Análisis Numérico

L. Héctor Juárez V.
Departamento de Matemáticas,
Universidad Autónoma Metropolitana.
hect@xanum.uam.mx

3 de septiembre de 2008

Índice general

Prólogo

Este documento contiene las notas del curso de *Análisis Numérico* diseñado para los alumnos de la maestría en Ciencias Matemáticas Aplicadas e Industriales y del Posgrado de la División de Ciencias Básicas e Ingeniería de la Universidad Autónoma Metropolitana, plantel Iztapalapa. El material incluye los temas cubiertos en un periódo trimestral y equivalente a 50 horas frente a grupo. Se recomienda que un mínimo de 12 horas de las 50 se dediquen a taller de cómputo, en donde los alumnos experimenten con los diferentes algoritmos y métodos analizados en clase de teoría. Cabe aclarar que el material también puede utilizarse en cursos con estudiantes avanzados (del último año) de ciencias e ingenieria de nivel licenciatura. Los prerrequisitos son: cálculo en una y varias variables, algebra lineal, y ecuaciones diferenciales ordinarias. Los temas se escogieron para cubrir un mínimo de material que permita introducir al estudiante a este fascinante campo de la matemática aplicada, con el objeto de que adquiera la formación y madurez necesaria que le permita abordar otros temas más avanzados y/ó especializados del análisis numérico. Se hace énfasis en el análisis y comprensión de los métodos y algorimtos, así como de los alcances y limitaciones de los mismos. Nuestro propósito es que el estudiante no solo aprenda a utilizar los métodos, sino que además sea capaz de elejir y diseñar la mejor estrategia para algún problema en particular. Por el momento no hemos incluido los ejemplos cubiertos en taller de cómputo, los códigos de cómputo correspondientes, ni tampoco hemos incluido una lista de ejercicios de cada tema. Estas son tareas pendientes que iremos cubriendo poco a poco.

Por lo pronto, a manera de introducción trataremos de describir de manera general el campo de estudio del análisis numérico y el propósito general del material que se presenta en este manuscrito.

El análisis numérico trata del estudio, análisis y desarrollo de algoritmos para obtener soluciones numéricas y computacionales de problemas expresados matemáticamente. Generalmente estos problemas matemáticos son modelos expresados en el lenguaje matemático de diversos fenómenos estudiados en las ciencias naturales y problemas de la ingeniería, abarcando recientemente otros campos como las finanzas, la economía, la biología y la medicina, entre otros. A menudo el análisis numérico es considerado como las matemáticas de la computación científica (la cual versa sobre la solución numérica en computadora de problemas matemáticos).

Los algoritmos que estudiaremos invariablemente son diseñados para utilizarlos en computadoras de alta velocidad. Por tanto, un paso crucial antes de obtener la solución del problema consiste en el desarrollo de un código o programa en algún lenguaje de programación para comunicarlo a la computadora. Dado que hay varias opciones no solo de lenguajes de

computación sino tambien de tipos de computadoras, el tema de programación se deja fuera de la ciencia del análisis numérico. Sin embargo, en el presente estudio a menudo haremos uso del ambiente *MATLAB* para los cálculos en algunos ejemplos y aclaramos que el lector puede programar los métodos en el ambiente o en el lenguaje de programación que el elija.

Existen varias fuentes de error en la solución de problemas que requieren de computación matemática o solución computacional:

1. ***Modelación del proceso***. La solución numérica de problemas no puede ser mejor que el modelo subyacente. Un modelo matemático de un problema en particular siempre parte de hipótesis simplificadoras y generalmente no describe el fenómeno de manera exacta.

2. ***Medida de datos para el problema***. Los datos observados ó los parámetros asociados a un problema pocas veces son exactos. Cuando se obtienen datos de un problema generalmente hay limitaciones en los equipos de medición y los datos son obtenidos con una precisión finita determinada.

3. ***Errores matemáticos ó de método***. Estos están asociados a la aproximación matemática ó modelo numérico de solución del problema. Estos errores comunmente son llamados errores de discretización y errores de truncamiento, y son inherentes a la metodología de aproximación que se utiliza para resolver el problema.

4. ***Error de redondeo***. Aparecen debido al número finito de dígitos disponibles en las computadoras o calculadoras utilizadas para el cálculo numérico. Los equipos de cálculo ó computo tienen una precisión limitada y no pueden almacenar todos los números de manera exacta, ocasionando pérdida de precisión en los cálculos.

En este curso estudiaremos el diseño e implementación de códigos que en lo posible minimizen errores de redondeo, estimen errores de truncamiento y proporcione alguna indicación de la sensibilidad del problema a errores en los datos y a los errores de redondeo.

Capítulo 1

Aritmética Computacional

La aritmética realizada en una computadora involucra operaciones con números que tienen un número finito de dígitos solamente. Así que inevitablemente muchos cálculos se realizan con representaciones aproximadas de los números verdaderos. Por ejemplo $1/3$ no puede representarse de manera exacta en una computadora, al igual que $\pi, e, \sqrt{2}$ y muchos otros números.

1.1. Representación de números por computadora

En una computadora digital los números generalmente se representan en el sistema binario (base 2). Por ejemplo, $(110)_2$ equivale a $1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 6$ en el sistema decimal; $(1011{,}11)_2$ equivale a $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 11{,}75$ en el sistema decimal.

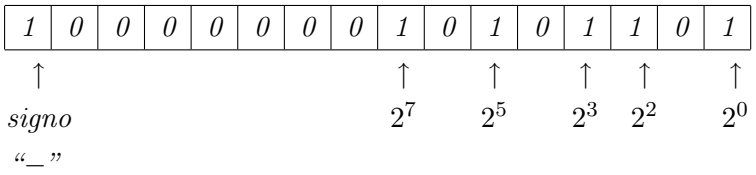
La longitud finita de la palabra de una computadora impone restricciones sobre la precisión que puede lograrse en la representación de los números. Por ejemplo, un número tan simple como $1/10$ no puede almacenarse exactamente en una máquina binaria dado que $1/10 = (0{,}0001100110011\ldots)_2$.

Además, debemos tomar en cuenta que tenemos dos técnicas de conversión relacionadas al comunicarnos con una computadora binaria: de decimal a binario (al introducir los datos), y de binario a decimal (al obtener los resultados). En cada conversión pueden presentarse errores.

1.1.1. Representación de números enteros

Los números enteros se pueden representar en una computadora utilizando el *método de la magnitud con signo*: El primer bit de la palabra se utiliza para indicar *el signo*: 0 para indicar "+", y 1 para indicar "−". Los bits sobrantes se usan para guardar *la magnitud del número.*

Ejemplo 1.1. *Si se usan 16 bits para representar los números enteros en la memoria de la computadora, el número* -173 *puede representarse por*



Ejemplo 1.2. ***Rango de enteros****. Determinar el rango de enteros que puede representarse en una computadora de 16 bits.*

Solución. El 1^{er} bit es para el signo. Sobran 15 bits con los cuales pueden representarse los números del 0 al $111111111111111 = 2^{15} - 1 = 32767$. Además como el 0 se representa por 0000000000000000, es redundante utilizar 1000000000000000 para representar -0, así que podemos representar un número adicional que puede ser -2^{15} ó 2^{15}. Por lo tanto, el rango de enteros que podemos representar con una computadora de 16 bits es de -32768 a 32767 ó bién de -32767 a 32768

1.1.2. Representación de punto flotante

Los números reales se representan en la computadora usando la *forma de punto flotante* (o notación científica normalizada). Esta consiste de una parte fraccionaria llamada *mantisa* y una parte entera llamada *exponente*, como se indica a continuación:

$$\pm \frac{m}{\beta^t} \beta^e = \pm m \beta^{e-t}$$

donde β es la *base del sistema* ($\beta = 10$ para el sistema decimal y $\beta = 2$ para el sistema binario), e representa *el exponente* y $t \geq 1$ es la p*recisión de la máquina.* La m se puede escoger como: $\frac{1}{\beta^t} \leq m < 1$, (la mantisa es m), ó $1 \leq m < \beta^t$, (la mantisa es $\frac{m}{\beta^t}$), dependiendo de la arquitectura.

Ejemplo 1.3. *La IBM 370 ó la 3000 consistía de un bit indicador del signo, un exponente e de 7 bits en base $\beta = 16$, y una mantisa m de 24 bits (en total 32 bits), con t=64. Cualquier número es de la forma: $m16^{e-64}$ con $\frac{1}{\beta^t} \leq m < 1$.*

En esta arquitectura el número de máquina

0	1 0 0 0 0 1 0	1 0 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0

representa el número decimal

$$[2^{-1} + 2^{-3} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-14}]16^{66-64} = 179{,}015625.$$

donde:

el primer bloque (de 1 bit) representa "+",
el segundo bloque (de 7 bits) representa $2^6 + 2^1 = 66$,
el tercer bloque (de 24 bits) representa $2^{-1} + 2^{-3} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-14}$.

El anterior número de máquina más pequeño es:

0	1 0 0 0 0 1 0	1 0 1 1 0 0 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1

$=179.0156097412109375$

El siguiente número de máquina más grande es:

0	1 0 0 0 0 1 0	1 0 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1

$= 179{,}0156402587890625$

Lo anterior significa que el número de máquina original debe representar no sólo a 179.015625, sino también a un número infinito de números reales que se encuentran entre este número y sus números de máquina más cercanos. Por otro lado, en esta arquitectura, el número más pequeño que se puede representar es:

0	0 0 0 0 0 0 0	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

$= 2^{-4} \cdot 16^{0-64} = 16^{-65} \approx 10^{-78}$

y el número más grande que se puede representar es:

0	1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

$\sim 1 \cdot 16^{127-64} = 16^{63} \approx 10^{76}$

de tal manera que los números que aparecen en los cálculos con magnitud menor a 16^{-65} resultan en lo que se denomina como *underflow*, y generalmente se le da el valor de cero. Los números con magnitudes mayores a 16^{63} resultan en lo que se llama *overflow*, y causa que los cálculos se detengan ó se obtenga un mensaje de error.

1.1.3. Limitaciónes en la representación de números en computadora

Del ejemplo anterior, concluimos que la limitación más fuerte para representar números en una computadora digital es que sólo se puede contar con un número finito de bits para cualquiera de las arquitecturas. De aquí que en una computadora digital

1. *Sólo se puede representar de manera exacta un número finito de números.*

2. *No se puede representar números arbitrariamente grandes o arbitrariamente pequeños.*

3. *Existen vacios ó claros entre los números con representación exacta.*

Afortunadamente en las computadoras modernas se pueden representar números suficientemente grandes y números suficientemente pequeños (desde un punto de vista práctico). Por ejemplo, en la aritmética más ampliamente usada en la actualidad (*IEEE, Institute of Electric and Electronic Engineers*) en doble precisión permite números tan grandes como $1{,}79 \times 10^{308} (= 2^{1024})$ y números tan pequeños como $2{,}23 \times 10^{-308} (= 2^{-1022})$. Sin embargo el vacío, claro o brecha entre números de máquina es un problema permanente en la computación. Por ejemplo, el intervalo [1,2] se representa por el conjunto discreto y finito, 1, $1 + 2^{-52}$, $1 + 2 \times 2^{-52}$, $1 + 3 \times 2^{-52}$, $\ldots$, $1 + 2^{52} \times 2^{-52} = 2$.

En la *aritmética IEEE de doble precisión* los claros o vacíos entre números de máquina adyacentes no son mayores a $2^{-52} \approx 2{,}22 \times 10^{-16}$, en *sentido relativo*. Esto puede parecer insignificante, y de hecho lo es para la mayoría de algoritmos "*estables*". Sin embargo, resulta catastrófico para los algoritmos "*inestables*". Más adelante precisaremos los conceptos de *estabilidad e inestabilidad numérica.*

En suma, los números de punto flotante en una computadora binaria están distribuidos en forma no uniforme, la mayoría de ellos concentrados cerca de cero. Esto es debido a que entre potencias adyacentes de 2 (intervalos binarios) siempre hay la misma cantidad de números de máquina e igual a 2^t (con t=52 para doble precisión). Por ejemplo en el intervalo $2^e \leq x \leq 2^{e+1}$ los números están igualmente espaciados con un incremento $\dfrac{2^e}{2^t}$, donde $2^e =$ *longitud del intervalo* y $2^t =$ *cantidad de números de punto flotante en el intervalo binario.* Por lo tanto, hay una distribución no uniforme de punto flotante, con una densidad mayor cerca del origen (es decir, el incremento $\dfrac{2^e}{2^t}$ es cada vez menor conforme el exponente e es más negativo).

1.2. Errores de redondeo

Cuando se realiza aritmética de punto flotante frecuentemente el número de dígitos en los resultados intermedios es cada vez mayor, mientras que el número significativo de dígitos permanece fijo. Por ejemplo, el producto de dos números de punto flotante en aritmética con 8 dígitos después del punto decimal será un número con 16 dígitos después del punto decimal, y debemos quitar ocho dígitos para redondear el resultado a 8 dígitos después del punto decimal. El redondeo es un concepto útil y fundamental en la ciencia computacional.

1.2.1. Forma normalizada de un número

Para introducir e ilustrar el concepto de redondeo utilizaremos la *forma decimal normalizada*, esto con el objeto de simplificar la exposición. Si x es un número real cualquiera, entonces tiene una representación decimal exacta de la forma $x = \pm 0.d_1 d_2 d_3 \cdots d_t d_{t+1} \cdots \times 10^n$, con $1 \leq d_1 \leq 9$, $0 \leq d_i \leq 9$, para $i = 1, 2, \ldots, t, t+1, \ldots$, y n un entero positivo o negativo.

Ejemplo 1.4. *La forma decimal normalizada se ilustra con los siguientes números fraccionarios:*

$$x = -\frac{1}{11} = -0{,}909090\ldots \times 10^{-1},$$
$$x = \frac{15}{11} = 0{,}13636363\ldots \times 10^{1}.$$

1.2.2. Forma de punto flotante: truncamiento y redondeo

La *forma de punto flotante*, denotada por $fl(x)$, se obtiene terminando la mantisa de x en t dígitos decimales, si suponemos que los números de máquina admiten a lo más t dígitos decimales después del punto decimal. Existen dos maneras de realizar esta terminación

1. **Truncamiento**. Se cortan los dígitos $d_{t+1}, d_{t+2}, \ldots$ para obtener $fl(x) = \pm 0.d_1 d_2 \cdots d_t \times 10^n$.

2. **Redondeo**. Se forma la suma $d_1 d_2 \cdots d_t.d_{t+1} + 0{,}5$ (donde el punto que está entre d_t y d_{t+1} es un punto decimal). La parte decimal de $fl(x)$ es $d_1 d_2 \cdots d_{t-1}\delta_t$ la cual representa la parte entera del resultado de la suma anterior.

Ejemplo 1.5. *La representación del número $x = 15/11$ en punto flotante con truncamiento*

a seis y siete cifras decimales es:

$$fl(x) = 0{,}136363 \times 10^1, \quad \textit{si } t = 6,$$
$$fl(x) = 0{,}1363636 \times 10^1, \quad \textit{si } t = 7.$$

Por otro lado, para obtener su representación con redondeo a seis y siete cifras decimales se hace lo siguiente: Si se toman seis cifras decimales ($t = 6$)*, entonces,* $136363{,}6 + 0{,}5 = 136364{,}1$*, cuya parte entera es 136364. Por lo tanto* $fl(x) = 0{,}136364 \times 10^1$ *(con* $\delta_6 = 4 \neq d_6 = 3$ *). Si se toman siete cifras decimales* ($t = 7$)*, entonces tenemos* $1363636{,}3 + 0{,}5 = 1363636{,}8$*, cuya parte entera es 1363636. Por lo tanto* $fl(x) = 0{,}1363636 \times 10^1$ *(con* $\delta_7 = 6 = d_7$*).*

El error que resulta al reemplazar un número real x por su número de punto flotante se denomina ***error de redondeo***. Este error esta acotado por $|x - fl(x)| \leq \frac{1}{2}10^{1-t}|x|p$, donde $p = 1$ cuando se hace redondeo, y $p = 2$ cuando se hace truncamiento.

Demostración. (Caso por truncamiento). Sea $x = 0.d_1d_2\cdots d_td_{t+1}\cdots \times 10^n$, $fl(x) = 0.d_1d_2\cdots d_t \times 10^n$, entonces

$$\begin{aligned}
|x - fl(x)| &= 0.\underbrace{0\cdots 0}_{t-ceros} d_{t+1}d_{t+2}\cdots \times 10^n \\
&= 0.d_{t+1}d_{t+2}\cdots \times 10^{n-t} \\
&= 0.d_{t+1}d_{t+2}\cdots \times 10^n \times 10^{-t} \times \frac{|x|}{|x|} \\
&= \frac{0.d_{t+1}d_{t+2}\cdots \times 10^n \times 10^{-t} \times |x|}{0.d_1d_2\cdots d_td_{t+1}\cdots \times 10^n}
\end{aligned}$$

Pero

$$\frac{0.d_{t+1}d_{t+2}\cdots}{0.d_1d_2\cdots} \leq 10$$

ya que

$$0.d_{t+1}d_{t+2}\cdots \leq d_1.d_2d_3\cdots$$

(donde el punto entre d_1 y d_2 es un punto decimal). Por lo tanto

$$|x - fl(x)| \leq 10^{1-t}|x|$$

cuando se hace truncamiento. Análogamente puede considerarse el caso de redondeo:

$$|x - fl(x)| \leq \frac{1}{2}10^{1-t}|x|.$$

Observación. En las expresiones anteriores la cantidad $|x - fl(x)|$ denota el *error absoluto* que se obtiene al representar el número x por su número de punto flotante, mientras que la cantidad $|x - fl(x)|/|x|$ denota el *error relativo* que se obtiene al reemplazar x por su correspondiente número de máquina o punto flotante $fl(x)$. Luego entonces el error relativo será menor a 10^{1-t} cuando se hace truncamiento y menor a $\frac{1}{2}10^{1-t}$ cuando se hace redondeo para representar los números en computadora. En el caso más general cuando la base es β en lugar de 10, se tiene que

$$|x - fl(x)| \leq \beta^{1-t}|x|, \tag{1.1}$$

cuando se hace truncamiento y

$$|x - fl(x)| \leq \frac{1}{2}\beta^{1-t}|x|, \tag{1.2}$$

cuando se hace redondeo.

1.2.3. Epsilon de máquina

La resolución del conjunto de números de punto flotante ó números de máquina para representar los números reales se establece a través del número conocido como *epsilon de máqina*, que se define como:

$$\epsilon_{maq} = \frac{1}{2}\beta^{1-t} \tag{1.3}$$

el cual representa la mitad de la distancia del número 1 a su número de máquina mayor más próximo. Algunos autores lo definen como

$$\epsilon_{maq} = \beta^{1-t}$$

pero en realidad lo que realmente importa es la propiedad siguiente de este número:

Propiedad del epsilon de máquina.

Para cada número real x, existe un número de punto flotante x' tal que

$$|x - x'| \leq \epsilon_{maq}|x|.$$

En realidad, comparando con las expresiones para el error de truncamiento y redondeo, encontramos que ésta es una forma de expresar que para cada número real x, existe ϵ con $|\epsilon| \leq \epsilon_{maq}$ tal que

$$fl(x) = x(1 + \epsilon). \tag{1.4}$$

Es decir, la diferencia entre un número real cualquiera y el número de punto flotante más cercano a él es siempre menor o igual a ϵ_{maq} en sentido relativo. Para los valores de β y t que se han usado en la mayoría de las computadoras el epsilon de máqina se encuentra en el rango de $10^{-35} \leq \epsilon_{maq} \leq 10^{-6}$. En aritmética IEEE de precisión simple y doble, el epsilon de máquina es $2^{-23} \approx 1{,}22 \times 10^{-7}$ y $2^{-52} \approx 2{,}2 \times 10^{-16}$, respectivamente.

1.3. Aritmética de punto flotante

1.3.1. Sistema de números de punto flotante

Representemos por $\mathbb{F}$ el sistema de números de punto flotante definidos por la base $\beta \geq 2$ y la precisión $t \geq 1$ (recordemos que $\beta = 2$, $t = 23$ para precisión simple y $\beta = 2$, $t = 52$ para precisión doble en el estandard IEEE). Entonces podemos escribir

$$\mathbb{F} = \{x \in \mathbb{R} : x = \pm \frac{m}{\beta^t}\beta^e, \quad \frac{1}{\beta^t} \leq m < 1, \quad e_{min} \leq e \leq e_{max}\}. \tag{1.5}$$

Recordemos que el epsilon de máquina se define por $\frac{1}{2}\beta^{1-t}$. A continuación presentamos una tabla de valores para diferentes arquitecturas (la mayoría de ellas de la década de los 80).

MÁQUINA	β	t	min e	max e	ϵ_{maq}
VAX(simple)	2	23	-128	127	$1{,}2 \times 10^{-7}$
VAX(doble)	2	55	-128	127	$2{,}8 \times 10^{-17}$
CRAY-1	2	48	-16384	16383	$3{,}6 \times 10^{-15}$
IBM-3081(simple)	16	6	-64	63	$9{,}5 \times 10^{-7}$
IBM-3081(doble)	16	14	-64	63	$2{,}2 \times 10^{-16}$

IEEE estadard	β	t	min e	max e	ϵ_{maq}
Simple	2	23	-126	127	$1{,}2 \times 10^{-7}$
Doble	2	52	-1022	1023	$2{,}2 \times 10^{-16}$
Cuadruple	2	112	-16382	16383	$1{,}9 \times 10^{-34}$

1.3.2. Operaciones de punto flotante

Los cálculos matemáticos generalmente se reducen a ciertas operaciones matemáticas elementales, de las cuales el conjunto clásico es $+,-,\times,\div$. Matemáticamente estos símbolos representan operaciones en $\mathbb{R}$. Sobre una máquina digital estas operaciones tienen análogos

llamadas operaciones de punto flotante sobre $\mathbb{F}$. Para distingir las operaciones aritméticas usuales de las operaciones de punto flotante, a estas últimas se les denota por los símbolos $\oplus, \ominus, \otimes, \oslash,$, respectivamente.

1.3.3. Axioma fundamental de la arimética de punto flotante

Una computadora puede construirse sobre el siguiente principio de diseño: Si x y y son dos números de punto flotante cualesquiera (x, $y \in \mathbb{F}$) y $*$ es una de las posibles operaciones $+$, $-$, $\times$, $\div$, con $\circledast$ su análogo de punto flotante, entonces

$$x \circledast y = fl(x * y) \qquad \textbf{(Propiedad de diseño)}. \tag{1.6}$$

Por la propiedad (1.4) del epsilon de máquina, encontramos que

$$fl(x * y) = x * y(1 + \epsilon) \quad \text{con} \quad |\epsilon| \leq \epsilon_{maq}.$$

Por lo tanto, concluimos que la computadora tiene una propiedad sencilla pero muy útil, expresada en el siguiente axioma, denominado el **Axioma fundamental de la arimética de punto flotante**:

Para todo $x, y \in \mathbb{F}$ existe ϵ con $|\epsilon| \leq \epsilon_{maq}$ tal que

$$x \circledast y = (x * y)(1 + \epsilon). \tag{1.7}$$

E*s decir, cada operación en la aritmética de punto flotante es exacta salvo un error relativo de a lo más ϵ_{maq}.*

Por lo tanto, si x, y son números de punto flotante

$$\begin{array}{l} x \oplus y = (x + y)(1 + \epsilon) \\ x \ominus y = (x - y)(1 + \epsilon) \\ x \otimes y = (x \times y)(1 + \epsilon) \\ x \oslash y = (x \div y)(1 + \epsilon) \end{array} \quad \text{para algún } \epsilon, \text{ con } |\epsilon| \leq \epsilon_{maq}.$$

siempre y cuando los resultados se mantengan dentro del sistema de punto flotante $\mathbb{F}$.

Algunas de las propiedades fundamentales de las operaciones en el sistema de números reales son válidas para las operaciones en el sistema de punto flotante, otras no lo son. Por ejemplo, las leyes conmutativa de suma y multiplicación si son válidas en punto flotante:

$$x \oplus y = y \oplus x, \qquad x \otimes y = y \otimes x$$

Sin embargo, la ley asociativa de la multiplicación no es válida en el sistema de punto flotante

$$(x \otimes y) \otimes z \neq x \otimes (y \otimes z).$$

Por ejemplo, si $x = \dfrac{\epsilon_{maq}}{3}$, $y = 3\epsilon_{maq}$, $z = \dfrac{\epsilon_{maq}}{11}$, entonces en el sistema de punto flotante IEEE de doble precisión, obtenemos en el ambiente MATLAB

$$x \otimes (y \otimes z) = 9{,}952403865943302 \times 10^{-49}$$

$$(x \otimes y) \otimes z = 9{,}952403865943303 \times 10^{-49}$$

y se observa una diferencia en el último dígito. La situación es más catastrófica con la suma y la resta. Puede suceder fácilmente que $(x \oplus y) \oplus z$ sea muy diferente a $x \oplus (y \oplus z)$ con aritmética de baja precisión. De la misma manera la ley distributiva $x(y + z) = xy + xz$ no es válida en aritmética de punto flotante.

Capítulo 2

Condicionamiento y Estabilidad

2.1. Normas de matrices

El objeto de esta sección es introducir el concepto de norma matricial. Este concepto es de fundamental importancia no solo en el análisis y el algebra lineal y sus aplicaciones, sino tambien en el análisis numérico. La razón por lo que se ha decidido introducirlo en esta parte del escrito es porque será muy útil a partir de la siguiente sección, especialmente para ilustrar los conceptos de condicionamiento y estabilidad.

2.1.1. Normas en espacios vectoriales

Sea V un espacio vectorial real. Una ***norma*** en V es una función no negativa $\|\cdot\|: V \to \mathbb{R}$ que satisface

1. $\|x\| \geq 0\,, \forall\, x \in \mathbb{R}\,, \quad$ y $\quad \|x\| = 0 \quad$ si y solo si $\quad x = 0$
2. $\|\alpha x\| = |\alpha| \|x\|, \quad \forall\, \alpha \in \mathbb{R}, \quad \forall\, x \in V$
3. $\|x + y\| \leq \|x\| + \|y\|\,, \quad \forall\, x, y \in V \quad$ (desigualdad del triángulo)

Con $V = \mathbb{R}^n$ (espacio de vectores n–dimensional) tenemos varias normas denominadas *normas–p*. Sea $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, entonces definimos las siguientes normas:

1. **norma–1**: $\quad \|x\|_1 = \sum_{i=1}^{n} |x_i|$
2. **norma–2**: $\quad \|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{\frac{1}{2}}$
3. **norma–p** : $\quad \|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}} \quad (p \geq 1)$

4. **norma**–∞ : $\|x\|_\infty = \text{máx}_{1\leq i\leq n}\,|x_i|$ (también llamada norma del máximo o uniforme)

La Figura 2.1 muestra la bolas unitarias bidimensionales ($n = 2$) en cada una de las normas anteriores. Por la *bola unitaria* en una norma $\|\cdot\|$ entendemos el conjunto $\{x \in \mathbb{R}^n : \|x\| \leq 1\}$. Obviamente la bola unitaria en la norma euclideana es esférica. Sin embargo, la bola unitaria en las otras normas tienen otra forma. A medida que p aumenta la bola unitaria se tranforma del diamante ó rombo al cuadrado, pasando por el círculo.

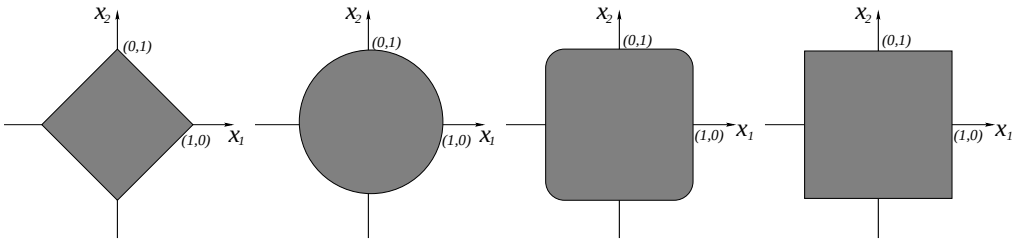


Figura 2.1: Bolas unitarias en la normas–1, 2, p, ∞, respectivamente.

Aparte de las normas–p, las normas más útiles en las aplicaciones son las *normas–p pesadas*, en donde cada una de las coordenadas del vector tiene su propio peso. Dada la norma $\|\cdot\|$, una norma con peso puede escribirse como

$$\|x\|_W = \|Wx\|$$

donde $W = \text{diagonal}(w_1, w_2, \ldots, w_n)$ con $w_i \neq 0$. Se puede generalizar el concepto de normas pesadas permitiendo que W sea una matriz arbitraria no singular, no necesariamente diagonal.

No es complicado verificar que las anteriores normas efectivamente satisfacen las propiedades 1, 2 y 3 en la definición de norma. Para cada una de estas normas las propiedades 1 y 2 se satisfacen trivialmente. Por otro lado, es fácil verificar la *desigualdad del triángulo* para la norma–1 y la norma–∞, pero para la norma euclideana o norma–2 es necesario hacer uso de la ***desigualdad de Cauchy–Schwartz***

$$\left|\sum_{i=1}^{n} x_i\, y_i\right| \leq \|x\|_2\, \|y\|_2 \qquad x = (x_1, \ldots, x_n)^T,\ y = (y_1, \ldots, y_n)^T\,.$$

A la desigualdad del triángulo para las normas–p

$$\|x + y\|_p \leq \|x\|_p + \|y\|_p$$

se le conoce como ***desigualdad de Minkowski*** y su demostración se encuentra en libros de calculo avanzado ó análisis matemático.

2.1.2. Continuidad y equivalencia de normas

Continuidad de la normas: Cualquier norma vectorial $\|\cdot\|$ en $\mathbb{R}^n$ es una función continua de $x \in \mathbb{R}^n$ (ver Isaacson y Keller).

Equivalencias de normas: Cualquier par de normas $\|\cdot\|,\ \|\cdot\|'$ en $\mathbb{R}^n$ son equivalentes. Es decir, para toda $x \in \mathbb{R}^n$ existen constantes positivas m y M tales que

$$m\|x\|' \leq \|x\| \leq M\|x\|', \quad \forall\, x \in \mathbb{R}^n\,.$$

Demostración. Es sufuciente con verificar el resultado cuando una de las normas es $\|\cdot\|_\infty$, ya que $\|\cdot\|$ y $\|\cdot\|'$ serán equivalentes si cada una de ellas es equivalente a $\|\cdot\|_\infty$. Consideremos la esfera unitaria en la norma ∞:

$$S_\infty = \{x \in \mathbb{R}^n \ :\ \|x\|_\infty = 1\}.$$

Este es un conjunto cerrado y acotado en $\mathbb{R}^n$. Dado que cualquier norma vectorial $\|\cdot\|$ es continua en $\mathbb{R}^n$, y en particular sobre S_∞, entonces toma sus valores máximo y mínimo dentro de S_∞. Sean entonces x^0 y x^1 los puntos sobre la esfera donde se toman el mínimo y el máximo, respectivemente:

$$\min_{x\in S_\infty} \|x\| = \|x^0\|, \quad \max_{x\in S_\infty} \|x\| = \|x^1\|.$$

Entonces

$$\|x^0\| \leq \|x\| \leq \|x^1\|, \quad \forall\, x \in S_\infty\,.$$

Ahora considermos un vector arbitrario $y \in \mathbb{R}^n\,,\ y \neq \vec{0}$, donde $\vec{0}$ denota al vector cero en $\mathbb{R}^n$. Como el vector normalizado $\dfrac{y}{\|y\|_\infty}$ se encuentra en S_∞, entonces

$$\|x^0\| \leq \left\| \frac{y}{\|y\|_\infty} \right\| \leq \|x^1\|.$$

Es decir,

$$\|x^0\|\|y\|_\infty \leq \|y\| \leq \|x^1\|\|y\|_\infty.$$

Por lo tanto,

$$m\|y\|_\infty \leq \|y\| \leq M\|y\|_\infty\,, \quad \forall\ y \in \mathbb{R}^n$$

con $m = \|x^0\|$ y $M = \|x^1\|$. Así que cualquier norma $\|\cdot\|$ en $\mathbb{R}^n$ es equivalente a la norma $\|\,\cdot\,\|_\infty$. Se concluye que cualesquier dos normas en $\mathbb{R}^n$ son equivalentes.

2.1.3. Normas matriciales inducidas

Cuando se hace análisis numérico de problemas donde intervienen matrices es útil tener el concepto de norma de una matriz y, en consecuencia, de la distancia entre matrices. Existe una forma geométrica (y aparentemente natural) que permite definir la norma de una matriz. Esta forma toma en cuenta el comportamieto de la matriz como un operador:

Si $A \in \mathbb{R}^{n\times n}$ es una matriz de $n \times n$, y $\|\cdot\|$ una norma vectorial en $\mathbb{R}^n$, entonces la norma de A inducida por esta norma vectorial se define por:

$$\|A\| := \sup_{x\neq\vec{0}} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\| \tag{2.1}$$

Nótese que estamos designando con el símbolo $\|\cdot\|$ tanto la norma vectorial como la norma matricial inducida. Sin embargo, no debe haber confusión cuando se trate de una ú otra, dado que el argumento será un vector ó una matriz y claramente indicará de que caso se trata. De la definición se infiere que para calcular la norma de una matriz dada A, basta con calcular el supremo de Ax sobre la esfera unitaria en la norma correspondiente.

Ejemplo 2.1. *La matriz*

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}$$

mapea $\mathbb{R}^2$ en $\mathbb{R}^2$ por medio de la transformación $\mathcal{T} : \mathbb{R}^2 \to \mathbb{R}^2$, $\mathcal{T}x = Ax$, $x = (x_1, x_2)^T$ (la T aquí indica la transpuesta, pues consideraremos que los vectores son vectores columna). Trataremos de ver cuál es la acción de A sobre bolas unitarias de $\mathbb{R}^2$ en las normas–1, 2, ∞: Sean $e_1 = (1, 0)^T$ y $e_2 = (0, 1)^T$ los vectores unitarios canónicos en $\mathbb{R}^2$. Entonces, $Ae_1 = (1, 0)^T$ y $Ae_2 = (2, 2)^T$.

En la norma–1, $\|Ae_1\|_1 = 1$ y $\|Ae_2\|_1 = 4$, luego el factor de amplificación es 4. Esta situación se ilustra en la Figura 2.2

En la norma–∞ el vector unitario x que tiene mayor amplificación bajo A es $x = (1, 1)^T$ ó su negativo, y el factor de amplificación es $\|Ax\|_\infty = \|(3, 2)^T\|_\infty = 3$. La Figura 2.3 muestra este caso.

*En la norma–2 (*norma Euclideana*) el vector unitario que es mayormente amplificado por A es el que se encuentra punteado en la figura de abajo (o su negativo), y el factor de amplificación es aproximadamente 2.9208, como se muestra en la Figura 2.4*

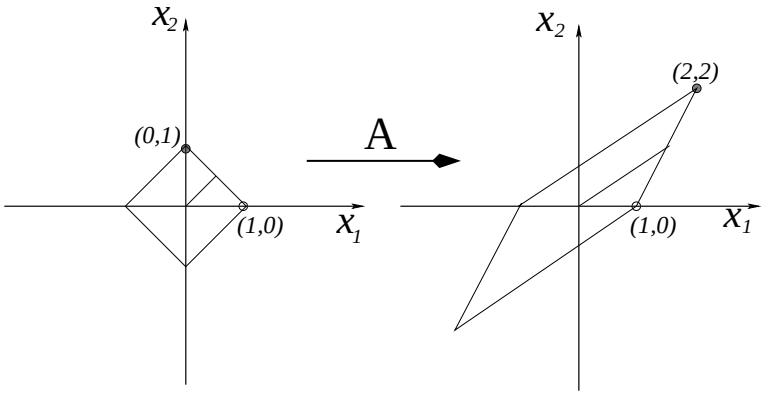


Figura 2.2: Norma–1, $\|A\|_1 = 4$.

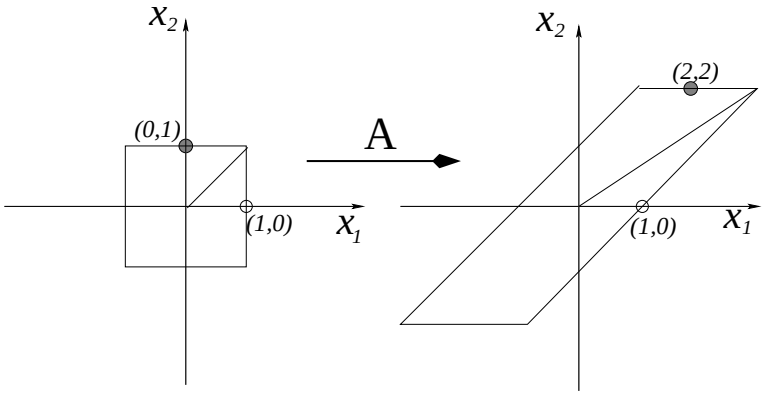


Figura 2.3: Norma–∞ $\|A\|_\infty = 3$.

2.1.4. Cálculo de normas matriciales

A continuación daremos fórmulas para calcular las normas matriciales inducidas por las normas–1, 2, ∞.

La norma–1 de una matriz: Si A es cualquier matriz de $n \times n$ entonces $\|A\|_1$ es igual a la *máxima suma por columnas* (de los coeficientes en valor absoluto) de A.

Demostración. Sea $A = [\vec{a}_1 \,|\, \vec{a}_2 \,|\, \ldots \,|\, \vec{a}_n]$, donde $\vec{a}_i$ es un vector columna en $\mathbb{R}^n$. Consideremos la bola unitaria en la norma–1: $\{x \in \mathbb{R}^n : \sum_{j=1}^n |x_j| \leq 1\}$. Cualquier vector Ax en la imagen de este conjunto satisface:

$$\|Ax\|_1 = \left\| \sum_{j=1}^n x_j \vec{a}_j \right\|_1 \leq \sum_{j=1}^n \|x_j \vec{a}_j\|_1 = \sum_{j=1}^n |x_j| \|\vec{a}_j\|_1 \leq \max_{1 \leq j \leq n} \|\vec{a}_j\|_1 \sum_{j=1}^n |x_j|$$

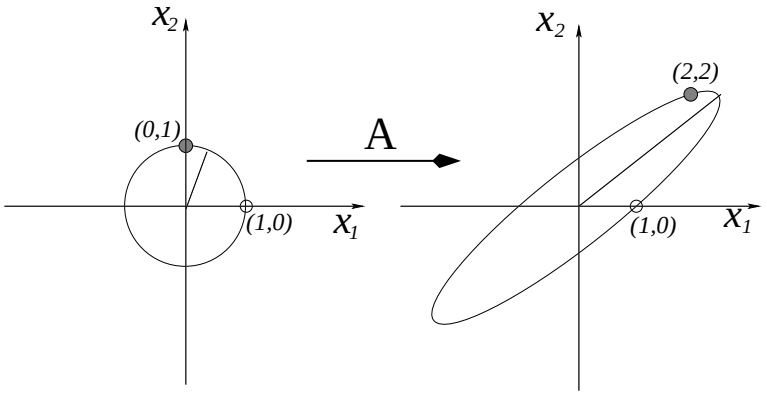


Figura 2.4: Norma–2 (Euclideana) $\|A\|_2 = 2{,}9208$

$$\leq \max_{1\leq j\leq n} \|\vec{a}_j\|_1 \qquad \Rightarrow \qquad \|Ax\|_1 \leq \max_{1\leq j\leq n} \|\vec{a}_j\|_1$$

Escogiendo $x = e_j = (0, \ldots, 1, \ldots, 0)$ (el 1 esta en la j–esima entrada), donde j maximiza $\|a_j\|_1$, obtenemos la cota máxima. Por lo tanto

$$\|A\|_1 = \max_{1\leq j\leq n} \|\vec{a}_j\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}| \qquad \text{(máxima suma por columna).} \tag{2.2}$$

La norma–∞ de una matriz: Utilizando un argumento muy similar al anterior, puede demostrarse que la norma-∞ de una matriz $A \in \mathbb{R}^{n\times n}$ es igual a la "máxima suma por renglón"

$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}| \qquad \text{(máxima suma por renglon).} \tag{2.3}$$

Ejemplo 2.2. *Dada la matriz*

$$A = \begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & -2 \end{bmatrix},$$

encontrar $\|A\|_1$ *y* $\|A\|_\infty$.

Solución. $\|A\|_1$ = máxima suma por columnas (en valor absoluto) = 3; mientras que $\|A\|_\infty$ = máxima suma por renglón (en valor absoluto) =4.

2.1.5. La norma–2 de una matriz

En este caso es más complicado hacer el cálculo. Para ello necesitamos los conceptos de valor propio, vector propio, así como del radio espectral de una matriz. Por ello revisaremos

brevemente estos conceptos antes de intentar encontrar la fórmula general.

Valores y vectores propios de una matriz. Dada una matriz A de $n \times n$ se dice que $\lambda \in \mathbb{C}$ es un *valor propio* de la matriz si existe $x \in \mathbb{R}^n$ tal que

$$x \neq \vec{0} \qquad y \qquad Ax = \lambda x,$$

y en este caso a x se le conoce como el *vector propio* asociado a λ. La condición "hay algún $x \neq \vec{0}$ tal que $Ax = \lambda x$", es equivalente a cualquiera de las siguientes dos condiciones:

1. La matriz $A - \lambda I$ es singular, donde I denota a la matriz identidad.
2. $det[A - \lambda I] = 0$.

Este determinante en la segunda condición es un polinomio en λ que se denomina *polinomio caracteristico de* A, y que denotaremos por $p(\lambda)$. Si A es de orden $n \times n$, entonces $p(\lambda)$ es de grado $\leq n$, y sus raices proporcionan los valores propios de la matriz.

Ejemplo 2.3. *Dada la matriz*

$$A = \begin{bmatrix} -1 & 0 & 1 \\ 2 & 0 & 1 \\ -1 & 0 & -2 \end{bmatrix}$$

su polinomio característico es

$$\begin{aligned} p(\lambda) &= det[A - \lambda I] = \begin{vmatrix} -1-\lambda & 0 & 1 \\ 2 & -\lambda & 1 \\ -1 & 0 & -2-\lambda \end{vmatrix} \\ &= (-1-\lambda)(-\lambda)(-2-\lambda) - \lambda \\ &= -\lambda[(\lambda+1)(\lambda+2)+1] = -\lambda[\lambda^2 + 3\lambda + 3]\,, \end{aligned}$$

y sus valores propios son la raices del polinomio: $\lambda_1 = 0$, $\lambda_2 = \dfrac{-3+\sqrt{3}i}{2}$, $\lambda_3 = \dfrac{-3-\sqrt{3}i}{2}$.

Radio espectral de una matriz. Dada una matriz A de $n \times n$, el *radio espectral* de A se define como el máximo de las magnitudes de sus valores propios, y se denota por $\rho(A)$, es decir,

$$\rho(A) = \text{máx}\{\ |\lambda| : \lambda \text{ es valor propio de } A\ \}.$$

Propiedad sobre los vectores propios de una matriz simétrica. Cualquier matriz simétrica A de $n \times n$ tiene un conjunto completo de vectores propios y este conjunto es ortogonal. Es decir, si $A = A^T$, entonces A tiene n vectores propios $\vec{u}_1, \ldots, \vec{u}_n \in \mathbb{R}^n$ tal que

$$\vec{u}_i^T \vec{u}_j = \delta_{ij} = \begin{cases} 1 & \text{si} \quad i = j\,, \\ 0 & \text{si} \quad i \neq j\,. \end{cases}$$

La norma-2 de una matriz: Si A es cualquier matriz de $n \times n$, entonces

$$\|A\|_2 = \sqrt{\rho(A^T A)}. \tag{2.4}$$

Demostración. Observese que $\|Ax\|_2^2 = (Ax)^T(Ax) = x^T A^T A x$. Pero la matriz $A^T A$ es una matriz simétrica y, por lo tanto, tiene un conjunto de n vectores propios $\vec{u}_1, \ldots, \vec{u}_n$ que generan $\mathbb{R}^n$ tales que

$$\vec{u}_i^T \vec{u}_j = \delta_{ij} \quad \text{y} \quad A^T A \vec{u}_i = \lambda_i \vec{u}_i, \qquad i,\, j = 1, \ldots, n.$$

Además

$$\|A\vec{u}_i\|_2^2 = (A\vec{u}_i)^T(A\vec{u}_i) = \vec{u}_i^T A^T A \vec{u}_i = \lambda_i \vec{u}_i^T \vec{u}_i = \lambda_i \geq = 0 \quad \forall i = 1, \ldots, n.$$

Dado que $x \in \mathbb{R}^n$ y los vectores propios forman una base, entonces $x = \sum_{i=1}^n \alpha_i \vec{u}_i$ para algunas constantes α_i, $i = 1, \ldots, n$, entonces

$$\begin{aligned} \|Ax\|_2^2 &= (Ax)^T(Ax) = x^T A^T A x = (\textstyle\sum_{i=1}^n \alpha_i \vec{u}_i)^T A^T A \sum_{i=1}^n \alpha_i \vec{u}_i \\ &= \textstyle\sum_{i=1}^n \alpha_i \vec{u}_i^T \sum_{i=1}^n \alpha_i A^T A \vec{u}_i = \sum_{i=1}^n \alpha_i \vec{u}_i^T \sum_{i=1}^n \alpha_i \lambda_i \vec{u}_i \\ &= \textstyle\sum_{i=1}^n \alpha_i^2 \lambda_i \leq \text{máx}\, \lambda_i \sum_{i=1}^n \alpha_i^2 \end{aligned}$$

Tomando $\|x\|_2 = 1$, se tiene que $\|x\|_2^2 = x^T x = \sum_{i=1}^n \alpha_i^2 = 1$. Así que

$$\|Ax\|_2^2 \leq \text{máx}\, \lambda_i = \rho(A^T A) \quad \Rightarrow \quad \|A\|_2 \leq \sqrt{\rho(A^T A)}$$

Para obtener la igualdad basta con encontrar algún $x \in \mathbb{R}$ con $\|x\|_2 = 1$ tal que $\|Ax\|_2^2 = \text{máx}\, \lambda_i$. Sea $x = \vec{u}_s$ el vector propio de $A^T A$ correspondiente al mayor valor propio, es decir, $\lambda_s = \text{máx}\, \lambda_i$. Entonces, claramente $\|\vec{u}_s\|_2 = 1$, y además

$$\begin{aligned} \|A\vec{u}_s\|_2^2 = (A\vec{u}_s)^T A \vec{u}_s = \vec{u}_s^T A^T A \vec{u}_s = \vec{u}_s^T \lambda_s \vec{u}_s = \lambda_s \|\vec{u}_s\|_2^2 \\ = \lambda_s = \text{máx}\, \lambda_i = \rho(A^T A). \end{aligned}$$

Concluimos que $\|A\|_2 = \sqrt{\rho(A^T A)}$.

Ejemplo 2.4. *Dada la matriz*

$$A = \begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & -2 \end{bmatrix},$$

calcular $\|A\|_2$.

Solución. Primero calculamos

$$A^T A = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 5 & -4 \\ -1 & -4 & 5 \end{bmatrix}.$$

El polinomio caracteríistio de $A^T A$ es

$$det\left[A^T A - \lambda I\right] = (6-\lambda)(-\lambda)(6-\lambda) + 3(3\lambda) = -\lambda(\lambda-3)(\lambda-9),$$

cuyas raices son $\lambda_1 = 0$, $\lambda_2 = 3$, $\lambda_3 = 9$. Por lo tanto $\|A\|_2 = \sqrt{\rho(A^T A)} = \sqrt{9} = 3$.

Observación: Si A es una matriz simétrica, entonces $A^T A = A^2$ y entonces los valores propios de $A^T A$ son de la forma λ_i^2, donde λ_i son los valores propios de A, pues

$$A^2\vec{u}_i = A(A\vec{u}_i) = A(\lambda\vec{u}_i) = \lambda A\vec{u}_i = \lambda\lambda\vec{u}_i = \lambda^2\vec{u}_i$$

para cualquier valor propio de $\vec{u}_i$ de A. Por lo tanto, si A es una **matriz simétrica** entonces

$$\|A\|_2 = \sqrt{\rho(A^T A)} = \sqrt{\rho(A^2)} = \sqrt{[\rho(A)]^2} = \rho(A).$$

Ejemplo 2.5. *Dada la matriz simétrica*

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix},$$

su polinomio característico es $p(\lambda) = det\,[A - \lambda I] = \lambda(2-\lambda)(\lambda-3)$. *Por lo tanto* $\|A\|_2 = \rho(A) = 3$.

2.1.6. Propiedades de las normas matriciales

Dadas dos matrices A y $B \in \mathbb{R}^{n\times n}$ y una *norma inducida* $\|\cdot\|$, se satisfacen las siguientes propiedades:

1. $\|Ax\| \leq \|A\| \, \|x\|, \quad \forall \, x \in \mathbb{R}^n$.
2. $\rho(A) \leq \|A\|$.
3. $\|AB\| \leq \|A\| \, \|B\|$.
4. $\|A + B\| \leq \|A\| + \|B\|$ (desigualdad del triángulo).

La propiedad 1 es consecuencia directa de la definición de norma matricial inducida. La propiedad 2 se obtiene de $\|A\| = \sup_{\|x\|=1} \|Ax\| \geq \|A\vec{u}_i\| = \|\lambda_i \vec{u}_i\| = |\lambda_i| \|\vec{u}_i\| = |\lambda_i|$ para cualquier vector propio unitario $\vec{u}_i$ de A con valor propio λ_i . La propiedad 3 es consecuencia directa de la propiedad 1. Finalmente la propiedad 4 es simplemente la desigualdad del triángulo para normas matriciales y es consecuencia de la desigualdad del triángulo en la norma vectorial asociada.

Una norma matricial no natural (no inducida). La norma matricial no inducida más conocida es la denominada norma de *Hilbert-Schmidt* o norma de *Frobenius* definida por

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

Por ejemplo, La norma de Frobenius para la matriz

$$A = \begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 2 \end{bmatrix}$$

es $\|A\|_F = \sqrt{(-1)^2 + 2^2 + (-1)^2 + 1^2 + 1^2 + 2^2} = \sqrt{12}$. No es posible expresar la norma de Frobenius en la forma $\sup_{\|x\|=1} \|Ax\|$ para alguna norma vectorial $\|\cdot\|$.

2.2. Condicionamiento y estabilidad

En esta sección discutiremos brevemente dos temas fundamentales del análisis numérico, a saber el condicionamiento y la estabilidad.

Condicionamiento: *Se refiere al comportamiento bajo perturbaciónes de un problema expresado matemáticamente.*

Estabilidad: *Se refiere al comportamiento bajo perturbaciones de un algoritmo utilizado para resolver un problema en una computadora.*

2.2.1. Condicionamiento de un problema

Un problema se puede considerar en forma abstracta como una función $f : X \to Y$ de un espacio normado X (de datos) a un espacio normado Y (de soluciones). La función generalmente no es lineal pero casi siempre es continua. Por ejemplo la operación de suma se puede representar por

$$f : \mathbb{R}^2 \to \mathbb{R}, \quad f(x) = x_1 + x_2,$$

con $X = \mathbb{R}^2$, $Y = \mathbb{R}$ y $x = (x_1, x_2)^T \in \mathbb{R}^2$. En la mayoría de los casos nos interesa el comportamiento de un problema en un punto específico $x \in X$ (una instancia del problema).

Un ***problema bién condicionado*** es aquel en el que cualquier perturbación pequeña de $x \in X$ ocasiona un pequeño cambio en $f(x)$.

En un ***problema mal condicionado*** pequeñas perturbaciones en los cambios de $x \in X$ ocasiona grandes cambios en los resultados de$f(x)$.

El significado de "pequeño" o "grande" en las anteriores oraciones depende del tipo de aplicación. En particular, en ocasiones es más apropiado medir perturbaciones en una escala absoluta, y en algunas otras es mejor medir las perturbaciones en forma relativa respecto a la norma del dato perturbado.

2.2.2. Número de condición absoluto

Sea δx una pequeña perturbación del dato $x \in X$, y denotemos por $\delta f = f(x + \delta x) - f(x)$ la correspondiente perturbación en el resultado. El *número de condición absoluto* del problema f en x se define por

$$\widehat{K} = \sup_{\delta x} \frac{\|\delta f\|_Y}{\|\delta x\|_X}, \tag{2.5}$$

en donde $\|\cdot\|_Y$ y $\|\cdot\|_X$ representan normas vectoriales en los espacios normados Y y X, respectivamente.

Ejemplo 2.6. *Dada una matriz $A \in \mathbb{R}^{n\times n}$, definimos el producto de esta matriz por un vector x por medio de*

$$f : \mathbb{R}^n \to \mathbb{R}^n, \quad f(x) = Ax \quad \forall x \in \mathbb{R}^n.$$

Encontrar el número de condición absoluto de este problema.

Solución. En este caso $X = Y = \mathbb{R}^n$ y sea $\|\cdot\| = \|\cdot\|_Y = \|\cdot\|_X$ cualquier norma vectorial en $\mathbb{R}^n$, entonces el número de condición absoluto de este problema es

$$\widehat{K} = \sup_{\delta x} \frac{\|\delta f\|}{\|\delta x\|} = \sup_{\delta x} \frac{\|f(x + \delta x) - f(x)\|}{\|\delta x\|} = \sup_{\delta x} \frac{\|A(x + \delta x) - Ax\|}{\|\delta x\|}$$

$$= \sup_{\delta x} \frac{\|A\delta x\|}{\|\delta x\|} = \|A\|$$

en donde $\|\cdot\|$ es la norma matricial inducida por la norma vectorial.

En el caso de que la función f (asociada al problema matemático) sea una función diferenciable, entonces dada la perturbación δx en el dato x, se tiene

$$f(x+\delta x) = f(x) + f'(x)\delta x + \mathcal{O}[\|\delta x\|^2],$$

Si denotamos la derivada $f'(x)$ por $J(x)$ (el Jacobiano en caso que f sea una función de varias variables), entonces

$$\delta f = f(x+\delta x) - f(x) = J(x)\delta(x) + \mathcal{O}[(\delta x)^2]\,.$$

Por lo tanto

$$\widehat{K} = \sup_{\delta x} \frac{\|\delta f\|}{\|\delta x\|} = \sup_{\delta x} \frac{\|J(x)\delta x\|}{\|\delta x\|} = \|J(x)\|\,.$$

Es decir, para un problema representado por una función diferenciable f con Jacobiano J, su número de condición absoluto en x es igual a la norma del Jacobiano en x.

