COMPUTACIÓN NUMÉRICA

Version 2015/16

Gonzalo Galiano Casas [1] ***Esperanza García Gonzalo*** [2]

Departamento de Matemáticas. Universidad de Oviedo

[1]E-mail: galiano@orion.ciencias.uniovi.es
[2]E-mail: espe@uniovi.es

Índice general

1. Aritmética finita y análisis de error **5**

1. Introducción 5
   1.1. La norma IEEE 754 5
   1.2. Representación decimal y binaria 6
   1.3. Paso del sistema decimal a binario y viceversa 7
2. Representación de enteros 8
3. Representación binaria en punto flotante según la norma IEEE 754 9
   3.1. Precisión simple 9
   3.2. Precisión doble 11
   3.3. Valores especiales 12
   3.4. Exactitud 14
   3.5. Redondeo 16
4. Error 19

2. Ecuaciones no lineales **21**

1. Introducción 21
   1.1. Orden de convergencia y criterio de parada 21
2. El método de bisección 22
3. El método de Newton 24
4. El método del punto fijo 25
5. El método de la secante 27

3. Interpolación y aproximación **29**

1. Interpolación 29
2. Interpolación polinómica: el polinomio de Lagrange 30
   2.1. Polinomios fundamentales de Lagrange 31
   2.2. Diferencias divididas 32
   2.3. Estimación del error 35

3. Interpolación por polinomios a trozos 35
3.1. Interpolación por splines 36
3.2. Estimación del error 38
4. Interpolación mediante polinomios trigonométricos 39
5. Aproximación 41
6. Método de mínimos cuadrados 42
7. Bases ortogonales 43
7.1. Aproximación mediante polinomios de Legendre 44
7.2. Aproximación mediante series de Fourier 46

4. Diferenciación e integración numéricas **49**
1. Derivación numérica 49
1.1. Derivadas de orden superior 51
1.2. Derivación numérica de funciones de varias variables 51
2. Integración numérica 53
2.1. Fórmula del punto medio 53
2.2. Fórmula del trapecio 54
2.3. Fórmula de Simpson 54
2.4. Fórmulas de orden superior 55
2.5. Fórmula de Gauss 55

5. Sistemas de ecuaciones lineales **57**
1. Introducción 57
2. Métodos directos 58
2.1. Gauss 58
2.2. Gauss-Jordan 61
2.3. Factorización LU 64
3. Métodos iterativos 66
3.1. Método de Jacobi 67
3.2. Gauss-Seidel 69
3.3. Convergencia de los métodos iterativos 71

6. Optimización **75**
1. Definición del problema de optimización 75
1.1. Ejemplos 77
2. Optimización sin restricciones en dimensión finita 78

2.1. Conceptos básicos y notación 78
2.2. Condiciones necesarias y suficientes de optimalidad local 79
2.3. Método de Newton 82
2.4. Método del gradiente 83
3. Optimización con restricciones en dimensión finita 84
3.1. Multiplicadores de Lagrange. Restricciones de igualdad 85
3.2. Método de penalización 86

Apéndice. Algunas definiciones importantes **90**

Bibliografía **94**

Capítulo 1

Aritmética finita y análisis de error

1. Introducción

Los números reales, que pueden tener infinitos dígitos, se almacenan en el ordenador utilizando solo un número finito de dígitos.

Los números se almacenan en los ordenadores con los formatos:

- **Entero** que permite el almacenamiento exacto de un conjunto de números enteros.
- **En punto flotante** que permite el almacenamiento exacto de un conjunto de números enteros y un conjunto de números racionales.

1.1. La norma IEEE 754

El formato en punto flotante usado habitualmente es el formato IEEE 754. IEEE significa Institute of Electrical and Electronics Engineers. Este estándar es el usado actualmente por casi todos los procesadores.

La primera norma IEEE 754 se publicó en el 1985 e incluía únicamente la representación de números en binario. Sus formatos básicos eran los de simple y el doble precisión. En el 2008 se publicó una segunda versión donde se incluía también la representación de números en decimal con dos formatos básicos y se añadía un formato básico binario con precisión cuádruple. El formato decimal es interesante desde el punto de vista de cálculos financieros y bancarios, que por ley, utilizan redondeos en base 10.

Los cinco formatos básicos y sus parámetros más importantes aparecen en la tabla 1.1. Además de los formatos básicos existen los formatos de precisión extendida y de precisión extensible. Los formatos de precisión extendida alargan la precisión de los formatos básicos proporcionando una mayor precisión y rango. En los formatos extensibles el usuario define la precisión y el rango.

Las FPUs (Floating Point Units) o coprocesadores matemáticos eran, antes de establecerse el estándar IEEE 754, circuitos integrados opcionales que se añadían a la placa base junto con el procesador principal, y que se encargaban de las operaciones en punto flotante que, previamente a la norma IEEE 754, eran propias de cada sistema operativo y compilador. Después del establecimiento de la norma IEEE 754 la existencia de este coprocesador matemático dejó de ser opcional

	Formatos binarios ($b=2$)			Formatos decimales ($b=10$)	
parámetro	binary32	binary64	binary128	decimal64	decimal128
precisión (p)	24	53	113	16	34
e_{max}	+127	+1023	+16383	+384	+6144

Cuadro 1.1: Principales parámetros de los formatos básicos de la norma IEEE 754

y se convirtió en estándar. Estos procesadores realizan tanto operaciones aritméticas básicas, como la suma, como operaciones más avanzadas como la raíz cuadrada y funciones trigonométricas, logaritmos, etc.

La mayoría de los procesadores actuales implementan en hardware el estándar de 1985 pero no el del 2008. Sólo existen unos pocos procesadores que implementen los formatos básicos nuevos del estándar 2008 en el hardware, es decir, los formatos decimales y el formato binario de cuádruple precisión. Excepto algunos procesadores, principalmente de IBM, en general, los formatos básicos nuevos del 2008 se ha implementado en software pero no en hardware.

Además de definir el almacenamiento de los números en punto flotante y su redondeo, la norma IEEE 754 incluye las principales operaciones aritméticas, conversiones de números entre los diferentes formatos y el manejo de las excepciones (situaciones dudosas que se pueden manejar de distintas maneras como dando un aviso y siguiendo con la ejecución del programa o simplemente abortando la ejecución).

La norma IEEE 754 no especifica la representación de enteros, aunque, lógicamente, sí la de los exponentes enteros que aparecen en la representación de números en punto flotante.

1.2. Representación decimal y binaria

La representación normalizada en punto flotante de un número $x \neq 0$ en base 10 es

$$x = \sigma \times \bar{x} \times 10^e,$$

donde $\sigma = \pm 1$, es el signo, $1 \leq \bar{x} < 10$ es la mantisa y $e \in \mathbb{Z}$ es el exponente.

Definición 1 *El número de dígitos en $\bar{x}$ es la* precisión de la representación.

Ejemplo 1.1 La representación normalizada en punto flotante *decimal* del número x es

$$x = 314{,}15 = 3{,}1415 \times 10^2$$

y entonces

$$\sigma = +1, \quad \bar{x} = 3{,}1415, \quad e = 2$$

está representado con 5 dígitos de precisión.

□

La representación normalizada en punto flotante de un número $x \neq 0$ en base 2 es

$$x = \sigma \times \bar{x} \times 2^e,$$

donde $\sigma = \pm 1$ es el signo, $(1)_2 \leq \bar{x} < (10)_2$ la mantisa y $e \in \mathbb{Z}$ es el exponente.

Ejemplo 1.2 El número binario $x = (10101{,}11001)_2 = (1{,}010111001)_2 \times 2^4$ tiene 10 dígitos de precisión. □

Ejemplo 1.3 Para $x = (101{,}001101)_2 = (5{,}203125)_{10}$ la representación decimal normalizada en punto flotante es:

$$\sigma = +1, \quad \bar{x} = 5{,}203125, \quad e = 0,$$

y la representación binaria normalizada en punto flotante:

$$\sigma = (1)_2, \quad \bar{x} = (1{,}01001101)_2, \quad e = (2)_{10} = (10)_2.$$

□

1.3. Paso del sistema decimal a binario y viceversa

En el sistema decimal el número 107,625 significa:

$$107{,}625 = 1 \cdot 10^2 + 7 \cdot 10^0 + 6 \cdot 10^{-1} + 2 \cdot 10^{-2} + 5 \cdot 10^{-3}.$$

En general, los ordenadores usan el sistema binario: sólo se almacenan 0 y 1. En el sistema binario, los números representan potencias de 2:

$$(107{,}625)_{10} = 2^6 + 2^5 + 2^3 + 2^1 + 2^0 + 2^{-1} + 2^{-3} = (1101011{,}101)_2$$

El paso de binario a decimal es directo:

$$(1101011{,}101)_2 = 2^6 + 2^5 + 2^3 + 2^1 + 2^0 + 2^{-1} + 2^{-3} = (107{,}625)_{10}.$$

El paso de decimal a binario lo realizamos convirtiendo primero la parte entera. Dividimos sucesivamente por 2 y los restos son los dígitos en base 2. Tomamos primero el último cociente (1) y luego los restos de derecha a izquierda

Cocientes	107	53	26	13	6	3	$\boxed{1}$
Restos	1	1	0	1	0	1	
						$\longleftarrow$	

Y luego convertimos la parte fraccionaria multiplicando por 2 y restando la parte entera. Los dígitos en binario son los restos. Tomamos los restos de izquierda a derecha

Decimal	0,625	0,25	0,5	0
Entera		$\boxed{1}$	0	1
		$\longrightarrow$		

El resultado es

$$(107{,}625)_{10} = (1101011{,}101)_2.$$

Representación binaria ($m = 4$ bits)	Enteros sin signo	Enteros con signo (signo en 1^{er} bit)	Enteros con signo $sesgo = 2^{m-1}$	Enteros con signo (exp.) $sesgo = 2^{m-1} - 1$
0000	0	$+0$	-8	Reservado
0001	1	$+1$	-7	-6
0010	2	$+2$	-6	-5
0011	3	$+3$	-5	-4
0100	4	$+4$	-4	-3
0101	5	$+5$	-3	-2
0110	6	$+6$	-2	-1
0111	7	$+7$	-1	0
1000	8	-0	0	1
1001	9	-1	1	2
1010	10	-2	2	3
1011	11	-3	3	4
1100	12	-4	4	5
1101	13	-5	5	6
1110	14	-6	6	7
1111	15	-7	7	Reservado

Cuadro 1.2: Representación de enteros con $m = 4$ bits.

2. Representación de enteros

Como dijimos, la norma IEEE 754 no se ocupa de la representación de enteros, pero como el exponente de la representación en punto flotante es un entero, vamos a dar unas breves nociones del almacenamiento de enteros en binario en el ordenador.

Si sólo necesitamos enteros positivos, con m bits podemos representar enteros con valores entre $(00\ldots00)_2 = (0)_{10}$ y $(11\ldots11)_2 = (2^m - 1)_{10}$. Por ejemplo, para $m = 4$ bits podríamos representar los enteros del 0 al 15, como se puede ver en la tabla 1.2.

Para representar enteros con signo hay varias estrategias, como reservar el primer bit para el signo, la representación al complemento de dos o la representación sesgada. En esta última, los números negativos se representan con valores consecutivos, empezando por el menor negativo y los positivos toman los valores siguientes. La representación de los números la obtenemos añadiendo el sesgo=2^{m-1} al número $x_r = x + 2^{m-1} \in [0, 2^m - 1]$. Es esta última representación la que se utiliza para los exponentes en punto flotante. No obstante, la representación sesgada para el exponente en punto flotante es diferente a la representación sesgada de enteros con signo en general. El motivo es que el primer y el último valor del exponente, se reservan para casos especiales (como infinito y `NaN`). El sesgo es diferente en los dos casos y el rango de números representados también. En el caso del exponente el sesgo es $2^{m-1} - 1$ y el rango de los números representados va es $[1 - sesgo, sesgo]$.

3. Representación binaria en punto flotante según la norma IEEE 754

Un número $x \neq 0$ en base 2 se representa en punto flotante como

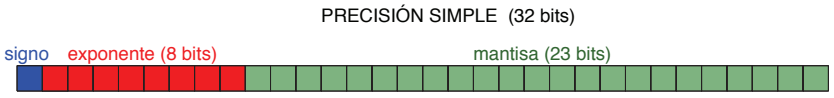
$$x = \sigma \times \bar{x} \times 2^e,$$

donde $\sigma = \pm 1$ es el signo, $(1)_2 \leq \bar{x} < (10)_2$ la mantisa y $e \in \mathbb{Z}$ es el exponente.

- En el bit del *signo* σ se almacena un 1 si el número es negativo y 0 si es positivo.
- La *mantisa* $\bar{x}$ normalizada tiene un valor $1 \leq \bar{x} < 2$. El primer dígito ha de ser distinto de 0. Este dígito en base dos solo puede ser 1 y por lo tanto, en una mantisa normalizada siempre es $a_1 = 1$, no es necesario almacenarlo y ganamos un bit. Esto se denomina *técnica del bit escondido*.
- El *exponente e* es un entero con signo y utiliza *representación sesgada*.

Los números se almacenan en "palabras" de 32 bits (precisión sencilla), 64 bits (doble precisión) y 128 bits (precisión cuádruple).

3.1. Precisión simple



$$x = \sigma \times (1.a_1 a_2 \ldots a_{23}) \times 2^e$$

Utiliza 32 bits (4 bytes) distribuidos en:

- 1 bit para el signo.
- 8 bits para el exponente.
- 23 bits para la mantisa.

Tiene una precisión de 24 dígitos binarios, uno más de los bits utilizados, 23, debido al *bit escondido*, que no se almacena porque siempre es 1.

El sesgo del exponente es $2^{m-1} - 1 = 2^{8-1} - 1 = 127$ y toma valores en $[1 - sesgo, sesgo] = [-126, 127]$.

Si tenemos 8 bits para el exponente con signo, significa que tenemos espacio para $2^8 = 256$ números en formato binario, $0 < E < (255)_{10}$. El primer valor 00000000 lo reservamos para cero y *números desnormalizados* y el último 11111111 para `Inf` (infinito) y `NaN` (Not a Number). Así que el exponente tomará valores $1 \leq E \leq 254$. Y como el sesgo es 127 estos valores representan a los exponentes $-126 \leq E - 127 \leq 127$, es decir, $-126 \leq e \leq 127$. Por lo tanto el exponente

máximo es $e_{max} = 127$, y el mínimo $e_{min} = -126$. Una ventaja de este sistema es que el inverso de un número cuyo exponente tenga el valor mínimo:

$$\frac{1}{\pm m 2^{e_{min}}} = \frac{1}{\pm m 2^{-126}} = \frac{1}{\pm m} 2^{126} < \frac{1}{\pm m} 2^{127}$$

no supera el valor máximo y no se produce *overflow*.

La representación sesgada hace más sencilla la comparación (si uno es mayor o menor que el otro) de números enteros. Cuando se quiere comparar dos números en punto flotante, se compara sus exponentes y, solo si sus exponentes son iguales, se continua comparando la mantisa.

Ejemplo 1.4 ¿Cómo se almacenaría en binario en precisión simple según la norma IEEE 754 el número $-118{,}625$?

Recordamos que en precisión simple tenemos 32 bits (4 bytes) en total, que se distribuyen de la siguiente manera:

- 1 bit para el signo,
- 8 para el exponente,
- 23 para la mantisa.

Parte decimal. Para convertir a binario la parte decimal, multiplicamos por 2, le quitamos la parte entera, que será nuestro dígito binario y repetimos el proceso hasta que nos quedemos con un cero

$$\begin{array}{rcccc} \textit{Decimal}: & 0{,}625 & 0{,}25 & 0{,}5 & 0 \\ \textit{Entera}: & 1 & 0 & 1 & \end{array}$$

y tomamos los dígitos de la parte entera:

$$0{,}101$$

Comprobamos que es correcto:

$$1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 0{,}625$$

Parte entera. Para convertir a binario la parte entera 118 dividimos de forma reiterada por 2 y guardamos los restos:

$$\begin{array}{rccccccc} \textit{cocientes}: & 118 & 59 & 29 & 14 & 7 & 3 & 1 \\ \textit{restos}: & 0 & 1 & 1 & 0 & 1 & 1 & \end{array}$$

y tomamos el último cociente y los restos en orden inverso:

$$1110110$$

El número completo en base dos se escribiría

$$1110110{,}101$$

Comprobamos que es correcto:

$$1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 118{,}625$$

Para almacenar según la norma IEEE, normalizamos el número:

$$1110110{,}101 = 1{,}110110101 \times 2^6$$

Por lo tanto la mantisa se almacena como

$$11011010100000000000000$$

Fijarse que no almacenamos el primer 1 debido a la técnica del bit escondido.

Exponente. El sesgo es $2^{m-1}-1 = 2^{8-1}-1 = 127$. El exponente sesgado en base 10 es $6+$ *sesgo* $= 6+127 = 133$. Si lo pasamos a base 2:

cocientes :	133	66	33	16	8	4	2	1
restos :		1	0	1	0	0	0	0

Tomamos el último cociente y los restos empezando por el último, el exponente es

$$10000101$$

Comprobamos que está bien

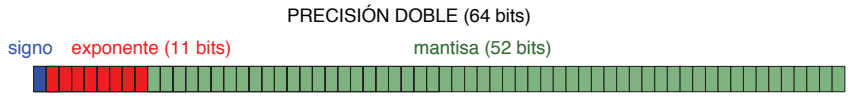
$$2^7 + 2^2 + 2^0 = 133$$

Como el signo del número es negativo, el bit del signo toma valor 1. Resumiendo, el número se almacenaría, en precisión simple como:

signo	exponente	mantisa
1	10000101	11011010100000000000000

□

3.2. Precisión doble



$$x = \sigma \times (1.a_1a_2\ldots a_{52}) \times 2^e$$

Utiliza 64 bits (8 bytes) distribuidos de la forma:

- 1 bit para el signo.
- 11 bits para el exponente.
- 52 bits para la mantisa.

Tiene una precisión de 53 dígitos binarios, uno más de los bits utilizados, 52, debido al *bit escondido*.

El sesgo del exponente es $2^{m-1}-1 = 2^{11-1}-1 = 1023$ y toma valores en $[-sesgo+1, sesgo] = [-1022, 1023]$.

	$m = 0$	$m \neq 0$
$e = 00000000000$	0	num. desnormalizados
$e = 11111111111$	$\pm$`Inf`	`NaN`

Cuadro 1.3: Excepciones en la representación en punto flotante con doble precisión

Si tenemos 11 bits para el exponente con signo, significa que tenemos espacio $2^{11} = 2048$ números en formato binario, $0 < E < (2047)_{10}$. El primer valor 00000000000 lo reservamos para cero y *números desnormalizados* y el último 11111111111 para `Inf` (infinito) y `NaN` (Not a Number). Así que el exponente tomará valores $1 \leq E \leq 2046$. Y como el sesgo es 1023 estos valores representan a los exponentes $-1022 \leq E - 1023 \leq 1023$, es decir, $-1022 \leq e \leq 1023$. Por lo tanto el exponente máximo es $e_{max} = 1023$, y el mínimo $e_{min} = -1022$.

3.3. Valores especiales

Los valores no normalizados para doble precisión se resumen en la tabla 1.3 y son los siguientes:

Infinito. El exponente contiene todo unos y la mantisa todo ceros. Aparece cuando se produce *overflow*.

Valor	signo	exponente	mantisa
$-\infty$	1	11111111	00000000000000000000
$+\infty$	0	11111111	00000000000000000000

`NaN` (Not a Number). El exponente contiene todo unos y la mantisa es distinta de cero. Hay dos clases `QNaN` (Quiet `NaN`) que significa *indeterminado* y `SNaN` (Signalling `NaN`) que significa *operación no válida.* Operaciones como $0/0$ o 0^0 devuelven `NaN`.

Valor	signo	exponente	mantisa
SNaN	0	11111111	10000000000000001000000
QNaN	1	11111111	00000010000000010000000

Cero. Como se asume que el bit escondido tiene valor 1, no es posible representar el cero con los valores normalizados. El cero se representa utilizando las dos representaciones siguientes:

Valor	signo	exponente	mantisa
-0	1	00000000	00000000000000000000000
$+0$	0	00000000	00000000000000000000000

Números desnormalizados. El exponente contiene todo ceros. Se asume que el bit escondido es cero y que el valor del exponente es el mínimo posible, es decir `00000001` (que equivale a -126 en precisión sencilla).

signo	exponente	mantisa
0	00000000	00001000010000000001000
1	00000000	01000100000000000010000

Números desnormalizados

Si consideramos que el bit escondido es cero, podemos representar números menores que R_{min}.

Estos números en la norma IEEE 754 se representan con exponente 00000000 en precisión simple y con exponente 00000000000 en precisión doble. Pero se interpreta que el valor de su exponente es el exponente mínimo, es decir -126 en precisión sencilla y -1022 en doble precisión.

El inconveniente de estos números es que su precisión es menor que 24 en precisión simple y menor que 53 en precisión doble.

Y su ventaja es que aumentan el rango de números a representar rellenando el espacio entre el menor número normalizado R_{min} y el cero.

Los números normalizados están, aproximadamente, espaciados logarítmicamente mientras que los desnormalizados están espaciados linealmente.

Ejemplo 1.5 El número siguiente está representado en precisión simple según la norma IEEE 754.

signo	exponente	mantisa
0	00000000	00010110000000000000000

¿Cuál es su valor en base 10?¿Cual es la precisión del número representado?

Como su exponente es 00000000 y su mantisa no es cero, es un número desnormalizado, su exponente es -126 y su bit escondido es 0. Por lo tanto representa el número

$$(0{,}0001011)\cdot 2^{-126}$$

que se corresponde con el número en base 10

$$(2^{-4}+2^{-6}+2^{-7})\cdot 2^{-126}\approx 1{,}0102\cdot 10^{-39}.$$

Este número tiene una precisión de sólo 20, puesto que a efectos de precisión no cuentan los tres ceros a la izquierda del primer uno de la mantisa.

El menor número normalizado, en valor absoluto, que se puede representar en precisión simple será

signo	exponente	mantisa
0	00000001	00000000000000000000000

$$(1{,}00\ldots00)\cdot 2^{-126}$$

en representación binaria. En decimal este número es:

$$R_{min}=2^{-126}\approx 1{,}1755\cdot 10^{-38}$$

que, lógicamente, es mayor que el mayor normalizado.

□

Ejemplo 1.6 ¿Cuál es el menor número desnormalizado en precisión simple?¿Y en doble precisión?

En precisión simple, el menor número desnormalizado sería

signo	exponente	mantisa
0	00000000	00000000000000000000001

que representa al número en base 2

$$(0{,}00000000000000000000001)\cdot 2^{-126}$$

y que se corresponde con el número en base 10

$$(2^{-23})\cdot 2^{-126} = 2^{-149} \approx 1{,}4013\cdot 10^{-45}$$

y tiene una precisión de 1. Razonando análogamente, el número más pequeño desnormalizado en doble precisión es

$$(2^{-52})\cdot 2^{-1022} = 2^{-1074} \approx 4{,}9407\cdot 10^{-324}$$

□

3.4. Exactitud

Si queremos medir la exactitud relativa de la aritmética en punto flotante, tenemos dos medidas:

Definición 2 El épsilon de la máquina *es la diferencia entre* 1 *y el número siguiente* $x > 1$ *que se puede almacenar de forma exacta.*

Definición 3 El entero más grande *es el entero más grande M tal que todos los enteros x, donde* $0 \leq x \leq M$*, se almacenan de forma exacta.*

El epsilon de la máquina. Si expresamos el número 1 en formato normalizado en precisión simple, sería

$$+1\times(1{,}00\ldots00)\times 2^0$$

el siguiente número que se puede almacenar de forma exacta es

$$1+\varepsilon = +1\times(1{,}00\ldots01)\times 2^0$$

y por lo tanto ε es

$$\varepsilon = +1\times(0{,}00\ldots01)\times 2^0$$

y si escribimos ε normalizado

$$\varepsilon = +1\times(0{,}00\overset{(23)}{\ldots}01)\times 2^0 = +1\times(1{,}00\overset{(23)}{\ldots}00)\times 2^{-23},$$

es decir, el epsilon de máquina en precisión simple es

$$\varepsilon = 2^{-23} \approx 1{,}19\times 10^{-7}.$$

	Decimal representado	Binario 25 dígitos	Mantisa 1.+23 bits	Exp	Representación
	1	000...001	1.00...000	0	Exacta
	2	000...010	1.00...000	1	Exacta
	3	000...011	1.10...000	1	Exacta
	4	000...100	1.00...000	2	Exacta
	⋮	⋮	⋮	⋮	⋮
	1677215	011...111	1.11...111	23	Exacta
$M = 2^{24} \rightarrow$	1677216	100...000	1.00...0000	24	Exacta
	1677216	100...001	1.00...0001	24	**Redondeada**
	1677218	100...010	1.00...0010	24	Exacta
	1677220	100...011	1.00...0011	24	**Redondeada**
	1677220	100...100	1.00...0100	24	Exacta
	⋮	⋮	⋮	⋮	⋮

Cuadro 1.4: Representación de enteros en punto flotante con precisión simple

Razonándo de forma análoga, el epsilon de máquina en precisión doble es

$$\varepsilon = 2^{-52} \approx 2{,}22 \times 10^{-16}.$$

El entero más grande. Si p es la precisión del formato, el entero más grande es:

$$M = 2^p.$$

Vamos a justificarlo con la tabla 1.4 para precisión simple y la tabla 1.5 para precición doble.

En precisión simple, podemos ver en la tabla 1.4 que todos los enteros anteriores a $M = 2^{24}$ admiten representación exacta porque podemos almacenar todos sus dígitos. En el caso de $M = 2^{24}$ no podemos almacenar el último dígito, pero al ser este 0 y siguiendo las normas de redondeo (que veremos en el punto 3.5) M se redondea al número más próximo acabado en 0 y, en este caso, no pierde dígitos y M se almacena de forma exacta. No sucede lo mismo con el siguiente, cuyo dígito más a la derecha es un 1, y al redondear el número este se almacena como el número anterior y ya no se almacena de forma exacta. A partir de este número algunos enteros se podrán almacenar de forma exacta y otros no. Como en base 10 este entero es

$$M = 2^{24} = 1677216$$

podemos interpretar que enteros con hasta 6 dígitos en base 10 se almacenan de forma exacta en punto flotante binario.

El número entero siguiente 1677217 ya no tiene representación exacta porque se redondea y es representado por 1677216. El número entero siguiente 1677218, sí que se representa de forma exacta porque, aunque no se pueden almacenar todos los bits, el bit despreciado es cero. El entero siguiente 1677219, se redondea a 1677220 y tampoco se puede representar de forma exacta.

El mismo razonamiento se puede hacer para doble precisión. Este está ilustrado en la tabla 1.5 y en este caso, el mayor entero es

$$M = 2^{53} = 9007199254740992$$

y podemos por tanto interpretar que enteros con hasta 15 dígitos en en base 10 se almacenan de forma exacta en punto flotante binario.

	Decimal representado	Binario 54 dígitos	Mantisa 1.+52 bits	Exp	Representación
	1	000...001	1.00...000	0	Exacta
	2	000...010	1.00...000	1	Exacta
	3	000...011	1.10...000	1	Exacta
	4	000...100	1.00...000	2	Exacta
	⋮	⋮	⋮	⋮	⋮
	9007199254740991	011...111	1.11...111	52	Exacta
$M = 2^{53} \rightarrow$	9007199254740992	100...000	1.00...0000	53	Exacta
	9007199254740992	100...001	1.00...0001	53	**Redondeada**
	9007199254740994	100...010	1.00...0010	53	Exacta
	9007199254740996	100...011	1.00...0011	53	**Redondeada**
	9007199254740996	100...100	1.00...0100	53	Exacta
	⋮	⋮	⋮	⋮	⋮

Cuadro 1.5: Representación de enteros en punto flotante con precisión doble

Overflow y underflow

El mayor número normalizado, en valor absoluto, que se puede representar en doble precisión será, en representación binaria

$$(1{,}11\ldots11)\cdot 2^{1023}$$

Como habíamos dicho, el primer 1 no hace falta guardarlo y nos quedan 52 bits donde almacenamos los demás unos. Por lo tanto, en decimal este número será:

$$R_{max} = (1+1\cdot 2^{-1}+1\cdot 2^{-2}+1\cdot 2^{-3}+\cdots+1\cdot 2^{-52})\cdot 2^{1023} \approx 1{,}7977\cdot 10^{308}$$

El menor número, en valor absoluto, normalizado que puede representar en doble precisión será,

$$(1{,}00\ldots00)\cdot 2^{-1022}$$

en representación binaria. Por lo tanto, en decimal este número será:

$$R_{min} = 2^{-1022} \approx 2{,}2251\cdot 10^{-308}$$

¿Qué sucede si intentamos almacenar un número mayor en valor absoluto que R_{max}? Se produce un *error de overflow*. En la mayor parte de los ordenadores se aborta la ejecución. El formato IEEE 754 puede darle soporte asignándole los valores simbólicos $\pm\infty$. A menudo, se debe a errores de programación, que deben ser corregidos.

