Guía de Cálculo Numérico

Elaborada por: Ramón Medina

1

Sistemas de Numeración y Errores

1.1 Sistemas de Numeración y Errores

1.1.1 Tipos de Errores

Error por Truncamiento. Se le da este nombre a los errores ocasionados por el método en sí (el nombre se origina del hecho de que los métodos numéricos generalmente pueden ser comparados con una serie de Taylor truncada) y es el error al que se ha prestado más atención. Para los métodos iterativos, de ordinario este error puede ser reducido por medio de iteraciones repetidas pero, ya que la vida es finita y el tiempo de computadora es caro, es necesario quedar satisfecho con las aproximaciones a la respuesta analítica exacta.

Error de Redondeo. Todos los dispositivos de cálculo representan números con alguna imprecisión. Las computadoras digitales que son los dispositivos normales para la implantación de los métodos numéricos, casi siempre utilizan números de punto flotante con una palabra de longitud fija. Los valores verdaderos no son expresados exactamente por tales representaciones. A esto se le llama un error de *redondeo*, ya sea que la fracción decimal esté redondeada, acortada después del dígito final.

Al resolver un problema matemático por medio de una calculadora, debemos estar conscientes de que los números decimales que calculamos quizá no sean exactos. Estos números casi siempre se redondean cuando los registramos. Aun cuando los números se redondeen de manera intencional, el número limitado de dígitos de la calculadora puede provocar errores de redondeo.

En una computadora electrónica, los errores de redondeo aparecen por las mismas razones y afectan los resultados de los cálculos. En algunos casos, los errores de redondeo causan efectos muy serios y hacen que los resultados de los cálculos carezcan por completo de sentido. Por lo tanto, es importante aprender algunos aspectos básicos de las operaciones aritméticas en las computadoras y comprender bajo que circunstancias pueden ocurrir severos errores de redondeo.

1.1.1 Base de los Números

El sistema numérico que usamos cotidianamente se llama *sistema decimal*. La base del sistema numérico decimal es 10. Sin embargo, las computadoras no usan el sistema decimal en los cálculos ni en la memoria, sino que usan el binario. Este sistema es natural para las computadoras ya que su memoria consiste en un enorme número de dispositivos de registro magnético y electrónico, en los que cada elemento sólo tiene los estados de “encendido” y “apagado”.

Sin embargo, si examinamos los lenguajes de máquina, pronto nos percatamos que se usan otros sistemas numéricos, en particular el octal y el hexadecimal. Estos sistemas son parientes cercanos del binario y pueden traducirse con facilidad al o del binario. Las expresiones en octal o hexadecimal son más cortas que en binario, por lo que es más sencillo que las personas las lean y comprendan. El hexadecimal también proporciona un uso más eficiente del espacio de la memoria para los números reales.

La base de un sistema numérico también recibe el nombre de *raíz*. Para el sistema decimal ésta es 10; para el sistema octal es 8 y 2 para el binario. La raíz del sistema hexadecimal es 16.

La base de un número se denota por medio de un subíndice; por ejemplo, $(3.224)_{10}$ es 3.224 en base 10 (decimal), $(1991.11)_2$ es 1001.11 en base 2 (binario) y $(18C7.90)_{16}$ es 18C7.90 en base 16 (hexadecimal).

El valor decimal de un número en base *r*, por ejemplo,

$$(abcdefg.hijk)_r$$

se calcula como

$$ar^6 + br^5 + cr^4 + dr^3 + er^2 + fr + g + hr^{-1} + ir^{-2} + jr^{-3} + kr^{-4}$$

Es común representar sin subíndice, los números que están en base 10.

La representación en base *r* de un número decimal, se obtiene mediante divisiones sucesivas del número por la base, como se muestra en el ejemplo siguiente.

Sea el número decimal $(123456)_{10}$, para obtener su equivalente hexadecimal se procede como sigue:

123456	16				
0	7716	16			
	4	482	16		
		2	30	16	
			14	1	16
				1	0

El proceso de divisiones sucesivas termina, cuando el cociente de una de las operaciones resulte cero. Los dígitos que constituyen la nueva representación son los residuos de cada una de las operaciones, acomodados en orden inverso al de aparición. Así el valor obtenido es $(1E240)_{16}$.

1.1.2 Números dentro del *hardware* de la computadora

Un *bit* es a abreviatura de dígito binario (*binary digit*) y representa un elemento de memoria que consta de posiciones de encendido y apagado, a la manera de un dispositivo semiconductor o un punto magnético en una superficie de registro. Un *byte* es un conjunto de bits considerado como una unidad, que normalmente está formado por 8 bits.

Las formas en que se usan los bits para los valores enteros y de punto flotante varían según el diseño de una computadora.

Enteros. En el sistema de numeración binario, la expresión matemática de un entero es

$$\pm a_k a_{k-1} a_{k-2} \cdots a_2 a_1 a_0$$

donde a^i es un bit con valor 0 o 1. Su valor decimal es

$$I = \pm[a_k 2^k + a_{k-1} 2^{k-1} + \cdots + a_2 2^2 + a_1 2 + a_0]$$

En una computadora, el valor máximo de *k* en la ecuación anterior, está limitado por el diseño del *hardware*.

En un computador personal, se usan 2 bytes (16 bits) para representar un entero. El primer bit registra el signo: positivo si es 0, negativo si es 1. Los restantes 15 bits se usan para los a^i. Por lo tanto el máximo entero positivo es

$$(0111111111111111)_2$$

su equivalente decimal es

$$\sum_{i=0}^{14} 2^i = 32767$$

Una forma de almacenar un número negativo es utilizar los mismos dígitos que el número positivo de la misma magnitud, excepto que el primer bit se pone en 1. Sin embargo, muchas computadoras usan el *complemento a dos* para almacenar números negativos. Por ejemplo, el complemento a dos para $(-32767)_{10}$ es

$$(1000000000000001)_2$$

Los bits del número anterior, se obtienen a partir de la representación binaria del máximo entero positivo (32767), cambiando los 0 por 1 y añadiendo 1 al resultado. Ene l complemento de dos, se determina primero el valor decimal como si los 16 bits expresaran un número positivo. Si este número es menor que 2^{15}, o 32768, se le interpreta como positivo. Si es mayor o igual, entonces se transforma en un número negativo restándole 2^{16}. En el ejemplo anterior del número binario, el equivalente decimal de éste en la ecuación es $Z = 2^{15} + 1$, por lo que la resta da

$$32768 + 1 - 2^{16} = 32768 + 1 - 65536 = -32767$$

Números Reales. El formato para un número real en una computadora difiere según el diseño de *hardware* y *software*.

Los número reales en un computador personal se almacenan en el formato de punto flotante formalizado en binario. En precisión simple, se usan 4 bytes, o 32 bits, para almacenar un número real. Si se introduce como dato un número decimal, primero se convierte al binario más cercano en el formato normalizado:

$$(\pm 0.abbbbbb\cdots bbbb)_2 \times 2^z$$

donde *a* siempre es 1, cada *b* es un dígito binario 0 o 1 y *z* es un exponente que también se expresa en binario. Existen 24 dígitos para la mantisa incluyendo la *a* y las *b*.

Los 32 bits se distribuyen de la manera siguiente. El primer bit se usa para el signo de la mantisa, los siguientes 8 bits para el exponente *z* y los últimos 23 para la mantisa.

```
11111111  11111111  11111111  11111111  32 bits
seeeeeee  emmmmmmm  mmmmmmmm  mmmmmmmm
```

Se usa un bit (s) para el signo.

Se usan 8 bits (e) para el exponente.

Se usan 23 bits (m) para la mantisa.

En el formato de punto flotante normalizado, el primer dígito de la mantisa siempre es 1, por lo que no se almacena físicamente. Esto explica por qué una mantisa de 24 bits se almacena en 23.

Si los 8 bits asignados al exponente se usan sólo para enteros positivos, el exponente puede representar desde 0 hasta $2^8 - 1 = 255$, aunque puede incluir números negativos. Para registrar exponentes positivos y negativos, el exponente en decimal es sumado con 128 y después convertido a binario (complemento a dos). Por ejemplo, si el exponente es -3, entonces -3 + 128 = 125 se convierte a binario y se almacena en los 8 bits. Por lo tanto, los exponentes que se pueden almacenar en 8 bits van desde 0 - 128 = -128 hasta 255 - 128 = 127.

1.1.2 Errores de Redondeo en una Computadora

Errores de redondeo al almacenar un número en memoria. La causa fundamental de errores en una computadora se atribuye al error de representar un número real mediante un número limitado de bits.

Al intervalo entre 1 y el siguiente número distinguible de 1 se le llama ε. Esto significa que ningún número entre 1 y $1 + \varepsilon$ se puede representar en la computadora. En el caso de un computador digital, cualquier número $1 + \alpha$ se redondea a 1 si $0 < \alpha < \varepsilon/2$, o se redondea a $1 + \varepsilon$ si $\varepsilon/2 \leq \alpha$. Así, se puede considerar que $\varepsilon/2$ es el máximo error posible de redondeo para 1. En otras palabras, cuando se halla 1.0 en la memoria, el valor original pudo ser alguno de entre $1 - \varepsilon/2 < x < 1 + \varepsilon/2$. El épsilon de la máquina se puede determinar mediante el siguiente algoritmo:

```
Hacer E igual a 1
Mientras E + 1 sea mayor que 1
      Imprimir E
      Hacer E igual a E/2
Fin de Mientras
```

El último valor impreso por el algoritmo es igual al épsilon de la máquina. Los épsilon para simple y doble precisión en un computador personal son:

Simple:	*1.19E-7*
Doble:	*2.77E-17*

El error de redondeo implicado en el almacenamiento de cualquier número real R en memoria es aproximadamente igual a $\varepsilon R/2$, si el número se redondea por exceso y εR si se redondea por defecto.

Efecto de los errores por redondeo. Si se suman o restan números, la representación exacta del resultado quizá necesite un número de dígitos mucho mayor que el necesario para los números sumados o restados.

Existen dos situaciones en los que aparecen muchos errores por redondeo: (a) cuando se suma (o se resta) un número muy pequeño de uno muy grande y (b) cuando un número se resta de otro que es muy cercano.

El error de un número provocado por el redondeo aumenta cuando el número de operaciones aritméticas también se incrementa.

Para probar el primer caso en la computadora, sumemos 0.00001 a la unidad diez mil veces. El diseño de un programa para este trabajo sería:

```
Hacer SUMA igual a 1
Desde I igual a 1 hasta 10000
      Hacer SUMA igual a SUMA más 0.00001
Fin de Desde
```

```
Imprimir SUMA
```

Se plantea como ejercicio codificar este algoritmo para determinar el resultado arrojado por el computador y contrastarlo con el resultado correcto.

Otro problema se presenta cuando dos números que debiesen ser matemáticamente idénticos, no siempre lo son en las computadoras. Por ejemplo, consideremos las ecuaciones

$$y = A / B$$
$$w = y * B$$
$$z = A - w$$

donde *A* y *B* son constantes. Desde un punto de vista matemático, *w* es igual a *A*, por lo que *z* debe anularse. Si estas ecuaciones se calculan en una computadora, *z* se anula o es un valor no nulo pero muy pequeño, dependiendo de los valores de *A* y *B*. Esto es posible probarlo mediante el siguiente programa

```
Hacer A igual a cos(0,3)
Desde K igual a 1 hasta 20
      Hacer B igual a sen(K)
      Hacer Z igual a A / B
      Hacer W igual a Z * B
      Hacer Y igual a A - W
      Imprimir A, B, W, Y
Fin de Desde
```

Lo que ocurre en la computadora, es que aparece un error de redondeo cuando se calcula Z = A / B y W = Z * B y se almacenan. Así, W = Z * B en la sexta línea del programa, no es exactamente igual a A. La magnitud relativa del error por redondeo atribuida a la multiplicación o división entre una constante y al almacenamiento del resultado en la memoria es casi igual al épsilon de la máquina.

El error de un número provocado por el redondeo aumenta cuando el número de operaciones aritméticas también se incrementa.

Causas de errores por redondeo. Para explicar cómo surgen los errores por redondeo, consideremos el cálculo de 1 + 0,00001 en un computador personal. Las representaciones binarias de 1 y 0,00001 son, respectivamente,

$$(1)_{10} = (0{,}1000\ 0000\ 0000\ 0000\ 0000\ 0000)_2 \times 2^{1}$$
$$(0{,}00001)_{10} = (0{,}1010\ 0111\ 1100\ 0101\ 1010\ 1100)_2 \times 2^{-16}$$

La suma de estos dos números es

$$(1)_{10} + (0{,}00001)_{10} =$$
$$(0{,}1000\ 0000\ 0000\ 0000\ 0101\ 0011\ \underline{1110\ 0010\ 1101\ 0110\ 0})_2 \times 2^{1}$$

Sin embargo los números subrayados se redondean ya que la mantisa tiene 24 bits. Por lo tanto, el resultado de este cálculo se guarda en memoria como

$$(1)_{10} + (0{,}00001)_{10} = (0{,}1000\ 0000\ 0000\ 0000\ 0101\ 0100)_2 \times 2^1$$

que es equivalente a $(1{,}0000100136)_{10}$.

Así, siempre que se sume 0,00001 a 1, el resultado agrega 0,0000000136 como error. Al repetir diez mil veces la suma de 0,00001 a 1, se genera un error de exactamente diez mil veces 0,0000000136. A este error se le conoce como *error de redondeo*.

1.1.3 Error Absoluto y Error Relativo

Dos métodos para medir errores de aproximación lo constituyen el *error absoluto* y el *error relativo*. El error absoluto viene dado por la siguiente expresión

$$Error\ absoluto = valor\ verdadero - valor\ aproximado$$

y el error relativo, por la formula

$$Error\ relativo = \frac{valor\ verdadero - valor\ aproximado}{valor\ verdadero}$$

El error relativo suele ser un mejor indicador de la precisión, ya que es independiente de la escala usada.

Ejemplo:
Si $p = 0{,}3000 \times 10^1$ y $p^* = 0{,}3100 \times 10^1$, el error absoluto es

$$Error\ absoluto = |p - p^*| = |0{,}3000 - 0{,}3100| = 0{,}01$$

$$Error\ relativo = |p - p^*| / p = |0{,}3000 - 0{,}3100| / 0{,}3000 = 0{,}3333 \times 10^{-1}$$

1.1.4 Dígitos Significativos

Se dice que el número p^* aproxima a p con t **dígitos significativos** (o cifras) si t es el entero más grande no negativo para el cual

$$\frac{|p - p^*|}{|p|} < 5 \times 10^{-t}$$

Ejemplo:
Para que p^* aproxime a 1000 con cuatro cifras significativas, usando la definición, p^* debe satisfacer

$$\left| \frac{p^* - 1000}{1000} \right| \leq 5 \times 10^{-4}$$

Esto implica que *999, 5 < p* < 1000,5* y está de acuerdo con la definición intuitiva de cifras significativas.

1.2 Problemas

1.2.1 Suponga que *p** aproxima a *p* con 3 dígitos significativos. Encuentre el intervalo en el cual *p** debe estar, si *p* es

(a) 150 (b) 900
(c) 1500 (d) 90

1.2.2 Considere los siguientes valores de *p** y *p* y calcule el error absoluto y relativo en cada caso.

(a) p= π, p*= 3,1 (b) p= 1/3, p* = 0,333
(c) p= π/1000, p* = 0,0031 (d) p= 100/3, p* = 33,3

1.2.3 ¿A cuántas cifras significativas aproxima *p** a *p* en el ejercicio 1.2.2?

1.2.4 Convierta los siguientes números de máquina representados en punto flotante, en números decimales

(a) 00011110110010000100000000000000
(b) 10011110110010000100000000000000
(c) 00100001010010000100000000000000
(d) 10100001010010000100000000000000

1.2.5 Si se usan 8 bits para representar los enteros positivos y negativos en complemento a dos, ¿cuál es el entero positivo más grande y el negativo más pequeño (en magnitud) en decimal?.

1.2.6 Se tienen dos números enteros de 16 bits:

$(111111111111111)_2$
$(100000000000110)_2$

Determine los valores decimales correspondientes.