Ejemplo 2.7. *Consideremos el problema de la sustracción de dos números:* $f : \mathbb{R}^2 \to \mathbb{R}$, $f(x_1, x_2) = x_1 - x_2$, *con* $x = (x_1, x_2)^T$. *Dado que*

$$J = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} & \dfrac{\partial f}{\partial x_2} \end{bmatrix} = [1 \quad -1].$$

El número de condición absoluto en la norma–∞, *y en cualquier instancia* $x = (x_1, x_2)^T \in \mathbb{R}^2$ *es constante e igual a*

$$\widehat{K} = \|J(x)\|_\infty = 2.$$

2.2.3. Número de condición relativo

Cuando estamos interesados en cambios relativos, necesitamos la noción de condición relativa. Por ejemplo, en el caso simple en el que X=Y con norma vectorial $\|\cdot\|$, el ***número de condición relativo*** $K = K(x)$ del problema f en x se define por

$$K = \sup_{\delta x} \left(\frac{\|\delta f\|/\|\delta x\|}{\|f(x)\|/\|x\|} \right) = \sup_{\delta x} \left(\frac{\|\delta f\|/\|f(x)\|}{\|\delta x\|/\|x\|} \right). \tag{2.6}$$

En el caso en que f sea diferenciable, podemos expresar el número de condición relativo en términos del Jacobiano

$$K = \frac{\|J(x)\|}{\|f(x)\|/\|x\|} = \frac{\|J(x)\|\,\|x\|}{\|f(x)\|}. \tag{2.7}$$

Tanto el número de condición absoluto como el número de condición relativo tiene sus usos, pero este último es más importante en el análisis numérico. Esto es debido a que la aritmérica de punto flotante utilizada en las computadoras introduce errores relativos. Por esta razón al número de condición relativo para un problema f se le conoce simplemente como *número de condición* a secas. Entonces, decimos que un problema es *bién condicionado* si K (su número de condición) es pequeño, por ejemplo 1, 10, 10^2. Decimos que el problema es *mal condicionado* si K es grande, por ejemplo 10^6, 10^{16}, etc.

Ejemplo 2.8. *Consideremos el problema de calcular $\sqrt{x}$ para $x > 0$.*

$$f(x) = \sqrt{x}, \quad J = f'(x) = \frac{1}{2\sqrt{x}}$$

$$K = \frac{\|J\|\,\|x\|}{\|f(x)\|} = \frac{(\frac{1}{2\sqrt{x}})x}{\sqrt{x}} = \frac{1}{2}$$

Por lo tanto, el problema es bién condicionado.

Ejemplo 2.9. *Consideremos una vez más el problema de obtener el escalar $f(x) = x_1 - x_2$ a partir del vector $x = (x_1, x_2)^T \in \mathbb{R}^2$. Utilizando la norma $\|\cdot\|_\infty$ en $\mathbb{R}^2$, anteriormente encontramos que*

$$J(x) = [1 \quad -1] \quad y \quad \|J(x)\|_\infty = 2,$$

así que

$$K = \frac{\|J(x)\|_\infty \|x\|_\infty}{|f(x)|} = \frac{2\,\text{máx}\{|x_1|, |x_2|\}}{|x_1 - x_2|}$$

y, en consecuencia, el problema es mal condicionado cuando x_1 es muy cercano a x_2, pues en este caso $|x_1 - x_2| \approx 0$, lo cual coincide con nuestra intuición sobre el "error de cancelación".

Ejemplo 2.10. *La determinación de las raices de un polinomio dado los coeficientes es un ejemplo clásico de un problema mal condicionado. Por ejemplo el polinomio $y^2 - 2y + 1 = (y - 1)^2$ tiene una raíz doble $y = 1$. Una pequeña perturbación en los coefecientes puede ocasionar*

un cambio drástico en las raices. Por ejemplo, $y^2 - 2y + (1-\epsilon) = (y - 1 + \sqrt{\epsilon})(y - 1 - \sqrt{\epsilon})$ *para cualquier* $\epsilon > 0$. *En este caso el dato* $x = (1, -2, 1)^T \in X = \mathbb{R}^3$ *denota los coeficientes del polinomio y el resultado* $f(x) = (1,\ 1)^T \in Y = \mathbb{C}^2$ *denota las raices del mismo. Entonces*

$$\delta x = (1, -2, 1 - \epsilon)^T - (1, -2, 1)^T = (0, 0, -\epsilon)^T,$$

$$\delta f = (1 - \sqrt{\epsilon}, 1 + \sqrt{\epsilon})^T - (1, 1)^T = (-\sqrt{\epsilon}, \sqrt{\epsilon})^T.$$

Utilizando la norma-∞ *tanto en* $X = \mathbb{R}^3$ *como en* $Y = \mathbb{C}^2$ *obtenemos*

$$K = \sup_{\delta x} \left(\frac{\|\delta f\|_\infty / \|\delta x\|_\infty}{\|f(x)\|_\infty / \|x\|_\infty} \right) = \sup_{\epsilon > 0} \left(\frac{\sqrt{\epsilon}/\epsilon}{1/2} \right) = \sup_{\epsilon} \frac{2}{\sqrt{\epsilon}} = \infty.$$

Por lo tanto, no es recomendable construir un algorimto numérico a partir de los coeficientes del polinomio para el cálculo de sus raices.

Ejemplo 2.11. *El problema del cálculo de los valores propios de una matriz no simétrica es frecuentemente mal condicionado. Un ejemplo sencillo para darse cuenta de esto es considerar la matriz*

$$A = \begin{bmatrix} 1 & 1000 \\ 0 & 1 \end{bmatrix}$$

cuyos valores propios son $\lambda_1 = 1$, $\lambda_2 = 1$. *Si perturbamos uno de los coeficientes fuera de la diagonal por 0.001, por ejemplo el coeficiente 0, obtenemos la matriz*

$$A' = \begin{bmatrix} 1 & 1000 \\ 0{,}001 & 1 \end{bmatrix},$$

cuyos valores propios son ahora $\lambda_1' = 0$, $\lambda_2' = 2$, *obteniendo un cambio drástico en los valores propios. De hecho, esto mismo sucede con la matriz*

$$\begin{bmatrix} 1 & 10^n \\ 0 & 1 \end{bmatrix},$$

con $n \in \mathbb{N}$, *cuando la perturbamos por*

$$\begin{bmatrix} 1 & 10^n \\ 10^{-n} & 1 \end{bmatrix}.$$

Ejemplo 2.12. *Consideremos nuevamente el problema de la multiplicación de una* ***matriz fija*** $A \in \mathbb{R}^{n\times n}$ *por un vector* $x \in \mathbb{R}^n$, $f(x) = Ax$. *El número de condición (relativo) de este problema es:*

$$K_x = {}^1 \sup_{\delta x} \left(\frac{\|\delta f\|/\|\delta x\|}{\|f(x)\|/\|x\|} \right) = \sup_{\delta x} \frac{\|A\delta x\| \ \|x\|}{\|\delta x\| \ \|Ax\|} = {}^2\|A\| \frac{\|x\|}{\|Ax\|}$$

pero

$$\|x\| = {}^3\|A^{-1}Ax\| \leq \|A^{-1}\| \ \|Ax\|$$

Por lo tanto

$$K \leq \|A^{-1}\| \ \|A\| \quad \Rightarrow \quad K = \alpha \|A\| \ \|A^{-1}\|.$$

Para ciertos valores de x, $\alpha = 1$ *y* $K = \|A\| \ \|A^{-1}\|$.

Ejemplo 2.13. *(Solución de un sistema lineal de ecuaciones) Dada* $A \in \mathbb{R}^{n\times n}$ *no singular y* $b \in \mathbb{R}^n$, *el problema consiste en calcular* $x \in \mathbb{R}^n$ *tal que* $Ax = b$. *Los datos son* A *y* b, *y el mapeo en este caso es* $f : \mathbb{R}^{n\times n} \times \mathbb{R}^n \to \mathbb{R}^n, \quad f(A, b) = x = A^{-1}b$. *Con el objeto de simplificar la discusión supongamos* A *fija, como en el ejemplo anterior, y que sólo varía* b, *es decir,* $f : \mathbb{R}^n \to \mathbb{R}^n, \quad f(b) = x = A^{-1}b$. *Como el Jacobiano en este caso es* $J(b) = A^{-1}$, *entonces el número de condición en* b *es*

$$K_b = \frac{\|J(b)\| \ \|b\|}{\|A^{-1}b\|} = {}^4 \frac{\|A^{-1}\| \ \|Ax\|}{\|x\|}$$

Dado que hay una correspondencia 1-1 entre x *y* b *(pues* A *es invertible), podemos calcular el "peor número de condición"*

$$K = \sup_{\substack{b\in\mathbb{R}^n \\ b\neq\vec{0}}} K_b = \sup_{\substack{b\in\mathbb{R}^n \\ b\neq\vec{0}}} \|A^{-1}\| \frac{\|Ax\|}{\|x\|} = \|A^{-1}\| \sup_{\substack{x\in\mathbb{R}^n \\ x\neq\vec{0}}} \frac{\|Ax\|}{\|x\|} = \|A^{-1}\| \ \|A\|$$

y observamos que el máximo número de condición no depende de b.

Nota. En los dos ejemplos anteriores el máximo número de condición del problema fué $\|A^{-1}\| \ \|A\|$. Esta expresión es muy común en el álgebra lineal numérica, y se le llama *número de condición de la matriz* A, y se escribe

$$K(A) = \|A^{-1}\| \ \|A\|.$$

---

[1] $\delta f = f(x + \delta x) - f(x) = A(x + \delta x) - Ax = A\delta x$

[2] $\sup_{\delta x} \frac{\|A\delta x\|}{\|\delta x\|} = \|A\|$

[3] Suponiendo que A es no singular

[4] $b = Ax \quad \Rightarrow \quad x = A^{-1}b$

Observación: Si $A \in \mathbb{R}^{n\times n}$ es una matriz simétrica no singular, $\|A\|_2 = \rho(A) = \lambda_{\text{máx}}(A)$, y en la norma-2:

$$K(A) = cond(A) = \frac{|\lambda_{\text{máx}}(A)|}{|\lambda_{\text{mín}}(A)|}$$

donde $\lambda_{\text{máx}}(A) = \text{máx}\{|\lambda| : \lambda \text{ es valor propio de } A\}$, y $\lambda_{\text{mín}}(A)$ se define en forma análoga.

EJEMPLOS DE MATRICES MAL CONDICIONADAS

Ejemplo 2.14. *La matriz de Hilbert*

$$H_n = \begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{i} & \frac{1}{i+1} & \cdots & \frac{1}{n+i-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{bmatrix}$$

es simétrica e invertible. Su número de condición en la norma euclideana $\|\cdot\|_2$ para diferentes valores de n es:

n	$K(H_n)$
10	$1{,}6 \times 10^{13}$
20	$2{,}45 \times 10^{28}$
40	$7{,}65 \times 10^{58}$

Estos números de condición crecen muy rápidamente con n. Un sistema $H_{20}\, x = b$ no puede resolverse en forma apropiada en doble precisión ya que $K(H_{20}) \sim 10^{28}$. La matriz de Hilbert es un prototipo de una matriz mal condicionada y (ver Gantschi)

$$K(H_n) = \|H_n\|_2 \, \|H_n^{-1}\|_2 \sim \frac{(\sqrt{2}+1)^{4n+4}}{2^{15/4}\sqrt{\pi n}} \quad \textit{cuando} \quad n \to \infty.$$

En clase de laboratorio consideraremos con más detalle esta matriz.

Ejemplo 2.15. *Una matriz de Vandermonde es de la forma*

$$V_n = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & t_n^2 & \cdots & t_n^{n-1} \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad \textit{con } t_1, t_2, \ldots, t_n \in \mathbb{R}.$$

Si los parámetros $t_1, t_2, \ldots, t_n$ se escogen igualmente espaciados entre -1 y 1, es decir, $t_i = -1 + \frac{2(i-1)}{n-1}$, $i = 1, \ldots, n$, entonces (ver Gantschi)

$$K(V_n) = \|V_n\|_1 \|V_n^{-1}\|_1 \sim \frac{1}{\pi} e^{-\pi/4} e^{n(\frac{\pi}{4} + \frac{1}{2}\ln 2)} \quad \textit{cuando} \quad n \to \infty.$$

Algunos valores numéricos se muestran en la tabla siguiente

n	$K(V_n)$
10	$1{,}36 \times 10^4$
20	$1{,}05 \times 10^9$
40	$6{,}93 \times 10^{18}$
80	$3{,}15 \times 10^{38}$

En aritmética IEEE de doble precisión sólo es posible calcular estos números de condición para $n = 10, 20$ y 40 como veremos en clase de laboratorio.

2.2.4. Estabilidad de los algoritmos

Sería deseable que los algoritmos numéricos proporcionaran soluciones exactas a los problemas numéricos. Sin embargo, debido a que las computadoras solo pueden representar un sistema numérico discreto, esto no es posible en general. Como ya hemos visto anteriormente, el error de redondeo jugará un papel importante en este caso. La noción de estabilidad es la forma común de caracterizar *lo que es posible*, es decir, de obtener *la respuesta correcta* aunque no sea la respuesta exacta.

Algoritmos. Un algoritmo puede verse como un mapeo $\widetilde{f} : X \to Y$ el cual esta asociado a un problema matemático $f : X \to Y$. Para precisar la definición de algoritmo consideremos:

1. Un problema $f : X \to Y$
2. Una computadora cuyo sistema de punto flotante satisface el axioma fundamental de la aritmética de punto flotante (1.7).
3. Un algoritmo $\widetilde{f}$, para el poblema f, y su implementación en computadora (programa).

Dado el dato $x \in X$, este dato es redondeado en la computadora para obtener en numero de punto flotante $\widetilde{x} = x(1+\epsilon)$ con $\|\epsilon\| \leq \epsilon_{maq}$, y este último número después proporcionado como entrada al programa. Al correr el programa el resultado es una colección de números de punto flotante que pertenece a Y. Llamemos al resultado $\widetilde{f}(x)$, el cual generalmente es

distinto a $f(x)$. Esquemáticamente

$$\begin{array}{ccccccc} \text{Entrada} & & \text{Redondeo} & & \text{Aritmética de punto flotante} & & \text{Salida} \\ x & \rightarrow & \widetilde{x} & \rightarrow & \text{Programa (conjunto de operaciones)} & \rightarrow & \widetilde{f}(x) \end{array}$$

Mínimamente el resultado $\widetilde{f}(x)$ debe ser afectado por errores de redondeo, pero también, y dependiendo de las circunstacias, puede ser alterado por otras complicaciones, como son otros programas corriendo al mismo tiempo ó también por tolerancias de convergencia. Asi que $\widetilde{f}(x)$ puede de hecho tomar valores diferentes, en corridas diferentes, es decir, $\widetilde{f}$ puede ser un mapeo multivariado.

Precisión. $\widetilde{f}$ no será un mapeo continuo salvo en casos excepcionales. Lo central es que el algoritmo aproxime "en forma adecuada" el problema f. Para cuantificar la precisión da la aproximación podemos considerar el

$$\text{error absoluto}: \quad \|\widetilde{f}(x) - f(x)\|$$

ó el

$$\text{error relativo}: \quad \frac{\|\widetilde{f}(x) - f(x)\|}{\|f(x)\|}$$

en alguna norma $\|\cdot\|$. Nosotros consideramos el error relativo ya que, como hemos dicho anteriormnte, en el análisis numérico se prefieren cantidades relativas para cuantificar errores. Entonces podriamos decir que $\widetilde{f}$ *es un buen algoritmo* para el problema f si el error relativo es de orden del epsilon de máquina, es decir, si

$$\frac{\|\widetilde{f}(x) - f(x)\|}{\|f(x)\|} = \mathcal{O}(\epsilon_{maq})$$

Estabilidad: si el problema f es mal condicionado, entonces el objetivo de precisión $\mathcal{O}(\epsilon_{maq})$ es excesivamente ambicioso, dado que cualquier perturbación en el dato $x \in X$ ocasionará un gran cambio en el resultado. El error de redondeo es inevitable al utilizar la computadora. Así que, en lugar de buscar precisión en todos los casos (lo cual es imposible), a lo más que podemos aspirar es a mantener cierta *estabilidad* en los resultados.

Definición: Decimos que un *algoritmo* $\widetilde{f}$ para un problema f es *estable* si para cada $x \in X$

$$\frac{\|\widetilde{f}(x) - f(\widetilde{x})\|}{\|f(\widetilde{x})\|} = \mathcal{O}(\epsilon_{maq}) \quad \text{para algún} \quad \widetilde{x} \quad \text{tal que} \quad \frac{\|\widetilde{x} - x\|}{\|x\|} = \mathcal{O}(\epsilon_{maq}) \tag{2.8}$$

En palabras, la anterior definición puede expresarse como "un algoritmo estable proporciona la respuesta *casi* correcta a la pregunta *casi* correcta".

2.2.5. Estabilidad regresiva (Backward-Stability)

Algunos algoritmos en el análisis numérico satisfacen una condición que es a la vez *más fuerte y más simple* que estabilidad a secas.

Definición: Decimos que un algoritmo $\tilde{f}$ para un problema f es *estable regresivo* si para cada $x \in X$

$$\tilde{f}(x) = f(\tilde{x}) \quad \text{para algún} \quad \tilde{x} \quad \text{con} \quad \frac{\|\tilde{x} - x\|}{\|x\|} = \mathcal{O}(\epsilon_{maq}) \tag{2.9}$$

Esquemáticamente tenemos

	Dato				Resultado
(exacto)	x	$\rightarrow$	redondeo + algoritmo	$\rightarrow$	$\tilde{f}(x)$
(perturbado)	$\tilde{x}$	$\rightarrow$	solución analítica	$\rightarrow$	$f(\tilde{x})$

donde $\tilde{x} = x(1+\epsilon)$ con $|\epsilon| \leq \epsilon_{maq}$. Arriba, $f(\tilde{x})$ es la respuesta correcta a la pregunta casi correcta $\tilde{x}$.

Ejemplo 2.16. *Analizar la estabilidad de la aritmética de punto flotante para el caso de la sustracción de dos números.*

Solución. El problema y el algorimto se pueden escribir de la siguiente manera

Problema: $f : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad f(x_1\,,\, x_2) = x_1 - x_2.$

Algoritmo: $\tilde{f} : \mathbb{R} \rightarrow \mathbb{R}, \quad \tilde{f}(x_1\,,\, x_2) = fl(x_1) \ominus fl(x_2).$

Por el axioma fundamental de la aritmética de punto flotante

$$\begin{aligned}
\tilde{f}(x_1\,,\, x_2) &= (fl(x_1) - fl(x_2))(1+\epsilon_3) \\
&= (x_1(1+\epsilon_1) - x_2(1+\epsilon_2))(1+\epsilon_3) \\
&= x_1(1+\epsilon_1+\epsilon_3+\epsilon_1\epsilon_3) - x_2(1+\epsilon_2+\epsilon_3+\epsilon_2\epsilon_3) \\
&= x_1(1+\epsilon_4) - x_2(1+\epsilon_5) \\
&= \tilde{x}_1 - \tilde{x}_2 \\
&= f(\tilde{x}_1\,,\, \tilde{x}_2)\,,
\end{aligned}$$

donde $|\epsilon_4|, |\epsilon_5| \leq 2\epsilon_{maq} + \mathcal{O}(\epsilon_{maq}^2)$. Además x_1, x_2 satisfacen

$$\frac{|\tilde{x}_1 - x_1|}{|x_1|} = \mathcal{O}(\epsilon_{maq}), \quad \frac{|\tilde{x}_2 - x_2|}{|x_2|} = \mathcal{O}(\epsilon_{maq}).$$

Análogamente, se puede verificar que los algoritmos $\oplus, \otimes, \oslash$ son todos *estable regresivos.* Asimismo, en forma análoga también mostrarse que el producto interno de dos vectores $x^T y$, x, $y \in \mathbb{R}^n$ es estable regresivo. Por otro lado, el producto externo $xy^T \in \mathbb{R}^{n\times x}$ es estable, pero no estable regresivo, pues $A = xy^T$ es una matriz de rango 1 exactamente, y $\tilde{A}$, la matriz calculada no puede generalmente escribirse en la forma $(x + \delta x)(y + \delta y)^T$, pues muy probablementese se obtendría una matriz de rango mayor a 1.

El uso del polinomio caracteristico $p(x) = det[A - \lambda I]$ para calcular los valores propios de una matriz A es un algoritmo inestable. De hecho, anteriormente hemos encontrado que el cálculo de raices de un polinomio es un problema mal condicionado.

2.2.6. Precisión de un algoritmo estable regresivo

Supongase que tenemos un algoritmo $\tilde{f}$ para un problema f que es estable regresivo. ¿El resultado proporcionado por el algoritmo $\tilde{f}$ será adecuado o preciso? La respuesta depende de si el problema f es bién o mal condicionado, es decir, depende del número de condición $K = K(x)$ de f. Si $K(x)$ es pequeño, los resultados serán precisos en sentido relativo, pero si $K(x)$ es grande el error crecerá proporcionalmente:

Análisis: El error relativo se define por

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|}.$$

Como el algoritmo se supone estable regresivo, entonces

$$\tilde{f}(x) = f(\tilde{x}) \text{ para algún } \tilde{x} \text{ tal que } \frac{\|\tilde{x} - x\|}{\|x\|} = \mathcal{O}(\epsilon_{maq}),$$

y el error relativo se puede estimar de la siguiente manera

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = \frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|} = \frac{\|\delta f(x)\|}{\|f(x)\|} \leq {}^5K(x)\frac{\|\delta x\|}{\|x\|} = K(x)\mathcal{O}(\epsilon_{maq}).$$

Es decir

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leq K(x)\mathcal{O}(\epsilon_{maq})\,, \tag{2.10}$$

y el error crece proporcionalmente a $K(x)$ en términos del ϵ_{maq}.

[5] $K(x) = \sup_{\delta x}(\frac{\|\delta f(x)\|}{\|f(x)\|} / \frac{\|\delta(x)\|}{\|x\|})$

El proceso que hemos seguido en el anterior análisis es conocido como *análisis regresivo del error*. Se obtiene una estimación del error en dos pasos: en el primer paso se investiga la condición del problema; en el segundo paso se investigar la estabilidad del algoritmo. Podemos entonces concluir que si el algortimo es estable, entonces *la precisión final refleja el número de condición*. Es decir, el mejor algoritmo para la mayoría de los problemas no produce mejores resultados que calcular la solución exacta para datos ligeramente perturbados.

Capítulo 3

Solución de Sistemas de Ecuaciones Lineales

Uno de los problemas más frecuentemente encontrados en la computación científica es el de la solución de sistemas de ecuaciones algebraicas lineales. Este problema consiste en encontrar $x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$ tal que $Ax = b$, donde A es una matriz de $n \times n$ y $b = (b_1, b_2, \ldots, b_n)^T \in \mathbb{R}^n$ son dados. Este problema aparece muy a menudo en muchas de las aplicaciones de la matemática, ciencias e ingeniería. Algunos ejemplos son el ajuste de datos, problemas de optimización, aproximación de ecuaciones diferenciales y de ecuaciones integrales. En el presente estudio, salvo en el siguiente capítulo, sólo consideraremos sistemas de ecuaciones lineales cuadrados que tengan solución única. Algunas de las condiciones más conocidas para que el sistema $Ax = b$ tenga solución única son:

1. A es una matriz no-singular (invertible)
2. La única solución de $Ax = \vec{0}$ es $x = \vec{0}$
3. $\det(A) \neq 0$.

3.1. Eliminación de Gauss

El método más conocido (y, en muchos casos, el más popular) para resolver sistemas de ecuaciones algebraicas lineales es el *método de eliminación de Gauss*. La idea básica de este método consiste en manipular las ecuaciones por medio de operaciones elementales para tranformar el sistema original en un sistema equivalente que sea más sencillo de resolver. Las *operaciones elementales* en la eliminación de Gauss son tres:

1. *Multiplicación de una ecuación por una constante no cero.*

2. *Sustracción del múltiplo de una ecuación de otra ecuación.*

3. *Intercambio de ecuaciones.*

Si alguna de estas operaciones se aplican a algún sistema de ecuaciones el sistema obtenido será *equivalente* al original. Lo mismo sucede cuando se realiza una cadena de estas operaciones. Nuestro objetivo es resolver el sistema $Ax = b$, donde $A = (a_{ij})$, $1 \leq i, j \leq n$, $b = (b_1, b_2, \ldots, b_n)^T$, que en forma explícita es:

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

Si a este sistema le llamamos $A^{(1)}x = b^{(1)}$, para indicar el estado original del sistema, entonces el *proceso de eliminación de Gauss* es como se muestra a continuación:

1^{er} Paso de eliminación. Si $a_{11}^{(1)} \neq 0$, podemos eliminar la incógnita x_1 de las demás ecuaciones. El paso típico es restar de la $i-$ésima ecuación ($i = 1, 2, \ldots, n$) la primera multiplicada por

$$m_{i1} = a_{i1}^{(1)}/a_{11}^{(1)} \quad i = 2, 3, \ldots, n$$

A m_{i1} se le denomina *multiplicador* asociado a la $i-$ésima ecuación en el primer paso de eliminación. Después de realizar esta operación la $i-$ésima ecuación tendrá nuevos coeficientes $a_{ij}^{(2)}$ y $b_i^{(2)}$ cuyos valores son:

$$a_{i1}^{(2)} = 0$$

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}\, a_{1j}^{(1)} \quad \text{para } j = 2, 3, \ldots, n$$

$$b_i^{(2)} = b_i^{(1)} - m_{i1}\, b_1^{(1)}$$

Haciendo lo anterior para cada renglón $i = 2, \ldots, n$, obtenemos el nuevo sistema $A^{(2)}x = b^{(2)}$ que es:

$$\begin{array}{rcl} a_{11}^{(1)}x_1 + \quad a_{12}^{(1)}x_2 + \cdots + a_{1n}^{(1)}x_n & = & b_1^{(1)} \\ a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n & = & b_2^{(2)} \\ \vdots \qquad\qquad\qquad \vdots & & \vdots \\ a_{n2}^{(2)}x_2 + \cdots + a_{nn}^{(2)}x_n & = & b_n^{(2)} \end{array}$$

Nota. Observese que si se va a resolver computacionalmente el problema, para almacenar los coefecientes a_{ij} y b_i, podemos escribir sobre los $a_{ij}^{(1)}$ los nuevos $a_{ij}^{(2)}$ justamente calculados. Podemos almacenar también los multiplicadores m_{i1} en donde teniamos los coeficientes $a_{i1}^{(1)}$, y recordando que todos los elementos debajo de la diagonal de la primera columna de $A^{(2)}$ son realmente cero. Más adelante veremos porqué es útil almacenar los multiplicadores.

2º paso de eliminación. En este paso el objetivo es eliminar la incógnita x_2 de la tercera ecuación a la última ecuación. Si $a_{22}^{(2)} \neq 0$, primero se calculan los multiplicadores

$$m_{i2} = a_{i2}^{(2)}/a_{22}^{(2)}, \quad i = 3, \ldots, n\,.$$

Los nuevos coeficientes $a_{ij}^{(3)}$ y $b_i^{(3)}$ de la i–ésima ecuación serán:

$$a_{i2}^{(3)} = 0$$

$$a_{ij}^{(3)} = a_{ij}^{(2)} - m_{i2}\, a_{2j}^{(2)} \quad \text{para } j = 3, \ldots, n$$

$$b_i^{(3)} = b_i^{(2)} - m_{i2}\, b_2^{(2)}$$

Haciendo lo anterior para cada renglón $i = 3, \ldots, n$, obtenemos el nuevo sistema $A^{(3)}x = b^{(3)}$ que es:

$$\begin{array}{rrrcl}
a_{11}^{(1)}x_1+ & a_{12}^{(1)}x_2+ & a_{13}^{(1)}x_3 + \cdots + a_{1n}^{(1)}x_n & = & b_1^{(1)} \\
 & a_{22}^{(2)}x_2+ & a_{23}^{(2)}x_3 + \cdots + a_{2n}^{(2)}x_n & = & b_2^{(2)} \\
 & & a_{33}^{(3)}x_3 + \cdots + a_{3n}^{(3)}x_n & = & b_3^{(3)} \\
 & & \vdots \qquad\qquad \vdots & & \vdots \\
 & & a_{n3}^{(3)}x_3 + \cdots + a_{nn}^{(3)}x_n & = & b_n^{(3)}
\end{array}$$

Continuando de esta manera, y después de $n-1$ pasos de eliminación, obtenemos un *sistema triangular superior*

$$\begin{array}{rrrrcl}
a_{11}^{(1)}x_1+ & a_{12}^{(1)}x_2+ & a_{13}^{(1)}x_3 + \cdots + & a_{1n}^{(1)}x_n & = & b_1^{(1)} \\
 & a_{22}^{(2)}x_2+ & a_{23}^{(2)}x_3 + \cdots + & a_{2n}^{(2)}x_n & = & b_2^{(2)} \\
 & & a_{33}^{(3)}x_3 + \cdots + & a_{3n}^{(3)}x_n & = & b_3^{(3)} \\
 & & & \vdots & & \vdots \\
 & & & a_{nn}^{(n)}x_n & = & b_n^{(n)}
\end{array}$$

que denotaremos por $A^{(n)}x = b^{(n)}$. El proceso anterior se termina sin problemas siempre y cuando ninguno de los coeficientes $a_{11}^{(1)}$, a_{22}^{2}, $\ldots$, $a_{nn}^{(n)}$, denominados ***pivotes***, sea cero. Cuando se realiza computacionalmente este procedimiento la matriz se rescribe en forma sucesiva,

en cada paso de eliminación, almacenando los nuevos coeficientes $a_{ij}^{(k)}$ y los correspondientes multiplicadores m_{ik} en los lugares asociados a las variables eliminadas. Al término del proceso de eliminación obtenemos un *sistema triangular superior* $Ux = b$ (donde $U = A^{(n)}$, $b = b^{(n)}$) el cual es equivalente al sistema original, es decir este nuevo sistema tiene exactamente la misma solución que el sistema original. Sin embargo, este nuevo sistema puede resolverse muy fácilmente por medio de la técnica de sustitución hacia atrás ó *sustitución regresiva*:

$$x_n = b_n/a_{nn}$$

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

en donde hemos suprimido los superíndices para simplificar la notación. Entonces, suponiendo que en el proceso de eliminación ninguno de los pivotes $a_{ii}^{(i)}$ es cero, el algoritmo de eliminación de Gauss puede escribirse de la siguiente manera:

Algoritmo de eliminación de Gauss

Dados los coeficientes a_{ij} de la matriz A, y los coeficientes b_i de b
Para $k = 1, 2, \ldots, n-1$ /* *Pasos de eliminación* */
. Para $i = k+1, \ldots, n$
. . $m := a_{ik}/a_{kk}$ /* *Multiplicador asociado al renglón i* */
. . Para $j = k+1, \ldots, n$
. . . $a_{ij} := a_{ij} - ma_{kj}$
. . Fín
. . $b_i := b_i - mb_k$
. Fín
Fín
$x_n = b_n/a_{nn}$ /* *Sustitución regresiva* */
Para $i = n-1, n-2, \ldots, 1$
. $x_i := b_i$
. Para $j = i+1, \ldots, n$
. . $x_i := x_i - a_{ij}x_j$
. Fín
. $x_i := x_i/a_{ii}$
Fín

Ejemplo 3.1. *Dada la matriz A y el vector b*

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 8 \\ 30 \\ 41 \end{bmatrix}$$

aplicar el método de eliminación de Gauss para calcular la solución del sistema $A\,x = b$. La solución exacta de este sistema es $x = (-1, 2, 1, 3)^T$.

Solución. En la práctica, para aplicar el método de eliminación de Gauss, es útil escribir solo los coeficientes de la matriz A en el lado izquierdo y los del vector b en el lado derecho, sin incluir las incógnitas. Entonces, el sistema inicial se puede escribir de la siguiente manera:

$$A^{(1)}x = b^{(1)}: \quad \left[\begin{array}{cccc|c} \mathbf{2} & 1 & 1 & 0 & 1 \\ 4 & 3 & 3 & 1 & 8 \\ 8 & 7 & 9 & 5 & 30 \\ 6 & 7 & 9 & 8 & 41 \end{array}\right]$$

El proceso de eliminación de Gauss se muestra a continuación:

1^{er} paso de eliminación.

Pivote: $a_{11} = 2$.

Multiplicadores:

$$2^{\text{o}} \text{ renglón:} \quad m_{21} = \frac{a_{21}}{a_{11}} = \frac{4}{2} = 2$$
$$3^{\text{er}} \text{ renglón:} \quad m_{31} = \frac{a_{31}}{a_{11}} = \frac{8}{2} = 4$$
$$4^{\text{o}} \text{ renglón:} \quad m_{41} = \frac{a_{41}}{a_{11}} = \frac{6}{2} = 3$$

Entonces,

restamos del segundo renglón el primero multiplicado por $m_{21} = 2$,

restamos del tercer renglón el primero multiplicado por $m_{31} = 4$,

restamos del cuarto renglón el primero multiplicado por $m_{41} = 3$,

con lo cual obtenemos:

$$A^{(2)}x = b^{(2)}: \qquad \left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 1 \\ & \mathbf{1} & 1 & 1 & 6 \\ & 3 & 5 & 5 & 26 \\ & 4 & 6 & 8 & 38 \end{array}\right]$$

2º paso de eliminación

Pivote: $a_{22} = 1$.

Multiplicadores:

$$3^{\text{er}} \text{ renglón:} \quad m_{32} = \frac{a_{32}}{a_{22}} = \frac{3}{1} = 3$$
$$4^{\text{o}} \text{ renglón:} \quad m_{42} = \frac{a_{42}}{a_{22}} = \frac{4}{1} = 4$$

Entonces

restamos del tercer renglón el segundo multiplicado por $m_{32} = 3$,

restamos del cuarto renglón el segundo multiplicado por $m_{42} = 4$,

y se obtiene:

$$A^{(3)}x = b^{(3)}: \qquad \left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 1 \\ & 1 & 1 & 1 & 6 \\ & & \mathbf{2} & 2 & 8 \\ & & 2 & 4 & 14 \end{array}\right]$$

3er paso de eliminación

Pivote : $a_{33} = 2$.

Multiplicadores:

$$4^{\text{o}} \text{ renglón:} \quad m_{43} = \frac{a_{43}}{a_{33}} = \frac{2}{2} = 1$$

Entonces, restando del cuarto renglón el tercero, pues $m_{43} = 1$, se obtiene

$$A^{(4)}x = b^{(4)}: \qquad \left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 1 \\ & 1 & 1 & 1 & 6 \\ & & 2 & 2 & 8 \\ & & & 2 & 6 \end{array}\right] \Rightarrow \begin{array}{rcl} 2x_1 + x_2 + x_3 & = & 1 \\ x_2 + x_3 + x_4 & = & 6 \\ 2x_3 + 2x_4 & = & 8 \\ 2x_4 & = & 6 \end{array}$$

Sustitución regresiva. En el sistema triangular superior obtenido hacemos sustitución regresiva para encontrar la solución.

$$\begin{aligned} x_4 &= \frac{6}{2} = 3 \\ x_3 &= \frac{8-2x_4}{2} = \frac{8-6}{2} = 1 \\ x_2 &= \frac{6-x_3-x_4}{1} = \frac{6-1-3}{1} = 2 \\ x_1 &= \frac{1-x_2-x_3}{2} = \frac{1-2-1}{2} = -1 \end{aligned}$$

3.2. Factorización LU

En la sección anterior hemos visto como el proceso de eliminación de Gauss transforma un sistema lineal completo en un sistema triangular superior por medio de la aplicación de operaciones elementales de eliminación. Este proceso de eliminación se puede interpretar desde un punto de vista meramente matricial. Es decir, cada paso de eliminación se puede escribir en forma compacta por medio de la multiplicación de una matriz. Por ejemplo, para el sistema $Ax = b$ con

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 8 \\ 30 \\ 41 \end{bmatrix},$$

el primer paso de eliminación se puede expresar multiplicando el sistema por una matriz triangular inferior. Esta matriz triangular inferior contiene unos en la diagonal y los multiplicadores con signo contrario en sus posiciones correspondientes. El resultado se muestra a continuación

$$L_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -4 & 0 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{bmatrix} \quad \Longrightarrow \quad L_1 A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 3 & 5 & 5 \\ 0 & 4 & 6 & 8 \end{bmatrix}, \qquad L_1 b = \begin{bmatrix} 1 \\ 6 \\ 26 \\ 38 \end{bmatrix},$$

obteniendo la matriz y lado derecho al final del primer paso de eliminación. En forma análoga, el segundo paso de eliminación equivale a premultiplicar el sistema anterior por la matriz

triangular inferior (con los multiplicadores correspondientes con signo contrario)

$$L_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & -4 & 0 & 1 \end{bmatrix}$$

En este caso se obtiene

$$L_2L_1A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix}, \qquad L_2L_1b = \begin{bmatrix} 1 \\ 6 \\ 8 \\ 14 \end{bmatrix}.$$

Finalmente, el tercer paso de eliminación equivale a premultiplicar el último sistema por la matriz

$$L_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

obteniendo

$$L_3L_2L_1A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \qquad L_3L_2L_1b = \begin{bmatrix} 1 \\ 6 \\ 8 \\ 6 \end{bmatrix}.$$

Si denotamos esta última matriz triangular superior por U, y la matriz $L_3L_2L_1$ por L^{-1}, entonces esta claro que

$$A = LU$$

El cálculo de la matriz L es sencillo como veremos a continuación. Observese que $L = (L_3L_2L_1)^{-1} = L_1^{-1}L_2^{-1}L_3^{-1}$, y basta con calcular las inversas de las matrices L_1, L_2 y L_3. El cálculo de estas inversas es trivial. Por ejemplo

$$L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -4 & 0 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{bmatrix}.$$

Análogamente las inversas de L_2 y L_3 se obtienen simplemente cambiando el signo de sus coeficientes debajo de la diagonal, y su producto es:

$$L = L_1^{-1}L_2^{-1}L_3^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 3 & 1 & 0 \\ 3 & 4 & 1 & 1 \end{bmatrix}$$

La cual es una matriz triangular inferior con unos en la diagonal, y con los multiplicadores debajo de la diagonal. A esta matriz L se le conoce como la ***matriz de multiplicadores***. Podemos generalizar el resultado anterior:

Factorización *LU. Si en el proceso de eliminación de Gauss ninguno de los pivotes $a_{ii}^{(i)}$ es cero, entonces la matriz A se puede factorizar en la forma $A = LU$. La matriz L es triangular inferior con unos en la diagonal y con los multiplicadores debajo de la diagonal. La matriz U es la matriz triangular superior que se obtiene al final del proceso de eliminación ($U = A^{(n)}$) y contiene los pivotes en la diagonal. Es decir,*

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & & \ddots & \\ l_{n1} & \dots & l_{n(n-1)} & 1 \end{bmatrix} \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n)} \end{bmatrix}$$

donde $l_{ij} = m_{ij}$ para $i > 1$ son los multiplicadores que se obtienen en el proceso de eliminación de Gauss.

Observación. Como $a_{ii}^{(i)} \neq 0$, entonces A es no singular y

$$detA = det(LU) = (detL)(detU) = (1)(\prod_{i=1}^{n} a_{ii}^{(i)}) = \text{producto de los pivotes.}$$

Solución del sistema $Ax = b$ utilizando la factorización LU

Sea el sistema $Ax = b$ con $A \in \mathbb{R}^{n\times n}$ invertible, $b \in \mathbb{R}^n$. Supongase que ya tenemos una factorización $A = LU$. Entonces, el sistema de ecuaciones también su puede escribir en la forma $LUx = b$. Si hacemos $Ux = y$, entonces $Ly = b$, y por lo tanto el sistema puede resolverse en dos pasos:

1. Se resuelve el sistema triangular inferior $Ly = b$ utilizando *Sustitución hacia adelante ó progresiva*:

$$\begin{aligned} y_1 &= b_1, \\ y_i &= b_i - \sum_{j=1}^{i-1} l_{ij} y_j \,, \qquad i = 2, \ldots, n. \end{aligned}$$

2. Una vez obtenido y del paso anterior, se resuelve el sistema triangular superior$Ux = y$ utilizando *Sustitución hacia atrás ó regresiva*:

$$\begin{aligned} x_n &= y_n / a_{nn}, \\ x_i &= (y_i - \sum_{j=i+1}^{n} u_{ij}\, x_j)/u_{ii} \,, \qquad i = n-1, n-2, \ldots, 1, \end{aligned}$$

en donde hemos denotado por u_{ij} a los coeficientes $a_{ij}^{(i)}$, $j \geq i$, de la matriz U.

Ejemplo 3.2. *Resolver el sistema del sistema de ecuaciones anterior utilizando factorización LU.*

Solución. Del proceso de elminación de Gauss obtenemos:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 3 & 1 & 0 \\ 3 & 4 & 1 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 8 \\ 30 \\ 41 \end{bmatrix}$$

Entonces, $Ly = b$ es

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 3 & 1 & 0 \\ 3 & 4 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ 30 \\ 41 \end{bmatrix}$$

y la solución por sustitución progresiva es

$$\begin{aligned} y_1 &= 1 \\ y_2 &= 8 - 2y_1 = 8 - 2 = 6 \\ y_3 &= 30 - 4y_1 - 3y_2 = 30 - 4 - 18 = 8 \\ y_4 &= 41 - 3y_1 - 4y_2 - y_3 = 41 - 3 - 24 - 8 = 6 \end{aligned}$$

Luego $Ux = y$ es

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ 8 \\ 6 \end{bmatrix}$$

y la solución por sustitución regresiva es

$$\begin{aligned} x_4 &= 6/2 = 3 \\ x_3 &= (8 - 2x_4)/2 = (8 - 6)/2 = 1 \\ x_2 &= (6 - x_3 - x_4)/1 = (6 - 3 - 1)/1 = 2 \\ x_1 &= (1 - x_2 - x_3 - 0x_4)/2 = (1 - 2 - 1)/2 = -1 \end{aligned}$$

Por lo tanto, la solución es la misma que la obtenida anteriormente.

3.3. Inestabilidad del método de eliminación de Gauss

El método de eliminación de Gauss, como se ha presentado hasta el momento, desafortunadamente no es un buen método práctico de propósito general para resolver sistemas de ecuaciones lineales. Como veremos en esta seccón este no es método estable regresivo. De hecho su posible inestabilidad, en algunos casos, está asociada a una dificultad muy simple: ***para algunas matrices el método no funciona debido que se corre el peligro de dividir por cero***.

Ejemplo 3.3. *La matriz*

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

tiene rango completo (es invertible) y es bién condicionada, debido a que

$$p(\lambda) = det(A - \lambda I) = -\lambda(1 - \lambda) - 1 = \lambda^2 - \lambda - 1 \Rightarrow \lambda_{1,2} = \frac{1 \pm \sqrt{5}}{2}$$

$$A \text{ simétrica} \Rightarrow K(A) = cond(A) = \frac{|\lambda_{\text{máx}}|}{|\lambda_{\text{mín}}|} = \frac{1 + \sqrt{5}}{1 - \sqrt{5}} = \frac{3 - \sqrt{5}}{2} \approx 2{,}618.$$

Sin embargo, el método de eliminación de Gauss falla en el primer paso debido a que el primer pivote es cero.

Al introducir una pequeña perturbación en la matriz anterior se revelan otras dificultades. Por ejemplo, supongase que aplicamos eliminación de Gauss a la matriz perturbada

$$\tilde{A} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

Ahora el método no fallará en el primer paso de eliminación. En este caso al segundo renglón le restamos el primero multiplicado por 10^{20} (pivote $= 10^{-20}$, multiplicador $1/10^{-20} = 10^{20}$). Suponiendo que realizamos las anteriores operaciónes en aritmética exacta, obtenemos la factorización LU con

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \qquad U = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

Sin embargo, en aritmética de punto flotante *IEEE* de doble precisión con $\epsilon_{maq} = 2{,}224 \times 10^{-16}$, el número $1 - 10^{20}$ no puede representarse en forma exacta y es redondeado al número de punto flotante más cercano: En *MATLAB*, $1 - 10^{20} = -10^{20}$. Tomando en consideración este hecho las matrices de punto flotante en la factorización en realidad serán

$$\tilde{L} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \qquad \tilde{U} = \begin{bmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix}$$

las cuales son cercanas en sentido relativo a las matrices exactas L y U. Esto significa que hemos calculado la factorización en forma estable. Sin embargo, cuando multiplicamos $\tilde{L}$ por $\tilde{U}$ aparece un problema inesperado: desgraciadamente $\tilde{A} \neq \tilde{L}\tilde{U}$, dado que

$$\tilde{A} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix} \qquad \text{y} \qquad \tilde{L}\tilde{U} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 0 \end{bmatrix}.$$

La diferencia es el coeficiente $a_{22} = 1$, el cual es muy grande comparado con el valor de la perturbación 10^{-20}. Cuando intentamos resolver el sistema $Ax = b$ por medio de la factorización $\tilde{L}\tilde{U}x = b$ aparece otro problema: el resultado es muy diferente al exacto. Por ejemplo, con $b = (1, 0)^T$ el sistema $Ax = b$ es

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

y la solución exacta es $x = (-1, 1)^T$. Por otro lado, el sistema $\tilde{L}\tilde{U}x = b$ es

$$\begin{bmatrix} 10^{-20} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

La solución en dos etapas de este sistema proporciona

$$\tilde{L}y = b: \quad \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \tilde{y} = (1, -10^{20})^T$$

$$\tilde{U}x = \tilde{y}: \quad \begin{bmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -10^{20} \end{bmatrix} \quad \Rightarrow \quad \tilde{x} = (0, 1)^T,$$

lo cual muestra que la solución de punto flotante $\tilde{x}$ es muy diferente a la solución exacta x. Un análisis cuidadoso de lo que ocurre en este ejemplo revela que la eliminación de Gauss ha calculado la factorización LU establemente ($\tilde{L}$ es cercana a L y $\tilde{U}$ es cercana a U), pero no ha resuelto $Ax = b$ establemente ($\tilde{x}$ no es cercana a x). Una explicación de este fenómeno es que a pesar de que la factorización LU se ha hecho en forma estable, esta factorización no es estable regresiva. Las siguientes lineas muestran esta aseveración:

$$f : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}, \quad f(A) = LU$$

$$\frac{\|A - \tilde{A}\|_\infty}{\|A\|_\infty} = \frac{10^{-20}}{2} \approx \mathcal{O}(\epsilon_{maq}) \quad \text{pero} \quad f(\tilde{A}) \neq \tilde{f}(A)\,,$$

pues $f(\tilde{A})$ representa la factorización exacta de la matriz perturbada $\tilde{A}$, es decir

$$f(\tilde{A}) = LU = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix} \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix},$$

y $\tilde{f}(A)$ representa la factorización aproximada (de punto flotante) de la matriz exacta, dentro de la computadora, es decir

$$\tilde{f}(A) = \tilde{L}\tilde{U} \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix} \begin{bmatrix} 10^{-20} & 1 \\ 1 & -10^{20} \end{bmatrix}.$$

En el caso general para matrices A de orden $n \times n$ con n grande, la situación empeora. El método de eliminación de Gauss en no es ni estable regresivo ni tampoco estable como algoritmo general para encontrar factorizaciones del tipo LU. Además de esto, las normas de las matrices triangulares L y U obtenidas pueden ser mucho mayores que la norma de la matriz misma A, introduciendo fuentes adicionales de inestabilidad en las faces de sustitución progresiva y regresiva para resolver los sistemas triangulares. En el ejemplo anterior

$$\|\tilde{A}\|_\infty = 2 \quad \text{mientras que} \quad \|L\|_\infty = 10^{20} + 1 \quad \text{y} \quad \|U\|_\infty = 10^{20} - 1,$$

lo cual muestra que, efectivamente, las normas de los factores L y U son desproporcionalmente mayores que la de la matriz dada. $\tilde{A}$.

3.4. Técnicas de pivoteo

Si bién no podemos eliminar la inestabilidad completamente, si podemos controlarla permutando el orden de los renglones y columnas de la matriz del sistema de ecuaciones. A esta técnica se le conoce como *pivoteo* y ha sido usada desde la aparición de las computadoras (alrededor de 1950). El propósito del pivoteo es asegurar que los factores L y U no sean tan grandes comparados con la matriz A. Siempre que las cantidades que aparecen en la eliminación sean manejables, los errores de redondeo se mantendrán controlados y el algoritmo será estable regresivo.

3.4.1. Pivoteo completo

La idea es la siguiente: en el k–ésimo paso de eliminación debemos escoger un pivote de entre los coeficientes del subsistema con matriz (para simplificar la exposición los superíndices se han suprimido)

$$A(k:n,k:n) \equiv \begin{bmatrix} a_{k,k} & a_{k,k+1} & \dots & a_{k,n} \\ a_{k+1,k} & a_{k+1,k+1} & \dots & a_{k+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,k} & a_{n,k+1} & \dots & a_{n,n} \end{bmatrix}.$$

El pivote no necesariamente es $a_{k,k}$ como lo hemos considerado hasta ahora. Con el objeto de controlar el crecimiento de los coeficientes en las matrices de factorización L y U es conveniente escoger como pivote a aquel coeficiente que tiene valor absoluto máximo:

$$|a| = \max_{k\le i,j\le n} |a_{ij}| = |a_{lm}|\,.$$

Hecho esto se procede a hacer el intercambio del renglón k con el renglón l, y de la columna k con la columna m, y se continua la eliminación en la forma usual calculando los multiplicadores y los nuevos coeficientes. Los multiplicadores que se obtienen son tales que

$$m_{i\,k} = \frac{a_{i\,k}}{|a|} \le 1\,, \quad i = k+1,\dots,n$$

y, en consecuencia, ninguno de los coeficientes de la matriz L al final del proceso de eliminación (ó factorización) será mayor a uno. A esta estrategia se le denomina *pivoteo completo*. Sin embargo, esta estrategia en muy poco usada por dos razones:

1. En el paso k hay $(n-k+1)^2$ posibilidades para buscar el máximo, y el costo para seleccionar los pivotes en los $n-1$ pasos de eliminación implica $\mathcal{O}(n^3)$ operaciones, lo cual es excesivo.

2. Hay que darle seguimiento al intercambio de renglones y columas.

3.4.2. Pivoteo parcial

En la práctica, es posible encontrar pivotes tan útiles como los encontrados con pivoteo completo realizando un mucho menor número de operaciones de busqueda. El método más común se denomina *pivoteo parcial.* En esta estrategia se intercambia solamente dos renglones en cada paso de eliminación. Así, en el $k-$ésimo paso de eliminación se escoge como pivote

$$|a| = \max_{k \leq i \leq n} |a_{ik}| = |a_{lk}|$$

y se intercambian los renglones k y l. En este caso hay $n-k+1$ posibilidades para el pivoteo en el $k-$ésimo paso, y por lo tanto el número de operaciones de busqueda en todo el proceso de eliminación es en total $\mathcal{O}(n^2)$ (en realidad $n(n-1)/2$).

Como es usual con otras operaciones en el álgebra lineal numérica, el intercambio de renglones puede expresarse por medio de un producto de matrices. Como vimos anteriormente un paso de eliminación corresponde a la multiplicación izquierda por una matriz triangular inferior L_k en el $k-$ésimo paso. El pivoteo parcial complica un poco más el proceso pues ahora es necesario multiplicar por una *matriz de permutación* P_k por la izquierda antes de cada eliminación.

Matrices de permutación

Una matriz de permutación es una matriz con ceros en todos lados excepto por un coeficiente 1 en cada renglón y columna. Por ejemplo la matriz

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

tiene un sólo 1 en cada renglón y columna, y en todas las demas entradas tiene ceros. Cualquier matriz de permutación es el producto de matrices de permutación elemental. Una matriz de permutación elemental se obtiene de la matriz identidad permutando dos de sus renglones (o dos de sus columnas) solamente. Por ejemplo, las matrices

$$P_{12} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{y} \quad P_{34} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

son matrices de permutación elementales que se obtienen de la matriz identidad en $\mathbb{R}^{4\times 4}$ permutando los renglones (ó columnas) 1 y 2, y permutando los renglones (ó columnas) 3 y 4 respectivamente. La matriz de permutación P dada un poco más arriba se puede expresar como el producto de estas dos matrices, pues

$$P = P_{12}P_{34} = P_{34}P_{12}.$$

Dada cualquier matriz $A \in \mathbb{R}^{4\times 4}$, el producto $P_{12}A$ intercambia los renglones 1 y 2 de la matriz A, y el producto AP_{12} intercambia las columnas 1 y 2 de la matriz A:

$$P_{12}A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} a_{21} & a_{22} & a_{23} & a_{24} \\ a_{11} & a_{12} & a_{13} & a_{14} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$AP_{12} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{12} & a_{11} & a_{13} & a_{14} \\ a_{22} & a_{21} & a_{23} & a_{24} \\ a_{32} & a_{31} & a_{33} & a_{34} \\ a_{42} & a_{41} & a_{43} & a_{44} \end{bmatrix}$$

3.4.3. Factorización LU con pivoteo parcial

Tomando en cuenta el intercambio de renglones en cada paso para realizar el pivoteo parcial, encontramos que, para una matriz no–singular $A \in \mathbb{R}^{n\times n}$, al término de los $n-1$ pasos de eliminación se obtiene la siguiente factorizacón

$$L_{n-1}P_{n-1}\cdots L_2P_2L_1P_1A = U\,.$$

El siguiente ejemplo ilustra esta aseveración.

Ejemplo 3.4. *Encontrar la factorización LU con pivoteo parcial para la matriz*

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = A^{(1)}$$

Solución.