¿Y qué sucede si intentamos almacenar un número menor en valor absoluto que R_{min}? Se produce un *error de underflow*. Desde la inclusión de los números desnormalizados, que se sitúan entre R_{min} y el cero, se utiliza el valor desnormalizado más cercano se pierde precisión. Es lo que se llama un *underflow gradual*. Si el número es menor que el menor desnormalizado se sustituye por cero y la ejecución continua.

3.5. Redondeo

Si escribimos un número x con notación en punto flotante en base 10 como

$$x = \pm d_0.d_1d_2\ldots \times 10^n = \pm \left(\sum_{k=0}^{\infty} d_k 10^{-k} \right) \times 10^n,$$

donde $d_0 \neq 0$ y $0 \leq d_k \leq 9$, para $k = 1, 2, \ldots$

Y si escribimos un número x con notación en punto flotante en base 2 como

$$x = \pm 1.d_1d_2\ldots \times 2^e = \pm \left(\sum_{k=0}^{\infty} d_k 2^{-k} \right) \times 2^e,$$

donde $d_0 \neq 0$ y $0 \leq d_k \leq 1$, para $k = 1, 2, \ldots$

Tanto para el caso decimal como el binario, si la mantisa tiene más de $p+1$ dígitos decimales, es decir,

$$d_k \neq 0 \quad \text{para algunos} \quad k > p-1,$$

entonces x no tiene un representación en punto flotante exacta con precisión p y cae entre dos números consecutivos $x^- < x < x^+$. Como representación de x elegiremos uno de los dos dependiendo del *método de redondeo* usado. La norma IEEE 754 reconoce 4 sistemas de redondeo:

- Hacia arriba.
- Hacia abajo.
- Hacia cero.
- Hacia el par más cercano.

Las dos formas habituales de redondeo son estas dos últimas a las que llamaremos respectivamente *truncamiento* y *redondeo*.

Redondeo decimal

Si tenemos un número real en base 10 cualquiera

$$x = \pm d_0.d_1d_2\ldots \times 10^n = \pm \left(\sum_{k=0}^{\infty} d_k 10^{-k} \right) \times 10^n$$

Truncamiento. Con $p+1$ dígitos:

$$x^* = \pm d_0.d_1d_2\ldots d_p \times 10^p,$$

Redondeo. Con p dígitos:

$$x^* = \begin{cases} \pm d_0.d_1d_2\ldots d_{p-1} \times 10^n & \text{si } 0 \leq d_p \leq 4, \\ \pm (d_0.d_1d_2\ldots d_{p-1} + 10^{-p+1}) \times 10^n & \text{si } 5 \leq d_p \leq 9, \\ \text{al número acabado en par más cercano} & \text{si } d_p = 5 \text{ y } d_{p+k} = 0. \end{cases}$$

Ejemplo 1.7 Redondear los siguientes números en base 10:

número	precisión	truncado	redondeado
0,999953	4	0,9999	1,000
0,433309	3	0,433	0,433
0,433500	3	0,433	0,434
0,434500	3	0,434	0,434

□

Redondeo binario

Si tenemos un número real en base 2 cualquiera

$$x = \pm 1.d_1 d_2 \ldots \times 2^e = \pm \left(\sum_{k=0}^{\infty} d_k 2^{-k} \right) \times 2^e$$

Truncamiento. Con $p+1$ dígitos:

$$x^* = \pm 1.d_1 d_2 \ldots d_p \times 2^e,$$

Redondeo. Con p dígitos:

$$x^* = \begin{cases} \pm 1.d_1 d_2 \ldots d_{p-1} \times 2^e & \text{si } d_p = 0, \\ \pm (1.d_1 d_2 \ldots d_{p-1} + 10^{-p+1}) \times 2^e & \text{si } d_p = 1, \\ \text{al número acabado en par más cercano} & \text{si } d_p = 1 \text{ y } d_{p+k} = 0. \end{cases}$$

Ejemplo 1.8 Redondear los siguientes números en base 2:

número	precisión	truncado	redondeado
1,1111	3	1,11	10,0
1,1101	3	1,11	1,11
1,0010	3	1,00	1,00
1,0110	3	1,01	1,10

□

Comparación entre truncado y redondeo en binario. La representación en punto flotante con precisión p de x puede expresarse como

$$x^* = x(1+\gamma), \quad \text{donde} \quad \gamma = \begin{cases} [-2^{-p+1}, 0] & \text{si truncamos,} \\ [-2^{-p}, 2^{-p}] & \text{si redondeamos.} \end{cases}$$

Por lo tanto, el mayor error de truncamiento es el doble que el mayor error de redondeo y el error de truncamiento es siempre no positivo, mientras que el error de redondeo puede cambiar de signo. Por lo tanto, lo errores se amplifican menos si usamos redondeo.

Ejemplo 1.9 Sea $x = (1{,}1001101)_2 = (1{,}6015625)_{10}$. Lo aproximamos por:

- truncamiento a 5 dígitos binarios,

$$x^* = (1{,}1001)_2 = (1{,}5625)_{10}$$

$$\gamma = -\frac{x - x^*}{x} = -0{,}0243902\ldots \in \left[-2^{-4}, 0\right]$$

- redondeo a 5 dígitos binarios,

$$x^* = (1{,}1010)_2 = (1{,}625)_{10}$$

$$\gamma = -\frac{x - x^*}{x} = 0{,}0146341\ldots \in \left[-2^{-5}, 2^{-5}\right].$$

□

4. Error

Los errores de redondeo que se deben a que la aritmética de la computación es finita son pequeños en cada operación, pero pueden acumularse y propagarse si un algoritmo tiene muchas operaciones o iteraciones, resultando en una diferencia grande entre la solución exacta y la solución calculada numéricamente. Este efecto se conoce como *inestabilidad numérica* del algoritmo.

Ejemplo 1.10 Para la sucesión $s_k = 1 + 2 + \ldots + k$, for $k = 1, 2, \ldots$, si calculamos

$$x_k = \frac{1}{s_k} + \frac{2}{s_k} + \ldots + \frac{k}{s_k},$$

el resultado es

$$x_k = 1 \text{ para todos los } k = 1, 2, \ldots$$

Sin embargo, en precisión simple obtenemos

k	x_k^*	$\lvert x_k - x_k^* \rvert$
10^1	1,000000	0,0
10^3	0,999999	$1{,}0 \times 10^{-7}$
10^6	0,9998996	$1{,}004 \times 10^{-4}$
10^7	1,002663	$2{,}663 \times 10^{-3}$

□

Definición 4 El error absoluto *que se comete al aproximar el valor de x con x^* es*

$$e_a = |x - x^*|$$

Definición 5 El error relativo *que se produce al aproximar el valor de x con x^**

$$e_r = \frac{|x - x^*|}{|x|}.$$

El error relativo es independiente de la escala y por tanto es más significativo que el error absoluto, como podemos ver en el siguiente ejemplo.

Ejemplo 1.11 Dar los errores absolutos y relativos correspondientes a los valores de x al aproximarlos con x^*.

x	x^*	e_a	e_r
$0{,}3 \times 10^1$	$0{,}31 \times 10^1$	$0{,}1$	$0{,}333... \times 10^{-1}$
$0{,}3 \times 10^{-3}$	$0{,}31 \times 10^{-3}$	$0{,}1 \times 10^{-4}$	$0{,}333... \times 10^{-1}$
$0{,}3 \times 10^4$	$0{,}31 \times 10^4$	$0{,}1 \times 10^3$	$0{,}333... \times 10^{-1}$

□

Definición 6 *Decimos que* x^* aproxima a x *con* p dígitos significativos *si* p *es el mayor entero no negativo tal que el error relativo satisface*

$$\frac{|x-x^*|}{|x|} \leq 5 \times 10^{-p}.$$

Ejemplo 1.12 Estudiemos los dígitos significativos en los casos siguientes:

$x^* = 124{,}45$ aproxima $x = 123{,}45$ con $p = 2$ dígitos significativos porque

$$\frac{|x-x^*|}{|x|} = \frac{1}{123{,}45} = 0{,}0081 \leq 0{,}05 = 5 \times 10^{-2}.$$

$x^* = 0{,}0012445$ aproxima $x = 0{,}0012345$ con $p = 2$ dígitos significativos porque

$$\frac{|x-x^*|}{|x|} = \frac{0{,}00001}{0{,}0012345} = 0{,}0081 \leq 0{,}05 = 5 \times 10^{-2}.$$

$x^* = 999{,}8$ aproxima $x = 1000$ con $p = 4$ dígitos significativos porque

$$\frac{|x-x^*|}{|x|} = \frac{0{,}2}{1000} = 0{,}0002 \leq 0{,}0005 = 5 \times 10^{-4}.$$

□

Capítulo 2

Ecuaciones no lineales

1. Introducción

En este capítulo estudiaremos métodos numéricos para calcular aproximaciones de las raíces o ceros de ecuaciones no linelaes del tipo

$$f(x) = 0,$$

donde $f : \mathbb{R} \to \mathbb{R}$ es una función dada. En general, las soluciones de (1) no pueden ser expresadas en forma explícita. Además, incluso cuando existe una fórmula que las identifica, a menudo es demasiado complicada como para que resulte útil.

Los métodos numéricos que estudiaremos son de naturaleza iterativa. Comenzando por una aproximación inicial y mediante algún algoritmo, se produce una sucesión de aproximaciones que, presumiblemente, convergerán a la raíz deseada.

Los métodos iterativos deben ser truncados o parados después de un número finito de pasos, por lo que sólo obtendremos aproximaciones de las raíces que buscamos. Además, los errores de redondeo que se generan en las evaluaciones de la función $f(x)$ también limitarán la precisión de cualquier método numérico.

Con ciertos métodos, como el método de biseccion, es suficiente saber el intervalo inicial que contiene la raíz para asegurar la convergencia, aunque ésta sea lenta. Sin embargo, otros métodos, aunque más rápidos, son más sensibles a la elección de la aproximación inicial. Por ello, normalmente se usa un método híbrido en el cual se comienza con, por ejemplo, el método de biseccion para acercarnos a la solución y después se aplica un método más eficiente para refinar esta aproximación, como el método de Newton.

1.1. Orden de convergencia y criterio de parada

En las líneas anteriores hemos introducido unos conceptos que merece la pena precisar. Los métodos numéricos de cálculo de raíces de una función son *métodos iterativos*, es decir, mediante algún algoritmo construímos una sucesión

$$x_0, x_1, \ldots, x_k, \ldots$$

tal que $\lim_{k\to\infty} x_k = \alpha$, y esperamos que, por continuidad de la función f, se tenga

$$\lim_{k\to\infty} f(x_k) = f(\alpha) = 0.$$

El orden de convergencia de un método está relacionado con la idea intuitiva de velocidad de convergencia de la sucesión con respecto a k, que es un concepto útil para comparar métodos. Supongamos que la sucesión x_k converge a $\alpha \in \mathbb{R}$. Decimos que x_k *converge a* α *con orden de convergencia* p si

$$\lim_{k\to\infty} \frac{|x_k - \alpha|}{|x_{k-1} - \alpha|^p} = \lambda \neq 0.$$

En los casos particulares

- $p = 1$, decimos que la convergencia es *lineal*,
- $p = 2$, la convergencia es *cuadrática*.

Un método numérico se dice de *orden* p si genera una sucesión que converge a la solución con un orden de convergencia p.

Por otra parte, puesto que la sucesión generada con los algoritmos, x_k, suele ser infinita, se hace necesario establecer un criterio de parada de las iteraciones. El criterio más crudo es el de establecer un número máximo de iteraciones. Sin embargo, tal criterio no proporciona ninguna información sobre la exactitud de la aproximación. Los criterios más habituales se basan en

- La diferencia absoluta entre dos iteraciones sucesivas,

$$|x_k - x_{k-1}| < \varepsilon.$$

- La diferencia relativa entre dos iteraciones sucesivas,

$$\frac{|x_k - x_{k-1}|}{|x_k|} < \varepsilon.$$

- El residuo en la iteración k,

$$|f(x_k)| < \varepsilon.$$

En la práctica, suelen usarse una combinación de estos criterios conjuntamente con el establecimiento de un número máximo de iteraciones, para prevenir bucles infinitos (porque ε sea demasiado pequeño) o, simplemente, tiempos de ejecución demasiado largos.

2. El método de bisección

Normalmente es aconsejable comenzar por reunir información cualitativa sobre las raíces a aproximar. Por ejemplo, trataremos de determinar cuántas raíces existen y su localización aproximada. Esta información puede conseguirse mediante la representación gráfica de la función $f(x)$, la cual suele ser una herramienta útil para determinar el número de raíces y los intervalos que contienen cada raíz.

Ejemplo 2.1 Consideremos la ecuación

$$f(x) = (x/2)^2 - \text{sen}(x) = 0.$$

En la Figura 2.1 mostramos los gráficos de $y = (x/2)^2$ e $y = \text{sen}(x)$. Mediante la simple observación, podemos determinar que la única raíz positiva se encuentra en el intervalo $(1{,}8, 2)$, siendo $\alpha \approx 1{,}9$. □

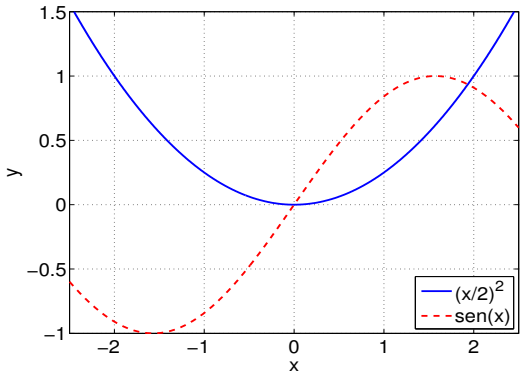


Figura 2.1: Gráfica de las curvas $y = (x/2)^2$ e $y = \text{sen}(x)$.

El siguiente teorema puede usarse para inferir si el intervalo $[a,b]$ contiene al menos una raíz de la ecuación $f(x) = 0$.

Teorema 2.1 (Teorema de los valores intermedios) *Supongamos que la función $f(x)$ es continua para todo $x \in [a,b]$, con $f(a) \neq f(b)$, y que k es un valor situado entre $f(a)$ y $f(b)$. Entonces, existe un punto $\xi \in (a,b)$ tal que $f(\xi) = k$.*

En particular, si $f(a)f(b) < 0$ entonces la ecuación $f(x) = 0$ tiene, al menos, una raíz en el intervalo (a,b).

El método de biseccion hace un uso sistemático del teorema de los valores intermedios. Supongamos que $f(x)$ es continua en el intervalo $[a_0, b_0]$ y que $f(a_0)f(b_0) < 0$. En lo que sigue, determinaremos una sucesión anidada de intervalos $I_k = [a_k, b_k]$ tales que

$$(a_0, b_0) \supset (a_1, b_1) \supset (a_2, b_2) \supset \cdots$$

los cuales contendrán la raíz de la ecuación. Los intervalos se determinan recursivamente como sigue. Dado $I_k = (a_k, b_k)$, calculamos el punto medio

$$m_k = \frac{a_k + b_k}{2} = a_k + \frac{1}{2}(b_k - a_k),$$

y $f(m_k)$. Esta última expresión tiene la ventaja de que permite calcular el punto medio sin errores de redondeo.

Podemos asumir que $f(m_k) \neq 0$ pues de otro modo habríamos localizado la raíz. El nuevo intervalo se determina como sigue

$$I_{k+1} = (a_{k+1}, b_{k+1}) = \begin{cases} (m_k, b_k) & \text{si } f(m_k)f(a_k) > 0, \\ (a_k, m_k) & \text{si } f(m_k)f(a_k) < 0. \end{cases}$$

De esta construcción se sigue que $f(a_{k+1})f(b_{k+1}) < 0$, y por tanto el intervalo I_{k+1} también contiene una raíz de $f(x) = 0$.

Tras n aplicaciones del método de bisección, la raíz estará contenida en el intervalo (a_n, b_n), de longitud $2^{-n}(b_0 - a_0)$. Es decir, si tomamos a m_n como una aproximación de la raíz de $f(x)$, tenemos la estimación del error absoluto

$$|\alpha - m_n| < 2^{-n}(b_0 - a_0). \tag{2.1}$$

En cada paso ganamos un dígito binario de exactitud de la aproximación, o bien, en promedio, un dígito decimal por cada 3,3 iteraciones. De este modo, encontrar un intervalo de longitud δ que contenga una raíz llevará unas $\log_2((b_0 - a_0)/\delta)$ evaluaciones de $f(x)$.

La fórmula (2.1) nos permite deducir que el método de bisección tiene un orden de convergencia lineal. Claramente, el test de parada estará basado en el error absoluto entre dos iteraciones, el cual nos permite determinar el número de iteraciones precisas.

Ejemplo 2.2 El método de biseccion aplicado a la ecuación $f(x) = 0$, con $f(x) = (x/2)^2 - \text{sen}(x) = 0$ e $I_0 = (1{,}8, 2)$ produce la sucesión de intervalos $[a_k, b_k]$, donde

k	a_k	b_k	m_k	$f(m_k)$
1	1.8	2	1.9	< 0
2	1.9	2	1.95	> 0
3	1.9	1.95	1.925	< 0
4	1.925	1.95	1.9375	> 0
5	1.925	1.9375	1.93125	< 0
6	1.93125	1.9375	1.934375	> 0

De modo que, tras seis iteraciones, obtenemos $\alpha \in (1{,}93125, 1{,}934375)$, un intervalo de longitud $0{,}2 \times 2^{-6} = 0{,}003125$. □

El tiempo de ejecución requerido por el método de biseccion es proporcional al número de evaluaciones de la función $f(x)$ y, por tanto, la convergencia es lenta. Pero independiente de la regularidad de la función. Para funciones regulares, por ejemplo las derivables, otros métodos tales como el de Newton proporcionan una convergencia mucho más rápida.

3. El método de Newton

La única información que usa el el método de bisección es el signo de $f(x)$ en los extremos de los sucesivos intervalos generados por el mismo. Si la función es regular, por ejemplo derivable, se puede construir un método más eficiente explotando los valores alcanzados por $f(x)$ y su derivada en las aproximaciones sucesivas.

Sea $f : [a, b] \to \mathbb{R}$ una función derivable, y consideremos la aproximación por la recta tangente a f en el punto $x_k \in (a, b)$ dada por

$$y(x) = f(x_k) + f'(x_k)(x - x_k).$$

Si fijamos x_{k+1} d modo que $y(x_{k+1}) = 0$, es decir, que sea una aproximación de una raíz de $f(x)$, obtenemos

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k \geq 0, \tag{2.2}$$

siempre que $f'(x_k) \neq 0$. La fórmula (2.2) se conoce como método de Newton y corresponde a calcular el cero de $f(x)$ reemplazando localmente $f(x)$ por su recta tangente en x_k.

Observemos que para inicializar el método de Newton necesitamos una primera aproximación del cero, x_0. Esta elección es delicada puesto que el método no converge, en general, para cualquier elección de la aproximación incial. En la práctica, se puede obtener un posible valor inicial x_0 recurriendo a una cuantas iteraciones del método de bisección o, alternativamente, a través de una investigación de la gráfica de $f(x)$.

Si x_0 se escoge apropiadamente y α es un cero *simple* (esto es, $f'(\alpha) \neq 0$) entonces el método de Newton converge. Además, si $f''(x)$ es continua, puede demostrarse que el método tiene convergencia cuadrática.

El test de parada más utilizado para el método de Newton y, en general, para los métodos de punto fijo que estudiaremos en la sección 4, es el de la diferencia absoluta entre dos iterantes consecutivos

$$|x_{k+1} - x_k| < \varepsilon, \tag{2.3}$$

para cierta tolerancia ε. Como en el caso del método de bisección, en la práctica, limitaremos el número de iteraciones a un máximo prefijado, para evitar bucles infinitos.

El método de Newton puede extenderse fácilmente a sistemas de ecuaciones no lineales. En efecto, si $\mathbf{f} : A \to \mathbb{R}^N$, con $A \subset \mathbb{R}^N$ viene dada por

$$\begin{cases} f_1(x_1, x_2, \ldots, x_N) = 0, \\ f_2(x_1, x_2, \ldots, x_N) = 0, \\ \vdots \\ f_N(x_1, x_2, \ldots, x_N) = 0, \end{cases}$$

entonces el método de Newton para resolver $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, donde $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ y $\mathbf{f} = (f_1, \ldots, f_N)$, viene dado como sigue: dado $\mathbf{x}_0 \in A$, para $k = 0, 1, \ldots$ y hasta la convergencia, definimos

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \big(J_{\mathbf{f}}(\mathbf{x}_k)\big)^{-1} \mathbf{f}(\mathbf{x}_k),$$

donde $J_{\mathbf{f}}(\mathbf{x}_k)$ es la matriz jacobiana de $\mathbf{f}(\mathbf{x})$ evaluada en $\mathbf{x}_k$, esto es

$$\big(J_{\mathbf{f}}(\mathbf{x}_k)\big)_{ij} = \frac{\partial f_i}{\partial x_j}(\mathbf{x}_k).$$

Obsérvese que al igual que para funciones escalares $f'(x_k)$ debe ser distinta de cero, para funciones vectoriales debe satisfacerse que exista la inversa del jacobiano, para lo cual basta con que $\det\big(J_{\mathbf{f}}(\mathbf{x}_k)\big) \neq 0$. En cuanto al test de parada, sustituiremos (2.3) por

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon.$$

4. El método del punto fijo

En esta sección introduciremos una clase general de métodos iterativos entre los que se encuentra, por ejemplo, el método de Newton.

Se dice que la función $g : [a,b] \to \mathbb{R}$ tiene un punto fijo $\alpha \in [a,b]$ si $g(\alpha) = \alpha$. El método del punto fijo se basa en la iteración

$$x_{k+1} = g(x_k), \quad k \geq 0, \tag{2.4}$$

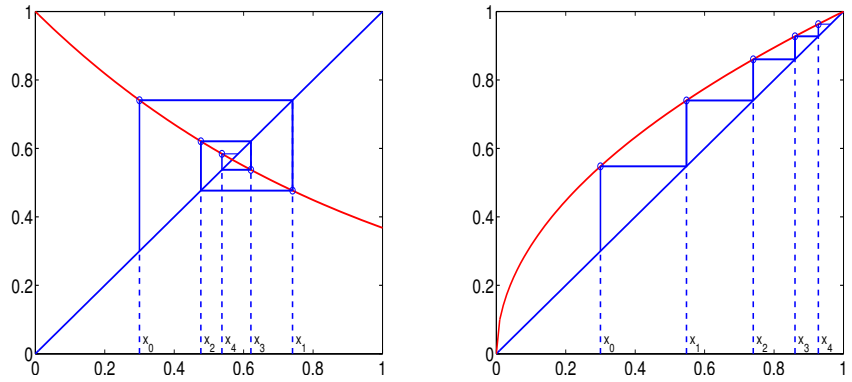
donde x_0 es la aproximación inicial que debemos proporcionar.

El método del punto fijo es de una gran generalidad y da pie a la introducción de otros algoritmos cuando se particulariza la función g. Por ejemplo, si queremos aproximar un cero de la función $f : [a,b] \to \mathbb{R}$ por el método del punto fijo, basta con definir $g(x) = x + f(x)$, de modo que si α es un punto fijo de g entonces también será una raíz de f. Sin embargo, no existe una manera única de expresar esta equivalencia, como muestra el siguiente ejemplo.

Ejemplo 2.3 La ecuación $x + \ln(x) = 0$ puede reescribirse, por ejemplo, como

$$(i)\ x = -\ln(x), \quad (ii)\ x = \mathrm{e}^{-x}, \quad (iii)\ x = \frac{x + \mathrm{e}^{-x}}{2}.$$

Observemos que cada una de estas ecuaciones da lugar a diferentes esquemas de punto fijo. □



(a) $x_{k+1} = \mathrm{e}^{-x_k}$, con $x_0 = 0{,}3$ (b) $x_{k+1} = \sqrt{x_k}$, con $x_0 = 0{,}01$

Figura 2.2: Ejemplos de iteraciones de punto fijo: convergente (izquierda) y divergente de la raíz más cercana (derecha).

Una interpretación gráfica del método del punto fijo puede verse en la Figura 2.2. Como se aprecia, en ciertos casos el método puede no converger incluso comenzando con una aproximación inicial arbitrariamente cercana a la raíz. Por ello, se hace necesario establecer criterios que aseguren la convergencia del método.

Teorema 2.2 (Teorema de la aplicación contractiva) *Sea g una función definida en el intervalo $[a,b] \subset \mathbb{R}$ y $x_0 \in [a,b]$ una aproximación inicial de la iteración de punto fijo dada por (2.4). Supongamos que*

1. *$g(x) \in [a,b]$ para todo $x \in [a,b]$,*
2. *g es diferenciable en $[a,b]$,*
3. *Existe una constante $k < 1$ tal que $|g'(x)| \leq k$ para todo $x \in [a,b]$.*

Entonces g tiene un único punto fijo $\alpha \in [a,b]$, y la sucesión x_n definida en (2.4) converge a α al menos con orden lineal. Más precisamente,

$$\lim_{n\to\infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|} = g'(\alpha).$$

En general, el test de parada utilizado para detener las iteraciones de punto fijo se basa en la diferencia absoluta entre dos iterantes consecutivos, como ya introdujimos para el método de Newton, véase (2.3).

Observación 2.1 *El método de Newton puede obtenerse a partir del método del punto fijo cuando se toma*

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

5. El método de la secante

Uno de los principales inconvenientes del método de Newton es que necesitamos evaluar la derivada de la función en los puntos definidos por la iteración. En ocasiones esto no es posible debido a que la función sólo es conocida en un número finito de puntos. Imaginemos, por ejemplo, el caso en que la función es obtenida mediante la toma de datos de una variable física, como la temperatura.

El método de la secante es una variante del método de Newton en el cual se aproxima la derivada de la función, $f'(x)$ por un cociente incremental. Puesto que

$$f'(x) = \lim_{y \to x} \frac{f(x) - f(y)}{x - y},$$

podemos aproximar $f'(x_{k-1})$ mediante

$$f'(x_{k-1}) \approx \frac{f(x_{k-1}) - f(x_{k-2})}{x_{k-1} - x_{k-2}}.$$

De este modo, obtenemos el esquema iterativo siguiente. Dadas dos aproximaciones iniciales x_0 y x_1, se toma, para $k = 2, 3 \ldots$,

$$x_k = x_{k-1} - f(x_{k-1}) \frac{x_{k-1} - x_{k-2}}{f(x_{k-1}) - f(x_{k-2})}, \tag{2.5}$$

siempre que $f(x_{k-1}) \neq f(x_{k-2})$.

Cuando el método de la secante converge, el término $|x_{k-1} - x_{k-2}|$ ha de hacerse, eventualmente, muy pequeño, y como consecuencia el cociente $(x_{k-1} - x_{k-2})/(f(x_{k-1}) - f(x_{k-2}))$ será determinado con una exactitud numérica pobre, ya que si las aproximaciones x_{k-1}, x_{k-2} están muy cerca de la raíz α, entonces el error de redondeo puede ser grande.

Sin embargo, un análisis del error nos permite concluir que, en general, las aproximaciones satisfacen $|x_{k-1} - x_{k-2}| \gg |x_{k-1} - \alpha|$ y, por tanto, la contribución dominante al error de redondeo viene de $f(x_{k-1})$.

