Resumen MN

The FurfiOS Corporation

Febrero 2021

Índice general

Este apunte fue hecho en base a las clases teóricas de la Dra. Isabel Méndez Díaz del Segundo Cuatrimestre 2020, los apuntes de Franco Frizzo, Guido Tagliavini Ponce, y Julián Sackmann, complementado con bibliografía de la materia:

- Burden [2](Capítulo 1): utilizado para completar el capítulo de errores numéricos.
- Bjorck [1](Capítulo 6): utilizado para completar el capítulo de ceros de funciones.
- Demmel [4](Capítulo 3): utilizado para completar el capítulo de CML.
- Highman [5](Capítulos 1 y 4): utilizado para completar el capítulo de errores numéricos, y motivación para el pivoteo en la eliminación gaussiana y la factorización PLU.
- Meyer [6](Capítulo 5): utilizado para completar la motivación detrás de resolver CML.
- Sauer [7](Capítulo 1): utilizado para dar un caso de mal funcionamiento del método de regla falsa
- Shakiban [3](Capítulo 1): utilizado para completar la explicación de la factorización PLU.

Acá tienen el *link* para editar el overleaf `https://www.overleaf.com/5845356168twvmvzdqmpcy`.

Capítulo 1

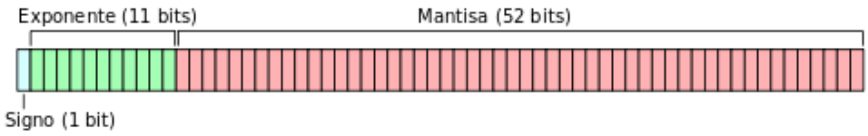
Aritmética de la computadora

Dado que en una computadora todos los números son representados mediante una cantidad de dígitos finita y fija, valores como π o $\sqrt{2}$ no se pueden manipular con completa exactitud, pues al ser irracionales tienen infinitos decimales no periódicos. Los irracionales no son los únicos números no representables correctamente en una computadora: aquellos racionales con una cola decimal no periódica suficientemente grande tampoco lo serán. En una computadora sólo se pueden representar de forma exacta un subconjunto de los números racionales. Esto hace que al realizar cómputos con números reales se generen **errores numéricos**.

1.1. Representación estándar IEEE

El estándar fijado por la IEEE contempla varias representaciones que se distinguen por su precisión. Las dos más frecuentemente utilizadas son *single* (32 bits) y *double* (64 bits). Las dos restantes son *half* (16 bits) y *quadruple* (128 bits). Todas estas representaciones son binarias y de punto flotante.

La precisión *double* tiene la siguiente estructura:



- **Signo** (s): 1 bit. El número representado es positivo si $s = 0$ y negativo si no.
- **Exponente** (e): 11 bits, en notación exceso $2^{10} - 1$.
- **Mantisa** (m): 52 bits. Se considera una mantisa normalizada a 1. Es decir, el número representado tiene mantisa $(1, m)_2$.

En definitiva, el número representado es

$$(-1)^s \cdot (1, m)_2 \cdot 2^{(e)_2 - (2^{10} - 1)}$$

Decimos que un cómputo genera *underflow* si su resultado es menor que el mínimo positivo representable, en módulo. Análogamente, decimos que genera *overflow* si su resultado es mayor que el máximo positivo representable, en módulo.

1.1.1. Aproximación de los reales mediante números de máquina

En esta sección estudiaremos cuán eficaz es la aproximación de un número real mediante un sistema con las características del presentado previamente. El tipo de sistemas a los que nos referimos son representaciones de punto flotante con una longitud de mantisa fija y exponente acotado.

En general, tenemos dos modos de aproximación. Consideremos la escritura

$$x = (0, d_1 \cdots d_k d_{k+1} \cdots) \cdot 10^e$$

con $d_1 \neq 0$ (esta escritura es única dado que $d_1 \neq 0$). Entonces dos formas de aproximar x son:

- **Truncamiento.** Simplemente descartamos los dígitos $d_{k+1}, d_{k+2}, \cdots$, para obtener

$$fl(x) = (0, d_1 \cdots d_k) \cdot 10^e$$

- **Redondeo.** Si $d_{k+1} < 5$ entonces truncamos. Si no, sumamos $0, \underbrace{0 \cdots 05}_{k+1 \text{ dígitos}} \cdot 10^e = 5 \cdot 10^{-(k+1)} \cdot 10^e$ a x y truncamos. En este último caso lo que queda es

$$fl(x) = [(0, d_1 \cdots d_k) + 10^{-k}] \cdot 10^e$$

Una forma equivalente de enunciar estos criterios es la siguiente. Sea x^- es el máximo número de máquina menor o igual que x. Análogamente, sea x^+ el mínimo número de máquina mayor o igual que x. Entonces, el truncamiento aproxima por x^- mientras que el redondeo aproxima por aquel valor de x^- o x^+ más cercano a x.

En general, las computadoras utilizan la aproximación por redondeo.

1.1.2. Distribución de los números de máquina sobre la recta real

Los números de máquina no están uniformemente distribuidos. Intuitivamente, cuanto más nos alejemos de 0, más esparcidos estarán los números de máquina.

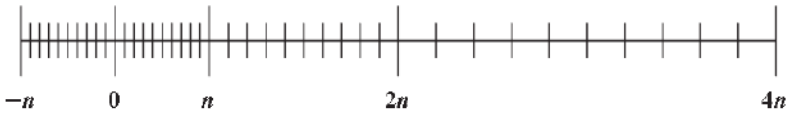


Figure 10.20 Density of Floating-Point Numbers

Esta distribución puede parecer extraña. Contrariamente a lo intuitivo, que haría pensar que una distribución uniforme sería más útil, esta distribución exponencial de los números de máquina resulta práctica pues se basa en la idea de que cuanto más chicos sean los números del rango en el que estamos trabajando, nos va a interesar poder identificar las diferencias más pequeñas. Contrariamente, cuanto más grandes sean los números con los que trabajemos, no nos va a interesar poder distinguir con tanta precisión. Además, esta distribución hace que la cota para el error relativo

$$\left| \frac{y - fl(y)}{y} \right|$$

mediante aritmética finita de k dígitos sean independientes del número que se va a representar, siendo esta 10^{-k+1}.

1.2. Error absoluto y error relativo

Definición 1.2.1. Sea $x \in \mathbb{R}$. Sea $x^* \in \mathbb{R}$ un valor que pretende aproximar a x. La medidas más útiles respecto a la precisión de x^* son

- El error o error real $x - x^*$.
- El error absoluto $|x - x^*|$.
- El error relativo $\frac{|x-x^*|}{|x|}$.

Notemos que la diferencia entre el error absoluto y el error relativo es que este último es independiente de la escala de x. Luego, el error relativo está relacionado con la noción de tener correctamente representados a los dígitos significativos. Es por este motivo que en el contexto de las ciencias de la computación, donde las respuestas a los problemas pueden variar enormemente en magnitud, usualmente va a ser de mayor interés el error relativo.

Cuando trabajamos con vectores $x \in \mathbb{R}^n$, $x^* \in \mathbb{R}^n$, el error relativo se define con una norma vectorial $\|\bullet\|$, tal que

$$\text{error relativo } = \frac{\|x - x^*\|}{\|x\|}$$

También podemos utilizar como medida al *error relativo por componentes*, que se define como

$$\max_i \frac{|x_i - x_i^*|}{|x_i|}$$

medida que es ampliamente utilizada en el contexto del análisis del error.

Un criterio que impondremos en un algoritmo, siempre que sea posible, es que los pequeños cambios en los datos inicial produzcan pequeños cambios en los resultados finales. Un algoritmo que satisface esta propiedad recibe el nombre de **estable**; de lo contrario es **inestable**. Para formalizar este concepto, veamos la siguiente definición

Definición 1.2.2. Supongamos que se presenta un error E_0 en alguna etapa de los cálculos, y E_n representa la magnitud del error después de n operaciones.

- Si $E_n \approx n \cdot C \cdot E_0$, con C constante, entonces se dice que el crecimiento del error es **lineal**.
- Si $E_n \approx C^n \cdot E_0$, para $C > 1$ constante, entonces se dice que el crecimiento del error es **exponencial**.

Normalmente, el crecimiento lineal del error es inevitable, y cuando C y E_0 son pequeños, los resultados son aceptables. Si el crecimiento del error fuera exponencial, entonces el error se volvería inaceptablemente grande, incluso para valores chicos de n. Por lo tanto, un algoritmo con crecimiento exponencial del error es **inestable**, mientras que un algoritmo que presenta un crecimiento lineal del error es **estable**.

1.3. Fuentes de Errores

Hay tres fuentes principales de errores en los cómputos numéricos: redondeo, incertidumbre en los datos, y truncamiento. Los errores asociados al redondeo son inevitables, al ser consecuencia de trabajar con aritmética de precisión finita. Si bien sus efectos pueden ser reducidos a partir de utilizar una aritmética de mayor precisión (doble o ampliada), esto requiere de un mayor costo en cuanto al tiempo de cómputo.

Los errores asociados a la incertidumbre de los datos son siempre una posibilidad cuando estamos resolviendo problemas prácticos, y pueden aparecer de distintas maneras: errores de medición de cantidad físicas, errores al momento de guardar los datos en la computadora, o pueden ser el resultado de errores en cómputos anteriores. Los efectos que generan los errores en los datos son, generalmente, más fáciles de comprender que los efectos que generan los errores de redondeo, ya que estos pueden ser analizados

utilizando teoría de perturbación sobre el problema a resolver, mientras que los errores de redondeo requieren de análisis específico del método utilizado.

Los errores asociados al truncamiento tienen que ver con, por ejemplo en el caso de la regla del trapecio, tomar una cantidad finita de una serie, y asumir que el resultado se parece al límite de la sucesión. Los términos omitidos constituyen a los errores de truncamiento.

> *Los errores de redondeo y la inestabilidad son importantes, y los analistas numéricos siempre serán los expertos en estos temas y se esforzarán por asegurarse de que los incautos no tropiecen con ellos. Pero nuestra misión central es calcular cantidades que normalmente son incomprensibles, desde un punto de vista analítico, y hacerlo a la velocidad del rayo.*

Algunos errores conceptuales y mitos son:

- *La cancelación en la resta de dos números casi iguales siempre causa errores graves.* Vimos que en el caso de $z + (x - y)$ la cancelación resulta inocua.
- *Los errores de redondeo pueden abrumar un cálculo solo si se acumulan un gran número de ellos.* Muy a menudo, la inestabilidad no se debe a la acumulación de millones de errores de redondeo. sino por el crecimiento insidioso de unos pocos errores de redondeo.
- *Un cálculo breve libre de cancelación, underflow y overflow debe ser preciso.* Vimos que cuando trabajamos con sistemas mal condicionados podemos obtener errores graves en los cálculos.
- *Aumentar la precisión con la que se realiza un cálculo aumenta la precisión de la respuesta.* Esto solo vale cuando no tenemos ninguna otra fuente de errores numéricos.
- *La respuesta final calculada de un algoritmo no puede ser más precisa que cualquiera de las cantidades intermedias, es decir, los errores no se pueden cancelar.*
- *Los errores de redondeo solo pueden obstaculizar, y no ayudar, el éxito de un cálculo.*

1.3.1. Errores de redondeo clásicos

Cancelación Catastrófica

Al restar dos números cercanos, el resultado estará próximo a cero, lo que puede ocasionar que se pierdan dígitos significativos. Este fenómeno se conoce como *cancelación catastrófica*, y tiene un gran impacto en el error relativo.

A modo de ejemplo, supongamos nuevamente que la precisión es de $k = 5$ dígitos, y sean que $y = 1/3 * 3$ e $x = 1$, entonces

$$\begin{aligned} x \ominus y &= fl(fl(x) - fl(y)) \\ &= fl(0,99999 - 1) \\ &= fl(0,00001) \\ &= 0,00001 \end{aligned}$$

Sin embargo $x - y = 0$, dando un error relativo de 1. El problema está en que, si bien la resta es exacta, esta genera un resultado del mismo tamaño que el error original en y. En otras palabras, la resta eleva la importancia del error anterior.

Para realizar un análisis más fino sobre el fenómeno de la cancelación, consideremos la resta $\hat{x} = \hat{a} - \hat{b}$, donde $\hat{a} = a \cdot (1 + \Delta_a)$, y $b = \hat{b} \cdot (1 + \Delta_b$, donde los términos Δ_a, Δ_b representan al error relativo de a y

b, respectivamente. Si ahora queremos calcular el error relativo de $x = a - b$, obtenemos

$$\begin{aligned}
\left|\frac{x-\hat{x}}{x}\right| &= \left|\frac{a-b-(\hat{a}-\hat{b}}{a-b}\right| \\
&= \left|\frac{a-b-(a\cdot(1+\Delta_a)-b\cdot(1+\Delta_b)}{a-b}\right| \\
&= \left|\frac{-a\cdot\Delta_a - b\cdot\Delta_b}{a-b}\right| \\
&\leq \frac{|a\cdot\Delta_a| + |b\cdot\Delta_b|}{|a-b|} \\
&\leq \text{máx}\left(|\Delta_a, \Delta_b| \cdot \frac{|a|+|b|}{|a-b|}\right)
\end{aligned}$$

Luego, podemos esperar que cuando $|a-b| \ll |a|+|b|$ ocurra una cancelación catastrófica. Este análisis nos muestra que la cancelación genera que errores relativos en $\hat{a}$ y $\hat{b}$ se magnifican. En otras palabras, la cancelación pone en evidencia errores que venimos arrastrando de cálculos previos.

Es importante notar que la cancelación no es siempre un error grave. Por ejemplo, si estamos tratando con datos iniciales sin error, o si tenemos una expresión del tipo $z + (x - y)$, con $z \gg x \approx y$.

Multiplicación por números grandes o división por números pequeños

En este caso, se produce una amplificación del error absoluto acarreado. Supongamos que x^* es una aproximación de máquina de x. Dividiendo a x^* por un número muy pequeño, digamos 10^{-n} para cierto $n > 0$, obtenemos el número de máquina $x^*/10^{-n}$ que aproxima a $x/10^{-n}$ con un error absoluto de

$$|x^*/10^{-n} - x/10^{-n}| = |x^* - x| \cdot 10^n$$

El error absoluto $|x^* - x|$ del primer redondeo se ve amplificado en un factor de 10^n.

Suma

Las sumas de números de punto flotante ocurren al evaluar productos internos, medias, variaciones, normas y todo tipo de funciones no lineales. Aunque a primera vista la sumatoria puede parecer que ofrece poco margen para el ingenio algorítmico, la habitual "suma recursiva"(con varios ordenamientos) es sólo una de las diversas técnicas posibles. Veamos por qué es necesario tener alternativas a la suma convencional.

Supongamos que nuestra aritmética tiene una precisión de $k = 5$ dígitos de mantisa. Sean $x = 0,88888888 \cdot 10^7$ e $y = 0,1 \cdot 10^2$. Entonces

$$\begin{aligned}
x \oplus y &= fl(fl(x) + fl(y)) \\
&= fl(0,88888 \cdot 10^7 + 0,1 \cdot 10^2) \\
&= fl(0,888881 \cdot 10^7) \\
&= 0,88888 \cdot 10^7
\end{aligned}$$

Es decir, el término x ha absorbido a y. Ahora, revisemos los principios de la suma.

En la mayoría de los contextos de la programación traduciríamos la sumatoria $\sum_{i=1} nx_i$ como

Entrada: $x_1, \ldots, x_n$
Salida: s

```
s ← 0
for i = 1, ..., n do
    s ← s + x_i
return s
```

Este algoritmo es conocido como *suma recursiva*. Como los errores de redondeo individuales dependen de los operandos que estén siendo sumados, la precisión de la suma computada $\hat{s}$ varía dependiendo del ordenamiento de los x_i. Dos propuestas interesantes para su ordenamiento son de menor a mayor $|x_1| \leq |x_2| \leq \cdots \leq |x_n|$, y de mayor a menor $|x_1| \geq |x_2| \geq \cdots \geq |x_n|$.

Otro método posible es la *suma de a pares*, en donde cada x_i es sumado de a pares de la siguiente manera

$$y_i = x_{2i-1}x_{2i}, \quad \text{para } i = 1 : \left\lfloor \frac{n}{2} \right\rfloor \text{ con } (y_{(n+1)/2} = x_n \text{si } n \text{ es impar}$$

, aplicando este proceso de manera iterativa $\log_2(n)/$ veces. La suma de a pares es una opción atractiva para la programación paralela, ya que cada uno de los pasos puede hacerse en paralelo.

