Apuntes de teórico - Métodos Numéricos

Instituto de Matemáticas y Estadística "Rafael Laguardia"
Facultad de Ingeniería, Universidad de la República
2016 - Montevideo, Uruguay

Nota importante: El presente material forma parte de una versión **en proceso de revisión** de un texto teórico para la asignatura "Métodos Numéricos". Por lo tanto, para preparar los exámenes se debe utilizar el material de las clases teóricas y la **bibliografía recomendada**.

Índice general

Capítulo 1

Errores y Representación

1.1. Aritmética de Punto Flotante y Errores

Consideremos un proceso o sistema real (**PR**) del cual se desea conocer el comportamiento de un determinado parámetro (**x**) conociendo información de base (**d**). Consideraremos dos formas de resolver problemas de este tipo:

- **Experimentación.** A través de la realización de experimentos es posible obtener valores reales del comportamiento que se desea estudiar. A pesar de esto, en muchos casos, la experimentación resulta costosa (ensayos de materiales, inspección de recursos naturales) en otros casos, se tardaría demasiado (políticas económicas, sistemas biológicos).
- **Modelamiento computacional.** utilizando herramientas de las ciencias exactas es posible formular problemas matemáticos que modelan o describen el comportamiento de dichos sistemas. A través del uso de *métodos numéricos* es posible resolver estos problemas y así, predecir el comportamiento de los sistemas, aunque obteniendo un error con respecto al proceso real en todos los casos.

En la Figura 1.1 se representan los dos métodos propuestos.

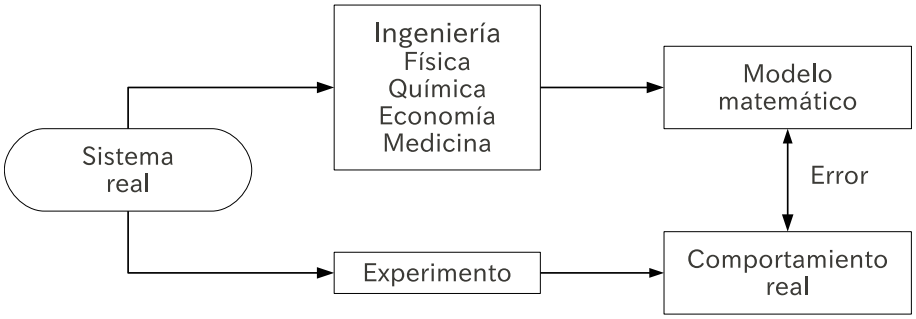

Figura 1.1: Simulación vs. Experimentación

Errores Al resolver problemas reales utilizando modelos computacionales siempre existen errores en la solución. Algunas de las fuentes de error mas habituales son las siguentes:

- Errores de medición en los datos de entrada del modelo (afectando el modelo y la calidad de las soluciones).
- Errores en el modelo matemático utilizado (considerando hipótesis que no se adapten a la realidad).
- Errores en el método numérico que resuelve el problema matemático (error de truncamiento, convergencia, etc).
- Errores en las operaciones de punto flotante (error computacional).
- Error Humano (errores en programación) y de Máquina (defectos de hardware).

En este capítulo nos centraremos en los errores de representación de punto flotante y de operaciones entre números representados en punto flotante.

1.2. Error de Operaciones de Máquina: Aritmética de Punto Flotante

Para resolver los problemas anteriormente mencionados, es necesario operar con números reales. Sea cual fuere el computador utilizado, la cantidad de números representables es finita, por lo tanto, siempre va a existir el error debido a la propia *representación* o almacenamiento del número. Al almacenar un número x podremos guardar algún número próximo a éste de los cuales la máquina pueda representar. Para poder trabajar en este tema, recordemos el Sistema Posicional de números reales.

Definición 1.2.1 (Sistema Posicional)**.** Sea una base $\beta \in \mathbb{N}$ con $\beta \geq 2$, y sea x un número real, entonces puede ser escrito como

$$x = d_n \cdot \beta^n + d_{n-1} \cdot \beta^{n-1} + \cdots + d_0 \cdot \beta^0 + d_{-1} \cdot \beta^{-1} + \ldots \qquad d_i \in \mathbb{N} \quad 0 \leq d_i \leq \beta - 1 \, i = 0 \ldots n$$

siendo esta espresión única (salvo excepciones como $0{,}999\cdots = 1$).

Particularmente utilizaremos los sistemas decimal y binario correspondientes a β igual a 10 y 2, respectivamente.

Sistema decimal Este es el sistema utilizado habitualmente para representar números. El valor de la base es 10, como se muestra a continuación:

$$5432,05 = 5 \cdot 10^3 + 4 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0 + 0 \cdot 10^{-1} + 5 \cdot 10^{-2}$$

La cantidad de bits almacenables de un computador es múltiplo de 2, por lo tanto es razonable también estudiar el sistema que utiliza base 2, el cual es llamado sistema binario.

suma (+)	producto (·)
$0+0=0$	$0\cdot 0=0$
$1+0=1$	$1\cdot 0=0$
$0+1=1$	$0\cdot 1=0$
$1+1=10$	$1\cdot 1=1$

Cuadro 1.1: Tabla de operaciones binarias

Sistema binario En este sistema de representación, el conjunto de dígitos se reduce a 0 y 1. Recordemos que se deben redefinir las operaciones suma y producto las cuales están dadas por la tabla 1.1.

Conversión de sistema de representación

A través de un ejemplo veremos cómo realizar la conversión entre distintos sistemas para representar el mismo número.

Ejemplo 1.2.1 (Conversión de sistema decimal a binario)**.** Obtendremos la representación binaria del número $176,524$.

$$(176,524)_{10} \longrightarrow (?)_2$$

parte entera: comenzamos por convertir la parte entera dividiendo por dos.

divisor	dividendo	cociente	resto
176	2	88	0
88	2	44	0
44	2	22	0
22	2	11	0
11	2	5	1
5	2	2	1
2	2	1	0

tomamos el último cociente, luego todos los restos ascendiendo hasta llegar al último resto. En este caso obtendríamos 10110000. Esto es equivalente a:

$$\begin{aligned}(176)_{10} &= 2\cdot 88 = 2^2\cdot 44 = 2^3\cdot 22 = 2^4\cdot 11\\ \ldots &= 2^4 + 2^4\cdot 10 = 2^4 + 2^5\cdot 5\\ \ldots &= 2^4 + 2^5 + 2^5\cdot 4 = 2^4 + 2^5 + 2^7\\ (176)_{10} &= (10110000)_2\end{aligned}$$

parte fraccional: ahora consideramos la parte fraccional y la multiplicamos por 2 reiteradamente, definiendo los dígitos binarios en función de que los resultados sean o no mayores que 1, como se describe en las ecuaciones siguientes:

$$(0,524)_{10} \longrightarrow (?)_2$$

$$\begin{aligned}
0,524 \cdot 2 = 1,048 \geq 1 &\Rightarrow d_{-1} = 1 \\
0,048 \cdot 2 = 0,096 \leq 1 &\Rightarrow d_{-2} = 0 \\
0,096 \cdot 2 = 0,192 \leq 1 &\Rightarrow d_{-3} = 0 \\
0,192 \cdot 2 = 0,384 \leq 1 &\Rightarrow d_{-4} = 0 \\
0,384 \cdot 2 = 0,768 \leq 1 &\Rightarrow d_{-5} = 0 \\
0,768 \cdot 2 = 1,536 \geq 1 &\Rightarrow d_{-6} = 1 \\
&\vdots \\
(0,524)_{10} &= (0,100001...)_2
\end{aligned}$$

Por lo tanto obtenemos la conversión del número:

$$(176,524)_{10} \longrightarrow (10110000,100001\ldots)_2$$

△

1.2.1. Representación de punto fijo

Los números reales con representación en **Punto fijo** en base β presentan un número fijo de decimales y están dados por la siguiente expresión:

$$\mathrm{PF}(x) = (-1)^s \cdot d_n\, d_{n-1} \ldots d_0\,,\, d_{-1}\, d_{-2} \ldots d_{-m}$$

donde:

- s: parámetro del signo, pudiendo valer 1 o 0.
- $0 \leq d_i \leq \beta - 1$.

Por tanto, si el sistema es binario, se utilizarán $n + m + 2$ bits para esta representación.

1.2.2. Representación de punto flotante

Los números reales con representación en **Punto flotante normalizado** en base β están dados por la siguiente expresión:

$$\mathrm{PF}(x) = (-1)^s \cdot 0, \underbrace{a_1\, a_2 \ldots a_p}_{\text{mantisa: } m} \cdot \beta^e = (-1)^s \cdot m \cdot \beta^{e-p}$$

donde:

- $0 \leq a_i \leq \beta - 1$.
- $L \leq e \leq U$.
- s: parámetro del signo, pudiendo valer 1 o 0.

- m: mantisa.
- e: exponente.

La normalización viene en el hecho de que $\beta^{-1} \leq m < 1$, evitando que los números tengan distintas representaciones posibles.

Si β es 2, m tendrá p bits asignados, e tendrá q bits asignados, y s tendrá 1 bit asignado, por lo que la suma de bits asignados será $N = 1 + q + p$. Al variar los bits asignados para cada uno de estos parámetros obtenemos distintos grados de precisión y rangos de números representables.

La Norma *IEEE 754* del año 1985, establece un estándar para la representación de números en punto flotante. En la misma se definen dos tipos de precisión: precisión simple (32-bits) y precisión doble (64-bits). En el cuadro 1.2 se describen ambos sistemas. La norma define también otras variantes de los mismos que no presentaremos aquí. Veamos cuál es el rango de

Precisión	N	s	p	q
simple	32	1	23	8
doble	64	1	52	11

Cuadro 1.2: tipos de precisión según *IEEE 754*

números representables normalizados para cada uno de estos sistemas de representación. Para ello analizamos cuál es el rango válido para exponente y mantisa. En el caso de precisión simple, por ejemplo, el exponente se almacena en 8 bits binarios, por lo que podemos almacenar

$$2^8 = 256 \text{ números}$$

de esta forma no es posible representar exponentes negativos, por lo que se resta 127 al número almacenado obteniendo un rango viable para representar números entre 0 y 1 fácilmente.

Para el caso del exponente tenemos:

$$E = e - (2^{q-1} - 1) = e - d \qquad d = 2^{q-1} - 1$$

se reservan $e = 00\ldots0$ y $e = 11\ldots1$
Rangos números normalizados para el exponente:

$$e_{min} = -2^{q-1} + 2 \qquad e_{max} = 2^{q-1} - 1$$

Precisión	e_{min}	e_{max}
simple	-126	127
doble	-1022	1023

Ahora vemos que para obtener el mínimo real normalizado:

$$Real_{min} = 1,0\ldots0 \cdot 2^{e_{min}}$$

Precisión	$\text{Real}_{min}(2)$	$\text{Real}_{min}(10)$
simple	$1 \cdot 2^{-126}$	$1,2 \cdot 10^{-38}$
doble	$1 \cdot 2^{-1022}$	$1,8 \cdot 10^{-308}$

y para obtener el máximo real normalizado:

$$Real_{max} = 1,11\ldots1 \cdot 2^{e_{max}}$$

Precisión	$\text{Real}_{max}(2)$	$\text{Real}_{max}(10)$
simple	$1,1\ldots1 \cdot 2^{127}$	$3,4 \cdot 10^{38}$
doble	$1,1\ldots1 \cdot 2^{1023}$	$1,8 \cdot 10^{308}$

En la figura 1.2 podemos ver un esquema de la distribución de los números reales representables normalizados próximos al número uno.

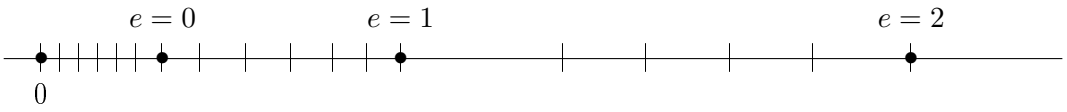

Figura 1.2: Distribución de números reales representables

Cero: se usa $e = 00\ldots0$ y mantisa nula $a_i = 0 \qquad i = 1\ldots p.$

$$0 = (s\ \underbrace{00\ldots0}_{m}\ \underbrace{00\ldots0}_{e} \rightarrow \begin{cases} s = 0 & \Rightarrow +0 \\ s = 1 & \Rightarrow -0 \end{cases}$$

Desnormalizados: se usa $e = 00\ldots0$ y mantisa no nula $\exists a_i \neq 0, \qquad i = 1\ldots p$. Son de la forma:

$$x_d = (-1)^s \cdot 0, a_1\, a_2\, \ldots\, a_p \cdot 2^{e_{min}}$$

$$Real_{min}(\text{desnormalizados}) = 0,0\ldots1 \cdot 2^{e_{min}}$$

Precisión	$\text{Real}_{min}(2)$(des.)	$\text{Real}_{min}(10)$(des.)
simple	$1 \cdot 2^{-127-23}$	$1,4 \cdot 10^{-45}$
doble	$1 \cdot 2^{-1022-52}$	$4,9 \cdot 10^{-324}$

Extiende el rango de representación próximo a cero pero con precisión limitada.

1.2.3. Aproximación de reales a punto flotante:

Redondeo y truncamiento: Como hemos visto, los sistemas de representación pueden representar una cantidad finita de números, por lo tanto al desear representar un número real que no esté incluido en el mismo, el computador deberá aproximarlo a otro. Para esta aproximación existen dos métodos habitualmente usados:

- **Redondeo:** aproxima el real al número representable más cercano. Si está equidistante, se aproxima al que tiene el dígito menos representativo igual a 0.
- **Truncamiento:** aproximo el real al número representable menor más próximo.

Límites de representación: Existen casos particulares de aproximación cuando el número a aproximar está fuera del rango abarcado por el sistema.

- **Overflow:** $x \in \mathbb{R}$ es mayor en magnitud que el mayor número representable por el sistema. Se considera como $x =$ **Inf** y es almacenado con mantisa nula, $s = 0$ y exponente $e = 11\ldots1$. Sucede lo mismo cuando si x es menor que el número mas negativo representable, y se almacena como **-Inf**.
- **Underflow:** $x \in \mathbb{R}$ es no nulo pero tiene menor magnitud que cualquier número representable (es muy próximo a cero).

1.2.4. Épsilon de máquina

Definición 1.2.2 (Épsilon de máquina)**.** Llamaremos *épsilon de máquina* (ε_{mach}) a la separación entre los números 1 y el siguiente número representable de un sistema de punto flotante.

En Octave y Matlab existe la función *eps*, la cual nos da el valor ε_{mach} para precisión doble ($2,22 \cdot 10^{-16}$) y simple ($1,19 \cdot 10^{-7}$) (ver help eps). Realice en Octave las siguientes operaciones:

```
>> a = 1 + 1.10 e-16 ;     [Enter]
>> a - 1                   [Enter]
   ans = 0                 [ a es exactamente 1 en PF]

>> a = 1 + 1.12 e-16 ;     [Enter]
>> a - 1                   [Enter]
   ans = 2.2204e-16        [ a es distinto a 1 en PF]

>> eps                     [Enter]
   ans = 2.2204e-16        [el epsilon de maquina]
```

podemos verificar de esta forma que Octave utiliza redondeo ya que para pasar de 1 al siguiente número debemos sumarle $\varepsilon_{mach}/2$.

1.3. Errores absolutos y relativos

Sea $\|.\|$ la norma euclideana en $\mathbb{R}^n$, consideremos las siguientes definiciones.

Definición 1.3.1 (Error absoluto)**.** Sean $\mathbf{x} \in \mathbb{R}^n$ un vector de valores incógnita o desconocido y $\bar{\mathbf{x}} \in \mathbb{R}^n$ una aproximación de $\mathbf{x}$. Definimos el error absoluto de $\mathbf{x}$ ($\Delta_{\mathbf{x}}$) como la norma de la diferencia entre estos valores:

$$\Delta_{\mathbf{x}} = \|\mathbf{x} - \bar{\mathbf{x}}\|$$

Definición 1.3.2 (Error relativo)**.** Sea $\mathbf{x} \in \mathbb{R}^n$ un vector de valores incógnita o desconocido y $\bar{\mathbf{x}} \in \mathbb{R}^n$ una aproximación de $\mathbf{x}$. Definimos el error relativo de $\mathbf{x}$ ($\delta_{\mathbf{x}}$) como la norma de la diferencia entre estos valores sobre la norma de $\mathbf{x}$:

$$\delta_{\mathbf{x}} = \frac{\|\mathbf{x} - \bar{\mathbf{x}}\|}{\|\mathbf{x}\|} \qquad \mathbf{x} \neq \vec{0}$$

1.4. Error de representación en Punto Flotante

Dado un valor real $x \in \mathbb{R}$ y su representación normalizada $PF(x)$ con p dígitos luego de la coma

$$x = 1, a_1 a_2 \dots a_p a_{p+1} \cdots \cdot 2^{exp} \qquad PF(x) = 1, a_1\, a_2 \dots\, a'_p \cdot 2^{exp}.$$

Para calcular el error de la aproximación de punto flotante, aplicaremos la definición de error relativo

$$\delta_x = \frac{|PF(x) - x|}{|x|} = \frac{|1, a_1\, a_2 \dots\, a_p\, a_{p+1} - 1, a_1\, a_2 \dots\, a'_p| \cdot 2^{exp}}{|1, a_1\, a_2 \dots\, a_p\, a_{p+1}| \cdot 2^{exp}}$$

simplificamos

$$\delta_x = \frac{|0, 0\, 0 \dots\, (a_p - a'_p)\, a_{p+1} \dots|}{1} \leq \varepsilon_{mach} < 2^{-p-1}$$

por lo tanto,

$$\frac{|PF(x) - x|}{|x|} \leq \varepsilon_{mach}.$$

Proposición 1.4.1. *Sea $x \in \mathbb{R}$ un real y $PF(x)$ su representación, entonces*

$$PF(x) = x(1 + \delta_x) \quad |\delta_x| \leq \varepsilon_{mach}$$

Demostración.

$$\frac{|PF(x) - x|}{|x|} = \frac{|x(1 + \delta_x) - x|}{|x|} = \frac{|x\delta_x|}{|x|} = |\delta_x| \leq \varepsilon_{mach}$$

□

1.4.1. Operaciones en punto flotante

La aritmética en punto flotante no es asociativa, ni tampoco distributiva. Las operaciones se hacen por etapas y en cada operación se aplica el redondeo correspondiente por lo tanto el resultado es alterado al cambiar los factores. Veamos un ejemplo.

Ejemplo 1.4.1 (Operaciones en Octave)**.** Realice las siguientes operaciones en Octave:

```
 >> (1 + 1.1e-16) + 1.1e-16
ans = 1

 >> 1 + (1.1e-16 + 1.1e-16)
ans = 1.0000
```

podemos concluir que en el primer caso, en el paréntesis se obtiene 1 como resultado y luego 1 nuevamente. En el segundo caso dentro del paréntesis se obtiene el épsilon de máquina por lo tanto al ser sumado a 1 se obtiene el NPF siguiente a 1. △

Al analizar el costo de la ejecución de algoritmos debemos contar la cantidad de operaciones que se realizan, por lo tanto definiremos una unidad de conteo de las mismas.

Definición 1.4.1 (*flop*)**.** Denotaremos por *flop* a una simple operación de punto flotante (suma, resta, producto o división).

En el caso de un producto escalar de dos vectores de n elementos, la cantidad de *flops* es igual a $2n-1$.

1.4.2. Error al aproximar con números representables

Sean las operaciones $+, \times, -$ y $/$ para denominador no nulo, se cumple: $x \in \mathbb{R}, x = PF(x)$ $y \in \mathbb{R}, y = PF(y)$

$$PF(x \circ y) = (x \circ y)\,(1 + \delta_{x+y}) \qquad |\delta_{x+y}| \leq \varepsilon_{mach}$$

siendo $\circ$ alguna de las operaciones consideradas. Esto se debe a que las máquinas operan con más precisión que la utilizada en la representación.

1.4.3. Error al aproximar con números reales

Sean x e y dos números reales con su respectivas representaciones y errores de representación

$$x \in \mathbb{R}, \quad PF(x) = x(1+\delta_x), \quad |\delta_x| \leq \varepsilon_{mach}$$

$$y \in \mathbb{R}, \quad PF(y) = y(1+\delta_y), \quad |\delta_y| \leq \varepsilon_{mach}$$

calcularemos cual es el error cometido al operar.

Suma

$$\delta_+ = \frac{|x+y-(PF(x)+PF(y))|}{|x+y|} = \frac{|x\,\delta_x + y\,\delta_y|}{|x+y|} = \frac{x\,|\delta_x| + y\,|\delta_y|}{x+y}$$

por lo tanto

$$\delta_+ \leq \frac{x+y}{x+y}\,\varepsilon_{mach} \leq \varepsilon_{mach}$$

vemos que el error al sumar está acotado y su cota es igual a la de la representación de punto flotante.

Resta

$$\delta_- = \frac{|x-y-(PF(x)-PF(y))|}{|x-y|} = \frac{|x\,\delta_x - y\,\delta_y|}{|x-y|} = \frac{x\,|\delta_x| + y\,|\delta_y|}{x-y}$$

por lo tanto

$$\delta_- \leq \frac{x+y}{|x-y|}\,\varepsilon_{mach}$$

este error no está acotado si $x \approx y$

$$\delta_- \leq \frac{2x}{|x-y|}\,\varepsilon_{mach} \leq \frac{2\varepsilon_{mach}}{|1-y/x|}$$

Por lo tanto al realizar resta de números muy próximos podemos obtener un error grande y no detectarlo. Este fenómeno lleva el nombre de cancelación catastrófica.

Ejemplo 1.4.2 (Cancelación catastrófica)**.** Al querer calcular la solución de la ecuación

$$x^2 - 56x + 1 = 0 \quad \Rightarrow \quad r_{1,2} = 28 \pm \sqrt{783}$$

Consideremos que tenemos 5 cifras de precisión. La primer raíz se puede calcular sin error considerable.

$$r_1 = 28 + \sqrt{783} \approx 28 + 27{,}982 = 55{,}982 \pm 0{,}005 \quad (5 \text{ cifras})$$

al redondear la raíz, al calcular la segunda raíz se obtiene como resta de dos valores muy próximos

$$r_2 = 28 - \sqrt{783} = 28 - 27{,}98213\cdots \approx 0{,}018 \pm 0{,}005 \quad (2 \text{ cifras})$$

Una forma de evitar esto es reescribir la ecuación para evitar el redondeo de la raíz y la resta

$$x^2 - 56x + 1 = (x - r_1)(x - r_2) = x^2 - (r_1 + r_2)x + r_1\, r_2$$

utilizamos el valor r_1 calculado y despejamos r_2

$$r_2 = \frac{1}{r_1} = \frac{1}{55{,}982} = 0{,}17862 \quad (5 \text{ cifras})$$

△

Observación 1.4.1. Conviene reescribir fórmulas para evitar problemas numéricos como cancelación catastrófica, overflow, etc.

1.5. Cálculo de derivadas por cocientes incrementales

Sea f una función real $f(x) : \mathbb{R} \longrightarrow \mathbb{R}$, de clase C^2. Recordamos la definición de la derivada primera

$$\frac{df}{dx}(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}.$$

No es posible calcular límites numéricamente, por lo tanto, es necesario estimarla. Veremos a continuación diferentes maneras de realizar esta aproximación. Para calcular numéricamente la misma, se realizan aproximaciones como por ejemplo el cociente incremental $\Delta_{f(x),h}$ para un paso $h \in \mathbb{R}^+$ pequeño.

También analizaremos los errores cometidos al llevar este problema a la computadora. Las fuentes de errores que analizaremos son dos:

- Error debido a no trabajar con precisión infinita, al que llamaremos **error de redondeo**.
- Error debido al truncamiento de la serie infinita en el desarrollo de Taylor del que se despeja la derivada, al que llamaremos **error de truncamiento**.

1.5.1. Diferencia hacia adelante

Esta aproximación consiste en aproximar la derivada de la función como el cociente incremental. Utiliza el propio punto x y $x+h$.

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

A continuación calcularemos una cota superior para el error absoluto entre el valor real de la derivada y la representación en punto flotante de el cociente incremental.

$$\begin{aligned}
\text{Error}_{\text{absoluto}} &= \left| f'(x) - \frac{PF(f(x+h)) - PF(f(x))}{h} \right| \\
\dots &= \left| f'(x) + \Delta_{f(x),h} - \Delta_{f(x),h} - \frac{PF(f(x+h)) - PF(f(x))}{h} \right| \\
\dots &\leq \left| f'(x) + \Delta_{f(x),h} \right| + \left| \Delta_{f(x),h} - \frac{PF(f(x+h)) - PF(f(x))}{h} \right| \\
\text{Error}_{\text{absoluto}} &\leq \text{Error}_{\text{trunc}} + \text{Error}_{\text{PF}}
\end{aligned}$$

Este error tiene dos componentes, una debida a el truncamiento de los términos del desarrollo de Taylor, y el error de punto flotante de la propia representación.

Error de truncamiento

Planteamos Taylor próximo al punto x:

$$f(x+h) = f(x) + f'(x)\,h + f''(c)\,\frac{h^2}{2} \qquad c \in [x, x+h]$$

entonces podemos despejar el cociente incremental de paso h

$$\Delta_{f(x),h} - f'(x) = \frac{f(x+h) - f(x)}{h} - f'(x) = f''(c)\frac{h}{2}$$

por lo tanto obtenemos una buena aproximación para el error de truncamiento

$$\text{Error}_{\text{trunc.}} = \frac{|f''(c)|}{2}\,h \approx \frac{|f''(x)|}{2}\,h$$

Se concluye por tanto que el error de truncamiento es de orden h. Esto significa que menor será el error cuanto menor sea el paso h.

Error de punto flotante

$$PF(f(x+h)) = f(x+h)\,(1+\delta_h) \qquad |\delta_h| \leq \varepsilon_{mach}$$

$$PF(f(x)) = f(x)\,(1+\delta_f) \qquad |\delta_f| \leq \varepsilon_{mach}$$

$$\begin{aligned}
\text{Error}_{\text{PF}} &= \left| \frac{PF(f(x+h)) - PF(f(x))}{h} - \frac{f(x+h) - f(x)}{h} \right| \\
\dots &= \frac{|f(x+h)\,\delta_{f(x+h)} - f(x)\,\delta_{f(x)}|}{h} \\
\dots &\leq \frac{|f(x+h)|\,|\delta_{f(x+h)}| + |f(x)|\,|\delta_{f(x)}|}{h} \\
\dots &\leq \frac{|f(x+h)| + |f(x)|}{h}\,\varepsilon_{mach} \\
\text{Error}_{\text{PF}} &\leq \frac{2|f(x)|\,\varepsilon_{mach}}{h}
\end{aligned}$$

Al contrario de lo visto para el error de truncamiento, este error es inversamente proporcional al paso. Esto significa que un paso demasiado pequeño incrementa el error relacionado a la representación en punto flotante.

Es así que este análisis combina restricciones contrapuestas, ya que por un lado se requiere un paso pequeño para minimizar el error de truncamiento, pero por otro un paso demasiado pequeño complica el trabajo en punto flotante.

Por tanto, el error total combinando ambas fuentes de error toma la forma:

$$\text{Error}_{\text{total}} \leq \frac{2|f(x)|\,\varepsilon_{mach}}{h} + \frac{|f''(x)|}{2}\,h$$

Paso óptimo Dado que h puede ser elegido, es importante buscar el valor de h que minimice el error total. Para ello utilizamos la expresión del error obtenida y la derivamos para encontrar un mínimo

$$\frac{d\text{Error}}{dh} = 0$$

utilizando la expresión del error total obtenemos

$$\frac{-2|f(x)|\,\varepsilon_{mach}}{h^2} + \frac{|f''(x)|}{2} = 0$$

por lo tanto

$$h_{opt} = 2\sqrt{\frac{|f(x)|}{|f''(x)|}}\,\sqrt{\varepsilon_{mach}}$$

por ejemplo si $f''(x) \approx O(f(x))$ entonces $h_{opt} \approx \sqrt{\varepsilon_{mach}} = 10^{-8}$

1.5.2. Diferencia centrada

En esta aproximación se utilizan los puntos $x + h$ y $x - h$ y podemos ver que el error de truncamiento es de orden h^2

$$\Delta_{f(x),h} = \frac{f(x+h) - f(x-h)}{2\,h} \qquad E_{trunc} = O(h^2)$$

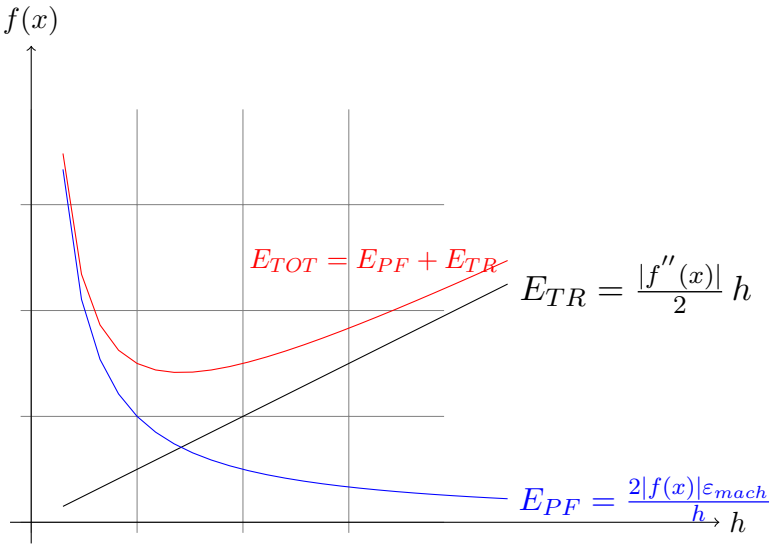

Figura 1.3: h óptimo

Aplicamos el desarrollo de Taylor en el punto x tomando como paso h y $-h$:

$$f(x+h) = f(x) + f'(x)\,h + f''(x)\,\frac{h^2}{2!} + f(c)'''\,\frac{h^3}{3!} \qquad c \in [x, x+h]$$

$$f(x-h) = f(x) - f'(x)\,h + f''(x)\,\frac{h^2}{2!} - f(d)'''\,\frac{h^3}{3!} \qquad d \in [x-h, x]$$

luego restamos miembro a miembro, obteniendo

$$f(x+h) - f(x-h) = 2h\,f'(x) + (f'''(c) + f'''(d))\,\frac{h^3}{3!}$$

por lo tanto el error de truncamiento será:

$$E_{trunc} = |\Delta f_{x,h} - f'(x)| = |f'''(c) + f'''(d)|\,\frac{h^3}{3!}\,\frac{1}{2\,h} \approx \frac{|f'''(x)|}{3!}\,h^2$$

1.5.3. Aproximación de derivada segunda

Partimos de la definición de derivada segunda

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

y aproximamos las derivadas utilizando cocientes incrementales.

Fórmula hacia adelante usando la aproximación de diferencia hacia adelante para cada derivada, obtenemos

$$\Delta^2 f_x = \frac{\Delta f_{x+h} - \Delta f_x}{h} = \frac{f(x+2h) - f(x+h) - (f(x+h) - f(x))}{h^2}$$

obteniendo una primer fórmula

$$\Delta^2 f_x = \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2}.$$

Fórmula centrada si aproximamos la derivada $f'(x)$ utilizando diferencia hacia atrás, obtenemos la siguiente expresión

$$\Delta^2 f_x = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}.$$

Error de truncamiento
Calculemos el error de truncamiento para la fórmula centrada. Aplicamos Taylor con paso h y $-h$ y sumamos miembro a miembro:

$$\begin{aligned} f(x+h) &= f(x) + f'(x)h + f''(x)\,\tfrac{h^2}{2} + f'''(x)\,\tfrac{h^3}{3!} + f^{iv}(c)\,\tfrac{h^4}{4!} \\ &+ \\ f(x-h) &= f(x) - f'(x)h + f''(x)\,\tfrac{h^2}{2} - f'''(x)\,\tfrac{h^3}{3!} + f^{iv}(d)\,\tfrac{h^4}{4!} \\ f(x+h) + f(x-h) &= 2\,f(x) + f''(x)h^2 + f^{iv}(x)\,\tfrac{h^4}{4!} \end{aligned}$$

por lo tanto el error de truncamiento está dado por la siguiente expresión

$$\left| \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - f''(x) \right| \cong \frac{f^{iv}(x)}{4!}\, h^2$$

1.6. Extrapolación de Richardson

El método de la diferencia centrada visto en la sección 1.5.2 ilustra un abordaje interesante para reducir el error de truncamiento de la derivada.

Una generalización de lo allí realizado podría ser considerar la fórmula de la diferencia centrada pero ahora evaluada en $\frac{h}{10}$:

$$\Delta_{f(x),\frac{h}{10}} = \frac{f(x+\frac{h}{10}) - f(x-\frac{h}{10})}{2\frac{h}{10}} = f'(x) + \frac{f'''(x)}{3!}\frac{h^2}{10^2} + \frac{f^v(x)}{5!}\frac{h^4}{10^4} + \dots$$

Entonces, combinando convenientemente la expresión anterior con la de la diferencia centrada llegamos a:

$$\Delta_{f(x),h} - 100\Delta_{f(x),\frac{h}{10}} = (1-100)f'(x) + o(h^4)$$

Es decir que logramos reducir el orden de aproximación de la derivada de una función a orden h^4 (nótese que con el primer planteamiento, aproximando f' con el cociente incremental simple el orden era h):

$$\frac{\Delta_{f(x),h} - 100\Delta_{f(x),\frac{h}{10}}}{1-100} = f'(x) + o(h^4)$$

Esto bien puede generalizarse a cualquier aproximación que admita una expresión del error de truncamiento como la expansión de una serie de potencias. De esta manera, combinando cuidadosamente estas técnicas es posible mejorar el orden del error.

Sea $x \in \mathbb{R}$ un valor que se desea estimar a partir de cierta formulación $T(h)$ y que verifica la siguiente expresión:

$$T(h) = x + a_0\, h^{p_1} + O(h^{p_2}) \qquad 1 \leq p_1 < p_2$$

Se observa que es posible despejar x y obtener una aproximación de orden p_1, debido al término $a_0\, h^{p_1}$. En el caso que se desee obtener una nueva aproximación de x de mayor orden (p_2) podemos aplicar el siguiente razonamiento.

$$T(h) = x + a_0\, h^{p_1} + O(h^{p_2})$$
$$T(q\, h) = x + a_0\, q^{p_1}\, h^{p_1} + O(h^{p_2})$$

multiplico la primera por q y resto miembro a miembro

$$T(q\, h) - T(h)\, q^{p_1} = (1 - q^{p_1})\, x + O(h^{p_2})$$

logrando eliminar el término $a_o\, h^{p_1}$ y obteniendo una aproximación de x de orden p_2:

$$R(h) = \frac{T(q\, h) - T(h)\, q^{p_1}}{1 - q^{p_1}} = x + O(h^{p_2})$$

podemos reescribir la fórmula para reducir errores numéricos debido a operaciones, obteniendo la expresión general de la aproximación de Richardson:

$$R(h) = T(h) + \underbrace{\frac{T(q\, h) - T(h)}{1 - q^{p_1}}}_{\text{corrección de Richardson}} .$$

Solicitamos al lector reconocer la analogía de esta expresión general con lo que realizamos en el ejemplo de la diferencia centrada donde $T(h)$ jugaría el rol de $\Delta_{f(x),h}$; x que es el valor a estimar era en ese caso la derivada de f; $p_1 = 2$ y $p_2 = 4$; el valor de q elegido fue $1/10$; y se encontró una mejor estimación $R(h) = \frac{\Delta_{f(x),h} - 100\Delta_{f(x),\frac{h}{10}}}{1-100}$.

1.7. Propagación de errores

Dada una función $f(\mathbf{x}) : \mathbb{R}^n \longrightarrow \mathbb{R}$, $z = f(\mathbf{x})$. Es útil poder estimar cual será el error o la variación en el valor de z al tener un error o variación conocido para los valores x_i.

Fórmula 1

$x_i \in \mathbb{R} \quad i = 1, \ldots, n$ valores desconocidos exactos
$\bar{x}_i \in \mathbb{R} \quad i = 1, \ldots, n$ valores conocidos aproximados
$\bar{x}_i = x_i + \varepsilon_i \quad i = 1, \ldots, n$

De esta forma los errores absolutos en cada componente del vector $\mathbf{x}$ son:

$$\Delta_{x_i} = |\varepsilon_i| \in \mathbb{R}^+ \qquad i = 1, \ldots, n$$

Ahora queremos evaluar la función $f(\mathbf{x})$:

$$z = f(x_1, x_2, \ldots, x_n)$$

pero dado que cada componente tiene errores logramos evaluar

$$\bar{z} = f(x_1 + \varepsilon_1, \ldots, x_n + \varepsilon_n)$$

por lo que deseamos saber cual es el error que estamos comentiendo en z

$$\Delta_z = |f(x_1 + \varepsilon_1, \ldots, x_n + \varepsilon_n) - f(x_1, \ldots, x_n)|$$

Aplicando Taylor:

$$f(x_1 + \varepsilon_1, \ldots, x_n + \varepsilon_n) \approx f(x_1, \ldots, x_n) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(x_1, \ldots, x_n) \cdot \varepsilon_i$$

por lo tanto aplicando la definición del error de z

$$\Delta_z \approx \left| \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(x_1, \ldots, x_n) \cdot \varepsilon_i \right| \leq \sum_{i=1}^{n} \left| \frac{\partial f}{\partial x_i}(x_1, \ldots, x_n) \right| \cdot |\varepsilon_i|$$

por lo tanto obtenemos que el error absoluto de z está acotado superiormente por la suma de los errores absolutos de x_i multiplicados por la derivada parcial de f correspondiente:

$$\Delta_z \leq \sum_{i=1}^{n} \left| \frac{\partial f}{\partial x_i} \right| \cdot \Delta_{x_i}$$

Fórmula 2 En este caso consideramos el error de cada componente ε_i como variables aleatorias independientes con esperanza cero y varianza finita:

$$\mathrm{E}(\varepsilon_i) = 0 \qquad \mathrm{Var}(\varepsilon_i) = \Delta_{x_i}^2 < \infty$$

$$\Delta_z^2 = \mathrm{Var}(z) = \mathrm{Var}(f(x_1 + \varepsilon_1, \ldots, x_n + \varepsilon_n))$$

recordando Taylor:

$$f(x_1 + \varepsilon_1, \ldots, x_n + \varepsilon_n) \approx f(x_1, \ldots, x_n) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(x_1, \ldots, x_n) \cdot \varepsilon_i$$

elevamos al cuadrado ambos lados de la igualdad y despreciamos algunos términos, obteniendo:

$$\mathrm{Var}(f(x_1 + \varepsilon_1, \ldots, x_n + \varepsilon_n)) \approx \sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i}\right)^2 \cdot \mathrm{Var}(\varepsilon_i)$$

por lo tanto obtenemos:

$$\Delta_z = \sqrt{\sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i}\right)^2 \cdot \Delta x_1^2}$$

1.8. Ejercicios

Ejercicio 1. Encuentre experimentalmente los siguientes valores de su calculadora, con dos cifras de precisión:

(A) El valor $\varepsilon_{\text{mach}}$ definido como el mínimo x tal que la representación en punto flotante de $1 + x$ es mayor que 1.

(B) El mayor número representable.

(C) El menor número positivo representable.

Ejercicio 2. Sea $P_n = (x_n, y_n)$, $n \in \mathbb{N}$, la sucesión generada, a partir de valores iniciales x_0, y_0, por la fórmula de recurrencia $\begin{cases} x_{n+1} = \{2x_n + y_n\} \\ y_{n+1} = \{x_n + y_n\} \end{cases}$ donde $\{u\}$ es la parte decimal de u.

(A) Muestre que si $x_0 = y_0 = \frac{1}{2}$ entonces P_n es periódica con $P_{n+3} = P_n$.

(B) Analice lo que sucede si $x_0 = y_0 = \frac{1}{3}$.

(C) Implemente un programa que calcule y grafique los primeros 100 puntos P_n de cada una de las cucesiones anteriores. Explique el resultado obtenido.

Ejercicio 3. *Errores relativo y absoluto.*

(A) Al determinar una constante C, se obtuvo el valor 92.34 con un error relativo de un 0.1 %. ¿En qué intervalo se encuentra C? ¿Cuál es el error absoluto?

(B) ¿Cuántos dígitos del número $\sqrt{22}$ deben darse para determinarlo con un error relativo no exceda el 0.1 %?

(C) En una medición se obtiene el valor $v = 17261$. Se sabe que el error relativo es del 1 %. ¿Cómo debería escribirse v para reflejar este hecho?

Ejercicio 4. *Representación interna de números.* Una computadora tiene un sistema de punto flotante decimal con 5 dígitos de precisión y 2 dígitos para el exponente. ¿Cuántos números diferentes pueden representarse con dicha arquitectura? ¿Cuáles son la menor y la mayor separación entre números representables consecutivos? Estime el valor ε_{match} (ver Ejercicio 1).

