

Enjeu.



**UNIVERSIDAD NACIONAL
AUTONOMA DE MEO**

FACULTAD DE INGENIE

**TRABAJO ESCRITO
(APUNTES)
Computación Aplicada
Ingeniería Petrolera.**

Q u e p r e s e n

Luis Roberto Samani Huerta

para obtener el Tlo de

INGENIERO PETRO



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

CONTENIDO

CAPITULO		Página
I	INTRODUCCION	
II	SISTEMAS DE ECUACIONES LINEALES Y NO-LINEALES	13
III	INTERPOLACION	48
IV	INTEGRACION NUMERICA	90
V	SOLUCION DE ECUACIONES DIFERENCIALES ORDINARIAS	146
	APENDICE A	157
	APENDICE B	163
	APENDICE C	166
	BIBLIOGRAFIA	181

CAPITULO I

INTRÓDUCCION

Todos aquellos ingenieros que por el carácter de su trabajo emplean la computadora deben saber, al menos básicamente, cuales son las restricciones de la computadora; su eficiencia en la ejecución de operaciones tales como la suma y la multiplicación; cuantos números pueden ser representados en una computadora; como evitar los errores de truncamiento en la elaboración de programas, etc.

El tema de este capítulo gira precisamente alrededor de estos conceptos. El propósito aquí es familiarizar al lector con el ambiente de la computadora.

Números en Punto Flotante.

Uno de los aspectos más importantes en el ambiente de las computadoras es el de la aproximación de los números reales o los números en punto flotante. Tal aproximación puede observarse en el cálculo simple de una fracción. Por ejemplo

$$\frac{1}{3} \text{ igual a } 0.333333\dots$$

En el mundo real y finito de las computadoras, este número puede ser representado únicamente con cierta precisión

$$\frac{1}{3} \text{ aproximadamente igual a } 0.333333$$

Diversos métodos han sido propuestos para la representación de números reales en computadores. El más empleado de ellos es el de los números en punto flotante. Los números en punto flotante forman un conjunto F , el cual está caracterizado por cuatro parámetros: un número base B , un número de precisión t y un rango de exponente (L, U) , de tal forma que cada número en punto flotante x en F puede representarse como:

$$x = \pm \left(\frac{d_1}{B} + \frac{d_2}{B^2} + \frac{d_3}{B^3} + \frac{d_4}{B^4} + \dots + \frac{d_t}{B^t} \right) \cdot B^e$$

donde los enteros d_1, d_2, \dots, d_t satisfacen que:

$$0 \leq d_i \leq B-1 \quad \forall i = 1, \dots, t$$

$$\text{y } L \leq e \leq U$$

En todos aquellos casos donde x sea distinta de cero, $x \in F$ y $d_1 \neq 0$, se dice que, según la representación anterior, x está normalizada.

A "e" se le nombra exponente, y al número

$$f = \left(\frac{d_1}{B} + \frac{d_2}{B^2} + \dots + \frac{d_t}{B^t} \right) \text{ se le llama "fracción".}$$

La tabla que a continuación se muestra presenta algunos ejemplos de los parámetros más empleados en la representación en Punto Flotante.

COMPUTADORA	β	t	L	U	MACHEPS*
UNIVAC 1108	2	27	-128	127	1.49×10^{-8}
Honeywell 6000	2	27	-128	127	1.49×10^{-8}
PDP-11	2	28	-128	127	7.45×10^{-9}
Control Dta 6600	2	48	-976	1070	7.11×10^{-15}
Cray-1	2	48	-16384	8191	7.11×10^{-15}
Illiac-IV	2	48	-16384	16383	7.11×10^{-15}
Setun (rusa)	3	18	?	?	7.74×10^{-9}
Burroughs B5500	8	13	-51	77	1.46×10^{-11}
Hewlett Packard HP-45	10	10	-98	100	1.0×10^{-9}
IBM 360 y 370 (short precision)	16	6	-64	63	9.54×10^{-7}
IBM 360 y 370 (long precision)	16	14	-64	63	2.22×10^{-16}
Maniac II	65536	2.69	-7	7	7.25×10^{-9}

*La columna macheps representa un valor aproximado de β^{1-t}

El conjunto de números en punto flotante F , no es continuo, ni siquiera finito, y contiene exactamente $2(\beta-1) \beta^{t-1} (U-L+1) + 1$ números; además, estos números no se encuentran igualmente espaciados a lo largo de su rango de valores. Únicamente para potencias sucesivas de β , el espaciamiento es el mismo.

Según la fórmula anterior

F_{IBM} contiene $1.7293823 \times 10^{19} + 1$ números

F_{univac} contiene $3.4359738 \times 10^{10} + 1$ números

F_{CDC} contiene $5.7617928 \times 10^{17} + 1$ números

La figura 1 muestra un conjunto hipotético F de 33 puntos para el caso de un sistema donde $\beta = 2$, $t = 3$, $L = -1$, y $U = 2$. En la misma figura puede observarse que no todos los números reales pueden ser representados en este sistema hipotético. Por lo tanto, cada número en F debe de representar un intervalo completo de números reales. Si x es un número real el cual cae dentro de un cierto intervalo de valores en F, entonces representaremos con $f_2(x)$ al número en F más cercano a x .

El error relativo en la aproximación, puede demostrarse, queda expresado como una función de los parámetros β y t :

$$\left| \frac{f_2(x) - x}{x} \right| \leq \frac{1}{2} \beta^{1-t}$$

Considérese el ejemplo siguiente. El número decimal 0.1 es seleccionado frecuentemente como incremento en ciertos algoritmos iterativos. Pregunta: ¿son 10 pares de tamaño 0.1 equivalentes a un par de tamaño 1.0? y la respuesta es ¡No!, no al menos en un sistema de punto flotante cuya base sea dos ($\beta = 2$), o una potencia de 2. Esto es debido a que 0.1 no tiene una representación finita en potencias de $\frac{1}{2}$, esto es:

$$\frac{1}{10} = \frac{0}{2^1} + \frac{0}{2^2} + \frac{0}{2^3} + \frac{1}{2^4} + \frac{1}{2^5} + \frac{0}{2^6} + \dots$$

$$(0.1)_{10} = (0.000110011001100\dots)_2$$

$$(0.121212121212121\dots)_4$$

$$(0.063146314631463\dots)_8$$

$$(0.199999999999999\dots)_{16}$$

donde los subíndices denotan la base β . Las cantidades a la derecha han

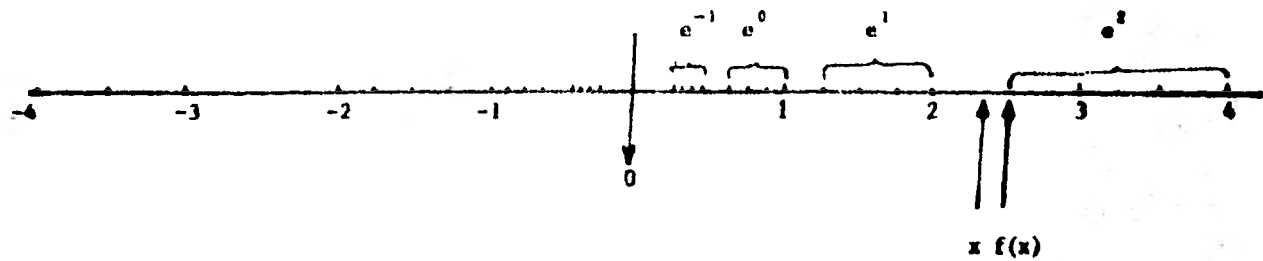


Figura 1. Conjunto hipotético de 33 números en punto flotante. Números entre llaves están igualmente espaciados. Un valor real x es aproximado por el número $f(x)$ más cercano en el conjunto.

sido truncadas después de t dígitos, y cuando 10 de ellas se han sumado, el resultado no ha sido 1.

Otra operación común es la de sumar dos números en punto fijo x e y cuyo resultado $x \oplus y \notin F$ es un elemento frecuentemente no en F: el valor real de la suma es aproximado en este caso por $f_t(x + y)$. Lo ideal sería que si $x + y$ estuviera en el rango de F, se tendría $x + y = x \oplus y = f_t(x + y)$. En la mayoría de las computadoras, este ideal es obtenido o casi obtenido para ciertos valores de x y y.

La diferencia entre $x \oplus y$ y $x + y$ es el error de redondeo introducido por la suma en punto flotante \oplus . Propiedades similares pueden observarse en la resta, multiplicación y en la división.

En el ejemplo de los 33 elementos podemos observar que:

$$\frac{5}{4} \in F, \frac{3}{8} \in F$$

$$\frac{5}{4} + \frac{3}{8} = \frac{13}{8} \neq \frac{5}{4} \oplus \frac{3}{8} = \frac{3}{2} \text{ ó } \frac{7}{4}$$

$$\frac{13}{8} - \frac{3}{2} = \text{error de truncamiento o de redondeo}$$

y $\frac{7}{2} + \frac{7}{2} = 7$ no está dentro de F, ya que 7 es mayor que el más grande de los números generados (Overflow).

La operación de la multiplicación ($x \cdot y$) puede producir "Overflows" más frecuentemente, ya que ella abarca $2t$ o $2t-1$ dígitos significativos. Por lo tanto la multiplicación produce truncamientos con más frecuencia

que la adición.

NOTA.- En el ambiente de una computadora las operaciones en punto flotante de suma y multiplicación son conmutativas.

La exactitud de la operación suma en punto flotante puede ser caracterizada por el término "Machine-epsilon", o sea el número ϵ más pequeño, tal que: $1 \oplus \epsilon > 1$

Existen diversas formas de computar ϵ (o un valor aproximado de ϵ)

```

EPS      = 1
10 EPS   = 0.5 * EPS
EPSP1   = EPS + 1.
IF(EPSP1.GT.1) GO TO 10

```

El siguiente es un ejemplo de error por truncamiento. Sea la función e^x . Deseamos programar un algoritmo que permita efectuar el cálculo de e^x para cualquier número x en punto flotante.

Expandiendo e^x en series se tiene:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Si $\beta = 10$ y $t = 5$ caracterizan el sistema, y se desea calcular el valor de $e^{-5.5}$, substituyendo en la serie se tiene

$$\begin{aligned}
e^{-5.5} &= 1.0000 \\
&\quad -5.5000 \\
&\quad 15.125 \\
&\quad -27.730 \\
&\quad 38.129
\end{aligned}$$

$$\begin{array}{r}
 -41.942 \\
 38.446 \\
 -30.208 \\
 20.768 \\
 -12.692 \\
 6.9803 \\
 -3.4902 \\
 1.5997 \\
 \cdot \\
 \cdot \\
 \cdot \\
 \hline
 0.0026363
 \end{array}$$

La suma ha sido calculada empleando únicamente 25 términos ya que los términos subsecuentes no la modifican. ¿Es la respuesta satisfactoria? Sabemos que el resultado correcto es

$$e^{-5.5} = 0.00408677$$

Nótese, por otro lado, que algunos de los términos son mayores, por varias veces, a la respuesta final. Por ejemplo el número 38.129 tiene ya, en sí mismo, un error de truncamiento tan grande como el resultado final. En efecto, el cuarto dígito decimal se ha perdido

$$38.129 ?$$

mismo que juega un papel importante en el resultado final.

Una solución, aunque costosa, sería la de efectuar las operaciones empleando un número mayor de dígitos significativos. Sin embargo, una solución más práctica sería la de calcular $e^{5.5}$ y luego obtener su

recíproco,

$$e^{-5.5} = \frac{1}{e^{5.5}} = \frac{1}{1 + 4.4 + 15.125 + \dots} = \frac{0.0040865}{t = 5}$$

con lo cual el error se reduciría a un 0.007 por ciento.

Como conclusión podemos observar que el cómputo de ciertas operaciones no tiene que ser necesariamente complicado para incurrir en serios errores de truncamiento.

Ejercicio # 1

Evaluación de la Función Error.

El propósito de este ejercicio es mostrar que algunas funciones matemáticas pueden ser, en ocasiones, difíciles de calcular y que diversas aproximaciones pueden proporcionar resultados completamente diferentes.

Un ejemplo de tales funciones es la función error, empleada en diversos problemas de Ingeniería Petrolera, Estadística, etc.; la cual se define como:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Dado que muchos sistemas computacionales tienen esta función implementada, el problema puede darse por resuelto. Pero supongamos que la función no está disponible en uno de estos sistemas y que uno trata por diferentes medios de computarla.

A continuación se mencionan tres métodos, los cuales varían en eficiencia, exactitud y requerimientos de memoria. Escriba un programa el cual emplee cada uno de los métodos siguientes en el cálculo de la función error. Deberá evaluar $\text{erf}(x)$ para argumentos de x en el rango $0 \leq x \leq 5$, a intervalos de 0.5. En aquellos métodos donde se consideran series infinitas, dé el resultado empleando 5 términos y 10 términos. Evalúe la función error empleando la función $\text{ERF}(\)$ del sistema y compárela con sus resultados calculados.

Use simple-precisión en todos sus cálculos. Entregue el listado del programa con resultados, así como una breve descripción de éstos.

METODO 1

La serie de Taylor es frecuentemente empleada en la representación de muchas funciones. Por ejemplo, empleando tal serie en la función exponencial, uno puede substituir en la función error

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x (1 - t^2 + t^4/2! + \dots)$$

e integrar cada término.

Escriba la serie resultante y úsela en la evaluación de $\operatorname{erf}(x)$.

METODO 2

Una razón por la cual la serie de Taylor es inexacta para valores grandes de x es que existe un error substancial por cancelación, debido a la naturaleza alternante de la serie. Si se integrara por partes, se obtendría la serie

$$\operatorname{erf}(x) = \frac{2x e^{-x^2}}{\sqrt{\pi}} \left(1 + \frac{2x^2}{1 \cdot 3} + \frac{(2x^2)^2}{1 \cdot 3 \cdot 5} + \frac{(2x^2)^3}{1 \cdot 3 \cdot 5 \cdot 7} + \dots \right)$$

Esta serie no es alternante y por lo tanto no sufre del problema de cancelación. Use esta serie en la evaluación de $\operatorname{erf}(x)$. ¿Qué opina de ella desde el punto de vista de su eficiencia?

METODO 3

Finalmente, uno puede intentar una función de aproximación. Por ejemplo, una función racional en ocasiones es más exacta cuando el mismo número de coeficientes (comparada con la serie) es empleado.

Tal aproximación podría ser:

$$\operatorname{erf}(x) \approx 1 - \frac{1}{(1 + a_1x + a_2x^2 + a_3x^3 + a_4x^4)^4}$$

donde

$$a_1 = 0.278393$$

$$a_2 = 0.230389$$

$$a_3 = 0.000972$$

$$a_4 = 0.078108$$

Evalúe $\operatorname{erf}(x)$ y diga si ésta es más exacta o no que la aproximación de los dos casos anteriores.

CAPITULO II

SISTEMA DE ECUACIONES LINEALES Y NO-LINEALES

En prácticamente todas las ramas de la ingeniería uno de los problemas más frecuentemente encontrados es el de la solución de sistemas de ecuaciones lineales. Tales sistemas se representan en forma matricial como $Ax = b$, donde A indica una matriz cuadrada de orden $n \times n$, b un vector columna de n términos independientes y x un vector columna de n componentes desconocidos.

En cursos de Algebra Lineal el lector ha podido aprender diferentes técnicas para la solución de tales sistemas. La regla de Cramer y el método de eliminación Gaussiana son dos ejemplos de dichas técnicas, las cuales no analizaremos en este curso. La gran mayoría de los programas de cómputo diseñados para la solución de sistemas de ecuaciones lineales se basan precisamente en el método de eliminación Gaussiana. Un ejemplo de estos programas son los programas DECOMP y SOLVE, enlistados al final del capítulo. Una ventaja de los programas DECOMP y SOLVE sobre otros programas convencionales es la de, aparte de evaluar el vector columna x , poder calcular el número de la condición de la matriz A . Este número indica qué tan cercana está la matriz de ser singular. Un número de condiciones muy "grande" indicaría una matriz sumamente inestable, es decir, que cualquier pequeño cambio en alguno de sus coeficientes produciría en x un resultado totalmente diferente. Este número asigna de cierto modo un grado de confiabilidad en la evaluación del vector x . A una matriz no-singular, el programa DECOMP le aso-

cia un número de condición igual a 1, y a una matriz singular un número de condición igual a 10^{32} . Cualquier otro número entre 1 y 10^{32} indicaría la posición relativa de la matriz con respecto a la no-singularidad o a la singularidad.

El número de la condición de la matriz A se define como

$$\text{Cond (A)} = \frac{\max_x \frac{\|Ax\|}{\|x\|}}{\min_x \frac{\|Ax\|}{\|x\|}}$$

donde $\|x\|$ indica la norma del vector x.

Es importante mencionar que aún cuando DECOMP y SOLVE pueden resolver cualquier sistema de forma $Ax = b$, existen otros algoritmos que son más eficaces bajo determinadas circunstancias particulares. Por ejemplo si la matriz A fuera tridiagonal, el algoritmo de Thomas proporcionaría la solución del vector x a través de un número de operaciones mucho menor al requerido por el método de eliminación Gaussiana.

Algoritmo de Thomas.

Sea A una matriz tridiagonal de forma

$$A = \begin{pmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & a_3 & b_3 & c_3 & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot \\ & & & & & \cdot & \cdot & \cdot \\ & & & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & & & a_n & b_n \end{pmatrix}$$

entonces A puede ser factorizada como el producto de dos matrices,

$A = LU$ donde

$$L = \begin{pmatrix} a_1 & & & & & \\ a_2 & a_2 & & & & \\ & a_3 & a_3 & & & 0 \\ & & \cdot & \cdot & & \\ & & & \cdot & \cdot & \\ 0 & & & & \cdot & \cdot \\ & & & & & \cdot & \cdot \\ & & & & & & a_n & a_n \end{pmatrix}$$

y

$$U = \begin{pmatrix} 1 & \beta_1 & & & & \\ & 1 & \beta_2 & & & 0 \\ & & \cdot & \cdot & & \\ & & & \cdot & \cdot & \\ 0 & & & & \cdot & \cdot \\ & & & & & \cdot & \cdot \\ & & & & & & 1 & \beta_{n-1} \\ & & & & & & & 1 \end{pmatrix}$$

matrices L y U.

Un sistema de ecuaciones $Ax = f$ puede expresarse como $L Ux = f$. Haciendo $Ux = y$ se obtendrá el sistema $Ly = f$, el cual se resuelve en forma directa por sustitución hacia adelante

$$y_1 = f_1/\alpha_1$$

$$y_i = \frac{f_i - a_i y_{i-1}}{\alpha_i}, \quad i = 2, 3, \dots, n$$

Una vez calculado el vector y se puede evaluar directamente x en el sistema $Ux = y$, por sustitución hacia atrás

$$x_n = y_n$$

$$x_i = y_i - \beta_i x_{i+1}, \quad i = n-1, n-2, \dots, 1$$

En el apéndice A se describen algoritmos similares para la solución de sistemas de ecuaciones lineales cuyas matrices presentan formas bi-tridiagonal, tri-tridiagonal y penta-diagonal.

Aplicación de la Solución de Sistemas de Ecuaciones Lineales al Problema del Flujo Transitorio Uni-dimensional en Yacimientos Confinados

La solución de la ecuación uni-dimensional de difusión hidráulica es requerida en problemas de flujo lineal transitorio (una fase) a través de medios porosos.

$$\frac{\partial^2 p}{\partial x^2} = \frac{1}{\eta} \frac{\partial p}{\partial t}$$

es la ecuación de difusión donde

p - presión

x - coordenada

t - tiempo

η - coeficiente de difusión

Aproximado por diferencias finitas sabemos que

$$\frac{\partial^2 p}{\partial x^2} = \frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{\Delta x^2}$$

$$\frac{\partial p}{\partial t} = \frac{p_{i,j+1} - p_{i,j}}{\Delta t}$$

Sustituyendo en la ecuación de difusión podemos resolver explícitamente para $p_{i,j+1}$.

Obviamente es necesario conocer las condiciones de frontera para que el problema esté bien definido; sean éstas por ejemplo

$$p(0,t) = p(1,t) = p^* \quad \forall t > 0$$

$$p(x,0) = p_0(x) \quad \forall x$$

Substituyendo en la ecuación de difusión las aproximaciones por diferencias finitas, obtendremos en forma explícita

$$p_{i,j+1} = p_{i,j} - \frac{\Delta t \eta}{\Delta x^2} (p_{i+1,j} - 2p_{i,j} + p_{i-1,j})$$

donde $p_{i,j}$ es el valor de la presión P en el nodo $i\Delta x, j\Delta t$

La fórmula anterior representa un esquema explícito de fácil solución, pero el cual no ofrece las mejores características de convergencia y estabi-

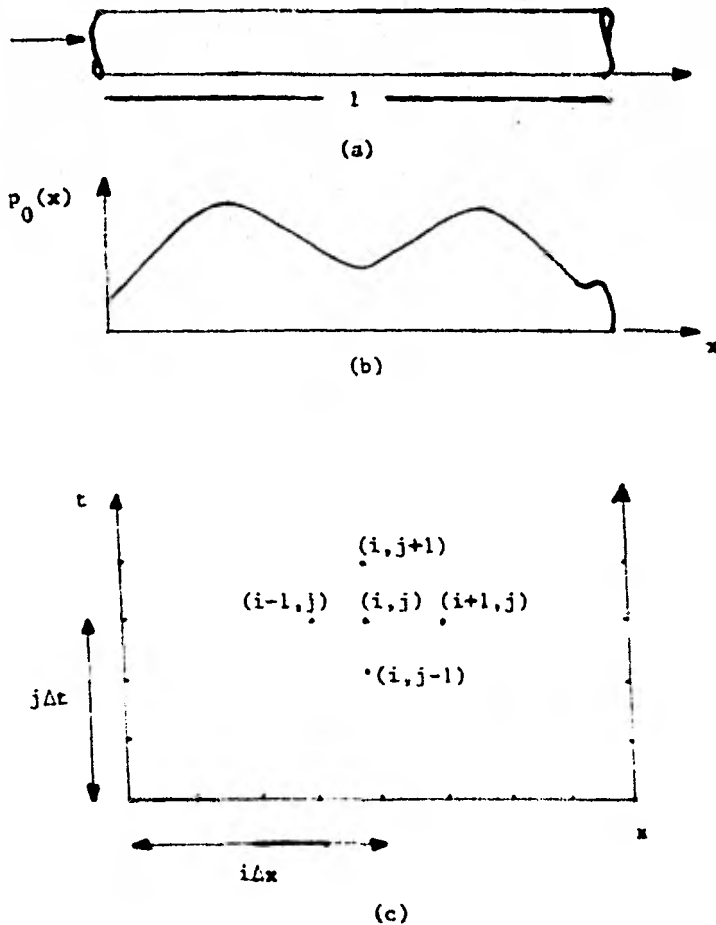


Figura 2. (a) Sección cilíndrica del material representando condiciones frontera. (b) $p_0(x)$ es la función condiciones iniciales. (c) Posición de los nodos donde una solución aproximada será calculada

lidad.

DEFINICION.

Convergencia: Si en un punto (x_i, t_j) se obtiene que

$$\lim_{\Delta x, \Delta t \rightarrow 0} |P_{i,j} - P(x_i, t_j)| = 0$$

$$\Delta x, \Delta t \rightarrow 0$$

para un determinado esquema, entonces se dice que el esquema es convergente.

Estabilidad:

Fijos Δx y Δt , y observando el comportamiento del algoritmo a medida que $t \rightarrow \infty$, es decir, observando que los errores no se amplifiquen cuando $t \rightarrow \infty$, entonces, se dice que el esquema es estable.

$$\lim_{t \rightarrow \infty} |P_{i,j} - P(x_i, t_j)| < 1$$

En el caso particular del esquema explícito anterior, si $\Delta t / \Delta x^2 \leq 1/2$, entonces habrá convergencia y estabilidad en el algoritmo.

Existen otros esquemas en los cuales los valores de Δt y Δx pueden ser seleccionados independientemente. Aproximando la derivada $\partial^2 p / \partial x^2$ por incrementos en el tiempo $j+1$, y la derivada $\partial p / \partial t$ por incremento en el tiempo $j+1$, tenemos:

$$\partial^2 p / \partial x^2 = \frac{P_{i+1,j+1} - 2P_{i,j+1} + P_{i-1,j+1}}{\Delta x^2}$$

$$\partial p / \partial t = \frac{P_{i,j+1} - P_{i,j}}{\Delta t}$$

o bien

$$\frac{P_{i,j+1} - P_{i,j}}{\eta \Delta t} = \frac{P_{i+1,j+1} + 2P_{i,j+1} + P_{i-1,j+1}}{\Delta x^2}$$

Esta ecuación tiene como incógnitas $P_{i,j+1}$, $P_{i-1,j+1}$ y $P_{i+1,j+1}$ y puede escribirse de la manera siguiente:

$$P_{i-1,j+1} - \rho P_{i,j+1} + P_{i+1,j+1} = \frac{-\Delta x^2}{\eta \Delta t} P_{i,j} \quad \text{para } i = 1, 2, \dots, n-1.$$

donde $\rho = \frac{1}{\eta \Delta t} + \frac{2}{\Delta x^2}$

además, dadas las condiciones frontera

$$P_{0,j} = P^* = P_{n,j} \quad , \quad \forall j$$

$$\text{y } P_{i,0} = P_0 \quad \Delta x \quad , \quad \forall i$$

El sistema de ecuaciones anterior puede escribirse en forma matricial como

$$\begin{array}{c|ccc|ccc|ccc} -\rho & 1 & & & P_{1,j+1} & & P_{1,j} + \frac{\eta \Delta t}{\Delta x^2} P^* & & & \\ 1 & -\rho & 1 & & P_{2,j+1} & & & & & \\ & & 1 & -\rho & 1 & & P_{2,j} & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & 1 & -\rho & 1 & P_{n-2,j} & & \\ & & & & & 1 & -\rho & P_{n-1,j} + \frac{\eta \Delta t}{\Delta x^2} P^* & & \end{array} = -\frac{\Delta x^2}{\eta \Delta t}$$

El resultado es un sistema tridiagonal, cuya solución es fácil de hallar empleando el Algoritmo de Thomas.

Este esquema, puede demostrarse, es incondicionalmente estable.

Es posible, más aún, introducir un refinamiento extra (Esquema Crank-Nicolson), tomando diferencias centrales en el tiempo

$$j + \frac{1}{2}$$

$$\frac{\partial p}{\partial t} = \frac{p_{i,j+1} - p_{i,j}}{\Delta t}$$

y promediando la aproximación de $\frac{\partial^2 p}{\partial x^2}$ en el tiempo $j + \frac{1}{2}$, con lo que se obtendría:

$$\frac{p_{i,j+1} - p_{i,j}}{\eta \Delta t} = \frac{\frac{1}{2} (p_{i+1,j+1} - 2p_{i,j+1} + p_{i-1,j+1}) + \frac{1}{2} (p_{i+1,j} - 2p_{i,j} + p_{i-1,j})}{\Delta x^2}$$

Puede demostrarse que este sistema, como el anterior, es tridiagonal.

Ejercicio # 2

Un ingeniero petrolero requiere diseñar la sección de la torre de perforación mostrada en la figura 3

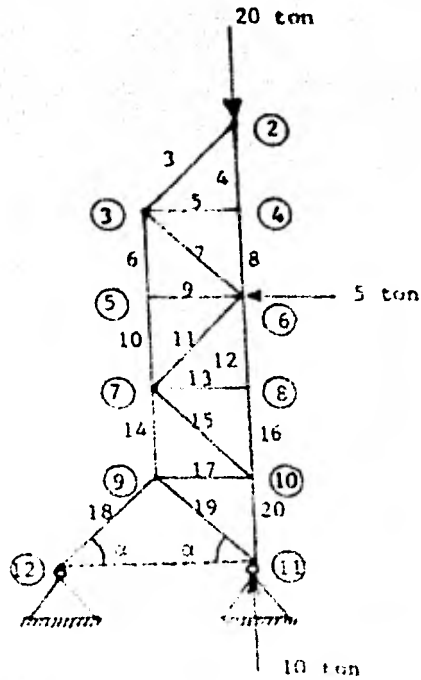


Figura 3. Diseño torre de perforación

Como parte del diseño resulta necesario conocer las fuerzas que actúan en cada pieza a fin de poder determinar el diámetro de la pieza. En la parte superior de la estructura una fuerza de 20 toneladas simula el efecto del levantamiento de la tubería de perforación cuando ésta se introduce o extrae. En la parte media de la estructura una fuerza de 5

toneladas simula el efecto debido al apoyo de las tuberías contra la estructura. Y, en la parte inferior, una fuerza de 10 toneladas simula el peso de la estructura.

Si F_x denota las componentes de las fuerzas horizontales y F_y las componentes de las fuerzas verticales, y si se consideran condiciones estáticas de equilibrio, el problema puede plantearse para su solución de la manera siguiente:

$$\text{Junta 2 } \left\{ \begin{array}{l} \sum F_y = f_4 + rf_3 - 20 = 0 \end{array} \right.$$

$$\text{Junta 3 } \left\{ \begin{array}{l} \sum F_x = rf_3 + f_5 + rf_7 = 0 \\ \sum F_y = f_6 + rf_7 - rf_3 = 0 \end{array} \right.$$

$$\text{Junta 4 } \left\{ \begin{array}{l} \sum F_x = f_5 = 0 \\ \sum F_y = f_8 - f_4 = 0 \end{array} \right.$$

$$\text{Junta 5 } \left\{ \begin{array}{l} \sum F_x = f_9 = 0 \\ \sum F_y = f_{10} - f_6 = 0 \end{array} \right.$$

$$\text{Junta 6 } \left\{ \begin{array}{l} \sum F_x = f_9 + rf_7 + rf_{11} - 5 = 0 \\ \sum F_y = f_{12} + rf_{11} - rf_7 - f_8 = 0 \end{array} \right.$$

$$\text{Junta 7 } \left\{ \begin{array}{l} \sum F_x = f_{13} + rf_{11} + rf_{15} = 0 \\ \sum F_y = f_{14} + rf_{15} - rf_{11} - f_{10} = 0 \end{array} \right.$$

$$\text{Junta 8 } \left\{ \begin{array}{l} \sum F_x = f_{13} = 0 \\ \sum F_y = f_{16} - f_{12} = 0 \end{array} \right.$$

$$\text{Junta 9 } \left\{ \begin{array}{l} \sum F_x = rf_{18} - rf_{19} - f_{17} = 0 \\ \sum F_y = rf_{18} + rf_{19} - f_{14} = 0 \end{array} \right.$$

$$\text{Junta 10 } \left\{ \begin{array}{l} \sum F_x = f_{17} + rf_{15} = 0 \\ \sum F_y = f_{20} - rf_{15} - f_{16} = 0 \end{array} \right.$$

$$\text{Junta 11 } \left\{ \sum F_y = 10 - rf_{19} - f_{20} = 0 \right.$$

Escriba un programa que emplee las subrutinas DECOMP y SOLVE en la solución del problema planteado.

Métodos Iterativos para la Solución de Sistemas de Ecuaciones No-Lineales.

Sea

$$\begin{aligned} f_1(x_1, \dots, x_n) &= 0 \\ f_2(x_1, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, \dots, x_n) &= 0 \end{aligned} \quad (1)$$

un sistema de n ecuaciones no-lineales, real, con n incógnitas, donde la función $f_i(x) = 0$ es la i -ésima función real no lineal, y sea $\alpha = [\alpha_1, \dots, \alpha_n]^t$ el vector solución buscado. Si el vector α es la solución, entonces se debe satisfacer que $f_i(\alpha) = 0$, $\forall i = 1, \dots, n$

Consideremos ahora las n funciones $F_i(x)$ definidas de tal manera que

$$x_i = F_i(x), \quad \forall i = 1, \dots, n \quad (2)$$

las cuales implican que

$$f_j(F_1(x), F_2(x), \dots, F_n(x)) = 0, \quad \forall j = 1, \dots, n$$

Lo que se pretende aquí es reordenar el sistema (1) en un nuevo sistema más "conveniente" (2).

En particular, observando la expresión (2), se tiene que

$\alpha_i = F_i(\alpha)$. Sea el vector $x_0 = [x_{1,0}, \dots, x_{n,0}]^t$ una aproximación inicial del vector solución α .

Aproximaciones sucesivas pueden definirse a partir de la función o sistema (2), de la manera siguiente:

$$x_{i,k+1} = F_i(x_k) \quad (3)$$

Supongamos ahora la existencia de una región R donde $|x_j - \alpha_j| \leq h$, $\forall j = 1, \dots, n$, y que para toda $x \in R$ existe un número positivo $\mu < 1$, tal que

$$\sum_{j=1}^n \left| \frac{\partial F_i(x)}{\partial x_j} \right| \leq \mu, \quad \forall i = 1, \dots, n.$$

Demostraremos a continuación que si la aproximación inicial del vector α (vector x_0) está dentro de la región R, entonces la función (3) converge a la solución del sistema (1), esto es

$$\lim_{k \rightarrow \infty} x_k = \alpha$$

Empleando el teorema del valor-medio tenemos que:

$$x_{i,k} = F_i(x_{k-1})$$

$$\alpha_i = F_i(\alpha)$$

$$x_{i,k} - \alpha_i = F_i(x_{k-1}) - F_i(\alpha)$$

$$x_{i,k} - \alpha_i = \sum_{j=1}^n (x_{j,k-1} - \alpha_j) \frac{\partial F_i}{\partial x_j} \left[\alpha + \zeta_{i,k-1} (x_{k-1} - \alpha) \right] \quad (4)$$

donde

$$0 < \zeta_{i,k-1} < 1$$

Tomando valor absoluto en ambos lados de la expresión (4)

$$\left| x_{i,k} - \alpha_i \right| = \left| \sum_{j=1}^n (x_{j,k-1} - \alpha_j) \frac{\partial F_1}{\partial x_j} \right|$$

y considerando la desigualdad del triángulo

$$\left| x_{i,k} - \alpha_i \right| \leq \sum_{j=1}^n \left| x_{j,k-1} - \alpha_j \right| \left| \frac{\partial F_1}{\partial x_j} \right|$$

Ahora, si el punto $x_{j,k-1}$ cae dentro de la región R, se tiene que

$$\left| x_{i,k} - \alpha_i \right| \leq h \sum_{j=1}^n \left| \frac{\partial F_1}{\partial x_j} \right| \leq \mu h < h, \quad \forall i = 1, \dots, n$$

mostrándose con ésto que el punto x_k cae también dentro de la región R. Igualmente, de la expresión anterior

$$\left| x_{j,k} - \alpha_j \right| \leq \sum_{j=1}^n \left| x_{j,k-1} - \alpha_j \right| \left| \frac{\partial F_1}{\partial x_j} \right|$$

y procediendo por inducción, tenemos que

$$\left| x_{i,k} - \alpha_i \right| \leq \sum_{j=1}^n \max_j \left(\left| x_{j,k-1} - \alpha_j \right| \right) \left| \frac{\partial F_1}{\partial x_j} \right|$$

donde "max" representa el mayor de todos los valores absolutos $\left| x_{j,k-1} - \alpha_j \right|$. Efectuando operaciones

$$\left| x_{i,k} - \alpha_i \right| \leq \max_j \left(\left| x_{j,k-1} - \alpha_j \right| \right) \sum_{j=1}^n \left| \frac{\partial F_1}{\partial x_j} \right|$$

$$\text{ó } |x_{j,k} - \alpha_j| \leq \mu \max_j (|x_{j,k-1} - \alpha_j|) \leq \mu^k h$$

y ya que si esto es cierto, también se debe de cumplir que

$$|x_{j,k-1} - \alpha_j| \leq \mu \max_j (|x_{j,k-2} - \alpha_j|) \dots \text{etc.}$$

$$\text{hasta } |x_{j,1} - \alpha_j| \leq \mu \max_j (|x_{j,0} - \alpha_j|) \leq \mu h$$

Finalmente, $\mu < 1$ implica que $\lim_{k \rightarrow \infty} \mu^k h = 0$, o lo que es lo mismo

$$\text{que } \lim_{k \rightarrow \infty} |x_{j,k} - \alpha_j| = 0$$

Si en particular las funciones $F_j(x)$ fuesen lineales, entonces una solución más simple sería emplear el método de Jacobi.