Un tercer método es el método de *inserción*. Se tiene una lista ordenada $x_1, \cdots, x_n$ de menor a mayor. Luego, se realiza la suma $x_1 + x_2$, y el resultado se inserta en la lista $x_3 \cdots x_n$ de forma tal que la lista siga estando ordenada de menor a mayor.

En general, hay una gran variedad de métodos de suma de donde elegir. Para cada método el error puede variar significativamente dependiendo de los datos, dentro del rango permitido por la cota del error. Sin embargo, algunos guías específicas para la elección del método pueden ser dadas.

- Si es importante una alta precisión, considere implementar la suma recursiva con mayor precisión; si es factible, esto puede ser menos costoso (y más preciso) que usar uno de los métodos alternativos en la precisión de trabajo.
- Para la mayoría de los métodos, los errores son, en el peor de los casos, proporcionales a n. Si n es muy grande, la suma de a pares es atractiva.
- Si todos los x_i tienen el mismo signo, la suma recursiva con orden creciente y el método de inserción resultan buenas opciones.
- Para sumas con una mala cota para el error generado por una cancelación, la suma recursiva con el orden decreciente es atractiva, aunque no se puede garantizar que logre la mejor precisión.

Las consideraciones de costo computacional y la forma en que se generan los datos pueden descartar algunos de los métodos. La suma recursiva y la suma por pares se pueden implementar en $O(n)$, mientras que el resto de los métodos son más costosos, ya que requieren de realizar búsquedas u ordenamientos.

1.4. Aritmética anidada

La pérdida de precisión debido a un error de redondeo también se puede reducir al reacomodar los cálculos o bien reduciendo el número de cálculos. Una operación típica es la evaluación de polinomios; por ejemplo

$$2x^3 + 4x^2 + 5x + 15$$

Esta expresión que requiere de 8 productos y 3 sumas. Como enfoque alternativo, este polinomio se puede expresar de forma **anidada** como

$$((2x + 4)x + 5)x + 15$$

de manera tal que esta expresión solo requiere de 3 productos y 3 sumas.

En general, los polinomios *siempre* deberían expresarse en forma anidada antes de realizar una evaluación, ya que esta forma minimiza el número de cálculos aritméticos requeridos, disminuyendo (en general) el error generado.

1.5. Epsilon de máquina

Llamamos epsilon de máquina al máximo error relativo que puede cometerse por redondeo. Lo notamos ε.

La noción de ε permite dar cotas superiores sobre el error cometido al realizar distintas operaciones en una máquina, independientemente de las características del sistema de representación de números que utilice la misma. Las cuatro operaciones estándar que realiza una computadora son,

$$
\begin{aligned}
x \oplus y &= fl(fl(x) + fl(y)) \\
x \ominus y &= fl(fl(x) - fl(y)) \\
x \otimes y &= fl(fl(x) \times fl(y)) \\
x \oslash y &= fl(fl(x) / fl(y))
\end{aligned}
$$

que representan, respectivamente, la suma, la resta, el producto y el cociente de dos números reales x e y.

Sean $x, y \in \mathbb{R}$ no nulos, con igual signo. En cualquier máquina con suma $\oplus$, que utilice redondeo, vale

$$
\frac{|(x+y) - (x \oplus y)|}{|x+y|} \leq 2\varepsilon + \varepsilon^2
$$

Capítulo 2

Repaso de Álgebra Lineal

Este capítulo está dedicado a hacer un repaso del álgebra lineal. En particular, nos vamos a enfocar en recordar algunas definiciones, conceptos, propiedades sobre vectores y matrices que nos van a resultar útiles para el desarrollo de algunos temas que veremos a lo largo de la materia.

2.1. Vectores

Vamos a empezar recordando lo que es un vector y sus operaciones básicas:

2.1.1. Operaciones entre vectores

- **Vector**: $v \in \mathbb{R}^n$ n-upla de coficientes reales.

$$v = \{v_1, v_2, ..., v_n\}$$

- **Suma**: $w = vu$ con $w_i = v_i + u_i$, para $i = 1, ..., n$. Es conmutativa y asociativa.
- **Multiplicación por escalar**: Sea $\alpha \in \mathbb{R}, w = \alpha v$, con $w_i = \alpha v_i$, para $i = 1, ..., n$
- **Producto interno** (o escalar): $\langle u, v \rangle = \sum_{i=1}^{n} u_i v_i$
- **Combinación lineal**: Dado un conjunto de vectores $v_k \in \mathbb{R}^n$, con $k = 1, ..., K$, w es combinación lineal del conjunto si:

$$w = \sum_{k=1}^{K} \alpha_k v_k, \text{ con } w \in \mathbb{R}^n$$

Dentro de los vectores en $\mathbb{R}^n$ vamos a destacar al vector nulo $0 = (0, 0, ..., 0)$. El vector nulo siempre se puede escribir como una combinación lineal de cualquier conjunto de vectores, ya que basta tomar todos los escalares de la combinaciones lineal igual a 0.

Sin embargo, dependiendo del conjunto de vectores elegido, a veces es posible escribir al vector nulo como una combinación lineal donde no todos los coeficientes sean nulos. Por ejemplo, consideremos los siguiente vectores:

$$\begin{array}{ll} v_1 = (-1, 0, 0) & \alpha_1 = 2 \\ v_2 = (2, 1, 0) & \alpha_2 = 1 \\ v_3 = (0, -3, 0) & \alpha_3 = 1/3 \\ v_4 = (1, 5, 3) & \alpha_4 = 0 \end{array} \Longrightarrow 0 = \sum_{i=1}^{4} \alpha_i v_i$$

Entonces, hemos logrado una representación del vector nulo, mediante una combinación lineal donde no todos los coeficientes son nulos. Nuevamente, esto no siempre es posible, y depende del conjunto de vectores que estemos considerando. Esto da pie a la noción de vectores **linealmente independientes** o **linealmente dependientes**:

- **Vectores li**: $\sum_{k=1}^{K} \alpha_k v_k = 0 \Longleftrightarrow \alpha_k = 0 \forall k = 1, ..., K$
- **Vectores ld**: $\exists\, \alpha_k$ con $k = 1, ..., K$ no todos nulos tal que $\sum_{k=1}^{K} \alpha_k v_k = 0$.

Es decir, en caso de que la única manera de escribir al vector nulo como combinación lineal sea tomando todos los escalares iguales a 0, diremos que se trata de un conjunto de vectores linealmente independientes. En el caso de que existan escalares no todos nulos, de tal manera que la combinación lineal nos dé el vector nulo, diremos que los vectores son linealmente dependientes.

Dado el conjunto de vectores $v_1, ..., v_K$, si consideramos todas las combinaciones lineales, eso da pie a lo que se conoce como **subespacio generado**:

- **Subespacio generado** $S = \{x \in \mathbb{R}^n \text{ tal que } x = \sum_{k=1}^{K} \alpha_k v_k\}$

Dentro de un subespacio, el cardinal del conjunto de vectores linealmente independientes lo llamaremos la **dimensión** del subespacio. A cualquier conjunto linealmente independiente cuyo cardinal coincida con la dimensión del subespacio lo llamaremos **base** del subespacio, y tienen la propiedad de que cualquier vector del subespacio puede ser escrito como combinación lineal de ellos.

2.2. Matrices

De la misma manera en la que hemos recordado los vectores, vamos a recordar las matrices. En este caso, estamos hablando de arreglos bi-dimensionales (tenemos dos parámetros m = cantidad de filas, n = cantidad de columnas):

$$A \in \mathbb{R}^{m \times n},\ A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{in} \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

2.2.1. Operaciones entre matrices

De la misma manera en la que recordamos operaciones entre vectores, podemos recordar algunas operaciones entre las matrices $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$:

- **Suma**: definida si $m = p$, $n = q$, $C \in \mathbb{R}^{n \times m}$, y es una operación conmutativa y asociativa:

$$C = A + B,\ \text{con } c_{ij} = a_{ij} + bij,\ \text{para } i = 1, ..., m,\ j = 1, ..., n$$

- **Producto por escalar**: $C \in \mathbb{R}^{n \times m}$, $\alpha \in \mathbb{R}$:

$$C = \alpha A \text{ con } c_{ij} = \alpha a_{ij},\ \text{para } i = 1, ..., m,\ j = 1, ..., n$$

- **Producto entre matrices**: Sea $C \in \mathbb{R}^{m \times q}$, $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $AB = C$. Definida si $n = p$:

$$C_{ij} = \sum_{k=1}^{n} a_{ik}.b_{kj} \quad \forall i \in [1...m],\ j \in [1...p]$$

Lema 1: Sea $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$ y a_i la columna i-ésima de A:

$$Ax = \sum_{i=1}^{n} a_i.x_i$$

Aquí podemos notar que Ax no es otra cosa que una combinación lineal de las columnas de A.

Lema 2: Sea , $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times m}$, a_i la columna i-ésima de A, y b_i^t la fila i-ésima de B:

$$AB = \sum_{k=1}^{m} a_k.b_k^t$$

Lema 3: Sea , $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times n}$, a_i^t la fila i-ésima de A, y b_i la columna i-ésima de B:

$$fila_i(AB) = a_i^t.B$$
$$col_i(AB) = A.b_i$$

Una primera observación es que la multiplicación de matrices **no es conmutativa**, incluso en el caso en que las dimensiones sean las mismas.

2.2.2. Matrices especiales

Dentro de las matrices, vamos a recordar algunas en particular:

- **Matriz identidad**: es una matriz cuadrada, en la diagonal tiene 1s, y fuera de la diagonal tiene 0s. $I \in \mathbb{R}^{n\times n}$, con $I_{ij} = 0$ si $i \neq j$, e $I_{ii} = 1$:

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

- **Matriz diagonal**: es una matriz cuadrada, y por fuera de la diagonal tiene 0s. $D \in \mathbb{R}^{n\times n}$, con $d_{ij} = 0$ si $i \neq j$:

$$\begin{bmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}$$

- **Matriz triangular superior**: es una matriz cuadrada, y por debajo de la diagonal (no incluida) tiene 0s. $U \in \mathbb{R}^{n\times n}$ con $u_{ij} = 0$ si $i > j$

$$\begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & * \end{bmatrix}$$

- **Matriz triangular inferior**: es una matriz cuadrada, y por arriba de la diagonal (no incluida) tiene 0s. $L \in \mathbb{R}^{n\times n}$ con $l_{ij} = 0$ si $i < j$

$$\begin{bmatrix} * & 0 & \cdots & 0 \\ * & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & * \end{bmatrix}$$

Propiedades

- Una propiedad interesante de las matrices triangulares es que **el producto de triangulares inferiores (superiores) es triangular inferior (superior)**.
- Otra propiedad de utilidad es que el determinante de una matriz triangular es igual al producto de los elemento de la diagonal.
- Notemos que una matriz diagonal es, en particular, una matriz triangular (tanto superior, como inferior).

Hay un concepto relacionado con las matrices, que dada una matriz $A \in \mathbb{R}^{m\times n}$, el **rango** de A se define como la cantidad máxima de columnas (o filas) linealmente independientes.

En el caso de que la matriz sea cuadrada, a veces existe lo que se conoce como la **inversa** de la matriz:

- **Definición:** Si A es una matriz cuadrada, y $\exists B$ del mismo tamaño que A, tal que $AB = BA = I$, entonces A es inversible (no singular) y B es la inversa de A, y se denota $B = A^{-1}$.
- $AA^{-1} = A^{-1}A = I$
- A es inversible $\iff rango(A) = n \iff det(A) \neq 0$
- Si A tiene inversa, entonces es única.
- $(A^{-1})^{-1} = A$.
- Si A y B cuadradas, y AB inversible, entonces $(AB)^{-1} = B^{-1}A^{-1}$.
- La inversa (si existe) de una matriz diagonal es una matriz diagonal, y en particular $(D^{-1})_{ii} = \frac{1}{d_{ii}}$.
- La inversa (si existe) de una matriz triangular inferior (superior) es una matriz triangular inferior (superior).
- Si una matriz $A \in \mathbb{R}^{n\times n}$ es triangular e inversible, entonces $a_{ii} \neq 0$ para todo $i = 1, \cdots, n$, y además $(A_{ii}^{-1} = \frac{1}{a_{ii}}$.

Hay ciertas matrices cuyos coeficientes guardan cierta relación. Entre ellas está el conjunto de matrices **estrictamente diagonal dominante**. Decimos que una matriz es *edd* cuando:

$$|a_{ii}| > \sum_{j\neq i} |a_{ij}| \, \forall i = 1, ..., n$$

Es decir, decimos que una matriz es *edd* si cada elemento de la diagonal es mayor estricto en módulo que la suma del resto de los elementos de su fila. Una de las propiedades de este tipo de matrices es que son inversibles (lo veremos en detalle más adelante).

Vamos a pasar a unas matrices muy particulares que son las **matrices de permutación**. Las matrices de permutación son matrices cuadradas, que son iguales a la identidad, pero se tienen desordenadas las columnas (o filas), es decir, son una permutación de la matriz identidad. $P \in \mathbb{R}^{n\times n}$:

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

Este tipo de matrices nos permiten permutar matrices, alterando el orden original de las filas (PA) o de las columnas (AP) de la matriz A, de la misma forma en la que la matriz de permutación P tenía cambiado el orden respecto de la identidad:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} a_1^t \\ a_2^t \\ a_3^t \\ a_4^t \end{bmatrix} = \begin{bmatrix} a_2^t \\ a_4^t \\ a_3^t \\ a_1^t \end{bmatrix}$$

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} a_4 & a_1 & a_3 & a_2 \end{bmatrix}$$

Obs: como las matrices de permutación son permutaciones de la identidad, podemos almacenar únicamente el orden alterado de las columnas de la identidad, el cual, en este caso, sería $[4, 1, 3, 2]$.

Otro tipo de matrices que vamos a mencionar son las **matrices elementales**. Vamos a empezar con las que se conocen como matrices elementales de tipo 1, que son muy parecidas a la identidad, salvo que tienen un escalar α en algún lugar de la diagonal:

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cuando multiplicamos a una matriz A por una matriz elemental, multiplicamos la fila (EA) o la columna (AE) por α:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} a_1^t \\ a_2^t \\ a_3^t \\ a_4^t \end{bmatrix} = \begin{bmatrix} a_1^t \\ \alpha.a_2^t \\ a_3^t \\ a_4^t \end{bmatrix}$$

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_1 & \alpha.a_2 & a_3 & a_4 \end{bmatrix}$$

El segundo tipo de matriz elemental (matriz elemental de tipo 2) es aquella que es igual a la identidad, pero por fuera de la diagonal tiene un escalar α no nulo:

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \alpha & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cuando multiplicamos a una matriz A cualquiera por este tipo de matrices es el siguiente:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \alpha & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} a_1^t \\ a_2^t \\ a_3^t \\ a_4^t \end{bmatrix} = \begin{bmatrix} a_1^t \\ a_2^t \\ \alpha.a_1 + a_3^t \\ a_4^t \end{bmatrix}$$

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \alpha & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_1 + \alpha.a_3 & a_2 & a_3 & a_4 \end{bmatrix}$$

Obs: Las matrices elementales son inversibles, y las matrices de permutación son ortogonales.