1.2.7 Hallar el épsilon de la máquina para un computador personal, usando el algoritmo descrito en la sección 1.1.2.

1.2.8 Codifique y ejecute el segundo algoritmo descrito en la sección 1.1.2.

1.2.9 Codifique y ejecute el tercer algoritmo descrito en la sección 1.1.2.

2

Preliminares Matemáticos

2.1 Teorema del Valor Medio

Si $f \in C[a,b]$ y f es diferenciable en (a,b), entonces existe un número c, $a < c < b$, tal que

$$f'(c) = \frac{f(b) - f(a)}{b - a}$$

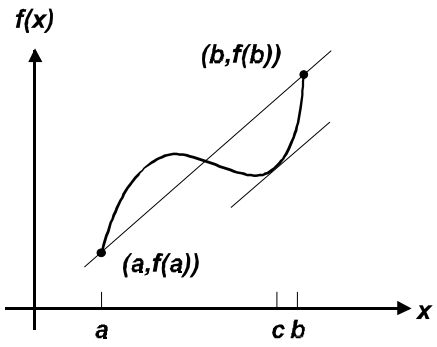

FIGURA 2.1

2.2 Teorema del Valor Medio Ponderado para Integrales

Si $f \in C[a,b]$, g es integrable en $[a,b]$, y $g(x)$ no cambia de signo en $[a,b]$, entonces existe un número c, $a < c < b$, tal que:

$$\int_a^b f(x)g(x)dx = f(c)\int_a^b g(x)dx$$

Si $f \in C[a,b]$ y K es un número cualquiera entre $f(a)$ y $f(b)$, entonces existe c en (a,b) tal que $f(c) = K$ (ver figura 2.1).

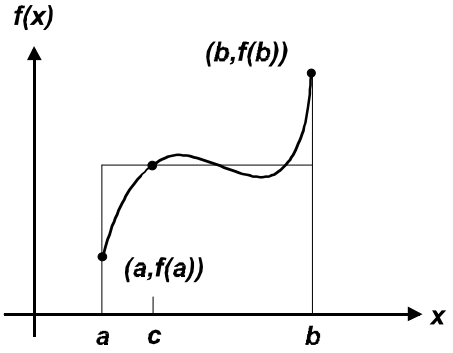

FIGURA 2.2

2.3 Teorema de Rolle

Supongamos que $f \in C[a,b]$ y f es diferenciable en (a,b). Si $f(a) = f(b) = 0$, entonces existe un número c, $a < c < b$, tal que $f'(c) = 0$.

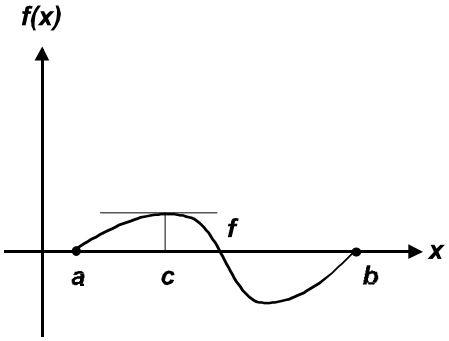

FIGURA 2.3

2.4 Teorema del Valor Intermedio

Si $f \in C[a,b]$ y K es un número cualquiera entre $f(a)$ y $f(b)$, entonces existe c en (a,b) tal que $f(c) = K$.

Ejemplo:

Para demostrar que $x^5 - 2x^3 + 3x^2 - 1 = 0$ tiene una solución en el intervalo [0,1], consideremos la función $f(x)= x^5 - 2x^3 + 3x^2 - 1$. Claramente f es continua en [0,1] y f(0) = -1 mientras que f(1) = 1. Como $f(0) < 0 < f(1)$, el teorema del Valor Intermedio implica que existe un número x, con $0 < x < 1$, para el cual $x^5 - 2x^3 + 3x^2 - 1 = 0$.

Como se ve en el ejemplo anterior, el teorema del Valor intermedio es importante para ayudar a determinar cuándo existen soluciones a ciertos problemas. Sin embargo, no da la manera de encontrar estas soluciones.

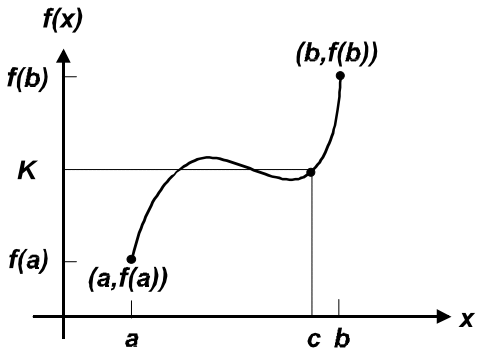

FIGURA 2.4

2.5 Teorema de Taylor

Supongamos que $f \in C^n[a,b]$ y $f^{(n+1)}$ existe en [a,b]. Sea $x_0 \in [a,b]$. Para toda $x \in [a,b]$, existe $\xi(x)$ entre x_0 y x tal que:

$$f(x) = P_n(x) + R_n(x),$$

donde

$$P_n(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

$$= \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$$

y

$$R_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)^{n+1}$$

A $P_n(x)$ se le llama el **polinomio de Taylor de grado n** para f alrededor de x_0 y a $R_n(x)$ se le llama el **residuo** (o **error por truncamiento**) asociado con $P_n(x)$. La serie infinita que se obtiene tomando el límite de $P_n(x)$ cuando n ® ¥ se denomina **Serie de Taylor** para f alrededor de x_0. En el caso de que $x_0 = 0$, el polinomio de Taylor se conoce frecuentemente como **polinomio de Maclaurin** y la serie de Taylor se denomina **serie de Maclaurin**.

El término **error de truncamiento** generalmente se refiere al error involucrado al usar sumas finitas o truncadas para aproximar la suma de una serie infinita.

2.6 Teorema Fundamental del Álgebra

Si P es un polinomio de grado $n \geq 1$, entonces $P(x) = 0$ tiene al menos una raíz (posiblemente compleja).

Corolario

Si $P(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$ es un polinomio de grado $n \geq 1$, entonces existen constantes únicas $x_1, x_2, \ldots, x_n$, posiblemente complejas, y enteros positivos, $m_1, m_2, \ldots, m_n$ tales que

$$\sum_{i=1}^{k} m_i = n$$

y

$$P(x) = a_n (x - x_1)^{m_1} (x - x_2)^{m_2} \cdots (x - x_k)^{m_k}$$

Este corolario afirma que los ceros de un polinomio son únicos y que si cada cero x_i es contado tantas veces como su multiplicidad m_i, entonces un polinomio de grado n tiene exactamente n ceros.

Corolario

Sean P y Q polinomios a los más de grado n. Si $x_1, x_2, \ldots, x_k$, $k > n$, son números distintos de $P(x_1) = Q(x_1)$ para $i = 1, 2, \ldots, k$, entonces $P(x) = Q(x)$ para todo valor de x.

2.7 Problemas

2.7.1 Sea $f(x) = 1 - e^x + (e-1)sen((p/2)x)$. Demuestre que $f'(x)$ es cero cuando menos una vez en $[0,1]$.

2.7.2 Demuestre que la ecuación $x^3 = e^x senx$ debe tener por lo menos una solución en $[1,4]$.

2.7.3 Demuestre que la ecuación $x = 3^{-x}$ tiene una solución en $[0,1]$.

2.7.4 Use el teorema del Valor Intermedio y el teorema de Rolle para demostrar que la gráfica de $f(x) = x^3 + 2x + k$ cruza el eje x exactamente una vez, independientemente del valor de la constante k.

2.7.5 Encuentre el polinomio de Taylor de cuarto grado para f alrededor de $x_0 = 0$ si $f(x) = e^x cosx$. Use este polinomio para aproximar $f(p/16)$, y encuentre una cota para el error en esta aproximación.

2.7.6 Use un polinomio de Taylor alrededor de $p/4$ para aproximar $cos42°$ con una precisión de 6 dígitos significativos.

3

Solución de Ecuaciones No Lineales

Las soluciones de una ecuación no lineal se llaman *raíces o ceros*. Los siguientes son algunos ejemplos de ecuaciones no lineales:

a) $1 + 4x - 16x^2 + 3x^3 + 3x^4 = 0$

b) $f(x) - a = 0,\ a < x < b$

c) $$\frac{x(2{,}1-0{,}5x)^{1/2}}{(1-x)(1{,}1-0{,}5x)^{1/2}} - 3{,}69 = 0, 0 < x < 1$$

d) $\tan(x) = \tanh(2x)$

La primera es un ejemplo de ecuación polinomial, que puede aparecer como una ecuación característica para una ecuación diferencial ordinaria lineal, entre otros problemas. El segundo ejemplo es equivalente a evaluar $f^{1}(a)$, donde $f(x)$ es cualquier función y f^{1} es su función inversa. El tercer ejemplo es un caso especial del inciso *b)*. El cuarto ejemplo es una ecuación trascendental.

TABLA 3.1. Resumen de los métodos para encontrar raíces

Nombre	Necesidad de especificar un intervalo que contenga a la raíz	Necesidad de la continuidad de *f'*	Tipo de ecuaciones	Otras características especiales
Bisección	"si"	no	cualquiera	Robusto, aplicable a funciones no analíticas.
Falsa posición	"si"	"si"	cualquiera	Convergencia lenta en un intervalo grande.
Falsa posición modificada	"si"	"si"	cualquiera	Más rápido que el método de la falsa posición.
Método de Newton	no	"si"	cualquiera	Rápido; se necesita calcular *f'*; aplicable a raíces complejas.
Método de secante	no	"si"	cualquiera	Rápido; no se requiere calcular *f'*.
Sustitución sucesiva	no	"si"	cualquiera	Puede no converger.

La razón principal para resolver ecuaciones no lineales por medio de métodos computacionales es que esas ecuaciones carecen de solución exacta, excepto para muy pocos problemas. La solución analítica de las ecuaciones polinomiales existe sólo hasta el orden cuatro, pero no existe soluciones en forma exacta para órdenes superiores. Por lo tanto, las raíces de esas ecuaciones no lineales se obtienen mediante métodos computacionales basados en procedimientos iterativos.

3.1 Algoritmo de Bisección

La primera técnica, basada en el teorema de Valor Intermedio, se llama **algoritmo de bisección** o **método de búsqueda binaria**. Supongamos que tenemos una función continua *f*, definida en el intervalo *[a,b]*, con *f(a)* y *f(b)* de signos distintos. Entonces se tiene que *$ p / a < p < b* y *f(p) = 0*. El método requiere de dividir repetidamente a la mitad a los subintervalos de *[a,b]* y, en cada paso, localizar la mitad que contiene a *p*. Para empezar, tomamos $a_1 = a$ y $b_1 = b$, y p_1 el punto medio de *[a,b]*; o sea,

$$p_1 = ½(a_1 + b_1)$$

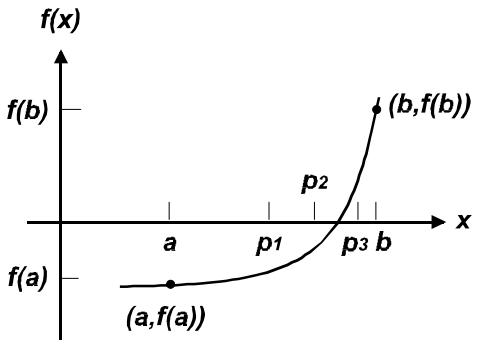

FIGURA 3.1

Si $f(p_1) = 0$, entonces $p = p_1$; si no, entonces $f(p_1)$ tiene el mismo signo que $f(a_1)$ o $f(b_1)$. Si $f(p_1)$ y $f(a_1)$ tienen el mismo signo, entonces $p \in (p_1 , b_1)$, y tomamos $a_2 = p_1$ y $b_2 = b_1$. Si $f(p_1)$ y $f(b_1)$ son del mismo signo, entonces $p \in (a_1 , p_1)$, y tomamos $a_2 = a_1$ y $b_2 = p_1$. Ahora replicamos el proceso al intervalo $[a_2,b_2]$. Esto produce el siguiente algoritmo

Entrada: extremos *a*, *b*; tolerancia *TOL*; máximo número de iteraciones N_0.
Salida: solución aproximada *p* o mensaje de fracaso.

```
Hacer i igual a 1
Mientras i sea menor o igual que N0
   Hacer p = a + (b - a)/2 (Calcular pi)
   Si f(p) es igual a cero ó (b - a)/2 es menor o igual que TOL, entonces
       Mostrar (p); (procedimiento completado satisfactoriamente)
       Parar
   Fin de Si
   Hacer i igual a i + 1
   Si f(a)f(p) es mayor que 0 entonces
       Hacer a igual a p (Calcular ai, bi)
   Si no
       Hacer b igual a p
   Fin de Si
Fin de Mientras
Mostrar ('El método fracasó después de N0 iteraciones')
Parar
```

Otros procedimientos de paro que pueden también aplicarse en el paso 4 del algoritmo, son los que se muestran a continuación. Seleccione una tolerancia $e > 0$ y genere $p_1,\ldots, p_N$ hasta que una de las siguientes condiciones se satisfaga

$$\left|P_N - P_{N-1}\right| < \varepsilon$$

$$\frac{\left|P_N - P_{N-1}\right|}{\left|P_N\right|} < \varepsilon, P_N \neq 0$$

$$\left|f(P_N)\right| < \varepsilon$$

Desafortunadamente, pueden surgir dificultades usando cualquiera de estos criterios de paro. Sin conocimiento adicional acerca de *f* o *p*, la desigualdad es el mejor criterio de paro que puede aplicarse porque verifica al error relativo. Cuando usamos una computadora para generar las aproximaciones, es una buena idea añadir una condición que imponga un máximo al número de iteraciones realizadas.

El algoritmo de bisección, aunque conceptualmente claro, tiene inconvenientes importantes. Converge muy lentamente y una buena aproximación intermedia puede ser desechada sin que nos demos cuenta. Sin embargo, el método tiene la propiedad importante de que converge siempre a una solución y, por esta razón se usa frecuentemente para "poner en marcha" a los métodos más eficientes que se presentarán más adelante en este capítulo.

3.2 Método de la Falsa Posición

Se basa en interpolación lineal y es análogo al método de bisección, puesto que el tamaño del intervalo que contiene la raíz se reduce mediante iteración. Sin embargo, en vez de bisectar en forma monótona, se utiliza una interpolación lineal ajustada a dos extremos para encontrar una aproximación de la raíz. Así, si la función está bien aproximada por la interpolación lineal, entonces las raíces estimadas tendrán una buena precisión y, en consecuencia, la iteración convergerá más rápido que cuando se utiliza el método de bisección.

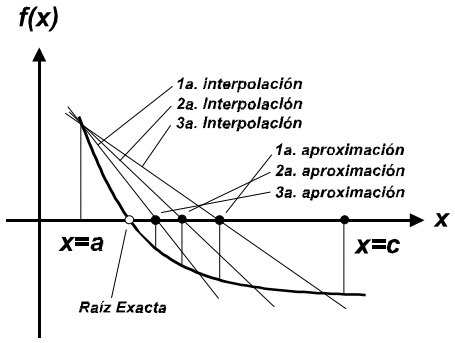

FIGURA 3.2

Dado un intervalo *[a,c]* que contenga a la raíz, la función lineal que pasa por *(a,f(a))* y *(c,f(c))* se escribe como

$$y = f(a) + \frac{f(c) - f(a)}{c - a}(x - a)$$

o, despejando x,

$$x = a + \frac{c + a}{f(c) - f(a)}(y - f(a))$$

La coordenada *x* en donde la línea intersecta al eje *x* se determina al hacer $y = 0$ en la ecuación anterior

$$b = a - \frac{c+a}{f(c)-f(a)}\, f(a) = \frac{af(c)-cf(a)}{f(c)-f(a)}$$

Después de encontrar *b*, el intervalo *[a,c]* se divide en *[a,b]* y *[b,c]*. Si *f(a)f(b) £ 0*, la raíz se encuentra en *[a,b]*; en caso contrario, está en *[b,c]*. Los extremos del nuevo intervalo que contiene a la raíz se renombran *a* y *c*. El procedimiento de interpolación se repite hasta que las raíces estimadas convergen.

La desventaja de este método es que pueden aparecer extremos fijos, como se muestra en la figura 3.2, en donde uno de los extremos de la sucesión de intervalos no se mueve del punto original, por lo que las aproximaciones de la raíz, denotadas por b_1, b_2, b_3, ... convergen a la raíz exacta solamente por un lado. Los extremos fijos no son deseables debido a que hacen más lenta la convergencia, en particular cuando el intervalo inicial es muy grande o cuando la función se desvía de manera significativa de una línea recta en el intervalo. El método de la falsa posición modificado elimina esta dificultad.

3.3 Método de la Falsa Posición Modificada

En este método, el valor de *f* en un punto fijo se divide a la mitad si este punto se ha repetido más de dos veces. El extremo que se repite se llama *extremo fijo*. La excepción a esta regla es que para *i = 2*, el valor de *f* en un extremo se divide entre 2 de inmediato si no se mueve.

El efecto de dividir el valor de *y* es que la solución de la interpolación lineal se hace cada vez más cercana a la verdadera raíz.

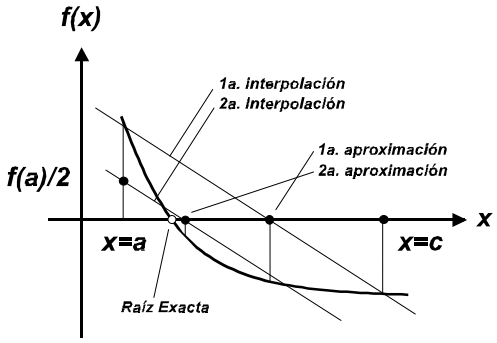

FIGURA 3.3

3.4 Método de Newton

Este método (también llamado método de Newton-Raphson) encuentra una raíz, siempre y cuando se conozca una estimación inicial para la raíz deseada. Utiliza rectas tangentes que se evalúan analíticamente. El método de Newton se puede aplicar al dominio complejo para hallar raíces complejas.