Ejemplo.

$$\text{Sean } f_1(x_1, x_2) = \frac{1}{2} \operatorname{sen} x_1 x_2 - \frac{x_2}{4\pi} - \frac{x_1}{2} = 0$$

$$\text{y } f_2(x_1, x_2) = (1 - \frac{\pi}{4}) (e^{2x_1} - e) + ex_2/\pi - 2ex_1 = 0$$

Reescribiendo las ecuaciones en la forma del sistema (2)

$$x_1 = F_1(x_1, x_2) = \operatorname{sen} x_1 x_2 - x_2/2\pi$$

$$x_2 = F_2(x_1, x_2) = 2\pi x_1 - (\pi - 1/4) (e^{2x_1 - 1} - 1)$$

Escogiendo como valores iniciales $x_{1,0} = 0.4$ y $x_{2,0} = 3.0$

$$\begin{array}{l} \text{Iteración} \\ \text{Jacobi} \end{array} \quad \left\{ \begin{array}{l} x_{1,1} = F_1(x_{1,0}, x_{2,0}) = \operatorname{sen}(1.2) - 3/2\pi = .455 \\ x_{2,1} = F_2(x_{1,0}, x_{2,0}) = 2\pi(.4) - (\pi - 1/4)(e^{-.2} - 1) = 3.0 \end{array} \right.$$

Iteración
Gauss-Seidel

$$\begin{aligned}x_{1,1} &= F_1(x_{1,0}, x_{2,0}) \\x_{2,1} &= F_2(x_{1,1}, x_{2,0}) \\x_{1,2} &= F_1(x_{1,1}, x_{2,1}) \\x_{2,2} &= F_2(x_{1,2}, x_{2,1}) \\&\vdots \\&\vdots\end{aligned}$$

Con lo cual la convergencia se aceleraría.

Método de Newton-Raphson en la Solución de Sistema de Ecuaciones No-Lineales.

Partiendo del sistema de ecuaciones no-lineales (1), podemos definir la matriz $\theta(x)$ cuyos elementos estén dados por la expresión

$$\theta_{ij}(x) = \frac{\partial F_i(x)}{\partial x_j}$$

El determinante de la matriz $\theta(x)$ no es otro que el Jacobiano del sistema evaluado en el vector x .

Definamos la función vectorial $f(x)$ como

$$f(x) = (f_1(x), \dots, f_n(x))^t$$

El proceso iterativo de Newton Raphson puede iniciarse considerando un vector inicial

$$x_0 = (x_{1,0}, \dots, x_{n,0})^t$$

y un esquema

$$x_{k+1} = x_k + \delta_k$$

donde δ_k vendría a ser la solución del sistema de ecuaciones lineales

$$B(x_k) \delta_k = -f(x_k)$$

Aplicación al Diseño de un Sistema de Recolección y Distribución de Gas*

El problema del diseño de sistemas de recolección de gas puede resolverse por medio del método de Stoner el cual está basado en la solución de ecuaciones que simulan el flujo de gas en sistemas de recolección. Este método tiene la ventaja de que incorpora en el sistema elementos tales como tuberías, compresoras, válvulas y pozos.

Modelo.

Considérese el sistema de recolección de gas mostrado en la figura 4, el cual consiste de una red de tuberías, compresoras, pozos, etc.

El régimen a considerar es el de flujo permanente. Simplificando el sistema, este puede representarse por nodos y conectores. Los nodos representan puntos donde los elementos del sistema se inician o terminan. Los conectores son los medios que permiten el intercambio de masa de un nodo a otro (figura 5).

Denominemos con la letra P al conjunto de conectores y con la letra n al número de nodos.

*Tomás L.J., 1974. Transporte de Gas en Régimen Permanente. Proyecto D-341A. Publicación No. 74RII/164. Instituto Mexicano del Petróleo.

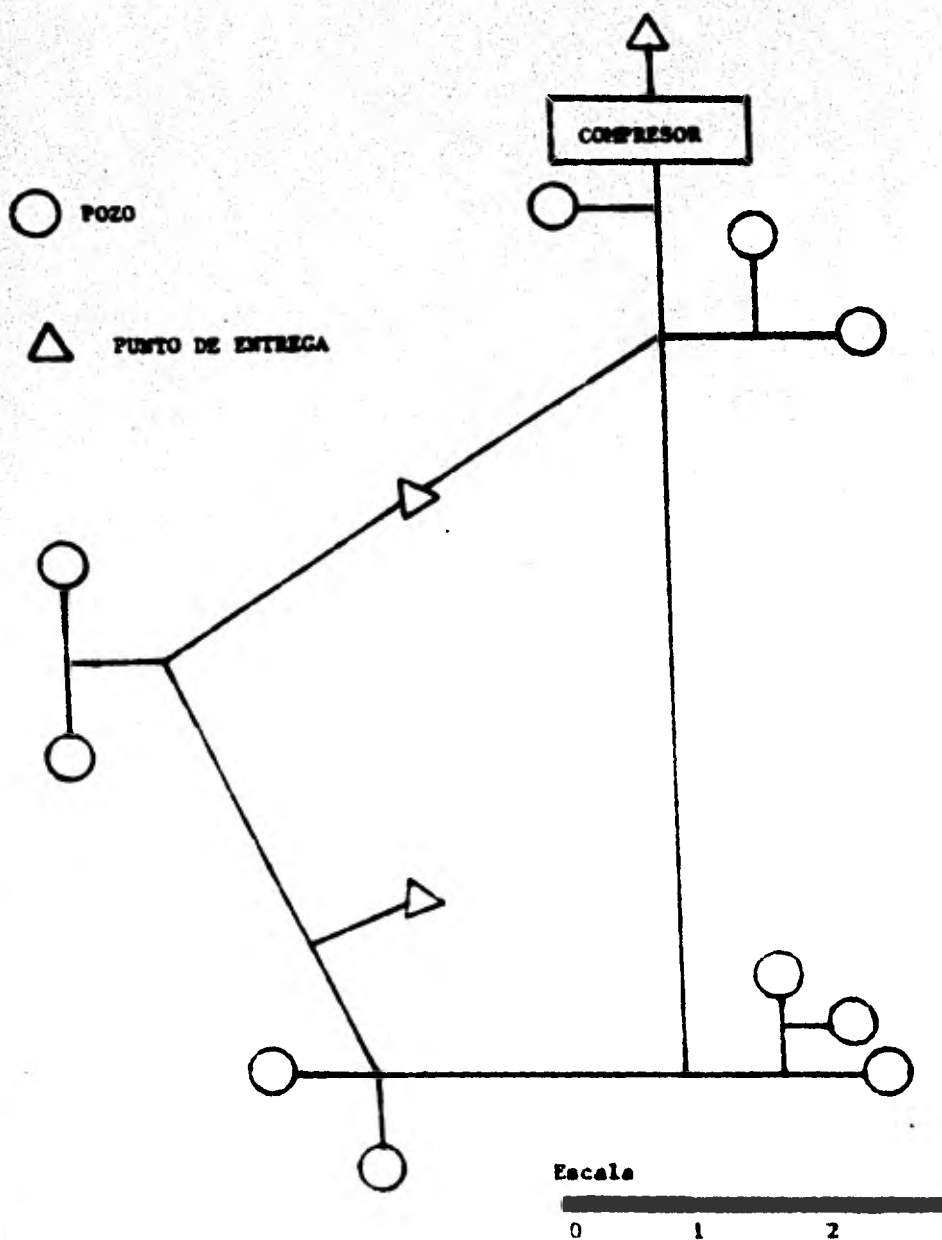


Figura 4. Representación esquemática de un sistema de recolección de gas.

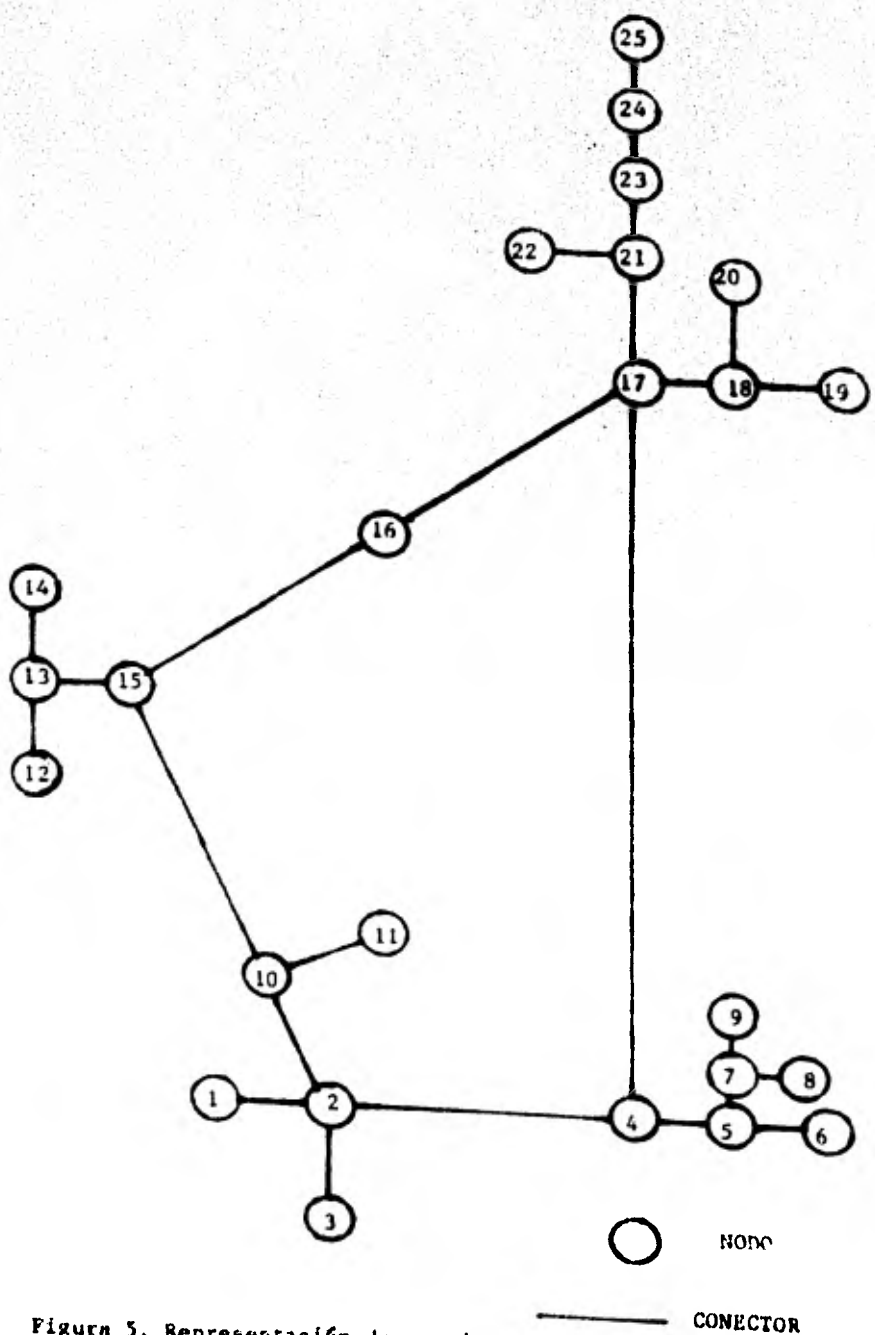


Figura 5. Representación de un sistema de recolección de gas por medio de nodos y conectores.

En un sistema como el descrito, se debe satisfacer la ley de conservación de masa en cada uno de los nodos

$$F_i = \sum_{j/(i,j) \in P} S_{ij} q_{ij} + Q_i = 0, \quad \forall i = 1, \dots, n \quad (5)$$

donde

S_{ij} es una variable que indica el sentido del flujo

si $S_{ij} = 1$ el flujo va del nodo i al nodo j , y

si $S_{ij} = -1$ el flujo va del nodo j al nodo i

q_{ij} es el gasto de gas que fluye a través del conector de los nodos i y j .

Q_i indica la adición o extracción de masa al sistema a través del nodo i .

Por ejemplo, en el caso del nodo 1 se tendría como ecuación de conservación de masa (figura 5)

$$F_1 = S_{12} q_{12} + Q_1 = 1$$

Cada gasto q_{ij} puede expresarse en términos de las diferencias de presiones en los extremos del conector como

$$q_{ij} = C_{ij} |p_i^2 - p_j^2|^n \quad (6)$$

donde

C_{ij} es un coeficiente de transmisión de la tubería, el cual depende de la geometría del tubo, de las condiciones de flujo y de la composición del gas.

P_i es la presión en el nodo i , y

n es un exponente que depende de la forma de la ecuación.

Substituyendo la expresión (6) en (5) se obtiene

$$F_i = \sum_{j/i, j \in P} S_{ij} C_{ij} |P_i^2 - P_j^2|^n + Q_i = 0, i \in n \quad (7)$$

Considerando el sistema total de nodos y conectores, la suma de los "flujos exteriores" (adición o extracción de masa) deberá ser igual a cero.

$$\sum_{i=1}^n Q_i = 0 \quad (8)$$

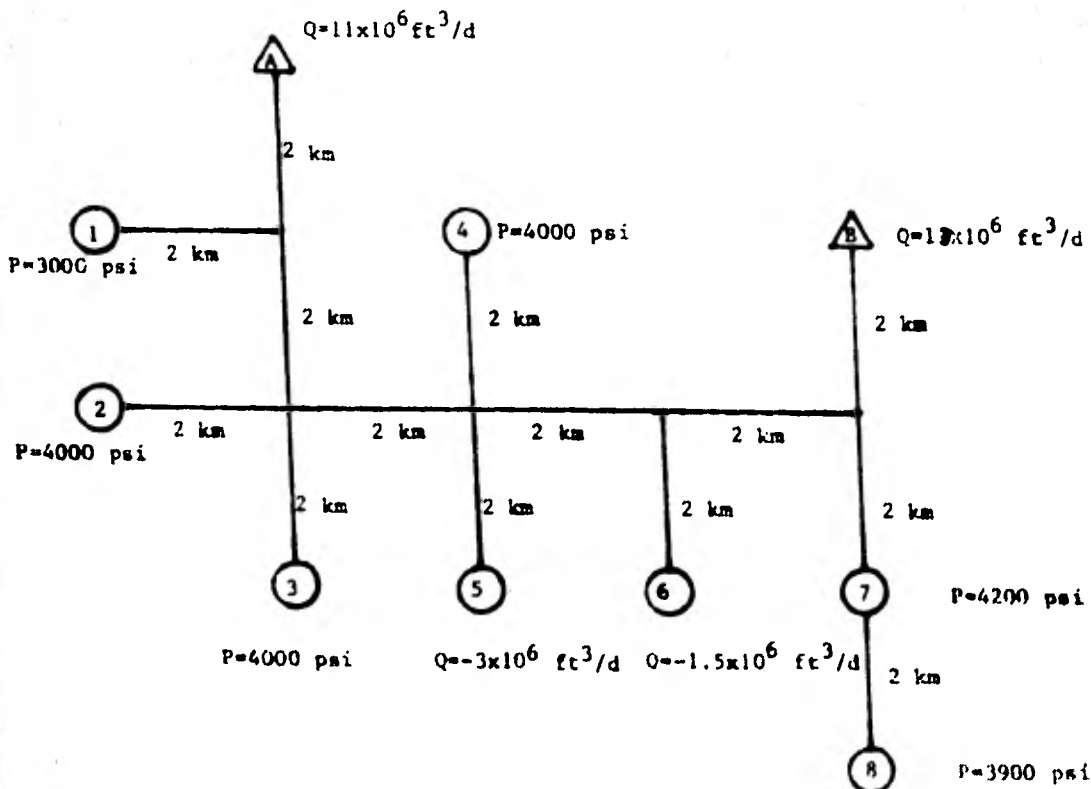


Figura 6. Campo "Las Margaritas". Estudio de un diseño de recolección de gas.

El problema aquí consiste en estimar el conjunto de gastos y presiones, Q_i y P_i , para cada nodo tal que éstos satisfagan las expresiones (7 y 8).

El sistema de ecuaciones (7), (la ecuación (8) está implícita en el sistema (7)), define un sistema de n ecuaciones con $2n$ incógnitas. Por lo tanto, para poder resolver el sistema será necesario asignar n valores, obteniéndose con ello un sistema compatible de n ecuaciones con n incógnitas.

Sin embargo, aún cuando el sistema es compatible, éste es no-lineal, por lo que resulta necesario emplear, en su solución, un método de solución de sistemas de ecuaciones no-lineales, tal como el de Newton-Raphson.

El procedimiento de solución propuesto es el siguiente:

- (I).- Asignar valores a n variables, presiones y/o gastos.
- (II).- Asignar valores iniciales a las n variables restantes.
- (III).- Substituir los valores de los pasos (I) y (II) en el sistema (7) y obtener el valor correspondiente de F_i , $\forall i = 1, \dots, n$
- (IV).- Probar si $\max_j |F_j|$ es menor o igual a cierta tolerancia fijada. Si esto ocurre el problema se da por resuelto, siendo los valores del paso (II) la solución. Si esto no ocurre, continuar con el paso siguiente.

(V).- Calcular el valor de las derivadas parciales

$$\left. \frac{\partial F_i}{\partial x_j} \right|_{x_k}$$

y resolver el sistema

$$G(x_k) \delta_k = -f(x_k)$$

donde δ_k representa al vector de las incógnitas.

(VI).- Calcular el nuevo valor de las incógnitas según el esquema

$$x_{k+1} = x_k + \delta_k$$

(VII).- Regresar al paso (III) y repetir el procedimiento.

Observaciones.

Es conveniente hacer notar que el sistema de ecuaciones del paso (V) es lineal y que la matriz resultante contiene una gran cantidad de elementos nulos. Puede observarse también que la estructura de la matriz no varía a lo largo del proceso iterativo.

Ejercicio # 3

El siguiente ejercicio es un ejemplo de aplicación a un caso real.

El campo Las Margaritas, situado en el Distrito Frontera Noreste, está produciendo gas a través de 8 pozos y 17 tuberías según el arreglo de la figura 6. El número entre paréntesis indica la longitud en kilómetros de cada línea. Todos los diámetros interiores son de 2 pulgadas.

La caída de presión en la tubería puede calcularse según la fórmula

$$q_{ij} = 842.69 E \left[\frac{(P_i^2/Z_i) - (P_j^2/Z_j)}{0.6215 L_{ij}} \right]^{0.5} d_{ij}^{2.665}$$

donde

q_{ij} = gasto de gas en ft^3/d

E = factor de eficiencia

P_i = presión en el nodo i en psi

Z_i = factor de desviación del gas a la presión P_i

L_{ij} = longitud de la línea en km.

d_{ij} = diámetro de la línea en in.

Usando como base el sistema de recolección de la figura (7), se desea

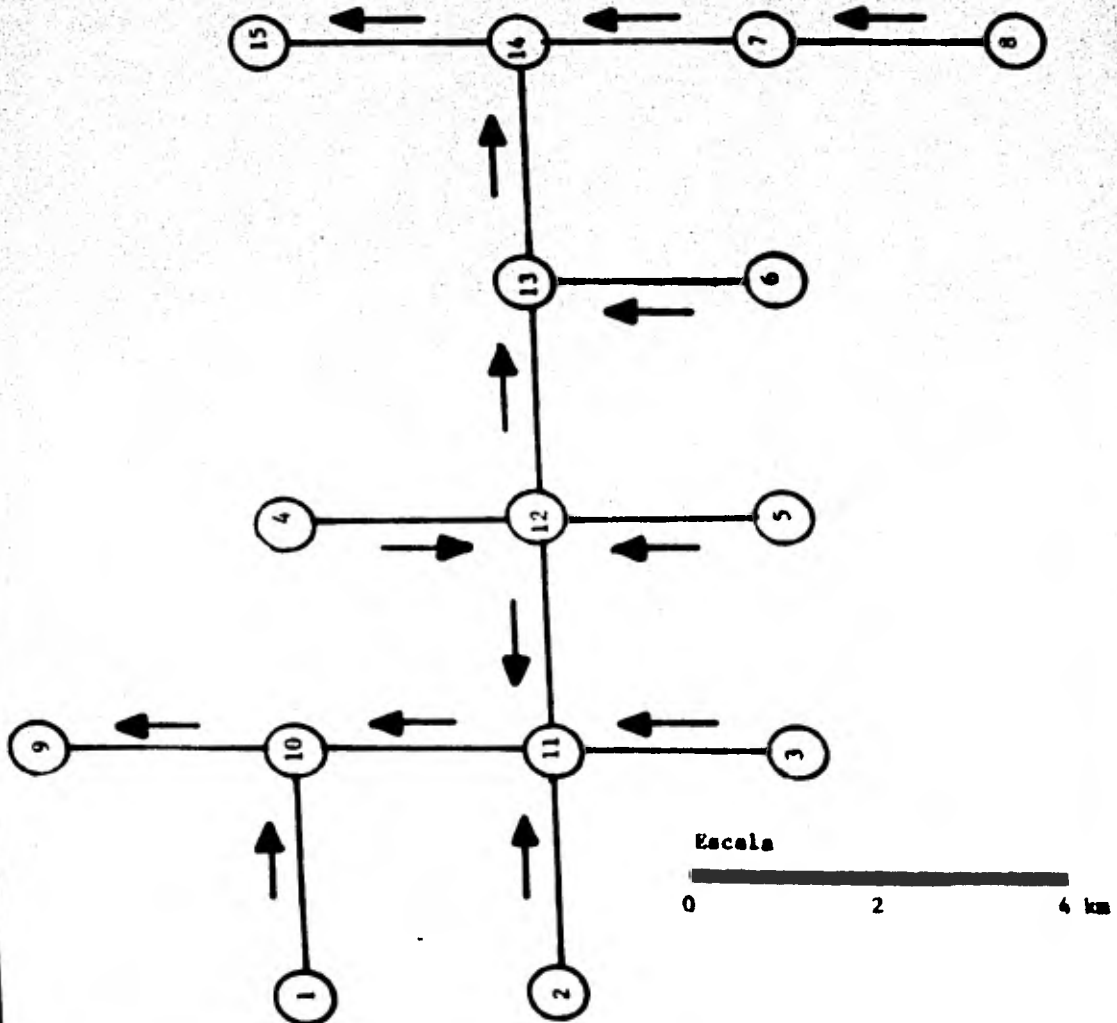


Figura 7. Representación esquemática en términos de nodos y conectores del sistema de recolección en el campo "Las Margaritas".

Las flechas indican el sentido del flujo a través de los conectores.

entregar en el punto A un gasto de $11 \times 10^6 \text{ ft}^3/\text{d}$ de gas, y en el punto B de $15 \times 10^6 \text{ ft}^3/\text{d}$. Además, se conoce la presión en la cabeza de 5 pozos y el gasto en 2 pozos más, tal como se ilustra en la misma figura.

Se desea determinar:

- (i) A qué presión se entregará el gas en los puntos A y B.
- (ii) Cuál será el gasto Q_i y la presión P_i en cada nodo donde la presión y/o el gasto se desconozcan.
- (iii) Cuál será el gasto q_{ij} a través de las líneas.
- (iv) El número de iteraciones necesarias para alcanzar la solución.

El diseño de sistemas de distribución en un campo es un problema importante a resolver, ya que permite: (a) analizar el sistema de distribución de gas, (b) cuantificar el efecto que produciría cualquier cambio en el sistema, (c) determinar la capacidad máxima del sistema, y (d) estudiar el comportamiento del yacimiento, considerando la interacción entre los pozos en la superficie.

El sistema de ecuaciones resultante para este caso particular se deriva fácilmente de la ecuación (7)

$$F_i = \sum_{j/(1,j) \in P} S_{ij} q_{ij} + Q_i = 0, \quad i = 1, \dots, n$$

Substituyendo la ecuación del gasto se tiene

$$F_i = \sum_{j/(i,j) \in P} S_{ij} 842.69 E \left| \frac{P_i^2/Z_i - P_j^2/Z_j}{0.6215 L_{ij}} \right|^{0.5} d_{ij}^{2.665} + Q_i$$

Asumiendo un factor de eficiencia de 0.80, un factor de compresibilidad constante e igual a 1, y observando que en el sistema

$d_{ij} = 2''$ y $L_{ij} = 2 \text{ km}$ $\forall i, j$, se tiene que la expresión anterior se reduce a

$$F_i = \sum_{j/(i,j) \in P} K S_{ij} (P_i^2 - P_j^2)^{1/2} + Q_i, \quad \forall i = 1, \dots, n$$

siendo $\sum_{i=1}^n Q_i = 0$

$k = 3835$ y $n = 15$.

En forma explícita el sistema de ecuaciones resultante queda:

para $n = 1$, $k(P_1^2 - P_{10}^2)^{1/2} + Q_1 = F_1$

para $n = 2$, $k(P_2^2 - P_{11}^2)^{1/2} + Q_2 = F_2$

para $n = 3$, $k(P_3^2 - P_{11}^2)^{1/2} + Q_3 = F_3$

para $n = 4$, $k(P_4^2 - P_{12}^2)^{1/2} + Q_4 = F_4$

para $n = 5$, $k(P_5^2 - P_{12}^2)^{1/2} - 3 \times 10^6 = F_5$

para $n = 6$, $k(P_6^2 - P_{13}^2)^{1/2} - 1.5 \times 10^6 = F_6$

para $n = 7$, $k(P_7^2 - P_{14}^2)^{1/2} - k(P_7^2 - P_8^2)^{1/2} + Q_7 = F_7$

para $n = 8$, $k(P_8^2 - P_7^2)^{1/2} + Q_8 = F_8$

para $n = 9$, $-k(P_9^2 - P_{10}^2)^{1/2} + 11 \times 10^6 = F_9$

$$\text{para } n = 10, -k(P_{10}^2 - P_1^2)^{1/2} - k(P_{10}^2 - P_{11}^2)^{1/2} + k(P_{10}^2 - P_9^2)^{1/2} = F_{10}$$

$$\text{para } n = 11, -k(P_{11}^2 - P_2^2)^{1/2} - k(P_{11}^2 - P_3^2)^{1/2} - k(P_{11}^2 - P_{12}^2)^{1/2} + k(P_{11}^2 - P_{10}^2)^{1/2} = F_{11}$$

$$\text{para } n = 12, k(P_{12}^2 - P_{11}^2)^{1/2} - k(P_{12}^2 - P_2^2)^{1/2} + k(P_{12}^2 - P_{13}^2)^{1/2} - k(P_{12}^2 - P_9^2)^{1/2} = F_{12}$$

$$\text{para } n = 13, -k(P_{13}^2 - P_{12}^2)^{1/2} - k(P_{13}^2 - P_6^2)^{1/2} + k(P_{13}^2 - P_{14}^2)^{1/2} = F_{13}$$

$$\text{para } n = 14, -k(P_{14}^2 - P_{13}^2)^{1/2} - k(P_{14}^2 - P_7^2)^{1/2} + k(P_{14}^2 - P_{15}^2)^{1/2} = F_{14}$$

$$\text{para } n = 15, -k(P_{15}^2 - P_{14}^2)^{1/2} + 13 \times 10^6 = F_{15}$$

donde $P_1, P_2, P_3, P_6, P_7,$ y P_9

son presiones conocidas.

Aplicue el método de Newton-Raphson para resolver el sistema anterior considerando los siguientes valores iniciales:

$$Q_1 = -4 \times 10^6 \text{ ft}^3/\text{d}$$

$$P_5 = 3000 \text{ psi}$$

$$Q_2 = -4 \times 10^6 \text{ ft}^3/\text{d}$$

$$P_6 = 3100 \text{ psi}$$

$$Q_3 = -4 \times 10^6 \text{ ft}^3/\text{d}$$

$$P_9 = 100 \text{ psi}$$

$$Q_6 = -4 \times 10^6 \text{ ft}^3/\text{d}$$

$$P_{10} = 1500 \text{ psi}$$

$$Q_7 = -4 \times 10^6 \text{ ft}^3/\text{d}$$

$$P_{11} = 3400 \text{ psi}$$

$$Q_8 = -5.5 \times 10^6 \text{ ft}^3/\text{d}$$

$$P_{12} = 3500 \text{ psi}$$

$$P_{13} = 3600 \text{ psi}$$

$$P_{14} = 2000 \text{ psi}$$

$$P_{15} = 100 \text{ psi}$$

Stoner ha recomendado, con el propósito de acelerar el proceso de

convergencia, substituir en las primeras iteraciones, el vector

δ_k por el vector δ_k o $\delta_k + \frac{\delta_k}{2}$, $\forall k = 0, 1$

Considere esta recomendación en su programa.

C
C
C

GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING

C
C
CDO 35 K = 1, N-1
KPL = K+1

FIND PIVOT

N = K

DO 15 I = KPL, N

15

IF (ABS(A(I,K)) .GT. ABS(A(M,K))) M = I

CONTINUE

IPVT(K) = M

IF (M .NE. K) IPVT(M) = -IPVT(M)

Y = A(M,K)

A(M,K) = A(K,K)

A(K,K) = Y

C
C
C

SKIP STEP IF PIVOT IS ZERO

IF (Y .EQ. 0.000) GO TO 35

C
C
C

COMPUTE MULTIPLIERS

DO 20 I = KPL, N

A(I,K) = -A(I,K)/Y

20

CONTINUE

C
C
C

INTERCHANGE AND ELIMINATE BY COLUMNS

DO 30 J = KPL, N

Y = A(M,J)

A(M,J) = A(K,J)

A(K,J) = Y

IF (Y .EQ. 0.000) GO TO 30

DO 25 I = KPL, N

A(I,J) = A(I,J) - A(I,K)*Y

25

CONTINUE

30

CONTINUE

35

CONTINUE

C
C
C

COND = (1-NORM OF A)/(AN ESTIMATE OF 1-NORM OF A-INVERSE)
ESTIMATE OBTAINED BY ONE STEP OF INVERSE ITERATION FOR THE
SMALL SINGULAR VECTOR. THIS INVOLVES SOLVING TWO SYSTEMS
OF EQUATIONS, (A-TRANPOSE)W = E AND A*Z = Y WHERE E
IS A VECTOR OF +1 OR -1 CHOSEN TO CAUSE GROWTH IN Y.
ESTIMATE = (1-NORM OF Z)/(1-NORM OF Y)

C
C
C
C
C
C
C

SOLVE (A-TRANPOSE)W = E

C
C
C
C
C
C
C

DO 50 K = 1, N

Y = 0.000

IF (K .EQ. 1) GO TO 45

KML = K-1

DO 40 I = 1, KML

Y = Y + A(I,K)*WORK(I)

C
C
C

40

CONTINUE

45

EK = 1.000

IF (Y .LT. 0.000) EK = -1.000

IF (A(K,K) .EQ. 0.000) GO TO 90

00000500

00000610

00000620

00000630

00000640

00000650

00000660

00000670

00000680

00000690

00000700

00000710

00000720

00000730

00000740

00000750

00000760

00000770

00000780

00000790

00000800

00000810

00000820

00000830

00000840

00000850

00000860

00000870

00000880

00000890

00000900

00000910

00000920

00000930

00000940

00000950

00000960

00000970

00000980

00000990

00001000

00001010

00001020

00001030

00001040

00001050

00001060

00001070

00001080

00001090

00001100

00001110

00001120

00001130

00001140

00001150

00001160

00001170

00001180

00001190

00001200


```

      WORK(K) = -(EK + T)/A(K,K)
50 CONTINUE
   DO 60 KB = 1, N-1
      K = N - KB
      T = 0.000
      KP1 = K+1
      DO 55 I = KP1, N
         T = T + A(I,K)*WORK(K)
55 CONTINUE
      WORK(K) = T
      M = IPVT(K)
      IF (M .EQ. K) GO TO 60
      T = WORK(M)
      WORK(M) = WORK(K)
      WORK(K) = T
60 CONTINUE
C
      YNORM = 0.000
      DO 65 I = 1, N
         YNORM = YNORM + DABS(WORK(I))
65 CONTINUE
C
      SOLVE A*Z = Y
C
      CALL SOLVE(MOIN, M, A, WORK, IPVT)
C
      ZNORM = 0.000
      DO 70 I = 1, N
         ZNORM = ZNORM + DABS(WORK(I))
70 CONTINUE
C
      ESTIMATE CONDITION
C
      CONO = ANORM*ZNORM/YNORM
      IF (CONO .LT. 1.000) CONO = 1.000
      RETURN
C
      1-BT-1
C
60 CONO = 1.000
   IF (A(1,1) .NE. 0.000) RETURN
C
      EXACT SINGULARITY
C
90 CONO = 1.00+32
   RETURN
   END
SUBROUTINE SOLVE(MOIN, M, A, B, IPVT)
C
C   INTEGER MOIN, M, IPVT(M)
C   DOUBLE PRECISION A(MOIN,M),B(M)
C
C   SOLUTION OF LINEAR SYSTEM, A*X = B .
C   DO NOT USE IF DECOMP HAS DETECTED SINGULARITY.
C
C   INPUT..
C
C   MOIN = DECLARED ROW DIMENSION OF ARRAY CONTAINING A .
C   M = ORDER OF MATRIX.

```

```

0001200
0001210
0001220
0001230
0001240
0001250
0001260
0001270
0001280
0001290
0001300
0001310
0001320
0001330
0001340
0001350
0001360
0001370
0001380
0001390
0001400
0001410
0001420
0001430
0001440
0001450
0001460
0001470
0001480
0001490
0001500
0001510
0001520
0001530
0001540
0001550
0001560
0001570
0001580
0001590
0001600
0001610
0001620
0001630
0001640
0001650
0001660
0001670
0001680
0001690
0001700
0001710
0001720
0001730
0001740
0001750
0001760
0001770
0001780
0001790

```


CAPITULO III

INTERPOLACION

La interpolación es uno de los problemas que con más frecuencia se enfrenta el ingeniero petrolero en la práctica. En este capítulo se describen algunas técnicas de interpolación.

Considérese un problema en el cual se tiene un conjunto de puntos en R^2 definidos por sus coordenadas

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

y donde

$$x_1 < x_2 < \dots < x_n$$

Los valores y_i pertenecen a observaciones efectuadas de un cierto fenómeno a las condiciones x_i .

El problema de la interpolación lineal consiste en construir una función f tal que, el valor de f bajo las condiciones x_i sea igual a las observaciones y_i .

$$f(x_i) = y_i, \forall i = 1, \dots, n.$$

Al mismo tiempo, la función f debe tomar "valores razonables" entre los puntos dato cuando se use con propósitos de interpolación. En la práctica, existen tantos criterios para definir "valor razonable" como problemas distintos. Por ejemplo, si las observaciones y_i provienen

de una función matemática "suave", la técnica de los splines cúbicos puede proporcionar buenos resultados. Para valores y_i experimentales y provenientes de observaciones más o menos imprecisas, no será necesario forzar a que la función ajustada pase por los puntos dato. Bastará, en este caso, con el ajuste de una función global para obtener resultados aceptables.

En términos matemáticos, el suavizamiento de una función está relacionado con el valor absoluto de la segunda derivada de la función

$$|f''(x)|,$$

y la simplicidad está asociada con el grado de la función.

El problema de la interpolación principia pues con la definición de "función razonable".

La mayoría de las funciones $f(x)$ se construyen a partir de otras funciones más elementales. La combinación lineal de monomios (x^k) produce una función polinómica. En general, la forma de las funciones interpoladoras es

$$f(x) = \sum_{i=1}^m \lambda_i f_i(x)$$

Si las funciones f_i estuvieran definidas como

$$f_i = \text{Sen } ix \quad \delta \quad f_i = \text{Cos } ix,$$

la función interpoladora $f(x)$ representaría un polinomio trigonométrico.

Si f_1 estuviese definida como

$$f_1 = \frac{a_0 + a_1x + \dots + a_nx^n}{b_0 + b_1x + \dots + b_nx^n}.$$

$f(x)$ representaría una función racional.

Estas son algunas de las funciones interpoladoras que a continuación se presentan.

Interpolación Polinomial.

Histórica y pragmáticamente, las funciones polinomiales constituyen el tipo de funciones más empleadas en procesos de interpolación.

Una función polinomial de grado n .

$$p_n(x) = a_0 + a_1x + \dots + a_nx^n$$

es fácil de derivar e integrar, y sus coeficientes pueden estimarse sin dificultad. Además, según el teorema de Weierstran*, cualquier función continua $h(x)$ puede ser aproximada dentro de un cierto intervalo cerrado por una función polinomial única.