2.3. Transformaciones Lineales

Finalmente, vamos a recordar algunos otros conceptos que relacionan a las matrices con las transformaciones lineales o con los sistemas de ecuaciones. Recordemos la definición de transformación lineal:

Definición: Si $T : V \to W$ es una función de un espacio vectorial V a un espacio vectorial W, entonces T se denomina una transformación lineal de V a W si y solo si, $\forall\, u,\, v \in V,\, c \in \mathbb{R}$, vale que:

- $T(u + v) = T(u) + T(v)$
- $T(c.u) = c.T(u)$

En este caso, vamos a decir que el **Espacio Imagen** definido por la matriz A es el conjunto de vectores en $y \in \mathbb{R}^n$ tal que existe $x \in \mathbb{R}^n$ con $Ax = y$:

$$Im(A) = \{y \in colm / \exists x \in \mathbb{R}^n, Ax = y\}$$

Además del espacio imagen, existe un espacio definido para las transformaciones lineales que es el espacio nulo o **Núcleo** de A. El $Nu(A)$ es el conjunto de $x \in \mathbb{R}^n$ tales que $Ax = 0$:

$$Nu(A) = \{x \in \mathbb{R}^n / Ax = 0\}$$

En el caso de que las columnas de A sean linealmente independientes, la única manera de escribir al 0 como combinación lineal de las columnas de A va a ser con todos los coeficientes nulos. En cambio, si las columnas son linealmente dependientes, van a existir valores de $x \neq 0$ tales que $Ax = 0$. Por lo tanto:

$$Nu(A) \neq \{0\} \iff \text{ las columnas de A son li}$$

2.4. Operaciones Varias

- **Traspuesta**
  - **Definición:** Si A es cualquier matriz $m \times n$, entonces la **traspuesta** de A, denotada por A^T, se define como la matriz $n \times m$ que resulta de intercambiar los renglones y las columnas de A. Es decir, la i-ésima columna de A^T es el i-ésimo renglón de A.
  - $a^t_{ij} = a_{ji}$ para todo $i = 1, ..., m,\, j?1, ..., n$
  - $(A^t)^t = A$
  - $(A + B)^T = A^T + B^T$
  - $(AB)^T = B^T.A^T$
  - $(A^t)^{-1} = (A^{-1})^t$
- **Traza**:
  - **Definición**: La traza de una matriz A $\in \mathbb{R}^{n\times m}$:

$$tr(A) = \sum_{i}^{n} a_{ii}$$

  - $tr(AB) = tr(BA)$.
  - $tr(A) = tr(A^T)$.

- **Determinante**:
  - **Definición**: Sea A una matriz cuadrada, entonces:
    1. Si $A = [a]$ es una matriz 1×1, entonces $det(A) = a$.
    2. Si A es ina matriz $n \times n$, con $n > 1$, el **menor** M_{ij} e el determinante de la submatriz $(n-1) \times (n-1)$ de A obtenida al quitar le i-ésima fila y la j-ésima columna de la matriz A.
    3. El **cofactor** A_{ij} asociado con M_{ij} está definido por $A_{ij} = (-1)^{i+j} M_{ij}$.
    4. El **determinante** de la matriz $An \times n$, cuando $n > 1$, está dado ya sea por:

       $$det(A) = \sum_{j=1}^{n} a_{ij} A_{ij}, \text{ para cualquier } i = 1, 2, \cdots, n,$$

       o mediante

       $$det(A) = \sum_{i=1}^{n} a_{ij} A_{ij}, \text{ para cualquier } j = 1, 2, \cdots, n,$$
  - $det(k.A) = k^n.det(A)$.
  - Si $det(A) \neq 0$, entonces A es inversible.
  - $det(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}) = ad - bc$

Si bien parece que existen $2n$ definiciones diferentes del $det(A)$, dependiendo de la columna o fila seleccionada, todas ellas llevan al mismo resultado numérico. Luego, es más conveniente calcular el $det(A)$ a lo largo de la fila o la columna con la mayor cantidad de ceros.

Se puede mostrar que la complejidad del cálculo del determinante de una matriz general $n \times n$ mediante esta definición es de $O(n!)$. Incluso para valores relativamente pequeños de n, el número de cálculo se vuelve difícil de manejar. Por lo tanto, en vez de utilizar la definición del determinante, se utilizan las siguientes propiedades:

i) Si cualquier fila o columna A sólo tiene entradas cero, entonces $\det A = 0$.

ii) Si A tiene dos filas o dos columnas iguales, entonces $\det A = 0$.

iii) Si $\tilde{A}$ se obtiene a partir de A mediante la operación $(E_i) \leftrightarrow (E_j)$, con $i \neq j$, entonces $\det \tilde{A} = -\det A$.

iv) Si $\tilde{A}$ se obtiene a partir de A mediante la operación $(\lambda E_i) \rightarrow (E_i)$, entonces $\tilde{A} = \lambda \det A$.

v) Si $\tilde{A}$ se obtiene a partir de A mediante la operación $(E_i + \lambda E_j) \rightarrow (E_i)$ con $i \neq j$, entonces $\det \tilde{A} = \det A$.

vi) Si B también es una matriz $n \times n$, entonces $\det AB = \det A \det B$.

vii) $\det A^t = \det A$.

viii) Cuando A^{-1} existe, $\det A^{-1} = (\det A)^{-1}$.

ix) Si A es una matriz triangular superior, triangular inferior o diagonal, entonces $A = \prod_{i=1}^{n} a_{ii}$. ■

La parte ix) del teorema 6.16 indica que el determinante de una matriz triangular es simplemente el producto de sus elementos diagonales. Al emplear las operaciones de fila dadas en las partes iii), iv) y v), podemos reducir una matriz cuadrada determinada a la forma triangular para encontrar su determinante.

Luego, podemos calcular el determinante en un orden cúbico.

2.5. Proposiciones equivalentes

Si A es una matriz $n \times n$, entonces las siguientes proposiciones son equivalentes:

- A es inversible.
- $Ax = 0$ solo vale para $x = 0$.
- $Ax = b$ tiene exactamente una solución para todo término independiente b.
- $det(A) \neq 0$.
- Las columnas de A son linealmente independientes.
- Las filas de A son linealmente independientes.
- El rango de A es n.
- La dimensión del núcleo de A es 0.
- $A^T A$ es inversible.
- $\lambda = 0$ no es autovalor de A.

Capítulo 3

Resolución de sistemas lineales

En este capítulo nos vamos a dedicar a **sistemas de ecuaciones lineales**, centrándonos en **algoritmos de resolución**. Vamos a analizar sus propiedades, tales como la eficiencia, el costo, la aplicabilidad, la inestabilidad numérica, etc. Comenzamos recordando lo que es un sistema de ecuaciones lineales.

La resolución de estos sistemas es un problema importante y frecuente en el análisis numérico, ya que estos son útiles a la hora de modelar matemáticamente el comportamiento de problemas provenientes de diversas disciplinas, como la física y la ingeniería, para ser tratados en forma computacional. En muchos de estos modelos aparecen ecuaciones que, o bien son lineales, o pueden aproximarse bien mediante ecuaciones lineales. Estos sistemas también aparecen en la resolución de ecuaciones diferenciales, que son cruciales para muchas disciplinas.

Un **sistema de ecuaciones lineales** es un conjunto de ecuaciones de la forma:

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

donde los $a_{i,j}$ y los b_i son números reales.

En particular nos vamos a restringir al caso en el que la cantidad de variables dadas en el sistema coincide con la cantidad de ecuaciones que tiene el sistema.

Si definimos a A como la **matriz asociada** al sistema, que no es otra cosa que la matriz que tiene los coeficientes de las ecuaciones, b al **término independiente**, que corresponde a los coeficientes que aparecen del lado derecho de las ecuaciones, y a x al vector de ecuaciones. Entonces resolver un sistema de ecuaciones no es otra cosa que resolver el sistema $Ax = b$:

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

Esta representación nos facilitará tanto su comprensión como su tratamiento computacional.

Las variables $x_1, \ldots, x_n$ se denominan las **incógnitas** del sistema. Una **solución** de un sistema de ecuaciones lineales es un conjunto de valores para las incógnitas que satisfacen simultáneamente todas las ecuaciones.

Un sistema de ecuaciones lineales puede no tener solución, tener solución única, o tener infinitas soluciones. Si la matriz asociada al sistema es inversible (o, lo que es lo mismo, sus columnas son lineal-

mente independientes) la solución será única. Si, por el contrario, la matriz es singular, podría pasar que el sistema no tenga solución o que tenga infinitas de ellas.

Un sistema de la forma $\mathbf{A} \cdot \mathbf{y} = \mathbf{0}$ se denomina **homogéneo**. Las soluciones de un sistema homogéneo forman un subespacio vectorial. Además, la totalidad del conjunto de soluciones de cualquier sistema $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ puede obtenerse obteniendo una solución particular del mismo, y luego sumarle a x cualquiera de las soluciones del sistema homogéneo asociado, pues $\mathbf{A}\cdot$(x + y)$= A \cdot x + A \cdot y = A \cdot x = b$.

Como nos estamos restringiendo al caso de igual cantidad de variables que de ecuaciones, la matriz $A \in \mathbb{R}^{n \times n}$, el vector $b \in \mathbb{R}^n$, y buscamos un vector $x \in \mathbb{R}^n$ tal que $Ax = b$.

Si recordamos algunos conceptos vistos en el repaso de Álgebra Lineal, cuando teníamos un sistema $Ax = b$, hacer el producto Ax no es otra cosa que una combinación lineal de las columnas de A:

$$Ax = b$$
$$Ax = a_1x_1 + a_2x_2 + \cdots + a_nx_n$$

Entonces, resolver el sistema de ecuaciones no es otra cosa que buscar la combinación lineal de las columnas de A que nos dé como resultado el vector b. Si recordamos, además, la relación entre las matrices y las transformaciones lineales, sabemos que encontrar un x tal que $Ax = b$ va a ser posible, únicamente, en caso de que $b \in Im(A)$. Si $b \notin Im(A)$, entonces no es posible hallar un x tal que $Ax = b$, y por lo tanto el sistema **no tiene solución**.

Por otro lado, si $b \in Im(A)$, entonces podemos escribir a b como una combinación lineal de las columnas de A. En el caso de que exista una única manera de escribir a b como combinación lineal de las columnas de A, la **solución va a ser única**. Si, en cambio, tenemos más de una manera de escribir a b, como combinación lineal de las columnas de A, entonces vamos a tener **infinitas soluciones**.

¿De qué va a depender esto? Va a depender del **rango** de la matriz A. Es decir, si las columnas de A son linealmente independientes, entonces la solución va a ser única. Si, en cambio, las columnas son linealmente dependientes, entonces vamos a tener infinitas soluciones.

Un concepto que vamos a recordar sobre los sistemas de ecuaciones son los **sistemas de ecuaciones equivalentes**. Son aquellos sistemas que tienen el mismo conjunto de soluciones:

$$\forall x \in \mathbb{R}^n,\ Ax = b \Longleftrightarrow Bx = d$$

3.1. Resolución de Sistemas Fáciles

Vamos a comenzar por resolver sistemas de ecuaciones que denominamos fáciles. El primer sistema de ecuaciones que vamos a considerar es aquel que tiene como matriz asociada una matriz diagonal:

$$A = D = \begin{bmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}$$

Entonces, ¿a qué tipo de sistema corresponde a una matriz de este estilo? Corresponde a un sistema en el cual en las distintas ecuaciones vamos a tener involucradas una única variable:

$$d_{11}x_1 = b_1$$
$$d_{22}x_2 = b_2$$
$$\vdots$$
$$d_{nn}x_n = b_n$$

Para resolver este sistema de ecuaciones, vamos a dividir el análisis en dos casos:

- **Caso 1**: Todos los elementos de la diagonal son distintos de cero:

$$
\begin{aligned}
x_1 &= \frac{b_1}{d_1 1} \\
x_2 &= \frac{b_2}{d_2 2} \\
&\vdots \\
x_n &= \frac{b_n}{d_n n}
\end{aligned}
$$

  De esta manera nos queda determinada la única solución que tiene el sistema, y es única porque, como los términos d_{ii} son todos distintos de 0, las columnas de A son linealmente independientes, y por lo tanto hay una única manera de escribir al vector b como combinación lineal de las columnas de A.

  Cuando hablamos de algoritmos, una de las propiedades que vamos a analizar es la cantidad de operaciones elementales (sumas, restas, divisiones, multiplicaciones) que están involucradas en el algoritmo. En este caso, la cantidad de operaciones elementales que tenemos son n cocientes, y por lo tanto el algoritmo tiene complejidad $O(n)$.

- **Caso 2**: existe algún elemento de la diagonal tal que $d_{ii} = 0$:

$$d_{ii}x_i = b_i, \text{ con } d_{ii} = 0$$

  - Si el término independiente $b_i \neq 0$, como x_i está multiplicada por un valor nulo, entonces no existe ningún valor para x_i que satisfaga esta ecuación, y por lo tanto podemos afirmar que el sistema no tiene solución.

  - Si, en cambio, el término independiente $b_i = 0$, cualquier valor que le demos a la variable x_i va a ser válido. Es decir, tenemos infinitas posibilidades.

  Por lo tanto, si, para todo el sistema de ecuaciones, ocurre que todos los términos nulos corresponden a términos independientes nulos, entonces el sistema tiene infinitas soluciones. Basta que exista un término nulo en la diagonal asociado a un término independiente no nulo para que el sistema no tenga solución.

El segundo caso que vamos a analizar, como caso fácil, es aquel caso en el que la matriz asociada al sistema al sistema es una matriz triangular superior:

$$U = \begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & * \end{bmatrix}$$

$$
\begin{gathered}
u_{11}x_1 + u_{12}x_2 + \cdots + u_{1n}x_n = b_1 \\
u_{22}x_2 + \cdots + u_{2n}x_n = b_2 \\
\vdots \\
u_{n-1n-1}x_{n-1} + u_{n-1n}x_n = b_{n-1} \\
u_{nn}x_n = b_n
\end{gathered}
$$

Al igual que hicimos con los sistemas diagonales, vamos a considerar dos casos:

- **Caso 1**: Todos los elementos de la diagonal son distintos de cero: Si recordamos que el determinante de una matriz triangular es el producto de los elementos de la diagonal, concluimos que el determinante va a ser distinto de 0, eso nos permite afirmar que la matriz es inversible, por lo tanto

sus columnas van a ser linealmente independientes, y por lo tanto la solución del sistema existe y es única. Veamos cómo lo podemos obtener:

$$
\begin{aligned}
x_n &= \frac{b_n}{u_{nn}} \\
x_{n-1} &= \frac{b_{n-1} - u_{n-1n}x_n}{u_{n-1n-1}} \\
&\vdots \\
x_1 &= \frac{b_1 - u_{12}x_2 \cdots - u_{1n}x_n}{u_{11}}
\end{aligned}
$$

De esta manera, hemos logrado identificar las n variables que intervienen en el sistema, de forma unívoca. Ahora, vamos a analizar la cantidad de operaciones involucradas en el algoritmo:

- En el primer paso, tenemos 1 cociente.
- En el segundo paso, tenemos 1 cociente, 1 producto, 1 resta.
- En el paso j-ésimo, correspondiente a la variable x_i, tenemos 1 cociente, $(n-i)$ productos, $(n-i)$ restas.
- En el último paso, tenemos 1 cociente, $(n-1)$ productos, $(n-1)$ restas.

Por lo tanto, el costo total en operaciones elementales del algoritmo es la suma de todas estas operaciones: n cocientes, $n(n-1)/2$ productos, $n(n-1)/2$ restas, y por lo tanto el algoritmo tiene complejidad $O(n^2)$. Este algoritmo se conoce como *Backward Substitution*, y si bien es más costoso con respecto al que corresponde a sistemas diagonales, sigue siendo relativamente barato.

- **Caso 2**: En caso de que alguno de los elementos de la diagonal sea nulo, no vamos a poder dividir por ese término, y entonces habrá que analizar si el sistema o bien tiene infinitas soluciones, o bien no tiene solución.

En caso de sistemas con una matriz asociada triangular inferior es simétrico al caso de las matrices triangulares superiores. En lugar de empezar por la última ecuación, se empieza por la primera, y el algoritmo se conoce como *Foward Substitution*.

3.2. Resolución de Sistemas Generales

3.2.1. Eliminación Gaussiana

Lo que tenemos hasta el momento son algoritmos para resolver sistemas diagonales, y para resolver sistemas triangulares. A continuación, vamos a abordar el caso de sistemas de ecuaciones donde la estructura de la matriz asociada al sistema no guarda ninguna particularidad, así que los llamamos **sistemas generales**. La idea va a ser construir un sistema equivalente cuya matriz asociada sea de las fáciles (triangular o diagonal), y como los sistemas equivalentes comparten el mismo conjunto solución, resolviendo estos sistemas fáciles, resolvemos el problema original.