1er paso de eliminación: Claramente el pivote debe ser 8 y hay que intercambiar los renglones 1 y 3

$$P_1A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix}$$

los multiplicadores son: $m_{21} = \frac{4}{8} = \frac{1}{2}$, $m_{31} = \frac{2}{8} = \frac{1}{4}$, $m_{41} = \frac{6}{8} = \frac{3}{4}$. Luego

$$L_1P_1A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ -1/4 & 0 & 1 & 0 \\ -3/4 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & -1/2 & -3/2 & -3/2 \\ 0 & -3/4 & -5/4 & -5/4 \\ 0 & 7/4 & 9/4 & 17/4 \end{bmatrix} = A^{(2)}$$

2o paso de eliminación: Ahora el pivote (para el subsistema 3×3) es $7/4$, debemos intercambiar los renglones 2 y 4

$$P_2L_1P_1A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & -1/2 & -3/2 & -3/2 \\ 0 & -3/4 & -5/4 & -5/4 \\ 0 & 7/4 & 9/4 & 17/4 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & 7/4 & 9/4 & 17/4 \\ 0 & -3/4 & -5/4 & -5/4 \\ 0 & -1/2 & -3/2 & -3/2 \end{bmatrix}$$

los multiplicadores ahora son: $m_{32} = \frac{-3/4}{7/4} = -\frac{3}{7}$, $m_{42} = \frac{-1/2}{7/4} = -\frac{2}{7}$. Así que

$$L_2P_2L_1P_1A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 3/7 & 1 & 0 \\ 0 & 2/7 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & 7/4 & 9/4 & 17/4 \\ 0 & -3/4 & -5/4 & -5/4 \\ 0 & -1/2 & -3/2 & -3/2 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & 7/4 & 9/4 & 17/4 \\ 0 & 0 & -2/7 & 4/7 \\ 0 & 0 & -6/7 & -2/7 \end{bmatrix}$$
$$= A^{(3)}$$

3er paso de eliminación: El pivote (para el subsistema 2×2) es -6/7. Entonces, intercambiamos los renglones 3 y 4

$$P_3L_2P_2L_1P_1A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & 7/4 & 9/4 & 17/4 \\ 0 & 0 & -2/7 & 4/7 \\ 0 & 0 & -6/7 & -2/7 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & 7/4 & 9/4 & 17/4 \\ 0 & 0 & -6/7 & -2/7 \\ 0 & 0 & -2/7 & 4/7 \end{bmatrix}$$

El multiplicador es $m_{43} = \dfrac{-2/7}{-6/7} = \dfrac{1}{3}$. Finalmente

$$L_3P_3L_2P_2L_1P_1A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/3 & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & 7/4 & 9/4 & 17/4 \\ 0 & 0 & -6/7 & -2/7 \\ 0 & 0 & -2/7 & 4/7 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 0 & 7/4 & 9/4 & 17/4 \\ 0 & 0 & -6/7 & -2/7 \\ 0 & 0 & 0 & 2/3 \end{bmatrix}$$
$$= U\,.$$

En el anterior ejemplo hemos encontrado entonces que

$$L_3P_3L_2P_2L_1P_1A = U.$$

Con un poco más de trabajo podemos reescribir esta última igualdad en forma más adecuada. Para ello, definimos

$$L_3' = L_3, \qquad L_2' = P_3L_2P_3^{-1}, \qquad L_1' = P_3P_2L_1P_2^{-1}P_3^{-1}.$$

Se puede verificar directamente que estas últimas matrices son triangulares inferiores y que $L_3'L_2'L_1'P_3P_2P_1 = L_3P_3L_2P_2L_1P_1$. Por lo tanto

$$L_3'L_2'L_1'P_3P_2P_1A = U\,.$$

Entonces, podemos escribir

$$PA = LU \quad \text{con} \quad P = P_3P_2P_1\,, \quad L = (L_3'L_2'L_1')^{-1}\,.$$

Un cálculo directo muestra que

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \qquad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3/4 & 1 & 0 & 0 \\ 1/2 & -2/7 & 1 & 0 \\ 1/4 & -3/7 & 1/3 & 1 \end{bmatrix}.$$

La matriz U ya se calculó al término del proceso de eliminación.

A la anterior factorización se le denomina *factorización LU* de la matriz A con *estrategia de pivoteo simple o parcial*. Por supuesto la factorización LU corresponde, estrictamente hablando, no a A sino a una permutación de la matriz A, a saber PA. Este algoritmo se muestra a continuación:

Algoritmo de factorización LU con pivoteo parcial

Dados los coeficientes a_{ij} de A y los coeficientes b_j de b

Para $k = 1, 2, \ldots, n-1$
. Encontrar $p \geq k$ tal que $|a_{pk}| = \text{máx}_{k \leq i \leq n} |a_{ik}|$
. Intercambiar los renglones p y k (si $p \neq k$)
. Si $|a_{kk}| = 0,$ escribir: "la matriz es singular". Parar y salir
. Si no, hacer para $i = k+1, \ldots, n$
. . $m := a_{ik}/a_{kk}$
. . para $j = k+1, \ldots, n$
. . . $a_{ij} := a_{ij} - m a_{kj}$
. . Fín
. . $b_i := b_i - m b_k$
. Fín
Fín

3.5. Cálculo de la inversa de una matriz

Dada la matriz $A \in \mathbb{R}^{n \times n}$ no singular, el cálculo de su inversa en forma aproximada se puede encontrar resolviendo n sistemas de ecuaciones lineales por medio del método de eliminación de Gauss con pivoteo parcial, como se indica a continuación:

Sea $A^{-1} = [\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n]$, donde $\vec{x}_i \in \mathbb{R}^n$ es el $i-$ésimo vector columna de A^{-1}, entonces

$$AA^{-1} = [A\vec{x}_1, A\vec{x}_2, \ldots, A\vec{x}_n] = I = [\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n]$$

donde $\vec{e}_i = (0, \ldots, 1, \ldots, 0)^T$ es el vector columna con 1 en el $i-$ésimo lugar y 0 en las demás entradas. Luego la igualdad se cumple si

$$A\vec{x}_i = \vec{e}_i, \qquad i = 1, 2, \ldots, n.$$

Resolviendo este conjunto de sistema de ecuaciones lineales, encontramos los vectores columna $\vec{x}_i$ de la matriz inversa A^{-1}. Como los n sistemas de ecuaciones lineales tienen la misma matriz, se puede aplicar eliminación con pivoteo con lado derecho $\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n$ en forma simultanea.

$$[A|I] \quad \rightarrow \quad \text{eliminación con pivoteo} \quad \rightarrow \quad [U|J]$$

donde I es la matriz identidad y U es una matriz triangular superior. La inversa se obtiene resolviendo por sustitución regresiva los sistemas

$$U\vec{x}_i = \vec{j}_i$$

donde $\vec{j}_i$ son los vectores columna de la matriz J, y los $\vec{x}_i$ los vectores solución, es decir, los vectores columna de A^{-1}.

3.6. Estabilidad del método de eliminación de Gauss con pivoteo

El método de eliminación de Gauss con pivoteo parcial es inestable para ciertas matrices (un número pequeño en realidad), pero estable en la práctica (es decir, en la gran mayoría de los problemas que aparecen en la aplicaciones). El análisis de estabilidad de la eliminación de Gauss con pivoteo parcial es complicado y ha sido uno de los temas más difíciles del análisis numérico desde 1950. La inestabilidad del método de eliminación de Gauss, con o sin pivoteo, puede aparecer si uno de los factores L y U es grande comparado con la matriz A. Así que el propósito del pivoteo, desde el punto de vista de la estabilidad, es asegurar que L y U no sean muy grandes. De tal manera que cuando las cantidades intermedias en el proceso de eliminación sean de tamaño manejable, los errores de redondeo serán pequeños, y el método será estable regresivo. Estas ideas se precisan en el siguiente teorema que se establece para la eliminación de Gauss sin pivoteo, pero también puede aplicarse al caso con pivoteo si A representa la matriz original con renglones y columnas permutadas adecuadamente.

Teorema 3.5. *Si $A \in \mathbb{R}^{n\times n}$ tiene una factorización LU y esta se realiza utilizando eliminación de Gauss sin pivoteo en una computadora que satisface el axioma fundamental de la aritmética de punto flotante, entonces las matrices $\tilde{L}$, $\tilde{U}$ satisfacen*

$$\tilde{L}\tilde{U} = A + \delta A \quad con \quad \frac{\|\delta A\|}{\|L\|\,\|U\|} = \mathcal{O}(\epsilon_{maq}) \quad para\ alguna \quad \delta A \in \mathbb{R}^{n\times n}.$$

Aclaración: Observese que el denominador es $\|L\|\,\|U\|$ y no $\|A\|$ en la expresión de arriba. Si $\|L\|\,\|U\|$ fuera de tamaño comparable con $\|A\|$ ($\|L\|\,\|U\| = \mathcal{O}(\|A\|)$) entonces la eliminación de Gauss es un método estable regresivo. Si por el contrario, $\|L\|\,\|U\| \gg \mathcal{O}(\|A\|)$ entonces el algoritmo es inestable.

Crecimiento de factores

Consideremos ahora el caso de la eliminación de Gauss con pivoteo parcial. En este caso se obtiene que la matriz de multiplicadores L tiene entradas que son menores o iguales a 1 en valor absoluto, es decir, $\|L\| = \mathcal{O}(1)$ en cualquier norma matricial $\|\cdot\|$. Por lo tanto

$$\frac{\|\delta A\|}{\|L\|\,\|U\|} = \frac{\|\delta A\|}{\|U\|} = \mathcal{O}(\epsilon_{maq}).$$

De aquí que en este caso se concluye que el algoritmo es estable regresivo si $\|U\| = \mathcal{O}(\|A\|)$. Para que la norma de U sea comparable con la norma de A basta que los coeficientes de U no sean mucho mayornes que los de A. Es decir, hay que considerar como se amplifica los coeficientes al reducir A a la matriz U en el proceso de eliminación. En particular, sea

$$\rho = \frac{\max |u_{ij}|}{\max |a_{ij}|}$$

el *factor de crecimiento*. Si ρ es de orden 1, entonces no hay mucho crecimiento, y el proceso de eliminación es estable. Por otro lado si ρ es mucho mayor que $\mathcal{O}(1)$, entonces podemos esperar que haya inestabilidad en el proceso de eliminación. Observese que si $\rho = \mathcal{O}(1)$, entonces de la igualdad anterior se tiene $\|U\| = \mathcal{O}(\rho(\|A\|))$. Esto se resume en el siguiente teorema.

Teorema 3.6. *Supongase que la factorización $PA = LU$ se lleva a cabo por medio de eliminación de Gauss con pivoteo parcial en una computadora que satisface el axioma fundamental de la aritmética de punto flotante. Entonces las matrices calculadas $\tilde{P}$, $\tilde{L}$ y $\tilde{U}$ satisfacen*

$$\tilde{L}\tilde{U} = \tilde{P}A + \delta A \quad con \quad \frac{\|\delta A\|}{\|A\|} = \mathcal{O}(\rho\,\epsilon_{maq})$$

para alguna $\delta A \in \mathbb{R}^{n\times n}$. Asimismo la eliminación Gaussiana es un método estable regresivo si $\rho = \mathcal{O}(1)$ uniformemente sobre todas las matrices en $\mathbb{R}^{n\times n}$, y no lo es en caso contrario.

El peor caso de inestabilidad

Considere la matriz

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix}.$$

Al aplicar el método de eliminación de Gauss se tiene

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 0 & 16 \end{bmatrix}.$$

Al hacer los cálculos detallados el lector puede convencerse que no se hace intercambio de renglones aún y cuando se aplique pivoteo parcial. En esta matriz 5×5, el factor de crecimiento es $\rho = \frac{\text{máx}\,|u_{ij}|}{\text{máx}\,|a_{ij}|} = 16/1 = 2^4$. Por otro lado, para la matriz análoga de orden $n \times n$ el factor de crecimiento al hacer eliminación de Gauss es $\rho = 2^{n-1}$. Así que, de acuerdo al teorema anterior, tendriamos

$$\tilde{L}\tilde{U} = \tilde{P}A + \delta A \quad \text{con} \quad \frac{\|\delta A\|}{\|A\|} = \mathcal{O}(\rho\epsilon_{maq}) = \mathcal{O}(2^{n-1}\,\epsilon_{maq}).$$

Este resutlado implica que habría una perdida de $n-1$ bits de precisión al hacer eliminación de Gauss con pivoteo, lo cual es intolerable para cálculos practicos a medida que el tamaño n del sistema aumenta. En realidad este es un ejemplo extremo de corte puramente académico y, afortunadamente, la gran mayoría de las matrices que aparecen como resultado de problemas prácticos no exhiben este tipo de comportamiento. De hecho, en 50 años de computación, no se ha visto que aparezcan, bajo circunstancias naturales, problemas matriciales que exhiben una inestabilidad tan dramática.

3.7. Método de Factorización de Choleski

Para matrices simétricas y definidas positivas el proceso de eliminación de Gauss, y por tanto la factorización LU, puede realizarse en forma más eficiente. El método que se utiliza se denomina *factorización de Choleski*. Este algorimto opera en el lado izquierdo y derecho de la matriz explotando la simetría. Este algoritmo descompone las matrices simétricas y definidas positivas en factores triangulares haciendo la mitad de las operaciones que las necesarias para matrices generales.

3.7.1. Matrices definidas positivas

Una matriz $A \in \mathbb{R}^{n\times n}$ se dice que es *definida positiva* si $x^T Ax > 0$ para todo $x \in \mathbb{R}^n$ con $x \neq \vec{0}$. Si A es una matriz definida positiva, algunas de sus propiedades importantes son:

1. A es no singular.

2. Los valores propios de A son todos reales y positivos.

3. El determinante de la matriz A y de cada uno de sus n menores.

$$A(1:k,1:k) \equiv \begin{bmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} \end{bmatrix},$$

   $k = 1, \ldots, n$, es siempre mayor a cero.

4. Cualquier submatriz principal de A de la forma $A(1:k,1:k)$ ó de la forma

$$A(k:n,k:n) \equiv \begin{bmatrix} a_{kk} & \cdots & a_{kn} \\ \vdots & & \vdots \\ a_{nk} & \cdots & a_{nn} \end{bmatrix},$$

   $k = 1, \ldots, n$, es definida positiva.

5. Cada uno de los pivotes obtenidos en el poceso de eliminación de Gauss aplicado a la matriz A es mayor a cero.

Se deja al lector verificar las propiedades 3 y 5. Aquí verificaremos el resto de las propiedades.

Verificación de la propiedad 1. La propiedad 1 es consecuencia directa de la propiedad 3.

Verificación de la propiedad 2. Si λ es un valor propio de $A \in \mathbb{R}^{n\times n}$ y $x \in \mathbb{R}^n$ es el vector propio correspondiente, entonces $x \neq \vec{0}$ y

$$x^T Ax = x^T \lambda x = \lambda \|x\|_2^2$$

así que

$$\lambda = \frac{x^T Ax}{\|x\|_2^2} > 0\,.$$

Verificación de la propiedad 4. Para todo vector $x = [x_1, \ldots, x_k]^T \in \mathbb{R}^k$ no cero se tiene

$$[x_1, \ldots, x_k]A(1:k, 1:k)\begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} = [x_1, \ldots, x_k, 0, \ldots, 0]A\begin{bmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} > 0\,,$$

por ser A definida positiva. Por lo tanto la submatriz $A(1:k, 1:k)$ es definida positiva. En forma análoga se puede verificar que la submatriz $A(k:n, k:n)$ es definida positiva si A lo es.

3.7.2. Factorización de Choleski

Sea $A \in \mathbb{R}^{n\times n}$ una matriz simétrica y definida positiva. Nuestro propósito es descomponer esta matriz en factores triangulares explotando las propiedades de la matriz. Con el objeto de simplificar la exposición primero realizaremos un paso de eliminación para el caso especial en el que $a_{11} = 1$, es decir cuando la matriz es de la forma

$$A = \begin{bmatrix} 1 & \omega_2 & \ldots & \omega_n \\ \omega_2 & & & \\ \vdots & & A(2:n, 2:n) & \\ \omega_n & & & \end{bmatrix}$$

donde $\omega_2 = a_{12}$, $\omega_3 = a_{13}$,..., $\omega_n = a_{1n}$ y $A(2:n, 2:n)$ es la submatriz principal inferior de orden $n-1$. Al realizar el primer paso de eliminación sin pivoteo en forma matricial obtenemos

$$A = \begin{bmatrix} 1 & \omega_2 \ \ldots \ \omega_n \\ \omega_2 & \\ \vdots & A(2:n, 2:n) \\ \omega_n & \end{bmatrix} = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ \omega_2 & 1 & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ \omega_n & 0 & \ldots & 1 \end{bmatrix}\begin{bmatrix} 1 & \omega_2 \ \ldots \ \omega_n \\ 0 & \\ \vdots & \quad M_2 \\ 0 & \end{bmatrix} = L_1 A^{(1)}$$

donde ω_2, ω_3,..., ω_n son los multiplicadores y $M_2 = A(2:n, 2:n) - \omega\omega^T$ con $\omega\omega^T$ el producto externo dado por

$$\begin{bmatrix} \omega_2 \\ \vdots \\ \omega_n \end{bmatrix}\begin{bmatrix} \omega_2 & \ldots & \omega_n \end{bmatrix} = \begin{bmatrix} \omega_2\omega_2 & \ldots & \omega_2\omega_n \\ \omega_3\omega_2 & \ldots & \omega_3\omega_n \\ \vdots & \ddots & \vdots \\ \omega_n\omega_2 & \ldots & \omega_n\omega_n \end{bmatrix}.$$

Al término del primer paso de eliminación se inducen ceros en la primera columna, pero quisieramos mantener simetría. Observese que la matriz $A^{(1)}$ no es simétrica aún y cuando M_2 lo es. Para obtener simetría, procedemos en forma análoga haciendo eliminación derecha en el primer renglón de $A^{(1)}$ (sustracción de múltiplos de la primera columna de la restante columna). Obtenemos

$$\begin{aligned} A &= \begin{bmatrix} 1 & 0 & \dots & 0 \\ \omega_2 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ \omega_n & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & M_2 & \\ 0 & & & \end{bmatrix} \begin{bmatrix} 1 & \omega_2 & \dots & \omega_n \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \\ &= \quad L_1 \qquad\qquad A_s^{(1)} \qquad\qquad L_1^T \end{aligned}$$

Observe que la matriz $A_s^{(1)}$ ahora si es simétrica, dado que M_2 los es. La idea de la factorización de Choleski es continuar este proceso con la submatriz $A_s^{(1)}$ y así sucesivamente hasta obtener una matriz identidad en el último paso de eliminación simétrica.

Queremos extender el anterior proceso para el caso en que la matriz A definida positiva sea tal que $a_{11} > 0$ en lugar de $a_{11} = 1$. La generalización se obtiene ajustando algunos elementos de las matrices L_1 por un factor $\sqrt{a_{11}}$. Concretamente, si

$$A = \begin{bmatrix} a_{11} & \omega_2 & \dots & \omega_n \\ \omega_2 & & & \\ \vdots & & A(2:n, 2:n) & \\ \omega_n & & & \end{bmatrix},$$

entonces

$$\begin{aligned} A &= \begin{bmatrix} \sqrt{a_{11}} & 0 & \dots & 0 \\ \frac{\omega_2}{\sqrt{a_{11}}} & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ \frac{\omega_n}{\sqrt{a_{11}}} & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & M_2 & \\ 0 & & & \end{bmatrix} \begin{bmatrix} \sqrt{a_{11}} & \frac{\omega_2}{\sqrt{a_{11}}} & \dots & \frac{\omega_n}{\sqrt{a_{11}}} \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \\ &= \quad L_1 \qquad\qquad A_s^{(1)} \qquad\qquad L_1^T \end{aligned}$$

donde

$$M_2 = A(2{:}n, 2{:}n) - \frac{\omega\omega^T}{a_{11}}.$$

La matriz $A_s^{(1)}$ es simétrica, pues las submatrices $A(2:n,\, 2:n)$ y $\omega\omega^T$ lo son. Además $A_s^{(1)}$ también es definida positiva, pues si $x \in \mathbb{R}^n$, $x \neq \overrightarrow{0}$, entonces $y = (L_1^{-1})^T x \neq \overrightarrow{0}$ dado que L_1^{-1} es no singular y, por lo tanto

$$x^T A^{(1)} x = x^T L_1^{-1} A (L_1^{-1})^T x = y^T A y > 0$$

Volviendo a aplicar el mismo procedimiento a $A_s^{(1)}$ se obtiene

$$A = L_1\, L_2\, A_s^{(2)}\, L_2^T L_1^T$$

donde

$$A_s^{(2)} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & K & \\ 0 & 0 & & \end{bmatrix},$$

con K matriz simétrica y definida positiva. Luego, el proceso puede continuarse en forma sucesiva (pues todas las submatrices K son definidas positivas y simétricas) hasta obtener

$$A = \underbrace{L_1 L_2 \dots L_n}_{L}\, I\, \underbrace{L_n^T L_{n-1}^T \dots L_1^T}_{L^T}.$$

Lo anterior se resume en el siguiente resultado

Teorema 3.7. *Cualquier matriz $A \in \mathbb{R}^{n\times n}$ simétrica y definida positiva tiene una única factorización de Choleski $A = LL^T$, donde L es una matriz triangular inferior no singular.*

Nota: Dado que $L^T = U$ es una matriz triangular superior, también podemos escribir $A = U^T U$

3.7.3. El algoritmo de Choleski

Cuando el algoritmo de Choleski se programa sólo se necesita almacenar la parte triangular superior de A ó bién la parte triangular inferior. Esta simplificación permite reducir el número de operaciones a la mitad para lograr la factorización $A = LL^T$. El algortimo se puede construir realizando comparación de los coeficientes:

Sean $A = (a_{ij})_{1\leq i,\, j\leq n}$ con $a_{ij} = a_{ji}$, y $L = (l_{ij})_{1\leq i,\, j\leq n}$ con $l_{ii} \neq 0$, $i = 1, \dots, n$ y $l_{ij} = 0$ si $j > i$. Comparando los coeficientes en la ecuación matricial $A = LL^T$, se obtiene

$$a_{ii} = (i\text{-ésimo renglón de } L) \times (i\text{-ésima columna de } L^T),$$

es decir

$$a_{ii} = \sum_{k=1}^{i} l_{ik} l_{ik} = \sum_{k=1}^{i-1} l_{ik}^2 + l_{ii}^2 \,.$$

Por lo tanto

$$l_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right)^{1/2} \qquad i = 1, \ldots, n.$$

Análogamente

$$a_{ij} = (i\text{-ésimo renglón de } L) \times (j\text{-ésima columna de } L^T) = \sum_{k=1}^{\min(i,j)} l_{ik} l_{jk} \,.$$

Considerando el caso $i > j$:

$$a_{ij} = \sum_{k=1}^{j} l_{ik} l_{jk} = \sum_{k=1}^{j-1} l_{ik} l_{jk} + l_{ij} l_{jj} \,,$$

entonces

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}}{l_{jj}} \qquad i = j+1, \ldots, n.$$

Obervese que en este método no hay intercambio de renglones o pivoteo. A continuación se muestra el algoritmo de factorización de Cholesky.

Algoritmo

$l_{11} = \sqrt{a_{11}}$

Para $i = 2, \ldots, n$

· $l_{i1} = a_{i1}/l_{11}$

Fín

Para $j = 2, \ldots, n-1$

· $l_{jj} = \left(a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 \right)^{1/2}$

· Para $i = j+1, \ldots, n$

· · $l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) / l_{jj}$

· Fín

Fín

$l_{nn} = \left(a_{nn} - \sum_{k=1}^{n-1} l_{nk}^2 \right)^{1/2}$

Ejemplo 3.8. *Aplicar el algoritmo de Choleski para factorizar en la forma* $A = LL^T$ *la siguiente matriz, que es simétrica definida positiva.*

$$A = \begin{bmatrix} 4 & -2 & 0 & -4 \\ -2 & 10 & 3 & 2 \\ 0 & 3 & 2 & 3 \\ -4 & 2 & 3 & 29 \end{bmatrix}.$$

Solución. La matriz es simétrica, y puede verificarse (calculando sus valores propios) que es definida positiva. Entonces, aplicando el algoritmo anterior obtenemos

$l_{11} = \sqrt{a_{11}} = \sqrt{4} = 2$

.....

$l_{21} = a_{21}/l_{11} = -2/2 = -1$

$l_{31} = a_{31}/l_{11} = 0/2 = 0$

$l_{41} = a_{41}/l_{11} = -4/2 = -1$

.....

$l_{22} = (a_{22} - l_{21}^2)^{1/2} = (10 - (-1)^2)^{1/2} = 3$

$l_{32} = (a_{32} - l_{31}l_{21})/l_{22} = (3 - (0)(-1))/3 = 1$

$l_{42} = (a_{42} - l_{41}l_{21})/l_{22} = (2 - (-2)(-1))/3 = 0$

.....

$l_{33} = (a_{33} - l_{31}^2 - l_{32}^2)^{1/2} = (2 - 0^2 - 1^2)^{1/2} = 1$

$l_{43} = (a_{43} - l_{41}l_{31} - l_{42}l_{32})/l_{33} = (3 - (-2)(0) - (0)(1))/1 = 3$

.....

$l_{44} = (a_{44} - l_{41}^2 - l_{42}^2 - l_{43}^2)^{1/2} = (29 - (-2)^2 - 0^2 - 3^2)^{1/2} = 4$

Luego la factorización de Choleski es

$$\begin{bmatrix} 4 & -2 & 0 & -4 \\ -2 & 10 & 3 & 2 \\ 0 & 3 & 2 & 3 \\ -4 & 2 & 3 & 29 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ -1 & 3 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ -2 & 0 & 3 & 4 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & -2 \\ 0 & 3 & 1 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

$$A \qquad = \qquad L \qquad L^T.$$

3.7.4. Estabilidad del algoritmo de Choleski

Todas las sutilezas que aparecen en el análisis de estabilidad de la eliminación de Gauss desparecen cuando se utiliza la factorización de Choleski. El algoritmo de Choleski no utiliza ninguna técnica de pivoteo y siempre es estable. Podemos verificar esta aseveración estimando el factor de crecimiento ρ al hacer la factorización. Sabemos que

$$a_{ii} = \sum_{k=1}^{i} l_{ik}^2 \,.$$

y si suponemos que $|l_{ik}| \geq 1$, entonces

$$|l_{ik}| \leq |l_{ik}|^2 \leq a_{ii} \leq \max_{1 \leq i,j \leq n} |a_{ij}| \quad \forall\, i,k.$$

Esto implica que

$$\max |l_{ik}| \leq \max |a_{ij}| \,.$$

Por lo tanto

$$\rho = \frac{\max |l_{ik}|}{\max |a_{ij}|} \leq 1 \,.$$

En consecuencia, el factor de crecimiento es $\mathcal{O}(1)$, y el algoritmo siempre es estable regresivo.

Capítulo 4

Problemas de mínimos cuadrados lineales. Factorización QR

El término "mínimos cuadrados" describe un enfoque frecuentemente usado para resolver sistemas de ecuaciones sobredeterminados ó especificados inexactamente en algún sentido apropiado. En lugar de resolver las ecuaciones exactamente, buscaremos solamente minimizar la suma de los cuadrados de los residuales.

El criterio de mínimos cuadrados tiene interpretaciones estadísticas importantes. Si se hacen suposiciones probabilísticas apropiadas acerca de la distribución del error, el enfoque de mínimos cuadrados produce lo que se conoce como la estimación de máxima verisimilitud de los parámetros. Apesar de que ciertas suposiciones probabilísticas no se cumplan, durante años se ha verificado que los métodos de mínimos cuadrados producen resultados útiles.

4.1. Ajuste de curvas

Existen muchos problemas en las aplicaciones que pueden abordarse utilizando el enfoque de mínimos cuadrados. Una fuente común que da origen a problemas de mínimos cuadrados es el ajuste de curvas a un conjunto de datos dados: Sea x una variable independiente y sea $y(x)$ una función desconocida de x la cual queremos aproximar. Suponiendo que tenemos m observaciones

$$(x_1,\, y_1),\, (x_2,\, y_2), \ldots,\, (x_m,\, y_m)\,,$$

donde $y_i = y(x_i)$, $i = 1, 2, \ldots,\, m$, la idea es modelar $y(x)$ por medio de una combinación de n funciones base $\phi_1(x)$, $\phi_2(x), \ldots,\, \phi_n(x)$. Es decir, suponer que la función que se ajusta a

los datos es una combinación lineal de la forma

$$y(x) \approx c_1\phi_1(x) + c_2\phi_2(x) + \ldots + c_n\phi_n(x)$$

en donde generalmente el número de funciones base n es menor que el número de datos m. Es decir, $m \geq n$. Entonces, los datos deben satisfacer de manera aproximada

$$y(x_i) \approx c_1\phi_1(x_i) + c_2\phi_2(x_i) + \ldots + c_n\phi_n(x_i), \quad i = 1, 2, \ldots, m\,.$$

La última expresión constituye un sistema de m ecuaciones con n incógnitas c_1, c_2,..., c_n, que en forma matricial puede expresarse de la siguiente manera:

$$\begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \ldots & \phi_n(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \ldots & \phi_n(x_2) \\ \vdots & & \ddots & \vdots \\ \phi_1(x_m) & \phi_2(x_m) & \ldots & \phi_n(x_m) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \quad m \geq n.$$

A la matriz de este sistema $A = (a_{ij})$ con $a_{ij} = \phi_j(x_i)$ se le denomina ***matriz de diseño***. Dado que $m \geq n$, entonces el sistema tiene más renglones (ecuaciones) que columnas (inógnitas). A este tipo de sistemas se les denomina *sobredeterminados* y generalmente no tienen solución. Las funciones base $\phi_i(x)$, $i = 1, \ldots, n$, pueden ser funciones no lineales de x, pero los coeficientes y parámetros c_j aparecen en el modelo en forma lineal cuando se trata de un ajuste lineal.

Dependiendo del problema particular y el objeto de estudio, las funciones base $\phi_i(x)$ pueden escogerse de muchas maneras, e incluso pueden depender de ciertos parámetros. Algunas elecciones comunes pueden ser, entre otras

- Polinomios: $\phi_i(x) = x^{i-1}$.
- Funciones racionales: $\phi_i(x) = x^{i-1}/(\alpha_0 + \alpha_1 x + \cdots + \alpha_{n-1}x^{n-1})$, con $\alpha_0, \ldots, \alpha_{n-1}$ parámetros dados.
- Exponenciales: $\phi_i(x) = e^{-\lambda_i x}$, con parámetros de decaimiento λ_i.
- Gaussianas: $\phi_i(x) = e^{-\left(\frac{x-\mu_i}{\sigma_i}\right)^2}$, con medias y varianzas μ_i, σ_i.

para $i = 1, \ldots, n$. En el presente estudio solo consideraremos el estudio de ajuste de datos por medio de polinomios.

4.2. Ajuste por medio de polinomios

4.2.1. Polinomio de interpolación

Cuando se tienen m observaciones $(x_1,\, y_1)$, $(x_2,\, y_2), \ldots,\, (x_m,\, y_m)$, y se escogen funciones base polinomiales $\phi_i(x) = x^{i-1}$, $i = 1, \ldots,\, n$, la curva de ajuste toma la forma

$$y(x) = c_1 + c_2 x + \cdots + c_n x^{n-1} \quad \text{con} \quad m \geq n.$$

El caso particular en el que $m = n$ da lugar al problema de encontrar el *polinomio de interpolación*. Es decir, suponiendo que los m puntos son distintos nuestro problema consiste en encontrar el polinomio de grado menor o igual a $m - 1$

$$p(x) = c_1 + c_2 x + \cdots + c_m x^{m-1}$$

que interpola los puntos (x_1, y_1), $(x_2, y_2), \ldots,\, (x_m, y_m)$, es decir, $p(x_i) = y_i$, $i = 1, \ldots,\, m$. El sistema de ecuaciones obtenido es un sistema cuadrado de la forma

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{m-1} \\ \vdots & & & \ddots & \vdots \\ 1 & x_m & x_m^2 & \ldots & x_m^{m-1} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}}_{\vec{c}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}_{\vec{y}}$$

que se denomina *sistema cuadrado de Vandermonde*. La matriz A del sistema se denomina *matriz de Vandermonde* y es *no-singular* si los puntos $x_1,\, x_2, \ldots,\, x_m$ son diferentes. De hecho puede demostrarse que

$$det(A) = \prod_{i>j} (x_i - x_j) \neq 0$$

por ser los x_i, $i = 1, \ldots,\, m$, distintos. Por lo tanto, los coeficientes $c_1,\, c_2, \ldots,\, c_m$, son únicos si los puntos de interpolación son todos distintos.

Ejemplo 4.1. *Encontrar el polinomio de grado* ≤ 10 *que interpola los 11 puntos*

$$\begin{aligned} x &= \text{-3.0, -2.0, -1.0, 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0} \\ y &= 0.0,\ \ 0.0,\ \ 0.0,\ 1.0,\ 1.0,\ 1.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0 \end{aligned}$$

Solución. Podemos utilizar el ambiente *MATLAB* para construir la matriz asociada de Vandermonde de orden 11×11

$$\begin{bmatrix} 1 & -3 & (-3)^2 & \dots & (-3)^{10} \\ 1 & -2 & (-2)^2 & \dots & (-2)^{10} \\ & \vdots & & \ddots & \vdots \\ 1 & 7 & (7)^2 & \dots & (7)^{10} \end{bmatrix}$$

Tambien puede observarse que la matriz está mal condicionada pues $cond(A) \sim 10^9$, lo cual permite anticipar que debemos utilizar pivoteo para resolver el sistema cuadrado de Vandermonde. Utilizando el algorimto de factorización LU con pivoteo encontramos la siguiente solución aproximada (para los coeficientes del polinomio)

$$\begin{aligned} c \quad = \quad & 1.000000,\ \text{-}0.059524,\ \text{-}0.454683,\ 0.761078,\ \text{-}0.112004,\ \text{-}0.208160 \\ & 0.072535,\ \ 0.005704,\ \text{-}0.005804,\ 0.000901,\ \text{-}0.000045 \end{aligned}$$

La Figura 4.1 muestra los datos junto con la gráfica del polinomio de interpolación calculado. Observese que el ajuste no es satisfactorio pues, aunque la curva $y = p(x)$ pasa por los puntos

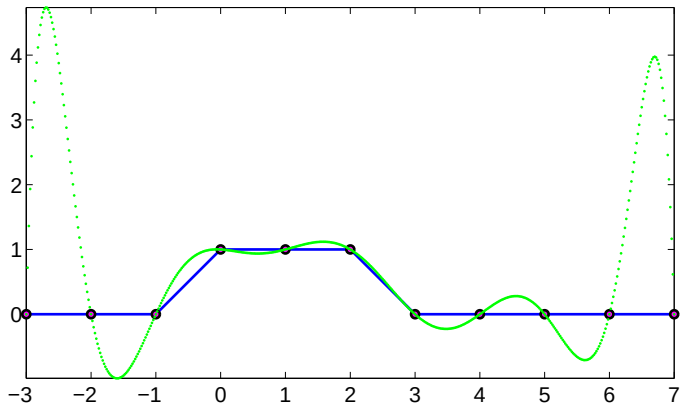


Figura 4.1: Polinomio de interpolación con 11 puntos.

de interpolación, cerca de los extremos muestra oscilaciones fuertes. Estas oscilaciones son un artificio típico de cualquier proceso de interpolación como veremos en el capítulo 4. Este mal ajuste empeora si se utilizan más puntos de interpolación como puede constatarse si en lugar de 11 puntos se usan 21 puntos de interpolación: los 11 puntos ya dados mas los 10 puntos intermedios adicionales. Los valores de x y y en este caso son

$$
\begin{aligned}
x &= \text{-3.0, -2.5, -2.0, -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5,} \\
&\quad \text{4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0} \\
y &= \text{0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0, 0.5, 0.0, 0.0,} \\
&\quad \text{0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0}
\end{aligned}
$$

La Figura 4.2 muestra la gráfica del polinomio de interpolación obtenido con los 21 puntos. Como puede observarse la solución no es buena, pero ilustra lo que pasa si se aumenta el

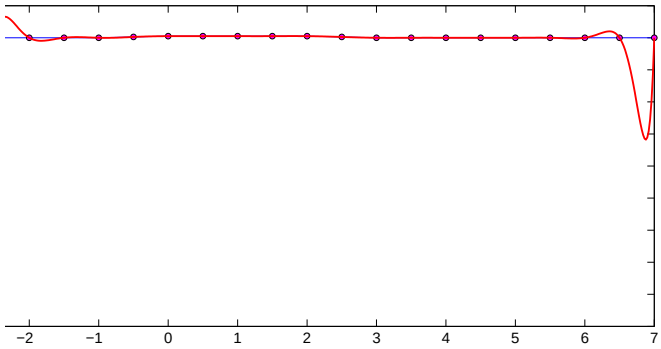


Figura 4.2: Polinomio de interpolación con 21 puntos.

número de puntos de interpolación: las oscilaciones en los extremos son de mucho mayor magnitud. Observese que el número de condición la matriz de Vandermonde en este caso aumenta respecto de aquel en el caso anterior, pues ahora $cond(A) \sim 10^{18}$. De hecho el proceso de interpolación puede ser mal condicionado (sensitivo a perturbaciones en los datos) si se utilizan puntos igualmente espaciados como en los ejemplos anteriores. Para evitar este problema se podrían utilizar los puntos de interpolación distribuidos en forma no uniforme. Sin embargo, en las aplicaciones uno no puede escojer los puntos a modo.

4.2.2. Mínimos cuadrados polinomiales

Sin cambiar los datos podemos obtener mejores resultados reduciendo el orden del polinomio. Dados los puntos $(x_1, y_1), \ldots, (x_m, y_m)$, consideremos el polinomio

$$p(x) = c_1 + c_2 x + \cdots + c_n x^{n-1} \quad \text{con } n < m\,.$$

Este polinomio será un ajuste de mínimos cuadrados para los datos si minimiza la suma de cuadrados

$$\sum_{i=1}^{m} (p(x_i) - y_i)^2\,.$$

Es decir, para encontrar el ajuste de mínimos cuadrados debemos encontrar el vector de coeficientes $c = (c_1, \ldots, c_n)^T$ que resuelve el problema

$$\min_{c\in\mathbb{R}^n} \sum_{i=1}^{m} (c_1 + c_2 x_i + \cdots + c_n x_i^{n-1} - y_i)^2 \,.$$

Esta suma de cuadrados es igual al cuadrado de la norma euclideana del residual, $\|r\|_2^2$, para el sistema rectangular de Vandermonde:

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{n-1} \\ & \vdots & & \ddots & \vdots \\ 1 & x_m & x_m^2 & \ldots & x_m^{n-1} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}}_{c} \approx \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}_{y}, \quad n < m \,,$$

es decir, $r = Ac - y$. Entonces, el problema consiste en resolver

$$\min_{c\in\mathbb{R}^n} \|Ac - y\|_2^2$$

dados A de orden $m \times n$ y $y \in \mathbb{R}^m$ con $m > n$.

4.3. Método de ecuaciones normales

¿Cómo se puede resolver el problema anterior? Más aún, ¿cómo podemos resolver los problemas de mínimos cuadrados en general?

A continuación intentaremos dar respuesta a estas preguntas. Con el objeto de hacer más clara la exposición cambiamos un poco la notación: denotamos por x a la incógnita c, y por b el lado derecho y, en el anterior problema. Así pues, nuestro objetivo es encontrar el punto Ax más cercano al vector b, es decir el que ***minimiza la norma del residual*** $r = b - Ax$.

4.3.1. Proyección ortogonal sobre el espacio imagen

La idea principal para generar algoritmos que resuelvan el anterior problema de minimización descansa en el concepto de ***proyección ortogonal***. Si denotamos por $Im(A)$ al espacio imagen de A, y por

$$P : \mathbb{R}^m \to Im(A)$$

la proyección ortogonal que mapea $\mathbb{R}^m$ sobre el espacio imagen de A, entonces el valor de x que minimiza la norma de $r = b - Ax$ es aquel que satisface $Ax = Pb$. Esta idea se ilustra en

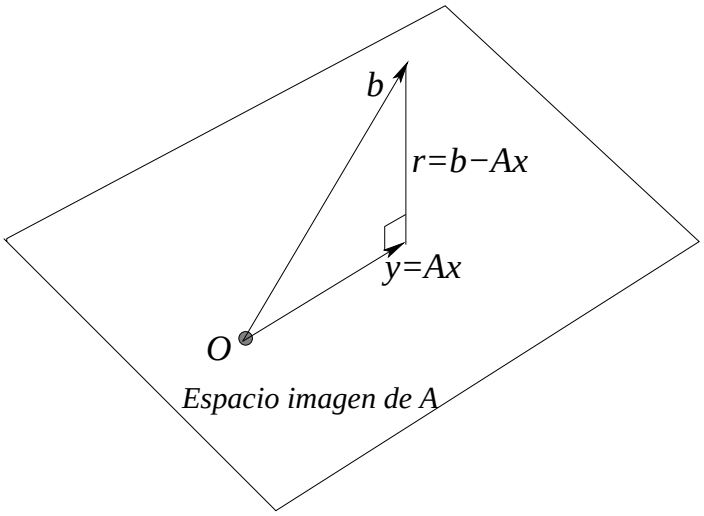


Figura 4.3: El espacion imagen de A es ortogonal al residual.

la Figura 4.3. En otras palabras, el residual $r = b - Ax$ debe ser ortogonal al espacio imagen de A. El espacio imagen de A es generado por los vectores columna de A. Es decir, si

$$A = [\quad a_1 \quad | \quad a_2 \quad | \quad \cdots \quad | \quad a_n \quad],$$

donde a_i es el i-ésimo vector columna de A, entonces

$$Im(A) = gen\{a_1, a_2, \ldots, a_n\}.$$

Demostración. Dado $x \in \mathbb{R}^n$ con $x = (x_1, x_2, \ldots, x_n)^T$, $Ax \in gen\{a_1, \ldots, a_n\}$ pues

$$\begin{aligned} Ax &= [\quad a_1 \quad | \quad a_2 \quad | \quad \cdots \quad | \quad a_n \quad]\, x \\ &= x_1 a_1 + x_2 a_2 + \cdots + x_n a_n. \end{aligned}$$

□

4.3.2. Sistema de ecuaciones normales

Del resultado anterior se tiene que $r = b - Ax$ es ortogonal a $Im(A) = gen\{a_1, a_2, \ldots, a_n\}$

$$\begin{array}{lll} \text{si y solo si} & a_i^T r = 0 & \forall\, i = 1, \ldots, n \\ \text{si y solo si} & A^T r = \vec{0} & \\ \text{si y solo si} & A^T(b - Ax) = \vec{0} & \\ \text{si y solo si} & A^T A x = A^T b & \end{array}$$

Este último sistema de ecuaciones para x es conocido como el ***sistema de ecuaciones normales*** para el problema de mínimos cuadrados. El siguiente resultado es de fundamental importancia.

Teorema 4.2. *Si la matriz A es de orden $m \times n$ con $m \geq n$ y tiene rango completo (todos sus vectores columna son linealmente independiente), entonces A^TA es una matriz cuadrada de orden $n \times n$ que es no singular.*

Demostración. Dado que A es de $m \times n$, entonces A^T es de orden $n \times m$ y A^TA debe ser de orden $n \times n$. Para demostrar que "A de rango completo implica A^TA no singular", basta con demostrar que si A^TA es singular entonces es de rango deficiente:

$$\begin{aligned} A^TA \text{ singular} &\Rightarrow A^TAx = \vec{0} \text{ para algún vector } x \neq \vec{0} \text{ en } \mathbb{R}^n \\ &\Rightarrow x^TA^TAx = 0, \quad x \neq \vec{0} \\ &\Rightarrow \|Ax\|_2^2 = 0, \quad x \neq \vec{0} \\ &\Rightarrow Ax = \vec{0}, \quad x \neq \vec{0} \\ &\Rightarrow A \text{ es singular} \\ &\Rightarrow A \text{ tiene rango deficiente.} \end{aligned}$$

Otro resultado importante es

Teorema 4.3. *Si A es de rango completo, entonces A^TA es una matriz simétrica y positiva definida.*

Demostración. Como A es de rango completo, A^TA es no singular, y además claramente simétrica. Ahora bién, dado $x \neq \vec{0}$ en $\mathbb{R}^n$, entonces, como A^TA es no singular, se satisface $x^TA^TAx = \|Ax\|_2^2 > 0$. Se concluye que A^TA debe ser positiva definida. □

De los dos resultados anteriores se deduce que si la matriz A es de rango completo, entonces la solución del sistema de ecuaciones normales ($A^TAx = A^Tb$) es única e igual a

$$x = (A^TA)^{-1}A^Tb.$$

Además esta solución puede obtenerse por medio del algoritmo de factorización de Choleski dado que A^TA es simétrica y definida positiva.

Nota. Cuando A es de rango completo, dado que $x = (A^TA)^{-1}A^Tb$ es única, a la matriz $A^+ \equiv (A^TA)^{-1}A^T$ se le denomina la *seudoinversa* de A. Además como $x = A^+b$, entonces $Ax = AA^+b$ y $P = AA^+$ es la *matriz de proyección*, es decir $AA^+b = Pb$ es la proyección de b sobre el espacio imagen de A.

4.3.3. Algoritmo de ecuaciones normales

Si A es una matriz $m \times n$ con $m \geq n$, y A es de rango completo, entonces $A^T A$ es no singular, simétrica y definida positiva. Por tanto el sistema de ecuaciones normales $A^T Ax = A^T b$ puede resolverse utilizando el método de Choleski:

Algoritmo. Dados A de $m \times n$ de rango completo y $b \in \mathbb{R}^m$

1. Calcular $A^T A$ y el vector $A^T b$.
2. Calcular la factorización de Choleski $A^T A = LL^T$.
3. Resolver para $z \in \mathbb{R}^n$ el sistema triangular inferior $Lz = A^T b$ (sustitución progresiva)
4. Resolver para $x \in \mathbb{R}^n$ el sistema triangular superior $L^T x = z$ (sustitución regresiva)

Ejemplo 4.4. *Considere de nuevo el problema de ajustar los once datos del problema 4.1, pero ahora utilizando un polinomio de grado 6,* $p(x) = c_1 + c_2 x + \cdots + c_6 x^5 + c_7 x^6$.

Solución. Debemos encontrar los coeficientes $c = (c_1, c_2, \ldots, c_7)^T$. En este caso la matriz de diseño A es la matriz de Vandermonde de tamaño $m \times n = 11 \times 7$

$$\begin{bmatrix} 1 & -3 & (-3)^2 & \ldots & (-3)^6 \\ 1 & -2 & (-2)^2 & \ldots & (-2)^6 \\ & \vdots & & \ddots & \vdots \\ 1 & 7 & (7)^2 & \ldots & (7)^6 \end{bmatrix}$$

y el lado derecho es el vector

$$b = y = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

Observese que A es de rango completo, ninguna columna es combinación lineal de las otras. Entonces los coeficientes $c = (c_1, c_2, \ldots, c_7)$ satisfacen la ecuación

$$A^T A\, c = A^T b$$

donde $A^T A$ es simétrica definida positiva de orden 7×7. Observese que el número de condición de la matriz es $\kappa(A^T A) \sim 10^{10}$. Utilizando el método de factorización de Choleski

para resolver el sistema de ecuaciones normales, en aritmética de doble precisión, obtenemos la solución

$$c_1 = \ 0.822788975730632, \quad c_2 = \ 0.412287124639659, \quad c_3 = -0.173651218063367,$$
$$c_4 = -0.043013848896040, \quad c_5 = \ 0.012531422825567, \quad c_6 = \ 0.000286576168915,$$
$$c_7 = -0.000130718954247$$

La Figura 4.4 ilustra la gráfica del polinomio de ajuste junto con los datos. El nuevo polinomio no interpola los datos, pero captura el comportamiento global de mejor manera que cualquiera de los polinomios de interpolación encontrados anteriormente.

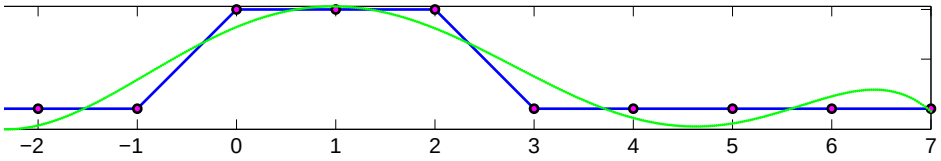


Figura 4.4: Polinomio de interpolación de grado 6.

Observación. Si hay miles de observaciones y sólo unos pocos parámetros, la matriz de diseño A es muy grande, pero la matriz A^TA es pequeña y de orden del número de parámetros, $n \times n$.

Precaución. El método vía ecuaciones normales para resolver problemas de mínimos cuadrados aparece en muchos libros de estadística y métodos numéricos. *Este método debe utilizarse con cautela* pues el sistema de ecuaciones normales es más mal condicionado que el sistema sobredeterminado original, es decir

$$\kappa(A^TA) = [\kappa(A)]^2 \, .$$

Con aritmética de precisión finita, las ecuaciones normales pueden llegar a ser singulares y, en consecuencia $(A^TA)^{-1}$ no existir aún y cuando las columnas de A sean linealmente independientes. Por ejemplo, consideremos la siguiente matriz

$$A = \begin{bmatrix} 1 & 1 \\ \epsilon & 0 \\ 0 & \epsilon \end{bmatrix} \qquad \text{con} \qquad 0 < \epsilon \ll 1.$$

Las dos columnas son casi paralelas, pero linealmente independientes. Con aritmética exacta obtenemos

$$A^TA = \begin{bmatrix} 1+\epsilon^2 & 1 \\ 1 & 1+\epsilon^2 \end{bmatrix}$$

Sin embargo, si $\epsilon < 10^{-8}$ y utilizamos aritmética de punto flotante de doble precisión obtenemos

$$A^T A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

la cual es una matriz singular.

4.4. Método de factorización QR por ortogonalización de Gram-Schmidt

El otro método principal para resolver el problema de mínimos cuadrados es el método de factorización QR. Este es un método clásico, moderno, y popular desde 1960 (Golub). De hecho, actualmente se considera que este método representa una de las ideas algoritmicas más importante en el álgebra lineal numérica.

4.4.1. Factorización reducida

Considere la matriz $A \in \mathbb{R}^{m\times n}$ con $m \geq n$ y sean $a_i \in \mathbb{R}^m$, $i = 1, \ldots, n$ los vectores columna de A:

$$A = [\quad a_1 \quad | \quad a_2 \quad | \quad \cdots \quad | \quad a_n \quad].$$

Los espacios sucesivos generados por los vectores columna de A tienen la siguiente propiedad

$$gen\{a_1\} \subseteq gen\{a_1, a_2\} \subseteq \ldots \subseteq gen\{a_1, \ldots, a_n\}.$$

La idea detrás de la factorización QR es construir una sucesión de vectores ortonormales $q_1, q_2, \ldots \in \mathbb{R}^m$ que generen estos espacios sucesivos. Supongamos que A es de rango completo, entonces sus vectores columna son linealmente independientes y queremos que

$$gen\{q_1, \ldots, q_i\} = gen\{a_1, \ldots, a_i\}, \quad i = 1, 2, \ldots, n$$

con $\|q_i\|_2 = 1$ y $q_i^T q_j = \delta_{ij}$. Para contruir este conjunto de vectores podemos utilizar el método de Gram-Schmidt:

$$\begin{aligned}
q_1 &= \frac{v_1}{\|v_1\|_2} \quad \text{con} \quad v_1 = a_1, \\
q_2 &= \frac{v_2}{\|v_2\|_2} \quad \text{con} \quad v_2 = a_2 - (q_1^T a_2) q_1, \\
q_3 &= \frac{v_3}{\|v_3\|_2} \quad \text{con} \quad v_3 = a_3 - (q_1^T a_3) q_1 - (q_2^T a_3) q_2.
\end{aligned}$$

En general, en el j-ésimo paso, suponiendo conocidos q_1, q_2,..., q_{j-1}, un vector q_j ortonormal a ellos esta dado por

$$q_j = \frac{v_j}{\|v_j\|_2} \quad \text{con} \quad v_j = a_j - (q_1^T a_j)q_1 - (q_2^T a_j)q_2 - \ldots - (q_{j-1}^T a_j)q_{j-1} = a_j - \sum_{i=1}^{j-1}(q_i^T a_j)q_i \,.$$

Si definimos $r_{ij} \equiv q_i^T a_j$, y el escalar $r_{jj} = \|v_j\|_2$, entonces

$$\begin{aligned} q_1 &= \frac{a_1}{r_{11}}\,, \\ q_2 &= \frac{a_2 - r_{12}\, q_1}{r_{22}}\,, \\ &\vdots \\ q_n &= \frac{a_n - \sum_{i=1}^{n-1} r_{in}\, q_i}{r_{nn}}\,. \end{aligned}$$

Por lo tanto,

$$\begin{aligned} a_1 &= r_{11}\, q_1\,, \\ a_2 &= r_{12}\, q_1 + r_{22}\, q_2\,, \\ &\vdots \\ a_n &= r_{1n}\, q_1 + r_{2n}\, q_2 + \cdots + r_{nn}\, q_n\,. \end{aligned}$$

Este conjunto de ecuaciones tienen la siguiente representación matricial

$$\begin{aligned} A &= \underbrace{\left[\begin{array}{c|c|c|c} a_1 & a_2 & \cdots & a_n \end{array}\right]}_{\text{matriz } m\times n} \\ &= \underbrace{\left[\begin{array}{c|c|c|c} q_1 & q_2 & \cdots & q_n \end{array}\right]}_{\text{matriz } m\times n} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ 0 & r_{22} & \ldots & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & r_{nn} \end{bmatrix}}_{\text{matriz } n\times n} \\ &= \qquad \hat{Q} \qquad\qquad \hat{R} \end{aligned}$$

El siguiente algoritmo construye la factorización $\hat{Q}\hat{R}$ encontrada:

Algoritmo de Gram-Schmidt clásico

Para $j = 1, \ldots, n$

· $v_j = a_j$

· Para $i = 1, \ldots, j-1$

· · $r_{ij} = q_i^T a_j$

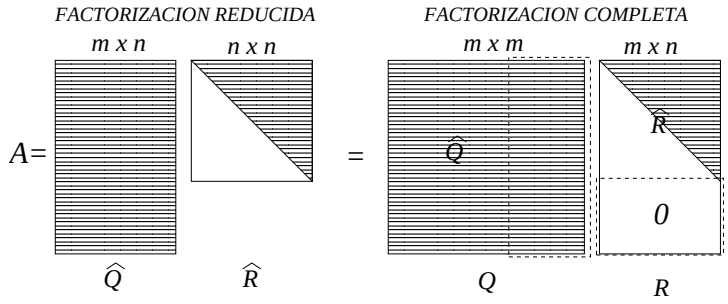
- $\cdot \quad \cdot \quad v_j = v_j - r_{ij}\, q_i$
- Fín
- $r_{jj} = \|v_j\|_2$
- $q_j = v_j / r_{jj}$

Fín

Teorema 4.5. *Si $A \in \mathbb{R}^{m\times n}$ con $m \geq n$ es de rango completo, existe una única factorización reducida QR, $A = \hat{Q}\hat{R}$ con $r_{ii} > 0$, $i = 1, \ldots, n$. (Ver Trefethen–Bau)*

4.4.2. Factorización completa

Una factorización completa QR de $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ va más allá agregando $m - n$ columnas ortonormales a $\hat{Q}$, y agregando $m - n$ renglones de ceros a $\hat{R}$ de manera tal que obtenemos una matriz ortogonal $Q \in \mathbb{R}^{m\times m}$ y otra matriz $R \in \mathbb{R}^{m\times n}$ triangular superior. Esquemáticamente



En la factorización completa las columnas q_j para $j = n+1, \ldots, m$, son ortogonales a $Im(A)$ (el espacio imagen de A). Observese que la matriz completa Q tiene la propiedad

$$Q^T Q = \begin{bmatrix} q_1^T \\ q_2^T \\ \vdots \\ q_m^T \end{bmatrix} [\, q_1 \mid q_2 \mid \cdots \mid q_m \,] = I$$

debido a que $q_i^T q_j = \delta_{ij}$. Por tanto $Q^{-1} = Q^T$. A las martrices con esta propiedad se les denomina ***matrices ortogonales***.