Observemos que la fórmula (2.5) no debe simplificarse como

$$x_k = \frac{x_{k-2} f(x_{k-1}) - x_{k-1} f(x_{k-2})}{f(x_{k-1}) - f(x_{k-2})},$$

puesto que esta fórmula puede dar lugar a errores de cancelación cuando $x_{k-1} \approx x_{k-2}$ y $f(x_{k-1}) f(x_{k-2}) > 0$. Incluso la fórmula (2.5) puede no resultar segura dado que cuando $f(x_{k-1}) \approx f(x_{k-2})$ podríamos

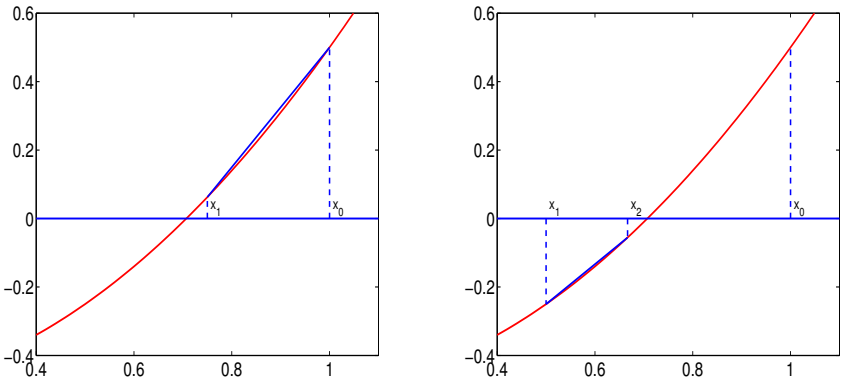


Figura 2.3: Una iteración de los métodos de Newton (izquierda) y de la secante (derecha) para la función $f(x) = x^2 - 0{,}5$.

dividir por cero o por números cercanos a cero y producir un *overflow*. Por ello, resulta más conveniente realizar las iteraciones del siguiente modo

$$s_{k-1} = \frac{f(x_{k-1})}{f(x_{k-2})}, \quad x_k = x_{k-1} + \frac{s_{k-1}}{1 - s_{k-1}}(x_{k-1} - x_{k-2}),$$

donde la división por $1 - s_{k-1}$ solo se lleva a cabo si $1 - s_{k-1}$ es suficientemente grande.

Finalmente, puede demostrarse que el orden de convergencia del método de la secante es menor que el de Newton, y viene dado por $p = (1 + \sqrt{5})/2 \approx 1{,}618\ldots$. El criterio de parada utilizado es similar al del método de Newton.

Capítulo 3

Interpolación y aproximación

A menudo, en la solución de problemas, necesitamos calcular el valor de una función en uno o varios puntos. Sin embargo, pueden surgir inconvenientes tales como

- puede ser costoso en términos de uso de procesador o tiempo de máquina, por ejemplo, una función complicada que hay que evaluar muchas veces;
- es posible que solo tengamos el valor de la función en unos pocos puntos, por ejemplo, si proceden de los datos de un muestreo.

Una posible solución es sustituir esta función complicada o parcialmente desconocida por otra, sencilla, que podamos evaluar eficientemente. Estas funciones más sencillas suelen ser polinomios, polinomios trigonométricas, funciones racionales, etc.

1. Interpolación

La interpolación de una función dada, f, por otra función $\tilde{f}$ consiste en, dados los datos siguientes

- $n+1$ puntos distintos $x_0,\ x_1,\ldots,\ x_n$,
- $n+1$ valores de f en dichos puntos, $f(x_0)=\omega_0,\ f(x_1)=\omega_1,\ldots,\ f(x_n)=,\omega_n$,

hallar una función *sencilla*, $\tilde{f}$, que verifique $\tilde{f}(x_i)=\omega_i$, con $i=0,1,\ldots,n$.

A los puntos $x_0,\ x_1,\ldots,x_n$ se les denomina *nodos de interpolación*, y a la función $\tilde{f}$ *interpolante de* f *en* $x_0,\ x_1,\ldots,x_n$. En lo que sigue, consideraremos dos tipos de interpolantes:

- La interpolante polinómica, del tipo

$$\tilde{f}(x)=a_0+a_1x+a_2x^2+\ldots+a_nx^n=\sum_{k=0}^{n}a_kx^k.$$

- La interpolante trigonométrica, del tipo

$$\tilde{f}(x)=a_{-M}e^{-iMx}+\ldots+a_0+\ldots+a_Me^{iMx}=\sum_{k=-M}^{M}a_ke^{ikx},$$

donde $M = n/2$ si n es par, y $M = (n-1)/2$ si n es impar. Recordemos que i denota la unidad imaginaria, y que $e^{ikx} = \cos(kx) + i\,\mathrm{sen}(kx)$.

- La interpolante polinómica a trozos, del tipo

$$\tilde{f}(x) = \begin{cases} p_1(x) & \text{si } x \in (\tilde{x}_0, \tilde{x}_1) \\ p_2(x) & \text{si } x \in (\tilde{x}_1, \tilde{x}_2) \\ \cdots & \\ p_m(x) & \text{si } x \in (\tilde{x}_{m-1}, \tilde{x}_m) \end{cases}$$

donde $\tilde{x}_0, \ldots, \tilde{x}_m$ forman una subdivisión o *partición* del intervalo que contiene a los nodos de interpolación, (x_0, x_n), y $p_i(x)$ son polinomios.

2. Interpolación polinómica: el polinomio de Lagrange

Buscamos un polinomio interpolador (sustituímos la notación $\tilde{f}$ por P_n)

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n, \tag{3.1}$$

que satisfaga las condiciones

$$P_n(x_0) = \omega_0, \quad P_n(x_1) = \omega_1, \quad P_n(x_2) = \omega_2, \quad \ldots \quad P_n(x_n) = \omega_n. \tag{3.2}$$

Evaluando la expresión del polinomio (3.1) en los nodos de interpolación e igualando a los valores ω_i, obtenemos que las condiciones (3.2) son equivalentes a que los coeficientes del polinomio sean solución del sistema de ecuaciones lineales

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_n \end{pmatrix}.$$

La matriz de coeficientes del sistema es de tipo V*andermode*,

$$A = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix},$$

cuyo determinante viene dado por

$$\det(A) = \prod_{0 \le l \le k \le n} (x_k - x_l).$$

Claramente, como los nodos de interpolación son distintos, tenemos $\det(A) \neq 0$, y por tanto el sistema tiene solución única, es decir, existe un único polinomio P_n que cumple las condiciones (3.2).

Tal polinomio, P_n, es denominado *polinomio de interpolación de Lagrange* en los puntos $x_{0,}x_1, \ldots, x_n$ relativo a los valores $\omega_{0,}\omega_1, \ldots, \omega_n$.

Si el número de nodos, n, es grande la resolución del sistema lineal anterior puede ser muy costosa. Sin embargo, existen métodos alternativos que facilitan enormemente el cálculo del polinomio de Lagrange, entre ellos, los que usan los *polinomios fundamentales de Lagrange*, y las *diferencias divididas*. Comenzamos con los primeros.

2.1. Polinomios fundamentales de Lagrange

Es un resultado básico que paara cada $i = 0, 1, \ldots, n$, existe un único polinomio ℓ_i de grado $\leq n$ tal que $\ell_i(x_k) = \delta_{ik}$, donde δ_{ik} denota la *delta de Kronecker*[1]. Dicho polinomio viene dado por

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}. \tag{3.3}$$

A los polinomios $\ell_0, \ell_1, \ldots, \ell_n$ se les denomina *polinomios fundamentales de Lagrange* de grado n. Observemos que estos polinomios solo dependen de los nodos de interpolación, x_i, no de los valores, ω_i. Es decir, los polinomios fundamentales no son interpolantes, sino una herramienta cómoda para construir los interpolantes.

El *polinomio de interpolación de Lagrange* en los puntos $x_{0,}x_1, \ldots, x_n$ relativo a los valores $\omega_{0,}\omega_1, \ldots, \omega_n$ viene dado por

$$P_n(x) = \omega_0 \ell_0(x) + \omega_1 \ell_1(x) + \cdots + \omega_n \ell_n(x).$$

Claramente, como en el nodo x_i el único polinomio fundamental distinto de cero es $\ell_i(x)$ (que vale uno en x_i), tendremos

$$P_n(x_i) = \omega_i,$$

para $i = 0, \ldots, n$, es decir, $P_n(x)$ satisface las condiciones de interpolación.

Ejemplo 3.1 Consideremos, para $i = 0, 1, 2$, los nodos $x_i = i$ y los valores $\omega_i = f(x_i)$, con $f(x) = 1/(x+1)$. Tenemos que

$$\ell_0(x) = \frac{x - x_1}{x_0 - x_1} \frac{x - x_2}{x_0 - x_2} = \frac{x-1}{-1} \frac{x-2}{-2} = \frac{1}{2}(x-1)(x-2).$$

Análogamente, obtenemos

$$\ell_1(x) = -x(x-2), \quad \ell_2(x) = \frac{1}{2}x(x-1),$$

de modo que

$$P_2(x) = \frac{1}{2}(x-1)(x-2) - \frac{1}{2}x(x-2) + \frac{1}{6}x(x-1).$$

□

Un inconveniente de este modo de calcular el polinomio de Lagrange es que, una vez obtenido el polinomio de grado n, si vemos que la aproximación no es suficientemente buena, deberemos rehacer todos los cálculos para deducir el polinomio de grado $n+1$. Para evitar este inconveniente tenemos el *método de diferencias divididas*, ideado por Newton.

[1] $\delta_{ik} = 0$ si $i \neq k$, $\delta_{ik} = 1$ si $i = k$.

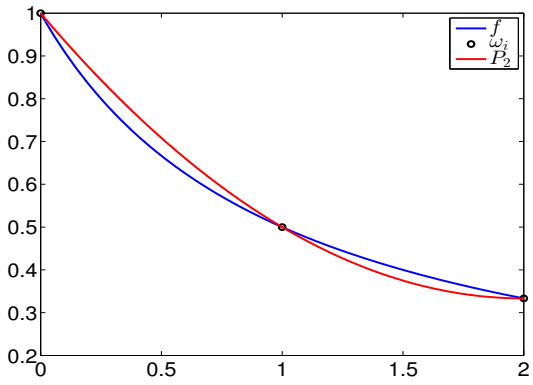


Figura 3.1:

2.2. Diferencias divididas

Podemos reescribir el polinomio de interpolación de Lagrange como

$$P_n(x) = c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + \cdots + c_n(x-x_0)\cdots(x-x_n), \tag{3.4}$$

donde $c_0, \ldots, c_n$ son constantes a determinar. Para $x = x_0$ tenemos que $P_n(x_0) = c_0$, y también, por las condiciones de interpolación, que $P_n(x_0) = \omega_0$, es decir, obtenemos $c_0 = \omega_0$.

Dividiendo la expresión (3.4) por $(x-x_0)$ y teniendo en cuenta que $c_0 = \omega_0$, obtenemos

$$\frac{P_n(x) - \omega_0}{x - x_0} = c_1 + c_2(x-x_1) + \cdots + c_n(x-x_1)\cdots(x-x_n), \tag{3.5}$$

y evaluando en $x = x_1$ deducimos que

$$c_1 = \frac{P_n(x_1) - \omega_0}{x_1 - x_0} = \frac{\omega_1 - \omega_0}{x_1 - x_0}.$$

Siguiendo con esta idea, dividimos la expresión (3.5) por $(x-x_1)$ para obtener

$$\frac{1}{x-x_1}\left(\frac{P_n(x) - \omega_0}{x - x_0} - \frac{\omega_1 - \omega_0}{x_1 - x_0}\right) = c_2 + c_3(x-x_2) + \cdots + c_n(x-x_2)\cdots(x-x_n),$$

y evaluando en $x = x_2$ deducimos que

$$c_2 = \frac{1}{x_2 - x_1}\left(\frac{\omega_2 - \omega_0}{x_2 - x_0} - \frac{\omega_1 - \omega_0}{x_1 - x_0}\right).$$

Con unas sencillas manipulaciones aritméticas, podemos reescribir c_2 como

$$c_2 = \frac{\dfrac{\omega_2 - \omega_1}{x_2 - x_1} - \dfrac{\omega_1 - \omega_0}{x_1 - x_0}}{x_2 - x_0}.$$

Recapitulando, e introduciendo la notación usual de las diferencias divididas, tenemos

$$\begin{aligned}
c_0 &= [\omega_0] = \omega_0, \\
c_1 &= [\omega_0, \omega_1] = \frac{\omega_1 - \omega_0}{x_1 - x_0}, \\
c_2 &= [\omega_0, \omega_1, \omega_2] = \frac{\dfrac{\omega_2 - \omega_1}{x_2 - x_1} - \dfrac{\omega_1 - \omega_0}{x_1 - x_0}}{x_2 - x_0}.
\end{aligned}$$

La observación clave es que podemos expresar las diferencias divididas de segundo orden, $[\omega_0, \omega_1, \omega_2]$, usando solo las de primer orden, $[\omega_1, \omega_2]$ y $[\omega_0, \omega_1]$. En efecto,

$$[\omega_0, \omega_1, \omega_2] = \frac{[\omega_1, \omega_2] - [\omega_0, \omega_1]}{x_2 - x_0}.$$

A partir de estas ideas, definimos las:

- diferencias divididas de orden 0

$$[\omega_i] = \omega_i \quad \text{para } i = 0, 1, \ldots, n,$$

- diferencias divididas de orden k $(k = 1, \ldots, n)$

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

  para $i = 0, 1, \ldots, n-k$.

En la práctica, el cálculo de las diferencias divididas se dispone como en la siguiente tabla:

x_0	ω_0	$[\omega_0, \omega_1]$	$[\omega_0, \omega_1, \omega_2]$	$\cdots$	$[\omega_0, \omega_1, \ldots, \omega_n]$
x_1	ω_1	$[\omega_1, \omega_2]$	$[\omega_1, \omega_2, \omega_3]$	$\cdots$	
x_2	ω_2	$[\omega_2, \omega_3]$	$[\omega_2, \omega_3, \omega_4]$	$\cdots$	
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	
x_{n-1}	ω_{n-1}	$[\omega_{n-1}, \omega_n]$			
x_n	ω_n				

Una vez calculadas las diferencias divididas asociadas a un problema de interpolación polinómica, el correspondiente polinomio de interpolación de Lagrange de grado n puede calcularse de la siguiente manera (Fórmula de Newton):

$$\begin{aligned} P_n(x) = {} & [\omega_0] + [\omega_0, \omega_1](x - x_0) + [\omega_0, \omega_1, \omega_2](x - x_0)(x - x_1) + \cdots + \\ & + [\omega_0, \omega_1, \ldots, \omega_n](x - x_0)(x - x_1) \cdots (x - x_{n-1}), \end{aligned}$$

siendo la principal ventaja de esta formulación que los polinomios de Lagrange de órdenes sucesivos pueden calcularse recursivamente, es decir

$$P_n(x) = P_{n-1}(x) + [\omega_0, \omega_1, \ldots, \omega_n](x - x_0)(x - x_1) \cdots (x - x_{n-1}).$$

Observación 3.1 *A menudo se usa la notación $f[x_0, x_1, \ldots, x_n]$ en lugar de $[\omega_0, \omega_1, \ldots, \omega_n]$. En tal caso, la fórmula de Newton se escribe así:*

$$\begin{aligned} P_n(x) = {} & f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots + \\ & + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1}). \end{aligned} \tag{3.6}$$

Ejemplo 3.2 Consideremos de nuevo los datos del Ejemplo 3.1, esto es, para $i = 0,1,2$, los nodos $x_i = i$ y los valores $\omega_i = 1/(i+1)$. Tenemos que

$$\begin{aligned}
&[\omega_i] = \omega_i,\\
&[\omega_0,\omega_1] = \frac{\omega_1-\omega_0}{x_1-x_0} = \frac{\frac{1}{2}-1}{1-0} = -\frac{1}{2},\\
&[\omega_1,\omega_2] = \frac{\omega_2-\omega_1}{x_2-x_1} = \frac{\frac{1}{3}-\frac{1}{2}}{1-0} = -\frac{1}{6},\\
&[\omega_0,\omega_1,\omega_2] = \frac{[\omega_1,\omega_2]-[\omega_0,\omega_1]}{x_2-x_0} = \frac{-\frac{1}{6}+\frac{1}{2}}{2} = \frac{1}{6}.
\end{aligned}$$

De modo que la tabla es

$$\begin{array}{c|ccc}
0 & 1 & -\frac{1}{2} & \frac{1}{6}\\
1 & \frac{1}{2} & -\frac{1}{6} & \\
2 & \frac{1}{3} & &
\end{array}$$

y el polinomio de Lagrange queda como

$$P_2(x) = 1 - \frac{1}{2}x + \frac{1}{6}x(x-1).$$

Si incorporamos un nuevo dato en el punto $x_3 = 3$, dad poro $\omega_3 = 1/4$, solo debemos calcular las diferencias divididas que nos faltan, esto es

$$\begin{aligned}
&[\omega_2,\omega_3] = \frac{\omega_3-\omega_2}{x_3-x_2} = \frac{\frac{1}{4}-\frac{1}{3}}{1-0} = -\frac{1}{12},\\
&[\omega_1,\omega_2,\omega_3] = \frac{[\omega_2,\omega_3]-[\omega_1,\omega_2]}{x_3-x_1} = \frac{-\frac{1}{12}+\frac{1}{6}}{2} = \frac{1}{24},\\
&[\omega_0,\omega_1,\omega_2,\omega_3] = \frac{[\omega_1,\omega_2,\omega_3]-[\omega_0,\omega_1,\omega_2]}{x_3-x_0} = \frac{\frac{1}{24}-\frac{1}{6}}{3} = -\frac{1}{24},
\end{aligned}$$

añadirlas a la tabla,

$$\begin{array}{c|cccc}
0 & 1 & -\frac{1}{2} & \frac{1}{6} & -\frac{1}{24}\\
1 & \frac{1}{2} & -\frac{1}{6} & \frac{1}{24} & \\
2 & \frac{1}{3} & -\frac{1}{12} & & \\
3 & \frac{1}{4} & & &
\end{array}$$

y así obtendremos el polinomio de Lagrange de orden 3,

$$P_3(x) = 1 - \frac{1}{2}x + \frac{1}{6}x(x-1) - \frac{1}{24}x(x-1)(x-2).$$

□

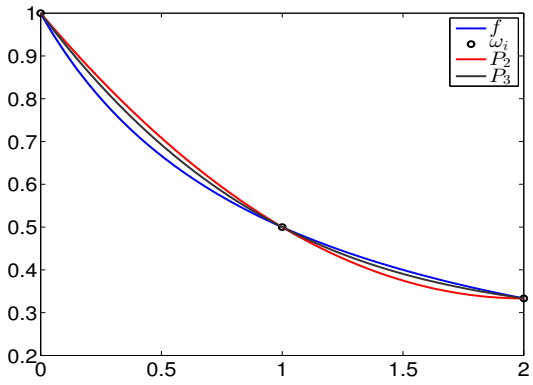


Figura 3.2:

2.3. Estimación del error

Utilizando el resultado siguiente podemos evaluar el error obtenido cuando reemplazamos f por su polinomio de interpolacion de Lagrange P_n.

Teorema 3.1 *Supongamos que*

- $f:[a,b]\to\mathbb{R}$ *es* $n+1$ *veces derivable en* $[a,b]$ *con derivadas continuas.*
- $x_0,x_1,\ldots,x_n\in[a,b]$
- $\omega_i=f(x_i)$, *para* $i=0,1,\ldots,n$.

Entonces, para todo $x\in[a,b]$ *se tiene*

$$|f(x)-P_n(x)|\leq \max_{y\in[a,b]}\left|f^{(n+1)}(y)\right|\frac{|(x-x_0)(x-x_1)\cdots(x-x_n)|}{(n+1)!}.$$

En el caso más habitual en el que los nodos están equiespaciados, es decir, $x_i=x_{i-1}+h$, para alguna constante $h>0$, la estimación del error puede simplificarse como sigue

$$\max_{x\in[a,b]}|f(x)-P_n(x)|\leq\frac{\sup_{y\in[a,b]}\left|f^{(n+1)}(y)\right|}{4(n+1)}h^{n+1}.$$

Desafortunadamente, no podemos deducir de esta estimación que el error tienda a 0 cuando el grado del polinomio tiende a infinito, a pesar de que $h^{n+1}/(4(n+1))$ tiende a 0 puesto que las derivadas $f^{(n)(x)}$ podrían tender a infinito en algún punto. De hecho, existen funciones para las cuales el límite puede ser incluso infinito.

3. Interpolación por polinomios a trozos

Como vimos en la sección anterior, cuando aumentamos el número de nodos en la interpolación de Lagrange sucede lo siguiente:

- aumenta el grado del polinomio de interpolación, con el inconveniente de que los polinomios de grado alto presentan muchas oscilaciones;
- no necesariamente mejorar la aproximación (disminuye el error). Para que mejore, todas las derivadas de la función interpolada deben estár acotadas uniformemente.

Un modo de evitar esta situación es introducir como interpolantes las llamadas *funciones polinómocas a trozos*. Aunque de este modo se pierde algo de regularidad en la interpolante obtenida, nos aseguramos de que el error disminuya cuando aumenta el número de nodos.

Un polinomio de grado n queda unívocamente determinado por $n+1$ puntos distintos. Así, la interpolación por polinomios de grado 0 a trozos (constantes a trozos) será aquella en la que los polinomios, en este caso constantes, vengan determinados por cada nodo, por ejemplo

$$\tilde{f}(x) = \begin{cases} \omega_0 & \text{si } x \in [x_0, x_1), \\ \omega_1 & \text{si } x \in [x_1, x_2), \\ \dots & \\ \omega_{n-1} & \text{si } x \in [x_{n-1}, x_n), \\ \omega_0 & \text{si } x = x_n. \end{cases}$$

Observemos que si $\omega_i \neq \omega_{i+1}$ entonces $\tilde{f}$ tiene una discontinuidad en x_{i+1}.

Análogamente, la interpolación por polinomios de grado 1 a trozos (lineales a trozos) será aquella en la que los polinomios, en este caso líneas rectas, vengan determinados por cada dos nodos,

$$\tilde{f}(x) = \omega_i + (\omega_{i+1} - \omega_i)\frac{x - x_i}{x_{i+1} - x_i} \quad \text{si } x \in [x_i, x_{i+1}]$$

para $i = 0, \dots, n-1$. En este caso, $\tilde{f}$ es siempre continua pero su primera derivada es, en general, discontinua en los nodos.

Junto a la interpolación constante a trozos y lineal a trozos, la interpolación por polinomios de grado 3 a trozos, o interpolación por *splines cúbicos*, son las más importantes dentro de este grupo de interpoladores.

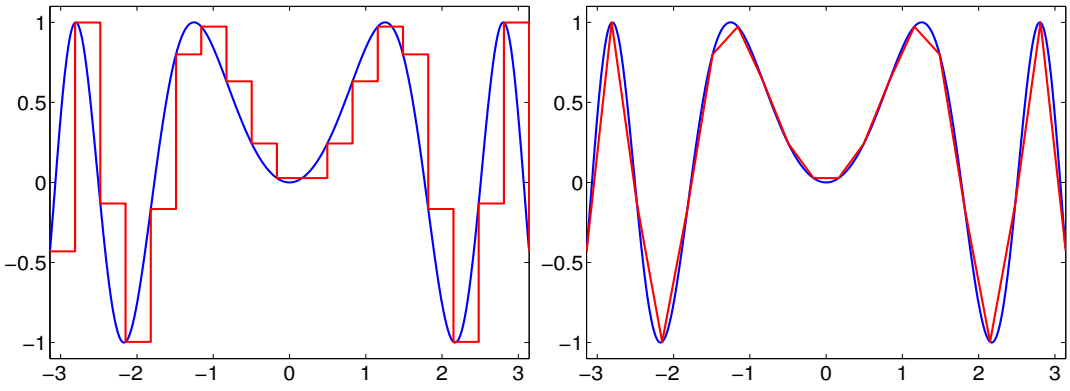


Figura 3.3: Izquierda: interpolación constante a trozos. Derecha: interpolación lineal a trozos.

3.1. Interpolación por splines

En general, el problema de interpolación por splines de orden p (o grado p) consiste en encontrar una función $\tilde{f}$ tal que

1. $\tilde{f}$ es una función derivable con cnotinuidad $p-1$ veces en el intervalo $[x_0, x_n]$;
2. $\tilde{f}$ es una función a trozos formada por los polinomios $\tilde{f}_0, \tilde{f}_1, \ldots, \tilde{f}_{n-1}$ que se definen, respectivamente, en $[x_0, x_1], [x_1, x_2], \ldots, [x_{n-1}, x_n]$ y que son de grado menor o igual que p;
3. los polinomios pasan por los nodos: $\tilde{f}_0(x_0) = \omega_0$, $\tilde{f}_1(x_1) = \omega_1$,..., $\tilde{f}_n(x_n) = \omega_n$.

Puede probarse que, para cada $p \geq 1$, este problema tiene solución. A estas soluciones, $\tilde{f}$, les llamamos *spline interpolador de orden p en los puntos* $x_0, x_1, \ldots, x_n$ *relativo a los valores* $\omega_0, \omega_1, \ldots, \omega_n$. El spline más utilizado es el de orden $p = 3$, conocido como *spline cúbico*.

Particularizando las condiciones anteriores al caso $p = 3$ vemos que el spline cúbico debe satisfacer

1. $\tilde{f}$ es 2 veces derivable en $[x_0, x_n]$ con derivada continua.
2. Cada uno de los polinomios $\tilde{f}_0, \tilde{f}_1, \ldots, \tilde{f}_{n-1}$ que componen los trozos de $\tilde{f}$ son de grado ≤ 3.
3. Los polinomios pasan por los nodos: $\tilde{f}(x_0) = \omega_0 \quad \tilde{f}(x_1) = \omega_1 \quad \ldots \quad \tilde{f}(x_n) = \omega_n$

Veamos cómo calcular los splines cúbicos.

Paso 1: Por definición, $\tilde{f}$ tiene derivada segunda continua en $[x_0, x_n]$. Por lo tanto $\tilde{f}''$ es continua y en particular

$$\begin{array}{lcl} \omega_0'' & = & \tilde{f}_0''(x_0), \\ \omega_1'' & = & \tilde{f}_0''(x_1) = \tilde{f}_1''(x_1), \\ \omega_2'' & = & \tilde{f}_1''(x_2) = \tilde{f}_2''(x_2), \\ \cdots & \cdots & \cdots \\ \omega_{n-1}'' & = & \tilde{f}_{n-2}''(x_{n-1}) = \tilde{f}_{n-1}''(x_{n-1}), \\ \omega_n'' & = & \tilde{f}_{n-1}''(x_n), \end{array}$$

donde ω_i'' denota el valor (desconocido) de $\tilde{f}''(x_i)$.

Paso 2: Los polinomios $\tilde{f}_i$ son de grado ≤ 3. Por lo tanto, $\tilde{f}_i''$ son de grado ≤ 1, es decir, rectas o constantes, con valores ω_i'' y ω_{i+1}'' en los extremos del intervalo $[x_i, x_{i+1}]$, respectivamente. Así, tenemos que, para $i = 0, \ldots, n-1$,

$$\tilde{f}_i''(x) = \omega_i'' \frac{x_{i+1} - x}{h_i} + \omega_{i+1}'' \frac{x - x_i}{h_i},$$

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

Paso 3: Integrando cada uno de estos polinomios con respecto a x, obtenemos

$$\tilde{f}_i'(x) = -\omega_i'' \frac{(x_{i+1} - x)^2}{2h_i} + \omega_{i+1}'' \frac{(x - x_i)^2}{2h_i} + c_i,$$

donde c_i es una constante de integración desconocida.

Paso 4: Integrado otra vez, obtenemos

$$\tilde{f}_i(x) = \omega_i'' \frac{(x_{i+1} - x)^3}{6h_i} + \omega_{i+1}'' \frac{(x - x_i)^3}{6h_i} + a_i(x_{i+1} - x) + b_i(x - x_i),$$

donde a_i y b_i son constantes de integración desconocidas y que absorben la constante c_i del Paso 3.

Paso 5: Determinamos las constantes a_i y b_i imponiendo las condiciones de interpolación:

$$\tilde{f}_i(x_i) = \omega_i \quad \tilde{f}_i(x_{i+1}) = \omega_{i+1}.$$

Así obtenemos, para $i = 0, \ldots, n-1$,

$$a_i = \frac{\omega_i}{h_i} - \omega_i'' \frac{h_i}{6}, \qquad b_i = \frac{\omega_{i+1}}{h_i} - \omega_{i+1}'' \frac{h_i}{6}.$$

Puesto que los valores ω_i'', para $i = 0, \ldots, n$, son desconocidos, el sistema resultante no está aún determinado. Usando que el interpolador $\tilde{f}$ es (dos veces) derivable con continuidad en $[x_0, x_n]$, obtenemos que debe satisfacerse en los nodos interiores

$$\tilde{f}_i'(x_{i+1}) = \tilde{f}_{i+1}'(x_{i+1}), \quad i = 0, \ldots, n-2,$$

lo que nos da $n-1$ ecuaciones lineales para la determinación de los $n+1$ valores ω_i''. Obviamente, el sistema sigue indeterminado puesto que tenemos dos incógnitas más que ecuaciones.

