

22  
250

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
Facultad de Ingeniería



APUNTES DE LA ASIGNATURA TECNICAS  
AVANZADAS DE PROGRAMACION

T E S I S  
Que para obtener el Titulo de  
INGENIERO PETROLERO  
p r e s e n t a  
JOSE ANTONIO RUIZ GARCIA

México, D. F.

1993

TESIS CON  
FALLA LE ORIGEN



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

# CONTENIDO

---

	PAGINA
INTRODUCCION	1
I CONCEPTOS BASICOS SOBRE PROGRAMACION	3
I.1 CONCEPTOS FUNDAMENTALES	3
I.2 NUMEROS EN PUNTO FLOTANTE	6
I.3 ERRORES DE PRECISION O REDONDEO	8
I.3.1 <i>Machine Epsilon</i>	10
I.3.2 Ejemplo de error por redondeo	11
I.4 ERRORES INHERENTES, DE TRUNCAMIENTO Y DE APROXIMACION FUNCIONAL	13
I.5 EJERCICIOS RESUELTOS	15
I.6 PROBLEMAS PROPUESTOS	26
II SOLUCION A PROBLEMAS MATRICIALES	28
II.1 SISTEMAS DE ECUACIONES LINEALES	28
II.1.1 Regla de Cramer	31
II.1.2 Método de Eliminación Gaussiana	32
II.1.3 Algoritmo de Thomas	34
II.1.4 Subrutinas DECOMP y SOLVE	38
II.1.5 Solución al problema de flujo bidireccional, monofásico, transitorio, a través de medios porosos	40
II.1.5.1 Convergencia	42
II.1.5.2 Estabilidad	42
II.1.5.3 Esquema Explicito	43
II.1.5.4 Esquema Implícito	46
II.1.5.5 Esquema de Crank-Nicolson	48
II.1.5.6 Simulación numérica de flujo bidireccional, monofásico, transi- torio, a través de medios porosos	51

	PAGINA
II.2 SISTEMAS DE ECUACIONES NO LINEALES	55
II.2.1 Método iterativo de Jacobi	58
II.2.2 Método iterativo de Gauss-Seidel	59
II.2.3 Método iterativo de Newton-Raphson	60
II.2.4 Solución al problema de diseño de un sistema de recolección y distribución de gas	62
II.3 EJERCICIOS RESUELTOS	72
II.4 PROBLEMAS PROPUESTOS	75
III INTERPOLACION Y APROXIMACION NUMERICA	78
III.1 INTERPOLACION EXACTA	78
III.1.1 Interpolación polinomial	80
III.1.2 Interpolación polinomial de Lagrange	83
III.1.3 Valuación de polinomios	84
III.1.4 Subrutina SPLINE	85
III.1.4.1 Interpolación Spline en tres dimensiones	92
III.1.5 Solución a la interpolación del factor de compresibilidad de un gas natural	93
III.2 APROXIMACION FUNCIONAL	97
III.2.1 Método de los Mínimos Cuadrados	100
III.2.2 Descomposición del valor singular (DVS) y subrutina SVD	103
III.2.3 Ajuste de familias de curvas	108
III.2.4 Introducción al Método de Estimación de Kriging	110
III.2.5 Cálculo de los coeficientes del modelo de optimización de la perforación	115
III.3 EJERCICIOS RESUELTOS	121
III.4 PROBLEMAS PROPUESTOS	125

	PAGINA
IV INTEGRACION NUMERICA	128
IV.1 REGLAS DEL RECTANGULO Y DEL TRAPEZOIDE	130
IV.2 CUADRATURA SPLINE	138
IV.3 REGLA DE SIMPSON	140
IV.4 CUADRATURA ADAPTADA	142
IV.4.1 Subrutina QUANCB	149
IV.5 CUADRATURAS GAUSSIANAS	150
IV.5.1 Polinomios ortogonales	151
a) Polinomios de Legendre	152
b) Polinomios de Laguerre	153
c) Polinomios de Chebyshev	153
d) Polinomios de Hermite	154
IV.5.2 Cuadratura de Gauss-Legendre	156
IV.5.3 Cuadratura de Gauss-Laguerre	159
IV.5.4 Cuadratura de Gauss-Chebyshev	161
IV.5.5 Cuadratura de Gauss-Hermite	163
IV.6 CALCULO DEL POTENCIAL O PSEUDO PRESION DE LOS GASES REALES	164
IV.7 PROBLEMAS PROPUESTOS	169
 V APROXIMACION A LA SOLUCION DE ECUACIONES DIFERENCIALES	 171
V.1 ECUACIONES DIFERENCIALES ORDINARIAS	171
V.1.1 Técnicas para la solución numérica de una Ecuación Diferencial Ordinaria	179
V.1.1.1 Método de Taylor	179
V.1.1.2 Método de Runge-Kutta	181
V.1.1.3 Método Multipasos	182
V.1.1.4 Problema con valores en la frontera	184
V.1.1.5 Subrutina RKF45	185
V.2 ECUACIONES DIFERENCIALES PARCIALES	187
V.2.1 Método de Diferencias Finitas	188
V.2.1.1 Esquema Explícito	190

	PAGINA
V.2.1.2 Errores de convergencia y estabilidad en el Esquema Explícito	191
V.2.1.3 Esquema Implícito	194
V.2.1.4 Errores de convergencia y estabilidad en el Esquema Implícito	196
V.2.1.5 Esquemas ponderados	196
V.2.1.6 Errores de convergencia y estabilidad en el Esquema Ponderado	197
V.2.1.7 Tipos de condiciones de frontera	198
V.2.1.8 Ecuaciones Diferenciales Parciales en dos dimensiones	198
V.3 EJERCICIOS RESUELTOS	198
V.4 PROBLEMAS PROPUESTOS	199
VI TECNICAS AVANZADAS DE PROGRAMACION APLICABLES AL AMBITO PETROLERO	201
VI.1 PROGRAMAS REGRESML Y SIMPP	202
VI.2 CONSTRUCCION Y USO DE PROGRAMAS DE BIBLIOTECA EN QUICK BASIC	207
VI.3 DISEÑO OPTIMO DE PROGRAMAS DE COMPUTO	209
1) Uso de subíndices múltiples	209
2) Manejo de funciones tabuladas	211
3) Uso de subrutinas	212
4) Entradas y salidas	213
5) Consideraciones adicionales	215
6) Facilidad de implementación	215
VI.4 PROBLEMAS PROPUESTOS	216
APENDICE	217
BIBLIOGRAFIA	219

## INTRODUCCION

Es difícil precisar cuándo aparecen en la historia de la humanidad los primeros números, pero es innegable que éstos surgieron ante la necesidad de cuantificar y organizar entes e ideas.

Poco a poco, el hombre tuvo que emplear números cada vez más grandes y complejos, y de manera paralela, debió crear máquinas capaces de representarlos rápida y eficazmente. Las primeras computadoras eran verdaderos "monstruos" que empleaban, debido a sus dimensiones, edificios completos para su instalación y operación, teniendo como componentes principales tubos al vacío o bulbos.

Posteriormente, con la invención del transistor, disminuyen las dimensiones del equipo y se acelera la velocidad de respuesta del procesador. No debe subestimarse la capacidad o complejidad de las computadoras sólo porque su tamaño se reduce cada vez más. En la actualidad, con el uso del circuito integrado y del L.S.I. (Large Scale Integration) o *chip*, se ha logrado incrementar la velocidad y la calidad de las respuestas del computador.

Aunque innumerables productos de *Hardware* y *Software* actualmente empleados, fueron concebidos y desarrollados hace muchos años, la industria de la Informática apenas empieza a madurar, y no existe alguna área de la Ciencia con la que no esté relacionada. Es en este contexto que se abordarán y expondrán tópicos sobre técnicas numéricas y de programación, aplicables a diversos problemas de Ingeniería Petrolera.

Los programas de cómputo y subrutinas básicas, presentados en este trabajo, se estructuraron en lenguaje *Basic* (ambiente *Quick Basic*, versión 4.5), y como todo *software*, son susceptibles de ser mejorados y optimizados. Los listados de los programas que logran la solución de problemas específicos del Área Petrolera, se excluyeron de este compendio con la finalidad de ahorrar espacio impreso; en su lugar, se anexa un disco flexible (*floppy disk* o *diskette*, de 3<sup>1</sup>/<sub>2</sub>"", doble densidad, *DD*, doble cara, *MF2*), dividido en subdirectorios, para la consulta y uso inmediato de los mismos; además, se incluyen las bases de datos necesarias para la buena ejecución de cada proyecto y las subrutinas básicas tratadas en estos apuntes. Para un conocimiento más amplio de los archivos contenidos en el *diskette* (denominado *disco de aplicación*), se recomienda ejecutar el archivo *INDICE.BAT* (archivo *Batch file*), o revisar el apéndice de este trabajo.

Como herramienta de apoyo se empleó una computadora personal (*PC*) *ZEOS 486SLC*, 100% compatible con *IBM*, tipo *AT* (procesador *80486*, 33 Mhz) y monitor cromático *VGA* (*Video Graphics Adapter*).

Es importante destacar la invaluable e incansable asesoría del M. en I. Néstor Martínez Romero y el desinteresado apoyo del Ing. Gustavo Hernández García, aspectos fundamentales en la elaboración de estos apuntes.



## CAPITULO I

### CONCEPTOS BASICOS SOBRE PROGRAMACION

Un conocimiento sólido sobre conceptos de computación es primordial para entender el funcionamiento de los diferentes computadores que se emplean actualmente y para aquellos que aparecerán en un futuro próximo.

A pesar de múltiples esfuerzos tecnológicos, las computadoras no son máquinas "perfectas", como la mayoría de la gente cree, debido a que sólo permiten el manejo de cierto rango de números reales, los cuales son representados con una cantidad determinada de cifras, incurriéndose, inevitablemente, en errores diversos.

La finalidad de este capítulo es describir dichos errores y generar en la actitud del lector un afán por evitarlos, para que así, al emplear equipos de cómputo en la solución de problemas reales, obtenga resultados con rapidez y precisión.

#### I.1 CONCEPTOS FUNDAMENTALES.

Un computador es un sistema digital que efectúa operaciones de cómputo. El término *digital* implica que la información es representada en el computador por medio de variables, las cuales poseen un número limitado de valores discretos. Las computadoras digitales funcionan de manera más confiable si únicamente emplean dos estados. Esto es debido a restricciones físicas de los componentes, aunado al hecho de que la lógica humana tiende a ser

binaria (es decir: cierto o falso, sí o no); entonces, los componentes digitales sólo admitirán dos valores, los dígitos 0 y 1 que conforman al sistema de números binarios.

Un dígito binario constituye una unidad básica denominada bit. La información se representa en las computadoras por grupos de bits, los que caracterizan no solamente a números binarios sino también a otros símbolos discretos cualesquiera, sean éstos, dígitos decimales o letras del alfabeto.

A diferencia de los números decimales, que emplean el sistema de base 10, los números binarios usan un sistema de base 2. Por ejemplo, el número binario 101101, representa una cantidad que puede transmutarse a un número decimal, multiplicando cada bit por la base 2 elevada a una potencia entera, de la manera siguiente:

$$(1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 45$$

Entonces, los seis bits (101101), caracterizan un número binario cuyo equivalente decimal es 45; sin embargo, estos bits podrían representar un código binario para una letra del alfabeto o un código de control para especificar alguna decisión lógica.

La información binaria se representa por cantidades físicas, nombradas *señales*. Las señales eléctricas, como el voltaje, existen a través del sistema digital en dos formas reconocibles, que representan a una variable binaria igual a 1 o a 0. Las terminales de entrada de un circuito digital aceptan señales binarias dentro de tolerancias permisibles y los circuitos, en las terminales de salida, emiten respuestas con señales binarias que también cumplen con dichas tolerancias.

Una unidad de memoria es una colección de registros de almacenamiento, junto con los circuitos necesarios para transferir

información dentro y fuera de los mismos. Si se puede tener acceso a los registros de memoria para realizar transferencia de datos cuando sea requerido (lectura/escritura), se habla de una Memoria de Acceso Aleatorio, RAM (*Random Access Memory*); si sólo es factible la obtención de información (lectura), entonces se hace mención de una Memoria de Solamente Lectura, ROM (*Read Only Memory*), en la cual, la información normalmente es creada por el fabricante.

Una unidad de memoria almacena información en grupos de bits, denominados *palabras* (*words*). Así, una *palabra* es una entidad de "n" bits. A una palabra de ocho bits comúnmente se le asigna el nombre de *byte*. Es costumbre referirse al número de bytes en una unidad de memoria con la literal K, la cual representa 1024 bytes, de esta manera 1K = 1024 bytes y 64K =  $2^{16}$  bytes. El rango más común de memoria RAM es de 1024 a  $2^{20}$  bytes. La Tabla 1.1 esboza una comparación entre estos conceptos informáticos y conceptos cotidianos.

CONCEPTO EN INFORMATICA	EQUIVALENCIA PRACTICA
8 bits = 1 byte 1 byte = 1 carácter	Nada equiparable. Cualquier letra, número, símbolo o espacio entre éstos.
3 K = 3072 bytes 1 Megabyte = $1 \times 10^6$ bytes	1 página (formato carta). Aprox. 333 páginas.

Tabla 1.1.

## I.2 NUMEROS EN PUNTO FLOTANTE.

Se han propuesto diferentes métodos para representar al conjunto infinito de números reales mediante el empleo de sistemas finitos de cómputo.

El método que actualmente se emplea en la gran mayoría de las computadoras es el denominado "Números en Punto Flotante", mediante el cual un número  $X$  puede obtenerse con la siguiente expresión:

$$X = \pm \left( \frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \dots + \frac{d_t}{\beta^t} \right) \beta^e$$

donde:

$d_1, d_2, \dots, d_t$ , son enteros que cumplen con  
 $d_1 > 0, 0 \leq d_i \leq \beta - 1 \quad (i = 2, \dots, t)$

Y:

$L \leq e \leq U$  ; siendo  $e$ , un número entero

Esta expresión conforma un conjunto  $F$  de números, caracterizado por cuatro parámetros: un número base  $\beta$ , un número de precisión  $t$  y un rango del exponente  $[L, U]$ .

Cuando  $X$  es diferente de cero,  $X \in F$ , y  $d_1 \neq 0$ , se dice que  $X$  está normalizado. El entero  $e$ , es llamado exponente,  $\beta^e$  es la parte entera y la fracción  $f$ , está dada por la expresión:

$$f = \left( \frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \dots + \frac{d_t}{\beta^t} \right)$$

La Tabla 1.2 muestra los valores usuales de los parámetros de la representación en punto flotante, empleados por diversas compañías. La columna  $\beta^{t-L}$  representa un valor estimado de la precisión aritmética del sistema (*Machine Epsilon*), la cual será tratada más adelante.

COMPUTADORA	$\beta$	t	L	U	$\beta^{t-L}$
Honeywell 6000	2	27	-128	127	1.49 E-08
Control Data 6600	2	48	-975	1070	7.11 E-15
Burroughs B5500	8	13	-51	77	1.46 E-11
Hewlett Packard HP-45	10	10	-98	100	1.00 E-09
Texas Instruments SR-5x	10	12	-98	100	1.00 E-11
IBM 360 y 370 :					
(short precision)	16	6	-64	63	9.54 E-07
(long precision)	16	14	-64	63	2.22 E-16

Tabla 1.2.

El conjunto de números en punto flotante,  $F$ , es finito y discontinuo; los elementos que contiene están dados por:

$$[2(\beta-1)\beta^{t-L}(U-L+1)+1]$$

Además no están igualmente espaciados a lo largo de su rango de valores. Aplicando la fórmula anterior, es posible estimar la cantidad de elementos que puede representar exactamente una computadora, por ejemplo:

$F$  IBM LONG PRECISION  $\approx 1.729362257 \text{ E}19$  números

$F$  HP 45  $\approx 3.582 \text{ E}12$  números

### 1.3 ERRORES DE PRECISION O REDONDEO.

Si suponemos un sistema en punto flotante (computadora hipotética), con los siguientes parámetros:  $\beta = 2$ ,  $t = 3$ ,  $L = -1$  y  $U = 2$ ,  $F$  estará integrado por 33 elementos, por tanto, no es posible que todos los números reales puedan ser representados y almacenados con este sistema, entonces cada número en  $F$  representa un intervalo de números reales. Si  $X$  representa un real, contenido en un cierto intervalo del conjunto  $F$ , entonces al número en  $F$  más próximo a  $X$  se le denotará como  $fl(x)$  (Figura 1.1).

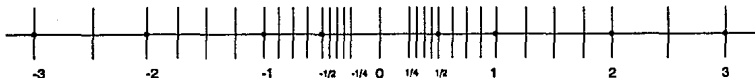


Figura 1.1.

El error relativo en que se incurre, es función de  $\beta$  y  $t$ :

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

En sistemas de punto flotante con  $\beta = 2$ , o igual a una potencia de 2, al sumar 10 elementos de tamaño 0.1 no se obtiene como resultado al número 1.0, puesto que 0.1 no tiene representación finita en potencias de 1/2. Matemáticamente, esto es:

$$\frac{1}{10} = \frac{0}{2^1} + \frac{0}{2^2} + \frac{0}{2^3} + \frac{1}{2^4} + \frac{1}{2^5} + \frac{0}{2^6} + \frac{0}{2^7} + \dots$$

Así, empleando subíndices para denotar la base  $\beta$ :

$$\begin{aligned} (0.1)_{10} &= (0.000110011001100\dots)_2 \\ &= (0.012121212\dots)_4 \\ &= (0.063146314\dots)_8 \\ &= (.199999999\dots)_{16} \end{aligned}$$

Los elementos del lado derecho de las expresiones anteriores han sido truncados después de "t" dígitos, y al sumar diez de ellos, el resultado arrojado no ha sido la unidad.

Al sumar dos números en punto flotante ( $x \oplus y$ ), donde  $\oplus$  representa la operación suma en punto flotante, comúnmente se tiene como resultado una cantidad que no está en  $F$ ; el valor real de la suma se puede aproximar por  $fl(x + y)$ . Lo ideal es que si  $(x \oplus y)$  cae dentro del rango de  $F$ , entonces  $(x + y) = (x \oplus y) = fl(x + y)$ , lo cual, en la gran mayoría de las computadoras se cumple sólo para algunos valores de  $x$  y de  $y$ .

El error por precisión o redondeo, está representado por la diferencia numérica, en valor absoluto, entre  $(x + y)$  y  $(x \oplus y)$ . Análogamente, puede calcularse en la resta, la multiplicación o la división.

En el sistema en punto flotante  $F$  supuesto con anterioridad se observa que:

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

$$\frac{5}{4} \oplus \frac{3}{8} \approx \frac{3}{2} \text{ ó } \frac{7}{4} \neq \frac{5}{4} + \frac{3}{8} = \frac{13}{8} \quad (\text{SUMA REAL})$$

La razón por la cual  $\frac{5}{4} + \frac{3}{8}$  no está en el conjunto  $F$ , se debe al espaciamiento de sus elementos. Ahora, si se desea realizar la

suma  $\frac{7}{2} + \frac{7}{2} = 7$ , no estaría en F, ya que el número siete sería mayor que el número más grande del conjunto, presentándose entonces una señal de *overflow*, interrumpiéndose en la mayoría de las computadoras el proceso de cálculo. En la multiplicación ( $xy$ ) se encuentra con mayor frecuencia esta señal, puesto que esta operación abarca  $2t$  o  $2t-1$  dígitos significativos.

Por otro lado, una señal de *underflow* suele desplegarse cuando se multiplican dos números ( $x, y$ ), diferentes de cero, cuyo resultado es más pequeño que el menor número, también diferente de cero, representado en F. Este desplegado, aunque es inusual, se presenta aún en la suma. Las operaciones de adición y de multiplicación en punto flotante no son distributivas ni asociativas pero, sí conmutativas.

Los errores de precisión se generan al emplear en los cálculos, series o números irracionales ( $\sqrt{2}$ ,  $\pi$ ,  $e$ , etcétera), ya que existe la imposibilidad de representarlos exactamente en la memoria de la computadora, entonces, al efectuarse alguna operación se usa la aproximación más cercana.

### I.3.1 MACHINE EPSILON.

La precisión de la suma en punto flotante puede determinarse por medio del *Machine Epsilon*, esto es, el número más pequeño ( $\epsilon$ ) tal que:

$$(1 \oplus \epsilon) > 1$$

Existen algunos métodos para calcular el valor (o una aproximación) de  $\epsilon$ . A continuación se presenta un programa, en



lenguaje *Basic*, que permite el cálculo aproximado del *Machine Epsilon* de un equipo de cómputo:

```

CLS
EPS = 1.0
DO
  EPS = 0.5 * EPS
  EPSP1 = EPS + 1.0
LOOP WHILE EPSP1 > 1.0
PRINT " ε = " ; EPS
END
  
```

Este concepto define la precisión y la calidad del computador, esto es, entre menor sea el valor de  $\epsilon$ , mayor será la precisión del procesador del equipo empleado. En la Tabla 1.3 se concentran los valores de  $\epsilon$ , para algunos equipos de cómputo.

Equipo de cómputo	Machine Epsilon ( $\epsilon$ )
PC XT, PRINTFORM (8088)	5.960464E-8
PC AT, ZEOS 486SLC (80486)	5.960464E-8
MICRO C., TIMEX SINCLAIR 1000	1.1641588E-10
MICRO C., CASIO FX-795P	5.820766091E-11

Tabla 1.3.

### 1.3.2 EJEMPLO DE ERROR POR REDONDEO.

Supóngase que desea evaluar la función  $e^x$  mediante un programa de cómputo. Tal función puede representarse como la sumatoria de una serie convergente infinita, o sea:

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

Si el sistema de cómputo a usar está caracterizado por  $\beta = 10$  y  $t = 5$ , y se necesita el valor de  $e^{-x}$ , para  $x = 5.5$ , entonces, sustituyendo en la serie anterior se tendrán los resultados siguientes para cada término:

$$\begin{array}{r}
 e^{-5.5} = \quad 1.0000 \\
 \quad - 5.5000 \\
 \quad + 15.125 \\
 \quad - 27.780 \\
 \quad + 38.129 \\
 \quad - 41.942 \\
 \quad + 38.446 \\
 \quad - 30.208 \\
 \quad + 20.768 \\
 \quad - 12.692 \\
 \quad + 6.9808 \\
 \quad - 3.4902 \\
 \quad + 1.5997 \\
 \quad \vdots \\
 \hline
 \quad 0.0026963
 \end{array}$$

El resultado final contempla sólo 25 términos, puesto que, los términos subsecuentes no lo modifican. ¿Es esta respuesta satisfactoria? ;No!, ya que el resultado correcto es:

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

Nótese que algunos términos, y consecuentemente las sumas intermedias, son mucho más grandes en magnitud que la respuesta final; por ejemplo, el quinto término, 38.129, involucra en sí mismo un error por redondeo tan considerable como el del resultado final, debido a que el cuarto dígito decimal se ha perdido. Por el impacto que tiene esa pérdida sobre el resultado, al fenómeno anterior se le denomina *cancelación catastrófica*.

Una solución podría ser el emplear un mayor número de cifras significativas, aunque esto generaría un incremento en el tiempo de ejecución y a menudo requiere de técnicas de programación

un tanto sofisticadas. Sin embargo, una solución práctica es la de calcular  $e^x$ , para  $x = 5.5$  y luego obtener su valor recíproco, es decir:

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

Obsérvese, que el resultado anterior reduce el error relativo a un 0.007% y además contiene 5 cifras significativas. Se puede concluir que el cálculo de ciertas operaciones, empleando una computadora, no debe ser excesivamente complicado para incurrir en grandes errores por redondeo.

Mediante técnicas adecuadas de programación siempre es posible reducir al mínimo los errores de redondeo.

#### I.4 ERRORES INHERENTES, DE TRUNCAMIENTO Y DE APROXIMACION FUNCIONAL.

Los modelos matemáticos de procesos físicos o naturales, indefectiblemente, acarrear *errores inherentes*, los cuales, son el resultado de una inadecuada o parcial interpretación del fenómeno natural a representar, de la aleatoriedad del mismo y de la falta de certeza de las mediciones experimentales. A menudo, un modelo incluye únicamente los rasgos más sobresalientes del proceso físico y es deliberadamente despojado de detalles superfluos, que son catalogados como de segundo término.

Aquellos algoritmos que usan sólo operaciones aritméticas y ciertas operaciones lógicas, como lo son las comparaciones algebraicas, son llamados *Métodos Numéricos*. El error que se

origina al aproximar la solución de un problema matemático a través de un método numérico, usualmente es denominado **error de truncamiento**. El aproximar una función  $f(x)$ , por medio de una función "conveniente",  $g(x)$ , promueve la aparición del **error de aproximación funcional**. Error que frecuentemente se presenta en dos casos. El primero, es reemplazar una función  $f(x)$ , difícil de evaluar o manipular (por ejemplo: diferenciales o integrales), por una expresión,  $g(x)$ , más simple. El segundo caso, es la interpolación de valores tabulados de funciones. La función  $f(x)$  es cuantitativamente conocida para un número finito de argumentos, llamados **puntos base**, es decir:

$x_0$	,	$f(x_0)$
$x_1$	,	$f(x_1)$
⋮	⋮	⋮
$x_L$	,	$f(x_L)$
⋮	⋮	⋮
$x_n$	,	$f(x_n)$

Entonces, debe generarse la función de aproximación,  $g(x)$ , que estime el valor de  $f(x)$ , para  $x \neq x_i, i = 0, 1, \dots, n$ .

La efectividad de una función  $g(x)$ , y por ende una disminución del error de aproximación funcional, depende de varios factores, entre ellos están los siguientes: el conocimiento de la función original, el origen y la precisión de los valores tabulados, y la precisión ofrecida por la aproximación empleada.

## 1.5 EJERCICIOS RESUELTOS.

PI-1. La serie de Taylor para calcular la función error es:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{n! (2n+1)}$$

la cual converge para todo valor de  $X$ . Escriba un programa que evalúe  $\operatorname{erf}(x)$  usando esta serie, considerando tantos términos como sea necesario, de manera que el primero en descartarse no altere considerablemente la suma acumulada. Como esta serie es alternante, el error causado por truncar la suma infinita debe ser menor que el error de redondeo. Dilucide el efecto del error de redondeo, comparando el valor calculado de  $\operatorname{erf}(x)$  con uno obtenido de tablas o de una subrutina publicada. Considere  $X = 0.5, 1, 5$  y  $10$ . Explique los resultados.

### Sugerencia:

Forsythe propone un ingenioso ciclo iterativo, en lenguaje **Basic**, a utilizar en parte del desarrollo de su programa<sup>(4)</sup>:

```
DO
  OLDS = S
  EN = EN + 1.0
  T = -XSO * T * (2.0 * EN - 1.0) / (EN * (2.0 * EN + 1.0))
  S = S + T
LOOP WHILE OLDS <> S
```

¿Qué valores deben asignarse a  $T$ ,  $S$ ,  $EN$  y  $XSO$  antes de iniciar con el ciclo? No olvide el factor  $2/\sqrt{\pi}$ .

### Solución:

Empleando la serie de Taylor "pura", esto es, sin

modificaciones, se obtuvieron los siguientes resultados (programa PI-1A en el *diskette* de consulta anexo, revisar apéndice):

x	erf(x)	NUMERO DE TERMINOS
0.5	0.52050024	4
1.0	0.84269917	7
5.0	3.87075E+07	35 *
10.0	4.094655E+28	35 *

Ahora, para utilizar el algoritmo propuesto por Forsythe, deben determinarse previamente, los valores iniciales de las variables involucradas. Si se desarrolla la serie de Taylor, se tiene:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \left[ x - \frac{x^3}{3} + \frac{x^5}{10} - \frac{x^7}{42} + \dots \right]$$

Por otro lado, para el algoritmo:

$$\text{EN} = 1 \quad \rightarrow \quad T = \frac{-XSQ(T)}{3}$$

$$\text{EN} = 2 \quad \rightarrow \quad T = \frac{XSQ^2(T)}{10}$$

$$\text{EN} = 3 \quad \rightarrow \quad T = \frac{-XSQ^3(T)}{42}$$

Para que ambas expresiones sean iguales, los valores iniciales que se asignarían deben ser: EN = 0, S = X, T = X y XSQ = X<sup>2</sup>. Se recomienda al lector la demostración de lo anterior.

\* MÁXIMO NÚMERO DE ITERACIONES PERMITIDAS POR EL EQUIPO DE COMPUTO UTILIZADO (PC-COMPATIBLE IBM, TIPO AT), SIN INCURRIR EN UN ERROR DE OVERFLOW.

Los resultados del uso del artificio se presentan a continuación (problema PI-1B en el disco flexible de consulta):

x	erf(x)	NUMERO DE ITERACIONES
0.5	0.52049989	6
1.0	0.84270066	10
5.0	13.89649200	76
10.0	1.960269E+38	69 *

Los resultados arrojados por ambos procedimientos son muy semejantes para valores del argumento iguales o inferiores a la unidad, sin embargo, su alternancia induce errores mayúsculos para valores grandes de x, provocados por cancelación. Los dos métodos presentan cierta dispersión con respecto a los valores tabulados para erf(x). Consúltense la Tabla 1.4 y la Figura 1.2.

x	erf(x)
0.5	0.6915
1.0	0.8413
3.0	0.9987
3.9	1.0000

TABLA DE VALORES DE erf(x)

Tabla 1.4.

\* MAXIMO NUMERO DE ITERACIONES PERMITIDAS POR EL EQUIPO DE COMPUTO UTILIZADO (PC-COMPATIBLE IBM, TIPO AT), SIN INCURRIR EN UN ERROR DE OVERFLOW.

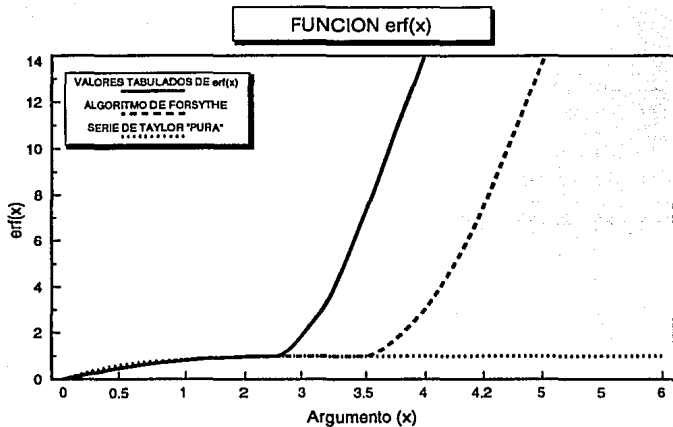


Figura 1.2.

PI-2. ¿Qué respuesta se obtiene al ejecutar el programa en *Basic* mostrado a continuación, en una computadora? Explique por qué.

```

REM problema PI-2
DEFDBL H, X-Y
CLS
X = 0.0
H = 0.1
FOR I = 1 TO 10
    X = X + H
NEXT I
Y = 1.0 - X
PRINT X ; ABS(Y)
END

```



Respuesta:

Al ejecutar el programa anterior en una computadora personal, tipo AT, se obtiene el siguiente resultado:

```
X = 1.000000014901161000000
Y = 0.000000014901161193848
```

El valor lógico a obtener para la variable X debía ser el de la unidad y cero para la variable Y, pero debido a la representación en punto flotante que posea la computadora en uso, se tendrán valores que tiendan a la unidad y a cero. En este caso el valor de Y representa el error de precisión absoluto, cometido al calcular X.

PI-3. ¿Es posible representar exactamente el número decimal 0.1 en su computadora? Si no, ¿cuál es el número, en punto flotante, más cercano? ¿Cuál es el número menor, también en punto flotante, más cercano? ¿Qué valor es asignado en cada una de las siguientes expresiones? Datos:  $X = 0.1$ ,  $Y = 0.1$ . Use el siguiente programa en lenguaje *Basic*:

```
REM problema PI-3
CLS
DEFDBL Y
X = 0.1
Y = 0.1
PRINT X , Y
Y = 0.1D0
PRINT Y
X = 1.0/10.0
Y = 1.0D0/10.0D0
PRINT X , Y
```

```

X = 1.0E-1
Y = 1.0D-1
PRINT X , Y
INPUT " X = " , X
INPUT " Y = " , Y
PRINT X , Y
END

```

**Resultados:**

Al ejecutar el programa anteriormente presentado, se obtienen los siguientes valores :

X	Y
0.1	0.1000000014901161
---	0.1
0.1	0.1
0.1	0.1
0.1	0.1

La computadora empleada representa con una exactitud de ocho decimales al número 0.1. Los números en punto flotante más cercanos a 0.1, superior e inferiormente, son aproximadamente iguales a  $0.1 \pm 1.4901161E-09$ .

PI-4. a) ¿Cómo representa internamente su computadora a los números  $1/2$ ,  $2/3$  y  $3/5$ ? Use una notación apropiada, por ejemplo, binaria, octal o hexadecimal. ¿Cómo son estos números representados en los sistemas de punto flotante de otras computadoras, como lo son IBM 360, CDC 6600, Univac 1108, Honeywell 6000, PDP-11, Burroughs 6500, etcétera?

b) Considere el siguiente programa en lenguaje *Basic*:

```

REM problema PI-4
CLS
H = 1.0/2.0
X = 2.0/3.0 - H
Y = 3.0/5.0 - H
E = ( X + X + X ) - H
F = ( Y + Y + Y + Y + Y ) - H
Q = F / E
PRINT "H = "; H
PRINT "X = "; X
PRINT "Y = "; Y
PRINT "E = "; E
PRINT "F = "; F
PRINT "Q = "; Q
END

```

La variable *Q* puede asumir diferentes valores que dependerán del punto flotante del *hardware* aritmético usado por la computadora. Trate de estimar el valor de *Q* para computadoras con las que esté familiarizado. Corra el programa en tantos equipos como le sea posible para corroborar los resultados. Explíquelos.

Resultados:

a) Empleando subíndices para denotar la base, se observa:

$$\begin{aligned}
 \left(\frac{1}{2}\right)_{10} &= (0.5)_{10} = (0.1)_2 = (0.1)_8 = (0.1)_{16} \\
 \left(\frac{1}{3}\right)_{10} &= (0.666666\bar{6})_{10} = (0.101010\bar{10})_2 = (0.525252\bar{52})_8 = (0.AAAAA\bar{A})_{16} \\
 \left(\frac{1}{5}\right)_{10} &= (0.6)_{10} = (.10011001)_2 = (0.46314631)_8 = (0.999999\bar{9})_{16}
 \end{aligned}$$

Por lo tanto, como los números  $2/3$  y  $3/5$  no tienen una representación finita en los sistemas en punto flotante más usuales (bases 2, 8 y 16), ya que son números periódicos, se generarán errores de precisión, al usar éstos y otros números similares, en la solución de algún problema.

La tabla que se muestra a continuación, indica el número base utilizado en el sistema en punto flotante de equipos de cómputo de uso común.

MARCA DEL EQUIPO	NUMERO BASE
IBM 360	16
CDC 6600	2
Univac 1108	2
Honeywell 6000	2
PDP 11	2
Burroughs 6500	8

b) Empleando una computadora tipo AT, y simple precisión para las variables E, F y G; se tiene:

H = 0.5  
 X = 0.1666667  
 Y = 0.1  
 E = 1.490116 E-08  
 F = 7.450581 E-09  
 G = 0.5

Contrariamente a lo esperado  $G \neq (0/0=\infty)$ , ya que se presentan errores de precisión y truncamiento, aunados al tipo de representación en punto flotante del equipo usado. Ahora, si se emplea doble precisión para todas las variables, se obtiene:

H = 0.5  
 X = 0.16666666666666667

Y = 0.1

E = -2.775557561562891D-17

F = 2.775557561562891D-17

G = -1 → G ≠ (0/0=∞)

PI-5. Evalúe la siguiente serie infinita:

$$\Phi(x) = \sum_{k=1}^{\infty} \frac{1}{k(k+x)}$$

para  $X = 0$  a  $1.0$  con incrementos de  $0.1$ , con un error inferior a  $0.5 \times 10^{-9}$ .

**Nota:**

Esto requiere emplear tanto el análisis humano como el poder de una computadora, ya que ni lo uno ni lo otro causaría éxito usándose individualmente. No desperdicie el tiempo de máquina de su computadora tratando de calcular la sumatoria sin razonar antes la posible solución (¿Cuánto tiempo podría desperdiciarse?).

**Sugerencia:**

Use el siguiente artificio<sup>(4)</sup>:

$$\frac{1}{k(k+1)} = \frac{1}{k} - \frac{1}{k+1}$$

para probar que  $\Phi(1) = 1$ . Entonces, exprese  $\Phi(x) - \Phi(1)$  como una serie que converge más rápido que la que define a  $\Phi(x)$ . Se tendrá que repetir este truco antes de obtener la serie para computar  $\Phi(x)$  para lograr una convergencia rápida.

Solución:

En una computadora tipo AT, sin utilizar el artificio sugerido, se obtuvieron los resultados que a continuación se muestran (programa PI-5A en el disco flexible de consulta):

X	$\Phi(X)$	ITERACIONES	TIEMPO (seg)
0.0	1.644725	4097	19.21
0.1	1.534399	4096	19.72
0.2	1.440670	4096	19.72
0.3	1.359873	4096	19.72
0.4	1.289369	4096	19.77
0.5	1.227203	4096	19.56
0.6	1.171897	4096	19.72
0.7	1.122311	4096	19.72
0.8	1.077551	4096	19.72
0.9	1.036903	4096	19.66
1.0	0.999853	5793	27.79

Ahora, sustituyendo el valor de  $X = 1$  en la serie infinita propuesta, además, aplicando propiedades de las series, se tiene:

$$\Phi(1) = \sum_{k=1}^{\infty} \frac{1}{k(k+1)} = \sum_{k=1}^{\infty} \frac{1}{k} - \sum_{k=1}^{\infty} \frac{1}{k+1}$$

Las series del lado derecho de la ecuación, son series conocidas, de las cuales se sabe que:

$$\sum_{k=1}^{\infty} \frac{1}{k} = 2$$

$$\sum_{k=1}^{\infty} \frac{1}{k+1} = 1$$

Entonces, reemplazando estos valores en la serie  $\Phi(1)$ :

$$\Phi(1) = \sum_{k=1}^{\infty} \frac{1}{k(k+1)} = 2 - 1 = 1$$

Con lo que se prueba que  $\Phi(1) = 1$ .

Por otro lado, expresando  $\alpha(x) = \Phi(x) - \Phi(1)$ , se observa lo siguiente:

$$\begin{aligned} \alpha(x) &= \Phi(x) - \Phi(1) \\ &= \sum_{k=1}^{\infty} \frac{1}{k(k+x)} - \sum_{k=1}^{\infty} \frac{1}{k(k+1)} \\ &= \sum_{k=1}^{\infty} \left[ \frac{1}{k(k+x)} - \frac{1}{k(k+1)} \right] \\ &= \sum_{k=1}^{\infty} \left[ \frac{k - kx}{k^2 + k^2x + k^2x^2} \right] \end{aligned}$$

Se deduce que la expresión resultante convergerá más aprisa que la serie original, puesto que en su denominador existen elementos elevados a potencias mayores que 1.

La manera más ágil de calcular  $\Phi(x)$  será entonces, computar la serie  $\alpha(x)$  y sumarle  $\Phi(1) = 1$ . Los resultados arrojados, usando este artificio (programa PI-5B en el *diskette* de consulta), son:

x	$\Phi(x)$	ITERACIONES	TIEMPO (seg)
0.0	1.644930	323	2.14
0.1	1.534603	312	2.14
0.2	1.440877	377	2.53
0.3	1.360080	361	2.41
0.4	1.289576	343	2.30
0.5	1.227410	406	2.74
0.6	1.172104	377	2.59
0.7	1.122519	432	2.96
0.8	1.077758	377	2.58
0.9	1.037110	261	1.76
1.0	1.000000	3	0.00

Se concluye que la rapidez con la cual una computadora puede ofrecer resultados, se incrementa cuando el análisis humano depura un proceso a ejecutar, es decir, no basta con poseer la computadora más sofisticada y costosa del mercado, si no se dispone de analistas de cómputo capaces de conformar una buena mancuerna.

## 1.6 PROBLEMAS PROPUESTOS.

PPI.1. Algunas funciones matemáticas pueden ser difíciles de calcular, además, al emplearse diversas aproximaciones para resolver un mismo problema, los resultados pueden ser disímboles. Para ejemplificar estas aseveraciones, realice un programa de cómputo que evalúe la función error por tres métodos, que enseguida se discutirán, para argumentos en el rango de  $0 \leq x \leq 5$ , a intervalos de 0.5. Dicha función está definida como:

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

En aquellos métodos que consideren series infinitas, encuentre el resultado usando 5 y 10 términos. Utilice simple precisión en todos sus cálculos. Compare los resultados con los ofrecidos por alguna subrutina disponible o por alguna tabla de  $\operatorname{erf}(x)$ ; incluya sus observaciones al respecto.

### Método 1.

La serie de Taylor puede utilizarse para representar a la función exponencial ( $e^x$ ). Entonces, sustituyéndola en la función error, se tiene:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x (1 - t^2 + \frac{t^4}{2!} - \frac{t^6}{3!} + \dots) dt$$

Por lo que, integrando, se obtiene:

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

Como puede observarse, la serie anterior es alternante, por lo cual será inexacta para valores grandes del argumento, ya que existirán errores sustanciales por cancelación.



Método 2.

Si se integra por partes la función  $\text{erf}(x)$ , se logra la siguiente serie:

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

Esta serie no es alternante. ¿Qué opina de su eficiencia?

Método 3.

Una función racional de aproximación (ajuste funcional), en ocasiones brinda mayor exactitud cuando se usa el mismo número de coeficientes comparada con la serie original. Tal aproximación, para el caso presentado, podría ser:

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

Donde:

$$a_1 = 0.278393$$

$$a_2 = 0.230389$$

$$a_3 = 0.000972$$

$$a_4 = 0.078108$$

$$|\epsilon(x)| \leq 5 \times 10^{-4}$$

¿Es esta aproximación más exacta que las dos anteriores?

## CAPITULO II

### SOLUCION A PROBLEMAS MATRICIALES

El concepto de *simulación de un sistema*, es decir, la simulación del comportamiento de un sistema, sea éste, por ejemplo, un yacimiento o una red de distribución de gas, tiene su base en el desarrollo y operación de un modelo físico y/o matemático, cuyo comportamiento refleje ante condiciones cualesquiera, el del sistema considerado.

Un modelo matemático es una ecuación, o un conjunto de ellas, que simula los procesos físicos que ocurren en un sistema al variar una o todas las condiciones iniciales. La Simulación Numérica de Yacimientos Petroleros, que es ejemplo de la aplicación de un modelo matemático, hace uso de métodos numéricos para lograr la solución de la(s) ecuación(es) que constituye(n) al sistema.