Teorema 4.6. *Cualqier matriz $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ tiene una factorización completa QR, $A = QR$ con $Q \in \mathbb{R}^{m\times m}$ matriz ortogonal y $R \in \mathbb{R}^{m\times n}$ matriz triangular superior (ver Trefethen–Bau).*

Habiendo obtenido una factorización completa QR de $A \in \mathbb{R}^{m\times n}$, el problema sobredeterminado $Ax = b$ con $b \in \mathbb{R}^m$ se puede expresar en la forma $QRx = b$, que a su vez es equivalente al sistema triangular superior

$$Rx = Q^T b \quad (\text{pues } Q^{-1} = Q^T)\,.$$

Este sistema en forma expandida es

$$\begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ 0 & r_{22} & \dots & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & r_{nn} \\ 0 & 0 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \\ f_{n+1} \\ \vdots \\ f_m \end{bmatrix}$$

con $f_i = (Q^T b)_i,\ i = 1, \dots, m$. Esta claro que este sistema se puede resolver utilizando sustitución regresiva, y que las últimas $m - n$ componentes de $Q^T b$ no intervienen en la solución. De hecho estas últimas componentes de $Q^T b$ estan relacionadas con el residual $r = b - Ax$, pues $Q^T r = Q^T b - Rx = (0, \dots, 0, f_{n+1}, \dots, f_m)^T$ y, en consecuencia, $||r||_2 = ||z||_2$.

Observese que en el caso que A sea una matriz cuadrada ($m = n$) no singular este algoritmo es útil para resolver sistemas lineales $Ax = b$. Sin embargo no es el método estandar porque requiere el doble de operaciones que el método de eliminación de Gauss o el método de factorización LU.

En la práctica las fórmulas de Gram-Schmidt no se aplican como se muestra en la pagina 82 debido a que la sucesión de operaciones resulta numéricamente inestable (sensible a errores de redondeo). Esta inestabilidad se produce debido a las sustracciones y a que los vectores q_j no son estrictamente ortogonales debido a errores de redondeo. Se pueden utilizar métodos de estabilización, cambiando el orden en que se realizan las operaciones. Sin embargo, hay un método más efectivo, estable por supuesto, para encontrar la factorización QR. El nuevo método hace uso de las propiedades de las *proyecciones otogonales*. Por tal motivo hacemos un paréntesis en nuestra discusión para estudiar dichas propiedades.

4.5. Proyecciones en $\mathbb{R}^n$

Una proyección en $\mathbb{R}^n$ es una matriz cuadrada P de $n \times n$ tal que $P^2 = P$. El ***espacio imagen*** de P se define por

$$Im(P) = \{v \in \mathbb{R}^n \mid v = Px, \text{ para algún } x \in \mathbb{R}^n\}.$$

El ***espacio nulo*** de P se define por

$$Nul(P) = \{x \in \mathbb{R}^n \mid Px = \vec{0}\}.$$

Ejemplo 4.7. *Verificar que la siguiente matriz cuadarada es una proyeccón y encontrar su espacio imagen y su espacio nulo.*

$$P = \frac{1}{5}\begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix}$$

Solución. P es una proyección en $\mathbb{R}^3$, pues $P^2 = P$. El espacio imagen de P es el conjunto de vectores $v \in \mathbb{R}^n$ de la forma

$$v = \frac{1}{5}\begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{5}(x_1 + 2x_3)\begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} = c\begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}, \quad c \in \mathbb{R}.$$

Es decir, $Im(P) = \{v \in \mathbb{R}^n \mid v = c[1,0,2]^T\} = gen\{[1,0,2]^T\}$ es la linea recta determinada por el vector $[1,0,2]^T$. El espacio nulo de esta misma proyeción es el conjunto de vectores $x = [x_1, x_2, x_3]^T$ que satisfacen

$$\frac{1}{5}\begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{matrix} x_1 + 2x_3 = 0 \\ 2x_1 + 4x_3 = 0 \end{matrix}$$

$$\Rightarrow \quad x_1 + 2x_3 = 0\,.$$

Es decir, $Nul(P) = \{[x_1, x_2, x_3]^T | x_1 + 2x_3 = 0\}$, el cual representa el plano con ecuación $x_1 + 2x_3 = 0$ y normal al vector $(1,0,2)^T$. La Figura 4.5 ilustra geométricamente el espacio imagen de P, la cual es una linea recta en el plano $x_1 - x_3$ de $\mathbb{R}^3$. El espacio nulo de P es el plano perpendicular a dicha recta. En dicha figura sólo se ilustra la intersección de este plano con el plano $x_1 - x_3$.

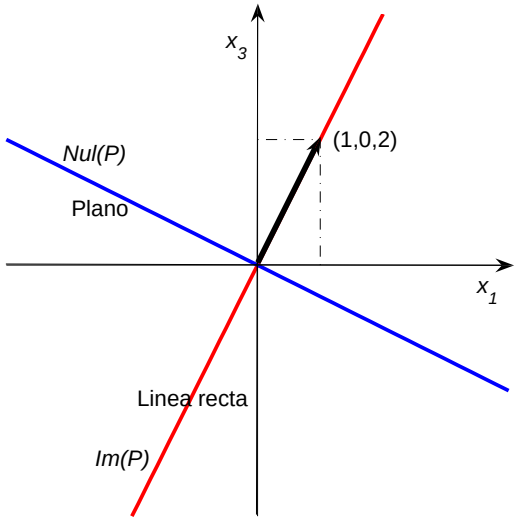


Figura 4.5: Ilustración del espacio imagen y el espacio nulo

4.5.1. Algunas propiedades de las proyecciones

1. Si $v \in Im(P)$, entonces $Pv = v$.
   Demostración. $v \in Im(P) \Rightarrow v = Px$ para algún $x \in \mathbb{R}^n \Rightarrow Pv = P^2v = Px = v$.

2. Dado $v \in \mathbb{R}^n$, v se puede escribir como $v = v_1 + v_2$ con $v_1 \in Im(P)$ y $v_2 \in Nul(P)$.
   Demostración. Sea $v_2 = v - Pv$, entonces $Pv_2 = P(v - Pv) = Pv - P^2v = \vec{0} \Rightarrow$ $v_2 \in Nul(P)$ y por lo tanto $v = v_1 + v_2$ con $v_1 = Pv \in Im(P)$.

3. Si P es una proyección en $\mathbb{R}^n$, entonces $I - P$ tambien es una proyección.
   Demostración. $(I - P)^2 = I - 2IP + P^2 = I - 2P + P = I - P$.

   A la proyección $I - P$ se le llama ***proyección complementaria*** de P.

4. Si P es una proyección en $\mathbb{R}^n$, entonces

   - $Im(P) = Nul(I - P)$: P proyecta sobre el espacio nulo de $I - P$.
   - $Im(I - P) = Nul(P)$: $I - P$ proyecta sobre el espacio nulo de P.

   Demostración. $v \in Im(P) \Rightarrow v = Px$, con $x \in \mathbb{R}^n \Rightarrow (I - P)v = v - Pv = Px - P^2x = \vec{0} \Rightarrow v \in Nul(I-P) \Rightarrow Im(P) \subseteq Nul(I-P)$. Por otro lado $v \in Nul(I-P)$ $\Rightarrow v - Pv = \vec{0} \Rightarrow v = Pv \in Im(P) \Rightarrow Nul(I - P) \subseteq Im(P)$. Con esto se concluye que $Im(P) = Nul(I - P)$. En forma análoga se verifica que $Im(I - P) = Nul(P)$.

Ejemplo 4.8. *Retomando la proyección del ejemplo anterior verificar las propiedades 2, 3, y 4 dado el vector arbitrario* $v = (a_1, a_2, a_3)^T \in \mathbb{R}^3$.

Solución.

$$P = \frac{1}{5}\begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix},$$

$$v_1 = Pv = \frac{1}{5}\begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} a_1 + 2a_3 \\ 0 \\ 2a_1 + 4a_3 \end{bmatrix} \in Im(P),$$

$$v_2 = v - Pv = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} - \frac{1}{5}\begin{bmatrix} a_1 + 2a_3 \\ 0 \\ 2a_1 + 4a_3 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} 4a_1 - 2a_3 \\ 5a_2 \\ -2a_1 + a_3 \end{bmatrix} \in Nul(P),$$

Esta claro que $v_1 + v_2 = v$, con lo cual 2 queda verificada.

La proyección complementaria de P es

$$I - P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{5}\begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} 4 & 0 & -2 \\ 0 & 5 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

y

$$(I - P)^2 = \frac{1}{25}\begin{bmatrix} 20 & 0 & -10 \\ 0 & 25 & 0 \\ -10 & 0 & 5 \end{bmatrix} = I - P.$$

Ademas la imagen de $v \in \mathbb{R}^3$ bajo $I - P$ está en $Nul(P)$, y más aún que

$$Im(I - P) = Nul(P)$$

La Figura 4.6 ilustra estas propiedades. En dicha figura se tiene que $Im(I - P) = Nul(P)$ es el plano en $\mathbb{R}^3$ con normal $(1, 0, 2)^T$, y $Nul(I - P) = Im(P)$ es la linea determinada por el vector $\vec{a} = (1, 0, 2)^T$.

De la discusión anterior concluimos que

5. Una proyección P en $\mathbb{R}^n$ separa el espacio completo en dos subespacios S_1 y S_2 con $S_1 \cap S_2 = \{\vec{0}\}$, y $S_1 + S_2 = \mathbb{R}^n$. Es decir, dado $v \in \mathbb{R}^n$, $v = v_1 + v_2$ con $v_1 = Pv$ y $v_2 = (I - P)v$, ó bién $v_1 = (I - P)v$ y $v_2 = Pv$.

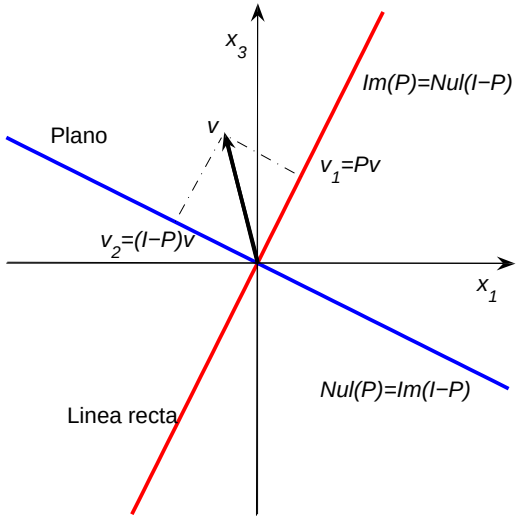


Figura 4.6: Ilustración de la proyecciòn complementaria.

4.5.2. Proyecciones ortogonales

Los tipos de proyecciones más importantes en el álgebra lineal numérica y en las aplicaciones son las denominadas ***proyecciones ortogonales***. Se dice que una *proyección* P *es ortogonal* si $P^T = P$. De hecho, una proyección ortogonal separa el espacio completo $\mathbb{R}^n$ en dos subespacios ortogonales $S_1 \perp S_2$ con $S_1 \cap S_2 = \{\vec{0}\}$, $S_1 + S_2 = \mathbb{R}^n$. Dado $v \in \mathbb{R}^n$, $v = v_1 + v_2$ con $v_1 = Pv$ y $v_2 = (I - P)v$, y si P es ortogonal, entonces

$$v_1^T v_2 = (Pv)^T (I - P)v = v^T P^T (I - P)v = v^T P(I - P)v = v^T (P - P^2)v = \vec{0}$$

En resumen

6. Si P en una proyección ortogonal, entonces Pv y $(I - P)v$ son ortogonales para toda $v \in \mathbb{R}^n$.

Ejemplo 4.9. *La proyección anterior*

$$P = \frac{1}{5}\begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix},$$

es una proyección ortogonal, pues P *es simetrica. La Figura 4.6 muestra la separación de* $\mathbb{R}^3$ *en dos subespacios ortogonales a saber* $Im(P) = Nul(I - P)$ *y* $Nul(P) = Im(I - P)$

Ejemplo 4.10. ***Proyección ortogonal de rango 1****. Dado $q \in \mathbb{R}^n$ con $\|q\|_2 = 1$, la matriz de rangp 1 dada por $P_q \equiv qq^T$ es una proyección ortogonal.*

Demostración. Primero, observese que P_q es una matriz de rango 1, pues toda columna es un múltiplo del vector q:

$$P_q = \begin{bmatrix} q_1q_1 & q_1q_2 & \cdots & q_1q_n \\ q_2q_1 & q_2q_2 & \cdots & q_2q_n \\ \vdots & & \ddots & \vdots \\ q_nq_1 & q_nq_2 & \cdots & q_nq_n \end{bmatrix}.$$

P_q es una proyección, pues

$$(P_q)^2 = (qq^T)(qq^T) = q(q^Tq)q^T = q\|q\|_2^2q^T = qq^T = P_q\,.$$

P_q es ortogonal, pues

$$(P_q)^T = (qq^T)^T = (q^T)^Tq^T = qq^T = P_q\,.$$

Su proyección complementaria es

$$(P_q)^T = I - P_q = I - qq^T\,.$$

Ejemplo 4.11. *Para cualquier $a \in \mathbb{R}^n$, $a \neq \vec{0}$, existe una proyección ortogonal de rango 1*

$$P_a = \frac{aa^T}{a^Ta}\,.$$

Demostración. Dado $a \in \mathbb{R}^n$, $a \neq \vec{0}$, $q = a/\|a\|_2$ es unitario y

$$qq^T = \frac{a}{\|a\|_2}\cdot\frac{a^T}{\|a\|_2} = \frac{aa^T}{\|a\|_2^2} = \frac{aa^T}{a^Ta}\,.$$

P_av produce la proyección del vector $v \in \mathbb{R}^n$ sobre la linea definida por el vector $a \in \mathbb{R}^n$. Su proyección complementaria

$$P_a^\perp = I - P_a = I - \frac{aa^T}{a^Ta},$$

es de rango $n-1$ y proyecta el vector $v \in \mathbb{R}^n$ sobre el hiperplano perpendicular al vector a. En otras palabras

P_a proyecta v sobre la linea determinada por a:

$$Im(P_a) = \{x \in \mathbb{R}^n \mid x = \alpha\, a,\ \alpha \in \mathbb{R}\}\,.$$

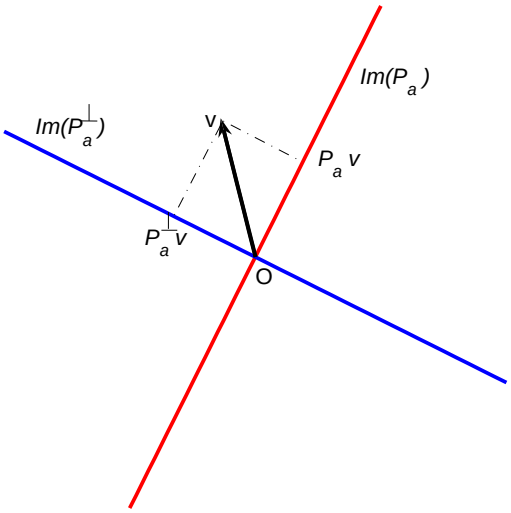


Figura 4.7: Proyeccíon ortogonal de rango 1 generada por un vector a.

$P_a^{\perp}$ proyecta sobre el hiperplano con vector normal $a = (a_1, a_2, \ldots, a_n)^T$:

$$Im(P_a^{\perp}) = \{x \in \mathbb{R}^n \mid a_1x_1 + a_2x_2 + \ldots + a_nx_n = 0\}\,.$$

En la Figura 4.7 se ilusta lo anterior.

La proyección

$$P = \frac{1}{5}\begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix}$$

es un caso particular de una proyección de rango 1 con $a = (1, 0, 2)^T$, pues

$$\frac{aa^T}{a^Ta} = \frac{1}{5}\begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}\begin{bmatrix} 1 & 0 & 2 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix}.$$

Ejemplo 4.12. ***Proyección ortogonal sobre la imagen de una matriz*** $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$. *Si* $A \in \mathbb{R}^{m\times n}$ *con* $m \geq n$ *es de rango completo, sabemos que* A^TA *es una matriz de* $n \times n$*, simétrica y definida positiva. La inversa generalizada derecha de* A*, se define como la matriz* $A^+ = (A^TA)^{-1}A^T$ *de orden* $n \times m$*. Esta matriz define una proyección ortogonal en el espacio* $\mathbb{R}^m$ *al premultiplicarse por* A*, pues* $P_A = AA^+ = A(A^TA)^{-1}A^T \in \mathbb{R}^{m\times m}$*, y, claramente* $P_A^2 = P_A$ *y* $P_A^T = P_A$*. Cualquier vector* $v \in \mathbb{R}^m$ *es proyectado sobre el espacio imagen de la matriz* A *por la proyección* P_A*. Es decir, el espacio generado por las columnas de* A *es igual a* $Im(P_A)$ *(ver Fig. 4.8).*

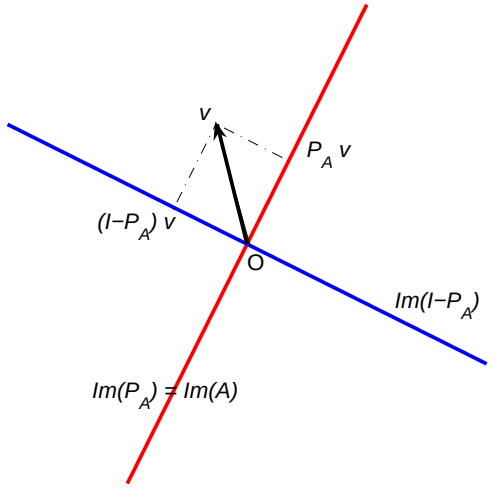


Figura 4.8: Proyección otogonal sobre la imagen de A.

Nota. Observese que $P_A = A(A^TA)^{-1}A^T$ es la generalización multidimensional de la proyección de rango 1, $Pa = aa^T/a^Ta$ con $a \in \mathbb{R}^m$, si observamos que a se puede ver como una de las columnas de la matriz A ó bién como una matriz de $m \times 1$.

4.6. Método de factorización QR por reflexiones de Householder

4.6.1. Triangularización de Householder

La *triangularización de Housholder* (Alston Householder, 1958) es el método más usado actualmente para encontrar la factorización QR debido a que es numéricamente más estable que el método de Gram-Schmidt. Este método consiste en un proceso de "***triangularización ortogonal***", en donde se construye una matriz triangular por una sucesión de operaciones matriciales semejante al proceso de eliminación de Gauss. Solo que en este caso se multiplica por matrices ortogonales Q_k, de tal manera que al final del proceso

$$Q_n \cdots Q_2 Q_1 A$$

resulta triangular superior. Cada matriz Q_k se escoge para introducir ceros debajo de la diagonal en la k-ésima columna. Por ejemplo, para una matriz A de 5×3, las operaciones

Q_k se aplican como se muetra a continuación:

$$\overbrace{\underbrace{\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}}_{A}}^{A^{(0)}} \overrightarrow{Q_1} \overbrace{\underbrace{\begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \end{bmatrix}}_{Q_1A}}^{A^{(1)}} \overrightarrow{Q_2} \overbrace{\underbrace{\begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & x \\ 0 & 0 & x \end{bmatrix}}_{Q_2Q_1A}}^{A^{(2)}} \overrightarrow{Q_3} \overbrace{\underbrace{\begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{Q_3Q_2Q_1A}}^{A^{(3)}} = R\,,$$

donde las x indican coeficientes no cero en general.

¿Cómo se construyen tales matrices ortogonales Q_k?

4.6.2. Reflexiones de Householder

Si $A \in \mathbb{R}^{m\times n}$ con $m \geq n$, cada Q_k se escoge como una matriz ortogonal de la forma

$$Q_k = \begin{bmatrix} I & 0 \\ 0 & H \end{bmatrix},$$

donde I es la matriz identidad de orden $(k-1)\times(k-1)$ y H es una matriz ortogonal de orden $(m-k+1)\times(m-k+1)$. La multiplicación por H debe introducir ceros debajo de la diagonal en la k–ésima columna. Esquemáticamente, esta operación se muestra a continuación

$$\begin{bmatrix} I & 0 \\ 0 & H \end{bmatrix}_{m\times m} \begin{bmatrix} a_{11} & \dots & a_{1k-1} & a_{1k} & \dots & a_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & a_{k-1k-1} & a_{k-1k} & \dots & a_{k-1n} \\ 0 & \dots & 0 & a_{kk} & \dots & a_{kn} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & a_{mk} & \dots & a_{mn} \end{bmatrix}_{m\times n}$$

La matriz de la izquierda es la matriz ortogonal Q_k de orden $m \times m$, mientras que la de la derecha es la matriz $A^{(k-1)}$ de orden $m \times n$. El resultado de la anterior multiplicación es la matriz $A^{(k)} \in \mathbb{R}^{m\times n}$, la cual tiene los mismos bloques que $A^{(k-1)}$, excepto el bloque

diagonal inferior que debe cambiar por

$$H \cdot \begin{bmatrix} a_{kk} & \dots & a_{kn} \\ \vdots & \ddots & \vdots \\ a_{mk} & \dots & a_{mn} \end{bmatrix} = \begin{bmatrix} a'_{kk} & a'_{kk+1} & \dots & a'_{kn} \\ 0 & a'_{k+1k+1} & \dots & a'_{k+1n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a'_{mk+1} & \dots & a'_{mn} \end{bmatrix}$$

Por supuesto, aquí el paso fundamental es la multiplicación de H por la primera columna de la submatriz de la derecha:

$$H \begin{bmatrix} a_{kk} \\ a_{k+1k} \\ \vdots \\ a_{mk} \end{bmatrix} = \begin{bmatrix} a'_{kk} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

donde $a'_{kk} = \|x_k\|$, con $x_k = (a_{kk}, \dots, a_{mk})^T$. Es decir, el algoritmo de Householder escoge H como una matriz particular llamada ***reflexión de Householder***. Esta matriz introduce los ceros correctos en la k–ésima columna:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-k+1} \end{bmatrix} \quad \longrightarrow \quad Hx = \begin{bmatrix} \|x\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|x\| e_1$$

con $e_1 = (1, 0, \dots, 0)^T$ de tamaño $m - k + 1$. Para obtener la reflexión de Householder H observamos en la Figura 4.9 que ésta refleja $\mathbb{R}^{m-k+1}$ a través del hiperplano H^+ ortogonal a $v = \|x\|e_1 - x$. Cuando la reflexión se aplica a cada punto en algún lado de H^+, el resultado es la imagen reflejada en el otro lado de H^+. En particular x es enviado a $\|x\|e_1$.

Para cualquier $y \in \mathbb{R}^{m-k+1}$, el vector

$$P\, y = \left(I - \frac{vv^T}{v^T v} \right) y$$

es la proyección ortogonal de y sobre el espacio H^+ (perpendicular a v). Para reflejar y a través de H^+, debemos continuar al doble de la distancia en la misma dirección como se ilustra en la Figura 4.10. La reflexión H debe ser entonces

$$H = I - 2\frac{vv^T}{v^T v} \qquad \text{con} \qquad v = \|x\|e_1 - x.$$

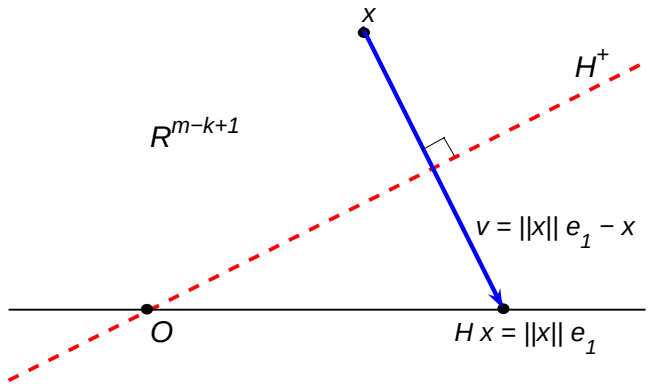


Figura 4.9: Reflexión de Householder para un vector dado x

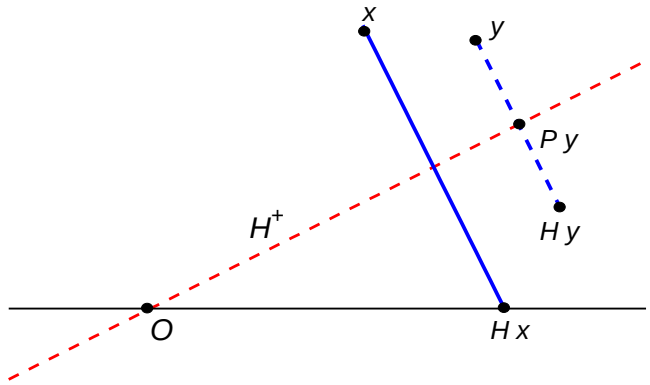


Figura 4.10: La reflexión de Householder $H = I - 2vv^T/v^Tv$.

Ejemplo 4.13. *Dado el vector* $x = (3, 4, 0)^T$*, encontrar la reflexión de Householder* H *tal que* $Hx = \|x\|e_1 = (5, 0, 0)^T$.

Solución.

$$v = \|x\|e_1 - x = \begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ -4 \\ 0 \end{bmatrix},$$

$$vv^T = \begin{bmatrix} 2 \\ -4 \\ 0 \end{bmatrix} \begin{bmatrix} 2 & -4 & 0 \end{bmatrix} = \begin{bmatrix} 4 & -8 & 0 \\ -8 & 16 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{y} \quad v^Tv = 20.$$

Luego

$$H = I - 2\frac{vv^T}{v^T v} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{10}\begin{bmatrix} 4 & -8 & 0 \\ -8 & 16 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 3/5 & 4/5 & 0 \\ 4/5 & -3/5 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Se puede verificar que H efectivamente satisface $Hx = \|x\|e_1 = (5,\, 0,\, 0)^T$. En la figura 4.11 se ilustra el resultado de este ejemplo.

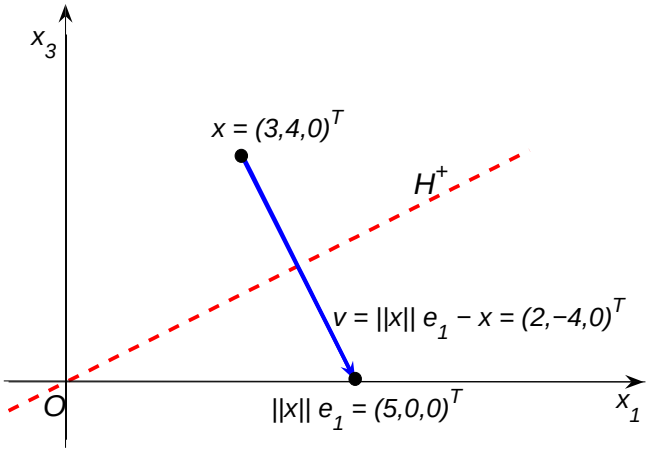


Figura 4.11: Reflexión de Householder del ejemplo 4.13.

El plano H^+ tiene ecuación $v_1x_1 + v_2x_2 + v_3x_3 = 0$, es decir, $2x_1 - 4x_2 = 0$ con x_3 arbitrario. Observese además que

$$HH^T = \begin{bmatrix} 3/5 & 4/5 & 0 \\ 4/5 & -3/5 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 3/5 & 4/5 & 0 \\ 4/5 & -3/5 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I$$

En general, se puede verificar facilmente que

$$HH^T = \left[I - 2\frac{vv^T}{v^T v}\right]\left[I - 2\frac{vv^T}{v^T v}\right]^T = I\,.$$

de modo que H es siempre una matriz ortogonal. Por lo tanto, las matrices Q_k en el proceso de triangularización de Householder son ortogonales, pues

$$Q_k Q_k^T = \begin{bmatrix} I & 0 \\ 0 & H \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & H^T \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & HH^T \end{bmatrix} = \begin{bmatrix} I_{k-1} & 0 \\ 0 & I_{m-k+1} \end{bmatrix} = I_m\,.$$

4.6.3. La mejor de las dos reflexiones

En la exposición anterior se ha simplificado el procedimiento. El vector x en realidad puede reflejarse ya sea a $\|x\|e_1$ ó bién a $-\|x\|e_1$, dando lugar a dos reflexiones, una a través del hiperplano H^+, y otra a través del hiperplano H^-, como se muestra en la Figura 4.12.

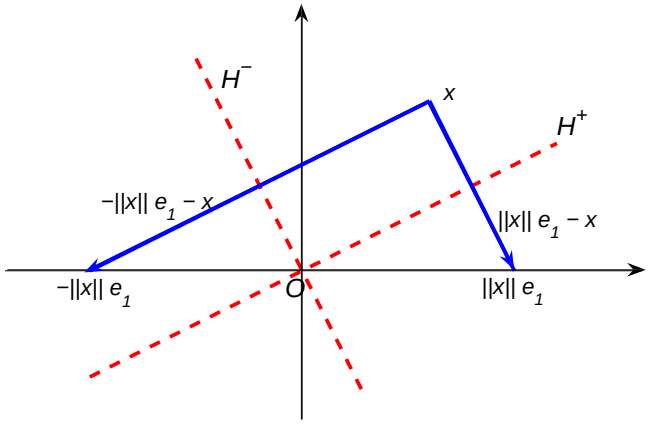


Figura 4.12: Las dos reflexiones de Householder

Desde el punto de vista matemático, cualquiera de las dos elecciones es satisfactoria. Sin embargo, para asegurar estabilidad en el algoritmo numérico se debe escoger aquella reflexión para la cual x se encuentre más alejado de su punto reflejado $\|x\|e_1$ ó $-\|x\|e_1$. Para lograr esto escogemos $v = -signo(x_1)\|x\|e_1 - x$, o bien podemos quitar el signo "$-$", y escoger $v = signo(x_1)\|x\|e_1 + x$. La función *signo* se define como:

$$signo(x_1) = \begin{cases} 1 \text{ si } x_1 \geq 0\,, \\ -1 \text{ si } x_1 < 0\,. \end{cases}$$

Vale la pena recordar que x_1 es la 1^{ra} coordenada de x. Las Figura 4.13 muestra ambos casos. La razón por la cual esta elección para v es conveniente es que si el ángulo entre H^+ y e_1 es muy pequeño, entonces el vector $v = \|x\|e_1 - x$ será más pequeño que x ó $\|x\|e_1$, y el cálculo de v representa la sustracción de cantidades casi iguales, con lo cual se obtienen errores de cancelación. Por otro lado, si escogemos el signo de tal forma que en lugar de restar sumemos, cortamos el efecto de cancelación y $\|v\|$ nunca será más pequeño que $\|x\|$, con lo cual aseguramos estabilidad en los cálculos.

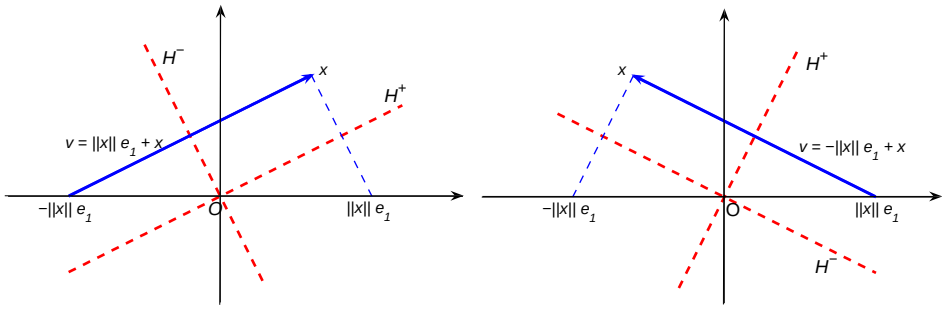


Figura 4.13: Gráfica de la izquierda: $x_1 > 0 \Rightarrow signo(x_1) = +1$. Gráfica de la derecha: $x_1 < 0 \Rightarrow signo(x_1) = -1$.

4.6.4. Algoritmo de factorización de Householder

Utilizando las ideas anteriores podemos construir el algoritmo que factoriza A en la forma $A = QR$, es decir $Q^T A = R$, con $Q = (Q_n \cdots Q_2 Q_1)^T$ matriz ortogonal y R matriz triangular superior. Sin embargo, en el siguiente algoritmo solamente calculamos el factor R y lo almacenamos en el mismo espacio de memoria que ocupa A. También se almacenan los n vectores de reflexión $v_1, \ldots, v_n$, para posibles usos posteriores.

Para $k = 1, \ldots, n$
. $x = A(k : m,\, k)$
. $v_k = signo(x_1)\|x\|e_1 + x$
. $v_k = v_k / \|v_k\|_2$
. $A(k : m,\, k : n) = A(k : m,\, k : n) - 2v_k\,(v_k^T\, A(k : m,\, k : m))$
Fín

Observese que en el algoritmo se han realizado dos simplificaciones:

1. Se ha denotado por v a $v/\|v\|_2$ pues la reflexión se puede escribir como
$$I - 2\left(\frac{vv^T}{v^T v}\right) = I - 2\left(\frac{v}{\|v\|_2}\frac{v^T}{\|v\|_2}\right).$$
2. Al multiplicar o aplicar la reflexión por un vector y obtenemos
$$\left(I - 2\frac{v}{\|v\|_2}\frac{v^T}{\|v\|_2}\right) y = y - 2\frac{v}{\|v\|_2}\left(\frac{v^T}{\|v\|_2}\, y\right),$$

de tal manera que cuando y es una de las columnas de la submatriz $A(k : m, k : n)$, entonces la forma económica de aplicar la reflexión a cada una de estas columnas es como se indica en el último renglón del algoritmo.

La matriz ortogonal $Q^T = Q_n \cdots Q_2 Q_1$ no se construye en el algoritmo anterior, pues esto tomaría trabajo adicional. Afortunadamente en muchas aplicaciones no es necesario obtener esta matriz. Por ejemplo, si se quiere resolver el sistema $Ax = b$ y sabemos que $Q^T A = R$, entonces $Rx = Q^T A\, x = Q^T b$ y basta con resolver el sistema $Rx = Q^T b$ utilizando sustitución regresiva. El lado derecho $Q^T b$ puede calcularse aplicando a b la misma sucesión de operaciones aplicadas a la matriz A:

Para $k = 1, \ldots, n$

$\vdots \quad b(k:m) = b(k:m) - 2v_k(v_k^T\, b(k:m))$

Fín

dado que los vectores de reflexión $v_1, \ldots, v_n$ se almacenan al encontrar R en el algoritmo anterior. En el problema de la solución del problema sobredetermnado $Ax = b$ inclusive no es necesario almacenar los vectores de reflexión v_k y el cálculo de los $Q^T b$ puede realizarse simultaneamente al cálculo de R. En este caso el algoritmo completo, incluyendo sustitución regresiva, sería:

Algortimo de triangularización de Housholder para resolver $Ax = b$

Para $k = 1, \ldots, n$

. $x = A(k:m,\, k)$

. $v = signo(x_1)\|x\|e_1 + x$

. $v = v/\|v\|_2$

. $A(k:m,\, k:n) = A(k:m,\, k:n) - 2v\,(v^T\, A(k:m,\, k:m))$

. $b(k:m) = b(k:m) - 2v(v^T\, b(k:m))$

Fín

$x(n) = b(n)/A(n,n)$

Para $k = n-1:-1:1$

$\vdots \quad x(k) = (b(k) - A(k, k+1:n) \cdot b(k+1:n))/A(k,k)$

Fín

Ejemplo 4.14. *Aplicar el algorimto de triangularización de Housholder para ajustar un polinomio de grado 6 a los datos del ejemplo 4.4.*

Solución. Primero construimos la matriz de diseño A como en el Ejemplo 4.4 y tomamos $b = y$. Aplicando el algoritmo de triangularización de Householder, en aritmética de punto flotante de doble presición se obtienen los siguientes valores para los coeficientes del poli-

nomio de grado seis:

$$c_1 = \ 0.822788975730155, \quad c_2 = \ 0.412287124640065, \quad c_3 = -0.173651218062985,$$
$$c_4 = -0.043013848896201, \quad c_5 = \ 0.012531422825541, \quad c_6 = \ 0.000286576168929,$$
$$c_7 = -0.000130718954248,$$

los cuales coinciden en hasta 12 cifras decimales con el resultado del Ejemplo 4.4, en donde se resolvió el problema por medio del algoritmo de ecuaciones normales. La diferencia de ambas soluciones tiene norma–2 igual a 7.52×10^{-13}. La gráfica del polinomio es indistinguible de aquella obtenida en el Ejemplo 4.4. Otra forma de medir la diferencia entre ambas soluciones es por medio de la norma–2 del residual $Ac - y$. En el caso del método de ecuaciones normales es 0.579791307012593, mientras que con el método QR se obtiene 0.579791307012592, mostrando que la diferencia es muy pequeña, y ambas soluciones se pueden considerar indistinguibles. Sin embargo, en la solución de problemas donde el número de datos, y por tanto el tamaño de los sistemas, es mayor la diferencia entre ambos métodos será mayor y favorable al método de factorización QR. Por ejemplo para el problema de ajustar un polinomio de grado 12 a los 20 datos

$x =$ 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0,
16.0, 17.0, 18.0, 19.0, 20.0

$y =$ 6.8, 3.7, 8.3, 5.1, 7.2, 4.3, 3.0, 1.9, 1.9, 6.8, 3.4, 5.5, 1.6, 6.8, 3.4,
5.5, 1.6, 6.9, 3.7, 8.6,

se obtiene la solución

$c =$ 12.907970210856304, -3.016629810172137, -8.667285831321209, 7.388121747944512,
-2.113195497831346, 0.183521281771445, 0.032056774991461, -0.010264106763814,
0.001240193327025, -0.000083809251109, 0.000003322253342, -0.000000072398408,
0.000000000671482

utilizando el algoritmo de ecuaciones normales, mientras que con el algoritmo QR con tringularización de Householder, se obtiene

$c =$ 354.3644019405224, -890.9842717372895, 904.5095592735649, -496.3782632641325
167.1779443154627, -036.8894657313046, 5.524135579250700, -0.570407247523000
0.040598148384200, -0.001954498782900, 0.000060739089900, -0.000001099075100

0.000000008792500

Como puede observarse, los resultados son muy diferentes. La gráfica de los polinomios de ajuste se muestra en la Figura 4.14. El método que produzca el menor residual proporciona la mejor solución. La norma de los residuales es 7.027 cuado se utiliza el método de ecuaciones normales, y es 5.507 cuando se utiliza el método QR. Por lo tanto el polinomio obtenido con el método QR ajusta mejor los datos dados.

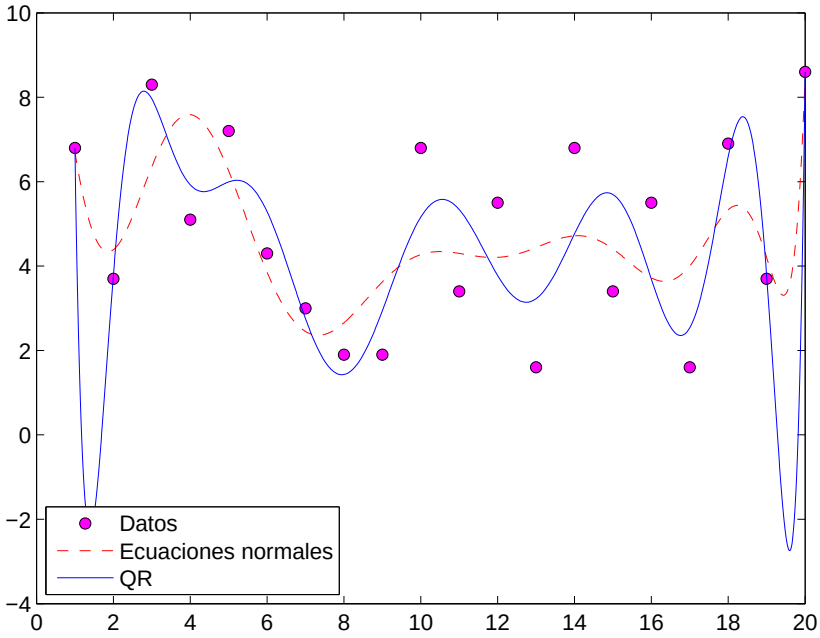


Figura 4.14: Ajuste con el algoritmo de ecuaciones normales y con el algoritmo QR

Capítulo 5

Solución de Ecuaciones no Lineales

El problema de resolver sistemas de ecuaciones no lineales aparece frecuentemente en la matemática computacional y en las aplicaciones en donde los modelos subyacentes para estudiar determinado fenómeno no pueden expresarse en forma lineal. Es decir los sistemas correspondientes no son de la forma $Ax = b$, donde A es una matriz y x, b son vectores. Este nuevo tipo de ecuaciones algebraicas puede representarse en forma funcional compacta $f(x) = 0$, donde $f : \mathbb{R}^n \to \mathbb{R}^n$ es una función vectorial, $x = (x_1, \ldots, x_n)^T$ y $0 = (0, \ldots, 0)^T$en $\mathbb{R}^n$. En forma desarrollada escribimos

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

donde $f_i(x_1, x_2, \ldots, x_n)$, $i = 1, \ldots, n$ son las funciones componentes de la función vectorial f. El caso $n = 1$ corresponde a una simple ecuación no-lineal en la incógnita $x \in \mathbb{R}$.

Frecuentemente, en las aplicaciones, un problema no lineal aparece en la forma de **problema de punto** fijo:

$$\text{Encontrar } x \in \mathbb{R}^n \text{ tal que } x = g(x),$$

donde $g : \mathbb{R}^n \to \mathbb{R}^n$ es una función vectorial. El problema, en este último sistema, consiste en encontrar un punto $\xi \in \mathbb{R}^n$ que satisface $\xi = g(\xi)$, y al cual se le denomina *punto fijo de g*. Cualquier ecuación o sistema de ecuaciones $f(x) = 0$ puede escribirse en la forma de punto fijo, y esto puede realizarse de muchas maneras. Por ejemplo, la ecuación $f(x) = x^2 - x - 2 = 0$ en una variable, puede expresarse de manera equivalente en cualquiera de las formas:

$$
\begin{array}{lll}
1. & x = x^2 - 2 & \Rightarrow \quad g(x) = x^2 - 2 \\
2. & x = \sqrt{x+2} & \Rightarrow \quad g(x) = \sqrt{x+2} \\
3. & x = 1 + \dfrac{2}{x} & \Rightarrow \quad g(x) = 1 + \dfrac{2}{x} \\
4. & x = \dfrac{x^2+2}{2x-1} & \Rightarrow \quad g(x) = \dfrac{x^2+2}{2x-1},
\end{array}
$$

entre otras. Cualquier punto fijo de estas funciones g será una raiz de la ecuación original $f(x) = 0$. Entonces, el problema del cálculo de raices de $f(x)$ se puede expresar como el cálculo de los puntos fijos de alguna función $g(x)$.

5.1. Método iterativo de punto fijo

Los métodos más ampliamente usados para encontrar soluciones de ecuaciones no lineales son los denominados ***métodos iterativos de punto fijo***. Estos métodos consisten en transformar la ecuación (o sistema de ecuaciones) original, $f(x) = 0$, en algún ***sistema equivalente de punto fijo*** $x = g(x)$ adecuado. Una vez hecho esto, se escoge algún x_0 como aproximación inicial al punto fijo ξ, y se genera una sucesión de aproximaciones mediante las iteraciones:

$$x_{n+1} = g(x_n), \quad n = 0,\ 1,\ 2, \ldots$$

Se espera que la sucesión $\{x_n\}$ converja al punto fijo ξ. Desgraciadamente esto no siempre ocurre como se muestra a continuación.

Ejemplo 5.1. *Consideremos el ejemplo sencillo de encontrar numéricamente una de las raices de* $f(x) = x^2 - x - 2 = 0$. *Sus raices exactas son* $x_1 = -1$ *y* $x_2 = 2$. *Nos concentraremos únicamente en el cálculo de la raiz* $\xi = 2$.

a) Con $g(x) = x^2 - 2$ y $x_0 = 2{,}1$, obtenemos $x_1 = 2{,}41$, $x_2 = 3{,}8081$, $x_3 = 12{,}50162561$, y $x_n \to \infty$ si $n \to \infty$.

b) Con $g(x) = \sqrt{x+2}$ y $x_0 = 2{,}1$, obtenemos $x_1 = 2{,}024846$, $x_2 = 2{,}006202$, $x_3 = 2{,}001550$, $x_4 = 2{,}000387, \ldots$, $x_{10} = 2{,}0000004$, y $x_n \to 2$ si $n \to \infty$.

c) Con $g(x) = 1 + \frac{2}{x}$ y $x_0 = 2{,}1$, obtenemos $x_1 = 1{,}952381$, $x_2 = 2{,}024390$, $x_3 = 1{,}987952$, $x_4 = 2{,}006061, \ldots, x_{18} = 2{,}0000004$, y $x_n \to 2$ si $n \to \infty$.

d) Con $g(x) = \frac{x^2+2}{2x-1}$ y $x_0 = 2{,}1$, obtenemos $x_1 = 2{,}003125$, $x_2 = 2{,}00003$, $x_3 = 2{,}000000000004$, y $x_n \to 2$ si $n \to \infty$.

En el primer caso, la función de iteración $g(x) = x^2 - 2$ no genera una sucesión $\{x_n\}$ convergente al punto fijo $\xi = 2$. Por el contrario, esta sucesión diverge a infinito. En los otros tres casos se produce una sucesión convergente al punto fijo $\xi = 2$. Observese que estas sucesiones convergen al punto fijo con mayor o menor "rapidez". Por ejemplo, la *función de iteración*

$$g(x) = \frac{x^2 + 2}{2x - 1}$$

aparentemente genera una sucesión que converge extremadamente rápido al punto fijo, mientras que las otras dos funciones de iteración generan sucesiones con una convergencia más lenta al punto fijo. Hasta el momento no hemos estudiado bajo que condiciones para la función de iteración g se obtiene una sucesión que converja efectivamente al punto fijo ξ. La respuesta a este problema viene dada por el ***teorema de punto fijo de Banach***, también llamado principio de mapeos contractantes, el cual estudiaremos en la siguiente sección. Una vez hecho este estudio podremos estudiar otros temas, como la rapidez de convergencia del método iterativo, y el diseño de algoritmos rápidos.

5.2. Teorema de punto fijo de Banach

Este teorema es muy importante y sirve como fuente para probar existencia y unicidad en diferentes ramas del Análisis y sus aplicaciones como: ecuaciones algebraicas, ecuaciones diferenciales ordinarias, ecuaciones diferenciales parciales, y ecuaciones integrales, entre otras. Este teorema da condiciones suficientes para la existencia y unicidad de un punto fijo para una clase de aplicaciones o funciones llamadas contracciones.

Definición 5.2. *Una aplicación $T : X \to X$ con $X \subseteq \mathbb{R}^n$ se denomina* ***contracción*** *sobre X si existe un número real K, con $0 < K < 1$, tal que para todo $x,\ y \in X$*

$$\|Tx - Ty\| \leq K\|x - y\|$$

en alguna norma vectorial $\|\cdot\|$.

Geométricamente esto significa que dos puntos cualesquiera $x,\ y \in X$ tienen imagenes más cercanas que ellos mismos. De ahí el nombre de *contracción* para la aplicación T.

Ejemplo 5.3. *La función $g : [1, 3] \to [1, 3]$ definida por $g(x) = \sqrt{x + 2}$ es una contracción en $X = [1, 3]$.*

Primero verificamos que $g(x) \in [1,3]$ si $x \in [1,3]$:

$$x \in [1,3] \;\Leftrightarrow\; 1 \leq x \leq 3 \;\Leftrightarrow\; 3 \leq x+2 \leq 5 \;\Leftrightarrow\; \sqrt{3} \leq \sqrt{x+2} \leq \sqrt{5}\,.$$

Por lo tanto, $g(x) = \sqrt{x+2} \in [1,3]$. Por otro lado, por el teorema del valor medio

$$|g(x) - g(y)| = |g'(\eta)|\,|(x-y)| \text{ con } \eta \text{ entre } x \text{ y } y\,,$$

y debido a que

$$g'(x) = \frac{1}{2\sqrt{x+2}} \quad \Rightarrow \quad \max_{x\in[1,3]} |g'(x)| = g'(1) = \frac{1}{2\sqrt{3}}\,.$$

Por lo tanto

$$|g(x) - g(y)| \leq \frac{1}{2\sqrt{3}}|x-y|\,.$$

Se concluye que g es una contracción con $K = \dfrac{1}{2\sqrt{3}} < 1$.

Teorema 5.4. ***Teorema de punto fijo de Banach****. Si X es un conjunto cerrado conexo de $\mathbb{R}^n$ y T es una contracción sobre X, entonces T tiene un* ***único punto fijo*** *en X, y dado cualquier punto de comienzo $x_0 \in X$, la sucesión generada por iteración de punto fijo*

$$x_{n+1} = T\,x_n, \quad n = 0,\ 1,\ 2, \ldots$$

convergerá al punto fijo ξ de T.

Demostración. Sea $x_0 \in X$, y sea $\{x_n\}$ la sucesión generada por iteración. Para cada $m \in \mathbb{N}$, $x_{m+1} - x_m = T\,x_m - T\,x_{m-1}$, y como T es una contracción entonces

$$\|x_{m+1} - x_m\| = \|T\,x_m - T\,x_{m-1}\| \leq K\,\|x_m - x_{m-1}\|\,,$$

con $0 < K < 1$. Procediendo recursivamente, obtenemos

$$\|x_{m+1} - x_m\| \leq K^m\,\|x_1 - x_0\| \quad \forall m \in \mathbb{N}\,.$$

Utilizando este resultado para toda $n > m$ se obtiene

$$\begin{aligned}
\|x_n - x_m\| &= \|x_n - x_{n-1} + x_{n-1} - x_{n-2} + x_{n-2} + \ldots + x_{m+1} - x_m\| \\
&\leq \|x_n - x_{n-1}\| + \|x_{n-1} - x_{n-2}\| + \ldots + \|x_{m+1} - x_m\| \\
&\leq (K^{n-1} + K^{n-2} + \ldots + K^m)\|x_1 - x_0\| \\
&\leq K^m(1 + K + K^2 + \ldots + K^{n-m-1})\|x_1 - x_0\| \\
&\leq K^m \frac{1-K^{n-m}}{1-K}\|x_1 - x_0\|\,,
\end{aligned}$$

y como $0 < K^{n-m} < 1$, entonces

$$\|x_n - x_m\| \leq K^m \frac{1}{1-K} \|x_1 - x_0\| .$$

Por lo tanto

$$\lim_{m,n\to\infty} \|x_n - x_m\| \leq \lim_{m\to\infty} K^m \frac{1}{1-K} \|x_1 - x_0\| = 0 ,$$

lo cual implica que la sucesión $\{x_n\}$ generada por iteración es una sucesión de Cauchy. Como T esta definida sobre el conjunto cerrado X la sucesión de Cauchy $\{x_n\}$ tiene un límite $\xi \in X$:

$$\xi = \lim_{n\to\infty} x_n$$

Este límite $\xi \in X$ es un punto fijo de T pues

$$\|\xi - T\xi\| = \|\xi - x_n + x_n - T\xi\| \leq \|\xi - x_n\| + \|x_n - T\xi\| = \|\xi - x_n\| + \|Tx_{n-1} - T\xi\|$$

implica que

$$\|\xi - T\xi\| \leq \|\xi - x_n\| + K\|x_{n-1} - \xi\| \to 0, \text{ cuando } n \to 0 .$$

Por lo tanto, $\xi = T\xi$. Además este es el único punto fijo de T en X, pues si hubiese otro, digamos η, entonces

$$\|\xi - \eta\| = \|T\xi - T\eta\| \leq K\|\xi - \eta\| < \|\xi - \eta\| ,$$

lo cual no es posible. Por tanto, ξ es el único punto fijo de T en X. □

En suma, para que una aplicación sea una buena función de iteración y produzca una sucesión convergente al punto fijo basta con que ella sea una contracción en un conjunto cerrado X que contenga al punto fijo. En ocasiones es difícil probar en forma directa que una función de iteración es una contracción. Pero si g es una función suave (tiene derivada continua en X), y si además

$$\|g'(x)\| \leq K, \ \forall x \in X \text{ con } 0 < K < 1 ,$$

entonces g es una contracción en X con $K = \max_{x\in X} \|g'(x)\| < 1$. En efecto, para $x,\ y, \in X$, por el teorema del valor medio, se tiene

$$\|g(x) - g(y)\| \leq \|g'(\eta)\| \, \|x - y\| \text{ con } \eta \text{ entre los puntos } x,\, y .$$

Así que

$$\|g(x) - g(y)\| \leq K\|x - y\| \text{ con } K = \max_{x\in X} \|g'(x)\| < 1 .$$

5.3. Cota del error en la iteración de punto fijo

La desigualdad

$$\|x_n - x_m\| \leq \frac{K^m}{1-K}\|x_1 - x_0\|\,,$$

obtenida anteriormente es válida para toda $n > m$. Luego, cuando $n \to \infty$, $x_n \to \xi$, y por tanto

$$\|x_m - \xi\| \leq \frac{K^m}{1-K}\|x_1 - x_0\| \quad \forall\, m \in \mathbb{N}\,.$$

Esta desigualdad, aparte de que proporciona una cota para el error en la m–ésima iteración, sirve para estimar el número de iteraciones necesarias par alcanzar una precisión determinada en el cálculo del punto fijo ξ:

$$n \geq \log\left(\frac{\|x_n - \xi\|}{\|x_1 - x_0\|}(1-K)\right)/\log K$$

Ejemplo 5.5. *Encontrar la raíz de* $f(x) = x^2 - x - 2 = 0$ *que se encuentra en el intervalo* $[1, 3]$.

Solución. Anteriormente se propusieron algunas funciones de iteración. La $1^{\underline{a}}$ función $g(x) = x^2 - 2$ es tal que su derivada es $g'(x) = 2x$, y en el intervalo $[1, 3]$ esta es mayor que 1, de hecho $2 \leq g'(x) \leq 3$. Luego $g(x) = x^2 - 2$ no define una contracción en $[1, 3]$, y esto explica porqué la sucesión generada por iteración de punto fijo diverge, a pesar de que escoge como punto inicial $x_0 = 2{,}1$ muy cercano al punto fijo $\xi = 2$. La Figura 5.1 ilustra esta situación.

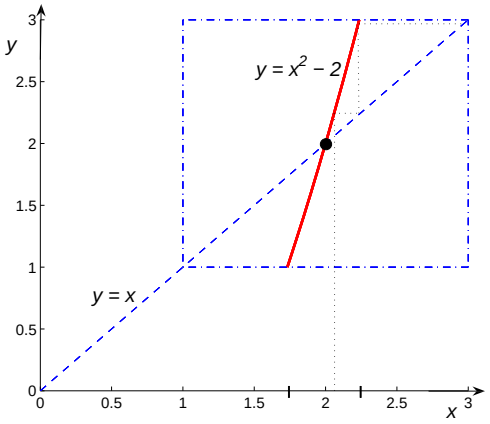


Figura 5.1: La función de iteración $g(x) = x^2 - 2$ es tal $|g'(x)| > 1$ en $[1, 3]$, y produce una sucesión divergente.