La existencia de los coeficientes a_i está asegurada ya que haciendo

$$y_i = p_n(x_i) = \sum_{k=0}^n a_k x_i^k \quad \forall i=1, \dots, n$$

*Ralston, A., 1965. *A First Course in Numerical Analysis*. McGraw-Hill.

se obtiene un sistema de ecuaciones lineales cuya matriz A será no lineal toda vez que se cumpla la condición de que

$$x_i \neq x_j, \quad \forall i \neq j.$$

En la práctica, sin embargo, se ha observado que la matriz A es sumamente mal condicionada. Un ejemplo clásico es el de valores x_i regularmente espaciados en el intervalo $[0, 1]$, los cuales al ser sustituidos en las funciones $1, x, x^2, \dots, x^n$, generan elementos positivos entre 0 y 1, produciéndose con ello una estrecha dependencia entre las columnas o entre los renglones de A.

Un método de interpolación más efectivo, que no presenta el problema de la dependencia, será descrito más adelante (Descomposición del Valor Singular).

Otra alternativa para la construcción de funciones interpoladoras es la de los polinomios de Lagrange.

$$L_j(x) = \frac{(x - x_0)(x - x_1) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n)}{(x_j - x_0)(x_j - x_1) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n)}$$

$$L_j(x) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$$

La función interpoladora de Lagrange de grado n se define a partir de la combinación lineal

$$f(x) = \sum_{j=1}^m y_j L_j(x)$$

En particular, nótese que el polinomio $y_j L_j(x)$ adquiere el valor y_j cuando x se sustituye por x_j , y toma el valor cero cuando $x_i \neq x_j$, $\forall i \neq j$.

Splines Cúbicos.

Una técnica de interpolación más sofisticada que las anteriores, es la de "Splines Cúbicos".

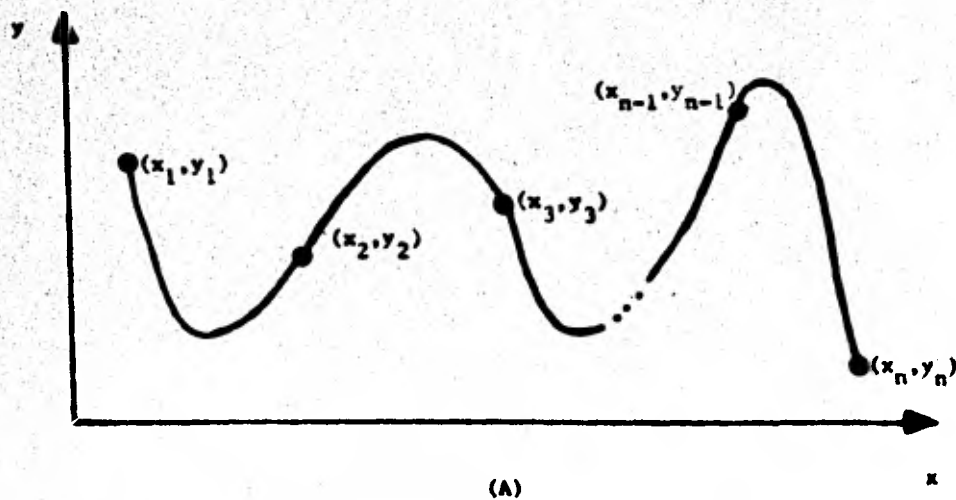
Las funciones cúbicas "spline" constituyen un desarrollo matemático reciente. Estas funciones se caracterizan por ser continuas y por tener primera y segunda derivadas continuas.

A diferencia de las técnicas de interpolación que emplean funciones polinómicas en las cuales a un conjunto de N datos se les ajusta un polinomio único de grado $N - 1$, en el método de las funciones cúbicas "spline" se ajustan $N-1$ polinomios de tercer grado, un polinomio por cada uno de los $N - 1$ intervalos definidos. Esta idea se ilustra gráficamente en la figura 8.

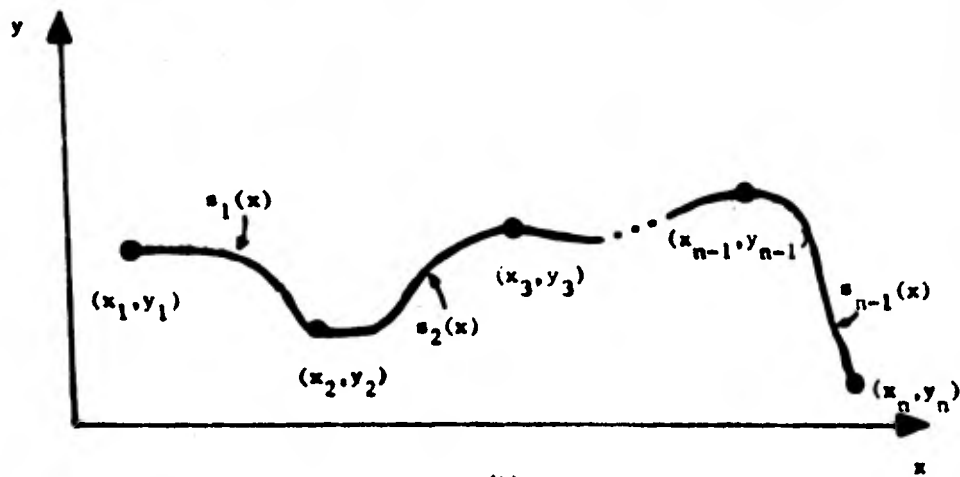
Gran parte de la teoría de los "splines" se inició con el teorema de Hólladay.

Sean las abscisas $a = x_0 < x_1 < \dots < x_N = b$

y las ordenadas $\{y_i\}$ ($i = 0, 1, 2, \dots, N$) dados.



(A)



(B)

Figura 8. (A) Función polinómica única de grado $n-1$ ajustada a un conjunto de n datos. (B) $n-1$ funciones cúbicas spline ajustadas en $n-1$ intervalos.

De todas las funciones $f(x)$ con segunda derivada continua en el intervalo $[a, b]$ tales que $f(x_i) = y_i$, $i = 0, 1, 2, \dots, N$, la función spline $s(x)$ con segunda derivada igual a cero en los extremos del intervalo, $s''(a) = s''(b) = 0$, minimiza la integral

$$\int_a^b (f''(x))^2 dx$$

Prueba:

$$0 \leq \int_a^b (f''(x) - s''(x))^2 dx$$

$$0 \leq \int_a^b (f''(x))^2 dx - 2 \int_a^b (f''(x) - s''(x)) s''(x) dx + \int_a^b (s''(x))^2 dx$$

$$0 \leq \int_a^b (f''(x))^2 dx - \int_a^b (s''(x))^2 dx + 2 \sum_{k=1}^{n-1} (f(x) - s(x)) s'''(x) \Big|_{x_k}^{x_{k+1}} \\ - 2(f'(x) - s'(x)) s''(x) \Big|_a^b$$

En el miembro del lado derecho, el tercer término desaparece debido a la condición de que

$f(x_i) = s(x_i) = y_i$, ($i = 0, 1, 2, \dots, N$); el cuarto término desaparece también según la condición de frontera

$$s''(a) = s''(b) = 0$$

Así

$$\int_a^b (s''(x))^2 dx \leq \int_a^b (f''(x))^2 dx$$

La función cúbica spline con condición $s''(a) = s''(b) = 0$ llamada también spline natural, es la función que posee la menor curvatura de todas las funciones que pueden interpolar entre puntos dato y cuya integral

$$\int_a^b (f''(x))^2 dx < \infty \text{ existe. En este sentido, la función cúbica}$$

spline es la función más suave que puede ajustarse a un conjunto de da tos.

Con el objeto de entender mejor la construcción de estas funciones es conveniente contar el número de parámetros que intervienen. En los $N - 1$ intervalos, existen $N - 1$ secciones separadas de curvas cúbicas, cada una con cuatro parámetros, haciendo un total de $4N - 4$ parámetros a determinar. El hecho de que la función $s(x)$ sea continua y tenga primera y segunda derivadas continuas en cada uno de los $N - 2$ nodos interiores x_i , introduce $3(N - 2)$ condiciones en s . Luego, el hecho de que $s(x_i)$ sea igual a y_i en cada uno de los nodos impone N condiciones más en $s(x)$, haciendo un total de $4N - 6$ condiciones. Para generar un sistema compatible es necesario contar entonces con dos condiciones más, mismas que pueden ser sugeridas por las condiciones frontera, $s''(a) = s''(b) = 0$.

La construcción de una función spline es un proceso simple y numéricamente estable. Considérese el subintervalo (x_i, x_{i+1}) y sea

$$h_i = x_{i+1} - x_i, \quad w = (x - x_i)/h_i, \quad \bar{w} = 1 - w.$$

Ya que x fluctúa sobre este subintervalo, w variará de 0 a 1 y \bar{w} de 1 a 0. Representamos la función spline en este subintervalo por medio de

$$s(x) = w y_{i+1} + \bar{w} y_i + h_i^2 [(w^3 - w) \sigma_{i+1} + (\bar{w}^3 - \bar{w}) \sigma_i]$$

donde σ_i y σ_{i+1} son ciertas constantes por determinar. Los dos primeros términos en esta expresión representan una interpolación lineal, mientras que los términos entre paréntesis rectangulares representan una corrección cúbica, la cual proporcionará la suavidad en la solución. Nótese que el término corrector desaparece en los extremos del subintervalo, de tal manera que

$$s(x_i) = y_i \quad \text{y} \quad s(x_{i+1}) = y_{i+1}$$

Según esto, la función $s(x)$ interpola exactamente los datos, cualesquiera que sean los valores σ_i .

Diferenciando la función $s(x)$ tres veces y usando la regla de la cadena, así como el hecho de que $w' = 1/h_i$ y $\bar{w}' = -1/h_i$, tenemos que

$$s'(x) = (y_{i+1} - y_i)/h_i + h_i [(3w^2 - 1) \sigma_{i+1} - (3\bar{w}^2 - 1) \sigma_i]$$

$$s''(x) = 6w \sigma_{i+1} + 6\bar{w} \sigma_i, \quad \text{y}$$

$$s'''(x) = 6(\sigma_{i+1} - \sigma_i)/h_i$$

Nótese que $s''(x)$ es una función lineal la cual interpola entre

los valores $6\sigma_i$ y $6\sigma_{i+1}$.

Consecuentemente $\sigma_i = s''(x_i)/6$. Esto explica el significado de σ_i , pero no determina su valor. Nótese también que $s''(x)$ es constante en cada subintervalo y que la cuarta derivada de $s(x)$ es igual a cero. Esto debe ser cierto, desde luego, ya que $s(x)$ es una función cúbica.

Evaluando $s'(x)$ en los puntos extremo del subintervalo tenemos

$$s'_+(x_i) = \theta_i - h_i (\sigma_{i+1} + 2\sigma_i)$$

$$s'_-(x_{i+1}) = \theta_i + h_i (2\sigma_{i+1} + \sigma_i)$$

donde

$$\theta_i = (y_{i+1} - y_i)/h_i$$

En la expresión anterior resulta necesario definir s'_+ y s'_- , ya que la fórmula de $s(x)$ se cumple únicamente en el intervalo $[x_i, x_{i+1}]$ de tal forma que las derivadas en los puntos extremo no están bien definidas. Con el objeto de obtener la continuidad deseada en $s'(x)$ se imponen las condiciones siguientes en los puntos interiores

$$s'_-(x_i) = s'_+(x_i), \quad i = 2, \dots, N-1.$$

Aún cuando el valor de $s'_-(x_i)$ se calcule al considerar el subintervalo $[x_{i-1}, x_i]$ su fórmula puede ser obtenida al reemplazar i por $i-1$ en $s'_-(x_{i+1})$, lo cual conduce a

$$\theta_{i-1} + h_{i-1} (2\sigma_i + \sigma_{i-1}) = \theta_i - h_i (\sigma_{i+1} + 2\sigma_i)$$

o bien a x_{i-1} y x_{i+1} se tiene:

$$h_{i-1} \sigma_{i-1} + 2(h_{i-1} + h_i) \sigma_i + h_i \sigma_{i+1} = Q_i - Q_{i-1}, \quad i = 2, \dots, N-1$$

La expresión anterior representa un sistema de $N - 2$ ecuaciones lineales con N incógnitas, σ_i , $i = 1, 2, \dots, N$. Bajo tales circunstancias, 2 condiciones adicionales deberán ser especificadas para poder obtener una solución única. Considerando un spline natural, el problema queda resuelto, ya que $s''(x_1) = s''(x_N) = 0$, implicará que

$$\sigma_1 \text{ y } \sigma_N \text{ sean ambas idénticas a cero.}$$

Un spline con estas condiciones frontera en las incógnitas σ 's define el sistema de $N - 2$ ecuaciones lineales con $N - 2$ incógnitas si guiente:

$$\begin{bmatrix} 2(h_1 + h_2) & h_2 & & & & & \\ h_2 & 2(h_2 + h_3) & h_3 & & & & \\ & & & \ddots & & & \\ & & & & h_{n-3} & 2(h_{n-3} + h_{n-2}) & h_{n-2} \\ & & & & h_{n-2} & 2(h_{n-2} + h_{n-1}) & \\ & & & & & & \sigma_{n-1} \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_{n-2} \\ \sigma_{n-1} \end{bmatrix} = \begin{bmatrix} Q_2 - Q_1 \\ Q_3 - Q_2 \\ \vdots \\ Q_{n-2} - Q_{n-3} \\ Q_{n-1} - Q_{n-2} \end{bmatrix}$$

Este sistema puede resolverse fácilmente empleando el algoritmo de Thomas.

En ocasiones, resulta más conveniente calcular los coeficientes actuales del spline cúbico b_i , c_i , y d_i , $i = 1, 2, \dots, N - 1$, para

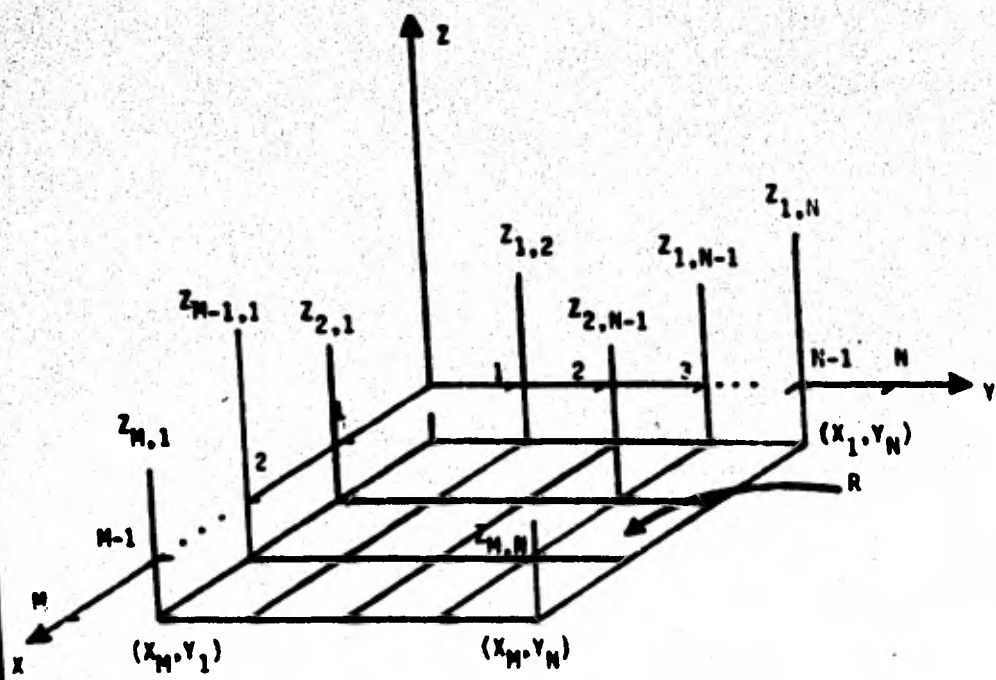


FIGURA 9. Región de aplicación de las funciones cúbicas spline.

El procedimiento es como sigue:

- a).- Para cada $y_j, j=1, \dots, N$ seleccionar las M parejas de puntos $(x_i, z_{ij}), i=1, \dots, M$ correspondientes y ajustar por el método descrito en la sección anterior las funciones spline $S_j^{(k)}(x), k=1, \dots, M-1$ y $j=1, \dots, N$ generándose con ello un total de $(M-1) \times N$ funciones cúbicas.

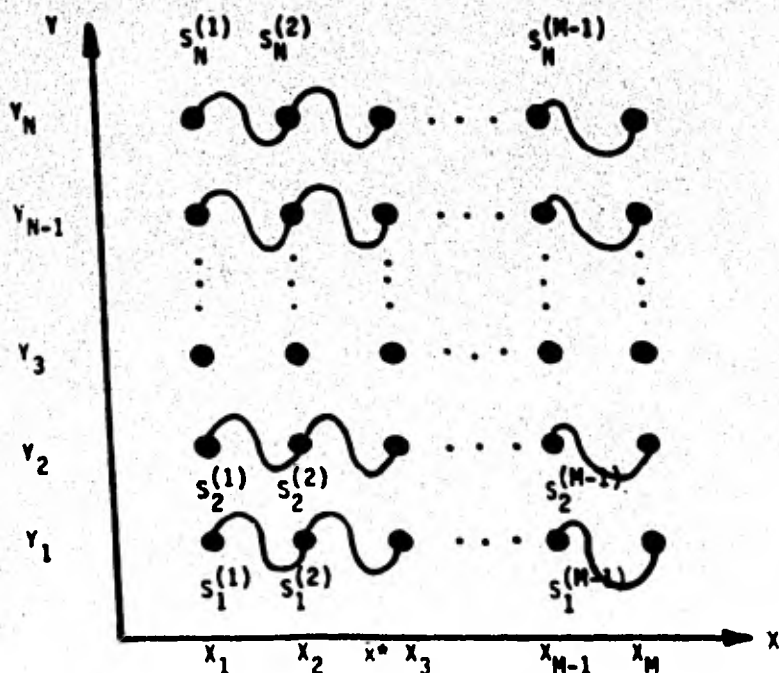


Figura 10. Secuencia de cálculo en la utilización de la técnica "cubic spline" en 3 dimensiones.

b).- Una primera aproximación \bar{z} en el punto (x^*, y^*) se obtiene seleccionando primeramente aquél conjunto de funciones

$$S_j^{(k)}(x), j = 1, \dots, N$$

que interpolen en el intervalo $[x_k, x_{k+1}]$, y el cual contenga a la abscisa x^* , es decir, $x_k \leq x^* \leq x_{k+1}$; enseguida, calcular en las funciones seleccionadas

$$S_j^{(k)}(x), j=1, \dots, N \text{ los valores } \bar{z}_{x^*,j}, \bar{z}_{x^*,j} = S_j^{(k)}(x^*)$$

c).- Considerando la pareja de valores

$$(y_j, \bar{z}_{x^*,j}), j = 1, \dots, N,$$

ajustar las funciones spline correspondientes y calcular finalmente en la función apropiada el valor \hat{z} , esto es

$$\hat{z} = s^{(\ell)}(y^*), \quad y_\ell \leq y^* \leq y_{\ell+1}, \quad \ell = 1, \dots, N-1$$

Esta técnica proporcionaría el mismo resultado en \hat{z} si en lugar de seleccionar en el punto (a) las parejas (x_i, z_{ij}) se hubiesen seleccionado las parejas (y_j, z_{ij}) , para cada x_i , $i = 1, \dots, M$.

Ajuste de Datos por Mínimos Cuadrados y Descomposición del Valor Singula

La de mínimos cuadrados es una de las técnicas de interpolación más empleada en todas las ramas de la ingeniería.

Supongamos que cierto conjunto de datos (x_i, y_i) , $i = 1, \dots, m$ son dados, siendo x la variable independiente y y la variable dependiente. Ambas están relacionadas mediante una función desconocida

$$y = y(x)$$

Por tal motivo la variable y será aproximada mediante una combinación lineal de n funciones básicas ϕ_j ,

$$y(x) = c_1 \phi_1(x) + c_2 \phi_2(x) + \dots + c_n \phi_n(x).$$

A esta combinación lineal se le conoce como modelo matemático lineal. 1

problema es seleccionar los n coeficientes c_1, \dots, c_n , de tal forma que el modelo se "ajuste" a los datos de alguna manera preestablecida. El modelo es lineal ya que esa es la forma en que los coeficientes aparecen. Las funciones $\phi_j(x)$ pueden ser no-lineales, sin embargo.

El modelo lineal más común es el polinomial

$$y(x) = c_1 + c_2x + c_3x^2 + \dots + c_nx^{n-1}$$

donde cada función $\phi_j(x)$ es igual a x^{j-1} .

Para $\phi_j(x) = \text{sen}jx$ el modelo lineal correspondiente sería

$$y(x) = c_1 \text{sen}x + c_2 \text{sen}2x + \dots + c_n \text{sen}nx$$

y para $\phi_j(x) = e^{\lambda_j x}$ sería

$$y(x) = c_1 e^{\lambda_1 x} + c_2 e^{\lambda_2 x} + \dots + c_n e^{\lambda_n x}$$

En el último caso, si las λ_j tuviesen que ser determinadas, entonces el modelo sería no-lineal.

En adelante, únicamente consideraremos el caso donde el número de puntos dato es mayor o igual a n , el número de coeficientes, $m \geq n$. De todos los métodos utilizados en la evaluación de los coeficientes c_j , el de mínimos cuadrados es el más empleado.

Los coeficientes c_j se determinan de tal forma que el cuadrado de las diferencias entre $\sum_{j=1}^n c_j \phi_j(x_i)$ y y_i es minimizado, es decir

$$\text{minimizar } \sum_{i=1}^m r_i^2 = \sum_{i=1}^m \left(\sum_{j=1}^n c_j \phi_j(x_i) - y_i \right)^2$$

c_j

Efectuando las derivadas parciales con respecto a cada coeficiente c_k , e igualando a cero se tiene:

$$\frac{\partial}{\partial c_k} \left[\sum_{i=1}^m \left(\sum_{j=1}^n c_j \phi_j(x_i) - y_i \right)^2 \right] = 0, \quad \forall k = 1, \dots, n$$

$$\delta \quad \sum_{i=1}^m 2 \phi_k(x_i) \left(\sum_{j=1}^n c_j \phi_j(x_i) - y_i \right) = 0, \quad \forall k = 1, \dots, n$$

sistema que puede ser expresado también como:

$$\sum_{i=1}^m \phi_k(x_i) \sum_{j=1}^n c_j \phi_j(x_i) = \sum_{i=1}^m \phi_k(x_i) y_i, \quad \forall k = 1, \dots, n$$

o

$$\sum_{j=1}^n c_j \sum_{i=1}^m \phi_k(x_i) \phi_j(x_i) = \sum_{i=1}^m \phi_k(x_i) y_i, \quad \forall k = 1, \dots, n$$

y que en forma matricial puede ser escrito simplemente como $Pc = q$ donde P es una matriz de orden $n \times n$ cuyos elementos son

$$\{P_{kj}\} = \sum_{i=1}^m \phi_k(x_i) \phi_j(x_i), \quad q \text{ es el vector de } n \text{ términos independientes } q_k = \sum_{i=1}^m \phi_k(x_i) y_i, \text{ y } c \text{ es el vector de coeficientes.}$$

En el caso particular donde la función $y(x)$ es aproximada por la ecuación de una línea recta $y(x) = c_1 + c_2 x$, la matriz P y el vector de términos independientes resultan iguales a

$$P = \begin{bmatrix} n & \sum x_1 \\ \sum x_1 & \sum x_1^2 \end{bmatrix} \quad q = \begin{bmatrix} \sum y_1 \\ \sum y_1 x_1 \end{bmatrix}$$

En principio, el sistema $Pc = q$ puede resolverse empleando las subrutinas DECOMP y SOLVE. Sin embargo, dado que P es una matriz simétrica, el espacio de memoria puede reducirse a la mitad, y también dado que P es una matriz positiva definida, no es necesario buscar el elemento pivote, ya que los elementos de la diagonal principal serán todos diferentes de cero.

La práctica ha demostrado que la matriz P tiene, en muchas ocasiones, un número condicional (COND) sumamente grande. Esto produce que cualquier pequeño error cometido en los datos se traduzca en un error amplificado en el cálculo de los coeficientes. Igualmente, en el caso extremo de funciones base $\phi_j(x)$ dependientes, la matriz P será singular, y su número condicional podrá considerarse como infinito.

Consecuentemente, cualquier método que evite la generación de números condicionales grandes en la matriz P , puede considerarse como un buen detector de dependencia lineal entre las funciones base. El número condicional COND calculado por la subrutina DECOMP es de ayuda, pero él, por sí sólo, no proporcionará una solución al problema.

A continuación, describiremos una técnica que permite detectar y manejar el problema de la dependencia en las funciones base.

El método más confiable que permite calcular los coeficientes del

problema de mínimos cuadrados está basado en la factorización de una matriz. Dicho método es conocido con el nombre de "descomposición del valor singular" (DVS).

El método se inicia con la generación de la "matriz de diseño" del análisis estadístico de experimentos, o sea con una matriz A de orden $m \times n$ cuyos elementos vienen definidos por la expresión

$$a_{ij} = \phi_j(x_i)$$

Si y denota al vector de m elementos $\{y_i\}$, y c al vector de coeficientes, entonces la aproximación del modelo matemático lineal

$$\sum_{j=1}^n c_j \phi_j(x_i) = y_i, \quad \forall i = 1, \dots, m$$

puede escribirse en forma matricial como

$$Ac = y,$$

El método DVS descompone la matriz A en tres matrices Σ , U y V . Σ es una matriz diagonal de orden $m \times n$ cuyos elementos no-negativos son los valores singulares de la matriz A . Las matrices $U_{m \times m}$ $V_{n \times n}$ son dos matrices ortogonales unitarias, las cuales son usadas en la transformación del sistema $Ac = y$ en un sistema equivalente $\Sigma \bar{c} = \bar{y}$. En otras palabras, si la matriz A puede expresarse como $U \Sigma V^t$, entonces la aproximación $Ac = y$ puede substituirse por

$$U \Sigma V^t c = y$$

Como U y V son dos matrices ortogonales, tales que $UU^t = I$ ó $U^{-1} = U^t$, la expresión anterior puede escribirse como

$$\Sigma \bar{c} = U^t y \quad (1)$$

o bien como $\Sigma \bar{c} = \bar{y}$ donde $\bar{y} = U^t y$ y $\bar{c} = V^t c$

Ahora bien, ¿cómo se construyen las matrices Σ , U y V , tales que Σ sea una matriz diagonal y U y V sean matrices ortogonales?

Los valores singulares de la matriz A no son otros que las raíces cuadradas positivas de los eigenvalores de la matriz AA^t , que por cierto son iguales a los eigenvalores de la matriz $A^t A$. Los vectores singulares izquierdo y derecho son los eigenvectores de las matrices AA^t y $A^t A$ respectivamente, y ellos constituyen las columnas de las matrices U y V .

Si las funciones base $\phi_j(x)$ fueran linealmente independientes en los puntos dato, entonces los valores singulares serían diferentes de cero.

El uso correcto del método de la descomposición del valor singular debe de considerar una tolerancia ζ la cual refleje la precisión de los datos originales. Cualquier valor singular σ_j mayor que ζ será aceptado y su correspondiente coeficiente \bar{c}_j podrá ser calculado como \bar{y}_j / σ_j . Por otra parte, cualquier valor singular σ_j menor que ζ será considerado como cero y el correspondiente coeficiente \bar{c}_j se igualará a cero. El cociente $\sigma_{\max} / \sigma_{\min}$ donde σ_{\max} es el mayor valor singular y σ_{\min} el menor valor singular, puede considerarse como otra forma alternativa de definir al número condicional de la matriz A

$$\text{COND}(A) = \sigma_{\max} / \sigma_{\min}$$

El método DVS se encuentra totalmente programado. La subrutina SVD (listada al final del capítulo) calcula, dada la matriz $A_{m \times n}$, las matrices U , Σ y V . El siguiente programa ilustra el uso de esta subrutina.

Ejemplo del Uso de la Subrutina SVD.

(I) Creación de la matriz de diseño A.

```

D0 20 I = 1, M
T (I) = abscisa del punto dato i.
Y (I) = ordenada del punto dato i.
D0 10 J = 1, N
A (I, J) = J-esima función base evaluada en el valor T(I)
10 CONTINUE
20 CONTINUE

```

(II) Llamada a la subrutina

```

CALL SVD (MDIM,M,N,A,SIGMA,.TRUE.,U,.TRUE.,V,IERR,WORK)
IF (IERR.NE.0) WRITE (6,13)
13 FORMAT ('ERROR ENCONTRADO EN S V D ')

```

(III) Búsqueda del máximo valor singular y detección de valores singulares despreciables. Inicialización del vector de coeficientes.

```

SIGMA1 = 0.
D0 30 J = 1, N

```

IF(SIGMA(J).GT.SIGMA1) SIGMA1 = SIGMA(J)

C(J) = 0.

30 CONTINUE

IV) Se proporciona un error de tolerancia relativo RELERR. Si los datos tienen una exactitud de 3 dígitos, entonces RELERR = 10⁻³, por ejemplo. Calcula la tolerancia en el error absoluto ϵ

TAU = RELERR*SIGMA1

Calcula el vector de coeficientes C(J) de acuerdo al desarrollo en la expresión (1).

D0 60 J = 1, N

IF(SIGMA(J).LE.TAU) GO TO 60

S = 0.

D0 40 I = 1, M

S = S+U(I,J)*Y(I) ... $\bar{y} = U^t y$

40 CONTINUE

S = S/SIGMA(J) ... $\bar{c} = \Sigma^{-1} \bar{y}$

D0 50 I = 1, N

C(I) = C(I)+S*V(I,J) ... $c = V \bar{c}$

50 CONTINUE

60 CONTINUE

En este programa M representa el número de datos y N el número de funciones base.

Ahora los coeficientes están listos para ser empleados.

```
WRITE(6,_) (C(J), J = 1, N)
```

- V) La raíz cuadrada de la suma del cuadrado de los errores residuales puede obtenerse a partir de la expresión $(\sum (A_i - y_i)^2)^{1/2}$

```
RSQ      = 0.
DO 90 I  = 1, M
RI       = 0.
DO 80 J  = 1, N
RI       = RI + C(J) * A(I, J)
80 CONTINUE
RSQ      = RSQ + (RI - Y(I)) ** 2
90 CONTINUE
R        = SQRT(RSQ)
:
```

Como ilustración del método DVS y del ajuste por mínimos cuadrados veamos el ejemplo siguiente:

En un yacimiento de petróleo se desea inyectar gas nitrógeno como técnica de recuperación secundaria. En el cálculo del gasto y de la presión de inyección es determinante conocer el valor del factor de compresibilidad Z del nitrógeno. La gráfica de la figura 11 proporciona el valor de Z en función de la presión y de la temperatura. Sin embargo, con el objeto de predecir y controlar la operación a través de una computadora, resulta necesario contar con una expresión analítica de Z .

Por tal motivo, se leyeron de la gráfica los valores de Z para todas las temperaturas indicadas y para las presiones de 0 psia a 4000 psia a intervalos de 500 psia. Como expresión analítica se eligió la función cúbica

$$Z = C_1 + C_2 P' + C_3 T' + C_4 P'^2 + C_5 P'T' + C_6 T'^2 + C_7 P'^3 + C_8 P'^2 T' + C_9 P'T'^2 + C_{10} T'^3$$

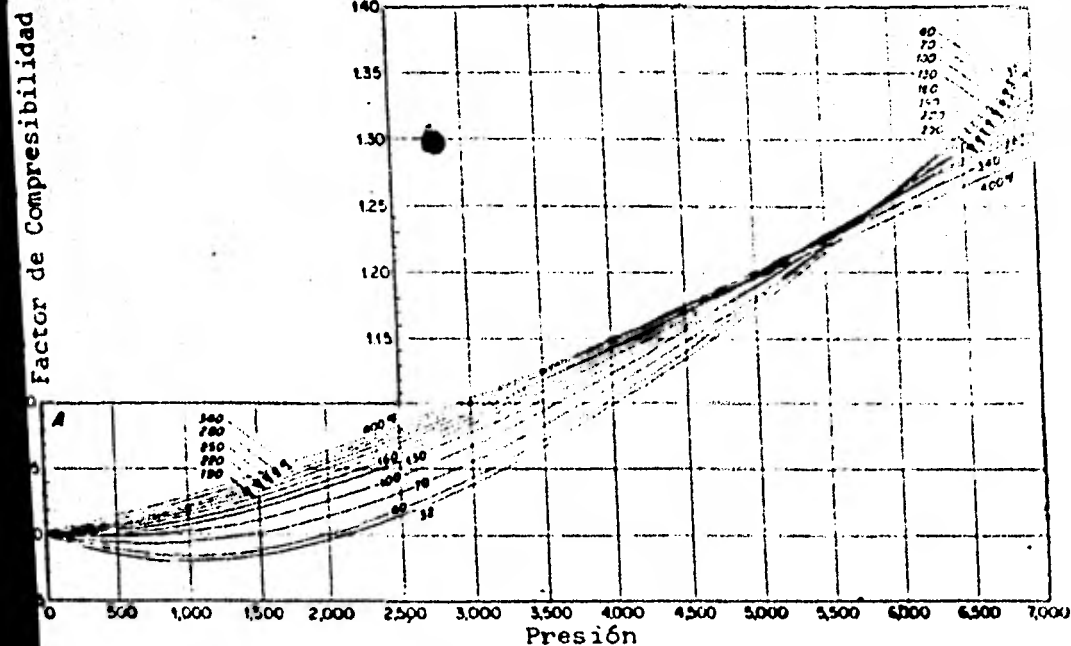


Figura 11. Factor de Compresibilidad "Z" del Nitrógeno.

donde P' y T' adoptarán diferentes valores según las alternativas:

$$\text{Alternativa A } T' = T, \quad P' = P$$

$$\text{Alternativa B } T' = \frac{T-184}{10}, \quad P' = \frac{P-2000}{100}$$

$$\text{Alternativa C } T' = \frac{T-184}{100}, \quad P' = \frac{P-2000}{1000}$$

donde P y T son los valores leídos de la gráfica, y las constantes 184 y 2000 son los valores leídos promedio de la temperatura y de la presión, respectivamente. Los divisores son factores de escala.

Considerando cada una de las tres alternativas, se efectuó el ajuste según mínimos cuadrados y según la técnica D V S. El resumen de los resultados obtenidos se muestra en la tabla 1. En la tabla, ρ es el coeficiente de correlación estadístico e indica la bondad en la predicción. Si $\rho = 1.00$ la predicción es "perfecta"; si $\rho = 0.00$ no hay tal predicción. $\sqrt{\sum r_i^2}$ es la raíz cuadrada de la suma de los errores al cuadrado e indica que tan cerca pasa la función ajustada a los puntos dato. Valores menores de $\sqrt{\sum r_i^2}$ denotan mayor cercanía. Como puede observarse, M-C permanece inalterable en los valores de ρ y $\sqrt{\sum r_i^2}$ ante las tres alternativas, lo cual es obvio ya que por construcción M-C sólo minimiza las desviaciones y no le afectan los cambios de escala en las variables independientes. D V S es, excepto en la alternativa A, mejor que M-C. Además, D V S es sensible a los cambios de escala.