El objetivo del presente capítulo, es mostrar técnicas numéricas simples que faciliten la solución de sistemas de ecuaciones lineales y no lineales.

#### II.1 SISTEMAS DE ECUACIONES LINEALES.

Un sistema de ecuaciones lineales es un conjunto de ecuaciones con la forma siguiente:

$$\begin{array}{r} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{array} \quad \dots(2.1)$$

que en forma matricial, puede representarse como:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Empleando notación algebraica, se tiene:  $A \bar{x} = \bar{b}$ , donde  $A$  representa a la matriz de coeficientes,  $\bar{x}$  al vector columna de incógnitas y  $\bar{b}$  al vector columna de términos independientes; a la matriz  $(A, \bar{b})$ , de orden " $m \times (n+1)$ ", se le asigna el nombre de *matriz ampliada del sistema*.

El grupo de " $n$ " valores  $x_1, x_2, \dots, x_n$  que satisfacen simultáneamente a todas las ecuaciones, es la solución del sistema. Dicha solución puede tipificarse en tres casos, éstos son:

- Primer caso. Solución única, se dice entonces que el sistema es *compatible* o *consistente*, y *determinado*. La matriz  $A$  es *no-singular*.
- Segundo caso. Si la solución es múltiple, se tiene un sistema *compatible* (*consistente*) e *indeterminado*. La matriz  $A$  es *singular*.
- Tercer caso. El sistema no admite solución, por lo que se le nombra *incompatible* o *inconsistente*. La matriz  $A$  es *singular*.

Es bien conocido que el determinante de una matriz coadyuva a la detección de su singularidad (si el  $\det A \neq 0$ ,  $A$  es no singular),

pero, al emplear equipos de cómputo en la solución de sistemas de ecuaciones, se generan errores (consultar capítulo I), que pueden provocar la emisión equivocada de criterios sobre el problema a resolver. Para esclarecer tal aseveración, supóngase una matriz diagonal de tamaño  $100 \times 100$ , cuyos elementos diagonales son todos iguales a 0.1. Es evidente que su determinante es distinto de cero ( $0.1^{100}$ ), no obstante, este valor es tan pequeño que cualquier computador arrojaría un valor de cero a pesar de que la matriz en cuestión es no singular. Debido al problema anterior, es conveniente introducir el concepto de *número de condición* de una matriz (Cond), el cual refleja de manera más eficaz que efectuando el cálculo de su determinante, la cercanía a la estabilidad o no singularidad. Matemáticamente, el número de condición se denota como:

$$\text{Cond}(A) = \frac{\max_x \frac{\|A \bar{x}\|}{\|\bar{x}\|}}{\min_x \frac{\|A \bar{x}\|}{\|\bar{x}\|}} \geq 1$$

El símbolo "  $\| \ \|$  " representa la norma o módulo de un vector. Es usual que el número de condición se exprese también de la siguiente forma:

$$\text{Cond}(A) = \frac{\lambda_{\max}}{\lambda_{\min}} \geq 1$$

donde:

$\lambda$  = Valor característico o *Eigen-valor*

Si A es una matriz cuadrada ( $n \times n$ ), sus valores característicos pueden obtenerse empleando el Método de las Potencias<sup>(4)</sup>, basado en la relación  $A \bar{x} = \lambda \bar{x}$ , o encontrando las raíces del polinomio<sup>(4)</sup>:

$$\det(A - \lambda I) = 0$$

donde:

$I =$  matriz identidad

Si  $\text{Cond} = 1$ , la matriz es no singular (estable) y el sistema de ecuaciones tendrá solución única, entonces un número de condición mucho mayor que la unidad indicará inestabilidad o singularidad, en este caso, con una pequeña variación en alguno de los coeficientes de la matriz, se obtendrá un vector  $\bar{x}$  diferente del originalmente calculado (soluciones múltiples). En pocas palabras, el número de condición esboza la sensibilidad del vector solución  $\bar{x}$ , a cambios en los elementos de  $A$  y  $\bar{b}$ .

Enseguida se detallarán las técnicas más empleadas en la solución de sistemas de ecuaciones lineales.

### II.1.1 REGLA DE CRAMER.

Este método resuelve sistemas de ecuaciones lineales de tamaño  $(n \times n)$ . Si el determinante de la matriz  $A$  es diferente de cero ( $\det A \neq 0$ ), el valor de la  $k$ -ésima incógnita se dilucidará efectuando el cociente de los determinantes de las matrices  $A_k$  y  $A$ , donde  $A_k$ , se obtiene al reemplazar la  $k$ -ésima columna de la matriz de coeficientes ( $A$ ), por el vector de términos independientes ( $\bar{b}$ ).

En otras palabras, sea  $A \bar{x} = \bar{b}$ , un sistema de "n" ecuaciones lineales con "n" incógnitas y sea  $A = [a_{i,j}]$  su matriz de coeficientes. Si el  $\det A \neq 0$ , entonces:

$$x_k = \frac{\det A_k}{\det A}, \quad k = 1, 2, \dots, n$$

donde:

$$A_k = [c_{i,j}] \quad , \quad c_{i,j} = \begin{cases} a_{i,j} & , \text{ para } j \neq k \\ b_i & , \text{ para } j = k \end{cases}$$

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

$$j = 1, 2, \dots, n$$

La Regla de Cramer se tipifica como una técnica "exacta", puesto que no es iterativa. Si se contempla un equipo de cómputo que ejecute 10,000 multiplicaciones/seg, siendo "n" el orden del determinante, se presentará lo siguiente<sup>(8)</sup>:

n	número de mult. $\approx n^3$	tiempo de máquina
5	100	0.01 seg
12	5 E+08	10 horas
20	2 E+18	10 años
30	3 E+32	10 <sup>24</sup> años

Se concluye que este método no es adecuado para resolver sistemas mayores de (4 x 4), pues se realizarán gran cantidad de operaciones (ya que implica el cálculo de determinantes), convirtiéndose en "lenta" la obtención del vector solución.

## II.1.2 METODO DE ELIMINACION GAUSSIANA.

El método de Gauss es utilizado para solucionar sistemas de ecuaciones lineales del tipo (m x n). Consiste en obtener sistemas equivalentes, a partir del original, aplicando **transformaciones elementales** sobre la matriz ampliada del sistema, hasta llevarlo a la **forma escalonada** (matriz triangular superior), esto es, cuando el número de ceros anteriores al primer elemento no nulo de cada

renglón aumenta al pasar de un renglón al siguiente, hasta llegar, eventualmente, a renglones cuyos elementos son todos nulos.

Las principales transformaciones elementales, son:

- a) Intercambio de un renglón por otro.
- b) Adición, término a término, de dos renglones.
- c) Multiplicación de la totalidad de los elementos de un renglón por un escalar diferente de cero.

Una vez reducido el sistema original a la forma escalonada, la solución se logra, de manera directa, por *sustitución hacia atrás* ( $x_i$  se obtiene por simple despeje de la  $i$ -ésima ecuación, entonces se sustituye en la ecuación " $i-1$ " y se calcula el valor de  $x_{i-1}$ , este procedimiento se continúa hasta tenerse los " $n$ " valores de  $x$ ).

Como este método no es de aproximaciones sucesivas, la solución debía ser exacta, sin embargo, debido a errores de redondeo o precisión, no lo es. Para minimizar el error, es recomendable seleccionar en cada ocasión como pivote al mayor elemento, en valor absoluto, del sistema<sup>(4)</sup> (se recomienda consultar el problema resuelto PII-1, página 72).

Considerando una computadora digital que ejecute 10,000 multiplicaciones/seg, siendo " $n$ " el orden del sistema, se tiene<sup>(4)</sup>:

$n$	núm. de mult. $\approx n^3/3$	tiempo de máquina
5	40	0.005 seg
12	500	0.05 "
20	2500	0.25 "
30	9000	1.0 "

Se observa que esta técnica no realiza una gran cantidad de operaciones, y tampoco invierte demasiado tiempo de

máquina, conforme el número de ecuaciones del sistema aumente.

### II.1.3 ALGORITMO DE THOMAS.

Se emplea para dar solución a sistemas de ecuaciones lineales de orden  $(n \times n)$ , cuya matriz de coeficientes presente un carácter "n-diagonal". Para efectos didácticos, considérese un sistema tri-diagonal, tal que:

$$A = \begin{bmatrix} b_1 & c_1 & & & & & & & 0 \\ a_2 & b_2 & c_2 & & & & & & \vdots \\ & a_3 & b_3 & c_3 & & & & & \vdots \\ & & \cdot & \cdot & \cdot & & & & \vdots \\ \vdots & & & & & & & & \vdots \\ 0 & \dots & & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & & a_n & b_n & \end{bmatrix}$$

Este algoritmo se basa en la Técnica de Descomposición L-U de una matriz, esto es:

$$A = L U$$

Donde, U es una matriz triangular superior y L una triangular inferior, o sea:

$$L = \begin{bmatrix} a_1 & & & & & & & & 0 \\ a_2 & a_2 & & & & & & & \vdots \\ & a_3 & a_3 & & & & & & \vdots \\ & & \cdot & \cdot & \cdot & & & & \vdots \\ \vdots & & & & & & & & \vdots \\ 0 & \dots & & & & & a_n & a_n & \end{bmatrix}$$



$$U = \begin{bmatrix} 1 & \beta_1 & & & & 0 \\ & 1 & \beta_2 & & & \vdots \\ & & & \ddots & & \vdots \\ & & & & & \vdots \\ & & & & & 1 & \beta_{n-1} \\ 0 & \dots & & & & & 1 \end{bmatrix}$$

El producto  $(L U)$  es:

$$LU = \begin{bmatrix} \alpha_1 & (\alpha_1 \beta_1) & & & & & 0 \\ \alpha_2 & (\alpha_2 \beta_1 + \alpha_2) & (\alpha_2 \beta_2) & & & & \vdots \\ & \alpha_3 & (\alpha_3 \beta_2 + \alpha_3) & (\alpha_3 \beta_3) & & & \vdots \\ & & \vdots & \vdots & & & \vdots \\ & & & \vdots & & & \vdots \\ & & & & (\alpha_{n-1}) & (\alpha_{n-1} \beta_{n-2} + \alpha_{n-1}) & (\alpha_{n-1} \beta_{n-1}) \\ 0 & \dots & & & \alpha_n & (\alpha_n \beta_{n-1} + \alpha_n) & \end{bmatrix}$$

Como  $A = L U$ , cada elemento de  $(L U)$  será igual al de  $A$ , por lo cual:

$$\alpha_i = a_i, \quad i = 2, 3, \dots, n$$

$$\alpha_i = b_i$$

$$\alpha_i = b_i - (\alpha_i \beta_{i-1}), \quad i = 2, 3, \dots, n$$

$$\beta_i = \frac{c_i}{\alpha_i}, \quad i = 1, 2, \dots, (n-1)$$

Entonces, un sistema de ecuaciones  $A \bar{x} = \bar{f}$  puede representarse como (recordemos que  $A = L U$ ):

$$L U \bar{x} = \bar{f}$$

Además, considerando que  $U \bar{x} = \bar{y}$ , se tendrá el sistema:

$$L \bar{y} = \bar{f}$$

el cual puede resolverse fácilmente por sustitución hacia adelante.  
De manera que:

$$y_1 = \frac{f_1}{a_1}$$

$$y_i = \frac{f_i - (a_{i1} y_{1-1})}{a_i}, \quad i = 2, 3, \dots, n$$

Finalmente, el sistema  $U \bar{x} = \bar{y}$ , se soluciona mediante una sustitución hacia atrás, por lo tanto:

$$x_n = y_n$$

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

Para una matriz penta-diagonal de orden  $(n \times n)$  como la siguiente, se tiene:

$$A = \begin{bmatrix} c_1 & d_1 & 0 & e_1 & & & \dots & 0 \\ b_2 & c_2 & d_2 & 0 & e_2 & & & \vdots \\ 0 & b_3 & c_3 & d_3 & 0 & e_3 & & \\ a_4 & 0 & b_4 & c_4 & d_4 & 0 & e_4 & \\ & & & \vdots & \vdots & \vdots & & \\ & & & \vdots & \vdots & \vdots & & \\ & & & & \vdots & \vdots & & \\ & & & & a_{n-1} & 0 & b_{n-1} & c_{n-1} & d_{n-1} \\ 0 & \dots & & & a_n & 0 & b_n & c_n \end{bmatrix}$$

Mientras que las matrices L y U serán:

$$L = \begin{bmatrix} \alpha_1 & & & & & & \dots & 0 \\ \beta_2 & \alpha_2 & & & & & & \vdots \\ \gamma_3 & \beta_3 & \alpha_3 & & & & & \\ \delta_4 & \gamma_4 & \beta_4 & \alpha_4 & & & & \\ \vdots & & & \vdots & \vdots & & & \\ \vdots & & & \vdots & \vdots & & & \\ 0 & \dots & & & \delta_n & \gamma_n & \beta_n & \alpha_n \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & U_{14} & V_{14} & W_{14} & & & & & 0 \\ & 1 & U_{12} & V_{12} & W_{12} & & & & \vdots \\ & & & & & & & & \vdots \\ & & & & & & & & \vdots \\ & & & & & & 1 & U_{n-2} & V_{n-2} \\ & & & & & & & 1 & U_{n-1} \\ & & & & & & & & 1 \end{bmatrix}$$

Con un desarrollo similar al anterior, se obtiene:

$$\begin{aligned} \delta_i &= a_i & , \quad i = 4, 5, \dots, n \\ \gamma_i &= 0 & , \quad i = 3 \\ \gamma_i &= -\delta_i u_{i-2} & , \quad i = 4, 5, \dots, n \\ \beta_i &= b_i & , \quad i = 2, 3 \\ \beta_i &= b_i - (\gamma_i u_{i-2}) - (\delta_i v_{i-2}) & , \quad i = 4, 5, \dots, n \\ \alpha_i &= c_i & , \quad i = 1 \\ \alpha_i &= c_i - (\beta_i u_{i-1}) & , \quad i = 2, 3 \\ \alpha_i &= c_i - (\beta_i u_{i-1}) - (\gamma_i v_{i-2}) - (\delta_i w_{i-2}) & , \quad i = 4, 5, \dots, n \\ u_i &= \frac{d_i}{\alpha_i} & , \quad i = 1 \\ u_i &= \frac{d_i - (\beta_i v_{i-1})}{\alpha_i} & , \quad i = 2, 3 \\ u_i &= \frac{d_i - (\beta_i v_{i-1}) - (\gamma_i w_{i-2})}{\alpha_i} & , \quad i = 4, 5, \dots, (n-1) \\ v_i &= 0 & , \quad i = 1 \\ v_i &= \frac{-\beta_i w_{i-1}}{\alpha_i} & , \quad i = 2, 3, \dots, (n-2) \\ w_i &= \frac{e_i}{\alpha_i} & , \quad i = 1, 2, \dots, (n-3) \end{aligned}$$

De igual manera, para los vectores  $\bar{y}$  y  $\bar{x}$ :

$$\begin{aligned} y_i &= \frac{f_i}{\alpha_i} & , \quad i = 1 \\ y_i &= \frac{f_i - (\beta_i y_{i-1})}{\alpha_i} & , \quad i = 2 \end{aligned}$$

$$y_i = \frac{\bar{r}_i - (\beta_i y_{i-1}) - (\gamma_i y_{i-2})}{\alpha_i}, \quad i = 3$$

$$y_i = \frac{\bar{r}_i - (\beta_i y_{i-1}) - (\gamma_i y_{i-2}) - (\delta_i y_{i-3})}{\alpha_i}, \quad i = 4, \dots, n$$

$$x_i = y_i, \quad i = n$$

$$x_i = y_i - (u_i x_{i+1}), \quad i = n-1$$

$$x_i = y_i - (u_i x_{i+1}) - (v_i x_{i+2}), \quad i = n-2$$

$$x_i = y_i - (u_i x_{i+1}) - (v_i x_{i+2}) - (w_i x_{i+3}), \quad i = n-3, n-4, \dots, 1$$

Esta técnica es equivalente al método de Eliminación Gaussiana, y con ella se evita el incremento del error asociado a la solución regresiva de las ecuaciones y se atenúan los requerimientos de memoria RAM. Además, se realizan cuando mucho  $5n$  multiplicaciones en comparación con  $n^3/3$  en el de Gauss y  $n!$  en la Regla de Cramer, para resolver un sistema  $(n, x, n)$ . En el *diskette* de consulta se presentan programas de cómputo para la solución de matrices tri y penta-diagonales, empleando el Algoritmo de Thomas y la técnica de Descomposición L-U, respectivamente (THOMAS.BAS y LU-PENTA.BAS).

#### II.1.4 SUBROUTINAS DECOMP Y SOLVE.

Existen sistemas de cómputo que incluyen en sus programas de biblioteca, programas para obtener la solución de sistemas de ecuaciones lineales, los cuales, generalmente, están basados en el método de Eliminación Gaussiana. Tal es el caso de las subrutinas DECOMP y SOLVE, que serán tratadas a continuación.

A DECOMP le corresponde la primera etapa de la eliminación de la matriz ampliada del sistema (dejar ceros abajo de la diagonal principal), mientras que SOLVE usa esos cálculos, realizando una

sustitución hacia atrás, para generar la solución del sistema. Estas subrutinas incluyen la técnica de *Pivoteo Parcial* (se sugiere consultar el problema resuelto PII-1, página 72), esto es, al k-ésimo paso de la primera etapa de la eliminación, se elige como pivote, al mayor elemento (en valor absoluto), contenido en la k-ésima columna no reducida. El renglón en que se ubica este pivote es intercambiado por el k-ésimo renglón, quedando el pivote en la posición (k, k). Similares intercambios se llevan a cabo en los elementos del vector de términos independientes.

Además, DECOMP estima un valor del número de condición (COND) de la matriz de coeficientes; si se detecta singularidad COND será igual a 1E+32, en caso contrario, será 1. Cualquier valor de COND entre 1 y 1E+32, indicará la posición relativa de la matriz con respecto a la no-singularidad.

Si se desea evaluar el determinante de la matriz de coeficientes, bastará con efectuar el producto de algunos elementos calculados por DECOMP, o sea:

$$\det(A) = \text{IPVT}(N) * A(1,1) * A(2,2) * \dots * A(N,N)$$

La inversa de la matriz de coeficientes (A), puede definirse como la matriz X, cuyas columnas  $\bar{x}_j$ , satisfacen:

$$A \bar{x}_j = \bar{e}_j \quad , j = 1, 2, \dots, n$$

Donde  $\bar{e}_j$  es la j-ésima columna de la matriz identidad (I). Por lo tanto, para obtener la matriz inversa X, es necesario invocar a DECOMP una sola vez y a SOLVE "n" ocasiones, una para cada columna de X. Si se detecta singularidad, X será incalculable<sup>(4)</sup>.

DECOMP y SOLVE son las subrutinas más eficientes para solucionar sistemas de ecuaciones lineales, pues no sólo ofrecen precisión en

los resultados, sino además, un tiempo de ejecución aceptable. En el *diskette* de consulta se presenta, en lenguaje *Basic*, un programa fuente para su control (DECOMP.BAS, ver apéndice).

### II.1.5 SOLUCION AL PROBLEMA DE FLUJO BIDIRECCIONAL, MONOFASICO, TRANSITORIO, A TRAVES DE MEDIOS POROSOS.

La simulación de un yacimiento se hace, en la mayoría de los casos, con el fin de estimar su comportamiento al ser sometido a diferentes programas de explotación. El objetivo de esta sección, es elaborar un simulador numérico sencillo. Por lo cual, se desarrollarán algunas técnicas para la solución de las ecuaciones involucradas en el modelo matemático que describe el flujo bidimensional, monofásico y transitorio a través de medios porosos.

La expresión que describe el flujo bidireccional, monofásico, en régimen transitorio (variable), de un fluido ligeramente compresible (agua o aceite) y de viscosidad y espesor constantes, que fluye en un medio poroso homogéneo y anisótropo, es:

$$k_x \frac{\partial^2 p}{\partial x^2} + k_y \frac{\partial^2 p}{\partial y^2} + \frac{119.56 \mu q}{h \Delta x \Delta y} = 119.56 \phi \mu c \frac{\partial p}{\partial t} \quad \dots (2.2)$$

donde:

- x = desplazamiento en la coord. x, [m].
- y = desplazamiento en la coord. y, [m].
- $k_x$  = permeabilidad en la direcc. x. [md].
- $k_y$  = permeabilidad en la direcc. y. [md].
- t = tiempo, [días].

$h$  = espesor del medio poroso, [m].  
 $p$  = presión, [kg/cm<sup>2</sup> abs].  
 $\mu$  = viscosidad, [cp].  
 $c$  = compresibilidad, [cm<sup>2</sup>/kg].  
 $q$  = gasto de fluido, a condiciones estándar, en función de espacio y tiempo (si es de producción se usa con signo negativo), [m<sup>3</sup>/día].

Además, las condiciones de frontera (CF) e iniciales (CI) son:

$$\begin{array}{l}
 \text{CF} \left\{ \begin{array}{l}
 \text{interna} \dots p(0,0,t) = p_0, \quad t \\
 \text{externa} \dots p(L,M,t) = p_1, \quad t
 \end{array} \right. \dots (2.3) \\
 \text{CI} \left\{ \begin{array}{l}
 p(x,y,0) = p(x,y) = \text{valor conocido}
 \end{array} \right.
 \end{array}$$

donde:

$L$  = longitud total en la direcc.  $x$ , [m].

$M$  = longitud total en la direcc.  $y$ , [m].

Para resolver la ecuación (2.2), se aplica el método de Diferencias Finitas (el cual es tratado en el capítulo V, sección V.2.1), teniéndose los esquemas Explícito, Implícito y de Crank-Nicolson, como alternativas de solución. Antes de abordar estas técnicas, se expondrán los conceptos de estabilidad y convergencia.

Los ciclos de cálculo necesarios para resolver un problema determinado, dependerán del tiempo de predicción, las características del yacimiento y de los fluidos, los gastos de inyección o de producción, el tamaño de las celdas de la malla y la precisión que se desee en los resultados.

Al disminuir los incrementos  $\Delta x$ ,  $\Delta y$  y  $\Delta t$ , se reduce el error de

truncamiento en la aproximación de las derivadas, pero, aumenta el tiempo de cálculo. Por otra parte, al emplear  $\Delta t$  grandes, además de incrementarse el error de truncamiento, se tiene la posibilidad de que se presente oscilación en los resultados, fallando el método. Por lo tanto, la  $\Delta t$  adecuada en cada problema se obtendrá haciendo un balance de los factores anotados con anterioridad.

### II.1.5.1 CONVERGENCIA.

Si en un punto  $(x_i, t_n)$  de un determinado esquema, se cumple que:

$$\lim_{\Delta x, \Delta t \rightarrow 0} |p_i^n - p(x_i, t_n)| \approx 0$$

$$i = 1, 2, \dots, L \quad , \quad n = 1, 2, \dots, t$$

el esquema es convergente<sup>(9)</sup>, es decir, tiende a la solución real. Gráficamente, esto se representa en la Figura 2.1.

### II.1.5.2 ESTABILIDAD.

Si en un esquema determinado, al mantener fijos los valores de los incrementos  $\Delta x$  y  $\Delta t$ , el error al aproximar la solución real no se amplifica mientras  $\Delta t \rightarrow \infty$ , se dice que es un esquema estable<sup>(9)</sup>. Matemáticamente, se tiene:

$$\lim_{t \rightarrow \infty} |p_i^n - p(x_i, t_n)| < 1$$

$$i = 1, 2, \dots, L \quad , \quad n = 1, 2, \dots, t$$



En otras palabras, si el valor calculado no varía considerablemente con respecto al real, se habla de un esquema estable. Esto se muestra en la Figura 2.2. Es importante apuntar que se tendrá estabilidad sólo si se presenta la convergencia.

### II.1.5.3 ESQUEMA EXPLICITO.

Partiendo de la ecuación (2.2), anteriormente expuesta, y considerando:

$$\beta_1 = \frac{119.56 \mu c}{h \Delta x \Delta y}$$

$$\beta_2 = 119.56 \phi \mu c$$

Se tiene:

$$k_x \frac{\partial^2 p}{\partial x^2} + k_y \frac{\partial^2 p}{\partial y^2} + \beta_1 = \beta_2 \frac{\partial p}{\partial t} \quad \dots (2.4)$$

Ahora, aproximando las derivadas parciales de la ecuación (2.4) por medio de Diferencias Finitas:

$$\frac{\partial p}{\partial t} = \frac{p_{i,j}^{n+1} - p_{i,j}^n}{\Delta t} \quad \dots (2.5)$$

$$\frac{\partial^2 p}{\partial x^2} = \frac{p_{i-1,j}^n - 2p_{i,j}^n + p_{i+1,j}^n}{(\Delta x)^2} \quad \dots (2.6)$$

$$\frac{\partial^2 p}{\partial y^2} = \frac{p_{i,j-1}^n - 2p_{i,j}^n + p_{i,j+1}^n}{(\Delta y)^2} \quad \dots (2.7)$$

**CONVERGENCIA**  
Para  $t=n$

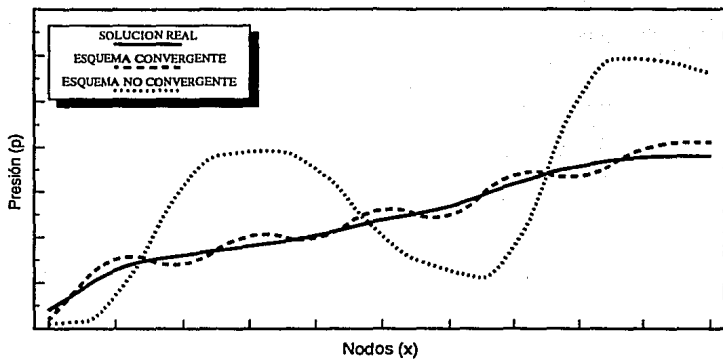


Figura 2.1.

**ESTABILIDAD**  
Nodo "1"

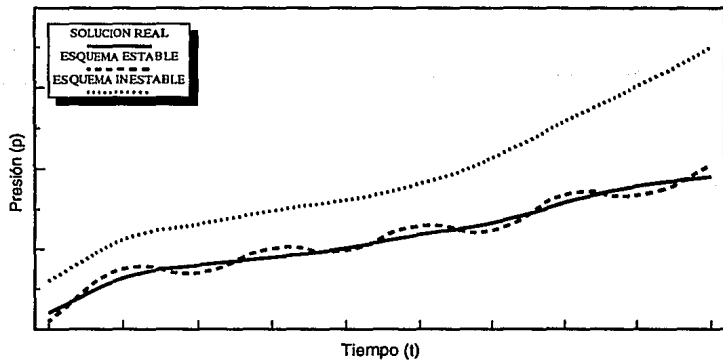


Figura 2.2.

En las relaciones anteriores, el subíndice "i" representa a la distancia o nodos en la dirección x (abscisa), "j" a los nodos en la dirección y (ordenada), y el superíndice "n", al tiempo.

Sustituyendo las ecuaciones (2.5), (2.6) y (2.7) en la (2.4), despejando  $p_{i,j}^{n+1}$  y factorizando, se llega a la siguiente expresión:

$$\begin{aligned}
 p_{i,j}^{n+1} = & \frac{\Delta t k_x}{\beta_x (\Delta x)^2} \left[ p_{i-1,j}^n + p_{i+1,j}^n \right] + \\
 & + \left[ 1 - 2 \frac{\Delta t}{\beta_x} \left[ \frac{k_x}{(\Delta x)^2} + \frac{k_y}{(\Delta y)^2} \right] \right] p_{i,j}^n + \\
 & + \frac{\Delta t k_y}{\beta_x (\Delta y)^2} \left[ p_{i,j-1}^n + p_{i,j+1}^n \right] + \frac{\Delta t \beta_x}{\beta_x} \dots (2.8)
 \end{aligned}$$

$$\begin{aligned}
 i &= 1, 2, \dots, L \\
 j &= 1, 2, \dots, M \\
 n &= 0, 1, \dots, t
 \end{aligned}$$

La ecuación (2.8) representa un esquema explícito, de sencilla solución, evaluado al tiempo "n" ( $t_n$ ). Este esquema es inusual en la práctica, puesto que es condicionalmente convergente y por ende, condicionalmente estable, se puede demostrar matemáticamente que la convergencia se cumplirá sólo para <sup>(23)</sup>:

$$\Delta t \leq \frac{1}{2[(\Delta x)^{-2} + (\Delta y)^{-2}]}$$

Por lo tanto, si la simulación de flujo a través de un medio poroso involucra grandes  $\Delta t$ , el riesgo de divergencia se hace patente.

Debido a que el empleo de Diferencias Finitas involucra una expansión en Serie de Taylor, se generan errores de truncamiento,

ya que no se consideran todos los términos de la serie. En la Figura 2.3 se presenta una porción de red que define la técnica explícita, en la cual, las cruces y los círculos indican los nodos implicados en la solución de las diferenciales de tiempo y espacio, respectivamente.

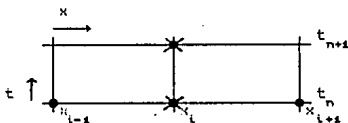


Figura 2.3. Nodos implicados en el esquema explícito, en la dirección  $x$ .

#### II.1.5.4 ESQUEMA IMPLICITO.

Este esquema, a diferencia del explícito, se evalúa al tiempo "n+1" ( $t_{n+1}$ ). Lo cual, puede ser apreciado en el segmento de malla de la Figura 2.4.

Conservando la nomenclatura y empleando Diferencias Finitas, se tiene entonces, para las derivadas parciales de la ecuación (2.4):

$$\frac{\partial p}{\partial t} = \frac{p_{i,j}^{n+1} - p_{i,j}^n}{\Delta t} \quad \dots (2.9)$$

$$\frac{\partial^2 p}{\partial x^2} = \frac{p_{i-1,j}^{n+1} - 2p_{i,j}^{n+1} + p_{i+1,j}^{n+1}}{(\Delta x)^2} \quad \dots (2.10)$$

$$\frac{\partial^2 p}{\partial y^2} = \frac{p_{i,j-1}^{n+1} - 2p_{i,j}^{n+1} + p_{i,j+1}^{n+1}}{(\Delta y)^2} \quad \dots (2.11)$$

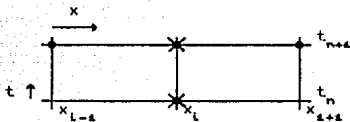


Figura 2.4. Nodos involucrados en el esquema implícito, en la dirección  $x$ .

Sustituyendo las expresiones (2.9), (2.10) y (2.11) en la (2.4), despejando  $p_{l,j}^n$  y factorizando, se encuentra la ecuación siguiente:

$$\begin{aligned}
 -\frac{\beta_x}{\Delta t} p_{l,j}^n &= \frac{k_x}{(\Delta x)^2} \left[ p_{l-1,j}^{n+1} + p_{l+1,j}^{n+1} \right] - \\
 &- \left[ 2 \left( \frac{k_x}{(\Delta x)^2} + \frac{k_y}{(\Delta y)^2} \right) \frac{\beta_x}{\Delta t} \right] p_{l,j}^{n+1} + \\
 &+ \frac{k_y}{(\Delta y)^2} \left[ p_{l,j-1}^{n+1} + p_{l,j+1}^{n+1} \right] + \beta_x \quad \dots (2.12)
 \end{aligned}$$

$$\begin{aligned}
 i &= 1, 2, \dots, L \\
 j &= 1, 2, \dots, M \\
 n &= 0, 1, \dots, t
 \end{aligned}$$

Suponiendo que:

$$\begin{aligned}
 \psi_1 &= \frac{k_x}{(\Delta x)^2} & \psi_2 &= \frac{k_y}{(\Delta y)^2} \\
 \psi_3 &= \left[ 2 \left( \frac{k_x}{(\Delta x)^2} + \frac{k_y}{(\Delta y)^2} \right) \frac{\beta_x}{\Delta t} \right]
 \end{aligned}$$

Entonces, se puede conformar un sistema de ecuaciones, cuya

matriz de coeficientes es penta-diagonal:

$$\begin{bmatrix}
 -\psi_3 & \psi_2 & 0 & \psi_1 & & & \dots & 0 \\
 \psi_2 & -\psi_3 & \psi_2 & 0 & \psi_1 & & & \vdots \\
 0 & \psi_2 & -\psi_3 & \psi_2 & 0 & \psi_1 & & \vdots \\
 \psi_1 & 0 & \psi_2 & -\psi_3 & \psi_2 & 0 & \psi_1 & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 0 & \dots & & & & & \psi_1 & 0 & \psi_2 & -\psi_3
 \end{bmatrix}
 \begin{bmatrix}
 p_{i,1}^{n+1} \\
 p_{i,2}^{n+1} \\
 \vdots \\
 p_{L,M-1}^{n+1} \\
 p_{L,M}^{n+1}
 \end{bmatrix}
 =
 \begin{bmatrix}
 \frac{-\beta_2}{\Delta t} p_{i,1}^n - \beta_1 \\
 \frac{-\beta_2}{\Delta t} p_{i,1}^n - \beta_1 \\
 \vdots \\
 \frac{-\beta_2}{\Delta t} p_{L,M-1}^n - \beta_1 \\
 \frac{-\beta_2}{\Delta t} p_{L,M}^n - \beta_1
 \end{bmatrix}$$

Como ya se mencionó en su oportunidad, la solución de una matriz penta-diagonal puede encontrarse mediante el uso del Algoritmo de Descomposición L-U. No se recomienda el uso de la subrutina DECOMP, puesto que se efectuarían operaciones innecesarias, debido a que se trata de una matriz rara (pocos elementos diferentes de cero).

El esquema implícito es incondicionalmente convergente y estable, no obstante, debe considerarse un rango sensato para el incremento de tiempo ( $\Delta t$ ), pues para valores elevados de él, se presentan errores de truncamiento.

#### II.1.5.5 ESQUEMA DE CRANK-NICOLSON.

El esquema de Crank-Nicolson calcula las diferenciales de las direcciones (x, y) al tiempo "n+1/2" ( $t_{n+(1/2)}$ ). Gráficamente se representa en la Figura 2.5. Es incondicionalmente estable y convergente, además, el ciclo de cálculo consta de dos etapas: una en la dirección x, empleando una ecuación implícita en x (barrido en x), y otra en la dirección y, recurriéndose a una expresión implícita sólo en y (barrido en y).

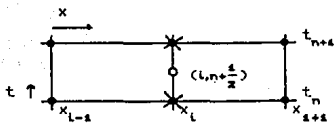


Figura 2.5. Nodos, en la dirección  $x$ , involucrados en el esquema de Crank-Nicolson.

Para el barrido en la dirección  $x$ , se tiene:

$$\frac{\partial p}{\partial t} = \frac{p_{i,j}^{n+1/2} - p_{i,j}^n}{\Delta t/2} \quad \dots (2.13)$$

$$\frac{\partial^2 p}{\partial x^2} = \frac{p_{i-1,j}^{n+1/2} - 2p_{i,j}^{n+1/2} + p_{i+1,j}^{n+1/2}}{(\Delta x)^2} \quad \dots (2.14)$$

$$\frac{\partial^2 p}{\partial y^2} = \frac{p_{i,j-1}^n - 2p_{i,j}^n + p_{i,j+1}^n}{(\Delta y)^2} \quad \dots (2.15)$$

Sustituyendo las ecuaciones (2.13), (2.14) y (2.15) en la (2.4), factorizando y despejando las  $p^{n+1/2}$ :

$$\begin{aligned} \frac{k_x}{(\Delta x)^2} [p_{i-1,j}^{n+1/2} + p_{i+1,j}^{n+1/2}] - \left[ 2 \left( \frac{k_x}{(\Delta x)^2} + \frac{\beta_2}{\Delta t} \right) \right] p_{i,j}^{n+1/2} + \beta_1 = \\ = - \frac{k_y}{(\Delta y)^2} [p_{i,j-1}^n + p_{i,j+1}^n] - \left[ 2 \left( \frac{k_y}{(\Delta y)^2} + \frac{\beta_2}{\Delta t} \right) \right] p_{i,j}^n \end{aligned} \quad \dots (2.16)$$

$$\begin{aligned} i &= 1, 2, \dots, L \\ j &= 1, 2, \dots, M \\ n &= 0, 1, \dots, t \end{aligned}$$

Ahora para el barrido en la dirección y:

$$\frac{\partial p}{\partial t} = \frac{p_{i,j}^{n+1} - p_{i,j}^{n+1/2}}{\Delta t/2} \quad \dots(2.17)$$

$$\frac{\partial^2 p}{\partial x^2} = \frac{p_{i-1,j}^{n+1/2} - 2p_{i,j}^{n+1/2} + p_{i+1,j}^{n+1/2}}{(\Delta x)^2} \quad \dots(2.18)$$

$$\frac{\partial^2 p}{\partial y^2} = \frac{p_{i,j-1}^{n+1} - 2p_{i,j}^{n+1} + p_{i,j+1}^{n+1}}{(\Delta y)^2} \quad \dots(2.19)$$

Sustituyendo las ecuaciones anteriores en la (2.4), factorizando y despejando las  $p^{n+1}$ :

$$\begin{aligned} \frac{k_y}{(\Delta y)^2} [p_{i,j-1}^{n+1} + p_{i,j+1}^{n+1}] - \left[ 2 \left[ \frac{k_y}{(\Delta y)^2} + \frac{\beta_2}{\Delta t} \right] \right] p_{i,j}^{n+1} + \beta_1 = \\ = - \frac{k_x}{(\Delta x)^2} [p_{i-1,j}^{n+1/2} + p_{i+1,j}^{n+1/2}] - \left[ 2 \left[ \frac{k_x}{(\Delta x)^2} + \frac{\beta_1}{\Delta t} \right] \right] p_{i,j}^{n+1/2} \end{aligned} \quad \dots(2.20)$$

$$\begin{aligned} i &= 1, 2, \dots, L \\ j &= 1, 2, \dots, M \\ n &= 0, 1, \dots, t \end{aligned}$$

Entonces, un intervalo de tiempo ( $\Delta t$ ), es decir, un ciclo de cálculo, se cubrirá al resolver la ecuación (2.16) primero y posteriormente la (2.20). Esto representa la solución de 2 sistemas de ecuaciones lineales, ambos con una matriz de coeficientes fija, por consiguiente, es recomendable emplear el Algoritmo de Descomposición L-U, en lugar de la subrutina DECOMP, ya que dicha subrutina realizaría operaciones innecesarias.



## II.1.5.6 SIMULACION NUMERICA DE FLUJO BIDIRECCIONAL, MONOFASICO, TRANSITORIO, A TRAVES DE MEDIOS POROSOS.

Para este fin, se empleará el Esquema Implícito, cuya solución se basa en la aplicación de la expresión (2.12) en todos los puntos (nodos), donde la presión es desconocida. Lo anterior conformará un sistema de ecuaciones como el expuesto en el subtema II.1.5.4, el cual se resuelve, dado el carácter de la matriz de coeficientes (consultar sección II.1.3), aplicando el Algoritmo de Descomposición L-U para una matriz pentadiagonal.

Suponga, un yacimiento como el ilustrado en la Figura 2.6. La malla que se utiliza es relativamente simple, 6 celdas en la dirección  $x$  y 6 en la  $y$ , esto se hace con el propósito de simplificar la ejecución del simulador. La frontera del yacimiento se aproxima por la línea perimetral de la misma figura. La localización de las celdas en la malla es en forma matricial, es decir, el índice " $i$ " para los renglones y " $j$ " para las columnas. Por lo tanto, el pozo indicado en el esquema tiene las coordenadas (3,3). Debe resaltarse que los pozos se suponen ubicados al centro de cada celda. A los pozos productores se les asigna, por convención, un gasto con signo negativo.

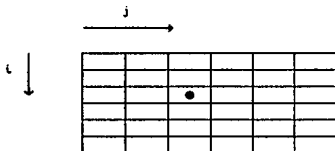


Figura 2.6. Ejemplo de malla de un yacimiento hipotético.

Para obtener la solución de la ecuación diferencial implicada,

es necesario introducir condiciones iniciales y de frontera, éstas son:

- a) Inicialmente el yacimiento tiene una presión uniforme.
- b) No hay flujo a través de su frontera.

Se puede contemplar el caso en el que la presión inicial no sea uniforme, y el gasto y la presión sean variables en los pozos. La condición dada en el inciso (b), se simula haciendo cero la permeabilidad (o en general la capacidad de flujo  $kh$ ) entre dos nodos, cuando por lo menos uno de ellos esté fuera del yacimiento.

El programa de cómputo que se expone, admite 4 pozos, sean éstos productores o inyectores. Un gasto nulo (cero) es indicativo de que el pozo no existe o está cerrado.

Se tendrán  $Nx$  ecuaciones con  $Ny$  incógnitas, siendo  $Nx$  el número de nodos en la dirección  $x$  y  $Ny$  el número de nodos en la  $y$ . Un ciclo de cálculo se completará cuando se solucione el sistema de ecuaciones para un intervalo de tiempo ( $\Delta t$ ), y la simulación llega a su fin cuando el número de ciclos cubra el tiempo de simulación supuesto.

En cada ciclo de cálculo se verifica que la presión de burbujeo ( $p_b$ ) no se halla alcanzado en aquellas celdas en las cuales se localice un pozo productor, en caso afirmativo, se simula el cierre del mismo haciendo el gasto correspondiente, igual a cero. Debe señalarse que abajo de la  $p_b$ , la simulación deja de ser válida, dado que la ecuación diferencial usada es para flujo monofásico.

El diagrama de flujo se muestra, de manera esquemática y condensada, en la página 53. El programa de cómputo está disponible en el disco flexible de consulta y tiene por nombre SIMUL.BAS (ver apéndice).

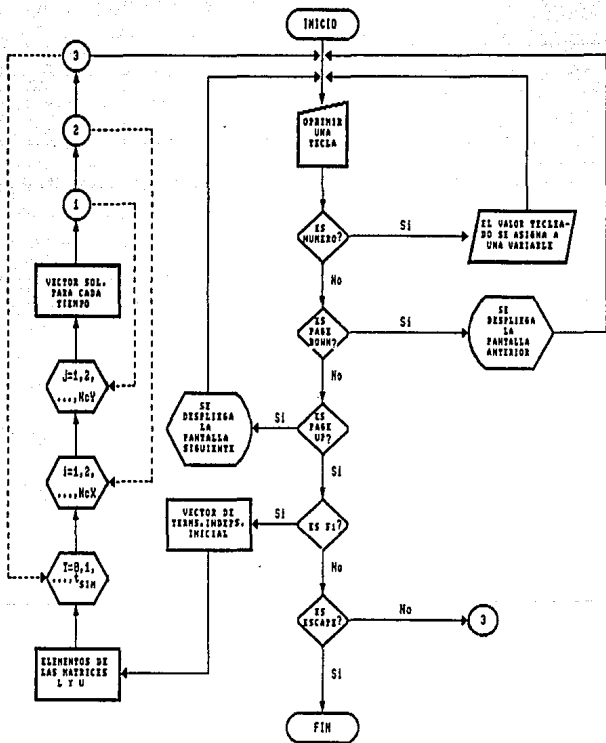


Figura 2.7. Diagrama de bloques del Simulador Bidireccional de Flujo en Medios Porosos.