La $2^{\underline{a}}$ funciíon de iteración $g(x) = \sqrt{x+2}$ produjo una sucesión convergente al punto fijo $\xi = 2$. En la Sección 3.2 se demostró que esta función es una contracción en $[1, 3]$, así que cualquier punto de comienzo x_0 en este intervalo producirá, bajo las iteraciones, una sucesión que converge al punto fijo $\xi = 2$. ¿Si comenzamos con $x_0 = 1$, cuál es el número de iteraciones para alcanzar un error menor a 10^{-4}? Como

$$|x_n - \xi| = 10^{-4},\ |x_0 - x_1| = |1 - \sqrt{3}| = \sqrt{3} - 1,\ K = \frac{1}{2\sqrt{3}}\,.$$

Entonces

$$n \geq \frac{\log\left(\frac{10^{-4}(1-\frac{1}{2\sqrt{3}})}{\sqrt{3}-1}\right)}{\log(\frac{2}{3\sqrt{3}})} \sim 7{,}436$$

Así que podemos tomar $n = 8$. La Tabla 5.1 muestra las iteraciones con precisión a 6 cifras decimales, junto con el error y la estimación del error. Observese que el error real siempre es

n	x_n	$e_n = \lvert x_n - \xi \rvert$	$\frac{K^n}{1-K}\lvert x_1 - x_0 \rvert$
0	1	1	1.029137
1	1.732051	0.267949	0.297086
2	1.931852	0.068148	0.085761
3	1.982890	0.017110	0.024757
4	1.995718	0.004282	0.007147
5	1.998930	0.001070	0.002063
6	1.999732	0.000268	0.000715
7	1.999933	0.000067	0.000172
8	1.999983	0.000017	0.000050

Cuadro 5.1: Iteraciones de $g(x) = \sqrt{x+2}$, el error y la estimación del error.

menor que la cota $\dfrac{K^n}{1-K}|x_1 - x_0|$, como debe de suceder. Además, en la práctica, el número de iteraciones para alcanzar un error menor que 10^{-4} es 7. La Figura 5.2 ilustra este ejemplo

Finalmente, la $3^{\underline{a}}$ función de iteración $g(x) = 1 + \dfrac{2}{x}$ también produce una sucesión convergente cuando $x_0 = 2{,}1$. Para saber que rango de valores puede tomar x_0 y obtener una sucesión que converja a $\xi = 2$ con esta función de iteración hacemos

$$|g'(x)| = \left|-\frac{2}{x^2}\right| \leq 1 \Leftrightarrow x^2 \geq 2 \Leftrightarrow x \geq \sqrt{2} \text{ ó } x \leq -\sqrt{2}\,.$$

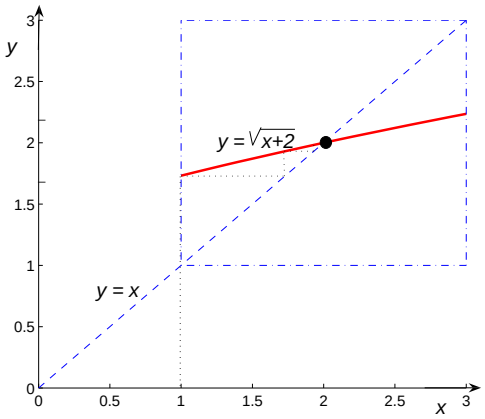


Figura 5.2: La función de iteración $g(x) = \sqrt{x+2}$ produce una sucesión monótona convergente.

Entonces para obtener una sucesión convergente $\xi = 2$ basta tomar $x_0 \geq \sqrt{2}$. Es decir, la función de iteración es una buena elección si nos restringimos al intervalo $[\sqrt{2}, 3]$. En este intervalo podemos tomar $x_0 = 1{,}5$, y obtenemos $x_0 = 1{,}5$, $x_1 = 2{,}333333$, $x_2 = 1{,}85714$, $x_3 = 2{,}07692$, $x_4 = 1{,}96296$,... Observese el caracter alternante de la sucesión alrededor de $\xi = 2$. Esto debido a que $g'(x) = -\dfrac{2}{x^2} < 0$ y, en particular, $g'(\xi) = g'(2) = -0{,}5$. La Figura 5.3 ilustra esta situación.

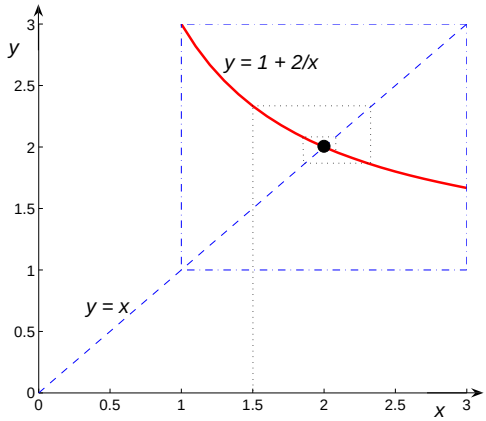


Figura 5.3: La función de iteración $g(x) = 1 + 2/x$ produce una sucesión alternante convergente.

En general, si g es la función de iteración y ésta es una contracción en X, con $\xi \in X$, se

satisfacen las siguientes propiedades para la sucesión generada por iteración de punto fijo:

1. Cuando $g'(x) > 0$ para toda $x \in X$. La sucesión x_n es creciente si $x_0 < \xi$. La sucesión x_n es decreciente si $x_0 > \xi$.

2. Cuando $g'(x) < 0$ para toda $x \in X$. La sucesión es alternante alrededor de ξ en la forma $x_0 < x_2 < x_4 < \ldots < \xi < \ldots < x_5 < x_3 < x_1$, o bien en la forma $x_1 < x_3 < x_5 < \ldots < \xi < \ldots < x_4 < x_2 < x_0$.

5.4. Orden de convergencia

Teorema 5.6. *Sea $f(x) = 0$ una ecuación no lineal y $x = g(x)$ su correspondiente ecuación de punto fijo. Bajo las siguientes condiciones:*

1. *g es una contracción sobre X.*

2. *$g \in \mathcal{C}^1(X)$ (g y g' son continuas en X).*

3. *g es estrictamente monótona sobre X ($g'(x) \neq 0\,,\ \forall\, x \in X$).*

se tiene que

$$\textit{Si } x_0 \neq \xi\,,\ \textit{entonces } x_n \neq \xi\,,\ \forall\, n \in \mathbb{N}\,,$$

es decir, el proceso iterativo no puede terminar en un número finito de pasos.

Demostración. Una demostración de este hecho se obtiene suponiendo lo contrario, es decir que $g(x_n) = x_n$ para algún n. Si n es el primer índice para el cual esto ocurre, entonces

$$x_n = g(x_{n-1}) \text{ y } x_n = g(x_n) \text{ con } x_{n-1} \neq x_n\,.$$

Luego, por el teorema del valor medio

$$0 = g(x_n) - g(x_{n-1}) = g'(n)(x_n - x_{n-1}) \text{ con } \eta \text{ entre } x_{n-1} \text{ y } x_n\,,$$

y debe tenerse $g'(\eta) = 0$ con $\eta \in X$, lo cual contradice la hipótesis de que $g'(x) \neq 0\ \forall x \in I$.

Comportamiento asintótico del error

Una vez hecho lo anterior, ahora podemos analizar como se comporta el error en el método iterativo de punto fijo la ir aumentando n:

$$e_{n+1} = x_{n+1} - \xi = g(x_n) - g(\xi) = g'(\eta_n)(x_n - \xi) = g'(\eta_n)e_n\,.$$

Es decir

$$e_{n+1} = g'(\eta_n)e_n \quad \text{con } \eta_n \text{ entre } x_n \text{ y } \xi\,.$$

Como η_n está entre x_n y ξ para cada n, y ξ es el límite de x_n, necesariamente se tiene que

$$\lim_{n\to\infty} \eta_n = \xi.$$

Además

$$\lim_{n\to\infty} g'(\eta_n) = g'(\xi),$$

por ser $g'(x)$ continua. Así que

$$e_{n+1} = g'(\eta_n)e_n = (g'(\xi) + \epsilon_n)e_n$$

donde $\varepsilon_n \to 0$ cuando $n \to \infty$. Luego

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n} = g'(\xi) + \lim_{n\to\infty} \varepsilon_n = g'(\xi)\,.$$

De aquí que, para valores suficientemente grandes de n (asintóticamente) se tendrá que

$$e_{n+1} \approx g'(\xi)e_n\,.$$

Esta última relación establece que el error en la iteración $n+1$ depende (más o menos) linealmente del error en la iteración n. Por tanto, podemos decir que la sucesión $\{x_n\}$ *converge linealmente* a ξ, y que el método de punto fijo es un método con orden de *convergencia lineal* o de orden uno. En general, como desconocemos $g'(\xi)$ no es posible estimar e_{n+1} en téminos de e_n. Lo único que sabemos es que e_{n+1} es igual a una constante (menor que 1) multiplicada por e_n, y en consecuencia $e_{n+1} < e_n$.

Con el objeto de ilustrar este tipo de convergencia, consideremos la función de iteración $g(x) = \sqrt{x+2}$ en el Ejemplo 3.5. Para esta función $g'(\xi) = g'(2) = 0{,}25$, así que asintóticamente tenemos que $e_{n+1} \approx g'(\xi)e_n = 0{,}25e_n$. Es decir, asintóticamente, el error en la siguiente iteración es un cuarto del error anterior. La Tabla 5.2 muestra que efectivamante así es.

Definición del orden de convergencia

Sea $\{x_n\}$ una sucesión que converge a un número ξ, y sea $e_n = x_n - \xi$. Si existe un número $p > 0$, y una constante $\lambda \neq 0$ tal que

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^p} = \lambda$$

error exacto	$0.25\ e_n$
$e_5 = 0{,}001070$	$- - - - -$
$e_6 = 0{,}000268$	0,00026750
$e_7 = 0{,}000067$	0,00006700
$e_8 = 0{,}000017$	0,00001675

Cuadro 5.2: Error real y error asintótico de la sucesión de punto fijo con función de iteración $g(x) = \sqrt{x+2}$.

entonces p se denomina el *orden de convergencia* de la sucesión y λ se denomina la *constante asintótica del error*. Entre mayor sea p, mayor será el orden de convergencia y la sucesión convergerá "más rápido" al límite ξ:

Si $p = 1$, la convergencia se denomina lineal, y asintóticamente

$$|e_{n+1}| \sim \lambda |e_n| \,.$$

Si $p = 2$, la convergencia se denomina cuadrática, y asintóticamente

$$|e_{n+1}| \sim \lambda |e_n|^2 \,.$$

Por otro lado, si $0 < \lambda < 1$ decrece, la sucesión también tendría una mejor convergencia, pero no tan dramática como aquella cuando p aumenta.

Ejemplo 5.7. *Ya hemos mostrado que la sucesión generada por la función de iteración* $g(x) = \sqrt{x+2}$ *converge linealmente al punto fijo* $\xi = 2$. *Por otro lado, la función de iteración* $g(x) = \dfrac{x^2+2}{2x-1}$ *produce una sucesión* $\{x_n\}$ *que aparentemente tiene un orden de convergencia mayor a uno.*

En realidad, esta sucesión converge cuadráticamente al punto fijo $\xi = 2$. Una forma de verificar esto es la siguiente: La convergencia cuadrática implica que existe $\lambda > 0$ tal que

$$|e_{n+1}| \sim \lambda |e_n|^2 \text{ para } n \text{ grande} \,.$$

Para verificar esto basta con ver que los cocientes $|e_{n+1}|/|e_n|^2$ producen un valor aproximadamente constante $\lambda \sim 0{,}3$.

5.5. La convergencia cuadrática

En la discusión de convergencia lineal supusimos $g'(x) \neq 0\,,\ \forall\, x \in X$, y en particular $g'(\xi) \neq 0$. Ahora queremos investigar el comportamiento asintótico del error si $g'(\xi) = 0$.

Suponiendo ahora que la contracción g satisface $g \in \mathcal{C}^2(X)$, y que $g''(x) \neq 0 \ \forall\, x \in X$, se puede demostrar que

1. si $x_0 \neq \xi$, entonces $x_n \neq \xi\,,\ \forall\, n \in \mathbb{N}$.

2. El orden de la convergencia de la sucesión generada por iteración de punto fijo es 2.

Demostración. Primero demostraremos que $e_n = x_n - \xi \neq 0\,,\ \forall\, n \in \mathbb{N}$. Supongase lo contrario, y sea n el primer entero para el cual $x_n = \xi$. Entonces

$$x_n = g(x_{n-1}) = g(x_n) \text{ con } x_{n-1} \neq x_n\,.$$

Por el teorema de Taylor

$$0 = g(x_n) - g(x_{n-1}) = g'(x_n)(x_n - x_{n-1}) + \frac{g''(c_n)}{2!}(x_n - x_{n-1})^2\,.$$

con c_n entre x_{n-1} y x_n. Como $g'(x_n) = g'(\xi) = 0$ y $x_{n-1} \neq x_n$ se debe cumplir $g''(c_n) = 0$, lo cual contradice la hipótesis $g''(x) \neq 0\ \forall x \in X$.

Ahora verificamos que bajo las suposiciones anteriores se obtiene convergencia cuadrática:

$$e_{n+1} = x_{n+1} - \xi = g(x_n) - g(\xi) = g'(\xi)(x_n - \xi) + \frac{1}{2}g''(\theta_n)(x_n - \xi)^2$$

con θ_n entre x_n y ξ. Dado que $g'(\xi) = 0$, se obtiene

$$e_{n+1} = \frac{1}{2}g''(\theta_n)e_n^2\,,$$

Como $\theta_n \to \xi$ cuando $n \to \infty$ y $g \in \mathcal{C}^2(X)$, entonces

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^2} = \frac{1}{2}|g''(\xi)|$$

lo cual implica que la sucesión $\{x_n\}$ converge cuadráticamente al punto fijo ξ. Por lo tanto, para valores grandes de n se tiene

$$|e_{n+1}| \approx \frac{1}{2}|g''(\xi)|\,|e_n|^2\,,$$

lo cual muestra que el error en la iteración $n+1$ es proporcional al cuadrado del error en la iteración n, con constante de proporcionalidad igual a $\lambda = \frac{1}{2}|g''(\xi)|$ (la constante del error asintótico).

Una mejor forma para ver el comportamiento de esquemas de segundo orden es tomar $d_n =$ número de decimales correctos en la iteración n, es decir

$$|e_n| = |x_n - \xi| = 10^{-d_n}\,,\ \text{con } d_n > 0\,.$$

Entonces

$$10^{-d_n} = |x_n - \xi| \leq \lambda |x_{n-1} - \xi|^2 = \lambda 10^{-2d_{n-1}}.$$

Luego

$$-d_n \leq \log_{10} \lambda - 2d_{n-1}\,.$$

Por lo tanto

$$d_n \geq 2d_{n-1} - \log_{10} \lambda\,.$$

Entonces si $\lambda < 1$, se tiene $-\log_{10} \lambda > 0$, y el número de decimales correctos en la iteración n es mayor que el doble de decimales correctos en la iteración $n-1$. Si $\lambda > 1$, se sigue cumpliendo que $d_n \geq 2d_{n-1}$ para n sufcientemente grande pues en este caso $d_n \gg \log \lambda$.

Ejemplo 5.8. *Consideremos de nuevo el problema de calcular la raíz $\xi = 2$ de la función $f(x) = x^2 - x - 2$. Verificar que la función de iteración*

$$g(x) = \frac{x^2 + 2}{2\,x - 1}$$

produce una sucesión que converge cuadráticamente a $\xi = 2$, tomando el valor de comienzo $x_0 = 2{,}1$.

Solución. El número de decimales correctos en la primera iteración es $d_1 = 2$ debido a que $x_1 = 2{,}003125$, por lo que el número de decimales en la segunda iteración será

$$d_2 \geq 2d_1 - \log_{10} \lambda = 4 - \log_{10} 0{,}3 = 4{,}52$$

Como $x_2 = 2{,}00003$, el número de decimales correctos en realidad es 4. La fórmula predice que el número de decimales correctos en la tercera iteración será

$$d_3 \geq 2d_2 - \log_{10} \lambda = 8 + 0{,}52 = 8{,}52$$

Pero en realidad se obtiene 11 decimales correctos, pues

$$x_3 = 2{,}000000000004$$

Convergencia de orden mayor a dos

Generalizando, podemos describir fácilmente los esquemos que convergen más rápidamente que los de segundo orden: supóngase que $g \in \mathcal{C}^k$ con $k > 2$, y que en el punto fijo ξ

$$g'(\xi) = g''(\xi) = \ldots = g^{(k-1)}(\xi) = 0\,,$$

y que $g^{(k)}(x) \neq 0\,,\ \forall\, x \in X$. Haciendo una expansión en series de Taylor obtenemos:

$$e_{n+1} = x_{n+1} - \xi = g(x_n) - g(\xi) = \frac{g^{(k)}(c_n)}{k!}(x_n - \xi)^k = \frac{g^{(k)}(c_n)}{k!}e_n^k\,.$$

por ser $g'(\xi) = g''(\xi) = \ldots = g^{(k-1)}(\xi) = 0$. Por lo tanto

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^k} = \lambda\,,$$

lo cual implica un orden de convergencia k, con constante asintótica del error $\lambda = |g^{(k)}(\xi)/k!|$.

5.6. El método de Newton

Puede tenerse la impresión que la discusión de convergencia cuadráica en la sección anterior no tiene mucho valor prático, ya que en una ecuación de la forma $x = g(x)$ esperariamos que la condición $g'(\xi) = 0$ se satisfaga solo accidentalmente. Sin embargo, mostraremos que, al menos para funciones diferenciales g, el procedimiento básico del método de iteración de punto fijo puede reformularse de tal manera que llegue a ser cuadráticamente convergente.

El problema general es calcular las raices de $f(x) = 0$ en algún intervalo, digamos $a \leq x \leq b$, que llamamos X. Sea $\phi(x)$ una función en $\mathcal{C}^{(1)}(X)$ que no tenga raices en X, es decir

$$0 < |\phi(x)| < \infty\,,\ \forall\, x \in X\,.$$

Entonces, la ecuación

$$x = g(x) \equiv x - \phi(x)f(x)\,,$$

tiene exactamente las mismas raices que $f(x)$ en $[a, b]$. Muchos de los métodos iterativos usuales pueden obtenerse con elecciones especiales de $\phi(x)$. Por ejemplo, para que el método de iteración de punto fijo tenga convergencia cuadrática, una de las condiciones es que $g'(\xi) = 0$. Como

$$g'(x) = 1 - \phi'(x)f(x) - \phi(x)f'(x)\,,$$

entonces para que

$$g'(\xi) = 1 - \phi'(\xi)f(\xi) - \phi(\xi)f'(\xi) = 0$$

se debe tener $\phi(\xi) = 1/f'(\xi)$. Por lo tanto, es suficiente con escoger $\phi(x) = 1/f'(x)$, y con esta elección se obtiene la función de iteración

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

Suponiendo $f'(x) \neq 0,\ \forall\, x \in X$, el algoritmo que resulta es conocido como el ***Método de Newton*** (también conocido como método de Newton-Raphson):

Escoger x_0 cercano a ξ, y determinar la sucesión $\{x_n\}$ por medio de la relación de recurrencia

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\,, \quad n = 0,\ 1,\ 2,\ \ldots$$

La convergencia de este método inmediatamente se sigue del teorema de convergencia cuadrática, dado que se escogió $g(x) = x - f(x)/f'(x)$ para satisfacer $g'(\xi) = 0$. Sin embargo, como veremos más adelante, en general debemos tener x_0 suficientemente cercano al punto fijo ξ para asegurar el éxito del algoritmo.

Ejemplo 5.9. *Encontrar al raiz de $f(x) = x^2 - x - 2$ en el intervalo $[1, 3]$ utilizando el método de Newton.*

Solución. El método de Newton para la ecuación correspondiente es: Dado x_0, generar $\{x_n\}$ por medio de

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \\ &= x_n - \frac{x_n^2 - x_n - 2}{2x_n - 1} \\ &= \frac{x_n^2 + 2}{2x_n - 1} \end{aligned}$$

La función de iteración $g(x) = x - f(x)/f'(x) = (x^2 + 2)/(2x - 1)$ es la misma que en los dos ejemplos anteriores. Esta función satisface

$$\begin{aligned} g'(\xi) &= \frac{f''(\xi)f(\xi)}{[f'(\xi)]^2} = \frac{2(\xi^2 - \xi - 2)}{(2\xi - 1)^2} = 0\,, \quad \text{pues } f(\xi) = 0, \\ g''(\xi) &= \frac{18}{(2\xi - 1)^3} = \frac{2}{3}\,, \quad \text{pues } \xi = 2, \\ \lambda &= |\frac{1}{2}g''(\xi)| = \frac{1}{3} \Longrightarrow e_{n+1} \approx \lambda e_n^2 = \frac{1}{3}\, e_n^2 \text{ para } n \text{ grande.} \end{aligned}$$

Por lo tanto, si comenzamos las iteraciones con $x_0 = 1.5$ obtenemos

n	x_n	e_n	λe_n^2
0	1.5	0.5	
1	2.125	0.125	0.083333
2	2.00480769230769	0.0048077	0.0052083
3	2.00000768001966	0.00000768	0.0000077
4	2.00000000001966	$1{,}97 \times 10^{-11}$	$1{,}966 \times 10^{-11}$

Observese como el valor asintótico λe_n^2 se aproxima a e_{n+1} cuando n aumenta. Además, dado que el número de decimales correctos en la segunda iteración es 2, en la tercera iteración será

$$d_3 \geq 2d_2 - \log_{10} \frac{1}{3} \cong 4{,}477$$

Como en la tercera iteración $d_3 = 5$, en la cuarta será aproximadamente

$$d_4 \geq 2d_3 + 0{,}477 = 10{,}477$$

y el valor real es $d_4 = 11$.

5.6.1. Interpretación geométrica del método de Newton

El método de Newton tiene una interpretación geométrica muy sencilla. Dada la ecuación $f(x) = 0$ en una variable, suponiendo conocido la aproximación x_n a la raíz ξ, x_{n+1} se calcula como

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\,,$$

es decir

$$f(x_n) + (x_{n+1} - x_n)f'(x_n) = 0\,.$$

Si reemplazamos x_{n+1} por la variable x, obtenemos la función $y = f(x_n) + f'(x_n)(x - x_n)$, la cual tiene como gráfica a la recta tangente a $f(x)$ en el punto $(x_n, f(x_n))$. Así que x_{n+1} es la intersección de esta recta con el eje x, como se ilustra en la figura siguiente.

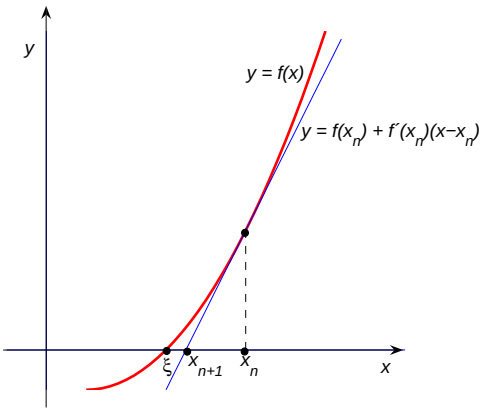


Figura 5.4: Interpretación geométrica del método de Newton

El método de Newton es uno de los métodos más usados para calcular numericamente raices de ecuaciones no lineales, pues si se escoge adecuadamente la aproximación inicial

x_0, la sucesión obtenida por iteración converge razonablemente rápido a la solución. Sin embargo, este es un método denominado *local*, pues si x_0 no es lo suficientemente cercano a ξ y la función $f(x)$ no tiene "buenas propiedades", se pueden presentar algunos problemas. El siguiente ejemplo muestra algunas de las situaciones indeseables en el método de Newton.

Ejemplo 5.10. *Considere el problema* $f(x) = \sin(x) = 0$ *en el intervalo* $X = [-\pi/2,\, \pi/2]$. *El método de Newton es: dado* $x_0 \in [-\pi/2,\, \pi/2]$, *generar* $\{x_n\}$ *por medio de*

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{\sin(x_n)}{\cos(x_n)} = x_n - \tan(x_n)\,.$$

Si escogemos $x_0 = \pi/2$, entonces $x_1 = -\infty$. De hecho, si escogemos x_0 cerca de $\pi/2$ ó $-\pi/2$, la linea tangente intersecta al eje x fuera del intervalo y muy lejos del mismo. Por ejemplo, tomando $x_0 = 1{,}4$ se obtiene $x_1 = -4{,}3979$. Esta situación se ilustra en la Figura 5.5.

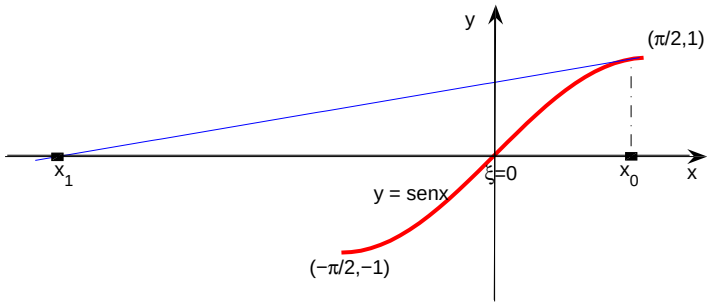


Figura 5.5: Situación indeseable en el método de Newton: x_1 cae muy lejos de ξ

Otro aspecto indeseable es que si escogemos $x_0 = x^*$ tal que

$$\tan(x^*) = 2x^* \quad (x^* \approx 1{,}1655\ldots)\,,$$

entonces

$$x_1 = x_0 - \tan(x_0) = x_0 - 2x_0 = -x_0$$
$$x_2 = x_1 - \tan(x_1) = -x_0 - \tan(-x_0) = x_0$$

y entramos dentro de un ciclo obteniendo $x_0,\ -x_0,\ x_0,\ -x_0, \ldots$ como se ilustra en la Figura 5.6.

La primera situación indeseable se suprime si exigimos que

$$a \le a - \frac{f(a)}{f'(a)} \le b\,, \quad \text{y} \quad a \le b - \frac{f(b)}{f'(b)} \le b$$

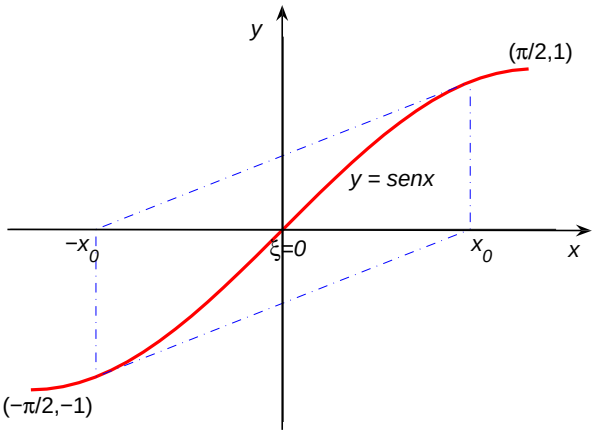


Figura 5.6: Situación indeseable en el método de Newton: ciclo infinito

es decir si

$$0 \leq -\frac{f(a)}{f'(a)} \leq b-a\,, \quad \text{y} \quad a-b \leq -\frac{f(b)}{f'(b)} \leq 0,$$

o equivalentemente si exigimos que

$$\left|\frac{f(a)}{f'(a)}\right| \leq b-a\,, \quad \text{y} \quad \left|\frac{f(b)}{f'(b)}\right| \leq b-a.$$

La segunda situación indeseable se suprime si exigimos que la función $f(x)$ no cambie de concavidad en el intervalo, es decir si exigimos

$$f''(x) \geq 0 \quad \text{ó} \quad f''(x) \leq 0\,,\ \forall\, x \in X\,.$$

Estas observaciones se incorporan para establecer condiciones suficientes para asegurar la convergencia del método de Newton.

5.6.2. Teorema de convergencia no-local para el méodo de Newton

El siguiente teorema resume las observaciones anteriores especificando condiciones suficientes para que el método de Newton converja a la raíz independientemente de como se escoja x_0 dentro del intervalo $[a,b] = X$.

Teorema 5.11. *Sea $f \in \mathcal{C}^2(X)$, y supóngase que f satisface las siguientes condiciones*

1. $f(a)f(b) < 0$,
2. $f'(x) \neq 0\,, \forall\, x \in X$,

3. $f''(x) \geq 0$ *ó* $f''(x) \leq 0$, $\forall\, x \in X$

4. $\left|\dfrac{f(a)}{f'(a)}\right| \leq b-a \quad y \quad \left|\dfrac{f(b)}{f'(b)}\right| \leq b-a,$

entonces el método de Newton convergerá la única solución ξ *de* $f(x)=0$*, para cualquier elemento* $x_0 \in X = [a,b]$

La literatura sobre el método de Newton es muy extensa. En particular, la convergencia puede demostrarse bajo varios conjuntos de condiciones diferentes a las del teorema anterior. Sin embargo, en este curso no consdieraremos otro tipo de condiciones.

Ejemplo 5.12. *El problema que hemos venido considerando,* $f(x) = x^2 - x - 2 = 0$*, cumple las condiciones del ejemplo anterior en el intervalo* $[1,3]$*:*

1. $f(1)f(3) = -8 < 0$*;*

2. $f'(x) = 2x - 1 > 0$ *en* $[1,3]$*;*

3. $f''(x) = 2 > 0\,,\ \forall\, x \in [1,3]$*;*

4. $\left|\dfrac{f(1)}{f'(1)}\right| = \dfrac{2}{1} \leq 3-1, \quad \left|\dfrac{f(3)}{f'(3)}\right| = \dfrac{4}{5} \leq 3-1$

5.6.3. Algunas modificaciones del método de Newton

El método de Newton requiere de la evaluación de f y f' en cada iteración. Mientras que en muchos problemas el cálculo de f' es trivial, en muchos otro puede llegar a representar un verdadero problema. Por otro lado, desde el punto de vista computacional, la precisión alcanzada con el método de Newton depende de la precisión con cual $\frac{f(x)}{f'(x)}$ puede evaluarse. Puede suceder que $f'(x)$ sea muy pequeño aunque no sea cero, y cualquier error al calcular $f'(x)$ será aumentado al dividir por esta cantidad. Por estas razones, en ocasiones es conveniente considerar otros métodos que no requieran evaluaciones de $f'(x)$ y que retengan algunas de las propiedades de convergencia del método de Newton.

Método de la cuerda (Whittaker)

Una forma fácil de evitar el cálculo de $f'(x_n)$ es reemplazar este valor por un valor constante m, obteniendo el método

$$x_{n+1} = x_n - \frac{f(x_n)}{m}\,.$$

Suponiendo que x_n es cercano a la raiz ξ, entre más cercana sea la constante m a $f'(\xi)$ mejor será el método.

Ejemplo 5.13. *Para calcular la raíz $\xi = 2$ de $f(x) = x^2 - x - 2 = 0$ en el intervalo $[1, 3]$ podemos escoger*

$$m = \frac{f'(1) + f'(3)}{2} = \frac{1+5}{2} = 3\,.$$

Si tomamos $x_0 = 1$, la regla $x_{n+1} = x_n - f(x_n)/m$ produce en doble precisión $x_1 = 1.6666666666667$, $x_2 = 1.96296296296296$, $x_3 = 1.99954275262917$, $x_4 = 1.99999993030828$, $x_5 = 2$, y observamos que en este caso la sucesión converge cuadráticamente.

En realidad, en este ejemplo se obtiene convergencia cuadrática porqué casualmente $m = 3$ es igual a $f'(2)$. Sin embargo, en general el método tiene convergencia lineal, y como ya mencionamos arriba la convergencia solo será cercana a la cuadrática cuando $m = f'(\xi)$. Si la estimación de m es buena, la convergencia puede ser muy rápida. Especialmente en los estados finales del método de Newton no es necesario recalcular $f'(x)$ en cada paso, y una buena estrategia es cambiar al método de la cuerda.

Método de la secante

En este caso el valor de la derivada $f'(x_n)$ en el método de Newton se reemplaza por el cociente de diferencias:

$$\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}\,,$$

obteniendo la fórmula de recurrencia:

$$x_{n+1} = \frac{x_{n-1}f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}\,.$$

En el método de la secante debemos encontrar dos iteraciones sucesivas $x_0,\ x_1$ antes de empezar a usar la fórmula de recurrencia. Sin embargo, solo es necesario evaluar la función una vez en cada paso, ya que el valor anterior $f(x_{n-1})$ puede retenerse. Esto representa una ventaja sobre el método de Newton donde forzosamente necesitamos dos evaluaciones, una para $f(x_n)$ y otra para $f'(x_n)$.

El orden de convergencia de la sucesión generada por el método de la secante no puede deducirse por un análisis semejante al de punto fijo, pues este método no puede expresarse en la forma de punto fijo $x_{n+1} = g(x_n)$. Una investigación detallada del método muestra que su orden de convergencia es fraccionario y se encuentra entre 1 y 2. Asintóticamente (Conte-deBoor)

$$|e_{n+1}| \sim \left|\frac{f''(\xi)}{2f'(\xi)}\right|^{1/p} |e_n|^p \quad \text{con} \quad p = \frac{1+\sqrt{5}}{2} \sim 1{,}618$$

A una convergencia de este tipo se le denomina *superlineal*

El método de la secante tiene una interpretación geométrica análoga a la del método de Newton. El valor x_{n+1} representa la intersección con el eje x de la recta que pasa por los puntos $(x_{n-1}, f(x_{n-1}))$ y $(x_n, f(x_n))$, de ahí el nombre de método de la secante. Esto se ilustra en la Figura 5.7.

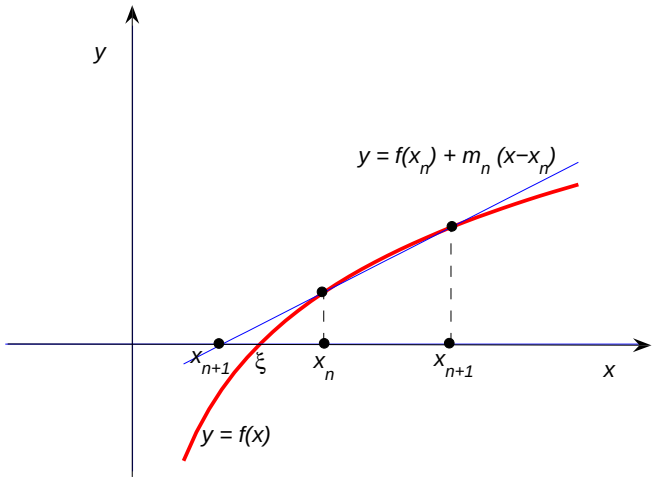


Figura 5.7: Ilustración geométrica del método de la secante

Ejemplo 5.14. *Utilizar el método de la secante para encontrar la solución de* $f(x) = x^2 - x - 2 = 0,\ x \in [1,3]$ *en el intervalo* $[1,3]$.

Solución. Escogiendo $x_0 = 1$, $x_1 = 3$, al aplicar el método de la secante, se obtienen los valores mostrados en la Tabla 5.3 cuando se utiliza aritmética de punto flotante de doble precisión en el ambiente *MATLAB*.

Analizando estos resultados se observa la denominada convergencia superlineal.

5.7. Método de punto fijo para sistemas de ecuaciones

Consideraremos brevemente el caso de sistemas no lineales de ecuaciones de la forma

$$f(x) = 0,$$

donde $f : \mathbb{R}^n \to \mathbb{R}^n,\ f = (f_1, \ldots, f_n)$. Es decir, f es una función vectorial con las funciones componentes $f_1, f_2, \ldots, f_n$, cada una de las cuales es una función escalar en las variables

n	x_n	e_n	$\left\lvert\frac{f''(\xi)}{2f'(\xi)}\right\rvert^{1/p} \lvert e_n\rvert^p$ (error asintótico)
0	1	1	
1	3	1	
2	1.66666666666667	0.33333333333333	
3	1.90909090909091	0.09090909090909	0.08572834524805
4	2.01176470588235	0.01176470588235	0.01047403661108
5	1.99963383376053	3.661662397×10^{-4}	3.830517471×10^{-4}
6	1.99999856948921	1.43051079×10^{-6}	1.39648907×10^{-6}
7	2.00000000017462	1.7462×10^{-10}	1.7722×10^{-10}
8	2.00000000000000	0	0

Cuadro 5.3: Resultados obtenidos por el método de la secante.

$x_1,\ x_2, \ldots,\ x_n :\ f_i(x) = f_i(x_1, x_2, \ldots, x_n)$, $i = 1, \cdots, n$. Primero expresamos el anterior sistema de ecuaciones en la forma de un sistema de ecuaciones de punto fijo

$$x = g(x)\,,$$

o bien, escribiendo componente a componente

$$x_i = g_i(x_1, x_2, \ldots, x_n)\,, \quad i = 1,\, 2, \ldots,\, n\,.$$

Esperamos que la función de iteración $g : \mathbb{R}^n \to \mathbb{R}^n$ tenga un punto fijo en un subconjunto $X \subset \mathbb{R}^n$. El algoritmo de iteración de punto fijo es: *Dado $x^0 \in X$, generamos la sucesión $\{x^k\}_{k=1}^{\infty}$ por medio de*

$$x^{k+1} = g(x^k)\,, \quad k = 0,\, 1,\, 2, \ldots$$

Está claro que para cada k, x^k es el vector con componentes $x_1^k,\, x_2^k, \cdots, x_n^k$.

El análisis de los métodos de iteración de punto fijo para sistemas es el mismo que aquel para ecuaciones de una variable. La única diferencia es que para los temas de convergencia y error utilizamos una norma vectorial en lugar del valor absoluto. Por ejemplo, para que la función de iteración sea una contracción basta con que en alguna norma matricial $\|\cdot\|$,

$$\|Jg(x)\| \leq K < 1\,, \qquad \forall\, x \in X\,,$$

donde $0 < K < 1$ es alguna constante, y $Jg(x)$ la matriz Jacobiana de g en $x \in X$:

$$Jg(x) = \Big(\frac{\partial g_i(x)}{\partial x_j}\Big)_{1\leq\, i,\, j\, \leq n}$$

La anterior es una condición suficiente y se puede jusificar utilizando del teorema del valor medio para derivadas:

$$g(x) - g(y) = Jg(\theta)\ (x - y)\,, \quad \forall\, x,\, y \in X\,, \quad \text{y} \quad \theta = x + \lambda(y - x)\,,$$

con $0 < \lambda < 1$. En consecuencia

$$\|g(x) - g(y)\| \leq \|Jg(\theta)\|\ \|(x - y)\| \leq K\|(x - y)\|$$

Por supuesto, la norma de la matriz Jacobiana es la norma matricial inducida por la norma

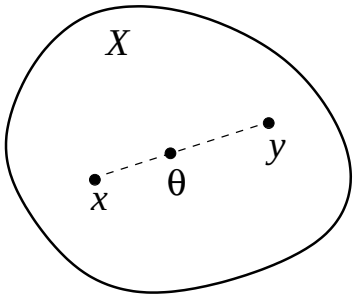


Figura 5.8: θ se encuentra entre x y y.

vectorial aplicada a la diferencia $x - y$ y también a $g(x) - g(y)$.

Ejemplo 5.15. *Encontrar una solución del sistema tres ecuaciones no lineales con tres incógnitas*

$$\begin{aligned} 3x_1 - \cos(x_2 x_3) &= \frac{1}{2} \\ x_1^2 - 81(x_2 + 0.1)^2 + \operatorname{sen} x_3 + 1.06 &= 0 \\ e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} &= 0 \end{aligned}$$

en el conjunto $X = [-1, 1]^3$.

Solucion. La función vectorial $f(x)$, $x = (x_1, x_2, x_3)^T$, que define el sistema de ecuaciones tiene las funciones componentes

$$\begin{aligned} f_1(x_1, x_2, x_3) &= 3x_1 - \cos(x_2 x_3) - \frac{1}{2} \\ f_2(x_1, x_2, x_3) &= x_1^2 - 81(x_2 + 0.1)^2 + \operatorname{sen} x_3 + 1.06 \\ f_3(x_1, x_2, x_3) &= e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} \end{aligned}$$

Un posible sistema de punto fijo se obtiene despejando x_1 de la primera ecuación, x_2 de la segunda y x_3 de la tercera:

$$\begin{aligned} x_1 &= \frac{1}{6} + \frac{1}{3}\cos(x_2 x_3) &&= g_1(x_1, x_2, x_3) \\ x_2 &= \frac{1}{9}\sqrt{x_1^2 + \operatorname{sen} x_3 + 1{,}06} - 0{,}1 &&= g_2(x_1, x_2, x_3) \\ x_3 &= -\frac{1}{20}e^{-x_1 x_2} - \frac{10\pi - 3}{60} &&= g_3(x_1, x_2, x_3) \end{aligned}$$

La matriz Jacobiana de la función de iteración

$$Jg(x) = \begin{bmatrix} \dfrac{\partial g_1}{\partial x_1} & \dfrac{\partial g_1}{\partial x_2} & \dfrac{\partial g_1}{\partial x_3} \\ \dfrac{\partial g_2}{\partial x_1} & \dfrac{\partial g_2}{\partial x_2} & \dfrac{\partial g_2}{\partial x_3} \\ \dfrac{\partial g_3}{\partial x_1} & \dfrac{\partial g_3}{\partial x_2} & \dfrac{\partial g_3}{\partial x_3} \end{bmatrix}$$

en este caso es

$$Jg(x) = \begin{bmatrix} 0 & -\dfrac{x_3}{3}\operatorname{sen}(x_2 x_3) & -\dfrac{x_2}{3}\operatorname{sen}(x_2 x_3) \\ \dfrac{x_1}{9\sqrt{x_1^2 + \operatorname{sen} x_3 + 1{,}06}} & 0 & \dfrac{\cos x_3}{18\sqrt{x_1^2 + \operatorname{sen} x_3 + 1{,}06}} \\ \dfrac{x_2}{20}e^{-x_1 x_2} & \dfrac{x_1}{20}e^{-x_1 x_2} & 0 \end{bmatrix}$$

Si tomamos $X = [-1, 1]^3$, es decir $-1 \le x_1,\, x_2,\, x_3 \le 1$, entonces la ecuación de punto fijo satisface $g(X) \subset X$, pues

$$\begin{gathered} \frac{1}{6} + \frac{1}{3}\cos(1) \le x_1 \le \frac{1}{6} + \frac{1}{3}\cos(0) \\ \frac{\sqrt{0 + \operatorname{sen}(-1) + 1{,}06}}{9} - 0{,}1 \le x_2 \le \frac{\sqrt{1 + \operatorname{sen}(1) + 1{,}06}}{9} - 0{,}1 \\ -\frac{e}{20} - \frac{10\pi - 3}{60} \le x_3 \le -\frac{e^{-1}}{20} - \frac{10\pi - 3}{60} \end{gathered}$$

es decir

$$\begin{aligned} 0{,}346 &\le g_1(x_1,\, x_2,\, x_3) \le 0{,}5 \\ -0{,}048 &\le g_2(x_1,\, x_2,\, x_3) \le 0{,}089 \\ -0{,}610 &\le g_3(x_1,\, x_2,\, x_3) \le -0{,}492\,. \end{aligned}$$

Por lo tanto, la función de iteración tiene su imagen dentro de $X = [-1, 1]^3$. Además, si consideramos la norma infinito, se obtiene $\|Jg(x)\|_\infty$ es el máximo de las cantidades

$$\frac{|\operatorname{sen}(x_2 x_3)|}{3}(|x_2| + |x_3|),\ \frac{|x_1| + |\cos x_3|/2}{9\sqrt{x_1^2 + \operatorname{sen} x_3 + 1{,}06}},\ \frac{e^{-x_1 x_2}}{20}(|x_1| + |x_2|)$$

con $-1 \leq x_1,\, x_2,\, x_3 \leq 1$. Es decir

$$\|Jg(x)\|_\infty \leq \text{máx}\left\{\frac{2\,\text{sen}(1)}{3},\ \frac{3/2}{9\sqrt{0{,}06}},\ \frac{2\,e}{20}\right\} \approx 0{,}6804\,,$$

de tal forma que g es una contracción en $X = [-1,1]^3$ con $K \leq 0{,}6804$. Si tomamos como punto de comienzo $x^0 = (0{,}2, 0{,}1, -0{,}1)^T$, se obtienen los valores mostrados en la Tabla 5.4

k	x_1	x_2	x_3
0	0.2	0.1	-0.1
1	0.49998333347222	0.01112036535646	-0.52260870926364
2	0.49999437090050	0.00005190019018	-0.52332154714020
3	0.49999999987705	0.00001447284345	-0.52359747812499
4	0.49999999999043	0.00000006935321	-0.52359841377852
5	0.5	0.00000001934170	-0.52359877386447
6	0.5	0.00000000009269	-0.52359877511476
7	0.5	0.00000000002585	-0.52359877559598
8	0.5	0.00000000000012	-0.52359877559765
9	0.5	0.00000000000003	-0.52359877559830
10	0.5	0.0	-0.52359877559830

Cuadro 5.4: Resultados de la iteración de punto fijo.

La solución exacta el sistema es $x_1 = \dfrac{1}{2},\ x_2 = 0,\ x_3 = -\dfrac{\pi}{6}$.

5.7.1. Aceleración de tipo Seidel en las iteraciones

Para sistemas de ecuaciones puede acelerarse la convergencia del método de punto fijo utilizando las últimas estimaciones $x_1^{k+1}, x_2^{k+1}, \ldots, x_{i-1}^{k+1}$ en lugar de $x_1^k, x_2^k, \ldots, x_{i-1}^k$ para calcular x_i^{k+1}, para $2 \leq i \leq n$, como se hace en el método de Gauss-Seidel para sistemas lineales. En el ejemplo anterior, las nueva realción de recursión al realizar aceleración del tipo Seidel es:

$$\begin{aligned}
x_1^{k+1} &= \frac{1}{6} + \frac{1}{3}\cos(x_2^k x_3^k) \\
x_2^{k+1} &= \frac{1}{9}\sqrt{(x_1^{k+1})^2 + \text{sen}\, x_3^k + 1{,}06} - 0{,}1 \\
x_3^{k+1} &= -\frac{1}{20}e^{-x_1^{k+1}x_2^{k+1}} - \frac{10\pi - 3}{60}
\end{aligned}$$

Tomando el mismo punto de comienzo $x^0 = (0{,}2,\ 0{,}1,\ -0{,}1)^T$, se obtienen los valores mostrados en la Tabla 5.5. Claramente se observa la aceleración de la convergencia.

k	x_1	x_2	x_3
0	0.2	0.1	-0.1
1	0.49998333347222	0.02222979355858	-0.52304612619137
2	0.49997746826183	0.00002815366194	-0.52359807179342
3	0.49999999996378	0.00000003762202	-0.52359877465775
4	0.5	0.00000000005028	-0.52359877559704
5	0.5	0.00000000000007	-0.52359877559830
6	0.5	0.0	-0.52359877559830

Cuadro 5.5: Resultados al aplicar acelación de tipo Seidel.

5.8. Método Newton para sistemas de ecuaciones

Al igual que en el caso de una variable el método de punto fijo convergerá cuadráticamente si $g \in \mathcal{C}^2(X)$), y si la matriz Jacobiana es nula en el punto fijo ξ, es decir si todas las derivadas parciales $\partial g_j/\partial x_i$ evaluadas en el punto fijo son cero. Asimismo, el método de Newton para sistemas de ecuaciones no lineales $f(x) = 0$, con $f : X \to \mathbb{R}^n$, se puede constuir en forma análoga a como lo hicmos en el caso escalar. En este caso hacemos la elección para la función de iteración en la forma

$$g(x) = x - A(x)f(x)\,,$$

donde la matriz $A(x) \in \mathbb{R}^{n\times n}$ depende de $x \in \mathbb{R}^n$. El sistema de ecuaciones $f(x) = 0$ y el sistema de punto fijo $x = g(x)$ tienen la misma solución ξ si la matriz $A(x)$ es no-singular para toda x del dominio de f, dado que en este caso

$$x = g(x) \iff A(x)f(x) = 0 \iff f(x) = 0\,.$$

Además, para que el método sea de segundo orden, se debe satisfacer

$$0 = Jg(\xi) = I - A'(\xi)\, f(\xi) - A(\xi)\, Jf(\xi) = I - A(\xi)\, Jf(\xi)\,.$$

Para que esto suceda basta pedir $A(x) = [Jf(x)]^{-1},\ \forall\, x$. Con esta elección se tiene

$$g(x) = x - [Jf(x)]^{-1} f(x)\,,$$

donde $Jf(x)$ debe ser invertible para toda x cerca de la raíz ξ (al menos). Por lo tanto, el método de Newton es:

Dado x^0 cerca de la solución ξ, generar la sucesión $\{x^k\}_{k=1}^{\infty}$ por medio de

$$x^{k+1} = x^k - [Jf(x^k)]^{-1} f(x^k)\,.$$

Para evadir el costo excesivo la evaluación de la inversa de $Jf(x^k)$ en cada iteración, podemos expresar la iteración en la forma

$$x^{k+1} = x^k + y\,,$$

donde $y \in \mathbb{R}^n$ resuelve el sistema lineal

$$Jf(x^k)y = -f(x^k)\,.$$

Por lo tanto el mayor costo del Método de Newton consiste resolver un sistema lineal de n ecuaciones con n incógnitas en cada iteración. Podemos utilizar el método de factorización LU para resolver estos sistemas lineales.

Si cada una de las funciones $f_i(x),\ i = 1,\ 2, \ldots, n$ tienen segundas derivadas continuas y además $Jf(x)$ es no singular $\forall\, x \in X$, entonces el método converge cuadráticamente si el punto inicial x^0 se escoge suficientemente cercano a la raíz ξ.

Ejemplo 5.16. *Consideremos el sistema no lineal de el ejemplo anterior. La matriz Jacobiana del sistema $f(x) = 0$ es*

$$Jg(x) = \begin{bmatrix} 3 & x_3\ \text{sen}(x_2x_3) & x_2\ \text{sen}(x_2x_3) \\ 2x_1 & -162(x_2 + 0{,}1) & \cos x_3 \\ -x_2e^{-x_1x_2} & -x_1e^{-x_1x_2} & 20 \end{bmatrix} \tag{5.1}$$

Tomando $x^0 = (0{,}2,\ 0,1,\ -0{,}1)^T$, como en los ejemplos anteriores, obtenemos los resultados mostrados en la Tabla 5.6. Se observa claramente la convergencia cuadrática del método.

k	x_1	x_2	x_3
0	0.2	0.1	-0.1
1	0.49986882803679	0.02161464530261	-0.52190738633341
2	0.50001730001008	0.00192386026501	-0.52354815221340
3	0.50000016585639	0.00001819182542	-0.52359829975132
4	0.50000000001512	0.00000000165766	-0.52359877555494
5	0.5	0.0	-0.52359877559830

Cuadro 5.6: Resultados obtenidos con el método de Newton.

Capítulo 6

Interpolación Polinomial e Integración Númerica

Los polinomios son utilizados como el medio básico de aproximación de funciones en muchas áreas como son:

a) Aproximación de derivadas e integrales,

b) Solución numérica de ecuaciones diferenciales e integrales,

c) Solución de ecuaciones no lineales,

entre otras, debido a que tienen una estructura muy simple. *La interpolación polinómial* es un área particular de la *teoría de aproximación* de funciones y, de hecho, es la técnica más usada para aproximar funciones, más ampliamente incluso que los métodos de mínimos cuadrados, funciones racionales, funciones trigonométricas y "splines". El *teorema de aproximación de Weierstrass* garantiza la existencia de un polinomio que aproxima a una función continua dada con la precisión deseada:

Teorema 6.1. Teorema de aproximación de Weierstrass. *Dada* $f \in \mathcal{C}[a,b]$*, para toda* $\varepsilon > 0$*, existe un polinomio* p_ε *tal que* $|f(x) - p_\varepsilon(x)| < \varepsilon, \quad \forall\, x \in [a,b]$.

Geométricamente, el teorema de aproximación de Weierstrass establece que para cualquier $\varepsilon > 0$ podemos encontrar una $franja$ con grosor 2ε alrededor de $f(x)$ sobre $[a,b]$ dentro de la cual se encuentra un polinomio P_ε, sin importar que tan pequeña sea ε. Sin embargo, este teorema es útil solo desde el punto de vista teórico, pero no lo es desde el punto de vista computacional, pues no ofrece un método constructivo para calcular dicho polinomio. Por

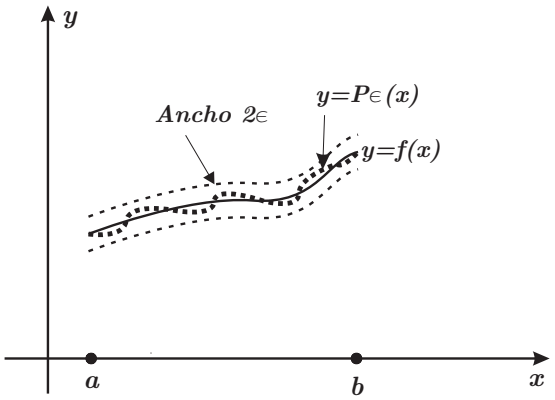


Figura 6.1: Teorema de aproximación de Weierstrass

tal motivo es necesario recurrir a otras herramientas para contruir polinomios que aproximen a las funciones.

6.1. Polinomio de Taylor

Una forma de aproximar una función "*suave*" localmente y cerca de un punto dado x_0, es por medio del *polinomio de Taylor*. Sea I_{x_0} un intervalo que contiene x_0. Entonces

Dada la función $f \in \mathcal{C}^{n+1}(I_{x_0})$, donde I_{x_0} es un intervalo que contiene x_0, el polinomio de Taylor de grado n que aproxima $f(x)$ en I_{x_0} es

$$p_n(x) = f(x_0) + f'(x_0)(x - x_0) + \ldots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \,.$$

El término de error o término complementario de la aproximación es

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

con ξ entre x y x_0. Este error aumenta conforme x se aleja del punto x_0, y también si la derivada $f^{n+1}(\xi)$ es muy grande.

Ejemplo 6.2. *El polinomio de Taylor de $f(x) = e^x$ cerca de $x_0 = 0$ es*

$$p_n(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!} \,,$$

y el error es

$$e_n(x) = \frac{e^\xi}{(n+1)!}\, x^{n+1} \,, \quad \text{con } \xi \text{ entre } 0 \text{ y } x.$$

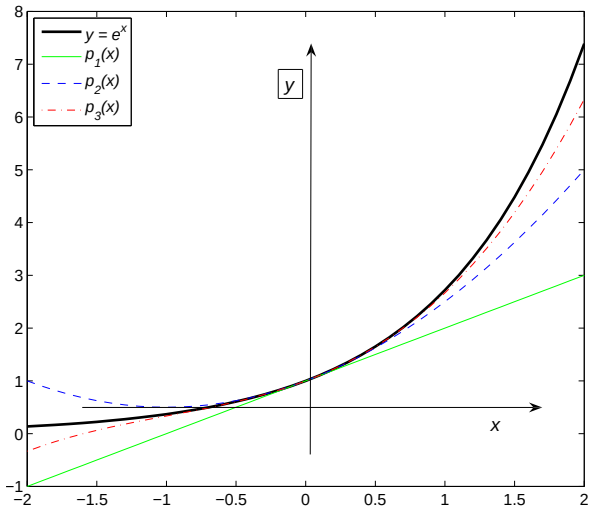


Figura 6.2: Polinomios de Taylor de orden 1, 2 y 3 para la función exponencial

La Figura 6.2 muestra la gráfica de la función exponencial junto con la gráfica de los polinomios de Taylor de grado 1, 2 y 3 en el intervalo $[-1, 1]$. Se observa que al aumentar el grado, la aproximación mejora, mientras que al aumentar $|x|$ la aproximación se deteriora en cada caso, como era de esperarse.