Ejercicio 5. *Cancelación catastrófica y desborde.*

(A) Se desea calcular numéricamente $\lim_{n\to\infty} \int_n^{n+1} \log(x)\, dx$. ¿Cómo puede reescribirse dicha integral para evitar efectos de cancelación catastrófica?

(B) Reescriba la expresión $\frac{e^x}{e^x+1}$ para poder evaluarla en valores grandes de x evitando efectos de desborde.

(C) Comente los inconvenientes que pueden surgir al implementar un programa para calcular la derivada de $\cos(x)$ utilizando el cociente incremental $\frac{\cos(x+h)-\cos(x)}{h}$. ¿Cómo reescribiría usted dicho cociente?

Ejercicio 6. *Errores en operaciones.*

(A) El diámetro interior de un tanque de agua esférico es de $1{,}5 \pm 0{,}05$ m. Calcule su volumen (con el error correspondiente) aproximando $\pi \simeq 3{,}1416$.

(B) Un campo rectangular mide aproximdamente 2000 por 3000 metros. ¿Con qué error deberán medirse los lados para obtener el área con un error inferior a un metro cuadrado?

Ejercicio 7. *Cálculo de la derivada con el cociente incremental.* Dada una función $f\colon I \to \mathbb{R}$ de clase C^∞, donde $I \subseteq \mathbb{R}$ es un intervalo, se desea calcular la derivada $f'(x)$ en un punto $x \in I$ usando la *diferencia hacia adelante* en x, es dicir, empleando la fórumla

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

Al aproximar numéricamente la derivada por la diferencia hacia adelante, esto es, evaluando en un h pequeño no nulo, se cometen dos tipos de errores. En primer lugar está el *error de truncamiento*, que proviene de tomar un h pequeño fijo en lugar del el límite cuando $h \to 0$, y en segundo lugar el *error de redondeo* que son los errores numéricos de la máquina, tanto en la representación como en las operaciones.

(A) Calcular la derivada de la función $f(x) = \sqrt{x}$ en el punto $x = 1$, con la diferencia hacia adelante y usando $h = 1{,}5^k$ con $k = 0, 1, \ldots, 100$. Graficar, usando escala logarítmica, el error absoluto cometido en función de h. Explicar el comportamiento observado.

(B) Usando los resultados vistos en clase sobre los errores de truncamento y de redondeo, estimar el valor de h óptimo para el calculo anterior. Coteje este valor de h con el resultado obtenido en la parte anterior.

(C) Repetir las partes (A) y (B) para la función $\tan(x)$ y el punto $x = 1{,}57$.

(D) Repetir las partes (A) y (B) para el cálculo de la derivada segunda de $f(x) = \sqrt{x}$ en el punto $x = 1$, usando la *discretización* siguiente:

$$f''(x) \simeq \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}.$$

Ejercicio 8. *Extrapolación de Richardson.* Considere las aproximaciones realizadas de las diferentes derivadas en el ejercicio enterior.

(A) A partir del vector de aproximaciones correspondientes a los diferentes valores de h, use extrapolación de Richardson para hallar un nuevo vector de aproximaciones. (El nuevo vector tendrá una entrada menos.)

(B) Calcule y grafique el error cometido, comparándolo con el correspondiente a las aproximaciones originales.

(C) Repita el procedimiento, extrapolando el último vector hallado.

Ejercicio 9. Se desea hallar las cuatro raíces de polinomio

$$P_4(x) = x^4 - 12x^3 + 54x^2 - 108x + 80{,}99999999999999.$$

(A) Resuelva el problema usando el comando `roots`. ¿Qué sucedió con el vector de coeficientes del polinomio?

(B) Observando que $P_4(x) = (x-3)^4 - 10^{-14}$, resuelva analíticamete el problema.

(C) Considere la ecuación $(x-3)^4 = 0$, con solución exacta $x = 3$ y la correspondiente solución del problema "perturbado" de las partes (A) y (B). Halle el error (variación) relativo de la solución.

(D) Halle la diferencia relativa en los coeficientes de la ecuación de las partes (A)-(B) y la de la parte (C). Extraiga conclusiones sobre el número de condición del problema, definido como la razón entre el error relativo en la solución y el error relativo en los datos de entrada.

Ejercicio 10. Se desea calcular los valores de la función exponencial a partir de su desarrollo en serie

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

(A) Use el programa dado para efectuar la suma anterior hasta $n = 100$, para un rango de valores de x:

(B) Investigue qué sucede con el error relativo en los resultados numéricos obtenidos. Use la función `exp` y grafique con `semilogy`. ¿Dónde se dan los peores resultados? Justifique.

(C) Piense una solución para hacer el cálculo en los valores anteriores con mejor precisión.

```
x=-20:20;
sum=ones(size(x));
t=x; n=1;
while n<100
   sum=sum+t;
   n=n+1;
   t=t.*x/n;
end
```

Ejercicio 11. (Ver Ejercicio 7) Se quiere obtener numéricamente la derivada de una función f mediante la siguiente discretización por diferencia centrada:

$$f'(x) \simeq \Delta f = \frac{f(x+h) - f(x-h)}{2h}.$$

(A) Halle una cota para el error de truncamiento debido a la discretización usada.

(B) Estime el error de redondeo debido al uso de aritmética de punto flotante.

(C) Estime el error total y el h óptimo.

(D) Usando lo anterior, estime $f'(x)$ y su error para $f(x) = e^x$ en $x = 0$. Compare Δf con $f'(x)$ para diferentes valores de h (p.ej. $h = 10^{-n}$, $n = 1, 2, \ldots$) y obtenga una gráfica experimental que verifique el h óptimo obtenido.

(E) Repita lo anterior para $f(x) = \text{sen}(x)$ en $x = 0$ y explique los resultados.

Ejercicio 12. En versiones anteriores de *Octave* se generaba un error al calcular $\text{arcsenh}(x)$ para valores negativos grandes. El objetivo de este ejercicio es analizar el problema y proponer una solución.

(A) Calcule $\text{arcsenh}(-10^{30})$ de dos formas: usando la fórmula $\text{arcsenh}(x) = \log(x + \sqrt{x^2 + 1})$ y utilizando la función `asinh`.

(B) Explique el resultado obtenido y proponga una forma de solucionarlo.

Ejercicio 13. *Propagación del error de redondeo.*

(A) Suponga que se conoce una cantidad $x > 0$ con error absoluto δx pequeño en relación a x. Si $y = \sqrt{x}$ estime el error absoluto δy en base a x y δx. Estime también el error relativo R_y en y en base x y al error relativo en x: $R_x = \delta x/x$.

(B) Si las cantidades $x_1, x_2 > 0$ se conocen con error δx, halle una cota al error absoluto en la cantidad $z = \sqrt{x_1} - \sqrt{x_2}$.

(C) Si $x_1 = 9 \times 10^{14} + 1$ y $x_2 = 9 \times 10^{14} - 1$, calcule en la computadora el valor z, llamando z_1 al resultado obtenido. Halle el error $\delta z = z - z_1$ cometido tomando como verdadero valor de z el resultado del cálculo $z = \frac{x_1 - x_2}{\sqrt{x_1} + \sqrt{x_2}}$.

(D) Compare δz con la cota obtenida en (B) suponiendo que el error se debe sólo a la propagación del error cometido al almacenar x_1 y x_2 en punto flotante.

(E) ¿Qué relación hay entre el error relativo inicial (en x_1 y x_2) y el final (en z)?

Capítulo 2

Sistemas de Ecuaciones Lineales

2.1. Introducción

Un sistema de m ecuaciones lineales y n incógnitas consta de un conjunto de relaciones algebraicas de la forma:

$$\sum_{j=1}^{n} a_{ij}\, x_j = b_i \qquad x_j,\, a_{ij},\, b_i \in \mathbb{R} \quad \forall\, i = 1 \dots m$$

el cual puede ser representado en notación matricial como $A\mathbf{x} = \mathbf{b}$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \in \mathcal{M}_{m\times n}(\mathbb{R}) \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

Este sistema tiene solución única si y solo si $m = n$ y $|A| \neq 0$, entonces decimos que es compatible determinado (CD). En este capítulo trabajaremos con sistemas de esta clase.

En el caso de $m > n$ existen más ecuaciones que variables, por lo tanto, si el vector $\mathbf{b}$ no pertenece al espacio generado por las columnas de A, no existe una solución que verifique todas las ecuaciones, decimos que el sistema es incompatible. Más adelante veremos técnicas para resolver este tipo de problemas.

Métodos directos e indirectos Los métodos de resolución de sistemas lineales pueden ser clasificados en las siguientes dos categorías:

- Directos: obtenemos solución luego de un número finito de iteraciones. Si tenemos precisión infinita la solución es exacta.
- Indirectos: obtenemos una aproximación de la solución $(\bar{x}_k)$, luego de k iteraciones. La solución se mejora sucesivamente con cada paso.

2.2. Métodos directos

Los métodos directos generan soluciones aproximadas luego de realizar una cantidad finita de pasos. Uno de los ejemplos más habituales es la escalerización gaussiana.

2.2.1. Solución de sistemas triangulares

Comenzaremos por abordar un caso particular de sistemas lineales. Sea el siguiente sistema 3×3 triangular inferior no singular:

$$\begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

Las matrices con bloques con ceros serán representadas esquemáticamente de la siguiente forma:

$$A\,\mathbf{x} = \mathbf{b} \qquad A = \begin{pmatrix} \diagdown & 0 \\ & \diagdown \end{pmatrix}$$

Dado que el sistema es no singular, las entradas de la diagonal a_{ii}, $i = 1, 2, 3$ son distintas a cero, por lo tanto lo podemos resolver de la siguiente forma:

$$x_1 = \frac{b_1}{a_{11}}, \tag{2.1}$$

$$x_2 = \frac{b_2 - a_{21}x_1}{a_{22}}, \tag{2.2}$$

$$x_3 = \frac{b_3 - a_{31}x_1 - a_{32}x_2}{a_{33}}. \tag{2.3}$$

Este algoritmo puede ser extendido para sistemas triangulares inferiores de orden n, de la forma $A\,\mathbf{x} = \mathbf{b}$. Este método tiene el nombre de sustitución hacia adelante:

$$\begin{aligned} x_1 &= \frac{b_1}{a_{11}} \\ x_i &= \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}\ x_j \right), \qquad i = 2, \ldots, n \end{aligned}$$

El pseudo-código del método es el siguiente:

- paso 1: $x_1 = \frac{b_1}{a_{11}}$
- paso i: $x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}\ x_j \right), \qquad i = 1, \ldots, i-1$

recordando la definición de *Flops* del capítulo 1, podemos calcular el costo computacional del método. Se realizan $n\,(n+1)/2$ multiplicaciones-divisiones mientras que el número de sumas-restas es $n\,(n-1)/2$, por lo tanto, el costo es n^2 *flops*.

En el caso que el sistema lineal sea triangular superior,

$$A\,\mathbf{x} = \mathbf{b} \qquad A = \begin{pmatrix} \diagdown & \\ 0 & \diagdown \end{pmatrix}$$

podemos utilizar un método análogo llamado sustitución hacia atrás (BS[1]):

$$\begin{aligned} x_n &= \frac{b_n}{a_{nn}} \\ x_i &= \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^{n} a_{ij}\, x_j \right), \qquad i = n-1, \ldots, 1 \end{aligned}$$

Pseudo-código:

- paso n: $x_n = b_n / a_{nn}$
- paso i, $i = n-1, \ldots, 1$: $x_i = \frac{b_i - \sum_{k=i+1}^{n} a_{ik}\, x_k}{a_{ii}}$

2.2.2. Escalerización Gaussiana (EG)

El Método de Escalerización Gaussiana (MEG) es un método directo para resolución de Sistemas de Ecuaciones Lineales.

Permite llevar un sistema general no singular a uno equivalente triangular superior, por medio de la aplicación de operaciones elementales (intercambio y combinación lineal de filas).

Algoritmo: Paso k: si $a_{kk}^{(k)} \neq 0$ será llamado *pivot*

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \ldots & a_{1k}^{(1)} & \ldots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \ldots & a_{2k}^{(2)} & \ldots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & a_{kk}^{(k)} & \ldots & a_{kn}^{(k)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & a_{nk}^{(k)} & \ldots & a_{nn}^{(k)} \end{bmatrix}$$

$$\begin{aligned} a_{ij}^{(k+1)} &= a_{ij}^{(k)} - l_{ik}\, a_{kj}^{(k)} \qquad & l_{ik} &= \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \\ b_i^{(k+1)} &= b_i^{(k)} - l_{ik}\, b_k^{(k)} \quad i = k+1, \ldots, n; \quad j = k, \ldots, n \end{aligned}$$

Las cantidades de flops de *EG* por cada bloque del código son las siguientes:

[1] BS por Backwards Substitution

Algoritmo 1 Pseudo-código: EG no eficiente

Sea A una matriz con entradas $A_{i,j} = a(i,j)$

```
for k = 1 → n − 1 do
  for i = k + 1 → n do
    l(i, k) ← a(i, k)/a(k, k)
    a(i, k) ← 0
    for j = k + 1 → n do
      a(i, j) ← a(i, j) − l(i, k) ∗ a(k, j)
    end for
  end for
end for
```

- loop j: $2(n-k)$
- loop i: $(2(n-k)+1)\ (n-k) = 2(n-k)^2 + (n-k)$
- loop k: $\sum_{k=1}^{n-1} 2(n-k)^2 + (n-k)$

Tomando el loop k y desarrollando obtenemos:

$$\begin{aligned} flops(EG) &= \sum_{k=1}^{n-1} 2n^2 - 4n\,k + 2k^2 + n - k \\ \ldots &= 2\sum_{k=1}^{n-1} k^2 - (4n+1)\sum_{k=1}^{n-1} k + 2n^2 + n \\ \ldots &= 2\,\frac{(n-1)n(2n-1)}{6} - (4n+1)\frac{(n-1)n}{2} + 2n^2 + n \end{aligned}$$

Por lo tanto, la cantidad de *flops* necesarios para resolver $A\,\mathbf{x} = \mathbf{b}$ usando EG y sustitución hacia adelante es la siguiente:

$$EG + BS = O(2/3\,n^3) + O(n^2) \approx O(2/3\,n^3)$$

Observación 2.2.1. Se recuerda que utilizando el MEG, sin pivoteo, el valor del determinante se mantiene inalterado, y una vez escalerizada la matriz, el determinante resulta en el producto de los valores de la diagonal, por tanto, acotamos la cantidad de *flops* para resolver $|A|$:

$$EG + \prod_{i=1}^{n} a_{ii}^{(i)} = O(2/3\,n^3) + O(n) \approx O(2/3\,n^3)$$

Se sugiere comparar con el costo computacional del cálculo utilizando la fórmula general o descomposición por filas (deberá observar que la cantidad de multiplicaciones son $n!(n-1)$ y luego se deben sumar $n!$ términos).

2.2.3. Descomposición LU (sin pivoteo)

Un segundo método directo de resolución de SL es mediante la llamada Descomposición LU. Esta descomposición se basa en el método de escalerización gaussiana pero permite ahorros computacionales en algunos casos como desarrollaremos luego de presentar el método.

Dada una matriz $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, tal que $|A| \neq 0$, existen dos matrices L y U tal que $A = LU$.

$$\begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ l_{21} & 1 & 0 & \vdots & \vdots \\ l_{31} & l_{32} & 1 & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & 0 \\ l_{n1} & \dots & \dots & l_{nn-1} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ \vdots & 0 & u_{33} & \dots & u_{3n} \\ \vdots & \dots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & u_{nn} \end{bmatrix} = A$$

por lo tanto:

$$a_{ij} = \sum_{k=1}^{r} l_{ik}\, u_{kj} \qquad r = \text{mín}\{i, j\}$$

tomando la convención de que $l_{ii} = 1$.

Obsérvese que los l_{ij} son los mismos coeficientes determinados por el algoritmo y los u_{ij} son las entradas de la matriz escalerizada, es decir que escribimos $u_{ij} = a_{ij}^{(i)}$.

Demostraremos la anterior relación para el cálculo de las entradas a_{ij} de A a partir de las entradas de L y U realizando los pasos de EG:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik}\, a_{kj}^{(k)} \qquad k = 1, \dots, p; \quad p = i - 1 \text{ si } i \leq j; \quad p = j \ \textit{si}\ i > j$$

Si sumamos la ecuación anterior para los distintos valores de $k = 1, \dots, p$, y reordenando:

$$\sum_{k=1}^{p} a_{ij}^{(k+1)} - \sum_{k=1}^{p} a_{ij}^{(k)} = -\sum_{k=1}^{p} l_{ik}\, a_{kj}^{(k)}$$

Ahora, los términos de la izquierda se cancelan entre las sumatorias:

$$a_{ij}^{(p+1)} - a_{ij} = -\sum_{k=1}^{p} l_{ik}\, a_{kj}^{(k)}$$

y como

$$a_{ij}^{(p+1)} = \begin{cases} a_{ij}^{(i)} & i \leq j \\ 0 & i > j \end{cases}$$

y además teníamos que $u_{ij} = a_{ij}^{(i)}$, se concluye reescribiendo las entradas que:

$$a_{ij} = \sum_{k=1}^{r} l_{ik}\, u_{kj} \qquad r = \text{mín}\{i, j\}$$

con lo que se termina la prueba.

El número de *flops* para realizar la descomposición L.U. es igual a las operaciones requeridas para aplicar la escalerización gaussiana. De esta forma, enumeremos las operaciones necesarias para resolver un sistema lineal aplicando L.U. :

$$\begin{aligned} A = LU \quad (EG) \quad &= O(2/3n^3) \\ L\,\mathbf{y} = \mathbf{b} \quad (BS) \quad &= O(n^2) \\ U\,\mathbf{x} = \mathbf{y} \quad (FS) \quad &= O(n^2) \\ \textbf{Total} \quad \textit{flops} \quad &\approx O(2/3n^3) \end{aligned} \tag{2.4}$$

Una característica importante de este método está en la resolución de múltiples sistemas lineales, es decir, varios sistemas con la misma matriz A. Por ejemplo, calculemos la cantidad de operaciones necesarias para resolver m sistemas lineales utilizando L.U.:

$$\begin{array}{c} A\,\mathbf{x}_i = \mathbf{b}_i \\ i = 1, \ldots, m \end{array} \Rightarrow \left\{ \begin{array}{l} A = L\,U \\ for\ i = 1, \ldots, m \\ \quad A\,\mathbf{y}_i = \mathbf{b}_i \\ \quad U\,\mathbf{x}_i = \mathbf{y}_i \\ end \end{array} \right. \Rightarrow flops = O(2/3\,n^3 + 2mn^2)$$

Número de *flops* para hallar A^{-1} con L.U.

$$A\,X = \mathrm{I} \Rightarrow \mathbf{b}_j = \mathrm{I}_j = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad j = 1, \ldots, n \quad \Rightarrow flops = O(2/3n^3 + 2n^3) = O(8/3n^3)$$

Observación 2.2.2. Dado un sistema lineal $A\,\mathbf{x} = \mathbf{b}$, no es económico hallar $\mathbf{x}$ calculando $A^{-1}\,\mathbf{b}$.

2.2.4. EG con pivotes

¿Qué pasa con el método de escalerización si en algún paso i, se llega a $a_{kk}^{(k)} = 0$? Es imposible realizar el algoritmo ya que este valor debería ser el denominador en los cocientes que modifican las entradas a partir de allí, en particular, se requiere para el cálculo de los $l(i,k)$.

Entonces, lo que podemos hacer es intercambiar filas (pivotear), ya que supusimos que A es invertible, entonces existe $a_{pk}^{(k)} \neq 0$, $k+1 \leq p \leq n$.

Lo anterior es cierto con aritmética exacta. Si la aritmética es PF entonces conviene pivotear si algún $a_{kk}^{(k)} << 1$.

Ejemplo 2.2.1 (Error sin pivoteo)**.** Arit. PF 5 dígitos con redondeo: $\pm a_1\,,\ a_2\,a_3\,a_4\,a_5\ \times 10^e$

Consideremos el siguiente sistema de ecuaciones lineales:

$$(S)\ \left[\begin{array}{ccc|c} 10 & -7 & 0 & 7 \\ -3 & 2,099 & 6 & 3,901 \\ 5 & -1 & 5 & 6 \end{array}\right]$$

y lo resolvemos aplicando EG sin utilizar pivoteo:

$$A^{(1)}\ \left[\begin{array}{ccc|c} 1,0000 \times 10^1 & -7,0000 \times 10^0 & 0,0000 \times 10^0 & 7,0000 \times 10^0 \\ -3,0000 \times 10^0 & 2,099 \times 10^0 & 6,0000 \times 10^0 & 3,9010 \times 10^0 \\ 5,0000 \times 10^0 & -1,0000 \times 10^0 & 5,0000 \times 10^0 & 6,0000 \times 10^0 \end{array}\right]$$

En el segundo paso obtenemos una entrada a_{22} con un valor muy bajo comparado con el resto de los valores de la matriz.

$$A^{(2)}\ \left[\begin{array}{ccc|c} 1,0000 \times 10^1 & -7,0000 \times 10^0 & 0,0000 \times 10^0 & 7,0000 \times 10^0 \\ 0 & -1,0000 \times 10^{-3} & 6,0000 \times 10^0 & 6,0010 \times 10^0 \\ 0 & 2,5000 \times 10^0 & 5,0000 \times 10^0 & 2,5000 \times 10^0 \end{array}\right]$$

en el tercer paso obtenemos un error en la componente b_3

$$A^{(3)} \left[\begin{array}{ccc|c} 1,0000 \times 10^1 & -7,0000 \times 10^0 & 0,0000 \times 10^0 & 7,0000 \times 10^0 \\ 0 & -1,0000 \times 10^{-3} & 6,0000 \times 10^0 & 6,0010 \times 10^0 \\ 0 & 0 & 1,5005 \times 10^4 & 1,5004 \times 10^4 \end{array}\right]$$

obtenemos la solución:

sol. numérica	sol. exacta	
$\hat{x}_1 = -2,8000 \times 10^{-1}$	$x_1 = 0$	$\times$ mal
$\hat{x}_2 = -1,4000 \times 10^0$	$x_2 = -1$	$\times$ mal
$\hat{x}_3 = 9,9993 \times 10^{-1}$	$x_3 = 1$	$\cong$ Ok

△

Observación 2.2.3. Si $1,5004 \times 10^4$ hubiera sido $1,5005 \times 10^4$ (valor exacto) entonces la solución numérica sería igual a la exacta, por lo tanto se concluye que el error está provocado por no haber pivoteado cuando el valor a_{22} era pequeño.

En general pivoteamos si $a_{ii}^{(i)}$ vale cero o cualquier valor que sea "mucho menor" que el resto de las entradas de $A^{(i)}_{i\ldots n,i\ldots n}$. En el ejemplo 2.2.1 un valor "mucho menor" que los otros corresponde a menor que un 1 % por ejemplo.

Si pivoteamos entonces EG es numericamente estable.

Estrategias de pivoteo Presentaremos dos formas de elegir la entrada de la matriz que utilizaremos como pivot.

Pivoteo parcial Se compara el valor del pivot actual con todas las entradas siguientes de esa columna (por debajo de k), eligiendo como pivot al de mayor magnitud.

$$\arg \max_{r \in \{k,\ldots,n\}} |a_{rk}^{(k)}| = p \Rightarrow \text{ pivot: } a_{pk}$$

Luego de elegir la fila p, debemos intercambiar la fila k por la p y continuar con EG. La cantidad de comparaciones totales que se realizan es del orden de n^2.

Pivoteo completo Se compara el valor del pivot actual con cualquier otra entrada por debajo de esa fila y a la derecha de esa columna, eligiendo como pivot al de mayor magnitud.

$$\arg \max_{r \in \{k,\ldots,n\}, s \in \{k,\ldots,n\}} |a_{rs}^{(k)}| = p, q \Rightarrow \text{ pivot: } a_{pq}$$

En este caso, además de intercambiar la fila k por la fila p, también debemos intercambiar la columna k por la q. En este caso la cantidad de comparaciones totales es del orden de n^3.

2.2.5. Descomposición LU con intercambio de filas

Teorema 2.2.1. *Sea $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ no singular. Se puede descomponer A como $PA = LU$, con P una matriz de permutación.*

Observación 2.2.4. PA intercambia las filas de A, mientras que AP intercambia sus columnas. Compruébelo tomando $A \in \mathcal{M}_{2\times 2}(\mathbb{R})$ genérica y $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Haciendo EG con pivoteo formamos $P \in \mathcal{M}_{n\times n}(\mathbb{R})$ matriz de permutaciones de filas

$$P\,A = \tilde{A} = L\,U \Rightarrow PA = LU$$

Solución de un SL con descomposición PLU

Hallar la descomposición PLU. Luego $PA\mathbf{x} = P\mathbf{b}$. Llamamos $P\mathbf{b} = \mathbf{b}'$.

Con $\mathbf{b}'$ planteamos el sistema $LU\mathbf{x} = \mathbf{b}'$, el cual se resuelve mediante sustitución hacia atrás y hacia adelante, primero resolviendo un sistema auxiliar $L\mathbf{y} = \mathbf{b}'$, donde $\mathbf{y} = U\mathbf{x}$, y luego obtenemos $\mathbf{x}$ resolviendo $U\mathbf{x} = \mathbf{y}$.

$$L\,U\,\mathbf{x} = \mathbf{b}' \;\Rightarrow \begin{cases} P,\, L,\, U \\ \mathbf{b}' = P\,\mathbf{b} \\ L\,\mathbf{y} = \mathbf{b}' \\ U\,\mathbf{x} = \mathbf{y} \end{cases} \Rightarrow \mathbf{x}$$

2.2.6. Almacenamiento económico

Definición 2.2.1 (Matrices Esparsas)**.** $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ es esparsa si la mayoría de sus entradas son nulas.

Las matrices esparsas aparecen en muchísimos problemas tanto de ingeniería como de otras ramas y su manipulación son un vivo tema de investigación.

Sea A una matriz esparsa con elementos no nulos, una técnica de almacenamiento podría ser guardar el valor de la entrada junto con sus índices de fila y columna:

$$\left.\begin{array}{ccc} (i_1 & j_1 & a_{i_1,j_1}) \\ (i_2 & j_2 & a_{i_2,j_2}) \\ & \vdots & \\ (i_m & j_m & a_{i_m,j_m}) \end{array}\right\} \text{ si una entrada } a_{ij} \text{ no está en la lista, entonces es nula.}$$

¿Cómo puede ser esto eficiente si por cada entrada debo almacenar tres valores en lugar de uno? Veamos, si se guardan las ternas se requieren $3m$ NPF (números en punto flotante). Por otra parte, si se guarda llena, es decir con todos los ceros que corresponden se necesitarán n^2 NPF. Por tanto, si la matriz es esparsa, $m << n$, entonces entonces $3m << n^2$, lográndose una interesante reducción de espacio de almacenamiento.

Observación 2.2.5. En general A^{-1} no es esparsa aunque A lo sea.

2.2.7. Estructura de banda

Una matriz $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ es una *matriz banda* si los valores no nulos de la matriz se presentan solamente en una banda entorno a la diagonal.

Formalmente, la matriz tiene una estructura de banda si:

$$a_{i,j} = 0 \text{ si } \begin{cases} i - j > k_1 \\ j - i > k_2 \end{cases} \quad \text{con } k_1, k_2 \geq 0.$$

Definimos entonces el *ancho de banda* de una matriz como $k_1 + k_2 + 1$.

Ejemplo 2.2.2. Algunos casos particulares:

- Una matriz diagonal es una matriz banda con $k_1 = k_2 = 0$ y su ancho de banda es 1.
- Una matriz banda es tridiagonal cuando $k_1 = k_2 = 1$.
- Una matriz es triangular superior si $k_1 = n - 1$ y $k_2 = 0$.

△

Para este tipo de matrices, muchos algoritmos de resolución pueden optimizarse reduciendo considerablemente tanto la complejidad como el costo computacional y así el tiempo de ejecución.

Un ejemplo de esto es el Algoritmo de Thomas para matrices tridiagonales, en el cual la descomposición LU pasa a tener un orden lineal.

2.2.8. Otros métodos directos

Antes de pasar a los métodos indirectos de resolución, mencionaremos algunos otros métodos directos para que el lector amplíe su visión sobre los mismos e indague sus ventajas y desventajas.

Asumiremos que los sistemas son compatibles determinados.

Cálculo de matriz inversa

Dado el sistema $A\mathbf{x} = \mathbf{b}$, es posible determinar $\mathbf{x}$ operando algebraicamente en la ecuación: $A^{-1}A\mathbf{x} = A^{-1}\mathbf{b} \Rightarrow \mathbf{x} = A^{-1}\mathbf{b}$.

Por tanto, hallar la matriz inversa de A es otro mecanismo para resolver un sistema de ecuaciones.

Notamos sin embargo que el método para el cálculo de la matriz inversa en el que se construye la matriz ampliada $[A|I]$ y se reduce hasta obtener $[I|A^{-1}]$ involucra doblemente escalerización Gaussiana (ya que se debe escalerizar hacia "abajo" y luego hacia "arriba"), para luego realizar una multiplicación matriz por vector. Se sugiere comparar este costo con respecto a $EG + BS$.

Regla de Cramer

Dado el sistema $A\mathbf{x} = \mathbf{b}$, la Regla de Cramer devuelve cada entrada del vector solución solamente como un cociente de determinantes que dependen de la matriz A y del vector $\mathbf{b}$. En efecto:

$$x_i = \frac{\det(A_i)}{\det(A)}$$

donde A_i es la matriz resultante de reemplazar la columna i-ésima de la matriz A por el vector $\mathbf{b}$.

Ejemplo 2.2.3. Sea $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$, y $\mathbf{b} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$.

Entonces $\det(A) = 1$, $\det(A_1) = \begin{vmatrix} 3 & 1 \\ 2 & 1 \end{vmatrix} = 1$, $\det(A_2) = \begin{vmatrix} 2 & 3 \\ 1 & 2 \end{vmatrix} = 1$.

Resultando en $\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. △

Se deja como ejercicio calcular el costo computacional de este método para su comparación con los anteriores.

2.3. Estabilidad de sistemas lineales

La resolución de sistemas lineales mediante métodos numéricos involucra tanto errores de redondeo por el pasaje de números a una representación finita, como en el caso de la resolución de estos sistemas utilizando métodos iterativos, el truncamiento de una sucesión que converge a la solución real del sistema. Es por esta razón que es necesario definir una noción de cercanía en los espacios con los que estaremos trabajando. Es así que comenzamos la sección introduciendo algunos conceptos y resultados referentes a normas vectoriales y matriciales.

2.3.1. Norma de vectores

Trabajaremos en el espacio vectorial $\mathbb{R}^n$ con las operaciones suma y producto interno habituales entre vectores. Una norma es una función $\|\cdot\|$ que verifica las siguientes propiedades:

$$(\mathbb{R}^n, \mathbb{R}, +, \cdot) \text{ } e.v., \qquad \|\cdot\| : \mathbb{R}^n \longrightarrow \mathbb{R}$$

1. $\|\mathbf{u}\| \geq 0 \quad \forall \mathbf{u} \in \mathbb{R}^n \quad$ y $\quad \|\mathbf{u}\| = 0 \quad \Leftrightarrow \quad \mathbf{u} = \vec{0}$
2. $\|\lambda\,\mathbf{u}\| = |\lambda|\,\|\mathbf{u}\| \qquad \forall \lambda \in \mathbb{R}, \quad \forall \mathbf{u} \in \mathbb{R}^n$
3. $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\| \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^n$

Norma-p Dado un vector de $\mathbb{R}^n$, $\mathbf{x} = (x_1, \ldots, x_n)^t$. La Norma-$p$ del vector será:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_1|^p \right)^{1/p} \qquad 1 \leq p < \infty$$

Para distintos valores de p se obtienen distintas variantes de la norma:

$$\begin{aligned} \|\mathbf{x}\|_1 = & \quad \textstyle\sum_{i=1}^{n} |x_i| & p = 1 \\ \|\mathbf{x}\|_2 = & \quad \sqrt{\textstyle\sum_{i=1}^{n} x_i^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}} & p = 2 \quad \text{(Euclidiana)} \end{aligned}$$

Norma Infinito En el caso de que $p = \infty$ podemos tomar el límite a partir de la definición:

$$\begin{aligned} \|\mathbf{x}\|_\infty &= \lim_{p \to \infty} \|\mathbf{x}\|_p \\ \|\mathbf{x}\|_\infty &= \max_i |x_i| \lim_{p \to \infty} \underbrace{\left(\sum_{j=1}^{n} \alpha_j^p \right)^{1/p}}_{1} \qquad \alpha_j = \frac{x_j}{\max_i |x_i|} \in [0,1] \; j = 1, \ldots, n \\ \|\mathbf{x}\|_\infty &= \max_i |x_i| \end{aligned}$$

2.3.2. Norma de matrices

La norma de matrices, en este apuntes será una función definida en el espacio vectorial de las matrices con entradas reales y las operaciones habituales de suma y producto de matrices.

$$(\mathcal{M}_{n\times n}(\mathbb{R}), \mathbb{R}, +, *) \; e.v., \qquad \|\cdot\| : \mathcal{M}_{n\times n}(\mathbb{R}) \longrightarrow \mathbb{R}$$

1. $\|A\| \geq 0 \quad \forall A \in \mathcal{M}_{n\times n}(\mathbb{R}) \quad$ y $\quad \|A\| = 0 \quad \Leftrightarrow \quad A = \vec{0}$
2. $\|\lambda\, A\| = |\lambda|\, \|A\| \qquad \forall \lambda \in \mathbb{R}, \quad \forall A \in \mathcal{M}_{n\times n}(\mathbb{R})$
3. $\|A + B\| \leq \|A\| + \|B\| \quad \forall A, B \in \mathcal{M}_{n\times n}(\mathbb{R})$

Diremos además que una norma de matrices es *submultiplicativa* si $\|AB\| \leq \|A\|\|B\| \quad \forall A, B \in \mathcal{M}_{n\times n}(\mathbb{R})$.

Definición 2.3.1 (Norma compatible)**.** Una norma matricial $\|\cdot\|_M$ es compatible con una vectorial $\|\cdot\|_v$ si

$$\|A\mathbf{x}\|_v \leq \|A\|_M \|\mathbf{x}\|_v \quad \forall A \in \mathcal{M}_{n\times n}(\mathbb{R}), \forall \mathbf{x} \in \mathbb{R}^n$$

A partir de aquí no haremos referencia explícita a qué norma estamos considerando lo que se deducirá del argumento que tome la norma según corresponda.

Definición 2.3.2 (Norma inducida, o norma operador)**.**

$$\|A\| = \max_{\mathbf{x} \neq \vec{0}} \frac{\|A\,\mathbf{x}\|}{\|\mathbf{x}\|}$$

Esta norma es inducida a partir de alguna norma de vectores.

Ejercicio 2.3.1. Verificar que la norma inducida cumple las propiedades de norma.

Podemos ver que esta norma cumple las siguientes propiedades:

Proposición 2.3.1.

$$\|A\| = \max_{\|\mathbf{x}\|=1} \|A\,\mathbf{x}\|$$

Demostración.

$$\|A\| = \max_{\mathbf{x}\neq\vec{0}} \frac{\|A\,\mathbf{x}\|}{\|\mathbf{x}\|} = \max_{\mathbf{x}\neq\vec{0}} \|A\,\frac{\mathbf{x}}{\|\mathbf{x}\|}\| = \max_{\|\mathbf{y}\|=1} \|A\,\mathbf{y}\|$$

□

Comentamos que la propiedad anterior ilustra que el valor de la norma de la matriz es igual al valor de la norma del vector más deformado por la transformación $A\mathbf{x}$.

Proposición 2.3.2. *La norma operador es compatible, es decir,*

$$\|A\,\mathbf{x}\| \leq \|A\|\,\|\mathbf{x}\| \quad \forall A \in \mathcal{M}_{n\times n}(\mathbb{R}), \forall \mathbf{x} \in \mathbb{R}^n$$

Demostración.

$$\|A\,\mathbf{x}\| = \frac{\|A\,\mathbf{x}\|}{\|\mathbf{x}\|}\|\mathbf{x}\| \leq \left(\max_{\mathbf{z}\neq\vec{0}} \frac{\|A\,\mathbf{z}\|}{\|\mathbf{z}\|}\right)\|\mathbf{x}\| = \|A\|\,\|\mathbf{x}\|$$

□

Ejercicio 2.3.2. Demostrar que dadas dos matrices $n \times n$ y la norma inducida dada en 2.3.2, se cumple:

$$\|A\,B\| \leq \|A\|\,\|B\|$$

Ejemplos de normas Las siguientes son algunos ejemplos normas usualmente utilizadas:

- Norma 1:

$$\|A\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$$

- Norma 2:

$$\|A\|_2 = \max_{\|\mathbf{x}\|=1} \sqrt{\mathbf{x}^t A^t A \mathbf{x}} = \sigma_1(A)$$

- Norma Infinito:

$$\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$$

Aquí notamos $\sigma_1(A) = \sqrt{\lambda_1}$, con λ_1 el mayor valor propio de A^tA (que existe y es positivo).

Veamos que $\|A\|_\infty = \text{máx}_i \sum_{j=1}^n |a_{ij}|$.

$$\|A\mathbf{x}\|_\infty = \max_i |\sum_{j=1}^n a_{ij}x_j| \leq \max_i \sum_{j=1}^n |a_{ij}x_j| \leq \max_i \sum_{j=1}^n |a_{ij}|$$

Donde hemos usado en la última inecuación que $\|\mathbf{x}\|_\infty = 1 \Rightarrow \text{máx}_i |x_i| = 1$.

Por tanto, sabemos que $\|A\mathbf{x}\|_\infty \leq \text{máx}_i \sum_{j=1}^n |a_{ij}|$ con $\|\mathbf{x}\|_\infty = 1$. Ahora bien, debemos ver que se cumple la igualdad, es decir, probar que la cota se alcanza.

Esto es, $\exists \mathbf{y} \in \mathbb{R}^n$ tal que $\|A\mathbf{y}\|_\infty = \text{máx}_i \sum_{j=1}^n |a_{ij}|$ con $\|\mathbf{y}\|_\infty = 1$.

Sea i_0 el índice de la fila en el que se da el $\text{máx}_i \sum_{j=1}^n |a_{ij}| = \sum_{j=1}^n |a_{i_0 j}|$, entonces tomando $\mathbf{y}^t = (sg(a_{i_0 1}), sg(a_{i_0 2}), \ldots, sg(a_{i_0 n}))^t$. En tal caso, $\|\mathbf{y}\|_\infty = 1$ y además $\|A\mathbf{y}\|_\infty = \text{máx}_i \sum_{j=1}^n |a_{ij}|$. La cota se alcanza y por lo tanto es el máximo.

Es así que se concluye que $\|A\|_\infty = \text{máx}_i \sum_{j=1}^n |a_{ij}|$.

Ejercicio 2.3.3. Probar que $\|A\|_2 = \sigma_1(A) = \sqrt{\lambda_1}$, con λ_1 el mayor valor propio de A^tA. Puede ser de utilidad para completar la demostración formalizar:

1. $\|A\mathbf{x}\|_2^2 = (A\mathbf{x})^t(A\mathbf{x}) = \mathbf{x}^tA^tA\mathbf{x}$.
2. A^tA es simétrica y definida positiva ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$).
3. Si v_i son los vectores propios tal que $\|v_i\| = 1$, $v_i \perp v_j \quad \forall i \neq j$, $\mathbf{x} = \sum_{i=1}^n \alpha_i v_i$, con $\|\mathbf{x}\|_2 = 1 \Rightarrow \sum_{i=1}^n \alpha_i^2 = 1$.

Definición 2.3.3 (Radio espectral)**.** Sea A una matriz cuadrada $A \in \mathcal{M}_{n\times n}(\mathbb{R})$. Su radio espectral ρ está definido como el máximo de los valores absolutos de sus valores propios.

$$\rho(A) = \max_i |\lambda_i| \qquad A\mathbf{v}_i = \lambda_i \mathbf{v}_i \quad \forall i = 1, \ldots, n$$

Proposición 2.3.3. *La norma operador está acotada inferiormente por su radio espectral:*

$$\rho(A) \leq \|A\|$$

Demostración. Sea λ valor propio de A tal que $|\lambda| = \rho(A)$, y $\mathbf{v}$ correspondiente vector propio de norma 1. Entonces:

$$\|A\| = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\| \geq \|A\mathbf{v}\| = \|\lambda\mathbf{v}\| = |\lambda| = \rho(A). \tag{2.5}$$

□

Teorema 2.3.4 (Teorema del Radio Espectral)**.** *Sea A una matriz cuadrada $A \in \mathcal{M}_{n\times n}(\mathbb{R})$. Para todo $\varepsilon > 0$ existe alguna norma consistente $\|\cdot\|$ tal que*

$$\|A\| < \rho(A) + \varepsilon$$

Observación 2.3.1. El teorema 2.3.4 es equivalente a decir que el radio espectral es el ínfimo de todas las normas consistentes de una matriz.