### ENUNCIADO DEL PROBLEMA.

Los datos a considerar son los siguientes:  $k_x = k_y = 9$  [md],  $\mu = 1.2$  [cp],  $\phi = 0.25$ ,  $\Delta x = \Delta y = 200$  [m],  $\Delta t = 1$  [día],  $h = 47$  [m],  $p_i = 250$  [kg/cm<sup>2</sup> abs],  $p_o = 200$  [kg/cm<sup>2</sup> abs], y  $C = 0.00015$  [cm<sup>2</sup>/kg]. Además, se supone un pozo con un gasto de 55.3 [m<sup>3</sup>/día], cuyas coordenadas son (3,3). Por motivos prácticos, los datos se almacenan en un archivo de acceso directo (binario). Debe recordarse que el simulador se diseñó para 6 celdas en x y y.

Determinese el tiempo en que el medio poroso, aledaño al pozo productor, llegará a la  $p_o$ .

### DESCRIPCION DE RESULTADOS.

El tiempo que deberá transcurrir para que se alcance la  $p_o$ , contemplando un ritmo de extracción constante, es de 46 días. Este resultado fue obtenido en una computadora personal tipo AT, 100% compatible con IBM, en 2 minutos, para un tiempo de simulación de 50 días. La distribución de la presión a este tiempo, se indica en la Tabla 2.1. Es importante hacer notar que si el intervalo de tiempo ( $\Delta t$ ) se incrementa, disminuye la aproximación de la solución, y que a partir de un cierto valor de  $\Delta t$ , dependiendo del tamaño de las celdas de la malla, las características del yacimiento, etcétera, se presenta oscilación en los resultados.

227.96	222.67	220.21	220.04	221.06	222.56
217.53	212.29	209.29	210.07	212.13	216.02
212.69	206.69	206.66	204.92	206.06	212.66
211.18	206.17	203.29	204.26	206.73	211.29
210.62	206.16	204.08	204.30	206.37	210.79
210.73	206.13	204.21	204.25	206.17	210.52

Tabla 2.1. Distribución de la presión (kg/cm<sup>2</sup>) a 46 días de simulación.

## II.2 SISTEMAS DE ECUACIONES NO LINEALES.

Las ecuaciones no lineales, a diferencia de las lineales, no cumplen las siguientes **Condiciones de Linealidad**<sup>(2)</sup>:

Sean  $V$  y  $W$  dos espacios vectoriales sobre un campo  $K$ . Una transformación  $T:V \rightarrow W$  es lineal si para todo  $\vec{v}_1, \vec{v}_2 \in V$  y para todo escalar  $\alpha \in K$ , se verifica:

$$1.- T(\vec{v}_1 + \vec{v}_2) = T(\vec{v}_1) + T(\vec{v}_2) \quad (\text{PROPIEDAD DE SUPERPOSICION})$$

$$2.- T(\alpha \vec{v}_1) = \alpha T(\vec{v}_1) \quad (\text{PROPIEDAD DE HOMOGENEIDAD})$$

Los sistemas de ecuaciones no lineales están constituidos por funciones (circulares directas e inversas, logarítmicas, exponenciales, de potenciación y radicación), que involucran implícitamente a las variables o incógnitas, por lo cual, no es posible su solución mediante un "simple despeje" (solución directa), sino que deben emplearse métodos iterativos para su cálculo. Matemáticamente, tenemos:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ &\vdots \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \quad \dots (2.21)$$

El sistema anterior involucra "n" funciones reales, con "n" variables  $(x_1, x_2, \dots, x_n)$ , también reales, esto es:

$$\vec{x} = [x_1, x_2, \dots, x_n]^T$$

$$f_i(\vec{x}) = f_i(x_1, x_2, \dots, x_n) \quad , \quad i = 1, 2, \dots, n$$

Entonces, si la solución del sistema (2.21), es la siguiente:

$$\bar{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_n]^T$$

se cumple que:

$$f_i(\bar{\alpha}) = 0 \quad , \quad i = 1, 2, \dots, n$$

Por otro lado si consideramos las "n" funciones  $F(x)$ , de tal manera que:

$$x_i = F_i(x) \quad , \quad i = 1, 2, \dots, n \quad \dots(2.22)$$

Lo cual implica:

$$f_j(\bar{x}) = 0 \quad , \quad j = 1, 2, \dots, n$$

Tenemos, "n" ecuaciones (2.22) que constituyen un arreglo "más conveniente", para nuestros fines, que el sistema original (2.21). En particular sea:

$$\alpha_i = F_i(\bar{\alpha}) \quad , \quad i = 1, 2, \dots, n \quad \dots(2.23)$$

Tomando en cuenta, que una primera aproximación del vector  $\bar{\alpha}$ , es el vector inicial mostrado a continuación:

$$\bar{x}_0 = [x_{0_1}, x_{0_2}, \dots, x_{0_n}]^T$$

Las aproximaciones sucesivas pueden definirse entonces, como:

$$\bar{x}_k = [x_{k_1}, x_{k_2}, \dots, x_{k_n}]^T \quad , \quad k = 1, 2, \dots \quad \dots(2.24)$$

Cada elemento del vector (2.24) se obtiene a partir de la siguiente ecuación de recurrencia:

$$\begin{aligned} x_{k_i} &= F_i(x_{k-1_1}, x_{k-1_2}, \dots, x_{k-1_n}) \\ &= F_i(\bar{x}_{k-1}) \end{aligned} \quad \dots(2.25)$$

Suponga que existe una región  $\mathbb{R}$ , donde:

$$|x_j - \alpha_j| \leq \text{valor moderadamente grande}, \quad j = 1, 2, \dots, n$$

Y para todo  $\bar{x}$  en  $\mathbb{R}$ , existe un número positivo  $\mu < 1$ , tal que:

$$\sum_{j=1}^n \left| \frac{\partial F_i(\bar{x})}{\partial x_j} \right| \leq \mu$$

por lo tanto, si el vector inicial  $\bar{x}_0$  pertenece al conjunto  $\mathbb{R}$ , el método iterativo expresado por la ecuación (2.24), convergerá a la solución del sistema <sup>(2)</sup>, o sea:

$$\lim_{k \rightarrow \infty} \bar{x}_k = \bar{\alpha}$$

A diferencia de los métodos para sistemas de ecuaciones lineales, denominados "exactos", los concernientes a sistemas no lineales, acarrear errores de redondeo de un paso a otro. Esto, debido a que el resultado en cada iteración puede ser visualizado como una nueva suposición o como una aproximación al vector solución. Pueden permitirse errores sustanciales, con tal que no agiganten el error final o no sobrepasen la tolerancia establecida.

Aunque existen gran cantidad de métodos para la solución de sistemas de ecuaciones no lineales, se presentan a continuación sólo los más comunes: Jacobi, Gauss-Seidel y Newton-Raphson, siendo éste último la mejor alternativa.

## II.2.1 METODO ITERATIVO DE JACOBI.

Considérese un sistema no lineal cualquiera, como el sistema (2.21), en el cual, el vector solución es:

$$\bar{x} = [x_1, x_2, \dots, x_n]^T$$

y:

$$f_i(\bar{x}) = f_i(x_1, x_2, \dots, x_n) \quad , \quad i = 1, 2, \dots, n$$

Este método consiste en transformar el sistema original, en uno del tipo de la ecuación (2.22). Cuya solución se logrará mediante un ciclo iterativo, apoyándose en la ecuación de recurrencia (2.25), la cual es:

$$\begin{aligned} x_k^i &= F_i(x_{k-1}^1, x_{k-1}^2, \dots, x_{k-1}^n) \\ &= F_i(\bar{x}_{k-1}) \end{aligned}$$

Donde:

$$k = \text{número de iteración} \quad , \quad k = 1, 2, \dots$$

Además, se contempla una tolerancia y un vector inicial como el siguiente:

$$\bar{x}_0 = [x_0^1, x_0^2, \dots, x_0^n]^T$$

El grado de aproximación en la solución, casi siempre, puede ser mejorado al incrementar el número de iteraciones y depurar el proceso de cálculo en cada una de ellas. Esta técnica es de convergencia lenta y emplea memoria RAM para los elementos  $x_i^{k+1}$  y  $x_i^k$ , en cada iteración.



## II.2.2 METODO ITERATIVO DE GAUSS-SEIDEL.

Tomando en cuenta el sistema de ecuaciones (2.21), el cual tiene como vector solución a:

$$\bar{x} = [x_1, x_2, \dots, x_n]^T$$

y:

$$f_i(\bar{x}) = f_i(x_1, x_2, \dots, x_n) \quad , \quad i = 1, 2, \dots, n$$

Entonces:

$$x_i = F_i(\bar{x}) \quad , \quad i = 1, 2, \dots, n$$

La solución se basa en la utilización de la siguiente ecuación de recurrencia:

$$x_k^i = F_i(x_1^k, x_2^k, \dots, x_{i-1}^k, x_{i+1}^k, \dots, x_n^k) \quad \dots(2.26)$$

donde:

$$k = \text{número de iteración} \quad , \quad k = 1, 2, \dots$$

Se considera también, una tolerancia y un vector inicial, el cual puede ser:

$$\bar{x}_0 = [x_1^0, x_2^0, \dots, x_n^0]^T$$

Este método converge más rápido que el de Jacobi, ya que una vez calculado el componente  $x_{k+1}^i$ , lo aplica inmediatamente en la misma iteración.

### 11.2.3 METODO ITERATIVO DE NEWTON-RAPHSON.

El algoritmo para resolver un sistema de ecuaciones como el (2.21), usando el método de Newton-Raphson, puede resumirse así:

- 1) Elegir un vector inicial, esto es:

$$\bar{x}_k = \bar{x}_0 = [x_{1_0}, x_{2_0}, \dots, x_{n_0}]^T$$

- 2) Resolver el sistema de ecuaciones que se muestra a continuación:

$$\Phi(\bar{x}_k) \bar{\Delta}_k = -\bar{f}(\bar{x}_k) \quad \dots (2.27)$$

donde:

$$\phi_{ij}(\bar{x}_k) = \frac{\partial f_i}{\partial x_j}(\bar{x}_k) \quad \dots (2.28)$$

$$\bar{f}(\bar{x}_k) = [f_1(\bar{x}_k), f_2(\bar{x}_k), \dots, f_n(\bar{x}_k)]^T \quad \dots (2.29)$$

$$\bar{\Delta}_k = [\Delta x_k, \Delta z_k, \dots, \Delta n_k]^T \quad \dots (2.30)$$

$$i = 1, 2, \dots, n \quad ; \quad j = 1, 2, \dots, n$$

k = número de iteración (1, 2, ...)

El sistema anterior, como se observa, es lineal. El vector  $\bar{\Delta}_k$  es la solución del sistema (también es denominado vector de incrementos) y la matriz  $\Phi(\bar{x}_k)$  es el *Jacobiano*.

- 3) Calcular la aproximación de la solución del sistema de ecuaciones no lineales original (ecuación 2.21), o sea:

$$\bar{x}_{k+1} = \bar{x}_k + \bar{\Delta}_k \quad \dots (2.31)$$

## METODO DE NEWTON-RAPHSON

(Suponiendo un sistema de 3 ecuaciones con 3 incógnitas)

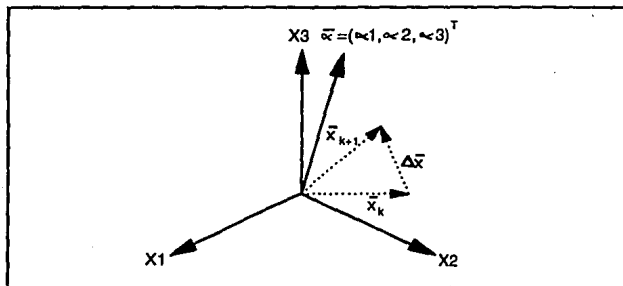


Figura 2.8.

- 4) Verificar la posible convergencia de la aproximación de la solución del sistema de ecuaciones no lineales. Se sugiere la siguiente prueba:

$$\left| x_{i,k+1} - x_{i,k} \right| \leq \text{tolerancia} \quad , \quad i = 1, 2, \dots, n$$

Si la expresión anterior se cumple para toda "i", entonces  $\bar{x}_{k+1}$  será la aproximación de la solución del sistema de ecuaciones no lineales ( $\bar{\alpha}$ ) (consultar Figura 2.8), finalizando el cálculo. Pero si la prueba falla para alguna "i", el proceso se repite a partir del paso 2. y el cálculo iterativo continuará hasta verificarse el criterio de tolerancia o hasta que se exceda algún límite de iteraciones fijado ( $k_{\text{máx}}$ ).

Si los componentes de  $\bar{f}(\bar{x})$  son continuos en la vecindad de un punto definido por el vector  $\bar{\alpha}$ , tal que:

$$\bar{f}(\bar{\alpha}) = 0$$

Y si el determinante de  $\bar{J}(\bar{\alpha}) \neq 0$  y  $\bar{x}_0$  está muy cercano a  $\bar{\alpha}$ , se asegura la convergencia del método<sup>(2)</sup>, es decir:

$$\lim_{k \rightarrow \infty} \bar{x}_k = \bar{\alpha}$$

Debido a que se usan los últimos valores  $x_{i-1}^{k+1}$  para calcular  $x_i^{k+1}$ , el requerimiento de memoria RAM es menor que en el método iterativo de Jacobi, además, su convergencia es mucho más rápida.

#### II.2.4 SOLUCION AL PROBLEMA DE DISEÑO DE UN SISTEMA DE RECOLECCION Y DISTRIBUCION DE GAS.

Es factible establecer un modelo para el flujo de gas en un sistema de recolección y/o distribución, cuando las condiciones de flujo en cada uno de sus elementos, pueden describirse en términos de relaciones matemáticas. También es posible desarrollar las ecuaciones analíticas necesarias para describir los efectos de interacción entre los componentes. El método más general y efectivo para analizar sistemas que manejan gas, es el propuesto por M. A. Stoner; en este subcapítulo se presentan los fundamentos del mismo y se discuten sus aplicaciones.

En la Figura 2.9, se muestra un sistema de recolección de gas natural con régimen de flujo permanente, comúnmente, éstos se constituyen por una red de tuberías y otros elementos, tales como: compresores, pozos, reguladores, válvulas, etcétera.

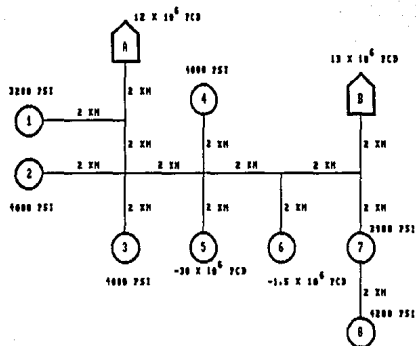


Figura 2.9. Campo "Las Margaritas", estudio de un diseño de recolección de gas.

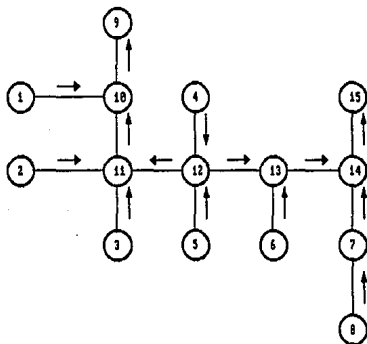


Figura 2.10. Representación por medio de nodos y conectores del sistema de recolección del campo "Las Margaritas". Las flechas indican el sentido de flujo a través de los conectores.

Cualquier red puede representarse como un sistema integral mediante *nodos* y *conectores de nodos*, como se ilustra en la Figura 2.10. Un nodo simboliza físicamente un punto de unión entre dos o más elementos, incluso, puede existir adición o extracción de masa al sistema a través de él. Los conectores, son elementos que permiten la transferencia de masa de un nodo a otro, por ejemplo, tuberías o válvulas.

Al conjunto de conectores se le denominará P, y N al de nodos. En un sistema como el descrito, la ley de conservación de masa en cada uno de los nodos debe satisfacerse, de manera analítica se tiene:

$$F_i = \sum_{j/(i,j) \in P} s_{ij} q_{ij} + Q_i = 0 \quad , \quad i \in N \quad \dots (2.32)$$

donde:

$s_{ij}$  = variable que indica el sentido de flujo, (+1) si el sentido de flujo es del nodo "i" al "j", y (-1) en caso contrario.

$q_{ij}$  = gasto de gas que fluye a través del conector de los nodos "i" y "j".

$Q_i$  = término que indica adición o extracción de masa al sistema a través del nodo "i".

Por ejemplo, considerando la Figura 2.10, la ecuación de conservación de masa para el nodo 1, es:

$$F_1 = s_{1,10} q_{1,10} + Q_1 = 0$$

Cada gasto  $q_{ij}$  puede expresarse en términos de la diferencia de presión en los extremos del conector, o sea:

$$q_{ij} = c_{ij} \left| p_i^2 - p_j^2 \right|^n \quad \dots (2.33)$$

donde:

$c_{ij}$  = coeficiente de transmisión de la tubería, el cual está en función de la geometría del tubo, de las condiciones de flujo y de la composición del gas.

$p_i$  = presión del fluido en el nodo "i".

$p_j$  = presión del fluido en el nodo "j".

$n$  = exponente que depende de la forma de la ecuación.

Al sustituir la ecuación (2.33) en la (2.32), se obtiene:

$$F_i = \sum_{j \in N, j \neq i} s_{ij} c_{ij} \left[ p_i^2 - p_j^2 \right]^n + Q_i = 0 \quad \dots (2.34)$$

$i \in N$

Además, si el sistema está balanceado se cumplirá que:

$$\sum_{i=1}^N Q_i = 0 \quad \dots (2.35)$$

De esta manera, el problema consiste en determinar un conjunto de valores  $Q_i$  y  $p_i$ , para toda "i"  $\in N$ , que satisfaga las ecuaciones (2.34) y (2.35). Sin embargo, se presentan N ecuaciones con 2N incógnitas (N valores de p más N valores de Q), por lo tanto, se requiere asignar valores a N variables (valores medidos), siendo las N restantes, las incógnitas del sistema. Dicha asignación, se hará de forma que las ecuaciones resultantes sean linealmente independientes, entonces, como la suma algebraica de los gastos exteriores debe ser igual a cero, solamente se puede suponer gastos en (N-1) nodos, quedando al menos uno como incógnita. Una posible forma para calcular la caída de presión en la tubería es mediante la ecuación de Panhandle<sup>(9)</sup>:

$$q_{ij} = 0.84267 E \left[ \frac{(p_i^2/z_i) - (p_j^2/z_j)}{0.6215 L_{ij}} \right]^{0.5} d_{ij}^{2.667}$$

donde:

$q_{ij}$  = gasto de gas a c.s., [MPCD].

$E$  = factor de eficiencia de la tubería, varía de 0.8 a 1.2.

$p_i, p_j$  = presiones en los nodos  $i$  y  $j$ , respectivamente, [lb/pg<sup>2</sup>].

$z_i, z_j$  = factores de desviación del gas a  $p_i$  y  $p_j$ , respectivamente.

$L_{ij}$  = longitud de la línea entre los nodos  $i$  y  $j$ , [km].

$d_{ij}$  = diámetro de la línea entre los nodos  $i$  y  $j$ , [pg].

De la expresión anterior y considerando la (2.33), se puede obtener el coeficiente de transmisión de la tubería ( $c_{ij}$ ), esto es:

$$c_{ij} = \frac{0.84269 E d_{ij}^{2.667}}{(0.6215 L_{ij})^{0.5}}$$

y para el sentido de flujo:

$$s_{ij} = \frac{p_i - p_j}{|p_i - p_j|}$$

Las ecuaciones que resultan de aplicar la ecuación (2.34) a cada uno de los nodos son no lineales. El método que se empleará para resolver dichas ecuaciones es el de Newton-Raphson. Para su aplicación deben asignarse valores iniciales a las incógnitas del sistema. Usando estos valores como base, el método proporciona un conjunto de correcciones que, sumadas a los valores anteriores, generarán que éstos tiendan a la solución. Este proceso conforma una iteración. Las iteraciones se continúan hasta que los valores obtenidos para las incógnitas satisfagan, dentro de cierta tolerancia, las ecuaciones (2.34).



Como ocurre con todo proceso iterativo, la convergencia del método está sujeta a la selección de los valores iniciales para las incógnitas. Entre mejor sea la selección, el proceso convergerá más aprisa. Para evitar la divergencia, Stoner<sup>(9)</sup> sugiere que en las dos primeras iteraciones, los valores de las correcciones se multipliquen por 1/2; aunque con esto, en las últimas etapas iterativas disminuye el ritmo de convergencia.

El simulador numérico desarrollado en este trabajo se estructuró en lenguaje *Basic* y permite manejar en forma gráfica el diseño de la red, mediante el desplazamiento del cursor en la pantalla del computador. El modo de gráficos seleccionado fue de 320 x 200 pixeles (*screen 1*), el cual facilita la ejecución del simulador en cualquier tipo de monitor que cuente con tarjeta controladora de gráficos, o en su defecto, debe emplearse un programa emulador (SIMCGA.EXE, EGC2UTIL.EXE, etcétera). El programa de cómputo se presenta en el *diskette* de consulta, bajo el nombre de REDGAS.BAS (ver apéndice).

Cabe mencionar que la matriz asociada al sistema de ecuaciones es "rala" y que su estructura no varía durante el proceso iterativo; no obstante, para fines didácticos se emplearán las subrutinas DECOMP y SOLVE, expuestas en la sección II.1.4, para resolver el sistema, por lo tanto, el tiempo de máquina se verá incrementado, aunque no de manera sustancial en problemas pequeños. En la página 68 se expone el diagrama de bloques simplificado (Figura 2.11).

#### DATOS DEL DISEÑO.

La prueba del simulador se hará empleando valores de una red de recolección de gas del campo Las Margaritas, Distrito Frontera Noreste, la cual, está conectada a 8 pozos y 17 tuberías de

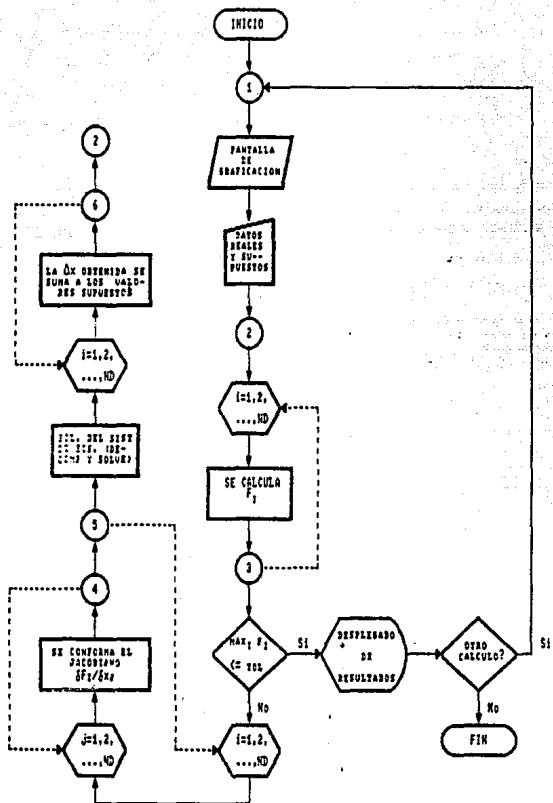


Figura 2.11. Diagrama de bloques del Simulador para Flujo de Gas en Régimen Permanente.

distribución (consultar Figura 2.9). Todos los diámetros interiores son de 2 pulgadas, excepto el de las líneas (9,10) y (14,15), que es de 4. Tomando como base el arreglo de la Figura 2.10, en el punto A deben entregarse 12 MMPCD de gas y en el B, 13 MMPCD. Se conoce la presión en la cabeza de 6 pozos y el gasto en los 2 restantes.

Es necesario determinar:

- a) A qué presión se entregará el gas en los puntos A y B.
- b) Cuál será el gasto  $Q_i$  y la presión  $p_i$  en cada nodo.
- c) El número de iteraciones en que se llega a la solución.

Los valores iniciales a considerar son (por convención a los gastos de producción se les asigna signo negativo):

$$\begin{aligned}Q_1 &= -2.9 \text{ [MMPCD]}, \\Q_2 &= -3.6 \text{ [MMPCD]}, \\Q_3 &= -3.6 \text{ [MMPCD]}, \\Q_4 &= -3.1 \text{ [MMPCD]}, \\Q_7 &= -1.2 \text{ [MMPCD]}, \\Q_8 &= -6.0 \text{ [MMPCD]}, \\p_1 &= 3990 \text{ [lb/pg}^2\text{]}, \\p_2 &= 3700 \text{ [lb/pg}^2\text{]}, \\p_7 &= 1000 \text{ [lb/pg}^2\text{]}, \\p_{10} &= 3100 \text{ [lb/pg}^2\text{]}, \\p_{11} &= 3880 \text{ [lb/pg}^2\text{]}, \\p_{12} &= 3900 \text{ [lb/pg}^2\text{]}, \\p_{13} &= 3750 \text{ [lb/pg}^2\text{]}, \\p_{14} &= 3400 \text{ [lb/pg}^2\text{]}, \\p_{15} &= 1050 \text{ [lb/pg}^2\text{]}.\end{aligned}$$

Todos los datos del sistema de recolección del presente análisis se incluyen en el programa de cómputo, así como su representación gráfica (red), a manera de ejemplo de aplicación.

## DISCUSION DE RESULTADOS.

Con fines de simplificación del problema, se supuso que el factor de desviación del gas ( $z$ ) es igual a 1 en todo instante de la simulación, y que el factor de eficiencia es igual a 0.8.

En base a experimentación, se determinó una exigua modificación al método de Stoner para acelerar la convergencia en redes de más de 5 nodos, ésta se basa en que en las tres primeras iteraciones se multiplique el vector de correcciones por 0.5. Con este criterio, se logró obtener la solución en 7 iteraciones y 69.64 segundos; considerando el método tradicional se ejecutan 12 iteraciones en 119.63 segundos, para alcanzar resultados análogos (empleando, en ambos casos, un computador personal tipo AT, 100% compatible con IBM). En la Tabla 2.2 se plasman estos resultados.

nodo	$p$ [lb/pg <sup>2</sup> ]	$Q$ [MPCD]
1	3200.00	-3002.10*
2	4000.00	-3605.55*
3	4000.00	-3605.55*
4	4000.00	-3131.66*
5	3993.15*	-3000.00
6	3769.29*	-1500.00
7	3900.00	-1168.66*
8	4200.00	-5986.49*
9	3063.79*	12000.00
10	3103.05*	0.00
11	3888.26*	0.00
12	3916.00*	0.00
13	3749.00*	0.00
14	3426.17*	0.00
15	3384.46*	13000.00

Tabla 2.2. Resultados del simulador para el campo Las Margaritas.

\* valores resultantes para las incógnitas

Debe indicarse que es preponderante elegir la ecuación correcta para evaluar las pérdidas de presión en tuberías, puesto que los resultados que proporciona el modelo están fincados totalmente en dicha selección. También, debe determinarse un factor de eficiencia que, unido a la ecuación, reproduzca las caídas de presión reales.

Puede observarse que simular sistemas de recolección y/o distribución de gas proporciona información esencial, como lo es: cuantificar el efecto que produciría cualquier cambio en el mismo, determinar su capacidad máxima, y estudiar el comportamiento del yacimiento, considerando la interacción entre los pozos.

## II.3 EJERCICIOS RESUELTOS.

PII-1. Sea el sistema:

$$\begin{bmatrix} 10 & -7 & 0 \\ -3 & 2.099 & 6 \\ 5 & -1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 3.901 \\ 6 \end{bmatrix}$$

Cuya solución exacta es  $[0, -1, 1]^T$ .

Empleando el método de Eliminación Gaussiana, investigue el efecto de los pivotes, sobre el error de precisión.

Suponga que se utiliza una computadora que maneja 5 cifras significativas.

Solución:

Conformando la matriz ampliada del sistema y eligiendo el valor del primer elemento de la ecuación 1 como pivote, se tiene:

1er. paso de la Eliminación Gaussiana:

- multiplicando la ec. 1 por 0.3 y sumándola a la ec. 2.
- multiplicando la ec. 1 por -0.5 y sumándola a la ec. 3.

$$\left[ \begin{array}{ccc|c} 10 & -7 & 0 & 7 \\ 0 & -0.001 & 6 & 6.001 \\ 0 & 2.5 & 5 & 2.5 \end{array} \right]$$

Ahora, seleccionando al segundo elemento de la ecuación 2 como pivote (recordemos que sólo se manejan 5 cifras significativas, por lo tanto, aquellas cantidades que rebasen esta restricción serán redondeadas):

2do. paso de la Eliminación Gaussiana:

- multiplicando la ec. 2 por  $2.5 \times 10^3$  y sumándola a la ec. 1, por lo que el nuevo renglón 1 es:

$$\left. \begin{aligned} 2.5 \times 10^3 (6) + 1.5000 \times 10^4 + 5 &= 1.5005 \times 10^4 \\ (6.001) 2.5 \times 10^3 + 1.5002 \times 10^4 + 2.33 &= 1.5006 \times 10^4 \end{aligned} \right\} \begin{array}{l} \text{matriz de} \\ \text{coeficientes} \\ \text{vector de} \\ \text{térms. indep.} \end{array}$$

$$\left[ \begin{array}{ccc|c} 10 & -7 & 0 & 7 \\ 0 & -0.001 & 6 & 6.001 \\ 0 & 0 & 1.5005 \times 10^4 & 1.5006 \times 10^4 \end{array} \right]$$

Entonces, por sustitución hacia atrás:

$$x_3 = \frac{1.5006 \times 10^4}{1.5005 \times 10^4} = 1.0001$$

$$-0.001 x_2 + 6 x_3 = 6.001$$

$$x_2 = \frac{6.001 - 6(1.0001)}{-0.001} = -0.4$$

$$10 x_1 - 7 x_2 = 7$$

$$x_1 = \frac{7 + 7(-0.4)}{10} = 0.42$$

La solución obtenida es  $[0.42, -0.4, 1.0001]^T$ , la cual difiere en extremo de la solución exacta  $[0, -1, 1]^T$ . Si no existe error de redondeo acumulado, causado por la ejecución de miles de operaciones, y la matriz es no singular, ¿cuál es el problema? La dificultad se presenta al elegir un pivote de valor pequeño en el segundo paso de la eliminación. Esto introdujo el uso de un multiplicador de  $2.5 \times 10^3$ , por lo que, la ecuación final involucra coeficientes que son  $10^3$  veces mayores que los originalmente planteados. Se deja al lector la verificación de que al intercambiar el segundo renglón por el primero, no existirán multiplicadores grandes, y por ende, el resultado será exacto.

Se puede concluir que si se seleccionan pivotes con valores absolutos grandes, que impliquen multiplicadores con valor absoluto menor o igual 1, se obtendrán resultados satisfactorios. A este criterio se le denomina *Pivoteo Parcial*.

PII-2. Elabore un programa de cómputo para obtener el determinante y la inversa de una matriz "cuadrada" cualquiera. Emplee las subrutinas DECOMP y SOLVE. Considere la siguiente matriz, para evaluar el funcionamiento del programa:

$$A = \begin{bmatrix} 2 & -7 & 4 \\ 1 & 9 & -6 \\ -3 & 8 & 5 \end{bmatrix}$$

$$A^{-1} = \begin{bmatrix} \frac{93}{295} & \frac{67}{295} & \frac{6}{295} \\ \frac{19}{295} & \frac{22}{295} & \frac{16}{295} \\ \frac{7}{47} & \frac{1}{47} & \frac{5}{47} \end{bmatrix}$$

Solución:

Mediante la subrutina DECOMP se puede obtener el determinante de una matriz (A), para ello, se recurre a la siguiente relación, la cual involucra elementos calculados por la subrutina mencionada:

$$\det(A) = \text{IPVT}(N) * A(1,1) * A(2,2) * \dots * A(N,N)$$

Por otro lado, la inversa de la matriz de coeficientes (A), puede definirse como la matriz X, cuyas columnas  $\bar{x}_j$ , satisfacen:

$$A \bar{x}_j = \bar{e}_j, \quad j = 1, 2, \dots, n$$

donde:

$\bar{e}_j$  = es la j-ésima columna de la matriz identidad (I)



Entonces, la matriz inversa (X), se obtendrá invocando a DECOMP en una sola ocasión y a SOLVE "n" veces, una para cada columna de X. Si se detecta singularidad, X será indeterminable<sup>(1)</sup>. El programa de cómputo se presenta en el *diskette* de consulta, y se intitula INVERSA.BAS. (ver apéndice). Los resultados arrojados por él, son los siguientes:

$$X = \begin{bmatrix} 0.3957 & 0.2851 & 0.0255 \\ 0.0553 & 0.0936 & 0.0681 \\ 0.1489 & 0.0213 & 0.1064 \end{bmatrix} ; \quad \det (A) = 235$$

Como puede observarse, existe coincidencia total entre los elementos de la matriz inversa calculada (X) y la dada como dato ( $A^{-1}$ ), debe apuntarse también, que se redondeo a cuatro decimales significativos el resultado mostrado.

#### II.4 PROBLEMAS PROPUESTOS.

PPII-1. Se requiere diseñar la sección de la torre de perforación que se muestra en la Figura 2.12. Por lo tanto, deben calcularse las fuerzas que actúan en cada barra, con el propósito de determinar el diámetro de éstas. En la parte superior, una fuerza de 20 toneladas simula el efecto de tensión, debido a la extracción o introducción de tubería. En la parte media, una fuerza de 5 toneladas representa el efecto originado por el apoyo de la tubería contra la estructura. En la parte inferior, una fuerza de 10 toneladas simula el peso de la estructura metálica.

Si las fuerzas horizontales se denotan como  $F_x$  y las verticales como  $F_y$ , además, considerando condiciones estáticas (equilibrio),

el problema puede representarse de la manera siguiente<sup>(3)</sup>:

$$\begin{aligned}
 \text{Junta 2} & \left\{ \begin{array}{l} \Sigma F_y = f_4 + rf_8 - 20 = 0 \\ \Sigma F_x = rf_8 + f_9 + rf_7 = 0 \\ \Sigma F_y = f_6 + rf_7 - rf_8 = 0 \end{array} \right. \\
 \text{Junta 3} & \left\{ \begin{array}{l} \Sigma F_x = f_9 = 0 \\ \Sigma F_y = f_8 - f_4 = 0 \end{array} \right. \\
 \text{Junta 4} & \left\{ \begin{array}{l} \Sigma F_x = f_9 = 0 \\ \Sigma F_y = f_{10} - f_6 = 0 \end{array} \right. \\
 \text{Junta 5} & \left\{ \begin{array}{l} \Sigma F_x = f_9 + rf_7 + rf_{11} - 5 = 0 \\ \Sigma F_y = f_{12} + rf_{11} - rf_7 - f_8 = 0 \end{array} \right. \\
 \text{Junta 6} & \left\{ \begin{array}{l} \Sigma F_x = f_{12} + rf_{11} + rf_{15} = 0 \\ \Sigma F_y = f_{14} + rf_{15} - rf_{11} - f_{10} = 0 \end{array} \right. \\
 \text{Junta 7} & \left\{ \begin{array}{l} \Sigma F_x = f_{12} = 0 \\ \Sigma F_y = f_{16} - f_{12} = 0 \end{array} \right. \\
 \text{Junta 8} & \left\{ \begin{array}{l} \Sigma F_x = rf_{18} - rf_{19} - f_{17} = 0 \\ \Sigma F_y = rf_{18} + rf_{19} - f_{14} = 0 \end{array} \right. \\
 \text{Junta 9} & \left\{ \begin{array}{l} \Sigma F_x = f_{17} + rf_{15} = 0 \\ \Sigma F_y = f_{20} - rf_{15} - f_{16} = 0 \end{array} \right. \\
 \text{Junta 10} & \left\{ \begin{array}{l} \Sigma F_x = f_{17} + rf_{15} = 0 \\ \Sigma F_y = f_{20} - rf_{15} - f_{16} = 0 \end{array} \right. \\
 \text{Junta 11} & \left\{ \begin{array}{l} \Sigma F_y = 10 - rf_{19} - f_{20} = 0 \end{array} \right.
 \end{aligned}$$

Realice un programa que involucre a las subrutinas DECOMP y SOLVE, para resolver el problema planteado.

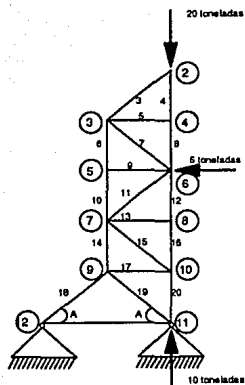


Figura 2.12. Diseño de torre de perforación.

PPII-2. La importancia de poseer una herramienta que permita simular el comportamiento de un sistema de conducción de fluidos en dos fases, con régimen estacionario, a través de tuberías, es incalculable, ya que con ella se pueden diseñar redes para la recolección y distribución de fluidos, así como optimizar los sistemas existentes. En base al simulador de flujo de gas en régimen permanente, elabore un programa de cómputo que simule el flujo bifásico en tuberías horizontales, empleando el método propuesto por Beggs y Brill (referencia: Brill, J. P. y Beggs, H. D., *Two phase flow in Pipes*, abril 1979).

## CAPITULO III

### INTERPOLACION Y APROXIMACION NUMERICA

En cualquier rama de la Ciencia, sea ésta: Ingeniería, Matemáticas Aplicadas, etcétera, es necesario lograr soluciones, prácticas y rápidas, de fenómenos o procesos físicos y químicos. En ocasiones, tales eventos se conocen sólo de manera tabulada o gráfica, es entonces, cuando la posibilidad de interpolar los datos o de generar una función de aproximación se torna primordial, pues con ello se obtiene una solución referida a los parámetros en estudio. Esta técnica evita el reproducir el proceso o fenómeno, con el consecuente ahorro de tiempo e inversión.

#### III.1 INTERPOLACION EXACTA.

Suponga que se cuenta con un conjunto de "n" pares de puntos (puntos base), definidos en el conjunto  $\mathbb{R}^2$ , tal que:

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

Los valores de las ordenadas ( $y_i$ ), son el resultado de alguna observación experimental o de alguna relación matemática y corresponden a las condiciones  $x_i$ . Si se desea conocer (interpolar) el valor de y para  $x \neq x_i$ , deberá crearse un función f, de tal manera que:

$$f(x_i) = y_i \quad , \quad i = 1, 2, \dots, n$$

## ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

Además, la citada función debe proporcionar valores *razonables* para  $f(x)$ ,  $x \neq x_i$ . El criterio para definir valor razonable varía de acuerdo al problema analizado y está supeditado a los requerimientos del analista. Si las ordenadas  $(y_i)$ , conforman una función matemática suave y poseen un cierto número de cifras decimales exactas, se esperará que la solución sea satisfactoria.

Si los puntos  $(x_i, y_i)$ , se derivan de observaciones experimentales muy precisas, de forma que pueden considerarse *libres* de error, es apropiado interpolarlos mediante una función suave, como la que ofrece la técnica de los Splines Cúbicos (que se expondrá posteriormente). Si por el contrario, provienen de experimentos relativamente burdos o imprecisos, es absurdo *obligar* a que la función interpoladora se ajuste exactamente a todos los puntos dato. En tal caso, será suficiente con ajustar una función global para lograr resultados aceptables.

El punto clave de la interpolación, es definir como debe comportarse una *función razonable* a través de los puntos dato. Debido a que los puntos base pueden interpolarse con un número infinito de funciones distintas, es necesario emitir algún criterio para elegir alguna de ellas. El criterio típico se sustenta en la suavidad y simplicidad de la función a seleccionar.

La suavidad de una función está dada por el valor máximo de  $|f''(x)|$ . Entre menor sea dicho valor, la suavidad irá en aumento. La simplicidad se refiere al grado de la función y también debe ser el mínimo valor posible.

Aunque existen diversas funciones de interpolación —monomiales, polinomiales, trigonométricas o de Fourier, exponenciales y racionales— se detallarán sólo las polinomiales, por ser las más empleadas (consultar sección III.2, Aproximación Funcional).

### III.1.1 INTERPOLACION POLINOMIAL.

Dadas las parejas de valores  $(x_i, f(x_i)=y_i)$ ,  $i = 0, 1, 2, \dots, n$  y el polinomio de interpolación :

$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

El criterio más evidente para determinar los coeficientes de  $p_n(x)$ , es el satisfacer la siguiente expresión:

$$p_n(x_i) = f(x_i) \quad , \quad i = 0, 1, 2, \dots, n \quad \dots(3.1)$$

Entonces, el polinomio  $p_n(x)$  de  $n$ -ésimo grado debe reproducir exactamente a  $f(x)$ , para los  $n+1$  argumentos  $x = x_i$ . Este razonamiento, basado en el teorema de Weierstrass, es especialmente importante, ya que existe un polinomio único de grado " $n$ " o menor, que cumpla tal aseveración<sup>(4)</sup>. Un polinomio con esas características es denominado *polinomio de interpolación de  $n$ -ésimo grado*.

Nótese, que la ecuación (3.1) establece que debe satisfacerse el valor de  $p_n(x)$ , para toda  $x_i$ , pero de ninguna manera garantiza la aproximación exacta de  $f(x)$  para  $x \neq x_i$ , es decir, para argumentos diferentes de los puntos base. En la Figura 3.1 se presenta de manera esquemática un polinomio interpolador para  $n = 3$ .

Aunque la mayoría de las fórmulas de interpolación que la literatura brinda, aparentemente sean disímbolas, aquellas que empleen como información idénticos puntos base y el criterio de la ecuación (3.1), para calcular los coeficientes  $a_0, a_1, \dots, a_n$ , deben ser básicamente iguales.

## INTERPOLACION POLINOMIAL

Para 4 pares de puntos dato

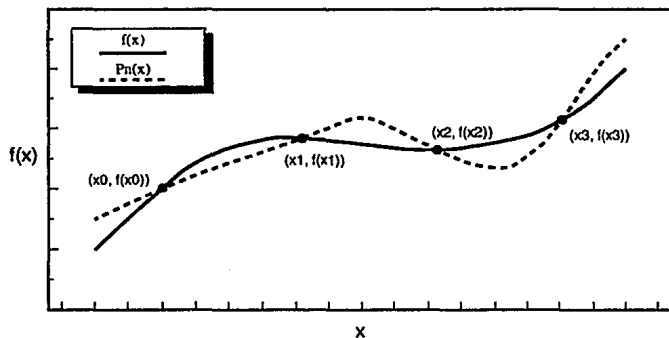


Figura 3.1.