La figura 12 muestra dos gráficas donde el valor real del factor de compresibilidad Z , en las abscisas, ha sido graficado contra el valor calculado Z^* , en las ordenadas. La gráfica A corresponde al ajust

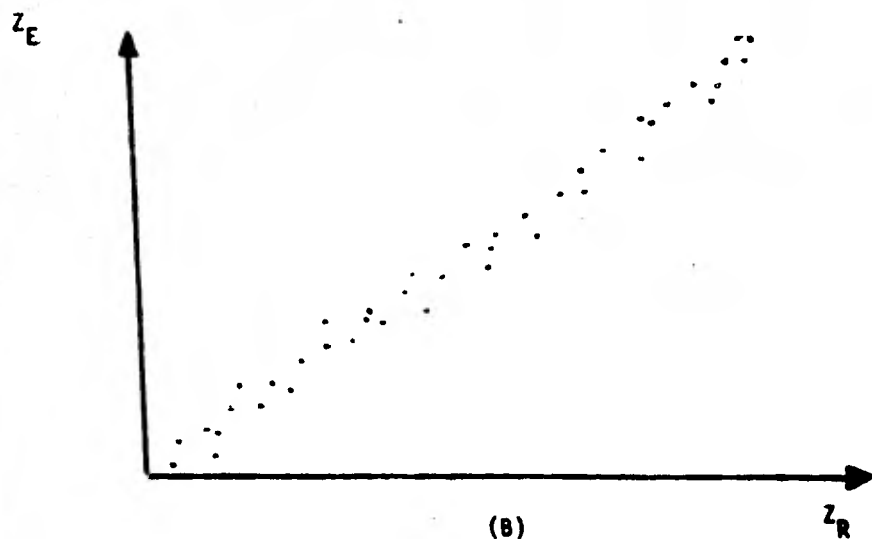
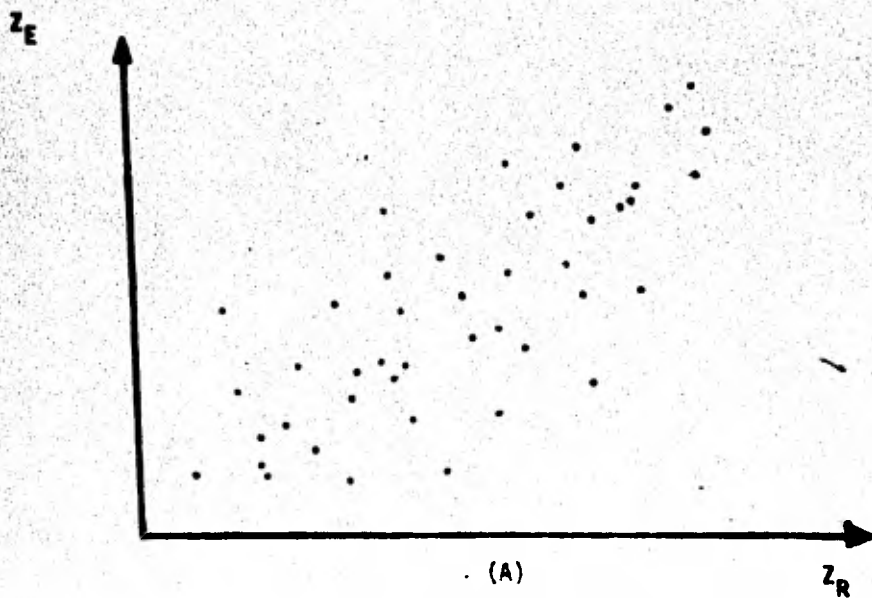


FIGURA 12. (A) Diagrama de correlación entre variable real (abscisa) y variable estimada (ordenada) por mínimos cuadrados.
(B) Diagrama de correlación entre variable real (abscisa) y variable estimada (ordenada) por DVS.

Tabla 1. Resumen de resultados del ajuste MC VS DVS

		Alternativa		
		A	B	C
M.C.	ρ	0.48	0.48	0.48
	$\sqrt{\Sigma \epsilon_i^2}$	0.4361	0.4361	0.4361
	COND	0.4591×10^{23}	0.7448×10^8	0.1462×10^3
D.V.S.	ρ	0.79	1.00	1.00
	$\sqrt{\Sigma \epsilon_i^2}$	6.8109	0.0349	0.0349
	COND	0.2789×10^{11}	0.8425×10^4	0.1220×10^2

te por M-C. (alternativas A, B y C) y la gráfica B corresponde al ajuste según D V S (alternativa C.). En la tabla 1, COND representa el número de condición de la matriz del sistema. En todas las alternativas el método D V S produce un número de condición siempre mucho menor al calculado por M-C.

Los valores de los coeficientes C_i de la función ajustada por la técnica D V S y bajo la alternativa C fueron:

$$C_1 = 0.10437360 \times 10^1$$

$$C_2 = 0.35185506 \times 10^{-1}$$

$$C_3 = 0.17852805 \times 10^{-1}$$

$$C_4 = 0.70853098 \times 10^{-2}$$

$$C_5 = 0.32667433 \times 10^{-2}$$

$$C_6 = -0.54567901 \times 10^{-2}$$

$$C_7 = -0.34567901 \times 10^{-3}$$

$$C_8 = -0.29738655 \times 10^{-3}$$

$$C_9 = -0.79286510 \times 10^{-2}$$

$$C_{10} = 0.12825284 \times 10^{-2}$$

La función ajustada proporciona un ajuste confiable para valores de T y P dentro del rango $32 \leq T \leq 400^\circ\text{F}$ y $0 \leq P \leq 4000$ psia.

Kriging - Una Nueva Filosofía Dentro de los Procesos de Interpolación.

En contraste con los métodos bi-dimensionales de interpolación, interpolar en tres dimensiones resulta sumamente costoso en tiempo de cómputo y los requisitos exigidos a la información pueden, en muchas ocasiones, no ser satisfechos. Considérese el siguiente problema donde se desea obtener un plano configurado de las permeabilidades calculadas a través de pruebas de presión efectuadas en un cierto número de pozos (Figura 13). Como suele ocurrir en estos casos, lo primero que se observa es que la información no está distribuida regularmente en el espacio. Esto impediría la aplicación del método de las funciones bi-cúbicas o del método de mínimos cuadrados descrito en el Apéndice B. Otra característica importante de la información es que ésta se encuentra agrupada en ciertas porciones del área de estudio formando lo que se conoce como "nubes de información". El ajuste de una superficie polinomial produciría, bajo estas circunstancias, resultados incoherentes ya que la información más aislada estaría ejerciendo fuerte influencia sobre los resultados de

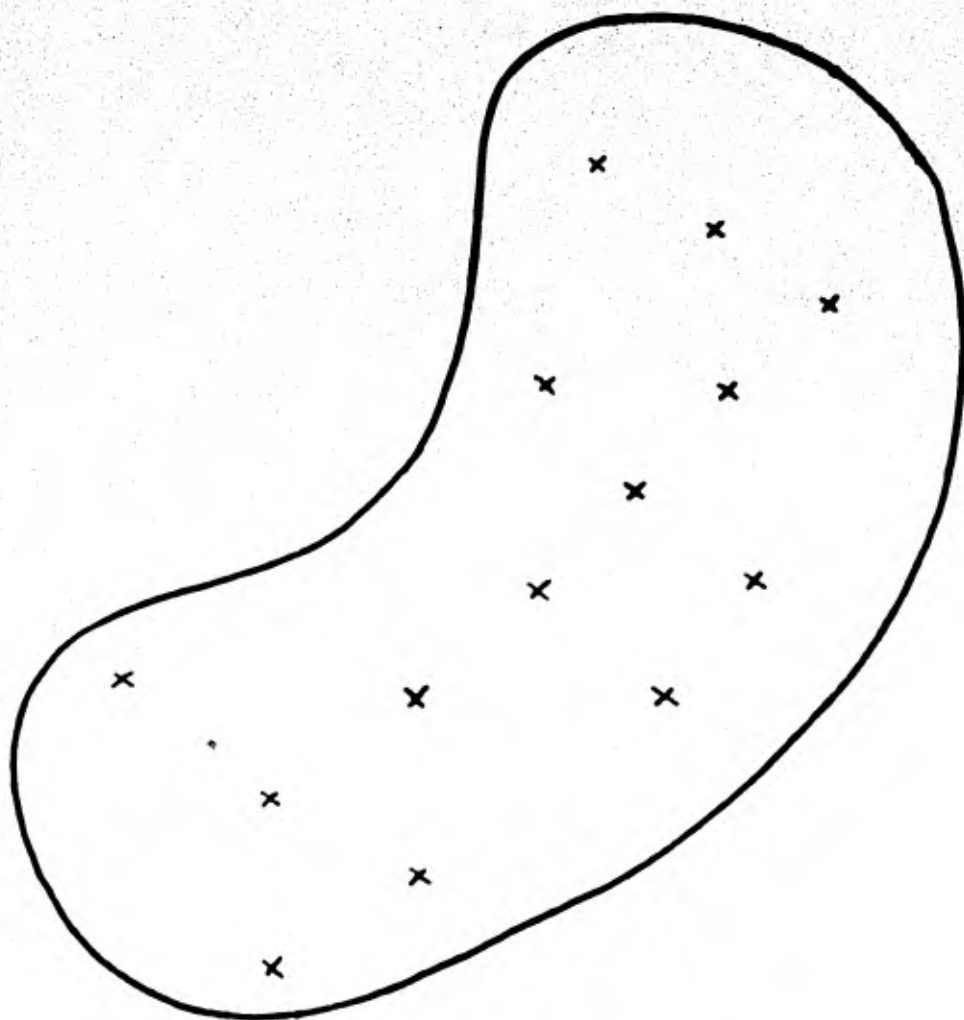


FIGURA 13. Representación gráfica de un campo petrolífero donde localizaciones de pozos perforados han sido indicadas con cruces. A través de pruebas de presión, permeabilidades han sido calculadas en cada pozo.

los coeficientes del polinomio.

Por tales motivos, se han creado otros métodos de interpolación, tales como el de ponderación con respecto al inverso de la distancia, ponderación con respecto al inverso del cuadrado de la distancia, etc., los cuales empleados conjuntamente con la técnica de búsqueda octal pueden producir resultados "aceptables". Todos estos métodos, sin embargo, no pueden evitar el error inherente de todo proceso de interpolación.

Existe un método que por diseño minimiza precisamente el error del proceso de interpolación y el cual se conoce como método Kriging de estimación. Tratar de explicar este método en toda su extensión tomaría por sí sólo un curso completo. Aquí nos limitaremos a resumir los principios básicos.

La piedra angular del método Kriging es una función denominada semi-variograma la cual expresa el grado de correlación espacial existente entre la información. Una vez estimada la función semi-variograma, es posible expresar el error, o mejor dicho, el valor promedio del error en términos del semi-variograma. Lo que resta es simplemente una aplicación del método de los multiplicadores de Lagrange sobre la expresión del error.

Si lo que se busca es estimar el valor de la variable desconocida Z empleando un estimador Z' definido como

$$Z' = \sum_{i=1}^n \lambda_i Z_i$$

donde los n valores Z_i son datos conocidos, entonces el método Kriging

proporcionaré los n valores λ_i tales que el valor esperado del cuadrado de la diferencia entre Z y Z' , esto es, $E[(Z-Z')^2]$ es el mínimo.

EJERCICIO # 4

Escriba un programa para la solución del siguiente problema. Un gas natural conteniendo 0.7% de nitrógeno fue sometido a un análisis de laboratorio donde el factor de compresibilidad Z fue medido. La tabla siguiente muestra los resultados obtenidos.

Presión (psia)	Temperatura			
	32°F	100°F	190°F	280°F
1.4	0.6885	0.8213	0.9097	0.9557
1.6	0.6593	0.8044	0.9009	0.9516
1.8	0.6433	0.7896	0.8943	0.9486
2.0	0.6369	0.7805	0.8899	0.9472
3.0	0.6981	0.7901	0.8932	0.9571
4.0	0.8103	0.8600	0.9356	0.9890
5.0	0.9333	0.9516	1.005	1.0384

Empleando el algoritmo de "bicubic-splines" descrito, interpole los valores de Z considerando valores de presión de 1.4 psia a 5.0 psia a intervalos de 0.1 psia, y considerando valores de temperatura de 32°F a 100°F a intervalos de 2°F y de 100°F a 280°F a intervalos de 10°F.

Comente la exactitud en la interpolación del método.

EJERCICIO # 5

En yacimientos productores de aceite cuando éste fluye por la tubería de producción del pozo se libera gas produciendo un flujo en dos fases. Poettmann y Carpenter han derivado una expresión analítica que permite calcular la caída de presión en tuberías verticales con flujo multifásico

$$\frac{\Delta p}{\Delta h} = \frac{1}{144} \left[\rho + \frac{f(q_M)^2}{7.413 \times 10^{10} \rho d^5} \right]$$

donde

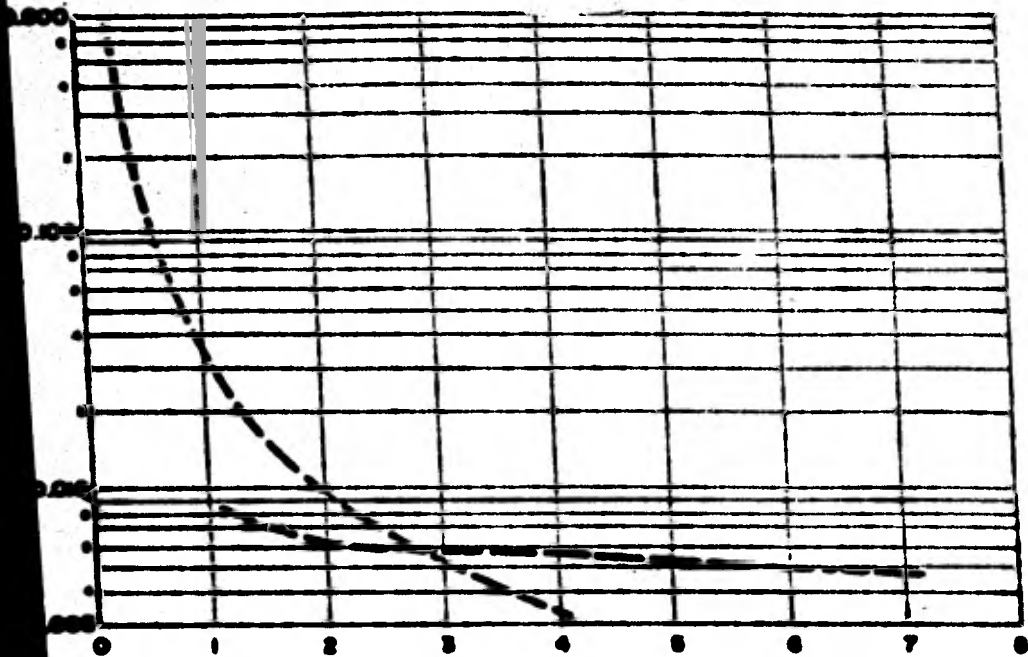
- ρ es la densidad de la mezcla gas-aceite,
- q_M es el gasto de aceite por masa de la mezcla,
- f es el factor de pérdidas de energía el cual depende del gasto q_M ,
- d es el diámetro interno de la tubería, y
- $\frac{\Delta p}{\Delta h}$ es el gradiente de presión.

Se desea encontrar, dado un cierto diámetro de tubería d , el gasto q_M que producirá la mínima caída de presión en la tubería.

Del análisis sabemos que derivando $\Delta p/\Delta h$ con respecto al gasto q_M e igualando a cero, obtendremos las condiciones buscadas en el gasto que producirán la caída mínima de presión. Sin embargo, el factor de pérdidas de energía f depende también del gasto q_M , por lo que resulta necesario generar una función que exprese f en términos de q_M . Dicha

función deberá ser substituída por f antes de proceder a la derivación.

La curva de abajo, obtenida experimentalmente por Bazendell, expresa el factor de pérdida de energía en términos del gasto qM/d . Empleando la técnica de mínimos cuadrados o la de descomposición del valor singular, ajuste una función a la curva mostrada. Substituya la función ajustada en la ecuación de Poettmann y Carpenter y encuentre una expresión del gasto qM en términos del diámetro d , para la cual la caída de presión en la tubería sea mínima.



$$\frac{qM}{d} = 10^{-6}$$

```

SUBROUTINE SPLINE (M, X, Y, B, C, D)
  INTEGER M
  DOUBLE PRECISION X(N), Y(N), B(N), C(N), D(N)

  C THE COEFFICIENTS B(I), C(I), AND D(I), I=1,2,...,M ARE COMPUTED
  FOR A CUBIC INTERPOLATING SPLINE

  B(I) = Y(I) + B(I)*(X-X(I)) + C(I)*(X-X(I))**2 + D(I)*(X-X(I))**3
  FOR X(I) .LE. X .LE. X(I+1)

  INPUT..

  M = THE NUMBER OF DATA POINTS OR KNOTS (N.O.T)
  X = THE ABSISSAS OF THE KNOTS IN STRICTLY INCREASING ORDER
  Y = THE COORDINATES OF THE KNOTS

  OUTPUT..

  B, C, D = ARRAYS OF SPLINE COEFFICIENTS AS DEFINED ABOVE.

  USING P TO DENOTE DIFFERENTIATION,

  P(I) = S(X(I))
  C(I) = SP(X(I))
  D(I) = SPP(X(I))/2
  B(I) = SPP(X(I))/6 (DERIVATIVE FROM THE RIGHT)

  THE ACCOMPANYING FUNCTION SUBPROGRAM SEVAL CAN BE USED
  TO EVALUATE THE SPLINE.

  INPUT: M, ID, I
  DOUBLE PRECISION Y

  I=1
  IF (M .LT. 2) RETURN
  IF (M .LT. 3) GO TO 50

  SET UP TRIAGONAL SYSTEM

  B = DIAGONAL, D = OFFDIAGONAL, C = RIGHT HAND SIDE.

  D(1) = X(2) - X(1)
  C(1) = (Y(2) - Y(1))/D(1)
  DO 10 I = 2, M-1
    D(I) = X(I+1) - X(I)
    D(I) = 2.*D(I-1) + D(I)
    C(I+1) = (Y(I+1) - Y(I))/D(I)
    C(I) = C(I+1) - C(I)
  10 CONTINUE

  END CONDITIONS. THIRD DERIVATIVES AT X(1) AND X(N)
  OBTAINED FROM DIVIDED DIFFERENCES

  B(1) = -D(1)
  B(N) = -D(N-1)
  C(1) = 0.
  C(N) = 0.
  IF (M .EQ. 3) GO TO 15
  C(1) = C(1)/(X(4)-X(2)) - C(2)/(X(3)-X(1))
  C(N) = C(N-1)/(X(N)-X(N-2)) - C(N-2)/(X(N-1)-X(N-3))
  C(1) = C(1)*D(1)**2/(X(4)-X(1))
  C(N) = -C(N)*D(N-1)**2/(X(N)-X(N-3))

  FORWARD ELIMINATION

  15 DO 20 I = 2, M
    Y = D(I-1)/B(I-1)
    B(I) = B(I) - Y*D(I-1)
    C(I) = C(I) - Y*C(I-1)
  20 CONTINUE

```

00000000
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710

```

C
C  BACK SUBSTITUTION
C
      C(N) = C(N)/B(N)
      DO 30 I = 1, N-1
        I = N-I
        C(I) = (C(I) - B(I)*C(I+1))/B(I)
      30 CONTINUE
C
C  C(I) IS NOW THE SIGNAL(I) OF THE TEXT
C
C  COMPUTE POLYNOMIAL COEFFICIENTS
C
      B(N) = (Y(N) - Y(N-1))/B(N-1) + B(N-1)*C(N-1) + 3.*C(N)
      DO 40 I = 1, N-1
        B(I) = (Y(I+1) - Y(I))/B(I) - B(I)*C(I+1) + 2.*C(I)
        C(I) = (C(I+1) - C(I))/B(I)
        C(I) = 3.*C(I)
      40 CONTINUE
      C(N) = 3.*C(N)
      B(N) = B(N-1)
      RETURN
C
      DO 50 B(I) = (Y(I)-Y(I))/X(I)-X(I+1)
      C(I) = 0.
      B(I) = 0.
      RETURN
      END

```

```

0C000720
0C000730
0C000740
0C000750
0C000760
0C000770
0C000780
0C000790
0C000800
0C000810
0C000820
0C000830
0C000840
0C000850
0C000860
0C000870
0C000880
0C000890
0C000900
0C000910
0C000920
0C000930
0C000940
0C000950
0C000960
0C000970
0C000980
0C000990

```

```

C
C  DOUBLE PRECISION FUNCTION SEVAL(M, U, X, T, B, C, D)
C  INTEGER N
C  DOUBLE PRECISION U, X(N), Y(N), B(N), C(N), D(N)
C
C  THIS SUBROUTINE EVALUATES THE CUBIC SPLINE FUNCTION:
C
C  SEVAL = Y(I) + B(I)*(U-X(I)) + C(I)*(U-X(I))**2 + D(I)*(U-X(I))**3
C
C  WHERE X(I) .LT. U .LT. X(I+1), USING HORNOR'S RULE
C
C  IF U .LT. X(1) THEN I = 1 IS USED.
C  IF U .GE. X(N) THEN I = N IS USED.
C
C  INPUT..
C
C  N = THE NUMBER OF DATA POINTS
C  U = THE ABSCISSA AT WHICH THE SPLINE IS TO BE EVALUATED
C  X,Y = THE ARRAYS OF DATA ABSCISSAS AND COORDINATES
C  B,C,D = ARRAYS OF SPLINE COEFFICIENTS COMPUTED BY SPLINE
C
C  IF U IS NOT IN THE SAME INTERVAL AS THE PREVIOUS CALL, THEN A
C  BINARY SEARCH IS PERFORMED TO DETERMINE THE PROPER INTERVAL.
C
C  INTEGER I, J, K
C  DOUBLE PRECISION DX
C  DATA I/1/
C  IF ( U .GE. X(1) ) I = 1
C  IF ( U .LT. X(I) ) GO TO 10
C  IF ( U .LE. X(I+1) ) GO TO 35
C
C  BINARY SEARCH
C
      10 I = 1
         J = N+1
      20 K = (I+J)/2
         IF ( U .LT. X(K) ) J = K
         IF ( U .GE. X(K) ) I = K
         IF ( J .BT. I+1 ) GO TO 30
C
C  EVALUATE SPLINE
C
      30 DX = U - X(I)
         SEVAL = Y(I) + DX*B(I) + DX**2*C(I) + DX**3*D(I)
         RETURN
      END

```

```

0C001000
0C001010
0C001020
0C001030
0C001040
0C001050
0C001060
0C001070
0C001080
0C001090
0C001100
0C001110
0C001120
0C001130
0C001140
0C001150
0C001160
0C001170
0C001180
0C001190
0C001200
0C001210
0C001220
0C001230
0C001240
0C001250
0C001260
0C001270
0C001280
0C001290
0C001300
0C001310
0C001320
0C001330
0C001340
0C001350
0C001360
0C001370
0C001380
0C001390
0C001400
0C001410
0C001420
0C001430
0C001440

```


C	IERR IS SET TO	32440061
C	ZERO FOR NORMAL RETURN.	32440062
C	K IF THE K-TH SINGULAR VALUE HAS NOT BEEN	32440063
C	DETERMINED AFTER 30 ITERATIONS:	32440064
C		32440065
C	RVL IS A TEMPORARY STORAGE ARRAY.	32440066
C		32440067
C	QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO G. S. GARBON,	32440068
C	APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY	32440069
C		32440070
C	-----	32440071
C MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING	32440072
C	THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.	32440073
C	MACHEP = 16.0000*(-13) FOR LONG FORM ARITHMETIC	32440074
C	ON S360	32440075
C	DATA MACHEP/Z3410000000000000/	32440076
C		32440077
C	IERR = 0	32440078
C		32440079
C	DO 100 I = 1, M	32440080
C		32440081
C	DO 100 J = 1, N	32440082
C	U(I,J) = A(I,J)	32440083
C	100 CONTINUE	32440084
C HOUSEHOLDER REDUCTION TO BIDIAGONAL FORM	32440085
C	G = 0.000	32440086
C	SCALE = 0.000	32440087
C	X = 0.000	32440088
C		32440089
C	DO 100 I = 1, M	32440090
C	L = I + 1	32440091
C	RVL(I) = SCALE * G	32440092
C	G = 0.000	32440093
C	S = 0.000	32440094
C	SCALE = 0.000	32440095
C	IF (I .GT. M) GO TO 210	32440096
C		32440097
C	DO 120 K = I, M	32440098
C	120 SCALE = SCALE + DASS(U(K,I))	32440099
C		32440100
C	IF (SCALE .EQ. 0.000) GO TO 210	32440101
C		32440102
C	DO 130 K = I, M	32440103
C	U(K,I) = U(K,I) / SCALE	32440104
C	S = S + U(K,I)**2	32440105
C	130 CONTINUE	32440106
C		32440107
C	F = U(I,I)	32440108
C	G = -DSIGN(DSQRT(S),F)	32440109
C	H = F * G - S	32440110
C	U(I,I) = F - G	32440111
C	IF (I .EQ. N) GO TO 190	32440112
C		32440113
C	DO 150 J = L, N	32440114
C	S = 0.000	32440115
C		32440116
C	DO 140 K = I, N	32440117
C	140 S = S + U(K,I) * U(K,J)	32440118
C		32440119
C		32440120

```

      F = S / H
C      DO 150 K = 1, M
      U(K,J) = U(K,J) + F * U(K,I)
150  CONTINUE
C      DO 190 K = 1, M
      U(K,I) = SCALE * U(K,I)
C      DO 210
      W(I) = SCALE * G
      G = 0.000
      S = 0.000
      SCALE = 0.000
      IF (I .GT. M .OR. I .EQ. N) GO TO 290
C      DO 220 K = 1, N
      SCALE = SCALE + DABS(U(I,K))
C      IF (SCALE .EQ. 0.000) GO TO 290
C      DO 230 K = 1, N
      U(I,K) = U(I,K) / SCALE
      S = S + U(I,K)**2
      CONTINUE
C      F = U(I,1)
      G = -DSIGN(DSQR(S),F)
      H = F * G - S
      U(I,1) = F - G
C      DO 240 K = 1, N
      RV(I,K) = U(I,K) / H
C      IF (I .EQ. N) GO TO 270
C      DO 260 J = 1, M
      S = 0.000
C      DO 280 K = 1, N
      S = S + U(I,K) * U(I,K)
C      DO 260 K = 1, N
      U(I,K) = U(I,K) - S * RV(I,K)
      CONTINUE
C      DO 290 K = 1, M
      U(I,K) = SCALE * U(I,K)
C      DO 300 K = 1, M
      U(K,I) = DMAX1(X,DABS(U(I,1)),DABS(RV(I,1)))
      CONTINUE
C      ***** ACCUMULATION OF RIGHT-HAND TRANSFORMATIONS *****
      IF (LAST.MATV) GO TO 410
      ***** FOR I=N STEP -1 UNTIL 1 DO *****
      DO 400 II = 1, N
      Y = M + 1 - II
      IF (II .EQ. N) GO TO 390
      IF (IS .EQ. 0.000) GO TO 360
      DO 320 J = 1, N
      ***** DOUBLE DIVISION AVOIDS POSSIBLE UNDERFLOW *****

```

```

32440121
32440122
32440123
32440124
32440125
32440126
32440127
32440128
32440129
32440130
32440131
32440132
32440133
32440134
32440135
32440136
32440137
32440138
32440139
32440140
32440141
32440142
32440143
32440144
32440145
32440146
32440147
32440148
32440149
32440150
32440151
32440152
32440153
32440154
32440155
32440156
32440157
32440158
32440159
32440160
32440161
32440162
32440163
32440164
32440165
32440166
32440167
32440168
32440169
32440170
32440171
32440172
32440173
32440174
32440175
32440176
32440177
32440178
32440179
32440180

```


C	320	V(I,J) = (U(I,J) / U(I,L)) / S	32440101
		DO 330 J = L, N	32440102
		S = 0.000	32440103
C			32440104
		DO 340 K = L, N	32440105
		S = S + U(I,K) * V(K,J)	32440106
C	340		32440107
		DO 350 K = L, N	32440108
		V(K,J) = V(K,J) + S = V(K,I)	32440109
C	350	CONTINUE	32440100
			32440101
C			32440102
		DO 360 J = L, N	32440103
		V(I,J) = 0.000	32440104
		V(J,I) = 0.000	32440105
C	360	CONTINUE	32440106
			32440107
C			32440108
		DO 370 J = L, N	32440109
		V(I,I) = 1.000	32440100
		G = RV(I)	32440101
		L = I	32440102
C	400	CONTINUE	32440103
		***** ACCUMULATION OF LEFT-HAND TRANSFORMATIONS *****	32440104
C	410	IF (.NOT. MATU) GO TO 510	32440203
		*****FOR I=MIN(M,N) STEP -1 UNTIL 1 DO -- *****	32440204
C		MI = M	32440205
		IF (M .LT. N) MI = N	32440206
C			32440207
		DO 500 II = 1, MI	32440208
		I = MI + 1 - II	32440209
		L = I + 1	32440210
		G = W(I)	32440211
		IF (I .EQ. N) GO TO 430	32440212
C			32440213
		DO 420 J = L, N	32440214
		U(I,J) = 0.000	32440215
C	420		32440216
		IF (G .EQ. 0.000) GO TO 475	32440217
		IF (I .EQ. MI) GO TO 440	32440218
C			32440219
		DO 450 J = L, N	32440220
		S = 0.000	32440221
C			32440222
		DO 440 K = L, M	32440223
		S = S + U(K,I) * U(K,J)	32440224
C	440	***** DCUBLE DIVISION AVOIDS POSSIBLE UNDERFLOW *****	32440225
		F = (S / U(I,I)) / G	32440226
C			32440227
		DO 450 K = I, M	32440228
		U(K,J) = U(K,J) + F * U(K,I)	32440229
C	450	CONTINUE	32440230
			32440231
C	460	DO 470 J = I, M	32440232
		U(J,I) = U(J,I) / G	32440233
C	470		32440234
		GO TO 490	32440235
C			32440236
		DO 480 J = I, M	32440237
		U(J,I) = 0.000	32440238
C	480		32440239
		DO 490 J = I, M	32440240
		U(I,I) = U(I,I) + 1.000	32440241
C	490		32440242

```

590 CONTINUE
C      ::::::::::: DIAGONALIZATION OF THE BIDIAGONAL FORM :::::::::::
510 EPS = MACHEP * X
C      ::::::::::: FOR K=N STEP -1 UNTIL 1 DO -- :::::::::::
      DO 700 KK = 1, N
        K1 = N - KK
        K = K1 + 1
        ITS = 0
C      ::::::::::: TEST FOR SPLITTING.
C      FOR L=K STEP -1 UNTIL 1 DO -- :::::::::::
520   DO 530 LL = 1, K
        LL = K - LL
        L = LL + 1
        IF (DABS(RV1(L)) .LE. EPS) GO TO 545
C      ::::::::::: RV1(L) IS ALWAYS ZERO. SO THERE IS NO EXIT
C      THROUGH THE BOTTOM OF THE LOOP :::::::::::
        IF (DABS(W(L))) .LE. EPS) GO TO 540
530   CONTINUE
C      ::::::::::: CANCELLATION OF RV1(L) IF L GREATER THAN 1 :::::::::::
540   C = 0.000
        S = 1.000
C
        DO 560 I = L, K
          F = S * RV1(I)
          RV1(I) = C + RV1(I)
          IF (DABS(F) .LE. EPS) GO TO 565
          G = W(I)
          H = DSQRT(F*F+G*G)
          W(I) = H
          C = G / H
          S = -F / H
          IF (.NOT. MATU) GO TO 560
C
          DO 550 J = 1, M
            Y = U(J,LL)
            Z = U(J,I)
            U(J,LL) = Y * C + Z * S
            U(J,I) = -Y * S + Z * C
550   CONTINUE
C
560   CONTINUE
C      ::::::::::: TEST FOR CONVERGENCE :::::::::::
565   Z = W(K)
        IF (L .EQ. K) GO TO 650
C      ::::::::::: SHIFT FROM BOTTOM 2 BY 2 MINOR :::::::::::
        IF (ITS .EQ. 10) GO TO 1000
        ITS = ITS + 1
        X = W(L)
        Y = W(K1)
        G = RV1(K1)
        H = RV1(K)
        F = ((Y - Z) * (Y + Z) + (G - H) * (G + H)) / (2.000 * H * Y)
        G = DSQRT(F*F+1.000)
        F = ((X - Z) * (X + Z) + H * (Y / (F + DSIGN(G,F)) - H)) / X
C      ::::::::::: NEXT QR TRANSFORMATION :::::::::::
        C = 1.000
        S = 1.000
C
        DO 600 I1 = L, M
          I = I1 + 1

```

```

32440241
32440242
32440243
32440244
32440245
32440246
32440247
32440248
32440249
32440250
32440251
32440252
32440253
32440254
32440255
32440256
32440257
32440258
32440259
32440260
32440261
32440262
32440263
32440264
32440265
32440266
32440267
32440268
32440269
32440270
32440271
32440272
32440273
32440274
32440275
32440276
32440277
32440278
32440279
32440280
32440281
32440282
32440283
32440284
32440285
32440286
32440287
32440288
32440289
32440290
32440291
32440292
32440293
32440294
32440295
32440296
32440297
32440298
32440299
32440300

```

```

G = RV1(I)
Y = W(I)
H = S * G
G = C * G
Z = DSQRT(F*F+H*H)
RV1(I1) = Z
C = F / Z
S = H / Z
F = X * C + G * S
G = -X * S + G * C
H = Y * S
Y = Y * C
IF (.NOT. MATV) GO TO 570

C
DO 570 J = 1, N
  X = V(J,I1)
  Z = V(J,I)
  V(J,I1) = X * C + Z * S
  V(J,I) = -X * S + Z * C
570 CONTINUE

C
575 Z = DSQRT(F*F+H*H)
  W(I1) = Z
C ***** ROTATION CAN BE ARBITRARY IF Z IS ZERO *****
  IF (Z .EQ. 0.000) GO TO 580
  C = F / Z
  S = H / Z
580 F = C * G + S * Y
  X = -S * G + C * Y
  IF (.NOT. MATU) GO TO 600

C
DO 590 J = 1, N
  ( = U(J,I1)
  Z = U(J,I)
  U(J,I1) = Y * C + Z * S
  U(J,I) = -Y * S + Z * C
590 CONTINUE

C
600 CONTINUE

C
RV1(L) = 0.000
RV1(K) = F
W(K) = X
GO TO 520

C
***** CONVERGENCE *****
650 IF (Z .GE. 0.000) GO TO 700
C ***** W(K) IS MADE NON-NEGATIVE *****
  W(K) = -Z
  IF (.NOT. MATV) GO TO 700

C
DO 690 J = 1, N
690 V(J,K) = -V(J,K)

C
700 CONTINUE

C
GO TO 1001
-C ***** SET ERRCR -- NO CONVERGENCE TO A
  SINGULAR VALUE AFTER 30 ITERATIONS *****
C
1000 IERR = K
1001 RETURN

C
***** LAST CARD OF SVD *****
END

```

```

32440301
32440302
32440303
32440304
32440305
32440306
32440307
32440308
32440309
32440310
32440311
32440312
32440313
32440314
32440315
32440316
32440317
32440318
32440319
32440320
32440321
32440322
32440323
32440324
32440325
32440326
32440327
32440328
32440329
32440330
32440331
32440332
32440333
32440334
32440335
32440336
32440337
32440338
32440339
32440340
32440341
32440342
32440343
32440344
32440345
32440346
32440347
32440348
32440349
32440350
32440351
32440352
32440353
32440354
32440355
32440356
32440357
32440358
32440359
32440360
32440361
32440362

```

CAPITULO IV

INTEGRACION NUMERICA

Existen diversos métodos numéricos que permiten efectuar la Integración y/o la diferenciación de funciones. El método más apropiado para cada problema particular dependerá principalmente de la cantidad de información disponible sobre la función. Los problemas básicos que trataremos en este capítulo serán los de las funciones reales de una sola variable x , definida sobre un intervalo $[a, b]$. Estas funciones pueden dividirse en 4 categorías:

- I) La función $f(x)$ está bien definida y puede ser evaluada en cualquier valor real de la variable x sobre el intervalo $[a, b]$.
- II) Los valores $f(x_i)$ están disponibles únicamente en ciertos puntos x_i del intervalo $[a, b]$.
- III) La definición de la función puede ser extendida analíticamente al plano de los complejos.
- IV) Una fórmula explícita de la función $f(x)$ se encuentra disponible en una representación apropiada para su manipulación simbólica.

En todas aquellas funciones que caen dentro de las dos primeras categorías, la diferenciación numérica es básicamente más difícil que la

efectuar la Intg
 más apropiado pa-
 e la cantidad de
 básicos que tra-
 ales de una sola
 Funciones pueden

ser evaluada en
 e el intervalo

ente en ciertos

noida analítica-

se encuentra dis-
 na su manipula-

as dos primeras
 ás difícil que la

es debido a que la diferenciación tiende a amplificar
 presente en la información y la integración tienda a
 uir tal error. Así mismo, en aquellos casos donde los
 acción $f(x)$ son conocidos, o pueden ser calculados con
 y donde se requieren únicamente las primeras deriva-
 n, entonces los métodos basados en funciones splines o
 micas proporcionan, en general, resultados satisfacto-
 derivadas de alto orden son deseadas, o si los valores
 presentan "ruido", entonces los resultados pueden ser ine

la tercera categoría, podemos considerar aquellas fun-
 presiones trigonométricas u otras funciones elementales
 las funciones a integrar y/o diferenciar. Finalmente,
 en el dominio de los complejos C es factible de hacerse,
 integración en C puede producir resultados satisfactorios.

enguaje de los métodos numéricos, el término "regla" o
 s empleado en la definición de aquellos algoritmos cuyas
 irven para calcular aproximadamente ciertas integrales de-
 ste capítulo estudiaremos exclusivamente aquellas reglas o
 aplicables al tipo de funciones descritas en las categorías

tángulo y del Trapezoide.

a, b] un intervalo finito para la variable x , el cual se ha

dividido en n subintervalos o p\u00e1neles $[x_i, x_{i+1}]$, $i = 1, \dots, n$.