El método de Newton se obtiene a partir del desarrollo de Taylor. Supóngase que el problema es encontrar una raíz de *f(x) = 0*. Al utilizar el desarrollo de Taylor de *f(x)* en torno a una estimación x_0, la ecuación se puede escribir como

$$f(x) = 0 = f(x_0) + f'(x_0) + O(h^2)$$

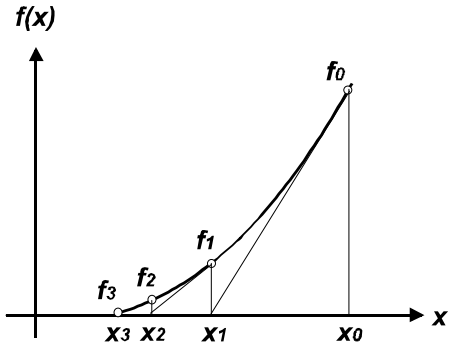

FIGURA 3.4

donde $h = x - x_0$. Al despejar *x* en la ecuación anterior no se obtiene el valor exacto debido al error de truncamiento, pero la solución se acerca en mayor medida al *x* exacto, que lo que se aproxima al estimado x_0. Por lo tanto, al repetir la solución utilizando el valor actualizado como una nueva estimación, se mejora la aproximación en forma sucesiva.

El algoritmo se muestra de manera gráfica en la figura 3.4. El valor x_0 es una estimación inicial para la raíz. A continuación se obtiene la función lineal que pasa por (x_0, y_0) en forma tangencial. La intersección de la recta tangente con el eje *x* se denota como x_1 y se considera como una aproximación de la raíz. Se repite el mismo

procedimiento, utilizando el valor actualizado como una nueva estimación, se mejora la aproximación en forma sucesiva.

El algoritmo se muestra de manera gráfica en la figura 3.4. El valor x_0 es una estimación inicial para la raíz. A continuación se obtiene la función lineal que pasa por (x_0, y_0) en forma tangencial. La intersección de la recta tangente con el eje x se denota como x_1 y se considera como una aproximación de la raíz. Se repite el mismo procedimiento, utilizando el valor más actualizado como una estimación para el siguiente ciclo de iteración.

La recta tangente que pasa por $(x_0, f(x_0))$ es

$$g(x) = f'(x_0)(x-x_0) + f(x_0)$$

La raíz de $g(x) = 0$ denotada por x_1 satisface

$$f'(x_0)(x_1-x_0) + f(x_0) = 0$$

Al resolver la ecuación anterior se obtiene

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Las aproximaciones sucesivas a la raíz se escriben como

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

Obtener la primera derivada de una función dada puede ser una tarea difícil o imposible. En tal caso, se puede evaluar $f'(x_i)$ mediante una aproximación por diferencias, en vez de en forma analítica. Por ejemplo, se puede aproximar $f'(x_{i-1})$ mediante la aproximación por diferencias hacia adelante,

$$f'(x_{i-1}) = \frac{f(x_{i-1}+h) - f(x_{i-1})}{h}$$

donde h es un valor pequeño - por ejemplo $h = 0{,}001$.

Los errores pequeños en la aproximación por diferencias no tienen un efecto observable en la razón de convergencia del método de Newton. La precisión del resultado final no se ve afectada por la aproximación por diferencias.

El método de Newton se puede aplicar para hallar raíces complejas. Si el lenguaje de programación permite variables complejas, se puede aplicar fácilmente al caso de las raíces complejas un programa de computadora diseñado sólo para raíces reales.

Algoritmo de Newton-Raphson

Para encontrar una solución de $f(x) = 0$ dada una aproximación inicial p_0:

Entrada: aproximación inicial p_0; tolerancia *TOL*; número máximo de iteraciones N_0.

```
Hacer i igual a 1
Mientras i sea menor o igual que N0
   Hacer p = p0 - f(p0)/f'(p0) (Calcular pi)
   Si | p - p0 | < TOL entonces
        Mostrar (p); (Procedimiento completado satisfactoriamente)
        Parar
        Hacer i igual a i + 1
        Hacer p0 igual a p (Redefinir p0)
Fin de Mientras
Mostrar ('El método fracasó después de N0 iteraciones')
             (Procedimiento completado sin éxito)
Parar
```

3.5 Método de la Secante

Este método es muy similar al de Newton. La principal diferencia con el método de Newton es que *f'* se aproxima utilizando los dos valores de iteraciones consecutivas de *f*. Esto elimina la necesidad de evaluar tanto a *f* como a *f'* en cada iteración. Por lo tanto, el método de la secante es más eficiente, particularmente cuando *f* es una función en la que se invierte mucho tiempo al evaluarla. El método de la secante también está íntimamente ligado con el método de la falsa posición, ya que ambos se basan en la fórmula de interpolación lineal, pero el primero utiliza extrapolaciones, mientras que el segundo utiliza únicamente interpolaciones.

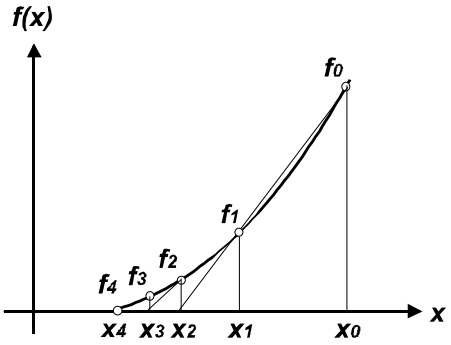

FIGURA 3.5

Las aproximaciones sucesivas para la raíz en el método de la secante están dadas por

$$x_n = x_{n-1} - y_{n-1}\frac{x_{n-1} - x_{n-2}}{y_{n-1} - y_{n-2}}, \quad n = 2, 3\ldots$$

donde x_0 y x_1 son dos suposiciones iniciales para comenzar la iteración.

Si los x_{n-1} y x_n consecutivos son muy cercanos, entonces y_{n-1} y y_n están muy cercanos, por lo que aparece un error de redondeo significativo en la ecuación anterior. Este problema se puede evitar de dos formas: (a) cuando y_n es menor que un valor fijado de antemano, x_{n-2} y y_{n-2} quedan fijos (o congelados) de ahí en adelante, o (b) x_{n-2} y y_{n-2} se reemplazan por $x_{n-2} + x$ y $y(x_{n-2} + x)$ donde x es un número pequeño prescrito pero lo suficientemente grande como para evitar serios errores de redondeo, El método de la secante puede converger a una raíz no deseada o puede no converger del todo si la estimación inicial no es buena.

Algoritmo de la Secante

Para encontrar una solución de $f(x) = 0$, dadas las aproximaciones iniciales p_0 y p_1:

Entrada: aproximaciones iniciales p_0, p_1; tolerancia *TOL*; número máximo de iteraciones N_0
Salida: solución aproximada *p* o mensaje de fracaso

Hacer *i* igual a 2
Hacer $q_0 = f(p_0)$
Hacer $q_1 = f(p_1)$
Mientras *i* sea menor o igual que N_0
  Hacer $p = p_1 - q_1(p_1 - p_0)/(q_1 - q_0)$ (Calcular p_i)
  Si $|p - p_1| < TOL$ entonces
    Mostrar *(p)*; (Procedimiento completado satisfactoriamente)
    Parar
  Fin de Si
  Hacer *i* igual a *i + 1*
  Hacer p_0 igual a p_1; (Redefinir p_0, q_0, p_1, q_1)
  Hacer q_0 igual a q_1
  Hacer p_1 igual a *p*
  Hacer q_1 igual a *f(p)*
  Mostrar ('El método fracasó después de N_0 iteraciones')
    (Procedimiento completado sin éxito)
  Parar

3.6 Problemas

1. Determine la raíz positiva de $x^2 - 0{,}9x - 1{,}52 = 0$ en el intervalo *[1,2]* mediante el método de bisección, con una tolerancia de 0,001.
2. Calcula la raíz de $tan(x) = 3{,}5$ en el intervalo *[0,p]* mediante el método de bisección, con una tolerancia de 0,005.
3. Codifique el algoritmo correspondiente al Método de Bisección y determine un intervalo de tamaño 0,5 para cada raíz positiva de las siguientes ecuaciones

   i) $f(x) = 0{,}5e^{x/3} - sen(x) = 0,\ x > 0$
   ii) $f(x) = log_e(1 + x) - x^2 = 0$

4. Encuentre todas las raíces positivas de las ecuaciones siguientes mediante el método de bisección con una tolerancia de 0,001

   i) $tan(x) - x + 1 = 0,\ 0 < x < 3p$
   ii) $sen(x) - 0{,}3e^x = 0,\ x > 0$
   iii) $-x^3 + x + 1 = 0$
   iv) $16x^5 - 20x^3 + x^2 + 5x - 0{,}5 = 0$

5. Determine las raíces de las siguientes ecuaciones mediante el método de la falsa posición modificada

   i) $f(x) = 0{,}5exp(x/3) - sen(x),\ x > 0$
   ii) $f(x) = log(1+x) - x^2$
   iii) $f(x) = exp(x) - 5x^2$
   iv) $f(x) = x^3 + 2x - 1 = 0$
   v) $f(x) = (x + 2)^{1/2}$

6. La función de transferencia de un sistema está dada por

$$F(s) = \frac{H(s)}{1 + G(s)H(s)}$$

donde

$$G(s) = \frac{1}{s}\exp(-0{,}1s), H(s) = K$$

Busque las raíces de la ecuación característica *1 + G(s)H(s) = 0* para *K= 1, 2 y 3* mediante el método de la falsa posición modificada.

7. Encontrar aproximaciones a 10^{-4} de todos los ceros reales de los siguientes polinomios usando el método de Newton.

i) $P(x) = x^3 - 2x^2 - 5$
ii) $P(x) = x^3 + 3x^2 - 1$
iii) $P(x) = x^3 - x - 1$
iv) $P(x) = x^4 + 2x^2 - x - 3$

8. $P(x) = 10x^3 - 8{,}3x^2 + 2{,}295x - 0{,}21141 = 0$ tiene una raíz en $x = 0{,}29$. Use el método de Newton con una aproximación inicial $x_0 = 0{,}28$ para tratar de encontrar esta raíz. ¿Qué pasa?. Suponga que la única raíz que se desea es $x = 0{,}29$; ¿cómo podría obtener una aproximación lo suficientemente buena para que el método de Newton converja a $x = 0{,}29$?.

9. Codifique, empleando el lenguaje de programación de su preferencia, el algoritmo de la secante.

10. Codifique, empleando el lenguaje de programación de su preferencia, el algoritmo de Newton.

4

Interpolación y Aproximación de Funciones

En este capítulo se considerará la pregunta: “Dados los valores de una función desconocida correspondiente a ciertos valores de *x*, ¿cuál es el comportamiento de la función?”. El propósito en determinar el comportamiento de la función, tal como se evidencia por las muestras de los pares de datos *(x, f(x))*, tiene varias interrogantes. Se desearía aproximar otros valores de la función para valores e *x* no tabulados (interpolación y extrapolación), y estimar la integral de *f(x)* y su derivada. Estos últimos objetivos conducirán a formas de resolver ecuaciones diferenciales.

La estrategia que se utilizará para aproximarse a los valores desconocidos de la función es directa. Se encontrará un polinomio que satisfaga un conjunto de puntos seleccionados $(x_i, f(x_i))$ y, se supondrá que el polinomio y la función se comportan casi de la misma manera, sobre el intervalo en cuestión. Entonces los valores de los polinomios, deben ser estimaciones razonables de los valores de la función desconocida. Cuando el polinomio es de primer grado, éste conduce a la interpolación lineal familiar. Estaremos interesados en polinomios de grado mayor que el primero, y así poder aproximar funciones que están lejos de ser lineales, o poder obtener buenos valores a partir de una tabla con un mayor espaciamiento.

Una de las razones más importantes para el estudio del tema, está en el trabajo de detalle para la integración y diferenciación numérica.

Si se desea encontrar un polinomio que pase a través de los mismos puntos que nuestra función desconocida, se puede establecer un sistema de ecuaciones que involucre los coeficientes del polinomio. Por ejemplo, supóngase que se desea ajustar un polinomio cúbico a los siguientes datos:

x	1,0	2,7	3,2	4,8	5,6
f(x)	14,2	17,8	22,0	38,3	51,7

El polinomio interpolante podrá ser a lo sumo, de grado uno menos que el número de puntos disponibles; en esta caso particular el grado máximo del polinomio de interpolación será cuatro. Se construye entonces un sistema de ecuaciones con polinomios de grado cuatro de la forma $ax^4 + bx^3 + cx^2 + dx + e$,

$$
\begin{aligned}
\text{donde } x = 1{,}0 \quad & a(1{,}0)^4 + b(1{,}0)^3 + c(1{,}0)^2 + d(1{,}0) + e = 14{,}2 \\
x = 2{,}7 \quad & a(2{,}7)^4 + b(2{,}7)^3 + c(2{,}7)^2 + d(2{,}7) + e = 17{,}8 \\
x = 3{,}2 \quad & a(3{,}2)^4 + b(3{,}2)^3 + c(3{,}2)^2 + d(3{,}2) + e = 22{,}0 \\
x = 4{,}8 \quad & a(4{,}8)^4 + b(4{,}8)^3 + c(4{,}8)^2 + d(4{,}8) + e = 38{,}3 \\
x = 5{,}6 \quad & a(5{,}6)^4 + b(5{,}6)^3 + c(5{,}6)^2 + d(5{,}6) + e = 51{,}7
\end{aligned}
$$

La solución de estas ecuaciones proporciona el polinomio. Entonces es posible estimar los valores de la función en algún valor de *x*, por ejemplo *x= 3,0*, sustituyendo este valor de *x* en el polinomio.

Se busca una mejor y más sencilla forma de encontrar los polinomios interpolantes. El procedimiento anterior es algo engorroso, en especial si se desea que el nuevo polinomio se ajuste o pase por el punto (5,6; 51,7), o para ver que diferencia resultaría si se utilizara un polinomio cuadrático en lugar de uno cúbico.

4.1 Interpolación y el Polinomio de Lagrange

Considérese el problema de determinar un polinomio de grado 1 que pase por los puntos distintos (x_0, y_0) y (x_1, y_1). Este problema es el mismo que el de aproximar una función f, para la cual $f(x_0)= y_0$ y $f(x_1)= y_1$, por medio de un polinomio de primer grado, interpolando entre, o coincidiendo con, los valores de f en los puntos dados.

Se el polinomio

$$P(x) = \frac{(x - x_1)}{(x_0 - x_1)} y_0 + \frac{(x - x_0)}{(x_0 - x_1)} y_1$$

Cuando $x= x_0$, $P(x_0)= y_0= f(x_0)$ y cuando $x= x_1$, $P(x_1)= y_1= f(x_1)$ así que P tiene las propiedades requeridas.

La técnica usada para construir a P es el método de "interpolación" usado con frecuencia en las tablas trigonométricas o logarítmicas. Lo que puede ser no tan obvio es que P es el único polinomio de grado 1 o menor con la propiedad de interpolación.

Para generalizar el concepto de interpolación lineal, considérese la construcción de un polinomio de grado a lo más n que pase por los $n+1$ puntos $(x_0, f(x_0))$, $(x_1, f(x_1))$, $(x_2, f(x_2))$, ..., $(x_n, f(x_n))$. El polinomio lineal que pasa por $(x_0, f(x_0))$, $(x_1, f(x_1))$ se construye usando los cocientes

$$L_0(x) = \frac{(x - x_1)}{(x_0 - x_1)} \qquad \text{y} \qquad L_1(x) = \frac{(x - x_0)}{(x_0 - x_1)}$$

Cuando $x= x_0$, $L_0(x_0)= 1$ mientras que $L_1(x_0)= 0$. Cuando $x= x_1$, $L_0(x_1)= 0$ mientras que $L_1(x_1)= 1$. Para el caso general es necesario construir, para cada $k= 0, 1, ..., n$, un cociente $L_{n,k}(x)$ con la propiedad de que $L_{n,k}(x_i)= 0$ cuando $i \neq k$ y $L_{n,k}(x_k)= 1$. Entonces,

$$L_{n,k}(x) = \frac{(x - x_0)...(x - x_{k-1})(x - x_{k+1})...(x - x_n)}{(x_k - x_0)...(x_k - x_{k-1})(x_k - x_{k+1})...(x_k - x_n)} = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{(x - x_i)}{(x_k - x_i)}$$

Ahora que se conoce la forma de $L_{n,k}$ es fácil describir al polinomio interpolante. Este polinomio se llama *polinomio interpolante de Lagrange* y se define a continuación.