¿Cómo hacemos para construir un sistema equivalente? Para transformar un sistema en otro equivalente, se aplica una serie de operaciones sobre las ecuaciones del mismo, que no modifican su conjunto de soluciones. Estas operaciones son las siguientes:

- Permutar el orden de las ecuaciones (multiplicar por una matriz de permutación: $P[A, b]$).
- Multiplicar ecuaciones por un escalar no nulo (multiplicar por una matriz elemental de tipo 1: $E_{t1}[A, b]$).
- Sumar/restar ecuaciones (multiplicar por una matriz elemental de tipo 2: $E_{t2}[A, b]$).

No se modifica el conjunto solución ya que las matrices que permiten realizar estas operaciones son inversibles, y por lo tanto $E \cdot Ax = E \cdot b \iff Ax = b$.

Basado en esta propiedad, se desarrolla el **Método de Eliminación Gaussiana**. Este consiste

en convertir un sistema de ecuaciones general, sin ninguna estructura, a un sistema equivalente cuya matriz asociada sea triangular superior. Una vez obtenida el sistema equivalente, se puede aplicar el procedimiento de Backward Substitution, y así encontrar las soluciones del sistema original.

El mismo opera sobre la **matriz aumentada** ($[A, b]$) del sistema, que es la matriz

$$A = \left.\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array}\right.$$

Como cada iteración efectúa cambios sobre la matriz $\tilde{\mathbf{A}}$, utilizaremos la notación $\tilde{\mathbf{A}}^{(k)}$ para referirnos al resultado luego de la k-ésima iteración del proceso, mientras que con $a_{ij}^{(k)}$ y $b_i^{(k)}$ haremos referencia a cada uno de sus elementos.

La idea del algoritmo es aplicar operaciones de filas en forma consecutiva hasta llevar $\tilde{\mathbf{A}}$ a una forma triangular superior. El método itera sobre las columnas de la matriz, buscando en cada paso colocar ceros en los lugares que se encuentran debajo de la diagonal. Es decir, en la k-ésima iteración, todas las columnas hasta la $k-1$ tienen ceros debajo de la diagonal, asegurando que tras $n-1$ iteraciones la matriz quedará en forma triangular superior.

En la k-ésima iteración se resta a las filas $k+1, \cdots, n$ un múltiplo de la fila k-ésima, con un factor $m_i^{(k)}$ correspondiente. Esto significa que, para todo $i = k+1, \dots, n$, los coeficientes de la fila i-ésima quedarán

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - m_i^{(k)} \cdot a_{kj}^{(k-1)},$$

y como se quiere dejar un 0 en la columna k-ésima, es decir, $a_{ik}^{(k)} = 0$, debe tomarse, para cada fila i, el multiplicador

$$m_{ik} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}$$

Es importante notar que solo es posible efectuar el k-ésimo paso del algoritmo si $a_{kk}^{(k-1)} \neq 0$. Si $a_{kk}^{(k-1)} = 0$, el algoritmo falla.

Como cada paso del algoritmo coloca ceros debajo de la diagonal en la columna k-ésima, y no modifica los ceros que fueron ubicados en otras columnas por los pasos previos, la matriz $\tilde{\mathbf{A}}^{(n-1)}$ que se obtiene tras $n-1$ iteraciones del proceso es triangular superior.

A continuación se presenta el algoritmo en forma de pseudocódigo:

Algoritmo de Eliminación Gaussiana

Entrada: $\mathbf{A} \in \mathbb{R}^{n \times n}$ y $\mathbf{b} \in \mathbb{R}^n$.
Salida: $\mathbf{x} \in \mathbb{R}^n$ tal que $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$.

```
for k = 1, ..., n-1 do
    if a_kk ≠ 0 then
        for i = k+1, ..., n do
            m_ik ← a_ik / a_kk
            F_i ← F_i − m_ik · F_k
    else
        fallar
```

donde F_i se corresponde con la i-ésima fila de la matriz ampliada del sistema. El costo total del algoritmo es :

$$\text{Costo } = \sum_{i=1}^{n-1} (n-i)(n-i+2) \text{ p } + (n-i)(n-i+2) \text{ r } + (n-i) \text{ c}$$

donde p = producto, r = resta, c = cociente. Por lo tanto, su complejidad es aproximadamente $O(\frac{n^3}{3})$.

3.2.2. EG con pivoteo

Ahora bien, dada una matriz, puede ocurrir que en la k-ésima iteración nos aparezca un elemento nulo en la posición a_{kk}, y por lo tanto no se cumpla la condición necesaria ¿qué podemos hacer en ese caso? Vamos a analizar:

$$\begin{bmatrix} * & * & \cdots & * & \cdots & * \\ 0 & * & \cdots & * & \cdots & * \\ 0 & 0 & * & \cdots & \cdots & * \\ \vdots & \vdots & \ddots & 0 & * & \cdots \\ \vdots & \vdots & \ddots & * & * & \cdots \\ \vdots & \vdots & \ddots & * & * & \cdots \end{bmatrix}$$

Veamos qué dos posibilidades pueden ocurrir:

- **Caso 1**: Una primera posibilidad es que nos encontremos, efectivamente, con un término nulo en a_{kk}, pero que el resto de los elementos ($a_{k+1}k$ en adelante) de esa columna también sean nulos. Recordando que el objetivo de este paso es lograr ceros en la columna, al ya tener la columna llena de ceros, no hay necesidad de efectuar ese paso, y se puede pasar a la siguiente columna.

- **Caso 2**: Si nos encontramos con un elemento no nulo en el resto de la columna, no podemos pasar al siguiente paso. Sin embargo, sabemos que si a un sistema de ecuaciones permutamos el orden de las filas, el sistema que obtenemos es un sistema equivalente. Entonces, podemos pensar en realizar una permutación entre la fila k y una fila j (con $j > k$) que tenga un elemento no nulo en la columna k, obteniendo un sistema equivalente con un elemento no nulo en la posición $\tilde{A}_{kk}$, pudiendo continuar con el algoritmo.

  En ambos casos pudimos solucionar el problema, y por lo tanto toda matriz A admite resolución aplicando Eliminación Gaussiana con permutaciones.

Cuando se busca implementar la Eliminación Gaussiana, hay que tener en cuenta que en la computadora se trabaja con aritmética finita. Cuando se trabaja con aritmética finita, las operaciones elementales pueden presentar errores. Entonces, es deseable que el algoritmo tratara evitar algunos errores que pueden ser significativos, propiedad que se conoce como *estabilidad numérica*. Por ejemplo, en aritmética finita, si se realiza se divide por un número chico, se incrementa el error absoluto en las operaciones, lo cual no es deseable.

Supongamos que queremos calcular la factorización LU de la matriz A

$$A = \begin{bmatrix} \epsilon & -1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ l_{21} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}, \qquad 0 < \epsilon \ll 1.$$

Si aplicamos el algoritmo de eliminación gaussiana sin pivoteo, la factorización LU nos queda $u_{11} = \epsilon, u_{12} = -1, l_{21} = \epsilon^{-1}, u_{22} = 1 + \epsilon^{-1}$. Sin embargo, cuando trabajamos con aritmética de punto flotante,

si ϵ es lo suficientemente chico, entonces $\hat{u}_{22} = fl(1 + \epsilon^{-1}$ se evalúa a $\epsilon^{-}1$, ya que $\epsilon^{-1} \gg 1$. Asumiendo que l_{21} es computado de forma exacta, entonces nos queda

$$\begin{aligned} A - \hat{L}\hat{U} &= \begin{bmatrix} \epsilon & -1 \\ 1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ \epsilon^{-1} & 1 \end{bmatrix} \begin{bmatrix} \epsilon & -1 \\ 0 & \epsilon^{-1} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Por lo tanto, la factorización LU computada falla en reproducir a A. Notemos que la matriz A está muy bien condicionada ($\kappa_\infty(A) = \frac{4}{1+\epsilon}$), por lo que no es un problema de que el sistema esté mal condicionado. El problema es la elección de ϵ como *pivote*. Luego, podemos concluir que es necesario tener algunas estrategias que nos permitan reducir este tipo de escenarios en los que se compromete la solución del sistema.

En cada paso del algoritmo de Eliminación Gaussiana, llamamos **pivote** al elemento de la diagonal sobre el cual estamos trabajando (en el paso k-ésimo, el pivote es $a_{k,k}^{(k-1)}$). La técnica de **pivoteo** consiste en realizar operaciones sobre la matriz, intercambiando sus filas (o sus columnas) para modificar el pivote sin alterar las soluciones del sistema asociado.

La idea es la siguiente:

- Si lo que molesta es dividir por números chicos, en el paso k-ésimo podemos permutar por aquella fila j-ésima (con $j > k$, y $a_{jk}^{k-1} \neq 0$) que tenga coeficiente más grande en módulo. Esta estrategia se conoce como **pivoteo parcial**.

  El **pivoteo parcial** consiste en intercambiar el pivote por un elemento de la misma columna, considerando el propio pivote y los elementos que se encuentran por debajo de él, y eligiendo el de mayor valor absoluto. Por lo tanto, se lleva a cabo intercambiando dos filas de la matriz. Esta técnica solo requiere considerar, a lo sumo, n posibles valores para el pivote. Garantiza que se elegirá un pivote no nulo (a menos que el elemento de la diagonal y todos los que estén debajo sean nulos), y permite mejorar la estabilidad numérica.

- Otra estrategia, conocida como **pivoteo completo**, considera toda la submatriz que falta reducir, eligiendo como pivote al elemento de mayor valor absoluto. Se lleva a cabo intercambiando dos filas y dos columnas de la matriz (intercambiar columnas equivale a alterar el orden de las variables del sistema, por lo que los intercambios de columnas deberán ser revertidos en la solución que se obtenga).

  Esta técnica permite mejorar aún más la estabilidad numérica, pero es poco utilizada por resultar considerablemente menos eficiente, ya que la búsqueda del pivote tiene una complejidad cuadrática.

Estas estrategias, de ninguna manera, aseguran evitar todos los problemas asociados a trabajar con aritmética finita. Son estrategias que tratan de reducir estas complicaciones, pero no las anulan.

Capítulo 4

Factorización LU

En este capítulo vamos a derivar la factorización LU de una matriz, y vamos a ver cómo se puede utilizar para resolver sistemas de ecuaciones lineales, de forma eficiente.

Hasta el momento, la herramienta que tenemos para resolver un sistema de ecuaciones lineales es la Eliminación Gaussiana, el cual consiste en transformar al sistema general a uno equivalente cuya matriz asociada sea triangular superior, mediante una cantidad de operaciones en orden cúbico, y luego resolver ese sistema triangular usando *Backward Substitution* (orden cuadrado).

Recordemos que la EG se aplica a la matriz aumentada ($[A, b]$), y por lo tanto el resultado depende no solo de A, sino que también del término independiente b. Luego, si nos llegaran a plantear otro sistema de ecuaciones en el cual, únicamente, se cambia el término independiente, nos vemos en la obligación de tener que realizar la EG desde el inicio. Es decir, por cada sistema de ecuaciones que se plantee, siempre vamos a tener un costo cúbico para llegar al sistema triangular, para luego resolverlo con costo cuadrático.

Luego, la factorización LU busca evitar tener ese costo cúbico por cada sistema de ecuaciones que se plantee. ¿Qué es la factorización LU? Consiste en tener escrita a la matriz A como el producto de una matriz triangular inferior (L) por una matriz triangular superior (U):

$$A = LU$$

¿Por qué nos va a ser útil esta factorización? Si se tiene el sistema $Ax = b$, y conocemos la factorización LU de la matriz A, entonces podemos reescribir este sistema como $LU \cdot x = b$. Si denotamos a $Ux = y$, y resolvemos primero un sistema $Ly = b$, y después un segundo sistema $Ux = y$, entonces este último x va a ser solución del sistema original, pues

$$\left.\begin{array}{r} Ly = b \\ Ux = y \end{array}\right\} \Longrightarrow L(Ux) = b \Longleftrightarrow Ax = b$$

¿Cuál fue la ventaja? Partimos de un sistema general, sin ninguna estructura en particular, a tener que resolver dos sistemas de ecuaciones, ambos triangulares, que se pueden resolver con un costo cuadrático, en vez de un costo cúbico. Nos falta ver de cómo podemos obtener la factorización LU de A. Para eso, vamos a pensar en el proceso de Eliminación Gaussiana.

4.1. Buscando la factorización LU

Vamos a suponer que se puede realizar el proceso de EG sin tener que realizar ninguna permutación de filas (siempre nos encontramos con un elemento no nulo en el pivote).

Vamos a considerar una matriz elemental de tipo 2, que tenga la siguiente pinta:

$$E = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -m_{21} & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

donde m_{21} hace referencia al multiplicador que utilizamos en la eliminación gaussiana.

Vamos a considerar a la matriz original A, a la cual vamos a multiplicar por izquierda por esta matriz elemental, y lo que queremos ver es que esta multiplicación, en realidad, lo que va a estar realizando es que a la fila 2 le resta $m_{21}F_1$, sin modificar al resto de las filas:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -m_{21} & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} a_{11}^0 & a_{12}^0 & \cdots & a_{1j}^0 & \cdots & a_{1n}^0 \\ a_{21}^0 & a_{22}^0 & \cdots & a_{2j}^0 & \cdots & a_{2n}^0 \\ a_{31}^0 & a_{32}^0 & \cdots & a_{3j}^0 & \cdots & a_{3n}^0 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{n1}^0 & a_{n2}^0 & \cdots & a_{nj}^0 & \cdots & a_{nn}^0 \end{bmatrix} =$$

$$\begin{bmatrix} a_{11}^0 & a_{12}^0 & \cdots & a_{1j}^0 & \cdots & a_{1n}^0 \\ a_{21}^0 - m_{21}a_{11}^0 & a_{22}^0 - m_{21}a_{12}^0 & \cdots & a_{2j}^0 - m_{21}a_{1j}^0 & \cdots & a_{2n}^0 - m_{21}a_{1n}^0 \\ a_{31}^0 & a_{32}^0 & \cdots & a_{3j}^0 & \cdots & a_{3n}^0 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{n1}^0 & a_{n2}^0 & \cdots & a_{nj}^0 & \cdots & a_{nn}^0 \end{bmatrix}$$

Luego, con esta matriz elemental, podemos realizar el cálculo de $F_2 - m_{21}F_1$. Luego, nos gustaría poder encontrar una matriz que nos permita expresar el primer paso de la eliminación gaussiana en forma matricial.

Si colocamos todos los multiplicadores del primer paso en la primer columna de una matriz M^1, completando con la identidad, obtenemos una matriz que permite expresar matricialmente el primer paso de la EG:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ -m_{21} & 1 & 0 & \cdots & 0 & \cdots & 0 \\ -m_{31} & 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ -m_{i1} & 0 & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ -m_{n1} & 0 & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} a_{11}^0 & a_{12}^0 & \cdots & a_{1j}^0 & \cdots & a_{1n}^0 \\ a_{21}^0 & a_{22}^0 & \cdots & a_{2j}^0 & \cdots & a_{2n}^0 \\ a_{31}^0 & a_{32}^0 & \cdots & a_{3j}^0 & \cdots & a_{3n}^0 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{i1}^0 & a_{i2}^0 & \cdots & a_{ij}^0 & \cdots & a_{in}^0 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{n1}^0 & a_{n2}^0 & \cdots & a_{nj}^0 & \cdots & a_{nn}^0 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}^0 & a_{12}^0 & \cdots & a_{1j}^0 & \cdots & a_{1n}^0 \\ a_{21}^0 - m_{21}a_{11}^0 & a_{22}^0 - m_{21}a_{12}^0 & \cdots & a_{2j}^0 - m_{21}a_{1j}^0 & \cdots & a_{2n}^0 - m_{21}a_{1n}^0 \\ a_{31}^0 - m_{31}a_{11}^0 & a_{32}^0 - m_{31}a_{12}^0 & \cdots & a_{3j}^0 - m_{31}a_{1j}^0 & \cdots & a_{3n}^0 - m_{31}a_{1n}^0 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{i1}^0 - m_{i1}a_{11}^0 & a_{i2}^0 - m_{i1}a_{12}^0 & \cdots & a_{ij}^0 - m_{i1}a_{1j}^0 & \cdots & a_{in}^0 - m_{i1}a_{1n}^0 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{n1}^0 - m_{n1}a_{11}^0 & a_{n2}^0 - m_{n1}a_{12}^0 & \cdots & a_{nj}^0 - m_{n1}a_{1j}^0 & \cdots & a_{nn}^0 - m_{n1}a_{1n}^0 \end{bmatrix}$$

Obs: esta matriz es triangular inferior.