Sean $x_1 = a$, $x_{n+1} = b$ donde

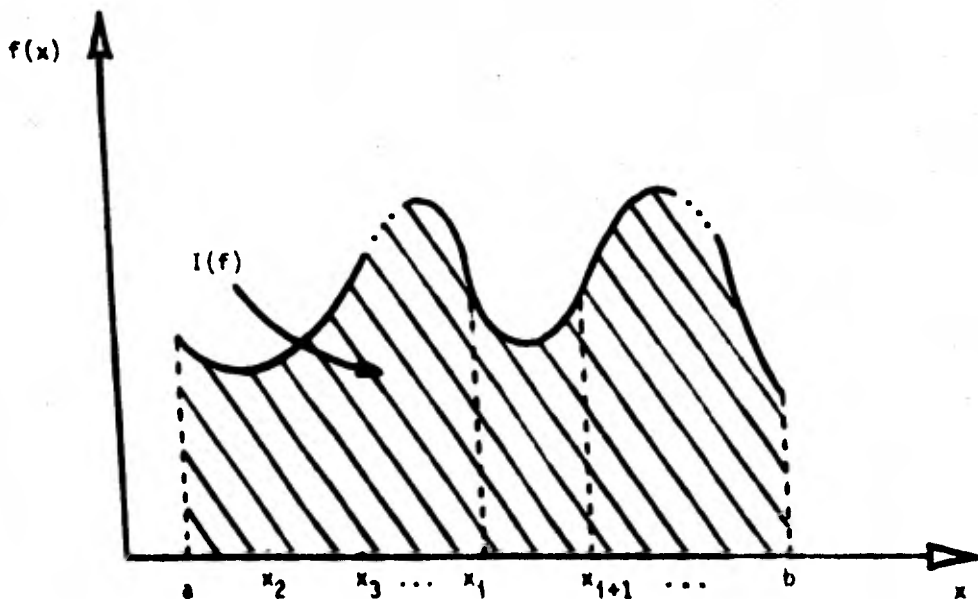
$$x_1 < x_2 < x_3 < x_4 < \dots < x_{n+1},$$

siendo $h_i = x_{i+1} - x_i$, el ancho de los p\u00e1neles, y sea $f(x)$ una funci\u00f3n definida en el mismo intervalo $[a, b]$.

Una aproximaci\u00f3n a la integral $I(f) = \int_a^b f(x) dx$ es deseada.

Tal aproximaci\u00f3n podr\u00eda derivarse a trav\u00e9s de la suma de las integrales sobre p\u00e1neles h_i , es decir:

$$I(f) = \sum_{i=1}^n I_i, \text{ donde } I_i = I_i(f) = \int_{x_i}^{x_{i+1}} f(x) dx.$$



Una regla de cuadratura simple es una fórmula que permite aproximar las integrales I_1 . Una regla de cuadratura compuesta es una fórmula que aproxima la integral $I(f)$ mediante una suma de reglas de cuadratura simples aplicadas a cada integral I_1 .

Dos de las reglas de cuadratura simple más usadas son la regla del rectángulo y la del trapecoide. La regla del rectángulo utiliza en su estimación los valores de la función $f(x)$ en los puntos medios de los paneles.

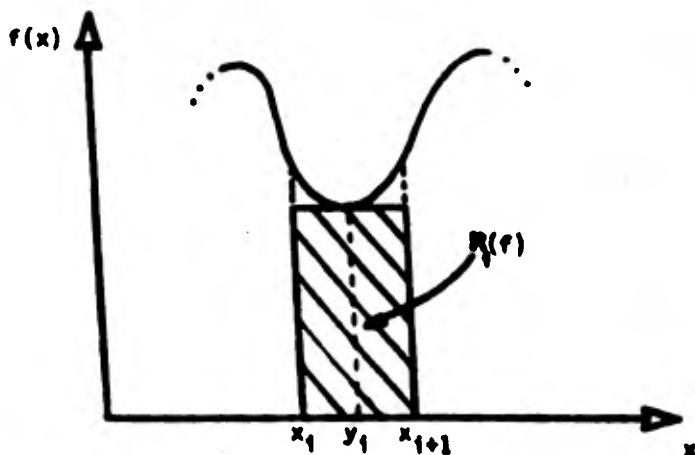
$$y_1 = \frac{x_i + x_{i+1}}{2}, \quad i = 1, \dots, n,$$

aproximando cada integral I_1 mediante el área de un rectángulo de base h_1 y de altura $f(y_1)$, es decir

$$I_1 = h_1 f(y_1)$$

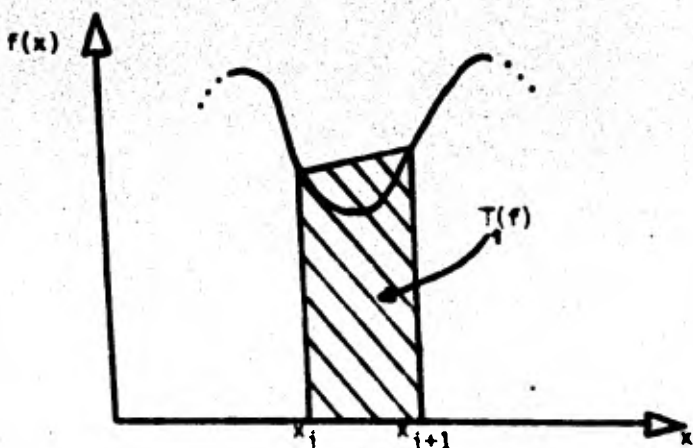
Según lo anterior, la regla compuesta del rectángulo resulta

$$R(f) = \sum_{i=1}^n h_i f(y_i)$$



La regla del trapecoide emplea los valores de la función en los extremos del pñel y aproxima cada integral a través del área de un trapecoide cuya base es h_i y cuya altura varía linealmente desde $f(x_i)$ a $f(x_{i+1})$, es decir

$$I_i(f) = h_i \left(\frac{f(x_i) + f(x_{i+1})}{2} \right)$$



La regla compuesta del trapecoide resulta entonces

$$T(f) = \sum_{i=1}^n h_i \frac{f(x_i) + f(x_{i+1})}{2}$$

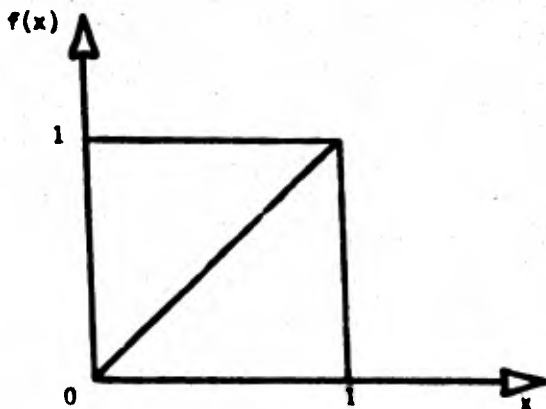
Puede mostrarse que si $f(x)$ es una función continua (o simplemente integrable según Riemann) en el intervalo $[a, b]$, y si $h = \max_i h_i$, entonces ambas reglas convergen al resultado exacto a medida que el ancho de

los subintervalos tiende a cero,

$$\lim_{h \rightarrow 0} R(f) = I(f) \quad \text{y} \quad \lim_{h \rightarrow 0} T(f) = I(f).$$

La pregunta obligada es ahora: ¿Qué tan rápido convergen estas reglas? La regla del rectángulo está basada en la aproximación de una constante en cada subintervalo, mientras que la regla del trapecoide está basada en una interpolación lineal. Intuitivamente, uno podría esperar mayor exactitud en la regla trapezoidal. Sin embargo, observamos que pasa en el caso de la función

$$f(x) = x \text{ definida sobre el intervalo } [0, 1].$$



La regla trapezoidal obviamente no da error, ya que la interpolación lineal concuerda con la función $f(x)$. Sin embargo, la regla rectangular da un resultado también sin error a pesar del hecho de que la función interpoladora en el punto $x = \frac{1}{2}$ no concuerda con la función $f(x)$. El error promedio es, en ambos casos, cero.

Es este ejemplo típico, o simplemente es suerte de la cuadratura del rectángulo? ¿Qué regla es en general más exacta?

Para dar respuesta a estas interrogantes, considere una función $f(x)$ y su expresión en series de Taylor con respecto al punto y_1 localizado en el centro del panel $[x_i, x_{i+1}]$, esto es

$$f(x) = f(y_1) + (x-y_1) f'(y_1) + \frac{1}{2!} (x-y_1)^2 f''(y_1) + \frac{1}{3!} (x-y_1)^3 f'''(y_1) + \frac{1}{4!} (x-y_1)^4 f^{(4)}(y_1) + \dots$$

La primer suposición consiste en considerar que los términos " ... " son menos significativos que los términos mostrados explícitamente.

Observemos lo siguiente:

$$\text{en la integral } \int_{x_i}^{x_{i+1}} f(x) dx = \int_{x_i}^{x_{i+1}} [f(y_1) + (x-y_1) f'(y_1) + \dots] dx$$

se obtiene que las potencias impares al ser integradas dancero:

$$\int_{x_i}^{x_{i+1}} (x-y_1)^p dx = \begin{cases} h_i, & \text{si } p = 0 \\ 0, & \text{si } p = 1 \\ \frac{h_i^3}{12}, & \text{si } p = 2 \\ 0, & \text{si } p = 3 \\ \frac{h_i^5}{80}, & \text{si } p = 4 \end{cases}$$

Por lo tanto, la integral I_i puede ser expresada como

$$\int_{x_i}^{x_{i+1}} f(x) dx = \underbrace{h_i f(y_i)}_{R(f)} + \frac{1}{24} h_i^3 f''(y_i) + \frac{1}{1920} h_i^5 f^{(4)}(y_i) + \dots$$

donde, para valores de h_i pequeños, el error en la aproximación por la cuadratura del rectángulo (R) será $\frac{1}{24} h_i^3 f''(y_i)$, más algunos términos menos significativos.

Volviendo a la serie de Taylor y substituyendo en ella el valor $x=x_i$ primeramente, y posteriormente el valor $x = x_{i+1}$, tenemos:

$$f(x_i) = f(y_i) - \frac{1}{2} h_i f'(y_i) + \frac{1}{8} h_i^2 f''(y_i) - \frac{1}{48} h_i^3 f'''(y_i) + \frac{1}{384} h_i^4 f^{(4)}(y_i) + \dots$$

y

$$f(x_{i+1}) = f(y_i) + \frac{1}{2} h_i f'(y_i) + \frac{1}{8} h_i^2 f''(y_i) + \frac{1}{48} h_i^3 f'''(y_i) + \frac{1}{384} h_i^4 f^{(4)}(y_i) + \dots$$

de donde sumando las dos expresiones anteriores se obtiene que

$$\frac{f(x_i) + f(x_{i+1})}{2} = f(y_i) + \frac{1}{8} h_i^2 f''(y_i) + \frac{1}{384} h_i^4 f^{(4)}(y_i) + \dots$$

Combinando esto con la expresión anterior,

$$\int_{x_i}^{x_{i+1}} f(x) dx = h_i f(y_i) + \frac{1}{24} h_i^3 f''(y_i) + \frac{1}{1920} h_i^5 f^{(4)}(y_i) + \dots$$

y considerando que

$$h_i f(y_i) = h_i \left(\frac{f(x_i) + f(x_{i+1})}{2} \right) - \frac{1}{8} h_i^3 f''(y_i) - \frac{1}{384} h_i^5 f^{(4)}(y_i) + \dots$$

se obtiene que

$$\int_{x_i}^{x_{i+1}} f(x) dx = h_i \underbrace{\left(\frac{f(x_i) + f(x_{i+1}))}{2} \right)}_{T(f)} - \frac{1}{12} h_i^3 f''(y_i) - \frac{1}{480} h_i^5 f^{(5)}(y_i) + \dots$$

Para valores pequeños de h_i , el error en la aproximación de la regla trapezoidal (T) en un panel es:

$$- \frac{1}{12} h_i^3 f''(y_i),$$

más otros términos menos significativos. El error total de cada regla será la suma de los errores en todos los paneles, es decir, haciendo

$$E = \frac{1}{24} \sum_{i=1}^n h_i^3 f''(y_i)$$

$$F = \frac{1}{1920} \sum_{i=1}^n h_i^5 f^{(5)}(y_i)$$

el error total resulta:

$$I(f) = R(f) + E + F + \dots \quad (\text{Rectángulo})$$

$$I(f) = T(f) - 2E - 4F + \dots \quad (\text{Trapezoide})$$

Para valores de $h_i < 1$ entonces $h_i^5 \ll h_i^3$ y por lo tanto $F \ll E$, provisto que $f^{(5)}$ no se comporte ruidosamente.

Conclusiones

- a) R es cerca de 2 veces más exacta que T.

b) La diferencia entre los valores de $R(f)$ y $T(f)$ puede emplearse en la estimación del error en la integración

$$R(f) - T(f) = -3E - 5F = -3E$$

c) Duplicando el número de pñeles $h_i \rightarrow \frac{h_i}{2}$, se cuadruplica la exactitud ya que, por una parte, E se reduce por un factor de $\frac{1}{8}$, y por otra, el número de pñeles se incrementa en 2, lo cual hace que el término total se vea reducido en $\frac{1}{4}$. Estrictamente hablando, el factor no es $\frac{1}{4}$, ya que la función f'' no es generalmente constante y los términos de mayor orden ejercen alguna influencia. Desde un punto de vista práctico, en ambos casos, R y T , al doblar el número de pñeles, cuadruplican su exactitud.

Esta técnica de doblar repetidamente el número de pñeles y de calcular mediante R y T el error, puede programarse para producir un método, el cual automáticamente pueda determinar el número de pñeles necesarios, de tal forma que la integral aproximada sea computada dentro de cierta tolerancia de error prescrita. Esta técnica será empleada en reglas de cuadratura más sofisticadas.

Cuadratura de Simpson.

En la sección anterior hemos definido las cuadraturas R y T

$$R(f) = \sum_{i=1}^n h_i f(y_i), \quad y_i = \frac{x_i + x_{i+1}}{2}$$

$$T(f) = \sum_{i=1}^n \frac{h_i}{2} [f(x_{i+1}) + f(x_i)]$$

y además hemos probado que los errores de estas cuadraturas están dados por:

$$I(f) - R(f) = E + F + \dots$$

$$I(f) - T(f) = -2E - 4F + \dots$$

donde

$$E = \frac{1}{24} \sum_{i=1}^n h_i^3 f''(y_i) \quad \text{y} \quad F = \frac{1}{1920} \sum_{i=1}^n h_i^3 f'''(y_i).$$

Combinando estas dos cuadraturas podemos producir otra cuadratura donde el error no contenga los términos E.

$$\text{Sea } S(f) = \frac{2}{3} R(f) + \frac{1}{3} T(f)$$

$$\text{ó } S(f) = \sum_{i=1}^n \frac{1}{6} h_i [f(x_i) + 4f(y_i) + f(x_{i+1})]$$

Esta cuadratura recibe el nombre de Simpson y puede derivarse, como en los dos casos anteriores, integrando pieza a pieza una función, parabólica en este caso, la cual interpole los datos.

El error puede obtenerse directamente de las fórmulas de los errores en R y T.

$$\begin{aligned} I(f) - S(f) &= \frac{2}{3} [I(f) - R(f)] + \frac{1}{3} [I(f) - T(f)] \\ &= \frac{2}{3} (E + F + \dots) + \frac{1}{3} (-2E - 4F + \dots) \\ &= \left(\frac{2}{3} - \frac{2}{3}\right) E + \left(\frac{2}{3} - \frac{4}{3}\right) F + \dots \end{aligned}$$

$$= -\frac{2}{3}F + \dots$$

$$= -\frac{1}{2880} \sum_{i=1}^n h_i^5 f^{(5)}(y_i) + \dots$$

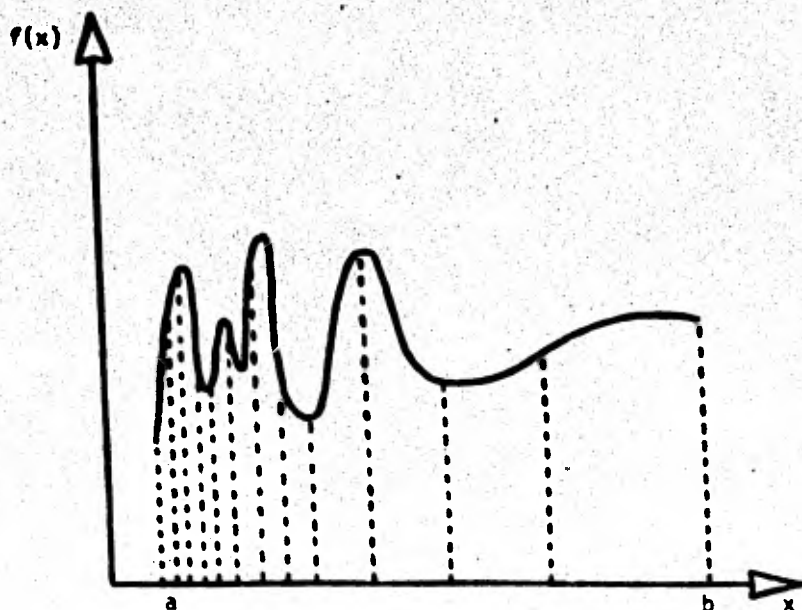
La cuadratura $S(f)$ está basada en una interpolación de segundo grado y su error contiene derivadas de cuarto orden, por lo tanto, $S(x)$ será una cuadratura exacta para funciones cúbicas.

Cuando la longitud del intervalo es dividida, $h_1 \rightarrow \frac{h_1}{2}$, el término h_1^5 decrece por un factor de $\frac{1}{32}$, al mismo tiempo el número de sumandos se duplica siendo la reducción del error total, un factor aproximadamente igual a $\frac{1}{16}$.

Esta técnica de combinar dos aproximaciones con errores similares para obtener una aproximación más exacta puede ser continuada. Por ejemplo, los valores de $S(f)$ pueden combinarse con las cuadraturas R o T , para obtener errores del orden de h_1^7 y $f^{(7)}(y_1)$. La regla general que describe este procedimiento se conoce con el nombre de cuadratura de Romberg.

Regla de Cuadratura Adaptada.

Esta regla es un algoritmo numérico el cual, empleando una o dos cuadraturas básicas, determina automáticamente el tamaño del subintervalo de tal manera que ciertos requisitos de exactitud sean satisfechos. El principio básico de esta regla consiste en seleccionar un tamaño relativamente grande de intervalos donde la función a integrar tenga un



comportamiento suave, y un intervalo pequeño donde la función cambie abruptamente.

Esta regla requiere de la siguiente información:

- (i) un intervalo finito $[a, b]$,
- (ii) una función a integrar $f(x)$ definida en el intervalo $[a, b]$,
- (iii) y un error de tolerancia ϵ .

El algoritmo de cuadratura adaptada calculará una cantidad Q tal que

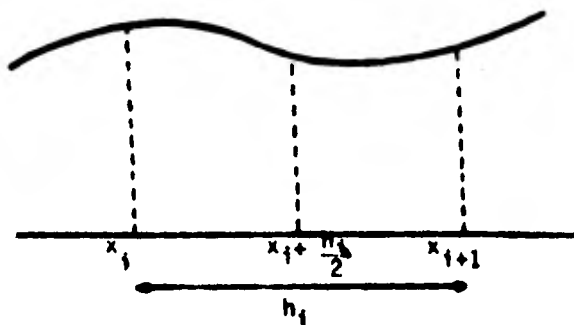
$$|Q - \int_a^b f(x)dx| \leq \epsilon$$

El intervalo de integración se divide en n subintervalos

$[x_i, x_{i+1}]$, el número de subintervalos dependerá de ξ y de la función $f(x)$.

Sea, como antes, $h_i = x_{i+1} - x_i$ ($x_1 = a$ y $x_{n+1} = b$). El esquema o regla de la cuadratura adaptada aplica dos cuadraturas diferentes a cada subintervalo. Sean los dos resultados de estas cuadraturas P_i y Q_i . Por ejemplo, un esquema basado en la cuadratura de Simpson emplearía la fórmula de 2- pánels siguiente:

$$P_i = \frac{h_i}{6} \left[f(x_i) + 4f\left(x_i + \frac{h_i}{2}\right) + f(x_i + h_i) \right]$$



y un esquema de Simpson con 4-pánels emplearía

$$Q_i = \frac{h_i}{12} \left[f(x_i) + 4f\left(x_i + \frac{h_i}{4}\right) + 2f\left(x_i + \frac{h_i}{2}\right) + 4f\left(x_i + 3\frac{h_i}{4}\right) + f(x_i + h_i) \right].$$

P_i y Q_i son dos aproximaciones distintas de la integral

$$\int_{x_i}^{x_{i+1}} f(x) dx = I_i.$$

El principio de la cuadratura adaptada consiste en comparar estas dos aproximaciones y obtener de la comparación una medida de la exactitud de cada aproximación.

Si el resultado es aceptable una de las dos se toma como el valor de la integral, de lo contrario se subdivide el intervalo en dos o más partes y el proceso se repite en los subintervalos más pequeños.

Es importante notar que la evaluación de la función $f(x)$ se simplifica, ya que como puede observarse los términos P_i y Q_i tienen ciertas partes comunes. Q_i contiene únicamente dos términos que no están en P_i . En lo que resta, deberá entenderse que Q_i se obtiene aplicando la regla de P_i dos veces, una vez para cada mitad del intervalo. Esta idea facilitará el análisis de la regla adaptada.

Consideremos por un momento que la regla de P_i da la respuesta exacta cuando la función integrante es un polinomio de grado $p-1$, o equivalentemente, $f^{(p)}(x) = 0$. Expandiendo por series de Taylor con respecto al punto central de un subintervalo, puede demostrarse que

$$I_i - P_i = ch_i^{p+1} f^{(p)} \left(x_i + \frac{h_i}{2} \right) + \dots$$

(ver esta misma expresión en la cuadratura del rectángulo pg. 98).

Como hemos supuesto que Q_i es la suma de dos P_i 's calculados en subintervalos de longitud $\frac{h_i}{2}$, podemos concluir que

$$I_1 - Q_1 = c \left(\frac{h_1}{2}\right)^{p+1} \left[f^{(p)} \left(x_1 + \frac{h_1}{4}\right) + f^{(p)} \left(x_1 + 3 \frac{h_1}{4}\right) \right] + \dots$$

Pero dado que

$$f^{(p)} \left(x_1 + \frac{h_1}{4}\right) + f^{(p)} \left(x_1 + 3 \frac{h_1}{4}\right) = 2f^{(p)} \left(x_1 + \frac{h_1}{2}\right) + \dots$$

los dos errores estarán relacionados por

$$I_1 - Q_1 = \frac{2}{2^{p+1}} (I_1 - P_1) + \dots = \frac{1}{2^p} (I_1 - P_1) + \dots$$

Esto indica que al partir el subintervalo, el error decrecerá cerca de 2^p veces.

Rearreglando términos:

$$I_1 \left(1 - \frac{1}{2^p}\right) = Q_1 - \frac{1}{2^p} P_1 + \dots$$

$$I_1 (2^p - 1) = 2^p Q_1 - P_1 + \dots$$

$$I_1 - \frac{1}{2^p - 1} Q_1 = \frac{2^p}{2^p - 1} Q_1 - \frac{P_1}{2^p - 1} - \frac{1}{2^p - 1} Q_1 + \dots$$

$$I_1 - \frac{Q_1}{2^p - 1} = Q_1 - \frac{P_1}{2^p - 1} + \dots$$

$$\frac{1}{2^p - 1} (P_1 - Q_1) = Q_1 - I_1 + \dots$$

Finalmente, el error en la expresión más exacta Q_1 es cerca de $\frac{1}{2^p - 1}$ veces la diferencia entre las dos aproximaciones.

El objetivo esencial será pues, dividir cada subintervalo hasta que

la desigualdad

$$\frac{1}{2^{p-1}} |P_1 - Q_1| \leq \frac{h_1}{b-a} \varepsilon \quad \text{sea satisfecha.}$$

ε es la tolerancia requerida. La expresión $\frac{h_1}{b-a} \varepsilon$ resultará evidente a continuación. Si todo el intervalo $[a, b]$ fuera cubierto por n subintervalos, entonces el resultado sería:

$$Q = \sum_{i=1}^n Q_i$$

Tomando en cuenta lo anterior y despreciando los términos de alto orden tenemos:

$$\begin{aligned} |Q - \int_a^b f(x) dx| &= \left| \sum_{i=1}^n Q_i - I_1 \right| \\ &\leq \sum_{i=1}^n |Q_i - I_i| \\ &\leq \frac{1}{2^{p-1}} \sum_{i=1}^n |P_i - Q_i| \\ &\leq \frac{1}{2^{p-1}} \frac{2^{p-1}}{b-a} \varepsilon \sum_{i=1}^n h_i = \varepsilon \end{aligned}$$

lo cual coincide con lo planteado arriba.

No se debe pasar por alto que en este análisis se requiere de la suposición de que $f^{(p)}(x)$ sea una función continua y proporcional a $h_i^{p+1} f^{(p)}(x)$.

Hasta aquí y por simplicidad hemos empleado el criterio de error absoluto.

es decir

$$|Q - \int f| < \epsilon$$

En ocasiones el criterio de error relativo, el cual es independiente de la escala de f , es empleado

$$\frac{|Q - \int f|}{|\int f|} < \epsilon$$

Esto complica el análisis ya que el denominador $|\int f|$ puede ser cero. Además una buena aproximación del denominador solo se obtendría hasta el final del cálculo.

Otro criterio podría ser

$$\frac{|Q - \int f|}{\int |f|} < \epsilon$$

en cuyo caso el denominador no sería cero, al menos que la función $f(x)$ fuese idéntica a cero y en cuyo caso el análisis sería igualmente complicado.

De cualquier manera el usuario de la cuadratura adaptada debe estar siempre conciente del tipo de error que se está empleando en la subrutina paquete que utilice para aproximar integrales.

Al final del capítulo se enlista la subrutina QUANCB la cual aproxima integrales por el método de la cuadratura adaptada.

Cuadratura Spline.

La técnica de interpolación de las funciones cúbicas "spline" puede emplearse en la obtención de resultados interesantes en problemas de integración.

Para el intervalo $x_1 \leq x \leq x_{i+1}$, sea

$$s(x) = f_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3$$

la función spline que interpola $f(x)$ en el intervalo $[x_1, x_{i+1}]$.

Si $a = x_1$ y $b = x_{n+1}$, la integral $\int_a^b f(x)dx$ puede ser aproximada por la integral $\int_a^b s(x)dx$.

$$\int_a^b s(x)dx = \sum_{i=1}^n (h_i f_1 + \frac{1}{2} h_i^2 b_1 + \frac{1}{3} h_i^3 c_1 + \frac{1}{4} h_i^4 d_1)$$

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

Los coeficientes b_1 , c_1 y d_1 pueden obtenerse por medio de la técnica cubic-spline ya descrita. (Dado que existen $n+1$ nodos, la subrutina SPLINE será llamada con $N = n + 1$).

La función $s(x)$ puede representarse también como (ver capítulo III):

$$s(x) = w f_{i+1} + \bar{w} f_i + h_i^2 [(w^3 - \bar{w}) \sigma_{i+1} + (\bar{w}^3 - w) \sigma_i]$$

dado

$$w = \frac{x - x_1}{h_1} = 1 - \bar{w}, \quad \sigma_1 = \frac{s''(x_1)}{6} = \frac{c_1}{3}$$

y donde las incógnitas σ_1 son calculadas a partir de un sistema de ecuaciones tridiagonal.

Para obtener la fórmula de la cuadratura observemos que:

$$\int_{x_1}^{x_{1+1}} s(x) dx = h_1 \int_0^1 s(w) dw$$

$$\text{Además, } \int_0^1 w dw = \frac{1}{2}, \quad \text{y } \int_0^1 (w^3 - w) dw = \frac{-1}{4}$$

Entonces, substituyendo en la primera integral se obtiene que

$$\int_{x_1}^{x_{1+1}} s(x) dx = h_1 \left(\frac{f_1 + f_{1+1}}{2} \right) - h_1^3 \left(\frac{\sigma_1 + \sigma_{1+1}}{4} \right)$$

En otras palabras, la fórmula de la cuadratura spline es la misma que la fórmula de la cuadratura trapezoidal, más un "término de corrección", el cual contiene a la variable σ_1 . Una fórmula más conveniente para el cálculo en el intervalo total sería

$$\int_a^b s(x) dx = \sum_{i=1}^n h_i \frac{f_i + f_{i+1}}{2} - h_i^3 \frac{c_i + c_{i+1}}{12}.$$

En este caso, únicamente los datos (x, f) y el arreglo de segundas derivadas c_i , calculado por SPLINE, serían requeridos.

Con respecto a la exactitud podemos observar que

$$h_i^3 \frac{c_i + c_{i+1}}{12} = h_i^3 \frac{s''(x_i) + s''(x_{i+1})}{24}$$

$$= \frac{h_i^3}{12} f''(y_i) \quad \text{donde } y_i = \frac{x_i + x_{i+1}}{2}$$

Por lo tanto e interesantemente, el término de corrección provisto por la cuadratura SPLINE aproxima al término error de la cuadratura trapezoidal. En la literatura no se han reportado resultados prácticos donde se haya aplicado la cuadratura SPLINE; sin embargo, por los resultados mostrados aquí, ésta parece ser de gran utilidad.

Por último, debemos observar que la cuadratura spline no presenta la forma convencional de otras cuadraturas, es decir,

$$\int_a^b f(x) dx = \sum_{i=1}^{n+1} \alpha_i f(x_i).$$

En la cuadratura spline cada coeficiente α_i dependerá de los valores $f(x_i)$ en forma más complicada.

Aproximación de Integrales por el Método de Monte Carlo.

Supongamos que se desea integrar la función $g(x)$ en el intervalo $a \leq x \leq b$,

$$I = \int_a^b g(x) dx.$$

La metodología a emplear consiste en seleccionar una función de densidad arbitraria $P_V(x)$ de una variable aleatoria V definida en el intervalo $[a, b]$, es decir

$$P_V(x) \geq 0 \quad \forall x \in [a, b]$$

$$\int_a^b P_V(x) dx = 1$$

Posteriormente, además de la variable aleatoria V , se requiere definir la siguiente función aleatoria

$$H = \frac{g(v)}{P_V(v)}$$

Ahora bien, la definición de valor esperado $E[f_V(v)] = \int_a^b f(v)P_V(v)dv$ puede aplicarse directamente en la función H , esto es

$$E[H] = \int_a^b \frac{g(v)}{P_V(v)} P_V(v)dv = I$$

En vez de una variable aleatoria H , consideremos ahora n variables aleatorias independientes e idénticamente distribuidas

$$H_1, H_2, H_3, \dots, H_n \quad \text{tales que}$$

$$E [H_i] = I, \quad \forall i = 1, \dots, n$$

$$\text{Var} [H_i] = \text{Var} [H], \quad \forall i = 1, \dots, n$$

Si formamos una nueva variable aleatoria H^* , definida como

$$H^* = H_1 + \dots + H_n, \text{ entonces}$$

$$E[H^*] = nI$$

$$\text{Var}[H^*] = n\text{Var}[H]$$

Según el teorema del Límite Central, la distribución de la nueva variable aleatoria H^* tenderá a la distribución normal, acercándose a $N(\mu, \sigma)$ a medida que $n \rightarrow \infty$, es decir,

$$\Pr\{|H^* - nI| < 3\sqrt{\text{Var } H^*}\} = 0.997$$

$$\delta \quad \Pr\left\{\left|\sum_{i=1}^n H_i - nI\right| < 3\sqrt{n\text{Var } H}\right\} = 0.997.$$

Dividiendo entre n

$$\Pr\left\{\left|\frac{1}{n}\sum_{i=1}^n H_i - I\right| < 3\sqrt{\frac{\text{Var } H}{n}}\right\} = 0.997.$$

lo que también se interpreta como una variable aleatoria $\frac{1}{n}\sum_{i=1}^n H_i$ cercana al valor I con probabilidad 0.997 y con un error menor a $3\sqrt{\frac{\text{Var } H}{n}}$.

A medida que n crece se observa que

$$\frac{1}{n}\sum_{i=1}^n H_i = \frac{1}{n}\sum_{i=1}^n \frac{g(v_i)}{P_V(v_i)} = I.$$

Ahora bien, según hemos visto en el cálculo aproximado de I , pue-

de emplearse la distribución de cualquier variable aleatoria V . En cualquier caso, el estimador será insesgado ya que

$$E[H] = E\left[\frac{g(V)}{P_V(V)}\right] = I$$

y su varianza será $\text{Var}[H]$. La pregunta que ahora se presenta es, ¿cómo reducir la varianza del error?, esto es,

$$\min \text{Var}[H] = \min(E[H^2] - I^2) = \min\left(\int_a^b \frac{g^2(x)}{P_V(x)} dx - I^2\right).$$

A continuación mostraremos que $\text{Var}[H]$ se minimiza si se elige una función de densidad $P_V(x)$ proporcional a $|g(x)|$.

Haciendo $u = \frac{g(x)}{\sqrt{P_V(x)}}$ y $v = \sqrt{P_V(x)}$ en la desigualdad de Schwarz

$$\left| \int_a^b |u(x)v(x)| dx \right|^2 \leq \int_a^b u^2(x) dx \int_a^b v^2(x) dx$$

se obtiene:

$$\left[\int_a^b |g(x)| dx \right]^2 \leq \int_a^b \frac{g^2(x)}{P_V(x)} dx \int_a^b P_V(x) dx = \int_a^b \frac{g^2(x)}{P_V(x)} dx \quad (1)$$

y de acuerdo a la definición de la varianza se tiene entonces que

$$\text{Var}[H] \geq \left[\int_a^b |g(x)| dx \right]^2 - I^2.$$

Seleccionando $P_V(x)$ proporcional a $|g(x)|$, es decir:

$$P_V(x) = \frac{|g(x)|}{\int_a^b |g(x)| dx}$$

obtendremos que la expresión (1) se convierte en la identidad, ya que

$$\int_a^b \frac{g^2(x) dx}{\int_a^b |g(x)| dx} = \int_a^b \frac{|g(x)| |g(x)| dx}{|g(x)|} \cdot \int_a^b |g(x)| dx = \left[\int_a^b |g(x)| dx \right]^2,$$

y por consiguiente $\text{Var}[H]$ se convierte en igualdad.

En la práctica no se seleccionan funciones $P_V(x)$ complejas ya que ello haría sumamente laborioso el método Monte Carlo. Por otro lado, la función $g(x)$ puede ser una gafa para la selección de la función $P_V(x)$.

Así mismo, las integrales de forma

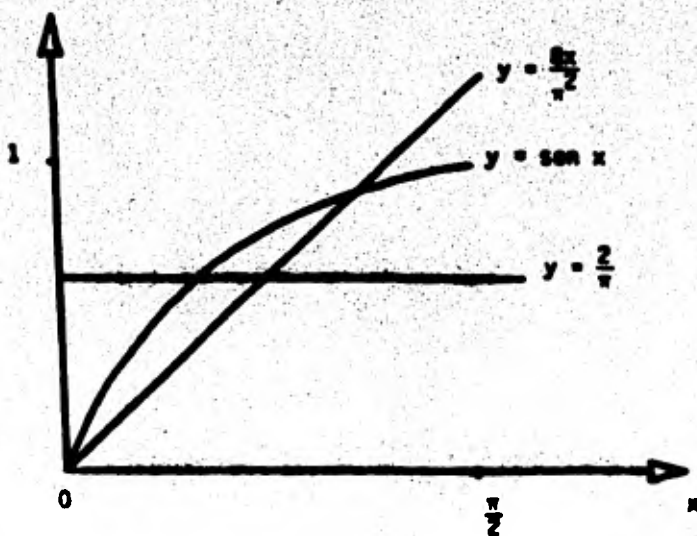
$$I = \int_a^b g(x) dx$$

no se resuelven por el método Monte Carlo, sino por métodos de cuadratura. En cambio, en el caso de integrales múltiples la situación cambia, el método de la cuadratura se vuelve sumamente complejo, mientras que Monte Carlo permanece prácticamente inalterable.