2.3.3. Número de condición

Definición 2.3.4 (Número de condición de una matriz)**.** Dada una matriz cuadrada $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ decimos que su número de condición k está dado por la siguiente expresión

$$k(A) = \|A\| \left\|A^{-1}\right\|$$

Observación 2.3.2. $k(A) \geq 1$: $k(A) = \|A\|\|A^{-1}\| \geq \|AA^{-1}\| = \|Id\| = 1$

2.3.4. Análisis de perturbaciones

Debido a los errores de representación, en vez de resolver un sistema $A\mathbf{x} = \mathbf{b}$ estaremos resolviendo $(A + \delta_A)\mathbf{x} = (\mathbf{b} + \delta_\mathbf{b})$. Esto significa que existirán perturbaciones tanto en A como en $\mathbf{b}$, y la solución hallada se apartará de la original en $\mathbf{x} + \delta\mathbf{x}$. Analizaremos el error relativo de estas perturbaciones.

Aplicación a estimación de error ($\mathbf{b} + \delta_\mathbf{b}$): Si consideramos un sistema de ecuaciones lineales y adicionamos solamente un vector de errores $\delta_\mathbf{b}$ al término independiente, obtendremos una solución que consistirá en la solución sin error, $\mathbf{x}$ con un vector de errores adicionado $\delta_\mathbf{x}$

$$A\left(\mathbf{x} + \delta_\mathbf{x}\right) = \mathbf{b} + \delta_\mathbf{b}$$

dado que suponemos que A es invertible podemos escribir

$$\mathbf{x} + \delta_\mathbf{x} = A^{-1}(\mathbf{b} + \delta_\mathbf{b}) = A^{-1}\,\mathbf{b} + A^{-1}\,\delta_\mathbf{b}$$

dado que $A^{-1}\,\mathbf{b} = \mathbf{x}$ anulamos el x de ambos lados y aplicamos norma, obteniendo

$$\|\delta_\mathbf{x}\| = \|A^{-1}\,\delta_\mathbf{b}\| \leq \|A^{-1}\|\,\|\delta_\mathbf{b}\|$$

desigualdad que puede ser dividida por la norma de x para obtener

$$\frac{\|\delta_\mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{\|A^{-1}\|\,\|\delta_\mathbf{b}\|}{\|\mathbf{x}\|}$$

multiplicamos y dividimos por $\|A\|$ y obtenemos el número de condición en el numerador

$$\frac{\|\delta_\mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{k(A)\,\|\delta_\mathbf{b}\|}{\|A\|\,\|\mathbf{x}\|}$$

por otra parte es simple ver que se cumple

$$\frac{1}{\|A\|\,\|\mathbf{x}\|} \leq \frac{1}{\|\mathbf{b}\|}$$

por lo tanto obtenemos

$$\boxed{\frac{\|\delta_\mathbf{x}\|}{\|\mathbf{x}\|} \leq k(A)\,\frac{\|\delta_\mathbf{b}\|}{\|\mathbf{b}\|}}$$

Aplicación a estimación de error $(A+\delta_A)$: Si ahora consideramos una perturbación en A:

$$(A+\delta_A)(\mathbf{x}+\delta_\mathbf{x}) = \mathbf{b} \Rightarrow (A+\delta_A)\delta_\mathbf{x} + A\mathbf{x} + \delta_A\mathbf{x} = \mathbf{b}$$

$$A\delta\mathbf{x} + \delta_A(\mathbf{x}+\delta_\mathbf{x}) = 0 \Rightarrow \delta_\mathbf{x} = -A^{-1}\delta_A(\mathbf{x}+\delta_\mathbf{x})$$

Aplicando normas:

$$\|\delta_\mathbf{x}\| \leq \|A^{-1}\|\|\delta_A\|\|\mathbf{x}+\delta_\mathbf{x}\| \Rightarrow \boxed{\frac{\|\delta_\mathbf{x}\|}{\|\mathbf{x}+\delta_\mathbf{x}\|} \leq k(A)\frac{\|\delta_A\|}{\|A\|}}$$

En general el número de condición de A se estima por otro método (si el problema está mal condicionado díficilmente pueda conocer A^{-1}).

Ejemplo 2.3.1. Si el error relativo en $\mathbf{b}$ es bajo, $\frac{\|\delta_\mathbf{b}\|}{\|\mathbf{b}\|} = 10^{-3}$, pero el número de condición es alto, $k(A) = 10^6$, entonces, la perturbación a la salida puede llegar a ser grande: $\frac{\|\delta_\mathbf{x}\|}{\|\mathbf{x}\|} \leq 10^3$. $\triangle$

Error residual: Sea $\mathbf{x}^*$ una aproximación a la solución de $A\mathbf{x} = \mathbf{b}$, definimos el residuo $\mathbf{r} = b - A\mathbf{x}^*$.

Nos preguntamos, ¿si el residuo es "pequeño", se cumplirá qué el error en la solución es también "pequeño"?

Veamos:

$$\mathbf{r} = b - A\mathbf{x}^* = A\mathbf{x} - A\mathbf{x}^* = A(\mathbf{x}-\mathbf{x}^*) \Rightarrow \mathbf{x}-\mathbf{x}^* = A^{-1}\mathbf{r}$$

Por otra parte:

$$\|\mathbf{r}\| = \|A(\mathbf{x}-\mathbf{x}^*)\| \leq \|A\|\|\mathbf{x}-\mathbf{x}^*\|$$

Por lo que combinando ambas ecuaciones:

$$\frac{\|\mathbf{r}\|}{\|A\|} \leq \|\mathbf{x}-\mathbf{x}^*\| \leq \|A^{-1}\|\|\mathbf{r}\|$$

Por otra parte:

$$A\mathbf{x} = \mathbf{b} \Rightarrow \|\mathbf{b}\| \leq \|A\|\|\mathbf{x}\|$$
$$\mathbf{x} = A^{-1}\mathbf{b} \Rightarrow \|\mathbf{x}\| \leq \|A^{-1}\|\|\mathbf{b}\|$$

Por lo que combinando ambas ecuaciones:

$$\frac{\|\mathbf{b}\|}{\|A\|} \leq \|\mathbf{x}\| \leq \|A^{-1}\|\|\mathbf{b}\|$$

Finalmente, juntando todo se llega a:

$$\frac{\|\mathbf{r}\|}{\|b\|}\frac{1}{\|A\|\|A^{-}1\|} \leq \frac{\|\mathbf{x}-\mathbf{x}^*\|}{\|\mathbf{x}\|} \leq \|A^{-1}\|\|A\|\frac{\|\mathbf{r}\|}{\|b\|}$$

$$\frac{\|\mathbf{r}\|}{\|b\|}\frac{1}{k(A)} \leq \frac{\|\mathbf{x}-\mathbf{x}^*\|}{\|\mathbf{x}\|} \leq k(A)\frac{\|\mathbf{r}\|}{\|b\|}$$

Mencionamos como conclusiones que si $\mathbf{r}$ es pequeño y $k(A)$ es elevado (matriz mal condicionada), entonces el residuo no da información sobre la calidad de la solución x^*.

Por otra parte, si $\mathbf{r}$ es pequeño y $k(A)$ es bajo (matriz bien condicionada), entonces el residuo da información sobre la calidad de la solución x^*.

Ejemplo 2.3.2. Sea $A = \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}$, y el sistema $A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Entonces $A^{-1} = \begin{bmatrix} \frac{-1}{1-\epsilon} & \frac{1}{1-\epsilon} \\ \frac{1}{1-\epsilon} & \frac{-\epsilon}{1-\epsilon} \end{bmatrix}$.

Calculando $k(A)$ usando $\|\cdot\|_\infty$ obtenemos que $\|A\|_\infty = \text{máx}\{1+|\epsilon|, 2\}$ y que $\|A^{-1}\|_\infty = \text{máx}\{\frac{2}{|1-\epsilon|}, \frac{1-|\epsilon|}{|1-\epsilon|}\}$.

Entonces, para $\epsilon \approx 1$, tenemos que $\|A\|_\infty \approx 2$ y $\|A^{-1}\|_\infty \approx \frac{2}{|1-\epsilon|}$.

Así, $k(A) \approx \frac{4}{|1-\epsilon|}$ que va a ser muy grande (ya que $\epsilon \approx 1$). △

Ejemplo 2.3.3. Sea $A = \begin{bmatrix} 1,2969 & 0,8648 \\ 0,2161 & 0,1441 \end{bmatrix}$, y $\mathbf{b} = \begin{bmatrix} 0,8642 \\ 0,1440 \end{bmatrix}$.

Supongamos que $\mathbf{x}^* = \begin{bmatrix} 0,9911 \\ -0,4870 \end{bmatrix}$.

Entonces $\mathbf{r} = \mathbf{b} - A\mathbf{x}^* = \begin{bmatrix} -10^{-8} \\ 10^{-8} \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Sin embargo la solución real es $\mathbf{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

En este caso lo que ocurre es que $k(A) \approx 3,3 \times 10^8$, por lo que $3,35 \times 10^{-17} \leq \frac{\|\mathbf{x}-\mathbf{x}^*\|}{\|\mathbf{x}\|} \leq 3,78$. △

2.4. Métodos indirectos

Los métodos indirectos como mencionamos al inicio del capítulo se basan en aproximar sucesivamente la solución a un sistema. Ya tenemos una noción de cercanía de acuerdo a lo trabajado en la sección anterior, por lo que tenemos los ingredientes para establecer si una sucesión es convergente o no.

2.4.1. Método de Jacobi

Consideremos un sistema $A\mathbf{x} = \mathbf{b}$, con $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ invertible, $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Sabemos que para la solución real se cumple: $b_i = \sum_{j=1}^{n} a_{ij}x_j$.

$b_i = a_{ii}x_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j = b_i$

De donde, asumiendo $a_{ii} \neq 0$: $x_i = \frac{b_i - \sum_{j=1 j\neq i}^{n} a_{ij}x_j}{a_{ii}} \quad \forall i = 1, \ldots, n.$

Esta igualdad no tiene sentido, ya que requiere conocer el vector $\mathbf{x}$ para hallar el vector $\mathbf{x}$.

Sin embargo podemos generar un método iterativo de la forma:

$$\text{(Jacobi):} \begin{cases} x_i^{k+1} = \frac{b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j^k}{a_{ii}} & \forall i = 1, \ldots, n \\ \mathbf{x}^0 \text{ punto inicial} & \mathbf{x}^0 = \begin{bmatrix} x_1^0 \\ \vdots \\ x_n^0 \end{bmatrix}. \end{cases}$$

Este método se llama Método de Jacobi y el punto $\mathbf{x}^0$ es el punto inicial. Ahora bien, podríamos preguntarnos ¿será que esta sucesión así definida converge a la solución del sistema $A\mathbf{x} = \mathbf{b}$? Veremos más adelante que, efectivamente, si se cumplen ciertas condiciones sobre la matriz A el método converge. Estudiaremos también cómo se debe elegir el punto $\mathbf{x}^0$, si es necesario que esté en alguna región particular, cuál es la velocidad de convergencia del método, etc.

Antes de continuar observemos otra forma de escribir la iteración del método:

$$x_i^{k+1} = x_i^k + \frac{b_i - \sum_{j=1}^{n} a_{ij}x_j^k}{a_{ii}} \quad \forall i = 1, \ldots, n$$

Implementación de Jacobi:

Algoritmo 2 Pseudo-código: Jacobi

Sea A una matriz con entradas $A_{i,j} = a(i,j)$, y vectores $\mathbf{b}$ y $\mathbf{x}^0$

```
k = 1
error = ínf
while error > tolerancia & k < max_iteraciones do
  for i = 1 → n do
    x_i^{k+1} = x_i^k + (b_i − Σ_{j=1}^{n} a_ij x_j^k) / a_ii
  end for
  error = norm(x^{k+1} − x^k)
  k = k + 1
end while
```

Los valores de tolerancia del error y máximo de iteraciones se determinarán de acuerdo a la experiencia del usuario y requerimientos del problema.

2.4.2. Método de Gauss-Seidel (GS)

El método de Gauss-Seidel introduce una variante del método anterior. Para ello, es importante observar que el método de Jacobi, requiere conocer completamente el vector $\mathbf{x}^k$ para calcular

$\mathbf{x}^{k+1}$. Sin embargo, las entradas del vector $\mathbf{x}^{k+1}$ se van calculando una a una, y podrían utilizarse para para calcular la entrada siguiente, en lugar de utilizar las del vector anterior. Es decir, para calcular x_i^{k+1} se precisan $x_1^k, x_2^k, \ldots, x_{i-1}^k, x_{i+1}^k, \ldots, x_n^k$. Pero las primeras $x_1^{k+1}, x_2^{k+1}, \ldots, x_{i-1}^{k+1}$ ya están calculadas. Por tanto, el método de Gauss-Seidel hace uso de esa información y en general veremos que se mejora el método.

Explícitamente, la iteración queda expresada:

$$(Gauss-Seidel): \begin{cases} x_i^{k+1} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij} x_j^k}{a_{ii}} & \forall i = 1, \ldots, n \\ \\ \mathbf{x}^0 \text{ punto inicial} & \mathbf{x}^0 = \begin{bmatrix} x_1^0 \\ \vdots \\ x_n^0 \end{bmatrix}. \end{cases}$$

Observación 2.4.1. Nuevamente debemos pedir $a_{ii} \neq 0$.

Ejercicio 2.4.1. Modificar el código de Jacobi para implementar Gauss-Seidel.

2.4.3. Expresión Matricial de Jacobi y Gauss-Seidel

Dado un sistema de ecuaciones lineales

$$A\,\mathbf{x} = \mathbf{b}, \qquad A = \begin{pmatrix} \searrow & -F \\ D & \\ -E & \searrow \end{pmatrix}$$

aplicamos la descomposición de la matriz A en sus componentes triangular inferior $-E$, diagonal D y triangular superior $-F$.

Veamos cómo es posible escribir los métodos vistos hasta el momento en forma matricial.

Jacobi:

$$A = D - E - F \qquad \Rightarrow \qquad D\mathbf{x} = (E+F)\mathbf{x} + \mathbf{b}$$

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

lo podemos escribir como

$$\begin{cases} \mathbf{x}^{(k+1)} = Q_J\,\mathbf{x}^{(k)} + \mathbf{r}_J & Q_J = D^{-1}(E+F) \\ \mathbf{x}^{(0)} = \mathbf{x}_0 & \mathbf{r}_J = D^{-1}\,\mathbf{b} \end{cases}$$

Gauss-Seidel:

$$A = D - E - F \qquad \Rightarrow \qquad (D-E)\mathbf{x} = F\mathbf{x} + \mathbf{b}$$

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

lo podemos escribir como

$$\begin{cases} \mathbf{x}^{(k+1)} = Q_{GS}\,\mathbf{x}^{(k)} + \mathbf{r}_{GS} & Q_{GS} = (D-E)^{-1}F \\ \mathbf{x}^{(0)} = \mathbf{x}_0 & \mathbf{r}_{GS} = (D-E)^{-1}\,\mathbf{b} \end{cases}$$

2.4.4. Método Iterativo Matricial

Los métodos indirectos para resolver sistemas lineales son iterativos. Por lo tanto resulta útil expresar las operaciones que realizan de forma $\mathbf{x}^{(k+1)} = g(\mathbf{x}^{(k)})$. Dado que $\mathbf{x}^{(k)}$ es un vector en el caso general, se trabaja con matrices, quedando expresado de la siguiente forma:

$$(M): \begin{cases} \mathbf{x}^{(k+1)} = Q\mathbf{x}^{(k)} + \mathbf{r} & \mathbf{x}^{(k)} \in \mathbb{R}^n \quad k = 0, 1, \dots \\ \mathbf{x}^{(0)} = \mathbf{x}_0 & Q \in \mathcal{M}_{n\times n}(\mathbb{R}),\ \mathbf{r} \in \mathbb{R}^n \end{cases}$$

En general, dado el sistema $A\mathbf{x} = \mathbf{b}$, elegimos una matriz $M \in \mathcal{M}_{n\times n}(\mathbb{R})$ invertible y planteamos:

$$A\mathbf{x} = \mathbf{b} \Leftrightarrow M\mathbf{x} = (M - A)\mathbf{x} + b$$

En los métodos iterativos se escoge una matriz M relacionada con A (observar Jacobi y G-S) y se genera una sucesión de vectores $\{\mathbf{x}^{(k)}\}_{k\geq 0}$ a partir de la ecuación

$$M\mathbf{x}^{(k+1)} = (M - A)\mathbf{x}^{(k)} + b$$

Si $\{\mathbf{x}^{(k)}\}_{k\geq 0}$ resulta convergente, la convergencia será hacia la solución de $A\mathbf{x} = \mathbf{b}$. Veremos que muchas veces es posible elegir un $\mathbf{x}^{(0)}$ inicial arbitrario (cualquiera).

Por tanto, si escribimos el sistema como $M\mathbf{x}^{(k+1)} = (M - A)\mathbf{x}^{(k)} + b$, la iteración estacionaria será:

$$\begin{cases} \mathbf{x}^{(k+1)} = Q\,\mathbf{x}^{(k)} + \mathbf{r} & Q = M^{-1}(M - A) \\ \mathbf{x}^{(0)} = \mathbf{x}_0 & \mathbf{r} = M^{-1}\,\mathbf{b} \end{cases}$$

Observación 2.4.2. Es una iteración estacionaria porque tanto Q como r no dependen del paso k, y es de orden 1 porque $\mathbf{x}^{(k+1)}$ depende solamente del valor anterior $\mathbf{x}^{(k)}$ (y no de pasos anteriores).

Observación 2.4.3. $\lim_{k\to\infty} M\mathbf{x}^{(k+1)} = \lim_{k\to\infty}(M - A)\mathbf{x}^{(k)} + b$. Si $\lim_{k\to\infty} \mathbf{x}^{(k+1)} = \mathbf{x}^*$ tenemos que: $\lim_{k\to\infty} M\mathbf{x}^{(k+1)} = M\mathbf{x}^* = (M - A)\mathbf{x}^* + b = \lim_{k\to\infty}(M - A)\mathbf{x}^{(k)} + b$.

Definición 2.4.1 (Punto Fijo)**.** Dado un método iterativo matricial (M). Diremos que $\mathbf{x}^*$ es un punto fijo del mismo si y solo si

$$\mathbf{x}^* = Q\mathbf{x}^* + r$$

En este caso decimos que el método iterativo es consistente.

Lema 2.4.1. *Dada una matriz cuadrada* $A \in \mathcal{M}_{n\times n}(\mathbb{R})$

$$\lim_{k\to\infty} A^k = 0 \quad \Leftrightarrow \quad \rho(A) < 1$$

Observación 2.4.4. Notamoms que aquí 0 representa la matriz nula.

Demostración. ($\Rightarrow$) Por absurdo supongamos que $\rho(A) \geq 1$, entonces existe λ valor propio de A tal que $|\lambda| \geq 1$ y $Av = \lambda v$, con v vector propio asociado a λ.

$$\Rightarrow \|A^k\| \geq \frac{\|A^k v\|}{\|v\|} = |\lambda|^k \xrightarrow[k\to+\infty]{} +\infty \Rightarrow \lim_{k\to\infty} A^k \neq 0$$

($\Leftarrow$) Como $\rho(A) < 1$, entonces existe $\varepsilon > 0$ tal que $\rho(A) < 1 - \varepsilon$ y además existe una norma inducida $\|\cdot\|_\varepsilon$ que cumple $\|A\|_\varepsilon \leq \rho(A) + \varepsilon < 1$. Ahora, $\|A^k\| \leq \|A\|^k < 1$ y entonces $\|A^k\|$ está acotada por $\|A\|^k$ que tiende a cero con k. Así $\lim_{k\to\infty} A^k = 0$. □

Observación 2.4.5. Hay métodos eficientes para estimar el radio espectral.

Denotaremos al vector de error en el paso k-ésimo como $e^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$.

Proposición 2.4.2. $\lim_k \mathbf{x}^{(k)} = \mathbf{x}^*$ *sii* $\lim_k e^{(k)} = \vec{0}$ *sii* $\lim_k \|e^{(k)}\| = 0$

Teorema 2.4.3. *La iteración estacionaria* $\begin{cases} \mathbf{x}^{(k+1)} = Q\mathbf{x}^{(k)} + \mathbf{r} \\ \mathbf{x}^{(0)} \in \mathbb{R}^n \end{cases}$ *es convergente sii* $\rho(Q) < 1$.

Demostración. Sea la solución del sistema $\mathbf{x}^* \in \mathbb{R}^n$ tal que $\mathbf{x}^* = Q\mathbf{x}^* + \mathbf{r}$ y $e^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$ el error en el paso k. Utilizando que $\mathbf{x}^*$ es punto fijo de la iteración tenemos que:

$$e^{(k+1)} = \mathbf{x}^{(k+1)} - \mathbf{x}^* = Q\mathbf{x}^{(k)} + r - (Q\mathbf{x}^* + \mathbf{r}) = Q(\mathbf{x}^{(k)} - \mathbf{x}^*) = Qe^{(k)}.$$

Luego, por inducción en los naturales tenemos que $e^{(k)} = Q^k e^{(0)}$. Vamos ahora a probar el directo y el recíproco en partes:

($\Rightarrow$) Supongamos por absurdo que $\rho(Q) \geq 1$. En tal caso, existe un valor propio λ de Q, con $|\lambda| \geq 1$, y un vector propio $v \neq 0$ tal que $Qv = \lambda v$. Elijamos $\mathbf{x}^{(0)}$ de modo que $e^{(0)} = v$. Esto es posible tomando $\mathbf{x}^{(0)} = \mathbf{x}^* + e^{(0)}$. Por lo anteriormente observado, tenemos que:

$$e^{(k)} = Q^k e^{(0)} = Q^k v = \lambda^k v,$$

y tomando normas, tenemos que $\|e^{(k)}\| = |\lambda|^k \|v\|$. Tomando límites en ambos miembros, se consigue que $\lim_k \|e^{(k)}\| \neq 0$, pues $|\lambda| \geq 1$ y $\|v\| \neq 0$. Esto es decir que el error no tiende al vector nulo, o equivalentemente, que la sucesión $\{x^k\}_{k\in\mathbb{N}}$ no converge a $\mathbf{x}^*$, en contradicción con la hipótesis.

($\Leftarrow$) Por el Teorema del Radio Espectral, el radio espectral es el ínfimo de las normas operadores. Sea ε igual a la mitad de la distancia entre $\rho(Q)$ y 1, es decir, $\varepsilon = \frac{1-\rho(Q)}{2}$. Por definición de ínfimo, existe una norma operador $\|\cdot\|_\varepsilon$ tal que $\|Q\|_\varepsilon - \rho(Q) < \varepsilon$. Pero entonces:

$$\|Q\|_\varepsilon < \rho(Q) + \varepsilon = \frac{2\rho(Q) + (1 - \rho(Q))}{2} = \frac{1 + \rho(Q)}{2} < 1.$$

Hemos conseguido así una norma $\|\cdot\|_\varepsilon$ compatible con una vectorial tal que $\|Q\|_\varepsilon < 1$. Como $e^{(k)} = Q^k e^{(0)}$, tomando normas en cada miembro tenemos que:

$$\|e^{(k)}\| = \|Q^k e^{(0)}\| \leq (\|Q\|_\varepsilon)^k \|e^{(0)}\|,$$

y tomando límite con k tenemos que $0 \leq \lim_k \|e^{(k)}\| \leq (\|Q\|_\varepsilon)^k \|e^{(0)}\| = 0$. La única opción válida es que $\lim_k \|e^{(k)}\| = 0$, y por la primera propiedad de una norma tenemos que la sucesión de vectores $\{e^{(k)}\}_{k\in\mathbb{N}}$ converge al vector nulo. Esto significa que $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}}$ converge a $\mathbf{x}^*$.

□

Observación 2.4.6. Recalcamos que esto significa que $\mathbf{x}^{(k)}$ converge independientemente del dato inicial $\mathbf{x}^{(0)}$.

Podemos ver como corolario que si en alguna norma $\|Q\| < 1 \Rightarrow \rho(Q) \leq \|Q\| < 1$, la iteración es convergente.

El teorema anterior es de gran utilidad ya que permite establecer si un método será convergente o no dependiendo de la matriz Q, para cualquier método que pueda escribirse como una ecuación estacionaria. Sin embargo debemos encontrar la matriz Q asociada al sistema $A\mathbf{x} = \mathbf{b}$. A continuación veremos algunos criterios para determinar la convergencia a partir de características de la matriz A.

Definición 2.4.2 (Matriz diagonal dominante)**.** Sea A una matriz $n \times n$, decimos que es diagonal dominante por filas si y solo si

$$\sum_{j=1, j\neq i}^{n} |a_{ij}| < |a_{ii}| \qquad i = 1, \ldots, n$$

Proposición 2.4.4 (Convergencia de Jacobi)**.** *A es diagonal dominante por filas (o por columnas) $\Rightarrow$ la sucesión generada por Jacobi converge.*

Demostración. Caso filas:

$$\sum_{j=1, j\neq i}^{n} |a_{ij}| < |a_{ii}| \Rightarrow \sum_{j=1, j\neq i}^{n} \frac{|a_{ij}|}{|a_{ii}|} < 1 \Rightarrow \|Q_J\|_\infty < 1 \Rightarrow \text{ Jacobi converge}$$

Caso columnas: Ejercicio. □

Proposición 2.4.5 (Convergencia de Gauss-Seidel)**.** *A es diagonal dominante por filas (o por columnas) $\Rightarrow$ la sucesión generada por Gauss-Seidel converge.*

Observación 2.4.7. Las proposiciones anteriores indican que si la matriz A es estrictamente diagonal dominante por filas (o por columnas), entonces tanto Jacobi como Gauss-Seidel son convergentes, para toda condición inicial x_0. Esto es una condición suficiente. Es decir que el método podría ser convergente aunque A no sea estrictamente diagonal dominante.

Ejercicio 2.4.2. Se considera $A = \begin{pmatrix} 3 & -1 \\ 1 & \beta \end{pmatrix}$. Sin calcular $\rho(Q)$, indicar un rango de valores de β que asegure convergencia de Jacobi y Gauss-Seidel.

Observación 2.4.8.

- Para matrices esparsas, los métodos iterativos permiten encontrar la solución en forma rápida y eficiente.
- No se usan para matrices mal condicionadas.
- Hay variantes para acelerar la convergencia.
- Típicamente Gauss-Seidel es más rápido que Jacobi.
- El radio espectral determina la velocidad de convergencia.

Velocidad de convergencia de MIG

Dado un método iterativo, de la forma $\begin{cases}\mathbf{x}^{(k+1)} = Q\mathbf{x}^{(k)} + \mathbf{r} \\ \mathbf{x}^{(0)} = \mathbf{x}_0\end{cases}$ con $\mathbf{x}^{(k)}, \mathbf{r} \in \mathbb{R}^n$, $Q \in \mathcal{M}_{n\times n}(\mathbb{R})$.

Sean $\{\lambda_1, \ldots, \lambda_n\}$ los valores propios de Q que asumiremos reales y satisfaciendo la relación: $\rho(Q) = |\lambda_1| > |\lambda_2| > \cdots > |\lambda_{n-1}| > |\lambda_n| \geq 0$. Sean $\{v_1, \ldots, v_n\}$ los vectores propios de Q asociados a dichos valores propios. Entonces, si

$$\left.\begin{array}{l} e^{(k+1)} = \mathbf{x}^{(k+1)} - \mathbf{x}^* \\ \rho(Q) < 1 \end{array}\right\} \Rightarrow \frac{\|e^{(k+1)}\|}{\|e^{(k)}\|} \xrightarrow[k\to\infty]{} \rho(Q)$$

$e^{(k)}$ es el error en la aproximación en el paso k-ésimo.

Supongamos que $e^{(0)} = \sum_{i=1}^n \alpha_i v_i$, sabemos que el error en el k-ésimo paso satisface $e^{(k)} = Q^k e^{(0)}$. Con lo cual:

$$e^{(k)} = \sum_{i=1}^n Q^k \alpha_i v_i = \sum_{i=1}^n (\lambda_i)^k \alpha_i v_i = (\lambda_1)^k \alpha_1 v_1 + \sum_{i=2}^n (\lambda_i)^k \alpha_i v_i,$$

de la misma manera tenemos que:

$$e^{(k+1)} = (\lambda_1)^{k+1} \alpha_1 v_1 + \sum_{i=2}^n (\lambda_i)^{k+1} \alpha_i v_i.$$

Entonces: $\lim_{k\to+\infty} \frac{\|e^{(k+1)}\|}{\|e^{(k)}\|} = \lim_{k\to+\infty} \frac{\|(\lambda_1)^{k+1}\alpha_1 v_1 + (\lambda_1)^{k+1}\sum_{i=2}^n \frac{(\lambda_i)^{k+1}}{(\lambda_1)^{k+1}}\alpha_i v_i\|}{\|(\lambda_1)^k \alpha_1 v_1 + (\lambda_1)^k \sum_{i=2}^n \frac{(\lambda_i)^k}{(\lambda_1)^k}\alpha_i v_i\|} =$

$$= \lim_{k\to+\infty} \frac{\|(\lambda_1)^{k+1}\alpha_1 v_1\|}{\|(\lambda_1)^k \alpha_1 v_1\|} = |\lambda_1|.$$

Ejemplo 2.4.1. Si $\rho(Q_1) = 0,4$ y $\rho(Q_2) = 0,75$ el método convergerá más rápido para Q_1. △

Ejemplo 2.4.2. Calcular y comparar los valores de $\rho(Q_J)$ y $\rho(Q_{GS})$ para $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$.

En primer lugar, sabemos que ambos métodos son convergentes por ser la matriz A diagonal dominante.

$Q = M^{-1}(M - A)$. Denotando $A = D - E - F$, donde $-E$ es la matriz subdiagonal inferior y $-F$ es la matriz por encima de la diagonal D; en Jacobi se tiene: $Q_J = D^{-1}(E + F)$ (se toma $M = D$). En Gauss-Seidel se tiene: $Q_{GS} = (D - E)^{-1}(F)$ (se toma $M = D - E$).

Tendremos que: $Q_J = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}$, y sus valores propios son $1/2$ y $-1/2$ con lo cual $\rho(Q_J) = \frac{1}{2}$.

Además $Q_{GS} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix}$, y sus valores propios son 0 y $1/4$ con lo cual $\rho(Q_{GS}) = \frac{1}{4}$.

Gauss-Seidel convergerá el doble de rápido que Jacobi en este caso. △

Condición de parada de MIG

Nos preguntamos a continuación cómo saber cuándo detener la iteración, ya que no conocemos el valor real de la solución, es decir, no podemos imponer $\|e^{(k)}\| = \|\mathbf{x}^{(k)} - \mathbf{x}^*\| < \varepsilon$. Podríamos imponer $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \varepsilon$, pero qué relación tendría esto con la condición usando la solución verdadera:

$$\begin{aligned}\mathbf{x}^{(k+1)} - \mathbf{x}^* &= Q(\mathbf{x}^{(k)} - \mathbf{x}^*) \\ &= -Q(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) + Q(\mathbf{x}^{(k+1)} - \mathbf{x}^*)\end{aligned}$$

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| \leq \|Q\|\|(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})\| + \|Q\|\|(\mathbf{x}^{(k+1)} - \mathbf{x}^*)\|$$

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|(1 - \|Q\|) \leq \|Q\|\|(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})\|$$

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| \leq \frac{\|Q\|}{1 - \|Q\|}\|(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})\|$$

Es así que como decíamos, imponiendo $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \varepsilon$ obtendríamos una cota para el error en el paso $k+1$ que está relacionado con la diferencia entre los valores calculados en pasos anteriores:

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| \leq \frac{\|Q\|}{1 - \|Q\|}\,\varepsilon$$

Como caso particular, si $\|Q\| < \frac{1}{2}$, entonces $\frac{\|Q\|}{1-\|Q\|} < 1$, por lo que para alcanzar una cota de error ε alcanza que $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \approx \varepsilon$.

2.5. Métodos de Sobrerrelajación

Las técnicas de sobrerrelajación para resolución de sistemas lineales usando métodos iterativos tienen el cometido es lograr convergencia partiendo de métodos iterativos que no resultan convergentes, o acelerar la velocidad de convergencia de los que sí son convergentes. Para ello, se realiza una especie relajación convexa entre los puntos hallados en los pasos k y el punto $k+1$ hallado por el método elegido:

$$\mathbf{x}^{k+1} = \omega \mathbf{x}_{(M)}^{(k+1)} + (1-\omega)\mathbf{x}^{(k)} \quad \forall i = 1, \ldots, n$$

El valor de ω se escoge optimizando la velocidad de convergencia.

Observación 2.5.1. Si $\omega > 1$ se llama sobrerrelajación (aceleran convergencia de Jacobi y G-S).

Si $\omega < 1$ se llama subrelajación (Jacobi y G-S no convergen).

2.5.1. Sobrerrelajación de Jacobi

La iteración queda:

$$x_i^{k+1} = \frac{\omega}{a_{ii}} \left[b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^k \right] + (1-\omega) x_i^k \quad \forall i = 1, \ldots, n$$

Y la matriz de iteración:

$$Q = \omega Q_j + (1-\omega) Id$$

Ejemplo 2.5.1. El método de Jacobi no converge para un sistema $Ax = b$ con matriz:

$$A = \begin{pmatrix} 1 & -6 \\ 2 & 3 \end{pmatrix}.$$

y vector $b = (1, 0)^t$.

Esto puede deducirse ya que aunque en este caso la matriz A no es diagonal dominante, se puede utilizar el criterio de $\rho(Q_J)$. La matriz de iteración es:

$$Q_J = \begin{pmatrix} 0 & 6 \\ -\frac{2}{3} & 0 \end{pmatrix}$$

Sus valores propios son $\lambda = \pm 2i$. Por lo tanto $\rho(Q_J) = 2 > 1$ y se concluye que el método no es convergente.

Sin embargo, consideremos la relajación del método de Jacobi donde $Q = \omega Q_j + (1-\omega) Id$ es la nueva matriz de iteración. Analicemos si existe $\omega > 0$ que garantice convergencia de este método para el sistema lineal anterior. Ahora:

$$Q = \begin{pmatrix} 1-\omega & 6\omega \\ -\frac{2}{3}\omega & 1-\omega \end{pmatrix}$$

Sus valores propios son $\lambda = (1-\omega) \pm 2i\omega$. Busquemos $\omega > 0$ para que ambos valores propios tengan magnitud inferior a 1. Tenemos que $|\lambda|^2 = (1-\omega)^2 + 4\omega^2 = 5\omega^2 - 2\omega + 1 < 1$, o equivalentemente, $5\omega^2 - 2\omega < 0$. Usando que $\omega > 0$ y factorizando, tenemos que si $\omega \in (0, 2/5)$ se asegura convergencia. Conseguimos así relajar el método de Jacobi en un caso que no era convergente y traducirlo a otro método que sí es convergente. △

2.5.2. Sobrerrelajación sucesiva

El método SOR (por *Successive Over-Relaxation*) es la aplicación a Gauss-Seidel de una relajación análoga a la realizada en la sección anterior.

La iteración queda:

$$x_i^{k+1} = \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij} x_j^k \right] + (1-\omega) x_i^k \quad \forall i = 1, \ldots, n$$

Se toma $M = \frac{1}{\omega} D - E,\ \omega \in (1,2)$ ($\omega = 1$ es GS).

Ejemplo 2.5.2.

$$A = \begin{bmatrix} 2 & 2 & 2 \\ -3 & 3 & 5 \\ -2 & 4 & -2 \end{bmatrix} \Rightarrow M = \begin{bmatrix} \frac{2}{\omega} & 0 & 0 \\ -3 & \frac{3}{\omega} & 0 \\ -2 & 4 & \frac{-2}{\omega} \end{bmatrix}$$

△

Observación 2.5.2.

- El ω óptimo es aquel que minimiza $\rho(Q_{SOR})$.

$$\omega_{opt} = \min_{\omega \in (1,2)} \rho(Q_{SOR})$$

- En general $\rho(Q_{SOR})$ es no lineal en ω.

Capítulo 3

Ecuaciones no lineales

3.1. Ecuaciones no lineales en $\mathbb{R}$

Consideramos $f : \mathbb{R} \to \mathbb{R}$ función no lineal, queremos encontrar una raíz de f. O sea, $x^* \in \mathbb{R}$ tal que $f(x^*) = 0$.

Si existe y conocemos la función inversa de f, la podemos aplicar a la ecuación no lineal y obtener la raíz:

$$f^{-1}(f(x^*)) = f^{-1}(0) \Rightarrow x^* = f^{-1}(0)$$

En general no hay manera de hallar la inversa, o es muy costoso. Por lo que buscamos una aproximación de la solución mediante algún método numérico.

3.1.1. Métodos de punto fijo

Tenemos $f(x^*) = 0$, $f : \mathbb{R} \to \mathbb{R}$.

Supongamos que existe y conozco $g : \mathbb{R} \to \mathbb{R}$ tal que:

$$f(x) = x - g(x) \ \forall x \in \mathbb{R}$$

entonces

$$f(x^*) = 0 \iff x^* - g(x^*) = 0 \iff x^* = g(x^*)$$

Genero el método iterativo que tiene a x^* como punto fijo:

$$\begin{cases} x_{k+1} = g(x_k) \\ x_0 \in \mathbb{R} \end{cases}$$

con x_0 cercano a la raíz x^*.

Ejemplo 3.1.1. Estudiaremos el comportamiento de los métodos iterativos para la función $f(x) = \log(x^2) - x/2$. La solución más próxima a cero es: $x^* = 1{,}4296...$.

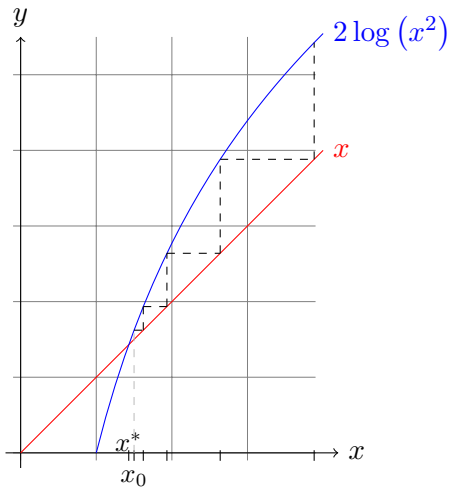

Figura 3.1: Primer método de punto fijo para la ecuación $\log(x^2) = x/2$

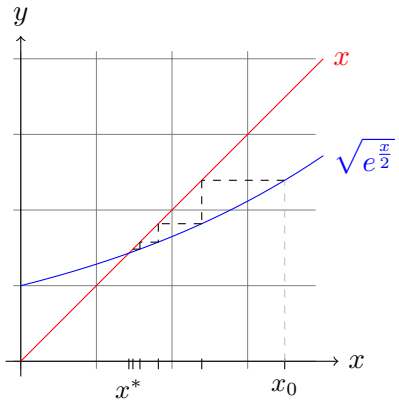

Figura 3.2: Segundo método de punto fijo para la ecuación $\log(x^2) = x/2$

1.
$$\log(x^2) - x/2 = 0 \iff 2\log(x^2) = x$$
tomamos $g_1(x) = 2\log(x^2)$ y obtenemos el método de punto fijo, que no converge, como se ve en la Figura 3.1:
$$\begin{cases} x_{k+1} &= 2\log(x_k^2) \\ x_0 &= 1{,}5 \end{cases}$$

2.
$$\log(x^2) - x/2 = 0 \iff 2\log(x^2) = x \iff x^2 = e^{\frac{x}{2}} \iff x = +\sqrt{e^{\frac{x}{2}}}$$
tomamos ahora la función $g_2(x) = \sqrt{e^{\frac{x}{2}}}$ y obtenemos el método:
$$\begin{cases} x_{k+1} &= \sqrt{e^{x_k/2}} \\ x_0 &= 1{,}5 \end{cases}$$

Como se puede ver en la Figura 3.2, $|g'(x^*)| < 1$, x^* es aproximadamente 1,4296, y se observa también que el método converge.

△

Profundizaremos sobre estos métodos en la Sección 3.2.

3.1.2. Método de Bipartición (o Bisección)

Antes de comentar el método recordemos el teorema de Bolzano:

Teorema 3.1.1 (Teorema de Bolzano). *Sea $f : [a, b] \to \mathbb{R}$ función continua tal que $f(a) \cdot f(b) < 0$ entonces existe $x^* \in [a, b]$ tal que $f(x^*) = 0$.*

El teorema nos da una idea para un algoritmo, supongamos que f es continua en un intervalo $I_0 = [a_0, b_0]$ tal que $f(a_0) \cdot f(b_0) < 0$, esto significa que en los extremos del intervalo la función tiene distinto signo, por lo que debe haber una raíz de f en I_0. Recursivamente generamos sub intervalos de I_0 tomando su punto medio en cada iteración que llamaremos m_k, tales que estén en la hipótesis del teorema de Bolzano y por lo tanto contienen siempre una raíz.

