IND, K).

c.- DESCRIPCION DE LOS ARGUMENTOS:

- N: Variable entera que contiene el número de pares ordenados $(x_i, f(x_i))$; $i = 1, 2, \dots, n$ tomados como muestras.
- M: Variable entera que contiene el orden de las diferencias finitas más altas a determinarse. M siempre debe ser menor que N.
- X: Arreglo real de 8 bytes que contiene los N puntos bases, $x_i = 1, 2, \dots, N$, tomados de la variable independiente x; los cuales deben estar ordenados de menor a mayor.
- Y: Arreglo real de 8 bytes, que contiene las N muestras, $f(x_i)$, tomadas de la función.
- TABLA: Arreglo bidimensional de 8 bytes, de tamaño $k \times k$, al que se le asigna todas las diferencias finitas calculadas desde orden 1 hasta M. Si hay interés en la forma de asignación, referirse al Apéndice 1.
- IND: Variable entera a la que se le asigna el

valor 1 cuando $N \leq M$. En cambio cuando hay consistencia se asigna a IND el valor cero.

K: Variable entera cuyo valor debe ser igual al dimensionamiento que tiene en el proceso principal el arreglo TABLA.

Ninguno de los argumentos de entrada sufre alteración durante el proceso de cálculo de TABLAD.

FUNCION EVAPOL.

a.- PROPOSITO:

Evalúa el polinomio de colocación por diferencias finitas divididas de Newton de grado d , $P_d(x)$, utilizado como función aproximante de una función $f(x)$ conocida en forma discreta para n valores x_i ; $i = 0, 1, 2, \dots, n$, de la variable independiente x .

b.- FORMA DE USO:

YR = EVAPOL (M, N, NGRAD, ARGX, X, Y, TABLA, IND, K).

YR debe ser una variable real de 8 bytes, pues EVAPOL es de 8 bytes.

c.- DESCRIPCION DE LOS ARGUMENTOS:

M: Variable entera que contiene el orden de la o de las diferencias finitas más altas obtenidas ya por la subrutina TABLAD.

N: Variable entera que contiene el número de pares ordenados $(x_i; f(x_i)); i = 1, 2, \dots, n$, que se conocen de la función $f(x)$ a la que se desea aproximar por el polinomio de Newton de grado d .

NGRAD: Variable entera que contiene el grado del polinomio que se quiere utilizar como función aproximante.

ARGX: Variable real de 8 bytes, que contiene el valor de la variable x , $\bar{x} \neq x_i$, para el cual se desea conocer el valor de la función, $f(\bar{x})$, por aproximación, mediante la evaluación del polinomio de Newton.

X: Arreglo real de 8 bytes que contiene los n puntos bases, x_i , tomados de la variable independiente x . Estos puntos bases, deben estar ordenados de menor a mayor.

Y: Arreglo real de doble precisión que contiene las n muestras, $y_i = f(x_i)$, tomadas de la variable dependiente.

TABLA: Arreglo bidimensional real de doble precisión de tamaño $k \times k$, que contiene todas

las diferencias finitas que debieron ser calculadas ya por la subrutina TABLAD.

IND: Variable entera a la que se le asigna el valor -1 cuando $NGRAD > M$. En caso contrario $IND = 0$.

K: Variable entera cuyo valor debe ser igual al dimensionamiento que tiene en el programa principal el arreglo TABLA.

EVAPOL: Nombre del subprograma de función, es una variable real de doble precisión a la que se le asigna el valor evaluado del polinomio de Newton. Cuando $IND = -1$, a esta variable se le asigna el valor cero.

Este subprograma de función asume que las diferencias finitas ya fueron determinadas por la subrutina TABLAD.

Ninguno de los argumentos de entrada (M, N, NGRAD, ARGX, X, Y, TABLA, K) no sufren alteración durante el proceso de cálculo de EVAPOL.

B I B L I O G R A F I A

B I B L I O G R A F I A

- 1.- Carnahan, B., Luther, H.A., Wilkes, J.O., *Applied Numerical Methods*, Wiley, 1969.
- 2.- Scheid, F., *Análisis Numérico*, Mc Graw-Hill, Schaum, 1968.
- 3.- Conte, S.D., De Boor, C., *Análisis Numérico*, Mc Graw-Hill, 1974.
- 4.- Schich, W., Merz, C.J., *Fortran para Ingeniería*, Mc Graw-Hill, 1974.
- 5.- Villavicencio, A.J., *Optimización del Alumbrado Público*, Tesis de Grado, 1978.