Esta matriz recibe el nombre de **primera matriz de transformación gaussiana**. Luego, si ahora consideramos una matriz M^i en la que colocamos en la columna i los multiplicadores del paso i-ésimo de la EG, completando con la identidad, entonces obtenemos una matriz que nos permite expresar de forma matricial el i-ésimo paso de Gauss:

$$\begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & & \vdots & \cdots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -m_{i+1i} & 1 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -m_{ni} & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} a_{11}^{i-1} & \cdots & a_{1i}^{i-1} & \cdots & a_{1n}^{i-1} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ 0 & \cdots & a_{ii}^{i-1} & \cdots & a_{in}^{i-1} \\ 0 & \cdots & a_{i+1i}^{i-1} & \cdots & a_{i+1n}^{i-1} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ 0 & \cdots & a_{ni}^{i-1} & \cdots & a_{nn}^{i-1} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}^{i-1} & a_{12}^{i-1} & \cdots & a_{1i}^{i-1} & \cdots & a_{1n}^{i-1} \\ 0 & a_{22}^{i-1} & \cdots & a_{2i}^{i-1} & \cdots & a_{2n}^{i-1} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & a_{ii}^{i-1} & \cdots & a_{in}^{i-1} \\ 0 & 0 & \cdots & a_{i+1i}^{i-1} - m_{i+1i}a_{ii}^{i-1} & \cdots & a_{i+1n}^{i-1} - m_{i+1i}a_{in}^{i-1} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & a_{ni}^{i-1} - m_{ni}a_{ii}^{i-1} & \cdots & a_{nn}^{i-1} - m_{ni}a_{in}^{i-1} \end{bmatrix}$$

Luego, si asumimos que $a_{ii}^{(i-1)} \neq 0$, podemos expresar todos los pasos de la EG de forma matricial de la siguiente manera:

$$M^{n-1}M^{n-2}\cdots M^1 A = U \text{ con } U \text{ triangular superior}$$

$$\text{con } M^i = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -m_{i+1i} & 1 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -m_{ni} & 0 & \cdots & 1 \end{bmatrix}$$

Con esto obtuvimos una expresión matricial de la Eliminación Gaussiana.

Recordemos que nuestro objetivo es obtener la factorización LU de A. Hasta ahora sabemos cómo obtener la U (aplicando EG sobre A), y podemos pensar que si obtenemos la inversa del producto de matrices M^i habremos obtenido también la L, considerando que:

- M^i son triangulares inferiores
- El producto de triangulares inferiores es triangular inferior, por lo que el producto de M^i es triangular inferior.
- La inversa de una triangular inferior también es triangular inferior, por lo que la inversa del producto de M^i también es triangular inferior.

$$M^{n-1}M^{n-2}\cdots M^1 A = U$$

$$A = (M^1)^{-1} \cdots (M^{n-2})^{-1}(M^{n-1})^{-1} \cdot U$$

Sin embargo, calcular $N-1$ inversas para obtener la factorización resulta demasiado costoso, por lo que necesitamos poder decir más cosas sobre estas matrices para encontrar una forma eficiente de calcular L.

Estas matrices M^i son matrices muy particulares. Por un lado, son triangular inferior, con todos los elementos de la diagonal iguales a 1, por lo que son inversibles. Además son muy parecidas a la identidad, salvo en la columna i-ésima, por lo que podemos reescribirlas de la siguiente manera:

$$M^i = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -m_{i+1i} & 1 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -m_{ni} & 0 & \cdots & 1 \end{bmatrix} = I - \begin{bmatrix} 0 \\ \vdots \\ m_{i+1i} \\ \vdots \\ m_{ni} \end{bmatrix} \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}$$

$$M^i = I - m_i^t e_i$$

con $m_i = (0, \ldots, m_{i+1i}, \cdots, m_{ni})$ y e_i el i-ésimo vector canónico.

En base a esto, buscaremos caracterizar la inversa de M^i, que ya sabemos que existe. Supongamos que $I + m_i^t e_i$ es la inversa de M^i. Para comprobar que esto es así, veamos que $M^i \cdot (I + m_i^t e_i) = I$:

$$(I - m_i^t e_i)(I + m_i^t e_i) = I + m_i^t e_i - m_i^t e_i - m_i^t e_i m_i^t e_i = I - m_i^t e_i m_i^t e_i$$

$$\text{pero} \quad e_i m_i^t = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ m_{i+1i} \\ \vdots \\ m_{ni} \end{bmatrix} = 0$$

Entonces $(I - m_i^t e_i)(I + m_i^t e_i) = I \Longrightarrow (I - m_i^t e_i)^{-1} = I + m_i^t e_i$

Con esto conseguimos caracterizar las inversas de las M^i, por lo que reemplazando nos queda:

$$A = (I + m_1^t e_1)(I + m_2^t e_2) \cdots (I + m_{n-1}^t e_{n-1}) \cdot U$$

Ahora, si desarrollamos este producto, nos queda:

$$A = (I + m_1^t e_1 + m_2^t e_2 \cdots + m_{n-1}^t e_{n-1}) \cdot U$$

y si lo expresamos de forma matricial:

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ m_{21} & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ m_{i1} & m_{i2} & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & \cdots & \cdots & 1 \end{bmatrix} \cdot U$$

Ahora, la construcción de esta matriz triangular inferior nos viene gratis, porque lo único que tenemos que hacer para construirla es poner debajo de cada una de las columnas los multiplicadores que utilizamos para realizar los distintos pasos de la Eliminación Gaussiana, y con esto obtuvimos la factorización LU de la matriz A.

4.2. Propiedades

Cuando hablamos de factorización LU, siempre vamos a estar pensando en la matriz L y la matriz U que surgen de la eliminación gaussiana, y por lo tanto la matriz L siempre va a verificar que:

- los elementos de la diagonal valen 1.
- por debajo de la diagonal están los multiplicadores que se utilizan durante el proceso de EG.

, y la matriz U es la matriz triangular superior, asociada a un sistema equivalente al sistema de la matriz A, que surge de aplicar EG sobre A.

No toda matriz tiene factorización LU, pues es necesario que sea posible realizar el proceso de EG sin permutaciones. Es un error muy común pensar que si la matriz es inversible, siempre vamos a poder hallar la factorización LU, y eso no es cierto. Un contraejemplo sencillo es:

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

que es inversible y no tiene factorización LU (ni siquiera podemos hacer el primer paso de Gauss). La factorización LU está estrechamente asociada a poder realizar la eliminación gaussiana.

- Vimos que A sea inversible no sirve para saber si tiene factorización LU. Nos gustaría alguna propiedad que nos asegure la existencia de la misma:
  - Si y solo si A tiene todas sus submatrices principales inversibles, entonces tiene factorización LU. Las submatrices principales son aquellas submatrices formadas a partir de las primeras k filas y las primeras k columnas.
  - Si bien la propiedad anterior asegura la existencia de la factorización LU, y por lo tanto es una propiedad teórica importante, pero no es del todo práctica (es demasiado costosa de verificar). Por lo tanto, nos gustaría una propiedad más sencilla para poder afirmar la existencia de la misma.

    Existen matrices que, por las características que tienen, sí podemos afirmar que tienen factorización LU, y además sus características son fáciles de comprobar. Entre estas matrices se encuentran las matrices **estrictamente diagonal dominante**, es decir que $|a_{ii}| > \sum_{j \neq i} a_{ij}$, $\forall i = 1, \cdots, n$). La idea es que las matrices *edd* son inversibles, y como toda submatriz principal de una matriz *edd* también es *edd*, en particular, también es inversible, y por lo tanto la tienen factorización LU.
- Si una matriz s inversible y tiene factorización LU, entonces la factorización LU es **única**.

4.3. Factorización PLU

¿Qué es lo que ocurre cuando encontrábamos un elemento nulo en la EG? Sabemos que la Eliminación Gaussiana puede continuar mediante permutaciones de filas. Sin embargo, por la definición que hemos hecho de la factorización LU sabemos que, en caso de ser necesaria la permutación, esta no existe.

Sin embargo, si se realizan todos los intercambios de filas requeridos por adelantado, entonces el algoritmo de eliminación puede continuar sin requerir de más permutaciones. Es decir, va a existir lo que llamamos la factorización PLU, es decir la factorización LU de la matriz original permutada:

$$PA = LU$$

donde P es la matriz de permutación resultante de aplicar eliminación gaussiana con permutaciones, L es una matriz triangular inferior con 1s en la diagonal, y U es una matriz triangular superior con los elementos pivotes en la diagonal.

La matriz P realizará un seguimiento de las permutaciones realizadas durante el proceso de eliminación gaussiana. Es decir, cada vez que se intercambian dos filas de A, se intercambiarán las mismas dos filas de P.

Una vez que se establece la factorización PLU, la solución al sistema original $Ax = b$ se obtiene aplicando el mismo algoritmo de *Backward* y *Foward substitution* presentado anteriormente. Explícitamente, primero multiplicamos el sistema $Ax = b$ por la matriz de permutación, lo que lleva a

$$\begin{aligned} PA \cdot x &= Pb \\ LU \cdot x &= Pb \end{aligned}$$

y luego resolvemos ambos sistemas triangulares, como hicimos anteriormente

$$\begin{aligned} Ly &= Pb \\ Ux &= y \end{aligned}$$

Notemos que, como todo sistema de ecuaciones se puede resolver mediante la EG con permutaciones, todas las matrices tienen factorización PLU. El costo de obtener la factorización PLU es de orden cúbico. Además, esta forma de obtener una factorización LU permite realizar pivoteo durante la eliminación gaussiana, en busca de reducir el error numérico.

Capítulo 5

Normas vectoriales y matriciales, y Número de condición

En este capítulo vamos a analizar la sensibilidad de un sistema de ecuaciones cuando modificamos algún valor en los coeficientes de la matriz o del término independiente. Queremos ver cómo varía la solución frente a cambios en los datos de entrada del problema.

Tener una forma de medir la **distancia** entre vectores y enre matrices nos va a permitir analizar la convergencia de los métodos iterativos que resuelvan sistemas lineales, y determinar si una matriz está mal condicionada, permitiendo reconocer este problema, y así evitar obtener soluciones erróneas al aplicar métodos directos (como la Eliminación Gaussiana con o sin pivoteo).

Recordemos que a la hora de resolver problemas que involucran números reales utilizando la computadora, siempre debe tenerse en cuenta que el abordaje de los mismos es **numérico**, es decir, opera tan solo con aproximaciones de los números reales, dentro de lo permitido por la aritmética finita de la computadora.

Esto produce, inevitablemente, errores de redondeo que pueden ocasionar pérdida de exactitud en los resultados. Por lo tanto, es importante tener cuidado en evitar que dichos errores se propaguen de formas no deseadas.

5.1. Normas Vectoriales

Para hacer esto, nos va a resultar muy útil recordar algunos conceptos del álgebra lineal, y vamos a comenzar con las normas vectoriales. Las normas vectoriales son funciones definidas en $\mathbb{R}^n$, que toman valores reales, y que cumplen con las siguientes propiedades:

$f : \mathbb{R}^n \longrightarrow \mathbb{R}$ es una norma sii:

- $f(x) > 0$ si $x \neq 0$.
- $f(x) = 0 \Longleftrightarrow x = 0$.
- $f(\alpha x) = |\alpha| f(x) \, \forall \alpha \in \mathbb{R}$
- $f(x + y) \leq f(x) + f(y)$ (desigualdad triangular).

Cualquier función, definida en $\mathbb{R}^n$, que toman valores reales, y cumplan con estas cuatro propiedades es una norma vectorial. Veamos algunas de ellas:

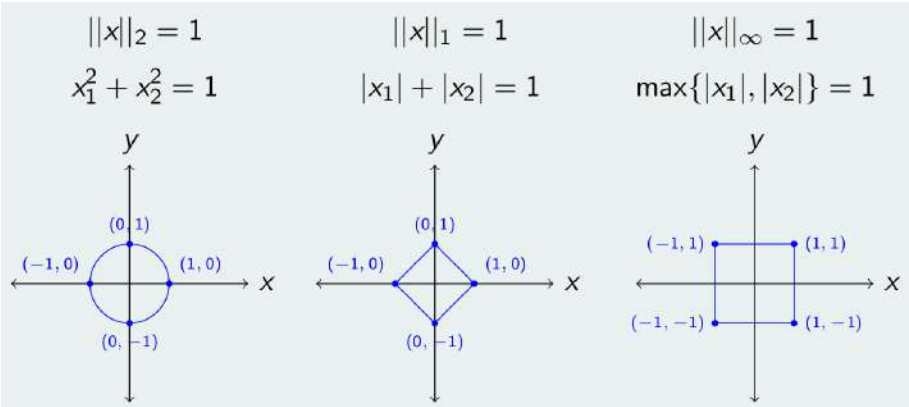
- Norma 2 o Norma Euclídea: $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$

- Norma 1: $\|x\|_1 = \sum_{i=1}^{n} |x_i|$

- Norma p (generalización de norma 1 y 2): $\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$

- Norma Infinito: $\|x\|_\infty = \max_{i=1,\cdots,n} |x_i|$

Dada una norma, existe una región del espacio que se caracteriza por tener el valor de esa norma igual a 1, a la que llamaremos **Circunferencia de radio 1**. Esta región del espacio nos va a resultar útil, más adelante, para algunas definiciones. Si pensamos en $\mathbb{R}^2$, la región en el espacio tendría la siguiente pinta:



5.2. Normas Matriciales

Así como existen las normas vectoriales, también tenemos las **normas matriciales**. En este caso, la función debe estar definida en el espacio de las matrices $\mathbb{R}^{m\times n}$, toma valores reales y va a ser una norma si y solo si cumple con las siguientes propiedades:

- $F(A) > 0$, $\forall A \neq 0$.

- $F(A) = 0$ sii $A = 0$.

- $F(\alpha A) = |\alpha| F(A)$, $\forall \alpha \in \mathbb{R}$.

- Desigualdad triangular: $F(A + B) \leq F(A) + F(B)$.

  Dentro del conjunto de normas matriciales, existe un subconjunto que cumple con una propiedad adicional, y que se las conoce como normas sub-multiplicativas, que están definidas para el caso $m = n$, y la propiedad dice que:

- $F(AB) \leq F(A)F(B)$.

  Trabajar con normas que cumplan esta propiedad adicional nos va a resultar muy útil.

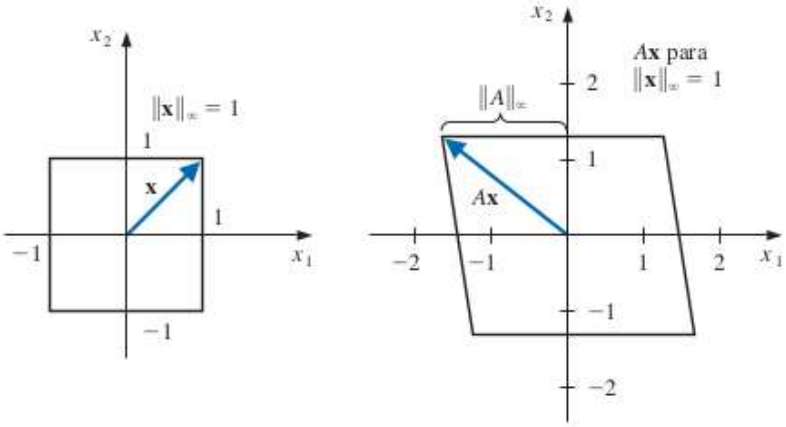
Algunos ejemplos de normas matriciales son:

- **Norma de Frobenius**: $\|A\|_F = \sqrt{\left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2\right)}$

- **Norma M**: $\|A\|_M = \max_{i,j} |a_{ij}|$.

Dentro de las normas matriciales tenemos un subconjunto de normas llamadas **normas matriciales**

inducidas o naturales. Si consideramos la transformación lineal asociada a la matriz A, $T : \mathbb{R}^n \longleftarrow \mathbb{R}^m$, tenemos un vector en $\mathbb{R}^n$ cuya imagen está en $\mathbb{R}^m$, y la norma inducida busca relacional la norma del vector con la de su imagen.