Si $x_0, x_1, ..., x_n$ son $(n+1)$ número diferentes y f es una función cuyos valores están dados en estos puntos, entonces existe un único polinomio P de grado a lo más n con la propiedad de que

$f(x_k)= P(x_k)$ para cada $k= 0, 1, ..., n.$

Este polinomio está dado por

$$P(x) = f(x_0)L_{n,0}(x) + ... + f(x_n)L_{n,n}(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x)$$

donde

$$L_{n,k}(x) = \frac{(x-x_0)...(x-x_{k-1})(x-x_{k+1})...(x-x_n)}{(x_k-x_0)...(x_k-x_{k-1})(x_k-x_{k+1})...(x_k-x_n)} = \prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{(x-x_i)}{(x_k-x_i)}, \quad \text{para cada } k = 0, ..., n$$

4.2 Polinomio Interpolantes en puntos con igual separación

El problema se simplifica considerablemente, si los valores de la función están dados a intervalos regulares de la variable independiente, así que primero se considerará este caso.

4.2.1 Tabla de Diferencias

Resulta conveniente arreglar los datos en una tabla con los valores *x* en orden ascendente. Además de las columnas para *x* y *f(x)*, se deberán tabular las diferencias de los valores funcionales. La tabla que se muestra a continuación es llamada *tabla de diferencias*.

x	*f(x)*	*Df(x)*	$D^2f(x)$	$D^3f(x)$	$D^4f(x)$	$D^5f(x)$	$D^6f(x)$
0,0	0,000	0,203	0,017	0,024	0,020	0,032	0,127
0,2	0,203	0,220	0,041	0,044	0,052	0,159	
0,4	0,423	0,261	0,085	0,096	0,211		
0,6	0,684	0,346	0,181	0,307			
0,8	1,030	0,527	0,488				
1,0	1,557	1,015					
1,2	2,572						

Los términos calculados en la tabla de diferencias, permiten determinar los coeficientes de polinomios interpolantes. Es convencional que la letra *h* sea la diferencia uniforme de los valores *x*, es decir, $h = Dx$. Utilizando subíndices para representar el orden de los valores *x* y *f(x)* se definen las primeras diferencias de la función como: $Df_1 = f_2 - f_1$, $Df_2 = f_3 - f_2$, ..., $Df_i = f_{i+1} - f_i$. De una manera semejante se definen las diferencias segundas y de orden más elevado. La *n*-ésima diferencia de la función se define como:

$$\Delta^n f_i = f_{i+n} - nf_{i+n-1} + \frac{n(n-1)}{2!} f_{i+n-2} - \frac{n(n-1)(n-2)}{3!} f_{i+n-3} + ...$$

El patrón de los coeficientes en la ecuación anterior, es el arreglo de valores en el desarrollo binomial. Las diferencias segundas y de mayor orden se obtienen por sustracción de las diferencias anteriores.

Cuando *f(x)* se comporta como un polinomio para el conjunto dado de datos, la tabla de diferencias tiene propiedades especiales. La siguiente tabla muestra una función sobre el dominio $x = 1$ hasta $x = 6$, y *f(x)* se comporta como x^3.

x	$f(x)$	$Df(x)$	$D^2f(x)$	$D^3f(x)$	$D^4f(x)$	$D^5f(x)$	$D^6f(x)$
0	0	1	6	6	0	0	0
1	1	7	12	6	0	0	
2	8	19	18	6	0		
3	27	37	24	6			
4	64	61	30				
5	125	91					
6	216						

Observe que las terceras diferencias son iguales en todos los casos, así que las cuartas y superiores serán cero. Es posible demostrar que las diferencias de *n*-ésimo orden de un polinomio de *n*-ésimo grado serán iguales, por lo que las diferencias de (*n+1*)-ésimo orden serán cero.

4.2.2 Polinomio de Avance de Newton-Gregory

Cuando la función que ha sido tabulada, se comporta como un polinomio (esto se puede decir observando que sus diferencias de orden *n*-ésimo sean iguales o casi), se le puede aproximar al polinomio que se le parece. El problema consiste entonces en encontrar los medios más sencillos para escribir el polinomio de *n*-ésimo grado correspondiente.

Quizá las forma más fácil de escribir un polinomio que pasa por un conjunto de puntos equiespaciados, es el polinomio de avance de Newton-Gregory:

$$\begin{aligned} P_n(x_s) &= f_0 + s\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + \frac{s(s-1)(s-2)}{3!}\Delta^3 f_0 + \ldots \\ &= f_0 + \binom{s}{1}\Delta f_0 + \binom{s}{2}\Delta^2 f_0 + \binom{s}{3}\Delta^3 f_0 + \binom{s}{4}\Delta^4 f_0 + \ldots \\ &= \sum_{n=0}^{k}\binom{s}{n}\Delta^n f_0 \end{aligned}$$

En esta ecuación la notación $\binom{s}{n}$ viene dada por el número de combinaciones de *s* cosas tomadas *n* a la vez, lo cual es conocido como *razones factoriales*. Es posible demostrar que $P_n(x)$ formada de acuerdo a la ecuación anterior, se ajusta a todos los *n+1* puntos dados.

Si, sobre el dominio de x_0 a x_n, $P_n(x)$ y $f(x)$ tienen los mismos valores en los datos tabulados de *x*, resulta razonable suponer que estarán cercanos en los valores intermedios de *x*. Esta suposición es la base del uso de $P_n(x)$ como un polinomio interpolante. Se hace énfasis en el hecho de que en general $f(x)$ y $P_n(x)$ no serán la misma función. Por lo tanto, hay algún error que se debe esperar en la estimación de tal interpolación. Usamos el polinomio de la ecuación anterior como un polinomio de interpolación, dejando a *s* tomar valores no enteros. Obsérvese que, para cualquier valor de *x*,

$$s = \frac{x - x_0}{h}$$

4.2.3 Polinomio de Retroceso de Newton-Gregory

Algunas veces es conveniente escribir el polinomio interpolante en otras formas. El polinomio de retroceso de Newton-Gregory es:

$$P_n(x) = f_0 + \binom{s}{1}\nabla f_0 + \binom{s+1}{2}\nabla^2 f_0 + \binom{s+2}{3}\nabla^3 f_0 + \binom{s+3}{4}\nabla^4 f_0 + \ldots.$$
$$= \sum_{n=0}^{k}\binom{s+n-1}{n}\nabla^n f_0$$

donde $\tilde{N}f_i = f_i - f_{i-1}$

4.3 Polinomio Interpolantes en puntos no equiespaciados

Las fórmulas de interpolación de Newton-Gregory descritas en la sección anterior se restringen a puntos con igual separación. Sin embargo, a menudo aparece a necesidad de escribir un polinomio para puntos con separación no uniforme. El modelo de interpolación de Newton-Gregory puede extenderse a los puntos con separación no uniforme utilizando las diferencias divididas.

4.3.1 Tabla de Diferencias Divididas

Supóngase que P_n es el polinomio de Lagrange de grado a lo más n que coincide con la función f en los números distintos $x_0, x_1, \ldots, x_n$. Las diferencias divididas de f con respecto a $x_0, x_1, \ldots, x_n$, se pueden derivar demostrando que P_n tiene la representación

$$P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1)+\ldots+a_n(x-x_0)(x-x_1)\text{---}(x-x_{n-1})$$

con constantes apropiadas $a_0, a_1, \ldots, a_n$.

Para determinar la primera de estas constantes, a_0, nótese que si $P_n(x)$ puede escribirse en la forma de la ecuación anterior, entonces evaluando P_n en x_0 deja solamente el término constante a_0; esto es, $a_0 = P_n(x_0) = f(x_0)$.

Similarmente, cuando P_n se evalúa en x_1, los únicos términos distintos de cero en la evaluación de $P_n(x_1)$ son la constante y el término lineal,

$$f(x_0) + a_1(x_1-x_0) = P_n(x_1) = f(x_1);$$

así que

$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Aquí se introduce lo que se conoce como *notación de diferencia dividida*. La diferencia dividida cero de la función f, con respecto a x_i, se denota por $f[x_i]$ y es simplemente la evaluación de f en x_i,

$$f[x_i] = f(x_i)$$

Las diferencias divididas restantes se definen inductivamente; la primera diferencia dividida de f con respecto a x_i y x_{i+1}, se denota por $f[x_i, x_{i+1}]$ y está definida por

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$$

La k-ésima diferencia dividida de f relativa a $x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+k}$ está dada por

$$f[x_i, x_{i+1}, \ldots, x_{i+k}] = \frac{f[x_i, x_{i+1}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}$$

Entonces el polinomio interpolante P_n es

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)$$
$$+ \ldots + f[x_0, \ldots, x_n](x - x_0)\ldots(x - x_{n-1})$$

o como

$$P_n(x) = f[x_0] + \sum_{k=1}^{n}\left[f[x_0, \ldots, x_k] \prod_{i=0}^{k-1}(x - x_i) \right]$$

Esta ecuación se conoce como la *fórmula de diferencia dividida interpolante de Newton*.

Problemas

1. Escriba la fórmula de interpolación de Lagrange ajustada a los puntos dados en la siguiente tabla:

x_i	0,00	0,25	0,50	0,75	1,00
$f(x_i)$	0,9162	0,8109	0,6931	0,5596	0,4055

2. Ajuste $sen(x)$ en $[0, 2p]$ con el polinomio de interpolación de Lagrange de orden 4 y 8, utilizando puntos con igual separación.

3. Deduzca el polinomio de interpolación de Newton-Gregory hacia adelante que pasa por los puntos dados en la siguiente tabla:

x_i	0,50	1,00	1,50	2,00	2,50	3,00
$f(x_i)$	1,143	1,000	0,828	0,667	0,533	0,428

4. Deduzca el polinomio de interpolación utilizando el método de Newton para puntos no equiespaciados, ajustado a los datos dados en la siguiente tabla:

x_i	0	10	30	50	70	90	100

$f(x_i)$	1,792	1,308	0,801	0,549	0,406	0,317	0,284

5

Diferenciación e Integración Numérica

Se tiene que construir una hoja de techo corrugado usando una máquina que comprime una hoja plana de aluminio convirtiéndola en una cuya sección transversal tiene la forma de una onda de la función seno.

Supongamos que se necesita una hoja corrugada de cuatro pies de longitud, que cada onda tiene una altura de 1 pulgada desde la línea central y que cada onda tiene un período de aproximadamente 2π pulgadas. El problema de encontrar la longitud de la hoja plana consiste en determinar la longitud de arco de la curva dada por *f(x)=sen(x)* de *x=0* a *x=48* (pulgadas). Sabemos, del cálculo, que esta longitud se puede expresar como:

$$L = \int_0^{48} \sqrt{1 + \left(\frac{df(x)}{dx} \right)^2}\, dx = \int_0^{48} \sqrt{1 + (\cos x)^2}\, dx$$

así que el problema se reduce a evaluar esta integral. Aun cuando la función seno es una de las funciones matemáticas más comunes, el cálculo de su longitud de arco da lugar a la llamada integral elíptica de segunda clase, que no se puede evaluar usando métodos ordinarios. En esta sección se estudian métodos aproximados que reducen problemas de este tipo a ejercicios elementales.

5.1 Diferenciación Numérica

La diferenciación numérica, o aproximación por diferencias, se utiliza para evaluar las derivadas de una función por medio de sus valores en los puntos de una retícula. Las aproximaciones por diferencias son importantes en la solución de ecuaciones diferenciales ordinarias y parciales.

Para ilustrar la diferenciación numérica, consideremos una función *f(x)*. Supongamos que se desea evaluar la primera derivada de *f(x)* en $x = x_0$. Si se conocen los valores de *f* en $x_0 - h$, x_0 y $x_0 + h$, donde *h* es el tamaño del intervalo entre dos puntos consecutivos en el eje *x*, entonces se puede aproximar *f'(x)* mediante el gradiente de la interpolación lineal A, B o C. Estas tres aproximaciones se llaman respectivamente las aproximaciones por diferencias *hacia adelante*, *hacia atrás* y *central*. Sus fórmulas matemáticas son como sigue:

a) Aproximación que utiliza A (aproximación por diferencias hacia adelante)

$$f'(x_0) \cong \frac{f(x_0 + h) - f(x_0)}{h}$$

b) Aproximación que utiliza B (aproximación por diferencias hacia atrás)

$$f'(x_0) \cong \frac{f(x_0) - f(x_0 - h)}{h}$$

c) Aproximación que utiliza C (aproximación por diferencias central)

$$f'(x_0) \cong \frac{f(x_0 + h) - f(x_0 - h)}{2h}$$

5.2 Integración Numérica

Los métodos de integración numérica se pueden utilizar para integrar funciones dadas, ya sea mediante una tabla o en forma analítica. Incluso en el caso en que sea posible la integración analítica, la integración numérica puede ahorrar tiempo y esfuerzo si sólo se desea conocer el valor numérico de la integral.

Esta sección analiza los métodos numéricos que se utilizan para evaluar integrales de una variable:

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

así como integrales dobles:

$$I = \int_a^b \int_{u(x)}^{v(x)} f(x, y)dydx$$

donde las funciones *f(x)* y *f(x, y)* pueden estar dadas en forma analítica o mediante una tabla.

Los métodos de integración numérica se obtienen al integrar los polinomios de interpolación. Por consiguiente, las distintas fórmulas de interpolación darán por resultado distintos métodos de integración numérica.

5.2.1 Regla del Trapecio

Esta regla es un método de integración numérica que se obtiene al integrar la fórmula de interpolación lineal. Se escribe en la forma siguiente:

$$I = \int_a^b f(x)dx = \frac{b-a}{2}[f(a) + f(b)] + E$$

donde el primer término del lado derecho es la regla del trapecio (fórmula de integración) y *E* representa el error. En la figura 5.2 se muestra gráficamente la integración numérica por medio de la ecuación anterior. El área sombreada por debajo de la recta de interpolación (la cual puede denotarse como *g(x)*) es igual a la integral calculada mediante la regla del trapecio, mientras que el área por debajo de la curva *f(x)* es el valor exacto. Por lo tanto, el error de la estimación de la integral por este método es igual al área entre *g(x)* y *f(x)*.

La fórmula dada anteriormente se puede extender a varios intervalos y se puede aplicar *N* veces al caso de *N* intervalos con una separación uniforme *h*, para así obtener la regla extendida del trapecio:

$$I = \int_a^b f(x)dx = \frac{h}{2}[f(a) + 2\sum_{j=1}^{N-1} f(a + jh) + f(b)] + E$$

El error de la regla del trapecio se define como

$$E = \int_a^b f(x)dx - \frac{b-a}{2}[f(a) + f(b)]$$

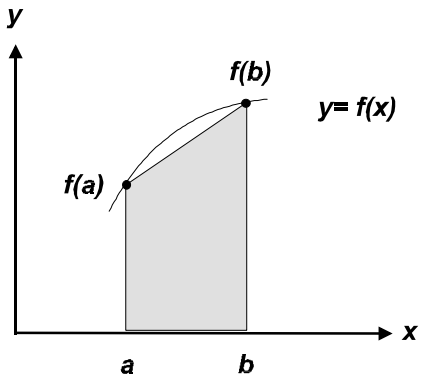

Figura 5.1 - Regla del Trapecio

donde el primer término es la integral exacta y el segundo es la regla del trapecio. Para analizar la ecuación se utilizarán los desarrollos en Serie de Taylor de *f(x)*, *f(a)* y *f(b)* en torno a $x_0 = (a + b) / 2$, con la hipótesis de que *f* es analítica en $a \pounds x \pounds b$.

El error de la regla extendida del trapecio es la suma de los errores en todos los intervalos y viene dada por:

$$E \cong -\frac{1}{12}\frac{(b-a)^3}{N^3}\sum_{i=1}^{N} f''(X_i)$$

donde $h = (b - a) / N$ y X_i es el punto medio entre x_i y x_{i+1}. Si definimos *F"* como el promedio de *f"*, es decir,

$$F'' = \sum_{i=1}^{N} f''(X_i) / N$$

la estimación del error se puede expresar como:

$$E \cong -\frac{1}{12}(b-a)h^2 F''$$

Para un dominio fijo *[a , b]* el error es proporcional a h^2.