Ejemplo de aplicación:

Sea la integral $I = \int_0^{\frac{\pi}{2}} \text{sen } x \, dx$

cuya solución sabemos es 1.



Con propósito de ilustración considérense las dos funciones de densidad siguientes:

$$(I) y = p_V(x) = \frac{2}{\pi} \quad \text{y} \quad (II) p_V(x) = \frac{8x}{\pi^2}$$

La función de densidad (I) pertenece a una variable aleatoria con distribución uniforme definida en el intervalo $[0, \frac{\pi}{2}]$. Intuitivamente se observa que la función de densidad (II) satisface mejor las recomendaciones de proporcionalidad, por lo tanto, de ella se esperarán los mejores resultados. Para cada caso se tiene:

$$(a) p_V(x) = \frac{2}{\pi}$$

$$v = v_{\min} + U (v_{\max} - v_{\min})$$

$$v = \frac{U\pi}{2}, \text{ ya que } v_{\max} = \frac{\pi}{2} \text{ y } v_{\min} = 0.$$

$$\text{Según lo anterior, } I = \frac{1}{n} \sum_{i=1}^n \frac{g(v_i)}{P_V(v_i)} = \frac{\pi}{2n} \sum_{i=1}^n \text{sen } v_i.$$

Generando 10 valores aleatorios U_1, \dots, U_{10} se producen, según la fórmula $v = \frac{U\pi}{2}$, otros 10, v_1, \dots, v_{10} y el valor de la integral resulta

$$I = 0.952$$

$$(b) P_V(x) = \frac{8x}{\pi^2}$$

$$\int_0^v \frac{8x}{\pi^2} dx = U$$

$$v = \frac{\pi}{2} \sqrt{U}$$

$$I = \frac{\pi^2}{8n} \sum_{i=1}^n \frac{\text{sen } v_i}{v_i}$$

para $n = 10$, $I = 1.016$

La medida del orden del error estaría dada por $3\sqrt{\frac{\text{Var}(H)}{n}}$, un resultado más práctico sería $0.675\sqrt{\frac{\text{Var}(H)}{n}}$, entonces por uno de los casos vistos se tendría el error siguiente:

$$\text{Var } [H]_1 = 0.256$$

$$\text{Var } |M|_{II} = 0.016$$

Integración Numérica a Través de Polinomios Ortogonales.

Polinomios Ortogonales.

Definición: Dos funciones, $g_m(x)$ y $g_n(x)$, obtenidas a partir de una familia de funciones $\{g_k(x)\}$, se dicen ser ortogonales con respecto a la función $w(x)$ en el intervalo $[a, b]$, si ellas satisfacen que:

$$\int_a^b w(x) g_m(x) g_n(x) dx = 0, \quad \forall m \neq n \quad y$$

$$\int_a^b w(x) g_m^2(x) dx = c(m) \neq 0 \quad \forall m.$$

Como ejemplo de estas funciones pueden citarse las funciones seno y coseno ($\sin Rx$, $\cos Rx$).

La propiedad de ortogonalidad se interpreta, en términos de espacios vectoriales, como la perpendicularidad existente entre dos vectores, en un espacio n -dimensional (n muy grande) (Figura 14). Los elementos o vectores de este espacio vectorial no son otros que los formados con la familia $\{g_k(x)\}$.

La familia de polinomios $\{x^k\}$, $k = 0, 1, 2, \dots$, define un espacio vectorial cuyos elementos no son ortogonales. Otras familias, tales como las de los polinomios de Legendre, Laguerre, Chebyshev y Hermite, además de definir un espacio vectorial, sus elementos son ortogonales.

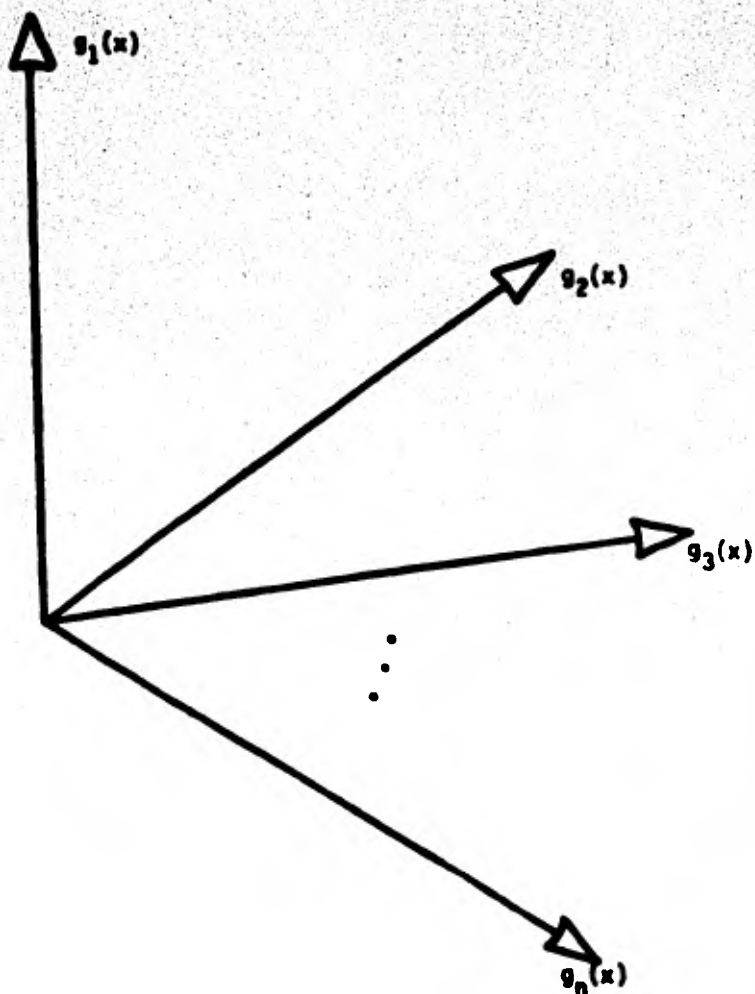


Figura 14.- Familia de funciones ortogonales representada por medio de un espacio vectorial n -dimensional. Cada vector en el espacio representa una función.

(i) Polinomios de Legendre $\{P_n(x)\}$.

Los polinomios de Legendre satisfacen la propiedad de ortogonalidad en el intervalo $[-1, 1]$ con función peso $w(x) = 1$, es decir:

$$\int_{-1}^1 P_m(x) P_n(x) dx = \begin{cases} 0, & m \neq n \\ c(m), & m = n \end{cases}$$

Los tres primeros polinomios de Legendre se definen como:

$$P_0(x) = 1$$

$$P_1(x) = x$$

$$P_2(x) = \frac{1}{2} (3x^2 - 1)$$

Los polinomios de Legendre subsecuentes pueden obtenerse a partir de la fórmula de recurrencia siguiente:

$$P_n(x) = \frac{2n-1}{n} x P_{n-1}(x) - \frac{n-1}{n} P_{n-2}(x)$$

(ii) Polinomios de Laguerre $\{L_n(x)\}$

Los polinomios de Laguerre satisfacen la propiedad de ortogonalidad en el intervalo $[0, \infty]$ con función peso $w(x) = e^{-x}$, o sea que:

$$\int_0^{\infty} e^{-x} L_m(x) L_n(x) dx = \begin{cases} 0, & m \neq n \\ c(m), & m = n \end{cases}$$

Los tres primeros polinomios de Laguerre son iguales a:

$$L_0(x) = 1$$

$$L_1(x) = 1 - x$$

$$L_2(x) = x^2 - 4x + 2$$

La fórmula de recurrencia de los polinomios de Laguerre es:

$$L_n(x) = (2n - x - 1) L_{n-1}(x) - (n - 1)^2 L_{n-2}(x).$$

(iii) Polinomios de Chebyshev $\{T_n(x)\}$.

Los polinomios de Chebyshev son ortogonales en el intervalo $[-1, 1]$ con función peso $w(x) = (1 - x^2)^{-\frac{1}{2}}$, dando

$$\int_{-1}^1 (1 - x^2)^{-\frac{1}{2}} T_m(x) T_n(x) dx = \begin{cases} 0, & m \neq n \\ c(m), & m = n \end{cases}$$

Los tres primeros polinomios de Chebyshev se definen así:

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_2(x) = 2x^2 - 1$$

y la fórmula de recurrencia es:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$$

iv) Polinomios de Hermite $\{H_n(x)\}$.

Los polinomios de Hermite satisfacen la propiedad de ortogonalidad

en el intervalo $(-\infty, \infty)$ con función peso $w(x) = e^{-x^2}$, esto es:

$$\int_{-\infty}^{\infty} e^{-x^2} H_m(x) H_n(x) dx = \begin{cases} 0, & m \neq n \\ c(m), & m = n \end{cases}$$

Los tres primeros polinomios de Hermite son:

$$H_0(x) = 1$$

$$H_1(x) = 2x$$

$$H_2(x) = 4x^2 - 2$$

con fórmula de recurrencia $H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x)$

Cada uno de los polinomios definidos arriba es un polinomio con coeficientes reales, de grado n en la variable x y con n raíces reales distintas, las cuales caen dentro del intervalo de integración de cada polinomio. Como ejemplo, todas las raíces del polinomio de Legendre $P_n(x)$ caen dentro del intervalo $[-1, 1]$.

Una propiedad importante de los polinomios ortogonales es la de generar el espacio vectorial de todos los otros polinomios, ortogonales o no.

Sea $p_n(x)$ un polinomio arbitrario de grado n

$$p_n(x) = \sum_{i=0}^n a_i x^i$$

El polinomio $p_n(x)$ puede representarse como la combinación lineal de cualquiera de las familias de polinomios ortogonales mencionadas arriba:

$$P_n(x) = \sum_{i=0}^n B_i Z_i(x)$$

siendo $Z_i(x)$ el polinomio de grado i de una de las familias de polinomios ortogonales.

Ejemplo:

El polinomio de cuarto grado

$$P_4(x) = \sum_{i=0}^4 \alpha_i x^i \text{ puede expresarse de manera Única en términos}$$

de los polinomios de Legendre, es decir:

$$P_4(x) = \sum_{i=0}^4 B_i P_i(x) \quad 6$$

$$P_4(x) = B_0 + B_1 x + B_2 \left(\frac{3}{2} x^2 - \frac{1}{2}\right) + B_3 \left(\frac{5}{2} x^3 - \frac{3}{2} x\right) + B_4 \left(\frac{35}{8} x^4 - \frac{15}{4} x^2 + \frac{3}{8}\right)$$

Agrupando términos según el grado de la variable x

$$P_4(x) = \left(B_0 - \frac{B_2}{2} + \frac{3}{8} B_4\right) + \left(B_1 - \frac{3}{2} B_2\right)x + \left(\frac{3}{2} B_2 - \frac{15}{4} B_4\right)x^2 + \frac{5}{2} B_3 x^3 + \frac{35}{8} B_4 x^4$$

e igualando con los coeficientes α se deduce que:

$$B_4 = \frac{8}{35} \alpha_4$$

$$B_3 = \frac{2}{5} \alpha_3$$

$$B_2 = \frac{2}{3} \left(\alpha_2 + \frac{6}{7} \alpha_4\right)$$

$$B_1 = \alpha_1 + \frac{3}{5} \alpha_3$$

$$y \quad p_0 = a_0 + \frac{1}{3} a_2 + \frac{1}{5} a_4$$

Es fácil de verificar que el polinomio

$$p_4(x) = x^4 + 3x^3 - 2x^2 + 2x - 1$$

puede ser expresado también como:

$$p_4(x) = -\frac{22}{15} p_0(x) + \frac{19}{5} p_1(x) - \frac{16}{21} p_2(x) + \frac{6}{5} p_3(x) + \frac{6}{35} p_4(x)$$

Cuadraturas Gaussianas.

En general, el método de las cuadraturas gaussianas propone aproximar la integral

$$\int_a^b w(x) F(x) dx$$

por medio de una suma de n términos

$$\sum_{i=1}^n W_i F(T_i),$$

siendo los T_i y W_i , ciertos nodos y ponderadores, respectivamente. La función $w(x)$, en la integral, no es otra que la función peso asociada a alguna familia de polinomios ortogonales. Los nodos T_i son las raíces del polinomio ortogonal de grado n perteneciente a la misma familia anterior y el cual es empleado como herramienta de solución del problema. Cada elemento ponderador W_i es la solución de la integral efectuada sobre un po

linomio de grado l (de la misma familia anterior), en el intervalo $[a, b]$.

En otras palabras, muchas de las técnicas de integración numérica, tales como Simpson, Romberg, etc. (donde $w(x) = 1$) emplean puntos equidistantes x_i , lo cual puede ser conveniente pero no necesario. En los métodos Gaussianos no se seleccionan puntos equidistantes, sino más bien se emplean puntos (nodos) que son las raíces de un polinomio ortogonal de grado n definido sobre el intervalo $[a, b]$ y con función peso, la función $w(x)$.

La razón de los nodos y ponderadores resultará más clara con lo que se explique a continuación.

(i)* Cuadratura Gauss-Legendre.

En el método de la cuadratura de Legendre, la herramienta de solución a la integral

$$\int_a^b f(x) dx, \text{ son los polinomios de Legendre.}$$

El método consiste en aproximar la función $f(x)$ por medio de un polinomio cualquiera de grado n que pase por n puntos x_i y que sea igual a $f(x_i)(V_i)$, más una cierta función error $R_n(x)$, la cual contenga el error en la aproximación

$$f(x) = p_n(x) + R_n(x) \quad (1)$$

La función $p_n(x)$ puede ser expresada en términos de polinomios Lagrangianos (ver Capítulo III) como:

$$P_n(x) = \sum_{i=0}^n L_i(x) f(x_i) \quad (2)$$

donde cada función $L_i(x)$ es el polinomio de grado n igual a:

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \left(\frac{x - x_j}{x_i - x_j} \right), \text{ y la función error, según el método de}$$

Lagrange de interpolación, es

$$R_n(x) = \prod_{i=0}^n (x - x_i) \frac{f^{(n+1)}(\xi)}{(n+1)!}, \quad a < \xi < b \quad (3)$$

Substituyendo las expresiones (2) y (3) en (1) se tiene:

$$f(x) = \sum_{i=0}^n L_i(x) f(x_i) + \left[\prod_{i=0}^n (x - x_i) \right] \frac{f^{(n+1)}(\xi)}{(n+1)!}, \quad a < \xi < b \quad (4)$$

Siendo el intervalo de ortogonalidad de los polinomios de Legendre $[-1, 1]$, es necesario, antes de integrar la función $f(x)$, efectuar un cambio de variable a fin de modificar el intervalo de integración de $[a, b]$ a $[-1, 1]$.

Haciendo

$z = \frac{2x - (a+b)}{b - a}$, la función $f(x)$ se convierte en una nueva función en z ,

$$f(x) = F(z) = f\left(\frac{z(b - a) + (a + b)}{2}\right)$$

quedando la expresión (4) definida entonces como:

$$F(z) = \sum_{i=0}^n L_i(z) F(z_i) + \left[\prod_{i=0}^n (z - z_i) \right] \frac{F^{(n+1)}(\xi)}{(n+1)!}, \quad -1 < \xi < 1 \quad (5)$$

$$\text{donde } L_i(z) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{z - z_j}{z_i - z_j}$$

Si la función $f(x)$ (ó $F(z)$) fuera un polinomio de grado $2n+1$, entonces, el término $\sum_{i=0}^n L_i(z) F(z_i)$ sería un polinomio de, a lo más, grado n , el término $\prod_{i=0}^n (z - z_i)$ sería un polinomio de grado $n+1$, y el término $\frac{F^{(n+1)}(\xi)}{(n+1)!}$ sería un polinomio de grado n .

Considerando éste último término igual a un polinomio de grado n , $q_n(z)$, e integrando entre -1 y 1 , la expresión (5) se convierte en:

$$\int_{-1}^1 F(z) dz = \int_{-1}^1 \sum_{i=0}^n L_i(z) F(z_i) dz + \int_{-1}^1 \left[\prod_{i=0}^n (z - z_i) \right] q_n(z) dz.$$

En conclusión, la integral $\int_{-1}^1 F(z) dz$ puede aproximarse por la expresión

$$\sum_{i=0}^n W_i F(z_i) \quad (6)$$

con un error dado por la expresión

$$\text{Error} = \int_{-1}^1 \prod_{i=0}^n (z - z_i) q_n(z) dz.$$

siendo

$$W_1 = \int_{-1}^1 L_1(z) dz = \int_{-1}^1 \prod_{\substack{j=0 \\ j \neq 1}}^n \left(\frac{z - z_j}{z_1 - z_j} \right) dz. \quad (7)$$

El objetivo fundamental será seleccionar aquellos valores z_j que anulen (o minimicen) el error en la aproximación. La ortogonalidad de los polinomios de Legendre puede emplearse con este propósito.

Ahora bien, expandiendo los polinomios $q_n(z)$ y $\prod_{i=1}^n (z - z_i)$ en términos de polinomios de Legendre se tiene:

$$q_n(z) = \sum_{j=0}^n c_j P_j(z) \quad y \quad (8)$$

$$\prod_{i=0}^n (z - z_i) = \sum_{i=0}^{n+1} b_i P_i(z). \quad (9)$$

El producto $q_n(z) \prod_{i=0}^n (z - z_i)$ resulta entonces igual a:

$$\sum_{i=0}^{n+1} \sum_{j=0}^n b_i c_j P_i(z) P_j(z), \text{ mismo que al ser integrado entre } [-1, 1] \text{ se}$$

reduce a:

$$\text{Error} = \sum_{i=0}^n b_i c_i \int_{-1}^1 [P_i(z)]^2 dz$$

Para que el error sea igual a cero será menester igualar los $n+1$ coeficientes b_i a cero. Por otro lado, en la expresión (9), si todos los b_i ($i = 0, \dots, n$) son iguales a cero, exceptuando el último, se ob

tiene que:

$$\prod_{i=0}^n (z - z_i) = b_{n+1} P_{n+1}(z). \quad (10)$$

En esta última expresión el polinomio de la izquierda tiene como coeficientes del término de mayor orden, la unidad. Por lo tanto, si la expresión (10) se cumple, b_{n+1} deberá ser igual al inverso del coeficiente del término de mayor orden del polinomio $P_{n+1}(z)$. Mas aún, si ambos polinomios son iguales, ellos deberán tener las mismas raíces, o lo que es lo mismo, las raíces de $P_{n+1}(z)$ deberán de ser las mismas que las raíces de $\prod_{i=0}^n (z - z_i)$, que son precisamente, todas las z_i .

Habiendo definido los puntos z_i como las raíces del polinomio de Legendre de grado $n+1$, $P_{n+1}(z)$, los pesos W_i en la aproximación integral (expresión (6)) se definen de manera inmediata a través de la expresión (7).

Todo este desarrollo se ha basado en la suposición inicial de una función $f(x)$ polinómica de grado menor o igual a $2n+1$. Si éste no fuese el caso, se incurriría en un error en la aproximación igual a:

$$\text{Error}_n = \frac{2^{2n+3} [(n+1)!]^4}{(2n+3) [(2n+2)!]^3} F^{(2n+2)}(\xi), \quad -1 < \xi < 1.$$

(ii) Cuadratura Gauss-Laguerre.

El desarrollo del método de la cuadratura Gauss-Laguerre, es similar al anterior. Los nodos z_i son, en este caso, las raíces del polino-

nie de Laguerre de grado $n+1$, $L_{n+1}(z)$, y los pesos asociados a cada nodo están dados por la expresión

$$W_i = \int_0^{\infty} e^{-z} L_i(z) dz = \int_0^{\infty} e^{-z} \prod_{\substack{j=0 \\ j \neq i}}^n \left(\frac{z - z_j}{z_i - z_j} \right) dz.$$

El error en la aproximación integral de funciones no-polinómicas está dado por:

$$E_n = \frac{[(n+1)!]^2}{(2n+2)!} F^{(2n+2)}(\xi), \quad 0 < \xi < \infty.$$

(iii) Cuadratura Gauss-Chebyshev.

Análogamente, los nodos z_i son las raíces del polinomio $T_{n+1}(z)$, los pesos W_i están dados por la expresión

$$W_i = \int_{-1}^1 \frac{1}{\sqrt{1-z^2}} L_i(z) dz,$$

y el error para funciones no polinómicas es igual a:

$$E_n = \frac{2\pi}{2^{2n+2} (2n+2)!} F^{(2n+2)}(\xi), \quad -1 < \xi < 1.$$

(iv) Cuadratura Gauss-Hermite

En este caso, z_i serán las raíces de $H_{n+1}(z)$, los pesos W_i estarán definidos por

$$W_i = \int_{-\infty}^{\infty} e^{-z^2} L_i(z) dz$$

y el error en funciones no-polinómicas será

$$E_n = \frac{(n+1)! \sqrt{\pi}}{2^{n+1} (2n+2)!} F(2n+2)(\xi), \quad -1 < \xi < 1.$$

Conclusiones.

- (I) Cualquier función $f(x)$ que exista en el intervalo de integración puede ser integrada empleando alguna de las cuatro cuadraturas anteriores.
- (II) Si la función peso $w(x)$ no coincide con la función peso de las cuadraturas Gaussianas, entonces es recomendable usar la siguiente transformación:

$$\int_a^b f(x) dx = \int_a^b w(x) \frac{f(x)}{w(x)} dx = \int_a^b w(x) g(x) dx$$

donde $w(x)$ sería la función peso apropiada.

- (III) Si el intervalo de integración no coincide con el intervalo de la cuadratura Gaussiana, entonces la siguiente transformación puede emplearse:

$$z = \frac{2x - (a+b)}{b-a}, \quad \text{intervalo } (-1, 1).$$

La subrutina GAUSSQ listada al final del capítulo calcula, según sea el tipo de cuadratura a emplear, los valores de los nodos T_i y ponderadores W_i para diferentes valores de N , el número de términos en la aproximación.

Ejercicio # 6

Distribución de Velocidades Adimensionales.

Gill y Scher^o derivaron una expresión de la distribución de velocidades de un fluido en una tubería circular modificando la ecuación de Prandtl.

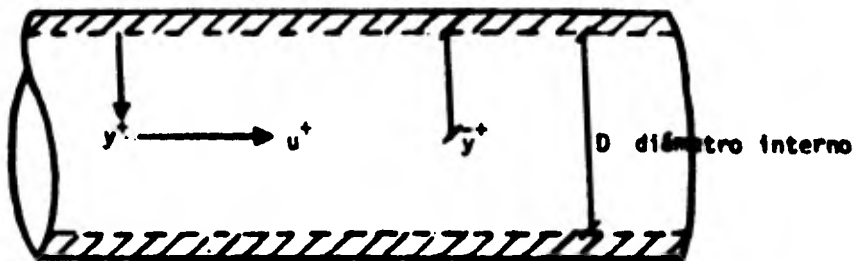
La velocidad adimensional u^+ en la dirección x está dada en función de una distancia adimensional y^+ , medida a partir de la cara interna de la tubería.

$$u^+ = \int_0^{y^+} \frac{-1 + \sqrt{1 + 4cd}}{2c} dy^+ = \int_0^{y^+} \frac{2d}{1 + \sqrt{1 + 4cd}} dy^+$$

donde

$$c = (0.36)^2 (y^+)^2 (1 - e^{-d(y^+/y^+)})^2$$

$$d = 1 - \frac{y^+}{y^+}$$



* W.W. Gill and M.Scher, "A Modification of the Momentum Transport Hypothesis", A.I.Ch.E. Journal, 7, 61-65 (1961).

\bar{y}^+ es la distancia adimensional correspondiente al eje de la tuberfa y está dada por

$$\bar{y}^+ = \frac{N_{Re}}{2} \sqrt{\frac{f}{2}}$$

ϕ es una función empírica de \bar{y}^+

$$\phi = \frac{\bar{y}^+ - 60}{22}$$

N_{Re} - número de Reynolds y

f - factor de fricción de Fanning.

Ambos, N_{Re} y f son parámetros adimensionales que dependen de las propiedades físicas del fluido, la rugosidad de la tuberfa y la velocidad del fluido, principalmente.

N_{Re} está dado por la fórmula

$$N_{Re} = \frac{vD\rho}{\mu}$$

donde

v - velocidad media del fluido	}	en unidades compatibles
D - diámetro interno		
ρ - densidad		
μ - viscosidad		

Para el caso de tuberfas circulares no muy rugosas donde N_{Re} fluctúa entre 2000 y 5000, f puede representarse adecuadamente por la ecuación de

Blasius

$$f = 0.079 N_{Re}^{-0.25}$$

Escriba un programa general que efectúe el cálculo de la integral

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

empleando la cuadratura de Chebyshev y de Legendre con n puntos de expansión. Sustituya apropiadamente $f(x)$ y los intervalos (a, b) en la expresión de la velocidad adimensional u^+ .

Compare los resultados obtenidos de la integral por las dos cuadraturas para valores del número de Reynolds de

$$N_{Re} = 5000, 10000, 25000 \text{ y } 50000$$

y para distancias dimensionales de

$$y^+ = 1, 2, 5, 10, 20, 50, 100, 200, 500 \text{ y } 1000$$

empleando

2, 4, 6, 10 y 15 puntos de expansión.

Grafique en papel semilogarítmico los resultados de $\log y^+$ vs. u^+ obtenidos cuando el número de partes en la expansión es 15. Comente los resultados.

Ejercicio # 7

Considere la siguiente integral:

$$I = \int_{0.01}^{1.0} \frac{dx}{e^x - 1}$$

a).- Evalúese analíticamente.

b).- Aproxime I empleando la cura de N paneles igualmente espaciados. Dada la singular en el cerca del intervalo de integración $.01$, es prouentades con la exactitud del método. Corra un pr con N_{10} .

c).- Reescriba I como:

$$I = \int_{0.01}^{1.0} \left(\frac{1}{e^x - 1} - \frac{1}{x} \right) dx + \frac{dx}{x}$$

Evalúe la segunda integral anamente usando la cuadratura de Simpson, entonces sumos val tener I . Use, otra vez, $N = 10, 20$ y 40 . ¿Obisultads que en el inciso (b)? ¿Porqué?

d).- Repita los incisos (b) y ando l)QUANCB en lugar de la regla de Simpson. Use utranca, ambos, el error absoluto y el error relativo, la el n)iones evaluadas requeridas. ¿Es QUANCB más etue la pson?


```

C
C
C   INITIALIZE RUNNING BARS TO ZERO.
C
C   FLAG = 0.000
C   RESULT = 0.000
C   CC(1) = 0.000
C   ENLIST = 0.000
C   AREA = 0.000
C   NCFUN = 0
C   IF (A .EQ. 0) RETURN
C
C
C   *** STAGE 2 ***  INITIALIZATION FOR FIRST INTERVAL
C
C   LEV = 0
C   NIM = 1
C   X0 = A
C   X(16) = 0
C   CFREV = 0.000
C   F0 = FUM(X0)
C   STONE = (B - A) / 16.000
C   X(1) = (X0 + X(16)) / 2.000
C   X(4) = (X0 + X(13)) / 2.000
C   X(8) = (X0 + X(10)) / 2.000
C   X(12) = (X0 + X(4)) / 2.000
C   X(6) = (X(4) + X(8)) / 2.000
C   X(10) = (X(8) + X(12)) / 2.000
C   X(14) = (X(12) + X(16)) / 2.000
C   DO 25 J = 2, 16, 2
C     F(J) = FUM(X(J))
C 25 CONTINUE
C   NCFUN = 0
C
C
C   *** STAGE 3 ***  CENTRAL CALCULATION
C   SECURES QPFR,X0,X2,X4,...,X16,F0,F2,F4,...,F16.
C   CALCULATES X1,X3,...,X15, F1,F3,...,F15,QLEFT,QRIGHT,QMOM,QDIFF,AREA.
C
C 30 X(1) = (X0 + X(16)) / 2.000
C   F(1) = FUM(X(1))
C   DO 35 J = 3, 15, 2
C     X(J) = (X(J-1) + X(J+1)) / 2.000
C     F(J) = FUM(X(J))
C 35 CONTINUE
C   NCFUN = NCFUN + 0
C   STEP = (X(16) - X0) / 16.000
C   QLEFT = (.5*(F0 + F(1)) + M1*(F(1)+F(7)) + M2*(F(2)+F(6))
C 1 + M3*(F(3)+F(5)) + M4*(F(4)) + STEP
C   QRIGHT(LEV)=1*(M0*(F(8)+F(16))+M1*(F(9)+F(15))+M2*(F(10)+F(14))
C 1 + M3*(F(11)+F(13)) + M4*(F(12))) + STEP
C   QMOM = QLEFT + QRIGHT(LEV+1)
C   QDIFF = QMOM - CFREV
C   AREA = AREA + QDIFF
C
C
C   *** STAGE 4 ***  INTERVAL CONVERGENCE TEST
C
C   ESTERR = QABS(QDIFF) / 1003.000
C   TOLERR = QMAX(ABSERR,RELERR=QABS(AREA)) * (STEP/STONE)
C   IF (LEV .LT. LEVMAX) GO TO 50
C   IF (M0FUN .GT. LEVMAX) GO TO 60
C   IF (M0FUN .GT. NCFUN) GO TO 60
C   IF (LEVERR .LT. TOLERR) GO TO 70

```

```

00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110

```

```

C
C
C   *** STAGE 5 *** NO CONVERGENCE
C   LOCATE NEXT INTERVAL.
C
C   50 NIN = 2*NIN
C   LEV = LEV+1
C
C   STORE RIGHT HAND ELEMENTS FOR FUTURE USE.
C
C   DO 52 I = 1, 8
C     FSAVE(I,LEV) = F(I,0)
C     XSAVE(I,LEV) = X(I,0)
C   52 CONTINUE
C
C   ASSEMBLE LEFT HAND ELEMENTS FOR INTERMEDIATE USE
C
C   GPREV = GLEFT
C   DO 55 I = 1, 8
C     J = -I
C     F(I+J,0) = F(J,0)
C     X(I+J,0) = X(J,0)
C   55 CONTINUE
C   GO TO 30
C
C   *** STAGE 6 *** TROUBLE SECTION
C   NUMBER OF FUNCTION VALUES IS ABOUT TO EXCEED LIMIT.
C
C   60 NOPIN = 2*NOPIN
C   LEVMAX = LEVOUT
C   FLAG = FLAG * (B - X0) / (B - A)
C   GO TO 70
C
C   CURRENT LEVEL IS LEVMAX.
C
C   62 FLAG = FLAG * 1.000
C
C   *** STAGE 7 *** INTERVAL CONVERGED
C   ADD CONTRIBUTIONS INTO RUNNING SUMS.
C
C   70 RESULT = RESULT * QNEW
C   ERREST = ERREST * ESTERN
C   COR11 = COR11 * QDIFF / 1023.000
C
C   LOCATE NEXT INTERVAL.
C
C   72 IF (NIN .EQ. 2*(NIN/2)) GO TO 75
C   NIN = NIN/2
C   LEV = LEV-1
C   GO TO 72
C   75 NIN = NIN * 2
C   IF (LEV .LE. 0) GO TO 60
C
C   ASSEMBLE ELEMENTS REQUIRED FOR THE NEXT INTERVAL.
C
C   GPREV = ORIENT(LEV)
C   X0 = X(16)
C   F0 = F(16)
C   DO 78 I = 1, 8
C     F(I,0) = FSAVE(I,LEV)
C     X(I,0) = XSAVE(I,LEV)
C   78 CONTINUE
C   GO TO 30
C
C   *** STAGE 8 *** FINALIZE AND RETURN
C
C   80 RESULT = RESULT * COR11
C
C   MAKE SURE ERREST NOT LESS THAN ROUND-OFF LEVEL.
C
C   IF (ERREST .EQ. 0.000) RETURN
C   82 TEMP = DABS(RESULT) * ERREST
C   IF (TEMP .GE. DABS(RESULT)) RETURN
C   ERREST = 2.000*ERREST
C   GO TO 62
C   END

```

```

00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660
00001670
00001680
00001690
00001700
00001710
00001720
00001730
00001740
00001750
00001760
00001770
00001780
00001790
00001800
00001810
00001820
00001830
00001840
00001850
00001860

```

***** START OF GAUSSG *****
LAST UPDATED: FEBRUARY, 1978
SUBROUTINE GAUSSG(KIND, N, ALPHA, BETA, EPTS, ENPTS, S, T, M)

THIS SET OF ROUTINES COMPUTES THE NODES T(J) AND WEIGHTS
W(J) FOR GAUSSIAN-TYPE QUADRATURE RULES WITH PRE-ASSIGNED
NODES. THESE ARE USED WHEN ONE WISHES TO APPROXIMATE
INTEGRAL (FROM A TO B) F(X) W(X) DX

BY
SUM W F(T)
J=1 J J

(NOTE W(X) AND W(J) HAVE NO CONNECTION WITH EACH OTHER.)
HERE W(X) IS ONE OF SIX POSSIBLE NON-NEGATIVE WEIGHT
FUNCTIONS (LISTED BELOW), AND F(X) IS THE
FUNCTION TO BE INTEGRATED. GAUSSIAN QUADRATURE IS PARTICULARLY
USEFUL ON INFINITE INTERVALS (WITH APPROPRIATE WEIGHT
FUNCTIONS), SINCE THEN OTHER TECHNIQUES OFTEN FAIL.

ASSOCIATED WITH EACH WEIGHT FUNCTION W(X) IS A SET OF
ORTHOGONAL POLYNOMIALS. THE NODES T(J) ARE JUST THE ZEROS
OF THE PROPER N-TH DEGREE POLYNOMIAL.

INPUT PARAMETERS (ALL REAL NUMBERS ARE IN DOUBLE PRECISION)

KIND AN INTEGER BETWEEN 1 AND 6 GIVING THE TYPE OF
QUADRATURE RULE:

KIND = 1: LEGENDRE QUADRATURE, W(X) = 1 ON (-1, 1)

KIND = 2: CHEBYSHEV QUADRATURE OF THE FIRST KIND
W(X) = 1/SQRT(1 - X^2) ON (-1, +1)

KIND = 3: CHEBYSHEV QUADRATURE OF THE SECOND KIND
W(X) = SQRT(1 - X^2) ON (-1, 1)

KIND = 4: HERMITE QUADRATURE, W(X) = EXP(-X^2) ON
(-INFINITY, +INFINITY)

KIND = 5: JACOBI QUADRATURE, W(X) = (1-X)^ALPHA * (1+X)^BETA
ON (-1, 1), ALPHA, BETA .GT. -1.
NOTE: KIND=2 AND 3 ARE A SPECIAL CASE OF THIS.

KIND = 6: GENERALIZED LAGUERRE QUADRATURE, W(X) = EXP(-X)*
X^ALPHA ON (0, +INFINITY), ALPHA .GT. -1

N THE NUMBER OF POINTS USED FOR THE QUADRATURE RULE
ALPHA REAL PARAMETER USED ONLY FOR GAUSS-JACOBI AND GAUSS-
LAGUERRE QUADRATURE (OTHERWISE USE 0.00).

BETA REAL PARAMETER USED ONLY FOR GAUSS-JACOBI QUADRATURE--
(OTHERWISE USE 0.00)

EPTS (INTEGER) NORMALLY 0, UNLESS THE LEFT OR RIGHT END-
POINT (OR BOTH) OF THE INTERVAL IS REQUIRED TO BE A
NODE (THIS IS CALLED GAUSS-RADAB OR GAUSS-LOBATTO
QUADRATURE). THEN EPTS IS THE NUMBER OF FIXED
ENDPOINTS (1 OR 2).

ENPTS REAL ARRAY OF LENGTH 2, CONTAINS THE VALUES OF
ANY FIXED ENDPNTS, IF EPTS = 1 OR 2.