En particular, la norma matricial inducida busca la máxima alteración relativa en la norma de un vector, al aplicarle la transformación asociada a la matriz A:

$$\|A\| = \operatorname*{máx}_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

Una definición equivalente consiste en considerar la región dentro del dominio de la transformación lineal asociada a la matriz A cuya norma vectorial sea igual a 1, para luego buscar en la imagen de esa región aquel vector de máxima norma:

$$\operatorname*{máx}_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \operatorname*{máx}_{x \neq 0} A\Big(\frac{z}{\|z\|}\Big) = \operatorname*{máx}_{\|z\|=1} \|Az\|$$

Algunos ejemplos para $n = m$:

- Norma 1: $\|A\|_1 = \operatorname*{máx}_{x:\|x\|_1=1} \|Ax\|_1$
- Norma 2: $\|A\|_2 = \operatorname*{máx}_{x:\|x\|_2=1} \|Ax\|_2$
- Norma p: $\|A\|_p = \operatorname*{máx}_{x:\|x\|_p=1} \|Ax\|_p$
- Norma ∞: $\|A\|_\infty = \operatorname*{máx}_{x:\|x\|_\infty=1} \|Ax\|_\infty$

Una de las ventajas que tiene la norma infinito, la norma 1, y la norma 2 es que tienen una fórmula cerrada para calcularla:

$$\|A\|_\infty = \operatorname*{máx}_{i=1,\cdots,n} \sum_{j=1}^{n} |a_{ij}| = \operatorname*{máx}_{i=1,\cdots,n} \|a_i^t\|_1, \text{ siendo } a_i^t \text{ la fila i-ésima de A.}$$

$$\|A\|_1 = \operatorname*{máx}_{j=1,\cdots,n} \sum_{i=1}^{n} |a_{ij}| = \operatorname*{máx}_{j=1,\cdots,n} \|a_i\|_1, \text{ siendo } a_i \text{ la columna i-ésima de A.}$$

$$\|A\|_2 = \sigma_1 \text{ (lo veremos más adelante)}$$

5.3. Número de condición

Las normas matriciales nos brindan herramientas para caracterizar sistemas de ecuaciones problemáticos, donde pequeños errores numéricos pueden magnificarse y producir soluciones considerablemente inexactas. Los sistemas que presentan este tipo de inconvenientes se dice que están **mal condicionados**.

Es fácil visualizar qué causa que un sistema 2×2 esté mal condicionado. Geométricamente, dos ecuaciones con dos incógnitas representan dos líneas rectas, y el punto de intersección es la solución para el sistema. Un sistema mal condicionado representa dos líneas rectas que son casi paralelas.

Si dos líneas rectas son casi paralelas y si una de las líneas está inclinada solo ligeramente, entonces el punto de intersección (es decir, la solución del sistema lineal 2×2 asociado) se modifica drásticamente.

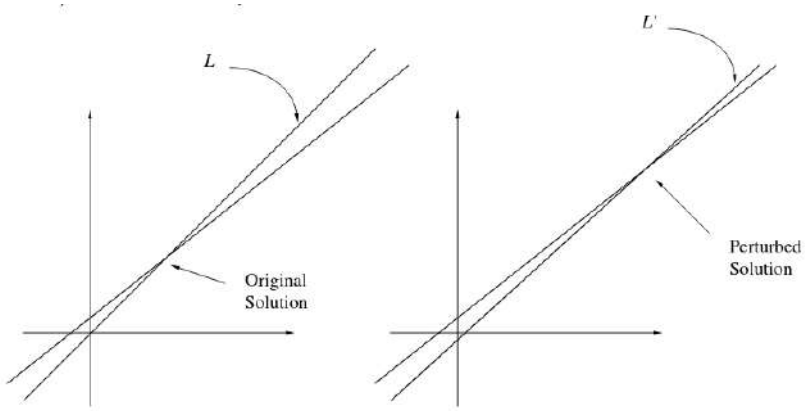


Figura 5.1: Sistema de ecuaciones mal condicionado.

Debido a que los errores de redondeo pueden verse como perturbaciones de los coeficientes originales del sistema, emplear incluso una técnica numérica generalmente buena (muy cerca de ser aritmética exacta) en un sistema mal condicionado conlleva el riesgo de producir resultados sin sentido.

El problema surge del hecho de que los coeficientes se obtienen, a menudo, empíricamente y, por lo tanto, solo se conoce que se encuentran dentro de un cierto rango. Para un sistema mal acondicionado, una pequeña perturbación en cualquiera de los coeficientes puede significar que puede existir una perturbación extremadamente grande en la solución. Este hecho vuelve tan poco confiable a los resultados obtenidos que incluso obtener la solución exacta sea totalmente inútil.

La siguiente proposición formaliza esta idea intuitiva.

Sea $A \in \mathbb{R}^{n \times n}$ inversible. Sea x^* solución de $Ax = b$. Sea $\tilde{x}$ solución de $Ax = \tilde{b}$. Si $\|\cdot\|$ es una norma inducida cualquiera, entonces

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

Notemos que en esta definición queda implícito que una matriz debe ser inversible para poder calcular su número de condición.

Este número de condición va a jugar un papel fundamental cuando estudiemos la sensibilidad de un sistema, es decir cómo puede cambiar la solución de un sistema de ecuaciones al modificar alguno de los términos.

Para relacionar el error relativo entre x^* y $\tilde{x}$ con los cambios relativos en el término independiente aparece la siguiente propiedad:

$$\frac{\|\mathbf{x}^* - \tilde{\mathbf{x}}\|}{\|\mathbf{x}^*\|} \leq \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{b}\|} \cdot \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\|.$$

Notemos que estos errores se relacionan justamente con el número de condición de la matriz A.

Esta desigualdad nos dice que, si tenemos pequeños cambios relativos en el término independiente, se puede esperar tener pequeños cambios relativos en el vector solución, siempre y cuando el número de condición de la matriz A sea chico. Si, en cambio, el número de condición es muy grande, entonces no podemos asegurar que pequeños en el b implican pequeños cambios en el x. El concepto de cambios "pequeños" va a depender del contexto en el que estemos trabajando.

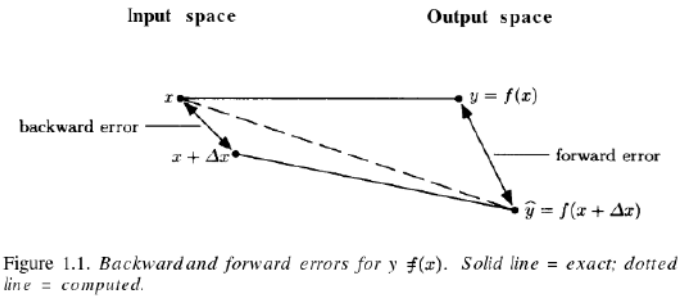


Figura 5.2: Sistema de ecuaciones mal condicionado.

En general, para resolver un mismo problema vamos a tener varios sistemas que nos permiten encontrar la solución, de los cuales algunos van a estar bien condicionados y otros no.

5.4. Propiedades Varias

Normas Vectoriales:

- $\|x\|_\infty \leq \|x\|_1$, $\|x\|_2 \leq n\|x\|_\infty$. Obs: **No vale para normas matriciales**.
- **Desigualdad C-S**: $|x^T y| \leq \|x\|_2 \|y\|_2$.
- $\|x\|_2^2 = x^t x$.

Normas Matriciales:

- $\|A\|_M \leq \|A\|_2 \leq n\|A\|_M$.
- Si se trata de una norma inducida, entonces:
  - $\|I\| = 1$.
  - $\|Ax\| \leq \|A\|\|x\|$.
  - $\|AB\| \leq \|A\|\|B\|$.

Número de condición

- Si $\|.\|$ es inducida, $\kappa(I) = 1$.
- Si $\|.\|$ es sub-multiplicativa, $\kappa(A) \geq 1$
- También podemos relacionar el error absoluto entre x^* y $\tilde{x}$ con los cambios absolutos en el término independiente mediante la siguiente desigualdad: $\|x^* - \tilde{x}\| \leq \|b - \tilde{b}\|\|A^{-1}\|$.

Capítulo 6

Matrices SDP

Este capítulo está dedicado a las matrices **simétricas definidas positivas**, que son matrices muy particulares, para las cuales vamos a demostrar que la eliminación gaussiana no tiene inconvenientes durante su proceso, por lo que va a existir la factorización LU, y además vamos a caracterizar ciertas propiedades de esa factorización LU.

Empezamos por la definición de una matriz simétrica definida positiva. Una matriz es *sdp* si y solo si es simétrica y que, dado cualquier vector no nulo, multiplicando a izquierda y a derecha a la matriz A por ese vector se obtiene un número positivo:

Sea $A \in \mathbb{R}^{n\times n}$, se dice *sdp* sii:

- $A = A^T$, es decir A es **simétrica**.
- $x^t Ax > 0$ para todo $x \in \mathbb{R}^n$, $x \neq 0$, es decir A es **definida positiva**.

Propiedades:

- Una de las primeras propiedades que tienen las matrices *sdp* es que son inversibles.
- Otra propiedad que tienen las matrices *sdp* es que todos los coeficientes en la diagonal son positivos ($a_{ii} > 0$).
- Las matrices *sdp* tienen la propiedad de que todas sus submatrices principales también son *sdp*, luego todas las submatrices principales van a ser inversibles, y por lo tanto A va a tener factorización LU.

  Otra manera de llegar a la misma conclusión sobre la existencia de la factorización LU de una matriz *sdp*. Para eso vamos a hacer uso de las siguientes propiedades:

- A es *sdp* $\Longleftrightarrow$ $B^t AB$ es *sdp*, con B inversible.
- La submatriz conformada por las filas 2 a n y por las columnas 2 a n después del primer paso de gauss es *sdp*.

  El primer paso de la eliminación gaussiana se puede aplicar ya que $a_{11} > 0$ al ser A una matriz *sdp*. Luego, como la submatriz $\tilde{A}$ es *sdp*, en particular, el elemento pivote a_{22} ($\tilde{a_{11}}$) es mayor que 0, y por lo tanto se puede aplicar el siguiente paso de la eliminación gaussiana sin permutaciones.

6.1. Buscando la Factorización de Cholesky

Lo que vamos a ver ahora es si la factorización LU de una matriz *sdp*, que ya sabemos que existe, por el hecho de que la matriz sea *sdp*:

$$
\begin{aligned}
A &= LU \\
A^t &= (LU)^t = U^t L^t
\end{aligned}
$$

Como A es simétrica, $A = A^t$, por lo que $LU = U^t L^t$. Además, como L es triangular inferior con $1s$ en la diagonal, es inversible (y por lo tanto L^t también lo es).

$$
\begin{aligned}
LU = U^t L^t &\Longrightarrow \\
U(L^t)^{-1} &= L^{-1} U^t
\end{aligned}
$$

Notemos que tanto U como L^t son triangular superior, y por lo tanto $U(L^t)^{-1}$ es triangular superior. Por otro lado, tanto L^{-1} como U^t son triangular inferior, y por lo tanto $L^{-1}U^t$ es triangular inferior. Luego, estamos en condiciones de asegurar que la matriz $U(L^t)^{-1}$ o la matriz $L^{-1}U^t$ (son la misma) es al mismo tiempo triangular inferior como triangular superior, y por lo tanto es una matriz diagonal D. Luego

$$
\begin{aligned}
U(L^t)^{-1} = L^{-1}U^t = D &\Longrightarrow \\
U = DL^t &\Longrightarrow \\
A = LDL^t &
\end{aligned}
$$

Hasta ahora solo usamos que A es simétrica y que tiene factorización LU. Veamos qué podemos decir sobre los coeficientes de la matriz D.

Por un lado, sabemos que, al ser A una matriz *sdp*, vale que $x^t A x > 0$ para todo x no nulo. Por otro lado, sabemos que existe $\tilde{x}$ no nulo tal que $L^t x = e_i$ (la solución sería $\tilde{x} = (L^t)^{-1} e_i$), donde e_i es el i-ésimo vector canónico $(0, \cdots, 0, 1, 0, \cdots, 0)$, entonces:

$$
\begin{aligned}
\tilde{x}^t A \tilde{x} &= \tilde{x}^t L D L^t \tilde{x} \\
&= (L^t \tilde{x})^t D (L^t \tilde{x}) \\
&= e_i^t D e_i \\
&= d_{ii}
\end{aligned}
$$

y como $x^t A x > 0$ para todo x no nulo, $\tilde{x}^t A \tilde{x} = d_{ii}$, y $\tilde{x} \neq 0$, entonces $d_{ii} > 0$ para todo $i = 1, \cdots, n$.

Luego, podemos considerar la matriz $\sqrt{D}$, donde $(\sqrt{D})_{ij} = \sqrt{d_{ij}}$, y sabemos que existe y está bien definido porque los elementos no nulos de D ya sabemos que son positivos, y además se cumple que $D = \sqrt{D} \cdot \sqrt{D}$.

¿Para qué vamos a utilizar esto? Volviendo a la estructura $A = LDL^t$, podemos reemplazar a D por $\sqrt{D} \cdot \sqrt{D}$, obteniendo

$$
\begin{aligned}
A &= L\sqrt{D} \cdot \sqrt{D} L^t \\
&= L\sqrt{D} \cdot (L^t \sqrt{D}^t)^t \\
&= L\sqrt{D} \cdot (L^t \sqrt{D})^t (*) \\
&= \tilde{L} \cdot \tilde{L}^t
\end{aligned}
$$

$(*)$ pues al ser $\sqrt{D}$ diagonal, $\sqrt{D} = \sqrt{D}^t$.

Finalmente hemos obtenido $A = \tilde{L}\tilde{L}^t$, con $\tilde{L}$ triangular inferior, y esto es lo que se conoce como **Factorización de Cholesky**. Es decir, a toda matriz *sdp* la podemos factorizar como una matriz triangular inferior por su traspuesta. Notemos que $\tilde{L}$ no necesariamente tiene 1s en la diagonal, pero sabemos que los elementos de la diagonal son positivos y valen $\tilde{l}_{ii} = l_{ii}\sqrt{d_{ii}} = \sqrt{d_{ii}}$.

Sabiendo, entonces, que a una matriz *sdp* la podemos factorizar como una matriz triangular inferior por su traspuesta, podemos tratar de derivar directamente a esta factorización, sin necesidad de pasar por la factorización LU .

$$\begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{21} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \cdots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ni} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} \tilde{l}_{11} & 0 & \cdots & 0 \\ \tilde{l}_{21} & \tilde{l}_{22} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ \tilde{l}_{i1} & \tilde{l}_{i2} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ \tilde{l}_{n1} & \tilde{l}_{n2} & \cdots & \tilde{l}_{nn} \end{bmatrix} \begin{bmatrix} \tilde{l}_{11} & \tilde{l}_{21} & \cdots & \tilde{l}_{n1} \\ 0 & \tilde{l}_{22} & \cdots & \tilde{l}_{n2} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \tilde{l}_{ni} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \tilde{l}_{nn} \end{bmatrix}$$

Factorización de Cholesky

Entrada: $\mathbf{A} \in \mathbb{R}^{n\times n}$ definida positiva.
Salida: $\mathbf{L}$ triangular superior, con elementos positivos en la diagonal, tal que $\mathbf{A} = \mathbf{L}\cdot\mathbf{L}^{\mathrm{T}}$.

$l_{1,1} \leftarrow \sqrt{a_{1,1}}$
for $i = 2, \ldots, n$ **do**
  $l_{i,1} \leftarrow \dfrac{a_{i,1}}{l_{1,1}}$
for $j = 2, \ldots, n$ **do**
  $l_{j,j} \leftarrow \sqrt{a_{j,j} - \sum_{k=1}^{j-1}(l_{j,k})^2}$
  for $i = j+1, \ldots, n$ **do**
    $l_{i,j} \leftarrow \dfrac{1}{l_{j,j}} \cdot \left(a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}\cdot l_{j,k}\right)$
$l_{n,n} \leftarrow \sqrt{a_{n,n} - \sum_{k=1}^{n-1}(l_{n,k})^2}$

Puede observarse que la complejidad del algoritmo es $\mathrm{O}(n^3)$. Si bien se trata de la misma complejidad asintótica que la de obtener una factorización LU, las constantes son mejores; en la práctica, computar una factorización de Cholesky es aproximadamente el doble de rápido que obtener una factorización LU.