Algoritmo 3 Algoritmo de Bipartición

Sea f función continua en $I_0 = [a_0, b_0]$, tal que $f(a_0) \cdot f(b_0) < 0$ y N una cantidad máxima de iteraciones a realizar.

for $k = 0 \to N$ **do**
  $m_k \leftarrow (a_k + b_k)/2$
  if $f(m_k) = 0$ **then**
    return m_k
  else if $f(m_k) \cdot f(a_k) > 0$ **then**
    $I_{k+1} = [a_{k+1}, b_{k+1}] \leftarrow [m_k, b_k]$
  else $\{f(m_k) \cdot f(a_k) < 0\}$
    $I_{k+1} = [a_{k+1}, b_{k+1}] \leftarrow [a_k, m_k]$
  end if
end for

Observación 3.1.1. Para el algoritmo anterior:

$$|x^* - m_k| \leq |max\_error(k)| = \frac{b_k - a_k}{2} = \frac{b_{k-1} - a_{k-1}}{2^2} = \cdots = \frac{b_0 - a_0}{2^k} \quad \forall k > 0.$$

Por lo que $\lim_k |x^* - m_k| \leq 0$ y así:

$$\lim_k m_k = x^*$$

y para un k grande, m_k es una buena aproximación de x^*.

Observación 3.1.2. Si la función no es continua, el método puede converger a un punto de discontinuidad. (ej: piense en la función como en la Figura 3.3).

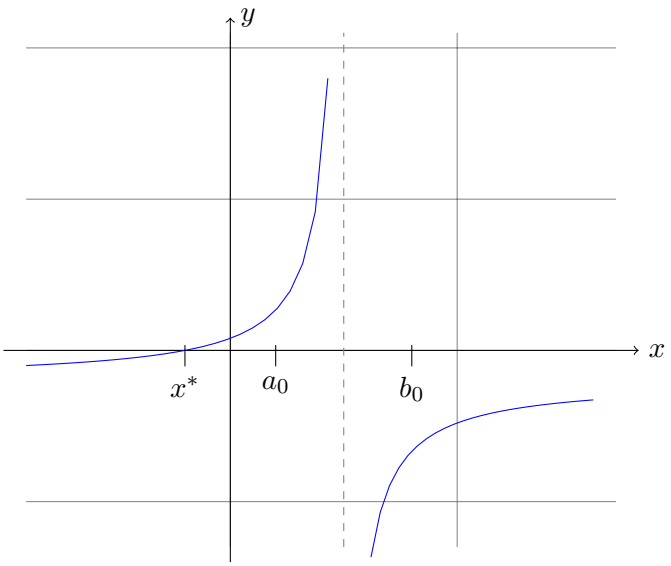

Figura 3.3: Ejemplo bipartición en función discontinua

Ejercicio 3.1.1.

- Sea $x^* \in [a, b]$ raiz de f, hallar la cantidad de iteraciones $\hat{k}$ necesarias para que el error usando el método de bipartición sea menor o igual que 10^{-n}.
- Si $[a, b] = [1, 2]$ y el error debe ser menor a 10^{-5}, determine $\hat{k}$.
- ¿Puede deducir cuántas iteraciones se requieren para ganar un dígito decimal de precisión?

Definición 3.1.1 (Orden y velocidad de convergencia)**.** Sea $\{x_k\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$ una sucesión convergente a $\alpha \in \mathbb{R}^n$. Sean $p > 0$ y $\beta > 0$ tales que:

$$\lim_{k\to\infty} \frac{\|x_{k+1} - \alpha\|}{\|x_k - \alpha\|^p} = \beta$$

Decimos que

- p es el *orden* de convergencia de la sucesión.
- β es la *velocidad* de convergencia.
- Si $p = 1$ decimos que la sucesión tiene convergencia lineal.
- Si $p > 1$ decimos que la sucesión tiene convergencia supralineal.
- Si $p = 2$ decimos que la sucesión tiene convergencia cuadrática, etc.

Ejemplo 3.1.2.

1. Sea la sucesión $a_k = \frac{1}{2^k}$. Tiene convergencia lineal a 0, con velocidad 1/2.

2. $b_0 = 1, b_1 = 1, b_2 = \frac{1}{4}, b_3 = \frac{1}{4}, b_4 = \frac{1}{16}, \ldots, b_k = \frac{1}{4^{\text{floor}(k/2)}}$. La sucesión b_k converge a 0, pero no entra dentro de la definición.

3. $c_k = \frac{1}{2^{2^k}}$, tiene convergencia cuadrática y velocidad 1.

△

Observación 3.1.3. Se demuestra que cuanto mayor es p mas rápido converge la sucesión y cuanto menor es β mas rápido converge la sucesión.

Observamos que para el método de bisección tenemos que en cada paso el máximo error se divide por dos y:

$$\lim_k \frac{|x^* - m_k|}{|x^* - m_{k-1}|} \leq \frac{1}{2}$$

por lo que el método tiene convergencia lineal y velocidad a lo sumo 1/2.

3.1.3. Método de Newton-Raphson

Supongamos ahora que la función con la que estamos trabajando es diferenciable. Veamos la deducción geométrica del método de Newton-Raphson.

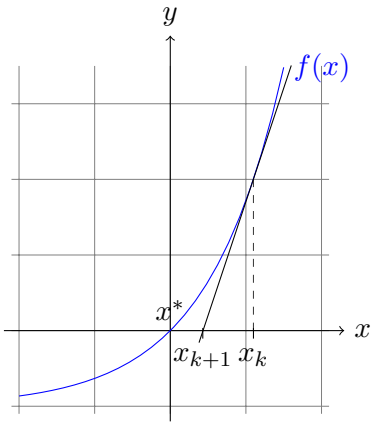

Figura 3.4: Método de Newton-Raphson

Dado $x_k \in \mathbb{R}$, hallamos la recta tangente a f en x_k y el punto de corte con el eje x, es x_{k+1}. La recta tangente a f en x_k es

$$r: \; y - f(x_k) = f'(x_k)(x - x_k)$$

si imponemos $y = 0$, se obtiene $x = x_k - \frac{f(x_k)}{f'(x_k)}$. Con lo que llegamos al método de Newton-Raphson:

$$\text{(N-R)}: \begin{cases} x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \\ x_0 \in \mathbb{R} \end{cases}$$

que se puede ver en la Figura 3.4.

Ejemplo 3.1.3. Sea $f(x) = x^2 - 2$, una de las raíces de esta función es $x^* = \sqrt{2}$. Si aplicamos Newton-Raphson a f, con punto inicial 1, obtenemos la siguiente iteración:

$$\begin{cases} x_{k+1} = x_k - \frac{x_k^2 - 2}{2x_k} \\ x_0 = 1 \end{cases}$$

En la siguiente tabla vemos los valores de la iteración para ciertos k y sus respectivos errores.

k	x_k	orden del error
0	1	-1
1	$3/2 = 1.5$	-2
2	$17/2 \approx 1{,}41117$	-3
3	$577/408 \approx 1{,}41421$	-6
4	$665857/470832 \approx 1{,}41421$	-12
5	-	-25
6	-	-49

△

Veamos una teorema que nos garantiza la convergencia del método de Newton-Raphson bajo ciertas hipótesis.

Teorema 3.1.2. *Sea $f : I \to \mathbb{R}$ función con derivada segunda continua, tal que $f(x^*) = 0$, $f'(x^*) \neq 0$ y $f''(x^*) \neq 0$. Entonces la sucesión generada por Newton-Raphson converge a x^* cuadraticamente y con velocidad $\left|\frac{f''(x^*)}{2f'(x^*)}\right|$, siempre que x_0 se elija suficientemente próximo a x^*.*

Demostración.

- Postergaremos la demostración de la convergencia para más adelante, una vez hayamos desarrollado más la teoría. En particular, obtendremos este resultado como aplicación del Corolario 3.2.4. Ver Proposición 3.2.5.
- Veamos el orden y velocidad de convergencia:

  Hallamos el desarrollo de Taylor de f en un entorno de x_k y evaluando en x^*, obtenemos

  $$0 = f(x^*) = f(x_k) + f'(x_k)(x^* - x_k) + \frac{f''(\xi_k)}{2}(x^* - x_k)^2, \ \xi_k \in (x^*, x_k)$$

  por lo que, dividiendo entre $f'(x_k)$

  $$x^* - x_{k+1} = x^* - \left(x_k - \frac{f(x_k)}{f'(x_k)}\right) = -\frac{f''(\xi_k)}{2f'(x_k)}(x^* - x_k)^2, \ \xi_k \in (x^*, x_k)$$

  y

  $$\frac{|x^* - x_{k+1}|}{|x^* - x_k|^2} = \left|\frac{f''(\xi_k)}{2f'(x_k)}\right|, \ \xi_k \in (x^*, x_k)$$

finalmente, tomando limite

$$\lim_k \frac{|x^* - x_{k+1}|}{|x^* - x_k|^2} = \lim_k \left| \frac{f''(\xi_k)}{2f'(x_k)} \right| = \left| \frac{f''(x^*)}{2f'(x^*)} \right|$$

ya que $\lim_k f''(\xi_k) = f''(x^*)$, $\lim_k f'(x_k) = f'(x^*)$ por continuidad de f'' y f', y $\lim_k \xi_k = x^*$.

□

Se deja como ejercicio probar:

- Si $f'(x^*) = 0$ el orden es 1.
- Si $f'(x^*) \neq 0$ y $f''(x^*) = 0$ el orden es mayor o igual a 3.

3.1.4. Método de la secante

El método de la secante es una modificación del método de Newton-Raphson, en vez de hallar la recta tangente a un punto, aproximamos la misma por una recta secante. Ver Figura 3.5.

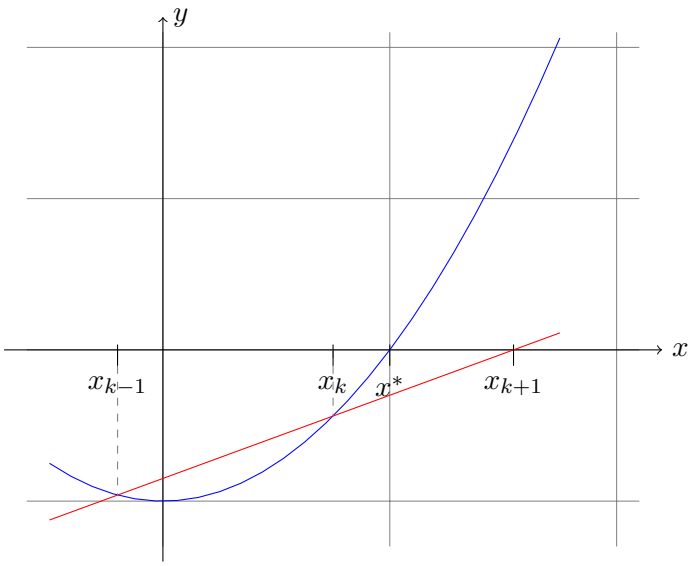

Figura 3.5: Método de la secante

Geométricamente, dados dos puntos de aproximación x_{k-1}, x_k, se halla la intersección de la recta que pasa por los puntos $(x_{k-1}, f(x_{k-1}))$ y $(x_k, f(x_k))$ con el eje x.

La ecuación de la recta es:

$$r : y - f(x_k) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}(x - x_k)$$

cuya intersección con el eje x es: $x = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k)$.

Obtenemos así el método de la secante:

$$\begin{cases} x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k) \\ x_0, x_1 \in \mathbb{R} \end{cases}$$

Observación 3.1.4.

- El método de la secante genera una sucesión de orden 2, es decir, que para el cálculo de x_{k+1} que requieren los valores de dos pasos anteriores (x_k y x_{k-1}).
- Reescribiendo el método:

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

  Si $x_k \approx x_{k-1} \Rightarrow$, es posible pensarlo como una variación del método de N-R, aproximando la derivada por medio de diferencias finitas, y esto es especialmente útil cuando el costo computacional de derivar la función es elevado. Por tanto, si bien el orden de convergencia es menor que el de Newton, al considerar el costo de evaluar $f'(x)$ puede resultar ventajoso usar la secante.
- No se asegura convergencia si los valores iniciales son demasiado lejanos o la raíz no es simple.

Vemos ahora un teorema sobre la convergencia del método de la secante, que no demostraremos.

Teorema 3.1.3. *Sea $f : I \to \mathbb{R}$ función con derivada segunda continua, tal que $f(x^*) = 0$, $f'(x^*) \neq 0$. Entonces la sucesión generada por el método de la secante converge a x^* siempre que x_0 se elija suficientemente próximo a x^* y en ese caso la convergencia es supralineal con $p = \frac{1+\sqrt{5}}{2} \approx 1,618$.*

Observación 3.1.5. Destacamos como curiosidad que $p = \frac{1+\sqrt{5}}{2}$ es la razón áurea.

3.1.5. Método de la regla falsa (o falsa posición)

El método de la Regla Falsa (o Falsa posición) requiere que la función f sea continua y requiere dos puntos iniciales a y b tal que sus valores funcionales son de distinto signo.

Es similar al de la secante o al de bipartición en el sentido de que el intervalo $[a_n, b_n]$ se va actualizando.

El procedimiento es el siguiente (ver Figura 3.6):

Se toma la cuerda dada por los puntos $(a, f(a))$ y $(b, f(b))$. Se intersecta con el eje x, es decir, que se hace $y = 0$ para obtener el punto x_1. Se genera un nuevo segmento $[a, b]$, sustituyendo o a o b por x_1 según sea el signo de $f(x_1)$ de forma que siempre $f(a)$ y $f(b)$ tengan signos opuestos.

Es decir:

Cuerda $(a, f(a)) - (b, f(b))$:

$$y = mx + d \quad \begin{cases} m = \frac{f(b) - f(a)}{b - a} \\ d = \frac{af(b) - bf(a)}{a - b} \end{cases}$$

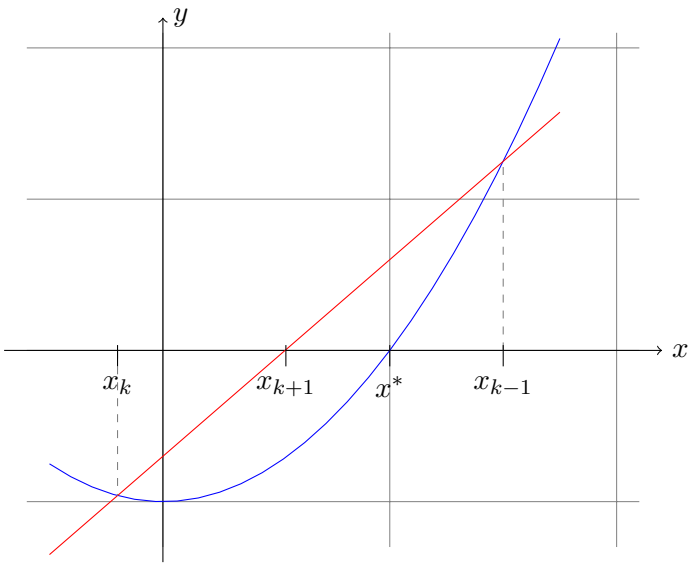

Figura 3.6: Método de la regla falsa

Haciendo $y = 0$:

$$x = \frac{-d}{m} = \frac{af(b) - bf(a)}{f(a) - f(b)} = x_1$$

Si $\begin{cases} f(a)f(x_1) < 0 \Rightarrow \text{ Repetimos en } [a, x_1] \\ f(x_1)f(b) < 0 \Rightarrow \text{ Repetimos en } [x_1, b] \end{cases}$

A diferencia del método de la secante, en que se debería utilizar la cuerda formada por x_k y x_{k+1} para construir el siguiente punto, en el método de la falsa regla debemos utilizar siempre puntos tal que sus valores funcionales sean de distinto signo. Por lo que, si tomamos como ejemplo la Figura 3.6, como x_k y x_{k+1} toman valores funcionales negativos, debemos buscar un punto anterior de la sucesión cuyo valor funcional sea positivo. En el ejemplo, utilizamos x_{k+1} con x_{k-1} y con ellos construimos la cuerda que generará el punto x_{k+2}

Observación 3.1.6.

- El método es siempre convergente (para funciones continuas).
- Sin embargo es de orden 1.
- Es aconsejable su uso mientras se está lejos de la raíz pero conviene cambiar a otro cerca de ella.
- Pese a ser similar al método de la secante, es más complejo, tiene menor orden, pero es globalmente convergente.

Ejercicio 3.1.2. Sea la función $f(x) = x^2 - 1$; $x_1 = 1,5$; $x_2 = 0,2$.

- Hallar x_3 y x_4 usando los 4 métodos vistos.
- Determinar si todos ellos convergen.

- Fundamentar cuál es el más conveniente.
- Varía de algún modo lo anterior si en lugar de $x_2 = 0,2$ se toma $x_2 = -0,2$.

Para concluir remarcamos que existe toda una galería de métodos aplicables a funciones. Aquí solamente se han presentado las más generales y/o sencillas.

3.2. Métodos Iterativos Generales

Sea $g : \mathbb{R} \to \mathbb{R}$ función, y el método:

$$(\mathrm{M}) \begin{cases} x_{k+1} = g(x_k) \\ x_0 \in \mathbb{R} \end{cases}$$

que genera una sucesión $\{x_k\}_{k\in\mathbb{Z}}$.

Ejemplo 3.2.1. En N-R, $g(x) = x - \frac{f(x)}{f'(x)}$. △

Definición 3.2.1. Decimos que $\alpha \in \mathbb{R}$ es punto fijo de (M) si $g(\alpha) = \alpha$.

Observación 3.2.1. Si $\alpha \in \mathbb{R}$ es punto fijo de $g(x)$, entonces es raiz de $f(x) = g(x) - x$.

Ejercicio 3.2.1. Hallar los puntos fijos de $g(x) = 2 - x^2$.

Definición 3.2.2. La función g es contractiva en $I \subset \mathbb{R}$ si existe $0 \leq m < 1$ tal que

$$|g(x) - g(y)| \leq m \cdot |x - y|$$

para todos $x, y \in I$.

Ejemplo 3.2.2. La función x^2 no es contractiva en $\mathbb{R}$ pues tomando $x = 3$, $y = 2$, no es posible hallar $m < 1$ que verifique la definición.

Sin embargo, sí lo es en el intervalo $(-\frac{1}{4}, \frac{1}{4})$:

$|x^2 - y^2| = |x + y||x - y|$ y como $|x + y| < \frac{1}{2}$, basta tomar $m = \frac{1}{2}$. △

Ejercicio 3.2.2. Dar otro ejemplo de una función contractiva en algún intervalo y probarlo.

Observación 3.2.2.

- Toda contracción es continua.
- Más aún, toda contracción es Lipschitz (con constante de Lipschitz m), y por tanto, uniformemente continua.

Teorema 3.2.1 (Punto Fijo). *Si X es un espacio métrico completo y $\varphi : X \to X$ una m-contracción con $m < 1$, entonces la sucesión $\{x_k\}_{k\in\mathbb{N}}$ tal que $x_{k+1} = \varphi(x_k)$ converge al punto fijo de φ, que además es único. El resultado no depende del elemento inicial $x_0 \in X$.*

Demostración. Vamos a probar el Teorema 3.2.1 en cuatro etapas:

i Toda m-contracción es continua: sea $f : X \to X$ una m-contracción y $\{x_k\}_{k\in\mathbb{N}}$ una sucesión en X tal que $x_k \to x$. Basta ver que $f(x_k) \to f(x)$. Efectivamente, como el espacio admite una métrica d tenemos que $d(f(x_k), f(x)) \leq md(x_k, x) \to 0$. En particular, tenemos por hipótesis que φ es continua.

ii Probemos ahora que la sucesión $\{x_k\}_{k\in\mathbb{N}}$ es de Cauchy. Sea $\varepsilon > 0$ arbitrario. Veamos que existe k_0 tal que si $l \geq k \geq k_0$ entonces $d(x_l, x_k) < \varepsilon$. Por definición de la sucesión y la desigualdad triangular de una métrica, tenemos que:

$$\begin{aligned} d(x_l, x_k) = d(\varphi^{(l)}(x_0), \varphi^{(k)}(x_0)) &\leq \sum_{i=0}^{l-k-1} d(\varphi^{(l-i)}(x_0), \varphi^{(l-i-1)}(x_0)) \\ &\leq \sum_{i=0}^{l-k-1} m^{l-i-1} d(x_0, x_1) = d(x_0, x_1) \sum_{j=k}^{l-1} m^j = d(x_0, x_1) \frac{m^l - m^k}{m-1} \\ &= d(x_0, x_1) \frac{m^k}{1-m}(1 - m^{l-k}) < \varepsilon, \end{aligned}$$

eligiendo $l \geq k$ suficientemente grande. Luego, $\{x_k\}_{k\in\mathbb{N}}$ es de Cauchy. Como X es completo, converge a cierto elemento α perteneciente al espacio X: $\lim_k x_k = \alpha \in X$.

iii Veamos ahora α es además punto fijo de φ, es decir que $\varphi(\alpha) = \alpha$. De hecho, por continuidad de φ tenemos que:

$$\varphi(\alpha) = \varphi(\lim_k x_k) = \lim_k \varphi(x_k) = \lim_k x_{k+1} = \alpha.$$

iv Finalmente, probemos la unicidad del punto fijo: si $\beta = \varphi(\beta)$, entonces:

$$d(\alpha, \beta) = d(\varphi(\alpha), \varphi(\beta)) \leq md(\alpha, \beta),$$

o equivalentemente:

$$d(\alpha, \beta)(1 - m) \leq 0.$$

Como $1 - m > 0$ y $d(\alpha, \beta) \geq 0$, la única posibilidad es que $d(\alpha, \beta) = 0$, y como d es una métrica entonces $\alpha = \beta$.

Hemos probado así que en un espacio métrico completo X, la reiterada aplicación de una m-contracción a un punto inicial cualquiera $x_0 \in X$ genera una sucesión que converge siempre al único punto fijo α de la contracción, siempre que $m < 1$. □

Teorema 3.2.2 (Convergencia)**.** *Supongamos que existe α punto fijo de (M) y $\delta > 0$ tal que g es contractiva en $B_{\alpha,\delta}$ con constante m. Luego, para todo $x_0 \in B_{\alpha,\delta}$ se cumple:*

1. $x_k \in B_{\alpha,\delta}$ $\forall k \in \mathbb{N}$.
2. $\lim_k x_k = \alpha$.
3. *α es el único punto fijo de (M) en $B_{\alpha,\delta}$.*

Demostración.

1. Hacemos inducción en k, sabemos que $x_0 \in B_{\alpha,\delta}$. Si $x_k \in B_{\alpha,\delta}$

$$|\alpha - x_{k+1}| = |g(\alpha) - g(x_k)| \leq m \cdot |\alpha - x_k| < 1 \cdot \delta = \delta$$

y $x_{k+1} \in B_{\alpha,\delta}$.

2. Sabemos que $|\alpha - x_k| \leq m \cdot |\alpha - x_{k-1}| \leq m^2 \cdot |\alpha - x_{k-2}| \leq \cdots \leq m^k \cdot |\alpha - x_0|$ por lo que $\lim_k |\alpha - x_k| \leq \lim_k m^k \cdot |\alpha - x_0| = 0$, ya que $m < 1$. Concluimos que $\lim_k x_k = \alpha$.

3. Supongo que existe $\beta \in B_{\alpha,\delta}$ otro punto fijo de (M), vemos que

$$|\alpha - \beta| = |g(\alpha) - g(\beta)| \leq m \cdot |\alpha - \beta| < |\alpha - \beta|$$

que es absurdo salvo que $\alpha = \beta$.

□

Lema 3.2.3. *Si g es derivable en un entorno de α y $|g'(x)| \leq m$, con $m < 1$, para todo $x \in B_{\alpha,\delta}$ entonces g es contractiva en $B_{\alpha,\delta}$.*

Demostración. Sean $x, y \in B_{\alpha,\delta}$, por el teorema del valor medio, existe $\xi \in (x, y)$ tal que $|g(x) - g(y)| = |g'(\xi)| \cdot |x - y|$. Y como $|g'(\xi)| \leq m$ queda demostrado el lema. □

Veamos una consecuencia directa del lema anterior.

Corolario 3.2.4. *Si α es punto fijo de (M), $g \in \mathcal{C}^1$ y $|g'(x)| \leq m < 1$ para todo $x \in B_{\alpha,\delta}$ con $\delta > 0$, entonces:*

1. *$x_k \in B_{\alpha,\delta}$ para todo $k \in \mathbb{N}$.*
2. *$\lim_k x_k = \alpha$.*
3. *α es el único punto fijo de (M) en $B_{\alpha,\delta}$.*

Ejemplo 3.2.3. Volvamos al ejemplo 3.1.1, tenemos la ecuación

$$\log\left(x^2\right) - \frac{x}{2} = 0$$

con solución aproximada $x^* \approx 1{,}4296$.

En el primer método iterativo usábamos la función $g_1(x) = 2\log\left(x^2\right)$. Vemos que $|g_1'(x^*)| = 4/x^* \approx 2{,}79796 > 1$, por lo cual el método no converge.

En el segundo método iterativo usamos la función $g_2(x) = \sqrt{e^{x/4}}$, con $|g_2'(x^*)| \approx 0{,}35740 < 1$ por lo que vemos que el método converge. △

Proposición 3.2.5. *La sucesión generada por Newton-Raphson en las hipótesis del Teorema 3.1.2 converge a x^*.*

Demostración. Si denotamos $g(x) = x - \frac{f(x)}{f'(x)}$ entonces el método de Newton-Raphson es un método de punto fijo con función g. Derivando g:

$$g'(x) = 1 - \frac{f'(x)f'(x) - f''(x)f(x)}{f'(x)^2}$$

y vemos que $g'(x^*) = 0$. Entonces, x^* es un atractor para g, por lo que existe $\varepsilon > 0$ tal que si $x_0 \in (x^* - \varepsilon, x^* + \varepsilon)$ la sucesión generada por el método converge. □

Teorema 3.2.6. *Si $g \in \mathcal{C}^p$, $p \in \mathbb{N}^+$. (M) es de orden p y velocidad $\frac{|g^{(p)}(\alpha)|}{p!}$ si y solo si $g^{(i)}(\alpha) = 0$ para $i = 1, \ldots, p-1$ y $g^{(p)}(\alpha) \neq 0$.*

Demostración. ($\Rightarrow$) Desarrollamos Taylor a g en α:

$$g(x) = g(\alpha) + \overbrace{\sum_{i=1}^{p-1} \frac{g^{(i)}(\alpha)}{i!}(x-\alpha)^i}^{=0} + \frac{g^{(p)}(\xi)}{p!}(x-\alpha)^p = \alpha + \frac{g^{(p)}(\xi)}{p!}(x-\alpha)^p$$

con $\xi \in (\alpha, x)$. Evaluando en x_k vemos que $x_{k+1} = g(x_k) = \alpha + \frac{g^{(p)}(\xi_k)}{p!}(\alpha - x_k)^p$, $\xi_k \in (\alpha, x_k)$. Luego

$$\frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} = \frac{|g^{(p)}(\xi_k)|}{p!} \longrightarrow_k \frac{|g^{(p)}(\alpha)|}{p!} \neq 0$$

ya que $\lim_k g^{(p)}(\xi_k) = g^{(p)}(\alpha)$ por continuidad.

($\Leftarrow$) Si el orden de convergencia del método es p, entonces si $0 \leq i < p$ vemos que

$$\lim_k \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^i} = \lim_k \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} |x_k - \alpha|^{p-i} = 0$$

ya que $(p - i) > 0$ y $\lim_k \frac{|x_{k+1}-\alpha|}{|x_k-\alpha|^p} < \infty$. Veamos ahora por inducción que $g^{(i)}(\alpha) = 0$. Para $i = 1$, desarrollamos Taylor a g en α y evaluamos en x_k y obtenemos $x_{k+1} - \alpha = g'(\xi_k)(x_k - \alpha)$, $\xi_k \in (\alpha, x_k)$. Luego $\frac{|x_{k+1}-\alpha|}{|x_k-\alpha|} = |g'(\xi_k)|$. Y tomando limite probamos que $g'(\alpha) = 0$. Usando inducción y Taylor se prueba el resto de las igualdades. □

A modo de resumen, para una función $f(x)$ es importante elegir adecuadamente la función $g(x)$ y el valor inicial x_0. Si se tienen dos métodos iterativos, serán convergentes si $|g'(x)| < 1$ en un entorno de la raíz x^*. El orden de convergencia viene dado por el estudio de la anulación de sus derivadas, elegimos el que tenga más derivadas nulas en x^*. Si ambos métodos tienen derivada primera no nula, la velocidad es lineal y convendrá usar la iteración en la cual el valor absoluto de $g(x^*)$ es menor.

Ejemplo 3.2.4. Sea $f(x) = \sin(x) - x^2$ que tiene raíz $x^* \approx 0,87$.

Tenemos las siguientes opciones:

- $g_1(x) = \sqrt{\sin(x)} \Rightarrow |g_1'(x)| = \left|\frac{\cos(x)}{2\sqrt{\sin(x)}}\right| < 0,45$ en $(0,8; 0,9)$. Sirve.

- $g_2(x) = \arcsin(x^2) \Rightarrow |g_2'(x)| = \left|\frac{2x}{\sqrt{1-(x^2)^2}}\right| > 2$ en $(0,8; 0,9)$. No sirve.
- $g_3(x) = \sin(x) - x^2 + x \Rightarrow |g_3'(x)| = |\cos(x) - 2x + 1| < 0,18$ en $(0,8; 0,9)$. Sirve.

Es así que tanto g_1 como g_3 son métodos convergentes con velocidad lineal. Ahora $g_1'(x^*) \approx 0,37$ y $g_3'(x^*) \approx 0,10$, por lo que convendrá usar g_3. △

3.2.1. Condiciones de parada

Entre las condiciones de parada de los métodos iterativos es posible considerar:

1. Parada por número de iteraciones ($k > k_{max}$). El algoritmo se detiene cuando el número de iteraciones llega a una cantidad fijada de antemano. Es adecuado cuando no se conoce el comportamiento del método para la función f considerada o también en conjunción con otras condiciones de parada que pudieran no satisfacerse a lo largo de la recursión. La condición evita que el algoritmo entre en un ciclo sin fin.

2. Parada por proximidad a la raíz x^*. Es adecuada cuando lo que se desea es hallar, con determinada precisión, la raíz x^*. Se busca detener el algoritmo cuando el error $|x^{(k)} - x^*|$ o bien el error relativo $\frac{|x^{(k)}-x^*|}{|x^*|}$, es menor que cierta tolerancia $\varepsilon > 0$ fijada de antemano. Como no se conoce a priori la raíz x^*, se hace la estimación $x^{(k+1)} \approx x^*$ con lo que el método se detiene si:

$$|x^{(k)} - x^{(k+1)}| < \varepsilon, \quad \text{o bien si} \quad \frac{|x^{(k)} - x^{(k+1)}|}{|x^{(k+1)}|} < \varepsilon.$$

3. Parada por proximidad a la anulación de f. Se utiliza cuando lo que se pretende es hallar valores de las variables que hagan que $f(x)$ sea pequeño. La condición de parada es $|f(x^{(k)})| < \varepsilon$, donde $\varepsilon > 0$ es una tolerancia fijada de antemano.

3.3. Sistemas de Ecuaciones no lineales

Consideramos ahora un sistema de n ecuaciones no lineales en n variables, $f_i : \mathbb{R}^n \to \mathbb{R}$,

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \qquad\qquad \vdots \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

que vectorialmente lo podemos expresar como $F(\mathbf{x}) = 0$ con $F = (f_1, f_2, \ldots, f_n)$, $F : \mathbb{R}^n \to \mathbb{R}^n$ y $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$.

3.3.1. Métodos Iterativos en las variables

Podemos definir métodos iterativos para resolver sistemas de ecuaciones con las ideas de los métodos de Jacobi, Gauss-Seidel y SOR, dado un método para resolver sistemas no lineales en $\mathbb{R}$. En el Algoritmo 4 vemos el método de Jacobi no lineal.

Algoritmo 4 Método de Jacobi no lineal

Sea (M) un método para resolver ecuaciones del tipo $f(x) = 0$ con $f : \mathbb{R} \to \mathbb{R}$, y $\mathbf{x}^{(0)} \in \mathbb{R}^n$.

$k \leftarrow 0$
repeat
  for $j = 1 \to n$ **do**
    $x_j^{(k+1)} \leftarrow$ solución mediante el método (M) de $f_j\left(x_1^{(k)}, x_2^{(k)}, \ldots, x_{j-1}^{(k)}, x_j^{(k+1)}, x_{j+1}^{(k)}, \ldots, x_n^{(k)}\right) = 0$
  end for
  $k \leftarrow k + 1$
until convergencia

Se puede aplicar la misma idea para los métodos de Gauss-Seidel y SOR.

3.3.2. Métodos Iterativos Generales

Generalizamos aquí la idea de los MIG para funciones reales a funciones en $\mathbb{R}^n$. Es posible probar los mismos resultados para $\mathbf{x}^{(k+1)} = g(\mathbf{x}^{(k)})$, con $g : \mathbb{R}^n \to \mathbb{R}^n$, convirtiendo valores absolutos en normas, derivadas en jacobiana. En particular, la convergencia requerirá las mismas hipótesis de continuidad y de dominio para g y que $\|J_g(\mathbf{x}^*)\| < 1$.

Ejemplo 3.3.1.

$$f(x_1, x_2) = \begin{pmatrix} e^{x_1+x_2^2} + x_1^2 \\ x_1^2 e^{-x_2} - x_2 x_1^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Entonces podemos generar un problema de punto fijo mediante:

$$g(x_1, x_2) = \begin{pmatrix} e^{x_1+x_2^2} + x_1^2 + x_1 \\ x_1^2 e^{-x_2} - x_2 x_1^2 + x_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

△

3.3.3. Newton-Raphson

Sea $F : \mathbb{R}^n \to \mathbb{R}^n$, $F(\mathbf{x}^*) = 0$, $\mathbf{x}^* \in \mathbb{R}^n$.

Se puede generalizar el método de Newton-Raphson a sistemas de ecuaciones no lineales. La idea es resolver la ecuación linealizada en un punto próximo a la solución.

Si suponemos F diferenciable en el punto $\mathbf{x}^{(k)}$, por el teorema de Taylor:

$$F(\mathbf{x}) = F(\mathbf{x}^{(k)}) + J_F(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) + O\left(\|\mathbf{x} - \mathbf{x}^{(k)}\|^2\right)$$

trunco el termino $O\left(\|\mathbf{x} - \mathbf{x}^{(k)}\|^2\right)$ y resuelvo suponiendo $F(x) \approx 0$, obteniendo el método:

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - J_F(\mathbf{x}^{(k)})^{-1} F(\mathbf{x}^{(k)}) \\ x^{(0)} \in \mathbb{R}^n \end{cases}$$

Notamos que el determinante de $J_F(\mathbf{x}^{(k)})$ debe de ser no nulo para todo k para poder definir el método. Vemos también que el caso en que $n = 1$ del método coincide con el método presentado en la Subsección 3.1.3.

En el Algoritmo 5 se puede ver el método de Newton-Raphson general (en el que no se calcula la inversa de J_F).

Algoritmo 5 Método de Newton-Raphson

Sea $\mathbf{x}^{(0)} \in \mathbb{R}^n$.
  $k \leftarrow 0$
  repeat
    $\mathbf{s}^{(k)} \leftarrow$ solución del sistema lineal $J_F(\mathbf{x}^{(k)})\mathbf{s}^{(k)} = -F(\mathbf{x}^{(k)})$
    $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$
    $k \leftarrow k + 1$
  until convergencia

Ejemplo 3.3.2. Sea $F(x, y, z) = (xy - z^2, y^2 - xz, 2x^2 - yxz - 1)$ con $\mathbf{x}^{(0)} = (1, 1, 0)$. Calculemos $\mathbf{x}^{(1)}$ aplicando Newton-Raphson.

$J_F(x, y, z) = \begin{pmatrix} y & x & -2z \\ -z & 2y & -x \\ 4x - yz & -xz & -yx \end{pmatrix}$. Por otro lado $F(1, 1, 0) = (1, 1, 1)^t$. Para calcular $\mathbf{x}^1$ tenemos el sistema:

$$J_F(1, 1, 0)\mathbf{s}^{(0)} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & -1 \\ 4 & 0 & -1 \end{pmatrix} \begin{pmatrix} s_1^{(0)} \\ s_2^{(0)} \\ s_3^{(0)} \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} = -F(1, 1, 0).$$

Resolviéndolo tenemos $\mathbf{s}^{(0)} = (s_1^{(0)}, s_2^{(0)}, s_3^{(0)})^t = (-1/3, -2/3, -1/3)^t$, y

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \mathbf{s}^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -\frac{1}{3} \\ -\frac{2}{3} \\ -\frac{1}{3} \end{pmatrix} = \begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \\ -\frac{1}{3} \end{pmatrix}.$$

Se deja como ejercicio computar de forma análoga $\mathbf{x}^{(2)}$ planteando y resolviendo el sistema: $J_F(\mathbf{x}^{(1)})\mathbf{s}^{(1)} = -F(\mathbf{x}^{(1)})$ y luego realizar $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \mathbf{s}^{(1)}$. △

Observación 3.3.1.

1. El numero de evaluaciones de funciones reales en cada caso es: $O(n^2)$ para J_F, y $O(n)$ para F.

2. Por cada paso necesito resolver un sistema lineal para el cual tiene un costo $O(2/3n^3)$ de flops.

3. Hay que calcular n^2 derivadas analíticamente por paso.

Es posible demostrar que el método es de orden 2.

Veremos a continuación algunas variantes.

3.3.4. Método de Newton amortiguado

Se varía la iteración y se convierte en:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha J_F(\mathbf{x}^{(k)})^{-1} F(\mathbf{x}^{(k)})$$

3.3.5. Método de Newton modificado

El método de Newton-Raphson es computacionalmente costoso y $J_F(\mathbf{x}^{(k)})$ puede esta mal condicionada, lo que hace difícil tener una buena aproximación de $\mathbf{x}^*$. Por estas razones se pueden definir ciertas modificaciones para evitar dichos problemas. El método de Newton modificado evita calcular la matriz Jacobiana y los $O(2/3n^3)$ flops por resolución del sistema, dejando constante $J_F(\mathbf{x}^{(p)})$. Es decir que J_F solo se recalcula para algunos pasos. O sea dejamos la matriz Jacobiana, más precisamente su descomposición, constante para una cantidad de pasos $p \geq 2$, luego se computa $J_F(\mathbf{x}^{(2p)})$ y se prosigue con el método sin variar la matriz por otras p iteraciones.

Algoritmo 6 Método de Newton modificado

Sea $\mathbf{x}^{(0)} \in \mathbb{R}^n$.

```
k ← 0
repeat
  LU ← J_F(x^(kp))
  for j = 0 → p − 1 do
    s^(kp+j) ← solución del sistema lineal, usando descomposición LU: LU · s^(kp+j) = −F(x^(kp+j))
    x^(kp+j+1) ← s^(kp+j) + x^(kp+j)
  end for
  k ← k + 1
until convergencia
```

Observación 3.3.2.

1. Aunque el método de Newton-Raphson converja, el método de Newton modificado puede no converger.

2. El método de Newton modificado es más lento que el de Newton-Raphson. Esto disminuye el orden de convergencia si J_F varía mucho (caso en que F es fuertemente no lineal).

3.3.6. Método de Steffensen

El método de Steffensen evita calcular J_F y la obtiene como cociente incremental:

$$\frac{\partial f_i(\mathbf{x}^{(k)})}{\partial x_j} \approx \frac{f_i(\mathbf{x}^{(k)} + h_j \mathbf{e}_j) - f_i(\mathbf{x}^{(k)})}{h_j}$$

con $\mathbf{e}_j = [0, 0, \ldots, 1, \ldots, 0]^t$, el j-ésimo vector de la base canónica.

Stefffensen toma $h_j = f_j(\mathbf{x}^{(k)})$. Así:

$$\frac{\partial f_i(\mathbf{x}^{(k)})}{\partial x_j} \approx \frac{f_i(\mathbf{x}^{(k)} + f_j(\mathbf{x}^{(k)}) \mathbf{e}_j) - f_i(\mathbf{x}^{(k)})}{f_j(\mathbf{x}^{(k)})}$$

De esta forma no es necesario conocer en forma explícita las derivadas parciales y se logra una velocidad supralineal.