6.2. Interpolación de Lagrange

La forma más usual de aproximar una función dada mediante un polinomio de interpolación es por medio del denominado *polinomio de interpolación de Lagrange*. Este polinomio se obtiene resolviendo el siguiente problema:

Dados $n+1$ *puntos distintos* $x_0,\ x_1, \ldots,\ x_n$ *y los valores* $f(x_0),\ f(x_1),\ \ldots,\ f(x_n)$ *de una función continua* $f(x)$ *definida en un intervalo* $[a, b]$, *encontrar un polinomio* $p_n(x)$ *de grado* $\leq n$ *tal que*

$$p_n(x_j) = f(x_j)\,, \quad j = 0,\ 1, \ldots,\ n\,.$$

Debido a que se deben satisfacer $n+1$ condiciones y se deben calcular $n+1$ coeficientes $c_0,\ c_1, \ldots,\ c_n$ que definen al polinomio

$$p_n(x) = c_0 + c_1 x + \ldots + c_n x^n\,,$$

debemos esperar que el problema tenga una solución única. Anteriormente hemos demostrado que, efectivamente, este polinomio es único si los puntos $x_j,\ j = 0,\ 1, \ldots,\ n$, son distintos,

dado que en este caso la matriz de Vandermonde es no–singular. En realidad, la existencia y unicidad no depende de la estructura de la matriz de Vandermonde, y podemos demostrar que el polinomio existe y es único por otros medios. Por ejemplo, la unicidad puede demostrarse como sigue:

Supongase que $p_n(x)$ y $q_n(x)$ son dos polinomios de grado $\leq n$ que interpolan a $f(x)$ en $x_j,\ j = 0, \ldots,\ n$, es decir

$$p_n(x_j) = q_n(x_j) = f(x_j)\,, \quad j = 0,\ 1, \ldots,\ n\,.$$

Entonces, $r_n(x) = p_n(x) - q_n(x)$ es un polinomio de grado $\leq n$ con $n+1$ raices $x_0,\ x_1, \ldots,\ x_n$. Pero cualquier polinomio de grado n con un número de raices mayor a n debe ser constante e igual a cero. Por lo tanto $r_n(x) \equiv 0,\ \forall\, x$, y en consecuencia $p_n(x) = q_n(x),\ \forall\, x \in [a, b]$.

Para probar la existencia de tal polinomio basta con construirlo. Para lograr esto primero construirmos los $n + 1$ polinomios de Lagrange $\ell_j(x),\ j = 0,\ 1, \ldots,\ n$, de grado $\leq n$. Cada uno de estos polinomios esta asociado a un nodo. Por ejemplo, el polinomio $\ell(x)$ tiene la propiedad de que $\ell_j(x_j) = 1$ y $\ell_j(x_j) = 0$ si $i \neq j$, y se puede representar como el producto

$$\ell_j(x) = \prod_{k=0,\, k\neq j}^{n} \frac{(x - x_k)}{(x_j - x_k)}$$

Esta claro que se satisface $\ell_j(x_i) = \delta_{ij}$. La Figura 6.3 muestra la gráfica de este polinomio cerca de su punto correspondiente x_j.

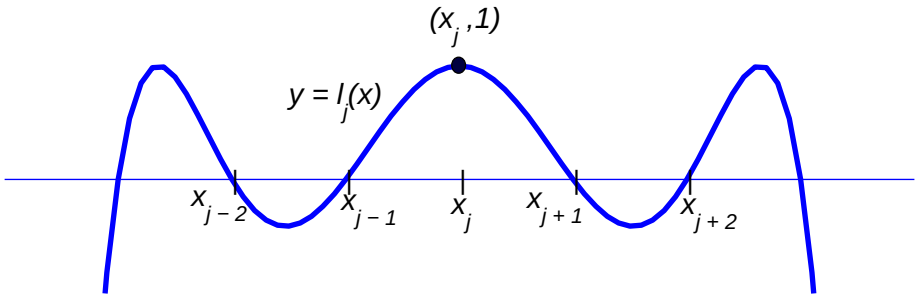


Figura 6.3: Polinomio básico de Lagrange asociado al punto x_j

Una vez que hemos construido los polinomios de Lagrange, se procede a construir el polinomio de interpolación como una combinación de los $n + 1$ polinomios de Lagrange

como se indica a continuación.

$$p_n(x) = \sum_{j=0}^{n} f(x_j)\ell_j(x).$$

Claramente este polinomio es de grado $\leq n$ y satisface

$$p_n(x_i) = \sum_{j=0}^{n} f(x_j)\ell_j(x_i) = \sum_{j=0}^{n} f(x_j)\delta_{ij} = f(x_i)\,, \quad i = 1, \ldots,\ n\,.$$

Aunque el polinomio de interpolación es único, hay varias formas de representarlo. Una de ellas es la anterior, y se le conoce como la *forma de Lagrange del polinomio de interpolación.*

Ejemplo 6.3. *Dado la función $f(x)$ definida sobre el intervalo $[a, b]$. El polinomio $p_1(x)$ de grado ≤ 1 que interpola $f(x)$ en dos puntos distintos x_0 y x_1 del intervalo es*

$$p_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)\,.$$

Este polinomio tiene como gráfica una recta. La Figura 6.4 ilustra esta situación.

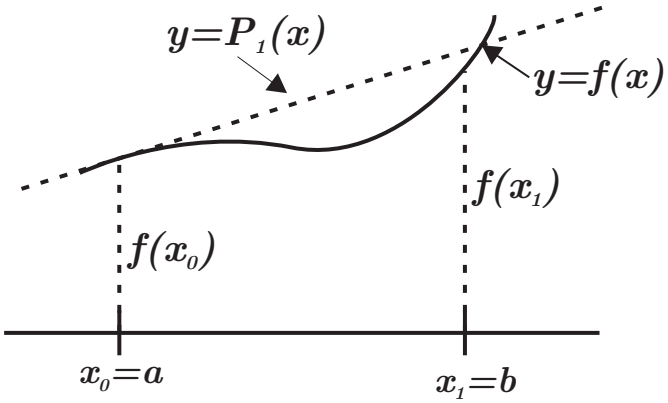


Figura 6.4: Polinomio de interpolación de Lagrange de grado 1.

El polinomio $p_2(x)$ de grado ≤ 2 y que interpola $f(x)$ tres puntos distintos x_0, x_1 y $x_2 = b$ del intervalo es

$$\begin{aligned} p_2(x) =& \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) \\ &+ \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2), \end{aligned}$$

el tiene como gráfica una parábola como se muestra el la Figura 6.5

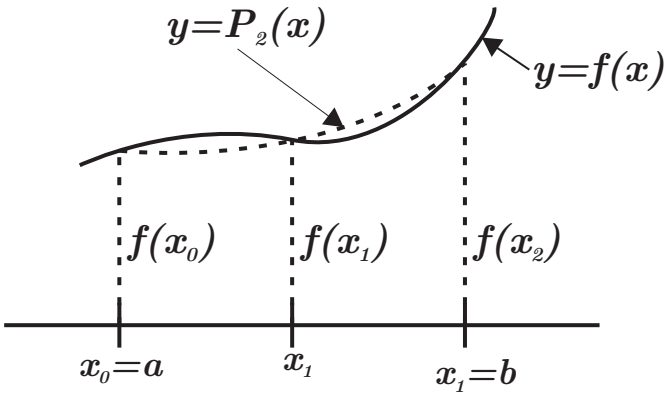


Figura 6.5: Polinomio de interpolación de Lagrange de grado 2.

6.3. Error en el polinomio de interpolación

La función $f(x)$ y su polinomio de interpolación coinciden en los puntos de interpolación. Sin embargo, también queremos el polinomio de interpolación para aproximar la función en puntos distintos a puntos los de interpolación, asi como estimar la diferencia $e_n(x) \equiv f(x) - p_n(x)$ en cualquier $x \in [a, b]$. Sin hipótesis adicionales para $f(x)$ no podemos decir gran cosa sobre el error $e_n(x)$. Sin embargo, si suponemos que $f(x)$ es suficientemente suave, es decir si la función tiene suficientes derivadas continuas, podemos establecer una estimación del error.

Teorema 6.4. *Si $f \in \mathcal{C}^{n+1}[a, b]$ y $p_n(x)$ es el polinomio de interpolación $n+1$ puntos distintos $x_0 = a,\ x_1, \ldots,\ x_n = b$, entonces para cada $x \in [a, b]$ existe $\xi(x) \in I[x_0, x_1, \ldots, x_n, x]$ (el intervalo cerrado más pequeño que contiene $x_0, x_1, \ldots, x_n, x$) tal que*

$$f(x) - p_n(x) \quad = \quad \frac{f^{n+1}(\xi(x))}{(n+1)!} w(x) \qquad con \quad w(x) = \prod_{k=0}^{n} (x - x_k).$$

Demostración. Si $x = x_i$ para algún $0 \leq i \leq n$, la igualdad se satisface trivialmente pues ambos lados de son iguales a cero. Así que supongase que $x \neq x_i,\ i = 0,\ 1, \ldots,\ n$, y sea

$$F(t) = f(t) - p_n(t) - \frac{f(x) - p_n(x)}{w(x)} w(t)\,, \quad t \in [a, b]\,.$$

Claramente $F(t)$ esta bién definida pues $w(x) \neq 0$ ya que $x \neq x_i,\ \forall\, i$. Además $F(t)$ es de clase $\mathcal{C}^{n+1}[a, b]$ y tiene al menos $n + 2$, a saber x_0, x_1, ..., x_n, x. Luego $F'(t)$ tiene al menos $n + 1$ ceros, $F''(t)$ tiene al menos n ceros , ..., y $F^{(n+1)}(t)$ tiene al menos un cero en $[a, b]$

que llamaremos $\xi(x)$. Por lo tanto

$$0 = F^{n+1}(\xi(x)) = f^{n+1}(\xi(x)) - 0 - \frac{f(x) - p_n(x)}{w(x)}(n+1)!\,.$$

Se concluye que

$$f(x) - p_n(x) = \frac{f^{n+1}(\xi(x))}{(n+1)!}w(x).$$

□

Ejemplo 6.5. *Cuando se hace interpolación lineal en los puntos x_0, x_1 la fórmula del error es*

$$f(x) - p_1(x) = \frac{f''(\xi(x))}{2}(x - x_0)(x - x_1) \text{ con } \xi(x) \in [x_0,\, x_1]\,.$$

Supopiendo $f \in \mathcal{C}^2[a,b]$, sea $M_2 = \text{máx}_{a\le x\le b}\,|f''(x)|$. El punto x^* donde $|w(x)| = |(x - x_0)(x - x_1)|$ toma su valor máximo es $x^* = \dfrac{1}{2}(x_0 + x_1)$ (ver Figura 6.6). Por lo tanto

$$\max_{a\le x\le b}|w(x)| = |w(x^*)| = \frac{h^2}{4}$$

donde $h = x_1 - x_0$.

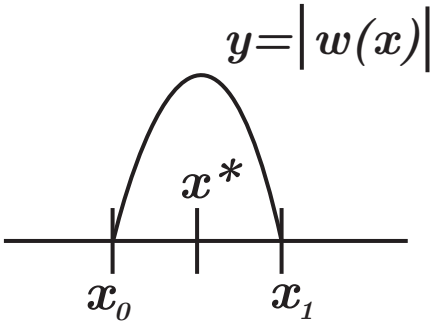


Figura 6.6: El punto máximo de $|w(x)|$

Utilizando estas expresiones se encuentra la siguiente cota para el error

$$\max_{a\le x\le b}|f(x) - p_1(x)| \le \frac{M_2}{2}\frac{h^2}{4} = \frac{M_2}{8}h^2\,.$$

es decir

$$\|f(x) - p_1(x)\|_\infty \le \frac{M_2}{8}h^2\,.$$

Se puede hacer el mismo cálculo en el caso de interpolación cuadrática en $x_0,\ x_1,\ x_2$ puntos igualmente espaciados, es decir $x_0,\ x_1 = x_0 + h,\ x_2 = x_0 + 2h$. En este caso se obtiene

$$f(x) - p_2(x) = \frac{f^{(3)}(\xi(x))}{3!}(x - x_0)(x - x_1)(x - x_2) \text{ con } \xi(x) \in (x_0, x_0 + 2h)\,,$$

es decir

$$\|f(x) - p_2(x)\|_\infty \leq \frac{M_3}{6}\|w\|_\infty = \frac{M_3}{9\sqrt{3}}h^3\,,$$

donde $M_3 = \text{máx}_{a\leq x\leq b}\,|f^{(3)}(x)|$ y $\|w\|_\infty = \dfrac{2}{3\sqrt{3}}h^3$.

Ejemplo 6.6. *Sea*

$$K(y) = \int_0^{\frac{\pi}{2}} \frac{dx}{\sqrt{1 - \text{sen}^2\, y - \text{sen}^2\, x}}\,.$$

Para ciertos valores de y medidos en grados se tiene $K(1) = 1{,}5709,\ K(4) = 1{,}5727,\ K(b) = 1{,}5751$. *Calcular* $K(3{,}5)$ *usando interpolación polinomial cuadrática.*

Solución. Los puntos de interpolación son $y_0 = 1,\ y_1 = 4,\ y_2 = 6$. El polinomio de Lagrange es

$$\begin{aligned} p_2(y) &= K(y_0)\frac{(y - y_1)(y - y_2)}{(y_0 - y_1)(y_0 - y_2)} + K(y_1)\frac{(y - y_0)(y - y_2)}{(y_1 - y_0)(y_1 - y_2)} \\ &\quad + K(y_2)\frac{(y - y_0)(y - y_1)}{(y_2 - y_0)(y_2 - y_1)} \end{aligned}$$

Por lo tanto

$$\begin{aligned} p_2(3{,}5) &= 1{,}5703\frac{(3{,}5 - 4)(3{,}5 - 6)}{(1 - 4)(1 - 6)} + 1{,}5727\frac{(3{,}5 - 1)(3{,}5 - 6)}{(4 - 1)(4 - 6)} \\ &\quad + 1{,}5751\frac{(3{,}5 - 1)(3{,}5 - 4)}{(6 - 1)(6 - 4)} \\ &= 1{,}57225 \end{aligned}$$

El número de operaciones para construir y evaluar el polinomio de interpolación en forma de Lagrange es:

a) Para evaluar $\ell_j(x) = \prod_{k=0,\ k\neq j}^{n} \dfrac{(x - x_k)}{(x_j - x_k)}$ se necesitan $2(n-1)$ multiplicaciones, 1 división, y $2n$ restas, para cada j. En total se necesitan $[2(n-1)+1](n+1) = (2n-1)(n+1)$ multiplicaciones/divisiones y $2n(n+1)$ restas.

b) Para evaluar $p_n(x) = \sum_{j=0}^{n} f(x_j)\ell_j(x)$ se requieren $n+1$ multiplicaciones, y n sumas.

El número total de operaciones es: $2n(n+1)$ multiplicaciones/divisiones y $n(2n+3)$ sumas/restas.

6.4. Forma de Newton del polinomio de interpolación

En ocasiones uno no sabe a ciencia cierta cuantos puntos de interpolación usar. Dada esta circunstancia sería deseable calcular los polinomios de interpolación $p_0(x),\ p_1(x),\ p_2(x), \ldots$ en forma sucesiva aumentando el número de puntos de interpolación hasta obtener una aproximación $p_n(x)$ satisfactoria para la función $f(x)$. En un proceso de este tipo el uso de la forma de Lagrange no es adecuada pues no es posible utilizar ninguna ventaja obvia del polinomio anterior p_{k-1} para construir el siguiente $p_k(x)$. Para este y otros propósitos, la forma de Newton del polinomio de interpolación es mejor. En esta forma, dada la función $f(x)$ y los puntos de interpolación $x_0,\ x_1, \ldots,\ x_n$, el polinomio de interpolación $p_n(x)$ se expresa utilizando los primeros n puntos de interpolación $x_0,\ x_1, \ldots,\ x_{n-1}$ como centros para escribir

$$\begin{aligned} p_n(x) = a_0 + a_1(x-x_0) &+ a_2(x-x_0)(x-x_1) + \ldots \\ &+ a_n(x-x_0)(x-x_1)\ldots(x-x_{n-1}) \end{aligned}$$

es decir

$$p_n(x) = a_0 + \sum_{j=1}^{n} a_j \prod_{k=0}^{j-1}(x-x_k)\,,$$

donde $a_0,\ a_1, \ldots,\ a_n$ son coeficientes a determinar. Antes de abordar el cálculo de estos coeficientes estudiaremos algunas propiedades de esta forma de representar el polinomio de interpolación. Estas propiedades nos permitirán verificar que esta representación permite la construcción de un polinomio de interpolacion de grado k a partir del de grado menor $k-1$.

Sea $q_k(x)$ la suma de los primeros $k+1$ términos de $p_n(x)$, es decir

$$\begin{aligned} q_k(x) = a_0 + a_1(x-x_0) &+ a_2(x-x_0)(x-x_1) + \ldots \\ &+ a_k(x-x_0)(x-x_1)\ldots(x-x_{k-1}) \end{aligned}$$

Entonces los términos restantes de $p_n(x)$ tienen como factor común el producto

$$(x-x_0)(x-x_1)\ldots(x-x_k)\,.$$

Así que

$$p_n(x) = q_k(x) + (x-x_0)(x-x_1)\ldots(x-x_k)r(x)$$

donde $r(x)$ es un polinomio de grado $\leq n-(k+1)$. Además $q_k(x)$ interpola $f(x)$ en los puntos $x_0,\ x_1,\ldots,\ x_k$, pues

$$\begin{aligned} q_k(x_j) &= p_n(x_j) - (x_j - x_0)(x_j - x_1)\ldots(x_j - x_k)r(x_j) \\ &= p_n(x_j) \quad \text{si } 0 \leq j \leq k \\ &= f(x_j) \end{aligned}$$

Entonces $q_k(x)$ es el único polinomio de interpolación $p_k(x)$ para $f(x)$ en $x_0,\ x_1,\ldots,\ x_k$, y podemos escribir

$$p_k(x) = q_k(x) + (x - x_0)(x - x_1)\ldots(x - x_k)\, r(x),$$

y con $k = n-1$, obtenemos

$$p_n(x) = p_{n-1}(x) + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

Entonces, el polinomio de interpolación $p_n(x)$ puede construirse paso a paso construyendo la sucesión de polinomios de interpolación $p_0(x)$, $p_1(x)$,..., $p_n(x)$, donde $p_k(x)$ se construye de $p_{k-1}(x)$ agregando el siguiente término en la forma de Newton , el cual es

$$a_k(x - x_0)(x - x_1)\ldots(x - x_{k-1}).$$

6.4.1. Cálculo de los coeficientes $a_0,\ a_1,\ldots,\ a_n$

Observese que cada coeficiente a_k es el *coeficiente principal* del polinomio $p_k(x)$ que interpola a $f(x)$ en los puntos $x_0,\ x_1,\ldots,\ x_k$. Además este coeficiente depende de los puntos y los valores de $f(x)$ en estos puntos pues

$$\begin{aligned} f(x_0) &= p_0(x_0) = a_0 \\ f(x_1) &= p_1(x_1) = f(x_0) + a_1(x_1 - x_0) \\ &\Rightarrow \quad a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \\ f(x_2) &= p_2(x_2) = p_1(x_2) + a_2(x_2 - x_0)(x_2 - x_1) \\ &\Rightarrow \quad a_2 = \frac{f(x_2) - p_1(x_2)}{(x_2 - x_0)(x_2 - x_1)} \\ &\vdots \\ f(x_k) &= p_k(x_k) = p_{k-1}(x_k) + a_k(x_k - x_0)(x_k - x_1)\ldots(x_k - x_{k-1}) \\ &\Rightarrow \quad a_k = \frac{f(x_k) - p_{k-1}(x_k)}{\prod_{j=0}^{k-1}(x_k - x_j)} \end{aligned}$$

Para indicar la dependencia de a_k de $x_0,\ x_1, \ldots,\ x_k$ y $f(x_0),\ f(x_1), \ldots,\ f(x_k)$ escribimos

$$a_k := f[x_0, x_1, \ldots, x_k]\,.$$

Esta expresión se denomina la k–ésima diferencia dividida de $f(x)$ en los puntos $x_0,\ x_1, \ldots,\ x_k$, y es igual al coeficiente principal del polinomio que interpola $f(x)$ en dichos puntos.

Podemos encontrar una forma más conveniente de calcular las diferencias divididas, en lugar del método recursivo anterior. Sabemos que

$$\begin{aligned} a_0 &= f(x_0) && \equiv f[x_0] \\ a_1 &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} && \equiv f[x_0, x_1] \end{aligned}$$

Las diferencias de orden mayor pueden construirse utilizando la fórmula

$$a_k = \frac{f[x_1, \ldots, x_k] - f[x_0, \ldots, x_{k-1}]}{(x_k - x_0)} \equiv f[x_0, x_1, \ldots, x_k]$$

La validez de esta última fórmula viene dada por el siguiente argumento debido a Neville:

Supóngase que la fórmula se ha obtenido para diferencias divididas hasta el orden $k-1$, y considerense los puntos $x_0,\ x_1, \ldots,\ x_k$. Sea $p_{k-1}(x)$ el polinomio de grado $\leq k-1$ que interpola $f(x)$ en $x_0,\ x_1, \ldots,\ x_{k-1}$, y sea $q_{k-1}(x)$ el polinomio de grado $\leq k-1$ que interpola $f(x)$ en $x_1,\ x_2, \ldots,\ x_k$ como se muestra en la Figura 6.7.

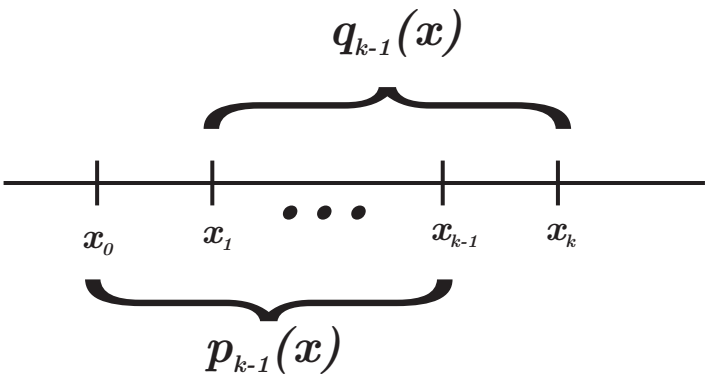


Figura 6.7: Polinomios en el argumento de Neville.

Luego, el polinomio

$$p(x) = \frac{x - x_0}{x_k - x_0} q_{k-1}(x) + \frac{x_k - x}{x_k - x_0} p_{k-1}(x),$$

es un polinomio de grado $\leq k$. Además se puede verificar directamente que este polinomio interpola $f(x)$ en los puntos $x_0,\ x_1, \ldots,\ x_k$. Por unicidad en el polinomio de interpolación, $p_k(x) = p(x)$, y en consecuencia

$$p_k(x) = \frac{x - x_0}{x_k - x_0} q_{k-1}(x) - \frac{x - x_k}{x_k - x_0} p_{k-1}(x).$$

Claramente

$$\text{coef. principal de } p_k = \frac{\text{coef. principal de } q_{k-1}}{x_k - x_0} - \frac{\text{coef. principal de } p_{k-1}}{x_k - x_0}$$

Es decir

$$f[x_0, x_1, \ldots, x_k] = \frac{f[x_1, \ldots, x_k] - f[x_0, \ldots, x_{k-1}]}{(x_k - x_0)}$$

Resumiendo, la **forma de Newton del polinomio de interpolación** en los puntos $x_0,\ x_1, \ldots,\ x_n$ para $f(x)$ es

$$\begin{aligned} p_n(x) &= f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \ldots \\ &\qquad + f[x_0, x_1, \ldots, x_n](x - x_0) \ldots (x - x_{n-1}) \\ &= f(x_0) + \sum_{k=1}^{n} f[x_0, \ldots, x_k] \prod_{j=0}^{k-1} (x - x_j) \end{aligned}$$

donde

$$f[x_0, x_1, \ldots, x_k] = \frac{f[x_1, \ldots, x_k] - f[x_0, \ldots, x_{k-1}]}{(x_k - x_0)}, \quad k = 1,\ 2, \ldots,\ n\,.$$

Esta última expresión se utiliza para generar las diferencias divididas en forma simple por medio de una tabla denominada "**tabla de diferencias divididas**". Por ejemplo, para el caso $n = 3$, esta tabla queda de la siguiente manera

x	$f[\cdot]$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot]$
x_0	$f(x_0)$			
		$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$		
x_1	$f(x_1)$		$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$	
		$f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$		$f[x_0, x_1, x_2, x_3]$
x_2	$f(x_2)$		$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}$	
		$f[x_2, x_3] = \frac{f(x_3) - f(x_2)}{x_3 - x_2}$		
x_3	$f(x_3)$			

donde

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}$$

Ejemplo 6.7. *Consideremos de nuevo el ejemplo anterior, donde* $K(1) = 1{,}5709$, $K(4) = 1{,}5727$ *y* $K(6) = 1{,}5751$. *Calcular* $K(3{,}5)$ *con interpolación cuadrática utilizando la forma de Newton.*

Solución. Primero calculamos la tabla de diferencias divididas:

y	$K[\cdot]$	$K[\cdot,\cdot]$	$K[\cdot,\cdot,\cdot]$
1	**1.5709**		
		$\frac{1{,}5727-1{,}5709}{4-1} = \mathbf{0{,}0006}$	
4	1,5727		$\frac{0{,}0012-0{,}0006}{6-1} = \mathbf{0.00012}$
		$\frac{1{,}5751-1{,}5727}{6-4} = 0{,}0012$	
6	1,5751		

Después, tomamos los coeficientes en la diagonal superior (en negrita) para construir el polinomio de interpolación:

$$\begin{aligned} p_2(y) &= K(1) + K[1,4](y-1) + K[1,4,6](y-1)(y-4) \\ &= 1{,}5709 + 0{,}0006(y-1) + 0{,}00012(y-1)(y-4), \end{aligned}$$

y evaluamos directamente en $y = 3{,}5$:

$$p_2(3{,}5) = 1{,}5709 + 0{,}0006(3{,}5-1) + 0{,}00012(3{,}5-1)(3{,}5-4) = 1{,}57225$$

Este resultado coincide con el encontrado con el polinomio de interpolación en forma de Lagrange.

6.4.2. Número de operaciones en la forma de Newton

Para hacer interpolación en la forma de Newton se requieren

a) 2 restas y 1 división para evaluar cada una de las $n(n+1)/2$ diferencias divididas. Es decir, se requieren $n(n+1)$ restas y $n(n+1)/2$ disiones.

b) $n(n+1)/2 + n$ sumas y $n(n+1)/2$ multiplicaciones para evaluar el polinomio .

Luego, el número total de operaciones es

$$\begin{aligned} n(n+1) + \frac{n(n+1)}{2} + n &= n[\frac{3(n+1)}{2} + 1] \text{ sumas/restas} \\ \frac{n(n+1)}{2} + \frac{n(n+1)}{2} &= n(n+1) \text{ multiplicaciones/divisiones} \end{aligned}$$

El número de operaciones para evaluar el polinomio puede reducirse expresando el polinomio de interpolación de Newton en forma anidada:

$$\begin{aligned}
p_n(x) &= a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-x_2) \\
&\qquad + \ldots + a_n(x-x_0)(x-x_1)\ldots(x-x_{n-1}) \\
&= a_0 + (x-x_0)[a_1 + a_2(x-x_1) + a_3(x-x_1)(x-x_2) \\
&\qquad + \ldots + a_n(x-x_1)\ldots(x-x_{n-1})] \\
&= a_0 + (x-x_0)[a_1 + (x-x_1)[a_2 + a_3(x-x_2) \\
&\qquad + \ldots + a_n(x-x_2)\ldots(x-x_{n-1})]] \\
&\vdots \\
&= a_0 + (x-x_0)[a_1 + (x-x_1)[a_2 + (x-x_2)[a_3 \\
&\qquad + \ldots + (x-x_{n-2})[a_{n-1} + a_n(x-x_{n-1})]\ldots]]]
\end{aligned}$$

cuya evaluación toma n multiplicaciones y $2n$ sumas. Así que para calcular $p_n(x)$ y evaluarlo se requieren en total

$$\begin{aligned}
n(n+1) + 2n &= n(n+3) \text{ sumas y} \\
\frac{n(n+1)}{2} + n &= n(\frac{n+3}{2}) \text{ multiplicaciones}
\end{aligned}$$

El algoritmo para evaluar el polinomio en forma anidada se puede escribir de la siguiente manera:

Algoritmo para evaluar $p_n(z)$ $A_n := a_n$
Para $k = n-1,\ n-2, \ldots,\ 0$ hacer
$A_k := a_k + A_{k+1}(z - x_k)$
Fín

Entonces $p_n(z) = A_0$. Los coeficientes a_k son las correspondientes diferencias divididas $f[x_0, x_1, \ldots, x_k]$.

Ejemplo 6.8. *Con este algoritmo, la evaluación del polinomio en el ejemplo anterior se escribe*

$$\begin{aligned}
A_2 &= a_2 = 0{,}00012 \\
A_1 &= a_1 + A_2(3{,}5 - x_1) = 0{,}0006 + 0{,}00012(3{,}5-4) = 0{,}00054 \\
A_0 &= a_0 + A_1(3{,}5 - x_0) = 1{,}5709 + 0{,}00054(3{,}5-1) = 1{,}57225
\end{aligned}$$

Por lo tanto $p_2(3{,}5) = 1{,}57225$.

6.4.3. Interpolación en un número creciente de puntos

Si conocemos el polinomio de interpolación $p_n(x)$ de $f(x)$ en $x_0,\ x_1, \ldots,\ x_n$, y agregamos un punto adicional x_{n+1}, entonces, como ya hemos anteriormente, el polinomio $p_{n+1}(x)$ que interpola $f(x)$ en $x_0,\ x_1, \ldots,\ x_n,\ x_{n+1}$ es

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

Ejemplo 6.9. *Anteriormente hemos encontrado el polinomio de grado dos que interpola* $K(y)$ *en* $y = 1,\ 4,\ 6$. *Si agregamos el valor* $K[0] = 1{,}5708$, *obtenemos la tabla:*

y	$K[\cdot]$	$K[\cdot,\cdot]$	$K[\cdot,\cdot,\cdot]$	$K[\cdot,\cdot,\cdot,\cdot]$
1	**1.5709**			
		0,0006		
4	1,5727		**0.00012**	
		0,0012		**-0.000001**
6	1,5751		0,000121	
		0,000717		
0	1,5708			

Por lo tanto, el polinomio $p_3(y)$ que interpola $K(y)$ en $y = 1,\ 4,\ 6,\ 0$, es

$$p_3(y) = p_2(y) + K[1, 4, 6, 0](y - 1)(y - 4)(y - 6)\,,$$

y su valor en $y = 3{,}5$ es

$$\begin{aligned} p_3(3{,}5) &= p_2(3{,}5) + (-0{,}000001)(2{,}5)(-0{,}5)(-2{,}5) \\ &= 1{,}57225 - 0{,}000003125 \\ &= 1{,}5722469 \end{aligned}$$

6.4.4. El error del polinomio en forma de Newton

Sea $f(x)$ definida en $[a, b]$ y $p_n(x)$ el polinomio que interpola a $f(x)$ en los puntos $x_0,\ x_1, \ldots,\ x_n$. Consideremos $\overline{x}$ un punto en $[a, b]$ distinto de los puntos de interpolación

$x_i,\ i = 0,\ 1, \ldots,\ n$, y sea $p_{n+1}(x)$ el polinomio que interpola $f(x)$ en los puntos $x_0,\ x_1, \ldots,\ x_n,\ \overline{x}$, entonces

$$p_{n+1}(x) = p_n(x) + f[x_0, \ldots, x_n, \overline{x}]w(x),$$

donde $w(x) = (x - x_0) \ldots (x - x_n)$. Así que

$$f(\overline{x}) = p_{n+1}(\overline{x}) = p_n(\overline{x}) + f[x_0, \ldots, x_n, \overline{x}]w(\overline{x})$$

es decir

$$e_n(\overline{x}) = f(\overline{x}) - p_n(\overline{x}) = f[x_0, \ldots, x_n, \overline{x}]w(\overline{x})$$

En general, para toda x en $[a, b]$

$$e_n(x) = f(x) - p_n(x) = f[x_0, \ldots, x_n, x]w(x)\,.$$

Anteriormente encontramos, en el análisis del error del polinomio de la forma de Lagrange, que si $f \in \mathcal{C}^{n+1}[a, b]$, entonces

$$e_n(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}w(x),\ \text{con}\ \xi(x) \in I_{[x_0,\ldots,x_k]}$$

Comparando las dos últimas expresiones arriba, hemos encontrado que si $f \in \mathcal{C}^{n+1}[a, b]$, entonces

$$f[x_0, \ldots, x_n, x] = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\,.$$

En general, se tiene que si $f \in \mathcal{C}^{(k)}(I_{[x_0,\ldots,x_k]})$, entonces

$$f[x_0, \ldots, x_k] = \frac{f^{(k)}(\xi)}{(k)!}\text{con}\ \xi \in I_{[x_0,\ldots,x_k]}\,. \tag{6.1}$$

6.5. Integración Numérica

6.5.1. Fórmulas de Newton–Cotes

Nuestro objetivo es estimar la integral de una función $f(x)$ sobre un intervalo dado $[a,\ b]$, especialmente cuando esta integral no puede calcularse analíticamente o cuando solo se conocen valores de f en un número finito de puntos. Una de las estrategias para aproximar esta integral consiste en aproximar $f(x)$ por el polinomio $p_k(x)$ que interpola f en los puntos conocidos $x_0, \ldots, x_k$ del intervalo. Entonces

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

Si utilizamos la forma de Lagrange del polinomio de interpolación

$$p_k(x) = \sum_{j=0}^{k} f(x_j)\ell_j(x)$$

obtenemos

$$\int_a^b f(x)dx \approx \sum_{j=0}^{k} w_j f(x_j) \qquad \text{con} \quad w_j = \int_a^b \ell_j(x)dx.$$

Entonces, la integral de $f(x)$ sobre $[a, b]$ se aproxima por medio de un promedio pesado de valores de $f(x)$ en los puntos de interpolación, en donde los pesos w_j son las integrales de las funciones de Lagrange $\ell_j(x)$. A estas fórmulas de integración numérica se les conoce como de **Newton-Cotes**. Por supuesto, también podemos realizar la aproximación utilizando la forma de Newton del polinomio de interpolación

$$p_k(x) = f(x_0) + \sum_{j=1}^{k} f[x_0, \ldots, x_j] \prod_{l=0}^{j-1}(x - x_l)\,.$$

En este caso obtenemos

$$\int_a^b f(x)dx \approx (b-a)f(x_0) + \sum_{j=1}^{k} f[x_0, \ldots, x_j] \int_a^b \prod_{l=0}^{j-1}(x - x_l)dx\,.$$

6.5.2. El error en las fórmulas de integración de Newton-Cotes

Como vimos anteriormente, el error en el polinomio de interpolación en forma de Newton es para toda $x \in I[x_0, x_1, \ldots, x_k]$ es

$$e_k(x) = f(x) - p_k(x) = f[x_0, \ldots, x_k, x]w_{k+1}(x)\,, \quad \text{con } w_{k+1}(x) = \prod_{j=0}^{k}(x - x_j).$$

Luego, el ***error en la integración numérica*** es:

$$e = \int_a^b e_k(x)dx = \int_a^b f[x_0, \ldots, x_k, x]w_{k+1}(x)dx\,.$$

El cálculo de esta última integral puede ser complicado. Sin embargo, puede simplificarse si consideramos dos casos:

- $1^{\underline{er}}$ **caso**: Si $w_{k+1}(x)$ *es de un solo signo* en $[a, b]$, es decir si $\int_a^b w_{k+1}(x)dx > 0$ ó $\int_a^b w_{k+1}(x)dx < 0$, entonces

$$e = f[x_0, \ldots, x_k, \xi] \int_a^b w_{k+1}(x)dx \quad \text{con} \quad \xi \in (a, b).$$

- 2^{o} **caso**: Si $\int_a^b w_{k+1}(x)dx = 0$, entonces se puede introducir un punto adicional x_{k+1} para que $w_{k+2}(x) = w_{k+1}(x)(x - x_{k+1})$ sea de un solo signo en $[a,\, b]$. Si esto es posible, entonces

$$e = f[x_0, \ldots, x_{k+1}, \xi] \int_a^b w_{k+2}(x)dx$$

Demostración. $1^{\underline{er}}$ **caso**: $f[x_0, \ldots, x_k, x]$ es una función continua de x en $[a, b]$ y toma sus valores máximo M y mínimo m en el intervalo

$$m \leq f[x_0, \ldots, x_k, x] \leq M\,.$$

Suponiendo $\int_a^b w_{k+1}(x)dx > 0,\ \forall\, x \in [a, b]$, se obtiene

$$m \int_a^b w_{k+1}(x)dx \leq \int_a^b w_{k+1}(x) f[x_0, \ldots, x_k, x]dx \leq M \int_a^b w_{k+1}(x)dx\,.$$

Por lo tanto

$$m \leq \frac{\int_a^b w_{k+1}(x) f[x_0, \ldots, x_k, x]dx}{\int_a^b w_{k+1}(x)dx} \leq M\,.$$

Por el teorema del valor intermedio para funciones continuas, debe existir un ξ en $[a, b]$ tal que

$$f[x_0, \ldots, x_k, \xi] = \frac{\int_a^b w_{k+1}(x) f[x_0, \ldots, x_k, x]dx}{\int_a^b w_{k+1}(x)dx},$$

de donde se sigue la expresión para el error. Se obtiene la misma expresión suponiendo $\int_a^b w_{k+1}(x)dx < 0,\ \forall\, x \in [a, b]$.

2^{o} **caso**: Sea x_{k+1} un punto adicional a $x_0, \ldots, x_k$. Entonces

$$\begin{aligned} f[x_0, \ldots, x_k, x_{k+1}, x] &= f[x_{k+1}, x_0, \ldots, x_k, x] \\ &= \frac{f[x_0, \ldots, x_k, x] - f[x_{k+1}, x_0, \ldots, x_k]}{x - x_{k+1}} \end{aligned}$$

$$\Rightarrow f[x_0, \ldots, x_k,\, x] = f[x_{k+1}, x_0, \ldots, x_k] + f[x_0, \ldots, x_k, x_{k+1}, x](x - x_{k+1})\,.$$

Luego

$$\begin{aligned} e &= \int_a^b f[x_0, \ldots, x_k] w_{k+1}(x)dx \\ &= f[x_{k+1}, x_0, \ldots, x_k] \int_a^b w_{k+1}(x)dx + \int_a^b f[x_0, \ldots, x_k, x_{k+1}, x] w_{k+2}(x)dx\,. \end{aligned}$$

La primera integral en la última expresión es cero por hipótesis. Además, si se escoge x_{k+1} de tal manera que $w_{k+2}(x) = w_{k+1}(x)(x - x_{k+1})$ sea de un solo signo en $[a, b]$, entonces se obtiene la expresión para el error:

$$e = f[x_0, \ldots, x_k, x_{k+1}, \xi] \int_a^b w_{k+2}(x)dx$$

Nota. En caso de que la función $f(x)$ sea suficientemente suave, cada diferencia dividida es igual a una derivada del mismo orden dividida por un factorial. Tomando en cuenta esta propiedad, se obtienen las siguientes expresiones para el error en la integración numérica:

- Caso 1: Si $w_{k+1}(x)$ es de un solo signo en $[a, b]$, entonces

$$e = f[x_0, \ldots, x_k, \xi] \int_a^b w_{k+1}(x)dx = \frac{f^{(k+1)}(\eta)}{(k+1)!} \int_a^b w_{k+1}(x)dx\,.$$

- Caso 2: Si $\int_a^b w_{k+1}(x)dx = 0$, se introduce un punto adicional x_{k+1} tal que $w_{k+2}(x) = w_{k+1}(x)(x - x_{k+1})$ sea de un solo signo en $[a, b]$, y

$$e = f[x_0, \ldots, x_{k+1}, \xi] \int_a^b w_{k+2}(x)dx = \frac{f^{(k+2)}(\eta)}{(k+2)!} \int_a^b w_{k+2}(x)dx\,.$$

Observese que el error en este caso tiene un orden mayor que el del caso 1.

6.5.3. Fórmulas de integración numérica más comunes

Consideraremos las reglas de Newton–Cotes más utilizadas en la práctica:

1. **Regla del rectángulo**.

   Consideremos un solo punto de interpolación x_0, es decir $k = 0$. En este caso $f(x)$ se aproxima por el polinomio constante $p_0(x) = f(x_0)$, y la integral de $f(x)$ se aproxima por

$$\int_a^b f(x)dx \approx \int_a^b f(x_0)dx = (b - a)f(x_0)\,.$$

   Si se escoge $x_0 = a$, entonces

$$\int_a^b f(x)dx \approx (b - a)f(a)\,.$$

   Como $w_1(x) = (x - a)$ es de un solo signo en $[a, b]$ y el error en este caso es

$$e = f[x_0, \xi] \int_a^b w_1(x)dx = f[a, \xi] \int_a^b (x - a)dx = f[a, \xi]\frac{(b - a)^2}{2}\,,$$

con ξ en $[a,b]$. Suponiendo $f \in \mathcal{C}^1[a,b]$, podemos expresar el error en la forma

$$e = \frac{f'(\eta)}{2}(b-a)^2$$

2. **Regla del trapecio**.

Considermos dos puntos de interpolación x_0 y x_1, es decir $k=1$. En este caso $f(x)$ se aproxima por el polinomio lineal

$$p_1(x) = f(x_0) + f[x_0,x_1](x-x_0),$$

y la integral de $f(x)$ sobre el intervalo $[a,b]$ se aproxima por

$$\int_a^b \{f(x_0) + f[x_0,x_1](x-x_0)\}dx = f(x_0)(b-a) + \frac{f[x_0,x_1]}{2}\{(b-x_0)^2 - (a-x_0)^2\}\,.$$

Si se escoge $x_0 = a,\ x_1 = b$, la regla de cuadratura es

$$\int_a^b f(x)dx \approx \frac{b-a}{2}\{f(a)+f(b)\}\,.$$

Además $w_2(x) = (x-a)(x-b) \leq 0$ es de un solo signo en $[a,b]$ (ver Fig. 6.8). Luego

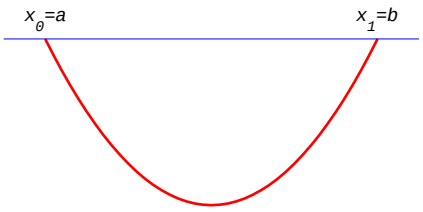


Figura 6.8: El polinomio cuadrático $w_2(x)$

el error en este caso se obtiene de

$$e = f[x_0,x_1,\xi]\int_a^b w_2(x)dx = f[a,b,\xi]\int_a^b (x-a)(x-b)dx = f[a,b,\xi]\frac{(a-b)^3}{6}\,,$$

con ξ en $[a,b]$. Suponiendo $f \in \mathcal{C}^2[a,b]$, entonces

$$e = -\frac{f^2(\eta)}{12}(b-a)^3\,, \quad \eta \text{ en } [a,b]\,.$$

3. **Regla de Simpson**.

   Consideremos tres puntos de interpolación x_0, x_1 y x_2, es decir $k = 2$. En este caso el polinomio de interpolación para $f(x)$ es

$$p_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)\,.$$

   Si elegimos $x_0 = a,\ x_1 = b,\ x_2 = \dfrac{a+b}{2}$, entonces la regla de cuadratura es

$$\begin{aligned}\int_a^b f(x)dx \approx \int_a^b p_2(x)dx &= f(a)(b-a) + f[a,b]\frac{(b-a)^2}{2} + f\left[a, b, \frac{a+b}{2}\right]\frac{(a-b)^3}{6} \\ &= \frac{b-a}{6}\left(f(a) + 4f(\frac{a+b}{2}) + f(b)\right)\end{aligned}$$

   En este caso se tiene que $w_3(x) = (x-a)(x-b)(x - \dfrac{a+b}{2})$, el cual satisface (ver Figura 6.9)

$$\int_a^b w_3(x)dx = 0\,.$$

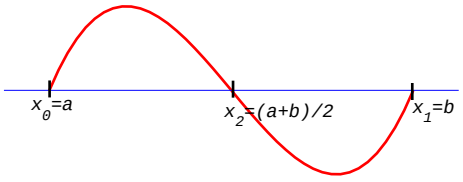


Figura 6.9: El polinomio cúbico $w_3(x)$

   Si introducimos $x_3 = \dfrac{a+b}{2}$ como punto adicional entonces

$$w_3(x) = (x-a)(x-b)(x - \frac{a+b}{2})^2 \leq 0,$$

   es de un solo signo en $[a, b]$. Por lo tanto

$$\begin{aligned}e &= f\left[a, b, \frac{a+b}{2}, \frac{a+b}{2}, \xi\right]\int_a^b (x-a)(x-b)\left(x - \frac{a+b}{2}\right)^2 dx \\ &= f\left[a, b, \frac{a+b}{2}, \frac{a+b}{2}, \xi\right]\left(-\frac{4}{15}\left(\frac{b-a}{2}\right)^5\right) = -\frac{f^4(\eta)}{90}\left(\frac{b-a}{2}\right)^5,\end{aligned}$$

   donde $\eta \in [a, b]$. En la última igualdad hemos supuesto que $f \in \mathcal{C}^4[a, b]$.

4. **Regla del punto medio**.

   En este caso volvemos a considerar un solo punto de interpolación, pero ahora tomamos el punto medio del intervalo: $x_0 = \dfrac{a+b}{2}$. Luego

$$f(x) \approx p_0(x) = f\left(\frac{a+b}{2}\right)$$

   y la regla de cuadratura es

$$\int_a^b f(x)dx \approx f\left(\frac{a+b}{2}\right)(b-a)\,.$$

   Además (ver Figura 6.10)

$$\int_a^b w_1(x)dx \approx \int_a^b (x - \frac{a+b}{2})dx = 0\,.$$

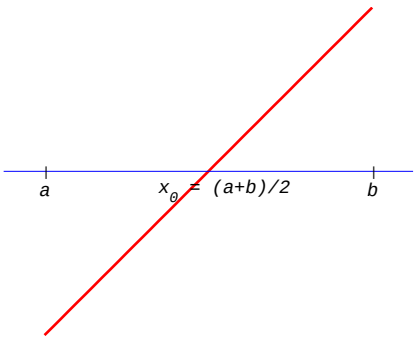


Figura 6.10: El polinomio lineal $w_1(x)$

   Si tomamos $x_1 = \dfrac{a+b}{2}$ como punto adicional, entonces la función $w_2(x) = (x - \dfrac{a+b}{2})^2$ es de un solo signo en $[a,b]$, y

$$e = f[x_0, x_1, \xi]\int_a^b w_2(x)dx = \frac{f^2(\eta)}{2}\int_a^b \left(x - \frac{a+b}{2}\right)^2 dx = \frac{f^2(\eta)}{24}(b-a)^3\,.$$

Ejemplo 6.10. *Consideremos la integral* $I = \int_0^1 e^{-x^2}\,dx$, *cuyo valor exacto a ocho cifras decimales es 0.74682413. Aplicando las reglas de integración anteriores, con* $a = 0$, $b = 1$, $\dfrac{a+b}{2} = 0{,}5$, $f(a) = 1$, $f(b) = e^{-1}$, $f(\dfrac{a+b}{2}) = e^{-1/4}$, *obtenemos*

REGLA DE CUADRATURA		RESULTADO
Rectángulo	$I \approx (b-a)f(a)$	$= \quad 1$
Punto medio	$I \approx (b-a)f(\frac{a+b}{2})$	$= \quad 0{,}7788007$
Trapecio	$I \approx \frac{(b-a)}{2}(f(a)+f(b))$	$= \quad 0{,}6839397$
Simpson	$I \approx \frac{(b-a)}{6}(f(a)+4f(\frac{a+b}{2})+f(b))$	$= \quad 0{,}7471804$

Los errores obtenidos con cada regla son

REGLA	ERROR REAL	ESTIMACION DEL ERROR
Rectángulo	0.25318	$\left\|\frac{f'(\xi)}{2}\right\|(b-a)^2 \leq 0{,}4288819$
Punto medio	0.03198	$\left\|\frac{f''(\xi)}{24}\right\|(b-a)^3 \leq 0{,}0833333$
Trapecio	0.06288	$\left\|-\frac{f''(\xi)}{12}\right\|(b-a)^3 \leq 0{,}166666$
Simpson	$3{,}6 \times 10^{-4}$	$\left\|-\frac{f^4(\xi)}{90}\right\|\left(\frac{b-a}{2}\right)^5 \leq 4{,}166610^{-3}$

6.6. Reglas compuestas de integración

Las estimaciones de $I = \int_a^b f(x)dx$ presentadas anteriormente no producen una buena aproximación, particularmente cuando el intervalo $[a, b]$ es grande. Introducir más puntos de interpolación para compensar el aumento en el tamaño del intervalo no es conveniente pues la acumulación del error debido a que el término $w_{k+1}(x) = \prod_{j=0}^{n}(x - x_j)$ es cada vez mayor conforme n aumenta. La alternativa práctica es dividir $[a, b]$ en un conjunto de subintervalos pequeños y aplicar las reglas simples de integración numérica en cada uno de estos subintervalos.

Sea $f(x)$ definida e integrable sobre $[a, b]$. Consideremos una subdivisión de $[a, b]$ en n subintervalos iguales de longitud $h = \dfrac{b-a}{n}$ como se ilustra en la Figura 6.11, y sean

$$x_i = a + ih, \ i = 0, \ 1, \ldots, \ n.$$

Figura 6.11: Subdivisión del intervalo en n subintervalos.

Esta claro que $x_0 = a,\ x_n = b$. Entonces

$$\int_a^b f(x)dx = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)dx\,,$$

y las integrales delante de la sumatoria se aproximan utilizando las reglas simples en cada subintervalo.

1. **Regla del trapecio compuesta**. Para cada $i = 1,\ 2, \ldots,\ n$, se tiene

$$\int_{x_{i-1}}^{x_i} f(x)dx \quad = \quad \frac{h}{2}\{f(x_{i-1}) + f(x_i)\} - \frac{f''(\xi_i)}{12}h^3\,,$$

para algún $\xi_i \in [x_{i-1},\, x_i]$. Entonces

$$\begin{aligned}\int_a^b f(x)dx &= \frac{h}{2}\sum_{i=1}^{n}[f(x_{i-1}) + f(x_i)] - \frac{h^3}{12}\sum_{i=1}^{n} f''(\xi_i) \\ &= \frac{h}{2}[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)] - \frac{h^3}{12}\sum_{i=1}^{n} f''(\xi_i)\end{aligned}$$

La última sumatoria se puede simplificar cuando $f \in \mathcal{C}^2[a, b]$, pues en este caso

$$m \leq f''(\xi_i) \leq M \quad \text{para cada } i = 1, \ldots, n\,,$$

en donde

$$m = \min_{a\leq x\leq b} f''(x) \quad \text{y} \quad M = \max_{a\leq x\leq b} f''(x)\,.$$

Así que

$$m \leq \frac{\sum_{i=1}^{n} f''(\xi_i)}{n} \leq M\,.$$

Por el teorema del valor intermedio aplicado a la función continua $f''(x)$ sobre $[a, b]$, existe ξ en $[a, b]$ tal que

$$f''(\xi) = \frac{\sum_{i=1}^{n} f''(\xi_i)}{n}$$

es decir

$$nf''(\xi) = \sum_{i=1}^{n} f''(\xi_i)$$

Por lo tanto, el término del error en la regla de cuadratura es

$$-\frac{h^3}{12} nf''(\xi) = -\frac{h^2}{12}(b-a)f''(\xi)$$

pues $nh = b - a$. Resumiendo, la regla del trapecio compuesta es

$$\int_a^b f(x)dx \approx \frac{h}{2}[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)]$$

y la expresión del error es

$$e = -\frac{b-a}{12} f''(\xi)h^2$$

2. **Regla de Simpson compuesta**

Sabemos que para cada $i = 1,\ 2, \ldots,\ n$

$$\int_{x_{i-1}}^{x_i} f(x)dx = \frac{h}{6}\left[f(x_{i-1}) + 4f(x_{i-\frac{1}{2}}) + f(x_i)\right] - \frac{f^4(\xi_i)}{90}\left(\frac{h}{2}\right)^5,$$

donde hemos usado la notación

$$x_{i-\frac{1}{2}} = \frac{x_{i-1} + x_i}{2}\,.$$

Sumando de $i = 1$ a $i = n$ obtenemos

$$\begin{aligned}\int_a^b f(x)dx &= \frac{h}{6}\sum_{i=1}^{n}[f(x_{i-1}) + 4f(x_{i-\frac{1}{2}}) + f(x_i)] - \frac{1}{90}\left(\frac{h}{2}\right)^5 \sum_{i=1}^{n} f^4(\xi_i) \\ &= \frac{h}{6}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + 4\sum_{i=1}^{n} f(x_{i-\frac{1}{2}}) + f(x_n)\right] \\ &\quad - \frac{b-a}{180} f^4(\xi_i)\left(\frac{h}{2}\right)^4,\end{aligned}$$

en donde en el término del error hemos utilizado un argumento similar al utilizado en la regla del trapecio:

$$\text{Suponiendo } f \in \mathcal{C}^4[a,b]: \quad \sum_{i=1}^{n} f^4(\xi_i) = nf^4(\xi)\,,$$

para algún ξ en $[a, b]$.