La interpolación polinomial puede representarse por un sistema de ecuaciones lineales, o sea:

$$\begin{aligned} a_0 + a_1 x_0 + a_2 x_0^2 + \dots + a_n x_0^n &= f(x_0) \\ a_0 + a_1 x_1 + a_2 x_1^2 + \dots + a_n x_1^n &= f(x_1) \\ a_0 + a_1 x_2 + a_2 x_2^2 + \dots + a_n x_2^n &= f(x_2) \\ &\vdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \dots + a_n x_n^n &= f(x_n) \end{aligned}$$

El determinante de la matriz de coeficientes, el cual se conoce como determinante de Vandermonde, es:

$$\begin{vmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{vmatrix}$$

Se observa que para  $x \neq x_i$ ,  $i \neq j$ , este determinante es diferente de cero, por lo tanto, existe una solución única para cada  $a_i$ , es decir, habrá un polinomio único  $p_n(x)$ , que se ajuste con exactitud a  $f(x)$  en los puntos dato.

Ya que los coeficientes del polinomio elegido pueden obtenerse al resolver el sistema de ecuaciones inherente a él, surge la pregunta: ¿Para qué emplear fórmulas de interpolación? Existen dos razones fundamentales. La primera, es que dar solución a sistemas de ecuaciones lineales de cualquier tamaño, no es una tarea sencilla, sobre todo, cuando se *hace a mano*. La segunda, y quizá la más importante, es que la matriz de coeficientes es mal condicionada. Un ejemplo clásico, es el de los valores  $x_i$  igualmente espaciados en el intervalo  $[0, 1]$ , que al ser sustituidos en las funciones  $1, x, x^2, \dots, x^n$ , producen elementos positivos entre 0 y 1, generándose una estrecha dependencia entre las columnas o renglones de la matriz de coeficientes.



### III.1.2 INTERPOLACION POLINOMIAL DE LAGRANGE.

La ecuación de interpolación de Lagrange es la siguiente:

$$\begin{aligned}
 p_n(x) = & a_0 (x - x_1) (x - x_2) (x - x_3) \dots (x - x_n) + \\
 & + a_1 (x - x_0) (x - x_2) (x - x_3) \dots (x - x_n) + \\
 & + a_2 (x - x_0) (x - x_1) (x - x_3) \dots (x - x_n) + \\
 & \vdots \\
 & + a_l (x - x_0) (x - x_1) \dots (x - x_{l-1}) (x - x_{l+1}) \dots (x - x_n) + \\
 & \vdots \\
 & + a_{n-1} (x - x_0) (x - x_1) \dots (x - x_{n-2}) (x - x_n) + \\
 & + a_n (x - x_0) (x - x_1) \dots (x - x_{n-2}) (x - x_{n-1}) \dots (3.2)
 \end{aligned}$$

En la ecuación (3.2), los coeficientes  $a_0, a_1, \dots, a_n$ , serán determinados de tal forma que:  $p_n(x_i) = f(x_i)$ ,  $i = 0, 1, 2, \dots, n$ .

Por lo tanto:

$$a_l = \frac{f(x_l)}{(x_l - x_0) (x_l - x_1) \dots (x_l - x_{l-1}) (x_l - x_{l+1}) \dots (x_l - x_n)} \quad \dots(3.3)$$

La forma condensada de la expresión de Lagrange se obtiene al sustituir la ecuación (3.3) en la (3.2), o sea:

$$p_n(x) = \sum_{l=0}^n L_l(x) f(x_l) \quad \dots(3.4)$$

Donde:

$$L_l(x) = \prod_{\substack{j=0 \\ j \neq l}}^n \frac{(x - x_j)}{(x_l - x_j)} \quad , \quad i = 0, 1, \dots, n$$

En la expresión (3.4), cada valor de  $f(x_i)$  es multiplicado por  $L_i(x)$ , el cual es un polinomio de ajuste de  $n$ -ésimo grado, ya que está constituido por " $n$ " factores  $(x - x_j)$ .

Una desventaja de esta técnica, es la gran cantidad de operaciones que deben realizarse cuando se desean varias interpolaciones con el mismo conjunto de puntos dato (el número de operaciones y el tiempo de ejecución son proporcionales a  $n^2$ ). Otra desventaja, se presenta cuando se pretende incrementar en uno, el grado del polinomio (adición de un nuevo término), puesto que se requiere el cálculo completo de todos los valores  $L_i(x)$ . Por lo tanto, el polinomio de interpolación de Lagrange no es recomendable cuando el grado del polinomio no se conoce *a priori*.

### III.1.3 VALUACION DE POLINOMIOS.

En algunos casos, en un programa de cómputo se requiere evaluar reiteradamente un polinomio para cierto número de argumentos. Es entonces, que se hace vital la rápida valuación de dicho polinomio. Considérese el siguiente polinomio:

$$p(x) = a_1 x^n + a_2 x^{n-1} + \dots + a_{n+1}$$

Cuya valuación en lenguaje *Basic* es:

```
P = A(N + 1)
FOR I = 1 TO N
  P = P + A(I) * X ^ (N - I + 1)
NEXT I
```

Usualmente se emplean  $n(n+1)/2$  multiplicaciones y " $n$ " adiciones.

Una técnica simple para optimizar la valuación de un polinomio es la denominada *regla de Horner*, conocida también como *división o sustitución sintética*, que consiste en un re-arreglo del polinomio original, esto es:

$$p(x) = a_{n+1} + x (a_n + x (a_{n-1} + x ( \dots (a_2 + a_1 x) \dots )))$$

Que en lenguaje *Basic* se puede valorar como:

```
P = A(1)
FOR I = 1 TO N
    P = P * X + A(I + 1)
NEXT I
```

El algoritmo anterior realiza solamente "n" multiplicaciones y adiciones. Esta técnica enarbola el nombre de W. G. Horner, porque él la publicó en 1819, sin embargo, fue propuesta 100 años antes por Isaac Newton.

### III.1.4 SUBROUTINA SPLINE.

Antes de describir las *bondades* de la técnica SPLINE, se mostrará su ámbito teórico. Las funciones cúbicas *Spline* constituyen una técnica interpoladora reciente y se caracterizan por ser continuas, además, su primera y segunda derivadas también lo son.

Las funciones cúbicas Spline ajustan n-1 polinomios de tercer grado, es decir, un polinomio para cada uno de los n-1 intervalos definidos. Esta es la principal diferencia con respecto a las técnicas de interpolación tradicionales, en las cuales se ajusta un

polinomio único a los "n" pares de puntos base (Figuras 3.2 a y b).

La teoría de los Splines se expone a continuación:

Sean las abscisas  $(a = x_0) < x_1 < \dots < x_n = b)$  y las ordenadas  $[y_i]$ ,  $i = 0, 1, 2, \dots, n$ . Puede demostrarse<sup>(9)</sup> que de todas las funciones  $f(x)$  con segunda derivada continua en el intervalo  $[a, b]$ , tales que  $f(x_i) = y_i$ , la función cúbica Spline  $s(x)$ , con segunda derivada igual a cero en los extremos del citado intervalo,  $s''(a) = s''(b) = 0$ , minimiza la integral:

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

Esto es:

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

La función cúbica Spline con condición  $s''(a) = s''(b) = 0$ , es denominada *Spline natural* y posee la menor curvatura (mayor suavidad) de entre todas las funciones que pueden interpolar los puntos dato.

En los  $n-1$  intervalos, se tiene igual número de secciones separadas de curvas cúbicas, cada una con 4 parámetros, por lo que habrá  $4n-4$  elementos a determinar. El hecho de que  $s(x)$  sea continua, y que su primera y segunda derivadas también lo sean en los  $n-2$  nodos interiores  $x_i$ , adiciona  $3(n-2)$  condiciones. Como  $s(x_i) = y_i$  en cada uno de los nodos, se introducen "n" condiciones más, siendo  $4n-6$  el número total de parámetros a determinar. Para generar un sistema compatible, deben incluirse dos condiciones extra, que pueden estar dadas por las condiciones de frontera,  $s''(a) = s''(b) = 0$ .

Construir una función Spline es un proceso simple y

### FUNCION POLINOMICA UNICA DE GRADO $n-1$

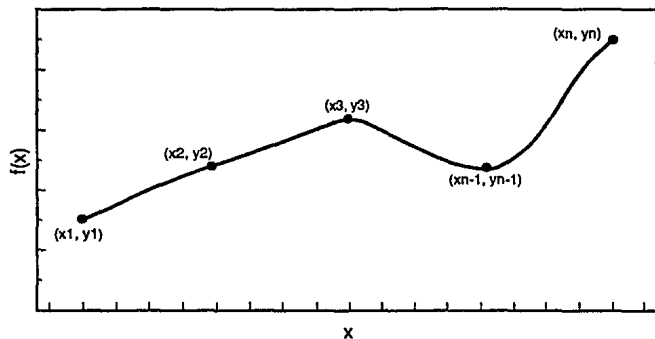


Figura 3.2a.

### $n-1$ FUNCIONES CUBICAS SPLINE AJUSTADAS EN $n-1$ INTERVALOS

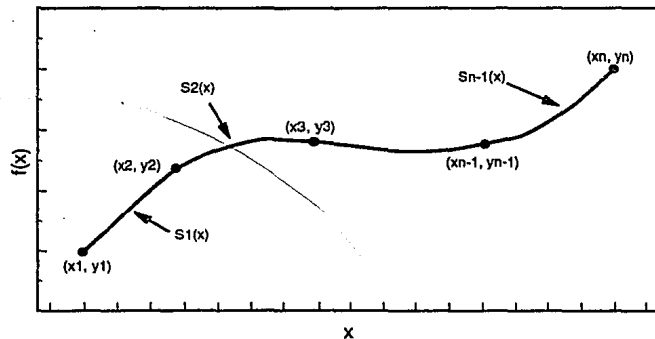


Figura 3.2b.

numéricamente estable. Supóngase el subintervalo  $(x_i, x_{i+1})$ , y que:

$$h_i = x_{i+1} - x_i, \quad w = \frac{x - x_i}{h_i}, \quad \bar{w} = 1 - w$$

Debido a que  $x$  fluctúa en tal subintervalo,  $w$  variará de 0 a 1 y  $\bar{w}$  de 1 a 0. Ahora, representando la función Spline 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:

$\sigma_i$  y  $\sigma_{i+1}$  son constantes por determinar.

Los dos primeros términos de la expresión anterior representan una interpolación lineal, mientras que los términos entre paréntesis rectangulares realizan una corrección cúbica, que generará la suavidad en la solución. Obsérvese que el término corrector desaparece en los puntos inicial y final del subintervalo, de manera que:

$$\begin{aligned} s(x_i) &= y_i \\ s(x_{i+1}) &= y_{i+1} \end{aligned}$$

Según esto, la función  $s(x)$  interpola exactamente los datos, sin importar cuales sean los valores de  $\sigma_i$ .

Considerando que:

$$w' = \frac{1}{h_i}, \quad \bar{w}' = -\frac{1}{h_i}$$

Además, diferenciando en tres ocasiones la función  $s(x)$  y utilizando la regla de la cadena, se obtiene:

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

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

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

Debe señalarse que  $s''(x)$  es una función lineal, la cual interpola entre los valores  $6\sigma_{i+1}$  y  $6\sigma_i$ . Consecuentemente:

$$\sigma_i = \frac{s''(x_i)}{6}$$

Lo que explica el significado de  $\sigma_i$ , pero, no determina su valor. Además,  $s'''(x)$  es constante en cada subintervalo y  $s''''(x) = 0$ , puesto que  $s(x)$  es una función cúbica. Si se evalúa  $s'(x)$  en los puntos extremo del subintervalo se tiene:

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

y:

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

donde:

$$\Delta_i = \frac{y_{i+1} - y_i}{h_i}$$

En las expresiones anteriores se emplea de manera temporal  $s'_+$  y  $s'_-$ , ya que la fórmula para  $s(x)$  se cumple sólo en el intervalo  $[x_i, x_{i+1}]$  y las derivadas en los puntos extremo no están claramente definidas. Con la finalidad de lograr la continuidad deseada en  $s'(x)$ , se fijan las siguientes condiciones en los puntos interiores:

$$s'_-(x_i) = s'_+(x_i) \quad , \quad i = 2, 3, \dots, n-1$$

Aún cuando  $s'_-(x_i)$  se calcule al considerar el subintervalo

$[x_{i-1}, x_i]$ , su fórmula puede obtenerse al reemplazar "i" por i-1 en  $s'(x_{i+1})$ , lo que origina:

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

Reordenado:

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

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

La expresión anterior genera un sistema de n-2 ecuaciones lineales con "n" incógnitas,  $\sigma_i$ ,  $i = 1, 2, \dots, n$ . Entonces, deben ser adicionadas dos condiciones más para lograr una solución única. Tal problema se resuelve considerando un Spline natural, puesto que  $s''(x_1) = s''(x_n) = 0$ , implicando que  $\sigma_1 = \sigma_n = 0$ .

Un Spline con estas condiciones frontera define un sistema de n-2 ecuaciones lineales y n-2 incógnitas, de forma que:

$$\begin{bmatrix} 2(h_1 + h_2) & h_2 & & & & \\ h_2 & 2(h_2 + h_3) & h_3 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ & & & & h_{n-1} & 2(h_{n-1} + h_n) \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_{n-2} \\ \sigma_{n-1} \end{bmatrix} = \begin{bmatrix} \Delta_2 - \Delta_1 \\ \Delta_3 - \Delta_2 \\ \vdots \\ \Delta_{n-2} - \Delta_{n-3} \\ \Delta_{n-1} - \Delta_{n-2} \end{bmatrix}$$



Este sistema puede resolverse mediante las subrutinas DECOMP y SOLVE, no obstante, la matriz de coeficientes presenta las siguientes características especiales:

- 1) Es tri-diagonal.
- 2) Es simétrica.
- 3) Para cualquier  $x_1 < x_2 < \dots < x_n$ , la matriz es no singular y generalmente está bien condicionada.

Por lo tanto, emplear el algoritmo de Thomas reduce el tiempo de ejecución y la cantidad de operaciones por realizar.

Si el Spline se va a evaluar muchas veces, es recomendable calcular y almacenar los coeficientes del Spline cúbico,  $b_i$ ,  $c_i$  y  $d_i$ ,  $i = 1, 2, \dots, n-1$ , para cada intervalo  $[x_i, x_{i+1}]$ , siendo:

$$s(x) = y_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

$$x_i \leq x \leq x_{i+1}$$

Dichos coeficientes, pueden obtenerse con las ecuaciones siguientes:

$$b_i = \frac{y_{i+1} - y_i}{h_i} - h_i(\sigma_{i+1} + 2\sigma_i)$$

$$c_i = 3\sigma_i$$

$$d_i = \frac{\sigma_{i+1} - \sigma_i}{h_i}$$

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

De esta manera se simplifican manipulaciones de  $s(x)$ , tales como derivaciones e integraciones.

La subrutina SPLINE tiene implementada la técnica descrita, para calcular los coeficientes de la función cúbica del mismo nombre. Los comentarios que aparecen en la rutina (chechar el *diskette* de consulta, archivo SPLINE.BAS), describen la forma de usarla. Una vez calculados los coeficientes por SPLINE, el subprograma SEVAL valga el Spline correspondiente al dato a interpolar. Tal subprograma, con formato de función (*function*), cuenta con una innovación (desarrollada en este trabajo), que permite elegir el tipo de búsqueda a ejecutar, sea ésta binaria o secuencial, cuando la variable independiente a interpolar no se encuentra en el mismo intervalo de la llamada previa, con la finalidad de ubicar el intervalo apropiado. Por último, es pertinente apuntar que los valores de la variable independiente (abscisas), deben proporcionarse en estricto orden ascendente ( $x_1 < x_2 < \dots < x_n$ ), para asegurar que los valores arrojados por SPLINE sean confiables y correctos.

#### III.1.4.1 INTERPOLACION SPLINE EN TRES DIMENSIONES.

Si se tienen "n" datos en el eje de las abscisas y "m" en el de las ordenadas, cada uno de ellos con su respectivo valor en el eje axial (Z), de tal forma que las coordenadas de cada punto sean:

$$(x_i, y_j, z_{ij})$$

$$i = 1, 2, \dots, n; j = 1, 2, \dots, m$$

Los (m x n) puntos base definen una región rectangular de interpolación en el plano X-Y. El problema será entonces definir el valor  $z^*$  en cualquier punto ( $x^*, y^*$ ). El procedimiento es el siguiente:

- a) Para cada  $y_j$  seleccionar las parejas de puntos ( $x_i, z_{ij}$ )

correspondientes y obtener las  $(n-1 \times m)$  funciones Spline  $s_{kj}(x)$ ,  $k = 1, 2, \dots, n-1$ , involucradas.

- b) Una primera aproximación  $z^{**}$  en el punto  $(x^*, y^*)$ , se logra al elegir aquél conjunto de funciones  $s_{kj}(x)$ ,  $j = 1, 2, \dots, m$ , que interpolen en el intervalo  $[x_k, x_{k+1}]$ , y el cual cumpla con  $x_k \leq x^* \leq x_{k+1}$ . Enseguida calcular:

$$z_{x^*j}^{**} = s_{kj}(x^*)$$

- c) Considerando la pareja de valores  $(y_j, z_{x^*j}^{**})$ , ajustar las funciones Spline respectivas y calcular en la función apropiada (intervalo correcto), el valor  $z^*$ . Análiticamente esto es:

$$z^* = s_l(y^*) \quad , \quad y_l \leq y^* \leq y_{l+1} \\ l = 1, 2, \dots, m-1$$

Este procedimiento arrojaría resultados idénticos para  $z^*$ , si en el inciso (a), en lugar de seleccionar las parejas  $(x_i, z_{ij})$ , se hubiesen elegido las  $(y_i, z_{ij})$ , para cada  $x_i$ .

### III.1.5 SOLUCION A LA INTERPOLACION DEL FACTOR DE COMPRESIBILIDAD DE UN GAS NATURAL.

Para tal efecto, se empleará el método de interpolación SPLINE en tres dimensiones. Entonces, considerando los parámetros involucrados, es decir, "n" datos de presión (abscisas), "m" de temperatura (ordenadas) y "n x m" del factor de compresibilidad del gas (cotas), se ejecutará el procedimiento siguiente:

- 1) Para cada  $T_j$ ,  $j = 1, 2, \dots, m$ , seleccionar el conjunto de valores  $(p_i, Z_{ij})$ ,  $i = 1, 2, \dots, n$ , y obtener las funciones Spline correspondientes, las cuales se valorarán en la abscisa de interés (presión a interpolar,  $P_{int}$ ). En lenguaje *Basic*, se tiene:

```

      :
FOR J = 1 TO M
  INPUT T(J)
  FOR I = 1 TO N
    INPUT P(I)
    INPUT Z(I)
  NEXT I
  CALL SPLINE (N, P(), Z(), B(), C(), D())
  ZAUX (J) = SEVAL (N, Pint, P(), Z(), B(), C(), D())
NEXT J
      :

```

- 2) Ahora, sólo restará invocar a *SPLINE* para el conjunto de valores  $(T_j, ZAUX_j)$ ,  $j = 1, 2, \dots, m$ . Debe señalarse que  $ZAUX_j$  es una primera aproximación del factor de compresibilidad y es el resultado del paso anterior. Finalmente, las nuevas funciones generadas se valorarán en la ordenada de interés (temperatura a interpolar,  $T_{int}$ ). En lenguaje *Basic*, esto puede escribirse como:

```

      :
CALL SPLINE (M, T(), ZAUX(), B(), C(), D())
Zint = SEVAL (M, Tint, T(), ZAUX(), B(), C(), D())
PRINT Zint
      :

```

El programa de cómputo que aquí se presenta (*INT3DIM.BAS*, consultar archivo *INDICE.BAT* del *diskette* anexo), se estructuró en

lenguaje *BASIC*, y permite, además de interpolar el factor de compresibilidad de un gas natural a partir de su presión y temperatura, crear una base de datos (archivos con extensión *DAT*) que contenga la información básica del citado gas, evitándose con esto, el capturar tal información cada vez que se requiera efectuar alguna interpolación. En la Figura 3.3, página 96, se muestra de manera simplificada el diagrama de bloques del programa expuesto.

#### DATOS BASICOS PARA LA INTERPOLACION.

Por un gasoducto fluye un gas natural que contiene 0.7% de nitrógeno. Fue analizado en laboratorio para obtener su factor de compresibilidad y así poder determinar su comportamiento a través de la red de distribución. Si las condiciones de transporte se modificaran, los datos del análisis dejarían de ser válidos, es entonces, cuando la ayuda de un procedimiento de interpolación se hace necesaria. En la Tabla 3.1 se muestran los resultados del análisis.

Presión (lb/pg <sup>2</sup> abs)	Factor de compresibilidad (Z)			
	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	1.0050	1.0384

Tabla 3.1. Datos del análisis de un gas natural.

Como puede observarse, la presión varía de 1.4 a 5.0 lb/pg<sup>2</sup>abs y la temperatura de 32 a 280°F, por lo que, los rangos de interpolación podrían ser de 0.5 a 6 lb/pg<sup>2</sup>abs para la presión y de 0 a 300°F para la temperatura. Esta consideración está contemplada en el programa de cómputo. También, debe aclararse que en dicho

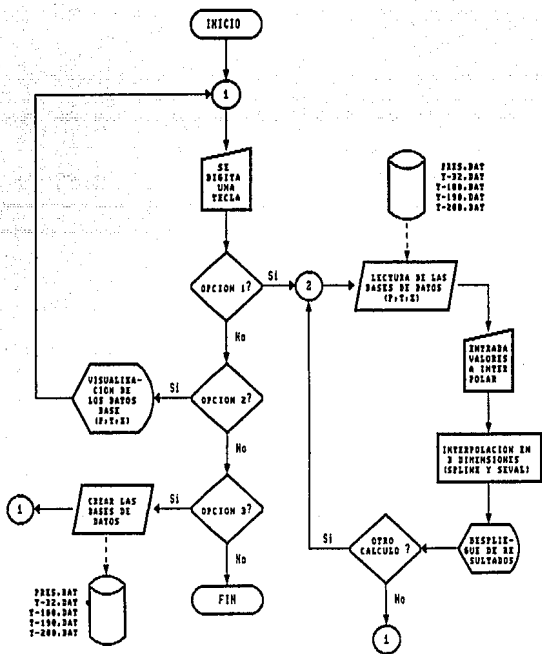


Figura 3.3. Diagrama de bloques del Programa para Interpolacion en 3 Dimensiones.

programa el número de bases de datos se restringe a 4 para la temperatura (con sus factores de compresibilidad correspondientes) y a uno para la presión.

#### DISCUSION DE RESULTADOS.

Los resultados obtenidos son aceptables, sin embargo, puesto que existe un rango muy extenso, mayor de 60°F, entre la información respectiva a cada temperatura, pueden presentarse resultados poco precisos. Al interpolar para valores alejados de los límites superior e inferior del intervalo de datos, se tienen resultados raros o ilógicos, esto es debido a que SPLINE permite interpolaciones satisfactorias, mas no extrapolaciones. Por lo tanto, mientras mayor continuidad posea la información proporcionada, más acertada será la interpolación ejecutada. Algunos resultados se muestran de manera tabular a continuación.

Presión (lb/pg <sup>2</sup> abs)	Factor de compresibilidad (Z)		
	50°F	150°F	250°F
1.0	0.8086	0.8961	0.9669
1.5	0.7173	0.8736	0.9379
2.5	0.6891	0.8436	0.9329
3.5	0.7681	0.8718	0.9558
4.5	0.8787	0.9398	1.0017
5.5	0.9844	1.0209	1.0697

Tabla 3.2. Resultados del simulador del factor de compresibilidad.

#### III.2 APROXIMACION FUNCIONAL.

En las secciones anteriores se establecieron métodos de interpolación para encontrar la ecuación de la curva que contiene a todos y cada uno de los "n" puntos base.

La curva continua de la Figura 3.4, representa la aproximación obtenida con algún método de interpolación. Ahora, se trata de encontrar la ecuación de una curva que, aunque no pase por todos los puntos, tenga pocas variaciones (sea suave, como la curva de trazo discontinuo de la Figura 3.4) y pase lo más cerca posible de todos ellos. Antes de aplicar cualquier técnica para tal problema, debe elegirse la forma de la curva que va a ser ajustada al conjunto de puntos dato. La ecuación de dicha curva puede obtenerse por conocimiento previo del problema, es decir, por la interpretación física del fenómeno, o en forma arbitraria, observando que ecuación conocida describe aproximadamente a esta curva.

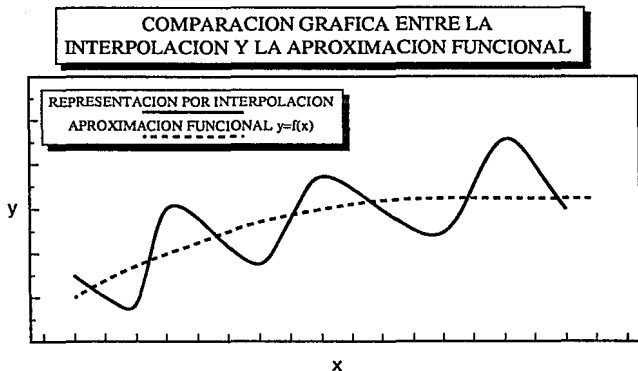


Figura 3.4.



Las funciones de aproximación  $g(x)$  más comunes, son aquellas que involucran combinaciones lineales de funciones base o simples,  $\{g_i(x)\}$ , y tienen la forma:

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

Las funciones base más usadas son las monomiales  $\{x^i\}$ ,  $i = 0, 1, 2, \dots, n$ , las de Fourier  $\{\sin kx, \cos kx\}$ ,  $k = 0, 1, 2, \dots, n$ , y las exponenciales  $\{e^{b_i x}\}$ ,  $i = 0, 1, 2, \dots, n$ .

Las combinaciones lineales de funciones monomiales desembocan en los polinomios de la forma:

$$f(x) \approx g(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

La combinación lineal de funciones de Fourier conforma las aproximaciones siguientes:

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

Las aproximaciones que emplean funciones exponenciales constituyen un modelo no lineal, o sea:

$$f(x) \approx g(x) = a_0 e^{b_0 x} + a_1 e^{b_1 x} + \dots + a_n e^{b_n x}$$

Las aproximaciones racionales, aunque se emplean en menor grado, son:

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

En las técnicas que se describirán a continuación, sólo se considerará el caso donde el número "n" de puntos dato, sea mayor o igual, que el número "m" de coeficientes de la función de aproximación (grado del polinomio).

### III.2.1 METODO DE LOS MINIMOS CUADRADOS.

Este método permite generar una función, que ajuste una curva suave a un conjunto de pares de puntos, tales que:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \quad , \quad x_1 < x_2 < \dots < x_n \\ i = 1, 2, \dots, n$$

Antes de mostrar las ecuaciones de esta técnica, debemos definir el concepto de *residuo*. Residuo es la diferencia de las ordenadas, entre la curva de ajuste y los datos base, para un valor de abscisa (x) dado (ver Figura 3.5). Representando como  $R_i$  a este residuo, analíticamente se tiene:

$$R_i = f(x_i) - y_i \quad , \quad i = 1, 2, \dots, n$$

El método de los Mínimos Cuadrados consiste, entonces, en determinar los valores de los coeficientes de una función de aproximación, de grado "m", de manera que se minimice la suma de los cuadrados de los residuos. La forma de tal función es:

$$y = f(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_m x^m$$

## RESIDUOS (Ri) EN UN AJUSTE DE CURVAS

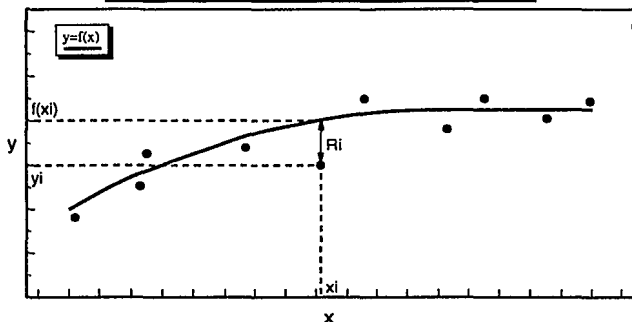


Figura 3.5.

El conjunto de ecuaciones a emplear, denominadas *normales*<sup>(40)</sup>, es:

$$\begin{aligned}
 na_0 + a_1 \sum_{l=1}^n x_l + a_2 \sum_{l=1}^n (x_l)^2 + \dots + a_m \sum_{l=1}^n (x_l)^m &= \sum_{l=1}^n y_l \\
 a_0 \sum_{l=1}^n x_l + a_1 \sum_{l=1}^n (x_l)^2 + a_2 \sum_{l=1}^n (x_l)^3 + \dots + a_m \sum_{l=1}^n (x_l)^{m+1} &= \sum_{l=1}^n x_l y_l \\
 a_0 \sum_{l=1}^n (x_l)^2 + a_1 \sum_{l=1}^n (x_l)^3 + \dots + a_m \sum_{l=1}^n (x_l)^{m+2} &= \sum_{l=1}^n (x_l)^2 y_l \\
 &\vdots \\
 a_0 \sum_{l=1}^n (x_l)^m + a_1 \sum_{l=1}^n (x_l)^{m+1} + \dots + a_m \sum_{l=1}^n (x_l)^{m+m} &= \sum_{l=1}^n (x_l)^m y_l
 \end{aligned}
 \tag{3.5}$$

En forma matricial este sistema puede ser escrito simplemente como:

$$P \bar{c} = \bar{q}$$

Donde:

$P$  = matriz de coeficientes del sistema (3.5)  
de orden  $(n \times n)$ .

$\bar{c}$  = vector de "m" coeficientes incógnita  $\{a_i\}$ ,  
del sistema (3.5).

$\bar{q}$  = vector de "n" términos independientes del  
sistema (3.5).

De primera mano, el sistema  $P \bar{c} = \bar{q}$ , puede solucionarse empleando las subrutinas DECOMP y SOLVE. Pero, dado que la matriz  $P$  es simétrica, la memoria requerida puede reducirse a la mitad, además,  $P$  es una matriz positiva definida, por lo cual no es necesario buscar un elemento pivote, ya que los elementos de la diagonal principal serán todos diferentes de cero.

Regularmente, la matriz  $P$  tiene un número de condición (COND), demasiado grande. Esto provoca que cualquier error cometido en los datos, por ínfimo que este sea, se traduzca en uno amplificado al calcular los coeficientes. De igual forma, cuando las funciones base  $\{q_i(x)\}$ , muestren dependencia, la matriz  $P$  será singular considerándose entonces, su número de condición como infinito.

Evidentemente, cualquier técnica que evite la generación de números de condición grandes en la matriz  $P$ , puede suponerse un buen detector de dependencia lineal entre las funciones base. En el siguiente subtema se describirá el método de la Descomposición del Valor Singular, que permite detectar y manejar el problema de la dependencia en las funciones base.

### III.2.2 DESCOMPOSICIÓN DEL VALOR SINGULAR (DVS) Y SUBROUTINA SVD.

El método de Descomposición del Valor Singular (DVS), calcula los coeficientes del problema de Mínimos Cuadrados y está basado en la factorización de la matriz de esos coeficientes.

La técnica inicia con la conformación de la *matriz de diseño* a partir de los datos base, o sea, una matriz  $A$  de orden  $(m \times n)$ , cuyos elementos están definidos por la siguiente expresión:

$$a_{ij} = g_i(x_j)$$

Si " $y$ " denota al vector de " $m$ " términos independientes  $\{y_i\}$ , y " $c$ " al de coeficientes, entonces la aproximación del modelo matemático lineal, está dada por:

$$\sum_{j=1}^n c_j g_j(x_i) \approx y_i \quad , \quad i = 1, 2, \dots, m$$

Y en forma matricial se tiene:

$$A \bar{c} \approx \bar{y}$$

El método DVS descompone a la matriz  $A$  en las matrices  $\Sigma$ ,  $U$  y  $V$ .  $\Sigma$  es una matriz diagonal de orden  $(m \times n)$  y sus elementos  $\{\sigma_i\}$  no-negativos, son los valores singulares de la matriz  $A$ . Las matrices  $U$ , de orden  $(m \times m)$ , y  $V$ , de  $(n \times n)$ , son ortogonales y unitarias, y se emplean en la transformación del sistema  $A \bar{c} \approx \bar{y}$ , en un sistema equivalente  $(\Sigma \bar{c} \approx \bar{y})$ . Por lo que, si  $A$  se expresa como  $U \Sigma V^t$ , se tendrá que:

$$U \Sigma V^t \bar{c} \approx \bar{y}$$

Dado el carácter ortogonal de  $U$  y  $V$  (esto es,  $U U^t = I$  y  $U^t = U^t$ ), la expresión anterior puede escribirse como:

$$\Sigma U^t \bar{c} \approx U^t \bar{y}$$

O también:

$$\Sigma \bar{c}' \approx \bar{y}'$$

Donde:

$$\begin{aligned} \bar{c}' &= U^t \bar{c} \\ \bar{y}' &= U^t \bar{y} \end{aligned}$$

Los valores singulares de la matriz  $A$ , están dados por las raíces cuadradas de los valores característicos de la matriz  $A A^t$ , que por cierto son iguales a los de la matriz  $A^t A$ . Los vectores singulares, izquierdo y derecho, son los vectores característicos de las matrices  $A A^t$  y  $A^t A$ , respectivamente, y constituyen las columnas de las matrices  $U$  y  $V$ .

Si las funciones base  $g_i(x)$  fueran linealmente independientes en el punto dato, entonces los valores singulares serían diferentes de cero.

El método DVS debe considerar una tolerancia  $\tau$ , la cual refleje la precisión de los datos originales. Cualquier valor singular  $\sigma_i$  mayor que dicha tolerancia, será aceptado y su correspondiente coeficiente  $c_j$  podrá evaluarse con la relación:

$$c_j = \frac{\bar{y}'_j}{\sigma_j}$$

Si algún valor singular  $\sigma_i$  es menor que  $\tau$ , se considerará nulo y su coeficiente asociado se igualará a cero. Una vez definidos los

valores singulares máximo y mínimo, puede efectuarse el cociente entre ambos para obtener, de manera alternativa, el número de condición de la matriz A, es decir:

$$\text{Cond}(A) = \frac{\sigma_{\text{máx}}}{\sigma_{\text{mín}}}$$

La técnica DVS se encuentra totalmente programada en la subrutina SVD (Singular Value Decomposition), la cual puede consultarse, junto con un programa fuente (SVD.BAS), en el disco flexible adjunto (ver apéndice). Las siguientes sentencias en lenguaje *Basic* permiten ilustrar el uso de la subrutina mencionada.

La matriz de diseño puede generarse con el siguiente grupo de instrucciones:

```
FOR I = 1 TO M
  T(I) = abscisa del i-ésimo punto base
  Y(I) = ordenada del i-ésimo punto base
  FOR J= 1 TO N
    A(I,J) = j-ésima función base evaluada en T(I)
  NEXT J,I
```

En el caso de aproximación polinomial, una manera eficiente de constituir a la matriz A, es:

```
FOR I = 1 TO M
  T(I) = ...
  Y(I) = ...
  A(I,1) = 1.0
  FOR J= 2 TO N
    A(I,J) = T(I) * A(I,J-1)
  NEXT J,I
```

A continuación, se debe incluir la llamada a SVD (leer los comentarios que aparecen en la subrutina para conocer los detalles

de su uso), esto es:

```
CALL SVD (NM, M, N, A, SIGMA, 1, U, 1, V, IERR, WORK)
IF IERR <> 0 THEN
  PRINT "ERROR DETECTADO POR SVD"
END
ENDIF
```

El siguiente segmento detecta el valor singular máximo y mínimo y ubica los valores singulares despreciables, además, se inicializa el vector de coeficientes:

```
SIGMAMAX = 0.0
SIGMAMIN = SIGMA(1)
FOR J = 1 TO N
  IF SIGMA(J) > SIGMAMAX THEN SIGMAMAX = SIGMA(J)
  IF SIGMA(J) < SIGMAMIN THEN SIGMAMIN = SIGMA(J)
  C(J) = 0.0
NEXT J
COND = SIGMAMAX / SIGMAMIN
```

Ahora, se fija un error relativo de tolerancia, RELERR. Si por ejemplo, los datos tienen una exactitud de 3 decimales, entonces RELERR =  $10^{-3}$ . Por lo tanto, la tolerancia ( $\tau$ ) en el error absoluto será:

```
TAU = RELERR * SIGMAMAX
```

El siguiente paso es obtener los coeficientes de la función de aproximación, en base a  $\sum U^i \bar{c} \approx U^i \bar{y}$ .

```
FOR J = 1 TO N
  IF SIGMA(J) <= TAU THEN GOTO 60
  S = 0.0
  FOR I = 1 TO M
    S = S + U(I,J) * Y(I)
  NEXT I
  S = S / SIGMA(J)
```



```

FOR I = 1 TO N
  C(I) = C(I) + S * V(I,J)
NEXT I
60 NEXT J

```

Nótese, que M es el número de elementos de Y y el de renglones en A y U. N es el número de elementos de C y el de renglones en V. Todas la matrices tienen N columnas.

Los coeficientes están listos para ser utilizados:

```

FOR I = 1 TO N
  PRINT "C(" ; I ; ") = " ; C(I)
NEXT I

```

Para evaluar el modelo en cualquier punto TT, se propone:

```

YY = 0.0
FOR J = 1 TO N
  YY = YY + C(J) * (j-ésima función base evaluada en TT)
NEXT J

```

Para modelos polinomiales, se recomienda emplear el esquema de Horner:

```

YY = 0.0
FOR JB = 1 TO N
  J = N + 1 - JB
  YY = TT * YY + C(J)
NEXT I

```

La raíz cuadrada de la suma del cuadrado de los residuos, es la cantidad que se está minimizando y puede calcularse con:

```

RSQ = 0.0
FOR I = 1 TO M
  RI = 0.0
  FOR J = 1 TO N

```

```

      RI = RI + C(J) * A(I,J)
    NEXT J
    RSD = RSD + (RI - Y(I))^2
  NEXT I
  R = SQR(RSD)

```

### III.2.3 AJUSTE DE FAMILIAS DE CURVAS\*

Considérese una función en tres dimensiones y con dos variables independientes,  $y = f(x, z)$ , la cual conforma una familia de curvas como las mostradas en la Figura 3.6. El procedimiento que se propone para ajustar un polinomio a esta familia de curvas, es el siguiente:

- 1) Ordenar de manera tabular los valores  $(x, y)$ , para  $z = z_i$ ,  $i = 1, 2, \dots, n$ .
- 2) Ajustar, mediante Mínimos Cuadrados, un polinomio a estos puntos, obteniéndose una expresión que de manera general tiene la forma:

$$y = a_{i,0} + a_{i,1}x + a_{i,2}x^2 + \dots + a_{i,m}x^m, \quad \text{para } z = z_i$$

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

Donde "n" representa el número de datos y "m" el grado del polinomio. Como es consabido, los coeficientes  $a_{i,0}$ ,  $a_{i,1}$ , ...,  $a_{i,m}$ , se determinan con el mismo ajuste polinomial.

- 3) El conjunto de ecuaciones del paso 2, puede reducirse a otro del tipo:

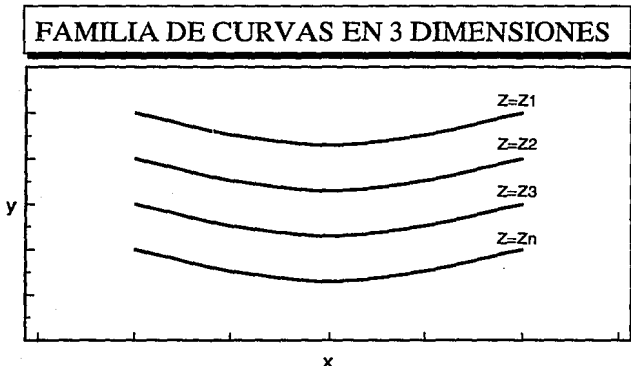
\* SEGUN NOTAS DEL M.C. J. TOMAS LIMON. H.

$$y = b_0 + b_1 x + b_2 x^2 + \dots + b_m x^m \quad \dots(3.6)$$

Donde los coeficientes  $b_l$ , son funciones de  $z$ , señalando que  $a_{l,0}, a_{l,1}, \dots, a_{l,m}$ , corresponden a  $z = z_l$ . Por lo tanto, los coeficientes de la ecuación (3.6), están definidos por las expresiones enseguida mostradas:

$$\begin{aligned} b_0 &= c_{0,0} + c_{0,1}z + c_{0,2}z^2 + \dots + c_{0,m}z^m \\ b_1 &= c_{1,0} + c_{1,1}z + c_{1,2}z^2 + \dots + c_{1,m}z^m \\ &\vdots \\ b_m &= c_{m,0} + c_{m,1}z + c_{m,2}z^2 + \dots + c_{m,m}z^m \end{aligned}$$

Los coeficientes  $c_{0,0}, c_{0,1}, c_{0,2}, \dots, c_{0,m}$ , de la ecuación para  $b_0$ , son el resultado de un ajuste polinomial en el que se han considerado las parejas de puntos  $(z_1, a_{1,0}), (z_2, a_{2,0})$ ,



x  
 Figura 3.6.

....  $(z_n, a_{n,0})$ . De manera análoga, los coeficientes  $c_{1,0}$ ,  $c_{1,1}$ ,  $c_{1,2}$ , ...,  $c_{1,m}$ , son el producto de un ajuste polinomial con los puntos  $(z_1, a_{1,1})$ ,  $(z_2, a_{2,1})$ , ...,  $(z_n, a_{n,1})$ . Y así sucesivamente, hasta calcular todos los coeficientes  $b_m$  restantes.

### III.2.4 INTRODUCCION AL METODO DE ESTIMACION DE KRIGING.

En la exploración y explotación petrolera, normalmente se tiene un conjunto finito de valores espacialmente distribuidos de la variable en estudio, a partir de los cuales, debe reconstruirse el fenómeno con la fidelidad y confiabilidad suficiente para la toma de decisiones que coadyuven al desarrollo de un yacimiento. En la Industria Petrolera la obtención de información es difícil y costosa, por tanto, es necesario cuantificar con más exactitud y veracidad las variables recabadas, utilizando nuevas y mejores técnicas para el procesamiento de datos.

La Geoestadística, que fue definida por Georges Matheron (1962) como "la aplicación de las funciones aleatorias al reconocimiento y estimación de fenómenos naturales", considera que las variables de dichos fenómenos son de carácter mixto, es decir están compuestas de dos partes, una estructural y otra estocástica. La Figura 3.7a muestra una gráfica de mediciones de porcentaje de mineral efectuadas a lo largo de cierta dirección en un yacimiento minero. Pueden apreciarse dos características: una local, de comportamiento errático o aleatorio, y otra general, con cualidades estructurales.