Existen varias estrategias para acabar de determinar el sistema de ecuaciones, dando lugar cada una de ellas a una variante distinta de los splines cúbicos. Por ejemplo, si fijamos el valor de 2 incógnitas, digamos $\omega_0'' = \omega_{n+1}'' = 0$, la variente se denomina *spline natural*, y los restantes ω_i, $i = 1, \ldots, n$ son solución única del sistema

$$\mathbf{H}\omega'' = \mathbf{d},$$

con

$$\mathbf{H} = \begin{pmatrix} 2(h_0+h_1) & h_1 & 0 & \cdots & 0 & 0 \\ h_1 & 2(h_1+h_2) & h_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2(h_{n-3}+h_{n-2}) & h_{n-2} \\ 0 & 0 & 0 & \cdots & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{pmatrix}$$

$$\omega'' = \begin{pmatrix} \omega_1'' \\ \omega_2'' \\ \vdots \\ \omega_{n-2}'' \\ \omega_{n-1}'' \end{pmatrix}, \qquad \mathbf{d} = \begin{pmatrix} \Delta_1 - \Delta_0 \\ \Delta_2 - \Delta_1 \\ \vdots \\ \Delta_{n-2} - \Delta_{n-3} \\ \Delta_{n-1} - \Delta_{n-2} \end{pmatrix},$$

donde $\Delta_i = (\omega_{i+1} - \omega_i)/h_i$.

3.2. Estimación del error

El siguiente resultado nos proporciona una estimación del error para la interpolación con polinomios a trozos. Observemos que cualquiera que sea el de los polinomios, podemos hacer disminuir el error tanto como deseemos tomando la distancia entre los nodos suficientemente pequeña.

Teorema 3.2 *Supongamos que*

- $f : [a,b] \to \mathbb{R}$ *es* $p+1$ *veces derivable en* $[a,b]$ *con derivadas continuas.*

- $x_0, x_1, \ldots, x_n \in [a,b]$
- $\omega_i = f(x_i)$, *para* $i = 0, 1, \ldots, n$.

Sea $\tilde{h} = \max\limits_{i=0,\ldots,n} h_i$. *Entonces, para todo* $x \in [a,b]$ *se tiene*

$$\left|f(x) - \tilde{f}(x)\right| \leq c\tilde{h}^{p+1} \max_{y\in[a,b]} \left|f^{(p+1)}(y)\right|,$$

siendo c *una constante independiente de* f, x *y* $\tilde{h}$.

Ejemplo 3.3 Tomemos la función $f : [0, 2\pi] \to \mathbb{R}$, $f(x) = \operatorname{sen}(x)$, y los nodos de interpolación $x_j = 2\pi j/N$, con $j = 0, 1, \ldots, N$. Entonces, $\tilde{h} = 2\pi/N$ y

$$\max_{y\in[0,2\pi]} \left|f^{(p+1)}(y)\right| \leq 1.$$

Deducimos que el error absoluto está acotado como

$$\left|\operatorname{sen}(x) - \tilde{f}(x)\right| \leq \frac{c}{N^{p+1}},$$

de modo que el error absoluto tiende a cero con un orden de convergencia $p+1$. □

4. Interpolación mediante polinomios trigonométricos

El objetivo más usual en la interpolación con polinomios trigonométricos es la interpolación de funciones periódicas, es decir, de funciones $f : [a,b] \to \mathbb{R}$ tales que $f(a) = f(b)$. Por comodidad y sin pérdida de generalidad[2], consideraremos en esta sección el intervalo $[a,b] = [0, 2\pi]$.

El interpolante, $\tilde{f}$, ha de satisfacer

$$\tilde{f}(x_j) = f(x_j), \quad \text{donde } x_j = \frac{2\pi j}{n+1}, \quad \text{para } j = 0, \ldots, n,$$

y tener la forma, si n es par,

$$\tilde{f}(x) = \frac{a_0}{2} + \sum_{k=1}^{M} \Big(a_k \cos(kx) + b_k \operatorname{sen}(kx)\Big), \tag{3.7}$$

con $M = n/2$, mientras que si n es impar será

$$\tilde{f}(x) = \frac{a_0}{2} + \sum_{k=1}^{M} \Big(a_k \cos(kx) + b_k \operatorname{sen}(kx)\Big) + a_{M+1} \cos((M+1)x), \tag{3.8}$$

con $M = (n-1)/2$. Usando la identidad $e^{ikx} = \cos(kx) + i\operatorname{sen}(kx)$ podemos reescribir (3.7) y (3.8) como

$$\tilde{f}(x) = \sum_{k=-M}^{M} c_k e^{ikx} \quad \text{si } n \text{ es par}, \qquad \tilde{f}(x) = \sum_{k=-M+1}^{M+1} c_k e^{ikx} \quad \text{si } n \text{ es impar},$$

[2] Si el periodo de la función es otro, por ejemplo T, el cambio de variable $x = 2\pi t/T$ la convierte en 2π-periódica.

donde

$$a_k = c_k + c_{-k}, \quad b_k = i(c_k - c_{-k}), \quad \text{para } k = 0,\dots,M, \qquad c_{M+1} = c_{-(M+1)} = a_{M+1}/2.$$

Usando la notación

$$\tilde{f}(x) = \sum_{k=-M+\mu}^{M+\mu} c_k e^{ikx},$$

con $\mu = 0$ sin n es par y $\mu = 1$ si n es impar, las condiciones de interpolación se escriben como

$$\tilde{f}(x_j) = \sum_{k=-M+\mu}^{M+\mu} c_k e^{ikjh} = f(x_j), \quad j = 0,\dots,n,$$

donde $h = 2\pi/(n+1)$.

Para calcular los coeficientes c_k, multiplicamos (3.10) por $e^{-imx_j} = e^{-imjh}$, con $m \in \mathbb{Z}$, y sumamos respecto j,

$$\sum_{j=0}^{n} \sum_{k=-M+\mu}^{M+\mu} c_k e^{ikjhe^{-imjh}} = \sum_{j=0}^{n} f(x_j) e^{-imjh}. \tag{3.9}$$

Usando la identidad

$$\sum_{j=0}^{n} e^{ijh(k-m)} = (n+1)\delta_{km},$$

obtenemos

$$\sum_{j=0}^{n} \sum_{k=-M+\mu}^{M+\mu} c_k e^{ikjhe^{-imjh}} = \sum_{k=-M+\mu}^{M+\mu} c_k (n+1)\delta_{km} = (n+1)c_m.$$

Finalmente, de (3.9) deducimos (intercambiando m por k)

$$c_k = \frac{1}{n+1} \sum_{j=0}^{n} f(x_j) e^{-ikjh}, \quad k = -(M+\mu),\dots,M+\mu.$$

Resumimos todos estos cálculos en la siguiente definición.

Definición 7 *Dada $f : [0, 2\pi] \to \mathbb{R}$, se define su* serie de Fourier discreta *en los nodos $x_j = jh$, con $h = 2\pi/(n+1)$ y $j = 0,\dots,n$ por*

$$\tilde{f}(x) = \sum_{k=-M+\mu}^{M+\mu} c_k e^{ikx}, \tag{3.10}$$

donde $c_k = \frac{1}{n+1} \sum_{j=0}^{n} f(x_j) e^{-ikjh}$ y con $M = n/2$ y $\mu = 0$ sin n es par, o $M = (n-1)/2$ y $\mu = 1$ si n es impar.

Ejemplo 3.4 Sea $f(x)$ una función cualquiera y consideremos los nodos $x_j = jh$ con $h = 2\pi/3$, para $j = 0, 1, 2$. Es decir, $x_0 = 0$, $x_1 = 2\pi/3$, $x_2 = 4\pi/3$ y $n = 2$. Entonces

$$c_k = \frac{1}{3}\big(f(0) + f\big(\frac{2\pi}{3}\big)e^{-ik\frac{2\pi}{3}} + f\big(\frac{4\pi}{3}\big)e^{-ik\frac{4\pi}{3}}\big),$$

luego

$$c_{-1} = \frac{1}{3}\big(f(0) + f(\frac{2\pi}{3})e^{i\frac{2\pi}{3}} + f(\frac{4\pi}{3})e^{i\frac{4\pi}{3}}\big)$$
$$c_0 = \frac{1}{3}\big(f(0) + f(\frac{2\pi}{3}) + f(\frac{4\pi}{3})\big),$$
$$c_1 = \frac{1}{3}\big(f(0) + f(\frac{2\pi}{3})e^{-i\frac{2\pi}{3}} + f(\frac{4\pi}{3})e^{-i\frac{4\pi}{3}}\big)$$

De modo que

$$\begin{aligned}\tilde{f}(x) = \sum_{k=-1}^{1} c_k e^{ikx} = \frac{1}{3}\Big[&\big(f(0) + f(\tfrac{2\pi}{3})e^{i\frac{2\pi}{3}} + f(\tfrac{4\pi}{3})e^{i\frac{4\pi}{3}}\big)e^{-ix} + \big(f(0) + f(\tfrac{2\pi}{3}) + f(\tfrac{4\pi}{3})\big) \\ &+ \big(f(0) + f(\tfrac{2\pi}{3})e^{-i\frac{2\pi}{3}} + f(\tfrac{4\pi}{3})e^{-i\frac{4\pi}{3}}\big)e^{ix}\Big] \\ = \frac{1}{3}\Big[&f(0)\big(1 + e^{-ix} + e^{ix}\big) + f(\tfrac{2\pi}{3})\big(1 + e^{-i(x-\frac{2\pi}{3})} + e^{i(x-\frac{2\pi}{3})}\big) \\ &+ f(\tfrac{4\pi}{3})\big(1 + e^{-i(x-\frac{4\pi}{3})} + e^{i(x-\frac{4\pi}{3})}\big)\Big].\end{aligned}$$

Usando las fórmulas trigonométricas, deducimos finalmente que

$$\tilde{f}(x) = \frac{1}{3}\Big[f(0)\big(1 + 2\cos(x)\big) + f(\tfrac{2\pi}{3})\big(1 + 2\cos(x - \tfrac{2\pi}{3})\big) + f(\tfrac{4\pi}{3})\big(1 + 2\cos((x - \tfrac{4\pi}{3})\big)\Big].$$

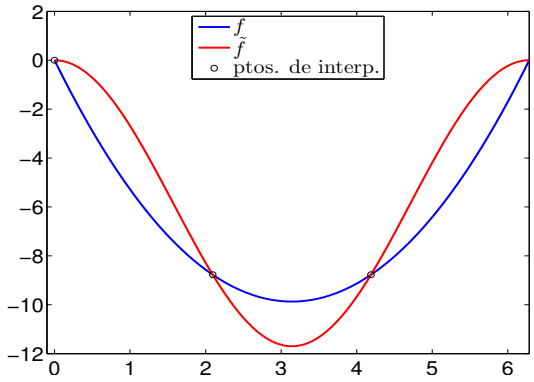


Figura 3.4: La función $f(x) = x(x - 2\pi)$ y su interpolante.

□

5. Aproximación

Ya vimos que la interpolación de Lagrange no garantiza una mejor aproximación de una función dada cuando el grado del polinomio crece. Este problema puede ser resuelto mediante la interpolación compuesta (tal como la interpolación lineal a trozos o por splines). Sin embargo, ninguna de ellas es apropiada para extrapolar información de los datos disponibles, esto es, para generar nuevos valores en puntos situados fuera del intervalo donde se dan los nodos de interpolación.

Para esta tarea usaremos los métodos de aproximación, en los cuales la condición de interpolación, $\tilde{f}(x_j) = f(x_j)$, no se satisface necesariamente.

6. Método de mínimos cuadrados

Supongamos que se dispone de los datos $\{(x_i, y_i), i = 0, \ldots, n\}$, donde ahora y_i podrían representar los valores $f(x_i)$ alcanzados por una función dada f en los nodos x_i. Para un entero dado $m \geq 1$ (usualmente, $m \ll n$) buscamos un polinomio $\tilde{f}$ de grado m (escribimos $\tilde{f} \in \mathcal{P}_m$) que satisfaga la desigualdad

$$\sum_{i=0}^{n} |y_i - \tilde{f}(x_i)|^2 \leq \sum_{i=0}^{n} |y_i - p_m|^2$$

para todo polinomio $p_m \in \mathcal{P}_m$. Si existe, $\tilde{f}$ será llamada *aproximación de mínimos cuadrados* en $\mathcal{P}_m$ del conjunto de datos $\{(x_i, y_i), i = 0, \ldots, n\}$. Salvo que $m \geq n$, en general no será posible garantizar que $\tilde{f}(x_i) = y_i$ para todo $i = 0, \ldots, n$.

Poniendo

$$\tilde{f}(x) = a_0 + a_1 x + \ldots + a_m x^m,$$

donde los coeficientes $a_0, \ldots, a_m$ son desconocidos, el problema puede ser replanteado como sigue: hallar $a_0, a_1, \ldots, a_m$ tales que

$$\Phi(a_0, a_1, \ldots, a_m) = \min_{\{b_i, i=0,\ldots,m\}} \Phi(b_0, b_1, \ldots, b_m),$$

donde

$$\Phi(b_0, b_1, \ldots, b_m) = \sum_{i=0}^{n} |y_i - (b_0 + b_1 x_i + \ldots + b_m x_i^m)|^2,$$

lo cual puede llevarse a cabo mediante las técnicas habituales del cálculo diferencial.

Resolvamos el problema en el caso $m = 1$, es decir, para un polinomio de aproximación lineal (regresión lineal, en término estadísticos). En este caso tenemos

$$\Phi(b_0, b_1) = \sum_{i=0}^{n} \left(y_i^2 + b_0^2 + b_1^2 x_1^2 + 2 b_0 b_1 x_i - 2 b_0 y_i - 2 b_1 x_i y_i^2 \right).$$

El punto (a_0, a_1) en el que Φ alcanza el mínimo viene determinado por

$$\frac{\partial \Phi}{\partial b_0}(a_0, a_1) = 0, \quad \frac{\partial \Phi}{\partial b_1}(a_0, a_1) = 0.$$

Calculando estas derivadas parciales obtenemos las condiciones

$$\sum_{i=0}^{n} \left(a_0 + a_1 x_i - y_i \right) = 0, \quad \sum_{i=0}^{n} \left(a_0 x_i + a_1 x_i^2 - x_i y_i \right) = 0,$$

que podemos reordenar como

$$a_0 (n+1) + a_1 \sum_{i=0}^{n} x_i = \sum_{i=0}^{n} y_i,$$
$$a_0 \sum_{i=0}^{n} x_i + a_1 \sum_{i=0}^{n} x_i^2 = \sum_{i=0}^{n} x_i y_i.$$

Este sistema de dos ecuaciones con dos incógnitas tiene por solución

$$a_0 = \frac{1}{D}\Big(\sum_{i=0}^{n} y_i \sum_{j=0}^{n} x_j^2 - \sum_{j=0}^{n} x_j \sum_{i=0}^{n} x_i y_i\Big),$$
$$a_1 = \frac{1}{D}\Big((n+1)\sum_{i=0}^{n} x_i y_i - \sum_{j=0}^{n} x_j \sum_{i=0}^{n} y_i\Big),$$

donde $D = (n+1)\sum_{i=0}^{n} x_i^2 - \Big(\sum_{i=0}^{n} x_i\Big)^2$. De este modo hemos calculado la *recta de mínimos cuadrados* $\tilde{f}(x) = a_0 + a_1 x$, que es la recta que mejor aproxima, en el sentido de los mínimos cuadrados, al conjunto de datos dado.

Ejemplo 3.5 Supongamos que el tiempo de ejecución, t, de un programa depende de un parámetro de entrada, j, y que hemos tomado las siguientes mediciones:

j	10	15	25	50	100
t	1	1.2	2	3.5	6

Realizando los cálculos necesarios, obtenemos la recta de regresión

$$\tilde{f}(x) = 0{,}5015 + 0{,}056x.$$

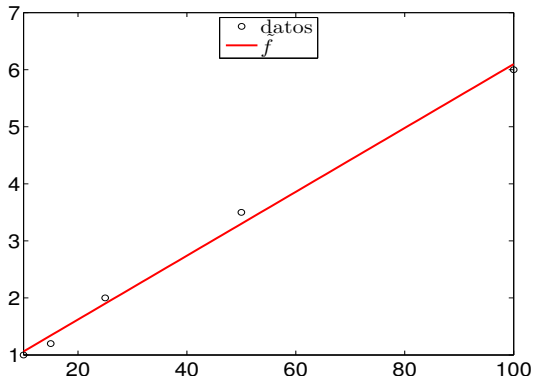


Figura 3.5:

□

7. Bases ortogonales

En esta sección trataremos el caso en el que la función a aproximar, f, es conocida en todo el intervalo $[a,b]$, y no simplemente en algunos de sus puntos. Nuestro objetivo será proporcionar, a partir de una función f que puede tener una expresión complicada, otra $\tilde{f}$ que se le *parezca* pero que tenga una expresión más simple, tal como un polinomio o una serie trigonométrica.

Al igual que se hace en el Álgebra Lineal, en la teoría de funciones se pueden introducir espacios de funciones, productos escalares (y, por tanto, distancias y relaciones de ortogonalidad),

bases de dichos espacios, etc. En este contexto, dadas dos funciones $f, g : [a,b] \to \mathbb{R}$, usaremos el producto escalar

$$< f, g >= \int_a^b f(x)g(x)dx.$$

7.1. Aproximación mediante polinomios de Legendre

Comencemos con un ejemplo. Consideremos el espacio de los polinomios de grado menor o igual que dos definidos en el intervalo $[-1,1]$, es decir

$$\mathcal{P}_2 = \{p(x) = a_0 + a_1 x + a_2 x^2 : a_0, a_1, a_2 \in \mathbb{R}, \quad x \in [-1,1]\}.$$

Obviamente, cualquiera de estos polinomios puede escribirse mediante una combinación lineal única de los polinomios

$$p_0(x) = 1, \quad p_1(x) = x, \quad p_2(x) = x^2.$$

En efecto, no hay más que poner $p(x) = a_0 p_0(x) + a_1 p_1(x) + a_2 p_2(x)$ para cualesquiera que sean a_0, a_1 y a_2. Por ello,

$$\mathcal{B}_2 = \{p_0(x), p_1(x), p_2(x)\}$$

es una base de $\mathcal{P}_2$. Como en Álgebra Lineal, cuando se utilizan bases ortogonales, nos gustaría escribir una descomposición del tipo

$$p(x) = \frac{< p, p_0 >}{< p_0, p_0 >} p_0(x) + \frac{< p, p_1 >}{< p_1, p_1 >} p_1(x) + \frac{< p, p_2 >}{< p_2, p_2 >} p_2(x),$$

que, de momento, no es posible ya que la base $\mathcal{B}_2$ no es ortogonal. En efecto, por ejemplo

$$< p_0, p_2 >= \int_{-1}^{1} x^2 dx = \frac{2}{3} \neq 0.$$

Sin embargo, podemos ortogonalizar[3] la base $\mathcal{B}_2$, obteniendo en nuestro ejemplo

$$\{p_0(x) = 1, p_1(x) = x, p_2(x) = \frac{3x^2 - 1}{2}\},$$

de modo que, ahora, la descomposición (7.1) se satisface. Comprobémoslo. Por una parte,

$$\begin{aligned}
< p, p_0 >&= \int_{-1}^{1} (a_0 + a_1 x + a_2 x^2) dx = 2a_0 + \frac{2a_2}{3}, \\
< p, p_1 >&= \int_{-1}^{1} (a_0 + a_1 x + a_2 x^2) x dx = \frac{2a_1}{3}, \\
< p, p_2 >&= \int_{-1}^{1} (a_0 + a_1 x + a_2 x^2) \frac{3x^2 - 1}{2} dx = \frac{8a_2}{30}.
\end{aligned}$$

Por otra parte, es fácil ver que

$$< p_0, p_0 >= 2, \quad < p_1, p_1 >= \frac{2}{3}, \quad < p_2, p_2 >= \frac{2}{5},$$

[3] Siempre se puede ortogonalizar una base mediante el procedimiento de Gram-Schmidt.

y, por tanto,

$$\frac{< p, p_0 >}{< p_0, p_0 >} p_0(x) + \frac{< p, p_1 >}{< p_1, p_1 >} p_1(x) + \frac{< p, p_2 >}{< p_2, p_2 >} p_2(x) = a_0 + \frac{a_2}{3} + a_1 x + \frac{2a_2}{3} \frac{3x^2 - 1}{2} = p(x).$$

Los polinomios ortogonales de la base dada en (7.1) son los llamados *polinomios de Legendre* de orden dos. En general, los polinomios de Legendre de grado n se definen por la fórmula

$$L_n(x) = (-1)^n \frac{1}{n! 2^n} \frac{d^n}{dx^n} (x^2 - 1)^n, \quad n = 1, 2, \ldots,$$

con $L_0(x) = 1$, y satisfacen

$$< L_n, L_n >= \int_{-1}^{1} L_n(x)^2 dx = \frac{2}{2n+1}.$$

Además, pueden ser obtenidos recursivamente mediante la fórmula

$$L_{n+1}(x) = \frac{2n+1}{n+1} x L_n(x) - \frac{n}{n+1} L_{n-1}(x), \quad n = 1, 2, \ldots,$$

con $L_0(x) = 1$ y $L_1(x) = x$.

Resumiendo, cualquier polinomio, $p(x)$, de grado menor o igual que n definido en el intervalo $[-1, 1]$ puede descomponerse (de forma exacta) en términos de la base

$$\mathcal{L}_n = \{L_0(x), L_1(x), \ldots, L_n(x)\}$$

mediante la fórmula

$$p(x) = \sum_{j=0}^{n} \frac{< p, L_j >}{< L_j, L_j >} L_j(x)$$

Análogamente, cualquier función $f : [-1, 1] \to \mathbb{R}$ puede *aproximarse* en términos de los polinomios de Legendre, mediante la expresión

$$f(x) \approx \tilde{f}(x) = \sum_{j=0}^{n} \frac{< f, L_j >}{< L_j, L_j >} L_j(x),$$

donde $\tilde{f}(x)$ es el polinomio que aproxima a $f(x)$.

De hecho, si la función f satisface ciertas propiedades de regularidad, la serie polinómica infinita es una representación alternativa de dicha función, es decir

$$f(x) = \lim_{n \to \infty} \sum_{j=0}^{n} \frac{< f, L_j >}{< L_j, L_j >} L_j(x).$$

Finalmente, observemos que si la función a aproximar está definida en un intervalo distinto de $[-1, 1]$, siempre podemos introducir un cambio de variable para situarla en dicho intervalo. En efecto, si $f : [a, b] \to \mathbb{R}$, y $x \in [a, b]$, podemos introdcir el cambio

$$t = -1 + 2\frac{x-a}{b-a} \to x = a + \frac{b-a}{2}(t+1),$$

de modo que la nueva función $g(t) = f(a + \frac{b-a}{2}(t+1))$ está definida en $[-1, 1]$. Ahora, si la aproximación por polinomios de Legendre viene dada por $\tilde{g}(t)$, entonces la de f vendrá dada por $\tilde{f}(x) = \tilde{g}(-1 + 2\frac{x-a}{b-a})$.

Ejemplo 3.6 Consideremos la función exponencial, $f(x) = e^x$ y aproximémosla por polinomios de Legendre de hasta grado dos. Tenemos

$$\begin{aligned} < f, L_0 >&= \int_{-1}^{1} e^x dx = e - \frac{1}{e}, \\ < f, L_1 >&= \int_{-1}^{1} e^x x dx = \frac{2}{e}, \\ < f, L_2 >&= \int_{-1}^{1} e^x \frac{3x^2-1}{2} dx = e - \frac{7}{e}. \end{aligned}$$

Entonces

$$\begin{aligned} e^x \approx \frac{e-\frac{1}{e}}{2} L_0(x) + \frac{3}{e} L_1(x) + \left(e - \frac{7}{e}\right)\frac{5}{2} L_2(x) &= \frac{e^2-1}{2e} + \frac{3}{e}x + \frac{5(e^2-7)}{2e}\frac{3x^2-1}{2} \\ &= \frac{33-3e^2}{4e} + \frac{3}{e}x + \frac{15(e^2-7)}{4e}x^2. \end{aligned}$$

□

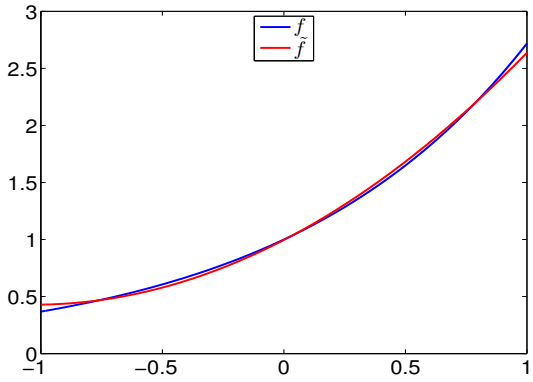


Figura 3.6:

7.2. Aproximación mediante series de Fourier

La idea de la sección anterior de representar o aproximar una función dada en términos de una combinación lineal de funciones más simples no está limitada a que estas funciones sean polinómicas. El ejemplo más importante de funciones no polinómicas que forman una base respecto la cual expresar otras funciones es el de las funciones trogonométricas.

La base de Fourier de funciones definidas en el intervalo $[0, 2\pi]$ viene dada por

$$\mathcal{F} = \{1, \text{sen}(x), \cos(x), \text{sen}(2x), \cos(2x), \ldots, \text{sen}(nx), \cos(nx), \ldots\},$$

que puede escribirse, usando la notación exponencial, como

$$\mathcal{F} = \{e^{inx}\}_{n=-\infty}^{n=\infty}.$$

Es fácil ver que esta base es ortogonal respecto el producto escalar

$$< f, g >= \int_0^{2\pi} f(x)\bar{g}(x)dx,$$

donde $\bar{g}(x)$ denota el conjugado complejo de $g(x)$[4]. En efecto, introduzcamos la notación $\phi_n(x) = e^{inx}$ y calculemos el producto escalar de dos elementos de la base distintos ($n \neq m$)

$$\begin{aligned} <\phi_n, \phi_m> &= \int_0^{2\pi} e^{inx} e^{-imx} dx = \int_0^{2\pi} e^{i(n-m)x} dx = \frac{1}{i(n-m)} \, e^{i(n-m)x} \Big|_0^{2\pi} \\ &= \frac{1}{i(n-m)} \big(\cos((n-m)2\pi) + i \operatorname{sen}((n-m)2\pi) - \cos(0) + i \operatorname{sen}(0) \big) \\ &= \frac{1}{i(n-m)} (1-1) = 0. \end{aligned}$$

Por otra parte, si $n = m$, tenemos

$$<\phi_n, \phi_n> = \int_0^{2\pi} e^{inx} e^{-inx} dx = \int_0^{2\pi} 1 dx = 2\pi.$$

Por tanto, dada una función periódica de periodo[5] 2π, $f : [0, 2\pi] \to \mathbb{R}$, podemos considerar una expresión análoga a (7.1) para los primeros $2M+1$ elementos de la base $\mathcal{F}$,

$$\tilde{f}(x) = \frac{1}{2\pi} \sum_{n=-M}^{M} <f, \phi_n> \phi_n(x),$$

donde hemos usado que $<\phi_n, \phi_n> = 2\pi$. Como ocurría para los polinomios de Legendre, la función f puede representarse como la serie infinita

$$f(x) = \frac{1}{2\pi} \lim_{M \to \infty} \sum_{n=-M}^{M} <f, \phi_n> \phi_n(x),$$

que es la llamada *serie de Fourier* de la función f. A los coeficientes

$$\hat{f}_n = \frac{1}{2\pi} <f, \phi_n> = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-inx} dx$$

se les denomina *coeficientes de Fourier* de f, de modo que la expresión de la serie queda como

$$f(x) = \sum_{n=-\infty}^{\infty} \hat{f}_n e^{inx}.$$

Usando las identidades trigonométricas, también es usual expresar la serie en términos de senos y cosenos

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n cos(nx) + b_n \operatorname{sen}(nx),$$

donde $a_n = \hat{f}_n + \hat{f}_{-n}$, $b_n = i(\hat{f}_n - \hat{f}_{-n})$, y $a_0 = \frac{1}{\pi} \int_0^{2\pi} f(x) dx$.

Ejemplo 3.7 Volvamos a la situación del Ejemplo 3.4 (véase la Figura 3.4) y apliquemos la aproximación de Fourier, en vez de la interpolación trigonométrica, como hicimos en dicho ejemplo. Tenemos, para $f(x) = x(x - 2\pi)$

$$\begin{aligned} \hat{f}_{-1} &= \frac{1}{2\pi} \int_0^{2\pi} x(x-2\pi) e^{-ix} dx = 2, \\ \hat{f}_0 &= \frac{1}{2\pi} \int_0^{2\pi} x(x-2\pi) dx = -\frac{2\pi^2}{3}, \\ \hat{f}_1 &= \frac{1}{2\pi} \int_0^{2\pi} x(x-2\pi) e^{ix} dx = 2, \end{aligned}$$

[4] Recordemos que si $z = a + bi$, entonces $\bar{z} = a - bi$, y si $z = e^{ai}$ entonces $\bar{z} = e^{-ai}$.