3.3.7. Método de Broyden*

El método de Broyden mejora las desventajas computacionales de Newton-Raphson, que vimos en la Observación 3.3.1, pero converge más lento. La idea general del método es usar el paso de Newton-Raphson pero no usar $J_F(\mathbf{x}^{(k)})$ sino una aproximación.

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - B_k^{-1} F(\mathbf{x}^{(k)}) \\ \mathbf{x}^{(0)} \in \mathbb{R}^n \end{cases}$$

donde B_k es la aproximación de Broyden de $J_F(\mathbf{x}^{(k)})$.

Desarrollamos F por Taylor en $\mathbf{x}^{(k)}$ y evaluando en $\mathbf{x}^{(k-1)}$ obtenemos la siguiente aproximación:

$$F(\mathbf{x}^{(k-1)}) - F(\mathbf{x}^{(k)}) \approx J_F(\mathbf{x}^{(k)})(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$$

Buscamos entonces B_k tal que $\mathbf{y}^{(k-1)} = B_k s_{k-1}$, donde $\mathbf{y}^{(k-1)} = F(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k-1)})$ y $\mathbf{s}^{(k-1)} = \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$, esta condición para B_k es llamada *condición Cuasi-Newton*. En el paso k conocemos $\mathbf{y}^{(k-1)}$ y $\mathbf{s}^{(k-1)}$. B_k tiene n^2 incógnitas y n ecuaciones, por lo que es un sistema indeterminado.

Broyden propone que B_k y B_{k-1} no difieran en direcciones ortogonales al paso $k-1$, o sea

$$(\mathbf{s}^{(k-1)})^T \cdot p = 0 \text{ implica } B_k \cdot p = B_{k-1} \cdot p \tag{3.1}$$

que es llamada la *condición de Broyden*.

La condición (3.1) es equivalente a que B_k minimice $\|\tilde{B} - B_{k-1}\|_F$ para $\tilde{B}$ solución del sistema $\tilde{B}\mathbf{s}^{(k-1)} = \mathbf{y}^{(k-1)}$. Donde $\|\cdot\|_F$ es la norma de Frobenius y esta definida por

$$\|A\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|}, \; A = \{a_{ij}\}_{i=1,j=1}^{n,n}$$

Proposición 3.3.1. *La matriz B_k dada por*

$$B_k = B_{k-1} + \frac{(\mathbf{y}^{(k-1)} - B_{k-1}\mathbf{s}^{(k-1)})(\mathbf{s}^{(k-1)})^T}{(\mathbf{s}^{(k-1)})^T\mathbf{s}^{(k-1)}}$$

cumple con la condición Cuasi-Newton y la condición de Broyden en las hipótesis hasta ahora mencionadas.

Demostración. Veamos la condición Cuasi-Newton,

$$\begin{aligned} B_k\mathbf{s}^{(k-1)} &= B_{k-1}\mathbf{s}^{(k-1)} + \left(\frac{(\mathbf{y}^{(k-1)} - B_{k-1}\mathbf{s}^{(k-1)})(\mathbf{s}^{(k-1)})^T}{(\mathbf{s}^{(k-1)})^T\mathbf{s}^{(k-1)}}\right)\mathbf{s}^{(k-1)} \\ &= B_{k-1}\mathbf{s}^{(k-1)} + \mathbf{y}^{(k-1)} - B_{k-1}\mathbf{s}^{(k-1)} \\ &= \mathbf{y}^{(k-1)} \end{aligned}$$

La condición de Broyden,

$$\begin{aligned} (\mathbf{s}^{(k-1)})^T \cdot p = 0 \Longrightarrow B_k \cdot p &= B_{k-1} \cdot p + \frac{(\mathbf{y}^{(k-1)} - B_{k-1}\mathbf{s}^{(k-1)})(\mathbf{s}^{(k-1)})^T \cdot p}{(\mathbf{s}^{(k-1)})^T\mathbf{s}^{(k-1)}} \\ &= B_{k-1} \cdot p \end{aligned}$$

□

Algoritmo 7 Método de Broyden

Dados $x_0 \in \mathbb{R}^n$ y B_0 matriz no singular $n \times n$.

for $k \geq 0$ **do**
- $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} - B_k^{-1} \cdot F(\mathbf{x}^{(k)})$
- $\mathbf{y}^{(k)} \leftarrow F(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^{(k)})$
- $\mathbf{s}^{(k)} \leftarrow \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$
- $B_{k+1} \leftarrow B_k + \frac{(\mathbf{y}^{(k)} - B_k\mathbf{s}^{(k)})(\mathbf{s}^{(k)})^T}{(\mathbf{s}^{(k)})^T\mathbf{s}^{(k)}}$

end for

Observación 3.3.3. En el Algoritmo 7:

1. B_0 se elije de forma que sea fácil de invertir, por ejemplo una matriz diagonal.
2. Sigue haciendo $O(n^3)$ flops por paso como Newton-Raphson, al invertir la matriz B_k.
3. Solo hace n llamadas a funciones contra las n^2 de Newton-Raphson.

Veamos ahora una manera de solucionar el problema de computar B_k^{-1} usando la formula de actualización de la inversa de Sherman Morrison,

Proposición 3.3.2. *Sea $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, $u, v \in \mathbb{R}^n$, definimos $M = A + uv^T$, tal que* $\det(M) \neq 0$. *Luego*

$$M^{-1} = (A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{\alpha}$$

donde $\alpha = 1 + v^TA^{-1}u \neq 0$.

Demostración.

$$\begin{aligned}(A+uv^T)\left(A^{-1}-\frac{A^{-1}uv^TA^{-1}}{\alpha}\right) &= \\ \mathrm{I}-\frac{uv^TA^{-1}}{\alpha}+uv^TA^{-1}-\frac{u\overbrace{\left(v^TA^{-1}u\right)}^{\in\mathbb{R}}v^TA^{-1}}{\alpha} &= \\ \mathrm{I}+uv^TA^{-1}\left(-\frac{1}{\alpha}+1-\frac{v^TA^{-1}u}{\alpha}\right) &= \mathrm{I}\end{aligned}$$

□

Aplicando la proposición 3.3.2 a $B_{k+1}=B_k+uv^T$, con $u=\frac{(\mathbf{y}^{(k)}-B_k\mathbf{s}^{(k)})}{(\mathbf{s}^{(k)})^T\mathbf{s}^{(k)}}$ y $v=\mathbf{s}^{(k)}$, podemos definir el Algoritmo 8.

Algoritmo 8 Algoritmo de Broyden usando Sherman Morrison

Dados $\mathbf{x}^{(0)}\in\mathbb{R}^n$ y $B_0\in M^{n\times n}(\mathbb{R})$.

for $k\geq 0$ **do**
- $\mathbf{x}^{(k+1)}\leftarrow\mathbf{x}^{(k)}-B_k^{-1}F(\mathbf{x}^{(k)})$
- $\mathbf{y}^{(k)}\leftarrow F(\mathbf{x}^{(k+1)})-F(\mathbf{x}^{(k)})$
- $\mathbf{s}^{(k)}=\mathbf{x}^{(k+1)}-\mathbf{x}^{(k)}$
- $B_{k+1}^{-1}=B_k^{-1}+\frac{\left(\mathbf{s}^{(k)}-B_k^{-1}\mathbf{y}^{(k)}\right)(\mathbf{s}^{(k)})^TB_k^{-1}}{(\mathbf{s}^{(k)})^TB_k^{-1}\mathbf{y}^{(k)}}$

end for

Observación 3.3.4. En el Algoritmo 8:

1. Hacemos n evaluaciones de funciones por paso como en Newton-Rapshon y Broyden.
2. Tenemos $O(n^2)$ de flops por paso al hallar B_k^{-1}, que es mejor que Newton-Raphson y Broyden.
3. El orden de convergencia es supralineal.

Capítulo 4

Mínimos Cuadrados

4.1. Problema de ajuste general

En general es el problema geométrico de ajustar una figura a una serie de puntos según algún criterio de mínima distancia. Por ejemplo ajustar elipses a puntos del plano, ver Figura 4.1.

Los datos surgen de una figura desconocida a la cual se le agrega ruido, buscamos recuperar a partir de estos datos con ruido la figura verdadera.

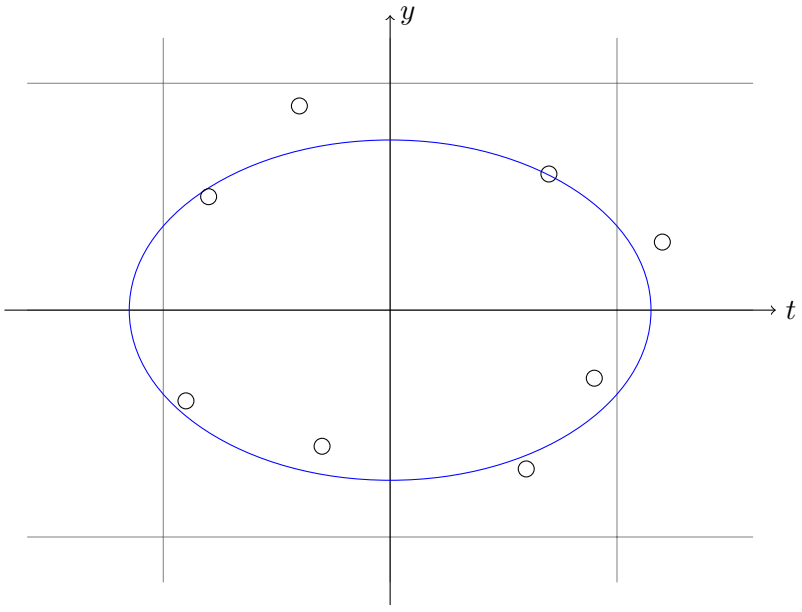

Figura 4.1: Elipse ajustada a puntos

4.2. Mínimos Cuadrados Lineales

Supongamos que tenemos ciertos datos dados por la tabla

t	y
t_1	y_1
$\vdots$	$\vdots$
t_m	y_m

y una función de ajuste a esos datos con parámetros $x_1, \cdots, x_n$, dada por $\Phi(x_1, x_2, \cdots, x_n, t) = \sum_{i=1}^{n} x_i \varphi_i(t)$. Donde los $\varphi_i(t) : \mathbb{R} \to \mathbb{R}$ pertenecen a una familia de funciones base conocidas linealmente independientes.

Observamos que $\Phi(X, t)$ es lineal con respecto a los x_i y en general $m > n$, o sea que esperamos mas datos que parámetros de ajuste. Buscamos entonces de todas las $\Phi(X, t)$ las que mejor se ajustan a los datos según un criterio de distancia que veremos a continuación.

Definición 4.2.1. En las hipótesis hasta ahora expuestas definimos el residuo como la siguiente función,

$$R(x_1, x_2, \cdots, x_n) = \begin{bmatrix} \Phi(x_1, x_2, \cdots, x_n, t_1) - y_1 \\ \Phi(x_1, x_2, \cdots, x_n, t_2) - y_2 \\ \vdots \\ \Phi(x_1, x_2, \cdots, x_n, t_m) - y_m \end{bmatrix} \in \mathbb{R}^n$$

que depende de los parámetros $x_1, x_2, \cdots, x_n$.

El problema de mínimos cuadrados lineal es encontrar $(\hat{x_1}, \hat{x_2}, \cdots, \hat{x_n})$ que minimice

$$\|R(x_1, x_2, \cdots, x_n)\|_2^2 = \sum_{j=1}^{m} \left(\Phi(x_1, x_2, \cdots, x_n, t_j) - y_j\right)^2$$

En notación matricial el problema queda planteado de la siguiente manera:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n, \; Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{R}^m$$

$$A = \begin{bmatrix} \varphi_1(t_1) & \varphi_2(t_1) & \cdots & \varphi_n(t_1) \\ \varphi_1(t_2) & \varphi_2(t_2) & \cdots & \varphi_n(t_2) \\ \vdots & \vdots & & \vdots \\ \varphi_1(t_m) & \varphi_2(t_m) & \cdots & \varphi_n(t_m) \end{bmatrix} \in M^{m \times n}$$

y definimos el resto como $R(X) = AX - Y$ y queremos minimizar $\|AX - Y\|_2^2$.

Teorema 4.2.1 (Ecuaciones Normales)**.** *Sean $A \in \mathcal{M}_{m \times n}(\mathbb{R})$, $Y \in \mathbb{R}^m$, $X \in \mathbb{R}^n$, entonces $\hat{X}$ minimiza $\|AX - Y\|_2^2$ si y solo si $A\hat{X} - Y$ es ortogonal a $Im(A)$ o $A^t\left(A\hat{X} - Y\right) = 0$.*

Observación 4.2.1.

- El sistema de ecuaciones $A^t AX = A^t Y$ se llaman ecuaciones normales.

- Si las columnas de A son l.i. entonces $|AA^t| \neq 0$ y existe una única solución a las ecuaciones normales.
- Si las columnas de A son l.d. entonces $|AA^t| = 0$ y existen infinitas soluciones a las ecuaciones normales.

Veremos tres demostraciones del teorema 4.2.1, en las que usaremos distintas herramientas de geometría, álgebra lineal y cálculo, y que permitirán además dar distinta significación al teorema.

Demostración 1 del teorema 4.2.1: Sea $\hat{X} \in \mathbb{R}^n$ tal que $A^t\left(y - A\hat{X}\right) = 0$. Para todo $w \in \mathbb{R}^n$ vemos que $Y - Aw = \left(Y - A\hat{X}\right) + \left(A\left(\hat{X} - w\right)\right)$, entonces para todo $w \in \mathbb{R}^n$

$$\begin{aligned}
\|Y - Aw\|_2^2 &= \left\|\left(Y - A\hat{X}\right) + \left(A\left(\hat{X} - w\right)\right)\right\|_2^2 \\
&= \left\|Y - A\hat{X}\right\|_2^2 + \left\|A\left(\hat{X} - w\right)\right\|_2^2 + 2\left(A\left(\hat{X} - w\right)\right)^t\left(Y - A\hat{X}\right) \\
&= \left\|Y - A\hat{X}\right\|_2^2 + \underbrace{\left\|A\left(\hat{X} - w\right)\right\|_2^2}_{\geq 0} + 2\left(\hat{X} - w\right)^t \underbrace{A^t\left(Y - A\hat{X}\right)}_{=0} \\
&\geq \left\|Y - A\hat{X}\right\|_2^2
\end{aligned}$$

y entonces $\hat{X}$ minimiza $\|Y - AX\|_2^2$.

Supongamos por absurdo que $\hat{X} \in \mathbb{R}^n$ minimiza $\|Y - AX\|_2^2$ y $A^t\left(Y - A\hat{X}\right) = Z \neq 0$. Dado $\varepsilon > 0$ definimos $w = \hat{X} + \varepsilon Z$, por lo visto anteriormente en la demostración sabemos que

$$\begin{aligned}
\|Y - Aw\|_2^2 &= \left\|Y - A\hat{X}\right\|_2^2 + \varepsilon^2\|AZ\|_2^2 - 2\varepsilon Z^t \underbrace{A^t\left(Y - A\hat{X}\right)}_{=Z} \\
&= \left\|Y - A\hat{X}\right\|_2^2 + \varepsilon^2\|AZ\|_2^2 - 2\varepsilon\|Z\|_2^2
\end{aligned}$$

y para llegar a una contradicción con respecto a la minimalidad, buscamos un ε tal que $\|Y - Aw\|_2^2 < \left\|Y - A\hat{X}\right\|_2^2$. Tenemos dos casos, $\|AZ\|_2^2 = 0$ o $\|AZ\|_2^2 \neq 0$.

Si $\|AZ\|_2^2 = 0$, tomo cualquier $\varepsilon > 0$ y funciona. Si $\|AZ\|_2^2 \neq 0$, tomo $\varepsilon = \frac{\|Z\|_2^2}{\|AZ\|_2^2}$ y

$$\begin{aligned}
\|Y - Aw\|_2^2 &= \left\|Y - A\hat{X}\right\|_2^2 + \frac{\|Z\|_2^4}{\|AZ\|_2^2} - 2\frac{\|Z\|_2^4}{\|AZ\|_2^2} \\
&= \left\|Y - A\hat{X}\right\|_2^2 - \frac{\|Z\|_2^4}{\|AZ\|_2^2} \\
&< \left\|Y - A\hat{X}\right\|_2^2
\end{aligned}$$

□

Seremos menos rigurosos en las siguientes dos demostraciones ya que la intención primordial es presentar las otras alternativas.

Demostración 2 del teorema 4.2.1: Consideremos la siguiente función que queremos minimizar:

$$\begin{aligned} s(X) = \|AX - Y\|_2^2 &= (AX - Y)^t (AX - Y) \\ &= X^t A^t AX - X^t A^t Y - Y^t AX + Y^t Y \\ &= X^t A^t AX - 2X^t A^t Y + Y^t Y \end{aligned}$$

donde hemos aplicado en la última igualdad que tanto $X^t A^t Y$ como $Y^t AX$ son escalares y uno es traspuesto del otro (es decir, $X^t A^t Y = (Y^t AX)^t$), por tanto son iguales.

Luego, como intentamos minimizar $s(X)$ imponemos $\nabla s(X) = \vec{0}$,

$\nabla s(X) = 2A^t AX - 2A^t Y = \vec{0}$

de donde se deducen las ecuaciones normales.

Puede el lector completar la demostración de ser mínimo, ya que $\nabla s(X) = \vec{0}$ solo implica punto crítico, observando los autovalores de la matriz Hessiana (si A es invertible, $A^t A$ es definida positiva).

□

Demostración 3 del teorema 4.2.1: Sea $AX = Y$, sabemos que este sistema es compatible si y solo si Y pertenece al espacio de columnas de A, o sea, si $Y \in Col(A)$. En general, en los PMC, justamente, esto no sucede ya que en general son muchas más ecuaciones que incóginitas, es decir, muchos más puntos que parámetros de ajuste, lo que hace que no exista un juego de parámetros que haga que el modelo pase por todos los puntos, y por tanto se debe buscar un set de parámetros que mejor se ajuste a los datos. Entonces Y queda fuera de este subespacio formado por las columnas de A (llamémosle $S = Col(A)$), $Y \notin Col(A)$, y el vector más próximo a Y que pertenece a $Col(A)$ es la proyección de Y sobre S: $P_S(Y)$.

Este vector $P_S(Y)$ pertenece a $Col(A)$, por tanto, existe $\hat{X}$ tal que $P_S(Y) = A\hat{X}$. Además sabemos que la proyección de un vector cumple que $Y - P_S(Y) \perp S$, es decir que $Y - A\hat{X}$ es ortogonal a $Col(A)$ y por tanto es ortogonal a todos los vectores en $Col(A)$, en particular, es ortogonal a las columnas A_i de A:

$Y - A\hat{X} \perp A_i \iff < Y - A\hat{X}, A_i >= 0 \iff A_i^t(Y - A\hat{X}) = 0$

Como esto debe cumplirse para todos los $i = 1, \ldots, n$, podemos ver que $Y - A\hat{X} \in \ker(A^t)$, y así $A^t(Y - A\hat{X}) = 0$.

Es interesante observar que es posible construir las implicancias que hemos mencionado en sentido inverso para realizar desarrollar el recíproco de la demostración, aunque se entiende que la parte más ilustrativa es la que hemos tratado aquí.

□

Veamos un ejemplo de mínimos cuadrados lineal.

Ejemplo 4.2.1. Supongamos que queremos aproximar los siguientes datos por una parábola,

t	y
1	3.2
2	10.2
3	21.4
4	36.3
5	55.1
6	78.3

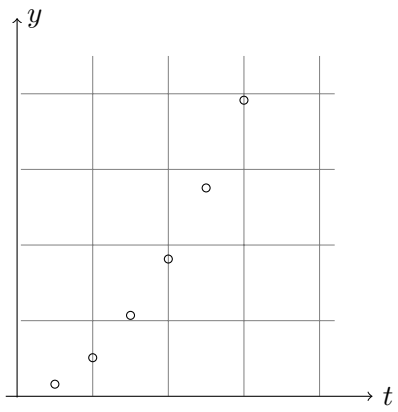

Figura 4.2: Ejemplo PMCL: Datos

En este caso, tenemos que $\Phi(x_1, x_2, x_3, t) = x_1t^2 + x_2t + x_3$, $\varphi_1(t) = t^2$, $\varphi_2(t) = t$, $\varphi_3(t) = 1$, y

$$Y = \begin{bmatrix} 3{,}2 \\ 10{,}2 \\ 21{,}4 \\ 36{,}3 \\ 55{,}1 \\ 78{,}3 \end{bmatrix}, \; A = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \\ 16 & 4 & 1 \\ 25 & 5 & 1 \\ 36 & 6 & 1 \end{bmatrix}, \; A^tA = \begin{bmatrix} 2275 & 441 & 91 \\ 441 & 91 & 21 \\ 91 & 21 & 6 \end{bmatrix}$$

$$A^tY = \begin{bmatrix} 5013{,}7 \\ 978{,}3 \\ 204{,}5 \end{bmatrix}$$

Resolviendo las ecuaciones normales vemos que $\hat{X} = \begin{bmatrix} 1{,}9893 \\ 1{,}0778 \\ 0{,}1400 \end{bmatrix}$

△

En general las ecuaciones normales pueden ser un problema mal condicionado para hallar la solución, por lo que buscamos métodos alternativos para solucionar las ecuaciones normales.

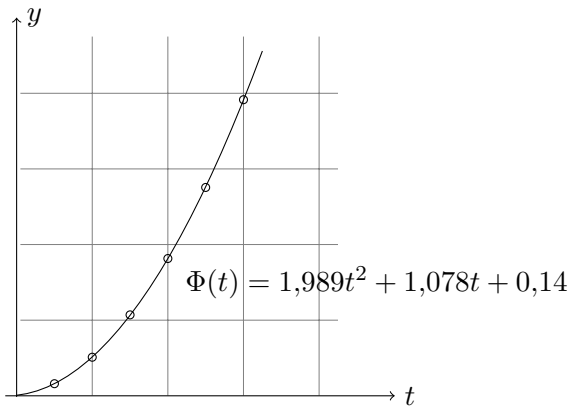

Figura 4.3: Ejemplo PMCL: Datos ajustados

4.3. Descomposición QR

La factorización QR de una matriz es la descomposición de la misma como producto de una matriz ortogonal Q por una triangular superior R.

Esta descomposición se utiliza computacionalmente para resolver sistemas lineales, se aplica a la resolución de problema de mínimos cuadrados y para hallar los valores propios de una matriz.

Teorema 4.3.1. *Sea $A \in \mathcal{M}_{m\times n}$, con $m > n$ con rango n. Existen matrices $Q \in \mathcal{M}_{m\times m}$, $R \in \mathcal{M}_{m\times n}$ tales que: $A = QR$, $Q^tQ = I_m$ o sea Q es ortogonal, y R es triangular superior o sea sus entradas $r_{i,j} = 0$ para todos los $i > j$.*

Proposición 4.3.2. *Sea $Q \in \mathcal{M}_{m\times m}$ ortogonal, o sea $Q^tQ = I_m$ entonces $\|QX\|_2 = \|X\|_2$ para todo $X \in \mathbb{R}^m$ y $Q^{-1} = Q^t$.*

Demostración. Veamos que $\|QX\|_2 = \|X\|_2$ para todo $X \in \mathbb{R}^m$.

$$\|QX\|_2^2 = (QX)^t(QX) = X^tQ^tQX = X^tX = \|X\|_2^2$$

□

Observación 4.3.1. En otras palabras, la proposición establece que las transformaciones ortonormales conservan la norma 2, o que la norma 2 es invariante ante transformaciones ortonormales.

Demostración del teorema 4.3.1: Sea A_i la columna i de A.

1. Definimos:
$$Q_1 = \frac{A_1}{\|A_1\|} \Rightarrow A_1 = r_{11}Q_1 \text{ con } r_{11} = \|A_1\|.$$

2. Veamos que $\{Q_1, \ldots, Q_{k-1}\}$ es un conjunto ortonormal ($Q_i \perp Q_j \quad i \neq j$, $\|Q_i\| = 1 \quad \forall i$).
   Y que además se cumple $[A_1, \ldots, A_{k-1}] = [Q_1, \ldots, Q_{k-1}]$.
   Ya probamos que esto vale para $k = 1$, vamos a probarlo por inducción.

3. Buscamos Q_k tal que $\|Q_k\| = 1$ y $Q_k \perp Q_j \quad j = 1, \ldots, k-1$.

   Definimos:

$$\tilde{Q_k} = A_k - \sum_{i=1}^{k-1} r_{ik} Q_i \text{ tal que } \{Q_1, \ldots, Q_{k-1}, \tilde{Q_k}\} \text{ sea ortogonal.}$$

   Debemos encontrar los r_{ik}:

$$\begin{aligned} < \tilde{Q_k}, Q_j > &= < A_k - \sum_{i=1}^{k-1} r_{ik} Q_i, Q_j > \\ &= < A_k, Q_j > - r_{jk} \qquad \text{porque } Q_i \perp Q_j \ i \neq j. \end{aligned}$$

   Entonces el producto escalar da 0 si se define $r_{jk} =< A_k, Q_j >$ y esto quiere decir que el conjunto $\{Q_1, \ldots, Q_{k-1}, \tilde{Q_k}\}$ es ortogonal.

   Luego definimos $Q_k = \frac{\tilde{Q_k}}{\|\tilde{Q_k}\|}$ y llegamos a que $\{Q_1, \ldots, Q_k\}$ es ortonormal.

   Finalmente llegamos a que $A_k = \sum_{i=1}^{k} r_{ik} Q_i$, lo que quiere decir que $[A_1, \ldots, A_k] = [Q_1, \ldots, Q_k]$.

La descomposición QR se obtiene utilizando los valores r_{ij} como las entradas de R, las columnas $\{Q_i\}_{i=1,\ldots,n}$ halladas anteriormente como columnas de Q que se completan con columnas $\{Q_l\}_{l=n+1,\ldots,m}$ de tal manera que $\{Q_1, \ldots, Q_n, Q_{n+1}, \ldots, Q_m\}$ conformen una base ortonormal.

La relación $A = QR$ se cumplirá ya que ambas matrices fueron construidas para satisfacer $A_k = \sum_{i=1}^{k} r_{ik} Q_i$.

□

Observamos para terminar que el proceso de construcción de Q es análogo al de ortonormalización de bases de Gram-Schmidt.

4.3.1. Aplicación de QR al PMCL

Aplicamos ahora los resultados anteriores al problema de mínimos cuadrados. Suponemos que las ecuaciones normales tienen solución única, o sea es un sistema compatible determinado, entonces buscamos:

$$\begin{aligned} \min_{X \in \mathbb{R}^m} \|AX - Y\|_2^2 &= \min_{X \in \mathbb{R}^m} \|QRX - Y\|_2^2 \\ &= \min_{X \in \mathbb{R}^m} \left\|Q\left(RX - Q^tY\right)\right\|_2^2 \\ &= \min_{X \in \mathbb{R}^m} \left\|RX - Q^tY\right\|_2^2 \\ &= \min_{X \in \mathbb{R}^m} \left\|R_1X - \left(Q^tY\right)_1\right\|_2^2 + \left\|\left(Q^tY\right)_2\right\|_2^2 \end{aligned}$$

donde $R_1 \in \mathcal{M}_{n\times n}$ es la matriz formada por las primeras n filas de R, y $\left(Q^tY\right)_1 \in \mathbb{R}^n$, $\left(Q^tY\right)_2 \in \mathbb{R}^{m-n}$ son los vectores formados por los primeros n y $(m-n)$ elementos de $Q^tY \in \mathbb{R}^m$, respectivamente. Minimizamos tomando $R_1X = \left(Q^tY\right)_1$, ya que $\left\|\left(Q^tY\right)_2\right\|_2^2$ no depende de X. Deducimos que tenemos que resolver el sistema $R_1X = \left(Q^tY\right)_1$.

Observación 4.3.2.

- R_1 es triangular superior, por lo que $R_1X = \left(Q^tY\right)_1$ se resuelve con sustitución hacia atrás.
- La resolución del PMCL con QR está bien condicionada.
- Son necesarias más operaciones para encontrar QR.

Ejemplo 4.3.1. Con los datos del ejemplo 4.2.1, vemos que

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \\ 16 & 4 & 1 \\ 25 & 5 & 1 \\ 36 & 6 & 1 \end{bmatrix} = QR$$

donde

$$Q = \begin{bmatrix} -0{,}021 & -0{,}343 & 0{,}838 & 0{,}112 & -0{,}040 & -0{,}405 \\ -0{,}084 & -0{,}521 & 0{,}168 & -0{,}006 & 0{,}346 & 0{,}757 \\ -0{,}189 & -0{,}535 & -0{,}224 & -0{,}613 & -0{,}488 & -0{,}122 \\ -0{,}335 & -0{,}383 & -0{,}335 & 0{,}754 & -0{,}213 & -0{,}122 \\ -0{,}524 & -0{,}065 & -0{,}168 & -0{,}205 & 0{,}706 & -0{,}391 \\ -0{,}755 & 0{,}417 & 0{,}279 & -0{,}042 & -0{,}311 & 0{,}283 \end{bmatrix}$$

$$R = \begin{bmatrix} -47{,}7 & -9{,}25 & -1{,}91 \\ 0 & -2{,}35 & -1{,}43 \\ 0 & 0 & 0{,}54 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; R_1 = \begin{bmatrix} -47{,}7 & -9{,}25 & -1{,}91 \\ 0 & -2{,}35 & -1{,}43 \\ 0 & 0 & 0{,}54 \end{bmatrix}$$

Entonces, el $X \in \mathbb{R}^3$ que minimice va a ser el X solución del sistema

$$\begin{bmatrix} -47{,}7 & -9{,}25 & -1{,}91 \\ 0 & -2{,}35 & -1{,}43 \\ 0 & 0 & 0{,}54 \end{bmatrix} X = \begin{bmatrix} -105 \\ -2{,}73 \\ 0{,}008 \end{bmatrix}$$

por lo que $X = \begin{bmatrix} 1{,}9893 \\ 1{,}0778 \\ 0{,}1400 \end{bmatrix}$.

△

4.4. Descomposición SVD

Una extensión del popular Teorema Espectral, generalmente trabajado en cursos iniciales de Álgebra Lineal, se conoce como Descomposición en Valores Singulares (SVD en inglés) y es de gran utilidad en múltiples aplicaciones en estadística (análisis de componente principal), teoría de control, compresión de imágenes, etc. Veremos algunos ejemplos de cálculo de esta descomposición, su aplicación al Problema de Mínimos Cuadrados Lineales y su utilización en otras aplicaciones sobre el final del capítulo.

El Teorema Espectral establece las condiciones bajo las cuales un operador o una matriz pueden ser diagonalizados (es decir, representadas como una matriz diagonal en alguna base). En particular, que cualquier matriz simétrica $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ puede descomponerse como $A = PDP^t$ donde D es diagonal y P ortogonal.

Aquí demostraremos que una matriz cualquiera $A \in \mathcal{M}_{m\times n}(\mathbb{R})$, es decir no necesariamente cuadrada, puede descomponerse como $A = USV^t$ donde S es diagonal y U y V son ortogonales (no necesariamente una inversa de la otra).

Trabajaremos primero en las demostraciones genéricas para transformaciones lineales para ver luego como corolario la deducción de la descomposición SVD. [1]

Por tanto, en lo que sigue, V y W son espacios vectoriales con producto interno sobre un cuerpo $\mathbb{K}$.

4.4.1. Descomposición en valores singulares de una transformación lineal

Recordamos en primer lugar un par de definiciones.

Definición 4.4.1. Un operador $T : V \to V$ es autoadjunto si

$$\langle T(v), v\rangle = \langle v, T(v)\rangle \geq 0, \quad \forall v \in V$$

Definición 4.4.2. Un operador autoadjunto $T : V \to V$ se dice no negativo si

$$\langle T(v), v\rangle \geq 0, \quad \forall v \in V$$

Si T es no negativo entonces resulta fácil probar que sus valores propios son todos no negativos, en efecto si λ es valor propio de T con vector propio $v \neq \vec{0}$ entonces

$$0 \leq \langle T(v), v\rangle = \lambda\langle v, v\rangle$$

y como en el último término el segundo factor es positivo, se deduce que $\lambda \geq 0$.

Definición 4.4.3. Un operador autoadjunto $T : V \to V$ se dice no positivo si

$$\langle T(v), v\rangle > 0, \quad \forall v \in V$$

[1] Es posible saltear esta sección y continuar la lectura en el Corolario 4.4.3. Sin embargo, se recomienda apretar los dientes y seguir para comprender los conceptos subyacentes de los que surge su obtención.

En este caso los valores propios son positivos.

El siguiente lema es útil en la demostración del teorema siguiente.

Lema 4.4.1. *Sean V y W espacios vectoriales con producto interno de dimensión finita y sea $T : V \to W$ una transformación lineal, entonces $T * T$ y $TT*$ son ambos autoadjuntos, no negativos y tienen el mismo rango que T.*

Demostración. Probaremos el resultado solo para T^*T, la prueba para TT^* es análoga y qeuda a cargo del lector. Veamos primero que T^*T es autoadjunto:

$$(T^*T)^* = T^*(T^*)^* = T^*T.$$

Ahora, veamos que T^*T es no negativo. En efecto:

$$\langle T^*T(v), v\rangle = \langle T(v), T(v)\rangle \geq 0, \quad \forall v \in V.$$

Finalmente observemos que $rango(T^*T) = rango(T)$. Para esto alcanza con probar que $\ker(T)\ker(T^*T)$:
Es inmediato verificar que,

$$\ker(T) \subset \ker(T^*T).$$

Además, si $v \in \ker(T^*T)$ entonces

$$0 = \langle T^*T(v), v\rangle = \langle T(v), T(v)\rangle,$$

por lo tanto $T(v) = \vec{0}$ y consecuentemente $v \in ker(T)$. Como $ker(T^*T) = ker(T)$ se deduce inmediatamente del teorema de las dimensiones que

$$\dim(Im(T^*T)) = \dim(Im(T))$$

lo cual concluye la prueba. □

Teorema 4.4.2. *Sean V y W espacios vectoriales con $\dim(V) = n$ y $\dim(W) = m$ y sea $T : V \to W$ una transformación lineal con $rango(T) = r$.*
Entonces existe $\mathcal{A} = \{v_1, \ldots, v_n\}$ base ortonormal de V, $\mathcal{B} = \{w_1, \ldots, w_m\}$ base ortonormal de W y escalares $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ tales que $T(v_i) = \sigma_i w_i$ si $i = 1, \ldots, r$ y $T(v_i) = \vec{0}$ si $i = r+1, \ldots, n$. Es decir:

$$T(v_i) = \sigma_i w_i \qquad con \qquad \sigma_i > 0 \quad \forall i = 1, \ldots, r \qquad y \qquad \sigma_i = 0 \quad i = r+1, \ldots, n.$$

Demostración. Sea $R = T^*T$, por el lema anterior se sabe que R es un operador lineal en V autoadjunto, no negativo y tiene rango r (pues $r = rango(T)$).
Por el teorema espectral existe $\mathcal{A} = \{v_1, \ldots, v_n\}$ base ortonormal de V tal que $R(v_i) = \lambda_i v_i \ \forall i = 1, \ldots, n$. Como $rango(T) = r$ se tiene que $\dim(\ker(T)) = n - r$ por lo tanto la base A se puede elegir de modo que $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$ y $\lambda_i = 0$ si $i = r+1, \ldots, n$. Para $i = 1, \ldots, r$ definimos

$$w_i = \frac{T(v_i)}{\sigma_i}$$

donde $\sigma_i = \sqrt{\lambda_i}$. Entonces se tiene que $T(v_i) = \sigma_i w_i \ \forall i = 1, \ldots, r$ y $T(v_i) = 0 \ \forall i = r+1, \ldots, n$ tal como se quería. Falta probar que podemos construir $\mathcal{B} = \{w_1, \ldots, w_n\}$ definida de este modo

y es una base ortonormal de V.
Veamos en primer lugar que $\mathcal{B}' = \{w_1, \ldots, w_r\} \subset W$ es un conjunto ortonormal y consecuentemente linealmente independiente. En efecto:

$$\langle w_i, w_j \rangle = \langle \frac{T(v_i)}{\sigma_i}, \frac{T(v_j)}{\sigma_j} \rangle = \langle \frac{v_i}{\sigma_i}, \frac{T^*T(v_j)}{\sigma_j} \rangle = \frac{1}{\sigma_i \sigma_j} \langle v_i, R(v_j) \rangle = \frac{1}{\sigma_i \sigma_j} \langle v_i, \lambda_j v_j \rangle = \frac{\lambda_j}{\sigma_i \sigma_j} \langle v_i, v_j \rangle$$

Por lo tanto, usando que $A = \{v_1, \ldots, v_n\}$ es una base ortonormal de V, resulta que $\mathcal{B}'$ es un conjunto ortonormal. Sea $S = SG(\{w_1, \ldots, w_r\})$, entonces $\mathcal{B}'$ es una base ortonormal de S. Además se puede construir $\mathcal{B}'' = \{w_{r+1}, \ldots, w_m\}$ base ortonormal de $S^\perp$ y por lo tanto $\mathcal{B} = \mathcal{B}' \cup \mathcal{B}'' = \{w_1, \ldots, w_r, w_{r+1}, \ldots, w_m\}$ es una base ortonormal de W con las propiedades deseadas.

□

4.4.2. Descomposición en valores singulares de una matriz

Corolario 4.4.3 (Descomposición SVD)**.** *Sea $A \in \mathcal{M}_{m\times n}(\mathbb{R})$ de rango r. Entonces existen $U \in \mathcal{M}_{m\times m}(\mathbb{R})$ y $V \in \mathcal{M}_{n\times n}(\mathbb{R})$ ambas ortogonales y $\Sigma \in \mathcal{M}_{m\times n}(\mathbb{R})$ diagonal con rango$(\Sigma) = r$ tal que $A = U\Sigma V^t$.*

Observación 4.4.1. La matriz Σ es de la forma: $\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} = \left[\begin{array}{c|c} \Sigma_r & 0^{r\times(n-r)} \\ \hline 0^{(m-r)\times r} & 0^{(m-r)\times(n-r)} \end{array}\right]$

donde $\Sigma_r = \begin{bmatrix} \sigma_1 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \sigma_r \end{bmatrix} \in \mathbb{R}^{r\times r}$ matriz diagonal que cumple $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$.
A los $\{\sigma_i\}_{i\in 1\ldots r}$ se les denomina *valores singulares* de A.
Además, $U = [U_1|U_2]$ con $U_1 \in \mathcal{M}_{m\times r}(R)$, $V = [V_1|V_2]$ con $V_1 \in \mathcal{M}_{n\times r}(\mathbb{R})$, y cumplen $A = U_1 \Sigma_r V_1^T$.

Demostración. Basta elegir $T : \mathbb{R}^n \to \mathbb{R}^m$ tal que $T(x) = Ax$, entonces $A = {}_{\mathcal{C}_m}((T))_{\mathcal{C}_n}$ donde $\mathcal{C}_i$ es la base canónica de $\mathbb{R}^i$. Por el teorema anterior se tiene que existen $\mathcal{A}$ y mcB bases ortonormales de $\mathbb{R}^n$ y $\mathbb{R}^m$ respectivamente tales que $T(v_i) = \sigma_i w_i$ con $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ y $\sigma_i = 0$ si $i > r$.

Por lo tanto, ${}_{\mathcal{B}}((T))_{\mathcal{A}} = \begin{bmatrix} \sigma_1 & & & & & & \\ & \sigma_2 & & & & & \\ & & \ddots & & & & \\ & & & \sigma_r & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix}$ Además

$$_{\mathcal{C}_m}((T))_{\mathcal{C}_n} = {}_{\mathcal{C}_m}((Id))_{\mathcal{B}\mathcal{B}}((T))_{\mathcal{A}\mathcal{A}}((Id))_{\mathcal{C}_n}$$

Las matrices de cambio de base ${}_{\mathcal{C}_m}((Id))_{\mathcal{B}}$ y ${}_{\mathcal{A}}((Id))_{\mathcal{C}_n}$ son ortogonales pues transforman una base ortonormal en otra. Por lo tanto, poniendo

$$U = {}_{\mathcal{C}_m}((Id))_{\mathcal{B}}, \qquad V = {}_{\mathcal{C}_n}((Id))_{\mathcal{A}}, \qquad \Sigma = {}_{\mathcal{B}}((T))_{\mathcal{A}}$$

se tiene que $A = U\Sigma V^t$ y además $rango(\Sigma) = r$, lo cual concluye la prueba. □

Observación 4.4.2. Los valores singulares de A son $\sigma_i = \sqrt{\lambda_i}$ donde λ_i es valor propio de A^TA. Se cumple además que $\lambda_i \geq 0$ ya que A^TA es semidefinida positiva.