Describir fenómenos distribuidos espacialmente como lo son las formaciones geológicas (cimas y bases) o las propiedades físicas de una roca (porosidad, permeabilidad), permite evaluar las

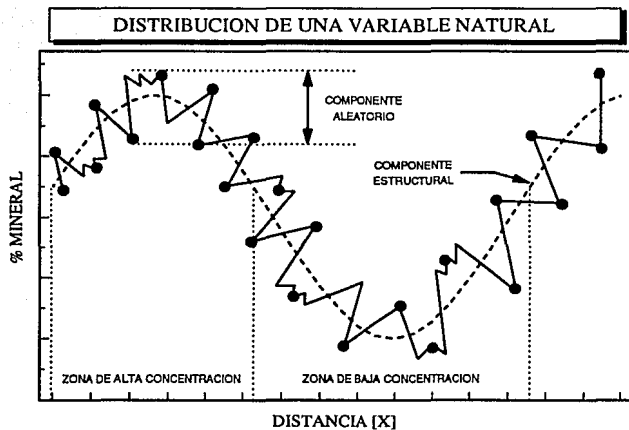


Figura 3.7a.

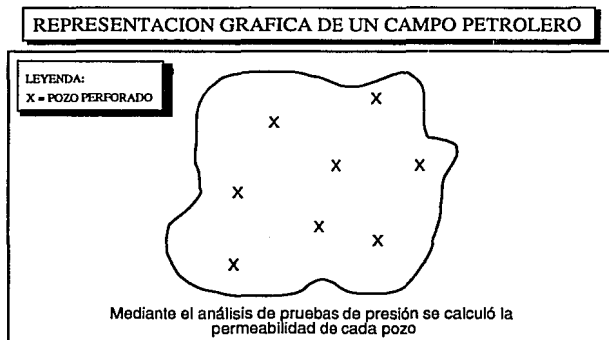


Figura 3.7b.

condiciones de saturación de agua, capacidad de flujo, índice de hidrocarburos, etcétera. Estas distribuciones pueden representarse por alguna función o modelo matemático, que resulta la mayoría de las veces tan complicado en su expresión como en su solución. Considérese que se desea obtener un plano configurado de las permeabilidades calculadas a través de pruebas de presión efectuadas en un cierto número de pozos (Figura 3.7b). 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 Spline o del método de Mínimos Cuadrados. Otra característica importante, es que la información se encuentra agrupada en ciertas porciones del Área en estudio, formando lo que se conoce como *nubes de información*. El ajuste de una superficie polinomial produciría, bajo estas circunstancias, resultados incoherentes debido a que la información más aislada estaría ejerciendo fuerte influencia sobre los coeficientes de los polinomios resultantes.

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, etcétera, 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 a todo proceso de interpolación. La Técnica de Estimación de Kriging (denominada así en honor del Doctor Daniel H. Krige de la escuela sudafricana), además de ser un método interpolador exacto y de minimizar el error, toma en cuenta las relaciones espaciales del fenómeno natural y permite la detección de rasgos característicos, tales como continuidad, variación en diferentes direcciones y la influencia de la variable alrededor de su vecindad. Contempla también, como ya se mencionó, que el valor de una variable es el resultado de 2 procesos, uno estructural y otro estocástico.

El Método de Kriging es una técnica de estimación local la cual proporciona el mejor estimador lineal de las características desconocidas del fenómeno en estudio. El objetivo de la estimación local es encontrar el estimador más eficaz del valor medio de una variable asociada a un dominio limitado, y de dimensiones menores a las dimensiones de la zona de cuasi-estacionaridad del fenómeno, es decir, aquella zona en la cual la función acumulativa de distribución permanece constante bajo efectos de traslación, en otras palabras, las distancias entre los puntos en estudio son menores o iguales a las de la zona considerada para propósitos de estimación.

La información requerida por este método consiste de un conjunto de datos (permeabilidad, porosidad, porcentaje de mineral, tiempos de reflexión, etcétera), e información estructural, es decir, el modelo del semivariograma que caracteriza la variabilidad en diferentes direcciones, de la zona estudiada. El semivariograma ( $\gamma$ ), define el grado de continuidad de la variable, su zona de influencia y su mayor o menor variación en distintas direcciones. De la forma de la gráfica de espaciamiento entre pares de datos ( $h$ ) vs. semivariograma  $\gamma(h)$ , se obtiene valiosa información acerca de una variable. Si se tienen puntos dispuestos regularmente a lo largo de una línea, el semivariograma puede calcularse, para incrementos de espaciamiento  $h$ , mediante la siguiente expresión:

$$\gamma(h) = 1/2 N \sum_{i=1}^N [Z(x_i+h) - Z(x_i)]^2$$

Donde  $Z(x_i)$  son los datos,  $x_i$  son las localizaciones tal que los datos estén disponibles en  $x_i$  y  $x_i+h$ , y  $N$  es el número de puntos.

La teoría desarrollada es como sigue: sea  $Z(x)$  una función aleatoria, con media  $E[Z(x)] = m$ , covariancia  $E\{Z(x+h) Z(x)\} - m^2 = C(h)$  y variograma  $E\{Z(x+h) - Z(x)\}^2 = 2\gamma(h)$ . El objetivo es estimar el valor medio de la variable  $Z_v(x_0)$  asociada al dominio

$V(x_{\alpha})$  con centro en el punto  $x_{\alpha}$ . Los datos experimentales pueden estar dados por el conjunto de valores  $\{Z_{v_{\alpha}}, \alpha=1, 2, \dots, n\}$ , donde cada valor  $Z_{v_{\alpha}}$  está definido sobre el soporte  $v_{\alpha}$  con centro en el punto  $x_{\alpha}$ . El valor  $Z_v(x_{\alpha})$  será estimado linealmente, a partir de los  $n$  datos experimentales, con el estimador  $Z_k^{\#}$ :

$$Z_k^{\#} = \sum_{\alpha=1}^n \lambda_{\alpha} Z_{v_{\alpha}}$$

Los  $n$  coeficientes  $\lambda_{\alpha}$  se calcularán asegurando que  $\sum \lambda_{\alpha} = 1$ , ya que así se garantiza que el valor esperado de  $Z_v$  sea igual al valor, también esperado, de  $Z_k^{\#}$ , es decir:

$$E[Z_k^{\#}] = E[\sum \lambda_{\alpha} Z_{v_{\alpha}}] = m \sum \lambda_{\alpha} = m = E[Z_v]$$

y la variancia de estimación  $\sigma_k^2$  se obtiene con:

$$\sigma_k^2 = \bar{C}(V, V) - 2 \sum_{\alpha=1}^n \lambda_{\alpha} \bar{C}(v_{\alpha}, V) + \sum_{\alpha=1}^n \sum_{\beta=1}^n \lambda_{\alpha} \lambda_{\beta} \bar{C}(v_{\alpha}, v_{\beta})$$

Aplicando el método de los Multiplicadores de Lagrange es posible encontrar el conjunto óptimo de coeficientes  $\lambda_{\alpha}$ , y por supuesto sujetos a la condición  $\sum \lambda_{\alpha} = 1$ . Al igualar a cero las  $n$  derivadas parciales:

$$\frac{\partial}{\partial \lambda_{\alpha}} \left[ \sigma_k^2 - 2\mu \sum_{\alpha=1}^n \lambda_{\alpha} \right] = 0$$

y al considerar la función restricción  $\sum \lambda_{\alpha} = 1$ , se define un sistema lineal de  $(n+1)$  ecuaciones y  $(n+1)$  incógnitas (los  $n$  coeficientes  $\lambda_{\alpha}$  más el multiplicador de Lagrange  $\mu$ ), el cual se denomina *Sistema Kríging*, esto es:

$$\sum_{\beta=1}^n \lambda_{\beta} \gamma(v_{\alpha}, v_{\beta}) + \mu = \bar{\gamma}(v_{\alpha}, V) \quad , \quad \alpha = 1, 2, \dots, n$$

$$\sum_{\alpha=1}^n \lambda_{\alpha} = 1$$



Una vez resuelto el sistema para los coeficientes  $\lambda_{\alpha}$ , la obtención de la variancia de estimación mínima  $\sigma_k^2$  es inmediata:

$$\sigma_k^2 = \sum_{\alpha=1}^n \lambda_{\alpha} \bar{\gamma}(v_{\alpha}, U) + \mu - \gamma(U, U)$$

El Sistema Kriging arrojará una solución única toda vez que la matriz de covariancia  $\bar{C}(v_{\alpha}, v_{\beta})$  sea una matriz definida positivamente. Dado que la técnica de Kriging minimiza el error del proceso de interpolación, permite la detección de rasgos característicos contemplando las relaciones espaciales del fenómeno natural en estudio, se le puede considerar como uno de los métodos de interpolación tridimensional más eficaces y confiables. A la fecha, se constituye como una herramienta de uso incipiente en el Área de Ingeniería de Yacimientos Petroleros, con aplicaciones específicas en la caracterización de yacimientos.

### III.2.5 CALCULO DE LOS COEFICIENTES DEL MODELO PARA LA OPTIMIZACION DE LA PERFORACION.

El objetivo de este proyecto, es obtener los coeficientes de la expresión que define el comportamiento de la velocidad de penetración de la barrena sobre la formación, a partir de información del proceso mismo de perforación, como lo es: la densidad equivalente de circulación, la profundidad de perforación, el gradiente de presión, el peso sobre la barrena, la velocidad de la mesa rotaria, el desgaste de los dientes de la barrena y el número de Reynolds.

El término de *optimización de la perforación* se aplica a procedimientos que permiten seleccionar la hidráulica a chorro, el

peso sobre la barrena, la velocidad de rotación, el tipo y las propiedades del lodo, el tipo de barrena, etcétera. El modelo que se adoptará es el propuesto por Bourgoyne y Young<sup>(41)</sup>, el cual sólo permite la optimización del peso sobre la barrena, la velocidad de perforación y la hidráulica a chorro. La ecuación que simula tal modelo de perforación, es:

$$\ln \left[ \frac{dD}{dt} \right] = a_1 + \sum_{i=2}^n a_i x_i$$

Donde:

$a_1, \dots, a_n$  = coeficientes del modelo.

$x_2 = 10,000 - D$  ; efecto de compactación normal.

$x_3 = D^{0.60} (q_p - 9.0)$  ; efecto de sobre-compactación.

$x_4 = D (q_p - p_e)$  ; efecto de la presión diferencial.

$x_5 = \ln \left[ \frac{\frac{W}{d} - \left(\frac{W}{d}\right)_t}{4.0 - \left(\frac{W}{d}\right)_t} \right]$  ; efecto del diámetro y del peso sobre la barrena.

$x_6 = \ln \left[ \frac{N}{100} \right]$  ; efecto de la velocidad de rotación.

$x_7 = -h$  ; efecto del desgaste de los dientes de la barrena.

$x_8 = \frac{\rho q}{350 \mu d_n}$  ; efecto hidráulico de la barrena (número de Reynolds).

$D$  = profundidad, [pie].

$g_p$  = gradiente de presión, [lb/gal].

$\rho_e$  = densidad equivalente de circulación, [lb/gal].

$d_n$  = diámetro de la tobera, [pg].

$\frac{dW}{dt}$  = peso sobre la barrena, [1000 lb/pg].

$\left(\frac{W}{d}\right)_t$  = peso inicial sobre la barrena, [1000 lb/pg].

$N$  = velocidad de la rotaria, [rpm].

$h$  = desgaste del diente de la barrena, [fracción].

$\rho$  = densidad del lodo, [lb/gal].

$q$  = velocidad de flujo, [gal/min].

$\mu$  = viscosidad aparente del lodo a 10000 [seg<sup>-1</sup>], [cp].

Para obtener el valor de los coeficientes del modelo es necesario emplear un procedimiento de aproximación funcional, por lo tanto, la técnica que determinará dichos coeficientes, será la de la Descomposición del Valor Singular (subrutina SVD), descrita en un subtema antecedente.

El programa desarrollado facilita la obtención de los 8 coeficientes del modelo, y además, la creación de archivos para la lectura/escritura de la información base (8 archivos con capacidad para 30 datos cada uno, extensión DTD). El lenguaje de programación adoptado es el *Basic* (*Quick Basic*).

El programa de cómputo se presenta en el *diskette* de consulta (OPTPERF.BAS, ver apéndice), y su diagrama simplificado de bloques, en la Figura 3.8.

#### DATOS DEL PROBLEMA PLANTEADO.

Teóricamente, sólo se requieren 8 datos para resolver el modelo de perforación pero, como la ecuación no simula al 100% las condiciones de la misma, debe usarse un número razonable, que se propone sea cuando menos 30. Los datos mencionados se tomaron del trabajo original de Bourgoyne y Young<sup>(14)</sup>, y pertenecen a una formación de las costas de Louisiana, E.U.A.; ellos se consignan en las Tablas 3.1 a y b (el peso inicial sobre la barrena es de 0.5 [1000 lb/pg]).

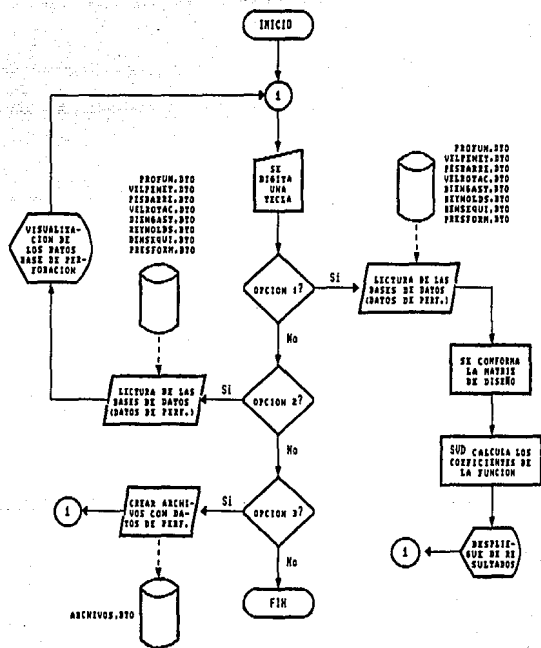


Figura 3.0. Diagrama de bloques del Programa para Obtener los Coeficientes del Modelo para la Optimización de la Perforación.

DISCUSION DE RESULTADOS.

Los coeficientes obtenidos al ejecutar el programa propuesto, son los siguientes:

$$a_1 = 6.0566E-08$$

$$a_2 = 2.5285E-04$$

NO. DATO	PROF. (pie)	VEL. DE PENET. (pie/hr)	PKSO SOBRE LA BARRERA (1000lb/pg)	VEL. DE ROTACION (rpm)	DESG. DEL DIENTE (frac)
1	9815.0	23.0	2.08	118.0	0.77
2	9820.0	23.0	1.15	126.0	0.38
3	10120.0	14.0	0.81	120.0	0.74
4	10250.0	10.0	0.95	87.0	0.15
5	10290.0	16.0	1.02	78.0	0.24
6	10500.0	19.0	1.60	81.0	0.61
7	10575.0	13.0	1.56	81.0	0.78
8	10840.0	16.6	1.63	67.0	0.38
9	10960.0	15.9	1.33	65.0	0.57
10	11060.0	15.7	2.08	60.0	0.72
11	11475.0	14.0	1.60	77.0	0.20
12	11775.0	13.5	2.21	58.0	0.12
13	11940.0	6.2	2.26	67.0	0.20
14	12070.0	9.6	2.07	84.0	0.08
15	12315.0	15.5	3.11	60.0	0.40
16	12900.0	31.4	2.82	85.0	0.42
17	12975.0	42.7	3.48	77.0	0.17
18	13055.0	38.6	3.20	75.0	0.30
19	13250.0	43.4	2.82	76.0	0.43
20	13705.0	12.5	1.60	81.0	0.56
21	14010.0	21.1	1.04	75.0	0.46
22	14445.0	19.0	1.76	64.0	0.16
23	14695.0	18.7	2.00	76.0	0.27
24	14905.0	20.2	2.35	75.0	0.33
25	15350.0	27.1	2.12	85.0	0.31
26	15740.0	14.8	2.35	78.0	0.31
27	16155.0	12.6	2.47	80.0	0.12
28	16325.0	14.9	3.76	81.0	0.50
29	17060.0	13.8	3.76	65.0	0.01
30	20265.0	9.0	3.41	60.0	0.01

Tabla 3.3a. Datos de perforación.

$a_1 = 5.3631E-04$   
 $a_4 = -9.8133E-05$   
 $a_5 = -3.0509E-08$   
 $a_6 = 2.0383E-08$   
 $a_7 = -2.9540E-08$   
 $a_8 = 2.2855E-08$

La exactitud de los resultados depende estrechamente de la

No. DATO	No. DE REYNOLDS	DENSIDAD EQUIV. (lb/gal)	GRADIENTE DE PRESION (lb/gal)
1	0.064	9.5	9.0
2	0.064	9.5	9.0
3	0.227	9.6	9.0
4	0.076	9.7	9.0
5	0.084	9.7	9.0
6	0.084	9.7	9.1
7	0.084	9.7	9.2
8	0.022	9.8	9.2
9	0.878	9.8	9.4
10	0.878	9.8	9.5
11	0.887	10.2	9.5
12	0.852	11.8	10.1
13	0.976	15.2	12.4
14	0.992	15.7	12.0
15	1.125	16.2	14.4
16	1.150	16.7	15.9
17	1.221	16.7	16.1
18	1.161	16.8	16.2
19	1.161	16.8	16.2
20	0.272	16.8	16.2
21	0.201	16.8	16.2
22	0.748	16.9	16.2
23	0.810	17.1	16.2
24	0.419	17.2	16.4
25	1.200	17.0	16.5
26	0.802	17.2	16.5
27	0.670	17.9	16.5
28	0.522	17.5	16.6
29	0.748	17.6	16.6
30	0.512	17.7	16.6

Tabla 3.3b. Datos de perforación (continuación).

forma de la ecuación del modelo y de los valores de los parámetros de perforación usados, es decir, los valores de  $x_2, \dots, x_9$ . Además, la velocidad de penetración y de rotación, y el peso sobre barrena, deben registrarse en intervalos cortos para asegurar que son representativos del tipo de formación atravesada. Es adecuado un intervalo de 2 a 5 pies, para garantizar el registro de datos representativos.

Una vez conformado el polinomio del modelo, puede obtenerse, mediante ecuaciones matemáticas que quedan fuera del alcance de este proyecto, la velocidad de rotación, el peso sobre la barrena y la hidráulica óptimos, así como, los costos por pie perforado. La finalidad de optimizar tales parámetros es lograr maximizar el ritmo de penetración.

### III.3 EJERCICIOS RESUELTOS.

PIII-1. Aplique el método de Kriging al conjunto de datos de porosidad mostrados en las Tablas 3.4 a y b. La gráfica del semivariograma promedio se muestra en la Figura 3.9. Esta información proviene de 65 pozos del Campo Miguel Alemán, situado en la porción sureste del Paleocanal de Chicontepec<sup>(16)</sup> (Figura 3.10).

#### Solución:

El programa de cómputo empleado se estructuró en lenguaje *Quick Basic*, y mediante la técnica de Kriging estima el valor de una

### DISTRIBUCION DE POZOS CON MUESTREO DE POROSIDAD

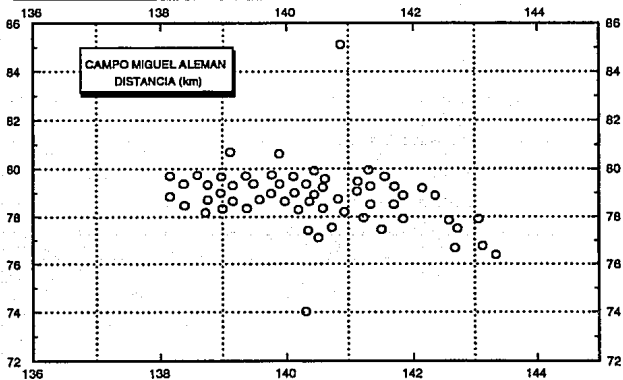


Figura 3.9.

### SEMIVARIOGRAMA PROMEDIO PARA LA POROSIDAD

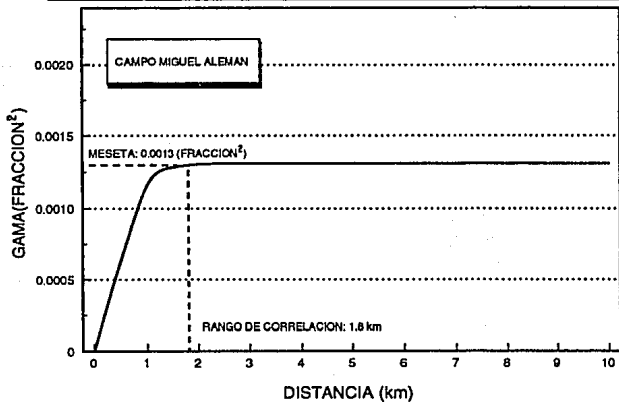


Figura 3.10.



MUESTRA	COORD. X (km)	COORD. Y (km)	PROXIMIDAD
1	142.24	76.40	0.108
2	142.12	76.78	0.10
3	142.70	76.68	0.12
4	142.74	77.00	0.16
5	142.56	77.80	0.18
6	142.06	77.92	0.18
7	140.52	77.12	0.17
8	140.26	77.42	0.16
9	140.74	77.54	0.21
10	141.52	77.46	0.16
11	141.24	77.96	0.19
12	141.80	77.92	0.17
13	140.20	78.20	0.108
14	140.60	78.26	0.14
15	140.94	78.20	0.22
16	138.72	78.18	0.15
17	139.00	78.24	0.125
18	139.28	78.26	0.14
19	138.16	78.24	0.17
20	138.76	78.70	0.17
21	138.28	78.48	0.14
22	139.16	78.66	0.17
23	139.58	78.72	0.17
24	140.28	78.64	0.14
25	140.24	78.74	0.13
26	141.22	78.54	0.14
27	140.46	78.98	0.20
28	140.12	79.68	0.19
29	140.22	79.26	0.16
30	140.59	79.60	0.26
31	140.46	79.92	0.17
32	138.97	79.00	0.15
33	139.76	78.97	0.175
34	138.27	79.28	0.16
35	138.76	79.24	0.24
36	139.16	79.21	0.11
37	139.48	79.28	0.15
38	139.90	79.26	0.247
39	138.59	79.74	0.21
40	138.98	79.67	0.11
41	138.16	79.71	0.24
42	139.27	79.70	0.21
43	139.77	79.74	0.108
44	139.98	78.64	0.19
45	140.14	79.00	0.166

Tabla 3.4a. Datos del Campo Miguel Alemán.

MUESTRA	COORD. X [ km ]	COORD. Y [ km ]	POROSIDAD
46	141.15	79.48	0.16
47	139.89	80.62	0.17
48	139.12	80.7	0.21
49	140.32	74.03	0.15
50	141.14	79.06	0.198
51	141.32	79.95	0.15
52	140.63	79.59	0.14
53	140.60	79.23	0.13
54	140.88	85.12	0.198
55	141.71	78.51	0.17
56	141.35	78.52	0.15
57	141.85	78.88	0.13
58	142.37	78.89	0.23
59	141.54	79.60	0.16
60	141.57	79.67	0.185
61	141.72	79.25	0.26
62	141.35	79.26	0.13
63	142.16	79.20	0.15
64	142.60	77.85	0.13
65	141.85	77.91	0.17

Tabla 3.4b. Datos del Campo Miguel Alemán (continuación).

variable aleatoria natural, en este caso porosidad. Para fines prácticos, el yacimiento se subdivide en 12 bloques cuadrados de 2 km de ancho cada uno (4 celdas en la dirección X y 3 en la Y). Al *Arigearse* cada celda se despliega una comparación, gráfica y numérica, entre el resultado obtenido y el calculado con métodos estadísticos convencionales.

Los resultados se observan en la Tabla 3.5 y se almacenan en un archivo llamado RESGLOBA.RES, en general son congruentes, pero se ven afectados por el número de datos circunscritos a la celda en análisis en cada paso del proceso de estimación (celdas 1, 4, y 12, con 0, 1 y 2 datos respectivamente).

Para hacer más dinámico el uso de este programa, se incluye un archivo en código ASCII a manera de base de datos (DATOS.REG,

consultar apéndice), el cual contiene los datos requeridos por la técnica de Kriging. El diagrama simplificado de flujo se presenta en la página 126 (Figura 3.11).

		No. CELDA			
		3	6	9	12
2		0.177	0.173	0.170	0.192
		0.162	0.163	0.206	0.181
		0.000	0.000	0.133	0.172
		1	4	7	10

Tabla 3.5. Estimación de la porosidad del Campo Miguel Alemán.

### III.4 PROBLEMAS PROPUESTOS.

PPIII-1. Poettmann y Carpenter\* han definido una expresión analítica que permite obtener la caída de presión en tuberías verticales con flujo multifásico, dicha expresión es la siguiente:

$$\frac{\Delta p}{\Delta n} = \frac{1}{144} \left[ \rho + \frac{f (qM)^2}{2.979E-5 \rho d^5} \right]$$

Donde:

- $\rho$  es la densidad de la mezcla sin resbalamiento, [lbm/pie<sup>3</sup>].
- $q$  es el gasto de aceite, [bl/día].
- $M$  es la masa asociada a un barril de aceite, [lbm/bl o a c.s.].
- $f$  es el factor de fricción, el cual depende del producto  $qM$ .
- $d$  es el diámetro interno de la tubería, [pgl].
- $\frac{\Delta p}{\Delta n}$  es el gradiente de presión, [lb/pg<sup>2</sup>pie].

\* POETTMMANN, F. H. Y CARPENTER, P. O., "THE MULTIPHASE FLOW OF GAS, OIL AND WATER THROUGH VERTICAL FLOW STRINGS WITH APPLICATION TO GAS LIFT INSTALLATIONS", DRILL AND PROD. PRAC., API, 1952.

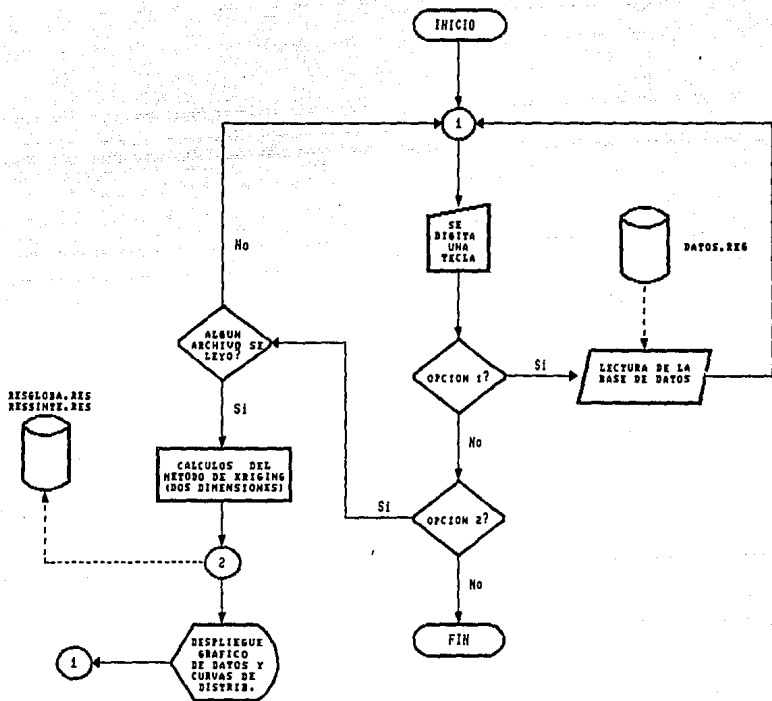


Figura 3.11. Diagrama de bloques del Programa para el Metodo de Kriging en 2 Dimensiones.

## CORRELACIONES DEL FACTOR DE FRICCIÓN

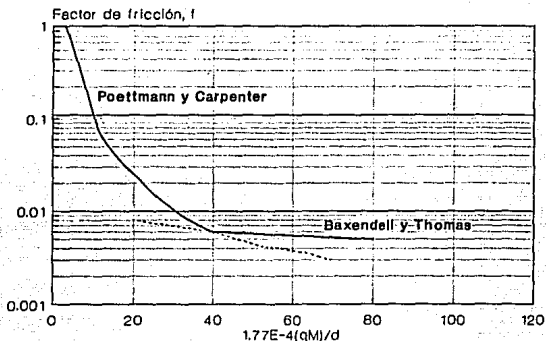


Figura 3.12. Correlaciones de Poettmann-Carpenter y Baxendell-Thomas.

Encuentre, dado un cierto diámetro de tubería ( $d$ ), el gasto de aceite por masa de la mezcla ( $qM$ ) que produzca la mínima caída de presión.

En base a la derivada de  $\Delta p/\Delta h$  con respecto al gasto ( $qM$ ) igualada a cero, se logra estimar el gasto que generará la caída mínima de presión. No obstante, el factor de fricción  $f$  depende también del gasto ( $qM$ ), por lo que resulta necesario encontrar una función que exprese  $f$  en términos de ( $qM$ ). Tal función deberá reemplazarse por  $f$  antes de proceder a la derivación. La Figura 3.12 expresa el factor de fricción  $f$  en términos de ( $qM$ )/ $d$ . Emplee la técnica de los Mínimos Cuadrados o la de Descomposición del Valor Singular, para ajustar una función a las curvas mostradas. Sustitúyala en la ecuación de Poettmann y Carpenter, y defina la expresión del gasto ( $qM$ ) en términos del diámetro  $d$ , para la cual la caída de presión en la tubería sea mínima.

## CAPITULO IV

### INTEGRACION NUMERICA

Evaluar una integral definida

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

por métodos matemáticos, es frecuentemente una tarea difícil, aún cuando  $f(x)$  presente una forma analítica simple. Afortunadamente, el Análisis Numérico ofrece diversas técnicas para integrar funciones definidas en forma tabular, obtenidas de algún experimento o simplemente valuando la función en cuestión. Una ventaja adicional de estos métodos numéricos, es que son fácilmente programables en cualquier equipo de cómputo, desde una calculadora de bolsillo hasta una computadora de gran capacidad.

En el área petrolera, es conveniente poseer técnicas sencillas y eficientes para dar solución a las integrales involucradas en el cálculo de algunos problemas, por ejemplo, puede citarse la Fugacidad o el Potencial de Gas Real,  $m(p)$ .

La elección del método de integración más apropiado para un problema particular, está regida por la cantidad de información disponible sobre la función. Básicamente, en este capítulo se consideran sólo aquellas funciones reales con una sola variable independiente,  $x$ , definidas en el intervalo  $[a, b]$ .

Tales funciones pueden agruparse en 4 categorías:

- 1) Los valores de  $f(x_i)$ , están disponibles sólo para ciertos puntos  $x_i$  del intervalo  $[a, b]$ .
- 2) La función  $f(x)$ , está bien definida y puede evaluarse para

cualquier valor real,  $x$ , en el intervalo  $[a, b]$ .

- 3) La definición de la función puede extenderse analíticamente al campo de los valores complejos de la variable  $x$ .
- 4) La función  $f(x)$  tiene una ecuación explícita disponible, con forma apropiada, para su manipulación simbólica.

Las funciones de la primera categoría son el resultado de mediciones experimentales para varios puntos  $x_i$ , los cuales a menudo no están uniformemente espaciados, o pueden haber sido obtenidos de tablas para valores equidistantes de  $x_i$ .

En las dos primeras categorías, se encuentran las funciones cuya diferenciación numérica es más difícil que la integración, esto ocurre puesto que la diferenciación numérica tiende a magnificar cualquier error inherente a los datos, mientras que la integración tiende a suavizar o disminuir el error. Si los valores de la función son conocidos o pueden calcularse con cierta exactitud, los métodos de integración basados en funciones Spline o funciones polinómicas presentan resultados satisfactorios, no obstante, si los valores de la función muestran ruido, entonces, los resultados pueden ser inexactos.

Como ejemplo de funciones en la tercera categoría, se encuentran las conformadas por complicadas expresiones trigonométricas o por funciones elementales; y si su extensión al dominio de los complejos es factible, la integración producirá resultados aceptables.

Para las funciones de la cuarta y última categoría, la diferenciación simbólica por computadora es más sencilla que la integración, ya que sólo involucra cálculos elementales.

En el ámbito de los Métodos Numéricos, el término *regla* o *cuadratura*, engloba algoritmos con los cuales es posible calcular, aproximadamente, integrales definidas. En lo sucesivo, se detallarán solamente las cuadraturas, o reglas, aplicables a las funciones contempladas por las dos primeras categorías expuestas.

#### IV.1 REGLAS DEL RECTANGULO Y DEL TRAPEZOIDE.

Suponga que  $[a, b]$  es un intervalo finito para la variable  $x$ , el cual está particionado en "n" subintervalos llamados paneles,  $[x_i, x_{i+1}]$ ,  $i = 1, 2, \dots, n$ ; además,  $x_1 = a$  y  $x_{n+1} = b$ ,  $x_1 < x_2 < \dots < x_{n+1}$ . Definiendo, el ancho de cada panel:

$$h_i = x_{i+1} - x_i \quad \dots (4.1)$$

Sea  $f(x)$  una función definida en el mismo intervalo  $[a, b]$  y supóngase que se desea una aproximación de la integral:

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

Sencillamente,  $I(f)$  puede expresarse como la suma de las integrales sobre cada panel  $h_i$  (consultar la Figura 4.1), o sea:

$$I(f) = \sum_{i=1}^n I_i$$

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, de manera individual, cada  $I_i$ . Una *regla de cuadratura*



compuesta, es una fórmula que calcula aproximadamente la integral  $I(f)$ , mediante la suma de las reglas de cuadratura simples de cada subintegral,  $I_i$ .

Las reglas de cuadratura simple más usadas son la del Rectángulo y la del Trapecoide. La Regla de Rectángulo valga la función  $f(x)$  en los puntos medios de los paneles, esto es, en el punto:

$$y_i = \frac{x_i + x_{i+1}}{2}, \quad i = 1, 2, \dots, n \quad \dots(4.2)$$

Y cada integral  $I_i$ , se obtiene como el área de un rectángulo de base  $h_i$ , y de altura  $f(y_i)$ , por lo que:

$$I_i \approx h_i f(y_i)$$

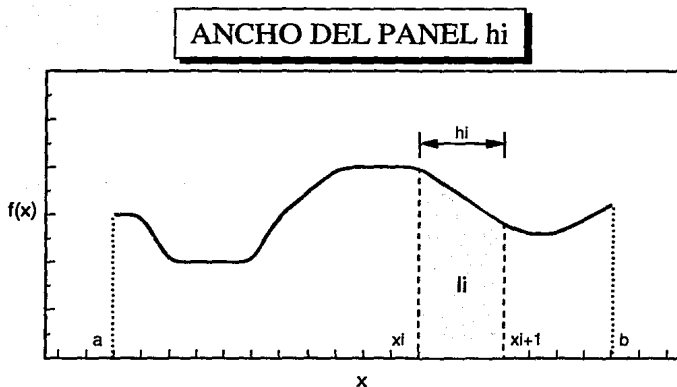


Figura 4.1.

En base a lo anterior, la Regla Compuesta del Rectángulo será la siguiente (ver Figura 4.2):

$$R(f) = \sum_{l=1}^n h_l f(y_l) \quad \dots (4.3)$$

Por otro lado, la Regla del Trapezoide valga la función en los puntos extremo de cada panel. La integral  $I_l$ , es aproximada por el área del trapezoido con base  $h_l$ , cuya altura varía, de izquierda a derecha, o viceversa, de  $f(x_l)$  a  $f(x_{l+1})$  (lo que puede apreciarse en la Figura 4.3). Por lo tanto:

$$I_l(f) \approx h_l \frac{f(x_l) + f(x_{l+1})}{2} \quad \dots (4.4)$$

Y la Regla Compuesta del Trapezoide, estará dada por la siguiente expresión:

$$T(f) = \sum_{l=1}^n h_l \frac{f(x_l) + f(x_{l+1})}{2} \quad \dots (4.5)$$

Si  $f(x)$  es una función continua —o simplemente integrable según Riemann— en el intervalo  $[a, b]$ , y si  $h = \max_l h_l$ , entonces, ambas cuadraturas convergen al resultado exacto conforme el ancho de cada subintervalo decrece<sup>(4)</sup>, es decir:

$$\lim_{h \rightarrow 0} R(f) = I(f)$$

$$\lim_{h \rightarrow 0} T(f) = I(f)$$

Ahora, surge la interrogante: ¿Qué tan rápido convergen estas reglas? La Cuadratura del Rectángulo emplea una interpolación constante (grado cero), en cada subintervalo; y la del Trapezoide, una interpolación lineal (grado uno). Por sentido común, podría esperarse mayor exactitud en la Regla Trapezoidal.

## REGLA DEL RECTANGULO

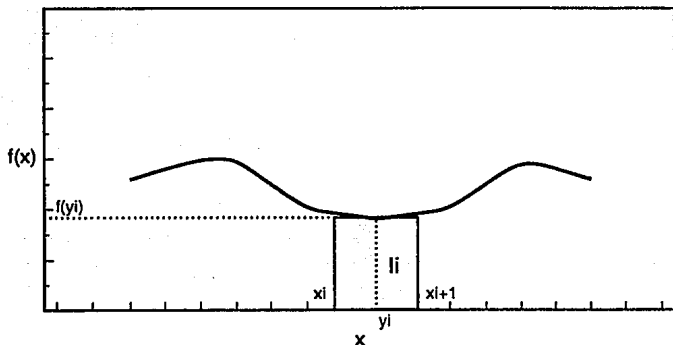


Figura 4.2.

## REGLA DEL TRAPEZOIDE

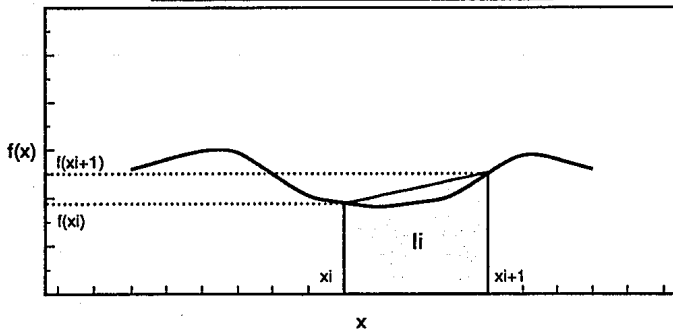


Figura 4.3.

Sean  $[a, b] = [0, 1]$ ,  $n = 1$  (un sólo panel) y la función  $f(x) = x$ , tal como se muestra en las Figuras 4.4 a y b. La Regla Trapezoidal es evidente que no generará error, ya que la interpolación lineal concuerda para cualquier punto con la función  $f(x)$ . Aún así, la Cuadratura del Rectángulo brinda un resultado también libre de error, a pesar de que la función interpoladora no coincide con la función  $f(x)$  en el punto  $x = 1/2$ . El error promedio es cero en ambos casos.

¿El ejemplo anterior es típico o simplemente es suerte de la Cuadratura del Rectángulo? ¿En general, qué regla es más exacta? Para responder estas inquietudes, se realizará un análisis basado en suposiciones sobre una función cualquiera.

Considere una función  $f(x)$ , que posee 5 derivadas continuas, cuyos valores no son muy grandes. También, considere el panel  $[x_i, x_{i+1}]$ . La expansión de  $f(x)$  con respecto al punto  $y_i$ , localizado en el centro del panel, es:

$$f(x) = f(y_i) + (x-y_i) f'(y_i) + \frac{1}{2} (x-y_i)^2 f''(y_i) + \frac{1}{6} (x-y_i)^3 f'''(y_i) + \frac{1}{24} (x-y_i)^4 f^{(4)}(y_i) + \dots$$

La suposición considerada es que los términos denotados por el símbolo "...", son despreciables, es decir, menos significativos que los mostrados explícitamente.

Al integrar cada uno de los términos de la serie anterior desde  $x_i$  a  $x_{i+1}$ , se observa lo siguiente:

FUNCION  $f(x)=x$  EN EL INTERVALO  $[0,1]$   
(REGLA DEL RECTANGULO)

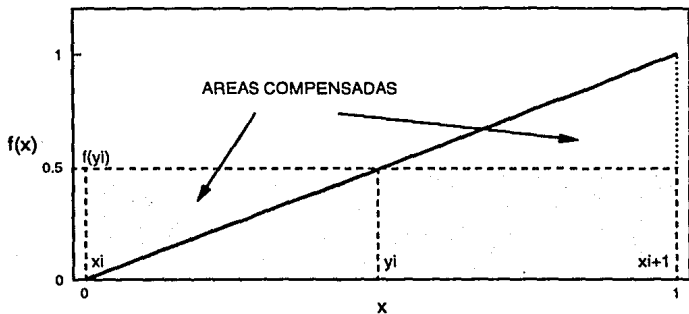


Figura 4.4a.

FUNCION  $f(x)=x$  EN EL INTERVALO  $[0,1]$   
(REGLA DEL TRAPEZOIDE)

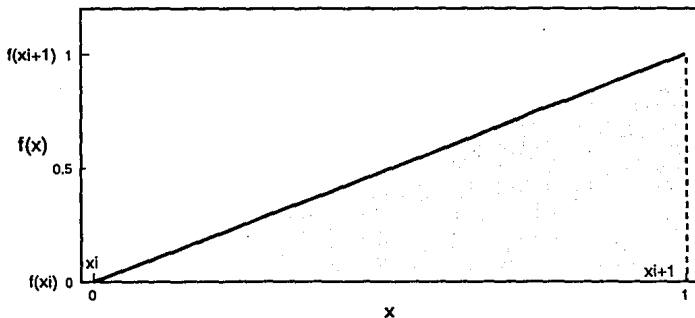


Figura 4.4b.

$$\int_{x_l}^{x_{l+1}} (x-y_l)^p dx = \begin{cases} h_l & \rightarrow p = 0 \\ 0 & \rightarrow p = 1 \\ \left(\frac{h_l^3}{12}\right) & \rightarrow p = 2 \\ 0 & \rightarrow p = 3 \\ \left(\frac{h_l^5}{80}\right) & \rightarrow p = 4 \end{cases}$$

Nótese, que las potencias impares al ser integradas son cero; consecuentemente:

$$\int_{x_l}^{x_{l+1}} f(x) dx = \underbrace{h_l f(y_l)}_{\text{Cuadratura del Rectángulo}} + \frac{1}{24} h_l^3 f''(y_l) + \frac{1}{1920} h_l^5 f^{(4)}(y_l) + \dots$$

La expresión anterior indica que para valores pequeños de  $h_l$ , el error para cada panel, en la aproximación por la Cuadratura del Rectángulo, es del orden de  $\frac{1}{24} h_l^3 f''(y_l)$ , más otros términos poco significativos.