[5] Si el periodo de la función es otro, por ejemplo T, el cambio de variable $x = 2\pi t/T$ la convierte en 2π-periódica.

de modo que

$$\tilde{f}(x) = 2(e^{-ix} + e^{ix}) - \frac{2\pi^2}{3} = 4\cos(x) - \frac{2\pi^2}{3}.$$

□

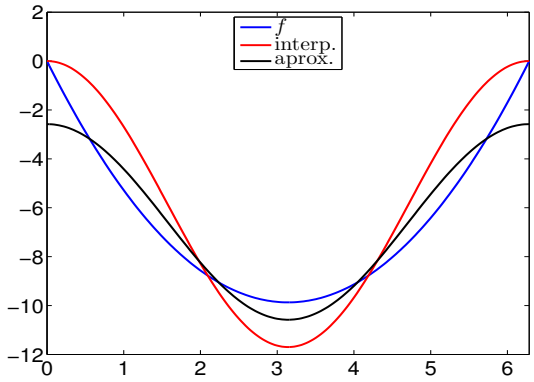


Figura 3.7: La función f, su interpolada trigonométrica y su aproximada por serie de Fourier.

Capítulo 4

Diferenciación e integración numéricas

En este capítulo introducimos métodos para la aproximación numérica de derivadas e integrales de funciones. Con respecto a la integración, es bien sabido que, en general, no siempre es posible hallar una primitiva en forma explícita de una función dada. De hecho, hay funciones para las que aunque se pueda calcular su prmitiva, esta tiene una expresión funcional demasiado complicada como para que su manejo sea práctico.

Otras situación frecuente es en la que la función que queremos integrar o derivar solo se conoce en un conjunto de puntos -no en todo un intervalo-, por ejemplo, cuando representa los resultados de una medida experimental.

En ambas situaciones es necesario considerar métodos numéricos para obtener un valor aproximado de la cantidad de interés, independientemente de lo difícil que sea la función a integrar o derivar.

1. Derivación numérica

Para una función $f : (a,b) \subset \mathbb{R} \to \mathbb{R}$ diferenciable con continuidad en un punto $x \in (a,b)$, su derivada puede ser calculada mediante los límites laterales

$$f'(x) = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h} = \lim_{h\to 0} \frac{f(x)-f(x-h)}{h},$$

siendo $h > 0$. Estas expresiones dan lugar a las aproximaciones más simples de la derivada: la aproximación mediante *diferencias finitas progresivas*, dada por

$$(\delta_+ f)(x) = \frac{f(x+h)-f(x)}{h},$$

y la aproximación mediante *diferencias finitas regresivas*, dada por

$$(\delta_- f)(x) = \frac{f(x)-f(x-h)}{h},$$

siendo $h > 0$ un número pequeño en ambas expresiones.

Para estimar el error cometido, basta desarrollar f en serie de Taylor. Si $f \in C^2(a,b)$ entonces

$$f(x+h) = f(x) + f'(x)h + \frac{f''(\xi)}{2}h^2,$$

donde $\xi \in (x, x+h)$. Tenemos entonces que

$$|(\delta_+ f)(x) - f'(x)| \leq ch,$$

para cierta constante $c > 0$ independiente de h, y por tanto la aproximación en diferencias finitas progresivas es de primer orden. Un razonamiento análogo nos permite deducir el mismo orden de aproximación para la aproximación en diferencias finitas regresivas.

Sin embargo, es posible deducir una aproximación de segundo orden de la derivada con el mismo coste computacional que las anteriores. En efecto, consideremos la aproximación mediante *diferencias finitas centradas*, dada por

$$(\delta f)(x) = \frac{f(x+h) - f(x-h)}{2h}.$$

El desarrollo de Taylor de orden tres nos proporciona las identidades

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(\xi_+)}{6}h^3, \tag{4.1}$$

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2}h^2 - \frac{f'''(\xi_-)}{6}h^3, \tag{4.2}$$

donde $\xi_+ \in (x, x+h)$ y $\xi_- \in (x-h, x)$. Restando ambas expresiones obtenemos, después de algunas manipulaciones

$$(\delta f)(x) - f'(x) = \frac{f'''(\xi_+) + f'''(\xi_-)}{12}h^2,$$

de donde se deduce

$$|(\delta f)(x) - f'(x)| \leq ch^2,$$

para cierta constante $c > 0$ independiente de h.

Normalmente, la derivación numérica se implementa sobre la malla uniforme de un intervalo, es decir, para $x_i = a + ih$, con $h = (b-a)/n$ e i recorriendo los índices $i = 0, \ldots, n$. En este caso, y para todos los esquemas de derivación numérica anteriores, aparece el *problema del borde*, debido a que las diferencias finitas no se pueden implementar en uno o ambos extremos del intervalo. En efecto, las diferencias progresivas no pueden ser evaluadas en x_n, debido a que necesitamos un nodo "x_{n+1}" que, en general, no está a nuestra disposición. Algo similar sucede con las diferencias regresivas en el nodo x_0, y con las centradas en x_0 y x_n.

Para resolver este problema recurrimos a la interpolación. Por ejemplo, para las diferencias centradas, que son de segundo orden de aproximación, consideramos el polinomio interpolatorio de Lagrange de grado 2 sobre los puntos x_0, x_1, x_2, (véase la fórmula de Newton (3.6) en el Capítulo 3)

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

Derivando y evaluando en $x = x_0$ obtenemos

$$f'(x_0) \approx p'(x_0) = f[x_0, x_1] + f[x_0, x_1, x_2](x_0 - x_1).$$

Teniendo en cuenta que la malla es uniforme y reemplazando las expresiones en diferencias divididas, deducimos

$$\begin{aligned} f'(x_0) &\approx \frac{f(x_1) - f(x_0)}{h} - \frac{f(x_2) - 2f(x_1) + f(x_0)}{2h} \\ &= \frac{1}{2h}\big(-3f(x_0) + 4f(x_1) - f(x_2)\big). \end{aligned}$$

Un razonamiento análogo nos proporciona la fórmula

$$f'(x_n) \approx \frac{1}{2h}\big(3f(x_n) - 4f(x_{n-1}) + f(x_{n-2})\big).$$

1.1. Derivadas de orden superior

Para calcular la segunda derivada o derivadas de orden superior, podemos encadenar los esquemas anteriores sucesivamente. Por ejemplo, la segunda derivada suele aproximarse mediante la fórmula

$$f''(x) \approx \big(\delta_+(\delta_- f)\big)(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}.$$

Para estimar el error de la aproximación, consideramos de nuevo los desarrollos de Taylor dados en (4.1), pero esta vez los sumamos, obteniendo

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{f'''(\xi_+) - f'''(\xi_-)}{6}h,$$

de donde

$$|\big(\delta_+(\delta_- f)\big)(x) - f''(x)| \leq ch,$$

es decir, la aproximación es lineal.

1.2. Derivación numérica de funciones de varias variables

El anterior procedimiento para la aproximación de funciones de una variable puede extenderse sin dificultad a funciones de varias variables. Sea $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ una función diferenciable con continuidad, y denotemos por (x,y) a un punto de Ω. Las derivadas parciales de f vienen dadas por

$$\frac{\partial f}{\partial x}(x,y) = \lim_{h\to 0} \frac{f(x+h,y) - f(x,y)}{h},$$
$$\frac{\partial f}{\partial y}(x,y) = \lim_{h\to 0} \frac{f(x,y+h) - f(x,y)}{h},$$

a las cuales podemos aplicar cualquiera de los esquemas en diferencias finitas vistos en la sección anterior.

A partir de las derivadas parciales definimos el *gradiente* de f

$$\nabla f(x,y) = \Big(\frac{\partial f}{\partial x}(x,y), \frac{\partial f}{\partial y}(x,y)\Big),$$

que nos proporciona las direcciones de mayor crecimiento y decrecimiento de la función f.

Para un campo vectorial $\mathbf{F} = (F_1, F_2) : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^2$, se define la *divergencia* de $\mathbf{F}$ como

$$\operatorname{div}\mathbf{F}(x,y) = \frac{\partial F_1}{\partial x}(x,y) + \frac{\partial F_2}{\partial y}(x,y),$$

la cual mide la diferencia entre el flujo saliente y el flujo entrante de un campo vectorial sobre la superficie que rodea a un volumen de control. Por tanto, si el campo tiene *fuentes* la divergencia será positiva y si tiene *sumideros* la divergencia será negativa.

Finalmente, mediante la composición del gradiente con la divergencia obtenemos un operador diferencial de segundo orden -ya que tiene dericadas segundas-, el *laplaciano*, dado por

$$\Delta f(x,y) = \operatorname{div} \nabla f(x,y) = \frac{\partial^2 f}{\partial x^2}(x,y) + \frac{\partial^2 f}{\partial y^2}(x,y).$$

Veamos con un ejemplo cómo se pueden calcular las aproximaciones numéricas de estos operadores diferenciales. Sea $\Omega = (a,b) \times (c,d)$, e introduzcamos las mallas de los intervalo (a,b) y (c,d) dadas por, respectivamente,

$$\begin{aligned} x_i &= a + ih, \quad \text{con } h = \frac{b-a}{n}, \quad i = 0,\dots,n \\ y_j &= c + jh, \quad \text{con } h = \frac{d-c}{m}, \quad j = 0,\dots,m. \end{aligned}$$

Observemos que, por comodidad, hemos asumido $(b-a)/n = (d-c)/m$. En general, el paso de las mallas, denotados por h_x y h_y, puede ser distinto.

A partir de estas mallas unidimensionales podemos construir una malla bidimensional del rectángulo Ω, dada simplemente por los puntos (x_i, y_j), $i = 0,\dots,n$, $j = 0,\dots,m$.

A partir de esta malla bidimensional podemos aproximar, por ejemplo, con diferencias progresivas

$$\begin{aligned} \nabla f(x_i,y_j) &\approx \frac{1}{h}\big(f(x_{i+1},y_j) - f(x_i,y_j), f(x_i,y_{j+1}) - f(x_i,y_j)\big), \\ \operatorname{div} \mathbf{F}(x_i,y_j) &\approx \frac{1}{h}\big(F_1(x_{i+1},y_j) - F_1(x_i,y_j) + F_2(x_i,y_{j+1}) - F_2(x_i,y_j)\big), \end{aligned}$$

para $i = 0,\dots,n-1$, $j = 0,\dots,m-1$ (observemos el problema en los bordes *superiores*). Una combinación de diferencias progresivas y regresivas nos conduce a

$$\Delta f(x_i,y_j) = \frac{1}{h^2}\big(f(x_{i+1},y_j) + f(x_{i-1},y_j) + f(x_i,y_{j+1}) + f(x_i,y_{j-1}) - 4f(x_i,y_j)\big),$$

para $i = 1,\dots,n-1$, $j = 1,\dots,m-1$ (observemos el problema en todos los bordes).

La deducción del error cometido con estas aproximaciones nos la proporciona, nuevemente, el desarrollo de Taylor, como se verá en los ejercicios.

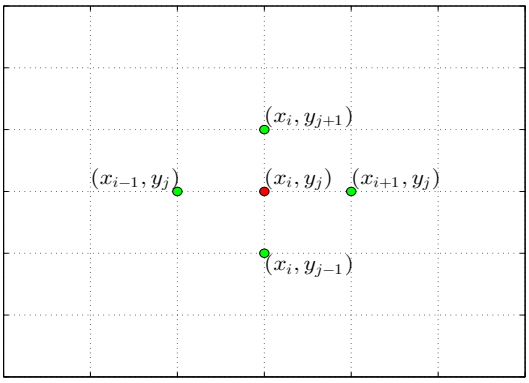


Figura 4.1: Nodos involucrados en la discretización del laplaciano

2. Integración numérica

En esta sección vamos a introducir algunas de las fórmulas clásicas de integración de funciones continuas $f:(a,b)\to\mathbb{R}$. Para abreviar la notación, escribiremos

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

Las fórmulas de integración para aproximar $I(f)$ se llaman *simples* si la aproximación se produce usando *todo* el intervalo (a,b), y *compuestas* si, previamente a la aplicación de la fórmula, se subdivide el intervalo (a,b) en un número n dado de subintervalos,

$$I_i=[x_i,x_{i+1}],\quad \text{con } i=0,\dots,n-1,$$

donde

$$x_i=a+ih,\quad i=0,\dots,n,\quad h=\frac{b-a}{n},$$

se utiliza que

$$I(f)=\sum_{i=0}^{n-1}\int_{I_i} f(x)dx,$$

y, entonces, se usa la fórmula de aproximación sobre cada subintervalo.

Para medir el error de las fórmulas de integración utilizaremos dos criterios. Si la regla es simple, diremos que tiene un *grado de exactitud r* si para cualquier polinomio de grado r, $p_r(x)$, la fórmula proporciona el resultado exacto de $I(p_r)$.

Si la regla es compuesta, usaremos el criterio habitual del orden de aproximación respecto el tamaño de los subintervalos.

2.1. Fórmula del punto medio

La fórmula del punto medio es la más sencilla de todas. Consiste en aproximar la función f en (a,b) por el valor que toma en el punto medio de dicho intervalo, es decir,

$$I_{pm}(f)=(b-a)f\Big(\frac{a+b}{2}\Big),$$

donde *pm* significa *punto medio*.

Para obtener una estimación del error usamos el desarrollo de Taylor. Suponiendo que f es derivable con continuidad en (a,b) tenemos

$$f(x)=f\Big(\frac{a+b}{2}\Big)+f'\Big(\frac{a+b}{2}\Big)\Big(x-\frac{a+b}{2}\Big)+\frac{f''(\xi)}{2}\Big(x-\frac{a+b}{2}\Big)^2,$$

con $\xi\in(a,b)$. Entonces

$$\begin{aligned}I(f)&=I_{pm}(f)+f'\Big(\frac{a+b}{2}\Big)\int_a^b\Big(x-\frac{a+b}{2}\Big)dx+\frac{f''(\xi)}{2}\int_a^b\Big(x-\frac{a+b}{2}\Big)^2dx\\&=I_{pm}(f)+\frac{f''(\xi)}{24}(b-a)^3.\end{aligned}\tag{4.3}$$

De modo que, como la estimación depende de la segunda derivada de f, deducimos que la fórmula tiene un grado de exactitud $r = 1$.

La correspondiente fórmula compuesta del punto medio se obtiene fácilmente

$$I_{pm}^{c}(f) = h\sum_{i=0}^{n-1} f\Big(\frac{x_i + x_{i+1}}{2}\Big),$$

donde c significa *compuesta*. Usando un argumento similar al de (4.3) deducimos

$$I(f) - I_{pm}^{c}(f) = (b-a)\frac{f''(\xi)}{24}h^2,$$

donde $\xi \in (a,b)$, luego la fórmula es de orden de aproximación cuadrático.

2.2. Fórmula del trapecio

La obtenemos aproximando la función f en (a,b) por su polinomio interpolatorio de orden 1, es decir,

$$I_t(f) = \int_a^b \Big(f(a) + \frac{f(b)-f(a)}{b-a}(x-a)\Big)dx = \frac{b-a}{2}\big(f(a)+f(b)\big).$$

El error inducido viene dado por

$$I(f) - I_t(f) = -\frac{(b-a)^3}{12}f''(\xi),$$

donde $\xi \in (a,b)$. Deducimos entonces que la fórmula del trapecio tiene grado de exactitud $r = 1$, como en el caso de la regla del punto medio.

La correspondiente fórmula compuesta viene dada por

$$I_t^c(f) = \frac{h}{2}\sum_{i=0}^{n-1}\big(f(x_i) + f(x_{i+1})\big),$$

y como en el caso de la fórmula del punto medio, su orden de aproximación es cuadrático.

2.3. Fórmula de Simpson

La obtenemos aproximando la función f en (a,b) por su polinomio interpolatorio de orden 2. La fórmula resultante es

$$I_s(f) = \frac{b-a}{6}\big(f(a) + 4f\Big(\frac{a+b}{2}\Big) + f(b)\big).$$

El error inducido viene dado por

$$I(f) - I_t(f) = -\frac{1}{16}\frac{(b-a)^5}{180}f^{(4)}(\xi),$$

donde $\xi \in (a,b)$. Por consiguiente, la fórmula de Simpson tiene grado de exactitud $r = 3$. La correspondiente fórmula compuesta viene dada por

$$I_t^c(f) = \frac{h}{6}\sum_{i=0}^{n-1}\big(f(x_i) + 4f\Big(\frac{x_i + x_{i+1}}{2}\Big) + f(x_{i+1})\big),$$

y puede usarse el desarrollo de Taylor para comprobar que proporciona una aproximación de cuarto orden.

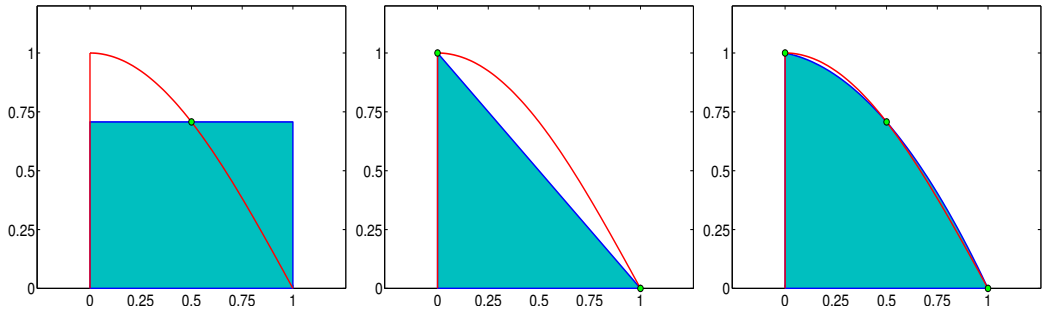


Figura 4.2: Fórmulas de integración del punto medio (izquierda), del trapecio (centro) y de Simpson (derecha).

2.4. Fórmulas de orden superior

Las fórmulas de integración numérica que hemos introducido hasta ahora para aproximar $I(f)$ utilizan polinomios interpolatorios de Lagrange de distinto orden para aproximar la función, y después integran dicho polinomio de forma exacta.

En general, podemos definir la aproximación

$$I_{apr}(f) = \int_a^b \Pi_n f(x)dx,$$

donde $\Pi_n f$ es el polinomio de interpolación de Lagrange de f de grado n en los nodos de una malla dada, x_i, $i = 0,\ldots,n-1$. Calculando esta integral obtenemos

$$I_{apr}(f) = \sum_{i=0}^{n} \alpha_i f(x_i),$$

donde

$$\alpha_i = \int_a^b \ell_i(x)dx, \quad i = 0,\ldots,n,$$

siendo ℓ_i es el $i-$ésimo polinomio fundamental de Lagrange de grado n, tal y como fue introducido en la fórmula (3.3) del Capítulo 3. De esta manera, la aproximación obtenida tendrá un grado de exactitud de, al menos, $r = n$.

2.5. Fórmula de Gauss

Inspirados por la expresión

$$I_{apr}(f) = \sum_{i=0}^{n} \alpha_i f(x_i), \tag{4.4}$$

podemos preguntarnos si existen elecciones de los *pesos* α_i y de los nodos x_i que nos proporcionen un grado de exactitud superior al dado por los polinomios de Lagrange.

Para simplificar la exposición nos restringiremos al intervalo de referencia $(-1,1)$, teniendo en cuenta que, una vez deducidos los nodos $\bar{x}_i$ y los pesos $\bar{\alpha}_i$ relativos a este intervalo, los podremos llevar al intervalo (a,b) mediante el cambio de variables

$$x_i = \frac{a+b}{2} + \frac{b-a}{2}\bar{x}_i, \quad \alpha_i = \frac{b-a}{2}\bar{\alpha}_i.$$

n	$\{\bar{x}_i\}$	$\{\bar{\alpha}_i\}$
1	$\{\pm 1/\sqrt{3}\}$	$\{1\}$
2	$\{\pm\sqrt{15}/5, 0\}$	$\{5/9, 8/9\}$
3	$\left\{\pm(1/35)\sqrt{525-70\sqrt{30}},\right.$ $\left.\pm(1/35)\sqrt{525+70\sqrt{30}}\right\}$	$\left\{(1/36)(18+\sqrt{30},\right.$ $\left.(1/36)(18-\sqrt{30}\right\}$
4	$\left\{0, \pm(1/21)\sqrt{245-14\sqrt{70}},\right.$ $\left.\pm(1/21)\sqrt{245+14\sqrt{70}}\right\}$	$\left\{128/225, (1/900)(322+13\sqrt{70},\right.$ $\left.(1/900)(322-13\sqrt{70}\right\}$

Cuadro 4.1: Nodos y pesos de la cuadratura de Gauss para los primeros valores de n.

La respuesta a la cuestión nos la proporcionan los polinomios de Legendre de grado hasta $n+1$, introducido en la Subsección 7.1 del Capítulo 3.

Puede demostrarse que el máximo grado de exactitud de la aproximación (4.4) es $r = 2n+1$, y que se obtiene mediante la *fórmula de Gauss*, cuyos nodos y pesos se determinan de la siguiente manera:

$$\begin{cases} \bar{x}_i = \text{ceros de } L_{n+1}(x), \\ \\ \bar{\alpha}_i = \dfrac{2}{(1-\bar{x}_i^2)\big(L'_{n+1}(\bar{x}_i)\big)^2}, \quad i = 0, \ldots, n. \end{cases}$$

Los pesos son todos positivos y los nodos pertenecen al intervalo $(-1,1)$. En la Tabla 2.5 recogemos los nudos y pesos de las fórmulas de cuadratura de Gauss con $n = 1,2,3,4$.

Si f es derivable con continuidad $2n+2$ veces, entonces el error de la aproximación viene dado por

$$I(f) - I_g(f) = \frac{2^{2n+3}((n+1)!)^4}{(2n+3)((2n+2)!)^3} f^{(2n+2)}(\xi),$$

donde $\xi \in (-1,1)$.

Capítulo 5

Sistemas de ecuaciones lineales

1. Introducción

Queremos resolver sistemas de ecuaciones lineales compatibles con el mismo número de ecuaciones que de incógnitas. Es decir, dados los números a_{ij} y b_j para $i, j = 1, 2, \ldots, n$ se trata de hallar los números $x_1, x_2, \ldots, x_n$ que verifican las n ecuaciones lineales

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n & = & b_2 \\ \vdots & & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n & = & b_n \end{array}$$

Donde $A = (a_{ij})_{i,j=1}^{n}$ es la matriz de coeficientes, $\mathbf{b} = (b_i)_{i=1}^{n}$ el vector del segundo miembro y $\mathbf{x} = (x_i)_{i=1}^{n}$ es el vector de incógnitas.

Usando esta notación matricial, el sistema se escribe

$$A\mathbf{x} = \mathbf{b}$$

Los métodos numéricos de resolución de sistemas lineales se pueden clasificar en directos e iterativos.

En los **métodos directos** la solución se calcula en un número finito de pasos conocidos *a priori.*

- Si usáramos aritmética con precisión infinita la solución calculada sería exacta, pero como, habitualmente, se usa artimética de precisión finita están sujetos a errores de redondeo.
- Son adecuados para resolver sistemas pequeños y sistemas grandes con matriz llena.

Como ejemplos de estos métodos veremos los métodos de **Gauss, Gauss-Jordan y Descomposición LU.**

Los **métodos iterativos** construyen una sucesión que converge a la solución del sistema. En este caso, además de los errores de redondeo, exite un error de truncamiento (debido a que realizamos un número finito de iteraciones). Como ejemplo de estos métodos veremos los métodos de **Jacobi y Gauss-Seidel.**

2. Métodos directos

2.1. Gauss

Gauss es transforma el sistema de ecuaciones lineales original en un sistema de matriz triangular superior equivalente (con las mismas soluciones).

En estos cálculos sólo intervienen la matriz de coeficientes y el vector del segundo miembro por lo que en lugar de operar con el sistema completo podemos operar con la matriz extendida:

$$[A|b] = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{pmatrix}$$

Triangularización

El sistema equivalente se obtiene realizando operaciones por filas

$$f_i \to f_i + \lambda f_j, \quad j \neq i.$$

Es decir, dada una fila f_i, obtenemos un sistema equivalente si le sumamos cualquier otra fila multiplicada por un real.

Si usamos la estrategia del pivote, intercambiaremos filas

$$f_i \leftrightarrow f_j$$

Una vez hemos triangularizado la matriz tenemos un sistema del tipo

$$U\mathbf{x} = \mathbf{b}'$$

donde U es una matriz triangular superior:

$$[U|b] = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} & b'_1 \\ 0 & u_{22} & u_{23} & \dots & u_{2n} & b'_2 \\ 0 & 0 & u_{33} & \dots & u_{3n} & b'_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & u_{nn} & b'_n \end{pmatrix}$$

Sustitución regresiva

La ecuación i-ésima del sistema, para $i = 1, 2, \dots, n$ es:

$$u_{ii}x_i + u_{ii+1}x_{i+1} + \cdots + u_{in}x_n = b'_i$$

Por lo tanto, para $i = n, n-1, \dots, 1$, si $u_{ii} \neq 0$ se tiene que

$$x_i = \frac{b'_i - u_{ii+1}x_{i+1} - \cdots - u_{in}x_n}{u_{ii}} = \frac{1}{u_{ii}}\left(b'_i - \sum_{j=i+1}^{n} u_{ij}x_j\right)$$

Ejemplo 5.1 Resolver, utilizando el método de Gauss, el sistema lineal:

$$\begin{array}{rrrcl} 2x & +3y & -z & = & 5 \\ 4x & +4y & -3z & = & 3 \\ -2x & +3y & -z & = & 1 \end{array}$$

Primero, triangularizamos la matriz extendida. Empezamos haciendo ceros en la primera columna por debajos del pivote **2**.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \left(\begin{array}{cccc} \mathbf{2} & 3 & -1 & 5 \\ \mathbf{4} & 4 & -3 & 3 \\ \mathbf{-2} & 3 & -1 & 1 \end{array} \right) \begin{array}{rcl} f_1' & = & f_1 \\ f_2' & = & f_2 - \frac{\mathbf{4}}{\mathbf{2}} f_1 \\ f_3' & = & f_3 - \frac{\mathbf{-2}}{\mathbf{2}} f_1 \end{array}$$

Seguimos haciendo ceros por debajo del pivote -2 en la segunda columna.

$$\begin{array}{c} f_1' \\ f_2' \\ f_3' \end{array} \left(\begin{array}{cccc} 2 & 3 & -1 & 5 \\ 0 & \mathbf{-2} & -1 & -7 \\ 0 & \mathbf{6} & -2 & 6 \end{array} \right) \begin{array}{rcl} f_1'' & = & f_1' \\ f_2'' & = & f_2' \\ f_3'' & = & f_3 - \frac{\mathbf{6}}{\mathbf{-2}} f_2' \end{array}$$

Y ya tenemos una matriz triangular superior (con ceros por debajo de la diagonal principal).

$$\begin{array}{c} f_1'' \\ f_2'' \\ f_3'' \end{array} \left(\begin{array}{cccc} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 0 & -5 & -15 \end{array} \right).$$

Una vez hemos triangularizado la matriz extendida, realizamos la sustitución regresiva, es decir, despejamos las incógnitas empezando por la ecuación de abajo y progresamos hacia arriba.

$$\begin{array}{rrrcr} 2x & +3y & -z & = & 5 \\ & -2y & -z & = & -7 \\ & & -5z & = & -15 \end{array}$$

Empezamos por la z, seguimos con la y y acabamos despejando x.

$$\begin{array}{rcl} z & = & -15/(-5) = 3 \\ y & = & (-7+z)/(-2) = (-7+3)/(-2) = 2 \\ x & = & (5-3y+z)/2 = (5-3(2)+(3))/2 = 1 \end{array}$$

□

La estrategia del pivote

En la triangularización, en la primera transformación de la matriz, obtenemos ceros por debajo de a_{11}, en el segundo paso hacemos ceros por debajo de a'_{22} y así sucesivamente. Estos elementos a_{ii} son los *pivotes*. Hay dos variantes del método de Gauss que contemplan la posibilidad de intercambiar los pivotes:

- En *Gauss con pivote parcial* se intercambian filas y se toma como pivote el elemento de mayor valor absoluto a_{ij} que está en la columna del pivote por debajo del pivote.
- En *Gauss con pivote total* se intercambian filas y columnas.

Es necesario usar Gauss con pivote, por ejemplo, cuando algún $a_{ii} = 0$ y cuando algún a_{ii} es muy pequeño. En el primer caso porque no podemos dividir por cero (como se puede observar en el ejemplo de Gauss, al realizar operaciones por filas, los pivotes siempre dividen) y en el segundo caso porque usar pivotes pequeños puede producir grandes errores de redondeo.

La estrategia del pivote es útil y necesaria en los mismos casos en el método de Gauss-Jordan y en la factorización LU.