S REAL SCRATCH ARRAY OF LENGTH N

OUTPUT PARAMETERS (BOTH DOUBLE PRECISION ARRAYS OF LENGTH N)

T WILL CONTAIN THE DESIRED NODES.

```

61.      C      M      WILL CONTAIN THE DESIRED WEIGHTS W(I,J).
62.      C
63.      C      SUBROUTINES REQUIRED
64.      C
65.      C      SOLVE, CLASS, AND INTGLS ARE PROVIDED. UNDERFLOW MAY SOMETIMES
66.      C      OCCUR, BUT IT IS HARMLESS IF THE UNDERFLOW INTERRUPTS ARE
67.      C      TURNED OFF. TO DO THIS, THE FIRST CALL OF THE MAIN PROGRAM
68.      C      SHOULD BE
69.      C      - CALL TRAPS (0, 0, 19999, 0, 0) IN MATRIX
70.      C
71.      C      OR
72.      C      CALL INIT
73.      C      IN FORTRAN 6 OR 8.
74.      C
75.      C      ACCURACY
76.      C
77.      C      THE ROUTINE WAS TESTED UP TO N = 512 FOR LEGENDRE QUADRATURE,
78.      C      UP TO N = 136 FOR HERMITE, UP TO N = 60 FOR LAGUERRE, AND UP
79.      C      TO N = 10 OR 20 IN OTHER CASES. IN ALL BUT TWO INSTANCES,
80.      C      COMPARISON WITH TABLES IN REF. 3 SHOWED 12 OR MORE SIGNIFICANT
81.      C      DIGITS OF ACCURACY. THE TWO EXCEPTIONS WERE THE WEIGHTS FOR
82.      C      HERMITE AND LAGUERRE QUADRATURE, WHERE UNDERFLOW CAUSED SOME
83.      C      VERY SMALL WEIGHTS TO BE SET TO ZERO. THIS IS, OF COURSE,
84.      C      COMPLETELY HARMLESS.
85.      C
86.      C      METHOD
87.      C
88.      C      THE COEFFICIENTS OF THE THREE-TERM RECURRENCE RELATION
89.      C      FOR THE CORRESPONDING SET OF ORTHOGONAL POLYNOMIALS ARE
90.      C      USED TO FORM A SYMMETRIC TRIAGONAL MATRIX, WHOSE
91.      C      EIGENVALUES (DETERMINED BY THE IMPLICIT QL-METHOD WITH
92.      C      SHIFTS) ARE JUST THE DESIRED NODES. THE FIRST COMPONENTS OF
93.      C      THE ORTHONORMALIZED EIGENVECTORS, WHEN PROPERLY SCALED,
94.      C      YIELD THE WEIGHTS. THIS TECHNIQUE IS MUCH FASTER THAN USING A
95.      C      ROOT-FINDER TO LOCATE THE ZEROS OF THE ORTHOGONAL POLYNOMIAL.
96.      C      FOR FURTHER DETAILS, SEE REF. 1. REF. 2 CONTAINS DETAILS OF
97.      C      GAUSS-RABAU AND GAUSS-LOBATTO QUADRATURE ONLY.
98.      C
99.      C      REFERENCES
100.      C
101.      C      1. GOLUB, G. H., AND WELSH, J. H., "CALCULATION OF GAUSSIAN
102.      C      QUADRATURE RULES," MATHEMATICS OF COMPUTATION 23 (APRIL,
103.      C      1969), PP. 221-238.
104.      C
105.      C      2. GOLUB, G. H., "SOME MODIFIED MATRIX EIGENVALUE PROBLEMS,"
106.      C      SIAM REVIEW 15 (APRIL, 1973), PP. 316-334 (SECTION 7).
107.      C
108.      C      3. STROUD AND SECREST, GAUSSIAN QUADRATURE FORMULAS. PRENTICE-
109.      C      HALL, ENGLEWOOD CLIFFS, N.J., 1966.
110.      C
111.      C      .....
112.      C
113.      C      DOUBLE PRECISION B(N), T(N), W(N), EMPDS(2), NUZERO, TI,
114.      C      X GAN, SOLVE, BSQRT, ALPHA, BETA
115.      C
116.      C      CALL CLASS (KIND, N, ALPHA, BETA, D, T, NUZERO)
117.      C
118.      C      THE MATRIX OF COEFFICIENTS IS ASSUMED TO BE SYMMETRIC.
119.      C      THE ARRAY T CONTAINS THE DIAGONAL ELEMENTS, THE ARRAY
120.      C      B THE OFF-DIAGONAL ELEMENTS.
121.      C      MAKE APPROPRIATE CHANGES IN THE LOWER RIGHT 2 BY 2
122.      C      SUBMATRIX.
123.      C
124.      C      IF (KPTS.EQ.8) GO TO 100

```

```

121.      IF (KPTS.EQ.2) GO TO 50
122.      C
123.      C      IF KPTS=1, ONLY T(N) MUST BE CHANGED
124.      C
125.      T(N) = SOLVE(ENDPTS(1), N, T, B)*B(N-1)**2 + ENDPTS(1)
126.      GO TO 100
127.      C
128.      C      IF KPTS=2, T(N) AND B(N-1) MUST BE RECOMPUTED
129.      C
130.      SO GAM = SOLVE(ENDPTS(1), N, T, B)
131.      T1 = ((ENDPTS(1) - ENDPTS(2))/(SOLVE(ENDPTS(2), N, T, B) - GAM))
132.      O(N-1) = SQRT(T1)
133.      T(N) = ENDPTS(1) + GAM*O1
134.      C
135.      C      NOTE THAT THE INDICES OF THE ELEMENTS OF B RUN FROM 1 TO N-1
136.      C      AND THUS THE VALUE OF B(N) IS ARBITRARY.
137.      C      NOW COMPUTE THE EIGENVALUES OF THE SYMMETRIC TRIDIAGONAL
138.      C      MATRIX, WHICH HAS BEEN MODIFIED AS NECESSARY.
139.      C      THE METHOD USED IS A QL-TYPE METHOD WITH ORIGIN SHIFTING
140.      C
141.      DO 105 I = 1, NDD
142.      DO 105 I = 2, N
143.      M(I) = 0.000
144.      C
145.      CALL INTQL2 (N, T, B, M, IERR)
146.      DO 110 I = 1, N
147.      M(I) = MZERO + M(I) * M(I)
148.      C
149.      RETURN
150.      END
151.      C
152.      C
153.      C
154.      DOUBLE PRECISION FUNCTION SOLVE(SHIFT, N, B, D)
155.      C
156.      THIS PROCEDURE PERFORMS ELIMINATION TO SOLVE FOR THE
157.      N-TH COMPONENT OF THE SOLUTION DELTA TO THE EQUATION
158.      C
159.      (JN - SHIFT*IGEN(TTT)) * DELTA = EN,
160.      C
161.      WHERE EN IS THE VECTOR OF ALL ZEROS EXCEPT FOR 1 IN
162.      THE N-TH POSITION.
163.      C
164.      THE MATRIX JN IS SYMMETRIC TRIDIAGONAL, WITH DIAGONAL
165.      ELEMENTS A(I), OFF-DIAGONAL ELEMENTS B(I). THIS EQUATION
166.      MUST BE SOLVED TO OBTAIN THE APPROPRIATE CHANGES IN THE LOWER
167.      2 BY 2 SUBMATRIX OF COEFFICIENTS FOR ORTHOGONAL POLYNOMIALS.
168.      C
169.      C
170.      DOUBLE PRECISION SHIFT, B(N), O(N), ALPHA
171.      C
172.      ALPHA = B(1) - SHIFT
173.      NN1 = N - 1
174.      DO 10 I = 2, NN1
175.      10 ALPHA = A(I) - SHIFT - B(I-1)**2/ALPHA
176.      SOLVE = 1.000/ALPHA
177.      RETURN
178.      END
179.      C
180.      C

```

181. C
182. C
183. C
184. C
185. C
186. C
187. C
188. C
189. C
190. C
191. C
192. C
193. C
194. C
195. C
196. C
197. C
198. C
199. C
200. C
201. C
202. C
203. C
204. C
205. C
206. C
207. C
208. C
209. C
210. C
211. C
212. C
213. C
214. C
215. C
216. C
217. C
218. C
219. C
220. C
221. C
222. C
223. C
224. C
225. C
226. C
227. C
228. C
229. C
230. C
231. C
232. C
233. C
234. C
235. C
236. C
237. C
238. C
239. C
240. C

```

SUBROUTINE CLASS(KIND, N, ALPHA, BETA, B, A, MUZERO)

  THIS PROCEDURE SUPPLIES THE COEFFICIENTS A(J), B(J) OF THE
  RECURRENCE RELATION

      O P (X) = (X - A ) P (X) - B P (X)
      J J          J J-1      J-1 J-2

  FOR THE VARIOUS CLASSICAL (NORMALIZED) ORTHOGONAL POLYNOMIALS,
  AND THE ZERO-TO MOMENT

      MUZERO = INTEGRAL W(X) DX

  OF THE GIVEN POLYNOMIAL'S WEIGHT FUNCTION W(X). SINCE THE
  POLYNOMIALS ARE ORTHONORMALIZED, THE TRIDIAGONAL MATRIX IS
  GUARANTEED TO BE SYMMETRIC.

  THE INPUT PARAMETER ALPHA IS USED ONLY FOR LAGUERRE AND
  JACOBI POLYNOMIALS, AND THE PARAMETER BETA IS USED ONLY FOR
  JACOBI POLYNOMIALS. THE LAGUERRE AND JACOBI POLYNOMIALS
  REQUIRE THE GAMMA FUNCTION.

  .....

  DOUBLE PRECISION A(N), B(N), MUZERO, ALPHA, BETA
  DOUBLE PRECISION A01, A252, G0ANNA, PI, SSQRT, AS
  DATA PI / 3.14159265358979300/

  NMI = N - 1
  GO TO (10, 20, 30, 40, 50, 60), KIND

      KIND = 1: LEGENDRE POLYNOMIALS P(X)
      ON (-1, +1), W(X) = 1.

10 MUZERO = 2.000
   DO 11 I = 1, NMI
      A(I) = 0.000
      B(I) = I
11 0(I) = ABI/DSQRT(4*ABI*ABI - 1.000)
   A(N) = B.000
   RETURN

      KIND = 2: CHEBYSHEV POLYNOMIALS OF THE FIRST KIND T(X)
      ON (-1, +1), W(X) = 1 / SQRT(1 - X*X)

20 MUZERO = PI
   DO 21 I = 1, NMI
      A(I) = 0.000
21 0(I) = 0.500
   0(1) = DSQRT(0.500)
   A(N) = 0.000
   RETURN

      KIND = 3: CHEBYSHEV POLYNOMIALS OF THE SECOND KIND U(X)
      ON (-1, +1), W(X) = SQRT(1 - X*X)

30 MUZERO = PI/2.000
   DO 31 I = 1, NMI
      A(I) = 0.000

```

```

31 B(I) = 0.000
   A(N) = 0.000
   RETURN

```

```

C
C      KIND = 4:  HERMITE POLYNOMIALS H(X) ON (-INFINITY,
C      +INFINITY), W(X) = EXP(-X**2)
C

```

```

40 MUZERO = DSQRT(PI)
   DO 61 I = 1, NM1
      A(I) = 0.000
   61 B(I) = DSQRT(1/2.000)
   A(N) = 0.000
   RETURN

```

```

C
C      KIND = 5:  JACOBI POLYNOMIALS P(ALPHA, BETA)(X) ON
C      (-1, +1), W(X) = (1-X)**ALPHA * (1+X)**BETA, ALPHA AND
C      BETA GREATER THAN -1
C

```

```

50 AB = ALPHA + BETA
   A01 = 2.000 + AB
   MUZERO = 2.000 ** (A0 + 1.000) * DBAMMA(ALPHA + 1.000) * DBAMMA(
X BETA + 1.000) / DBAMMA(AB)
   A(1) = (BETA - ALPHA)/A01
   B(1) = DSQRT(4.000*(1.000 + ALPHA)*(1.000 + BETA)/((A01 + 1.000)**
1 AB**AB))
   A2B2 = BETA*BETA - ALPHA*ALPHA
   DO 51 I = 2, NM1
      AB1 = 2.000*I + AB
      A(1) = A2B2/((AB1 - 2.000)*A01)
   51 B(I) = DSQRT(4.000*I*(I + ALPHA)*(I + BETA)*(I + AB)/
1 ((AB1**AB1 - 1)**AB1**AB1))
   AB1 = 2.000*I + AB
   A(N) = A2B2/((AB1 - 2.000)*AB1)
   RETURN

```

```

C
C      KIND = 6:  LAGUERRE POLYNOMIALS L(ALPHA)(X) ON
C      (0, +INFINITY), W(X) = EXP(-X) * X**ALPHA, ALPHA GREATER
C      THAN -1.
C

```

```

60 MUZERO = DBAMMA(ALPHA + 1.000)
   DO 61 I = 1, NM1
      A(I) = 2.000**I - 1.000 + ALPHA
   61 B(I) = DSQRT(I*(I + ALPHA))
   A(N) = 2.000**N - 1 + ALPHA
   RETURN
END

```

```

C
C      SUBROUTINE INTQL2(N, D, E, Z, IERR)

```

```

C
C      THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE INTQL2,
C      NUM. MATN. 12, 377-383(1966) BY MARTIN AND MILKINSON,
C      AS MODIFIED IN NUM. MATN. 15, 450(1976) BY DUBRILLE,
C      HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBR., 241-248(1971).
C      THIS IS A MODIFIED VERSION OF THE 'EISPACK' ROUTINE INTQL2.

```

```

C
C      THIS SUBROUTINE FINDS THE EIGENVALUES AND FIRST COMPONENTS OF THE
C      EIGENVECTORS OF A SYMMETRIC TRIIDIAGONAL MATRIX BY THE IMPLICIT QL
C      METHOD.
C

```

241.
242.
243.
244.
245.
246.
247.
248.
249.
250.
251.
252.
253.
254.
255.
256.
257.
258.
259.
260.
261.
262.
263.
264.
265.
266.
267.
268.
269.
270.
271.
272.
273.
274.
275.
276.
277.
278.
279.
280.
281.
282.
283.
284.
285.
286.
287.
288.
289.
290.
291.
292.
293.
294.
295.
296.
297.
298.
299.
300.


```

301. C
302. C
303. C
304. C
305. C
306. C
307. C
308. C
309. C
310. C
311. C
312. C
313. C
314. C
315. C
316. C
317. C
318. C
319. C
320. C
321. C
322. C
323. C
324. C
325. C
326. C
327. C
328. C
329. C
330. C
331. C
332. C
333. C
334. C
335. C
336. C
337. C
338. C
339. C
340. C
341. C
342. C
343. C
344. C
345. C
346. C
347. C
348. C
349. C
350. C
351. C
352. C
353. C
354. C
355. C
356. C
357. C
358. C
359. C
360. C

```

ON INPUT:

N IS THE ORDER OF THE MATRIX;

D CONTAINS THE DIAGONAL ELEMENTS OF THE INPUT MATRIX;

E CONTAINS THE SUBDIAGONAL ELEMENTS OF THE INPUT MATRIX
IN ITS FIRST N-1 POSITIONS. E(N) IS ARBITRARY;

Z CONTAINS THE FIRST ROW OF THE IDENTITY MATRIX.

ON OUTPUT:

D CONTAINS THE EIGENVALUES IN ASCENDING ORDER. IF AN
ERROR EXIT IS MADE, THE EIGENVALUES ARE CORRECT BUT
UNORDERED FOR INDICES 1, 2, ..., IERR-1;

E HAS BEEN DESTROYED;

Z CONTAINS THE FIRST COMPONENTS OF THE ORTHONORMAL EIGENVECTORS
OF THE SYMMETRIC TRIDIAGONAL MATRIX. IF AN ERROR EXIT IS
MADE, Z CONTAINS THE EIGENVECTORS ASSOCIATED WITH THE STORED
EIGENVALUES;

IERR IS SET TO
ZERO FOR NORMAL RETURN,
J IF THE J-TH EIGENVALUE HAS NOT BEEN
DETERMINED AFTER 30 ITERATIONS.

INTEGER I, J, K, L, M, N, II, NML, IERR
DOUBLE PRECISION D(N), E(N), Z(N), B, C, F, G, P, R, S, MACHEP
DOUBLE PRECISION DSQRT, DABS, DSIGN

..... MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING
THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.
MACHEP = 16.68584(-13) FOR LONG FORM ARITHMETIC
ON S360

DATA MACHEP/234168000000000000/

IERR = 0
IF (M.EQ. 1) GO TO 1001

E(N) = 0.000
GO TO 4 L = 1, M
J = 0

..... LOOK FOR SMALL SUB-DIAGONAL ELEMENT

105 DO 110 M = L, M
IF (M.EQ. N) GO TO 120
IF (DABS(E(M)) .LE. MACHEP * (DABS(D(M)) + DABS(D(M+1))))
X GO TO 120

110 CONTINUE

P = E(L)
IF (M.EQ. L) GO TO 240
IF (J.EQ. 30) GO TO 1000
J = J + 1

..... FORM SHIFT

B = (D(L+1) - P) / (2.000 * E(L))

```

361.      R = DSQRT(S**2+1.000)
362.      G = D(N) - P * Z(L) / (G + DSIGN(R, G))
363.      S = 1.000
364.      C = 1.000
365.      P = 0.000
366.      NML = N - L
367.  C
368.  C      :::::::::: FOR I=N-1 STEP -1 UNTIL L DO -- ::::::::::
369.      DO 200 II = 1, NML
370.          I = N - II
371.          F = G * Z(I)
372.          G = C * Z(I)
373.          IF (DABS(F) .LT. DABS(G)) GO TO 150
374.          G = G / F
375.          R = DSQRT(C**2+1.000)
376.          Z(I+1) = F * R
377.          S = 1.000 / R
378.          C = C * S
379.          GO TO 160
380.      150      G = F / G
381.          R = DSQRT(S**2+1.000)
382.          Z(I+1) = G * R
383.          C = 1.000 / R
384.          S = S * C
385.      160      G = D(I+1) - P
386.          R = (D(I) - G) * S + 2.000 * C * G
387.          P = S * R
388.          D(I+1) = G + P
389.          G = C * R - S
390.  C      :::::::::: FORM FIRST COMPONENT OF VECTOR ::::::::::
391.          F = Z(I+1)
392.          Z(I+1) = S * Z(I) + C * F
393.      200      Z(I) = C * Z(I) - S * F
394.  C
395.          D(L) = D(L) - P
396.          Z(L) = G
397.          D(N) = 0.000
398.          GO TO 185
399.      240 CONTINUE
400.  C
401.  C      :::::::::: ORDER EIGENVALUES AND EIGENVECTORS ::::::::::
402.      DO 300 II = 2, N
403.          I = II - 1
404.          K = I
405.          P = D(I)
406.  C
407.          DO 260 J = II, N
408.              IF (D(J) .GE. P) GO TO 260
409.              K = J
410.              P = D(J)
411.      260      CONTINUE
412.  C
413.          IF (K .EQ. I) GO TO 300
414.          D(K) = D(I)
415.          D(I) = P
416.          P = Z(I)
417.          Z(I) = Z(K)
418.          Z(K) = P
419.      300 CONTINUE
420.  C

```

421. GO TO 1001
422. C SET ERROR -- NO CONVERGENCE TO AN
423. C EIGENVALUE AFTER 30 ITERATIONS
424. 1000 IERR = L
425. 1001 RETURN
426. C LAST CARD OF INTQL2
427. END

CAPITULO V

SOLUCION NUMERICA DE ECUACIONES DIFERENCIALES ORDINARIAS

Ecuaciones Diferenciales Ordinarias.

En una ecuación diferencial de primer orden $y' = f(y,t)$, el objetivo es encontrar aquella función $y(t)$ que satisfaga la ecuación diferencial.

La ecuación diferencial

$$\frac{dy}{dt} = f(y,t) = y + t^2$$

tiene como solución una familia de curvas $y = y(t)$. Si la función $f(y,t)$ fuera igual a y , la ecuación diferencial tendría como solución la función $y(t) = Ce^t$.

Al elegir un valor inicial $y_0 = y(0)$ se seleccionaría una de las curvas que integran la familia de curvas.

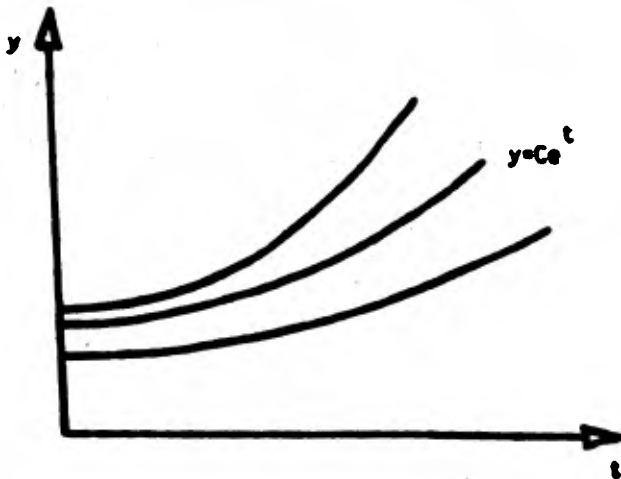


Figura 15. Familia de curvas solución a la EDO $y' = y$.

En el caso de ecuaciones diferenciales ordinarias (EDO) con varias variables dependientes, el problema consiste en resolver el sistema de EDO de forma

$$y' = f(y, z, t)$$

$$z' = g(y, z, t).$$

La solución analítica de este sistema involucra dos constantes, requiriéndose para ello dos piezas adicionales de información. Si los valores de y y z fuesen especificados para un cierto valor t_0 de la variable independiente t , entonces la solución sería única. El problema de la determinación de los valores de y y z para valores de $t > t_0$ es conocido con el nombre de "problema del valor inicial".

Finalmente, una EDO de orden n , donde la derivada de más alto orden es despejable, puede ser escrita también como un sistema de n ecuaciones diferenciales de primer orden introduciendo $n-1$ nuevas variables.

La ecuación diferencial ordinaria

$$u'' = g(u, u', t)$$

puede escribirse también como:

$$z' = g(u, z, t)$$

$$u' = z$$

Dado que en la práctica no todas las ecuaciones diferenciales pueden ser resueltas analíticamente, es deseable entonces definir otros mé-

todos de solución tales como los métodos numéricos. Además de la información requerida en la solución analítica de ecuaciones diferenciales, la solución numérica requiere de dos piezas adicionales de información:

- (i) Una expresión que indique el error a tolerar, y
- (ii) Una expresión que indique cuanto se desea pagar por alcanzar tal solución.

Las aproximaciones numéricas de interés en este capítulo involucran los métodos de paso a paso (o de diferencias o de variables discretas) los cuales consisten en la generación de una secuencia de puntos, t_0, t_1, \dots , con intervalo variable $h_n = t_{n+1} - t_n$. En cada punto t_n , la solución $y(t_n)$ es aproximada por un número y_n el cual es calculado a partir de ciertos valores anteriormente estimados.

En general, si K valores estimados son empleados en el cálculo del valor y_{n+1} , el método se denomina, método de K pasos. Cuando K es igual a 1, el método recibe el nombre de método de paso simple, y cuando $K > 1$ el método se nombra método de múltiplos pasos.

Un ejemplo del método de paso-simple es el método de Euler, donde el valor y_{n+1} es calculado a través de una extrapolación lineal del valor anterior y_n . Tal es el caso de la ecuación diferencial ordinaria

$$y' = f(y, t) \text{ con condición inicial } y(t_0) = y_0,$$

donde la pendiente de la solución $y(t)$ es calculada a partir de la condición inicial, esto es:

$$y'_0 = f(y_0, t_0)$$

y donde el valor estimado y_1 , correspondiente a $y(t_1)$ es calculado empleando los dos primeros términos de la serie de Taylor

$$y(t_1) = y_1 = y_0 + h_0 f(t_0, y_0).$$

Equivalentemente, para $t_2 = t_1 + h_1$, $y(t_2)$ será aproximada mediante la expresión

$$y_2 = y_1 + h_1 f(t_1, y_1)$$

y en general

$$y(t_{n+1}) = y_{n+1} = y_n + h_n f(t_n, y_n)$$

El método de Euler, sin embargo, puede crear errores considerables en la solución de algunas EDO. En el caso de la ecuación $y' = y$, donde e^t es la solución, a medida que t aumenta, el error será multiplicado por un factor e^t produciendo inestabilidad en la ED (Figura 15), en cambio, para $y' = -y$, donde e^{-t} es la solución, los errores se verán disminuidos a medida que t aumente de valor, produciéndose una solución estable (Figura 16).

A continuación se presentan los métodos numéricos de aproximación más comúnmente empleados para la solución de ecuaciones diferenciales ordinarias. Únicamente se presentarán los algoritmos de solución sin hacer alusión alguna al problema de inestabilidad.

(I) Método de Taylor

Sea $y(t)$ la solución de cierta EDO, la cual al expresarse en tér-

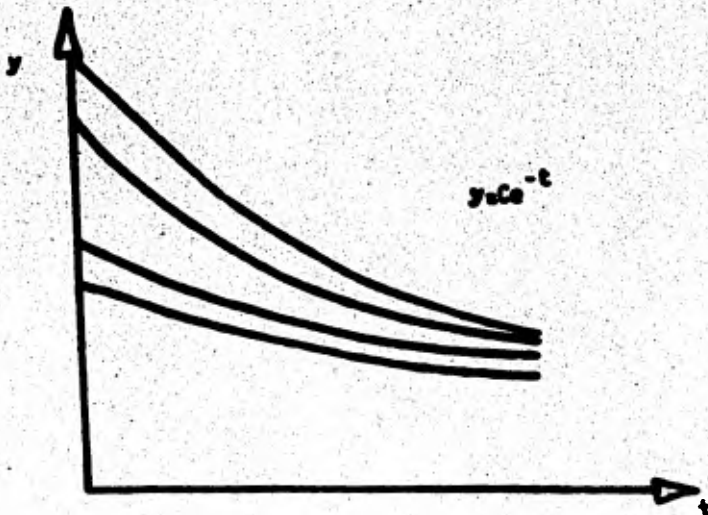


Figura 16. Familia de curvas solución a la EDO $y' = -y$.

minos de series de Taylor queda como:

$$y(t+h) = y(t) + h_n y'(t) + \frac{h_n^2}{2!} y''(t) + \dots$$

El método de Euler puede ser visto como un caso particular del método de Taylor. Una aproximación de orden p tendría la forma siguiente:

$$y_{n+1} = y_n + h_n y'_n + \frac{h_n^2}{2!} y''_n + \dots + \frac{h_n^p}{p!} y_n^{(p)}.$$

donde las derivadas y', y'', \dots serían evaluadas en términos de las derivadas parciales de la función f en la ecuación diferencial

$$y' = f(y, t).$$

Usando la regla de la cadena repetidas veces, pueden demostrarse las igualdades siguientes:

$$\begin{aligned}
 y'' &= f_y y' + f_t \\
 &= f_y \cdot f + f_t \\
 y''' &= f_{yy} y' \cdot f + f_y f_y y' + f_y f_t + f_{tt} + 2f_{ty} \cdot y' \\
 &= f_{yy} \cdot f^2 + f_y^2 \cdot f + f_y f_t + 2f_{ty} \cdot f + f_{tt} \\
 &\text{etc.}
 \end{aligned}$$

(ii) Método Runge-Kutta

Este método tiene la ventaja fundamental sobre el método de Series de Taylor de que no requiere la evaluación de las derivadas de segundo y mayor orden. La aproximación se obtiene evaluando la función $f(y,t)$ varias veces.

El método de Runge-Kutta de 4° orden, por ejemplo, está dado por la expresión

$$y_{n+1} = y_n + \frac{1}{6} (k_0 + 2k_1 + 2k_2 + k_3)$$

donde

$$k_0 = h_n f(y_n, t_n)$$

$$k_1 = h_n f\left(y_n + \frac{1}{2} k_0, t_n + \frac{h_n}{2}\right)$$

$$k_2 = h_n f\left(y_n + \frac{1}{2} k_1, t_n + \frac{h_n}{2}\right)$$

$$k_3 = h_n f(y_n + k_2, t_n + h_n)$$

Nótese que cuatro evaluaciones de la función $f(y,t)$ son requeridas

en cada paso $y_n \rightarrow y_{n+1}$.

El método de Runge-Kutta puede ser visto como una extensión del método de Simpson a la solución numérica de ecuaciones diferenciales. Si f fuera función de t únicamente, Simpson y Runge-Kutta serían idénticos

$$\int_{t_n}^{t_{n+1}} f(t) dt = \frac{h_n}{6} \left[f(t_n) + 4f\left(\frac{t_n + t_{n+1}}{2}\right) + f(t_{n+1}) \right]$$

El método de Runge-Kutta es fácil de programar y en la mayoría de los problemas es numéricamente estable. El método puede iniciarse por sí mismo, dado que sólo se requiere de una solución inicial y_0 . Además, el tamaño del intervalo h_n puede hacerse variar en cada evaluación.

(III) Método de Multipasos.

En los dos métodos anteriores los valores de y_{n+1} fueron calculados exclusivamente en base a los valores y_n, t_n y h_n . Es razonable pensar que si se empleara un número mayor de información, se mejoraría la exactitud. Por ejemplo, en la aproximación de $y(t_n)$ se podrían emplear los resultados previos,

$$y_{n-1}, y_{n-2}, \dots, y_{n-1}, f_{n-1}, f_{n-2}, \dots$$

siendo cada f_i igual a $f(y_i, t_i)$. El método de multipasos se basa en esta idea y es sumamente efectivo. La fórmula general del método de multipasos lineal es

$$y_{n+1} = \sum_{i=1}^k \alpha_i y_{n+1-i} + h_n \sum_{i=0}^k \beta_i f_{n+1-i}$$

donde k es un entero y α_i o β_i son coeficientes diferentes de cero. La fórmula es lineal en f .

Una vez que el método se ha inicializado, la evaluación del término y_{n+1} requerirá de los términos

$$y_n, y_{n-1}, \dots, y_{n-k+1}, \text{ y } f_n, f_{n-1}, \dots, f_{n-k+1}.$$

Es importante notar que si $\beta_0 = 0$, el método será explícito, y que si $\beta_0 \neq 0$ el método será implícito, ya que se requerirá el cálculo de ambos, f_{n+1} y y_{n+1} , en un mismo paso.

Por lo general, el método de multipasos se trabaja iterativamente de la manera siguiente:

- i) Un método explícito, llamado predictor, es aplicado primeramente.
- ii) Un método implícito, llamado corrector, es aplicado una o varias veces, a continuación.

Aplicando los dos métodos conjuntamente, el método completo se denomina método predictor-corrector. Un ejemplo es el método de Adams de cuarto orden, donde el método predictor está dado por la expresión

$$y_{n+1} = y_n + \frac{h_n}{24} (55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3})$$

y el método corrector viene expresado como

$$y_{n+1} = y_n + \frac{h}{24} (9 f_{n+1} + 19 f_n - 5 f_{n-1} + f_{n-2})$$

Ambas fórmulas son de 4° orden, es decir, requieren de cuatro evaluaciones de la función f en cada paso.

El algoritmo iterativo para el cálculo de y_{n+1} es como sigue:

(i) Empleando el método predictor calcule $y_{(n+1)}^{(0)}$, o sea una primera aproximación de y_{n+1} .

(ii) Evalúe la función

$$f_{n+1}^{(1)} = f(y_{n+1}^{(1)}, t_{n+1})$$

(iii) Calcule una mejor aproximación de y_{n+1} usando el método corrector

$(y_{n+1}^{(i)})$ es la i -ésima aproximación de y_{n+1} .

(iv) Si el valor absoluto de la diferencia $(y_{n+1}^{(i+1)} - y_{n+1}^{(i)})$ resulta mayor que una cierta tolerancia c , entonces incremente i en una unidad y vaya al paso (ii); de otra manera defina $y_{n+1} = y_{n+1}^{(i+1)}$ y el problema se da por resuelto.

Tres son los procesos que tienen lugar en este algoritmo:

(a) El paso del predictor (i), el cual denotaremos por la letra P.

(b) El paso de la evaluación (ii), denotado por E; y

(c) El paso de la corrección (iii) denotado por C.

El paso (iv) puede ser reemplazado por un paso en el cual se efectúan exactamente m iteraciones del corrector.

El método resultante se expresa en forma compacta como:

$$P(EC)^m$$

Problemas con Valores en la Frontera

Hasta aquí hemos discutido EDO con condiciones iniciales en la variable dependiente bien definidas. Ahora deseamos resolver un problema donde las condiciones de la variable dependiente se especifiquen para diversos valores de la variable independiente. Tal problema se reconoce como problema con valores en la "frontera".

Los métodos para la solución de estos problemas son completamente diferentes a los de valor inicial. Aquí daremos un método el cual permitirá reducir un problema con valores en la frontera a una secuencia de problemas con valor inicial.

Sea el siguiente sistema de EDO

$$y' = f(y, t) \text{ donde } y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

$$y_1(0) = \alpha$$

$$y_2(b) = \beta, \text{ y } b \neq 0$$

El algoritmo de solución es como sigue:

(i) Seleccione un valor ξ que aproxime el valor de $y_2(0)$.

(ii) Resuelva el problema de valor inicial

$\hat{y}' = f(\hat{y}, t)$, $\hat{y}(0) = \begin{pmatrix} a \\ \xi \end{pmatrix}$, donde \hat{y} es una aproximación de la función y .

(iii) Si el valor absoluto $|\hat{y}_2(b) - B|$ es menor que una tolerancia ϵ , entonces haga $y(t) = \hat{y}(t)$; de otra manera ajuste ξ y regrese al paso (ii).

El valor de ξ está estrechamente relacionado con los conceptos de región de convergencia del método de Newton-Raphson para la solución de sistemas de ecuaciones no-lineales .