Retomando una expansión en serie de Taylor y sustituyendo  $x = x_l$  y  $x = x_{l+1}$ , se obtiene:

$$f(x_l) = f(y_l) - \frac{1}{2} h_l f'(y_l) + \frac{1}{8} h_l^2 f''(y_l) - \frac{1}{48} h_l^3 f'''(y_l) + \frac{1}{384} h_l^4 f^{(4)}(y_l) + \dots$$

$$f(x_{l+1}) = f(y_l) + \frac{1}{2} h_l f'(y_l) + \frac{1}{8} h_l^2 f''(y_l) + \frac{1}{48} h_l^3 f'''(y_l) + \frac{1}{384} h_l^4 f^{(4)}(y_l) + \dots$$

Entonces, sumando y reordenando las expresiones anteriores:

$$\frac{f(x_l) + f(x_{l+1})}{2} = f(y_l) + \frac{1}{8} h_l^2 f''(y_l) + \frac{1}{384} h_l^4 f^{(4)}(y_l) + \dots$$

Combinando esta última expresión con la de la integral que representa el error de la Cuadratura del Rectángulo, se llega a:

$$\int_{x_l}^{x_{l+h}} f(x) dx = h_l \frac{f(x_l) + f(x_{l+h})}{2} - \frac{1}{12} h_l^3 f''(y_l) - \frac{1}{480} h_l^5 f^{(iv)}(y_l) + \dots$$

Puede observarse que el error para cada panel en la Regla del Trapezoide, cuando  $h_l$  es pequeña, es  $-\frac{1}{12} h_l^3 f''(y_l)$ , más otros términos menos significativos.

El error total para cada regla será la sumatoria del error presente en cada panel. Si se considera la siguiente convención:

$$E = \frac{1}{24} \sum_{l=1}^n h_l^3 f''(y_l)$$

$$F = \frac{1}{1920} \sum_{l=1}^n h_l^5 f^{(iv)}(y_l)$$

el error total resulta:

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

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

Si  $h_l$  es suficientemente pequeña, entonces  $h_l^5 \ll h_l^3$ , y si  $f^{(iv)}$  no presenta un comportamiento errático, entonces  $F \ll E$ .

Las conclusiones derivadas del análisis anterior son:

- Para la gran mayoría de las funciones  $f(x)$ , la Regla del Rectángulo es cerca de 2 veces más exacta que la del Trapezoide.

- La diferencia entre los valores obtenidos con cada regla puede usarse para estimar el error en la integración. Esto es:

$$R(f) + E + F + = T(f) - 2E - 4F$$

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

- La diferencia entre el resultado arrojado por el uso de una regla, antes y después de duplicar el número paneles, también ofrece una aproximación del error en la integración, pues al duplicarse el número de paneles se cuadruplica la exactitud de la cuadratura.

La técnica de aumentar repetidamente la cantidad de paneles al doble y de calcular el error, es susceptible de ser programada para generar un método que automáticamente determine el número de paneles necesarios, logrando que el valor de la integral aproximada esté dentro de una tolerancia de error preestablecida. Este método se aplica en cuadraturas más sofisticadas, que se expondrán, en detalle, más adelante (sección IV.4).

#### IV.2 CUADRATURA SPLINE.

La interpolación cúbica Spline permite obtener una fórmula sencilla de cuadratura. Para  $x_i \leq x \leq x_{i+1}$ , la función Spline que interpola a todos los puntos base de  $f(x)$ , es:

$$s(x) = f_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

Entonces para  $a = x_i$ ,  $b = x_{i+1}$  y  $h_i = x_{i+1} - x_i$ , siendo "n" el número de paneles, se define la Cuadratura Compuesta Spline como:



$$\int_a^b f(x) dx \approx \int_a^b s(x) dx \\ = \sum_{i=1}^n \left[ h_i f_i + \frac{1}{2} h_i^2 b_i + \frac{1}{3} h_i^3 c_i + \frac{1}{4} h_i^4 d_i \right] \quad \dots (4.6)$$

Los coeficientes  $b_i$ ,  $c_i$  y  $d_i$  se calculan con la subrutina SPLINE (descrita en el capítulo III). Como se tienen "n+1" nodos o puntos base, SPLINE debe invocarse con  $N = n+1$ .

La función  $s(x)$  puede ser representada también como (consultar sección III.1.4):

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

Donde:

$$w = 1 - \bar{w} = \frac{(x - x_i)}{h_i}, \quad \sigma_i = \frac{s''(x_i)}{6} = \frac{c_i}{3}$$

Las  $\sigma_i$  son la solución del sistema tridiagonal que se conforma con las ecuaciones anteriores, y que se discutió en la sección III.1.4. Continuando con el desarrollo, se observa que:

$$\int_{x_i}^{x_{i+1}} s(x) dx = h_i \int_0^1 s(w) dw$$

Además:

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

En consecuencia:

$$\int_{x_i}^{x_{i+1}} s(x) dx = h_i \frac{f_i + f_{i+1}}{2} - h_i^2 \frac{\sigma_i + \sigma_{i+1}}{4}$$

Es evidente que la fórmula de Cuadratura Spline es igual a la Trapezoidal, más un término de corrección que involucra a  $c_i$ . Una ecuación más simple, para calcular la Cuadratura Compuesta Spline es:

$$\int_a^b s(x) dx = \sum_{i=1}^n \left[ h_i \frac{f_i + f_{i+1}}{2} - h_i^3 \frac{c_i + c_{i+1}}{12} \right] \quad \dots(4.7)$$

Por lo tanto, sólo los vectores de datos  $(x, f)$  y el de segundas derivadas  $(c)$ , obtenido por SPLINE, se requerieren para evaluar la integral. La exactitud de la Cuadratura Spline puede calcularse mediante la siguiente expresión:

$$\begin{aligned} h_i^3 \frac{c_i + c_{i+1}}{12} &= h_i^3 \frac{s''(x_i) + s''(x_{i+1})}{24} \\ &\approx \frac{h_i^3}{12} f''(y_i) \end{aligned}$$

Se aprecia entonces, que el término de corrección provisto por la Cuadratura Spline aproxima el error de la Regla Trapezoidal.

En la literatura no se han consignado resultados prácticos sobre la utilización de la Cuadratura Spline, pero en base a este análisis, parece revestir gran utilidad; aunque se recomienda para casos en los cuales los puntos dato son fijos y los métodos automáticos, que se describirán más adelante, no están disponibles.

#### IV.3 REGLA DE SIMPSON.

En la sección IV.1, se expusieron las Cuadraturas del Rectángulo y del Trapezoide, éstas son:

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

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

Además, se probó que los errores de esas reglas están dados por:

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

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

Donde:

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

$$F = \frac{1}{1920} \sum_{i=1}^n h_i^5 f^{(4)}(y_i)$$

Combinando apropiadamente ambas reglas es factible crear una nueva cuadratura cuya fórmula de error no contenga los términos de E. Como  $R(f)$  es cerca de 2 veces más exacta que  $T(f)$ , la combinación idónea es<sup>(4)</sup>:

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

Y desarrollando la expresión anterior, se llega a:

$$S(f) = \sum_{i=1}^n \frac{1}{6} h_i \left[ f(x_i) + 4 f\left(\frac{x_i + x_{i+1}}{2}\right) + f(x_{i+1}) \right] \quad \dots(4.8)$$

El lector identificará la ecuación (4.8) como la Cuadratura Compuesta de Simpson, cuyo error puede obtenerse directamente de las fórmulas para los errores en las reglas del Trapezoide y del Rectángulo, o sea:

$$I(f) - S(f) = \frac{2}{3} [I(f) - R(f)] + \frac{1}{3} [I(f) - T(f)]$$

$$\begin{aligned}
&= \frac{h^2}{6} (E + F + \dots) + \frac{1}{3} (-2E - 4F + \dots) \\
&= \left(\frac{h^2}{6} - \frac{2}{3}\right) E + \left(\frac{h^2}{6} - \frac{4}{3}\right) F + \dots \\
&= -\frac{h^2}{3} F + \dots \\
&= -\frac{1}{2280} \sum_{l=1}^n h_l^5 f^{(4)}(y_l) + \dots
\end{aligned}$$

Obsérvese que  $S(f)$  está basada en una interpolación de segundo grado y que su término de error involucra derivadas de cuarto orden, por lo cual, la Cuadratura de Simpson es exacta para funciones cúbicas.

Si la longitud de cada panel es bipartida, entonces cada término  $h_l^5$  en la fórmula del error, decrece a razón de  $\frac{1}{32}$ . El número total de términos se incrementa al doble, por lo consiguiente, la reducción del error total es de cerca de  $\frac{1}{16}$ . Lo anterior debe contemplarse al emplear programas que automáticamente asignen la cantidad y el tamaño de los paneles.

La técnica de combinar 2 aproximaciones con errores similares para obtener una aproximación más exacta puede aplicarse también a esta cuadratura. Por ejemplo, los valores de  $S(f)$  para dos cantidades diferentes de paneles, pueden combinarse para lograr un nuevo valor, el cual incluirá un error de  $h_l^7$  y  $f^{(4)}(x)$ . La sistematización de esta técnica desemboca en el popular método de la Cuadratura de Romberg.

#### IV.4 CUADRATURA ADAPTADA.

Una Cuadratura Adaptada es un algoritmo numérico que utiliza una o dos cuadraturas básicas, determinando automáticamente el tamaño

de los subintervalos o paneles, para aproximar la solución de una integral definida; cumpliéndose a la vez, con ciertos lineamientos de exactitud, previamente impuestos. El tamaño de cada panel depende de la suavidad de la curva a integrar. De esta forma, el tamaño del subintervalo será relativamente grande, donde la función tenga un comportamiento suave; y será pequeño, cuando ésta cambie abruptamente. Esto se aprecia en la figura 4.5.

La mayor pérdida de tiempo de cómputo es debida al cálculo de  $f(x)$ . Por lo tanto, si dos cuadraturas, brindan resultados que cumplen con la exactitud especificada, considerando la misma función en ambos casos, aquella que use el menor número de valuaciones de  $f(x)$ , debe definirse como la más eficiente para ese problema en particular. Con la técnica descrita en el párrafo anterior, se logra un resultado que se apega a la exactitud deseada y que invierte el menor tiempo de cómputo posible.

### CUADRATURA ADAPTADA

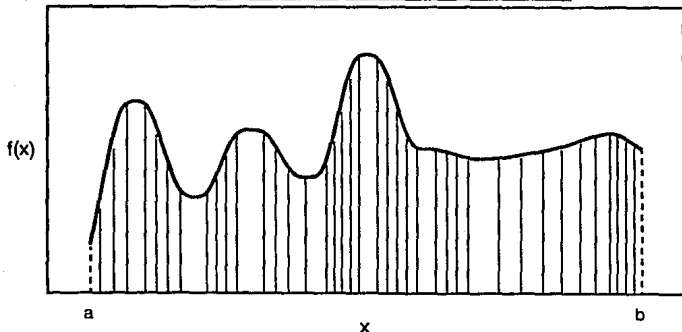


Figura 4.5.

La Cuadratura Adaptada requiere de la información siguiente:

- Un intervalo finito  $[a, b]$ .
- Un subprograma que valúe a  $f(x)$  para toda " $x$ ",  $x \in [a, b]$ .
- Un tolerancia de error ( $\epsilon$ ).

Entonces, el algoritmo calculará un valor  $Q$ , tal que:

$$\left| Q - \int_a^b f(x) dx \right| \leq \epsilon \quad \dots (4.9)$$

El intervalo de integración se divide en " $n$ " paneles,  $[x_i, x_{i+1}]$ , de manera que,  $x_1 = a$  y  $x_{n+1} = b$ . Dicho número, está en función de la exactitud solicitada ( $\epsilon$ ) y de  $f(x)$ . Como anteriormente fue definido, el ancho del panel es  $h_i = x_{i+1} - x_i$ .

Un esquema típico de Cuadratura Adaptada, aplica dos cuadraturas distintas en cada panel. Entonces, denotemos esos resultados como  $P_i$  y  $Q_i$ . Por ejemplo, un esquema que emplee la Regla de Simpson, se basa en la fórmula para dos paneles mostrada a continuación (ver Figura 4.6):

$$P_i = \frac{1}{6} h_i \left[ f(x_i) + 4f\left(x_i + \frac{h_i}{2}\right) + f(x_i + h_i) \right]$$

y en la siguiente para cuatro paneles:

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

Tanto  $P_i$  como  $Q_i$  son aproximaciones a la integral:

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

El principio fundamental de este esquema consiste en comparar las aproximaciones  $P_i$  y  $Q_i$ . Obteniéndose con esto, una estimación

de la exactitud de tales aproximaciones, si el resultado es satisfactorio, una de ellas se considerará como el valor de la integral del subintervalo en análisis; de lo contrario, se dividirá el intervalo en dos o más partes y se repetirá el proceso en estos subintervalos más pequeños.

La evaluación de la función  $f(x)$  se simplifica debido a que tanto  $P_i$  como  $Q_i$  poseen ciertos términos comunes entre sí.  $Q_i$  contiene únicamente dos términos extras que no están en  $P_i$ . De aquí en adelante, deberá suponerse que  $Q_i$  se obtiene aplicando la regla para  $P_i$  en dos ocasiones, una para cada mitad del intervalo. Esta es una técnica que facilitará el análisis de la Cuadratura Adaptada.

Considérese que la regla para  $P_i$  ofrece la respuesta exacta si la función por integrar es de grado  $p-1$ , o equivalentemente, si la  $p$ -ésima derivada,  $f^{(p)}(x)$ , es igual con cero.

## CUADRATURA ADAPTADA (REGLA DE SIMPSON)

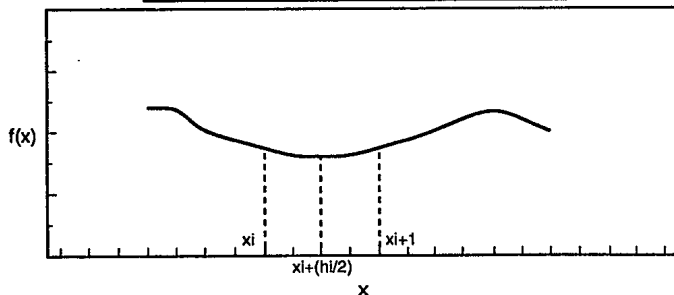


Figura 4.6.

Expandiendo la función por integrar en Series de Taylor, con respecto al punto medio del subintervalo, puede observarse que existe una constante "c", tal que:

$$I_l - P_l = c (h_l)^{p+1} f^{(p)}\left[x_l + \frac{h_l}{2}\right] + \dots$$

Como se supuso que  $Q_l$  es la suma de dos  $P_l$  calculadas en subintervalos de longitud  $h_l/2$ , se tiene:

$$I_l - Q_l = c \left(\frac{h_l}{2}\right)^{p+1} \left[ f^{(p)}\left[x_l + \frac{h_l}{4}\right] + f^{(p)}\left[x_l + \frac{3h_l}{4}\right] \right] + \dots$$

Pero como:

$$f^{(p)}\left[x_l + \frac{h_l}{4}\right] + f^{(p)}\left[x_l + \frac{3h_l}{4}\right] = 2f^{(p)}\left[x_l + \frac{h_l}{2}\right] + \dots$$

Entonces, los dos errores estarán relacionados por:

$$I_l - Q_l = \frac{2}{2^{p+1}} (I_l - P_l) + \dots = \frac{1}{2^p} (I_l - P_l) + \dots$$

Lo anterior indica que al bisectar el subintervalo, decrecerá el error en un factor de  $2^p$ . Resolviendo para la incógnita  $I_l$ , y reorganizando términos, se obtiene:

$$Q_l - I_l = \frac{1}{2^p - 1} (P_l - Q_l) + \dots$$

En otras palabras, el error en la expresión  $Q_l$  es de alrededor de  $1/(2^p - 1)$  veces la diferencia entre ambas aproximaciones.

El objetivo primario será entonces, bipartir cada subintervalo



hasta que se satisfaga la siguiente desigualdad:

$$\frac{1}{2^p - 1} \cdot |P_l - Q_l| \leq \frac{h_l}{b - a} \epsilon \quad \dots(4.10)$$

Donde,  $\epsilon$  es la tolerancia proporcionada por el usuario. Si el intervalo  $[a, b]$  fuese completamente cubierto por "n" subintervalos, el resultado sería:

$$Q = \sum_{l=1}^n Q_l$$

Considerando la igualdad anterior y despreciando los términos de alto orden, se tiene:

$$\begin{aligned} \left| Q - \int_a^b f(x) dx \right| &= \left| \sum_{l=1}^n Q_l - I_l \right| \\ &\leq \sum_{l=1}^n |Q_l - I_l| \\ &\leq \frac{1}{2^p - 1} \sum_{l=1}^n |P_l - Q_l| \\ &\leq \frac{1}{2^p - 1} \left( \frac{2^p - 1}{b - a} \right) \epsilon \sum_{l=1}^n h_l \\ &= \epsilon \end{aligned}$$

lo cual coincide con lo planteado en la ecuación (4.9).

Debe recordarse que este análisis requiere la suposición de que  $f^{(p)}(x)$  es continua, y que el error es proporcional a  $(h_l)^{p+\epsilon} f^{(p)}(x)$ . Aunque en la realidad esto no se cumpla, la cuadratura arrojará finalmente, un resultado apegado a la tolerancia fijada.

Por simplicidad, se ha manejado el criterio de error absoluto, o sea:

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

En numerosas ocasiones, se utiliza algún tipo de error relativo, el cual es independiente de factores de escala en la función  $f$ , complicándose su verificación por diversas razones. Un criterio de error relativo puro, es:

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

Como el denominador,  $\int f$ , puede ser cero, debido a la cancelación asociada a integrales oscilantes, el criterio puede ser imposible de satisfacer. Además, una buena aproximación del valor del denominador se lograría sólo hasta el final del cálculo.

Otras cuadraturas, usan un criterio de error relativo que involucra a  $\int |f|$ , es decir:

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

En la expresión anterior, el denominador será distinto de cero, a menos que, la función  $f$  fuese idéntica a cero.

El usuario de una subrutina de Cuadratura Adaptada debe tener en mente estas pruebas de exactitud, y por supuesto, debe checar la documentación particular para tal subrutina.

#### IV.4.1 SUBROUTINA QUANCB.

QUANCB es una subrutina que emplea una técnica de cuadratura adaptada para aproximar el valor de una integral definida. Su nombre se deriva de Cuadratura Adaptada de Newton-Cotes para 8 paneles (*Quadrature Adaptive of Newton-Cotes 8-panel*). Las fórmulas de Newton-Cotes constituyen una familia de cuadraturas o reglas para integración e interpolación polinomial, sobre puntos de evaluación igualmente espaciados. Las reglas del Rectángulo, del Trapezoide y de Simpson, son obtenidas integrando polinomios de 0, 1° y 2° grado; y son los primeros miembros de la familia. La fórmula de 8 paneles requiere la integración de un polinomio de 8° grado, pero igual que para la regla de Rectángulo o de Simpson, se permite un grado extra, de manera que, para polinomios de 9° grado se tendrán resultados exactos.

La subrutina se presenta, en lenguaje *Basic*, en el *diskette* de consulta (QUANCB.BAS), e incluye un subprograma, FUN(X), para valuar la integral  $f(x)$ ; el usuario debe proporcionar los límites, superior e inferior, de integración,  $A = a$  y  $B = b$ , y las tolerancias de error absoluto, ABSERR, y relativo, RELERR. La subrutina calcula la aproximación de la integral, RESULT, el error absoluto estimado, ERREST, el número de valuaciones requeridas de  $f(x)$ , NOFUN, y un indicador de la eficacia del cálculo, FLAG. Si FLAG es cero, RESULT cumplirá con la exactitud fijada; si FLAG es un valor mínimo, RESULT será aceptable, y en caso contrario, será inaceptable.

Como esta regla es exacta para polinomios de 9° grado, el análisis de la sección IV.4 puede aplicarse con  $p = 10$ . El error en  $Q_t$  es del orden de  $1/(2^{10} - 1) = \frac{1}{1023}$  veces el error en  $P_t$ , sólo si la curva por integrar es suave y se desprecian los términos de alto grado.

Entonces, los resultados de un subintervalo en particular son aceptables, si:

$$\frac{1}{1023} |P_i - Q_i| \leq \frac{h_i}{b-a} \epsilon$$

Cada subintervalo será bisectado, existiendo una cantidad límite, LEVMAX, la cual es de 30 biparticiones. Cuando tal límite es sobrepasado, el subintervalo se aprobará siempre y cuando se cumpla el criterio de exactitud establecido, pero dicho número se almacenará en la parte entera de FLAG. Por lo tanto, la parte entera del valor de FLAG, representa el número de subintervalos o paneles que se calcularon al superarse LEVMAX, y la parte decimal, indica el porcentaje de análisis del intervalo en el que se detecte la no convergencia a la tolerancia. También, puede averiguarse en que valor de la abscisa, x, se presentó el problema, mediante la siguiente relación:

$$X_0 = B - (\text{parte fraccional de FLAG}) \cdot (B - A)$$

Por ejemplo, si A = 0, B = 2 y FLAG = 91.21, 91 subintervalos fueron obtenidos, lográndose analizar el 21% del intervalo que muestra dificultades. Y el valor de "x" para el cual se detectó la no convergencia es  $X_0 = 2 - 0.21 \cdot (2-0) = 1.58$ .

#### IV.5 CUADRATURAS GAUSSIANAS.

Otra alternativa para obtener la aproximación de una integral definida la representan las Cuadraturas Gaussianas, que involucran una sumatoria ponderada de la función, evaluada en los puntos base, esto es:

$$\int_a^b f(x) dx \approx \sum_{i=0}^n w_i f(x_i)$$

Donde  $w_i$  es el peso o ponderador asociado a cada valor de la función  $f(x_i)$ ,  $i = 0, 1, \dots, n$ . Antes de proceder al desarrollo de tales cuadraturas es necesario exponer el tema de los Polinomios Ortogonales.

#### IV.5.1 POLINOMIOS ORTOGONALES.

Dos funciones,  $g_n(x)$  y  $g_m(x)$ , seleccionadas de una familia de funciones  $\{g_k(x)\}$ , son ortogonales, con respecto a la función ponderadora,  $w(x)$ , en el intervalo  $[a, b]$ , si:

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

$$\int_a^b w(x) [g_n(x)]^2 dx = c(n) \neq 0$$

En general, "c" depende de "n". Si estas relaciones se cumplen para toda "n", la familia de funciones,  $\{g_k(x)\}$ , constituye un conjunto de funciones ortogonales. Las funciones ortogonales más comunes son  $\{\sin kx\}$  y  $\{\cos kx\}$ .

La ortogonalidad, en términos de espacios vectoriales, se interpreta como la perpendicularidad existente entre dos vectores, en un espacio de "n" dimensiones, siendo "n" un valor muy grande (consultar la Figura 4.7).

La familia de polinomios  $\{x^k\}$ , define un espacio vectorial cuyos elementos no son ortogonales. Otras familias, como las constituidas por polinomios de Legendre, de Laguerre, de Chebyshev o de Hermite, definen un espacio vectorial y además son ortogonales.

**FAMILIA DE FUNCIONES ORTOGONALES**  
 (Cada función está representada por un vector en un espacio n-dimensional)

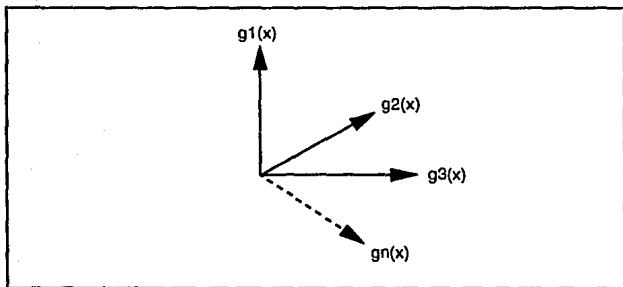


Figura 4.7.

Enseguida, se enuncian los principios básicos de las familias de polinomios ortogonales mencionadas:

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

Los polinomios de Legendre son ortogonales en el intervalo  $[-1, 1]$ , y su función ponderadora es  $w(x) = 1$ , esto es:

$$\int_{-1}^1 P_n(x) P_m(x) dx = 0 \quad ; \quad n \neq m$$

$$\int_{-1}^1 [P_n(x)]^2 dx = c(n) \neq 0$$

La ecuación de recurrencia es:

$$P_n(x) = \frac{(2n-1)x}{n} P_{n-1}(x) - \frac{n-1}{n} P_{n-2}(x) \quad \dots(4.11)$$

Por lo cual, los primeros polinomios de Legendre son:

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_2(x) &= \frac{1}{2} (3x^2 - 1) \end{aligned}$$

b) Polinomios de Laguerre  $\{L_n(x)\}$ .

Los polinomios de Laguerre son ortogonales en el intervalo  $[0, \infty]$ , y su función ponderadora es  $w(x) = e^{-x}$ , por lo que:

$$\begin{aligned} \int_0^{\infty} e^{-x} L_n(x) L_m(x) dx &= 0 \quad ; \quad n \neq m \\ \int_0^{\infty} e^{-x} [L_n(x)]^2 dx &= c(n) \neq 0 \end{aligned}$$

La ecuación de recurrencia es:

$$L_n(x) = (2n - x - 1) L_{n-1}(x) - (n-1) L_{n-2}(x) \quad \dots (4.12)$$

Entonces, los primeros polinomios de Laguerre son:

$$\begin{aligned} L_0(x) &= 1 \\ L_1(x) &= -x + 1 \\ L_2(x) &= x^2 - 4x + 2 \end{aligned}$$

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

Los polinomios de Chebyshev son ortogonales en el intervalo  $[-1, 1]$ , y su función ponderadora es  $w(x) = 1/\sqrt{1-x^2}$ , o sea:

$$\begin{aligned} \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} T_n(x) T_m(x) dx &= 0 \quad ; \quad n \neq m \\ \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} [T_n(x)]^2 dx &= c(n) \neq 0 \end{aligned}$$

La ecuación de recurrencia es:

$$T_n(x) = 2x T_{n-1}(x) - T_{n-2}(x) \quad \dots(4.13)$$

De tal forma, que los primeros polinomios de Legendre son:

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \end{aligned}$$

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

Los polinomios de Hermite son ortogonales en el intervalo  $[-\infty, \infty]$ , y su función ponderadora es  $w(x) = e^{-x^2}$ , es decir:

$$\begin{aligned} \int_{-\infty}^{\infty} e^{-x^2} H_n(x) H_m(x) dx &= 0 \quad ; \quad n \neq m \\ \int_{-\infty}^{\infty} e^{-x^2} [H_n(x)]^2 dx &= c(n) \neq 0 \end{aligned}$$

La ecuación de recurrencia es:

$$H_n(x) = 2x H_{n-1}(x) - 2(n-1) H_{n-2}(x) \quad \dots(4.14)$$

Los primeros polinomios de Hermite son:

$$\begin{aligned} H_0(x) &= 1 \\ H_1(x) &= 2x \\ H_2(x) &= 4x^2 - 2 \end{aligned}$$

Cada uno de los polinomios  $P_n(x)$ ,  $\mathcal{L}_n(x)$ ,  $T_n(x)$  y  $H_n(x)$ , es un polinomio con coeficientes reales, de  $n$ -ésimo grado en la variable "x", y con "n" raíces reales distintas dentro del intervalo de integración apropiado. Por ejemplo, todas las raíces de  $\mathcal{L}_n(x)$  caen dentro del intervalo  $[0, \infty]$ .

Un polinomio cualquiera, de grado  $n$ -ésimo,  $p_n(x) = \sum_{l=0}^n \alpha_l x^l$ , puede ser representado como una combinación lineal de cualquier



familia de polinomios ortogonales. Por lo que:

$$p_n(x) = \beta_0 Z_0(x) + \beta_1 Z_1(x) + \dots + \beta_n Z_n(x) = \sum_{l=0}^n \beta_l Z_l(x)$$

Donde  $Z_l(x)$  es un polinomio de  $l$ -ésimo grado de alguna familia de polinomios ortogonales, obtenido con la ecuación de recurrencia respectiva.

Por ejemplo, si expandemos un polinomio de cuarto grado,  $p_4(x)$ , en términos de los polinomios de Legendre:

$$\begin{aligned} p_4(x) &= \sum_{l=0}^4 \alpha_l x^l = \sum_{l=0}^4 \beta_l Z_l(x) = \\ &= \beta_0 + \beta_1 x + \beta_2 \left( \frac{3}{2} x^2 - \frac{1}{2} \right) + \beta_3 \left( \frac{5}{2} x^3 - \frac{3}{2} x \right) + \\ &\quad + \beta_4 \left( \frac{35}{8} x^4 - \frac{15}{4} x^2 + \frac{3}{8} \right) \end{aligned}$$

Agrupando términos:

$$\begin{aligned} p_4(x) &= \left( \beta_0 - \frac{1}{2} \beta_2 + \frac{3}{8} \beta_4 \right) + \left( \beta_1 - \frac{3}{2} \beta_3 \right) x + \\ &\quad + \left( \frac{3}{2} \beta_2 - \frac{15}{4} \beta_4 \right) x^2 + \frac{5}{2} \beta_3 x^3 + \frac{35}{8} \beta_4 x^4 \end{aligned}$$

Igualando con los coeficientes  $\alpha_l$ , se tiene:

$$\beta_4 = \frac{8}{35} \alpha_4$$

$$\beta_3 = \frac{2}{5} \alpha_3$$

$$\beta_2 = \frac{2}{3} \left( \alpha_2 + \frac{6}{7} \alpha_4 \right)$$

$$\beta_1 = \alpha_1 + \frac{3}{5} \alpha_3$$

$$\beta_0 = \alpha_0 + \frac{1}{3} \alpha_2 + \frac{1}{5} \alpha_4$$

El polinomio  $p_4(x) = x^4 + 3x^2 - 2x^2 + 2x$ , expresado por polinomios de Legendre (el lector puede verificarlo), es:

$$P_4(x) = -\frac{22}{15} P_0(x) + \frac{19}{5} P_1(x) - \frac{16}{21} P_2(x) + \frac{6}{5} P_3(x) + \frac{8}{35} P_4(x)$$

#### IV.5.2 CUADRATURA DE GAUSS-LEGENBRE.

El método de las Cuadraturas Gaussianas propone aproximar una integral definida, por medio de la siguiente relación matemática:

$$\int_a^b f(x) dx = \int_a^b w(x) f(x) dx \approx \sum_{l=0}^n w_l f(t_l)$$

Siendo  $w(x)$  la función peso asociada a alguna familia de polinomios ortogonales, cada nodo  $t_l$  es la raíz de un polinomio de grado "n" de tal familia, y cada ponderador  $w_l$  es la solución de la integral, efectuada mediante un polinomio de grado "i", de la misma familia, sobre el intervalo [a, b].

Muchas técnicas de integración numérica, como las cuadraturas del Rectángulo, del Trapecio, de Simpson y Spline, emplean puntos o nodos equidistantes,  $x_l$ , lo cual puede ser conveniente pero no necesario, además  $w(x) = 1$ . En contraste, en los métodos Gaussianos no se requiere de puntos igualmente espaciados, sino que se usan las raíces de un polinomio ortogonal de grado "n", definido en el intervalo [a, b], y con la función  $w(x)$  como función peso.

El método de Gauss-Legendre se apoya en los polinomios de Legendre para dar solución a una integral definida; la ecuación respectiva es:

$$\int_{-1}^1 F(z) dz \approx \sum_{l=0}^n w_l F(z_l) \quad \dots(4.15)$$

Donde:

$$w_i = \int_{-1}^1 L_i(z) dz = \int_{-1}^1 \prod_{\substack{j=c \\ j \neq i}}^n \left[ \frac{z - z_j}{z_i - z_j} \right] dz \quad \dots (4.16)$$

Siendo  $z_i$  la  $i$ -ésima raíz de un polinomio de Legendre de grado  $n+1$ ,  $P_{n+1}(z)$ . Dichas raíces pueden calcularse mediante la siguiente expresión:

$$\prod_{i=0}^n (z - z_i) = b_{n+1} P_{n+1}(z)$$

Algunos valores de  $z_i$  y de  $w_i$  para  $n = 1, 2, 3$  y  $4$  (correspondientes a las fórmulas para 2, 3, 4 y 5 puntos, respectivamente) se presentan en la Tabla 4.1<sup>(2)</sup>.

raíces ( $z_i$ )	$\int_{-1}^1 F(z) dz \approx \sum_{i=0}^n w_i F(z_i)$	factores ( $w_i$ )
$\pm 0.5773502692$	fórmula de dos puntos $n=1$	1.0000000000
0.0000000000	fórmula de tres puntos $n=2$	0.8888888889
$\pm 0.7745966692$		0.5555555556
$\pm 0.3399810436$	fórmula de cuatro puntos $n=3$	0.6521451549
$\pm 0.8611363116$		0.3478548451
0.0000000000	fórmula de cinco puntos $n=4$	0.3688888889
$\pm 0.5384698101$		0.4786286704
$\pm 0.9061798459$		0.2369268850

Tabla 4.1. Raíces de polinomios de Legendre,  $P_{n+1}(z)$ , y factores de peso para la Cuadratura de Gauss-Legendre.

Como puede observarse, el método de Gauss-Laguerre es aplicable en el intervalo  $-1 \leq z \leq 1$ , para extenderlo al intervalo  $a \leq x \leq b$ ,

se puede implementar un cambio de variable, es decir:

$$x = \frac{z(b-a) + a + b}{2}$$

De tal manera que:

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{z(b-a) + a + b}{2}\right) dz$$

Entonces, considerando la expresión (4.15), se llega a:

$$\int_a^b f(x) dx = \frac{b-a}{2} \sum_{i=0}^n w_i f\left(\frac{z_i(b-a) + a + b}{2}\right) \quad \dots(4.17)$$

La ecuación (4.17) es la fórmula general de la Cuadratura de Gauss-Legendre, y es más apropiada que la expresión (4.15) para cálculos mediante máquinas computadoras, puesto que no requiere de una transformación simbólica de  $f(x)$ ; en lugar de ella, los puntos base,  $z_i$ , son modificados y los factores de peso,  $w_i$ , son afectados por  $(b-a)/2$ .

Las expresiones para esta cuadratura son exactas si la función a integrar es un polinomio de grado menor o igual a  $2n+1$ , en caso contrario, el error en que se incurre está dado por:

$$E_n(z) = \frac{2^{2n+3} [(n+1)!]^4}{(2n+3) [(2n+2)!]^2} F^{(2n+2)}(\xi) \quad \dots(4.18)$$

$$\xi \in (-1, 1)$$

Si las magnitudes de las derivadas de alto orden de  $F(z)$ , disminuyen o no se incrementan sustancialmente, mientras que "n" crece, la Cuadratura de Gauss-Legendre será más precisa que otras Cuadraturas Gaussianas (que se describirán a continuación), para

valores equivalentes de "n".

#### IV.5.3 CUADRATURA DE GAUSS-LAGUERRE.

Los polinomios de Laguerre son la herramienta de la Cuadratura de Gauss-Laguerre. La ecuación de esta cuadratura es:

$$\int_0^{\infty} e^{-z} F(z) dz \approx \sum_{i=0}^n w_i F(z_i) \quad \dots (4.19)$$

Donde:

$$w_i = \int_0^{\infty} e^{-z} L_i(z) dz = \int_0^{\infty} e^{-z} \prod_{\substack{j=0 \\ j \neq i}}^n \left[ \frac{z - z_j}{z_i - z_j} \right] dz \quad \dots (4.20)$$

Siendo  $z_i$  la  $i$ -ésima raíz de un polinomio de Laguerre de grado  $n+1$ ,  $L_{n+1}(z)$ . Dichas raíces pueden calcularse mediante la expresión:

$$\prod_{i=0}^n (z - z_i) = b_{n+1} L_{n+1}(z)$$

Los valores de  $z_i$  y de  $w_i$  para  $n = 1, 2, 3$  y  $4$  (correspondientes a las fórmulas para 2, 3, 4 y 5 puntos, respectivamente) se presentan en la Tabla 4.2<sup>(2)</sup>.

Las ecuaciones para esta cuadratura son exactas si la función a integrar es un polinomio de grado menor o igual a  $2n+1$ , en caso contrario, el error en que se incurre está dado por:

$$E_n = \frac{[(n+1)!]^2}{(2n+2)!} F^{(2n+2)}(\xi) \quad \dots (4.21)$$

$$\xi \in (0, \infty)$$

raíces ( $z_i$ )	$\int_0^{\infty} e^{-x} F(z) dz \approx \sum_{i=0}^n w_i F(z_i)$	factores ( $w_i$ )
0.3857864376 3.4142135624	fórmula de dos puntos $n=1$	0.8525523006 0.1464468004
0.4157748568 2.2042208608 6.2899480829	fórmula de tres puntos $n=2$	0.7110980099 0.2785177886 0.0103842015
0.2229476896 1.7497611012 4.5266202969 9.2930709123	fórmula de cuatro puntos $n=3$	0.6031241042 0.2374186024 0.0388870023 0.0005292947
0.2625608197 1.4184080591 3.8064257710 7.0858100059 12.6408008442	fórmula de cinco puntos $n=4$	0.5217556106 0.2986668111 0.0789424497 0.0086117587 0.0000283700

Tabla 4.2. Raíces de polinomios de Laguerre,  $L_n(z)$ , y factores de peso para la Cuadratura de Gauss-Laguerre.

Para evaluar integrales de la forma:

$$\int_a^{\infty} e^{-x} f(x) dx \quad \dots(4.22)$$

Donde el valor del límite inferior  $a$ , es arbitrario y finito, debe realizarse un cambio lineal de variable,  $x = z + a$ . Por lo tanto:

$$\begin{aligned} \int_a^{\infty} e^{-x} f(x) dx &= \int_0^{\infty} e^{-(z+a)} f(z+a) dz \\ &= e^{-a} \int_0^{\infty} e^{-z} f(z+a) dz \quad \dots(4.23) \end{aligned}$$

Entonces, la expresión (4.19) se transformará en:

$$\int_a^{\infty} e^{-x} f(x) dx \approx e^{-a} \sum_{i=0}^n w_i f(z_i+a) \quad \dots(4.24)$$

#### IV.5.4 CUADRATURA DE GAUSS-CHEBYSHEV.

La Cuadratura de Gauss-Chebyshev se basa en los polinomios ortogonales de Chebyshev. La ecuación para esta cuadratura es:

$$\int_{-1}^1 \left[ 1/\sqrt{1-z^2} \right] F(z) dz \approx \sum_{i=0}^n w_i F(z_i) \quad \dots(4.25)$$

Donde los  $n+1$  valores  $z_i$ , son las raíces de polinomios de Chebyshev de grado  $n+1$ ,  $T_{n+1}(z)$ . Por lo cual:

$$z_i = \cos \left[ \frac{(2i+1)\pi}{(2n+2)} \right], \quad i = 0, 1, \dots, n \quad \dots(4.26)$$

En este caso, las  $w_i$  son todas iguales y tienen el valor  $\pi/(n+1)$ . Entonces la ecuación (4.25) se simplifica, quedando de la manera siguiente:

$$\int_{-1}^1 \left[ 1/\sqrt{1-z^2} \right] F(z) dz \approx \frac{\pi}{n+1} \sum_{i=0}^n F(z_i) \quad \dots(4.27)$$

Las expresiones para esta cuadratura son exactas si la función a integrar es un polinomio de grado menor o igual a  $2n+1$ , en caso contrario, el error en que se incurre está dado por:

$$E_n = \frac{2\pi}{2^{2n+2} (2n+2)!} F^{(2n+2)}(\xi) \quad \dots(4.28)$$

$$\xi \in (-1, 1)$$

Es evidente que el método de Gauss-Chebyshev es aplicable en el intervalo  $-1 \leq z \leq 1$ , para extenderlo al intervalo  $a \leq x \leq b$ , debe realizarse un cambio de variable:

$$x = \frac{z(b-a) + a + b}{2}$$

De tal manera que:

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{z(b-a) + a + b}{2}\right) dz$$

Reescribiendo la expresión anterior y considerando la función ponderadora de los polinomios de Chebyshev,  $w(x) = 1/\sqrt{1-x^2}$ , se llega a:

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 \left[1/\sqrt{1-z^2}\right] F(z) dz$$

Donde:

$$F(z) = \sqrt{1-z^2} f\left(\frac{z(b-a) + a + b}{2}\right)$$

La ecuación (4.27) puede transformarse en:

$$\int_a^b f(x) dx \approx \frac{\pi(b-a)}{2m} \sum_{l=1}^m \sqrt{1-z_l^2} f(x_l) \quad \dots(4.29)$$

Donde:

$$z_l = \cos\left[\frac{[2(i-1) + 1] \pi}{2m}\right] \quad \dots(4.30)$$

$$x_l = \frac{z_l(b-a) + a + b}{2} \quad \dots(4.31)$$

$$i = 1, 2, \dots, m$$

Aquí, "m" es el número de puntos a usar en la cuadratura, y  $x_l$  es simplemente una raíz  $z_l$ , de un polinomio de Chebyshev de m-ésimo grado transformado para el intervalo [a, b].



#### IV.5.5 CUADRATURA DE GAUSS-HERMITE.

La Cuadratura de Gauss-Hermite se basa en los polinomios ortogonales de Hermite. La expresión de esta cuadratura es:

$$\int_{-\infty}^{\infty} e^{-x^2} f(x) dx \approx \sum_{i=0}^n w_i f(x_i) \quad \dots (4.32)$$

Las  $x_i$  son las raíces de un polinomio de Hermite de grado  $n+1$ . Los valores de  $x_i$  y de  $w_i$  para  $n = 1, 2, 3$  y  $4$  (correspondientes a las fórmulas para 2, 3, 4 y 5 puntos, respectivamente) se presentan en la Tabla 4.3<sup>(2)</sup>.

raíces ( $x_i$ )	$\int_{-\infty}^{\infty} e^{-x^2} f(x) dx \approx \sum_{i=0}^n w_i f(x_i)$	factores ( $w_i$ )
$\pm 0.7071067811$	fórmula de dos puntos $n=1$	0.8862269255
$\pm 1.2247448714$ 0.000000000000	fórmula de tres puntos $n=2$	0.2954089752 1.1816959006
$\pm 1.6506801299$ $\pm 0.5246476233$	fórmula de cuatro puntos $n=3$	0.0818129354 0.8049140900
$\pm 1.0201828705$ $\pm 0.9385725646$ 0.0000000000	fórmula de cinco puntos $n=4$	0.0199592421 0.3936199292 0.9458087205

Tabla 4.3. Raíces de polinomios de Hermite,  $H_{n+1}(x)$ , y factores de peso para la Cuadratura de Gauss-Hermite.

Como en los casos anteriores, las expresiones para esta cuadratura son exactas si la función a integrar es un polinomio de grado menor o igual a  $2n+1$ , en caso contrario, el error en que se

incurrir está dado por:

$$E_n = \frac{(n+1) \int \sqrt{\pi}}{2^{2n+2} (2n+2)!} f^{(2n+2)}(\xi) \quad \dots (4.33)$$

$$\xi \in (-\infty, \infty)$$

La subrutina GAUSSQ (en lenguaje *Basic*), se incluye en el *diskette* de consulta (GAUSSQ.BAS), y permite elegir alguna de las Cuadraturas Gaussianas expuestas, para dar solución a una integral definida.