Ejemplo 4.4.1. Retomando el ejemplo 4.2.1, veamos que $A = U\Sigma V^T$ donde:

$$U = [U_1|U_2] = \left[\begin{array}{ccc|ccc} -0{,}025 & 0{,}434 & 0{,}795 & 0{,}112 & -0{,}040 & -0{,}405 \\ -0{,}099 & 0{,}536 & 0{,}109 & -0{,}006 & 0{,}346 & 0{,}757 \\ -0{,}194 & 0{,}505 & -0{,}281 & -0{,}613 & -0{,}488 & -0{,}122 \\ -0{,}339 & 0{,}339 & -0{,}375 & 0{,}754 & -0{,}213 & -0{,}122 \\ -0{,}525 & 0{,}041 & -0{,}173 & -0{,}205 & 0{,}706 & -0{,}391 \\ -0{,}750 & -0{,}391 & 0{,}324 & -0{,}042 & -0{,}311 & 0{,}283 \end{array}\right]$$

$$\Sigma = \left[\begin{array}{c} \Sigma_1 \\ \hline 0 \end{array}\right] = \left[\begin{array}{ccc} 48{,}6 & 0 & 0 \\ 0 & 2{,}72 & 0 \\ 0 & 0 & 0{,}473 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

$$V = V_1 = \left[\begin{array}{ccc} -0{,}981 & -0{,}182 & 0{,}070 \\ -0{,}191 & 0{,}823 & -0{,}535 \\ -0{,}040 & 0{,}538 & 0{,}842 \end{array}\right]$$

y

$$X = V_1\Sigma_1^{-1}U_1^TY = \left[\begin{array}{c} 1{,}9893 \\ 1{,}0778 \\ 0{,}1400 \end{array}\right]$$

△

4.4.3. Aplicación de SVD al PMCL

Veremos a continuación, cómo se aplica la descomposición SVD al PMCL. En este caso no necesitamos asumir que las ecuaciones normales tienen solución única.

Teorema 4.4.4. *Sea $b \in \mathbb{R}^m$ y $A \in \mathcal{M}_{n\times m}(\mathbb{R})$ de rango r.*
Sea $A = U\Sigma V^t$ su descomposición SVD.
Entonces, la solución al PMCL de norma mínima es el vector dado por

$$\hat{X} = V \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^t b$$

Demostración. Consideramos

$$\|b - AX\|_2 = \|U^t(b - AVV^tX)\|_2 = \|(\underbrace{U^tb}_{C} - \underbrace{U^tAV}_{\Sigma}\underbrace{V^tX}_{Z})\|_2$$

donde la primera igualdad es cierta por ser U^t ortogonal y $VV^t = Id$.
Luego, realizando los cambios de variables $Z = V^tX$ y $C = U^tb$ se tiene:

$$\|b - AX\|_2 = \|C - \Sigma Z\|_2 = \left\| \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} - \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} c1 - \Sigma_r z_1 \\ c_2 \end{pmatrix} \right\|_2$$

donde $z_1 \in \mathbb{R}^r$, $z_2 \in \mathbb{R}^{n-r}$, $c_1 \in \mathbb{R}^r$, y $c_2 \in \mathbb{R}^{m-r}$.

Esta norma se hace mínima cuando $c_1 = \Sigma_r z_1$ (z_2 arbitrario).
Tomando $z_2 = \vec{0} \in \mathbb{R}^{m-n}$, tendremos que $\|Z\|_2$ es mínima.
La norma $\|X\|_2$ también es mínima puesto que $\|X\|_2 = \|VZ\|_2 = \|Z\|_2$, ya que V es ortogonal.

Llamando $\hat{Z}$ a la solución de norma mínima, se tiene que

$$\hat{X} = V\hat{Z} = V \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = V \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^tb \Rightarrow \hat{X} = V \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^tb$$

□

4.5. Descomposición de Cholesky*

Teorema 4.5.1. *Sea $B \in \mathcal{M}^{n\times n}(\mathbb{R})$ una matriz simétrica definida positiva. Es decir*

- $B = B^t$
- $X^tBX > 0 \ \forall X \neq \vec{0}$

Entonces existe una matriz $C \in \mathcal{M}_{n\times n}(\mathbb{R})$ triangular inferior tal que $CC^t = B$.
A esta descomposición se le denomina Descomposición de Cholesky.

Demostración. Basta observar que B es diagonalizable con todos sus valores propios positivos y vectores propios formando una base ortogonal, $B = LDL^t$. Descomponiendo D como $D = \sqrt{D}\sqrt{D}$ y operando se concluye, por lo que se dejan los detalles al lector. □

Aplicación: Uso de Cholesky para resolver p problemas de mínimos cuadrados.

Supongamos que tenemos p problemas de mínimos cuadrados de la forma $PMCL_i = \text{mín}\,\|b_i - AX\|$ con $X \in \mathbb{R}^n$, $A \in \mathcal{M}^{n\times m}(\mathbb{R})$, $b_i \in \mathbb{R}^m$, $i \in 1, \ldots, p$.

Planteamos p sistemas de ecuaciones normales $A^tAX = A^tb_i$, con $i = 1, \ldots, p$.
Por su parte, A^tA es simétrica y definida positiva:

- $(A^tA)^t = A^t(A^t)^t = A^tA$
- $X^tA^tAX = \langle AX, AX\rangle = \|AX\|_2 > 0 \ \forall X \neq \vec{0}$

Por tanto, A^tA admite una descomposición de Cholesky, es decir, existe $C \in \mathcal{M}_{n\times n}(\mathbb{R})$ triangular inferior tal que $CC^t = A^tA$.

Sea $\hat{b}_i = A^tb_i$, entonces $A^tAX = A^tb_i$ se reescribe como

$$CC^tX = \hat{b}_i \quad i \in 1, \ldots, p$$

el cual puede descomponerse en dos pasos:

$$\begin{cases} C^tX = y \ \ (1) \\ Cy = \hat{b}_i \ \ (2) \end{cases}$$

Primero se resuelve el sistema triangular (2) para obtener $y \in \mathbb{R}^n$ y luego se resuelve el sistema triangular (1) para hallar la solución del i-ésimo problema de mínimos cuadrados.

Observación 4.5.1. La cantidad de operaciones requeridas para la resolución de los p problemas con este método es pn^2.

4.6. Mínimos Cuadrados No Lineales (PMCNL)

Veremos ahora el caso más general del problema de mínimos cuadrados. Lo que hicimos en el caso lineal era ajustar una función a ciertos datos y lineal respecto a ciertos parámetros. Supongamos como antes que tenemos ciertos datos dados por la siguiente tabla

t	y
t_1	y_1
$\vdots$	$\vdots$
t_m	y_m

y una función de ajuste $\Phi(x_1, x_2, \ldots, x_n, t) = f(X, t)$ no lineal con respecto a los parámetros de ajuste $x_1, x_2, \ldots, x_n$.

Ejemplo 4.6.1. La función $f(x_1, x_2, t) = \sin(x_1t)\, e^{x_2t}$, no es lineal con respecto a los parámetros x_1, x_2. △

Definición 4.6.1. Definimos el residuo, como antes, como la siguiente función:

$$R(x_1, x_2, \cdots, x_n) = R(X) = \begin{bmatrix} f(X,t_1) - y_1 \\ f(X,t_2) - y_2 \\ \vdots \\ f(X,t_m) - y_m \end{bmatrix} = F(X) - Y$$

donde denotamos

$$F(X) = \begin{bmatrix} f(X,t_1) \\ f(X,t_2) \\ \vdots \\ f(X,t_m) \end{bmatrix}$$

que depende de los parámetros $x_1, x_2, \cdots, x_n$.

El problema de mínimos cuadrados no lineal consiste en hallar un $\hat{X}$ que alcance el siguiente mínimo:

$$\min_{X \in \mathbb{R}^n} \|F(X) - Y\|_2^2$$

Presentamos el algoritmo de Gauss-Newton para encontrar una solución al problema de mínimos cuadrados no lineal. La idea del algoritmo es suponer que tenemos una buena aproximación de la solución X_k y linealizar el residuo en un entorno de X_k. Luego resolver un problema de mínimos cuadrados lineal para obtener una mejor aproximación X_{k+1}.

$$R(Z) \approx R(X_k) + J_R(X_k)(Z - X_k)$$

Buscamos minimizar $R(Z)$ y definimos el próximo punto X_{k+1} de la sucesión en el Z en que se da este mínimo. Por tanto, haciendo el cambio de variable $P_k = X_{k-1} - X_k$, en cada paso de la iteración se deberá hallar:

$$\min_{P_k \in \mathbb{R}^n} \|R(X_k) + J_R(X_k)P_k\|_2^2$$

Observación 4.6.1. $J_R(X_k) = -J_F(X_k)$

Por tanto, esquematizamos el algoritmo de resolución de PMCNL:

Algoritmo 9 Algoritmo de Gauss-Newton (PMCNL)

0: $X_0 \in \mathbb{R}^n$
k+1: $A_k \leftarrow J_F(X_k)$
$Y_k \leftarrow Y - F(X_k)$
Resuelvo PMCL:
$P_k \leftarrow$ solución de $\min_{P_k \in \mathbb{R}^n} \|Y_k - A_k P_k\|_2^2$
$X_{k+1} \leftarrow X_k + P_k$

Capítulo 5

Interpolación

El problema de interpolación consiste en encontrar una función $f(x)$ desconocida a partir de un conjunto de puntos $\{(x_i, y_i) : i = 0, \ldots, n\}$ tales que $f(x_i) = y_i$, que serán datos. Este problema es imposible de resolver ya que la ley subyacente que relaciona las variables podría ser cualquiera: existen infinitas funciones que pasan por esos puntos. Por tanto, el objetivo es encontrar buenas aproximaciones para $f(x)$, imponiendo además que la función que aproximará a $f(x)$ pase por todos los puntos.

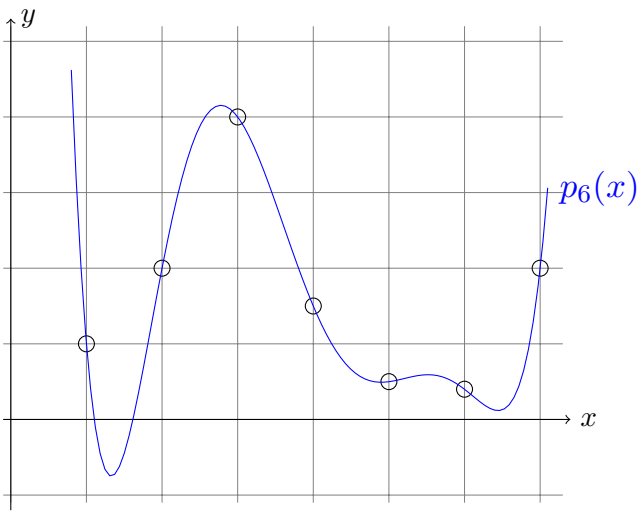

Figura 5.1: Interpolación polinómica

5.1. Interpolación de Vandermonde

La interpolación de Vandermonde se resume en encontrar un polinomio $P(x)$ tal que $P(x_i) = y_i = f(x_i)$ $\forall i = 0, \ldots, n$. Este es un método global ya que se busca una única expresión polinómica que funcione para la totalidad de los datos, a diferencia de ciertos métodos llamados a trozos, que veremos a partir de la Sección 5.6, en los que cada intervalo $[x_i, x_{i+1}]$ tiene una expresión diferente.

Encontrar el polinomio interpolante significa determinar sus coeficientes, por lo que si tenemos $n+1$ datos, podríamos plantear un polinomio de grado menor o igual a n. Entonces escribimos su expresión de esta forma:

$$P(x) = \sum_{j=0}^{j=n} a_j x^j$$

Imponiendo que el polinomio pase por los puntos dados obtenemos el siguiente sistema de ecuaciones:

$$\begin{cases} P(x_0) = \sum_{j=0}^{j=n} a_j x_0^j \\ P(x_1) = \sum_{j=0}^{j=n} a_j x_1^j \\ \vdots \\ P(x_n) = \sum_{j=0}^{j=n} a_j x_n^j \end{cases}$$

No confundir: aquí las incógnitas son los a_j; los x_j son datos al igual que los y_j.

Por lo tanto, la versión matricial del sistema anterior es:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{5.1}$$

A esta matriz que presenta una progresión geométrica en cada fila se le llama *Matriz de Vandermonde*, y la notaremos V.

Es así entonces que los coeficientes de $P(x)$ son solución del sistema 5.1.

Una propiedad de la Matriz de Vandermonde es que su determinante se expresa mediante la fórmula:

$$\det(V) = \prod_{0 \le i < j \le n} (x_j - x_i)$$

y como $x_i \neq x_j \ \forall i \neq j$ se llega a que el $\det(V)$ es no nulo, y así el sistema tiene solución.

Otra propiedad de la Matriz de Vandermonde es que tiene un número de condición grande, por lo que el sistema está mal condicionado y podemos cometer errores importantes en la determinación de los a_i.

Finalmente, otra problemática con este método para obtener $P(x)$ es que resulta difícil agregar nuevos puntos a la tabla, ya que se debe recalcular un nuevo sistema.

Antes de continuar con otros métodos de interpolación veamos algunos resultados interesantes.

5.1.1. Aproximación por polinomios

Los beneficios de la interpolación polinómica es que además de ser funciones simples, los computadores pueden evaluarlos directamente.

Pero además, toda función continua en un intervalo $[a, b]$ puede ser aproximada uniformemente por un polinomio tan cerca como se quiera. En otras palabras, los polinomios son densos en las funciones continuas sobre un intervalo cerrado.

Teorema 5.1.1 (Stone-Weierstrass). *Sea f continua en $[a, b]$. Entonces, $\forall \varepsilon > 0$, $\exists p(x)$ tal que $|f(x) - p(x)| < \varepsilon \quad \forall x \in [a, b]$.*

Veremos también que este teorema en realidad establece la existencia de un polinomio muy cercano a la función, pero que no necesariamente pasa por los puntos interpolantes. En realidad, cómo elegir los puntos para caer en la banda es aún un problema abierto.

5.1.2. Existencia y unicidad de P(x)

Haciendo uso de las propiedades de la Matriz de Vandermonde, se demuestra que siempre es posible encontrar un polinomio de grado menor o igual a n que pase por los $n + 1$ puntos.

Veremos ahora la unicidad:

Sean dos polinomios $p(x)$ y $q(x)$ tales que:

$$\begin{cases} p(x_i) = y_i & \forall i = 0, \ldots, n \quad gr(p) \leq n \\ q(x_i) = y_i & \forall i = 0, \ldots, n \quad gr(q) \leq n \end{cases}$$

Sea el polinomio $o = p - q \implies o(x_i) = p(x_i) - q(x_i) = y_i - y_i = 0 \; \forall i = 0, \ldots, n$.

Entonces el polinomio o tiene $n + 1$ raíces y $gr(o) \leq n \implies o \equiv 0 \implies p = q$.

¿Pero qué importancia tiene esto?

La unicidad del polinomio interpolante implica que independientemente del método que utilicemos para hallar un polinomio que pase por $n + 1$ puntos, sus coeficientes van a ser los mismos. Continuaremos viendo en la próxima sección el método de Lagrange y en la siguiente el método de Newton, ¡pero todos estos métodos van a arrojar el mismo resultado! Sí, así es, veremos al menos dos procedimientos más para hallar lo mismo. Por tanto, posiblemente el sagaz lector se preguntará: ¿Y entonces qué sentido tiene verlos? Lo invitamos a descubrirlo continuando la lectura.

5.2. Interpolación de Lagrange

El método de *interpolación de Lagrange* busca encontrar un polinomio $P(x)$ tal que $P(x_i) = y_i = f(x_i) \; \forall i = 0, \ldots, n$.

Sin embargo, el planteo de Lagrange es escribir $P(x)$ como combinación lineal de una base de polinomios $\{l_j(x)\}$ a los que les llamamos polinomios de Lagrange. Así,

$$P(x) = \sum_{j=0}^{j=n} y_j l_j(x)$$

y como debe cumplirse $P(x_i) = y_i$ queda que $P(x_i) = \sum_{j=0}^{j=n} y_j l_j(x_i)$, por tanto:

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

Esto significa que un polinomio de Lagrange genérico $l_k(x)$ se tiene que anular en todos los puntos x_i, salvo en x_k. De esta manera, en la sumatoria $P(x_i) = \sum_{j=0}^{j=n} y_j l_j(x_i)$ sólo sobrevivirá $l_i(x_i) = 1$ y por tanto se verificará que $P(x_i) = y_1 l_1(x_i) + \cdots + y_i l_i(x_i) + \cdots + y_n l_n(x_i) = y_i l_i(x_i) = y_i$.

Ahora bien, para que $l_j(x_i) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$ necesitamos que $l_j(x)$ tenga raíces en $x_0, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n$ y que $l_j(x_j) = 1$. Ingeniosamente, se comprueban estas condiciones expresando:

$$l_j(x) = \frac{(x - x_0)(x - x_1)(x - x_2)\ldots(x - x_{j-1})(x - x_{j+1})\ldots(x - x_n)}{(x_j - x_0)(x_j - x_1)(x_j - x_2)\ldots(x_j - x_{j-1})(x_j - x_{j+1})\ldots(x_j - x_n)}$$

Observamos que:

- No tenemos problemas de mal condicionamiento
- Los únicos errores numéricos aparecen en el cálculo de los $l_i(x)$ y al realizar las operaciones.

Nótese que el método de Lagrange permite expresar directamente el polinomio interpolante, sin efectuar cálculos. No obstante, tal expresión polinómica no es simple de manipular (derivar, integrar o evaluar).

5.3. Interpolación de Newton

El *método de Newton* propone construir el polinomio interpolante mediante un proceso iterativo en el que en cada paso se lo obliga a pasar por un nuevo punto. Este método es particularmente útil cuando tenemos el polinomio interpolante que pasa por los $\{(x_i, y_i\}_{i=0,\ldots,n}$ y nos llega un nuevo dato (x_{n+1}, y_{n+1}), o sea, es sencillo recalcular $P(x)$ para agregar puntos interpolados.

El planteo es de este modo:

Sea $P_k(x)$ tal que $P_k(x_i) = y_i \ \forall i = 0, \ldots, k$.

Buscar $P_{k+1}(x)$ tal que $P_{k+1}(x_i) = y_i \ \forall i = 0, \ldots, k, k+1$.

Escribimos:

$$P_{k+1}(x) = P_k(x) + q_{k+1}(x)$$

¿Qué condiciones debemos imponer?

1. El nuevo polinomio debe pasar por todos los puntos anteriores:
   $P_{k+1}(x_i) = \underbrace{P_k(x_i)}_{y_i} + q_{k+1}(x_i) = y_i \quad \forall i = 0, \ldots, k$
   $\implies q_{k+1}(x_i) = 0 \quad \forall i = 0, \ldots, k \quad gr(q_{k+1}) = k+1$
   Entonces
   $$q_{k+1}(x) = (x - x_0)(x - x_1) \ldots (x - x_k)a_k$$

2. El nuevo polinomio debe pasar por el nuevo punto:
   $P_{k+1}(x_{k+1}) = P_k(x_{k+1}) + q_{k+1}(x_{k+1}) = y_{k+1}$
   $\implies q_{k+1}(x_{k+1}) = y_{k+1} - P_k(x_{k+1})$
   De donde determinamos que:

$$a_k = \frac{y_{k+1} - P_k(x_{k+1})}{(x_{k+1} - x_0)(x_{k+1} - x_1) \ldots (x_{k+1} - x_k)}$$

A modo de resumen, dados los $n+1$ puntos a interpolar: $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, el método de Newton consiste en encontrar los a_i que satisfacen:

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

siendo

$$w_i(x) = \begin{cases} 1 & \text{si } i = 0 \\ \prod_{0 \le j < i}(x - x_j) & \text{si } i = 1, \ldots, n \end{cases}$$

A partir de esta base se plantea y resuelve el siguiente sistema:

$$\begin{array}{ll} (x_0, y_0): & a_0 = y_0 \\ (x_1, y_1): & a_0 + a_1(x_1 - x_0) = y_1 \\ (x_2, y_2): & a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) = y_2 \\ & \vdots \\ (x_n, y_n): & a_0 + \sum_{i=1}^{n} a_i \prod_{j=0}^{j=i-1}(x_i - x_j) = y_n \end{array}$$

Además, el valor de a_k puede expresarse en términos de las llamadas *diferencias divididas*:

$$a_0 = f[x_0],\ a_1 = f[x_0, x_1], \ldots,\ a_j = f[x_0, x_1, \ldots, x_k]$$

donde:

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

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

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i},$$

y en general:

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

Ejemplo 5.3.1. Expresemos el polinomio interpolante por los puntos $\{(0,0),(1,1),(2,2),(3,4)\}$ utilizando el método de Newton.

El sistema queda:

$$\begin{array}{ll} (0,0): & a_0 = 0 \\ (1,1): & a_0 + a_1(1-0) = 1 \Rightarrow a_1 = 1 \\ (2,2): & a_0 + a_1(2-0) + a_2(2-0)(2-1) = 2 \Rightarrow a_2 = 0 \\ (3,4): & a_0 + a_1(3-0) + a_2(3-0)(3-1) + a_3(3-0)(3-1)(3-2) = 4 \Rightarrow a_3 = \frac{1}{6} \end{array}$$

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

Mostramos a continuación cómo es posible obtener los coeficientes del polinomio utilizando el esquema de diferencias divididas:

$x_0 = 0$	$f[x_0] = \boxed{0}$			
		$f[x_0, x_1] = \frac{1-0}{1-0} = \boxed{1}$		
$x_1 = 1$	$f[x_1] = 1$		$f[x_0, x_1, x_2] = \frac{1-1}{2-0} = \boxed{0}$	
		$f[x_1, x_2] = \frac{2-1}{2-1} = 1$		$f[x_0, x_1, x_2, x_3] = \frac{\frac{1}{2}-0}{3-0} = \boxed{\frac{1}{6}}$
$x_2 = 2$	$f[x_2] = 2$		$f[x_1, x_2, x_3] = \frac{2-1}{3-1} = \frac{1}{2}$	
		$f[x_2, x_3] = \frac{4-2}{3-2} = 2$		
$x_3 = 3$	$f[x_3] = 4$			

△

5.4. Error de Interpolación Polinómica

Hasta el momento hemos intentado encontrar un polinomio que pase por el conjunto de puntos dados. Ahora bien, este polinomio es un "buen" polinomio para lo que queremos hacer que es a fin de cuentas aproximar la función subyacente.

Podríamos definir como medida del error para cada punto x, $E(x) = f(x) - P_n(x)$. Hacemos notar que en los datos del problema, este error es nulo, ya que $E(x_i) = f(x_i) - P_n(x_i) = y_i - y_i = 0$.

Pero con esta definición, cómo podemos establecer cuál es el error que cometemos si justamente $f(x)$ es desconocida.

Enunciamos a continuación un teorema que puede ser de utilidad si conocemos además algún otro dato de $f(x)$ por ejemplo, su velocidad de variación máxima (o pendiente).

Teorema 5.4.1 (Teorema de acotación del error en Interpolación Polinómica)**.** *Sea f de clase C^{n+1} en el intervalo $[x_0, x_n]$ y p_n el polinomio interpolante a f por las abscisas $x_0 < x_1 < \ldots < x_n$. Luego, para cada $x \in [x_0, x_n]$ existe $\gamma_x \in [x_0, x_n]$ tal que se cumple la siguiente igualdad para el error $E(x)$:*

$$E(x) = f(x) - p_n(x) = \frac{f^{n+1}(\gamma_x)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

Demostración. Tomemos $x \in [x_0, x_n]$ fijo. Si $x = x_i$ para algún i, la igualdad es evidente, pues $E(x_i) = 0$. Supongamos entonces que x no coincide con ninguna de las abscisas. Consideremos la función auxiliar $F : [x_0, x_n] \to \mathbb{R}$ tal que

$$F(t) = f(t) - p_n(t) - (f(x) - p_n(x)) \frac{\prod_{i=0}^{n}(t - x_i)}{\prod_{i=0}^{n}(x - x_i)}.$$

Se menciona, ya que puede resultar confuso, que la función F depende de la variable t, no de x, que pensamos fijo. Se observa que $F(x_i) = 0$ para cada $i = 0, \ldots, n$, y además $F(x) = 0$, por lo que F tiene al menos $n + 2$ raíces en $[x_0, x_n]$. Como f es de clase C^{n+1}, resulta que F también es de clase C^{n+1}. Sean $p_0 < p_1 \ldots < p_{n+1}$ las $n + 2$ raíces de F. Aplicando el Teorema de Rolle en cada intervalo podemos asegurar que existen $n + 1$ raíces $p'_0 < \ldots < p'_n$ para F' (nuevamente recalcamos que la derivada es respecto a t). Aplicando reiteradas veces el teorema de Rolle es posible asegurar la existencia de una raíz $\gamma_x \in [x_0, x_n]$ de la función $F^{(n+1)}$. Como p_n es un polinomio de grado n o menos, tenemos que su derivada de orden $n + 1$ es nula. Entonces, al derivar $n + 1$ veces la función F tenemos que:

$$F^{(n+1)}(\gamma_x) = f^{(n+1)}(\gamma_x) - (n+1)! \frac{f(x) - p_n(x)}{\prod_{i=0}^{n}(x - x_i)} = 0.$$

Finalmente despejando $f(x) - p_n(x)$ se prueba el enunciado. □

Corolario 5.4.2. $E(x) \leq \frac{(x_n - x_0)^{n+1}}{(n+1)!} |f^{n+1}(\gamma_x)|$

Corolario 5.4.3. $E(x) \leq \frac{(x_n - x_0)^{n+1}}{(n+1)!} \|f^{n+1}\|_{\infty,[x_0,x_n]}$

Parecería natural pensar que conociendo más puntos se mejora la interpolación, y se constataría si se observa que si n crece, también lo hace $(n+1)!$ y el error de interpolación debería disminuir. Sin embargo, podría darse el caso en que $f^{n+1}(\gamma_x)$ crezca más rápido que $(n+1)!$ y hacer que el error aumente.

Un caso destacable en que se aprecia esta particularidad se desarrolla en lo que sigue.

5.4.1. Fenómeno de Runge

El fenómeno de Runge es un problema que aparece al aproximar determinadas funciones aplicando interpolación polinómica con polinomios de alto grado utilizando nodos equidistantes. El nombre se debe a Carl Runge cuando exploraba el comportamiento de los errores al usar interpolación polinómica, descubriendo que en algunos casos, aumentar el número de puntos empeora la aproximación.

Una función en la que se puede observar el fenómeno es la llamada *función de Runge*: $f(x) = \frac{1}{1+25x^2}$.

En efecto, si se interpola usando nodos equidistantes entre -1 y 1, el polinomio resultante presenta oscilaciones hacia los extremos del intervalo, como se aprecia en la Figura 5.2.

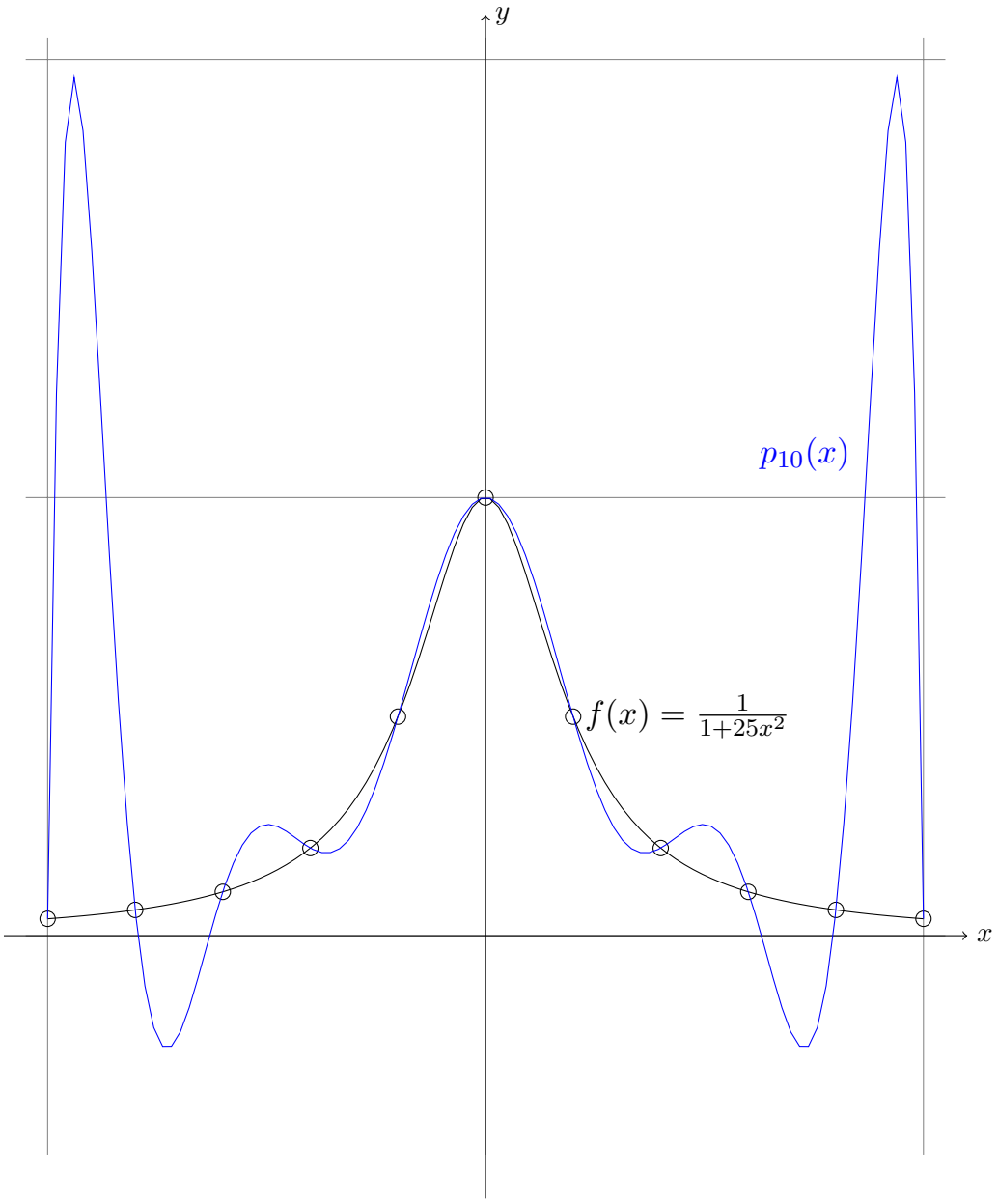

Figura 5.2: Función de Runge e interpolación

Se invita al lector a determinar la expresión genérica de los puntos $x_i \forall i = 0, \ldots, n$, implementar el método de resolución de Vandermonde, y verificar gráficamente el fenómeno al aumentar n.

El fenómeno demuestra que aumentar el grado del polinomio interpolante en general no es la

mejor opción pese a lo que intuitivamente se podría esperar al forzar al polinomio a pasar por más puntos de $f(x)$.

Por otro lado, por el Teorema de Stone-Weierstrass, la familia de funciones polinómicas permite aproximación uniforme de cualquier función continua dentro de un dominio compacto. No obstante, la correcta selección de polinomios interpolantes depende de la elección de las abscisas de interpolación. Este problema de selección de puntos interpolantes es tema actual de investigación.

Algunas ideas de soluciones a este problema son elegir puntos que no sean equidistantes (ej. nodos de Chevyshev), utilizar interpolación a trozos, o tal vez se podría imponer en el polinomio interpolante no solamente que pase por el punto, sino que lo haga con cierta pendiente, obteniendo así un mayor control sobre el polinomio.

5.5. Interpolación de Hermite

En muchas aplicaciones interesa considerar polinomios $P(x)$ que además de interpolar a $f(x)$, interpolan a $f'(x)$, es decir:

$$\begin{cases} P(x_i) = y_i = f(x_i) \\ P'(x_i) = y_i' = f'(x_i) \quad \forall i = 0, 1, \ldots, n \end{cases} \tag{5.2}$$

Con estas restricciones, se controla no solamente por qué puntos debe pasar el polinomio, sino que además, con qué pendiente debe pasar por ellos. Es decir, que el $P(x)$ pasará por los puntos (x_i, y_i) con derivada y_i', donde estos valores son datos conocidos.

Al método que desarrollaremos a continuación que interpola una función $f(x)$ y su derivada $f'(x)$ en $n+1$ puntos se le llama *Método de Hermite*.

Es interesante observar que en este caso, se tienen $2n+2$ condiciones a imponer, por lo que se busca un polinomio $P(x)$ de al menos grado $2n+1$, es decir tenemos $2n+2$ coeficientes a hallar.

Es así que se define el *Polinomio de Hermite* mediante la siguiente expresión:

$$H_{2n+1}(x) = \sum_{i=0}^{i=n} (y_i h_i(x) + y_i' \tilde{h}_i(x))$$

donde $h_i(x)$ y $\tilde{h}_i(x)$ pasan a ser nuevos polinomios, también de grado al menos $2n+1$ a determinar, pero que deben cumplir para todo $0 \leq i, j \leq n$:[1]

$$\begin{array}{ll} h_i(x_j) = \delta_{ij} & h_i'(x_j) = 0 \\ \tilde{h}_i(x_j) = 0 & \tilde{h}_i'(x_j) = \delta_{ij} \end{array}$$

¡Pero en esta ecuación hemos introducido $2n+2$ nuevos polinomios! ¿Cómo puede ser esto más conveniente?

[1] Recordamos que $\delta_{ij} = \begin{cases} 1 \text{ si } i = j \\ 0 \text{ si } i \neq j \end{cases}$

Antes de responder esto, veamos primero que las condiciones impuestas hacen que $H_{2n+1}(x)$ interpole $f(x)$ y $f'(x)$ en los puntos x_i, es decir, que es equivalente al sistema 5.2. O dicho de otro modo, evaluando en cualquiera de los puntos, por ejemplo en x_k, deberíamos constatar que $H_{2n+1}(x_k) = y_k$ y $H'_{2n+1}(x_k) = y'_k$.

En efecto, evaluando en x_k se anulan todas las $\tilde{h}_i$, pues $\tilde{h}_i(x_j) = 0 \forall i, j$, y además se apagan todas las h_i, salvo la k-ésima, pues $h_i(x_k) = 0$ para todos los i, salvo cuando $i = k$, que vale 1. Es decir, que en toda la sumatoria, solamente sobrevive el término $y_k h_k(x_k) = y_k$.

Se deja como ejercicio constatar que $H'_{2n+1}(x) = \sum_{i=0}^{i=n}(y_i h'_i(x) + y'_i \tilde{h}'_i(x))$ y verificar de forma análoga, qué ocurre al evaluar en x_k.

Finalmente, es posible demostrar que los $h_i(x)$ tienen la forma:

$$h_i(x) = [1 - 2l'_i(x_i)(x - x_i)](l_i(x))^2 \tag{5.3}$$

y que los $\tilde{h}_i(x)$ tienen la forma:

$$\tilde{h}_i(x) = (x - x_i)(l_i(x))^2$$

en función de los polinomios $l_i(x)$ de la base de Lagrange.

Vamos a deducir 5.3:

1. $h'_i(x_j) = 0 \quad \forall j \Rightarrow h'_i(x) = l_i(x)r(x)/r(x_i) = 0$
2. $h_i(x_j) = \delta_{ij} \Rightarrow h_i(x) = l_i(x)q(x)$

para ciertos $r(x)$ y $q(x)$ que completan el grado de $h'_i(x)$ y $h_i(x)$.

Derivamos $h'_i(x) = l'_i(x)q(x) + l_i(x)q'(x) = l_i(x)r(x)$.

Entonces l_i divide también a q y así $q(x) = l_i(x)s(x)$, para cierto $s(x)$.

Ahora, remplazando tenemos que $h_i(x) = l_i^2(x)s(x)$. Pero el grado de $l_i^2(x)$ es $2n - 2$, por lo que $s(x)$ debe ser un polinomio de grado 1 y así:

$h_i(x) = l_i^2(x)(ax + b)$

Para hallar a y b imponemos:

1. $h'_i(x_j) = 0 \Rightarrow 2l_i(x_i)l'_i(x_i)(ax_i + b) + l_i^2(x_i)a = 0$
2. $h_i(x_i) = 1 \Rightarrow l_i^2(x_i)(ax_i + b) = ax_i + b = 1$

que resulta en un sistema lineal de dos ecuaciones con dos incógnitas, con solución:

$$\begin{cases} a = -2l'_i(x_i) \\ b = 1 + 2l'_i(x_i)x_i \end{cases}$$

Sustituyendo se llega a 5.3.

Ejemplo 5.5.1. A modo de ejemplo, expresemos el polinomio interpolante de Hermite de la función $f(x) = sen(x)$ por los puntos $x_0 = 0$ y $x_1 = \pi/4$.

La tabla de datos es la siguiente:

x_i	y_i	y'_i
0	0	1
$\pi/4$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$

Los polinomios base de Lagrange son:

$$l_0(x) = \left(1 - \frac{\pi}{4}x\right)$$

$$l_1(x) = \frac{4}{\pi}x$$

Si recordamos la expresión dada anterior para el polinomio interpolante, se tiene en este caso:

$$H_3(x) = 0.h_0(x) + 1.\tilde{h}_0(x) + \frac{\sqrt{2}}{2}.h_1(x) + \frac{\sqrt{2}}{2}.\tilde{h}_1(x)$$

Ahora:

$$\tilde{h}_0(x) = x\left(1 - \frac{4}{\pi}x\right)^2$$

$$h_1(x) = \left[1 - \frac{8}{\pi}\left(x - \frac{\pi}{4}\right)\right]\left(\frac{4}{\pi}x\right)^2$$

$$\tilde{h}_1(x) = \left(x - \frac{\pi}{4}\right)\left(\frac{4}{\pi}x\right)^2$$

Por lo tanto el polinomio interpolante es:

$$H_3(x) = x\left(1 - \frac{4}{\pi}x\right)^2 + \frac{\sqrt{2}}{2}\left[1 - \frac{8}{\pi}\left(x - \frac{\pi}{4}\right)\right]\left(\frac{4}{\pi}x\right)^2 + \frac{\sqrt{2}}{2}\left(x - \frac{\pi}{4}\right)\left(\frac{4}{\pi}x\right)^2$$

△

El polinomio de Hermite está íntimamente relacionado con el polinomio de Newton en que ambos permiten ser derivados y calculados mediante el método de diferencias divididas. En este caso, debemos tomar los coeficientes de la diagonal de la tabla de diferencias divididas y multiplicarlos por el k-ésimo polinomio de la base de Hermite que tiene la siguiente expresión $\prod_{i=0}^{n}(x - x_i)$, así como lo hacíamos cuando generamos el polinomio de Newton.

Visualicemos el procedimiento mediante un ejemplo:

Ejemplo 5.5.2. Continuando con el caso de la interpolación de $f(x) = sen(x)$ mediante $H_3(x)$, ilustraremos cómo se puede obtener también el polinomio de Hermite utilizando el esquema de diferencias divididas. En ese caso:

0	0			
0	0	1		
$\pi/4$	$\frac{\sqrt{2}}{2}$	$\frac{2\sqrt{2}}{\pi}$	$\frac{4\left(2\sqrt{2}-\pi\right)}{\pi^2} \approx -0{,}1269$	
$\pi/4$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{2\sqrt{2}(\pi-4)}{\pi^2}$	$\frac{\pi\left(2\sqrt{2}+4\right)-16\sqrt{2}}{\pi^2}\frac{4}{\pi} \approx -0{,}1516$

Por lo tanto el polinomio interpolante de Hermite de la función $f(x) = sen(x)$ por las abscisas $x_0 = 0$ y $x_1 = \pi/4$ es:

$$H_3(x) = x + \frac{4\left(2\sqrt{2}-\pi\right)}{\pi^2}x^2 + \frac{\pi\left(2\sqrt{2}+4\right)-16\sqrt{2}}{\pi^2}\frac{4}{\pi}x^2(x-\pi/4)$$

$$H_3(x) = x - 0{,}1269x^2 - 0{,}1516x^2(x-\pi/4)$$

△

A continuación enunciaremos sin demostrar que el error de interpolación en el intervalo $[x_0, x_n]$ tiene la forma:

$$f(x) - H_{2n+1}(x) = \frac{f^{2n+2}(\xi(x))}{(2n+2)!}(x-x_0)^2(x-x_1)^2\ldots(x-x_n)^2,\ \xi(x) \in [x_0, x_n]$$

donde asumimos que los puntos se encuentran ordenados $x_0 \leq x_1 \leq \cdots \leq x_n$ por simplicidad.

La idea de la demostración consiste en llevar el problema a otro de interpolación clásica de puntos. Se tiene por un lado los $n+1$ puntos de interpolación $(x_i, y_i), i = 0, .., n$, y se introduce una sucesión de puntos artificiales (x_i^n, y_i^n) de manera que la pendiente generada por los segmentos $(x_i, y_i), (x_i^n, y_i^n)$ coincide con y_i' y cada x_i^n converge a x_i.

En el resultado es entonces idéntico al de interpolación polinómica, donde ahora figuran cuadrados (pues los puntos son "dobles").