Ejemplo 5.2 Resolver, utilizando el método de Gauss con pivote parcial, el sistema lineal:

$$\begin{array}{rrrcr} x & +y & -z & = & 0 \\ 2x & +y & +z & = & 7 \\ 3x & -2y & -z & = & -4 \end{array}$$

En la fase de triangularización, empezamos buscando el pivote en la primera columna. Seleccionamos como pivote el elemento de mayor valor absolutoy hacemos ceros por debajo del pivote.

$$\begin{pmatrix} \mathbf{1} & 1 & -1 & 0 \\ \mathbf{2} & 1 & 1 & 7 \\ \mathbf{3} & -2 & -1 & -4 \end{pmatrix} \Leftrightarrow \begin{matrix} f_1 \\ f_2 \\ f_3 \end{matrix} \begin{pmatrix} \mathbf{3} & -2 & -1 & -4 \\ \mathbf{2} & 1 & 1 & 7 \\ \mathbf{1} & 1 & -1 & 0 \end{pmatrix} \begin{array}{rcl} f_1' &=& f_1 \\ f_2' &=& f_2 - \frac{\mathbf{2}}{\mathbf{3}} f_1 \\ f_3' &=& f_3 - \frac{\mathbf{1}}{\mathbf{3}} f_1 \end{array}$$

Ahora el valor máximo entre el pivote y los elementos por debajo del pivote, máx $(7/3, 5/3)$, es el pivote $\mathbf{7/3}$, por lo que no hace falta intercambiar filas.

$$\begin{matrix} f_1' \\ f_2' \\ f_3' \end{matrix} \begin{pmatrix} 3 & -2 & -1 & -4 \\ 0 & \frac{\mathbf{7}}{\mathbf{3}} & \frac{5}{3} & \frac{29}{3} \\ 0 & \frac{\mathbf{5}}{\mathbf{3}} & -\frac{2}{3} & \frac{4}{3} \end{pmatrix} \begin{array}{rcl} f_1'' &=& f_1' \\ f_2'' &=& f_2' \\ f_3'' &=& f_3 - \frac{\mathbf{5/3}}{\mathbf{7/3}} f_2' \end{array}$$

Y ya tenemos una matriz triangular superior (con ceros por debajo de la diagonal principal)

$$\begin{matrix} f_1'' \\ f_2'' \\ f_3'' \end{matrix} \begin{pmatrix} 3 & -2 & -1 & -4 \\ 0 & \frac{7}{3} & \frac{5}{3} & \frac{29}{3} \\ 0 & 0 & -\frac{13}{7} & -\frac{39}{7} \end{pmatrix}.$$

Realizamos la sustitución regresiva despejando las incógnitas. Empezamos por la ecuación de abajo y progresamos hacia arriba.

$$\begin{array}{rrrcr} 3x & -2y & -z & = & -4 \\ & \frac{7}{3}y & +\frac{5}{3}z & = & \frac{29}{3} \\ & & -\frac{13}{7}z & = & -\frac{39}{7} \end{array}$$

Empezamos con la z

$$\begin{aligned} z &= -(39/7)/(-13/7) = 3 \\ y &= ((29/3) - (5/3)z)/(7/3) = 2 \\ x &= (-4 + 2y + z)/3 = 1 \end{aligned}$$

□

2.2. Gauss-Jordan

El fundamento del método de Gauss-Jordan es el mismo que el del método de Gauss. Pero en lugar de transformar el sistema en uno equivalente triangular, su objetivo es transformar el sistema de ecuaciones lineales original en un sistema de diagonal con las mismas soluciones.

Para ello se realizan las mismas operaciones que en el método de Gauss, es decir, operaciones con filas. Y, al igual que en Gauss, en los cálculos se utiliza la matriz de coeficientes y el vector del segundo miembro.

Ejemplo 5.3 Resolver un sistema lineal por Gauss-Jordan

$$\begin{array}{rrrcl} 2x & +3y & -z & = & 5 \\ 4x & +4y & -3z & = & 3 \\ -2x & +3y & -z & = & 1 \end{array}$$

Comenzamos escribiendo la matriz extendida y dividimos la primera fila por el pivote **2**.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \left(\begin{array}{cccc} \mathbf{2} & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{array} \right) \begin{array}{l} f_1 = f_1/\mathbf{2} \end{array}$$

A continuación hacemos ceros por debajo del pivote en la primera columna.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \left(\begin{array}{cccc} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{5}{2} \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{array} \right) \begin{array}{l} f_1 \\ f_2 = f_2 - 4f_1 \\ f_3 = f_3 - (-2)f_1 \end{array}$$

Repetimos ahora el proceso con la segunda fila. Dividimos la segunda fila por el pivote $-\mathbf{2}$.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \left(\begin{array}{cccc} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{5}{2} \\ 0 & \mathbf{-2} & -1 & -7 \\ 0 & 6 & -2 & 6 \end{array} \right) \begin{array}{l} f_2 = f_2/(\mathbf{-2}) \end{array}$$

Y hacemos ceros por encima y por debajo del pivote en la segunda columna.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \left(\begin{array}{cccc} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{5}{2} \\ 0 & 1 & \frac{1}{2} & \frac{7}{2} \\ 0 & 6 & -2 & 6 \end{array} \right) \begin{array}{l} f_1 = f_1 - (3/2)f_2 \\ f_2 \\ f_3 = f_3 - 6f_2 \end{array}$$

Y por último, repetimos las operaciones para la tercera fila. Dividimos la tercera fila por el pivote $-\mathbf{5}$.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \left(\begin{array}{cccc} 1 & 0 & -\frac{5}{4} & -\frac{11}{4} \\ 0 & 1 & \frac{1}{2} & \frac{7}{2} \\ 0 & 0 & \mathbf{-5} & -15 \end{array} \right) \begin{array}{l} f_3 = f_3/(\mathbf{-5}) \end{array}$$

Hacemos ceros por encima del pivote en la tercera columna.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \left(\begin{array}{cccc} 1 & 0 & -\frac{5}{4} & -\frac{11}{4} \\ 0 & 1 & \frac{1}{2} & \frac{7}{2} \\ 0 & 0 & 1 & 3 \end{array} \right) \begin{array}{l} f_1 = f_1 - (-5/4)f_3 \\ f_2 = f_2 - (1/2)f_3 \\ f_3 \end{array}$$

Y el sistema equivalente resultante es:

$$\left(\begin{array}{cccc} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array}\right) \begin{array}{l} x_1 = 1 \\ x_2 = 2 \\ x_3 = 3 \end{array}$$

Es decir, la solución del sistema viene dada por la columna de términos independientes.

□

Este método es particularmente adecuado para resolver simultáneamente sistemas con la misma matriz de coeficientes y términos independientes diferentes. Por ello es un método adecuado para calcular la inversa de una matriz.

Cálculo de la matriz inversa por Gauss-Jordan

La inversa de una matriz A $n \times n$, caso de existir, es una matriz $n \times n$ que llamaremos A^{-1} que verifica $AA^{-1} = I = A^{-1}A$.

Si consideramos que las columnas de A e I son

$$A = (c_1 c_2 \dots c_n) \text{ y } I = (e_1 e_2 \dots e_n)$$

como $AA^{-1} = I$

$$A\,(c_1 c_2 \dots c_n) = (e_1 e_2 \dots e_n)\,.$$

Podemos reescribirlo como

$$Ac_1 = e_1,\, Ac_2 = e_2, \dots,\, Ac_n = e_n$$

Es decir, las columnas de la matriz A^{-1} son las soluciones de n sistemas que tienen como matriz de coeficientes la matriz A y como término independiente las columnas de I. Es decir, si resolvemos simultaneamente estos n sistemas, las respectivas soluciones serán las columnas de A^{-1}.

Un método muy adecuado para resolver simultáneamente sistemas con la misma matriz de coeficientes es Gauss-Jordan.

El procedimiento tendría los siguientes pasos:

- Escribir una matriz $n \times 2n$ que consiste en la matriz dada A a la izquierda y la matriz identidad I de dimensión $n \times n$ a la derecha $[A|I]$.
- Mediante operaciones por filas, transformar la matriz A en la matriz I, de forma que obenemos a partir de $[A|I]$ obtenemos $\left[I|A^{-1}\right]$.
- Comprobar que $AA^{-1} = I = A^{-1}A$.

Ejemplo 5.4 Calcular la matriz inversa de A siendo

$$A = \left(\begin{array}{ccc} 3 & 2 & 3 \\ 2 & 1 & 1 \\ 3 & 1 & 1 \end{array}\right)$$

Comenzamos con la matriz extendida$[A|I]$y dividimos la primera fila por el pivote **3**.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array}\begin{pmatrix} \mathbf{3} & 2 & 3 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}\begin{array}{rcl} f_1 & = & f_1/\mathbf{3} \end{array}$$

A continuación, hacemos ceros por debajo del pivote en la primera columna.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array}\begin{pmatrix} 1 & \frac{2}{3} & 1 & \frac{1}{3} & 0 & 0 \\ 2 & 1 & 1 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}\begin{array}{rcl} f_1 & & \\ f_2 & = & f_2 - 2f_1 \\ f_3 & = & f_3 - 3f_1 \end{array}$$

Repetimos los dos pasos para la segunda fila. Dividimos la segunda fila por el pivote $-\frac{\mathbf{1}}{\mathbf{3}}$.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array}\begin{pmatrix} 1 & \frac{2}{3} & 1 & \frac{1}{3} & 0 & 0 \\ 0 & -\frac{\mathbf{1}}{\mathbf{3}} & -1 & -\frac{2}{3} & 1 & 0 \\ 0 & -1 & -2 & -1 & 0 & 1 \end{pmatrix}\begin{array}{rcl} f_2 & = & f_2/(-\frac{\mathbf{1}}{\mathbf{3}}) \end{array}$$

Y hacemos ceros por encima y por debajo del pivote en la segunda columna.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array}\begin{pmatrix} 1 & \frac{2}{3} & 1 & \frac{1}{3} & 0 & 0 \\ 0 & 1 & 3 & 2 & -3 & 0 \\ 0 & -1 & -2 & -1 & 0 & 1 \end{pmatrix}\begin{array}{rcl} f_1 & = & f_1 - (2/3)f_2 \\ f_2 & & \\ f_3 & = & f_3 - (-1)f_2 \end{array}$$

Repetimos los pasos para la tercera fila. Como el pivote ya es 1, hacemos ceros por encima del pivote en la tercera columna.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array}\begin{pmatrix} 1 & 0 & -1 & -1 & 2 & 0 \\ 0 & 1 & 3 & 2 & -3 & 0 \\ 0 & 0 & 1 & 1 & -3 & 1 \end{pmatrix}\begin{array}{rcl} f_1 & = & f_1 - (-1)f_3 \\ f_2 & = & f_2 - 3f_3 \\ f_3 & & \end{array}$$

Como a la izquierda ya tenemos la matriz I la matriz de la derecha será A^{-1}.

$$\left[I|A^{-1}\right] = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & -1 & 6 & -3 \\ 0 & 0 & 1 & 1 & -3 & 1 \end{pmatrix}$$

Por fin, comprobamos que es la matriz inversa $AA^{-1} = I$

$$AA^{-1} = \begin{pmatrix} 3 & 2 & 3 \\ 2 & 1 & 1 \\ 3 & 1 & 1 \end{pmatrix}\begin{pmatrix} 0 & -1 & 1 \\ -1 & 6 & -3 \\ 1 & -3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I$$

Y que $A^{-1}A = I$

$$A^{-1}A = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 6 & -3 \\ 1 & -3 & 1 \end{pmatrix}\begin{pmatrix} 3 & 2 & 3 \\ 2 & 1 & 1 \\ 3 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I$$

□

2.3. Factorización LU

En el método de factorización LU el objetivo es descomponer la matriz de coeficientes A en una matriz triangular superior U y una matriz triangular inferior L de forma que

$$A = LU$$

No todas las matrices cuadradas admiten factorización LU. Matrices que admiten factorización LU son, por ejemplo:

- Matrices A estrictamente diagonal dominantes:
  - por filas: $|a_{ii}| > \sum_{\substack{j=1 \\ j \neq 1}}^{n} |a_{ij}|$ para $i = 1, 2, \ldots, n$
  - por columnas $|a_{ii}| > \sum_{\substack{j=1 \\ j \neq 1}}^{n} |a_{ji}|$ para $i = 1, 2, \ldots, n$
- Matrices A simétricas y definidas positivas:

  $$A = A^T \quad \text{y} \quad \mathrm{x}^{\mathrm{T}} A \mathrm{x} > 0 \quad \text{para todo } \mathrm{x} \neq 0$$

  Como, caso de existir, la factorización LU de una matriz no es única, habitualmente se fija

  $$l_{ii} = 1 \quad \text{para } i = 1, 2, \ldots, n$$

  Y en este caso la matriz U, la matriz triangular superior, se obtiene al triangularizar A como en Gauss.

Consideramos el sistema de ecuaciones lineales

$$A\mathbf{x} = \mathbf{b}$$

donde la matriz A admite la factorización LU. Los pasos para resolver este sistema por factorización LU son:

1. Descomponemos la matriz A. Como $A\mathbf{x} = \mathbf{b} \Longleftrightarrow LU\mathbf{x} = \mathbf{b}$.
2. Resolvemos $L\mathbf{y} = \mathbf{b}$ utilizando el método de sustitución progresiva y obtenemos $\mathbf{y}$.
3. Resolvemos $U\mathbf{x} = \mathbf{y}$ utilizando el método de sustitución regresiva y obtenemos $\mathbf{x}$.

Sustitución progresiva

Si tenemos un sistema

$$L\mathbf{x} = \mathbf{b}$$

donde L es una matriz triangular inferior:

$$L = \begin{pmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{pmatrix}$$

La ecuación i-ésima del sistema, para $i = 1, 2, \dots, n$ es:

$$l_{i1}x_1 + l_{i2}x_2 + \cdots + l_{ii}x_i = b_i$$

Por lo tanto, para $i = 1, 2, \dots, n$, si $l_{ii} \neq 0$ entonces

$$x_i = \frac{b_i - l_{i1}x_1 - \cdots - l_{ii-1}x_{i-1}}{l_{ii}} = \frac{1}{l_{ii}}\left(b_i - \sum_{j=1}^{i-1} l_{ij}x_j\right)$$

Ejemplo 5.5 Resolver un sistema lineal por factorización LU

$$\begin{array}{rrrcr} x & +y & +z & = & 1 \\ -x & +y & & = & 0 \\ & -2y & +2z & = & -4 \end{array}$$

1. *Factorización.* En el primer paso hacemos ceros por debajo del elemento a_{11}.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \begin{pmatrix} \mathbf{1} & 1 & 1 \\ -1 & 1 & 0 \\ 0 & -2 & 2 \end{pmatrix} \quad \begin{array}{rcrcrr} f_1' & = & f_1 & & & \\ f_2' & = & f_2 & - & (-1)/1 & f_1 \\ f_3' & = & f_3 & - & 0/1 & f_1 \end{array}$$

   Los multiplicadores (en este ejemplo -1 y 0), que aparecen en rojo, son los elementos con los que construímos la matriz L. Los insertamos en la matriz, en lugar de los ceros creados. Repetimos el proceso creando ceros por debajo de a_{22}'.

$$\begin{array}{c} f_1' \\ f_2' \\ f_3' \end{array} \begin{pmatrix} 1 & 1 & 1 \\ -1 & \mathbf{2} & 1 \\ 0 & -2 & 2 \end{pmatrix} \quad \begin{array}{rcrcrr} f_1'' & = & f_1' & & & \\ f_2'' & = & f_2' & & & \\ f_3'' & = & f_3' & - & (-2/2) & f_2' \end{array}$$

   Y llegamos a la matriz que almacena simultáneamente L y U.

$$\begin{pmatrix} 1 & 1 & 1 \\ -1 & 2 & 1 \\ 0 & -1 & 3 \end{pmatrix}.$$

   Las matrices L y U son:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \quad U = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 3 \end{pmatrix}$$

2. *Sustitución progresiva.* Resolvemos el sistema triangular inferior $Ly = b$ y obtenemos y.

$$\begin{array}{rrrcr} y_1 & & & = & 1 \\ -y_1 & +y_2 & & = & 0 \\ & -y_2 & +y_3 & = & -4 \end{array} \qquad \begin{array}{rcrcrcr} y_1 & = & 1 & & & & \\ y_2 & = & y_1 & = & 1 & & \\ y_3 & = & -4 & +y_2 & = & & -3 \end{array}$$

3. *Sustitución regresiva.* Resolvemos el sistema triangular superior $Ux = y$ y obtenemos x, que era lo que buscábamos.

$$\begin{array}{rrrcr} x_1 & +x_2 & +x_3 & = & 1 \\ & 2x_2 & +x_3 & = & 1 \\ & & 3x_3 & = & -3 \end{array} \qquad \begin{array}{rcrcr} x_3 & = & -3/3 & = & -1 \\ x_2 & = & (1-x_3)/2 & = & 1 \\ x_1 & = & 1-x_2-x_3 & = & 1 \end{array}$$

□

3. Métodos iterativos

Un método iterativo construye una sucesión de vectores $\mathbf{x}_n$ tal que

$$\lim_{n\to\infty} \mathbf{x}_n = \mathbf{x}$$

Los métodos iterativos son, en general, más eficientes que los métodos directos para resolver sistemas de ecuaciones lineales grandes y de matriz hueca ya que se basan en la operación multiplicación de matriz por vector.

En general, en estos casos son ventajosos porque sólo es necesario almacenar los coeficientes no nulos de la matriz del sistema, si no se exige mucha precisión, se puede obtener una aproximación aceptable en un número pequeño de iteraciones y son menos sensibles a los errores de redondeo.

Sin embargo, en contraste con los métodos directos, En general, no es posible predecir el número de operaciones que se requieren para obtener una aproximación a la solución con una precisión determinada, el tiempo de cálculo y la precisión del resultado pueden depender de la elección de ciertos parámetros y generalmente no se gana tiempo por iteración si la matriz de coeficientes es simétrica.

Dada una aproximación inicial $\mathbf{x}^{(0)}$, un método iterativo genera una sucesión de aproximaciones $\mathbf{x}^{(k)}$, para $k = 1, 2, \ldots$, que converge a la solución del sistema.

Para generar esta sucesión, se repite el mismo esquema de operaciones hasta que se obtiene una aproximación a la solución con una precisión especificada de antemano, o se rebasa un número máximo de iteraciones.

Los métodos iterativos clásicos (lineales) se basan en reescribir el problema

$$A\mathbf{x} = \mathbf{b} \Longleftrightarrow \mathbf{x} = G\mathbf{x} + \mathbf{c}$$

donde G es una matriz $n \times n$ y $\mathbf{c}$ es un vector columna de dimensión n. $I - G$ ha de ser inversible.

Si tomamos un vector $\mathbf{x}^{(0)}$ como aproximación inicial a la solución realizamos las iteraciones $k = 1, 2, \ldots$ utilizando la fórmula recursiva

$$\mathbf{x}^{(k)} = G\mathbf{x}^{(k-1)} + \mathbf{c}$$

La matriz G se llama matriz de iteración y el vector $\mathbf{c}$ se llama vector de iteración.

3.1. Método de Jacobi

En este caso para obtener la matriz G realizamos la descomposición

$$A = L + D + U$$

donde

$$L = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{pmatrix} \quad D = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix} \quad U = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

Como

$$\begin{array}{l} A\mathbf{x} = \mathbf{b} \Rightarrow (L+D+U)\,\mathbf{x} = \mathbf{b} \Rightarrow \\ D\mathbf{x} = -(L+U)\,\mathbf{x} + \mathbf{b} \Rightarrow \mathbf{x} = -D^{-1}(L+U)\,\mathbf{x} + D^{-1}\mathbf{b} \end{array}$$

Tenemos

$$\mathbf{x}^{(k+1)} = -D^{-1}(L+U)\,\mathbf{x}^{(k)} + D^{-1}\mathbf{b}$$

y la matriz y el vector de iteración serían

$$\begin{array}{rl} G = & -D^{-1}(L+U) \\ \mathbf{c} = & D^{-1}\mathbf{b} \end{array}$$

Si elegimos una aproximación inicial $\mathbf{x}^{(0)}$ realizaríamos una serie de iteraciones hasta que se cumpliera un criterio de parada. Para cada iteración, para $i = 1, 2, \ldots, n$, calcularíamos los elementos del vector solución

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

Ejemplo 5.6 Resolver el siguiente sistema lineal por Jacobi

$$\begin{array}{rrrrcl} 10x_1 & -x_2 & +2x_3 & & = & 6 \\ -x_1 & +11x_2 & -x_3 & +3x_4 & = & 6 \\ 2x_1 & -x_2 & +10x_3 & -x_4 & = & 11 \\ 3x_2 & -x_3 & & +8x_4 & = & 15 \end{array}.$$

El criterio de parada será

$$\left\| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right\|_\infty \leq 0{,}01$$

Primero despejamos x_1 en la primera ecuación, x_2 en la segunda, ...

$$\begin{array}{rcl} x_1 & = & (6 + x_2 - 2x_3)/10 \\ x_2 & = & (6 + x_1 + x_3 - 3x_4)/11 \\ x_3 & = & (11 - 2x_1 + x_2 + x_4)/10 \\ x_4 & = & (15 - 3x_2 + x_3)/8 \end{array}$$

Teniendo en cuenta las ecuaciones definimos la sucesión

$$\begin{array}{rcl} x_1^{(k+1)} & = & (6 + x_2^{(k)} - 2x_3^{(k)})/10 \\ x_2^{(k+1)} & = & (6 + x_1^{(k)} + x_3^{(k)} - 3x_4^{(k)})/11 \\ x_3^{(k+1)} & = & (11 - 2x_1^{(k)} + x_2^{(k)} + x_4^{(k)})/10 \\ x_4^{(k+1)} & = & (15 - 3x_2^{(k)} + x_3^{(k)})/8 \end{array}$$

Primera iteración. Si el punto inicial es $\mathbf{x}^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)}\right)^t = (0,0,0,0)^t$

$$\begin{aligned} x_1^{(1)} &= (6 + x_2^{(0)} - 2x_3^{(0)})/10 = 0{,}6 \\ x_2^{(1)} &= (6 + x_1^{(0)} + x_3^{(0)} - 3x_4^{(0)})/11 = 0{,}545 \\ x_3^{(1)} &= (11 - 2x_1^{(0)} + x_2^{(0)} + x_4^{(0)})/10 = 1{,}1 \\ x_4^{(1)} &= (15 - 3x_2^{(0)} + x_3^{(0)})/8 = 1{,}875 \end{aligned}$$

Comprobamos si se cumple el criterio de parada

$$\left\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\right\|_\infty = \max_{1 \le i \le 4} \left(\left|\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(0)}\right|\right) = \max(0{,}6, 0{,}545, 1{,}1, 1{,}875) = 1{,}875 > 0{,}01$$

Como no se cumple, seguimos iterando.

Segunda iteración.

$$\begin{aligned} x_1^{(2)} &= (6 + x_2^{(1)} - 2x_3^{(1)})/10 && = (6 + 0{,}545 - 2(1{,}1))/10 = 0{,}435 \\ x_2^{(2)} &= (6 + x_1^{(1)} + x_3^{(1)} - 3x_4^{(1)})/11 && = (6 + 0{,}6 + 1{,}1 - 3(1{,}875))/11 = 1{,}886 \\ x_3^{(2)} &= (11 - 2x_1^{(1)} + x_2^{(1)} + x_4^{(1)})/10 && = (11 - 2(0{,}6) + 0{,}545 + (1{,}875))/10 = 1{,}22 \\ x_4^{(2)} &= (15 - 3x_2^{(1)} + x_3^{(1)})/8 && = (15 - 3(0{,}545) + 1{,}1)/8 = 1{,}808 \end{aligned}$$

Comprobamos si se cumple el criterio de parada

$$\left\|\mathbf{x}^{(2)} - \mathbf{x}^{(1)}\right\|_\infty = 0{,}357 > 0{,}01$$

Como no se cumple, seguimos iterando.

Realizamos más iteraciones hasta que se cumpla el criterio de parada.

$$\mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \mathbf{x}^{(1)} = \begin{pmatrix} 0{,}6 \\ 0{,}545 \\ 1{,}1 \\ 1{,}875 \end{pmatrix} \mathbf{x}^{(2)} = \begin{pmatrix} 0{,}435 \\ 0{,}189 \\ 1{,}222 \\ 1{,}808 \end{pmatrix} \mathbf{x}^{(3)} = \begin{pmatrix} 0{,}374 \\ 0{,}203 \\ 1{,}213 \\ 1{,}957 \end{pmatrix}$$

$$\mathbf{x}^{(4)} = \begin{pmatrix} 0{,}378 \\ 0{,}156 \\ 1{,}241 \\ 1{,}950 \end{pmatrix} \dots \mathbf{x}^{(6)} = \begin{pmatrix} 0{,}369 \\ 0{,}153 \\ 1{,}240 \\ 1{,}979 \end{pmatrix}$$

que se cumple en la iteración 6

$$\left\|\mathbf{x}^{(6)} - \mathbf{x}^{(5)}\right\|_\infty = 0{,}007 < 0{,}01$$

Por lo tanto, tomaremos como solución aproximada

$$\mathbf{x}^{(6)} = \begin{pmatrix} 0{,}369 \\ 0{,}153 \\ 1{,}240 \\ 1{,}979 \end{pmatrix}$$

Se puede ver que sus elementos difieren en el tercer decimal con la solución del sistema

$$\mathbf{x} = \begin{pmatrix} 0{,}368 \\ 0{,}154 \\ 1{,}239 \\ 1{,}972 \end{pmatrix}$$

□

3.2. Gauss-Seidel

En este caso para obtener la matriz G realizamos la descomposiciónComo

$$\begin{aligned} &A\mathbf{x} = \mathbf{b} \Rightarrow (L+D+U)\,\mathbf{x} = \mathbf{b} \Rightarrow \\ &(L+D)\mathbf{x} = -U\mathbf{x} + \mathbf{b} \Rightarrow \mathbf{x} = -(L+D)^{-1}U\mathbf{x} + (L+D)^{-1}\mathbf{b} \end{aligned}$$

Tenemos

$$\mathbf{x}^{(k+1)} = -(L+D)^{-1}U\mathbf{x}^{(k)} + (L+D)^{-1}\mathbf{b}$$

y la matriz y el vector de iteración serían

$$\begin{aligned} G = &\quad -(L+D)^{-1}U \\ \mathbf{c} = &\quad (L+D)^{-1}\mathbf{b} \end{aligned}$$

Si elegimos una aproximación inicial $\mathbf{x}^{(0)}$ realizaríamos una serie de iteraciones hasta que se cumpliera un criterio de parada. Para cada iteración, para $i = 1, 2, \ldots, n$, calcularíamos los elementos del vector solución

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

Ejemplo 5.7 Resolver el siguiente sistema lineal por Gauss-Seidel

$$\begin{array}{rrrrcr} 10x_1 & -x_2 & +2x_3 & & = & 6 \\ -x_1 & +11x_2 & -x_3 & +3x_4 & = & 6 \\ 2x_1 & -x_2 & +10x_3 & -x_4 & = & 11 \\ 3x_2 & -x_3 & & +8x_4 & = & 15 \end{array}$$

El criterio de parada será

$$\left\| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right\|_\infty \leq 0{,}01$$

Como en Jacobi, primero despejamos x_1 en la primera ecuación, x_2 en la segunda, ...

$$\begin{aligned} x_1 &= (6 + x_2 - 2x_3)/10 \\ x_2 &= (6 + x_1 + x_3 - 3x_4)/11 \\ x_3 &= (11 - 2x_1 + x_2 + x_4)/10 \\ x_4 &= (15 - 3x_2 + x_3)/8 \end{aligned}$$

Pero ahora, en cuanto hemos calculado la aproximación de una de las incógnitas ya la usamos

$$\begin{aligned} x_1^{(k+1)} &= (6+x_2^{(k)}-2x_3^{(k)})/10 \\ x_2^{(k+1)} &= (6+x_1^{(k+1)}+x_3^{(k)}-3x_4^{(k)})/11 \\ x_3^{(k+1)} &= (11-2x_1^{(k+1)}+x_2^{(k+1)}+x_4^{(k)})/10 \\ x_4^{(k+1)} &= (15-3x_2^{(k+1)}+x_3^{(k+1)})/8 \end{aligned}$$

Primera iteración. Si el punto inicial es $\mathbf{x}^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)}\right)^t = (0,0,0,0)^t$

$$\begin{aligned} x_1^{(1)} &= = (6+x_2^{(0)}-2x_3^{(0)})/10 && = 0{,}6(6+0-0)/10 = 0{,}6 \\ x_2^{(1)} &= (6+x_1^{(1)}+x_3^{(0)}-3x_4^{(0)})/11 && = (6+0{,}6+0-0)/11 = 0{,}6 \\ x_3^{(1)} &= = (11-2x_1^{(1)}+x_2^{(1)}+x_4^{(0)})/10 && = 1{,}1(11-2(0{,}6)+(0{,}6)+0)/10 = 1{,}04 \\ x_4^{(1)} &= (15-3x_2^{(1)}+x_3^{(1)})/8 && = (15-3(0{,}6)+(1{,}04))/8 = 1{,}78 \end{aligned}$$

Comprobamos si se cumple el criterio de parada

$$\left\|\mathbf{x}^{(1)}-\mathbf{x}^{(0)}\right\|_\infty = \max_{1\le i\le 4}\left(\left|\mathbf{x}_i^{(1)}-\mathbf{x}_{\mathrm{i}}{}^{(0)}\right|\right) = \max(0{,}6, 0{,}6, 1{,}04, 1{,}78) = 1{,}78$$

Como no se cumple, seguimos iterando.