#### IV.6 CALCULO DEL POTENCIAL O PSEUDO PRESION DE LOS GASES REALES.

En 1966 Al-Hussainy y colaboradores<sup>(14)</sup>, demostraron que una integral (que en lo sucesivo denominaremos *Potencial o Pseudo Presión de los Gases Reales*), puede usarse para cálculos ingenieriles que incorporen cambios en la viscosidad y compresibilidad de un gas. El potencial de gas real está definido como:

$$m(p) = 2 \int_{p_m}^p \frac{p}{\mu(p) Z(p)} dp$$

Donde  $p_m$  es la presión base. El potencial de gas real tiene las unidades de  $[(lb/pg^2 abs)^2 / cp]$ . Para flujo isotérmico de gas, la viscosidad,  $\mu$ , y el factor de compresibilidad,  $Z$ , son función de la presión, por lo que serán los únicos parámetros involucrados en la obtención del  $m(p)$ . Entonces, empleando correlaciones de propiedades físicas del gas, la función  $m(p)$  se solucionará fácilmente.

Al-Hussainy y colaboradores, también indicaron que el potencial de gas real puede usarse en el análisis de pruebas de presión en pozos de gas; coadyuvando al cálculo de la permeabilidad y del factor de daño, por citar algunos conceptos.

El programa propuesto facilita la solución del potencial de gas real, a cualquier presión y temperatura, además, permite almacenar la información generada, en archivos para su lectura o escritura continua (archivos con extensión DAT). El lenguaje de programación adoptado es el *Basic* (*Quick Basic*). Las Cuadraturas implementadas en el programa de cómputo son las siguientes: la del Rectángulo, la del Trapecio, la de Simpson, la Adaptada (QUANCS) y Spline. El programa se presenta en el disco flexible de consulta (M(P).BAS, ver apéndice), y el diagrama de bloques en la página 166.

La ejecución del programa requiere de datos de temperatura del yacimiento, gravedad específica del gas y presión a la que se desea obtener el  $m(p)$ . Los resultados se despliegan de manera tabular, y se considerará como la solución más exacta la ofrecida por QUANCS (debido a las "bondades" que posee, consultar sección IV.4.1), entonces, el error relativo se calculará en base al resultado arrojado por dicha subrutina.

Es oportuno señalar que la subrutina QUANCS incluye una modificación, la cual consiste en el reemplazo de la función FUN por la función SEVAL (y por ende se usará la subrutina SPLINE), para evaluar, mediante una interpolación, la integral en el punto deseado; incluso en aquellas técnicas que requieren evaluar la integral en presiones intermedias, se emplea la función SEVAL para tal fin (consultar listado del programa). El factor de compresibilidad del gas,  $Z$ , se calcula mediante la correlación de Hall-Yarborough<sup>(14)</sup>, la cual contempla gases con contenidos mínimos de impurezas ( $N_2$ ,  $CO_2$  y/o  $H_2S$ ); y su viscosidad, se obtiene con la correlación de Lee y colaboradores<sup>(12)</sup>.

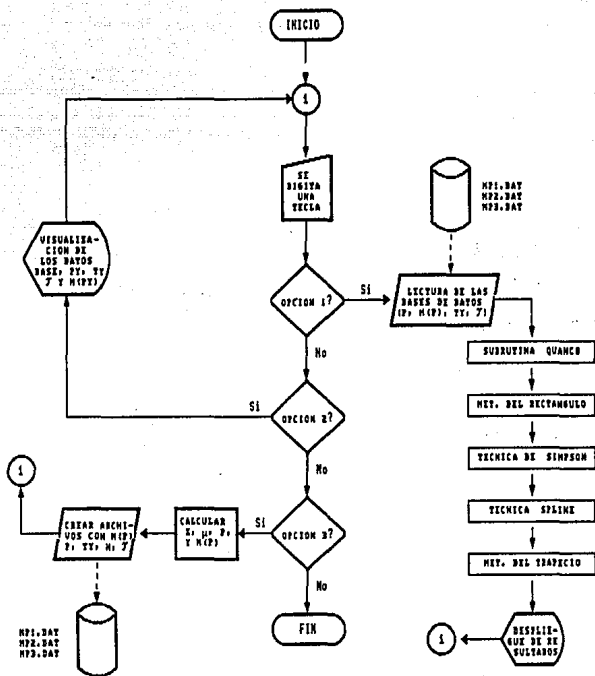


Figura 4.8. Diagrama de bloques del Programa para Obtener el Potencial de Gas Real,  $n(p)$ .

#### DATOS A CONSIDERAR.

La información usada en este ejemplo es<sup>(14)</sup>; temperatura del yacimiento de 200 °F y gravedad específica del gas igual a 0.85 (el gas está libre de impurezas).

Obtenga el potencial de gas real,  $m(p)$ , a la presión de 4400 [lb/pg<sup>2</sup>abs].

#### DISCUSION DE RESULTADOS.

El resultado publicado en el trabajo original<sup>(14)</sup>, que implica el uso de correlaciones gráficas, es el siguiente:

$$m(4400) = 1107.285 \times 10^{-6} [(lb/pg^2abs)^2/cp]$$

Los resultados logrados con el programa de cómputo se exponen en la Tabla 4.4. En ella puede apreciarse gran concordancia entre los valores obtenidos por las cuadraturas. El máximo error relativo, con respecto al valor del publicado original, lo presenta el método del Trapecio, debido a la metodología propia de la técnica (consultar sección IV.1); la magnitud de tal error es:

$$\text{error relativo} = \left| \frac{1060.002 - 1107.285}{1107.285} \right| \times 100 = 4.27 \%$$

Se puede concluir que los resultados ofrecidos por el programa son buenos, además de ser obtenidos de manera rápida y descartando el uso de correlaciones gráficas (que en muchos casos tienen una deficiente impresión). En la Figura 4.9 se muestra una gráfica de  $p$  vs.  $p/(\mu Z)$ , los datos se tomaron de las bases de datos generadas durante la ejecución del programa. El área bajo la curva representa el valor de  $m(p)$  en el intervalo de presión de 0 a 4400 [lb/pg<sup>2</sup>abs].

técnica	$m(4400 \text{ psia})$ [(lb/pg <sup>2</sup> abs) <sup>2</sup> /cp]	error relativo con respecto a QUANCE [%]
Sub. QUANCE	$1.060272 \times 10^9$	
Mét. Rectángulo	$1.060474 \times 10^9$	1.909992 E-02
Mét. de Simpson	$1.060817 \times 10^9$	4.208825 E-03
Técnica SPLINE	$1.060817 \times 10^9$	4.208825 E-03
Mét. Trapecio	$1.060002 \times 10^9$	2.544158 E-02

Tabla 4.4. Resultados del cálculo del  $m(p)$ .

## RESULTADOS DEL CALCULO DEL $m(p)$ T=200°F y S.G.=0.85

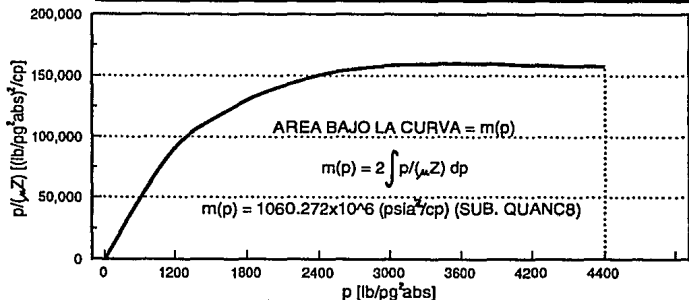


Figura 4.9.

#### IV.7 PROBLEMAS PROPUESTOS.

PPIV.1. Gill y Scher\*, modificando la fórmula de Prandtl, encontraron una expresión para la distribución de la velocidad de un fluido en una tubería de sección circular. La velocidad adimensional  $u^+$ , en la dirección  $x$ , está dada como una función de la distancia adimensional,  $y^+$ , medida a partir de la pared interna de la tubería (ver Figura 4.10), por medio de la siguiente ecuación:

$$u^+ = \int_0^{y^+} \frac{-1 + \sqrt{1 + 4cd}}{2c} dy^+ = \int_0^{y^+} \frac{2d}{1 + \sqrt{1 + 4cd}} dy^+$$

Donde:

$$c = 0.1296 (y^+)^2 (1 - e^{-\theta (y^+ / \bar{y}^+)})^2$$

Y:

$$d = 1 - \frac{y^+}{\bar{y}^+}$$

Aquí,  $\bar{y}^+$  es la distancia adimensional correspondiente al centro de la tubería (eje de la tubería), y viene dada por:

$$\bar{y}^+ = \frac{N_{Re}}{2} \sqrt{\frac{1}{f}/2}$$

Y  $\theta$ , es una función empírica de  $\bar{y}^+$ :

$$\theta = \frac{\bar{y}^+ - 60}{22}$$

\* W. N. GILL Y M. SCHER, 'A MODIFICATION OF THE MOMENTUM TRANSPORT HYPOTHESIS', A. I. CH. E. JOURNAL, 7, PÁGS. 61-65, 1961.

El número de Reynolds,  $N_{Re}$ , y el factor de fricción de Fanning,  $\bar{f}$ , son parámetros adimensionales que dependen de las propiedades físicas y velocidad del fluido, así como de las dimensiones y rugosidad de la tubería. La fórmula para el cálculo del número de Reynolds, es:

$$N_{Re} = \frac{v D \rho}{\mu}$$

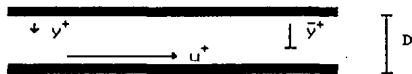


Figura 4.10. Distribución de la velocidad de un fluido.

Donde  $v$  es la velocidad media del fluido,  $D$  es el diámetro interno de la tubería, y  $\rho$  y  $\mu$  son la densidad y la viscosidad del fluido, respectivamente; todas estas variables en unidades compatibles. Para tuberías circulares, no muy rugosas, donde el rango del número de Reynolds varía de 2000 a 50000,  $\bar{f}$  puede representarse adecuadamente con la ecuación de Blásius, o sea:

$$\bar{f} = 0.079 (N_{Re})^{-0.25}$$

Realice un programa para calcular la integral

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

en base a las cuadraturas de Gauss-Chebyshev y de Gauss-Laguerre, para "n" puntos de expansión. Sustituya apropiadamente  $f(x)$  y los intervalos,  $[a, b]$ , en la expresión de la velocidad adimensional,  $u^+$ . Compare los resultados logrados con ambas cuadraturas, para  $N_{Re} = 5000, 10000, 25000$  y  $50000$ , con  $y^+ = 1, 2, 5, 10, 20, 50, 100, 200, 500$  y  $1000$ ; usando 2, 4, 6 10 y 15 puntos de expansión. Grafique en papel semi-logarítmico los resultados de  $\log y^+$  vs.  $u^+$ , obtenidos para 15 puntos y  $N_{Re} = 50000$ . Compare dicha gráfica con la presentada por Gill y Scher, y redacte comentarios al respecto.



## CAPITULO V

### APROXIMACION A LA SOLUCION DE ECUACIONES DIFERENCIALES

Aún cuando la mayoría de las ecuaciones diferenciales se resuelven al aplicar técnicas analíticas comunes, existen casos que representan fenómenos físicos que no tienen solución analítica. Afortunadamente, dichos casos pueden solucionarse mediante aproximaciones numéricas.

Por medio de ecuaciones diferenciales ordinarias puede describirse el comportamiento de innumerables procesos físicos, particularmente aquellos que están en función del tiempo (fenómenos transitorios). Por lo consiguiente, los métodos que dan solución a tales ecuaciones cobran gran importancia tanto para ingenieros como para científicos. En Ingeniería Petrolera, el empleo de ecuaciones diferenciales parciales, se enmarca principalmente en el área de Caracterización de Yacimientos (consultar sección II.1.5) Este capítulo está destinado a la descripción de las técnicas numéricas más usuales, para obtener la solución de ecuaciones diferenciales ordinarias y parciales.

#### V.1 ECUACIONES DIFERENCIALES ORDINARIAS.

Una ecuación diferencial ordinaria (EDO), de n-ésimo orden, puede representarse de la siguiente forma:

$$F\left(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \frac{d^3y}{dx^3}, \dots, \frac{d^ny}{dx^n}\right) = 0 \quad \dots(5.1)$$

La ecuación (5.1) es denominada de *n-ésimo* orden porque la máxima derivada es de orden "n", y *ordinaria* puesto que aparecen únicamente las derivadas totales (no muestra derivadas parciales o posee sólo una variable independiente, x). Una función y(x) que satisfaga esta ecuación, la cual debe ser cuando menos "n" veces diferenciable, será la solución. Para obtener una solución única (en general existen muchas funciones y(x) que satisfacen la ecuación (5.1)), es necesario contar con información adicional, es decir, valores de y(x) o de sus derivadas para algunos valores específicos de x; para una expresión de n-ésimo orden, "n" condiciones, de esa naturaleza, son suficientes para lograr una solución única, y(x). Si las "n" condiciones están referidas para el mismo valor de x ( $x_0$ , por ejemplo), entonces se tiene una *EDO con valores o condiciones iniciales*. Cuando se involucra más de un valor de x, el problema se denomina como *EDO con valores o condiciones en la frontera*.

Una EDO de n-ésimo orden puede representarse por un sistema de "n" ecuaciones de primer orden, definiéndose "n-1" nuevas variables. Considérese la siguiente expresión de segundo orden (ecuación de Bessel):

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 - p^2)y = 0$$

Donde p es una constante. Ahora, introduciendo una nueva variable,  $z = dy/dx$ , la expresión de segundo orden puede reescribirse como un par de ecuaciones de primer orden, o sea:

$$\begin{aligned} \frac{dy}{dx} - z &= 0 \\ x^2 \frac{dz}{dx} + xz + (x^2 - p^2)y &= 0 \end{aligned}$$

Como la gran mayoría de las EDO de alto orden pueden ser manejadas de manera análoga, se desarrollará solamente la solución numérica de ecuaciones de primer orden.

Una EDO de primer grado puede representarse de manera simbólica, de la siguiente manera:

$$y' = f(x,t)$$

La cual tiene una familia de curvas,  $y(t)$ , como solución. Si  $f(y, t) = y$ , para cualquier constante  $C$ , la función  $y(t) = C e^{-t}$  es la solución. Al elegir un valor inicial, digamos  $y(0)$ , se selecciona una curva de la familia, como se observa en la Figura 5.1. El valor inicial de la variable dependiente puede ser especificado para cualquier valor,  $t_0$ , de la variable independiente.

Como algunas EDO no tienen solución analítica, es usual emplear métodos numéricos para obtener una aproximación de la solución. Para esto, se requiere de la siguiente información:

- 1) Una expresión que regule el error tolerable en la solución.
- 2) Una expresión que indique la dificultad existente para lograr tal solución. Esto, normalmente, está en función del error tolerable.

Las aproximaciones que se desarrollarán en este capítulo, incluyen el uso de métodos Paso a Paso (también conocidos como métodos de Diferencias o de Variables Discretas). En estas técnicas se genera una serie de puntos  $t_0, t_1, t_2, \dots$  con un intervalo o paso, variable, o sea:

$$h_n = t_{n+1} - t_n \quad \dots (5.2)$$

En cada punto  $t_n$ , la solución,  $y(t_n)$ , se aproxima por un número  $y_n$ ,

que se calcula a partir de valores previamente obtenidos. Un método de Diferencias que provee una regla para calcular  $y_{n+1}$ , usando " $k$ " valores antecedentes  $y_n, y_{n-1}, \dots, y_{n-k+1}$ , es nombrado un método de  $k$  pasos. Si  $k = 1$ , será un método de Pasos Simples o Sencillos: y si  $k > 1$ , se denominará método Multipasos.

El método de Euler es un ejemplo de método de pasos simples, y el valor de  $y_{n+1}$  se calcula mediante una extrapolación lineal del valor anterior,  $y_n$ . Sea, por ejemplo, la siguiente EDO:

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

con condición inicial  $y(t_0) = y_0$ . La pendiente de la solución puede obtenerse para la condición inicial con la expresión  $y'_0 = f(y_0, t_0)$ . Por lo que, una aproximación de  $y_1 = y(t_1)$ , se obtiene usando los primeros dos términos de la serie de Taylor, esto es:

$$y(t_1) \approx y_1 = y_0 + h_0 f(y_0, t_0) = y_0 + y'_0 h_0$$

Análogamente para  $t_2 = t_1 + h_1$ , se tiene:

$$y(t_2) \approx y_2 = y_1 + h_1 f(y_1, t_1)$$

Y de manera general:

$$y(t_{n+1}) \approx y_{n+1} = y_n + h_n f(y_n, t_n)$$

A cada paso, la aproximación de la solución cruza de una curva de la familia solución a otra (Figura 5.2). Esta situación provoca errores considerables en la solución de algunas EDO. En el caso de la EDO  $y' = y$  (Figura 5.1), el método de Euler magnificaría el error a razón de  $e^t$ , mientras " $t$ " se incrementa. Este fenómeno se conoce como *inestabilidad de una ecuación diferencial*.

## FAMILIA DE SOLUCIONES DE $y' = y$

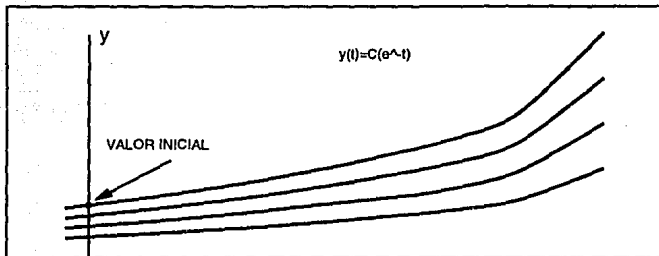


Figura 5.1.

## METODO DE EULER

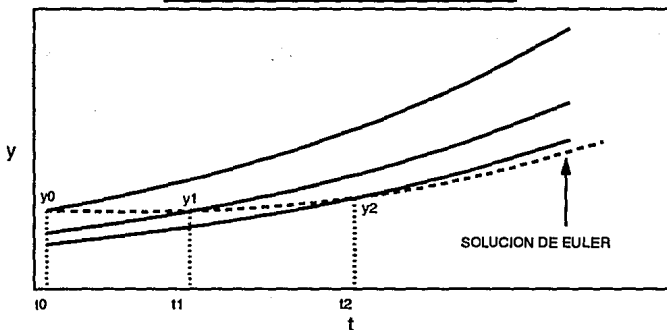


Figura 5.2.

## FAMILIA DE SOLUCIONES DE $y' = -y$

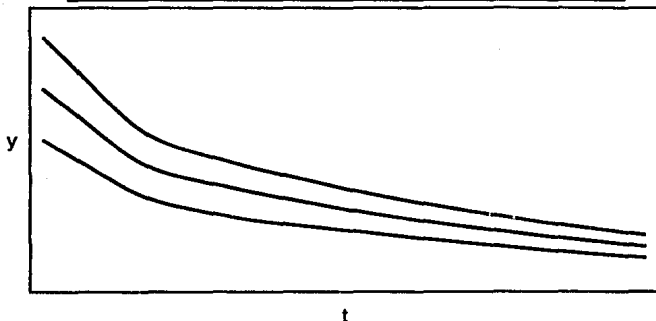


Figura 5.3.

Por otro lado, si la EDO es  $y' = -y$ , se tiene la familia de curvas mostradas en la Figura 5.3. Para dicha ecuación, los errores, decrementarán a medida que "t" disminuye. Esto se denomina *estabilidad de una ecuación diferencial*. Recapitulando, para la ecuación diferencial  $y' = \lambda y$ , donde  $\lambda$  es una constante, el error en la solución se verá afectado por el factor  $e^{\lambda t}$ , en tanto que "t" aumente. Si  $\lambda \leq 0$ , los errores no se magnificarán puesto que la ecuación se comportará establemente. Si  $\lambda > 0$ , la ecuación será inestable. Es conveniente señalar que una EDO estable, no necesariamente debe ser una expresión de sencilla solución numérica.

Los errores presentes en la aproximación de la solución de una EDO, son:

- 1) Error de redondeo.
- 2) Error de discretización.

El error de redondeo es inherente al equipo de cómputo (consultar sección I.3) y desafortunadamente es impredecible.

El error de discretización depende, exclusivamente, de la técnica empleada. Entonces, si todos los cálculos aritméticos se realizaran con precisión infinita, el único error presente sería éste. Se le subdivide en:

- Error de discretización local.
- Error de discretización global.

El error de discretización local se genera en un paso, si el valor previo es exacto y no existe error de redondeo. Para ser más claros, supóngase que  $u_n(t)$  es una función en "t", definida por:

$$u'_n = f(u_n, t)$$

$$u_n(t_n) = y_n$$

De esta manera,  $u_n(t)$  es la solución de la EDO, determinada a partir del valor  $t_n$  y no para la condición inicial original,  $t_0$ . Por lo cual, el error de discretización local,  $d_n$ , se define como la diferencia entre las soluciones, calculada y teórica, para un mismo valor  $t_n$ , o sea:

$$d_n = y_{n+1} - u_n(t_{n+1}) \quad \dots(5.3)$$

El error de discretización global es la diferencia entre la solución calculada (ignorando el redondeo), y la solución verdadera determinada para el valor  $t_0$ . Esto es:

$$e_n = y_n - y(t_n) \quad \dots(5.4)$$

Para una EDO, donde  $f(y, t)$  dependa de "y", el error en la solución para cualquier punto  $t_n$ , está regido por las soluciones

previamente calculadas. Como consecuencia, el error global será más grande que la sumatoria de los errores locales, si la EDO es inestable, y será menor que dicha sumatoria, si la EDO es estable. La estabilidad de una EDO, y por lo consiguiente el efecto de los errores local y global, está gobernada por el signo de  $\partial f/\partial y$ . Para ecuaciones no lineales,  $\partial f/\partial y$  puede alternar su signo, por lo que, la ecuación será inestable en algunas regiones y estable en otras. Para sistemas de EDO no lineales, la situación es aún más compleja.

Un concepto primordial en la evaluación de la exactitud de un método numérico, es el **orden**. El orden se define en términos del error de discretización local, obtenido cuando se aplica alguna técnica a problemas con solución sencilla. Se dice que un método es de orden "p", si existe un número "C", tal que:

$$|d_n| \leq C h_n^{p+1}$$

El número "C" depende de la derivada de la función que define la ecuación diferencial y de la longitud del intervalo sobre el cual se obtuvo la solución, y será independiente del número y del tamaño del paso, "n" y  $h_n$  respectivamente.

Una expresión para calcular el error total es:

$$\text{error total} \approx b \left[ C h + \frac{\epsilon}{h} \right]$$

donde:

$$b = t_f - t_0$$

$t_f$  = punto fijo final del intervalo de evaluación.

$\epsilon$  = error de redondeo.

El error total puede minimizarse, de manera aproximada, para el valor óptimo de "h", a través de la fórmula siguiente:

$$h_{\text{opt}} \approx \sqrt{\frac{\epsilon}{C}}$$



Finalmente, debe indicarse que conforme "h" disminuye, el error de discretización, normalmente también disminuirá, y la solución numérica comenzará a converger con respecto a la verdadera solución. No obstante, cuando el valor de "h" es excesivamente pequeño, la solución numérica divergirá a causa del error de redondeo.

#### V.1.1 TECNICAS PARA LA SOLUCION NUMERICA DE UNA ECUACION DIFERENCIAL ORDINARIA.

A pesar de que existen múltiples métodos para la aproximación de la solución de una EDO, se expondrán los tres más importantes.

##### V.1.1.1 METODO DE TAYLOR.

Si  $y(t)$  es la solución de una EDO, su expansión en términos de la serie de Taylor es:

$$y(t + h) = y(t) + hy'(t) + \frac{h^2}{2!} y''(t) + \dots$$

El método de Euler puede interpretarse como una aproximación al de Taylor (considerando sólo los dos primeros términos de la expansión). Si "y" tiene derivadas de alto orden, una técnica de orden "p", estará dada por:

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2!} y''_n + \frac{h^3}{3!} y'''_n + \dots + \frac{h^p}{p!} y_n^{(p)} \dots (5.5)$$

El primer término en despreciarse provee una estimación del error local de discretización y puede usarse para seleccionar el tamaño del paso (intervalo). Las derivadas  $y'(x)$ ,  $y''(x)$ , etcétera, pueden expresarse en términos de las derivadas parciales de la función,  $f$ . Si se hace esto, debe distinguirse la función  $f(y, t)$  con dos variables independientes, de la función  $f(y(t), t)$ , que posee sólo una variable independiente, obtenida al sustituir la solución " $y$ " en  $f$ . Iniciando con:

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

se pueden diferenciar ambos lados con respecto a " $t$ ", resultando:

$$y'' = f_y \cdot y' + f_t = f_y \cdot f + f_t$$

entonces, para la tercera diferencial:

$$y''' = f_{yy} f^2 + f_y^2 \cdot f + f_y \cdot f_t + 2f_{ty} \cdot f + f_{tt}$$

El proceso anterior es denominado *diferenciación total*. Por ejemplo, si  $f(x, t) = y^2 + t^2$ , se tendrá:

$$\begin{aligned} y' &= y^2 + t^2 \\ y'' &= 2y \cdot y' + 2t \\ y''' &= 6y^2 + 8y \cdot t^2 + 4yt + 2t^2 + 2 \end{aligned}$$

Nótese que aún cuando la función  $f$  es un polinomio de dos variables, sus derivadas totales se tornan más complicadas conforme incrementa el orden. En la práctica, este método tiene una aplicación muy limitada.

#### V.1.1.2 METODO DE RUNGE-KUTTA.

Esta técnica está diseñada para requerir solamente la evaluación de la primera derivada, lo cual representa una ventaja sobre el método de Taylor. La aproximación se logra evaluando varias veces la función  $f(y, t)$ . La expresión del método de Runge-Kutta de 4° orden, es:

$$y_{n+1} = y_n + \frac{1}{6} (k_0 + 2k_1 + 2k_2 + k_3) \quad \dots(5.6)$$

donde:

$$k_0 = hf(y_n, t_n)$$

$$k_1 = hf\left(y_n + \frac{1}{2}k_0, t_n + \frac{h}{2}\right)$$

$$k_2 = hf\left(y_n + \frac{1}{2}k_1, t_n + \frac{h}{2}\right)$$

$$k_3 = hf(y_n + k_2, t_n + h)$$

Debe resaltarse que en cada paso de cálculo, se realizan cuatro evaluaciones de la función  $f(y, t)$ . El método de Runge-Kutta es una extensión de la cuadratura de Simpson para EDO, esto es:

$$\int_{t_n}^{t_{n+1}} f(t) dt \approx \frac{h}{6} \left[ f(t_n) + 4f\left(\frac{t_n + t_{n+1}}{2}\right) + f(t_{n+1}) \right]$$

Si  $f(y, t)$  es función exclusiva de "t", ambas técnicas serán equivalentes.

Como en las técnicas de Runge-Kutta no es factible obtener el error local de discretización, se imposibilita la selección del tamaño del intervalo, sin embargo, son fácilmente programables y

numéricamente estables para casi toda EDO, además, puede iniciarse por sí mismo, puesto que sólo necesita una solución inicial  $y_0$ , para calcular  $y_{n+1}$ . En cualquier momento del cálculo de la aproximación, el tamaño del intervalo  $h_n$  es susceptible de modificarse (la subrutina RKF45, descrita en la sección V.2.5, incluye una moderna modificación que permite el control del tamaño  $h_n$ ).

### V.1.1.3 METODO MULTIPASOS.

En los métodos previamente expuestos,  $y_{n+1}$  se calcula a partir de una función que depende de  $y_n$ ,  $t_n$  y  $h_n$ . Es coherente pensar que empleando los valores de  $y_{n-1}$ ,  $y_{n-2}$ , ..., y  $f_{n-1}$ ,  $f_{n-2}$ , ..., donde  $f_i = f(y_i, t_i)$ , la aproximación obtenida sería más exacta. Los métodos Multipasos se basan en esta premisa y son sumamente efectivos. Para soluciones con gran exactitud, requieren menor cantidad de cálculos que las técnicas de Un-paso.

La fórmula general del método Multipasos (k-pasos) Lineal, es la siguiente:

$$y_{n+1} = \sum_{l=0}^k \alpha_l y_{n+1-l} + h \sum_{l=0}^k \beta_l f_{n+1-l} \quad \dots(5.7)$$

donde "k" es un número entero, y tanto  $\alpha_l$  como  $\beta_l$  son distintos de cero. La técnica se denomina lineal pues cada  $f_l$  se comporta linealmente, aunque,  $f$  puede no ser una función lineal de sus argumentos.

Una vez que el método ha iniciado, cada paso subsecuente requiere del cálculo de  $y_{n+1}$ , a partir de los valores, previamente

obtenidos o conocidos.  $y_n, y_{n-1}, \dots, y_{n-k+1}, f_n, f_{n-1}, \dots, f_{n-k+1}$ . Si  $\beta_0 = 0$ , la técnica se denomina explícita. Cuando  $\beta_0 \neq 0$  la técnica será implícita, ya que debe calcularse  $f_{n+1}$  para obtener  $y_{n+1}$ .

Un método Multipasos Predictor-Corrector, emplea una técnica explícita (predictor), seguida por una o más aplicaciones de una implícita (corrector). El método de Adams de 4° orden es un buen ejemplo de esta técnica, sus ecuaciones son:

ecuación predictora

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \quad \dots (5.8)$$

ecuación correctora

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) \quad \dots (5.9)$$

El algoritmo para el método Predictor-Corrector es como sigue:

- 1) Use la ecuación predictora para calcular  $y_{n+1}^{(0)}$ , como una aproximación inicial de  $y_{n+1}$ . Considere  $i = 0$ .
- 2) Obténgase y evalúese la derivada de la función, o sea:
 
$$f_{n+1}^{(i)} = f(y_{n+1}^{(i)}, t_{n+1})$$
- 3) Calcule una mejor aproximación  $y_{n+1}^{(i+1)}$  usando la ecuación correctora, para  $f_{n+1} = f_{n+1}^{(i)}$ .
- 4) Si  $|y_{n+1}^{(i+1)} - y_{n+1}^{(i)}| > \text{tolerancia}$ , incremente "i" en una unidad y regrese al punto 2; en otro caso,  $y_{n+1} = y_{n+1}^{(i+1)}$ , y el problema se da por resuelto.

En este algoritmo se ejecutan tres procesos independientes: el paso predictor (1), denominado P; el paso evaluador (2), llamado E; y el paso corrector (3) o paso C. El 4° paso puede sustituirse por un ciclo de "m" iteraciones correctoras. El esquema resultante se expresa en forma compacta de la siguiente forma:

$$P(EC)^m$$

Por lo que si se consideran la estabilidad y el error de

discretización, el método Predictor-Corrector de Adams, más sensible, será<sup>(1)</sup>:

FECE

#### V.1.1.4 PROBLEMA CON VALORES EN LA FRONTERA.

Hasta aquí se han expuesto EDO con valores iniciales, en las cuales la variable dependiente está disponible para algún valor de la variable independiente,  $t$ . En ocasiones es necesario resolver EDO, sujetas a los valores de la variable dependiente, obtenidos para diferentes valores de " $t$ ". Un problema de esa naturaleza es conocido como *problema con valores en la frontera*. Las técnicas para dar solución a estos problemas son totalmente disímiles de aquellas para problemas con valor inicial.

A continuación se expondrá un método para reducir un problema con valores en la frontera a una secuencia de problemas con valor inicial. Supóngase el sistema de EDO mostrado enseguida:

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

donde:

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

Sujeto a las condiciones  $y_1(0) = \alpha$  y  $y_2(b) = \beta$ ,  $b \neq 0$ .

Para solucionar tal sistema, se sugiere el siguiente método iterativo:

- 1) Elija un valor  $\xi$  que aproxime el valor de  $y_2(0)$ .

2) Resuelva el problema de valor inicial:

$$\hat{y}' = f(\hat{y}, t), \quad \hat{y}(0) = \begin{pmatrix} \alpha \\ \zeta \end{pmatrix}$$

3) Si  $|\hat{y}_2(b) - \beta| \leq \text{tolerancia}$ , entonces  $y(t) \approx \hat{y}(t)$ ; de otra forma ajuste  $\zeta$  y retorne al paso 2.

Como información adicional, se comentará que el valor de  $\zeta$ , está íntimamente relacionado con el concepto de región de convergencia, del método de Newton-Rhapson para sistemas de ecuaciones no lineales.

#### V.1.1.5 SUBRUTINA RKF45.

RKF45 es una subrutina, basada en la técnica de Runge-Kutta, para resolver una EDO con condiciones iniciales. Se requiere evaluar la función involucrada en seis ocasiones por cada paso de cálculo (para esto debe adicionarse un subprograma a modo de función). Cuatro de esos valores funcionales se combinan con un juego de coeficientes (propuestos por E. Fehlberg<sup>(1)</sup>), para generar un método de 4° orden, y uno de 5° orden se desarrolla al conjugar los seis valores funcionales con otro grupo de coeficientes. La comparación entre ambas cantidades proporciona una aproximación del error, valor con el cual se logra el control del tamaño del intervalo o paso,  $h_n$ . Como ya se indicó, las fórmulas usadas por RKF45 necesitan seis valores de la función involucrada por paso,  $k_1$  a  $k_6$ . Ellos están definidos por:

$$k_i = h_n f\left(y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j, t_n + \alpha_i h_n\right) \quad i = 1, 2, \dots, 6 \quad \dots (5.10)$$

El nuevo valor  $y_{n+1}$  se obtiene usando una combinación ponderada de las 6 cantidades  $k_i$ , es decir:

$$y_{n+1} = y_n + \sum_{i=1}^6 \gamma_i k_i \quad \dots(5.11)$$

Existen 27 coeficientes en total, 6 para  $\alpha_i$ , 15 para  $\beta_i$  y 6 para  $\gamma_i$ . La determinación de dichos coeficientes se logra al expandirse todas las  $k_i$  en series de Taylor, las expansiones se sustituyen en la fórmula para  $y_{n+1}$ , y el resultado se compara con la serie de Taylor para la solución real (definida en la sección V.1),  $u_n(t_{n+1})$ . Fehlberg también encontró un juego de coeficientes  $\gamma_i^*$ , que combinados con 4 de los 6 valores de  $k_i$ , producen una cantidad  $y_{n+1}^*$ , la cual brinda mayor precisión a la técnica de 4° orden. Esto es:

$$y_{n+1}^* = y_n + \sum_{i=1}^6 \gamma_i^* k_i \quad \dots(5.12)$$

RKF45 no calcula  $y_{n+1}^*$ , en su lugar, calcula una estimación del error,  $\sum_{i=1}^6 (\gamma_i - \gamma_i^*) k_i$ , la que sirve para regular el tamaño del paso,  $h_n$ .

Como corolario, se dirá que RKF45 es la implementación, de propósito general, más efectiva del método de Runge-Kutta, además, es fácil de entender y usar. Por si esto fuera poco, para demandas modestas de precisión (error de discretización local menor de  $10^{-5}$  ó  $10^{-6}$ ), RKF45 es tan eficiente como la más sofisticada técnica de Adams. En el *diskette* de consulta se presenta, codificada en lenguaje *Basic*, la subrutina RKF45 así como un programa fuente para su utilización (RKF45.BAS, revisar apéndice).



## V.2 ECUACIONES DIFERENCIALES PARCIALES.

Dentro del conjunto de las ecuaciones diferenciales parciales (EDP), las de 2° orden y forma parabólica revisten mayor interés para el Ingeniero Petrolero.

Las EDP de 2° orden, se clasifican como de tipo elíptico, hiperbólico o parabólico. Tal clasificación es posible si la ecuación puede reducirse, mediante transformaciones de las variables independientes, a la forma mostrada a continuación:

$$\sum_{i=1}^n A_i \frac{\partial^2 u}{\partial x_i^2} + \sum_{i=1}^n B_i \frac{\partial u}{\partial x_i} + Cu + D = 0 \quad \dots (5.13)$$

Los coeficientes  $A_i$ , evaluados en el punto  $(x_1, x_2, \dots, x_n)$ , pueden ser -1, 1 ó 0. La variable dependiente es "u" y la independiente es  $x_i$ . Los tipos principales de EDP, son:

- 1) Si todas las  $A_i \neq 0$  y tienen el mismo signo, la EDP es elíptica.
- 2) Si todas las  $A_i \neq 0$  y tienen, a excepción de una, el mismo signo, la EDP es hiperbólica.
- 3) Si alguna  $A_i = 0$  ( $A_k$ , por ejemplo), y las restantes  $A_i$  son distintas de cero y con igual signo, y si además, el coeficiente  $B_k$  de  $\partial u / \partial x_k$  es diferente de cero, la EDP es parabólica.

Como los coeficientes  $A_i$ ,  $B_i$ , C y D son función de las variables independientes  $x_1, x_2, \dots, x_n$ , la clasificación de una EDP puede variar de acuerdo al punto en particular, que esté siendo

considerado, en el espacio  $(x_1, x_2, \dots, x_n)$ . Es muy común, que una de las variables independientes sea el tiempo,  $t$ , y que las restantes sean coordenadas de distancia,  $x$ ,  $y$ , y  $z$ . Enseguida se desarrollará el método de Diferencias Finitas, que es el más "socorrido" para dar solución a una EDP.

### V.2.1 METODO DE DIFERENCIAS FINITAS.

La idea básica de cualquier método de aproximación, consiste en reemplazar el problema original por otro que sea más fácil de resolver, y cuya solución se aproxime a la del problema original.

El Método de Diferencias Finitas (MDF), tiene como principio fundamental el sustituir las derivadas parciales, que intervienen en la EDP de un problema en particular, por su aproximación en Diferencias Finitas. Considérese la siguiente ecuación:

$$\frac{\partial U}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{\Delta U}{\Delta x} = \frac{\Delta U}{\Delta x} \quad \dots(5.14)$$

En su aproximación, el valor asignado a  $\Delta x$  dependerá de cada problema. El MDF ejemplificado anteriormente, permite transformar un problema de Cálculo Diferencial en uno algebraico que será más fácil de solucionar.

El pilar fundamental en el desarrollo del MDF lo constituye la serie de Taylor. La expansión de  $f(x)$  en términos de tal serie, con respecto al punto " $x$ " y con incrementos positivos, puede expresarse como:

$$f(x + \Delta x) = f(x) + \Delta x \frac{df}{dx} + \frac{\Delta x^2}{2!} \frac{d^2 f}{dx^2} + \dots + \frac{\Delta x^n}{n!} \frac{d^n f}{dx^n} \quad \dots(5.15)$$

Y con incrementos negativos, se convierte en:

$$f(x + \Delta x) = f(x) - \Delta x \frac{df}{dx} + \frac{\Delta x^2}{2!} \frac{d^2 f}{dx^2} - \dots + \frac{\Delta x^n}{n!} \frac{d^n f}{dx^n} \quad \dots(5.16)$$

Empleando las dos expansiones desarrolladas, pueden derivarse aproximaciones en Diferencias Finitas para la primera y segunda derivadas de  $f(x)$ . Al despejar  $df/dx$  de la ecuación (5.15), se obtiene:

$$\frac{df}{dx} = \frac{f(x + \Delta x) - f(x)}{\Delta x} - \frac{1}{\Delta x} \left[ \frac{\Delta x^2}{2!} \frac{d^2 f}{dx^2} + \dots + \frac{\Delta x^n}{n!} \frac{d^n f}{dx^n} \right] \quad \dots(5.17)$$

Asumiéndose que  $\Delta x$  es pequeño, los términos dentro del paréntesis pueden despreciarse (tienen valores mínimos), además, si se denotan con  $\theta(x)$ , la ecuación (5.17) se transforma en:

$$\frac{df}{dx} = \frac{f(x + \Delta x) - f(x)}{\Delta x} + \theta(\Delta x) \quad \dots(5.18)$$

El término  $\theta(\Delta x)$  se lee como "de orden  $\Delta x$ ", pues precisamente  $\Delta x$  es la parte principal del primer término, es decir, el más significativo (el de mayor valor), de entre todos los que conforman la serie que lo define.

En conclusión, al sustituir la derivada de la función  $f(x)$  por  $(f(x+\Delta x) - f(x))/\Delta x$ , se cometerá un error de truncamiento del orden  $\Delta x$ ,  $\theta(\Delta x)$ .

Otra aproximación de  $df/dx$ , se obtiene a partir de la ecuación (5.16), esto es:

$$\frac{df}{dx} = \frac{f(x) - f(x - \Delta x)}{\Delta x} \quad \text{con un error de orden } \theta(\Delta x) \quad \dots(5.19)$$

Una tercera aproximación, con error de truncamiento menor, puede ser formulada al restar de la expresión (5.15) la (5.16), o sea:

$$\frac{df}{dx} = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} + \theta \langle (\Delta x)^2 \rangle \quad \dots(5.20)$$

Estas tres aproximaciones se conocen como Diferencias hacia Adelante, Diferencias hacia Atrás y Diferencias Centrales, respectivamente.

Por otro lado, si se suman las expresiones (5.15) y (5.16), se obtendrá una aproximación para la segunda derivada:

$$\frac{d^2 f}{dx^2} = \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x))}{(\Delta x)^2} + \theta \langle (\Delta x)^2 \rangle \quad \dots(5.21)$$

En caso de derivadas parciales, la aproximación puede obtenerse directamente de las ecuaciones (5.18), (5.19), (5.20) y (5.21); por ejemplo, para la segunda derivada parcial, se tiene:

$$\frac{\partial^2 f}{\partial x^2} = \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x))}{(\Delta x)^2} + \theta \langle (\Delta x)^2 \rangle \quad \dots(5.22)$$

Realizando un procedimiento similar, pueden aproximarse derivadas de mayor grado.