Ejemplo 6.11. *Estimar el número de subintervalos en los que hay que dividir el intervalo* $[0,1]$ *para que la regla compuesta del trapecio calcule* $\int_0^1 e^{-x^2}dx$ *con un valor correcto a 6 dígitos.*

Solución. El error en la regla de trapecio es

$$-\frac{h^2}{12}(b-a)f''(\xi) = -\frac{h^2}{12}f''(\xi) = -\frac{1}{12n^2}f''(\xi)$$

con $a=0$, $b=1$, y $h=\dfrac{1}{n}$. Entonces debemos pedir que

$$\frac{1}{12n^2}|f''(\xi)| \leq 10^{-6}\,.$$

Como no conocemos ξ, basta con calcular

$$M = \max_{0\leq x\leq 1}|f''(\xi)|$$

y despues determinar n tal que

$$\frac{1}{12n^2}M \leq 10^{-6}\,.$$

Pero $f''(x) = e^{-x^2}(4x^2-2)$ es creciente en $0\leq x\leq 1$, pues $f'''(x) = 4xe^{-x^2}(3-2x^2) > 0$ en el mismo intervalo. Luego el máximo de $|f''(x)|$ debe tomarse en $x=0$ ó en $x=1$, así que

$$M = \text{máx}\{|f''(0)|,\, |f''(1)|\} = \text{máx}\{2,\, 2e^{-1}\} = 2\,,$$

y n debe ser tal que $\dfrac{1}{12n^2}2 \leq 10^{-6}$. Es decir

$$n \geq \sqrt{\frac{10^6}{6}} \approx 408$$

El valor $n=408$ subintervalos es realmente una sobre–estimación del número necesario ya que estamos usando $M = \text{máx}\{|f''(x)| : 0\leq x\leq 1\}$ en lugar de $|f''(\xi)|$. A continuación mostramos los valores obtenidos usando diferentes valores de n.

n	I	error	Estimación del error
50	0.74679961	$2{,}452\times 10^{-6}$	$6{,}67\times 10^{-5}$
100	0.74681800	$6{,}13\times 10^{-6}$	$1{,}67\times 10^{-5}$
200	0.74682260	$1{,}53\times 10^{-6}$	$4{,}17\times 10^{-6}$
400	0.74682375	$3{,}8\times 10^{-7}$	$1{,}04\times 10^{-6}$
800	0.74682404	9×10^{-8}	$2{,}6\times 10^{-7}$

Notese que con $n = 400$ subintervalos ya se obtiene un error menor a 10^{-6}. En la tabla de arriba la estimación del error es

$$\frac{b-a}{2}\max_{0\le x\le 1}|f''(x)|h^2 = \frac{b-a}{2}Mh^2 = \frac{Mh^2}{12} = \frac{M}{12n^2} = \frac{2}{12n^2} = \frac{1}{6n^2}$$

Con el objeto de comparar hacemos lo mismo con el método de Simpson.

$$\frac{b-a}{180}|f^4(\xi)|\left(\frac{h}{2}\right)^4 \le \frac{1}{180}\max|f^4(x)|\frac{h^4}{16} = \frac{12h^4}{180\times 16} = \frac{1}{240n^4} \le 10^{-6}\,,$$

en donde $\max_{0\le x\le 1}|f^4(x)| = 12$. Por lo tanto

$$n \ge \sqrt[4]{\frac{10^6}{240}} \approx 8\,,$$

y, como era de esperarse, este resultado muestra un aumento drámatico en la precisión cuando se utiliza el método de Simpson. A continuación mostramos los resultados para diferentes valores de n

n	I	error	Estimación del error
2	0.74685538	$3{,}125\times 10^{-5}$	$2{,}604\times 10^{-4}$
4	0.74682612	$1{,}99\times 10^{-6}$	$1{,}628\times 10^{-5}$
6	0.74682452	$3{,}96\times 10^{-7}$	$3{,}215\times 10^{-6}$
8	0.74682426	$1{,}27\times 10^{-7}$	$1{,}017\times 10^{-6}$
22	0.74682413	$4{,}99\times 10^{-9}$	$1{,}780\times 10^{-8}$

en donde la estimación del error se obtuvo por medio de

$$\frac{b-a}{180}\max|f^{(4)}(x)|\frac{h^4}{16} = \frac{1}{240\,n^4}\,.$$

6.7. Fórmulas de cuadratura de Gauss

Todas las reglas de integración numérica que hemos obtenido son de la forma

$$\int_a^b f(x)dx \approx w_0 f(x_0) + w_1 f(x_1) + \ldots + w_n f(x_n) \tag{6.2}$$

donde $x_0,\ x_1,\ldots,\ x_n$ son puntos de interpolación y $w_0,\ w_1,\ldots,\ w_n$ son pesos tales que $\sum_{i=0}^n w_i/(b-a) = 1$. Si se escogen los w_i como $w_i = \int_a^b \ell_i(x)dx$, donde $\ell_i(x)$ es el polinomio de Lagrange asociado al punto x_i, entonces la regla es exacta al menos para polinomios de grado n. Sin embargo, es posible hacer que la regla sea exacta para polinomio de grado $2n+1$

si se escogen los puntos $x_0,\ x_1, \ldots,\ x_n$ de manera apropiada, y no igualmente separados como lo hemos hecho hasta ahora. Esta es la idea básica de las reglas de cuadratura de Gauss.

Como los valores $w_0,\ w_1, \ldots,\ w_n$ se escogen de manera arbitraria y los puntos $x_0,\ x_1, \ldots,\ x_n$ tienen como única restricción que $f(x)$ este definida en ellos, entonces hay $2n+2$ parámetros (grados de libertad) involucrados. Dado que los polinomios de grado $2n+1$ contienen $2n+2$ coeficientes, entonces esta es la clase de polinomios de mayor grado para los cuales se espera que la fórmula (6.2) arriba sea exacta.

Primero discutiremos brevemente el concepto de conjunto ortogonal de funciones.

Definición 6.12. *El conjunto de funciones $\phi_0(x),\ \phi_1(x), \ldots,\ \phi_n(x)$, es un conjunto ortogonal en $[a, b]$ con respecto a la función de peso $v(x)$ siempre y cuando para toda $0 \leq i,\ j \leq n$*

$$\int_a^b v(x)\phi_i(x)\phi_j(x)dx = \delta_{ij}\,.$$

Ejemplo 6.13. ***Polinomios de Legendre***

Los polinomios de Legendre pueden generarse en forma recursiva:

$$\begin{aligned} \phi_0(x) &= 1 \\ \phi_1(x) &= x \\ (k+1)\phi_{k+1}(x) &= (2k+1)\,x\,\phi_k(x) - k\phi_{k-1}(x)\,, \quad k = 1,\ 2, \ldots. \end{aligned}$$

Se puede verificar fácilmente que estos polinomios son ortogonales en el intervalo $[-1, 1]$ con respecto a la función de peso $v(x) = 1$. Algunas propiedades de estos polinomios son

1. El polinomio $\phi_k(x)$ tiene k raices distintas en $(-1, 1)$. Por ejemplo

$$\begin{aligned} \phi_2(x) &= \frac{3}{2}(x^2 - \frac{1}{3}) \text{ tiene raices } x = \pm\sqrt{\frac{1}{3}} \\ \phi_3(x) &= \frac{5}{2}(x^2 - \frac{3}{5})\,x \text{ tiene raices } x = 0,\ x = \pm\sqrt{\frac{3}{5}}. \end{aligned}$$

2. Los polinomios $\phi_0(x),\ \phi_1(x), \ldots,\ \phi_n(x)$, al ser ortogonales y tener grados $0,\ 1, \ldots,\ n$, generan el espacio de polinomios de grado $\leq n$.

Si escogemos como puntos de interpolación las raices $x_1,\ x_2, \ldots,\ x_n$ del polinomio de Legendre de grado n, $\phi_n(x)$ sobre $[-1,\, 1]$, entonces podemos expresar $f(x)$ por medio de su polinomio de interpolación de la siguiente manera

$$f(x) = \sum_{i=1}^{n} f(x_i)\ell_i(x) + \frac{f^{(n)}(\xi(x))}{n!}\prod_{i=1}^{n}(x - x_i)$$

Esta claro que la regla de cuadratura

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \quad \text{con} \quad w_i = \int_{-1}^{1} \ell_i(x)dx = \int_{-1}^{1} \prod_{j=1,\, j\neq i}^{n} \frac{(x - x_j)}{(x_i - x_j)}dx$$

integra de manera exacta al menos polinomios de grado $\leq n-1$ ó cualquier otra función suave cuya derivada de orden $n-1$ sea cero sobre $[a, b]$. Verifiquemos que esta regla en realidad es exacta para polinomios de grado $2n-1$:

Sea $p(x)$ un polinomio de grado $2n-1$. Si dividimos este polinomio entre el polinomio de Legendre de grado n, $\phi_n(x)$, obtenemos un polinomio cociente $q(x)$ de grado $\leq n-1$, y un polinomio residuo $r(x)$ de grado $\leq n-1$, tales que

$$p(x) = q(x)\phi_n(x) + r(x)\,.$$

Así que

$$\int_{-1}^{1} p(x)dx = \int_{-1}^{1} q(x)\phi_n(x)dx + \int_{-1}^{1} r(x)dx$$

Además como $q(x)$ es polinomio de grado $\leq n-1$, entonces es combinación lineal de los polinomios de Legendre $\phi_0(x),\ \phi_1(x), \ldots,\ \phi_{n-1}(x)$. Es decir, existen $d_0,\ d_1, \ldots,\ d_{n-1} \in \mathbb{R}$ tales que

$$q(x) = \sum_{i=0}^{n-1} d_i\phi_i(x).$$

Luego

$$\begin{aligned}\int_{-1}^{1} p(x)dx &= \sum_{i=0}^{n-1} d_i \int_{-1}^{1} \phi_i(x)\phi_n(x)dx + \int_{-1}^{1} r(x)dx \\ &= \int_{-1}^{1} r(x)dx\,,\end{aligned}$$

debido a la ortogonalidad de los polinomios de Legendre. Por otro lado como $r(x)$ es un polinomio de grado $\leq n-1$, entonces la última integral en la expresión anterior se puede sustituir por

$$\sum_{i=1}^{n} r(x_i)w_i \quad \text{con} \quad w_i = \int_{-1}^{1} \ell_i(x)dx\ = \int_{-1}^{1} \prod_{i\neq j} \frac{(x - x_j)}{(x_i - x_j)}dx\,.$$

n	x_i	w_i
2	$x_2 = -x_1 = 0{,}5773502692$	$w_1 = w_2 = 1$
3	$x_3 = -x_1 = 0{,}7745966692$	$w_1 = w_3 = \frac{5}{9}$
	$x_2 = 0$	$w_2 = \frac{8}{9}$
4	$x_4 = -x_1 = 0{,}8611363116$	$w_1 = w_4 = 0{,}3478548451$
	$x_3 = -x_2 = 0{,}3399810436$	$w_2 = w_3 = 0{,}6521451549$
5	$x_5 = -x_1 = 0{,}9061798459$	$w_1 = w_5 = 0{,}2369268851$
	$x_4 = -x_2 = 0{,}5384693101$	$w_2 = w_4 = 0{,}4786286705$
	$x_3 = 0$	$w_3 = \frac{128}{225} = 0{,}5688888889$

Cuadro 6.1: Puntos y pesos para reglas de cuadratura de Gauss.

Pero además, como cada punto x_i, $1 \leq i \leq n$, es raiz de $\phi_n(x)$, se tiene $p(x_i) = q(x_i)\phi_n(x_i) + r(x_i) = r(x_i)$. Por lo tanto,

$$\int_{-1}^{1} p(x)dx = \sum_{i=1}^{n} w_i p(x_i)$$

lo cual comprueba que la regla de la cuadratura es exacta para el polinomio arbitrario $p(x)$ de grado $\leq 2n-1$.

En la siguiente tabla se muestran valores $x_i,\ w_i$ a diez cifras decimales para distintos valores de n.

En la práctica, queremos integrar en un intervalo arbitrario $[a, b]$ y no solamente en $[-1, 1]$. En estos casos utilizamos el cambio de variable

$$t = \frac{2x - a - b}{b - a}$$

el cual transforma $[a, b]$ en $[-1, 1]$ si $a < b$. En este caso

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

y

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

La regla Gaussiana se aplica a esta última integral.

Ejemplo 6.14. *Calcular* $I = \int_0^1 e^{-x^2}dx$ *utilizando la fórmula Gausiana de 5 puntos* ($n = 5$).

Solución: $a = 0,\ b = 1$ implica $x = \dfrac{t+1}{2}$. Por lo tanto

$$\begin{aligned}\int_0^1 e^{-x^2}dx &= \int_{-1}^1 e^{-(\frac{t+1}{2})^2}\frac{1}{2}dt = \frac{1}{2}\int_{-1}^1 e^{-(\frac{t+1}{2})^2}dt \\ &\approx \frac{1}{2}\sum_{i=1}^{5} w_i e^{-(\frac{x_i+1}{2})^2} = 0{,}74682413\end{aligned}$$

utilizando los valores x_i y w_i de la tabla con $n = 5$.

Observación. Para obtener una precisión comparable con la regla del trapecio se requieren $2,800$ subdivisiones de $[0, 1]$, mientras que con la regla de Simpson se necesitan 20 subdivisiones aproximadamente.

Otros casos que llevan a reglas de integración de Gauss son

Intervalo	Función de peso $v(x)$	Nombre
$[-1, 1]$	$(1-x^2)^{1/2}$	$T_n(x)$, Chebishev
$[-1, 1]$	$(1-x)^\alpha(1+x)^\beta,\ \alpha, \beta > -1$	$P_n^{\alpha,\beta}(x)$, Jacobi
$[0, \infty)$	e^{-x}	$L_n(x)$, Laguerre
$(-\infty, \infty)$	e^{-x^2}	$H_n(x)$, Hermite

Capítulo 7

Aproximación Numérica de Ecuaciones Diferenciales Ordinarias

7.1. Conceptos básicos

Los problemas de valores iniciales ó ecuaciones diferenciales ordinarias ocurren en muchas áreas de la ciencia, pero más particularmente en la mecánica (incluyendo la mecánica celeste), donde el movimiento de partículas (planetas) obedece la $2^{\underline{a}}$ ley de Newton. Algunos astrónomos como Adams, Moulton, Cowell, contribuyeron al desarrollo de ténicas numéricas para resolver ecuaciones diferenciales ordinarias. Las ecuaciones diferenciales ordinarias también juegan un papel importante en la mecánica cuántica, en el estudio de dinámica de poblaciones, modelos biológicos, desintegración radiactiva, economía, circuitos eléctricos, entre otros campos. Los sistemas de ecuaciones diferenciales ordinarias aparecen también como parte importante de la busqueda de soluciones de las ecuaciones diferenciales parciales.

El problema típico de valores iniciales (que denotaremos por PVI) involucra sistemas de ecuaciones diferenciales de $1^{\underline{er}}$ orden de la forma

$$\frac{dy}{dt} = f(t, y), \quad a < t \leq b, \tag{7.1}$$

$$y(t_0) = y_0 \qquad (t_0 = a), \tag{7.2}$$

donde $y = y(t) = (y_1(t), y_2(t), \ldots, y_n(t))^T,\ y_0 = (y_{01}, y_{02}, \ldots, y_{0n})^T \in \mathbb{R}^n,\ f = (f_1, f_2, \ldots, f_n)^T$ función vectorial con cada una de sus funciones componentes $f_i = f_i(t, y_1, y_2, \ldots, y_n)$ de $\mathbb{R}^{n+1}$ en $\mathbb{R}$.

Ejemplos de ecuaciones diferenciales ordinarias (e.d.o.)

1. **Geometría**: La ecuación de una circunferencia de radio r con centro en el origen es $x^2 + y^2 = r^2$. Suponer $y = y(x)$, entonces derivando implícitamente la ecuación de la circunferencia, se obtiene:

$$2x + 2y\frac{dy}{dx} = 0 \Rightarrow \frac{dy}{dx} = -\frac{x}{y}$$

Si identificamos x con t entonces $y = y(t)$, y obtenemos la ecuación diferencial ordinaria

$$\begin{aligned} \frac{dy}{dt} &= -\frac{t}{y}, \\ y(0) &= r \end{aligned}$$

2. **Oscilador armónico simple**: Consideremos un bloque de masa m suspendido de un resorte. Si a partir de su posición de equilibrio estiramos ó encogemos el resorte, entonces despreciando la fricción con el aire, este bloque comienza a oscilar. La ecuación que describe el movimiento oscilatorio se obtiene a partir de la segunda Ley de Newton.

$$mx''(t) = -kx(t),$$

donde $x(t)$ indica la desviación del bloque respecto a la posición de equilibrio, k es la constante de rígidez (ó restitución) del resorte. La anterior ecuación ecuación puede

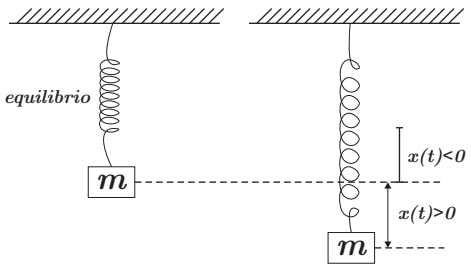


Figura 7.1: Bloque de masa m suspendido por un resorte.

reescribirse en la forma

$$x'' + w^2x = 0, \text{ con } w = \sqrt{k/m}.$$

A su vez, esta ecuación diferencial de segundo orden puede expresarse como un sistema de dos ecuaciones diferenciales de primer orden (7.1), con

$$y(t) = (y_1(t), y_2(t))^T, \tag{7.3}$$

$$f(t, y) = (y_2, -w^2y_1)^T = \begin{bmatrix} 0 & 1 \\ -w^2 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{7.4}$$

si hacemos la sustitución:

$$\begin{aligned} y_1(t) = x(t) \quad &\Rightarrow \quad y_1'(t) = y_2(t) \\ y_2(t) = x'(t) \quad &\Rightarrow \quad y_2'(t) = -w^2 y_1(t) \end{aligned}$$

En forma análoga, una ecuación diferencial ordinaria de orden n de la forma

$$y^{(n)}(t) = g(t, y(t), y'(t), \ldots, y^{(n-1)}(t)),$$

se puede transformar en un problema del tipo $y' = f(t, y)$, haciendo la sustitución:

$$\begin{aligned} y_1(t) = y(t) &\Rightarrow y_1'(t) = y'(t) = y_2(t) \\ y_2(t) = y'(t) &\Rightarrow y_2'(t) = y''(t) = y_3(t) \\ &\vdots \qquad\qquad\qquad\qquad \vdots \\ y_{n-1}(t) = y^{(n-2)}(t) &\Rightarrow y_{n-1}'(t) = y^{(n-1)}(t) = y_n(t) \\ y_n(t) = y^{(n-1)}(t) &\Rightarrow y_n'(t) = y^{(n)}(t) = g(t, y_1, y_2, \ldots, y_n) \end{aligned}$$

e identificando a $y(t)$ y $f(t, y)$ con

$$\begin{aligned} y(t) &= (y_1(t), y_2(t), \ldots, y_n(t))^T, \\ f(t, y) &= (y_2(t), y_3(t), \ldots, y_n(t), g(t, y_1(t), y_2(t), \ldots, y_n(t)))^T, \end{aligned}$$

respectivamente.

3. **Problema de dos cuerpos**. Un problema ligeramente más complicado de la mecánica celeste es el que describe la órbita de un cuerpo de masa m bajo la atracción de otro cuerpo de masa mucho mayor M como se ilustra en la Figura 7.2 .

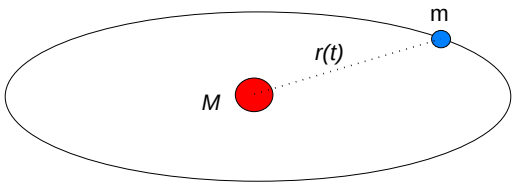


Figura 7.2: Dinámica de dos cuerpos.

Usando coordenadas cartesianas $x_1(t),\ x_2(t)$, centradas en el cuerpo más pesado, las ecuaciones son:

$$\begin{aligned} x_1''(t) &= -GMx_1(t)/r(t)^3, \\ x_2''(t) &= -GMx_2(t)/r(t)^3, \end{aligned}$$

donde G es la constante de gravitación universal y $r(t) = \sqrt{x_1(t)^2 + x_2(t)^2}$. Si hacemos la sustitución:

$$\begin{aligned} y_1(t) = x_1(t) &\Rightarrow y_1'(t) = x_1'(t) = y_3(t) \\ y_2(t) = x_2(t) &\Rightarrow y_2'(t) = x_2'(t) = y_4(t) \\ y_3(t) = x_1'(t) &\Rightarrow y_3'(t) = x_1''(t) = -GMy_1(t)/r(t)^3 \\ y_4(t) = x_2'(t) &\Rightarrow y_4'(t) = x_2''(t) = -GMy_2(t)/r(t)^3 \end{aligned}$$

con $r(t) = \sqrt{x_1(t)^2 + x_2(t)^2} = \sqrt{y_1(t)^2 + y_2(t)^2}$, obtenemos el sistema de ecuaciones diferenciales ordinarias de primer orden $y^{'}(y) = f(t, y)$ donde

$$\begin{aligned} y(t) &= (y_1(t), y_2(t), y_3(t), y_4(t))^T, \\ f(t, y) &= (f_1, f_2, f_3, f_4)^T, \end{aligned}$$

con

$$\begin{aligned} f_1(t, y) &= y_3, \\ f_2(t, y) &= y_4, \\ f_3(t, y) &= -GM\, y_1/(y_1^2 + y_2^2)^{3/2}, \\ f_4(t, y) &= -GM\, y_2/(y_1^2 + y_2^2)^{3/2}. \end{aligned}$$

El **teorema fundamental del cálculo** establece una conexión importante entre ecuaciones diferenciales e integrales

$$\frac{dy}{dt} = f(t, y) \quad \Longleftrightarrow \quad y(t + h) = y(t) + \int_t^{t+h} f(s, y(s))ds$$

No podemos usar integración numérica directamente en el problema de la derecha por que $y(s)$ es una función desconocida. De cualquier manera la idea básica es escoger una sucesión de valores h (pasos del tiempo) y a partir de dicha fórmula generar nuestra solución numérica. Un caso especial que hay que tomar en cuenta es cuando $f(t, y) = f(t)$ es solamente función de t. En este caso la solución de la ecuación diferencial autónoma consiste en encontrar la antiderivada de $f(t)$, es decir su integral:

$$y(t + h) = y(t) + \int_t^{t+h} f(s)ds$$

La solución numérica es entonces una sucesión de cuadraturas de la forma

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(s)ds$$

7.2. Existencia y unicidad de la solución

La existencia y unicidad de soluciones del problema (7.1)–(7.2) depende fundamentalmente de las propiedades de "suavidad" de la función $f(t, y)$. En particular se requiere que f sea continua respecto a t y Lipschitz continua respecto a y. Antes de ununciar el teorema, recordamos este último concepto.

Definición 7.1. *Una función $f : [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$ se dice que es* Lipschitz continua *respecto a la variable y si existe una constante $L > 0$ tal que*

$$\|f(t, y) - f(t, y^*)\| \leq L\|y - y^*\|, \quad \forall\ t \in [a, b],\ \forall\ y,\ y^* \in \mathbb{R}^n$$

en alguna norma vectorial $\|\cdot\|$ en $\mathbb{R}^n$

Ejemplo 7.2. *En el problema del oscilador armónico encontramos que $f(t, y) = (y_2, -w^2 y_1)^T$, con w constante. Entonces en la norma euclidiana tenemos*

$$\begin{aligned} \|f(t, y) - f(t, y^*)\|_2 &= \|(y_2 - y_2^*, -w^2(y_1 - y_1^*))^T\|_2 \\ &= \sqrt{(y_2 - y_2^*)^2 + w^4(y_1 - y_1^*)^2} \\ &\leq \max\{1, w^2\}\sqrt{(y_2 - y_2^*)^2 + (y_1 - y_1^*)^2} \\ &= \max\{1, w^2\}\|y - y^*\|_2 \end{aligned}$$

Por tanto, si tomamos $L = \max\{1, w^2\}$, entonces se satisface una condición de Lipschitz para $f(t, y)$ (en realidad en este caso f no depende de t).

Observese que la condición tipo Lipschitz es parecida a la propiedad que define a una aplicación como una contracción (ver capítulo 3). Sin embargo, la constante de Lipschitz no tiene que ser $L < 1$, pudiendo ser incluso muy grande, pero constante. Intuitivamente hablando, la condición de Lipschitz asegura que la función $f(t, y)$ no tenga cambios muy "bruscos" ó infinitos cuando y varía.

Teorema 7.3. ***Teorema de existencia y unicidad**. Si $f(t, y)$ es una función continua respecto a t en $[a, b]$, Lipschitz continua respecto a y en $\mathbb{R}^n$, entonces el problema de valores iniciales (PVI) tiene una solución única $y(t)$, $a \leq t \leq b$ para cualquier condición inicial $y_0 \in \mathbb{R}^n$. Además $y(t)$ depende continuamente de t_0 y y_0 (los datos iniciales).*

Nota 1. En ocasiones la condición de Lipschitz no se satisface para toda $y \in \mathbb{R}^n$, sino en un subconjunto Y de $\mathbb{R}^n$. En este caso el teorema de existencia y unicidad sigue siendo válido en $[a, b] \times Y$.

Nota 2. En ocasiones es muy difícil verificar una condición de Lipschitz en forma directa. Sin embargo si las derivadas parciales $\dfrac{\partial f_i}{\partial y_i}(t,y)$ son continuas y acotadas en $[a,b]\times\mathbb{R}^n$ ó $[a,b]\times Y$ (con $Y\subset\mathbb{R}^n$), entonces $f(t,y)$ es Lipschitz continua respecto a y en $\mathbb{R}^n$ ó Y, respectivamente.

Ejemplo 7.4. *La función $f(t,y)=-t/y$ no es Lipschitz continua en cualquier conjunto que tenga $y=0$. Sin embargo como $\partial f/\partial y=-t/y^2$ es continua respecto a t y respecto a toda $y\neq 0$, entonces $f(t,y)$ es Lipschitz continua en cualquier $t\in[a,b]$ y cualquier conjunto Y que no contenga $y=0$. Geométricamente es fácil observar que el problema no tiene solución si la condición inicial fuese $y(0)=0$ ó $y(0)=-r<0$, pues no hay circunferencias de radio nulo ó negativo.*

7.3. Métodos de aproximación con series de Taylor

En el presente estudio solo consideraremos métodos de aproximación de *variable discreta* del problema PVI. En estos métodos se encuentran aproximaciones y_i de la solución exacta $y(t_i)$ solo en puntos discretos $t_i\in[a,b]$, como se muestra en la Figura 7.3. Los métodos

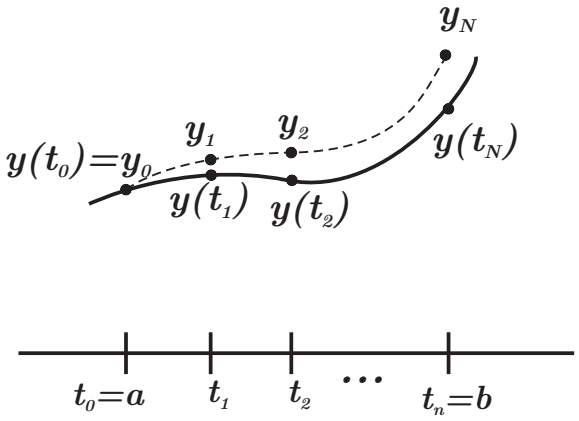


Figura 7.3: Aproximación de *variable discreta*.

de series de Taylor son métodos de este tipo. La idea para obtener este tipo de métodos consiste en lo siguiente: Sea $y(t)$ la solución de (7.1)–(7.2) al tiempo arbitrario t. Si h es un incremento del tiempo, entonces

$$y(t+h)=y(t)+y'(t)h+\frac{y''(t)}{2!}h^2+\frac{y'''(t)}{3!}h^3+\ldots. \tag{7.5}$$

Si $y(t)$ no es conocida, las derivadas de $y(t)$ tampoco lo son. Sin embargo, sabemos que

$$\begin{aligned} y'(t) &= f(t,y), \\ y''(t) &= f^{(1)}(t,y) = f_t(t,y) + f_y(t,y)f(t,y), \\ y'''(t) &= f^{(2)}(t,y) = f_t^{(1)}(t,y) + f_y^{(1)}(t,y)f(t,y), \\ &\vdots \\ y^{(k+1)}(t) &= f^{(k)}(t,y) = f_t^{(k-1)}(t,y) + f_y^{(k-1)}(t,y)f(t,y)\,, \end{aligned}$$

en donde $f^{(1)}$ y $f^{(2)}, \ldots, f^{(k)}$, $k \geq 0$, indican derivadas totales de f respecto a t. Por lo tanto (7.5) puede reescribirse como

$$y(t+h) = y(t) + hf(t,y) + \frac{h^2}{2}f^{(1)}(t,y) + \frac{h^3}{3!}f^{(2)}(t,y) + \ldots \tag{7.6}$$

Observese que si la función $f(t,y)$ no es simple, las derivadas $f^{(1)}$, $f^{(2)}$, $f^{(3)}$ serán cada vez más complicadas. Además por razones prácticas debemos limitarnos a un número finito de términos en la serie. Esta limitación restringe los valores de h para los cuales (7.6) es una buena aproximación a $y(t)$. Si cortamos la serie a $p+1$ términos ($p \geq 1$), se obtiene

$$\begin{aligned} y(t+h) &= y(t) + hf(t,y) + \frac{h^2}{2!}f^{(1)}(t,y) + \ldots + \frac{h^p}{p!}f^{(p-1)}(t,y) + \frac{h^{p+1}}{(p+1)!}f^{(p)}(\xi, y(\xi)) \\ &= y(t) + h\left[f(t,y) + \frac{h}{2!}f^{(1)}(t,y) + \ldots + \frac{h^{p-1}}{p!}f^{(p-1)}(t,y)\right] + \frac{h^{p+1}}{(p+1)!}f^{(p)}(\xi, y(\xi)) \end{aligned}$$

con ξ entre t y $t+h$. Entonces, podemos escribir

$$y(t+h) = y(t) + h\Phi(t,y;h) + \frac{h^{p+1}}{(p+1)!}f^{(p)}(\xi, y(\xi)), \tag{7.7}$$

con

$$\Phi(t,y;h) = \sum_{k=0}^{p-1} f^{(k)}(t,y)\frac{h^k}{(k+1)!} \tag{7.8}$$

El algoritmo de Taylor, y otros métodos basados en este tipo de algoritmos calculan y en t_{i+1} usando solamente información de y en t_i, donde $t_{i+1} = t_i + h_i$ es un incremento en tiempo de t_i. La forma más sencilla de llevar a cabo esta estrategia es dividir el intervalo $[a,b]$ en N subintervalos de igual longitud $h = (b-a)/N$, obteniendo el

Algoritmo de Taylor de orden *p*

1. Escoger $h = (b-a)/N$ y sean $t_i = t_0 + ih$, con $i = 0, 1, \ldots, N$.

2. Dado el valor inicial y_0, para cada $i = 0, 1, \ldots, N-1$ generar una aproximacion y_{i+1} de la solución exacta $y(t_{i+1})$ por medio de la relación de recurrencia

$$y_{i+1} = y_i + h\Phi(t_i, y_i; h), \tag{7.9}$$

Por ejemplo, cuando $p = 1$ se obtiene $\Phi(t, y; h) = f(t, y)$ y obtenemos el **método de Euler**, cuya relación de recurrencia es:

$$y_{i+1} = y_i + hf(t_i, y_i), \quad i = 0, 1, \ldots, N-1. \tag{7.10}$$

El **método de Taylor de 2^o orden** se obtiene con $p = 2$. En este caso $\Phi(t, y; h) = f(t, y) + \frac{h}{2}f^{(1)}(t, y)$, y la relación de recurrencia es

$$y_{i+1} = y_i + h\left[f(t_i, y_i) + \frac{h}{2}f^{(1)}(t_i, y_i)\right], \quad i = 0, 1, \ldots, N-1 \tag{7.11}$$

Ejemplo 7.5. *Considerese la ecuación diferencial ordinaria escalar*

$$y'(t) = 2y - y^2, \quad 0 < t \leq 2$$
$$y(0) = 1.$$

La solución exacta de esta ecuación es $y(t) = 2/(1+e^{-2t})$. En la Figura 7.4 y en la Tabla 7.1 se muestran los resultados con el método de Euler (7.10) y el método de Taylor (7.11) tomando $h = 0.1$. Tanto en la tabla como en la gráfica se observa una mejor aproximación con el

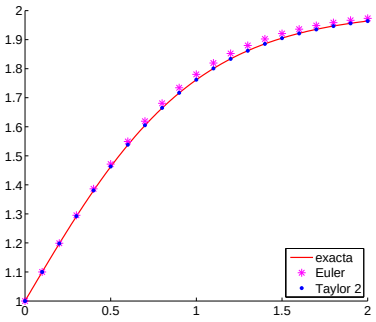


Figura 7.4: Gráficas de las soluciones con los métodos de Euler y Taylor de 2^o orden.

método de Taylor de orden 2 que con el método de Euler como era de esperarse. Observese que en este caso $f(t, y) = 2y - y^2$ y entonces $f^{(1)}(t, y) = (2-2y)(2y-y^2) = 2y(1-y)(2-y)$.

t	Sol. Euler	Sol. Taylor	Error Euler	Error Taylor
0.0	1.0000000	1.0000000	0.0000000	0.0000000
0.1	1.1000000	1.1000000	3.3200538e-4	3.3200538e-4
0.2	1.1990000	1.1980100	1.6246798e-3	6.3467978e-4
0.3	1.2950399	1.2921867	3.7272875e-3	8.7412722e-4
0.4	1.3863350	1.3809770	6.3860835e-3	1.0280498e-3
0.5	1.4714096	1.4632059	9.2924117e-3	1.0886996e-3
0.6	1.5491869	1.5381117	1.2137304e-2	1.0621176e-3
0.7	1.6190262	1.6053323	1.4658472e-2	9.6455088e-4
0.8	1.6807069	1.6648544	1.6670129e-2	8.1761418e-4
0.9	1.7343707	1.7169416	1.8072841e-2	6.4369998e-4
1.0	1.7804407	1.7620568	1.8846521e-2	4.6259408e-4
1.1	1.8195319	1.8007886	1.9032890e-2	2.8960765e-4
1.2	1.8523687	1.8337897	1.8714049e-2	1.3504982e-4
1.3	1.8797154	1.8617278	1.7992264e-2	4.6307489e-6
1.4	1.9023255	1.8852520	1.6973853e-2	9.9641180e-5
1.5	1.9209064	1.9049698	1.5758116e-2	1.7841400e-4
1.6	1.9360995	1.9214345	1.4430961e-2	2.3401902e-4
1.7	1.9484713	1.9351394	1.3062215e-2	2.6967832e-4
1.8	1.9585115	1.9465171	1.1705495e-2	2.8892268e-4
1.9	1.9666371	1.9559423	1.0399618e-2	2.9520355e-4
2.0	1.9731984	1.9637359	9.1707727e-3	2.9166246e-4

Cuadro 7.1: Comparación entre los métodos de Euler y Taylor de 2^o orden.

Ejemplo 7.6. *Consideremos la ecuación del oscilador armónico simple*

$$x''(t) + w^2\, x(t) = 0, \quad 0 < t \leq 2\pi,$$
$$x(0) = 1, \quad x'(0) = 0.$$

Transformamos la ecuación diferencial de segundo orden en un sistema de dos ecuaciones de primer orden $y^{'} = f(t, y)$ con y y $f(t, y)$ como se indica en (7.3)–(7.4). La condición inicial es $y(0) = (y_1(0), y_2(0))^T = (1, 0)^T$, además

$$f^{(1)}(t, y) = f_t(t, y) + f_y(t, y)f(t, y) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -w^2 & 0 \end{bmatrix} \begin{bmatrix} y_2 \\ -w^2 y_1 \end{bmatrix} = -w^2 \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Tomando $w = 2$, se obtiene

$$f(t, y) = (y_2, -4y_1)^T, \qquad f^{(1)}(t, y) = -4(y_1, y_2)^T,$$

y el método de Euler (7.10) y Taylor de 2^o orden (7.11) con $h = 0.1$ producen los resultados mostrados en la Figura 7.5. De nuevo es evidente la superioridad del método de Taylor de 2^o orden. Observese que en este caso la diferencia entre ambos métodos es aún mayor que en el caso anterior debido a que la solución con el método de Euler se deteriora cada vez más conforme avanza el tiempo.

A continuación introducimos algunas definiciones útiles para analizar los métodos de Taylor y otros métodos que estudiaremos más adelante.

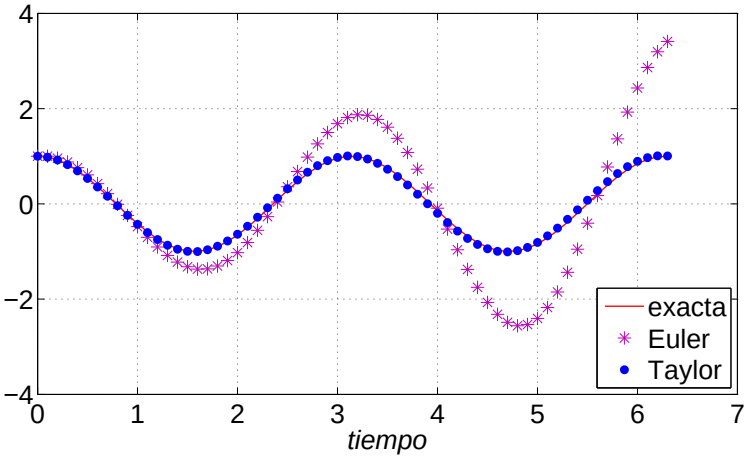


Figura 7.5: Soluciones numéricas para el oscilador armónico simple.

Definición 7.7. ***Error de truncamiento***. *El error de truncamiento del método de un paso*

$$y_{i+1} = y_i + h\Phi(t_i, y_i; h), \tag{7.12}$$

se define como

$$T(t, y; h) = \frac{1}{h}(y_{aprox} - y(t+h)),$$

donde

$$y_{aprox} = y(t) + h\Phi(t, y(t); h).$$

Entonces el error de truncamiento es una medida de la diferencia entre la soluciones aproximada y exacta de la ecuación diferencial por unidad del paso de tiempo h. Sustituyendo la expresión y_{aprox} en la expresión $T(t, y; h)$, obtenemos

$$T(t, y; h) = \Phi(t, y; h) - \frac{1}{h}[y(t+h) - y(t)] \tag{7.13}$$

Definición 7.8. ***Consistencia***. *El método de un paso (7.12) es consistente si*

$$T(t, y; h) \to 0, \quad \textit{cuando} \quad h \to 0.$$

De la relación (7.13) se obtiene que el método es consistente si

$$\begin{aligned} 0 &= \lim_{h\to 0} T(t, y; h) = \lim_{h\to 0} \Phi(t, y; h) - \lim_{h\to 0} \frac{y(t+h) - y(t)}{h} \\ &= \Phi(t, y; 0) - y'(t) = \Phi(t, y; 0) - f(t, y). \end{aligned}$$

Es decir, el método de un paso es consistente si

$$\Phi(t, y; 0) = f(t, y) \tag{7.14}$$

Definición 7.9. ***Orden del método*** *El método (7.12) se dice que tiene orden p, con p entero ≥ 1, si en alguna norma vectorial*

$$\|T(t, y; h)\| \leq c\, h^p \tag{7.15}$$

uniformemente sobre $[a, b] \times \mathbb{R}^n$, donde c no depende de t, y ó h.

Para expresar que el método es de orden p, usualmente escribimos

$$T(t, y; h) = \mathcal{O}(h^p), \quad h \to 0$$

Observese que si $p > 1$, entonces automáticamente se tiene consistencia en el método. El número p se denomina el **orden exacto del método** si la desigualdad (7.15) no se cumple para ningún número mayor a p.

Definición 7.10. ***Función principal del error***. *Si Φ determina un método de orden p, a una función $\tau : [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$ que satisfaga*

$$\tau(x, y) \neq 0 \quad y \quad T(t, y; h) = \tau(t, y)h^p + \mathcal{O}(h^{p+1}),\ h \to 0,$$

se denomina la función principal del error.

La función principal del error determina el término líder en el error de truncamiento. El número p en

$$T(t, y; h) = \tau(t, y)h^p + \mathcal{O}(h^{p+1})$$

es el orden exacto del método ya que $\tau(t, y) \neq 0$.

Nota. Todas las definiciones anteriores se hacen con la idea en mente de que $h > 0$ es un número pequeño. Entonces entre mayor es p más preciso es el método. Por ejemplo, el método de Taylor de 2^o orden ($p = 2$) es más preciso que el método de Euler ($p = 1$).

Ejemplo 7.11. *Ilustraremos las definiciones anteriores para el método de Euler:*

$$y_{i+1} = y_i + hf(t_i, y_i), \quad i = 0,\ 1,\ 2, \ldots,\ N - 1$$

Dado que $\Phi(t,y;h) = f(t,y)$ no depende de h, entonces $\Phi(t,y,0) = f(t,y)$ e inmediatamente concluimos que el método es consistente. El error de truncamiento es: (suponiendo que $y \in \mathcal{C}^2[a,b]$)

$$\begin{aligned} T(t,y;h) &= \Phi(t,y;h) - \frac{1}{h}(y(t+h) - y(t)) = f(t,y) - \frac{1}{h}(y(t+h) - y(t)) \\ &= y'(t) - \frac{1}{h}(hy'(t) + \frac{h^2}{2}y''(\xi)) = -\frac{h}{2}y''(\xi) = -\frac{h}{2}f^{(1)}(\xi,\, y(\xi)) \end{aligned}$$

donde $f^{(1)} \equiv f_t + f_y\, f$, con ξ entre t y $t+h$. Por lo tanto, si f y todas sus derivadas parciales de primer orden están uniformemente acotadas en $[a,b] \times \mathbb{R}^n$, entonces

$$\|T(t,y;h)\| \leq c\, h.$$

Esto muestra que efectivametne el método de Euler tiene $p = 1$.

Si hacemos la misma suposición de acotamiento uniforme sobre las segundas derivadas de y, entonces $y''(\xi) = y''(t) + \mathcal{O}(h)$, así que

$$T(t,y;h) = -\frac{h}{2}\left[f_t + f_y f\right](t,y) + \mathcal{O}h^2, \quad h \to 0,$$

lo cual demuestra que la función principal del error es

$$\tau(t,y) = -\frac{1}{2}[f_t + f_y f](t,y) = -\frac{1}{2}f^{(1)}(t,y).$$

Así pues, a menos que $f_t + f_y f \equiv 0$, el método de Euler tiene exactamente orden $p = 1$.

Ejemplo 7.12. *En forma análoga podemos hacer un análisis para el método de Taylor general (7.9).*

Suponiendo que $f \in \mathcal{C}^p$ sobre $[a,b] \times \mathbb{R}^n$ de (7.7), (7.8) y (7.13) se obtiene

$$T(t,y;h) = -\frac{h^p}{(p+1)!}f^{(p)}(\xi,\, y(\xi))$$

con ξ entre t y $t+h$. Así que

$$\|T(t,y;h)\| \quad \leq \quad \frac{c_p}{(p+1)!}h^p,$$

con $c_p =$ cota de $f^{(p)}(t,y)$ en $[a,b] \times \mathbb{R}^n$. Entonces, el método tiene exactamente el orden p, a menos que $f^p(t,y) \equiv 0$ y, por lo tanto, la función principal del error es

$$\tau(t,y) = -\frac{1}{(p+1)!}f^p(t,y) \tag{7.16}$$

Claramente el método es consistente, pues de (7.8) se obtiene

$$\Phi(t, y; 0) = f(t, y).$$

Una de las principales desventajas de los métodos de Taylor de orden mayor es que es necesario cálcular derivadas de orden superior $f^{(k)}(t, y)$. Por esta razón los investigadores optaron por buscar métodos de orden mayor al método de Euler pero sin la necesidad de calcular derivadas de $f(t, y)$. En la siguiente sección se introducen tales métodos, denominados métodos de Runge-Kutta.

7.4. Métodos de Runge-Kutta

En el método de Euler, para calcular $y(t + h)$ a partir de $y(t)$, se "corrige" el valor $y(t)$ sumando $hf(t, y)$, para obtener

$$y_{aprox} = y(t) + hf(t, y),$$

donde y_{aprox} denota una aproximación al valor exacto $y(t + h)$. La Figura 7.6 ilustra la diferencia entre y_{aprox} y $y(t + h)$. Se toma en cuenta que $f(t, y) = y'(t)$ representa la pendiente de la tangente a y en t.

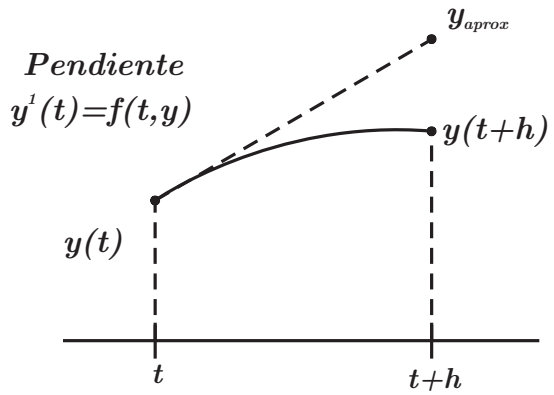


Figura 7.6: Interpretación geométrica del método de Euler.

Habiendo hecho esta descripción geométrica, observamos que para mejorar el cálculo es necesario seguir una mejor dirección a partir del punto $(t, y(t))$. Esto se logra reevaluando la pendiente a la mitad del intervalo $[t,\, t + h]$, y entonces seguir la pendiente revisada sobre

el intervalo completo. Es decir

$$\text{Pendiente revisada:} \qquad f\left(t+\frac{h}{2},\ y+\frac{h}{2}f(t,y)\right)$$

$$\text{Nuevo método:} \qquad y_{aprox} = y + h\, f\left(t+\frac{h}{2},\ y+\frac{h}{2}f(t,y)\right)$$

En este caso obtenemos

$$\Phi(t,y,h) = f(t+\frac{h}{2}, y+\frac{h}{2}f(t,y)).$$

Esta construcción se muestra en la Figura 7.7. Notese la característica anidada de f en la

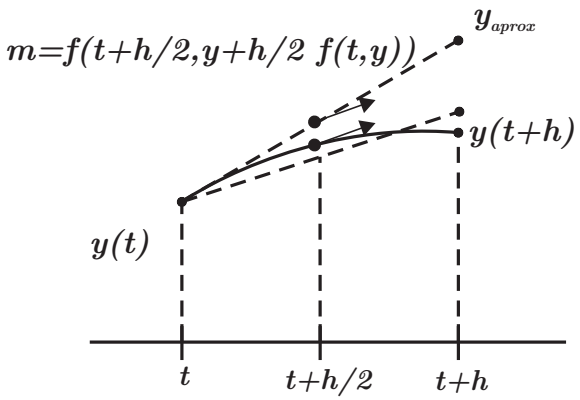


Figura 7.7: Interpretación del método de Euler mejorado.

fórmula de aproximación. Para propósitos de programación se deshace el anidamiento de f y escribimos:

Método de Euler modificado (ó mejorado)

$$\begin{cases} k_1(t,y) = f(t,y) \\ k_2(t,y) = f(t+\frac{h}{2}, y+\frac{h}{2}k_1) \\ y_{aprox} = y + hk_2 \end{cases} \tag{7.17}$$

En este método, hemos tomado dos pendientes de prueba: k_1 y k_2. La primera se toma en el punto inicial, y la segunda a la mitad del camino, y al final hemos tomado la segunda como pendiente para avanzar.

Es posible también tomar la segunda pendiente de prueba en el punto $(t+h, y+hf(t,y))$, pero entonces se corre el peligro de haber esperado mucho para revaluar la pendiente. Una

mejor opción es tomar la pendiente final como el promedio de las dos pendientes $m_1 = f(t, y)$ y $m_2 = f(t + h, y + hf(t, y))$. Con esta estrategia obtenemos el siguiente método, cuya interpretación gráfica se muestra en la Figura 7.8: **Método de Heun (ó del trapecio)**

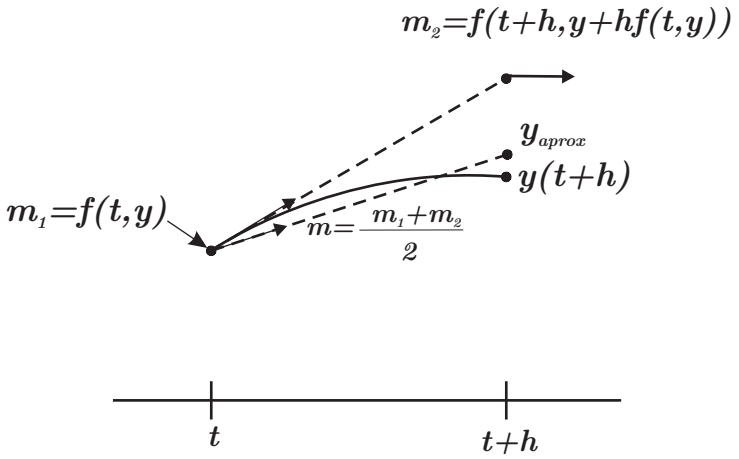


Figura 7.8: Interpretación del método del Trapecio.

$$\begin{cases} k_1(t, y) = f(t, y) \\ k_2(t, y) = f(t + h, y + hk_1) \\ y_{aprox} = y + \frac{h}{2}(k_1 + k_2) \end{cases} \tag{7.18}$$

En este caso

$$\Phi(t, y; h) = \frac{1}{2}[f(t, y) + f(t + h, y + hf(t, y))].$$

En los algoritmos usualmente no se utiliza la letra m para denotar las pendientes en los métodos. En su lugar se utliza la letra k. Las ventajas que se obtienen con estas modificaciones para revaluar la pendiente son básicamente dos:

1. Los métodos obtenidos tienen un orden $p = 2$, el cual es mayor que el método de Euler.
2. No se evalúan derivadas de f. Solo se evalúa f dos veces en cada paso de tiempo.

7.5. Métodos de Runge-Kutta de dos etapas

Una forma más sistemática para obtener los métodos anteriores (7.17) y (7.18) consiste en suponer que Φ es un promedio pesado de dos pendientes k_1 y k_2. Es decir

$$\Phi(t, y; h) = \alpha_1 k_1 + \alpha_2 k_2$$

con $\alpha_1 + \alpha_2 = 1$, $\alpha_1,\ \alpha_2 \geq 0$, y

$$\begin{aligned} k_1(t,y) &= f(t,y), \\ k_2(t,y) &= f(t+\mu h, y+\mu h k_1), \end{aligned}$$

En las pendientes anteriores el parámetro μ expresa que fracción de tiempo nos movemos entre t y $t+h$. La idea fundamental de los métodos de Runge–Kutta es intentar escoger los parámetros $\alpha_1,\ \alpha_2$ y μ de tal manera que el orden del método se pueda maximizar. Es decir, de tal forma que

$$\begin{aligned} T(t,y;h) &= \Phi(t,y;h) - \frac{y(t+h)-y(t)}{h} \\ &= \alpha_1 f(t,y) + \alpha_2 f(t+\mu h, y+\mu h k_1) - \frac{y(t+h)-y(t)}{h} \\ &= \mathcal{O}(h^p),\ h \to 0 \end{aligned}$$

con p máximo. Si escribimos $\Delta t = \mu h$ y $\Delta y = \mu h\, k_1 = \Delta t\, f(t,y)$, entonces

$$\begin{aligned} f(t+\mu h, y+\mu h k_1) &= f(t+\Delta t, y+\Delta y) \\ &= f(t,y) + \left[f_t \Delta t + f_y \Delta y\right](t,y) \\ &\qquad + \frac{1}{2}\left[f_{tt}\Delta t^2 + 2 f_{ty}\Delta t \Delta y + \Delta y^T f_{yy} \Delta y\right](t,y) + \ldots \\ &= f(t,y) + \mu h \left[f_t + f_y\, f\right](t,y) \\ &\qquad + \frac{\mu^2 h^2}{2}[f_{tt} + 2 f_{ty} + f^T f_{yy} f](t,y) + \mathcal{O}(h^3) \end{aligned}$$

Análogamente

$$\begin{aligned} \frac{y(t+h)-y(t)}{h} &= y'(t) + \frac{h}{2} y''(t) + \frac{h^2}{3!} y^{(3)}(t) + \mathcal{O}(h^3) \\ &= f(t,y) + \frac{h}{2} f^{(1)}(t,y) + \frac{h^2}{6} f^{(2)}(t,y) + \mathcal{O}(h^3) \\ &= f(t,y) + \frac{h}{2}[f_t + f_y f](t,y) \\ &\qquad + \frac{h^2}{6}[f_{tt} + 2 f_{ty} f + f^T f_{yy} f + f_y (f_t + f_y f)](t,y) + \mathcal{O}(h^3) \end{aligned}$$

Sustituyendo estas dos expresiones en la expresión del error de truncamiento, obtenemos:

$$\begin{aligned} T(t,y;h) &= \alpha_1 f + \alpha_2 \left(f + \mu h[f_t + f_y f] + \frac{\mu^2 h^2}{2}[f_{tt} + 2f_{ty}f + f^T f_{yy} f] \right) \\ &\quad - f - \frac{h}{2}[f_t + f_y f] - \frac{h^2}{6}[f_{tt} + 2f_{ty}f + f^T f_{yy} f + f_y(f_t + f_y f)] + \mathcal{O}(h^3) \\ &= (\alpha_1 + \alpha_2 - 1) f + (\alpha_2 \mu - 1/2) h[f_t + f_y f] \\ &\quad + \frac{h^2}{2}\left[(\alpha_2 \mu^2 - 1/3)(f_{tt} + 2f_{ty}f + f^T f_{yy} f) - \frac{1}{3} f_y (f_t + f_y f) \right] + \mathcal{O}(h^3) \end{aligned}$$

En esta expresión el primer coeficiente es cero pues $\alpha_1 + \alpha_2 = 1$; el segundo coeficiente es cero si $\alpha_2 \mu = 1/2$; el tercer coeficiente no puede hacerse cero, a menos que se impongan restricciones muy severas sobre f. Por lo tanto, el máximo orden alcanzable es $p = 2$ si escogemos α_1, α_2, μ, tales que

$$\alpha_1 + \alpha_2 = 1, \quad \alpha_2 \mu = \frac{1}{2}.$$

Si despejamos α_1 de la primera ecuación y μ de la segunda, se obtiene una familia de soluciones que dependen del parámetro $\alpha_2 \neq 0$:

$$\alpha_1 = 1 - \alpha_2 \quad \text{y} \quad \mu = \frac{1}{2\alpha_2}.$$

Se puede verificar muy facilmente que el método de Euler modificado se obtiene escogiendo $\alpha_2 = 1$, y el método de Heun se obtiene escogiendo $\alpha_2 = 1/2$. Otra posible elección se obtiene considerando la función principal del error (recuerdese que $\mu = 1/2\alpha_2$)

$$\tau(x,y) \quad = \quad \frac{1}{2}\left[(\frac{1}{4\alpha_2} - \frac{1}{3})(f_{tt} + 2f_{ty}f + f^T f_{yy} f) - \frac{1}{3} f_y (f_t + f_y f) \right](t,y)$$

Observese que la suma de los valores absolutos de los coeficientes se puede minimizar si escogemos $\alpha_2 = 3/4$, con lo cual se obtiene

$$\alpha_1 = \frac{1}{4}, \quad \alpha_2 = \frac{3}{4}, \quad \mu = \frac{2}{3}.$$

Con estos valores se obtiene el método:

Método RK2 óptimo

$$\begin{cases} k_1 = f(t,y) \\ k_2 = f(t + \frac{2}{3}h, y + \frac{2}{3}hk_1) \\ y_{aprox} = y(t) + \frac{h}{4}(k_1 + 3k_2) \end{cases}$$

Ejemplo 7.13. *Resolvemos de nuevo el problema del oscilador armónico simple con* $w = 2$*, utilizando el método de Euler modificado y el método de Heun con* $h = 0.1$ *en el intervalo* $0 \leq t \leq 2\pi$.