5.2.2 Regla de 1/3 de Simpson

La regla de 1/3 de Simpson se basa en la interpolación polinomial cuadrática (de segundo grado). El polinomio de Newton hacia adelante ajustado a tres puntos x_0, x_1, x_2, está dado por:

$$f(x) = f_0 + s(f_1 - f_0) + \frac{s(s-1)}{2}(f_2 - 2f_1 + f_0)$$

entonces,

$$\int_{x_0}^{x_2} f(x)dx = \int_{x_0}^{x_2}(f_0 + s(f_1 - f_0) + \frac{s(s-1)}{2}(f_2 - 2f_1 + f_0))dx$$

donde

$$s = \frac{x - x_0}{h}$$

haciendo el cambio de variable adecuado tenemos

$$\int_{x_0}^{x_1} f(x)dx = h\int_0^2 (f_0 + s(f_1 - f_0) + \frac{s(s-1)}{2}(f_2 - 2f_1 + f_0))ds$$

integrando en *ds* y sustituyendo $x_0 = a$ y $x_2 = b$ se obtiene la regla de 1/3 de Simpson:

$$I = \int_a^b f(x)dx = \frac{h}{3}[f(a) + 4f(X) + f(b)] + E$$

donde $h = (b - a) / 2$ $X = (a + b) / 2$. Esta se puede escribir en la forma equivalente

$$I = \frac{h}{3}[f_0 + 4f_1 + f_2] + E$$

donde $f_i = f(x_i) = f(a + ih)$. El error viene dado por:

$$E \cong -\frac{h^5}{90} f^{iv}(X)$$

El error se anula si *f(x)* es un polinomio de orden menor o igual que 3. La regla 1/3 de Simpson es fácil de aplicar en un computador y su precisión es suficiente para muchas aplicaciones.

La regla extendida de 1/3 de Simpson es una aplicación repetida de la ecuación antes descrita para un dominio dividido en un número par de intervalos. Si denotamos el número total de intervalos por *N* (par), la regla extendida de 1/3 de Simpson se escribe como:

$$I = \frac{h}{3}[f(a) + 4\sum_{\substack{i=1 \\ impar}}^{N-1} f(a + ih) + 2\sum_{\substack{i=2 \\ par}}^{N-2} f(a + ih) + f(b)] + E$$

donde $h = (b - a) / N$; la primera suma es únicamente sobre las *i* impares y la segunda es sólo sobre las *i* pares.

El término del error está dado por

$$E \cong -\frac{N}{2}\frac{h^5}{90} f^{iv}(X) = -(b - a)\frac{h^4}{180} f^{iv}(X)$$

donde $X = (a + b) / 2$.

Para un dominio fijo $[a , b]$ el error es proporcional a h^4. Una aproximación del error puede ser obtenida mediante la siguiente fórmula

$$E \cong -\frac{1}{90} h^5 f_1''''$$

5.2.3 Regla de 3/8 de Simpson

La regla de 3/8 de Simpson se obtiene al integrar una fórmula de interpolación polinomial de tercer grado. Para un dominio $[a, b]$ dividido en tres intervalos, se escribe como

$$I = \int_a^b f(x)dx = \frac{3}{8} h[f_0 + 3f_1 + 3f_2 + f_3] + E$$

donde $h = (b - a) / 3$, $f_i = f(a + ih)$ y E representa el error. El término del error se escribe como

$$E \cong -\frac{3}{80} h^5 f''''(X)$$

donde $X = (a + b) / 2$

La regla extendida de Simpson 3/8 se expresa como

$$I = \int_a^b f(x)dx = \frac{3}{8} h \sum_{\substack{i=0 \\ i \text{ multiplo de } 3}}^{N-3} (f_i + 3f_{i+1} + 3f_{i+2} + f_3) + E$$

La regla extendida de 1/3 se aplica a un número par de intervalos, mientras que la regla extendida de 3/8 se aplica a un número de intervalos que sea múltiplo de tres. Cuando el número de intervalos es impar pero sin ser múltiplo de tres, se puede utilizar la regla de 3/8 para los primeros tres o los últimos tres intervalos, y luego usar la regla de 1/3 para los intervalos restantes, que son un número par. Puesto que el orden del error de la regla de 3/8 es el mismo que el de la regla de 1/3, no se gana mayor exactitud que con la regla de 1/3 cuando uno puede elegir con libertad entre ambas reglas.

5.2.4 Fórmulas de Newton-Cotes

Los métodos de integración numérica que se obtienen al integrar las fórmulas de interpolación de Newton reciben el nombre de fórmulas de Newton-Cotes. La regla del trapecio y las dos reglas de Simpson son casos particulares de las fórmulas de Newton-Cotes, las cuales se dividen en fórmulas cerradas y abiertas.

Las fórmulas cerradas de Newton-Cotes tiene la forma:

$$\int_a^b f(x)dx = \alpha h[w_0 f_0 + w_1 f_1 + w_2 f_2 + \cdots + w_N f_N] + E$$

donde a y w_i son las constantes que aparecen en la tabla que se muestra a continuación y $f_n = f(x_n)$, $x_n = a + n$ y $h = (b - a) / N$

Esta ecuación recibe el nombre de *fórmula cerrada*, debido a que el dominio de integración está cerrado por el primer y último datos.

Tabla 5.1 - Constantes para las fórmulas cerradas de Newton-Cotes

N	**α**	***w_i, i= 0, 1, 2, ..., N***	***E***
1	1/2	1 1	$-\frac{1}{12}h^3 f''$
2	1/3	1 4 1	$-\frac{1}{90}h^5 f^{iv}$
3	3/8	1 3 3 1	$-\frac{3}{80}h^5 f^{iv}$
4	2/45	7 32 12 32 7	$-\frac{8}{945}h^7 f^{vi}$
5	5/288	19 75 50 50 75 19	$-\frac{275}{12096}h^7 f^{vi}$

Por otra parte, la integración de la ecuación pudiera extenderse más allá de los puntos extremos de los datos dados. Las fórmulas abiertas de Newton-Cotes se obtienen al extender la integración hasta un intervalo a la izquierda del primer dato y un intervalo a la derecha del último. Dichas fórmulas se escriben como:

$$\int_a^b f(x)dx = \alpha h[w_0 f_0 + w_1 f_1 + w_2 f_2 + \cdots + w_{N+2} f_{N+2}] + E$$

donde $h = (b - a) / (N + 2)$. Las constantes a y w_i se listan en la tabla XX, en donde w_0 y w_{N+2} se igualan a cero debido a que corresponden a los extremos del dominio. Puesto que w_0 y w_{N+2} se anulan, f_0 y f_{N+2} son datos ficticios, que en realidad no son necesarios.

Tabla 5.2 - Constantes para las fórmulas abiertas de Newton-Cotes

N	α	w_i, *i= 0, 1, 2, ..., N+2*	*E*
1	3/2	0 1 1 0	$-\frac{1}{12}h^3 f''$
2	4/3	0 2 -1 2 0	$-\frac{1}{90}h^5 f^{iv}$
3	5/24	0 11 1 1 11 0	$-\frac{3}{80}h^5 f^{iv}$
4	6/20	0 11 -14 26 -14 11 0	$-\frac{8}{945}h^7 f^{vi}$
5	7/1440	0 611 -453 562 562 -453 611 0	$-\frac{275}{12096}h^7 f^{vi}$

Si comparamos una fórmula abierta con una cerrada utilizando el mismo número *N* de datos, el error de la fórmula abierta es significativamente mayor que el de la fórmula cerrada. Por otro lado, se pueden utilizar las fórmulas abiertas cuando no se dispone de los valores de la función en los límites de integración.

5.2.5 Cuadratura de Gauss

Los métodos de integración presentados se establecieron para valores de *x* equiespaciados; esto significa que los valores *x* son predeterminados. Entonces, con una fórmula de tres términos hay tres parámetros., los coeficientes (factores de ponderación) aplicados a cada uno de los valores funcionales. Una fórmula con tres parámetros corresponde a un polinomio de segundo grado, uno menos que el número de parámetros. Gauss observó que si se elimina el requisito de que la función sea evaluada en valores *x* predeterminados, una fórmula de tres términos contendrá seis parámetros (los tres valores *x* ahora desconocidos, más tres ponderaciones), y corresponderá a un polinomio interpolante de grado 5. Las fórmulas basadas en este principio se llaman *fórmulas de cuadratura gaussiana*. Sólo pueden ser aplicadas cuando *f(x)* es conocida explícitamente, de manera que pueda ser evaluada en cualquier valor deseado de *x*.

Se determinarán los parámetros en el caso más sencillo de una fórmula de dos términos que contiene cuatro parámetros desconocidos:

$$\int_{-1}^{1} f(t)dt = af(t_1) + bf(t_2)$$

Se utiliza un intervalo simétrico de integración para simplificar la aritmética, y se llamará a la variable *t*. La fórmula es válida para cualquier polinomio de grado 3; por tanto se cumplirá si $f(t) = t^3$, $f(t) = t^2$, $f(t) = t$, y para $f(t)=1$:

$$f(t) = t^3: \qquad \int_{-1}^{1} t^3 dt = 0 = at_1^3 + bt_2^3;$$

$$f(t) = t^2: \qquad \int_{-1}^{1} t^2 dt = \frac{2}{3} = at_1^2 + bt_2^2;$$

$$f(t) = t: \qquad \int_{-1}^{1} tdt = 0 = at_1 + bt_2;$$

$$f(t) = 1: \qquad \int_{-1}^{1} 1 dt = 2 = a + b.$$

Multiplicando la tercera ecuación por t_1^2, y restando a la primera, se tiene

$$0 = 0 + b(t_2^3 - t_2 t_1^2) = b(t_2)(t_2 - t_1)(t_2 + t_1)$$

Esta ecuación es satisfecha por cualquiera de los siguientes valores $b=0$, $t_2=0$, $t_1=t_2$ o $t_2=-t_1$. Sólo las últimas de estas posibilidades son satisfactorias, ya que las otras no son válidas o bien reducen la fórmula a un único término. Se selecciona $t_1=-t_2$. Se encuentra entonces que

$$a = b = 1,$$
$$t_2 = -t_1 = \sqrt{\frac{1}{3}} = 0{,}5773,$$
$$\int_{-1}^{1} f(t)dt = f(-0{,}5773) + f(0{,}5773).$$

Es notable que sumando estos dos valores de la función, de el valor exacto de la integral para un polinomio cúbico sobre el intervalo de -1 a 1.

Supóngase que los límites de integración son de *a* a *b*, y no de -1 a 1 para los cuales se dedujo esta fórmula. Para usar los parámetros tabulados de la cuadratura gaussiana, se debe cambiar el intervalo de integración a (-1,1) por medio de un cambio de variables. Se reemplaza la variable dada por otra en la que esté relacionada linealmente de acuerdo con el siguiente esquema:

Si se tiene

$$x = \frac{(b-a)t + b + a}{2}, \quad \text{de manera que } dx = \left(\frac{b-a}{2}\right)dt,$$

entonces

$$\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{(b-a)t + b + a}{2}\right)dt.$$

El poder del método gaussiano, se debe al hecho de que sólo se necesitan dos evaluaciones funcionales. Si se usa la regla trapezoidal, que requiere también sólo dos evaluaciones, el margen de error sería significativamente mayor. La regla de Simpson 1/3, requiere de tres evaluaciones funcionales y genera un error algo más grande que la cuadratura gaussiana.

Para obtener los parámetros de las fórmulas gaussianas de mayor orden, se debe resolver de la misma forma un conjunto de ecuaciones simultáneas. EL conjunto de ecuaciones que resulta de escribir la definición de *f(t)*, como una sucesión de polinomios, no se resuelve con facilidad, debido a la no linealidad con respecto a *t*. En este caso se emplea la teoría de polinomios ortogonales; los valores de *t* que satisfacen las ecuaciones, son las raíces de los polinomios de Legendre.

Los polinomios de Legendre se definen de forma recursiva:

$$(n+1)L_{n+1}(x) - (2n+1)xL_n(x) + nL_{n+1}(x) = 0$$

con $L_0(x) = 1$, $L_1(x) = x$.
Entonces $L_2(x)$ es

$$L_2(x) = \frac{3xL_1(x) - (1)L_0(x)}{2} = \frac{3}{2}x^2 - \frac{1}{2}$$

aquí las raíces son $\pm\sqrt{1/3}$ =± 0,5773, precisamente los valores de *t* para la fórmula de dos términos.
Utilizando la relación recursiva, se encuentra

$$L_3(x) = \frac{5x^3 - 3x}{2}$$

$$L_4(x) = \frac{35x^4 - 30x^2 + 3}{8}, \quad \text{etc.}$$

Los factores de ponderación y los valores *t* de la cuadratura gaussiana hasta cuatro términos, se muestran en la siguiente tabla.

Tabla 5.3 - Valores de la cuadratura gaussiana

Número de términos	Valores de *t*	Factores de ponderación	Válido hasta el grado
2	-0,57735027	1,0	3
	0,57735027	1,0	
3	-0,77459667	0,55555555	5
	0,0	0,88888889	
	0,77459667	0,55555555	
4	-0,86113631	0,34785485	7
	-0,33998104	0,65214515	
	0,33998104	0,65214515	
	0,86113631	0,34785485	
5	-0,906179845	0,23692689	9
	-0,538469310	0,47862867	
	0,0	0,56888889	
	0,538469310	0,47862867	
	0,906179845	0,23692689	

5.2.6 Integración Numérica con Límites Infinitos

En esta sección se estudiarán las integrales del siguiente tipo:

$$\int_{-\infty}^{\infty} \exp(-x^2)dx$$

En este caso, la integral se extiende en un dominio infinito. Sin embargo, una función integrable en un dominio infinito o semi-infinito es casi nula, excepto en cierta parte del dominio. La contribución principal a la integral proviene de un dominio relativamente pequeño, en donde la función es distinta de cero en forma significativa.

Sea *f(x)* analítica en (-∞ , ∞), el método más eficiente para la integración numérica de

$$\int_{-\infty}^{\infty} f(x)dx$$

es la regla extendida del trapecio la cual está definida por:

$$I = h\sum_{i=-M}^{M} f(x_i)$$

donde $x_i = ih$ y M es un entero suficientemente grande.

5.2.7 Integración Numérica en un Dominio Bidimensional

Consideremos un dominio, en el que las fronteras izquierda y derecha son segmentos de recta verticales y las fronteras inferior y superior están dadas por curvas $y = d(x)$ y $y = c(x)$, respectivamente. La doble integral en el dominio se escribe como:

$$I = \int_a^b \left[\int_{c(x)}^{d(x)} f(x,y)dy \right] dx$$

Sin embargo, los problemas de dobles integrales no siempre se pueden escribir en la forma de la ecuación anterior y a menudo tiene formas distintas tales como:

$$I = \int_a^b \int_{c(x)}^{d(x)} f(x,y)dydx \quad \text{o} \quad I = \int_a^b dx \int_{c(x)}^{d(x)} dyf(x,y)$$

En cualquiera de estos casos, el problema debe reescribirse en la forma de la ecuación primera, antes de proseguir con la integración numérica.

El principio general de la integración numérica, es la doble aplicación de un método de integración numérica para las integrales de una sola variable, una vez para la dirección *y* y otra vez para la dirección *x*. Si definimos

$$G(x) \equiv \int_{c(x)}^{d(x)} f(x,y)dy$$

entonces, la ecuación original puede reexpresarse como

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

en la cual es posible aplicar cualquiera de los métodos de integración anteriormente descritos.
Apliquemos la regla extendida del trapecio al problema de integración doble:

$$I = \int_a^b \left[\int_{c(x)}^{d(x)} f(x,y)dy \right] dx$$

El rango de integración *[a , b]* se divide en *N* intervalos con igual separación, con una tamaño del intervalo dado por $h_x = (b - a) / N$. Los puntos de la retícula se denotarán como $x_0, x_1, x_2, ..., x_N$. Al aplicar la regla del trapecio en el eje *x*, obtenemos

$$\begin{aligned} I = \frac{h_x}{2}[& \int_{c(x_0)}^{d(x_0)} f(x_0,y)dy + 2\int_{c(x_1)}^{d(x_1)} f(x_1,y)dy \\ & + 2\int_{c(x_2)}^{d(x_2)} f(x_2,y)dy + \cdots + \int_{c(x_N)}^{d(x_N)} f(x_N,y)dy] \end{aligned}$$

pudiendo esta reescribirse en forma más compacta como

$$I = (h_x / 2)[G(x_0) + 2G(x_1) + 2G(x_2) + \cdots + G(x_N)]$$

donde

$$G(x) = \int_{c(x_i)}^{d(x_i)} f(x_i,y)dy$$

Al evaluar esta última ecuación, el dominio de integración $[c(x_i), d(x_i)]$ se divide en *N* intervalos con una tamaño de

$$h_y = \frac{1}{N}[d(x_i) - c(x_i)]$$

Los valores *y* de los puntos de la retícula se denotan como $y_{i,0}, y_{i,1}, y_{i,2}, ..., y_{i,N}$. Entonces, la integración mediante la regla extendida del trapecio da como resultado

$$\begin{aligned} G(x_i) &= \int_{c(x_i)}^{d(x_i)} f(x_i,y)dy \\ &= \frac{h_y}{2}\left[f(x_i,y_{i,0}) + 2f(x_i,y_{i,1}) + 2f(x_i,y_{i,2}) + \cdots + f(x_i,y_{i,N}) \right] \end{aligned}$$

La regla del trapecio utilizada en las integrales anteriores puede reemplazarse para cualquiera otro método de integración tales como Simpson, o las fórmulas abiertas o cerradas de Newton-Cotes.