Ejemplo 5.5.3. Es así que una cota superior posible para el error entre $f(x) = sen(x)$ y $H_3(x)$ en el intervalo $[0, \frac{\pi}{4}]$ es $E_{max} \leq \frac{\|f^{(4)}\|_\infty}{4!}(\pi/4)^2(\pi/4)^2$. En nuestro caso $f^{(4)}(x) = sen(x)$, por lo tanto $\max_{[0,\pi/4]} |f^{(4)}(x)| = \max_{[0,\pi/4]} sen(x) = sen(\pi/4) = \frac{\sqrt{2}}{2}$. Luego, una cota superior para el error (que no es rígida) es:

$$c = \frac{\sqrt{2}}{2}\frac{1}{4!}\left(\frac{\pi}{4}\right)^4 \simeq 1{,}121x10^{-2}$$

△

Finalmente, para cerrar esta sección, veremos que al igual que en el método de Newton, podemos construir el polinomio interpolante de Hermite mediante un proceso iterativo en el que en cada paso se lo obliga a pasar por un nuevo punto con cierto valor de derivada.

Si denotamos por $P_n(x)$ el polinomio de Hermite que interpola (x_i, y_i, y'_i) para $i = 0, \ldots, n$ y tenemos un punto adicional $(x_{n+1}, y_{n+1}, y'_{n+1})$, entonces

$$P_{n+1}(x) = P_n(x) + [a + b(x - x_{n+1})] \prod_{i=0}^{n} (x - x_i)^2$$

Imponiendo la condición $P_{n+1}(x_{n+1}) = y_{n+1}$ se despeja el valor de a y luego usando que la derivada de $\prod_{i=0}^{n}(x-x_i)^2$ es $\left(\sum_{i=0}^{n} \frac{2}{x-x_i}\right) \prod_{i=0}^{n}(x-x_i)^2$, se puede obtener de $P'_{n+1}(x_{n+1}) = y'_{n+1}$ el valor de b.

Observar que $p_0(x) = y_0 + y'_0(x - x_0)$.

5.6. Interpolación Lineal

La forma más básica de atacar el problema de interpolar un set de puntos consiste en algo que puede realizar casi cualquier persona: unir los puntos.

Matemáticamente, se debe expresar entre cada par de puntos la ecuación de una recta que pase por ellos:

$$y = L^{(i)}(x) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i) + y_i$$

Esta expresión es solamente válida en el intervalo $[x_i, x_{i+1}]$, habiendo distintas "fórmulas" para cada tramo, por lo que este método recibe el nombre de interpolación lineal a tramos.

Repasemos las ventajas de este tipo de interpolación:

- Es simple.
- Al agregar más puntos mejora la aproximación.
- Agregar puntos es fácil (por ser un método local).

Sin embargo, su principal desventaja es que, a diferencia de las anteriores formas de interpolación polinómica, no es derivable en los x_i. En las próximas secciones mostraremos otros tipos de interpolación a trozos para los que se obtienen curvas más suaves que subsanan la no derivabilidad de este método, pero antes, observemos que el error es:

$|L(x) - f(x)|_{[x_i, x_{i+1}]} = |(x - x_i)(x - x_{i+1})\frac{f''(\theta)}{2!}| \geq |h^2 \frac{f''(\theta)}{2!}|$ si suponemos $x_{i+1} - x_i = h$.

Es decir que si la distancia entre los x_i es h, el error es $O(h^2)$.

Se deja como ejercicio al lector interesado mostrar que el error entre las pendientes para $x \neq x_i$ es $|L'(x) - f'(x)| = O(h)$.

5.7. Splines cúbicos

La interpolación por medio de Splines resuelve el problema de encontrar una función de clase C^2 que interpole un conjunto de datos.

Para tal fin, realizaremos una interpolación a trozos utilizando en cada intervalo $I_i = [x_i, i_{i+1}]$ $\forall i = 0, \ldots, n-1$ un polinomio cúbico $s_i(x)$.

A estos polinomios se les impondrá no sólo por dónde deben pasar en sus extremos (como el caso anterior), sino además con una pendiente y una concavidad tal que se solapen con la pendiente y concavidad del próximo polinomio con el objetivo de obtener una curva dos veces derivable en todos los puntos.

Le llamaremos Spline a un polinomio interpolante cúbico a trozos con derivadas primeras y segundas continuas.

Por tanto, dos splines adyacentes quedan relacionados por las siguientes ecuaciones:

- $s_{i-1}(x_{i-1}) = y_{i-1} \quad 1 \leq i \leq n$
- $s_{i-1}(x_i) = y_i \quad 1 \leq i \leq n$
- $s'_{i-1}(x_i) = s'_i(x_i) \quad 1 \leq i \leq n$
- $s''_{i-1}(x_i) = s''_i(x_i) \quad 1 \leq i \leq n$

Se comenta que puede verse cada spline como un polinomio cúbico a trozos de Hermite.

Es así que entre cada par de puntos $[x_i, x_{i+1}]$ tengo un polinomio cúbico con 4 parámetros. Por ende tengo en total $4n$ incógnitas.

Por su parte, contando la cantidad de restricciones impuestas por cada item tenemos: n por pasar por el extremo izquierdo del intervalo, n por pasar por el extremo derecho del intervalo, $n-1$ para tener derivada continua, y $n-1$ para tener concavidad continua, totalizando $4n-2$.

Obsérvese que tenemos más incógnitas que ecuaciones, nos faltarían dos restricciones más que corresponden a las derivadas en los extremos $\{x_0, x_n\}$.

Las constantes y parámetros anteriores se pueden escribir a partir de los datos como:

- $h_i = x_{i+1} - x_i$
- $t = \frac{x - x_i}{h_i}$
- $\Delta_i = y_{i+1} - y_i$
- $d_i = s'_i(x_i)$

y además

$$s_i(x_i + th_i) = y_i + t\Delta_i + t(t-1)(\Delta_i - h_i d_i) + t^2(t-1)(h_i(d_i + d_{i+1}) - 2\Delta_i)$$

Veamos que pese a lo intrincado de la formulación, $s_i(x_i + th_i)$ cumple todo lo que debe:

1. $s_i(x_i) = y_i$ se constata sustituyendo $t = 0$.
2. $s_i(x_{i+1}) = y_{i+1}$ se verifica haciendo $t = 1$ resultando $s_i(x_i+h_i) = s_i(x_{i+1}) = y_i+\Delta_i = y_{i+1}$. Ahora derivamos respecto a t:

$$s_i'(x_i + th_i)h_i = \Delta_i + (2t-1)(\Delta_i - h_i d_i) + (3t^2 - 2t)(h_i(d_i + d_{i+1}) - 2\Delta_i)$$

3. $s_i'(x_i)h_i = \Delta_i - (\Delta_i - h_i d_i) = h_i d_i$
4. $s_i'(x_{i+1})h_i = \Delta_i + (\Delta_i - h_i d_i) + (h_i(d_i + d_{i+1}) - 2\Delta_i) = d_{i+1}$

Bien, ahora calculemos la derivada segunda:

$$s_i''(x_i + th_i)h_i^2 = 2(\Delta_i - h_i d_i) + (6t-2)(h_i(d_i + d_{i+1}) - 2\Delta_i)$$

e imponemos $s_i''(x_i) = s_{i+1}''(x_{i+1})$

$$\Rightarrow \frac{1}{h_i^2}[2\Delta_i - 2h_i d_i + 4h_i(d_i + d_{i+1}) - 8\Delta_i] = \frac{1}{h_{i+1}^2}[2\Delta_{i+1} - 2h_{i+1}d_{i+1} - 2h_{i+1}(d_{i+1} + d_{i+2}) + 4\Delta_{i+1}]$$

$$\Rightarrow \frac{1}{h_i^2}[-6\Delta_i + 2h_i d_i + 4h_i d_{i+1}] = \frac{1}{h_{i+1}^2}[6\Delta_{i+1} - 4h_{i+1}d_{i+1} - 2h_{i+1}d_{i+2}]$$

De donde finalmente se obtiene:

$$\frac{2}{h_i^2}d_i + (\frac{4}{h_i} + \frac{4}{h_{i+1}})d_{i+1} + \frac{2}{h_{i+1}}d_{i+2} = \frac{6}{h_{i+1}^2}\Delta_{i+1} + \frac{6}{h_i^2}\Delta_i$$

que es un sistema de ecuaciones lineales que permite calcular explícitamente los d_i.

Obsérvese que con los d_i es posible calcular los 4 coeficientes del Spline ya que tenemos 4 ecuaciones que lo determinan:

$$\begin{cases} s_i(x_i) = y_i \\ s_i(x_{i+1}) = y_{i+1} \\ s_i'(x_i) = d_i \\ s_i'(x_{i+1}) = d_{i+1} \end{cases}$$

Si especificamos d_0 y d_n se llama Spline completos. Si imponemos que $d_0 = d_n = 0$ se llama Spline natural.

Un último comentario sobre este tema es que la matriz resultante asociada al sistema con incógnitas d_i es un sistema tridiagonal para el cual existen técnicas eficientes de resolución.

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ \frac{2}{h_i} & [\frac{4}{h_i} + \frac{4}{h_{i+1}}] & \frac{2}{h_{i+1}} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix} \tag{5.4}$$

5.8. Curvas de Bézier*

Una curva de Bézier es una forma de interpolar puntos. Sin embargo, en este caso, dado un conjunto de puntos $P_0 = (x_0, y_0), \dots, P_n = (x_n, y_n)$, la curva de Bézier sólamente unirá los extremos de la serie, es decir P_0 con P_n, a los que se les llama "puntos de anclaje". Los otros puntos (por los cuales en general no pasará la curva) se denominan "puntos de control" y funcionan como controladores de la forma que tomará la curva, es decir, proporcionan información sobre la dirección y movimiento que trazará la curva.

Curva lineal de Bézier:

Dados los puntos P_0 y P_1, una curva lineal de Bézier es una recta entre los dos puntos:

$B(x) = (1-x)P_0 + xP_1 \quad \forall x \in [0,1]$.

Curva cuadrática de Bézier:

Dados los puntos P_0, P_1 y P_2, una curva cuadrática de Bézier viene dada por la siguiente función:

$B(x) = (1-x)^2 P_0 + 2x(1-x)P_1 + x^2 P_2 \quad \forall x \in [0,1]$.

En general, dados los puntos $P_0, P_1, \dots, P_n$, una curva de Bézier de grado n es:

$B(x) = \sum_{i=0}^{n} \binom{n}{i} (1-x)^{n-i} x^i P_i \quad \forall x \in [0,1]$.

Propiedades:

- El polígono formado por los puntos $P_0, P_1, \dots, P_n$ se denomina polígono de Bézier.
- La curva de Bézier se encuentra en el interior de la envolvente convexa del polígono de Bézier.
- El comienzo de la curva de Bézier es tangente al primer segmento del polígono de Bézier P_0P_1, lo mismo ocurre con el final de la curva y el último segmento del polígono.
- La curva de Bézier es C^∞.

Las curvas de Bézier encuentran su aplicación en sistemas CAD, dada su intuitiva interactividad con el usuario.

Capítulo 6

Ecuaciones Diferenciales

Muchas de las leyes de la naturaleza, así como múltiples problemas y aplicaciones de la ciencia, ingeniería, economía, pueden expresarse mediante ecuaciones diferenciales.

Definición 6.0.1. Una *ecuación diferencial* es una ecuación que relaciona las derivadas de una o más variables dependientes respecto a una o más variables independientes.

Definición 6.0.2. Una *ecuación diferencial ordinaria (EDO)* es una ecuación diferencial que relaciona una función desconocida de una única variable independiente con sus derivadas.

Por ejemplo, $y(x)$ es una función que depende de sólamente de la variable independiente x. Por tanto, $y' = x^2y + 2y$ es una EDO.

Definición 6.0.3. Se llama *orden* de una ecuación diferencial al orden de la derivada más alta que aparecen en la ecuación.

Ejemplo 6.0.1. La ecuación $y'' + e^x y y'^3 = \sin(x)$ tiene orden 2. △

Definición 6.0.4. Dada una función $f : \mathbb{R}^2 \to \mathbb{R}$, el *Problema de Valores Iniciales* consiste en hallar la función $y = y(x)$ tal que:

$$(PVI): \begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \in \mathbb{R}. \end{cases}$$

Es decir que es una EDO tal que su solución cumple cumple en el punto x_0 una condición suplementaria.

Mencionamos además a modo de ejemplo que en la EDO $y' = x^2y + 2y$, $f(x, y(x)) = x^2y + 2y$.

Para muchas ecuaciones diferenciales es posible determinar la solución exacta como se ve en otros cursos. Para muchas otras, esto no es posible y mediante métodos numéricos encontraremos soluciones aproximadas.

Es así entonces que nuestro objetivo es obtener una grilla $\{x_k\}$ y valores $\{y_k\}$ tales que y_k sea una aproximación de $y(x_k)$, siendo $y(x)$ la solución exacta de la ecuación diferencial.

En general, si se desea obtener una aproximación de $y(x)$ en un intervalo $[a, b]$ para x, entonces se toman valores equiespaciaciados para x. La expresión queda $x_k = a + \frac{k|b-a|}{N}$ para un cierto

N, con $k \in \{0, \ldots, N\}$. Al valor $h = \frac{|b-a|}{N}$ se le llama paso. Si los pasos no son equiespaciados, debemos explícitamente hacer referencia al paso k-ésimo: $h_k = x_{k+1} - x_k$.

Observación 6.0.1. Ecuaciones diferenciales de mayor orden se pueden expresar como ecuaciones diferenciales (vectoriales) de orden 1, por lo que bastaría desarrollar métodos de resolución para estas últimas.

Por ejemplo, consideremos una ecuación diferencial de orden 3 que puede escribirse de forma genérica como:

$h(y''', y'', y', y, x) = 0$

o bien

$y''' = g(y'', y, y, x)$

Entonces realizando los siguientes cambios de variables tenemos:

$$\begin{cases} z_1(x) = y \\ z_2(x) = y' \\ z_3(x) = y'' \end{cases} \Rightarrow \begin{cases} z_1'(x) = z_2(x) \\ z_2'(x) = z_3(x) \\ z_3'(x) = g(y'', y, y, x) \end{cases}$$

Ahora, tomando el vector $\vec{z} = (z_1, z_2, z_3)$, podemos reescribir la ecuación diferencial de orden 3, como una ecuación diferencial vectorial de orden 1:

$$\frac{d\vec{z}}{dx} = \vec{f}(z_1, z_2, z_3, x)$$

donde $\vec{f}(x) = \begin{pmatrix} z_2(x) \\ z_3(x) \\ g(z_3, z_2, z_1, x) \end{pmatrix}$.

6.1. Método de Euler hacia adelante

Sea el (PVI): $\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \in \mathbb{R}. \end{cases}$

En el método de Euler, o método de la tangente, se aproxima la derivada en el punto x_k mediante el cociente entre la diferencia de dos pasos consecutivos y_{k+1} e y_k y el paso h.

Se obtiene la *ecuación en diferencias*

$$\frac{y_{k+1} - y_k}{h} \approx y'(x_k) = f(x_k, y(x_k)) \approx f(x_k, y_k)$$

o sea $y_{k+1} = y_k + hf(x_k, y_k)$.

Definición 6.1.1. Una *ecuación en diferencias* es una relación que deben cumplir idénticamente los términos de una sucesión.

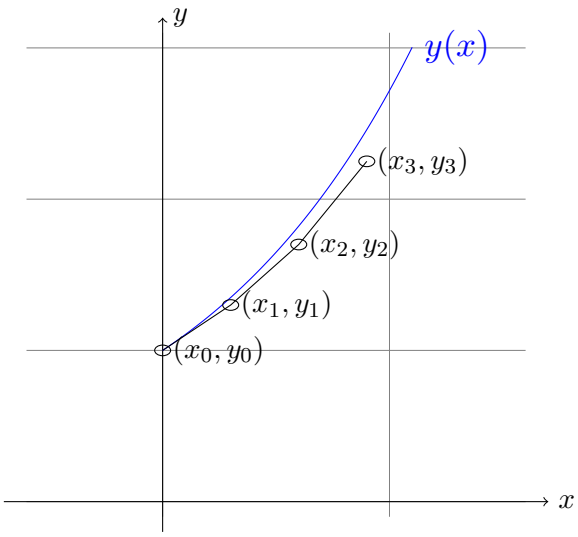

Figura 6.1: Método de Euler

Por tanto, el método de Euler hacia adelante consiste en la siguiente iteración:

$$(\textit{Euler hacia adelante}): \begin{cases} y_{k+1}^E = y_k^E + hf(x_k, y_k^E) \\ y_0^E = y_0 \end{cases}$$

donde en el caso de pasos regulares se toma $x_k = x_0 + kh$ con $h > 0$ fijo.

Otra forma de deducirlo es mediante integración en el intervalo $[x_k, x_{k+1}]$, se tiene la siguiente identidad:

$$y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} f(x, y(x))dx$$

El método de Euler hacia adelante se obtiene de estimar la segunda integral con un rectángulo con altura $f(x_k, y_k)$. Nótese que se conoce y_0:

$$y_{k+1} - y_k = hf(x_k, y_k)$$

Una tercera forma de verlo, consiste en considerar el desarrollo de Taylor:

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2}y''(\xi), \quad \xi \in (x, x+h)$$

Sustituyendo la derivada dada por la EDO en el desarrollo, se tiene

$$y(x+h) = y(x) + hf(x, y) + O(h^2)$$

Despreciando el término de segundo orden, evaluando en x_k, tenemos

$$y(x_{k+1}) = y(x_k) + hf(x_k, y(x_k))$$

Finalmente, aproximando $y(x_k)$ por y_k nuevamente obtenemos la iteración del método de Euler: $y_{k+1} = y_k + hf(x_k, y_k)$.

Ejemplo 6.1.1. Sea la EDO

$$\begin{cases} y' = xy^2 + y \quad x \in [0,1] \\ y(0) = 1 \end{cases}$$

Tomando $N = 10$, entonces $x_k = 0 + \frac{k(1-0)}{10} = \frac{k}{10}$, con $k \in \{0 \dots 10\}$

Sabemos que $f(x,y) = xy^2 + y$, entonces calculamos:

- $y_0 = 1$
- $y_1 = y_0 + \frac{1}{10} f(x_0, y_0) = 1 + \frac{1}{10} f(0,1) = \frac{11}{10}$
- $y_2 = y_1 + \frac{1}{10} f(x_1, y_1) = \frac{11}{10} + \frac{1}{10} f\left(\frac{1}{10}, \frac{11}{10}\right) = \frac{11}{10} + \frac{1}{10}\left[\frac{1}{10}\left(\frac{11}{10}\right)^2 + \frac{11}{10}\right] = \frac{11}{10} + \frac{11^2}{10^4} + \frac{11}{10^2}$
- $y_3 = y_2 + \frac{1}{10} f(x_2, y_2) = \dots$

△

Ejemplo 6.1.2. Sea $y' = y$, $y(0) = 1$. Sabemos que la solución exacta es $y(x) = e^x$. Vamos a hallar una aproximación de la solución en $[0,1]$. Tomamos $h_k = \frac{1}{n}$. Vamos a hallar $y_1, \dots, y_n$ para $x_1, \dots, x_n$.

- $y_0 = 1$
- $y_1 = 1 + \frac{1}{n} y_0 = 1 + \frac{1}{n}$
- $y_2 = \left(1 + \frac{1}{n}\right) + \frac{1}{n}\left(1 + \frac{1}{n}\right) = \left(1 + \frac{1}{n}\right)^2$ ⋮
- $y_k = \left(1 + \frac{1}{n}\right)^k \to y_n = \left(1 + \frac{1}{n}\right)^n$

Sabemos que y_n debería ser igual a $y(1) = e^1 = e$ y también que $\left(1 + \frac{1}{n}\right)^n \to e$ con n creciente. Se aprecia entonces que cuanto más chico es el h, mejor es la aproximación.

△

El pseudo-código para resolver un PVI usando Euler hacia adelante en $[0, x_f]$ podría ser:

Algoritmo 10 Pseudo-código: Euler hacia adelante

Dados $y(1) = y_0$; $x(1) = 0$; $i = 1$; h; f;
  while $x(i) < x_f$ **do**
    $y(i+1) \leftarrow y(i) + h * f(x(i), y(i))$
    $x(i+1) \leftarrow x(i) + h$
    $i \leftarrow i + 1$
  end while

6.2. Elementos de los métodos numéricos aplicados a EDOs

Definición 6.2.1. Un *método de un paso* consiste en un método donde y_k es el único dato que se usa en el paso en que se calcula y_{k+1}.

Ejemplo 6.2.1. El método de Euler es un método de un paso. △

Si el método requiere más de un dato previo para determinar y_{k+1} diremos que es un método multipaso.

6.2.1. Precisión

Definición 6.2.2. El *error global* E_{k+1} en el punto x_{k+1} es la diferencia entre y_{k+1} y el valor en x_{k+1} de la solución exacta al PVI con $y(x_0) = y_0$, es decir, la que pasa por (x_0, y_0).

Definición 6.2.3. El *error local* e_{k+1} en el punto x_{k+1} es la diferencia entre y_{k+1} y el valor en x_{k+1} de la solución exacta al PVI con $y(x_k) = y_k$, es decir, la que pasa por (x_k, y_k).

Si denotamos por $y(x_{k+1}, x_0, y_0)$ a la solución exacta de $\begin{cases} y' = f(x,y) \\ y(x_0) = y_0 \end{cases}$, entonces:

- *Error local* en el paso $k+1$ es: $e_{k+1}(h) = y(x_{k+1}, x_k, y_k) - y_{k+1}$.
- *Error global* en el paso $k+1$ es: $E_{k+1}(h) = y(x_{k+1}, x_0, y_0) - y_{k+1}$.

Anteriormente, el desarrollo de Taylor de $y(x)$ en x_k evaluado en x_{k+1} nos había brindado la expresión:

$y(x_{k+1}) = y(x_k) + h_k f(x_k, y(x_k)) + \frac{h_k^2}{2} y''(\xi_k), \quad \xi_k \in (x_k, x_{k+1})$

Por otra parte, la ecuación en diferencias para el método de Euler es:

$y_{k+1} = y_k + h_k f(x_k, y_k)$

Restando ambas expresiones se obtiene una expresión para el error global

$$\begin{aligned} E_{k+1} &= y(x_{k+1}) - y_{k+1} \\ &= \underbrace{y(x_k) - y_k + h_k[f(x_k, y(x_k)) - f(x_k, y_k)]}_{\text{Error de Propagación}} + \underbrace{\frac{h_k^2}{2} y''(\xi_k)}_{\text{Error Local}} \end{aligned}$$

El error local sería el único error que estaría cometiendo en y_{k+1} si y_k fuera igual a $y(x_k)$. Observe que si en la expresión anterior y_k fuera igual a $y(x_k)$ sólo permanecería el último término.

El error global viene de calcular términos anteriores, existiendo en cada paso un error local y acumulándose el error de propagación.

Ahora bien, si continuamos analizando el error de propagación:

$$f(x_k, y(x_k)) - f(x_k, y_k) = \frac{\partial f}{\partial y}(x_k, \theta)(y(x_k) - y_k)$$

$$E_{k+1} = \underbrace{\overbrace{[1 + h_k \frac{\partial f}{\partial y}(x_k, \theta)]}^{\text{Factor de amplificación}} (y(x_k) - y_k)}_{\text{Error de Propagación}} + \underbrace{\frac{h_k^2}{2} y''(\xi_k)}_{\text{Error Local}}$$

Por tanto, usando $h_k = h$:

- Si $|1 + h\frac{\partial f}{\partial y}(x_k, \theta)| < 1$ los errores **no** se amplifican.
- Si $|1 + h\frac{\partial f}{\partial y}(x_k, \theta)| > 1$ los errores se amplifican.

Buscamos un paso h tal que $|1 + h\frac{\partial f}{\partial y}| < 1 \Leftrightarrow -1 < 1 + h\frac{\partial f}{\partial y} < 1 \Leftrightarrow$

$\Leftrightarrow -2 < 1 + h\frac{\partial f}{\partial y} < 0$ lo cual conforma un intervalo de estabilidad.

- Si $\frac{\partial f}{\partial y} > 0$ siempre estamos fuera del intervalo de estabilidad.
- Si $\frac{\partial f}{\partial y} < 0$ podemos elegir h para caer dentro del intervalo de estabilidad (si $h_1, \ldots, h_k$ son distintos pedimos que todos cumplan). Buscaríamos $|h_k \frac{\partial f}{\partial y}| < 2$. Si L es una cota superior para $|\frac{\partial f}{\partial y}|$, bastaría tomar $h_k < \frac{2}{L}$.

En el caso vectorial ($y_k \in \mathbb{R}^n,\ x_k \in \mathbb{R}$), para el control del error de propagación, en lugar de $\frac{\partial f}{\partial y}(x_k, \theta)$ usamos $J_y(x_k, \theta)$, donde:

$$J_y = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \cdots & \frac{\partial f_1}{\partial y_m} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \cdots & \frac{\partial f_2}{\partial y_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial y_1} & \frac{\partial f_2}{\partial y_m} & \cdots & \frac{\partial f_m}{\partial y_m} \end{pmatrix}, \quad \theta \in \mathbb{R}^n$$

Por lo que ahora, el error de propagación tiene la siguiente expresión:

$$EP_{k+1} = (\mathrm{I} + hJ_y(x_k, \theta))\, E_k$$

donde:

- Si $\|\mathrm{I} + hJ_y(x_k, \theta)\| < 1$ los errores **no** se amplifican.
- Si $\|\mathrm{I} + hJ_y(x_k, \theta)\| > 1$ los errores se amplifican.

6.2.2. Estudio del error

Hipótesis: Se asume de ahora en adelante que existe una región $D \subset \mathbb{R}^2$ tal que

$$l \leq \frac{\partial f(x,y)}{\partial y} \leq L \quad \forall (x,y) \in D$$

Teorema 6.2.1. *Si los errores locales $e_k < \varepsilon$, entonces el error global cumple*

1. $|E_k(h)| = |y(x_k) - y_k| \leq \varepsilon \frac{e^{Lkh}-1}{e^{Lh}-1}$
2. $|E_k(h)| = |y(x_k) - y_k| \leq \frac{\varepsilon}{h} \frac{e^{Lkh}-1}{L} \leq c\frac{\varepsilon}{h}$
3. *Si $L = 0$, entonces* $|E_k(h)| = |y(x_k) - y_k| \leq k\varepsilon$

Desarrollamos a continuación la demostración del resultado 2. Si suponemos $h_k = h$, sabemos que:

$$E_{k+1} = [1 + h\frac{\partial f}{\partial y}(x_k, \theta)]E_k + \frac{h_k^2}{2}y''(\xi_k)$$

Además, si $\frac{\partial f}{\partial y}(x_k, \theta) < L$ y $e_k \leq \varepsilon \ \forall k$ queda:

$$|E_{k+1}| \leq (1 + hL)|E_k| + \varepsilon \quad \forall k$$

Haremos uso del siguiente lema:

Lema 6.2.2. *Sea w_k una sucesión que cumple $w_{k+1} \leq Aw_k + b,\ A > 0,\ A \neq 1,\ A, b \in \mathbb{R}$. Entonces:*

$$w_k \leq w_0 A^k + b\frac{A^k - 1}{A - 1}$$

Tomando $A = 1 + hL,\ b = \varepsilon$ tenemos

$$|E_k| \leq (1 + hL)^k |E_0| + \varepsilon \frac{(1 - hL)^k - 1}{1 + hL - 1}$$

Como $|E_k| = |y(x_k) - y_k|$ resulta que $|E_0| = 0$ y así

$$|E_k| \leq \varepsilon \frac{(1 - hL)^k - 1}{hL}$$

Por otra parte $e^{hL} = 1 + hL + \frac{(hL)^2}{2} + \ldots$, entonces $1 + hL \leq e^{hL}$.

$$|E_k| \leq \varepsilon \frac{(1 - hL)^k - 1}{hL} \leq \varepsilon \frac{e^{khL} - 1}{hL}$$

A su vez, como estamos resolviendo la ecuación en $[a, b]$, llegamos a que $kh \leq b - a$. Finalmente

$$|E_k| \leq \varepsilon \frac{e^{(b-a)L} - 1}{hL} = c\frac{\varepsilon}{h}$$

con c constante.

Vemos además que el error global E_k es un orden menor (en h) que el error local.

6.2.3. Control del error local

Error local debido al truncamiento

Sabemos que el error local tiene la expresión:

$e_{k+1} = \frac{h_k^2}{2} y''(\xi_k), \quad \xi_k \in (x_k, x_{k+1})$

$y'' \approx \frac{y'(x_{k+1})-y'(x_k)}{x_{k+1}-x_k} = \frac{f(x_{k+1},y(x_{k+1}))-f(x_k,y(x_k))}{x_{k+1}-x_k} \approx \frac{f(x_{k+1},y_{k+1})-f(x_k,y_k)}{x_{k+1}-x_k} = \tilde{y_k}''$

Por lo que llegamos a que:

$$e_{k+1} = \frac{h_k^2}{2} \tilde{y_k}''$$

y si buscamos $e_{k+1} < \varepsilon$ debemos elegir $h_k < \sqrt{\frac{2\varepsilon}{\tilde{y_k}''}}$.

Observación 6.2.1.

- Podemos buscar un h para que el error relativo sea menor que δ: imponiendo $\frac{e_{k+1}}{y_{k+1}} < \delta$, procedemos de igual forma para determinar h.
- En ocasiones, para cubrirnos elegimos $h_k < 0{,}9\sqrt{\frac{2\varepsilon}{\tilde{y_k}''}}$.
- A veces se toma $h_k = \text{mín}\{0{,}9\sqrt{\frac{2\varepsilon}{\tilde{y_k}''}}, \bar{h}\}$, donde $\bar{h}$ es un valor máximo prefijado para el paso.

Interesa trabajar con métodos para los cuales el error global debido al truncamiento es un infinitésimo en h. Recuérdese que por un teorema anterior, el error global por truncamiento se acota por $c\frac{\varepsilon}{h}$. Esto motiva la siguiente definición:

Definición 6.2.4. Se dice que el método es *consistente* si $\lim_{h\to 0} \frac{\text{máx}(e(h))}{h} = 0$

Definición 6.2.5. Se define como *orden de consistencia* al orden del infinitésimo $\frac{e(h)}{h}$.

Ejemplo 6.2.2. En el método de Euler $\frac{\text{máx}(e(h))}{h} = \frac{h^2}{2h}\|y''\|_{\infty,[a,b]} \to 0$, por lo que es consistente de orden 1. △

6.2.4. Estabilidad numérica

La estabilidad numérica refiere a la distancia entre la solución de la ecuación en diferencias y_k y la solución numérica $\bar{y}_k$ en la que aparecen errores de redondeo.

Es decir, la solución $\bar{y}_k$ es la que obtenemos cuando resolvemos la ecuación en diferencias en máquina.

Definición 6.2.6. Un método es *numéricamente estable* si $\bar{E}_k = \bar{y}_k - y_k$ se mantiene acotado al crecer k.

Error local debido al redondeo

Analicemos el error local debido al redondeo en el paso k:

$y_{k+1} = y_k + hf(x_k, y_k)$

$\bar{y}_{k+1} = \bar{y}_k + hf(x_k, \bar{y}_k) + \varepsilon_k$

donde ε_k es el error numérico introducido en cada paso.

Los errores numéricos se mantienen acotados si $|1 + h\frac{\partial f}{\partial y}(\theta)| < 1$

Nótese que esta restricción depende del **método** y del propio **problema** (a través de $\frac{\partial f}{\partial y}$).

Por tanto, salvo en el caso de Euler (y de otros métodos también simples) el estudio de la evolución de $\bar{E_k}$ es muy engorroso. En general, para estudiar este tipo de propagación se estudia qué ocurre en una ecuación diferencial lineal de variable compleja sencilla que se denomina **problema test** y que definiremos luego del siguiente ejemplo.

Ejemplo 6.2.3. $\begin{cases} y' = -y \\ y(0) = \alpha \end{cases}$ tiene solución $y(x) = \alpha e^{-x}$

Ahora, aplicando el método de Euler:

$y_{k+1} = y_k + h(-y_k) = (1-h)y_k$

$y_{k+1} = (1-h)^{k+1} y_0$

$|1 + h\frac{\partial f}{\partial y}| = |1 - h|$

Para que Euler sea numéricamente estable, h debe ser tal que $|1-h| < 1 \Rightarrow 0 < h < 2$. △

Problema Test

Definición 6.2.7. El *Problema Test* es la siguiente EDO:

$$(Problema\ Test): \begin{cases} y' = qy \\ y(0) = 1, \end{cases}$$

siendo q un número complejo arbitrario.

La solución al problema test es $y = e^{qx}$.

Definición 6.2.8. Dado un método iterativo que genera una sucesión $\{y_n\}$ al aplicar el Problema Test, su *región de estabilidad* R es el conjunto de complejos $z = hq$ tales que la sucesión $\{y_n\}$ permanece acotada. Formalmente:

$$R = \{z = hq \in \mathbb{C} : \exists k > 0, |y_n| < k,\ \forall n \in \mathbb{N}\}.$$

Calculemos la región de estabilidad para el método de Euler hacia adelante. Al aplicar Euler hacia adelante al Problema Test tenemos que:

$$y_{k+1} = y_k + h(qy_k) = (1 + hq)y_k = (1 + z)y_k.$$

Por inducción completa se obtiene que $y_k = (1+z)^k y_0 = (1+z)^k$. Esta sucesión permanece acotada si y solamente si $|1+z| \leq 1$. En términos gráficos, el complejo $z = hq$ debe permanecer dentro del disco unidad centrado en el complejo $-1+0i$.

$$R^E = \{hq \in \mathbb{C} : |1+hq| \leq 1\}$$

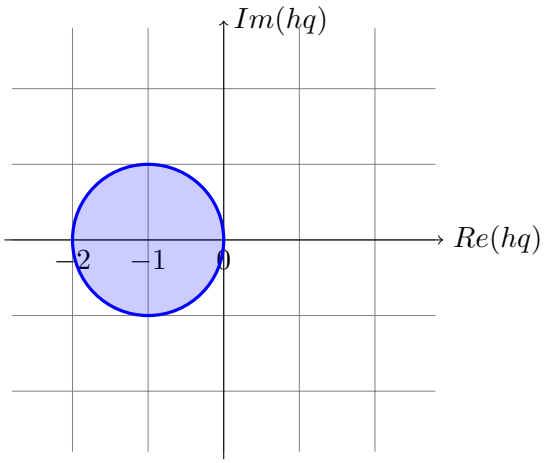

Figura 6.2: Región estabilidad (E)

La región de estabilidad caracteriza al método, ya que resulta de aplicárselo al problema test, y no a ningún problema concreto. En cambio, el valor admisible máximo para h sí depende del problema (a través de q).

Ejemplo 6.2.4. Si $q = -10 \Rightarrow 0 < h < 0,2$. Si $q = -10+10i \Rightarrow 0 < h < 0,1$. △

Observación 6.2.2. El Problema Test evita realizar, en métodos más complejos el estudio de propagación del error $\bar{E}_k = \bar{y_k} - y_k$.

6.2.5. Convergencia

Definición 6.2.9. Diremos que existe *convergencia* en un método cuando se cumple:

- Existencia de consistencia.
- Estabilidad numérica.

6.3. Otros métodos

6.3.1. Método del trapecio

Sea el (PVI): $\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \in \mathbb{R}. \end{cases}$

El método del trapecio se puede deducir mediante integración en el intervalo $[x_k, x_{k+1}]$, análogamente al método de Euler hacia adelante, obteniendo la siguiente identidad:

$$y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} f(x, y(x))dx$$

Sin embargo, en este caso la integral se aproxima con un trapecio con bases $f(x_k, y_k)$ y $f(x_{k+1}, y_{k+1})$ (en vez del rectángulo con altura $f(x_k, y_k)$) llegando a una relación similar:

$$y_{k+1} - y_k = \frac{h}{2}[f(x_k, y_k) + f(x_{k+1}, y_{k+1})$$

Por tanto, el método del Trapecio consiste en la siguiente iteración:

$$(Trapecio): \begin{cases} y_{k+1}^T = y_k^T + \frac{h}{2}[f(x_k, y_k^T) + f(x_{k+1}, y_{k+1}^T)] \\ y_0^T = y_0 \end{cases}$$

siendo $x_k = x_0 + kh$, $x_{k+1} = x_k + h$ en el caso de paso $h > 0$ fijo.

Definición 6.3.1. Un *método implícito* es un método donde y_{k+1} aparece también en el segundo miembro dentro de la función f, en caso contrario, es un *método explicito*.

Ejemplo 6.3.1. El método de Euler es un método explícito mientras que el método del Trapecio es un método implícito. △

Observación 6.3.1.

1. El método del trapecio es implícito, pero si f es lineal en su segunda variable, se puede obtener una expresión explícita para y_{k+1}.
2. Podemos usar Euler para encontrar un valor inicial para y_{k+1} y luego aplicar un esquema iterativo.
3. Veremos que el método del trapecio es de orden mejor que Euler.

Resolución de método implícito

Para poder determinar y_{k+1} en cada paso, una solución es pensar el problema como una ecuación de punto fijo ($x = g(x)$ en la que y_{k+1} toma el rol de x) y aplicar una iteración como las vistas en capítulos anteriores.

$$y_{k+1}^{n+1} = y_k + \frac{h}{2}[f(x_k, y_k) + f(x_{k+1}, y_{k+1}^n)] \tag{6.1}$$

Todos los valores con subíndice k son conocidos del paso anterior.

Un criterio suficiente de convergencia para la iteración anterior es:

$$\frac{h}{2}\left|\frac{\partial f}{\partial y}(x_k, y_k)\right| < 1 \quad \forall k$$

y cuanto menor se dicho valor, más rápida es la convergencia.

Debemos hallar un valor inicial para comenzar iteración, es decir, estimar y_{k+1}^0. Para ello, en ocasiones suele utilizarse un método explícito símple, por ejemplo, usando un paso de Euler.

$$y_{k+1}^0 = y_k + hf(x_k, y_k) \tag{6.2}$$

Esta última fórmula 6.2 se llama *predictor*, y la fórmula 6.1 *corrector*. El procedimiento general se llama método *predictor-corrector*.

La iteración se puede parar controlando la diferencia $y_{k+1}^{n+1} - y_{k+1}^n$ respecto de una tolerancia establecida, o fijando el número de iteraciones. La última idea es la más común. Es deseable elegir un predictor apropiado de forma que baste con una sola iteración (del corrector) para lograr la precisión deseada.

Son muy usadas las fórmulas de *Adams-Bashforth como predictor*, donde el valor inicial se expresa de la forma:

$$y_{k+1}^0 = y_k + \frac{h}{12}[23f(x_k, y_k) - 16f(x_{k-1}, y_{k-1}) + 5f(x_{k-2}, y_{k-2})] \tag{6.3}$$

En ocasiones, debido a la forma de f, no es necesario realizar la iteración asociada al punto fijo, por ejemplo, si f es una ecuación lineal en y.

Ejemplo 6.3.2. Apliquemos el método del trapecio a $\begin{cases} y' = xy \\ y(0) = 1 \end{cases}$

$$y_{k+1} = y_k + \frac{h}{2}\left(x_k y_k + x_{k+1} y_{k+1}\right)$$

Es posible despejar y_{k+1} y no es necesario aplicar predictor-corrector.

$$y_{k+1}\left(1 - \frac{h}{2}x_{k+1}\right) = y_k + \frac{h}{2}x_k y_k$$

$$y_{k+1} = \left(\frac{1 + \frac{h}{2}x_k}{1 - \frac{h}{2}x_{k+1}}\right) y_k$$

Esto se debe a que f es lineal. △

Veamos ahora un caso en que f es no lineal.

Ejemplo 6.3.3. Sea $y' = f(x, y) = ye^{xy} + y^2$

Aplicando el método del trapecio:

$$y_{k+1} = y_k + \frac{h}{2}\left(y_k e^{x_k y_k} + y_k^2 + y_{k+1} e^{x_{k+1} y_{k+1}} + y_{k+1}^2\right)$$

de la cual no es posible despejar y_{k+1}.