Segunda iteración.

$$\begin{aligned} x_1^{(2)} &= (6+x_2^{(1)}-2x_3^{(1)})/10 && = (6+0{,}6-2(1{,}04))/10 = 0{,}452 \\ x_2^{(2)} &= (6+x_1^{(2)}+x_3^{(1)}-3x_4^{(1)})/11 && = (6+0{,}452+1{,}04-3(1{,}78))/11 = 0{,}196 \\ x_3^{(2)} &= (11-2x_1^{(2)}+x_2^{(2)}+x_4^{(1)})/10 && = (11-2(0{,}452)+0{,}196+(1{,}78))/10 = 1{,}207 \\ x_4^{(2)} &= (15-3x_2^{(2)}+x_3^{(2)})/8 && = (15-3(0{,}196)+1{,}207)/8 = 1{,}953 \end{aligned}$$

Comprobamos si se cumple el criterio de parada

$$\left\|\mathbf{x}^{(2)}-\mathbf{x}^{(1)}\right\|_\infty = 0{,}404 > 0{,}01$$

Como no se cumple, seguimos iterando.

Realizamos más iteraciones hasta que se cumpla el criterio de parada.

$$\mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{x}^{(1)} = \begin{pmatrix} 0{,}6 \\ 0{,}6 \\ 1{,}04 \\ 1{,}78 \end{pmatrix} \quad \mathbf{x}^{(2)} = \begin{pmatrix} 0{,}452 \\ 0{,}196 \\ 1{,}207 \\ 1{,}953 \end{pmatrix} \quad \mathbf{x}^{(3)} = \begin{pmatrix} 0{,}378 \\ 0{,}157 \\ 1{,}235 \\ 1{,}971 \end{pmatrix}$$

$$\mathbf{x}^{(4)} = \begin{pmatrix} 0{,}369 \\ 0{,}154 \\ 1{,}239 \\ 1{,}972 \end{pmatrix}$$

que se cumple en la iteración 4

$$\left\|\mathbf{x}^{(4)}-\mathbf{x}^{(3)}\right\|_{\infty}=0{,}009<0{,}01$$

Por lo tanto, tomaremos como solución aproximada

$$\mathbf{x}^{(4)}=\begin{pmatrix}0{,}369\\0{,}154\\1{,}239\\1{,}972\end{pmatrix}$$

Se puede ver que sus elementos difieren en el tercer decimal con la solución del sistema

$$\mathbf{x}=\begin{pmatrix}0{,}368\\0{,}154\\1{,}239\\1{,}972\end{pmatrix}$$

□

3.3. Convergencia de los métodos iterativos

Los métodos iterativos para la resolución de sistemas lineales se basan en reescribir el problema

$$A\mathbf{x}=\mathbf{b}\Longleftrightarrow\mathbf{x}=G\mathbf{x}+\mathbf{c}$$

y tomando un $\mathbf{x}^{(0)}$ como aproximación inicial a la solución, para $k=1,2,\ldots$

$$\mathbf{x}^{(k)}=G\mathbf{x}^{(k-1)}+\mathbf{c}$$

Definición 8 *Una matriz A se dice estrictamente diagonal dominante si*

$$\sum_{j=1(i\neq j)}^{n}|a_{ij}|<|a_{ii}|\quad\text{para}\quad i=1,2,\ldots,n$$

Teorema 5.1 *Una condición suficiente de convergencia de los métodos de Jacobi y Gauss-Seidel es que la matriz de coeficientes del sistema lineal sea estrictamente diagonal dominante.*

Ejemplo 5.8 ¿Podemos garantizar la convergencia del método de Jacobi y Gauss-Seidel utilizando el criterio del teorema 5.1 en el sistema $Ax=b$ con

$$A=\begin{pmatrix}2&1&0\\2&5&-1\\0&-1&3\end{pmatrix}$$

Estudiamos si A es diagonal dominante.

$$A=\begin{pmatrix}2&1&0\\2&5&-1\\0&-1&3\end{pmatrix}\begin{array}{l}|2|<|1|+|0|\\|5|>|2|+|-1|\\|3|>|0|+|-1|\end{array}$$

A no es diagonal dominante. Por lo tanto, usando este criterio podemos garantizar la convergencia tanto del método de Jacobi como de el de Gauss-Seidel.

□

Definición 9 *El radio espectral de una matriz G es el módulo (o valor absoluto) máximo* $|\lambda_i|$ *de los autovalores de G.*

Teorema 5.2 *Dado un sistema lineal* $\mathbf{x} = G\mathbf{x} + \mathbf{c}$*, donde* $I - G$ *es inversible, las siguientes afirmaciones son equivalentes:*

- *El método iterativo asociado* $\mathbf{x}^{(k)} = G\mathbf{x}^{(k-1)} + \mathbf{c}$ para $k = 1, 2, \ldots, n$ *es convergente.*
- *El radio espectral de G es menor que* 1.
- *Alguna norma de G es menor que* 1.

Ejemplo 5.9 Estudiar si el método de Gauss-Seidel sería convergente para el sistema lineal $Ax = b$, siendo

$$A = \begin{pmatrix} 3 & 1 & 1 \\ 1 & 2 & -1 \\ 3 & 1 & 3 \end{pmatrix} \quad b = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}$$

La matriz $B_{G-S} = -(L+D)^{-1}U$ con

$$A = L + D + U = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 3 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}$$

Por lo que

$$L + D = \begin{pmatrix} 3 & 0 & 0 \\ 1 & 2 & 0 \\ 3 & 1 & 3 \end{pmatrix} \quad y \quad U = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}$$

Calculemos por Gauss-Jordan $(L+D)^{-1}$. Dividimos la primera fila por el pivote.

$$\begin{matrix} f_1 \\ f_2 \\ f_3 \end{matrix} \begin{pmatrix} 3 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 0 \\ 3 & 1 & 3 & 0 & 0 & 1 \end{pmatrix} \quad f_1 \rightarrow f_1/3$$

Hacemos ceros por debajo del pivote.

$$\begin{matrix} f_1 \\ f_2 \\ f_3 \end{matrix} \begin{pmatrix} 1 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 0 \\ 3 & 1 & 3 & 0 & 0 & 1 \end{pmatrix} \quad \begin{matrix} f_1 \rightarrow f_1 \\ f_2 \rightarrow f_2 - f_1 \\ f_3 \rightarrow f_3 - 3f_1 \end{matrix}$$

Dividimos la segunda fila por el pivote.

$$\begin{matrix} f_1 \\ f_2 \\ f_3 \end{matrix} \begin{pmatrix} 1 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 2 & 0 & -\frac{1}{3} & 1 & 0 \\ 0 & 1 & 3 & -1 & 0 & 1 \end{pmatrix} \quad f_2 \rightarrow f_2/2$$

Hacemos cero por debajo del pivote (por encima ya tenemos ceros).

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \begin{pmatrix} 1 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 1 & 0 & -\frac{1}{6} & \frac{1}{2} & 0 \\ 0 & 1 & 3 & -1 & 0 & 1 \end{pmatrix} \quad f_3 \rightarrow f_3 - f_2$$

Dividimos la tercera fila por el pivote.

$$\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \begin{pmatrix} 1 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 1 & 0 & -\frac{1}{6} & \frac{1}{2} & 0 \\ 0 & 0 & 3 & -\frac{5}{6} & -\frac{1}{2} & 1 \end{pmatrix} \quad f_3 \rightarrow f_3/3$$

Ya tenemos a la izquierda la matriz identidad.

$$\begin{pmatrix} 1 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 1 & 0 & -\frac{1}{6} & \frac{1}{2} & 0 \\ 0 & 0 & 1 & -\frac{5}{18} & -\frac{1}{6} & \frac{1}{3} \end{pmatrix}$$

Por lo tanto

$$(L+D)^{-1} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{1}{6} & \frac{1}{2} & 0 \\ -\frac{5}{18} & -\frac{1}{6} & \frac{1}{3} \end{pmatrix}$$

y

$$B_{G-S} = -(L+D)^{-1}U = -\begin{pmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{1}{6} & \frac{1}{2} & 0 \\ -\frac{5}{18} & -\frac{1}{6} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{6} & \frac{2}{3} \\ 0 & \frac{5}{18} & \frac{1}{9} \end{pmatrix}$$

Calculemos la norma 1 de esta matriz.

$$B_{G-S}^T = \begin{pmatrix} 0 & 0 & 0 \\ -\frac{1}{3} & \frac{1}{6} & \frac{5}{18} \\ -\frac{1}{3} & \frac{2}{3} & \frac{1}{9} \end{pmatrix} \begin{array}{l} 0+0+0=0 \\ \left|-\frac{1}{3}\right|+\left|\frac{1}{6}\right|+\left|\frac{5}{18}\right| = \frac{7}{9} \\ \left|-\frac{1}{3}\right|+\left|\frac{2}{3}\right|+\left|\frac{1}{9}\right| = \frac{10}{9} \end{array}$$

La norma 1 viene dada por

$$||B_{G-S}||_1 = \text{Max}\left(0, \frac{7}{9}, \frac{10}{9}\right) = \frac{10}{9} > 1$$

Como $||B_{G-S}||_1 < 1$ es condición suficiente, no necesaria, de convergencia, no podemos asegurar ni que converge ni que no converge. Calculemos la norma infinito.

$$B_{G-S} = \begin{pmatrix} 0 & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{6} & \frac{2}{3} \\ 0 & \frac{5}{18} & \frac{1}{9} \end{pmatrix} \begin{array}{l} 0+\left|-\frac{1}{3}\right|+\left|-\frac{1}{3}\right| = \frac{2}{3} \\ 0+\left|\frac{1}{6}\right|+\left|\frac{2}{3}\right| = \frac{5}{6} \\ 0+\left|\frac{5}{18}\right|+\left|\frac{1}{9}\right| = \frac{7}{18} \end{array}$$

La norma infinito es

$$||B_{G-S}||_\infty = \text{Max}\left(\frac{2}{3}, \frac{5}{6}, \frac{7}{18}\right) = \frac{5}{6} < 1$$

Como $||B_{G-S}||_\infty < 1$ podemos asegurar que converge.

Además, si estudiamos los autovalores de la matriz B_{G-S}:

$$\begin{vmatrix} 0-\lambda & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{6}-\lambda & \frac{2}{3} \\ 0 & \frac{5}{18} & \frac{1}{9}-\lambda \end{vmatrix} = 0$$

$$-\lambda\left(\frac{1}{6}-\lambda\right)\left(\frac{1}{9}-\lambda\right) - \frac{2}{3}\frac{5}{18}(-\lambda) = \frac{1}{6}\lambda + \frac{5}{18}\lambda^2 - \lambda^3 = (\lambda)\left(\frac{1}{6} + \frac{5}{18}\lambda - \lambda^2\right) = 0$$

Y los autovalores son

$$\lambda_1 = 0, \quad \lambda_2 = 0{,}57, \quad \lambda_3 = 0{,}29$$

que son todos menores que 1 en valor absoluto y esto es coherente con el resultado obtenido a partir de la norma infinito.

□

Capítulo 6

Optimización

El problema general de la Optimización se centra en el desarrollo de técnicas matemáticas que nos permitan deducir la existencia de puntos de mínimo y/o de máximo de aplicaciones $J : V \to \mathbb{R}$, así como de desarrollar métodos numéricos que nos permitan calcular aproximadamente dichos puntos.

De acuerdo a la naturaleza del conjunto V, el problema de optimización puede ser

- de dimensión finita, cuando $V \subset \mathbb{R}^n$,
- de dimensión infinita, cuando V es un espacio de funciones.

Otra clasificación importante de los problemas de optimización es la que los divide en *problemas sin restricciones*, o *problemas con restricciones*, también llamados *problemas condicionados*. Las restricciones suelen expresarse en términos de relaciones funcionales que limitan al conjunto V.

Observemos finalmente que maximizar la aplicación J es equivalente a minimizar $-J$. Por ello, solo trataremos el caso de la minimización, entendiendo que las condiciones y técnicas que introduzcamos pueden traducirse directamente al problema de maximización.

1. Definición del problema de optimización

Un problema de optimización puede presentarse de modo esquemático de la siguiente manera: una variable física, de decisión o de control debe ser elegida de modo óptimo, es decir, de modo que se optimize (minimize o maximize, según el caso) un criterio físico (acción, energía, ...), un criterio técnico (precisión, estabilidad, duración,...) o económico (coste, rentabilidad, productividad,...), todo respetando restricciones intrínsecas a la situación que se considera.

A continuación introducimos la terminología y la notación que usaremos para este tipo de problemas. Un *problema de optimización* está formado por

1. Un *criterio* (o *coste*, u *objetivo*), J, que aplica el espacio V de variables de decisión en $\mathbb{R}$,
$$J : V \to \mathbb{R}.$$
2. Las *restricciones*. En general, cualquier elemento del espacio V no es admisible como solución del problema, ya que debe satisfacer algunas restricciones que determinan el *espacio de*

soluciones. Estas restricciones aparecen en las aplicaciones de diversas formas, que pueden intervenir de forma simultánea:

a) *Restricciones funcionales de igualdad*

$$F(v) = 0, \tag{6.1}$$

donde $F : V \to \mathbb{R}^m$, con $m < n$. Diremos que la solución está sujeta a *m restricciones de igualdad*

$$F_i(v) = 0, \quad i = 1, ..., m.$$

b) *Restricciones funcionales de desigualdad*

$$G(v) \leq 0, \tag{6.2}$$

donde $G : V \to \mathbb{R}^p$, con $p < n$, y (6.2) tiene el siguiente sentido

$$G_j(v) \leq 0 \quad j = 1, ..., p.$$

Decimos entonces que tenemos *p restricciones de desigualdad*. Para una solución admisible, v, diremos que las restricciones de desigualdad están *saturadas* si $G_j(v) = 0$, y que no lo están si $G_j(v) < 0$.

c) Las restricciones de igualdad y desigualdad son casos particulares importantes de las restricciones expresadas como *restricciones de conjunto*, del tipo

$$v \in C,$$

donde $C \subset V$ es un conjunto dado.

En todo caso, el conjunto de restricciones define un subconjunto $U \subset V$, al que llamaremos *conjunto de soluciones admisibles*:

$$U := \{v : v \quad \text{satisface las restricciones}\}. \tag{6.3}$$

El problema de minimización consiste, por tanto, en encontrar $u \in U$ tal que

$$J(u) \leq J(v) \quad \text{para toda } v \in U, \tag{6.4}$$

Si tal u existe, diremos que es un *mínimo* de J, y que $J(u)$ es un *valor mínimo* del problema de minimización. A menudo usaremos la siguiente forma abreviada para referirnos a la formulación del problema de minimización:

$$\begin{cases} \text{mín} J(v) \\ v \in C, \quad F(v) = 0, \quad G(v) \leq 0. \end{cases}$$

En general, no existen técnicas para resolver el problema (6.4) en todo el conjunto U, es decir, para encontrar un *mínimo global o absoluto*. Por ello, normalmente, nos contentaremos con resolver el problema de hallar un *mínimo local o relativo* $\bar{u} \in U$, es decir, resolver

$$J(\bar{u}) \leq J(v) \quad \text{para toda } v \in U \cap B.$$

donde B es un entorno de $\bar{u}$. Claramente, un mínimo global es siempre un mínimo local, mientras que el recíproco es falso, en general.

1.1. Ejemplos

Optimización en dimensión finita: Programación lineal

En el caso de dimensión finita utilizaremos, normalmente, la notación f para el criterio (en vez de J) y x para las variables independientes (en vez de v).

Muchos problemas económicos o técnicos se presentan en la forma de un *programa lineal*:

$$\begin{cases} \text{mín } f(x) = c^T x \\ x \in \mathbb{R}^n, \quad Ax \geq b, \end{cases}$$

donde $c \in \mathbb{R}^n$ es un vector fila, $x \in \mathbb{R}^n$ es un vector columna, A es una matriz de orden $m \times n$, y $b \in \mathbb{R}^n$ es un vector columna.

El primer programa lineal (tratado en 1944) que fue el origen de lo que hoy se denomina *investigación operativa*, es un problema de *ración alimentaria*. Se dispone de n tipos de alimentos $x_1, \ldots, x_n$, y tomamos en consideración m parámetros alimentarios (proteínas, vitaminas, etc). Desiganamos por

- a_{ij} la cantidad del parámetro i contenida en una unidad del alimento j,
- b_j la cantidad mínima necesaria que debe ser introducida en cada ración, y
- c_j el coste de la unidad de alimento j.

Así, la ración alimentaria de coste mínimo, dada por x_j unidades del alimento j y satisfaciendo las restricciones de contenido mínimo del parámetro i, es solución del programa lineal

$$\begin{cases} \text{ínf} \sum_{j=1}^{n} c_j x_j \\ x_j \geq 0, \quad j = 1, ..., n, \quad \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, ..., m. \end{cases}$$

Optimización en dimensión infinita: El problema de la braquistocrona

El problema de la braquistocrona, o curva de descenso más rápido, es uno de los problemas de optimización en dimensión infinita más antiguos. La primera solución fue dada por Johann Bernoulli en 1696, aunque también dieron soluciones algunos contemporáneos suyos como Jacob Bernoulli, Leibniz y Newton.

El problema es: de entre todas las curvas que unen dos puntos A y B se desea hallar aquella a lo largo de la cual un punto material, moviéndose únicamente debido a la fuerza de la gravedad, va desde A hasta B en el menor tiempo.

Para resolver este problema debemos considerar todas las posibles curvas que unen A y B. A una determinada curva, γ, le corresponderá un valor determinado, T, del tiempo invertido para el descenso del punto material a lo largo de ella. El tiempo, T, dependerá de la elección de γ. De todas las curvas que unen A con B debemos hallar aquella a la que corresponda el menor valor de T. El problema puede plantearse de la siguiente forma.

Tracemos un plano vertical que pase por los puntos A y B. La curva de más rápido descenso debe evidentemente estar en él, así que podemos restringirnos a curvas sobre dicho plano. Tomemos

el punto A como el origen de coordenadas, el eje OX apuntando en la dirección de la gravedad y sea $B = (x_1, y_1)$, con $x_1 > 0$ y $y_1 \geq 0$. Consideremos una curva arbitraria descrita por la ecuación

$$y = y(x) \quad 0 \leq x \leq x_1. \tag{6.5}$$

Como la curva pasa por A y B, la función y debe verificar

$$y(0) = 0, \quad y(x_1) = y_1. \tag{6.6}$$

El movimiento de la masa puntual puede describirse por medio de la ley de la conservación de la energía del siguiente modo. En el punto A, en el que asumimos que la velocidad inicial es nula, se tiene

$$E = E_p mgh_A,$$

donde la energía, $E > 0$, es una constante y h_A es la altura a la que se encuentra el punto A. En cualquier punto por debajo será

$$\frac{1}{2}mv^2 + mgh = E,$$

luego $v^2 = 2g(h_A - h)$, y tomando la coordenada vertical como $x = h_A - h$, deducimos que la velocidad del movimiento del punto material es

$$v \equiv \frac{ds}{dt} = \sqrt{2gx},$$

siendo s una parametrización de la trayectoria del punto material. Deducimos que

$$dt = \frac{ds}{\sqrt{2gx}},$$

y como la longitud de arco de la curva viene dada por

$$ds = \sqrt{1 + y'(x)^2}dx,$$

tenemos que el tiempo empleado a lo largo de la curva y viene dado por

$$J(y) = \int_0^{x_1} \Big(\frac{1 + y'(x)^2}{2gx}\Big)^{1/2} dx. \tag{6.7}$$

Hallar la braquistocrona es equivalente a resolver el siguiente problema de minimización: de entre todas las posibles funciones diferenciables (6.5) que verifiquen las condiciones (6.6), hallar la que corresponda al menor valor de la integral (6.7).

2. Optimización sin restricciones en dimensión finita

2.1. Conceptos básicos y notación

Dada una función $f : \mathbb{R}^n \to \mathbb{R}$ que tenga derivadas parciales segundas continuas, denotamos por

$$\frac{\partial f}{\partial x_i}(x), \quad \text{y} \quad \nabla f(x) = \Big(\frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_n}(x)\Big),$$

a su *derivada parcial respecto x_i en x*, para $i = 1, \ldots, n$, y a su *gradiente* en x, respectivamente. La *matriz hessiana* de f en x viene dada a partir de las derivadas parciales segundas como

$$H_f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \ldots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \ldots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\ \ldots & \ldots & \ldots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \ldots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{pmatrix} \tag{6.8}$$

Una propiedad importante de la matriz hessiana es que es simétrica.

La norma euclídea de un vector $x \in \mathbb{R}^n$ viene dada por

$$\|x\| = \big(\sum_{i=1}^{n} x_i^2\big)^{1/2}.$$

La norma de vectores induce una norma en las matrices, definida de la siguiente manera:

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

En optimización, la *definición* de la matriz hessiana juega un papel importante.

Definición 10 *Una matriz cuadrada, A, de orden n se dice* definida positiva *si $x^T Ax > 0$ para todo $x \in \mathbb{R}^n$. Si la desigualdad anterior no es estricta entonces diremos que A es* semidefinida positiva.

El Teorema de Taylor es una herramienta básica que utilizaremos a menudo.

Teorema 6.1 (Teorema de Taylor) *Sea f dos veces diferenciable con continuidad en un entorno de un punto $x^* \in \mathbb{R}^n$. Entonces, para todo $e \in \mathbb{R}^n$ con $\|e\|$ suficientemente pequeña se tiene*

$$f(x^* + e) = f(x^*) + \nabla f(x^*)^T e + \frac{1}{2} e^T H_f(x) e + o(\|e\|^2). \tag{6.9}$$

Recordemos que un *entorno de radio* ρ *centrado en* x^* es el conjunto $B_\rho(x^*) = \{x \in \mathbb{R}^n : \|x - x^*\| < \rho\}$ (una bola $n-$dimensional). Por otra parte, la notación $o(t^2)$ (*o pequeña de* t^2) significa que

$$\lim_{t \to 0} \frac{o(t^2)}{t^2} = 0.$$

2.2. Condiciones necesarias y suficientes de optimalidad local

Dada una función $f : \mathbb{R}^n \to \mathbb{R}$ que tenga derivadas parciales segundas continuas, el programa usual que utiliza el cálculo diferencial para la localización de puntos de mínimo es el siguiente.

1. Resolvemos el sistema de ecuaciones de los puntos críticos, es decir, hallamos los $x^* \in \mathbb{R}^n$ tales que $\nabla f(x^*) = 0$, o escrito de otra manera,

$$\frac{\partial f}{\partial x_1}(x^*) = 0, \ldots, \frac{\partial f}{\partial x_n}(x^*) = 0. \tag{6.10}$$

2. Evaluamos la matriz hessiana de f en los puntos críticos, y comprobamos si dicha matriz es definida positiva

En caso afirmativo, x^* es un punto de mínimo local para f, es decir, existe un radio $\rho > 0$ tal que

$$f(x^*) \leq f(x) \quad \text{para todo } x \in B_\rho(x^*).$$

Veamos por qué este programa está justificado.

Teorema 6.2 (Condiciones necesarias de optimalidad) *Sea f dos veces diferenciable con continuidad y supongamos que x^* es un mínimo local. Entonces $\nabla f(x^*) = 0$ y $H_f(x^*)$ es semidefinida positiva.*

Demostración. Sea $v \in \mathbb{R}^n$ un vector dado. El Teorema de Taylor implica que

$$f(x^* + tv) = f(x^*) + t\nabla f(x^*)^T v + \frac{t^2}{2} v^T H_f(x) v + o(\|t\|^2).$$

Como x^* es un mínimo local, tenemos $f(x^* + tu) \geq f(x^*)$, para un t suficientemente pequeño. Entonces, dividiendo por t obtenemos

$$\nabla f(x^*)^T v + \frac{t}{2} v^T H_f(x) v + o(\|t\|) \geq 0. \tag{6.11}$$

Poniendo $t = 0$ y $v = -\nabla f(x^*)$ deducimos $\|\nabla f(x^*)\| = 0$, es decir, $\nabla f(x^*) = 0$. Ahora, usando esta identidad en (6.11), dividiendo por t y poniendo $t = 0$, obtenemos

$$\frac{1}{2} v^T H_f(x) v \geq 0.$$

□

La condición (6.10), aunque necesaria, no es suficiente para que x^* sea un punto de mínimo de f. Es decir, existen puntos críticos de f que no son mínimos. Para asegurarnos de que un punto crítico es, efectivamente, un mínimo, recurrimos al siguiente resultado.

Teorema 6.3 (Condiciones suficientes de optimalidad) *Sea f dos veces diferenciable con continuidad y supongamos que x^* es un punto crítico de f y que $h_f(x^*)$ es definida positiva. Entonces x^* es un mínimo local de f.*

Demostración. Sea $v \in \mathbb{R}^n$ un vector no nulo dado. Para t suficientemente pequeño, el Teorema de Taylor implica que

$$f(x^* + tv) = f(x^*) + \frac{t^2}{2} v^T H_f(x) v + o(\|t\|^2).$$

Como $H_f(x^*)$ es definida positiva, existe un número $\lambda > 0$ tal que

$$v^T H_f(x) v > \lambda \|v\|^2 > 0.$$

Entonces

$$f(x^* + tv) - f(x^*) = \frac{t^2}{2} v^T H_f(x) v + o(\|t\|^2) > \lambda \frac{t^2}{2} \|v\|^2 + o(\|t\|^2) > 0,$$

para todo $t \neq 0$ suficientemente pequeño. □

Ejemplo 6.1 Observemos que lo que nos dice el Teorema de Taylor es que una función f que tiene un mínimo local en x^* es local y aproximadamente, como un paraboloide. Por ejemplo, supongamos que $x^* = 0$ es un mínimo de una función bidimensional ($n = 2$). Tomando $e = (x_1, x_2)$ y despreciando el término $o(\|e\|^2)$, obtenemos

$$\begin{aligned} f(x_1,x_2) \approx & f(0,0) + x_1 \frac{\partial f}{\partial x_1}(0,0) + x_2 \frac{\partial f}{\partial x_2}(0,0) + \frac{1}{2}\Big(\frac{\partial^2 f}{\partial x_1^2}(0,0)x_1^2 + \frac{\partial^2 f}{\partial x_2^2}(0,0)x_2^2 \\ & + 2\frac{\partial^2 f}{\partial x_1 \partial x_2}(0,0)x_1 x_2 \Big) \\ = & f(0,0) + \frac{1}{2}\Big(\frac{\partial^2 f}{\partial x_1^2}(0,0)x_1^2 + \frac{\partial^2 f}{\partial x_2^2}(0,0)x_2^2 + 2\frac{\partial^2 f}{\partial x_1 \partial x_2}(0,0)x_1 x_2 \Big) \\ > & f(0,0) + \lambda(x_1^2 + x_2^2), \end{aligned}$$

para cierto $\lambda > 0$, debido a que $H_f(0)$ es definida positiva. □

Aunque, en general, una función puede tener múltiples puntos de mínimo local y el método diferencial no nos permite decidir cuál de ellos es el mínimo absoluto, existe una importante excepción, la de las funciones convexas.

Definición 11 *Se dice que un conjunto $\Omega \subset \mathbb{R}^n$ es* convexo *si para todos $x,\ y \in \Omega$, y para todo $\mu \in [0,1]$ se tiene*

$$\mu x + (1-\mu)y \in \Omega.$$

Se dice que una función $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ es convexa *si para todos $x,\ y \in \Omega$ y para todo $\mu \in [0,1]$ se tiene*

$$f(\mu x + (1-\mu)y) \leq \mu f(x) + (1-\mu) f(y).$$

No es difícil probar que si $\Omega \subset \mathbb{R}^n$ es convexo y $f : \Omega \to \mathbb{R}$ es convexa y diferenciable, entonces puede tener, a lo más, un único punto crítico. En caso de existir, dicho punto crítico corresponde a un mínimo global de la función.

Resolver el sistema de ecuaciones (6.10), también conocido como las *condiciones de optimalidad de primer orden*, de forma exacta no es posible, en general. Por ello introducimos métodos aproximados, basados en algoritmos iterativos.