Problemas

1. Evalúe las siguientes integrales utilizando la regla extendida del trapecio con intervalos de *N = 2, 4, 8, 16* y *32*.

   a) $3x^3 + 5x - 1$ [0, 1]
   b) $x^3 - 2x^2 + x + 2$ [0, 3]
   c) $x^4 + x^3 - x^2 + x + 3$ [0, 1]
   d) tan (x) [0, π / 4]
   e) e^x [0, 1]
   f) 1 / (2 + x) [0, 1]

2. Dados los siguientes valores

x	*f(x)*
0,0	0
0,1	2,1220
0,2	3,0244
0,3	3,2568
0,4	3,1399
0,5	2,8579
0,6	2,5140
0,7	2,1639
0,8	1,8358

   evalúe la integral de *f(x)* entre *0* y *0,8* empleando la regla extendida del trapecio.

3. Repita el ejercicio 1 utilizando la regla de Simpson con intervalos de *N = 4, 8* y *16*.

4. Obtenga la regla de 1/3 de Simpson integrando el polinomio de interpolación de Newton hacia adelante ajustado en x_0, $x_0 + h$ y $x_0 + 2h$.

5. Demuestre que las siguientes fórmulas para la regla extendida del trapecio son equivalentes y señale cuál es más eficiente computacionalmente. Justifique.

$$I = \int_a^b f(x)dx = \frac{h}{2}[f(a) + 2\sum_{j=1}^{N-1} f(a + jh) + f(b)]$$

$$I = \int_a^b f(x)dx = \frac{h}{2}\sum_{i=1}^{N-1}(f_i + f_{i+1})$$

6. Repita el ejercicio 2 utilizando la regla de Simpson.

7. Evalúe la integral de las siguientes funciones en el intervalo indicado, utilizando la regla de Simpson con intervalos de *N= 2, 4, 8, 16* y *32*.

   a) $y = \frac{1}{2 + \cos(x)}$ [0, π]

b) $y = \dfrac{\log(1+x)}{x}$ [1, 2]

c) $y = \dfrac{1}{1+\text{sen}^2(x)}$ [0, π/2]

8. Repita el ejercicio 1 utilizando la regla de Simpson con *N= 3, 7* y *11* intervalos.

9. Suponga que usted es un arquitecto y planea utilizar un gran arco de forma parabólica dado por *y=0,1x(30-x)* metros donde *y* es la altura desde el piso y *x* está en metros. Calcule la longitud total del arco utilizando la regla extendida de Simpson (divida el dominio desde *x = 0* hasta *x = 30* metros en 10 intervalos de la misma longitud). La longitud total del arco está dada por

$$L = \int_0^{30} \sqrt{1+(dy/dx)^2}\,dx$$

10. Evalúe la siguiente integral impropia en forma tan exacta como sea posible, utilizando la regla extendida del trapecio.

$$\int_{-\infty}^{\infty} \frac{\exp(-x^2)}{1+x^2}\,dx$$

11. Utilice la regla extendida de Simpson con 10 intervalos en cada dirección para evaluar la integral doble

$$I = \int_0^p \int_0^{\text{sen}(x)} \exp(-x^2 - y^2)\,dydx$$

12. Evalúe la siguiente integral doble mediante la regla de 1/3 de Simpson

$$I = \int_1^2 \int_0^{2-0,5x} \sqrt{x+y}\,dydx$$

13. Con *n = m= 3* aproxime las siguientes integrales dobles y compare con el valor exacto

a) $\int_{2,1}^{2,2} \int_{1,3}^{1,4} xy^2\,dydx$ b) $\int_0^1 \int_0^{\pi} (y\,\text{sen}\sqrt{x})\,dydx$

c) $\int_0^1 \int_1^2 \int_0^{0,5} e^{yzx}\,dxdydz$ d) $\int_{1,0}^{1,1} \int_0^x (x^2 + \sqrt{y})\,dydx$

14. Utilizando N=2 subintervalos, determine las estimaciones de las reglas trapezoidal y de Simpson para

$$I = \int_{-1}^{1} \sqrt{(1+x^3)}\,dx$$

15. Demuestre que la regla de Simpson es exacta para $f(x) = Ax^2 + Bx^2 + Cx + D$ considerando cada uno de los cuatro términos por separado sobre el intervalo $a \pounds x \pounds b$, con cualquier número (par) de subintervalos *N* a su elección.

16. Verifique que

$$\int_0^2 s(s-1)(s-2)\,ds = 0$$

17. Evalúe por medio de una fórmula de cuadratura gaussiana de tres términos.

$$\int_0^1 \frac{\operatorname{sen} x}{x}\, dx$$

18. Por computación con las fórmulas de cuadratura gaussiana de complejidad creciente, determine cuántos términos son necesarios para evaluar $\int_{1,8}^{3,4} e^x\, dx$ hasta cinco decimales de precisión. El valor exacto de la integral es 23,9144526.

6

Ecuaciones Diferenciales Ordinarias

Las ecuaciones diferenciales ordinarias comprenden una rama muy extensa del análisis numérico y, tal vez, con más complejidades que cualquier otro tema que se haya estudiado.

Una ecuación diferencial es una ecuación en donde aparecen funciones, sus derivadas, una o más variables independientes y una o más variables dependientes. Las ecuaciones diferenciales se dividen en dos grupos: ecuaciones diferenciales ordinarias (EDO), en las cuales aparece sólo una variable independiente (que se denota con *x*); y ecuaciones diferenciales parciales (EDP) en las que aparecen más de una. El tema de discusión será el de las ecuaciones diferenciales ordinarias.

Las ecuaciones diferenciales ordinarias se clasifican y estudian según su *orden*, el cual se define como el entero igual al número máximo de veces que se deriva la variable dependiente en la ecuación.

Los problemas de ecuaciones diferenciales ordinarias se clasifican en problemas con condiciones iniciales y problemas con condiciones en la frontera. Los métodos numéricos para los problemas con condiciones iniciales difieren en forma significativa de los que se utilizan para los problemas con condiciones en la frontera.

6.1 Ecuaciones en diferencias

En vista de que el procedimiento numérico usual para manejar las EDO es considerar una ecuación en diferencias relacionadas, es conveniente estudiar un poco este tópico.

En general, se denotará con una letra mayúscula la variable dependiente desconocida de una ecuación en diferencias, en particular *Y*. Si bien *Y* será una función de *x*, el interés se centrará en los valores de *Y* correspondientes a los puntos aislados $[x_j]$; se denotará con Y_j tales valores $Y(x_j)$.

Se sobreentenderá que la longitud de paso en *x* es *h*(y es posible que se permita que *h* varíe dependiendo de *x*, esto es, de *j*); la diferencia hacia adelante de *Y(x)* es

$$\Delta Y(x) = Y(x+h) - Y(x)$$

o, de manera equivalente

$$\Delta Y_j = Y_{j+1} - Y_j$$

la traslación (hacia adelante) de *Y* es

$$EY(x) = Y(x+h) \qquad \text{o} \qquad EY_j = Y_{j+1}$$

Una ecuación en la que aparece una función desconocida *Y* evaluada en dos o más puntos $\{x_j\}$ se llama ecuación en diferencias y se puede también llamar ecuación de traslación.

Aunque existe una teoría y un desarrollo correspondientes para ecuaciones en diferencias en el estilo del tratamiento tradicional de las ecuaciones diferenciales, el método a usarse para la resolución de éstas será el de sustituciones sucesivas.

6.2 Método de Euler

Este método fue ideado por Euler hace más de 200 años. Es fácil de entender y de usar, pero no es tan preciso como otros métodos que serán estudiados posteriormente. Sin embargo, el método de Euler sirve como punto de partida hacia técnicas alternativas que aparecerán según se consideren perfeccionamientos razonables de éste.

El objetivo del método es obtener una aproximación al problema de valor inicial bien planteado

$$\frac{dy}{dx} = f(t,y), \qquad a \le t \le b, \qquad y(a) = a$$

No se obtendrá una aproximación continua de la solución *y(t)*, sino que se generarán aproximaciones de *y* en varios puntos, llamados **puntos de red**, en el intervalo [*a*,*b*]. Una vez que se obtenga la solución aproximada en estos puntos, la solución aproximada en otros puntos en el intervalo se puede encontrar utilizando algunos de los procedimientos de interpolación existentes.

Los puntos de red se suponen uniformemente espaciados sobre el intervalo [*a*,*b*]. Esta condición se puede garantizar escogiendo un entero positivo *N* y seleccionando los puntos de red $\{t_0, t_1, t_2, \ldots, t_N\}$ donde

$$t_i = a + ih \qquad \text{para cada } i = 0, 1, 2, \ldots, N$$

La distancia común entre los puntos, $h = (b-a)/N$ se llama **tamaño de paso**.

Es posible derivar el método de Euler, empleando el teorema de Taylor. Supóngase que *y(t)*,solución única de la ecuación diferencial ordinaria planteada anteriormente, tiene dos derivadas continuas en [*a*,*b*], de tal manera que para cada $i = 0, 1, 2, \ldots, N-1$, $y(t_{i+1})$ puede escribirse como

$$y(t_{i+1}) = y(t_i) + (t_{i+1} - t_i)y'(t_i) + \frac{(t_{i+1} - t_i)^2}{2} y''(x_i)$$

para algún número ξ_i, donde $t_i < \xi_i < t_{i+1}$

Usando la notación $h = t_{i+1} - t_i$, tenemos

$$y(t_{i+1}) = y(t_i) + hy'(t_i, y(t_i)) + \frac{h^2}{2} y''(x_i)$$

y puesto que *y(t)* satisface la ecuación diferencial

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2} y''(x_i)$$

el método de Euler construye $w_i \approx y(t_i)$ para cada $i = 1, 2, \ldots, N$, donde

$w_0 = a$

$w_{i+1} = w_i + hf(t_i, w_i)$

La ecuación anterior se llama la **ecuación de diferencia** asociada con el método de Euler. La forma algorítmica del método de Euler se presenta a continuación.

```
ENTRADA:     a, b: valores extremos;
             N: número de intervalos;
             a: condición inicial
SALIDA:      aproximación w de y en los (N+1) valores de t
Hacer h igual a (b - a)/N;
Hacer t igual a a;
Hacer w = a;
Mostrar (t, w);
Desde i = 1 hasta N Hacer
  Hacer w igual a w + hf(t, w); (Calcular w_i)
  Hacer t igual a a + ih; (Calcular t_i)
  Mostrar (t, w).
Parar.
```

6.3 Método de Euler Modificado

El método de Euler modificado tiene dos motivaciones. La primera es que es más preciso que el anterior. La segunda es que es más estable.

Este método se obtiene al aplicar la regla del trapecio para integrar $y' = f(y,t)$

$$y_{n+1} = y_n + \frac{h}{2}\left[f(y_{n+1}, t_{n+1}) + f(y_n, t_n) \right]$$

Si f es lineal en y, la ecuación anterior se puede resolver fácilmente en términos de y_{n+1}. Por ejemplo, si la EDO está dada por $y' = ay + cos(t)$, la ecuación queda

$$y_{n+1} = y_n + \frac{h}{2}\left[ay_{n+1} + \cos(t_{n+1}) + ay_n + \cos(t_n) \right]$$

por lo tanto al despejar y_{n+1} se obtiene

$$y_{n+1} = \frac{1 + ah/2}{1 - ah/2} y_n + \frac{h/2}{1 - ah/2}\left[\cos(t_{n+1}) + \cos(t_n) \right]$$

6.4 Métodos de Runge-Kutta

Una desventaja fundamental de los métodos de Euler consiste en que los órdenes de precisión son bajos. Esta desventaja tiene dos facetas. Para mantener una alta precisión se necesita una *h* pequeña, lo que aumenta el tiempo de cálculo y provoca errores de redondeo.

En los métodos de Runge-Kutta, el orden de precisión aumenta al utilizar puntos intermedios en cada intervalo. Una mayor precisión implica además que los errores decrecen más rápido al reducir *h*, en comparación con los métodos con precisión baja.

Considérese una ecuación diferencial ordinaria

$$y' = f(y,t), \quad y(0) = y_0$$

Para calcular y_{n+1} en $t_{n+1} = t_n + h$, dado una valor de y_n, integramos la ecuación anterior en el intervalo $[t_n, t_{n+1}]$:

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(y,t)dt$$

Los métodos de Runge-Kutta se obtienen al aplicar un método de integración numérica a la integral del lado derecho de la expresión anterior.

6.4.1 Método de Runge-Kutta de segundo orden

Al aplicar la regla del Trapecio a la integral planteada se obtiene

$$\int_{t_n}^{t_{n+1}} f(y,t)dt \approx \frac{1}{2}h\left[f(y_n,t_n) + f(y_{n+1},t_{n+1})\right]$$

En esta ecuación, y_{n+1} es una incógnita, por lo que aproximamos el segundo término mediante $f(\bar{y}_{n+1}, t_{n+1})$, donde $\bar{y}_{n+1}$ es la primera estimación de y_{n+1} obtenida mediante el método de Euler hacia adelante. Este esquema se conoce como el método de Runge-Kutta de segundo orden y se resume como

$$\bar{y}_{n+1} = y_n + hf(y_n,t_n)$$
$$y_{n+1} = y_n + \frac{h}{2}\left[f(y_n,t_n) + f(\bar{y}_{n+1},t_{n+1})\right]$$

expresado en forma canónica

$$k_1 = hf(y_n,t_n)$$
$$k_2 = hf(y_n + k_1, t_{n+1})$$
$$y_{n+1} = y_n + \frac{1}{2}\left[k_1 + k_2\right]$$

6.4.2 Método de Runge-Kutta de tercer orden

Un método de Runge-Kutta más preciso que el anterior es resultado de un esquema de integración numérica de orden superior para el segundo término de la ecuación planteada. Al aplicar la regla 1/3 de Simpson para resolver la integral se obtiene

$$y_{n+1} = y_n + \frac{h}{6}\left[f(y_n, t_n) + 4 f(\bar{y}_{n+1/2}, t_{n+1/2}) + f(\bar{y}_{n+1}, t_{n+1}) \right]$$

donde $\bar{y}_{n+1}$ y $\bar{y}_{n+1/2}$ son estimaciones, puesto que no conocemos $y_{n+1/2}$ y y_{n+1}. Obtenemos la estimación de $\bar{y}_{n+1/2}$ mediante el método de Euler hacia adelante:

$$\bar{y}_{n+1/2} = y_n + \frac{h}{2} f(y_n, t_n)$$

La estimación de $\bar{y}_{n+1}$ es

$$\bar{y}_{n+1} = y_n + hf(y_n, t_n)$$

bien

$$\bar{y}_{n+1} = y_n + hf(\bar{y}_{n+1/2}, t_{n+1/2})$$

El esquema global tiene la forma siguiente

$$\begin{aligned}
k_1 &= hf(y_n, t_n) \\
k_2 &= hf\left(y_n + \frac{1}{2}k_1, t_n + \frac{h}{2} \right) \\
k_3 &= hf\left(y_n - k_1 + 2k_2, t_n + h \right) \\
y_{n+1} &= y_n + \frac{1}{6}\left(k_1 + 4k_2 + k_3 \right)
\end{aligned}$$

6.4.3 Método de Runge-Kutta de cuarto orden

El método de Runge-Kutta de cuarto orden se obtiene de una manera análoga a la del tercer orden, excepto que se utiliza un paso intermedio adicional para evaluar la derivada. Es posible escoger varias formas para el esquema de integración numérica a utilizar. El método de Runge-Kutta de cuarto orden tiene un error local proporcional a h^5.

La siguiente versión de Runge-Kutta de cuarto orden está basada en la regla 1/3 de Simpson y se escribe como

$$k_1 = hf(y_n, t_n)$$
$$k_2 = hf\left(y_n + \frac{k_1}{2}, t_n + \frac{h}{2}\right)$$
$$k_3 = hf\left(y_n + \frac{k_2}{2}, t_n + \frac{h}{2}\right)$$
$$k_4 = hf\left(y_n + k_3, t_n + h\right)$$
$$y_{n+1} = y_n + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

La segunda versión está basada en la regla 3/8 de Simpson

$$k_1 = hf(y_n, t_n)$$
$$k_2 = hf\left(y_n + \frac{k_1}{3}, t_n + \frac{h}{3}\right)$$
$$k_3 = hf\left(y_n + \frac{k_1}{3} + \frac{k_2}{3}, t_n + \frac{2h}{3}\right)$$
$$k_4 = hf\left(y_n + k_1 - k_2 + k_3, t_n + h\right)$$
$$y_{n+1} = y_n + \frac{1}{8}\left(k_1 + 3k_2 + 3k_3 + k_4\right)$$

6.5 Métodos de Pasos Múltiples

Los métodos del tipo Runge-Kutta (que incluyen los métodos de Euler y Euler modificado como caso especial), se llaman métodos de un paso debido a que sólo utilizan la información del último paso calculado. Con esto ellos tiene la capacidad de realizar el siguiente paso con un tamaño de paso diferente, y son ideales para comenzar la solución en donde sólo se dispone de las condiciones iniciales. Sin embargo, después que la solución ha comenzado, se puede obtener información adicional disponible acerca de la función (y sus derivadas). Un método de pasos múltiples es aquél que toma ventaja de este hecho.