#### V.2.1.1 ESQUEMA EXPLÍCITO.

Hasta esta parte puede concluirse que el MDF facilita la solución numérica de ecuaciones diferenciales, revisemos ahora, la metodología de su aplicación. Supóngase la siguiente EDP con sus

respectivas condiciones iniciales y de frontera:

$$\frac{\partial^2 p}{\partial x^2} = \frac{\partial p}{\partial t} \quad \dots(5.23)$$

$$\left. \begin{aligned} p(0, t) &= 0 \\ p(1, t) &= 0 \end{aligned} \right\} \text{condiciones de frontera}$$

$$p(x, 0) = 0 \quad \text{condición inicial}$$

El problema consiste en calcular la presión a cualquier distancia,  $x$ , y tiempo,  $t$ , de interés. Sustituyendo las derivadas parciales por sus respectivas diferencias finitas, se tendrá:

$$\frac{p_{i+1}^n - 2p_i^n + p_{i-1}^n}{(\Delta x)^2} + \theta((\Delta x)^2) = \frac{p_i^{n+1} - p_i^n}{\Delta t} + \theta(\Delta t) \quad \dots(5.24)$$

Nótese que  $f(x+\Delta x)$  se substituyó, para manejo de nomenclatura, por  $p_{i+1}$ , donde los subíndices manejan distancia y los superíndices tiempo. El superíndice "n" representa el tiempo actual y el "n+1" el tiempo al cual se predice la presión. Si los términos  $\theta((\Delta x)^2)$  y  $\theta(\Delta t)$  se desprecian, la aproximación de la EDP se reduce a:

$$\frac{p_{i+1}^n - 2p_i^n + p_{i-1}^n}{(\Delta x)^2} = \frac{p_i^{n+1} - p_i^n}{\Delta t} \quad \dots(5.25)$$

#### V.2.1.2 ERRORES DE CONVERGENCIA Y ESTABILIDAD EN EL ESQUEMA EXPLICITO.

El error en la aproximación numérica puede definirse como la

diferencia entre la solución exacta,  $P_i^n$ , y la calculada,  $Q_i^n$ , o sea:

$$e_i^n = P_i^n - Q_i^n \quad \dots(5.26)$$

Con la finalidad de cuantificar la magnitud de tal error y de definir los errores de convergencia, manipularemos algebraicamente algunas de las expresiones anteriores. Restando la ecuación (5.25) de la (5.24), se obtiene:

$$\frac{e_i^{n+1} - e_i^n}{\Delta t} = \frac{e_{i+1}^n - 2e_i^n + e_{i-1}^n}{(\Delta x)^2} + \theta((\Delta x)^2 + (\Delta t))$$

Haciendo  $r = \Delta t/(\Delta x)^2$  y despejando a  $e_i^{n+1}$ :

$$e_i^{n+1} = r e_{i+1}^n + (1 - 2r)e_i^n + r e_{i-1}^n + \theta((\Delta t)^2 + \Delta t(\Delta x)^2)$$

Ahora, asignando valores absolutos en ambos miembros de la expresión anterior y considerando que "r" está definida en el intervalo [0, 1/2], se tiene que:

$$|e_i^{n+1}| \leq r|e_{i+1}^n| + (1 - 2r)|e_i^n| + r|e_{i-1}^n| + c((\Delta t)^2 + \Delta t(\Delta x)^2) \quad \dots(5.27)$$

Donde el elemento  $c((\Delta t)^2 + \Delta t(\Delta x)^2)$  resulta de la definición siguiente: Un término A se denomina de orden  $(\Delta x)^n$  o  $\theta(\Delta x)^n$ , si existe una constante, c, positiva e independiente de  $\Delta x$ , tal que  $A < c(\Delta x)^n$ . Si esto se cumple, tanto  $(\Delta x)^n$  como A tenderán a cero, al menos con la misma velocidad.

Por otro lado, el máximo error al tiempo "n" puede estimarse con la siguiente expresión:

$$|e^n| = \max |e_i^n|, \quad i = 0, 1, \dots, n$$

Entonces, tomando en cuenta el máximo error, la ecuación (5.27) se reduce a:

$$|e^{n+1}| \leq m \Delta x |e^n| + c((\Delta t)^2 + \Delta t(\Delta x)^2) \quad \dots(5.28)$$

Reduciendo un paso en el tiempo, la desigualdad anterior puede transformarse en:

$$|e^n| \leq m \Delta x |e^{n-1}| + c((\Delta t)^2 + \Delta t(\Delta x)^2)$$

Misma que al sustituirse en la expresión (5.28), resulta:

$$|e^{n+1}| \leq |e^{n-1}| + 2c((\Delta t)^2 + \Delta t(\Delta x)^2)$$

Continuando con reducciones sucesivas en el tiempo, se llega a:

$$|e^{n+1}| \leq |e^0| + mc((\Delta t)^2 + \Delta t(\Delta x)^2)$$

Además, como  $e^0 = 0$ , por las condiciones iniciales y de frontera, el máximo error al tiempo "n+1", queda definido por la relación:

$$|e^{n+1}| \leq mc((\Delta t)^2 + \Delta t(\Delta x)^2) \quad \dots(5.29)$$

Se dice que un esquema en diferencias finitas es convergente, si en un punto  $(x, t)$  se cumple que:

$$\lim_{\substack{\Delta x \rightarrow 0 \\ \Delta t \rightarrow 0}} (P_{i,j} - Q_{i,j}) = 0 \quad \dots(5.30)$$

Con base en la expresión (5.29) y en la definición (5.30), se concluye que el esquema explícito es convergente si  $r \leq 1/2$ , es decir,  $(\Delta x)^2 > 2\Delta t$ . En otras palabras, si se restringe el cociente  $\Delta t/(\Delta x)^2$  a valores menores de 1/2, la presión pronosticada poseerá un grado de exactitud aceptable.

Otro punto relevante en la aplicación del MDF, lo constituye el error de estabilidad. Un sistema es inestable si el error aumenta conforme se incrementa el tiempo; esto, en notación algebraica, es:

$$\frac{e^{n+1}}{e^n} \leq 1 \Rightarrow \text{sistema estable} \quad \dots(5.31)$$

$$\frac{e^{n+1}}{e^n} > 1 \Rightarrow \text{sistema inestable}$$

Recapitulando, la ecuación (5.25) es un desarrollo explícito, en diferencias finitas, de la expresión (5.23), condicionalmente convergente y estable, para valores de  $\Delta t/(\Delta x)^2 < 1/2$ . El esquema se denomina **explícito**, ya que es posible el despeje de la única incógnita ( $p_i^{n+1}$ ), esto es:

$$p_i^{n+1} = \frac{\Delta t}{(\Delta x)^2} \left( p_{i+1}^n - 2p_i^n + p_{i-1}^n \right) + p_i^n \quad \dots(5.32)$$

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

Para emplear la ecuación (5.32) en un caso práctico, es necesario contemplar las restricciones establecidas para  $\Delta x$  y  $\Delta t$ . Por ejemplo, puede ser conveniente elegir incrementos pequeños en el espacio, sin embargo, ello conlleva a la utilización de incrementos en el tiempo más pequeños aún, con el consecuente incremento del tiempo de cómputo; y esto no es deseable, sobre todo cuando el equipo electrónico usado es rentado o "prestado".

### V.2.1.3 ESQUEMA IMPLICITO.

Si en la ecuación (5.23) la derivada espacial ( $\partial x$ ) es sustituida por diferencias finitas al tiempo "n+1", y no al tiempo "n" como en





#### V.2.1.4 ERRORES DE CONVERGENCIA Y ESTABILIDAD EN EL ESQUEMA IMPLICITO.

El error máximo de este esquema puede obtenerse mediante la siguiente expresión:

$$e = \frac{1}{1 + r[2 - \cos(\Delta x(n\pi/L))]} \quad \dots(5.35)$$

donde "n" es el número de celdas y L es la longitud total en estudio, además,  $|e| \leq 1$  para todo valor de "r", lo cual asegura la estabilidad del Esquema Implícito.

En base al Teorema de Equivalencia de Lax ("Dado un problema de valor inicial propiamente definido, y dada su aproximación por diferencias finitas, estabilidad es la condición necesaria y suficiente para la convergencia"), este esquema es convergente. Por lo tanto, el Esquema Implícito, representado por la ecuación (5.34), es incondicionalmente estable y convergente, ya que el error de aproximación no depende de  $\Delta x$  y/o de  $\Delta t$ .

#### V.2.1.5 ESQUEMAS PONDERADOS.

Un esquema de solución más general se obtiene al considerar familias de diferencias finitas a través de promedios ponderados. El esquema en diferencias finitas de la EDP (5.23), puede expresarse como:

$$\frac{\theta \delta^2 x p_l^{n+1} + (1 - \theta) \delta^2 x p_l^n}{(\Delta x)^2} = \frac{p_l^{n+1} - p_l^n}{\Delta t} \quad \dots(5.36)$$

donde:

$$\theta^2 x p_i^n = p_{i+1}^n - 2p_i^n + p_{i-1}^n \quad \dots(5.37)$$

En la ecuación (5.36), el error es del orden de  $\theta((\Delta x)^2 + (\Delta t)^2)$ . Debe observarse que el esquema es explícito si  $\theta = 0$  y totalmente implícito si  $\theta = 1$ .

#### 5.2.1.6 ERRORES DE CONVERGENCIA Y ESTABILIDAD EN EL ESQUEMA PONDERADO.

El esquema ponderado es condicionalmente estable y convergente de acuerdo con:

$$r \leq \frac{1}{2(1 - 2\theta)} \quad 0 \leq \theta \leq 1/2 \quad \dots(5.38)$$

e incondicionalmente estable y convergente para valores dentro del rango  $1/2 \leq \theta \leq 1$ . En la práctica, los valores de  $\theta$  más empleados son 0, 1/2 y 1. En el caso particular de que  $\theta = 1/2$ , el esquema implícito incondicional conformado es denominado como Esquema de Crank-Nicolson.

En resumen, con los esquemas implícitos se logra mejorar las condiciones de estabilidad, al introducir sistemas incondicionales, y obtener mayor exactitud al disminuir el orden del error.

### V.2.1.7 TIPOS DE CONDICIONES DE FRONTERA.

Cuando las condiciones de frontera son constantes, como en el caso estudiado, la EDP se clasifica como de tipo *Dirichlet* o de primera clase. Si tales condiciones están dadas en términos de derivadas parciales de la función incógnita, la EDP se clasifica como de tipo *Neumann* o de segunda clase. La combinación de estas condiciones de frontera genera una tercera clasificación, las EDP de tipo *Robin* o de tercera clase.

### V.2.1.8 ECUACIONES DIFERENCIALES PARCIALES EN DOS DIMENSIONES.

El desarrollo de la aproximación de una EDP en dos dimensiones, se discute ampliamente en el capítulo II, sección II.1.5, "Solución al problema de flujo bidireccional, monofásico, transitorio, a través de medios porosos".

### V.3 EJERCICIOS RESUELTOS.

PV-1. La función error está definida por la integral mostrada enseguida:

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

Pero también puede expresarse como la solución de una ecuación diferencial<sup>(4)</sup>, esto es:

$$y'(x) = \frac{2}{\sqrt{\pi}} e^{-x^2}$$

cuya condición inicial es  $y(0) = 0$ .

Realice un programa de cómputo que empleando la subrutina RKF45, despliegue una tabla de  $\text{erf}(x)$  para  $x = 0.0, 0.1, 0.2, \dots, 0.9, 1.0$ . Compare sus resultados con los obtenidos en el capítulo I (sección I.5. problema resuelto PI-1).

Solución:

El programa elaborado se presenta en el disco flexible de consulta (RKF45.BAS, subdirectorío CAP5, revisar el apéndice). Algunos resultados se consignan en la Tabla 5.1. Se observa que presentan un error relativo máximo del 34 %, con respecto a los valores tabulados de  $\text{erf}(x)$ . El tiempo total de ejecución del programa fue de 1.16 segundos.

x	erf(x)			
	subrutina RKF45	serie de Taylor	artificio Forsythe	valores tabulados
0.0	0.0	0.0	0.0	0.0
0.5	0.5641896	0.5205002	0.5204999	0.6915
1.0	1.1283790	0.8426992	0.8427007	0.8418

Tabla 5.1. Resultados del cálculo de  $\text{erf}(x)$  mediante RKF45.

#### V.4 PROBLEMAS PROPUESTOS.

PPV-1. N. Muskat desarrolló un procedimiento para predecir el comportamiento de yacimientos homogéneos, con distribución uniforme de presión, sin entrada de agua y sin segregación gravitacional de fluidos. La ecuación de dicho método es diferencial ordinaria de primer grado, y expresa la variación de la saturación de aceite a

cualquier presión:

$$\frac{dS_o}{dp} = \frac{S_o \left( X_p + \frac{k_g}{k_o} Y_p \right) - Z_p E_g}{1 + \left( \frac{k_g}{k_o} \right) \left( \frac{\mu_o}{\mu_g} \right)}$$

Donde:

$$X_p = \frac{E_g}{E_o} \frac{dR_o}{dp}$$

$$Y_p = \frac{\mu_o}{\mu_g} \frac{1}{E_o} \frac{dE_o}{dp}$$

$$Z_p = \frac{1}{E_g} \frac{dE_g}{dp}$$

Elabore un programa de cómputo que empleando RKF45 dé solución a estas ecuaciones.

\* GARAYOCHEA P. F. Y BASHBUSH B. J. L. 'APUNTES DE COMPORTAMIENTO DE LOS YACIMIENTOS', FACULTAD DE INGENIERÍA, U.N.A.M., 1987.

## CAPITULO VI

### TECNICAS AVANZADAS DE PROGRAMACION APLICABLES AL AMBITO PETROLERO

El objetivo de este capítulo, es proporcionar algunos conceptos de uso cotidiano en la programación de computadoras personales dentro de la Industria Petrolera.

En la primera parte se presentan los programas de cómputo SIMPP y REGRESML, que son útiles en la simulación del comportamiento de presión en pruebas de decremento y en el análisis de regresión lineal, respectivamente.

El examen minucioso de estos programas permitirá obtener un conocimiento de las técnicas básicas para el manejo de "menús interactivos", ayudas de uso de programa, graficación en pantalla, métodos estadísticos, y otros tópicos de interés, tales como el empleo de una antitransformada del plano de Laplace al plano Real para resolver ecuaciones diferenciales parciales (programa SIMPP).

Posteriormente, se hace una breve exposición de las ventajas, metodología de construcción, uso y mantenimiento de programas de biblioteca de *Quick Basic*; se hace énfasis en este lenguaje por su utilización generalizada en la UNAM, PEMEX e IMP. Además, se incluyen conceptos asociados con el "diseño óptimo de programas de cómputo", los cuales se consideran de gran utilidad para el desarrollo adecuado de programas computacionales.

## VI.1 PROGRAMAS REGRESML Y SIMPP.

El programa REGRESML contiene un modelo de regresión múltiple, que apoyado en técnicas de álgebra matricial, permite realizar ajustes de ecuaciones a conjuntos de datos usando el método de Mínimos Cuadrados, además, mediante procedimientos estadísticos se puede evaluar la *bondad* de dicho ajuste.

Frecuentemente, en el área de Ingeniería Petrolera se presentan problemas en los cuales dos o más variables están relacionadas y es necesario investigar la relación matemática que las une. REGRESML calcula los coeficientes del modelo de regresión propuesto por el usuario a un conjunto de datos, también efectúa cálculos estadísticos que incluyen análisis de varianza. Es posible realizar gráficas de las estadísticas de regresión para analizar en forma rápida el ajuste del modelo matemático al conjunto de datos.

Antes de predecir a partir de un modelo de regresión múltiple, es necesario tener la certeza de que los parámetros calculados para el modelo sean estadísticamente significativos. Un parámetro no significativo debe suponerse igual a cero y eliminarse. Pueden emplearse varias pruebas estadísticas si los errores están normalmente distribuidos, tales como la "F de Fisher" o la "t de Student" (las cuales están programadas en REGRESML). Es decir, no se busca únicamente una función matemática que nos diga de qué manera están relacionadas las variables, sino que se desea saber con qué precisión se puede predecir el valor de la variable dependiente en función de variables independientes conocidas. Las técnicas utilizadas para lograr este objetivo se conocen como *Métodos de Correlación*.

Los métodos de regresión se usan para determinar la mejor relación funcional entre las variables, mientras que los métodos de



correlación se utilizan para medir el grado de asociación de las distintas variables.

REGRESML presenta un *ambiente conversacional*, y ofrece: ayudas que pueden invocarse con la tecla F1, resultados en forma de tablas y gráficas (en pantalla), y maneja la ejecución del programa mediante opciones que se despliegan en forma de menús. El análisis detallado de este programa facilita la selección de procedimientos que posteriormente puedan ser utilizados como subrutinas en otros proyectos.

En lo referente al programa SIMFP, este constituye un simulador de presión que permite calcular pruebas de decremento en base a información de las características de la formación productora, del fluido producido y del tipo de terminación del pozo. El programa utiliza el *algoritmo de Stehfest* para realizar la antitransformada de Laplace (del plano de Laplace al Real), de la solución de la ecuación diferencial que caracteriza el flujo radial laminar, en un yacimiento homogéneo e isótropo, de espesor constante, conteniendo un fluido ligeramente compresible, con flujo isotérmico, considerando el efecto de almacenamiento de pozo y el factor de daño, esto es<sup>(19)</sup>:

$$L\{p_{VD}\} = \frac{K_0(\sqrt{p}) + s\sqrt{p} K_1(\sqrt{p})}{p[\sqrt{p} K_1(\sqrt{p}) + \bar{C}p\{K_0(\sqrt{p}) + s\sqrt{p} K_1(\sqrt{p})\}]}$$

Donde:

- L = Transformada de Laplace de un argumento.
- $p_{VD}$  = Caída de presión adimensional.
- $K_0$  = Función Bessel modificada de segunda clase.
- $K_1$  = Función Bessel modificada de primera clase.
- s = Factor de daño.
- p = Variable de la transformada de Laplace.
- $\bar{C}$  = Coeficiente adimensional de almacenamiento del pozo.

Las técnicas para realizar antitransformadas de Laplace se usan con mucha frecuencia debido a que es mucho más fácil obtener las soluciones de Ecuaciones Diferenciales en el plano de Laplace, y además, a que no siempre se tienen soluciones manejables computacionalmente en el plano Real; en el caso de la solución programada en SIMPP se requiere menos de un minuto para obtener la simulación de una prueba con 30 puntos, pero se emplea más de media hora para lograr el mismo objetivo si se utiliza la solución en el plano Real, puesto que están involucradas integrales definidas numéricamente (de difícil evaluación).

A partir del uso de SIMPP es posible diseñar eficientemente pruebas de decremento de presión, a fin de lograr el mayor provecho en la caracterización del yacimiento. En el artículo "Diseño de Pruebas de Incremento de Presión"<sup>(20)</sup>, se presentan ejemplos de aplicación de este tipo de simuladores a casos prácticos de campo, por lo cual se recomienda su lectura.

La técnica de Steffest, de acuerdo con publicaciones especializadas, ha tenido éxito al usarse en el área de pruebas de presión en yacimientos, pero es importante mencionar que esta metodología no siempre reporta buenos resultados con todas las ecuaciones diferenciales, en cuyo caso deberán utilizarse técnicas alternas.

Los diagramas de flujo simplificados de los programas REGRESML y SIMPP, se incluyen en las páginas 205 y 206, y los programas fuente pueden encontrarse en el *diskette* de consulta. REGRESML requiere para su correcto funcionamiento de un equipo de cómputo con monitor EGA/VGA y 640 Kb de memoria RAM.

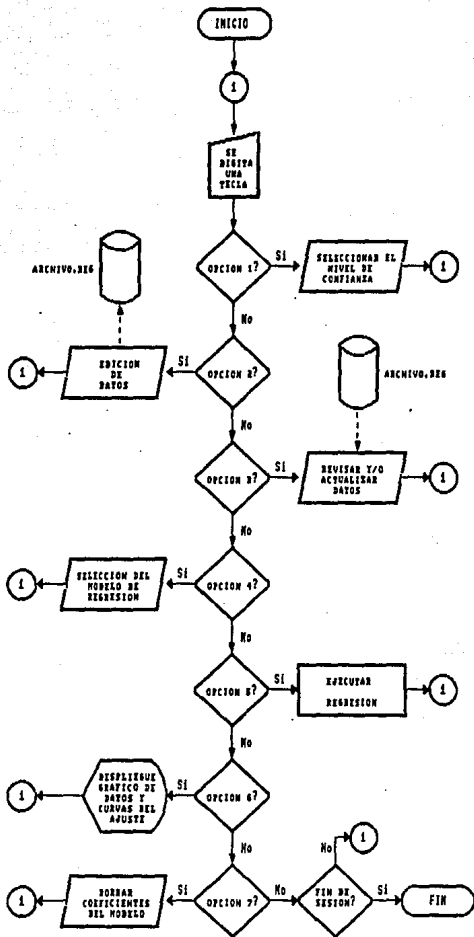


Figura 6.1. Diagrama de bloques del Programa REGRESML.

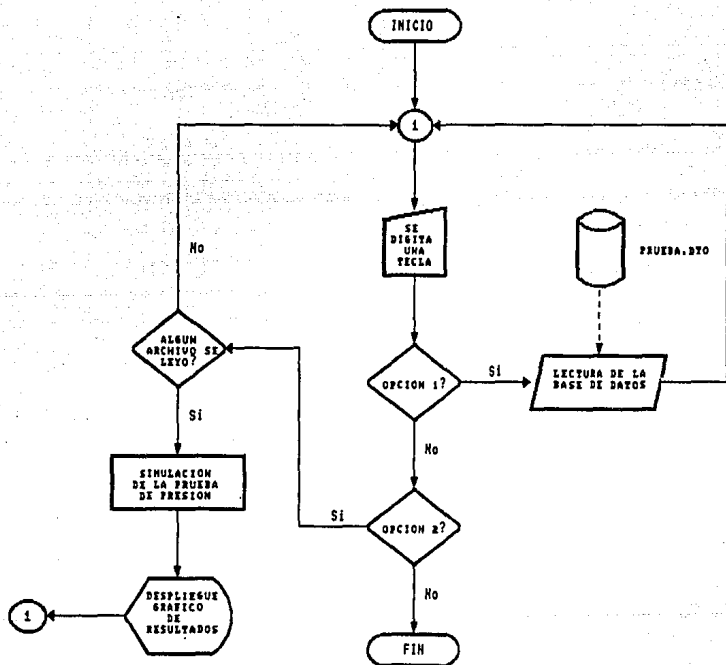


Figura 6.2. Diagrama de bloques del Programa para el Simulador de Pruebas de Decremento de Presión.

## VI.2 CONSTRUCCION Y USO DE PROGRAMAS DE BIBLIOTECA EN QUICK BASIC.

Generalmente en cualquier lenguaje de programación pueden construirse v/o utilizarse programas *de biblioteca* que contienen subrutinas probadas y que ejecutan algoritmos de uso frecuente.

Actualmente, a nivel internacional, es posible adquirir programas *de biblioteca* que ahorran al programador-analista una cantidad considerable de tiempo en la terminación de un proyecto de cómputo.

Una ventaja adicional del uso de programas *de biblioteca*, se presenta en el caso de tener problemas con algún programa en desarrollo o mantenimiento, puesto que sólo será necesario analizar el código de este último sin considerar las rutinas incluidas en los programas *de biblioteca*, lo cual reduce significativamente el proceso de corrección.

Por otro lado, el manejo de programas *de biblioteca* hace posible la utilización de subrutinas creadas en diferentes escenarios de programación tales como FORTRAN, ASSEMBLER, C, etcétera.

La construcción de programas *de biblioteca* es bastante simple; el primer paso es *ligar* todas las subrutinas, que con anterioridad fueron compiladas, para producir archivos con extensión (OBJ). A continuación se presenta un ejemplo:

```
LINK /O SUBR1.OBJ SUBR2.OBJ SUBR3.OBJ,MILIB.OLB,,BOLB45.LIB;
```

El programa *de biblioteca* que se genera al ejecutar este comando se llama MILIB.OLB y contiene las subrutinas SUBR1.OBJ, SUBR2.OBJ y SUBR3.OBJ, y adicionalmente las de BOLB45.LIB que incluyen funciones básicas (uso común) de *Quick Basic*.

El segundo paso es construir un programa *de biblioteca* paralelo usando los mismos archivos OBJ codificados con la primera instrucción, esto es:

LIB MILIB.LIB+SUBR1.OBJ+SUBR2.OBJ+SUBR3.OBJ;

Después de ejecutar esta última instrucción, es posible invocar el programa *de biblioteca* desde cualquier programa de cómputo (codificado en *Basic*) haciendo uso de la instrucción QB/L MILIB.

Es importante mencionar que la metodología explicada anteriormente constituye una de las dos posibles formas de crear programas *de biblioteca* en *Quick Basic*, la otra se deja como investigación para el lector interesado.

Una nota digna de comentarse, es que cuando se genera un programa ejecutable que requiere de algún programa *de biblioteca*, el código resultante sólo considera las rutinas que utiliza el programa, ignorando el resto de las incluidas en la *biblioteca*. *Quick Basic* permite construir programas *de biblioteca* a partir de otros y tiene un programa denominado QLBUMP.BAS que permite revisar el contenido de las mismas, es decir, proporciona un listado de todas las subrutinas implicadas.

La explicación anterior no intenta ser un resumen de todos los conceptos mencionados en los manuales de *Quick Basic* referentes a la creación, uso y mantenimiento de programas *de biblioteca*, el lector deberá dirigirse a ellos para un estudio a detalle.

### VI.3 DISEÑO OPTIMO DE PROGRAMAS DE COMPUTO.

El desarrollo de programas de cómputo eficientes, es la meta de todo buen analista. Comparando los tiempos de ejecución de dos programas realizados para el mismo fin, se observa que el tiempo es menor para el programa más eficiente. La rapidez de ejecución de un programa puede afectarse apreciablemente al utilizar códigos de programación (conjunto de instrucciones en *Fortran*, *Basic*, *Pascal*, etcétera) ineficientes. ¿Por qué preocuparnos por el tiempo de ejecución, si las computadoras realizan los procesos a velocidades muy grandes? Si bien es cierto que los equipos de cómputo modernos son capaces de realizar cientos de operaciones por segundo, también lo es el hecho de que existen problemas ingenieriles cuya solución requiere del manejo de demasiada información y de gran número de operaciones repetitivas.

Un programa eficiente de cómputo debe tomar en cuenta las siguientes características:

- Velocidad de ejecución.
- Entradas y salidas convenientes.
- Facilidad de implementación en otras máquinas.
- Mensajes de diagnóstico a errores cometidos por el usuario.

A continuación se enumeran algunas recomendaciones importantes para la obtención de programas eficientes:

1) Uso de subíndices múltiples.

Siempre que sea posible use un sólo subíndice. Suonga un programa (en lenguaje *Basic*), que calcule la saturación de aceite,  $S_o$ , en función de la saturación de gas y agua ( $S_g$  y  $S_v$ , respectivamente), para cada uno de los bloques en los cuales se ha discretizado un yacimiento (ver Figura 6.3):

```

DIM SW(10,10,10), SO(10,10,10), SG(10,10,10)
FOR I = 1 TO NX
  FOR J = 1 TO NY
    FOR K = 1 TO NZ
      SO(I,J,K) = 1.0 - SG(I,J,K) - SW(I,J,K)
    NEXT K,J,I
  
```

La computadora almacena la información contenida en cualquier arreglo en forma de vector, esto es:

$$\begin{bmatrix} 2 & 3 \\ 4 & 1 \end{bmatrix}$$

REPRESENTACION  
EN PAPEL

$$\begin{bmatrix} 2 \\ 4 \\ 3 \\ 1 \end{bmatrix}$$

REPRESENTACION INTERNA  
EN LA COMPUTADORA

Por tanto, en la ejecución del programa mostrado anteriormente, el compilador traduce a un sólo subíndice, M:

$$M=100*K-1+10*J-1+I$$

Este cálculo se realiza internamente cada vez que se ejecuta el FOR ... NEXT, lo cual genera 6 multiplicaciones, 6 sumas y 6 restas, para efectuar el proceso en cada celda (cálculo de So).

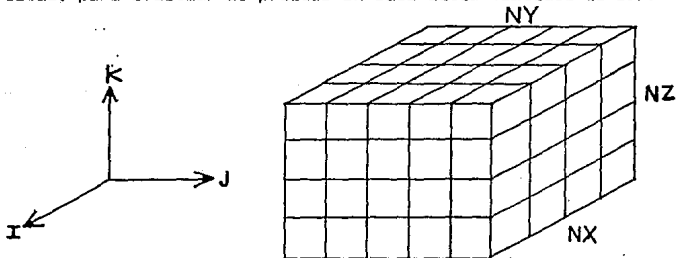


Figura 6.3. Yacimiento hipotético en tres dimensiones.



El mismo procedimiento puede ser ejecutado como:

```
DIM SW(1000), SG(1000), S6(1000)
M = 0
FOR I = 1 TO NX
  FOR J = 1 TO NY
    FOR K = 1 TO NZ
      M = M + 1
      SO(M) = 1.0 - S6(M) - SW(M)
    NEXT K,J,I
```

En el cual sólo se requiere una suma para cada asignación de  $S_0$  en una celda.

Otro procedimiento más eficaz, que no requiere de operaciones intermedias es el siguiente:

```
DIM SW(1000), SG(1000), S6(1000)
NXYZ = NX * NY * NZ
FOR M = 1 TO NXYZ
  SO(M) = 1.0 - S6(M) - SW(M)
NEXT M
```

## 2) Manejo de funciones tabuladas.

Muchas propiedades en programas de Ingeniería Petrolera se introducen en forma de tablas, por ejemplo propiedades de los fluidos y/o de la roca. El valor de una variable dependiente para su correspondiente variable independiente, se calcula por interpolación de los datos tabulados. Algunos autores han investigado acerca de algoritmos de búsqueda para interpolación, en datos ordenados en forma ascendente, con respecto a su valor numérico, y concluyen que la Investigación Secuencial es la más adecuada cuando se tiene un número reducido de datos (10 a 20), mientras que para un número mayor la Investigación Binaria es mejor.

La Investigación Secuencial consiste en localizar el intervalo

de interés mediante una búsqueda sistemática y ordenada en cada uno de los intervalos entre datos (si se tienen N datos se presentan N-1 intervalos). Un código en Basic, para obtener por interpolación un valor Y para un X dado, con Búsqueda Secuencial es:

```
FOR I = 1 TO N-1
  IF X > X(I) AND X < X(I+1) THEN GOTO 10
NEXT I
10 *** PROCEDIMIENTO DE INTERPOLACION ***
```

La Investigación Binaria, por otro lado, se basa en dividir el total de datos en dos partes iguales, posteriormente se investiga en que mitad queda el intervalo de interés, el cual se considera ahora como el total de datos. El proceso se repite hasta que se llega al intervalo adecuado. El siguiente programa realiza una Búsqueda Binaria:

```
I = 1
J = N + 1
DO
  K = (I + J)/2
  IF X < X(K) THEN J = K
  IF X >= X(K) THEN I = K
LOOP WHILE X > X(I+1)
*** PROCEDIMIENTO DE INTERPOLACION ***
```

### 3) Uso de subrutinas.

El llamado a una subrutina dentro de un programa requiere de una cierta cantidad de tiempo para búsqueda interna dentro de la memoria de la computadora, el cual se incrementa con el número de argumentos a transferir. Debe evitarse invocar frecuentemente una subrutina (por ejemplo dentro de un FOR ... NEXT), es mejor duplicar el código (conjunto de instrucciones) en esta parte del programa en lugar de llamarlo como subrutina; cuando sea necesario hacerlo, las variables deben pasarse por COMMON SHARED en lugar de ser transferidas como argumentos.

#### 4) Entradas y salidas.

Siempre que sea posible debe utilizarse para entrada/salida (INPUT/OUTPUT, I/O) la opción más apropiada a los datos a manejar. Si los datos van a consultarse o modificarse continuamente (archivos de cuentas corrientes en un banco, stock de un supermercado, control de pasajes de una línea aérea, etcétera), la mejor alternativa es emplear Archivos de Acceso Aleatorio o Directo. Los registros en estos archivos están numerados, lo que permite hacer referencia a un registro específico sin pasar por los que le anteceden, con lo que el acceso será inmediato (rápido), el inconveniente es que no se aprovecha eficazmente el espacio en disco flexible, ya que al almacenar registros de menor longitud que la establecida, se malgasta espacio (por ejemplo, si la longitud fija de cada registro es de 256 bytes y se sitúan sólo 125 en cada registro, aproximadamente el 50% de cada registro se convertirá en espacios en blanco). *Quick Basic* asigna por defecto una longitud fija de 128 bytes. Otra desventaja es que se necesita conocer el número que indica la posición del registro, además el proceso de I/O es un tanto difícil de programar. El siguiente es un ejemplo en lenguaje *Basic* para abrir un archivo aleatorio:

```
'DEFINICION DE UN REGISTRO LLAMADO YACIMIENTOS
TYPE YACIMIENTOS
    FORMACION AS STRING*15
    POROSIDAD AS REAL
    PERMEAB AS REAL
END TYPE

'DECLARACION DE UNA VARIABLE DEL TIPO YACIMIENTOS
DIM ROCA AS YACIMIENTOS

'APERTURA DEL ARCHIVO PARA ACCESO ALEATORIO
OPEN "PETRO.DAT" FOR RANDOM AS #1 LEN = LEN (ROCA)

'VERIFICACION DEL NUMERO DE REGISTROS EXISTENTE
REG = LOF(1)/LEN(ROCA)

'PROCESO DE I/O
INPUT "FORMACION ?"; ROCA.FORMACION
```

```

INPUT "POROSIDAD ?". ROCA.POROSIDAD
INPUT "PERMEABILIDAD ?". ROCA.PERMEAB
'ALMACENAR REGISTRO
PUT #1, REG, ROCA
'LEER REGISTRO
GET #1, REG, ROCA
PRINT "FORMACION :". ROCA.FORMACION
PRINT "POROSIDAD :". ROCA.POROSIDAD
PRINT "PERMEABILIDAD :". ROCA.PERMEAB
'CERRAR ARCHIVO ABIERTO
CLOSE #1

```

Cuando se tiene un número relativamente pequeño de datos con longitudes diversas, se recomienda el uso de Archivos de Acceso Secuencial, tales archivos emplean eficientemente el espacio en disco flexible (su tamaño depende exclusivamente de la longitud y cantidad de cada uno los registros, por lo que no se tienen espacios en blanco innecesarios). La secuencia de I/O es fácil de programar y como se leen todos los registros a la vez, el acceso es más lento comparado con el de los archivos directos. Cabe mencionar que el orden de escritura de variables debe respetarse al efectuar una lectura posterior. A continuación se presenta un procedimiento para manejo de archivos secuenciales:

```

'ALMACENAR REGISTRO
OPEN "PETRO.DAT" FOR OUTPUT AS #1
INPUT "FORMACION "; FORMACION : PRINT #1, FORMACION
INPUT "POROSIDAD "; POROSIDAD : PRINT #1, POROSIDAD
INPUT "PERMEABILIDAD "; PERMEABILIDAD : PRINT #1, PERMEAB
CLOSE #1 'CERRAR ARCHIVO ABIERTO
'LEER REGISTRO
OPEN "PETRO.DAT" FOR INPUT AS #1
INPUT #1, FORMACION : PRINT "FORMACION :"; FORMACION
INPUT #1, POROSIDAD : PRINT "POROSIDAD :"; POROSIDAD
INPUT #1, PERMEAB : PRINT "PERMEABILIDAD :"; PERMEAB

```

\*CERRAR ARCHIVO ABIERTO

CLOSE #1

Se puede concluir que el tiempo requerido para ejecutar operaciones de I/O es función del número de ítems (variables) en la lista de entrada/salida.

5) Consideraciones adicionales.

- a) Evite mezclar aritmética (conversiones INTEGER a REAL).
- b) En lugar de  $(2*A)$  use  $(A+A)$ .
- c) La multiplicación es más rápida que la división, por lo que en vez de  $(A/2)$  use  $(0.5*A)$ .
- d) Emplee  $(A*A)$  en lugar de  $(A^2)$ .
- e) Evite dejar expresiones constantes dentro de procesos iterativos.
- f) Codificar en *lenguaje de Máquina* comparado con el código que utiliza un compilador (*Fortran, Basic, Pascal*, etcétera), puede incrementar la velocidad de ejecución de 2 a 3 veces.
- g) Utilice *almacenamiento (arreglo) dinámico* en lugar de *estático*.

6) Facilidad de implementación.

- a) Evite usar caracteres especiales en los nombres de las variables.
- b) Es recomendable usar un máximo de 6 caracteres en los nombres asignados a una variable (recuerde: 1 caracter igual a un 1 byte). De preferencia emplear nombres que identifiquen fácilmente a las variables codificadas.
- c) Al capturar un programa de cómputo coloque comentarios alusivos a la instrucción codificada, a fin de que cualquier usuario pueda comprender el proceso, en caso de requerirse su modificación o inclusión como módulo (subrutina) en otro programa.

#### VI.4 PROBLEMAS PROPUESTOS.

PPVI-1. Modifique el programa REGRESML a fin de obtener una comparación estadística con la F de Fischer.

PPVI-2. Elabore un programa de biblioteca que incluya todas las subrutinas del programa REGRESML.

PPVI-3. Modifique el programa SIMFF a fin de simular pruebas de incremento de presión.

PPVI-4. Desarrolle una subrutina de graficación en pantallas de resolución VGA (screen 9) de uso general.

PPVI-5. Simule, empleando el programa SIMFF, una prueba de decremento de presión, considerando:

$$\begin{aligned}q_o &= 6917 \text{ BFD} \\B_o &= 2.3 \\k &= 50 \text{ mD} \\c_t &= 2.5316E-05 \text{ (lb/pg}^2\text{)}^{-1} \\r_w &= 0.208 \text{ pie} \\h &= 164 \text{ pie} \\p_i &= 8901.72 \text{ (lb/pg}^2\text{)} \\C_D &= 10,000 \\ \mu &= 0.39 \text{ cp} \\ \phi &= 0.02\end{aligned}$$

Calcule la presión para 0.1, 0.2, 0.3, 0.4, 0.5, 1, 2, 4, 6, 7, 8, 9, 10, 11, 12, 15, 17, 20, 30 y 40 horas. Aplique la técnica semilogarítmica de MDH para corroborar la calidad de los datos de presión calculados.

## APENDICE

---

Los archivos con extensión BAS requieren del compilador *Quick Basic* (versión 4.5) para ser ejecutados; los de extensión BAT deben invocarse directamente desde el *prompt* del Sistema Operativo (A>, C>, ...), y los de extensiones DAT, DTD, REG, RES y AYD (bases de datos y resultados), pueden accesarse mediante el compilador *Quick Basic* y/o cualquier editor de textos u hoja de cálculo.

### ARCHIVOS CONTENIDOS EN EL DISCO DE APLICACION

SUBDIRECTORIO	CONTENIDO (ARCHIVOS Y BASES DE DATOS)
RAIZ	LEEME.BAT
CAP1	MACHEPS.BAS PI-1A.BAS PI-1B.BAS PI-2.BAS PI-3.BAS PI-4.BAS PI-5A.BAS PI-5B.BAS
CAP2	THOMAS.BAS LU-PENTA.BAS DECOMP.BAS SIMUL.BAS ARCHIVO.DAT REDGAS.BAS INVERSA.BAS
CAP3	SPLINE.BAS INTSDIM.BAS PRES.DAT, T-32.DAT, T-100.DAT,

## SUBDIRECTORIO

## CONTENIDO (ARCHIVOS Y BASES DE DATOS)

## CAP3

T-190.DAT Y T-280.DAT  
SVD.BAS  
OPTPERF.BAS  
DENSEQUI.DTO, DIENGAST.DTO,  
PESBARRE.DTO, PRESFORM.DTO,  
PROFUN.DTO, REYNOLDS.DTO,  
VELPENET.DTO Y VELROTAC.DTO  
KRIGING.BAS  
DATOS.REG

## CAP4

QUANCB.BAS  
GAUSSO.BAS  
M(P).BAS  
MP1.DAT, MP2.DAT Y MP3.DAT

## CAP5

RKF45.BAS

## CAP6

REGRESML.BAS  
ARCHIVO.REG  
TFILE.AYD  
AYUDA1.AYD  
AYUDA2.AYD  
AYUDA3.AYD  
SIMFP.BAS  
PRUEBA.DTO



## BIBLIOGRAFIA

---

- 1.- Forsythe, G. E., Malcolm, M. A. y Moler, C. B., *Computer Methods for Mathematical Computations*, Prentice-Hall, Inc., Englewood Cliffs, Nueva Jersey, 1977.
- 2.- Carnahan, B., Luther, H. A. y Wilkes, J. O., *Applied Numerical Methods*, John Wiley & Sons, Londres, 1969.
- 3.- Berlanga Gutiérrez, J. M., *Apuntes de Computación Aplicada a la Ingeniería Petrolera*, Facultad de Ingeniería, U.N.A.M., 1980.
- 4.- Borrás García, H. E., Iriarte V. Balderrama, R. y Durán Cuevas, R., *Apuntes de Métodos Numéricos*, Facultad de Ingeniería, U.N.A.M., 1985.
- 5.- Solar González, E. y Speziale de Guzmán, L., *Apuntes de Álgebra Lineal*, Facultad de Ingeniería, U.N.A.M., 1985.
- 6.- Bitter, Gary. G., *Computación (Fundamentos, Aplicaciones, Programación)*, Addison-Wesley Iberoamericana, 1969.
- 7.- Mano, Morris M., *Arquitectura de Computadores*, Prentice-Hall Hispanoamericana S.A., 1989.
- 8.- Rodríguez Nieto, R. y De la Fuente García, G., *Curso Introductorio de Simulación de Yacimientos*, I.M.P., publicación No. 72 BH/094, 1972.
- 9.- Limón H., J. Tomás, *Transporte de Gas en Régimen Permanente*, I.M.P., publicación No. 17/74, proyecto D-341A, 1974.

- 10.- Luthe, R., Olivera, A. y Schutz, F., *Métodos Numéricos*, Editorial Limusa, México, 1979.
- 11.- Bourgoynne, A. T. Jr. y Young, F. S. Jr., *A Multiple Regression Approach to Optimal Drilling and Abnormal Pressure Detection*, S.P.E. Journal, Agosto, 1974.
- 12.- Garaicochea Petrirena, F., *Apuntes de Transporte de Hidrocarburos*, Facultad de Ingeniería, U.N.A.M., 1983.
- 13.- Martínez Romero, N., *Notas de clase de la cátedra Técnicas Avanzadas de Programación*, Facultad de Ingeniería, U.N.A.M., semestre escolar 1991-I.
- 14.- Dake, L. P., *Fundamentals of Reservoir Engineering*, Elsevier Publishing Co., Gran Bretaña, 1978.
- 15.- Hernández García, G. y Berlanga Gutiérrez, J. M., *Evaluación de Volúmenes Originales de Hidrocarburos Empleando Métodos Geoestadísticos (Aplicación al Campo Miguel Alemán-Chicontepec)*, revista Ingeniería Petrolera, Vol. XXIII, No. 3, marzo de 1983.
- 16.- Hernández García, G., *Curso de Geoestadística*, clave 00246B2P, I.M.F., marzo de 1992.
- 17.- Stehfest, Harald, *Algorithm 368 Numerical Inversion of Laplace Transforms*, Communications of the ACM, Vol. 13, No. 1, enero de 1970.
- 18.- Park, Chan S., *Interactive Microcomputer Graphics*, Addison Wesley Publishing Co., 1985.

- 19.- Agarwal, Ram G., Al-Hussainy, R. y Ramey, H. J. Jr., *An Investigation of Wellbore Storage and Skin Effect in Unsteady Liquid Flow: I. Analytical Treatment*, Transactions of AIME, Vol. 249, 1970.
- 20.- Martínez Romero, N. y León Ventura, R., *Diseño de Pruebas de Incremento de Presión*, XXIV Congreso de la Asociación de Ingenieros Petroleros de México.