APENDICE A

ALGORITMO DE THOMAS PARA LA SOLUCION DE MATRICES TRIDIAGONALES

Las ecuaciones son:

$$a_i u_{i-1} + b_i u_i + c_i u_{i+1} = d_i$$

$$\text{para } 1 \leq i \leq R$$

$$\text{con } a_1 = c_R = 0$$

El algoritmo es como sigue:

Primero, estime

$$\beta_i = b_i - \frac{a_i c_{i-1}}{\beta_{i-1}} \quad \text{con } \beta_1 = b_1$$

$$y_i = \frac{d_i - a_i y_{i-1}}{\beta_i} \quad \text{con } y_1 = \frac{d_1}{b_1}$$

Los valores de la variable dependiente se calculan a partir de

$$u_R = y_R \quad \text{y} \quad u_i = y_i - \frac{c_i u_{i+1}}{\beta_i}$$

ALGORITMO PARA LA SOLUCION DE MATRICES PENTADIAGONALES

Las ecuaciones son:

$$a_i u_{i-2} + b_i u_{i-1} + c_i u_i + d_i u_{i+1} + e_i u_{i+2} = f_i$$

$$\text{para } 1 \leq i \leq R$$

$$\text{con } a_1 = b_1 - a_2 = e_{R-1} = d_R = c_R = 0$$

El algoritmo es como sigue:

Primero calcule

$$\delta_1 = d_1 / c_1$$

$$\lambda_1 = e_1 / c_1$$

$$y_1 = f_1 / c_1$$

y

$$\mu_2 = c_2 - b_2 \delta_1$$

$$\delta_2 = (d_2 - b_2 \lambda) / \mu_2$$

$$\lambda_2 = a_2 \mu_2$$

$$\gamma_2 = (f_2 - b_2 \lambda) / \mu_2$$

es para valores de i entre $3 \leq i \leq (R-2)$, calcule

$$\beta_i = b_i - a_i \delta_{i-2}$$

$$\mu_i = c_i - \beta_i \delta_{i-1} - a_i \lambda_{i-2}$$

$$\delta_i = (d_i - \beta_i \lambda_{i-1}) / \mu_i$$

$$\lambda_i = a_i \mu_i$$

$$\gamma_i = (f_i - \beta_i \gamma_{i-1} - a_i \gamma_{i-2}) / \mu_i$$

calcule

$$\beta_{R-1} = b_{R-1} - a_{R-1} \delta_{R-2}$$

$$\mu_{R-1} = c_{R-1} - \beta_{R-1} \delta_{R-2} - a_{R-1} \lambda_{R-2}$$

$$\delta_{R-1} = (d_{R-1} - \beta_{R-1} \lambda_{R-2}) / \mu_{R-1}$$

$$\gamma_{R-1} = (f_{R-1} - \beta_{R-1} \gamma_{R-2} - a_{R-1} \gamma_{R-2}) / \mu_{R-1}$$

$$\beta_R = b_R - a_R \delta_{R-2}$$

$$\mu_R = c_R - \beta_R \delta_{R-1} - a_R \lambda_{R-2}$$

$$\gamma_R = (f_R - \beta_R \gamma_{R-1} - a_R \gamma_{R-2}) / \mu_R$$

γ y μ_i son usados únicamente en el cálculo de δ_i , λ_i y γ_i ; y no han de ser almacenados en memoria después de ser usados. Los δ_i y γ_i deben ser almacenados conforme son usados en la solución atrás. Esto es

$$u_R = \gamma_R$$

$$u_{R-1} = \gamma_{R-1} - \delta_{R-1} u_R$$

$$u_i = \gamma_i - \delta_i u_{i+1} - \lambda_i u_{i+2}$$

para $(R-2) \geq i \geq 1$

ALGORITMO PARA LA SOLUCION DE MATRICES BI-TRIANGULARES

Las ecuaciones son:

$$a_i^{(1)} u_{i-1} + a_i^{(2)} u_{i-2} + b_i^{(1)} v_i + b_i^{(2)} v_{i-1} + c_i^{(1)} u_{i+1} + c_i^{(2)} u_{i+2} = d_i^{(1)}$$

$$a_i^{(3)} u_{i-1} + a_i^{(4)} v_{i-1} + b_i^{(3)} u_i + b_i^{(4)} v_i + c_i^{(3)} u_{i+1} + c_i^{(4)} u_{i+2} = d_i^{(2)}$$

para $1 \leq i \leq R$

$$\text{con } a_i^{(m)} = c_i^{(m)} = 0$$

para $1 \leq m \leq 4$

El algoritmo es como sigue:

Primero calcule

$$\beta_i^{(1)} = b_i^{(1)} - a_i^{(1)} \lambda_{i-1}^{(1)} - a_i^{(2)} \lambda_{i-1}^{(2)}$$

$$\beta_i^{(2)} = b_i^{(2)} - a_i^{(2)} \lambda_{i-1}^{(2)} - a_i^{(3)} \lambda_{i-1}^{(3)}$$

$$\beta_i^{(3)} = b_i^{(3)} - a_i^{(3)} \lambda_{i-1}^{(3)} - a_i^{(4)} \lambda_{i-1}^{(4)}$$

$$\beta_i^{(4)} = b_i^{(4)} - a_i^{(4)} \lambda_{i-1}^{(4)} - a_i^{(1)} \lambda_{i-1}^{(1)}$$

$$\text{con } \beta_i^{(m)} = b_i^{(m)}$$

para $1 \leq m \leq 4$

$$\delta_i^{(1)} = d_i^{(1)} - a_i^{(1)} \gamma_{i-1}^{(1)} - a_i^{(2)} \gamma_{i-1}^{(2)}$$

$$\delta_i^{(2)} = d_i^{(2)} - a_i^{(2)} \gamma_{i-1}^{(2)} - a_i^{(3)} \gamma_{i-1}^{(3)}$$

$$\text{con } \delta_i^{(m)} = d_i^{(m)}$$

$$\text{y } \delta_i^{(4)} = d_i^{(4)}$$

$$\text{y } \mu_i = \beta_i^{(4)} \cdot \beta_i^{(3)} - \beta_i^{(3)} \cdot \beta_i^{(2)}$$

Los $\beta_i^{(m)}$, $\delta_i^{(m)}$ y μ_i son almacenados en memoria para facilitar el cálculo de las siguientes funciones y no requieren ser almacenados después de su utilización en

$$\lambda_i^{(1)} = (\beta_i^{(4)} c_i^{(1)} - \beta_i^{(3)} c_i^{(2)}) / \mu_i$$

$$\lambda_i^{(2)} = (\beta_i^{(3)} c_i^{(2)} - \beta_i^{(2)} c_i^{(3)}) / \mu_i$$

$$\lambda_i^{(3)} = (\beta_i^{(2)} c_i^{(3)} - \beta_i^{(1)} c_i^{(4)}) / \mu_i$$

$$\lambda_i^{(4)} = (\beta_i^{(1)} c_i^{(4)} - \beta_i^{(4)} c_i^{(1)}) / \mu_i$$

y

$$v_i^{(m)} (\beta_i^{(m)} \delta_i^{(m)} - \beta_i^{(m-1)} \delta_i^{(m-1)}) / \mu_i$$

$$v_i^{(m)} (\beta_i^{(m)} \delta_i^{(m)} - \beta_i^{(m-1)} \delta_i^{(m-1)}) / \mu_i$$

Los valores de $\lambda_i^{(m)}$ y $v_i^{(m)}$ deben ser almacenados conforme son empleados en la solución hacia atrás. Esto es

$$u_n = v_n^{(1)}$$

$$v_n = v_n^{(1)}$$

y

$$u_i = v_i^{(1)} - \lambda_i^{(1)} u_{i+1} - \lambda_i^{(1)} v_{i+1}$$

$$v_i = v_i^{(1)} - \lambda_i^{(1)} u_{i+1} - \lambda_i^{(1)} v_{i+1}$$

$$\text{para } (R - 1) \geq i \geq 1$$

ALGORITMO PARA LA SOLUCION DE MATRICES TRI-TRIDIAGONALES

Las ecuaciones son:

$$\begin{aligned} a_i^{(1)} u_{i-1} + a_i^{(2)} v_{i-1} + a_i^{(3)} w_{i-1} + b_i^{(1)} u_i + b_i^{(2)} v_i + b_i^{(3)} w_i \\ + c_i^{(1)} u_{i+1} + c_i^{(2)} v_{i+1} + c_i^{(3)} w_{i+1} = d_i^{(1)} \\ a_i^{(1)} u_{i-1} + a_i^{(2)} v_{i-1} + a_i^{(3)} w_{i-1} + b_i^{(1)} u_i + b_i^{(2)} v_i + b_i^{(3)} w_i \\ + c_i^{(1)} u_{i+1} + c_i^{(2)} v_{i+1} + c_i^{(3)} w_{i+1} = d_i^{(2)} \\ a_i^{(1)} u_{i-1} + a_i^{(2)} v_{i-1} + a_i^{(3)} w_{i-1} + b_i^{(1)} u_i + b_i^{(2)} v_i + b_i^{(3)} w_i \\ + c_i^{(1)} u_{i+1} + c_i^{(2)} v_{i+1} + c_i^{(3)} w_{i+1} = d_i^{(3)} \end{aligned}$$

$$\text{para } 1 \leq i \leq R$$

$$\text{con } a_i^{(m)} = c_i^{(m)} = 0 \quad \text{para } 1 \leq m \leq 9$$

algoritmo es como sigue:
 primero calcule

$$\begin{array}{l}
 \beta_i^{(1)} = b_i^{(1)} - a_i^{(1)} \lambda_{i-1}^{(1)} - a_i^{(2)} \lambda_{i-1}^{(2)} - a_i^{(3)} \lambda_{i-1}^{(3)} \\
 \beta_i^{(2)} = b_i^{(2)} - a_i^{(2)} \lambda_{i-1}^{(2)} - a_i^{(3)} \lambda_{i-1}^{(3)} - a_i^{(4)} \lambda_{i-1}^{(4)} \\
 \beta_i^{(3)} = b_i^{(3)} - a_i^{(3)} \lambda_{i-1}^{(3)} - a_i^{(4)} \lambda_{i-1}^{(4)} - a_i^{(5)} \lambda_{i-1}^{(5)} \\
 \beta_i^{(4)} = b_i^{(4)} - a_i^{(4)} \lambda_{i-1}^{(4)} - a_i^{(5)} \lambda_{i-1}^{(5)} - a_i^{(6)} \lambda_{i-1}^{(6)} \\
 \beta_i^{(5)} = b_i^{(5)} - a_i^{(5)} \lambda_{i-1}^{(5)} - a_i^{(6)} \lambda_{i-1}^{(6)} - a_i^{(7)} \lambda_{i-1}^{(7)} \\
 \beta_i^{(6)} = b_i^{(6)} - a_i^{(6)} \lambda_{i-1}^{(6)} - a_i^{(7)} \lambda_{i-1}^{(7)} - a_i^{(8)} \lambda_{i-1}^{(8)} \\
 \beta_i^{(7)} = b_i^{(7)} - a_i^{(7)} \lambda_{i-1}^{(7)} - a_i^{(8)} \lambda_{i-1}^{(8)} - a_i^{(9)} \lambda_{i-1}^{(9)} \\
 \beta_i^{(8)} = b_i^{(8)} - a_i^{(8)} \lambda_{i-1}^{(8)} - a_i^{(9)} \lambda_{i-1}^{(9)} - a_i^{(10)} \lambda_{i-1}^{(10)}
 \end{array}$$

con $\beta_i^{(m)} = b_i^{(m)}$ para $1 \leq m \leq 9$

$$\begin{array}{l}
 \delta_i^{(1)} = d_i^{(1)} - a_i^{(1)} \gamma_{i-1}^{(1)} - a_i^{(2)} \gamma_{i-1}^{(2)} - a_i^{(3)} \gamma_{i-1}^{(3)} \\
 \delta_i^{(2)} = d_i^{(2)} - a_i^{(2)} \gamma_{i-1}^{(2)} - a_i^{(3)} \gamma_{i-1}^{(3)} - a_i^{(4)} \gamma_{i-1}^{(4)} \\
 \delta_i^{(3)} = d_i^{(3)} - a_i^{(3)} \gamma_{i-1}^{(3)} - a_i^{(4)} \gamma_{i-1}^{(4)} - a_i^{(5)} \gamma_{i-1}^{(5)}
 \end{array}$$

con $\delta_i^{(m)} = d_i^{(m)}$ para $1 \leq m \leq 3$

Los $\beta_i^{(m)}$ son usados en la estimación de otras funciones y no requieren ser almacenados después del cálculo de:

$$\begin{array}{l}
 \Theta_i^{(1)} = \beta_i^{(1)} \rho_i^{(1)} - \rho_i^{(1)} \rho_i^{(1)} \\
 \Theta_i^{(2)} = \beta_i^{(2)} \rho_i^{(2)} - \beta_i^{(1)} \rho_i^{(1)} \\
 \Theta_i^{(3)} = \beta_i^{(3)} \rho_i^{(3)} - \rho_i^{(2)} \rho_i^{(2)} \\
 \Theta_i^{(4)} = \beta_i^{(4)} \rho_i^{(4)} - \rho_i^{(3)} \rho_i^{(3)} \\
 \Theta_i^{(5)} = \beta_i^{(5)} \rho_i^{(5)} - \rho_i^{(4)} \rho_i^{(4)} \\
 \Theta_i^{(6)} = \beta_i^{(6)} \rho_i^{(6)} - \rho_i^{(5)} \rho_i^{(5)} \\
 \Theta_i^{(7)} = \beta_i^{(7)} \rho_i^{(7)} - \rho_i^{(6)} \rho_i^{(6)} \\
 \Theta_i^{(8)} = \beta_i^{(8)} \rho_i^{(8)} - \rho_i^{(7)} \rho_i^{(7)} \\
 \Theta_i^{(9)} = \beta_i^{(9)} \rho_i^{(9)} - \rho_i^{(8)} \rho_i^{(8)}
 \end{array}$$

$$\mu_i = \alpha_i^{(1)} \beta_i^{(1)} + \alpha_i^{(2)} \beta_i^{(2)} + \alpha_i^{(3)} \beta_i^{(3)}$$

Los $\delta_i^{(k)}$, $\alpha_i^{(k)}$ y μ_i son empleados en la estimación de otras variables y no requieren ser almacenados después del cálculo

$$\lambda_i^{(1)} = (\alpha_i^{(1)} c_i^{(1)} + \alpha_i^{(2)} c_i^{(2)} + \alpha_i^{(3)} c_i^{(3)}) / \mu_i$$

$$\lambda_i^{(2)} = (\alpha_i^{(1)} c_i^{(2)} + \alpha_i^{(2)} c_i^{(3)} + \alpha_i^{(3)} c_i^{(1)}) / \mu_i$$

$$\lambda_i^{(3)} = (\alpha_i^{(1)} c_i^{(3)} + \alpha_i^{(2)} c_i^{(1)} + \alpha_i^{(3)} c_i^{(2)}) / \mu_i$$

$$\lambda_i^{(4)} = (\alpha_i^{(1)} c_i^{(4)} + \alpha_i^{(2)} c_i^{(5)} + \alpha_i^{(3)} c_i^{(6)}) / \mu_i$$

$$\lambda_i^{(5)} = (\alpha_i^{(1)} c_i^{(5)} + \alpha_i^{(2)} c_i^{(6)} + \alpha_i^{(3)} c_i^{(4)}) / \mu_i$$

$$\lambda_i^{(6)} = (\alpha_i^{(1)} c_i^{(6)} + \alpha_i^{(2)} c_i^{(4)} + \alpha_i^{(3)} c_i^{(5)}) / \mu_i$$

$$\lambda_i^{(7)} = (\alpha_i^{(1)} c_i^{(7)} + \alpha_i^{(2)} c_i^{(8)} + \alpha_i^{(3)} c_i^{(9)}) / \mu_i$$

$$\lambda_i^{(8)} = (\alpha_i^{(1)} c_i^{(8)} + \alpha_i^{(2)} c_i^{(9)} + \alpha_i^{(3)} c_i^{(7)}) / \mu_i$$

$$\lambda_i^{(9)} = (\alpha_i^{(1)} c_i^{(9)} + \alpha_i^{(2)} c_i^{(7)} + \alpha_i^{(3)} c_i^{(8)}) / \mu_i$$

$$\gamma_i^{(1)} = (\alpha_i^{(1)} \delta_i^{(1)} + \alpha_i^{(2)} \delta_i^{(2)} + \alpha_i^{(3)} \delta_i^{(3)}) / \mu_i$$

$$\gamma_i^{(2)} = (\alpha_i^{(1)} \delta_i^{(2)} + \alpha_i^{(2)} \delta_i^{(3)} + \alpha_i^{(3)} \delta_i^{(1)}) / \mu_i$$

$$\gamma_i^{(3)} = (\alpha_i^{(1)} \delta_i^{(3)} + \alpha_i^{(2)} \delta_i^{(1)} + \alpha_i^{(3)} \delta_i^{(2)}) / \mu_i$$

Los valores de $\lambda_i^{(k)}$ y $\gamma_i^{(k)}$ deben ser almacenados conforme son empleados en la solución hacia atrás. Esto es:

$$u_n = \gamma_n^{(1)}$$

$$v_n = \gamma_n^{(2)}$$

$$w_n = \gamma_n^{(3)}$$

$$u_i = \gamma_i^{(1)} - \lambda_i^{(1)} u_{i+1} - \lambda_i^{(2)} v_{i+1} - \lambda_i^{(3)} w_{i+1}$$

$$v_i = \gamma_i^{(2)} - \lambda_i^{(4)} u_{i+1} - \lambda_i^{(5)} v_{i+1} - \lambda_i^{(6)} w_{i+1}$$

$$w_i = \gamma_i^{(3)} - \lambda_i^{(7)} u_{i+1} - \lambda_i^{(8)} v_{i+1} - \lambda_i^{(9)} w_{i+1}$$

para $(R-1) \geq i \geq 1$

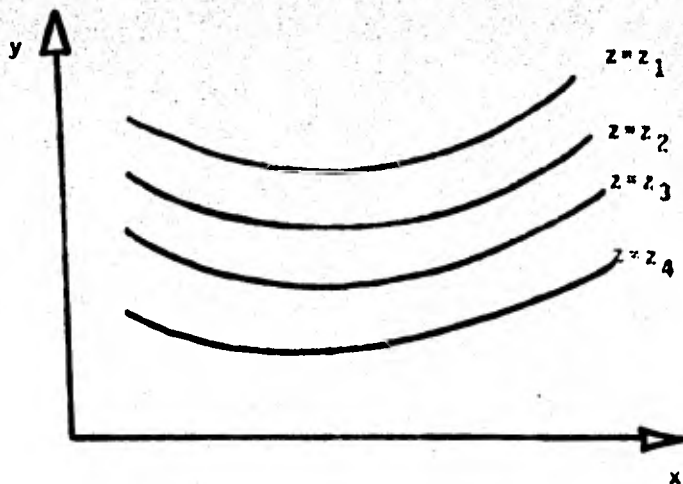
APÉNDICE B

PROCEDIMIENTO PARA EL AJUSTE DE FAMILIAS DE CURVAS*

Supongamos una función de dos variables independientes x y z .

$$y = y(x, z)$$

o gráficamente



Un procedimiento para ajustar un polinomio a esta familia de curvas es la siguiente:

- a) Leer los valores de x vs y , para $z = z_1$
- b) Ajustar un polinomio (por mínimos cuadrados) a estos puntos, obteniéndose una ecuación de la forma

$$y = a_{1,0} + a_{1,1}x + a_{1,2}x^2 + \dots + a_{1,n}x^n \quad \text{para } z = z_1$$

* Según notas de Tomás Linón.

en donde los coeficientes $a_{1,0}$, $a_{1,1}$, ..., $a_{1,n}$ se determinan mediante el ajuste polinomial.

c) Repetir los pasos (a) y (b) para las curvas correspondientes a $z = z_2$, $z = z_3$, ..., etc., obteniéndose

$$y = a_{2,0} + a_{2,1} x + a_{2,2} x^2 + \dots + a_{2,n} x^n \quad \text{para } z = z_2$$

$$y = a_{3,0} + a_{3,1} x + a_{3,2} x^2 + \dots + a_{3,n} x^n \quad \text{para } z = z_3 \dots \text{ etc.}$$

d) Este conjunto de polinomios se reduce a uno del tipo

$$y = b_0 + b_1 x + b_2 x^2 + \dots + b_n x^n$$

en donde los coeficientes b 's son funciones de z , notando que

$a_{1,0}$, $a_{2,0}$, $a_{3,0}$, ..., corresponden a $z = z_1$, $z = z_2$, $z = z_3$, ..., etc. Por lo tanto, los coeficientes en (a) están dados por expresiones del tipo

$$b_0 = c_{0,0} + c_{0,1} z + c_{0,2} z^2 + \dots$$

$$b_1 = c_{1,0} + c_{1,1} z + c_{1,2} z^2 + \dots$$

⋮

$$b_n = c_{n,0} + c_{n,1} z + c_{n,2} z^2 + \dots$$

Los coeficientes $c_{0,0}$, $c_{0,1}$, $c_{0,2}$, ..., de la ecuación para b_0 , son el resultado de un ajuste polinomial donde se han considerado los puntos $(z_1, a_{1,0})$, $(z_2, a_{2,0})$, $(z_3, a_{3,0})$, ..., etc.

De manera semejante, los coeficientes $c_{1,0}$, $c_{1,1}$, $c_{1,2}$ son el re-

sultado de un ajuste polinomial a los siguientes puntos

$$(z_1, a_{1,1}), (z_2, a_{2,1}), (z_3, a_{3,1}), \dots, \text{etc.}$$

y así sucesivamente hasta obtener los coeficientes de los polinomios de b_2, b_3, \dots, b_n .

APENDICE C

El siguiente es un listado por nombre de 99 subrutinas principales y alrededor de 140 subrutinas auxiliares. Estas subrutinas pueden constituir una herramienta útil para aquellos lectores que en el futuro su trabajo demande la elaboración de programas de cómputo ligados a problemas reales más específicos.

Las subrutinas se han clasificado en cinco grupos de acuerdo al problema que intentan resolver. Al margen izquierdo aparece el nombre de la subrutina principal, y a continuación, en el margen derecho, aparece una breve descripción del problema. En algunos casos, generalmente donde el problema a resolver es más complejo, se enlistan inmediatamente después de la descripción, un grupo de subrutinas auxiliares las cuales deberán acompañar a la subrutina principal.

PAQUETE DE SUBROUTINAS BASICAS

DECOMP Y SOLVE	Usadas conjuntamente permiten resolver el sistema $AX=B$. Donde A - matriz de coeficiente B - vector columna X - vector solución
SPLINE Y SEVAL	Usadas conjuntamente permiten interpolar aplicando el método de "splines" cúbicos.
QUANCP	Integra la función $f(x)$ entre los límites A y B usando el método de 8-púneles de Newton-Cotes.
RKF45	Diseñada para resolver sistemas de ecuaciones diferenciales de primer orden por el método de Runge-Kutta-Fehlberg de cuarto-quinto orden. RKFS FEHL.
ZEROIN	Calcula las raíces de la función $f(x)$ entre los intervalos AX y BX.
FMIN	Localiza el punto X dentro del intervalo AX, BX donde la función $f(x)$ adquiere un mínimo.

PAQUETE DE SUBROUTINAS NUMERO UNO

ZRPOIN	Calcula raíces de polinomios con coeficientes reales. UBERTST ZRPQLB ZRPQLC ZRPQLD ZRPQLE ZRPQLF ZRPQLG ZRPQLH ZRPQLI
FFT	Aplica transformada de Fourier a varias variables complejas.

Ajusta por mínimos cuadrados una serie de polinomios de tercer grado (splines) a un conjunto de puntos fijos en el plano.

ICSPEV
VERTST

Calcula una raíz de la ecuación no lineal $f(x)=0$.

Paquete de subrutinas empleadas en la solución de sistemas de ecuaciones no simétricas del orden de 150 o menor.

LSDCMP
LSDOLV
LSDMPR

Versión modificada de la subrutina "SPLINE" (ver SPLINE).

Versión modificada de la subrutina "SEVAL" (ver SEVAL).

Resuelve el problema ponderado de mínimos cuadrados siguiente:

$$\min_x \sum_{i=1}^N (W(i) * (B(i) - f(x, T(i)))^2)$$

donde $f(x, T(i))$ es un "modelo pre-especificado".

NECOMP
NOLVE
RESID
COVAR

Encuentra el mínimo de la función $f(x_1, \dots, x_N)$ usando el método "quasi-Newton" con primeras derivadas parciales.

LNSCH
OUTPT
CINCH
UPCHOL

Encuentra el mínimo de la función $f(x_1, \dots, x_N)$ usando el método "quasi-Newton" con diferencias finitas.

APROG
LINSCH
OUTPT
CINCH
UPCHOL

NSOIA

Resuelve un sistema de ecuaciones no lineales de forma $f(x)=0$, donde f y x son vectores de N componentes.

LSINVT*
INVKT2*

LCQNA

Intenta definir el punto X en el cual la función escalar $f(x)$, dos veces derivable y continua, adquiere su mínimo, sujeta a M ecuaciones restricción, pudiendo ser estas ecuaciones estructuradas con igualdades o con desigualdades.

DOT
LSOL
ELTSOL
LDLSOL
WQDLSL
PSQVVC
FMDEND
FORMCH
DELCON
ADDCON
KTRULT
NEWCON
PRJ:ISS
UPZTCZ
RFINEX
LNSRCH

QNA

Encuentra el mínimo de la función $f(x)$ de N variables independientes sujeta a no restricción.

LNSRCH
OUTT
FORMCH
HESS

ZAKLYT

Define las raíces de una función analítica compleja usando el método de Mueller.

VERTST

ZSYSTEM

Calcula el vector solución del sistema no lineal de N ecuaciones y N incógnitas $f(x)=0$.
Donde f y x son vectores de N componentes.

UERTST

Resuelve el siguiente problema de programación lineal:

minimiza la función lineal $C_1 X_1 + \dots + C_N X_N$ sujeta a las restricciones lineales

$$\begin{array}{rcccc} A_{11} X_1 + \dots + A_{1N} X_N & R(1) & B_1 & \\ \vdots & \vdots & \vdots & \\ A_{M1} X_1 + \dots + A_{MN} X_N & R(M) & B_M & \end{array}$$

donde cada $R(I)$ puede ser definida como

" \leq ", " \geq ", o " $=$ ".

LDLSOL
SQDLSL
PSPMVC
DFLCON
ADDCON
KTDULT
KFINEX
NEWCON

Regresión múltiple no lineal. Dado un conjunto de N observaciones, $Y(1), \dots, Y(N)$ de una variable dependiente Y , donde $Y(I)$ corresponde a la IV variable(s) independiente $T(I,1), \dots, T(I, IV)$, VARPRO trata de calcular un ajuste ponderado de mínimos cuadrados a la función ETA (el no delo) definida como:

$$ETA(\alpha, \beta; T) = \sum_{j=1}^L \beta_j \cdot \phi_j(\alpha; T) + \phi_{L+1}(\alpha; T)$$

donde ϕ son funciones no lineales. En otras palabras, determina los parámetros lineales $\beta(j)$ y el vector de parámetros no lineales α , minimizando

$$NORMA(RESIDUAL)^2 = \sum_{k=1}^N W_k^2 (Y_k - ETA(\alpha, \beta; T_k))^2$$

DPA
ORFAC1
ORFAC2
BACSUB
POSTPR
COV

INORM
INIT
VARERR

REALTR

Realiza la transformada de Fourier de un conjunto 2^N de datos reales.

LACRNI

Ajusta un polinomio de Lagrange de grado N , a $N+1$ puntos.

BESIO

Calcula valores de la función Bessel modificada de primera clase y orden cero.

PCMON
MONERR
NATSIO

BES2IO

Calcula valores de la función Bessel modificada de primera clase y orden cero multiplicada por $\text{EXP}(-X)$.

PCMON
MONERR
NATSIO

BES2II

Calcula valores de la función Bessel modificada de primera clase y orden uno multiplicada por $\text{EXP}(-X)$.

PCMON
MONERR
NATSII

BESII

Calcula valores de la función Bessel modificada de primera clase y orden uno.

PCMON
MONERR
NATSII

BESJO

Calcula valores de la función Bessel de primera clase y orden cero.

PCMON
MONERR

BESJI

Calcula valores de la función Bessel de primera clase y orden uno.

PCMON
MONERR

Calcula valores de la función Bessel modificada de segunda clase y orden cero.

PCNON
NONER
NATKO

Calcula valores de la función Bessel modificada de segunda clase y orden cero multiplicada por $\text{EXP}(-X)$.

PCNON
NONER
NATKO

Realiza suavizado de datos a través de "splines" cúbicos.

VERTST

Aproxima una función $f(x)$ por el método de Chebyshev.

LEQTY
LUDATP
LVELMP
VERTST

PAQUETE DE SUBROUTINAS NUMERO DOS

Solución de ecuaciones diferenciales ordinarias de orden variable.

Integra un sistema de hasta 20 ecuaciones diferenciales ordinarias de primer orden de la forma

$$\frac{dY(I)}{dt} = f(T, Y(1), \dots, Y(NEQM))$$

$Y(1)$ dada al tiempo T .

INTERP
STEP
ROOT

QUADSI

Aproxima la integral de la función $f(x)$ entre ciertos límites definidos por el tipo de cuadratura especificada.

GAUSSQ

Calculada nodos y coeficientes para la aproximación de una integral usando el método de la cuadratura. Aplicable en aproximaciones a la integral

$$\int_A^B f(x)W(x)dx = \sum_{j=1}^N W_j F(T_j)$$

SOLVE
CLASS
INTQL2

SQUANK

Integra la función FUS por el método de la cuadratura de SIMPSON ("sin ruido").

GPAR

Conjunto de subrutinas empleadas para la solución de sistemas de ecuaciones diferenciales ordinarias.

INTERP
STIFF
COSET
PSET
DRC
SOL

BCADRE

Integra la función $f(x)$ entre los límites A y B usando el método de extrapolación de Romberg.

UERTST

ODE

Integra un sistema de NEQN ecuaciones diferenciales ordinarias de primer orden.

DE
INTRP
STEP

BCSQUU

Integra una función "spline" cúbica entre los puntos A y B.

UERTST

BCCEVU

Evalúa derivadas parciales mixtas utilizando "splines" bicúbicas.

UERTST

PAQUETE DE SUBROUTINAS NUMERO TRES

RSCAB

Calcular valores característicos y vectores característicos del sistema $ABX = (\text{LAMBDA})X$.

donde A - matriz real y simétrica

B - matriz real y simétrica definida positivamente.

X - vector característico

LAMBDA - valor característico

REDUC2

TRED1

TQLRAT

TRED2

TQL2

REBAK

RSCBA

Calcula valores característicos y vectores característicos del sistema real y simétrico $BAX = (\text{LAMBDA})X$.

donde A - matriz real y simétrica

B - matriz real y simétrica definida positivamente

X - vector característico

LAMBDA - valor característico

REDUC2

TRED1

TQLRAT

TRED2

TQL2

REBAKB

RSC

Calcula valores característicos y vectores característicos del sistema real y simétrico $AX = (\text{LAMBDA})BX$

REDUC

TRED1

TQLRAT

TRED2

TQL2

REBAK

RSITIM

Mejora el vector solución del sistema lineal $AX = B$ por un método iterativo.

SOLVEN

Pondera los elementos del sistema lineal $AX = B$.

CHLEND
ENDSL1
CBNDIM

Usa el método de la descomposición de "Householder" en la matriz $A = Q * U$ (Q ortogonal, U triangular superior) para encontrar el vector que minimiza la norma euclidiana de $(AX - B)$.

Usa transformación "Householder" en la descomposición de una matriz no singular A en $A = QU$.

Pondera los elementos del sistema lineal $AX=B$ con la matriz diagonal D , tal que $DAX = DB$.

Pondera los elementos del sistema lineal $AX = B$ siguiendo el método de Cholesky.

CHLSKY
CHSLV1
CHITM

Dada la matriz A , define las matrices triangular inferior y superior L y U , y una matriz permutación P tal que $A = PLU$ o forma canónica de la matriz A .

Determina la "descomposición singular" $A = USV$ de una matriz real.

Encuentra valores y vectores característicos de una matriz real simétrica.

TRED1
TQLRAT
TRED2
TOL2

Encuentra valores y vectores característicos de una matriz real general.

BALANC
ELMHES
HQR
ELTRAN
HQR2
BALBAK

Encuentra valores y vectores característicos de una matriz real tridiagonal.

FIG1
INTQL1
FIG2
INTQL2

Encuentra valores y vectores característicos de una matriz real "compacta" y simétrica.

TRID3
TQLRAT
TQL2
TRBK3

Encuentra valores y vectores característicos de una matriz real en "banda" y simétrica.

BAKDR
TQLRAT
TQL2

Calcula valores y vectores característicos de una matriz real, simétrica y tridiagonal.

INTQL1
INTQL2

Encuentra la solución del sistema de ecuaciones $AX = B$.

DECOMP2
SOLVE2
INPRV2

INTPOT y IPTOLT

Ejecuta conjuntamente acumulaciones de productos escalares de dos vectores.

Calcula valores y vectores característicos de una matriz compleja general.

CBAL
CORTN
COM7R
COMQR2
CBANK2

- CHLND** Expresa una matriz A , positivamente definida, simétrica y en banda, como el producto de una matriz triangular inferior en banda y su transpuesta $A = L L^t$.
- CHDL1** Resuelve el sistema lineal $AX = B$ donde A es una matriz real, triangular inferior y en banda.
- CHDD1M** Por medio de un proceso iterativo, el vector solución del problema anterior, es mejorado.
- CHDLV1** Resuelve el sistema lineal $AX = B$ donde A es una matriz triangular inferior.
- CHITM** Por medio de un proceso iterativo, el vector solución del problema anterior, es mejorado.
- CHLSKY** Particiona la matriz A , real, simétrica y definida positivamente, en el producto de una matriz triangular inferior y su transpuesta $A = L L^t$.

PAQUETE DE SUBROUTINAS NUMERO CUATRO

- MMSTD** Calcula la media aritmética y la desviación estándar insesgada de los renglones de la matriz A .
- RECHSN** Calcula los coeficientes de regresión del sistema lineal $AX = B$.
- MMSTD**
WTVEC
COVARC
WMTX
WTBND*
NDEC*
SOLVEN
MSITM
- DISCR** Calcula un conjunto de funciones lineales las cuales sirven de índices en la clasificación de un individuo en distintos grupos (análisis de discriminación).

Calcula medias, desviaciones standard, sumas de productos cruzados de desviaciones y coeficientes de correlación.

Calcula la correlación canónica entre dos conjuntos de datos.

MINV *
 NROOT*
 EIGEN*

Generador de números aleatorios uniformemente distribuidos ($U(-0.5, 0.5)$). Recomendado sobre RANDU (versión de IBM).

Analiza un modelo simple de regresión lineal.

NOBET
 NOBETI
 RLPRDI
 UERTST

Analiza un modelo múltiple de regresión lineal.

LUFLMP
 NOBET
 NOBETI
 UERTST
 VMLFS

Grafica en la impresora hasta 10 funciones.

UERTST
 USMCCI *

Imprime histogramas.

UERTST

Imprime histogramas. Permite escribir dos frecuencias/barra.

UERTST

Determina un cierto número de regresiones óptimas basadas en subconjuntos de datos pertenecientes a aquel conjunto de datos de una regresión total.

MDFD
 RLEAP1
 RLEAP2
 RLEAP3
 UERTST

ULFAP

Imprime las regresiones óptimas calculadas por
ULFAP.

CGAMA

Genera pseudo-números aleatorios con distribución
Gamma (A,1). También puede usarse en la generación
de números aleatorios con distribución exponencial,
chi-cuadrada, chi, beta, t, y F.

CGNOR

CGUB

HERFI

MONRIS

MLRFCI

UERTST

CFIT

Prueba chi-cuadrada de bondad en el ajuste de cier-
tos datos a la distribución Gaussiana.

MDCDFI

UERTST

PTAUTO

Dada una serie de tiempo calcula:

1. Media y varianza
2. Autocovariancias
3. Autocovariancias y autocorrelaciones
4. Autocorrelaciones parciales

CTRNYC

Analiza tablas de contingencia.

UERTST

BFMIRI

Calcula medias, coeficientes de regresión con me-
didas de error y desviaciones standard para arre-
glos que puedan contener valores "perdidos".

UERTST

MDTPOS

Estima la densidad de probabilidad y la distribu-
ción acumulativa de una variable aleatoria con
distribución Poisson.

MDDT

Calcula valores de la distribución de probabili-
dad t "student".

UERTST

MDCDFI

Calcula valores de la distribución de probabili-
dad chi-cuadrada.

UERTST

Calcula valores de la distribución de probabilidad binomial.

UERTST

Calcula valores de la distribución inversa de probabilidad F.

MDBETA

MDSETI

UERTST

Distribución de probabilidad F.

MDEETA

UERTST

Calcula valores de la distribución de probabilidad F.

UERTST

Estima parámetros estadísticos en datos desagrupados.

UERTST

Estima parámetros estadísticos en datos agrupados.

UERTST

VABMXS *

VABMXF

Analiza datos en "diseño cuadrado latino".

UERTST

BIBLIOGRAFIA

1. Augx, J.M., Bass, D.M. Jr., and Whiting, R.L., 1960. Petroleum Reservoir Engineering. McGraw-Hill.
2. Saxendell, P.B., and Thomas, R., 1961. The Calculation of Pressure Gradients in High Rate Flowing Wells. Journal of Pet. Tech.
3. Beckmann, P., 1973. Orthogonal Polynomials for Engineers and Physicists. The Golem Press. Boulder, Colorado.
4. Carnahan, B., Luther, H.A., and Wilkes, J.O., 1969. Applied Numerical Methods. John Wiley & Sons.
5. Forsythe, G.E., Malcolm, M.A., and Moler, C.B., 1977. Computer Methods for Mathematical Computations. Prentice-Hall.
6. Gill, W.N., and Scher, M., 1961. A Modification of the Momentum Transport Hypothesis, A.I.Ch.E. Journal, 7, 61-65.
7. Isaacson, E., and Keller, H.B., 1966. Analysis of Numerical Methods. John Wiley & Sons.
8. Journel, A.G., and Huijbregts, Ch. J., 1978. Mining Geostatistics. Academic Press, London.
9. Krilov, V.I., 1962. Approximate Calculation of Integrals. Macmillan, New York.
10. McCracken, D.D., 1972. A Guide to Fortran IV Programming. Second Edition. John Wiley & Sons.
11. Poettmann, F.H., and Carpenter, P.G., 1952. The Multiphase Flow of Gas, Oil and Water Through Vertical Flow Strings with Application to the Design of Gas Lift Installations. Drill and Prod. Proc., API.
12. Ralston A., 1965. A First Course in Numerical Analysis. McGraw-Hill.
13. Ramson, I., Hornberger, G.M., and Molz, F. J., 1971. Numerical Methods in Subsurface Hydrology. Wiley-Interscience.

14. Söbel, I. N., 1976. Método de Monte Carlo. Lecciones Populares de Matemáticas. MIR, Moscú.
15. Stewart, G.W., 1973. Introduction to Matrix Computations. Academic Press, New York.
16. Tomás, L.J., 1974. Transporte de Gas en Régimen Permanente. Proyecto D-341A. Publication No. 74DH/164. Instituto Mexicano del Petróleo.