El principio que se encuentra detrás del método de pasos múltiples, es utilizar los valores anteriores de *y* y/o *y'* para construir un polinomio que se aproxime a la función derivada, y extrapolar este polinomio en el siguiente intervalo. La mayoría de los métodos utilizan valores equiespaciados anteriores, para hacer que la construcción de polinomios se fácil. El método de Adams es típico. El número de puntos utilizados por los que pasa, determina el grado del polinomio igual a la potencia de *h* en el término error global de la fórmula, el cual también es igual a uno más que el grado del polinomio.

Para deducir las relaciones del método de Adams, se escribe la ecuación diferencial *dy/dx = f(x,y)* en la forma

$$dy = f(x, y)dx,$$

y se integra entre x_n y x_{n+1}:

$$\int_{x_n}^{x_{n+1}} dy = y_{n+1} - y_n = \int_{x_n}^{x_{n+1}} f(x,y)dx.$$

Con el fin de integrar el término de la derecha de la igualdad, se aproxima *f(x,y)* como un polinomio en *x*, el cual se ajusta haciendo que pase por varios puntos. Si se ajusta a que pase por tres puntos, el polinomio aproximante será cuadrático. Si se ajusta a cuatro puntos, será cúbico. Mientras se ajuste a más puntos, mayor es la precisión.

Supóngase que se ajusta el polinomio de segundo grado para que pase por tres puntos, escribiendo esto como un polinomio de hacia atrás de Newton:

$$\int_{x_n}^{x_{n+1}} dy = y_{n+1} - y_n = \int_{x_n}^{x_{n+1}} \left(f_n + s\Delta f_{n-1} + \frac{(s+1)s}{2}\Delta^2 f_{n-2} + error \right) dx$$
$$= \int_{x=0}^{x=1} \left(f_n + s\Delta f_{n-1} + \frac{(s+1)s}{2}\Delta^2 f_{n-2} \right) hds + \int_{s=0}^{s=1} \frac{s(s-1)(s-2)}{6} h^3 f'''(\xi) hds.$$

En lo anterior, se ha cambiado la variable a *s* y se ha identificado a x_n como x_0. El intervalo de integración se hace desde *s = 0* hasta *s = 1*. Realizando la integración se obtiene

$$y_{n+1} = y_n + h\left[f_n + \frac{f_n - f_{n-1}}{2} + \frac{5\left(f_n - 2f_{n-1} + f_{n-2}\right)}{12} \right]$$
$$= y_n + \frac{h}{12}\left[23f_n - 16f_{n-1} + 5f_{n-2}\right] + O(h^4)$$

Obsérvese que la ecuación obtenida es similar a las fórmulas de un paso, en que el incremento de *y* es la suma ponderada de las derivadas por el tamaño del paso, pero difiere en que se utilizan valores anteriores, en lugar de estimados en la dirección de avance.

6.6 Métodos Predictor - Corrector

Un método predictor - corrector consta de un paso predictor y un paso corrector en cada intervalo. El predictor estima la solución para el nuevo punto y el corrector mejora su precisión. Los métodos de predictor - corrector utilizan la solución de los puntos anteriores, en lugar de utilizar puntos intermedios en cada intervalo.

Para explicar los métodos, considérese un intervalo de tiempo dividido de manera uniforme y supongamos que hemos calculado la solución hasta el tiempo *n*, por lo que es posible utilizar los valores de *y* y *y'* en los tiempos anteriores para calcular y_{n+1}.

Las fórmulas predictoras y correctoras se obtienen al sustituir una aproximación polinomial adecuada de *y'(t)* en la ecuación siguiente:

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(y,t)dt$$

El miembro más primitivo de los métodos predictor - corrector es el de segundo orden, que es idéntico al método de Runge-Kutta de segundo orden. Se obtiene un predictor de tercer orden al aproximar *y' = f(y,t)* con un polinomio de interpolación cuadrática, ajustado a f'_n, y'_{n-1} y y'_{n-2}:

$$y'(z) = \frac{1}{2h^2}\left[(z+h)(z+h)y'_n - 2z(z+2h)y'_{n-1} + z(z+h)y'_{n-2}\right] + E(z)$$

donde z es una coordenada local dada por $z = t - t_n$ y $E(z)$ es el error. La ecuación anterior viene dada por la interpolación de Lagrange ajustada a los valores y'_n, y'_{n-1} y y'_{n-2}. El error del polinomio viene dado por

$$E(z) = \frac{1}{3!}z(z+h)(z+2h)y^{(iv)}(\xi), \qquad t_{n-2} \le \xi \le t_{n+1}$$

En esta ecuación, la derivada del término del error es de cuarto orden, puesto que se ha ajustado un polinomio cuadrático a y'.

La ecuación inicial se puede reescribir en términos de la coordenada local $z = t - t_n$ como

$$y_{n+1} = y_n + \int_0^h y'(z)dz$$

Al sustituir el predictor de tercer orden obtenido en la ecuación anterior se obtiene

$$\bar{y}_{n+1} = y_n + \frac{h}{12}\left(23y'_n - 16y'_{n-1} + 5y'_{n-2}\right) + O(h^4)$$

donde la barra superior significa predictor. La ecuación obtenida recibe el nombre de *fórmula predictora de tercer orden de Adams - Bashforth*. El error de la ecuación se evalúa mediante la siguiente fórmula:

$$O(h^4) = \frac{3}{8}h^4 y^{(iv)}(\xi), \quad t_{n-2} \le \xi \le t_{n+1}$$

Para obtener la fórmula correctora, se necesita un valor predicho de y'_{n+1}, el cual se calcula sustituyendo este término en $y'(t) = f(y,t)$:

$$\bar{y}'_{n+1} = y_n + hf(\bar{y}'_{n+1}, t_{n+1})$$

El polinomio cuadrático ajustado a $\bar{y}'_{n+1}$, y'_n y y'_{n-1} se escribe como

$$y'(z) = \frac{1}{2h^2}\left[z(z+h)\bar{y}'_{n+1} - 2(z-h)(z+h)y'_n + z(z-h)y'_{n-1}\right] + E(z)$$

donde z es la coordenada local definida anteriormente. El error de esta ecuación es

$$E(z) = \frac{1}{3!}(z-h)z(z+h)y^{(iv)}(\xi), \qquad t_{n-1} \le \xi \le t_{n+1}$$

Se obtiene entonces la fórmula correctora

$$y_{n+1} = y_n + \frac{h}{12}\left(5\bar{y}'_{n+1} + 8y'_n - y'_{n-1}\right) + O(h^4)$$

y el error viene dado por

$$O(h^4) = -\frac{1}{24}h^4 y^{(iv)}(\xi), \qquad t_{n-1} \le \xi \le t_{n+1}$$

La ecuación obtenida es la *fórmula correctora de Adams - Moulton de tercer orden*. El conjunto de ecuaciones se llama *método predictor - corrector de Adamas de tercer orden*.

6.7 Sistemas de Ecuaciones y Ecuaciones de Mayor Orden

Hasta aquí sólo se ha considerado el caso de ecuaciones diferenciales de primer orden. La mayoría de las ecuaciones diferenciales que son modelos matemáticos de un problema físico son de mayor orden (orden superior), o aún un *conjunto* de ecuaciones diferenciales simultáneas de mayor orden. Por ejemplo,

$$\frac{w}{g}\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = f(x,t)$$

representa un sistema vibrante en el cual un resorte lineal con una constante de resorte *k*, restaura una masa desplazada de peso *w* en contra de una fuerza opuesta, cuya resistencia es *b* veces la velocidad. La función *f(x,t)* es una función de fuerza externa que actúa sobre la masa.

Una ecuación análoga de segundo orden describe el flujo de la electricidad en un circuito que contiene inductancia, capacitancia y resistencia. En este caso la función de fuerza externa representa la fuerza electromotriz aplicada. Los sistemas compuestos de masa - resorte y los circuitos eléctricos se pueden simular por medio de un sistema de tales ecuaciones de segundo orden.

Ahora se mostrará cómo una ecuación diferencial de orden superior, se reduce a un sistema de ecuaciones simultáneas de primer orden. Y luego se mostrará que éstas pueden ser resueltas por aplicación de los métodos que se estudiaron anteriormente.

Resolviendo para la segunda derivada, de ordinario se puede expresar una ecuación d segundo orden como:

$$\frac{d^2x}{dt^2} = f\left(x, t, \frac{dx}{dt}\right), \qquad x(t_0) = x_0, \qquad x'(t_0) = x'_0$$

En general se especifica el valor inicial de la función *x* y de su derivada. Este se transforma a un par de ecuaciones de primer orden por el sencillo procedimiento de definir las derivadas como una segunda función. Entonces, ya que $d^2x/dt^2 = (d/dt)(dx/dt)$,

$$\frac{dx}{dt} = y, \qquad x(t_0) = x_0,$$
$$\frac{dy}{dt} = f(x,t,y), \qquad y(t_0) = x_0'$$

Este par de ecuaciones de primer orden es equivalente a la ecuación original. Para ecuaciones de mayor orden, cada una de las derivadas de orden inferior se define como una nueva función, dando un conjunto de *n* ecuaciones de primer orden, que corresponden a una ecuación diferencial de *n*-ésimo orden. Para un sistema de ecuaciones de mayor orden, cada una se transforma de manera semejante, resultando un conjunto grande de ecuaciones de primer orden.

Problemas

1. Use el método de Euler para aproximar la solución de cada uno de los siguientes problemas de valor inicial:

   a) $y' = \left(\frac{y}{t}\right)^2 + \left(\frac{y}{t}\right), \qquad 1 \le t \le 1{,}2, \qquad y(1) = 1 \text{ con } h = 0{,}1$

   b) $y' = \operatorname{sen} t + e^{-t}, \qquad 0 \le t \le 1, \qquad y(0) = 0 \text{ con } h = 0{,}5$

   c) $y' = ty, \qquad 0 \le t \le 2, \qquad y(0) = 1 \text{ con } N = 4$

2. Dado el problema de valor inicial

   $$y' = \frac{2}{t}y + t^2 e^t, \qquad 1 \le t \le 2, \qquad y(1) = 0$$

   con solución exacta

   $$y(t) = t^2(e^t - e)$$

   Use el método de Euler con *h = 0,05* para aproximar la solución y compararla con los valores reales de *y*.

3. Aplique el método de Euler modificado con *h = 0,5* al problema

   $y' = x^2 - y^3, \quad y(1) = 1;$ determine $y(2{,}5)$

4. Aplique el método de Euler modificado con *h = 0,25* al problema

   $y' = x^3 - y^2, \quad y(0) = 0;$ determine $y(0{,}5)$

5. Resuelva la siguiente ecuación de segundo orden, utilizando Runge-Kutta de segundo orden

   $$y''(t) + 5y'(t) - 10y(t) = t, \qquad y(0) = 1, \qquad y'(0) = 0$$

6. Resuelva los siguientes problemas en *0 £ t £ 5* mediante los métodos de Euler, Runge-Kutta y Adams-Moulton con *h=0,1* y *h=0,01*:

a) $y'' + 8y = 0$, $y(0)=1,\ y'(0)= 0$

b) $y'' - 0{,}01(y')^2 + 2y = sen(t)$, $y(0)= 0, y'(0)= 1$

c) $y'' + 2ty' + ty = 0$, $y(0)= 1, y'(0)= 0$

d) $(e^t + y)y'' = t$, $y(0)= 1, y'(0)= 0$

7

Técnicas Iterativas en el Álgebra Matricial

7.1 Normas Matriciales y Vectoriales

Cuando se estudian entidades de componentes múltiples, como las matrices y los vectores, con frecuencia se necesita una forma de expresar su magnitud - alguna medida de su "grandeza" o "pequeñez" -. Para los números ordinarios, el valor absoluto indica qué tan grande es el número; pero para una matriz hay muchos componentes, cada uno de los cuales puede ser grande o pequeño en magnitud.

Cualquier buena medida de la magnitud de una matriz (*norma*), debe tener cuatro propiedades que son intuitivamente esenciales:

1. Las norma siempre tiene un valor mayor o igual a cero, y sólo es cero cuando la matriz es cero (con todos sus elementos iguales a cero).
2. La norma estará multiplicada por *k*, si la matriz está multiplicada por el escalar *k*.
3. La norma de la suma de dos matrices, no excederá a la suma de las normas.
4. La norma del producto de dos matrices, no excederá al producto de las normas.

De manera más formal, se pueden establecer estas condiciones, utilizando $||A||$ para representar la *norma de la matriz A*.:

1. $||A|| \geq 0$ y $||A|| = 0$ si y sólo si $A = 0$
2. $||kA|| = k||A||$
3. $||A + B|| \leq ||A|| + ||B||$
4. $||AB|| \leq ||A|| \, ||B||$

La tercera relación se llama *desigualdad del triángulo*. La cuarta es importante cuando se trata con el producto de dos matrices.

Para una clase especial de matrices, que se llama *vectores*, los conceptos expresados pueden ayudar. Para los vectores en un espacio bi o tridimensional, la longitud satisface los cuatro requerimientos, y es un buen valor para utilizar para la norma del vector. Esta norma se llama *norma euclidiana*, y se calcula con

$$\sqrt{x_1^2 + x_2^2 + x_3^2}$$

La norma euclidiana para vectores con más de tres componentes, se calcula generalizando:

$$\|x\|_e = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2} = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2}$$

Sin embargo, ésta no es la única forma de calcular una norma vectorial. La suma de los valores absolutos de las x_i se puede utilizar como una norma; el valor máximo de las magnitudes de las x_i también servirá. Estas tres normas se pueden interrelacionar definiendo la norma *p* como

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

Dependerá del problema, cuál de estas normas vectoriales es la mejor para usar.

Ejemplo. Calcúlense las normas 1, 2 e ∞ del vector *x*, si *x = (1,25, 0,02, -5,15, 0).*

$$\|x\|_1 = |1{,}25| + |0{,}02| + |-5{,}15| + |0| = 6{,}42;$$
$$\|x\|_2 = \left[(1{,}25)^2 + (0{,}02)^2 + (-5{,}15)^2 + (0)^2 \right]^{1/2} = 5{,}2996;$$
$$\|x\|_\infty = |-5{,}15| = 5{,}15$$

Las normas de una matriz se desarrollan por una correspondencia con las normas vectoriales. Las normas matriciales que correspondan a las anteriores, de la matriz *A*, se puede demostrar que son:

$$\|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}| = \quad \text{Suma columna máxima;}$$

$$\|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| = \quad \text{Suma fila máxima}$$

La norma de la matriz $\|A\|_2$ que corresponde a la norma 2 de un vector, no se calcula con rapidez. Está relacionada con los valores característicos de la matriz. Algunas veces tiene utilidad especial debido a que ninguna otra norma es más pequeña que esta norma. Por tanto, proporciona la medida más "cerrada" del "tamaño" de una matriz, pero es también la más difícil de calcular. Esta norma también se llama *norma espectral*.

Para una matriz de *m ´ n*, se puede parafrasear a la norma euclidiana y escribir como

$$\|A\|_e = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 \right)^{1/2}$$

Tómese en cuenta que si bien para un vector la norma 2 es igual que la norma euclidiana, no lo es así para una matriz.

Ejemplo. Calcúlense las normas euclidianas de *A*, *B*, y *C*; las matrices vienen dadas por

$$A = \begin{bmatrix} 5 & 9 \\ -2 & 1 \end{bmatrix};\ B = \begin{bmatrix} 0{,}1 & 0 \\ 0{,}2 & -0{,}1 \end{bmatrix};\qquad \text{y} \qquad C = \begin{bmatrix} 0{,}2 & 0{,}1 \\ 0{,}1 & 0 \end{bmatrix}.$$

$$\|A\|_e = \sqrt{25+81+4+1} = \sqrt{111} = 10{,}53; \qquad \|A\|_\infty = 14.$$
$$\|B\|_e = \sqrt{0{,}01+0+0{,}04+0{,}01} = \sqrt{0{,}06} = 0{,}2449 \qquad \|B\|_\infty = 0{,}3.$$
$$\|C\|_e = \sqrt{0{,}04+0{,}01+0{,}01+0} = \sqrt{0{,}06} = 0{,}2449; \qquad \|C\|_\infty = 0{,}3.$$

Los resultados del ejemplo se ven muy razonables; ciertamente *A* es "*mayor*" que *B* o *C*. Mientras que *B* ¹ *C*, ambas son igualmente *"pequeñas"*. La norma euclidiana es una buena medida de la magnitud de una matriz.