Hacemos la transformación de la ecuación de segundo orden a un sistema de dos ecuaciones de primer orden como en el ejemplo anterior. Los resultados numéricos con los métodos de Euler modificado y Heun son casi idénticos. En la Tabla 7.2 se muestra el error al utilizar ambos métodos así como la diferencia en valor absoluto de los resultados obtenidos con ambos métodos para los últimos 19 valores. Se observa una diferencia entre ambos métodos de orden 10^{-16}. En la Figura 7.9 se muestra una gráfica de la solución exacta $x(t) = \cos(2t)$ junto con las soluciones aproximadas.

t	Error Euler-mod	Error Heun	Dif. EM y Heun
4.5	3.1253742e-2	3.1253742e-2	7.7715612e-16
4.6	2.0823611e-2	2.0823611e-2	7.7715612e-16
4.7	9.0537770e-3	9.0537770e-3	8.8817842e-16
4.8	3.6095901e-3	3.6095901e-3	7.7715612e-16
4.9	1.6663189e-2	1.6663189e-2	6.6613381e-16
5.0	2.9566583e-2	2.9566583e-2	4.4408921e-16
5.1	4.1764547e-2	4.1764547e-2	2.2204460e-16
5.2	5.2710839e-2	5.2710839e-2	1.1102230e-16
5.3	6.1892396e-2	6.1892396e-2	5.5511151e-17
5.4	6.8852876e-2	6.8852876e-2	2.4980018e-16
5.5	7.3214496e-2	7.3214496e-2	4.4408921e-16
5.6	7.4697138e-2	7.4697138e-2	5.5511151e-16
5.7	7.3133807e-2	7.3133807e-2	6.1062266e-16
5.8	6.8481647e-2	6.8481647e-2	7.7715612e-16
5.9	6.0827900e-2	6.0827900e-2	8.8817842e-16
6.0	5.0390400e-2	5.0390400e-2	8.8817842e-16
6.1	3.7512415e-2	3.7512415e-2	7.7715612e-16
6.2	2.2651917e-2	2.2651917e-2	6.6613381e-16
6.3	6.3655670e-3	6.3655670e-3	6.6613381e-16

Cuadro 7.2: Comparación de los métodos de Euler modificado y Heun

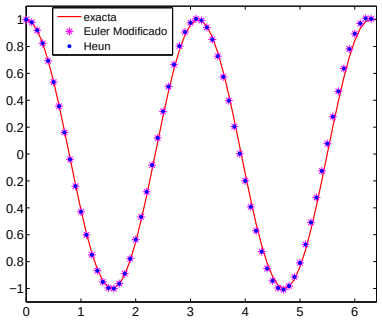


Figura 7.9: Comparación de los métodos de Euler mejorado y Heun.

7.6. Métodos de un paso de r–etapas

En forma análoga en como se introducen los métodos de dos etapas se pueden generar métodos de orden mayor. Por ejemplo, en un método de r etapas se proponen r "pendientes" de la forma

$$\begin{aligned} k_1(t,y) &= f(t,\, y), \\ k_2(t,y) &= f(t+\mu_2 h,\, y+h\lambda_{21}k_1), \\ k_3(t,y) &= f(t+\mu_3 h,\, y+h\lambda_{31}k_1+h\lambda_{32}k_2), \\ &\vdots \\ k_r(t,y) &= f(t+\mu_r h,\, y+h\lambda_{r1}k_1+h\lambda_{r2}k_2+\ldots+h\lambda_{r,r-1}k_{r-1}), \end{aligned}$$

donde los incrementos $\mu_2,\, \mu_3, \ldots, \mu_r$ se escogen para para que $\mu_2 = \lambda_{21},\, \mu_3 = \lambda_{31}+\lambda_{32}, \ldots,$ $\mu_r = \lambda_{r1}+\lambda_{r2}+\ldots+\lambda_{r,r-1}$. Posteriormente se forma un promedio ponderado de las pendientes propuestas, para obtener

$$\Phi(t,y;h) = \alpha_1 k_1 + \alpha_2 k_2 + \ldots + \alpha_r k_r$$

de tal manera que $\alpha_1+\alpha_2+\ldots+\alpha_r = 1$. Así que

$$y_{aprox} = y(t) + h\Phi(t,y;h)$$

Los parámetros introducidos se escogen en la forma adecuada para obtener el máximo orden en el error de truncamiento. Es decir de tal forma que

$$\begin{aligned} T(t,y;h) &= \Phi(t,y;h) - \frac{y(t+h)-y(t)}{h} = \sum_{i=1}^{r}\alpha_i k_i - \frac{y(t+h)-y(t)}{h} \\ &= \mathcal{O}(h^p), \quad h\to 0 \end{aligned}$$

con p máximo.

El procedimiento es más engorroso que en el caso de los métodos de dos etapas, y no intentaremos abordarlo con detalle en este escrito. Basta decir que el algoritmo general de r etapas requiere r evaluaciones de f por cada paso del tiempo. El método más popular es el denominado método de Runge-Kutta de cuarto orden. Este es un método de cuatro etapas y viene dado por

Método de Runge-Kutta clásico (r=4)

$$\begin{aligned} k_1(t,y) &= f(t,\, y), \\ k_2(t,y) &= f(t+\frac{h}{2},\, y+\frac{1}{2}hk_1) \\ k_3(t,y) &= f(t+\frac{h}{2},\, y+\frac{1}{2}hk_2) \\ k_4(t,y) &= f(t+h,\, y+hk_3) \\ y_{aprox} &= y(t)+\frac{h}{6}(k_1+2k_2+2k_3+k_4) \end{aligned}$$

Observese que en este caso

$$\Phi(t,y,;h)=\frac{1}{6}(k_1+2k_2+2k_3+k_4).$$

El método es de orden $p=4$, y cuando la función $f(t,y)$ no depende de y se puede demostrar facilmente que este método se reduce al método de Simpson para calcular $\int_{t_0}^{t} f(\tau)d\tau$. En forma análoga puede demostrarse que si $f(t,y)$ no depende explícitamente de y, el método de Euler modificado se reduce al método del punto medio para aproximar la integral de $f(t)$, y el método de Heun se reduce al método del trapecio.

A continuación se muestran los resultados numéricos que se obtienen cuando se utiliza el método de Runge-Kutta clásico con $h=0.1$, para resolver la ecuación del oscilador armónico simple con $w=2$ en $0\leq t\leq 2\pi$. En la tabla se muestran los valores de la solución numérica y el error para valores desde $t=4.5$ hasta $t=6.3$. Los resultados muestran la superioridad de este método respecto de los métodos anteriores.

t	Sol. RK-4	Error
4.5	-9.1106338e-1	6.6886305e-5
4.6	-9.7479683e-1	4.6788469e-5
4.7	-9.9966920e-1	2.3846367e-5
4.8	-9.8468894e-1	1.0861609e-6
4.9	-9.3045331e-1	2.7034469e-5
5.0	-8.3912447e-1	5.2941197e-5
5.1	-7.1434336e-1	7.7709332e-5
5.2	-5.6108451e-1	1.0024850e-4
5.3	-3.8545771e-1	1.1952256e-4
5.4	-1.9446450e-1	1.3459635e-4
5.5	4.2810183e-3	1.4467967e-4
5.6	2.0285570e-1	1.4916631e-4
5.7	3.9334320e-1	1.4766633e-4
5.8	5.6814960e-1	1.4003006e-4
5.9	7.2030612e-1	1.2636242e-4
6.0	8.4374693e-1	1.0702665e-4
6.1	9.3355101e-1	8.2637098e-5
6.2	9.8613826e-1	5.4040974e-5
6.3	9.9941230e-1	2.2289544e-5

Cuadro 7.3: Resultados con el método RK4 para el oscilador armónico simple.

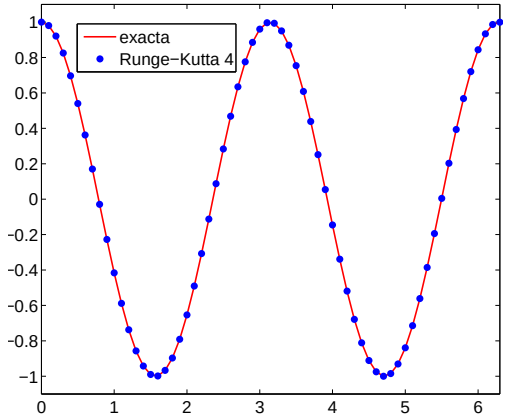


Figura 7.10: Gráfica de resultados con el método de Runge–Kutta 4.

7.7. Estabilidad y convergencia de los métodos de un paso

Para el estudio de la estabilidad y convergencia de los métodos de un paso, consideraremos el caso más general en el que el paso del tiempo puede ser variable:

$$y_{i+1} = y_i + h_i\,\Phi(t_i, y_i; h_i).$$

A continuación introducimos algunas definiciones y notación útil para el estudio de la convergencia y estabilidad de los métodos de un paso.

Mallas. Una *malla* en $[a, b]$ es un conjunto de puntos $\{t_i\}_{i=0}^{N}$ tales que

$$a = t_0 < t_1 < \ldots < t_N = b,$$

con *longitudes de malla* $h_i = t_{i+1} - t_i$, $i = 0, 1, \ldots, N-1$. A menudo se usa la letra h para señalar la colección de longitudes $\{h_i\}_{i=0}^{N-1}$. El *tamaño ó finura de la malla* se mide por medio de

$$|h| := \max_{0\le i\le N-1} h_i = \|h\|_\infty\,,$$

Si h_i es constante e igual a $(b-a)/N$ para toda $0 \le i \le N-1$, se dice que la malla es *uniforme*, de otra manera se dice que la malla es *no uniforme*. Si la malla no es uniforme se dice que el método es de *paso variable*.

Funciones de malla. La función vectorial

$$v_h = \{v_i\}_{i=0}^{N}, \quad \text{con} \quad v_i \in \mathbb{R}^n,$$

se denomina *función de malla*, y al conjunto ó colección de funciones de malla definidas sobre $[a,b]$ lo denotamos por $F_h[a,b]$. Podemos definir normas de funciones de malla. Por ejemplo, dado $v_h = \{v_i\}_{i=0^N} \in F_h[a,b]$, definimos

$$\|v_h\|_\infty = \max_{0\leq i\leq N} \|v_i\|,$$

en donde $\|v_i\|$ indica una norma vectorial de $v_i \in \mathbb{R}^n$.

Los métodos de un paso, y cualquier otro método de variable discreta, produce una función de malla $y_h = \{y(t_i)\}_{i=0}^N$ en los puntos de malla $\{t_i\}_{i=0}^N$. Es decir, dado el valor incial $y_0 \in \mathbb{R}^n$, $t_0 = a$, el método

$$t_{i+1} = t_i + h_i \tag{7.19}$$

$$y_{i+1} = y_i + h_i\Phi(t_i, y_i; h_i), \tag{7.20}$$

con $i = 0, 1, \ldots, N-1$, aproxima la solución del problema de valores inciales (7.1)–(7.2).

Operadores Residuales. Para el estudio de la estabilidad y convergencia es útil introducir los siguientes operadores residuales:

$$(Ry)(t) := y'(t) - f(t, y(t)), \quad \forall\, y \in (\mathcal{C}^{(1)}[a,b])^n.$$
$$(R_h y_h)_i := \frac{y_{i+1} - y_i}{h_i} - \Phi(t_i, y_i; h_i), \quad \forall\, y_h \in F_h[a,b]\,.$$

De la definición de estos operadores residuales, se deduce que:

Si $Ry \equiv \overrightarrow{0}$, entonces $y(t)$ resuelve la ecuación diferencial ordinaria (7.1).

Si $R_h y_h \equiv \overrightarrow{0}$, entonces $y_h = \{y_i\}_{i=0}^N$ resuelve la ecuación en diferencias (7.19)–(7.20).

Si hacemos la interpretación

$$(R_h y)(t) := \frac{y(t+h) - y(t)}{h} - \Phi(t, y; h),$$

entonces

$$(R_h y)(t) = -T(t, y; h).$$

7.7.1. Estabilidad

La estabilidad es una propiedad del esquema numérico solamente, y no tiene nada que ver con la precisión ó poder de aproximación del mismo. La estabilidad caracteriza la robustez del esquema con respecto a las perturbaciones pequeñas. Es decir, pequeños cambios en los datos , producen cambios pequeños en las soluciones numéricas. Veremos que la estabilidad del esquema junto con la consistencia del mismo, implica la convergencia de la solución numérica a la solución exacta.

Definición 7.14. *El método (7.19)–(7.20) es estable sobre* $[a,b]$ *si existe* $K > 0$*, independiente de cualquier malla* h *sobre* $[a,b]$*, tal que para cualesquier dos funciones de malla* $y_h = \{y_i\}_{i=0}^N$*,* $w_h = \{w_i\}_{i=0}^N$*, se satisface*

$$\|y_h - w_h\|_\infty \leq K(\|y_0 - w_0\| + \|R_h y_h - R_h w_h\|_\infty)$$

$\forall\, h$ *con* $|h|$ *pequeño.*

Esta definición está motivada por la siguiente propiedad: Supongase que $y_h = \{y_i\}_{i=0}^N$ es la solución de la ecuación en diferencias (7.19)–(7.20), obtenida con precisión infinita. Entonces

$$\begin{aligned} R_h y_h &= 0, \\ y(t_0) &= y_0. \end{aligned}$$

Ahora bién, cuando utilizamos una computadora o cualquier otro medio de cálculo para aplicar el esquema (7.19)–(7.20), la precisión es finita y el error de redondeo estará presente en los cálculos. Entonces, en lugar de obtener y_h obtenemos la solución en punto flotante $w_h = \{w_i\}_{i=0}^N$, la cual satisface

$$\begin{aligned} R_h w_h &= \varepsilon, \quad \text{con} \quad \varepsilon = \{\varepsilon_i\}_{i=0}^N \\ w_0 &= y_0 + \eta_0, \end{aligned}$$

en donde w_0 es la representación en punto flotante de y_0 (es decir, si $\eta_0 = (\eta_{01}, \eta_{02}, \ldots, \eta_{0n})^T$, entonces $|\eta_{0i}| \leq \varepsilon_{maq}$). Por lo tanto, la estabilidad del método (7.19)–(7.20) se satisface si existe $K > 0$ tal que

$$\|y_h - w_h\|_\infty \leq K(\|\eta_0\| + \|\varepsilon\|_\infty) \tag{7.21}$$

Esta última relación nos indica que el método (7.19)–(7.20) es estable si el cambio global en y_h es del mismo orden de magnitud que los errores residuales $\varepsilon = \{\varepsilon_i\}_{i=0}^N$ y el error inicial η_0. También observamos que $R_h w_h = \varepsilon$ significa que para $0 \leq i \leq N-1$, $(R_h w_h)_i = (\varepsilon)_i$, es decir

$$\frac{w_{i+1} - w_i}{h} - \Phi(t_i, w_i; h_i) = \varepsilon_i,$$

ó bién

$$w_{i+1} = w_i + h_i \Phi(t_i, w_i; h_i) + h_i \varepsilon_i.$$

De la última expresión se deduce que el efecto del error de redondeo se "desvanece" cuando $|h| \to 0$, pues

$$w_{i+1} \to w_i + h_i \Phi(t_i, w_i; h_i) \quad \text{cuando} \quad |h| \to 0.$$

Teorema 7.15. *El método (7.19)–(7.20) es estable si $\Phi(t,y;h)$ satisface una condición de Lipschitz con respecto a y sobre $[a,b]\times\mathbb{R}^n\times[0,h_0]$ con $h_0>0$. Es decir, si*

$$\|\Phi(t,y;h)-\Phi(t,y^*;h)\|\leq M\|y-y^*\|,\quad \forall y,\ y^*\in\mathbb{R}^n$$

Demostración. Sea $h=\{h_i\}_{i=1}^N$ una malla en $[a,b]$ y sean $y_h=\{y_i\}_{i=0}^N$, $w_h=\{w_i\}_{i=0}^N$ dos funciones de malla. Se satisface

$$\begin{aligned} y_{i+1} &= y_i+h_i\Phi(t_i,y_i;h_i)+h_i(R_hy_h)_i\\ w_{i+1} &= w_i+h_i\Phi(t_i,w_i;h_i)+h_i(R_hw_h)_i. \end{aligned}$$

Entonces

$$y_{i+1}-w_{i+1}=y_i-w_i+h_i\left[\Phi(t_i,y_i;h_i)-\Phi(t_i,w_i;h_i)\right]+h_i[(R_hy_h)_i-(R_hw_h)_i].$$

Sean $e_i=\|y_i-w_i\|$ y $d_i=\|(R_hy_h)_i-(R_hw_h)_i\|$ en alguna norma vectorial $\|\cdot\|$, y sea $\delta=\|d\|_\infty=\text{máx}_{0\leq i\leq N}\,d_i$. Debido a que Φ satisface una condición de Lipschitz y utilizando la desigualdad del triangulo, se obtiene:

$$e_{i+1}\leq(1+h_iM)e_i+h_i\delta,\qquad i=0,1,\ldots,N-1 \tag{7.22}$$

donde M es la constante de Lipschitz asociada a $\Phi(t,y;h)$. Por recurrencia de la desigualdad (7.22) obtenemos

$$e_{i+1}\leq\prod_{k=0}^{i}(1+h_kM)e_0+\left[h_0\prod_{k=1}^{i}(1+h_kM)+h_1\prod_{k=2}^{i}(1+h_kM)\right.$$
$$\left.+\cdots+h_{i-1}(1+h_iM)+h_i\right]\delta\,.$$

Se puede demostrar facilmente que cada uno de los productos es menor ó igual a $e^{(b-a)M}$ pues $1+h_kM\leq e^{h_kM}$ y $\sum_{k=1}^N h_k=b-a$. Así que

$$e_{i+1}\leq e^{(b-a)M}e_0+e^{(b-a)M}\sum_{k=0}^{i}h_k\delta\leq e^{(b-a)M}\left[e_0+(b-a)\delta\right]$$

En la última desigualdad se ultizó que $\sum_{k=0}^i h_k\leq b-a$. Por lo tanto

$$\begin{aligned}\|e\|_\infty=\|y_h-w_h\|_\infty &\leq e^{(b-a)M}\left[\|y_0-w_0\|+(b-a)\|R_hy_h-R_hw_h\|_\infty\right]\\ &\leq K\left[\|y_0-w_0\|+\|R_hy_h-R_hw_h\|_\infty\right]\end{aligned}$$

con $K=e^{(b-a)M}\,\text{máx}\{1,b-a\}$

Ejemplo 7.16. *El método de Euler es estable si $f(t,y)$ es Lipschitz continua respecto a y, dado que en el método de Euler $\Phi(t,y;h) = f(t,y)$.*

En realidad todos los métodos de un paso que se usan en la práctica satisfacen una condición de Lipschitz si $f(t,y)$ satisface una condición de este tipo. En este caso la constante de Lipschitz M para Φ puede expresarse en términos de la constante de Lipschitz L para f. Por ejemplo, para el método de Heun:

$$\Phi(t,y;h) = \frac{1}{2}(k_1 + k_2), \quad \text{con} \quad k_1 = f(t,y),\ k_2 = f(t+h, y+hk_1).$$

Entonces

$$\| \Phi(t,y;h) - \Phi(t,y^*;h) \| = \frac{1}{2} \| f(t,y) - f(t,y^*) + f(t+h, y+hk_1) - f(t+h, y^*+hk_1^*) \|$$

donde $k_1^* = f(t,y^*)$. Por la desigualdad del triángulo y la condición de Lipschitz se obtiene

$$\begin{aligned} \| \Phi(t,y;h) - \Phi(t,y^*;h) \| &\le \frac{L}{2}\|y - y^*\| + \frac{L}{2}\|y + hk_1 - y^* - hk_1^*\| \\ &\le L\|y - y^*\| + \frac{Lh}{2}\|f(t,y) - f(t,y^*)\| \le L\|y - y^*\| + \frac{L^2 h}{2}\|y - y^*\|. \end{aligned}$$

Tomando $M = L(1 + \frac{hL}{2})$ se obtiene la condición de estabilidad. □

7.7.2. Convergencia

Definición 7.17. *El método de un paso (7.19)–(7.20) es convergente si*

$$\max_{0 \le i \le N} \|y_i - y(t_i)\| \to 0 \text{ cuando } |h| \to 0,$$

es decir si $\|y_h - y\|_\infty \to 0$ cuando $|h| \to 0$.

Teorema 7.18. *Si el método (7.19)–(7.20) es consistente y estable sobre $[a,b]$, entonces el método es convergente.*

Demostración. Sea $y_h = \{y_i\}_{i=0}^N$ solución de (7.19)–(7.20), y sea $y = \{y(t_i)\}_{i=0}^N$ solución de (7.1)–(7.2) en los puntos de malla. Entonces, debido a la estabilidad

$$\|y_h - y\|_\infty \quad \le \quad K\{\|y_0 - y(t_0)\| + \|R_h y_h - R_h y\|_\infty\},$$

para alguna constante $K > 0$. Como $y_0 = y(t_0)$ y $R_h y_h = \overrightarrow{0}$, entonces

$$\|y_h - y\|_\infty \quad \le \quad K\|R_h y\| = K\|T(\cdot, y; h)\|_\infty = \mathcal{O}(|h|^p)$$

suponiendo un error de truncamiento de orden p. Claramente

$$\|y_h - y\|_\infty \to 0, \text{ cuando } |h| \to 0$$

□

Conclusión. Todos los métodos de un paso que se han considerado en esta sección son estables y convergentes, y de orden $p \geq 1$, bajo suposiciones razonables de suavidad para $f(t, y)$.

7.8. Estudio asintótico del error global

Hasta el momento solo hemos considerado el error de truncamiento para hablar de la precisión de un método de aproximación. Si el método es de orden p el error de truncamiento puede expresarse en términos de la función principal del error en la forma

$$T(t, y; h) = \tau(t, y)h^p + \mathcal{O}(h^{p+1}).$$

Como sabemos el error de truncamiento indica una medida del error por unidad de paso del tiempo. Esta claro que la distribución de este *error local* esta descrita por la función principal del error y juega el papel de término lider en el error de truncamiento. Quisieramos saber si, en forma análoga, el error global contiene un término lider que describe la distrubución del error global $\|y_h - y\|_\infty$ cuando $|h| \to 0$. Es decir, estamos interesados en el comportamiento asintótico de $y_i - y(t_i),\ i = 1, \ldots,\ N$. El siguiente teorema indica que, bajo suposiciones razonables, el término lider del error global es una función $\varepsilon(t)$ que resuelve el *problema variacional*

$$\begin{aligned} \frac{d\varepsilon}{dt} &= f_y(t, y)\varepsilon + \tau(t, y),\ a < t \leq b \\ \varepsilon(a) &= 0 \end{aligned}$$

Para simplificar la exposición consideraremos que la malla es uniforme, es decir $h_i = h$, $i = 1, \ldots,\ N$.

Teorema 7.19. *Supongase que*

a) $\Phi(t, y; h) \in \mathcal{C}^2[a, b] \times \mathbb{R}^n \times [0, h_0]$,

b) Φ *es un método de orden* $p \geq 1$ *en función del error principal* $\tau(t, y)$ *continua sobre* $[a, b] \times \mathbb{R}^n$.

c) $\varepsilon(t)$ es la solución del problema de valores iniciales.

$$\frac{d\varepsilon}{dt} = f_y(t, y(t))\varepsilon + \tau(t, y(t)),\ a < t \leq b \tag{7.23}$$

$$\varepsilon(a) = 0 \tag{7.24}$$

entonces

$$y_i - y(t_i) \quad = \quad \varepsilon(t_i)h^p + \mathcal{O}(h^{p+1}),\ \textit{cuando}\ |h| \to 0 \tag{7.25}$$

para $i = 1, \ldots, N$.

Antes de demostrar este teorema hacemos las siguientes observaciones:

1. El significado preciso de la expresión (7.25) es

   $$\|y_h - y - h^p\varepsilon\|_\infty = \mathcal{O}(h^{p+1}) \text{ cuando } |h| \to 0,$$

   donde $y = \{y(t_i)\}_{i=0}^N$, $y_h = \{y_i\}_{i=0}^N$, $\varepsilon = \{\varepsilon_i\}_{i=0}^N$ son funciones de malla y $\|\cdot\|_\infty$ se define como el valor máximo en valor absoluto de la función de malla.

2. Como $\Phi(t, y; 0) = f(t, y)$, entonces la primera suposición del teorema implica que $f \in \mathcal{C}^2$ sobre $[a, b]\times\mathbb{R}^n$, y esto último es más que suficiente para garantizar la existencia y unicidad de la solución $\varepsilon(t)$ del problema (7.23)–(7.24) sobre $[a, b]$.

Demostración. Utilizando el teorema de Taylor ó el teorema del valor medio, y la consistencia del método, se puede demostrar (Gautschi) que

$$\Phi(t_i, y_i; h_i) - \Phi(t_i, y(t_i); h_i) = f_y(t_i, y(t_i))\ \ [y_i - y(t_i)] + \mathcal{O}(h^{p+1}) \tag{7.26}$$

Para resaltar el término lider en el error global, definimos la función de malla $\xi = (y_h - y)/h^p$, es decir $\xi_i = (y_i - y(t_i))/h^p$ implica

$$\begin{aligned}
\frac{\xi_{i+1} - \xi_i}{h} &= \frac{1}{h}\left[\frac{y_{i+1} - y(t_{i+1})}{h^p} - \frac{y_i - y(t_i)}{h^p}\right] \\
&= \frac{1}{h^p}\left[\frac{y_{i+1} - y_i}{h} - \frac{y(t_{i+1}) - y(t_i)}{h}\right] \\
&= \frac{1}{h^p}\left[\Phi(t_i, y_i; h) - \{\Phi(t_i, y(t_i); h) - T(t_i, y(t_i); h)\}\right]
\end{aligned}$$

En la última igualdad se ha hecho uso de que

$$y_{i+1} = y_i + h\Phi(t_i, y_i; h) \quad \text{y} \quad T(t_i, y(t_i); h) = \Phi(t_i, y(t_i); h) - \frac{y(t_{i+1}) - y(t_i)}{h}$$

Además, si hacemos uso de la expresión del error de truncamiento en términos del error principal

$$T(t_i, y(t_i); h) = \tau(t_i, y(t_i))h^p + \mathcal{O}(h^{p+1}),$$

obtenemos, utilizando (7.26),

$$\begin{aligned} \frac{\xi_{i+1} - \xi_i}{h} &= \frac{1}{h^p}\left[\Phi(t_i, y_i; h) - \Phi(t_i, y(t_i); h) + \tau(t_i, y(t_i))h^p + \mathcal{O}(h^{p+1})\right] \\ &= f_y(t_i, y(t_i))\frac{y_i - y(t_i)}{h^p} + \tau(t_i, y(t_i)) + \mathcal{O}(h) \\ &= f_y(t_i, y(t_i))\xi_i + \tau(t_i, y(t_i)) + \mathcal{O}(h), \end{aligned}$$

$i = 1, \ldots, N$. Por lo tanto, despejando ξ_{i+1}, de la anterior ecuación, obtenemos

$$\xi_{i+1} = \xi_i + h\left[f_y(t_i, y(t_i))\xi_i + \tau(t_i, y(t_i))\right] + \mathcal{O}(h^2), \qquad \xi_0 = 0$$

la cual puede verse como el método de Euler para aproximar la ecuación (7.23). Dado que el método de Euler es estable y la función $g(t, y) = f_y(t, y(t)) + \tau(t, y(t))$ es lineal en y, ésta satisface una condición de Lipschitz. Entonces se satisface la condición

$$\|\xi - \varepsilon\|_\infty \leq K[\|\xi_0 - \varepsilon_0\| + \|R_h\xi - R_h\varepsilon\|_\infty] = K\|T(\cdot, \varepsilon; h)\| = \mathcal{O}(h)$$

debido a que $T(\cdot, \varepsilon; h)$ es el error de truncamiento en el método de Euler. Como $\xi = h^{-p}(y_h - y)$, entonces multiplicando por h^p la última desigualdad se obtiene

$$\|y_h - y + h^p\varepsilon\| = \mathcal{O}(h^{p+1}).$$

□

7.9. Monitoreo del error y control del paso

La mayoría de los programas modernos disponibles para resolver ecuaciones diferenciales ordinarias tienen integrada la posibilidad de monitorear errores locales de truncamiento y, en consecuencia, controlar la longitud del paso h sobre la base de la estimación de estos errores. Nosotros utilizaremos el resultado asintótico del teorema anterior para tratar de monitorear el error global dado por la ecuación (7.25). Observamos que para ello necesitamos calcular la matriz Jacobiana a lo largo de la trayectoria solución $f_y(t, y(t))$ y una estimación de la función principal del error $\tau(t, y(t))$, como se muestra en la ecuación (7.23).

Estimación del error global.

La idea principal para hacer una estimación del error global es hacer uso de las expresiones (7.23)–(7.25). Es decir, integrar la "ecuación variacional" (7.23) junto con la ecuación principal (7.1)–(7.2). Observese que en (7.25) necesitamos una estimación de $\varepsilon(t_i)$ con un error de orden a lo más $\mathcal{O}(h)$, dado que cualquier estimación con orden adicional sería absorbida por el término $\mathcal{O}(h^{p+1})$. Por lo tanto, $\varepsilon(t_i)$ puede obtenerse del *problema variacional* (7.23) por medio del método de Euler:

$$\varepsilon_{i+1} = \varepsilon_i + h[f_y(t_i, y_i)\varepsilon_i + \tau(t_i, y_i; h)].$$

Sin embargo tenemos un problema adicional: necesitamos calcular la función del error principal $\tau(t, y)$, lo cual generalmente no será posible. Pero, de nuevo, como solo necesitamos una estimación de primer orden ε_i, podemos sustituir $\tau(t, y)$ por una función $r(t, y\,; h)$ que satisfaga

$$r(t, y; h) = \tau(t, y) + \mathcal{O}(h), \quad h \to 0 \tag{7.27}$$

uniformemente sobre $[a, b]\times\mathbb{R}^n$. Más adelante presentaremos dos técnicas mediante las cuales se puede obtener $r(t, y; h)$.

Resumiendo, si $\Phi(t, y; h)$ satisface las condiciones del teorema 7.19 y si $r(t, y; h)$ es una estimación del error principal que satisface (7.27), generamos simultaneamente las funciones de malla $y_h = \{y_i\}_{i=0}^N$ y $\varepsilon_h = \{\varepsilon_i\}_{i=0}^N$ por medio de

$$\begin{aligned} t_{i+1} &= t_i + h && (7.28)\\ y_{i+1} &= y_i + h\Phi(t_i, y_i; h) && (7.29)\\ \varepsilon_{i+1} &= \varepsilon_i + h[f_y(t_i, y_i)\varepsilon_i + r(t_i, y_i; h)], && (7.30) \end{aligned}$$

donde $t_0 = a,\ y_0 = y(t_0)$ y $\varepsilon_0 = 0$. Entonces la *estimación del error global* de orden h^{p+1} viene dada por

$$y_i - y(t_i) = \varepsilon_i h^p, \qquad |h| \to 0$$

Estimación del error de truncamiento

Esta claro que para poder utilizar (7.28)–(7.30) debemos estimar $r(t, y; h)$ la cual es una aproximación de orden h para la función del error principal $\tau(t, y)$. A continuación describimos dos métodos que producen una estimación de $r(t, y; h)$.

1. Extrapolación local de Richardson a cero.

Funciona bién para cualquier método de un paso, pero se considera que es muy caro computacionalmente. Si Φ tiene orden p, el procedimiento es calcular

$$\begin{aligned} y_h &= y + h\Phi(t, y; h) \\ y_{h/2} &= y + \frac{h}{2}\Phi(t, y; h/2) \\ y_h^* &= y_{h/2} + \frac{h}{2}\Phi(t + h/2,\, y_{h/2}\,; h/2) \end{aligned}$$

y entonces una estimación de $r(t, y; h)$ es (Gautschi)

$$r(t, y; h) = \frac{1}{1 - 1/2^p} \frac{y_h - y_h^*}{h^{p+1}} \tag{7.31}$$

Observese que y_h^* es el resultado de aplicar Φ sobre dos pasos consecutivos de tamaño $h/2$, mientras que y_h es el resultado de una aplicación de Φ sobre el intervalo completo de tamaño h. Este procedimiento es costoso. Por ejemplo, para un método de Runge-Kutta de cuarto orden se necesitan 11 evaluaciones de f por paso, es decir casi el triple número de evaluaciones necesarias para un paso del método simple de Runge-Kutta. Por esta razón es que el método de extrapolación de Richardson normalmente se utiliza despues de cada dos pasos de Φ, es decir se calcula

$$\begin{aligned} y_h &= y + h\Phi(t, y; h) \\ y_{2h}^* &= y_h + h\Phi(t + h, y_h; h) \\ y_{2h} &= y + 2h\Phi(t, y; 2h) \end{aligned}$$

y entonces la expresión (7.31) se transforma en

$$r(t, y; h) = \frac{1}{2(2^p - 1)} \frac{y_{2h} - y_{2h}^*}{h^{p+1}} \qquad = \tau(t, y) + \mathcal{O}(h)$$

Para un método de Runge-Kutta clásico, el procedimiento necesita 7 evaluaciones de f por paso, es decir 3 evaluaciones adicionales a las necesarias para un paso simple de Runge-Kutta. Aún con este ahorro el procedimiento sigue siendo caro.

A continuación presentamos un método más eficiente.

2. Métodos de encajamiento (Runge–Kutta–Fehlberg).

Se pueden obtener métodos más eficientes para el control del paso, si en lugar de dos aproximaciones con pasos h y $h/2$ como en los métodos anteriores, se sigue la idea de Fehlberg: generar dos métodos de discretización diferentes, uno de orden p y otro de orden

$p+1$:

$$\begin{aligned} y(t+h) &\approx y(t) + h\,\Phi(t,y;h), \qquad \text{(orden } p\text{)} \\ y^*(t+h) &\approx y(t) + h\,\Phi^*(t,y;h), \qquad \text{(orden } p+1\text{)}. \end{aligned}$$

Dado que

$$\begin{aligned} \Phi(t,y;h) - \frac{y(t+h)-y(t)}{h} &= T(t,y;h) = \tau(t,y)h^p + \mathcal{O}(h^{p+1}), \\ \Phi^*(t,y;h) - \frac{y(t+h)-y(t)}{h} &= T^*(t,y;h) = \mathcal{O}(h^{p+1}), \end{aligned}$$

al restar y dividir por h^p, se obtiene

$$\frac{1}{h^p}[\Phi(t,y;h) - \Phi^*(t,y;h)] = \tau(t,y) + \mathcal{O}(h), \tag{7.32}$$

de tal forma que

$$r(t,y;h) = \frac{1}{h^p}[\Phi(t,y;h) - \Phi^*(t,y;h)] \tag{7.33}$$

es una aproximación de orden $\mathcal{O}(h)$ de la función principal del error $\tau(t,y)$ para el método de orden p con $\Phi(t,y;h)$. Para que el método valga la pena es necesario que sea más eficiente que los métodos de extrapolación. Siguiendo la idea de Fehlberg, esto se puede lograr "encajando" ó anidando un método de Runge–Kutta de orden p dentro de otro de orden $p+1$. La forma explícita de construir este tipo de procedimiento se muestra con el siguiente ejemplo.

Ejemplo 7.20. *Método de Runge–Kutta–Fehlberg con* $p = 2$ *y* $p+1 = 3$

Consideremos los métodos

$$\begin{aligned} y_{i+1} &= y_i + h\,\Phi(t_i,y_i;h), \qquad \text{con} \quad T(t,y;h) \sim \mathcal{O}(h^2), \\ y^*_{i+1} &= y_i + h\,\Phi^*(t_i,y_i;h), \quad \text{con} \quad T^*(t,y;h) \sim \mathcal{O}(h^3), \end{aligned}$$

donde Φ y Φ^* se construyen de la siguiente manera

$$\begin{aligned} \Phi(t,y;h) &= \alpha_1\,k_1(t,y;h) + \alpha_2\,k_2(t,y;h) + \alpha_3\,k_3(t,y;h), \\ \Phi^*(t,y;h) &= \alpha_1^*\,k_1(t,y;h) + \alpha_2^*\,k_2(t,y;h) + \alpha_3^*\,k_3(t,y;h) + \alpha_4^*\,k_4(t,y;h). \end{aligned}$$

Notese que en la construcción de Φ^* los valores k_1, k_2 y k_3 usados para calcular Φ se han reutilizado, por lo que solo se necesita un valor adicional k_4 para construir el segundo método

Φ^*. Las funciones k_1, k_2, k_3 y k_4 se construyen de la siguiente manera:

$$\begin{aligned} k_1 &= f(t,y), \\ k_2 &= f(t+\mu_2 h, y+h\lambda_{21}k_1), \\ k_3 &= f(t+\mu_3 h, y+h[\lambda_{31}k_1+\lambda_{32}k_2]), \\ k_4 &= f(t+\mu_4 h, y+h[\lambda_{41}k_1+\lambda_{42}k_2+\lambda_{43}k_3]), \end{aligned}$$

Las constantes μ_r, λ_{rj}, con $r = 2,\ 3,\ 4$, y $1 \leq j \leq r-1$, así como α_r, α_r^*, $r = 1,\ 2,\ 3,\ 4$, se calculan de tal forma que los errores de truncamiento sean $\mathcal{O}(h^2)$ para $\Phi(t,y;h)$ y $\mathcal{O}(h^3)$ para $\Phi^*(t,y;h)$. Este cálculo se realiza en forma análoga a como se hizo en el caso del método de Runge–Kutta de dos etapas. Si en el cálculo se utiliza la condición adicional $\mu_r = \sum_{j=1}^{r-1}\lambda_{rj}$, $r = 2,\ 3,\ 4$, se obtienen las ecuaciones siguientes para las constantes a determinar:

$$\begin{aligned} \alpha_1+\alpha_2+\alpha_3 &= 1, & \alpha_1^*+\alpha_2^*+\alpha_3^*+\alpha_4^* &= 1, \\ \mu_2\alpha_2+\mu_3\alpha_3 &= \frac{1}{2}, & \mu_2\alpha_2^*+\mu_3\alpha_3^*+\mu_4\alpha_4^* &= \frac{1}{2}, \\ \mu_2^2\alpha_2^*+\mu_3^2\alpha_3^*+\mu_4^2\alpha_4^* &= \frac{1}{3}, & \mu_2\lambda_{32}\alpha_3^*+(\mu_2\lambda_{42}+\mu_3\lambda_{43})\alpha_4^* &= \frac{1}{6}. \end{aligned}$$

Este sistema de ecuaciones tiene una infinidad de soluciones, y se pueden imponerse restricciones adicionales para reducir el costo del método. Por ejemplo, puede pedirse que k_4 en el i–ésimo paso sea reusable como k_1 en el $(i+1)$–ésimo paso. Es decir, podemos imponer

$$f(t+h, y+h\Phi) \equiv f(t+\mu_4 h, y+h[\lambda_{41}k_1+\lambda_{42}k_2+\lambda_{43}k_3]),$$

lo cual produce las restricciones adicionales

$$\mu_4 = 1, \quad \lambda_{41} = \alpha_1, \quad \lambda_{42} = \alpha_2, \quad \lambda_{43} = \alpha_3.$$

Se pueden imponer restricciones adicionales para mínimizar los coeficientes de la función principal del error. Sin embargo, no consideramos esto en detalle. En lugar de esto, escribimos a continuación un conjunto de valores para las constantes:

r	μ_r	λ_{r1}	λ_{r2}	λ_{r3}	α_r	α_r^*
1	0	0	– –	– –	214/891	533/2106
2	1/4	1/4	– –	– –	1/33	0
3	27/40	−189/800	729/800	– –	650/891	800/1053
4	1	214/891	1/33	650/891	– –	−1/78

Con estas constantes se obtiene el método siguiente al cual denominaremos *RKF–23*:

$$k_1 = f(t_i, y_i), \tag{7.34}$$

$$k_2 = f(t_i + \frac{h}{4}, y_i + \frac{h}{4}k_1), \tag{7.35}$$

$$k_3 = f(t_i + \frac{27}{40}h, y_i - \frac{189}{800}hk_1 + \frac{729}{800}hk_2), \tag{7.36}$$

$$k_4 = f(t_i + h, y_i + \frac{214}{819}hk_1 + \frac{1}{33}hk_2 + \frac{650}{891}hk_3), \tag{7.37}$$

$$y_{i+1} = y_i + \frac{h}{891}(214k_1 + 27k_2 + 650k_3), \tag{7.38}$$

$$y^*_{i+1} = y_i + \frac{h}{2106}(533k_1 + 1600k_3 - 27k_4). \tag{7.39}$$

Observese que debido a que

$$k_4(t_i, y_i; h) = f(t_i + h, y_i + h\Phi(t_i, y_i; h)) = f(t_{i+1}, y_{i+1}) = k_1(t_{i+1}, y_{i+1}; h)$$

entonces solo se requieren tres evaluaciones de f por paso.

Control del paso

El control del paso h se obtiene de la siguiente forma

$$y_{i+1} - y^*_{i+1} = h[\Phi(t_i, y_i; h)) - \Phi^*(t_i, y_i; h))] = h^3 r(t_i, y_i; h),$$

(ver ecuación (7.33) con $p = 2$). Supongase que queremos que el error global sea menor a un valor dado ϵ (tolerancia). Para que esto ocurra se debe cumplir que

$$\|y_{i+1} - y^*_{i+1}\| = h^3 \|r(t_i, y_i; h)\| \leq \epsilon\,. \tag{7.40}$$

Para que el nuevo paso del tiempo h_{new} sea exitoso, es decir produzca un error menor a ϵ, se debe satisfacer

$$\|r(t_{i+1}, y_{i+1}; h)\|\, h^3_{new} \leq \epsilon\,. \tag{7.41}$$

Pero, salvo errores de orden h,

$$r(t_{i+1}, y_{i+1}; h) \approx r(t_i, y_i; h) = \frac{\|y_{i+1} - y^*_{i+1}\|}{h^3}$$

por (7.40). Sustituyendo esta expresión en (7.41), obtenemos

$$\frac{\|y_{i+1} - y^*_{i+1}\|}{h^3} h^3_{new} \leq \epsilon$$

Por lo tanto, para garantizar que el error global sea menor a ϵ en cada paso debemos escoger

$$h_{new} \approx h \left(\frac{\epsilon}{\|y_{i+1} - y^*_{i+1}\|} \right)^{1/3}$$

Para el caso general, cuando las aproximaciones y_{i+1} y y^*_{i+1} se han realizado con métodos de orden p y $p+1$ respectivamente, se tiene

$$h_{new} \approx h \left(\frac{\epsilon}{\|y_{i+1} - y^*_{i+1}\|} \right)^{1/(p+1)} \tag{7.42}$$

Muchos autores, basados en una amplia experiencia numérica, recomiendan hacer el siguiente ajuste

$$h_{new} \approx q\, h$$

con

$$q = \alpha \left(\frac{\epsilon h}{\|y_{i+1} - y^*_{i+1}\|} \right)^{1/p} \quad \text{y} \quad \alpha \approx 0.9$$

Ejemplo 7.21. *Ecuación de Van der Pol.*

$$\begin{aligned} x''(t) - c(1 - x(t)^2)\, x'(t) + x(t) = 0, \quad 0 < t \leq 10\pi\,, \\ x(0) = 1\,, \\ x'(0) = 0. \end{aligned}$$

Con $c = 0$, se obtiene la ecuación del oscilador armónico simple. Si $c \neq 0$, la ecuación es no lineal. Primero, transformamos la ecuación en un sistema de dos ecuaciones diferenciales ordinarias de primer orden.

$$\begin{aligned} y_1(t) = x(t) \quad &\Rightarrow \quad y_1'(t) = y_2(t) \\ y_2(t) = x'(t) \quad &\Rightarrow \quad y_2'(t) = c(1 - y_1(t)^2) y_2(t) - y_1(t) \end{aligned}$$

En notación matricial se puede escribir

$$\begin{bmatrix} y_1' \\ y_2' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & c \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \begin{bmatrix} 0 \\ c y_1^2 y_2 \end{bmatrix}, \quad y_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Se resuelve numéricamente esta ecuación diferencial no lineal, para el caso $c = 3$, utilizando el método *RKF–23* (7.34)–(7.39) con tolerancia $\epsilon = 0.01$. La Figura 7.11 muestra las gráficas de $y_1(t) = x(t)$ y $y_2(t) = x'(t)$ contra t. La Figura 7.12 muestra el *retrato fase* de la solución,

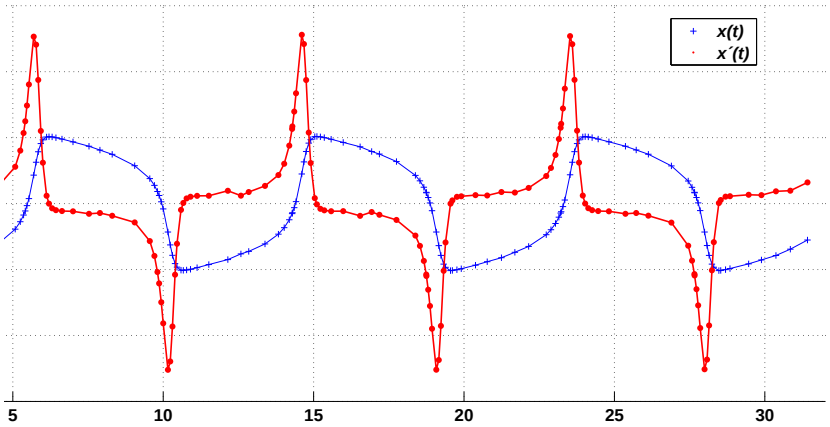


Figura 7.11: Solución de la ecuación de Van der Pol. Método *RKF–3* con $\epsilon = 0.01$

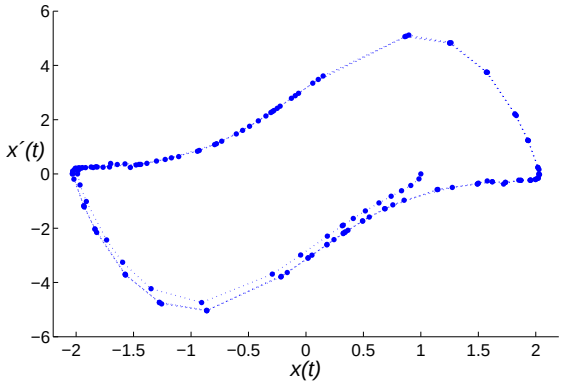


Figura 7.12: Retrato fase de la solución de la ecuación de Van der Pol.

es decir la gráfica de $x(t)$ contra $x'(t)$. Se observa que el retrato fase de la solución representa un ciclo límite no simple y la solución es períodica. En ambas figuras se observa claramente el tamaño variable de los pasos de tiempo. El número de pasos del tiempo en este ejemplo fué de 156. Con un método de paso fijo se toma $h = 0.1$ para obtener una precisión equivalente y se requieren $10\pi/0{,}1 \sim 314$ pasos del tiempo, es decir cerca del doble. Con el objeto de obtener una mejor solución, escogemos la tolerancia del error como $\epsilon = 10^{-4}$, y repetimos el cálculo. Las Figuras 7.13 y 7.14 muestran la solución. En este caso se necesitan 501 pasos del tiempo. Con un método de paso fijo se necesitarían 3142 pasos del tiempo para obtener la misma precisión, es decir más de 6 veces que con paso variable.

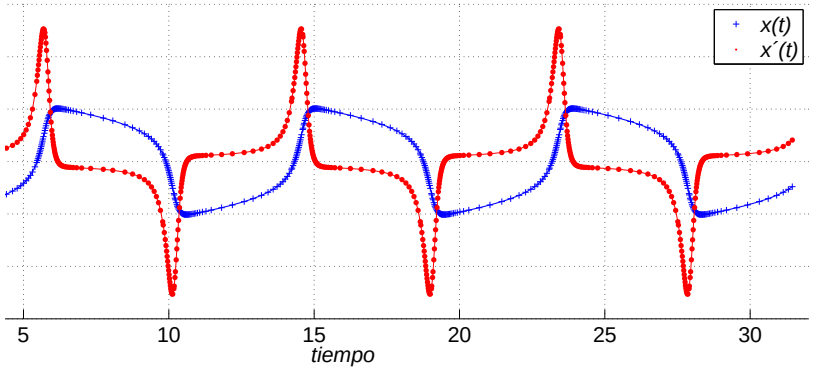


Figura 7.13: Solución con mayor precisión: $\epsilon = 10^{-4}$.

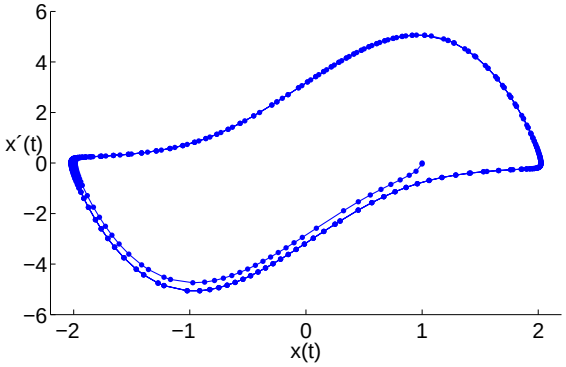


Figura 7.14: Retrato fase correspondiente.

7.9.1. Método Runge–Kutta–Fehlberg de cuarto orden

Este es un método que también es muy usado en la práctica. En este caso se combinan un método de orden $p = 4$ con uno de orden $p + 1 = 5$. Evitando los detalles de deducción

del método, solamente presentamos a continuación este método de paso variable:

$$
\begin{aligned}
k_1 &= f(t_i,\, y_i) \\
k_2 &= f\left(t_i + \frac{h}{4}\,, y_i + \frac{h}{4}k_1\right) \\
k_3 &= f\left(t_i + \frac{3h}{8},\, y_i + \frac{3}{32}hk_1 + \frac{9}{32}hk_2\right) \\
k_4 &= f\left(t_i + \frac{12h}{13},\, y_i + h\left[\frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right]\right) \\
k_5 &= f\left(t_i + h,\, y_i + h\left[\frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right]\right) \\
k_6 &= f\left(t_i + \frac{h}{2},\, y_i + h\left[-\frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right]\right) \\
y_{i+1} &= y_i + h\left(\frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5\right) \\
y^*_{i+1} &= y_i + h\left(\frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6\right)
\end{aligned}
$$

El paso de tiempo nuevo en cada iteración se calcula por medio de

$$h_{new} = 0.84\, h\left(\frac{\epsilon h}{\|y_{i+1} - y^*_{i+1}\|}\right)^{1/4}$$

donde ϵ es la tolerancia deseada para el error global. En ocasiones, en lugar de calcular y^*_{i+1} se calcula directamente la diferencia $e_{i+1} = \|y_{i+1} - y^*_{i+1}\|$. Por ejemplo, en el caso del método *RKF-45* (Runge–Kutta–Fehlberg de cuarto orden), se tiene

$$e_{i+1} = h\left\|\frac{1}{360}k_1 - \frac{128}{4275}k_3 - \frac{2197}{75240}k_4 + \frac{1}{50}k_5 + \frac{2}{55}k_6\right\|$$

Ejemplo 7.22. *Se resuelve la ecuación de Vander Pol con* $c = 3$ *utilizando el método* RKF–45 *con una tolerancia* $\epsilon = 10^{-4}$. *Las Figuras 7.15 y 7.16 muestran las gráficas de los resultados obtenidos. En este caso se realizaron 288 pasos de tiempo para obtener la solución.*

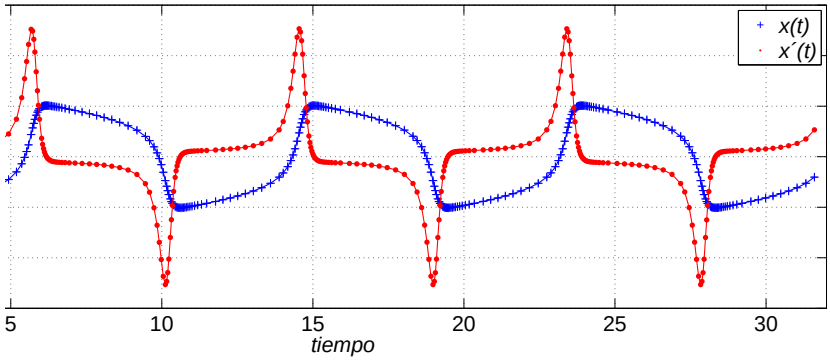


Figura 7.15: Solución con el método *RKF–45*.

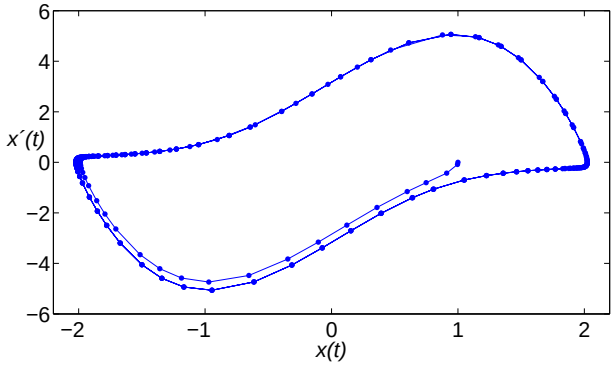


Figura 7.16: Retrato fase correspondiente.

Bibliografía

[1] E. Isaacson, H. B. Keller, *Analysis of Numerical Methods*, Wiley 1966.(Dover en 1994).

[2] W.Gautschi, *Numerical Analysis: An Introduction*, Birkhauser, 1998.

[3] L. N. Trefethen, D. Bau III, *Numerical Linear Algebra*, SIAM, 1997.

[4] S. D. Conte, C. de Boor, *Elementary Numerical Analysis*, Mc. Graw-Hill 3rd. Ed. 1980

[5] J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag 1st. Ed. 1980, 2nd. Ed. 199, 3rd. Ed. 2002.

[6] Cleve Moler, *Numerical Computing Whit Matlab*, Disponible en linea: http://www.mathworks.com/moler/index.html

[7] J. M. Ortega, *Numerical Analysis: A Second Course, Classics in Applied Mathematics*, SIAM, 1990.

[8] P. Henrici, *Elements of Numerical Analysis*, Wiley 1964.

[9] A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics*, Text in Applied Mathematics 37, Springer, 2000.

[10] E. Suli, D. F. Mayers, *An Introduction to Numerical Analysis*, CUP, 2003.

[11] S. Ross, *Simulation*, Academic Press, 2nd. Ed. 1997.

[12] G.Golub, C. Van Loan, *Matrix Computation*, John Hopkins University Press, 2nd. Ed. 1989, 3rd. Ed. 1996.

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press. Gratis versiones para F77, F90 y C en http://www.lanl.gov/numerical/index.html

[14] G. Engeln-Mullges, F. Uhlig, *Numerical Algorithms whit Fortran*, Springer-Verlag, Berlin, 1996. QA297 E5.6613

[15] Hammerlin, Gunther-Karl-Heinz Hoffmann, *Numerical Mathematics*, Springer-Verlag. QA297 H2.53

[16] R. Burden, J. D. Faires, *Análisis Numérico*, Grupo Editorial Iberoamericana, 2a. Ed. 1996.

[17] K. E. Atkinson, *An introduction to Numerical Analysis*, Wiley 1st Ed.1978, 2nd. Ed. 1989.

[18] D. Kincaid, W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, Books/Cole Pub. Co., 2nd Ed. 1996.