6.2. Propiedades Varias

- $A + A^T$ es una matriz simétrica.
- $A - A^T$ es una matriz antisimétrica.
- Toda matriz se puede escribir como la suma entre una matriz simétrica y una matriz antisimétrica.
- $e_i^t A e_i = a_{ii}$.
- Si A es definida positiva, entonces A^T también lo es.
- Si A es inversible, entonces AA^T es simétrica definida positiva.
- Si A no es inversible, AA^T es simétrica semi-definida positiva, es decir que $x^t AA^T x \geq 0$, para todo x.
- A tiene factorización de Cholesky $\iff$ A es *sdp*.

- El polinomio de grado 2 $p(x) = ax^2 + bx + c$ tiene discriminante $\Delta = b^2 - 4ac$, y vale que:
  - $\Delta > 0 \iff p(x)$ tiene dos raíces reales distintas.
  - $\Delta = 0 \iff p(x)$ tiene dos raíces coincidentes reales.
  - $\Delta < 0 \iff p(x)$ no tiene raíces reales.
- Si A es *sdp*, entonces $|x^T Ay| \leq \sqrt{x^T Ax}\sqrt{y^T Ay}$ ($<$ si x e y son li, $=$ si son ld).
- Si A es *sdp*, entonces $|aij| \leq a_{ii}a_{jj}$
- Si A es *sdp*, entonces el elemento de módulo máximo de A está en la diagonal.

Capítulo 7

Factorización QR

De momento hemos caracterizado la factorización LU de una matriz, y además vimos la utilidad que tiene esta factorización para resolver sistemas lineales. En este capítulo vamos a presentar otro tipo de factorización, que también nos va a resultar útil en el contexto de resolución de sistemas de ecuaciones lineales, pero además nos va a resultar valioso en el marco de otros problemas, que vamos a ver más adelante en la materia. La factorización QR utiliza matrices ortogonales, así que vamos a recordar qué es una matriz ortogonal.

Matrices ortogonales

Recordemos que dos vectores $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ se dicen **ortogonales** ($\mathbf{x} \perp \mathbf{y}$) si su producto interno $\langle \mathbf{x}, \mathbf{y} \rangle = x^t y$ es 0. Un conjunto de vectores es **ortogonal** si sus elementos son ortogonales dos a dos. Un conjunto de vectores es **ortonormal** si es ortogonal y la norma de todos sus elementos es 1. Los elementos de un conjunto ortonormal siempre son linealmente independientes.

Decimos que una matriz $\mathbf{Q} \in \mathbb{R}^{n \times n}$ es **ortogonal** si y solo si es una matriz inversible que tiene la particularidad de que la inversa es su transpuesta, es decir $QQ^t = Q^t Q = I$.

Otra manera de definir a una matriz ortogonal es vía la caracterización de ciertas propiedades que tienen sus columnas y filas. En el caso de las columnas, podemos ver que las columnas son ortogonales entre sí, y además tienen norma 2 igual a 1, es decir que forman un conjunto **ortonormal**. Las filas también son ortogonales entre sí y son de norma 2 igual a 1.

Sigamos con más propiedades de las matrices ortogonales:

- $\|Q\|_2 = 1$.
- $\kappa_2(Q) = 1$.

Notemos que esta propiedad es muy significativa, porque el número de condición nos habla de la estabilidad que podíamos esperar al momento de resolver un sistema de ecuaciones. Esto nos está diciendo que las matrices ortogonales son muy estables, y por lo tanto frente a pequeños movimientos en el término independiente, es de esperar pequeños movimientos en el vector solución.

- $\|Qx\|_2 = \|x\|_2$.

Esto nos dice que la transformación lineal definida por una matriz ortogonal no cambia la norma 2 de un vector. Es decir, la imagen de un vector tiene la misma norma 2 que el vector original.

- El producto de matrices ortogonales es también ortogonal, es decir, si $\mathbf{Q_1}$ y $\mathbf{Q_2}$ son ortogonales, $\mathbf{Q_1} \cdot \mathbf{Q_2}$ es ortogonal.
- $det(Q) = 1$ o -1.

- Las matrices de permutación P son matrices ortogonales.

Entremos ahora a la factorización QR. La idea va a ser que a una matriz, en principio, cuadrada, después vamos a ver cómo podemos generalizar esta factorización a matrices no cuadradas, la vamos a escribir como el producto de una matriz ortogonal Q por una matriz triangular superior R. ¿Para qué nos va a servir esto?

Si se tiene el sistema $Ax = b$, y se tiene la factorización QR de A, entonces podemos reemplazar a A como QR, es decir $QRx = b$. Ahora, como la matriz Q es ortogonal, sabemos que su inversa es la traspuesta, entonces vamos a multiplicar a izquierda por la traspuesta, y nos queda

$$\begin{aligned} Q \cdot Rx &= b \\ Q^t Q \cdot Rx &= Q^t b \\ Rx &= Q^t b \end{aligned}$$

Luego, el sistema original $Ax = b$ es equivalente al sistema $Rx = Q^t b$, entonces si hallamos la solución de $Rx = Q^t b$, habremos obtenido la solución del sistema original $Ax = b$. Pero, ¿cuál es la ventaja de este nuevo sistema? La ventaja es que tiene asociada una matriz que es triangular superior, para lo cual podemos aplicar *Backward Substitution* y encontrar la solución en un orden cuadrático.

Entonces, la ventaja de tener la factorización QR es que la resolución de un sistema de ecuaciones general se puede resolver vía la resolución de un sistema equivalente triangular superior en un orden cuadrático.

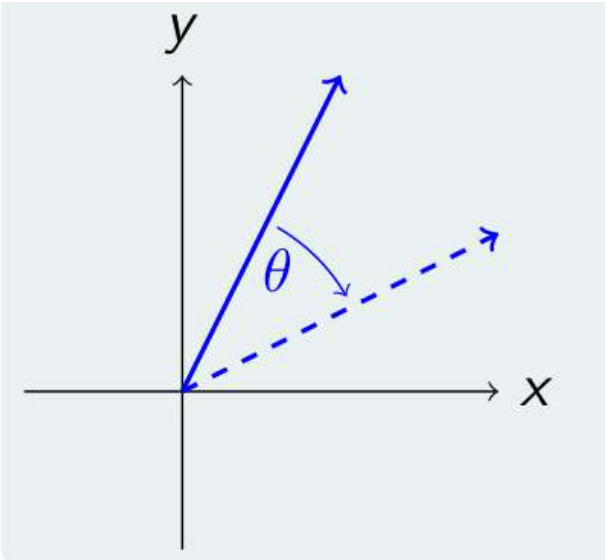
Por lo tanto, queda bien claro que la factorización QR, así como la factorización LU, nos sirve para resolver sistemas de ecuaciones de una manera más eficiente.

7.1. Buscando la factorización QR

Vista, entonces, la utilidad de la factorización QR, necesitamos de alguna metodología para encontrarla. Las dos formas que estudiaremos para computar la factorización QR de una matriz se basan en el mismo principio: ir aplicando sucesivas transformaciones a la matriz, todas ellas definidas por matrices ortogonales, hasta llevarla a una forma triangular superior. La diferencia está en el tipo de estas transformaciones.

7.1.1. Rotaciones en un ángulo θ

Comenzamos analizando en el plano cierto tipo de transformaciones lineales, que se conocen como **rotaciones**. La idea es que, dado un vector x, la imagen sea una rotación en un ángulo θ, en sentido horario, del vector original, preservando la norma 2 del vector original.



Veamos cómo podemos caracterizar a este tipo de transformaciones. Lo primero que podemos decir es que, como esta transformación preserva la norma 2 del vector original, es que la matriz asociada debe ser ortogonal:

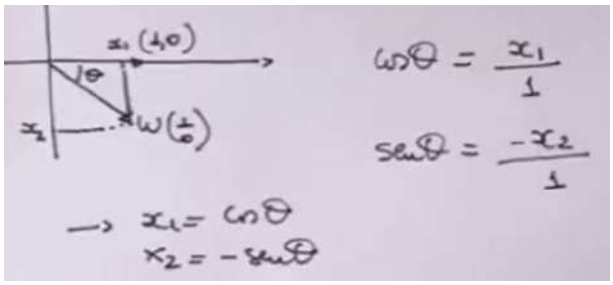
$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

Lo que queremos hacer es determinar los coeficientes de la matriz W. En primer lugar, observemos que:

$$We_1 = \begin{bmatrix} w_{11} \\ w_{21} \end{bmatrix} \; We_2 = \begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix}$$

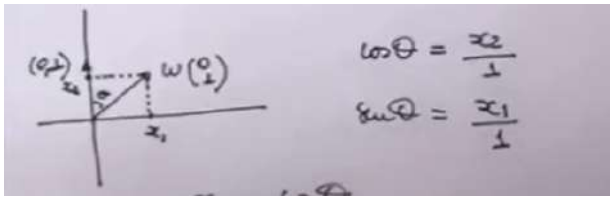
Con esto en mente, si logramos identificar las coordenadas de la imagen del primer vector canónico y la imagen del segundo vector canónico, entonces habremos determinado los coeficientes de la matriz asociada a la transformación lineal.

Comenzamos por el primer vector canónico $e_1 = (1, 0)$. Sabemos que la norma 2 de We_1 es igual a la norma 2 del original, por lo que tiene norma 2 igual a 1. Lo que queremos es identificar qué coordenadas tiene este vector imagen. También sabemos que el triangulo rectángulo que se forma tiene hipotenusa igual a 1, porque es la norma 2 del vector imagen. En un triángulo rectángulo podemos aplicar lo que sabemos de seno y coseno, y nos queda $x_1 = \cos\theta$, $x_2 = -\operatorname{sen}\theta$:



Entonces, ya tenemos identificada la primer columna de W: $We_1 = \begin{pmatrix} \cos\theta \\ -\operatorname{sen}\theta \end{pmatrix}$.

Ahora veamos qué pasa con el segundo vector canónico $e_2 = (0, 1)$. Nuevamente buscamos las coordenadas, y nos queda $x_2 = \cos\theta$, $x_1 = \operatorname{sen}\theta$:



Con esto tenemos identificada la segunda columna de W: $We_2 = (\cos\theta, \operatorname{sen}\theta)$.

En definitiva, hemos logrado caracterizar los cuatro coeficientes de la matriz asociada a la transformación que rota a todo vector del espacio, en un ángulo tita, en sentido horario, y esa matriz es:

$$\mathbf{W} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

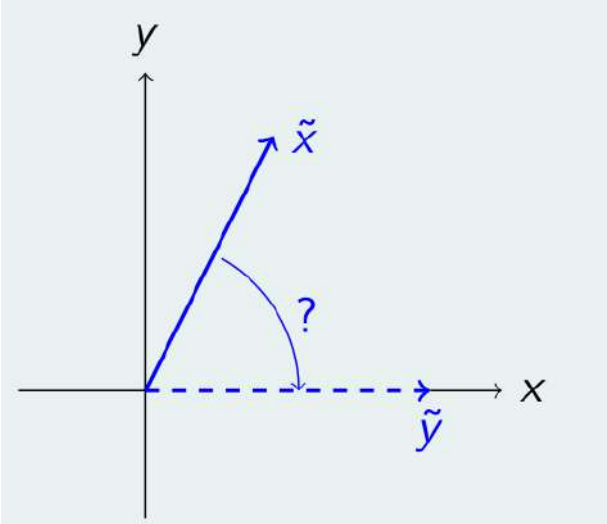
Notemos que, efectivamente, la matriz W es una matriz ortogonal, porque las columnas tienen norma 2 igual a 1, ya que $cos^2\,\theta + sen^2\,\theta = 1$, y las columnas son ortogonales entre sí, porque $\cos\theta \cdot \operatorname{sen}\theta - \operatorname{sen}\theta \cdot \cos\theta = 0$.

7.1.2. Rotaciones hacia el eje x

Ahora nos vamos a plantear el mismo problema, pero un poquito diferente. En este caso, en vez de darnos un ángulo, nos van a dar dos vectores de igual norma 2, y lo que vamos buscar es una rotación tal que la imagen del primer vector sea el otro vector:

$$W\tilde{x} = \tilde{y}$$

Entonces, lo que queremos es encontrar una rotación, que dado $\tilde{x}$, esa transformación sea tal que la imagen de $\tilde{x}$ sea un vector $\tilde{y}$ que tenga la misma norma que $\tilde{x}$ y tenga la segunda coordenada sea nula. Como los dos vectores van a tener la misma norma, la primera coordenada de $\tilde{y}$ debe ser $\|\tilde{x}\|_2$.



Hasta ahora sabemos que $\tilde{y} = \begin{bmatrix} \|\tilde{x}\|_2 \\ 0 \end{bmatrix}$, y sabemos que las rotaciones en sentido horario tienen la estructura $W = \begin{bmatrix} \cos\theta & \operatorname{sen}\theta \\ -\operatorname{sen}\theta & \cos\theta \end{bmatrix}$, y queremos que $W\tilde{x} = \tilde{y}$. Luego, tenemos un sistema de ecuaciones, cuyas incógnitas son $\cos\theta$ y $\operatorname{sen}\theta$, el cual podemos resolver:

$$\begin{cases} \tilde{x}_1 \cos\theta + \tilde{x}_2 \operatorname{sen}\theta = \|\tilde{x}\|_2 \\ \tilde{x}_2 \cos\theta - \tilde{x}_1 \operatorname{sen}\theta = 0 \end{cases}$$

Notemos que, en vez de estar buscando un $\tilde{x}$ tal que $W\tilde{x} = \tilde{y}$, lo que estamos haciendo es buscar los coeficientes de W tales que $W\tilde{x} = \tilde{y}$.

Una primera observación es que $\tilde{x}_2$ es no nula, porque si lo fuera $\tilde{x}$ estaría sobre el eje x, y no necesitaría rotar (se podría considerar a la identidad como la transformación buscada). Luego, para que el problema sea de interés, vamos a considerar que $\tilde{x}_2 \neq 0$. Luego, si $\tilde{x}_2 \neq 0$, pudiendo seguir despejando de la siguiente manera:

$$\begin{cases} \cos\theta\tilde{x}_1 + \operatorname{sen}\theta\tilde{x}_2 = \|\tilde{x}\|_2 \\ -\operatorname{sen}\theta\tilde{x}_1 + \cos\theta\tilde{x}_2 = 0 \end{cases}$$

$$\begin{aligned} &\Longrightarrow \\ \cos\theta &= \frac{\operatorname{sen}\theta\tilde{x}_1}{\tilde{x}_2} \\ &\Longrightarrow \\ \operatorname{sen}\theta \cdot \frac{\tilde{x}_2^2}{\tilde{x}_2} + \operatorname{sen}\theta \cdot \tilde{x}_2 &= \|\tilde{x}\|_2 \\ \operatorname{sen}\tilde{x}_1^2 + \operatorname{sen}\theta\tilde{x}_2^2 &= \tilde{x}_2\|\tilde{x}\|_2 \\ \operatorname{sen}\theta &= \frac{\tilde{x}_2}{\|\tilde{x}\|_2} \\ &\Longrightarrow \\ \cos\theta &= \frac{\tilde{x}_1}{\|\tilde{x}\|_2} \end{aligned}$$

Por lo tanto,

$$\tilde{W} = \begin{bmatrix} \dfrac{\tilde{x}_1}{\|\tilde{x}\|_2} & \dfrac{\tilde{x}_2}{\|\tilde{x}\|_2} \\ \\ -\dfrac{\tilde{x}_2}{\|\tilde{x}\|_2} & \dfrac{\tilde{x}_1}{\|\tilde{x}\|_2} \end{bmatrix}.$$

Entonces, logramos caracterizar a la matriz W que tiene la propiedad de que la rotación asociada a ella verifica que la imagen de $\tilde{x}$ es $\tilde{y}$, donde $\tilde{y}$ tiene la característica de que su segunda coordenada es nula, es decir $\tilde{y}_2 = 0$.

7.1.3. Método de Givens

Veamos para qué nos puede servir este tipo de transformaciones. Recordemos que queríamos encontrar la factorización QR, con Q una matriz ortogonal, y R una matriz triangular superior, de una matriz A de 2×2.