La importancia de las normas radica en que permiten expresar la precisión de la solución de un conjunto de ecuaciones en términos cuantitativos, al establecer la norma del vector error (la verdadera solución menos el vector de la solución aproximada). Las normas también se utilizan para resolver sistemas lineales.

7.2 Solución de Sistemas Lineales por Iteración

Además de los métodos directos, estudiados en los cursos de Álgebra Lineal, existen los métodos iterativos para solución de sistemas de ecuaciones lineales. En ciertos casos, se prefieren estos métodos en vez de los directos, por ejemplo cuando la matriz de los coeficientes es poco densa (tiene muchos ceros), ya que en estos casos pueden ser más rápidos. Además podrían requerir menor cantidad de memoria y en los cálculos manuales tienen la ventaja distintiva de ser autocorrectores si se comete un error; algunas veces se pueden utilizar para reducir el error por redondeo, en las soluciones calculadas por métodos directos. También es posible aplicarlos a sistemas de ecuaciones no lineales.

7.2.1 Estimaciones sucesivas de la solución (método de Jacobi).

Se explica el método con un ejemplo sencillo:

$$\begin{aligned} 8x_1 + x_2 - x_3 &= 8, \\ 2x_1 + x_2 + 9x_3 &= 12, \\ x_1 - 7x_2 + 2x_3 &= -4, \end{aligned}$$

la solución es $x_1 = 1$, $x_2 = 1$, $x_3 = 1$. Se comienza el esquema iterativo resolviendo cada ecuación para una de las variables, escogiendo cuando sea posible, para resolver la variable con el coeficiente más grande:

$$\begin{aligned} x_1 &= 1 - 0{,}125x_2 + 0{,}125x_3 && \text{(de la primera ecuacion)} \\ x_2 &= 0{,}571 + 0{,}143x_1 + 0{,}286x_3 && \text{(de la tercera ecuacion)} \\ x_3 &= 1{,}333 + 0{,}222x_1 - 0{,}111x_2 && \text{(de la segunda ecuacion)} \end{aligned}$$

Se comienza con alguna aproximación inicial al valor de las variables. Sustituyendo estas aproximaciones en los segundos miembros del conjunto de ecuaciones, genera nuevas aproximaciones más cercanas al valor verdadero. Los nuevos valores se sustituyen en los segundos miembros para generar una segunda aproximación, y el proceso se repite hasta que sean lo suficientemente similares los valores sucesivos de cada una de las variables. Para el conjunto de ecuaciones dado se obtiene:

	Primera	Segunda	Tercera	Cuarta	Quinta	Sexta	Séptima	Octava
x_1	0	1,000	1,095	0,995	0,993	1,002	1,001	1,000
x_2	0	0,571	1,095	1,026	0,990	0,998	1,001	1,000
x_3	0	1,333	1,048	0,969	1,000	1,004	1,001	1,000

Este procedimiento es conocido como el *método de Jacobi*, también llamado "método de los desplazamientos simultáneos", debido a que se cambia cada ecuación simultáneamente utilizando el conjunto más reciente de valores de *x*.

Algoritmo para la iteración Jacobi

Considérese que cada ecuación se puede expresar de la forma

$$x^{(n+1)} = Gx^{(n)} = b' - Bx^{(n)},$$

donde $x^{(n+1)}$ y $x^{(n)}$ se refieren a las iteraciones *n*-ésima y *(n+1)*-ésima de un vector, *G* es una transformación lineal, *b'* es la matriz columna de los términos independientes y *B* es la matriz de los coeficientes de las variables.

Para resolver un sistema de *N* ecuaciones lineales, es necesario reordenar las filas de manera que, los elementos diagonales tengan magnitudes tan grandes como sea posible, en relación a las magnitudes de otros coeficientes en la misma fila. Defínase al sistema reordenado como $Ax = b$. Comenzando con una aproximación inicial al vector $x^{(1)}$, calcule cada componente $x^{(n+1)}$ para $i= 1,2,..., N$, por:

$$x_i^{(n+1)} = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{a_{ij}}{a_{ii}} x_j^{(n)} \qquad , \qquad n= 1,2,...$$

Una condición suficiente para la convergencia es que

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{N} |a_{ij}|, \quad i= 1, 2, ..., N.$$

Cuando esto es verdad, $x^{(n)}$ convergerá a la solución, sin importar cuál vector inicial sea el que se use.

En el caso particular estudiado, la solución obtenida es exacta, más no existe garantía alguna de que esto sea así, debido a la naturaleza del sistema de ecuaciones y al error por redondeo. Lo común es entonces que el algoritmo reciba como entrada un cierto valor de tolerancia *TOL* con el cual se pueda establecer un criterio de paro como el que se muestra:

$$\frac{\left\| x^{(k)} - x^{(k+1)} \right\|}{\left\| x^{(k)} \right\|} < TOL$$

donde *TOL* es una valor mayor que cero y $x^{(k)}$ es el vector de soluciones en la k-ésima iteración. Para este propósito, se puede usar cualquier norma conveniente, siendo la más comúnmente usada la norma $l_{¥}$. Este criterio de paro puede ser usado para cualquiera de los métodos estudiados en esta sección.

7.2.2 Estimaciones sucesivas de la solución (método de Gauss-Seidel)

Cuando se ejecuta el método de Jacobi, todos los valores de *x* no se calculan "simultáneamente". La segunda estimación de x_1 se calculó antes de obtener la de x_2, y se tenían nuevos valores para x_1 y x_2 antes de que se mejorara el valor de x_3. En casi todos los casos los nuevos valores son mejores que los anteriores, y se le debe emplear en preferencia a los valores peores. Cuando se hace esto, el método se conoce por el nombre de "*Gauss-Seidel"*. En este método el primer paso consiste en reordenar el conjunto de ecuaciones, resolviendo cada ecuación para una de las variables en términos de las otras, exactamente como se hizo en el método de Jacobi. Luego se procede a mejorar cada valor *x* a su vez, utilizando siempre las aproximaciones más recientes de los valores de las otras variables. La rapidez de convergencia es mayor, tal como se muestra volviendo a calcular el mismo ejemplo anterior.

	Primera	Segunda	Tercera	Cuarta	Quinta	Sexta	Séptima	Octava
x_1	0	1,000	1,041	0,997	1,001	1,000		
x_2	0	0,714	1,014	0,996	1,000	1,000		
x_3	0	1,032	0,990	1,002	1,000	1,000		

Algoritmo para la iteración de Gauss-Seidel

Para resolver un sistema de *N* ecuaciones lineales, debe reordenarse con las filas de manera que los elementos de la diagonal tengan magnitudes tan grandes como sea posible, en relación a las magnitudes de los otros coeficientes de la misma fila. Defínase al sistema reordenado como $Ax = b$. Comenzando con una aproximación inicial al vector solución $x^{(1)}$, calcúlese cada componente de $x^{(n+1)}$, para $i= 1, 2, ..., N$, por

$$x_i^{(n+1)} = \frac{b_i}{a_{ii}} - \sum_{j=1}^{i-1} x_j^{(n+1)} - \sum_{j=i+1}^{N} \frac{a_{ij}}{a_{ii}} x_j^{(n)}, \quad n= 1, 2, ...$$

Una condición suficiente para la convergencia es que

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{N} |a_{ij}|, \quad i= 1, 2, ..., N$$

Cuando esto es verdadero, $x^{(n)}$ convergerá a la solución sin importar cuáles sean los vectores iniciales que se utilicen.

Es posible demostrar que el método de Gauss-Seidel siempre convergerá si lo hace el método de Jacobi, y lo hará más rápidamente. Si el método de Jacobi diverge, lo mismo sucederá con el método de Gauss-Seidel. El método de Gauss-Seidel siempre debe ser usado debido a su convergencia más rápida.

Estos métodos iterativos no convergerán para todos los conjuntos de ecuaciones, ni para todas las reordenaciones posibles de las ecuaciones. Cuando se puedan ordenar las ecuaciones de manera que, cada elementos en la diagonal es mayor en magnitud, que la suma de las magnitudes de los otros coeficientes en esa fila (sistema *diagonalmente dominante*), la iteración convergerá para cualesquiera valores iniciales. Esto es fácil de visualizar, debido a que todas las ecuaciones se pueden reexpresar en la forma:

$$x_i = \frac{b_i}{a_{ii}} - \frac{a_1}{a_{ii}} x_1 - \frac{a_2}{a_{ii}} x_2 - ...$$

El error en el siguiente valor de x_i, será la suma de los errores en todas las otras *x*, multiplicadas por los coeficientes de la ecuación, y si la suma de las magnitudes de los coeficientes es menor que la unidad, el error disminuirá

conforme proceda la iteración. La condición de convergencia anterior es sólo una condición suficiente; es decir, si se da la condición, el sistema siempre converge, pero algunas veces lo hace aún si la condición no se cumple.

La rapidez con la cual converjan las iteraciones, está relacionada con el grado de dominación de los términos de la diagonal.

7.2.3 Método de Relajación

Existe un método iterativo que converge más rápidamente que el método de Gauss-Seidel, y que se utiliza con ventaja en los cálculo manuales. Desafortunadamente no está bien adaptado a su aplicación en la computadora. El método se debe al ingeniero británico Richard Southwell, y ha sido aplicado a una amplia variedad de problemas. La importancia de estudiar el método está en que conduce a una técnica de aceleración importante llamada *sobrerrelajación*.

El *método de relajación* es un esquema que permite seleccionar la mejor ecuación, a ser usada para una tasa máxima de convergencia.

El método se muestra con el mismo ejemplo utilizado en las secciones anteriores.

$$\begin{aligned} 8x_1 + x_2 - x_3 &= 8, \\ 2x_1 + x_2 + 9x_3 &= 12, \\ x_1 - 7x_2 + 2x_3 &= -4, \end{aligned}$$

De nuevo se comienza reordenando las ecuaciones, pero en forma diferente de los métodos de Gauss-Seidel o Jacobi. Todos los términos se transponen a un sólo miembro y luego se dividen por el negativo de coeficiente más grande. Las ecuaciones se convierten en:

$$\begin{aligned} -x_1 - 0{,}125x_2 + 0{,}125x_3 + 1 &= 0 \\ -0{,}222x_1 - 0{,}111x_2 - x_3 + 1{,}333 &= 0 \\ 0{,}143x_1 + x_2 + 0{,}286x_3 + 0{,}571 &= 0 \end{aligned}$$

Si se comienza con algún conjunto inicial de valores y se lo sustituye en las ecuaciones, éstas no estarán satisfechas a menos, que por suerte, hayan caído en la solución; los primeros miembros no serán cero, sino que tendrán algún otro valor al que se llamará *residuo*, y se denotará por *Ir*. También es conveniente reordenar la ecuación de manera que los coeficientes -1 estén en la diagonal. El sistema queda entonces:

$$\begin{aligned} -x_1 - 0{,}125x_2 + 0{,}125x_3 + 1 &= R_1 \\ 0{,}143x_1 - x_2 + 0{,}286x_3 + 0{,}571 &= R_2 \\ -0{,}222x_1 - 0{,}111x_2 - x_3 + 1{,}333 &= R_3 \end{aligned}$$

Por ejemplo, con $x_1 = 0$, $x_2 = 0$, $x_3 = 0$, se tiene

$$R_1 = 1,\ R_2 = 0{,}571, \qquad R_3 = 1{,}333$$

El residuo mayor en magnitud R_3, nos dice que la tercera ecuación tiene el mayor error, y primero debe ser mejorada. El método obtiene el nombre de “relajación”, del hecho de que se hace un cambio x_3 para relajar a R_3 (el residuo más grande), de manera que se le pueda hacer cero. Observando los coeficientes de las diversas ecuaciones, se ve que

incrementando el valor de x_3 por uno, se decrementará R_3 por uno, lo que incrementará R_1 por 0,125 e incrementará a R_2 por 0,286. Para cambiar R_3 de su valor inicial de 1,333 a cero, se incrementa x_3 por la misma cantidad.

Entonces se selecciona el nuevo residuo de magnitud mayor, y se relaja a cero. Se continúa hasta que todos los residuos son cero y cuando esto es verdad, los valores de las *x* serán la solución exacta.

De ordinario, el método no se programa debido a que la búsqueda en la computadora para el primer residuo mayor es lenta, y añade el suficiente tiempo de ejecución, de manera que la aceleración no de un beneficio neto. Sin embargo, la búsqueda puede ser hecha con rapidez, escudriñando los residuos por medio del cálculo manual.

Southwell y sus colaboradores observaron que, en muchos casos, la relajación de los residuos a cero era menos eficiente que la relajación más allá de cero (*sobrerrelajación*) o que la relajación corta a cero (*subrelajación*). La razón de esto en una estrategia mejorada, es que un residuo cero no permanece cero; relajando el residuo de otra ecuación afecta al primer residuo de manera que, resulta apropiado anticiparse y permitir que esto suceda por la sobrerrelajación o subrelajación apropiada.

Hay aspecto del método de relajación de Southwell, que tiene influencia sobre la solución iterativa de las ecuaciones lineales, por medio de una computadora digital. Al utilizar el método de Gauss-Seidel, se puede acelerar la convergencia por "sobrerrelajación", esto es haciendo que los residuos pasen a otro lado del cero, en lugar de sólo relajar a cero. Se puede aplicar esto a la iteración de Gauss-Seidel modificando el algoritmo.

La relación estándar para la iteración de Gauss-Seidel para el conjunto de ecuaciones $Ax = b$, para x_i variable, se escribe como:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{N} a_{ij} x_j^{(k)} \right)$$

en donde los subíndices *(k+1)* indican que ésta es la *(k+1)*-ésima iteración. En el lado derecho de la igualdad se utilizan las estimaciones más recientes de x_j, los cuales serán ya sea $x_l^{(k)}$ o $x_j^{(k+1)}$.

Una forma algebraicamente equivalente a la ecuación anterior es:

$$x_i^{(k+1)} = x_i^{(k)} + \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^{N} a_{ij} x_j^{(k)} \right)$$

debido a que $x_i^{(k)}$ es a la vez sumada y restada del segundo miembro. En esta forma, se ve que las relajaciones de Gauss-Seidel y Southwell pueden tener una aritmética idéntica: el término que se suma a $x_i^{(k)}$ para obtener $x_i^{(k+1)}$ es exactamente el incremento que relaja al residuo de cero. La sobrerrelajación puede ser aplicada al algoritmo de Gauss-Seidel, si se añade a $x_i^{(k)}$ algún múltiplo del segundo término. Se puede mostrar que este múltiplo nunca debe ser más de 2 en magnitud (para evitar divergencia), y que el factor óptimo de sobrerrelajación yace entre 1,0 y 2,0. La ecuación de iteración tomará esta forma, donde *w* es el *factor de sobrerrelajación*.

$$x_i^{(k+1)} = x_i^{(k)} + \frac{w}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^{N} a_{ij} x_j^{(k)} \right)$$

El valor óptimo para *w* variará entre 1,0 y 2,0, dependiendo del tamaño de la matriz de coeficientes y de los valores de los coeficientes.

Problemas

1. Encontrar $\|A\|_\infty$ y $\|A\|_2$ para los siguientes vectores

   a) $x = (3, -4, 3/2)$.
   b) $x = (2,1,-3,4)$.
   c) $x = (sen(k), cos(k), 2^k)$ para un entero positivo fijo k.

2. Calcule $\|A\|_e$, $\|A\|_\infty$ y $\|A\|_1$ para la siguiente matriz.

$$A = \begin{bmatrix} 5 & -4 & -7 \\ -4 & 2 & -4 \\ -7 & -4 & 5 \end{bmatrix}$$

3. Resuelva los siguientes sistemas lineales por el método de Jacobi, usando $x^{(0)} = 0$ con $TOL = 10^{-2}, 10^{-3}$ y 10^{-4}.

a)
$$\begin{aligned} 2x_1 - x_2 + x_3 &= -1 \\ 3x_1 + 3x_2 + 9x_3 &= 0 \\ 3x_1 + 3x_2 + 5x_3 &= 4 \end{aligned}$$

b)
$$\begin{aligned} 2x_2 + 4x_3 &= 0 \\ x_1 - x_2 - x_3 &= 0{,}375 \\ x_1 - x_2 + 2x_3 &= 0 \end{aligned}$$

c)
$$\begin{aligned} 2x_1 - x_2 + 10x_3 &= -11 \\ 3x_2 - x_3 + 8x_4 &= -11 \\ 10x_1 - x_2 + 2x_3 &= 6 \\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25 \end{aligned}$$

d)
$$\begin{aligned} 4x_1 - 2x_2 &= 0 \\ -2x_1 + 5x_2 - x_3 &= 2 \\ -x_2 + 4x_3 + 2x_4 &= 3 \\ 2x_3 + 3x_4 &= -2 \end{aligned}$$

4. Repetir el ejercicio 3 usando Gauss-Seidel.

5. Repetir el ejercicio 3 usando Sobrerrelajación aplicado al método de Gauss-Seidel con $w = 1{,}2$.