Planteado como problema de punto fijo tenemos la siguiente expresión:

$$y_{k+1}^{n+1} = y_k + \frac{h}{2}\left(y_k e^{x_k y_k} + y_k^2 + y_{k+1}^n e^{x_{k+1}^n y_{k+1}^n} + (y_{k+1}^n)^2\right)$$

△

Orden de consistencia (T)

$$y(x_{k+1}) = y(x_k) + hy'(x_k) + \frac{h^2}{2}y''(x_k) + O(h^3)$$

$$y''(x_k) = \frac{y'(x_{k+1}) - y'(x_k)}{h} + O(h)$$

$$\Rightarrow y(x_{k+1}) = y(x_k) + \frac{h}{2}[f(x_k, y(x_k) + f(x_{k+1}, y(x_{k+1}))] + O(h^3)$$

Para analizar el error local consideramos $y_k = y(x_k)$.

$$y(x_{k+1}) = y_k + \frac{h}{2}[f(x_k, y(x_k) + f(x_{k+1}, y(x_{k+1}))] + O(h^3)$$

Por otra parte, el método del Trapecio es:

$$y_{k+1} = y_k + \frac{h}{2}[f(x_k, y_k) + f(x_{k+1}, y_{k+1})]$$

La diferencia entre estas dos últimas expresiones es un término de orden 3 ($O(h^3)$), y el error local es entonces $O(h^3)$. Esto quiere decir que el método del trapecio es consistente de orden 2.

Región de estabilidad (T)

En el caso del Trapecio, para calcular la región de estabilidad, se tiene que $y_{k+1} = y_k + \frac{h}{2}(qy_k + qy_{k+1})$, por lo que $(1 - qh/2)y_{k+1} = (1 + qh/2)y_k$, y por inducción (usando que $y_0 = 1$) tenemos que $y_k = \left(\frac{1+qh/2}{1-qh/2}\right)^k$. La acotación requiere que la base debe ser en magnitud igual a 1 o menor, por lo que debe cumplirse que $|1 + qh/2| \leq |1 - qh/2|$. El conjunto de complejos hq que verifican tal condición cumplen que la distancia al complejo -1 debe ser menor o igual que la distancia al complejo 1. Esto ocurre precisamente cuando qh se halla en el semiplano con parte real negativa, o el eje imaginario.

$$R^T = \{hq \in \mathbb{C} : Re\{hq\} \leq 0\}$$

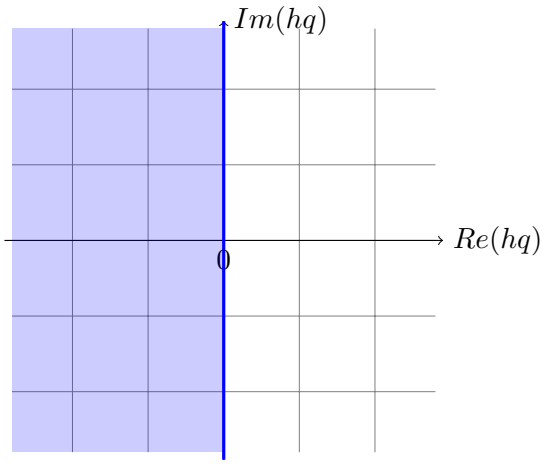

Figura 6.3: Región estabilidad (T)

6.3.2. Método de Euler hacia atrás

Sea el (PVI): $\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \in \mathbb{R}. \end{cases}$

En el método de Euler hacia atrás se puede obtener aproximando $y'(x_{k+1}) \approx f(x_{k+1}, y_{k+1}) \approx \frac{y_{k+1}-y_k}{h}$.

Luego, imponiendo la igualdad $fracy_{k+1} - y_k h = f(x_{k+1}, y_{k+1})$ tenemos $y_{k+1} = y_k + hf(x_{k+1}, y_{k+1})$.

Por tanto, el método de Euler hacia adelante consiste en la siguiente iteración:

$$(Euler\ hacia\ atrás): \begin{cases} y_{k+1}^{EA} = y_k^{EA} + hf(x_{k+1}, y_{k+1}^{EA}) \\ y_0^{EA} = y_0 \end{cases}$$

Otra forma de deducirlo es mediante integración en el intervalo $[x_k, x_{k+1}]$, al igual que en el método de Euler hacia adelante, se tiene la siguiente identidad:

$$y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} f(x, y(x))dx$$

Sin embargo, en este caso la integral se aproxima con un rectángulo con altura $f(x_{k+1}, y_{k+1})$ (en vez del rectángulo con altura y_k) llegando a una relación similar:

$$y_{k+1} - y_k = hf(x_{k+1}, y_{k+1})$$

Nuevamente se obtiene un método implícito, pero si f es lineal en su segunda variable, se puede obtener una expresión explícita para y_{k+1}.

Para el caso general, se procede de igual manera que para los métodos implícitos, considerando cada iteración como un problema de punto fijo.

Ejemplo 6.3.4. $y' = f(x,y) = xy$ Aplicando Euler hacia atrás

$$y_{k+1} = y_k + hx_{k+1} + y_{k+1} \Leftrightarrow y_{k+1}(1 - hx_{k+1}) = y_k \Leftrightarrow y_{k+1} = \left(\frac{y_k}{1 - hx_{k+1}}\right)$$

con $x_{k+1} = a - (k+1)h$. △

Ejemplo 6.3.5. f no lineal: $y' = f(x,y) = ye^{xy} + y^2$
Planteado como un problema de punto fijo tenemos

$$y_{k+1} = g(y_{k+1}) = y_k + hy_{k+1}e^{x_{k+1}y_{k+1}} + y_{k+1}^2$$

△

Orden de consistencia (EA)

Este método tiene orden de consistencia 1. Se deja su demostración como ejercicio.

Región de estabilidad (EA)

Para calcular la región de estabilidad para Euler hacia atrás, aplicamos el problema test:

$$y_{k+1} = y_k + hqy_{k+1} \Rightarrow y_{k+1}(1 - hq) = y_k \Rightarrow y_{k+1} = \frac{y_0}{(1 - hq)}^{k+1}$$

Entonces, para que los y_k queden acotados se tiene que cumplir que

$$\frac{1}{|1 - hq|} \leq 1 \Leftrightarrow |1 - hq| \geq 1$$

$$R^{EA} = \{hq \in \mathbb{C} : |1 - hq| \geq 1\}$$

6.3.3. Método del punto medio

Sea el (PVI): $\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \in \mathbb{R}. \end{cases}$

En el método de punto medio se puede obtener aproximando $y'(x_k) \approx f(x_k, y_k) \approx \frac{y_{k+1} - y_{k-1}}{2h}$.
Sin embargo, como el método no es de un paso, se utiliza Euler hacia adelante para el cálculo de y_1, resultando el método de punto medio en la siguiente iteración:

$$(Punto\ medio)\text{: } \begin{cases} y_{k+1}^{PM} = y_{k-1}^{PM} + 2hf(x_k, y_k^{PM}) \\ y_1^{PM} = y_0^{PM} + hf(x_0, y_0^{PM}) \\ y_0^{PM} = y_0 \end{cases}$$

Observación 6.3.2.
- El método de punto medio es de 2 pasos.

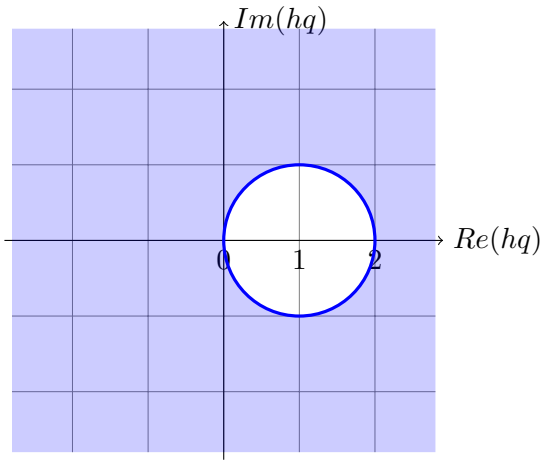

Figura 6.4: Región estabilidad (EA)

- Dada la siguiente fórmula:

$$y'(x_k) = \frac{y(x_{k+1}) - y(x_{k-1})}{2h} + O(h^2)$$

se deja como tarea al lector observar que el truncamiento es $O(h^3)$ por lo que el error global es $O(h^2)$.

Región de estabilidad (PM)

Aplicando el método al problema test:

$$y_{k+1} - 2hqy_k - y_{k-1} = 0$$

El polinomio característico resulta $\lambda^2 - 2hq\lambda - 1 = 0$, de donde las raíces son

$$\lambda = \frac{2hq \pm \sqrt{4h^2q^2 + 4}}{2} = hq \pm \sqrt{h^2q^2 + 1}$$

Si $\lambda_1 \neq \lambda_2$, la solución de la ecuación en diferencias queda:

$$y_k = c_1\lambda_1^k + c_2\lambda_2^k$$

Ahora, como $(\lambda - \lambda_1)(\lambda - \lambda_2) = \lambda^2 - 2hq\lambda - 1 \Rightarrow \lambda_1\lambda_2 = -1$, entonces $|\lambda_1||\lambda_2| = 1$.

Pero la estabilidad requiere que $|\lambda_i| \leq 1$, por lo que combinando las condiciones anteriores, las raíces deberán ser de la forma: $\lambda = e^{i\varphi}$.

Sustituyendo en el polinomio característico $e^{2i\varphi} - 2hqe^{i\varphi} - 1 = 0$ y despejando, los valores para h quedan determinados por:

$hq = \frac{e^{i\varphi} - e^{-i\varphi}}{2} = i\sin\varphi, \quad \varphi \in \mathbb{R}.$

Sin embargo, en el caso en que $hq = \pm i \Rightarrow \lambda^2 \mp 2i\lambda - 1 = (\lambda \pm i)^2 \Rightarrow \lambda = \mp i$ raíz doble.

La solución de la ecuación en diferencias queda entonces $y_k = (A+Bk)(\pm i)^k$, que no está acotada si $k \to \infty$, por tanto no debemos incluir los extremos del intervalo.

Luego, la región de estabilidad es el intervalo abierto de $-i$ a i, es decir, el método será estable sólo si q es imaginario puro.

$$R^{PM} = \{hq \in \mathbb{C} : Re\{hq\} = 0,\ |Im\{hq\}| < 1\}$$

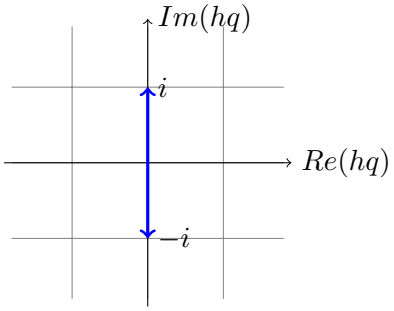

Figura 6.5: Región estabilidad (PM)

6.3.4. Método de Heun

Sea el (PVI): $\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \in \mathbb{R}. \end{cases}$

Cuando en el método del trapecio la predicción se realiza con Euler y no se realizan iteraciones de punto fijo, el procedimiento recibe el nombre de método de Heun:

$$(Heun): \begin{cases} y^H_{k+1} = y^H_k + \frac{h}{2}[f(x_k, y^H_k) + f(x_{k+1}, y^H_k + hf(x_k, y^H_k))] \\ y^H_0 = y_0 \end{cases}$$

Queda como ejercicio demostrar que su orden de convergencia es $O(h^2)$ como se verá en 6.4.1 y encontrar su región de estabilidad.

6.4. Métodos de Runge-Kutta

Los métodos de Runge-Kutta (R-K) son una familia de métodos multipaso de orden alto, que utilizan información en puntos interiores del intervalo $[x_k, x_{k+1}]$. Pueden ser explícitos, semi-explícitos o implícitos.

La mayor parte de los paquetes los incluyen como opción por defecto, lo que los hace extremadamente populares.

Primeramente, se introducirá la idea de los métodos de R-K.

6.4.1. Primer Método R-K

Comenzamos aplicando a un PVI el teorema de valor medio. Planteamos $y(x_k+1) = y(x_k) + hy'(\zeta_k) = y(x_k) + hf(\zeta_k, y(\zeta_k))$ con $\zeta_k = x_k + h\theta_k,\ \theta_k \in [0,1],\ \theta_k \in [x_k, x_{k+1}]$.

Tomemos $\theta_k = \frac{1}{2}$, por lo que $\zeta_k = x_k + h\frac{1}{2}$. ¿Cómo aproximamos $y(x_k + \frac{h}{2})$?

Aplicando el método de Euler con paso $\frac{h}{2}$ para la estimación se tiene que $y(x_k + \frac{h}{2}) \approx y_k + \frac{h}{2}f(x_k, y_k)$.

Por lo anterior, se obtiene el siguiente método explícito al cual llamamos Primer Método de Runge-Kutta:

$$y_{k+1} = y_k + hf(x_k + \frac{h}{2}, y_k + \frac{h}{2}f(x_k, y_k))$$

6.4.2. Segundo Método R-K

Alternativamente, usando la idea de evaluar la derivada en puntos intermedios (pero ahora promediando la derivada) se puede proceder como sigue:

$$y'(x_k + \frac{h}{2}) \approx \frac{1}{2}(y'(x_k) + y'(x_{k+1})) = \frac{f(x_k, y_k) + f(x_{k+1}, y_{k+1})}{2}$$

Tomando: $y(x_k) \approx y_k$ y aproximando $y(x_{k+1})$ por Euler, $y_{k+1} \approx y_k + hf(x_k, y_k)$. Así:

$$y'(x_k + \frac{h}{2}) \approx \frac{f(x_k, y_k) + f(x_{k+1}, y_k + hf(x_k, y_k))}{2}$$

Resultando el Segundo Método de Runge-Kutta explícito:

$$y_{k+1} = y_k + \frac{h}{2}[f(x_k, y_k) + f(x_{k+1}, y_k + hf(x_k, y_k)]$$

La generalización para los métodos R-K de orden 2 es:

$$(Runge - Kutta): \begin{cases} y_{k+1} = y_k + ak_1 + bk_2 \\ k_1 = hf(x_k, y_k) \\ k_2 = hf(x_k + \alpha h, y_k + \beta k_1) \end{cases} \tag{6.4}$$

Algunos de los métodos ya vistos corresponden a esta formulación genérica.

Ejercicio 6.4.1. Verificar que estos métodos pertenecen a la familia R-K con los siguientes parámetros:

- Euler: $a = 1,\ b = 0,\ \alpha = -,\ \beta = -$.
- Predictor-Corrector (Heun): $a = \frac{1}{2},\ b = \frac{1}{2},\ \alpha = 1,\ \beta = 1$.
- Punto medio: $a = 0,\ b = 1,\ \alpha = \frac{1}{2},\ \beta = \frac{1}{2}$.

6.4.3. Elección de parámetros según orden

Vamos a demostrar a continuación que con (a, b, α, β) adecuados, el orden de los métodos dados por 6.4 es 2.

Consideremos las siguientes ecuaciones:

$$y(x_{k+1}) = y(x_k) + hf(x_k, y(x_k)) + \frac{h^2}{2}y''(x_k) + O(h^3) \tag{6.5}$$

$$y'(x_k) = f(x_k, y(x_k)) \Rightarrow y'' = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial x} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}f \tag{6.6}$$

Por lo que sustituyendo 6.6 en 6.5 obtenemos:

$$y(x_{k+1}) = y(x_k) + hf(x_k, y(x_k)) + \frac{h^2}{2}\left[\frac{\partial f}{\partial x}(x_k, y(x_k)) + \frac{\partial f}{\partial y}(x_k, y(x_k))f(x_k, y(x_k))\right] + O(h^3)$$

Ahora, el método de R-K es:

$$y_{k+1} = y_k + ahf(x_k, y_k) + bhf(x_k + \alpha h, y_k + \beta k_1) \tag{6.7}$$

al que desarrollando por Taylor el último factor:

$$f(x_k + \alpha h, y_k + \beta k_1) = f(x_k, y_k) + \frac{\partial f}{\partial x}(x_k, y_k)\alpha h + \frac{\partial f}{\partial y}(x_k, y_k)\beta k_1 + O(h^2)$$

y sustituyendo en 6.7 llegamos a:

$$y_{k+1} = y_k + h\left[af(x_k, y_k) + bf(x_k, y_k)\right] + h^2\left[\alpha b\frac{\partial f}{\partial x}(x_k, y_k) + \beta b\frac{\partial f}{\partial y}(x_k, y_k)f(x_k, y_k)\right] + O(h^3)$$

Ahora, para encontrar el error local suponemos $y_k = y(x_k)$:

$$\begin{cases} y(x_{k+1}) = y_k + hf(x_k, y_k) + \frac{h^2}{2}\left[\frac{\partial f}{\partial x}(x_k, y_k) + \frac{\partial f}{\partial y}(x_k, y_k)f(x_k, y_k)\right] + O(h^3) \\ y_{k+1} = y_k + h\left[af(x_k, y_k) + bf(x_k, y_k)\right] + h^2\left[\alpha b\frac{\partial f}{\partial x}(x_k, y_k) + \beta b\frac{\partial f}{\partial y}(x_k, y_k)f(x_k, y_k)\right] + O(h^3) \end{cases}$$

Finalmente, para que ambas aproximaciones coincidan y el error local sea $O(h^3)$:

$$\begin{cases} a + b = 1 \\ \alpha b = \frac{1}{2} \\ \beta b = \frac{1}{2} \end{cases} \tag{6.8}$$

Corolario 6.4.1.

- *Heun es* $O(h^2)$.
- *P.M. es* $O(h^2)$.

6.4.4. Fórmula general para los métodos de Runge-Kutta explícitos

$$(Runge-Kutta): \begin{cases} y_{k+1} = y_k + \sum_{i=1}^{v} w_i k_i \\ k_i = hf(x_k + c_i h, \sum_{j=1}^{v-1} a_{ij} h_j) \quad i = 1, \ldots, v \end{cases}$$

Las constantes $\{w_i\}$, $\{c_i\}$, $\{a_{ij}\}$, $i, j = 1, \ldots, v$ se eligen adecuadamente de forma de tener una buena aproximación a la solución real.

En general los parámetros del modelo se calculan imponiendo que el error de truncamiento sea del orden más alto posible.

Ejemplo 6.4.1. Euler hacia adelante pertenece a esta familia de métodos. Basta tomar $v = 1,\ w_1 = 1,\ c_1 = 1.$ △

6.5. Problemas con condiciones de borde*

Sea la EDO $y'' = f(x, y, y')$ con *condiciones de borde* $y(a) = \alpha$, $y(b) = \beta$.

No hay teoría que asegure la existencia de solución; sin embargo se plantearán algunos esquemas que la hallarían si existiese.

Asumiremos que $f \in C^\infty$ en $[a, b]$.

Las ideas que se verán a continuación pueden usarse con otro tipo de condiciones de borde más complejas, por ejemplo, $p_0 y(b) + p_1 y'(b) = p_2$.

6.5.1. Método de los disparos

Supongamos que $y'(a) = t$ y resolvamos el PVI $y'' = f(x, y, y')$ con $y(a) = \alpha$. Obtendremos que el valor de y evaluado en b será un valor que dependerá de t, es decir $y(b, t)$, y queremos que dicho valor sea igual a β. O sea, el objetivo pasa a ser encontrar t tal que la función $y(b, t) = g(t) = \beta$, o expresado de otro modo $h(t) = y(b, t) - \beta = 0$, donde h es en general una función no lineal.

Quiere decir que el problema de condiciones de borde se puede plantear como la resolución de la ecuación $h(t) = 0$.

El método recibe este nombre porque para cada $y'(a) = t_i$ se obtendrá una curva que alcanzará una altura en b igual a $y(b, t_i)$ y el objetivo es "pegarle´ ´ al β.

Para resolver el problema es necesario utilizar un método iterativo para la resolución de ecuaciones no lineales, como ejemplo se utilizará el método de la secante.

Se estima t_0 y se determina numéricamente $g(t_0)$ con alguno los métodos estudiados (Euler, RK, etc.)

Se estima t_1 y análogamente se obtiene $g(t_1)$.

Utilizando el método de la secante, se aproximará a la solución hasta que esta converja al valor indicado por la condición de borde en el extremo derecho del intervalo:

$$t_2 = t_1 - [g(t_1) - \beta]\frac{t_1 - t_0}{g(t_1) - g(t_0)}$$

$$t_3 = t_2 - [g(t_2) - \beta]\frac{t_2 - t_1}{g(t_2) - g(t_1)}$$

Se nota que $t_i \to T_\infty$ en el caso que el método converja.

Capítulo 7

Integración Numérica

En cursos de Análisis Matemático se ve el concepto de Integral de Riemman. Se recuerda que el área bajo una curva se aproxima mediante rectángulos, y se definen sumas por exceso y por defecto. Cuando el ínfimo de todas las sumas por exceso coincide con el supremo de las sumas por defecto, decimos que la función es integrable Riemann, y denotamos $\int_a^b f(x)dx$ a tal integral.

El teorema fundamental del cálculo de Integrales establece que toda función seccionalmente continua es integrable, y más aún, si f es continua en $[a,b]$ tenemos que $F(x) = \int_a^x f(t)dt$ verifica $F'(x) = f(x)$. Este resultado induce la regla de Barrow, donde basta con hallar una primitiva F de f y usar que $\int_a^b f(x)dx = F(b) - F(a)$. Esta es la maquinaria que se utiliza tradicionalmente para calcular integrales definidas. No obstante, la función f no siempre admite una primitiva elemental, y el método anterior presenta deficiencias. Por ejemplo, la función $f : f(x) = e^{x^2}$ no admite primitiva elemental; sin embargo, su integral en cualquier intervalo real se puede estimar con alto grado de precisión.

El cometido de este capítulo es aproximar la integral definida $\int_a^b f(x)dx$ numéricamente. Si f es integrable Riemman pero no es continua, sabemos que un método válido de aproximación es tomar una suma por exceso y otra por defecto. El error será más pequeño a medida que tomamos refinamientos de menor norma (es decir, con intervalos de menor longitud). Esto lleva a un compromiso de Ingeniería entre precisión y esfuerzo computacional.

Si además de ser integrable f es continua en $[a,b]$, sabemos por el Teorema de Stone-Weierstrass que existe una sucesión de polinomios (que podemos seleccionar como interpolantes) que se aproximan uniformemente a f. Concretamente, dado un número real $\epsilon > 0$ cualquiera, existe cierto grado n y polinomio interpolante p_n tal que $|p_n(x) - f(x)| < \epsilon$, para todo x perteneciente al intervalo $[a,b]$. En estas condiciones tenemos que el error cometido al reemplazar $\int_a^b f(x)dx$ por $\int_a^b p_n(x)dx$ es:

$$E = \left| \int_a^b (f(x) - p_n(x))dx \right| \leq \int_a^b |f(x) - p_n(x)|dx \leq \epsilon(b-a),$$

y el error puede tomarse tan pequeño como sea deseable, eligiendo polinomios interpolantes de grado alto. Dada la facilidad de cómputo mediante polinomios (sumas, restas, derivación,

integración, evaluación), vamos a limitar nuestro estudio a la integración numérica mediante polinomios interpolantes. Más interesante es que las reglas de integración van a depender únicamente de la evaluación de la función f en ciertos puntos, como veremos a continuación.

Antes de presentar los primeros métodos de integración numérica recordemos al lector el Teorema del error en interpolación polinómica, pues será utilizado en el transcurso del capítulo:

Teorema 7.0.1. *Sea f de clase C^{n+1} en el intervalo $[x_0, x_n]$ y p_n el polinomio interpolante a f por las abscisas $x_0 < x_1 < \ldots < x_n$. Luego, para cada $x \in [x_0, x_n]$ existe $\gamma(x) \in [x_0, x_n]$ tal que se cumple la siguiente igualdad para el error $E(x)$:*

$$E(x) = f(x) - p_n(x) = \frac{f^{n+1}(\gamma(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

Vamos a requerir también una forma especial del Teorema del Valor Medio para integrales. La forma básica es la siguiente:

Teorema 7.0.2. *Si f es continua en $[a, b]$ entonces existe $c \in [a, b]$ tal que $f(c) = \frac{1}{b-a} \int_a^b f(x)dx$.*

Demostración. Por el Teorema de Weierstass, f alcanza máximo M y mínimo m en $[a, b]$. Luego $M(b-a)$ es una suma por exceso y $m(b-a)$ es suma por defecto, por lo que:

$$m(b-a) \leq \int_a^b f(x)dx \leq M(b-a).$$

Al dividir entre $(b-a)$ tenemos que:

$$m \leq \frac{1}{b-a} \int_a^b f(x)dx \leq M.$$

Ahora por el Teorema del valor intermedio, f alcanza a todos los valores intermedios a m y M dentro de $[a, b]$, y en particular existe $c \in [a, b]$ tal que $f(c) = \frac{1}{b-a} \int_a^b f(x)dx$. □

Utilizaremos en nuestro análisis una versión más general:

Teorema 7.0.3. *Sea f continua en $[a, b]$ y g integrable y no negativa en el mismo intervalo. Entonces, existe $c \in [a, b]$ tal que $\int_a^b f(x)g(x)dx = f(c) \int_a^b g(x)dx$.*

Demostración. Nuevamente por el Teorema de Weierstass, f alcanza el mínimo m y el máximo M en $[a, b]$. Reemplazando tenemos que:

$$m \int_a^b g(x)dx \leq \int_a^b f(x)g(x)dx \leq M \int_a^b g(x)dx.$$

Si $\int_a^b g(x)dx = 0$, tenemos entonces por lo anterior que $\int_a^b f(x)g(x)dx = 0$, y cualquier c verifica el enunciado. En caso contrario, como g es no negativa tendríamos que $\int_a^b g(x)dx > 0$. Dividiendo entre $\int_a^b g(x)dx$ resulta:

$$m \leq \frac{\int_a^b f(x)g(x)dx}{\int_a^b g(x)dx} \leq M.$$

Puesto que f es continua alcanza a todos los números comprendidos entre m y M, por lo que existe $c \in [a,b]$ tal que

$$f(c) = \frac{\int_a^b f(x)g(x)dx}{\int_a^b g(x)dx}.$$

□

7.1. Método del Punto Medio

Veamos primeramente el método basado en polinomio interpolante más simple, que es una interpolación en un solo punto, concretamente el punto medio. Sea $f : [a,b] \to \mathbb{R}$ continua. Denotemos de ahora en más $I = \int_a^b f(x)dx$, que es el número que queremos estimar.

La estimación más gruesa es tomarse el punto medio, y proponer

$$I_{PM} = \int_a^b f\left(\frac{a+b}{2}\right)dx = (b-a)f\left(\frac{a+b}{2}\right).$$

Vamos a calcular el error cometido al usar I_{PM} asumiendo que $f \in C^2[a,b]$. Para abreviar consideremos $x_m = \frac{a+b}{2}$. Por Taylor, para cada $x \in [a,b]$ existe $\gamma_x \in [a,b]$ tal que:

$$f(x) = f(x_m) + f'(x_m)(x-x_m) + \frac{f''(\gamma_x)}{2}(x-x_m)^2.$$

Se observa que $\int_a^b (x-x_m)dx = 0$, por ser $x-x_m$ una función impar respecto del punto medio x_m. Integrando en cada miembro y usando la forma general del Teorema del Valor medio, tenemos que el error $E = I - I_{PM}$ es:

$$E = f''(c)\int_a^b (x-x_m)^2 dx = \frac{f''(c)}{3}(b-x_m)^3 = \frac{f''(c)}{24}(b-a)^3.$$

Se observa que si $f(x) = \alpha x + \beta$ es una función lineal arbitraria (constante o polinomio de grado 1), entonces el error cometido es 0. En otras palabras, para integrar una recta en un intervalo basta con tomar el punto medio y multiplicarlo por el largo de la base.

Este método, tanto como los que veremos a continuación, se combinan del uso de partición en rectángulos del intervalo de estudio, y se aplican en cada intervalo. El lector podrá notar ventajas de estos métodos denominados compuestos, en términos del error.

7.2. Método del Trapecio

Consiste en tomar el polinomio interpolante por los extremos. Se estima I por la integral en $[a,b]$ de la función lineal $p_1(x) = f(a)\frac{x-b}{a-b} + f(b)\frac{x-a}{b-a}$, que llamaremos I_T. Procedemos ahora calculando el error cometido al estimar la integral por la de un trapecio. Por el Teorema del Error mediante interpolación polinómica, tenemos que:

$$E = I - I_T = \int_a^b (f(x) - p_1(x))dx = -\int_a^b \frac{f''(\gamma_x)}{2}(x-a)(b-x)dx.$$

Usando que $(x-a)(b-x) \geq 0$ y el caso general del Teorema de Valor Medio, tenemos que:

$$E = -\frac{1}{2}f''(c)\int_a^b (x-a)(b-x)dx = -\frac{f''}{12}(b-a)^3. \tag{7.1}$$

Se observa nuevamente que el error E es nulo cuando f es una recta.

7.3. Método de Newton-Cotes

Veremos una famlia de métodos que incluyen a los dos anteriores. Esencialmente, la familia de métodos de integración de Newton-Cotes proponen integrar polinomios interpolantes por abscisas equidistantes en $[a, b]$. Se distinguen métodos abiertos y cerrados:

- Los métodos cerrados incluyen a las abscisas a y b; concretamente, se consideran puntos $x_n = a + nh$, con $n = 0, \ldots, N$ y h tal que $h = \frac{b-a}{N}$. Un ejemplo es el método del trapecio (con $h = b - a$, o $N = 1$).
- Los métodos abiertos no incluyen a las abscisas a y b. Se consideran puntos $x = a + nh$, con $n = 1, \ldots, N+1$, y $a + (N+2)h = b$. Un ejemplo es el método del punto medio (con $h = \frac{b-a}{2}$ y $N = 0$).

Vamos a hallar una expresión para la integral estimada de un método cerrado de $N+1$ puntos. Podemos expresar el polinomio interpolante de f por las abscisas $x_0, \ldots, x_N$ mediante el método de Lagrange. Luego, la integral es estima mediante:

$$\begin{aligned} I_{N-C} &= \int_a^b \sum_{n=0}^{N} f(x_n) \frac{\prod_{j\neq n}(x - x_j)}{\prod_{j\neq n}(x_n - x_j)} dx \\ &= \sum_{n=0}^{N} f(x_n) w_n, \end{aligned}$$

donde $w_n = \int_a^b \frac{\prod_{j\neq n}(x-x_j)}{\prod_{j\neq n}(x_n-x_j)} dx$. Se observa que w_n no depende de f. Más aún, el estudiante puede comprobar al efectuar el cambio de variable $t = \frac{x-x_0}{h}$ que w_n tampoco depende de los extremos a y b. Por lo tanto, los pesos w_n de los métodos de Newton-Cotes están tabulados.

Es importante destacar que la implementación de la regla de integración de Newton-Cotes consiste en reemplazar pesos de una tabla y multiplicarlos por las evaluaciones correspondientes de f. Si bien es simple, no se aconseja tomar grados superiores a 7, debido a inestabilidad (los pesos w_i alternan de signo).

Para analizar el rendimiento de distintos métodos basados en polinomios interpolantes, nos gustaría saber cuán bien es posible integrar polinomios. Ello motiva la siguiente definición:

Definición 7.3.1. Una regla de integración tiene grado de exactitud m si integra sin error a todos los polinomios de grado m o menos, y existe algún polinomio de grado $m+1$ en el que comete error no nulo.

Mediante el uso del Teorema de error de interpolación polinómica es posible hallar el grado de exactitud de la familia de métodos de Newton Cotes. Observando que la cantidad de abscisas tanto en los métodos abiertos como los cerrados es $N+1$, podemos resumir el grado de exactitud como sigue:

Teorema 7.3.1. *El grado de exactitud de un método de Newton-Cotes vale:*

- $N+1$ *si* N *es par.*
- N *si* N *es impar.*

Se observa que el método de Punto Medio verifica $N = 0$, y vimos que su grado de exactitud es 1. Por otra parte también vimos que el método del Trapecio verifica $N = 1$, y su grado de exactitud también es 1. La demostración del caso general es más laboriosa. La idea es similar, y utiliza el teorema del error por interpolación polinómica para remplazar la diferencia de funciones por el error en cada x.

En la siguiente sección respondemos a la pregunta: ¿cuál es el método que logra el máximo grado de exactitud posible?

7.4. Regla de Gauss

Gauss se propone construir la regla que logra el máximo grado de exactitud posible. Mediante una abstracción de las reglas anteriores, podemos considerar una regla de $N+1$ puntos y la combinación lineal $I_G = \sum_{i=0}^{N} f(x_i)w_i$, donde los pesos w_i son reales arbitrarios (posiblemente negativos, a diferencia de Riemann), y las abscisas x_i pertenecientes al intervalo $[a, b]$ posiblemente no equiespaciadas (a diferencia de los métodos de Newton-Cotes).

Tenemos así $2N + 2$ grados de libertad, y si hay cierta noción de independencia, sería posible alcanzar un grado máximo de exactitud $2N + 1$. Este último es una cota superior del grado de exactitud, que es alcanzable si y solo si se integran exactamente todos los monomios x^i, con $i = 0, 1, \ldots, 2N + 1$. Es posible formular este problema en términos de un sistema no lineal (invitamos al lector a plantear el sistema no lineal). El problema admite solución solamente si el sistema no lineal es compatible determinado. Pese a que no es habitual hallar una solución exacta a un sistema no lineal, Gauss brinda una respuesta contundente a este problema. Antes de enunciar el resultado principal es necesario introducir las siguientes definiciones:

Definición 7.4.1. El espacio de funciones cuadrático integrables en $[a, b]$ es

$$L^2[a, b] = \{f : [a, b] \to \mathbb{R}, \int_a^b f^2(x)dx < \infty\}$$

Se observa que es un espacio vectorial, donde identificamos funciones que son idénticas salvo en un conjunto de puntos de medida nula.

Al espacio $L^2[a, b]$ es posible dotarlo de un producto interno de la siguiente manera: $< f, g >= \int_a^b f(x)g(x)dx$. Por el Teorema de Cauchy-Schwarz, se ve que fg es integrable, siempre que f y

g pertenezcan a $L^2[a,b]$. El lector puede comprobar que la operación define un producto interno. Asimismo, se puede ver que los monomios $\{x^i\}_{i\in\mathbb{N}}$ son una base pero no ortogonal. De ahora en más vamos a fijar $a=-1$ y $b=1$ sin pérdida de generalidad, pues mediante un cambio de variable podemos transportar una integral en $[a,b]$ a otra en $[-1,1]$.

Definición 7.4.2. La famlia de polinomios normalizados de Legendre $\{q_i\}_{i\in\mathbb{N}}$ se obtienen de aplicar el proceso de ortonormalización a la base de monomios $\{x^i\}_{i\in\mathbb{N}}$ en el espacio $L^2[-1,1]$.

Denotemos mediante $\mathbb{P}_n$ al espacio vectorial de todos los polinomios de grado n o menor. Por definición sabemos que $q_{N+1}\perp\mathbb{P}_n$. Ahora estamos en condiciones de definir la Regla de Gauss:

Definición 7.4.3 (Regla de Gauss)**.** La Regla de Gauss de $N+1$ puntos selecciona $x_0,\dots,x_N$ como las $N+1$ raíces del polinomio de Legendre normalizado q_{N+1}, y los pesos se deducen de integrar el polinomio interpolante de Lagrange de f por las abscisas $x_0,\dots,x_N$.

Ahora estamos en condiciones de probar el resultado principal del capítulo:

Teorema 7.4.1. *La Regla de Gauss de $N+1$ puntos alcanza la cota superior del grado máximo de exactitud $2N+1$.*

Demostración. Sea $p\in\mathbb{P}_{2N+1}$ un polinomio arbitrario. Debemos probar que la regla de Gauss lo integra exactamente, es decir, con error nulo. Por el algoritmo de división de Euclides, existen dos polinomios $u,r\in\mathbb{P}_N$ tales que $p(x)=u(x)q_{N+1}(x)+r(x)$. Además en las raíces $x_0,\dots,x_N$ de q tenemos que $p(x_i)=r(x_i)$. Por otra parte, $q_{N+1}\perp u$, pues $u\in\mathbb{P}_N$. La regla de Gauss de $N+1$ puntos es:

$$I_G=\sum_{i=0}^{N}p(x_i)\int_{-1}^{1}l_i(x)dx,$$

siendo $l_i(x)=\frac{\prod_{j\neq i}(x-x_i)}{\prod_{j\neq i}(x_j-x_i)}$ el polinomio de Lagrange por las abscisas $x_0,\dots,x_N$. Por otra parte, la integral de p en $[a,b]$ es:

$$\begin{aligned}\int_{-1}^{1}p(x)dx&=\int_{-1}^{1}u(x)q(x)dx+\int_{-1}^{1}r(x)dx\\&=0+\int_{-1}^{1}\sum_{i=0}^{N}r(x_i)l_i(x)dx\\&=\sum_{i=0}^{N}p(x_i)\int_{-1}^{1}l_i(x)dx=I_G.\end{aligned}$$

Luego, la regla de Gauss integra exactamente a todo polinomio $p\in\mathbb{P}_{2N+1}$, como se quería demostrar. □

7.5. Otras técnicas de integración

7.5.1. Conversión de problemas de integración a PVI

Consideremos la función $y(x)=\int_a^x f(x)dx$.

Entonces $y(a) = 0$ y además $y'(x) = f(x)$. Por tanto, el problema de hallar $I(f) = \int_a^b f(x)dx$ es equivalente a determinar $y(b)$ resolviendo el PVI

$$(PVI): \begin{cases} y'(x) = f(x) \\ y(a) = 0. \end{cases}$$

Podemos usar todos los métodos para resolución de EDOs vistos hasta ahora.

7.5.2. Método de Monte Carlo

El método de Monte Carlo es particularmente útil para el cálculo de integrales de alta dimensión, donde los métodos anteriores se vuelven intratables debido al crecimiento exponencial del número de evaluaciones con la dimensión.

Teorema 7.5.1. *Sea $\{U_i\}_{i\in\mathbb{N}}$ una sucesión de variables aleatorias i.i.d. con distribución uniforme en el intervalo $[a, b]$, y $f : [a, b] \to \mathbb{R}$ una función integrable en $[a, b]$, entonces:*

$$\frac{1}{n}\sum_{i=1}^{n} f(U_i) \xrightarrow[n\to\infty]{c.s.} \frac{1}{b-a}\int_a^b f(x)dx$$

Demostración. Consideramos la variable aleatoria $Y_n = \frac{1}{n}\sum_{i=1}^n f(U_i)$. Por la Ley Fuerte de los Grandes Números $Y_n = \frac{1}{n}\sum_{i=1}^n f(U_i) \xrightarrow[n\to\infty]{c.s.} \mathbb{E}(f(U_i)) = \int_{-\infty}^{\infty} f(x)f_U(x)dx$, donde $f_U(x)$ es la densidad de U. Como U es uniforme en $[a, b]$, la integral tiene soporte en dicho intervalo, con valor constante para $f_U(x) = \frac{1}{b-a}$. Así $\mathbb{E}(f(U_i)) = \int_a^b f(x)\frac{1}{b-a}dx = \frac{1}{b-a}\int_a^b f(x)dx$. □

Es decir que si nuestro objetivo es calcular $I(f) = \int_a^b f(x)dx$, basta con simular una cantidad suficientemente grande de variables aleatorias uniformes, lo que puede realizarse muy rápidamente en un computador. Luego, evaluar f en los valores que salieron, promediar, y finalmente multiplicar por el tamaño del dominio, $b - a$. Con esto estaremos estimando con un buen grado de exactitud el valor de la integral.

Esto se generaliza de forma directa a una integral múltiple en el hiper-rectángulo $\mathcal{H} = [a_1, b_1] \times \cdots \times [a_d, b_d]$.

Sea $\{U_i\}_{i\in\mathbb{N}}$ una sucesión de vectores aleatorios i.i.d. en $\mathbb{R}^d$ con distribución uniforme en $\mathcal{H}$, y $f : \mathcal{H} \to \mathbb{R}$ una función integrable en su dominio, entonces:

$$\frac{1}{n}\sum_{i=1}^{n} f(U_i) \xrightarrow[n\to\infty]{c.s.} \frac{1}{(b_1 - a_1)\dots(b_d - a_d)}\int_H f(\mathbf{x})d\mathbf{x}$$

con $\mathbf{x} = (x_1, \dots, x_d)$.

Más en general aun, sea una región cualquiera $\mathcal{D} \subset \mathbb{R}^d$, con volumen $V = \int_{\mathcal{D}} 1d\mathbf{x}$, este problema es equivalente a calcular $\int_{\mathcal{H}} f(\mathbf{x})d\mathbf{x}$ con $\mathcal{H}$ suficientemente grande tal que $\mathcal{D} \subset \mathcal{H} = [a_1, b_1] \times \cdots \times [a_d, b_d]$, y $f(\mathbf{x}) = \mathrm{I}_{\mathcal{D}}(\mathbf{x}) = \begin{cases} 0 \text{ si } \mathbf{x} \notin \mathcal{D} \\ 1 \text{ si } \mathbf{x} \in \mathcal{D} \end{cases}$.

Esto significa que la proporción de veces que las variables aleatorias caen en la región $\mathcal{D}$ respecto al total de variables simuladas está relacionada con su tamaño (área, volumen, según sea la dimensión). A modo de ejemplo, si se simula una cantidad suficientemente grande de variables, y se encuentra que la cuarta parte de ellas caen en la región $\mathcal{D}$, entonces su volumen es aproximadamente la cuarta parte del volumen de $\mathcal{H}$, el cual es conocido.