Ejemplo 6.2 Consideremos una función derivable definida en $\mathbb{R}$, esto es, $f : \mathbb{R} \to \mathbb{R}$. El sistema de ecuaciones (6.10) para los puntos críticos se reduce a una única ecuación

$$f'(x^*) = 0.$$

Usando el método de Newton para aproximar ceros de funciones, tal como introdujimos en la fórmula (2.2) del Capítulo 2, el algoritmo de aproximación de los puntos críticos de f viene dado por

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}, \quad \text{para } k = 1, 2, \ldots$$

donde x_0 es una aproximación inicial dada. Claramente, una condición necesaria para que este algoritmo converja es que $f''(x) \neq 0$ en el conjunto de iterandos anterior. De hecho, si lo que buscamos en un punto de mínimo, debemos tener $f''(x) > 0$ cerca del mínimo. □

2.3. Método de Newton

El método de Newton para funciones $f:\mathbb{R}^n \to \mathbb{R}$ se deduce usando el desarrollo de Taylor dado por la fórmula (6.9). Considerando la aproximación de segundo orden, es decir, despreciando el término $o(\|e\|^2)$, obtenemos

$$f(x) \approx f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T H_f(x_k)(x - x_k), \tag{6.12}$$

donde H_f es la matriz hessiana de derivadas parciales segundas de f, dada por (6.8). Para hallar la aproximación de un punto crítico de f, derivamos la parte derecha de (6.12) respecto x_j, para $j = 1, \ldots, n$, e igualamos a cero. Obtenemos el sistema de ecuaciones lineales

$$\nabla f(x_k) + H_f(x_k)(x - x_k) = 0,$$

de donde, llamando x_{k+1} a la solución, deducimos

$$x_{k+1} = x_k - (H_f(x_k))^{-1} \nabla f(x_k). \tag{6.13}$$

Observemos que el método de minimización de Newton, al igual que el correspondiente método de Newton para la aproximación de ceros de funciones, solo converge si la iteración inicial, x_0, está suficientemente cerca del mínimo. Para ello, debemos comprobar que la matriz $H_f(x_0)$ sea definida positiva.

Como la inversión de matrices es, en general, un cálculo costoso, al usar el método de Newton resolveremos, en vez de (6.13), el sistema

$$H_f(x_k) y = \nabla f(x_k), \tag{6.14}$$

y entonces, pondremos $x_{k+1} = x_k + y$. Una ventaja adicional de que la matriz Hessiana sea definida positiva es que admite una factorización de Cholesky, es decir, existe una matriz triangular inferior, L, con diagonal positiva tal que $H_f(x_k) = LL^T$. De este modo, una vez obtenida la factorización, el sistema (6.14) puede resolverse por sustitución progresiva.

Terminación de las iteraciones y estimación del error

Puesto que el método busca un punto crítico, un criterio razonable de parada de las iteraciones podría ser cuando

$$\|\nabla f(x_k)\| \leq \tau_r \|\nabla f(x_0)\|, \tag{6.15}$$

con $\tau_r \in (0,1)$, reflejando de este modo la disminución de la norma del gradiente. Sin embargo, si $\|\nabla f(x_0)\|$ es pequeño, podría no ser posible satisfacer (6.15) en la aritmética de coma flotante, de modo que las iteraciones no pararían nunca. Un criterio algo más exigente, pero también más seguro, se basa en la utilización de una combinación entre el error relativo y el absoluto, i.e.

$$\|\nabla f(x_k)\| \leq \tau_r \|\nabla f(x_0)\| + \tau_a,$$

donde τ_a es una tolerancia de error absoluto.

Terminamos esta sección con un resultado de convergencia.

Teorema 6.4 *Supongamos que f es* tres *veces diferenciable con continuidad, que x^* es un punto crítico de f y que $H_f(x^*)$ es definida positiva. Entonces, si x_0 está suficientemente cerca de x^*, las iteraciones del método de Newton (6.13) convergen cuadráticamente a x^*, es decir,*

$$\|x_{k+1} - x^*\| \leq K \|x_k - x^*\|^2,$$

para cierta constante $K > 0$.

2.4. Método del gradiente

En el método del gradiente, también conocido como método de descenso, se buscan direcciones apropiadas de modo que cuando se avanza en el iterando de x_k a x_{k+1} se disminuya en el valor de la función, es decir, se tenga $f(x_{k+1}) < f(x_k)$.

Para ello, definimos la iteración

$$x_{k+1} = x_k + \alpha_k d_k, \tag{6.16}$$

donde d_k es la dirección en la etapa k y $\alpha_k > 0$ la longitud del paso correspondiente. Del desarrollo de Taylor de primer orden de f, obtenemos

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) \approx f(x_k) + \alpha_k \langle \nabla f(x_k), d_k \rangle,$$

de modo que para que el descenso sea máximo debemos tomar la dirección opuesta a la de $\nabla f(x_k)$, esto es $d_k = -\nabla f(x_k)$, pues así

$$f(x_{k+1}) \approx f(x_k) - \alpha_k \|\nabla f(x_k)\|^2 \leq f(x_k),$$

ya que $\alpha_k > 0$.

Con ello, de (6.16) obtenemos

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k). \tag{6.17}$$

Para elegir la longitud del paso, definimos la función $\phi : \mathbb{R} \to \mathbb{R}$ dada por $\phi(\alpha) = f(x_k + \alpha d_k)$ y buscamos el α_k que minimiza ϕ. Observemos que, de este modo, hemos reducido el problema de minimización en n dimensiones a un problema de minimización en una dimensión, que puede ser resuelto, por ejemplo, por el método de Newton.

En la práctica, en vez de minimizar la función ϕ, se suele preferir la minimización de un interpolante de ϕ. Por ejemplo, como se tienen los datos

$$\phi(0) = f(x_k), \quad \phi(1) = f(x_k + d_k), \quad \text{y } \phi'(0) = \langle -d_k, d_k \rangle < 0,$$

puede tomarse la aproximación de $\phi(\alpha)$, para $\alpha \in [0,1]$, por el polinomio cuadrático

$$q(\alpha) = \phi(0) + \phi'(0)\alpha + (\phi(1) - \phi(0) - \phi'(0))\alpha^2,$$

cuyo mínimo global puede calcularse fácilmente. Por una parte, si $\phi(1) - \phi(0) - \phi'(0) < 0$, entonces el mínimo de q se encuentra en la frontera del intervalo $[0,1]$, y tomaremos $\alpha = 1$ ($\alpha = 0$ no está permitido, pues de este modo no hay avance del método, véase (6.16)).

Por otra parte, si $\phi(1) - \phi(0) - \phi'(0) > 0$, entonces ϕ posee el local dado por

$$\alpha_L = \frac{-\phi'(0)}{2(\phi(1) - \phi(0) - \phi'(0))} > 0,$$

de modo que tomeremos $\alpha = \text{mín}\{1, \alpha_L\}$.

Una propiedad inherente al método del gradiente es que la trayectoria que siguen los iterandos en su convergencia al mínimo es una trayectoria en zig-zag. En efecto, si α_k es el mínimo exacto de $\phi(\alpha)$ entonces, por la regla de la cadena,

$$0 = \phi'(\alpha_k) = \langle \nabla f(x_k + \alpha_k d_k), d_k \rangle,$$

es decir, $d_k = -\nabla f(x_k)$ y $\nabla f(x_k + \alpha_k d_k) = \nabla f(x_{k+1})$ son ortogonales.

Terminación de las iteraciones y estimación del error

Al igual que en el método de Newton, un test de parada razonable es el que se obtiene como combinación de los errores relativo y absoluto de la norma del gradiente,

$$\|\nabla f(x_k)\| \leq \tau_r \|\nabla f(x_0)\| + \tau_a,$$

donde $\tau_r \in (0,1)$ es una tolerancia para el error relativo y τ_a es una tolerancia para el error absoluto.

En general, el método de descenso no tiene buenas propiedades de convergencia. Dependiendo de la función, el método puede ser muy lento. Ilustramos este hecho con un ejemplo.

Ejemplo 6.3 Consideremos la función $f(x) = \frac{a}{2}x^2$, con $a \in (0,1)$, que tiene su único mínimo en $x^* = 0$. Un sencillo cálculo para el cómputo del paso $\alpha = \text{mín}\{1, \alpha_L\}$ nos muestra que $\alpha_L = 1/a$, de modo que debemos tomar $\alpha = 1$. Entonces, las iteraciones (6.17) son

$$x_{k+1} = x_k - f'(x_k) = (1-a)x_k,$$

con lo cual obtenemos solo una convergencia lineal:

$$|x_{k+1} - x_k| = a|x_k - x^*|.$$

Además, obtenemos por recursión que $x_k = (1-a)^k x_0$, con lo cual, si a está próximo a cero, la convergencia es extremadamente lenta. □

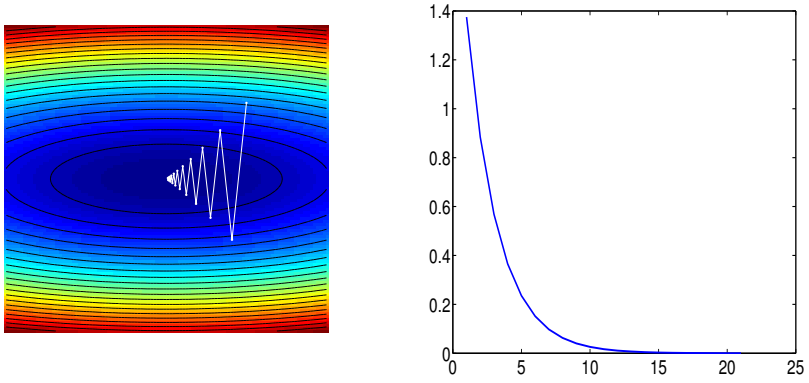


Figura 6.1: Trayectorias de descenso de x_k y de $f(x_k)$.

3. Optimización con restricciones en dimensión finita

La elección del método de resolución de un problema de optimización con restricciones depende fundamentalmente del tipo de restricciones que se impongan: igualdad, desigualdad, conjunto,...

En esta sección introduciremos dos métodos que son especialmente importantes. El *método de los multiplicadores de Lagrange* y el *método del penalty*. El primero se utiliza cuando las restricciones son de igualdad o desigualdad. El segundo es un método de aplicación general.

3.1. Multiplicadores de Lagrange. Restricciones de igualdad

El método de los multiplicadores de Lagrange nos permite usar las técnicas de minimización sin restricciones a problemas condicionados con restricciones de igualdad. Recordemos el planteamiento del problema a resolver.

Dada una función objetivo diferenciable $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$, y un conjunto de funciones diferenciables $\phi_i : \Omega \subset \mathbb{R}^n \to \mathbb{R}$, para $i = 1, \ldots, m$, con $m < n$, encontrar un mínimo x^* de f en Ω que satisfaga las restricciones de igualdad $\phi_i(x^*) = 0$ para todo $i = 1, \ldots, m$. Tenemos el siguiente reultado.

Teorema 6.5 (Condiciones necesarias para un mínimo relativo condicionado) *Supongamos que x^* es un punto del conjunto*

$$U = \{x \in \Omega : \phi_i(x) = 0, \quad 1 \leq i \leq m\} \subset \Omega, \tag{6.18}$$

tal que los m vectores $\nabla\Phi_i(x^*) \in \mathbb{R}^n$, con $i = 1, \ldots, m$, son linealmente independientes. Entonces, si f tiene un mínimo relativo en x^* relativo al conjunto U, existen m números $\lambda_i(x^*)$, $i = 1, \ldots, m$, tales que

$$\nabla f(x^*) + \lambda_1(x^*)\nabla\phi_1(x^*) + \ldots + \lambda_m(x^*)\nabla\phi_m(x^*) = 0. \tag{6.19}$$

Los números $\lambda_i(x^*)$ son llamados *multiplicadores de Lagrange*.

Aunque el Teorema 6.5 nos da un criterio para decidir si un punto x^* puede ser o no un mínimo condicionado, no nos proporciona niguna herramienta para hallar un tal candidato x^*.

La herramienta más común para encontrarlo es el uso de la *función Lagrangiana*. Denotemos por λ al vector $(\lambda_1, \ldots, \lambda_m)$, por $\phi : \mathbb{R}^n \to \mathbb{R}^m$ a la función $\phi(x) = (\phi_1(x), \ldots, \phi_m(x))$, y consideremos la función $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ dada por

$$L(x, \lambda) = f(x) + \lambda^T \phi(x) = f(x) + \sum_{i=1}^{m} \lambda_i \phi_i(x).$$

Si (x^*, λ^*) es un mínimo de L (sin restricciones) entonces debe ser $\nabla_{(x,\lambda)} L(x^*, \lambda^*) = 0$, es decir, se deben cumplir las n ecuaciones de la derivación respecto a x

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla\phi_i(x^*) = 0, \tag{6.20}$$

y las m ecuaciones de la derivación respecto a λ

$$\phi_i(x^*) = 0, \tag{6.21}$$

para $i = 1, \ldots, m$. Observemos que (6.21) es, precisamente, la condición de restricción (6.18), y que (6.20) es la condición (6.19). Concluimos, pues, que cualquier x^* tal que (x^*, λ^*) es un punto crítico de $L(x, \lambda)$ es un candidato a mínimo condicionado del problema de minimización con restricciones.

Ejemplo 6.4 Sean $f(x_1, x_2) = -x_2$ y $\phi(x_1, x_2) = x_1^2 + x_2^2 - 1$ ($n = 2,\ m = 1$). El conjunto de restricciones es, por tanto, la circunferencia

$$U = \{(x_1, x_2) \in \mathbb{R}^2 : x_1^2 + x_2^2 = 1\}.$$

La lagrangiana viene dada por

$$L(x_1, x_2, \lambda) = -x_2 + \lambda(x_1^2 + x_2^2 - 1).$$

Los puntos críticos de L se determinan por

$$\begin{aligned} 0 &= \frac{\partial L}{\partial x_1}(x^*, \lambda^*) = 2\lambda x_1, \\ 0 &= \frac{\partial L}{\partial x_2}(x^*, \lambda^*) = -1 + 2\lambda x_2, \\ 0 &= \frac{\partial L}{\partial \lambda}(x^*, \lambda^*) = x_1^2 + x_2^2 - 1. \end{aligned}$$

Resolviendo, obtenemos $x_1^* = 0$, $x_2^* = \pm 1$ y $\lambda^* = 1/2x_2^*$. □

Terminamos esta sección explicitando las condiciones suficientes de segundo orden para un mínimo condicionado con restricciones de igualdad.

Teorema 6.6 (Condiciones suficientes para un mínimo relativo condicionado) *Sean $x^* \in U$, con U el conjunto de restricciones dado por (6.18) y $\lambda \in \mathbb{R}^m$ tales que se satisface (6.19). Supongamos que la matriz hessiana de la función lagrangiana, L, con respecto a la variable x, dada por*

$$H(x^*) = H_f(x^*) + \lambda^T H_\phi(x^*)$$

es definida positiva en el conjunto $M = \{y \in \mathbb{R}^m : \nabla\phi(x^*)^T y = 0\}$. Entonces x^* es un mínimo condicionado de f en el conjunto U.

Observemos que, en el ejemplo anterior,

$$H(x^*) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} + \lambda^* \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \quad M = \{(y_1, y_2) \in \mathbb{R}^2 : x_2^* y_2 = 0\}.$$

Por tanto, $H(x^*)$ es definida positiva solo para $x^* = (0, 1)$. El otro punto crítico de la lagrangiana, $(0, -1)$, corresponde a un máximo condicionado.

3.2. Método de penalización

Nuevamente, con el método de penalización buscamos reducir un problema de minimización con restricciones a uno sin restricciones. En este caso, las restricciones pueden ser de un tipo más general que las de igualdad. El problema a resolver es el de, dada una función objetivo diferenciable $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$, y un conjunto $S \subset \Omega$, encontrar un mínimo x^* de f en S, es decir,

$$\min_{x \in S} f(x) \tag{6.22}$$

La idea del método de penalización es reemplazar la función objetivo $f(x)$ por otra función

$$f(x) + cP(x) \tag{6.23}$$

y resolver el problema sin restricciones. Para ello, tomamos c como una constante positiva y P satisfaciendo:

1. P continua en Ω,
2. $P(x) \geq 0$ para $x \in \Omega$, y
3. $P(x) = 0$ si y solo si $x \in S$.

Ejemplo 6.5 Supongamos que S está dado por m restricciones de desigualdad,

$$S = \{x \in \mathbb{R}^n : \phi_i(x) \leq 0, \quad i = 1, \ldots, m\}.$$

Un ejemplo de función penalización es

$$P(x) = \frac{1}{2} \sum_{i=1}^{m} \text{máx}(0, \phi_i(x))^2.$$

En la Figura 6.2 podemos ver un ejemplo de función $cP(x)$ en el caso unidimensional con $\phi_1(x) = x - b$ y $\phi_2(x) = a - x$. Para c grande, el mínimo de la función (6.23) estará en una región donde P es pequeño. Así, incrementando c esperamos que los puntos de mínimo correspondientes se aproximarán al conjunto S y, si están cerca unos de otros, minimizarán también f. Idealmente, cuando $c \to \infty$, la solución del problema con penalización convergerá a la solución del problema de minimización condicionado (6.22). □

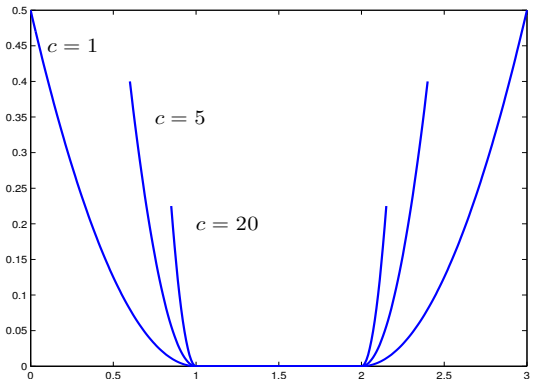


Figura 6.2: La función $cP(x)$ para varios valores de c.

El procedimiento para resolver el problema de minimización condicionado (6.22) mediante el método del penalización es el siguiente: Sea c_k una sucesión tal que, para todo $k = 1, 2, \ldots$,

- $c_k \geq 0$
- $c_{k+1} > c_k$,
- $\lim_{k \to \infty} c_k = \infty$.

Definamos la función

$$q(c, x) = f(x) + cP(x). \tag{6.24}$$

Para cada k, asumamos que el problema

$$\min q(c_k, x),$$

tiene una solución, x_k. Tenemos el siguiente resultado.

Teorema 6.7 *Sea x_k una sucesión generada por el método del penalización. Entonces, cualquier punto límite de la sucesión es una solución del problema de minimización condicionado (6.22).*

Observemos que el problema (6.24) puede ser resuelto, por ejemplo, por el método de Newton. En la demostración de este teorema usaremos el siguiente resultado auxiliar.

Lema 1 *Para todo $k = 1,2,\ldots$, se tiene*

$$q(c_k,x_k) \leq q(c_{k+1},x_{k+1}), \tag{6.25}$$
$$P(x_k) \geq P(x_{k+1}), \tag{6.26}$$
$$f(x_k) \leq f(x_{k+1}). \tag{6.27}$$

Además, si x^* es una solución del problema condicionado (6.22) entonces

$$f(x^*) \geq q(c_k,x_k) \geq f(x_k). \tag{6.28}$$

Demostración. Tenemos

$$\begin{aligned} q(c_{k+1},x_{k+1}) &= f(x_{k+1}) + c_{k+1}P(x_{k+1}) \geq f(x_{k+1}) + c_kP(x_{k+1}) \\ &\geq f(x_k) + c_kP(x_k) = q(c_k,x_k), \end{aligned}$$

lo que demuestra (6.25). También tenemos

$$f(x_k) + c_kP(x_k) \leq f(x_{k+1}) + c_{k+1}P(x_{k+1}), \tag{6.29}$$
$$f(x_{k+1}) + c_{k+1}P(x_{k+1}) \leq f(x_k) + c_{k+1}P(x_k). \tag{6.30}$$

Sumando (6.29) y (6.30) obtenemos

$$(c_{k+1} - c_k)P(x_{k+1}) \leq (c_{k+1} - c_k)P(x_k),$$

que prueba (6.26). Además,

$$f(x_k) + c_kP(x_k) \leq f(x_{k+1}) + c_kP(x_{k+1}),$$

y usando (6.26) obtenemos (6.27). Finalmente, si x^* es solución de (6.22) entonces $P(x^*) = 0$, luego

$$f(x^*) = f(x^*) + c_kP(x^*) \geq f(x_k) + c_kP(x_k) \geq f(x_k),$$

que prueba (6.28).□

Demostración del Teorema 6.7. Supongamos que $\bar{x}$ es un punto límite de alguna subsucesión de x_k, que denotamos por $\bar{x}_k$. Por continuidad, tenemos

$$\lim_{k\to\infty} f(\bar{x}_k) = f(\bar{x}). \tag{6.31}$$

Sea M el valor mínimo asociado al problema (6.22). De acuerdo al Lema 1, la sucesión de valores $q(c_k,x_k)$ es no decreciente y acotada por M. Por tanto, existe un $q^* \in \mathbb{R}$ tal que

$$\lim_{k\to\infty} q(c_k,\bar{x}_k) = q^* \leq M. \tag{6.32}$$

Restando (6.31) de (6.32) obtenemos

$$\lim_{k\to\infty} c_kP(\bar{x}_k) = q^* - f(\bar{x}). \tag{6.33}$$

Como $P(\bar{x}_k) \geq 0$ y $c_k \to \infty$, (6.33) implica

$$\lim_{k\to\infty} P(\bar{x}_k) = 0.$$

Usando la continuidad de P, esto implica $P(\bar{x}) = 0$, luego $\bar{x}$ satisface la restricción $\bar{x} \in S$. Finalmente, usando (6.28) deducimos $f(\bar{x}_k) \leq M$, y entonces

$$f(\bar{x}) = \lim_{k\to\infty} f(\bar{x}_k) \leq M.$$

□

Apéndice. Algunas definiciones importantes

Sea $x \in \mathbb{R}^n$. La *norma Euclidea x* se define como

$$\|x\| = \Big(\sum_{i=1}^{n} x_i^2\Big)^{1/2},$$

y la ℓ^∞ *norma de x* viene dada por

$$\|x\|_\infty = \max_{i=1,\ldots,n} |x_i|.$$

Una *matriz cuadrada*, A, *de orden n* es una colección ordenada de números, $a_{ij} \in \mathbb{R}$, para $i, j = 1, \ldots, n$,

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}.$$

Cuando el orden de la matriz se puede deducir facilmente del contexto, se usa la notación $A = (a_{ij})$.

La *transpuesta* de A, que se escribe A^T, es una matriz que se obtiene intercambiando las filas y las columnas de A,

$$A^T = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{pmatrix}.$$

Una matriz cuadrada, A, es *simétrica* si $A = A^T$. Una matriz cuadrada, A, es *definida positiva* si A es simétrica y

$$x^T A x > 0 \quad \text{para tidi } x \in \mathbb{R}^n, \quad x \neq 0.$$

Si la desigualdad no es estricta, se dice que A es *semidefinida positiva*.

Las normas vectoriales inducen normas matriciales haciendo:

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

El *polinomio característico* de una matriz cuadrada de orden n es un polinomio, P, de grado n, que es el mismo para matrices semejantes,

$$P(\lambda) = \det(A - \lambda I_n),$$

donde I_n es la matriz de identidad de orden n. Las n raíces de el polinomio característico, λ_i, para $i = 1,\ldots,n$, se denominan *autovalores* de A, y pueden ser reales o complejas. Si A is simétrica, entonces $\lambda_i \in \mathbb{R}$, para todo $i = 1,\ldots,n$. Además, si A es definida positiva entonces $\lambda_i > 0$, para todo $i = 1,\ldots,n$. El *radio espectral*, ρ, de A viene dado por el máximo autovalor en valor absoluto

$$\rho = \max_{i=1,\ldots,n} |\lambda_i|.$$

Sea $\Omega \subset \mathbb{R}^n$ un conjunto abierto, y $f : \Omega \to \mathbb{R}$ una función derivable dos veces con derivada segunda continua. La *derivada parcial de f con respecto a x_i*, evaluada en el punto $x \in \Omega$, se escribe

$$\frac{\partial f}{\partial x_i}(x).$$

Las derivadas parciales de orden superior se definen por composición de derivadas parciales de primer orden. Por ejemplo

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x)$$

es la segunda derivada parcial de f con respecto a x_i y x_j, evaluada en x. Una propiedad importante de las derivadas parciales segundas es que son independientes del orden en que se deriva

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x) = \frac{\partial^2 f}{\partial x_j \partial x_i}(x). \tag{A.34}$$

El *gradiente* de f en x es el vector

$$\nabla f(x) = \Big(\frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_n}(x)\Big).$$

Las derivadas parciales de segundo orden de f se reunen en una matriz que se denomina *matriz Hessiana de f*,

$$H_f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{pmatrix}. \tag{A.35}$$

Dado que se verifica (A.34), la matriz Hessiana es simétrica. La traza de la matriz Hessiana de f, es decir. la suma de los elementos de la diagonal principal, se llama *Laplaciana de f* en x, y se escribe como $\Delta f(x)$.

$$\Delta f(x) = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}(x). \tag{A.36}$$

Decimos que un conjunto $\Omega \subset \mathbb{R}^n$ es *convexo* si para todo $x,\ y \in \Omega$, y para todo $t \in [0,1]$

$$tx + (1-t)y \in \Omega.$$

Una función $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ es *convexa en el conjunto convexo* Ω si, para todo $x, y \in \Omega$, y para todo $t \in [0,1]$,

$$f(tx + (1-t)y) \leq t f(x) + (1-t) f(y).$$

f se dice *estrictamente convexa* si esta desigualdad es estricta para todo $x \neq y$ y $t \in (0,1)$.

Si f es derivable dos veces con derivada segunda continua, entonces es convexa en el conjunto convexo, Ω, si y solo si $H_f(x)$ es semidefinida positiva para todo $x \in \Omega$.

Sea $\Omega \subset \mathbb{R}^n$ un conjunto abierto, y $f : \Omega \to \mathbb{R}^m$ una función vectorial, $\mathbf{f} = (f_1, \ldots, f_m)$, continuamente diferenciable. La *matriz Jacobiana* de $\mathbf{f}$ es la $m \times n$ matriz dada por

$$J_f(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \ldots & \frac{\partial f_1}{\partial \partial x_n}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \ldots & \frac{\partial f_2}{\partial x_n}(x) \\ \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial x_1}(x) & \ldots & \frac{\partial f_m}{\partial x_n}(x) \end{pmatrix}. \tag{A.37}$$

Si $m = n$ entonces la matriz Jacobiana $\mathbf{f}$ es una matriz cuadrada, cuyo determinante se denomina *determinante Jacobiano* de $\mathbf{f}$ en x, y se escribe $|J_f(x)|$. Además, si $m = n$, la traza de $J_f(x)$ se denomina, *divergencia de* $\mathbf{f}(x)$, y se escribe $\operatorname{div} f(x)$.

$$\operatorname{div} \mathbf{f}(x) = \sum_{i=1}^{n} \frac{\partial f_i}{\partial x_i}(x). \tag{A.38}$$

Para una función real $f : \Omega \to \mathbb{R}$, la composición del gradiente y la divergencia es la Laplaciana,

$$\Delta f(x) = \operatorname{div}\big(\nabla f(x)\big).$$

La expansión de Taylor es una herramienta útil que usaremos a menudo.

Teorema 1.8 (Taylor) *Sea f una función dos veces derivable con derivada segunda continua en un entorno del punto $x^* \in \mathbb{R}^n$. Entonces , para todo $e \in \mathbb{R}^n$ con $\|e\|$ suficientemente pequña se tiene que*

$$f(x^* + e) = f(x^*) + \nabla f(x^*)^T e + \frac{1}{2} e^T H_f(x) e + o(\|e\|^2). \tag{A.39}$$

Entorno de radio ρ centrado en x^* es el conjunto $B_\rho(x^*) = \{x \in \mathbb{R}^n : \|x - x^*\| < \rho\}$ (bola $n-$dimensional). La notación $o(t^2)$ (*o pequeña*) significa

$$\lim_{t \to 0} \frac{o(t^2)}{t^2} = 0.$$

Bibliografía

[1] Burden, R., Faires, J.D., Métodos numéricos, Paraninfo (2004).

[2] Chapra.S.C., Canale,R.P., Métodos numéricos para ingenieros, McGraw Hill (2007) .

[3] Luenberger, D.G., Programación lineal y no lineal, Addison-Wesley (1989).

[4] Quarteroni, A., Saleri, F., Cálculo Científico con MATLAB y Octave, Springer-Verlag (2006).