Notemos que si conseguimos anular a_{22}, a al multiplicar a A por una matriz ortogonal W (cuya inversa es W^t, por ser una matriz ortogonal), entonces habremos conseguido la factorización QR, pues:

$$\begin{aligned} WA &= \begin{bmatrix} * & * \\ 0 & * \end{bmatrix} \\ \Longrightarrow WA &= R \\ \Longrightarrow W^tW \cdot A &= W^t \cdot R \\ A &= QR \end{aligned}$$

Ahora veamos cómo podemos encontrar a W para que anule a_{22}. Si tomamos al vector $\tilde{x} = a_1$, siendo a_1 la primer columna de A, y tomamos al vector $\tilde{y}$ como $\left(\begin{smallmatrix} \|\tilde{x}_2\| \\ 0 \end{smallmatrix}\right)$, sabemos que existe una rotación W tal que $W\tilde{x} = \tilde{y}$. Luego, si multiplicamos a A por esta matriz W, obtenemos

$$WA = \begin{bmatrix} \|\tilde{x}_2\| & * \\ 0 & * \end{bmatrix}$$

ya que $col_1(WA) = Wcol_1(A) = W\tilde{x} = \tilde{y}$.

Por lo tanto, obtuvimos la factorización QR de A, y concluimos que toda matriz 2×2 tiene factorización QR, y siempre la podemos encontrar vía la rotación que anula a_{22}. Veamos cómo podemos generalizar esto a matrices de $n \times n$. Vamos a ignorar, de momento, que la matriz A es de $n \times n$, y vamos a considerar únicamente los primeros dos coeficientes de la primer columna de A (a_{11}, a_{22})

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

Sabemos que existe una rotación W tal que la imagen de $\left(\begin{smallmatrix} a_{11} \\ a_{22} \end{smallmatrix}\right)$ es una vector con la segunda componente nula. Si embebemos esa matriz W de 2×2 en una matriz de $n \times n$, completando el resto con la matriz identidad, lo que obtenemos es también una matriz ortogonal:

$$W_{12} = \begin{bmatrix} \tilde{w}_{11} & \tilde{w}_{12} & 0 & \cdots & 0 \\ \tilde{w}_{21} & \tilde{w}_{22} & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

La utilidad de esta matriz es que si multiplicamos a A de $n \times n$ por esta matriz, se obtiene la rotación deseada sobre $\left(\begin{smallmatrix} a_{11} \\ a_{22} \end{smallmatrix}\right)$, afectando únicamente a las primeras dos filas de A.

$$W_{12}A = \begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ a_{31} & a_{32} & \cdots & a_{3n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

Luego, para triangular a la matriz A, la idea va a ser iterar sobre las columnas de A, e ir aplicando tantas rotaciones como sea necesario, hasta que se consiga triangular la matriz. Luego, la matriz de rotación $W_{ij} \in \mathbb{R}^{n \times n}$, que anula a $a_{ji}^{(i-1)}$, va a tener la siguiente estructura:

$$\begin{aligned} w_{i,i} &= \tilde{w}_{11}, & w_{i,j} &= \tilde{w}_{21}, \\ w_{j,i} &= \tilde{w}_{12}, & w_{j,j} &= \tilde{w}_{22}, \end{aligned}$$

con $\tilde{W} \in \mathbb{R}^{2\times 2}$ tal que $\tilde{W}\tilde{x} = \tilde{y}$, rotando al vector $\tilde{x} = \left(\begin{smallmatrix} a_{ii}^{(i-1)} \\ a_{ji}^{(i-1)} \end{smallmatrix}\right)$, al vector $\tilde{y} = \left(\begin{smallmatrix} \|\tilde{x}\|_2 \\ 0 \end{smallmatrix}\right)$, y ,

El resto de la matriz se completa con la matriz identidad. Notemos que los valores de las posiciones de **A** se van modificando a lo largo del proceso, por lo que siempre se consideran los obtenidos en la iteración anterior del algoritmo.

De esta forma, construyendo iterativamente las matrices de rotación y multiplicando **A** a izquierda, se llega a una forma triangular superior

$$\begin{aligned}\mathbf{R} &= (\mathbf{W}_{n-1,n}) \cdot (\mathbf{W}_{n-2,n} \cdot \mathbf{W}_{n-2,n-1}) \cdot \ldots \cdot (\mathbf{W}_{1,n} \cdot \ldots \cdot \mathbf{W}_{1,2}) \cdot \mathbf{A}. \\ &= \mathbf{W} \cdot \mathbf{A}\end{aligned}$$

donde $\mathbf{W}$ es el producto de todas las $\mathbf{W}_{i,j}$, y como el producto de matrices ortogonales es ortogonal, $\mathbf{W}$ es una matriz ortogonal. Tomando $\mathbf{Q} = \mathbf{W}^{\mathrm{T}}$, se obtiene una factorización QR para A:

$$\boxed{\mathbf{A} = \mathbf{Q} \cdot \mathbf{R}}$$

Análisis del costo Primero observemos el costo del producto

$$W_{1,2} \cdot A = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & \cdots & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,n} \end{bmatrix} =$$

$$\begin{pmatrix} \cos(\theta) \cdot a_{1,1} + \sin(\theta) \cdot a_{2,1} & \cos(\theta) \cdot a_{1,2} + \sin(\theta) \cdot a_{2,2} & \cdots & \cos(\theta) \cdot a_{1,n} + \sin(\theta) \cdot a_{2,n} \\ -\sin(\theta) \cdot a_{1,1} + \cos(\theta) \cdot a_{2,1} & -\sin(\theta) \cdot a_{1,2} + \cos(\theta) \cdot a_{2,2} & \cdots & -\sin(\theta) \cdot a_{1,n} + \cos(\theta) \cdot a_{2,n} \\ & fila_3 & & \\ & \vdots & & \\ & fila_n & & \end{pmatrix}$$

Como se puede observar, se realizan operaciones sólo en las primeras dos filas, cada una de las cuales toma $n \cdot (2$ productos $+ 1$ suma). Con lo cual, al realizar todo el producto matricial se realizan $4n$ productos y $2n$ sumas. Todos los productos matriciales hacen lo mismo, por lo cual por tenemos un consumo un total de $(n-1) \cdot (4n$ productos $+ 2n$ sumas).

En cada etapa voy operando con 1 fila menos que en la anterior. Luego, por etapa gasto:

- Etapa 2: $(n-2) \cdot (4(n-2+1)$ productos $+ 2(n-2+1)$ sumas)
- Etapa 3: $(n-3) \cdot (4(n-3+1)$ productos $+ 2(n-3+1)$ sumas)
- $\vdots$
- Etapa i: $(n-i) \cdot (4(n-i+1)$ productos $+ 2(n-i+1)$ sumas)

En conclusión, el costo total del algoritmo es de:

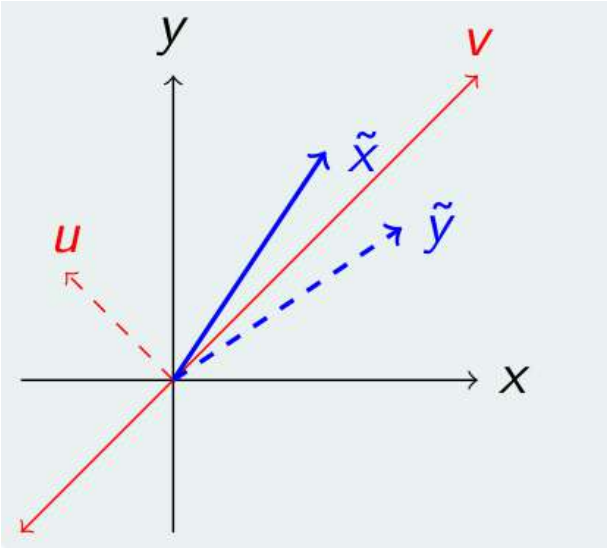
$$\sum_{j=1}^{n-1} (n-j) \cdot (4(n-j+1) \text{ productos } + 2(n-j+1) \text{ sumas}) \in O(\frac{4}{3} \cdot n^3)$$

Conclusión: Este algoritmo es conocido como el **Método de Givens**, y tiene complejidad, en el caso general, de $O(\frac{4}{3}n^3)$, el doble que la factorización LU. Este método presenta una gran ventaja si se está trabajando con matrices ralas (donde una gran cantidad de las posiciones está ocupada por ceros), ya que se puede aprovechar el hecho de que cada paso del algoritmo pone un cero en una posición particular de la matriz, por lo que se puede realizar una optimización aplicando la transformación únicamente en el caso en que no haya un 0 en dicha posición.

7.1.4. Reflexiones sobre un plano

Vamos a ver otra manera de hallar la factorización QR de una matriz. Primero lo vamos a pensar en $\mathbb{R}^{2\times 2}$, y después lo vamos a extender a $\mathbb{R}^{n\times n}$. En este caso se toma en cuenta transformaciones lineales conocidas como **reflexiones**, que lo que hacen es, dado un plano, reflejar a todo vector, respecto a ese

plano. Si vemos en este gráfico, el plano viene dado por el vector v, u es el vector ortogonal al plano, y la imagen de $\tilde{x}$ es $\tilde{y}$:



La transformación que estamos buscando tiene que cumplir al menos las siguientes características:

- $H\tilde{x} = \tilde{y}$.
- $Hu = -u$.
- $Hv = v$.

¿Qué características tiene esta reflexión o cómo podemos ir deduciéndola? Veamos qué es lo que está pasando. Tenemos un plano definido por la dirección v y u, con u ortogonal a v. Como u es ortogonal a v, forman una base del espacio $\mathbb{R}^2$, por lo que todo vector en $\mathbb{R}^2$ se puede escribir como combinación lineal de esta base. En particular, si

$$\begin{aligned} \tilde{x} &= \alpha v + \beta u \\ &\Longrightarrow \\ \tilde{y} &= \alpha v - \beta u \end{aligned}$$

pues como $\tilde{y}$ es la reflexión de $\tilde{x}$ sobre v, tiene la misma componente α por el lado de v y $-\beta$ por el lado de u.

Entonces, queremos que

$$\begin{aligned} H\tilde{x} &= \tilde{y} \\ &= \alpha v - \beta u \\ &= \alpha v + \beta u - 2\beta u \\ &= \tilde{x} - 2\beta u \end{aligned}$$

Si ahora escribimos a H como $H = I - W$, entonces lo que vamos a buscar es

$$\begin{aligned} H\tilde{x} &= (I - W)\tilde{x} \\ &= \tilde{x} - W\tilde{x} \\ &\Longleftrightarrow \\ 2\beta u &= W\tilde{x} \\ &= W(\alpha v + \beta u) \\ &= \alpha W v + \beta W u \end{aligned}$$

Por lo tanto, necesitamos que $Wv = 0$ y que $Wu = 2u$. Si encontramos una W que cumpla con estas propiedades, entonces habremos encontrado la H que buscábamos. Veamos de dónde podemos sacar una transformación con esta propiedad.

Vamos a suponer, sin pérdida de generalidad, que $u \in col2$, y vamos a definir una matriz $P = uu^t$, asumiendo $\|u\|_2 = 1$. Veamos qué propiedades tiene esta matriz:

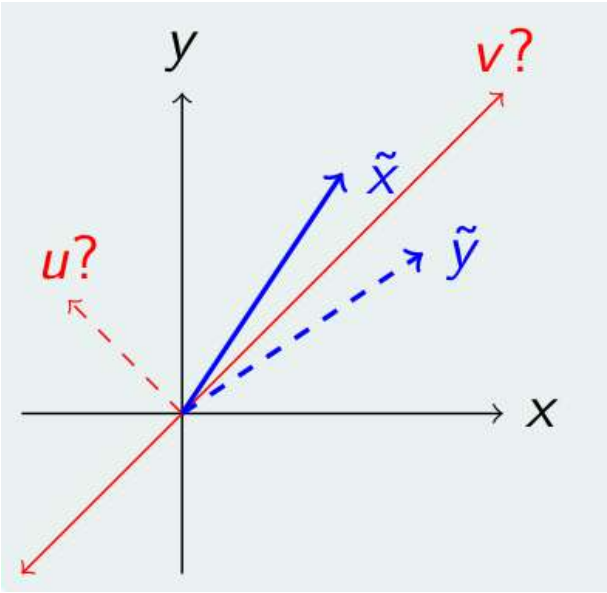
- P es simétrica.
- $P^2 = P$.
- $Pu = u$.
- $Pv = 0$.

Buscábamos una matriz W tal que $Wu = 2u$ y $Wv = 0$, por lo que si tomamos $W = 2P = 2uu^t$, y por lo tanto podemos tomar $H = I - 2uu^t$, obteniendo la matriz deseada. Luego, esta matriz H tiene las siguientes propiedades:

- $Hv = v$.
- $Hu = -u$.
- H es simétrica.
- H es ortogonal.

7.1.5. Método de Householder

Para encontrar la factorización QR aplicando reflexiones, primero debemos poder resolver un problema ligeramente distinto al anterior. Hasta el momento nos daban el plano, y reflejábamos respecto de ese plano. Ahora, nos van a dar dos vectores $\tilde{x}$, $\tilde{y}$ de igual normal 2, y queremos encontrar una reflexión tal que la imagen de $\tilde{x}$ sea el segundo.



La propuesta va a ser determinar cuál es el plano generado por v y u tal que, si reflejamos a $\tilde{x}$ respecto a este plano, la imagen de $\tilde{x}$ es $\tilde{y}$. Es decir, hallar v y u tal que $(I - 2uu^t) \cdot \tilde{x} = \tilde{y}$. Se puede demostrar que

$$v = \tilde{x} + \tilde{y}$$
$$u = \frac{\tilde{x} - \tilde{y}}{\|\tilde{x} - \tilde{y}\|_2}$$
$$H = I - 2uu^t$$

cumple con lo pedido. Notemos que normalizamos a u para que $\|u\|_2 = 1$.

Veamos cómo podemos usar estas transformaciones para encontrar una factorización QR de una matriz. Vamos a empezar viendo el caso de una matriz A de 2×2 a la cual queremos encontrar su factorización QR. El objetivo sería anular el a_{22} mediante la aplicación de matrices ortogonales.

Siguiendo la idea de la propiedad que nos dice que, dado un $\tilde{x}$ y un $\tilde{y}$ de igual norma, podemos reflejar a $\tilde{x}$ sobre $\tilde{y}$, vamos a considerar a $\tilde{x}$ como la primer columna de A, y a $\tilde{y}$ va a ser un vector cuya segunda componente sea nula, y además cuente con la misma norma 2 que $\tilde{y}$.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$
$$\tilde{x} = \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix} \qquad \tilde{y} = \begin{pmatrix} \|\tilde{x}\| \\ 0 \end{pmatrix}$$

Entonces, sabemos que existe una H tal que la imagen de $\tilde{x}$ es $\tilde{y}$, y por lo tanto, cuando hacemos HA obtenemos el siguiente resultado:

$$HA = \begin{bmatrix} \|\tilde{x}\|_2 & * \\ 0 & * \end{bmatrix}$$

pues $col_1(HA) = Hcol_1(A) = H\tilde{x} = \tilde{y}$. De esta manera, hemos obtenido una matriz triangular superior, al aplicarle una reflexión a la matriz A. Luego, para obtener la factorización QR simplemente hacemos

$$HA = R$$
$$H^t HA = H^t R$$
$$\boxed{A = QR}$$

Para el caso de una matriz de $n \times n$ va a ser muy similar. Para el primer paso de la triangulación, si consideramos a

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad \tilde{x} = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \quad \tilde{y} = \begin{pmatrix} \|\tilde{x}\|_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

sabemos que existe una reflexión $H_1 \in \mathbb{R}^{n \times n}$ tal que la imagen de $\tilde{x}$ es $\tilde{y}$, de manera tal que podemos obtener

$$H_1 A = \begin{bmatrix} \|\tilde{x}\|_2 & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & a_{n2} & \cdots & a_{nn} \end{bmatrix} = A^1$$

y de esta manera conseguimos anular desde la segunda hasta la n-ésima componente de la primer columna de A. Veamos cómo podemos continuar con el resto de las columnas.

La idea va a ser que para cada $k \in \{1, \ldots, n-1\}$, se toma como $\tilde{x}$ a los últimos $n-k+1$ elementos de la columna k-ésima, y luego se le aplica el mismo proceso. Sin embargo, la matriz de reflexión es de $n-k+1 \times n-k+1$, y necesitamos que esta pueda multiplicar a $A^{(k-1)} \in \mathbb{R}^{n \times n}$. Por lo tanto, debemos rellenar esta matriz de reflexión con la identidad:

$$\tilde{x} = \begin{pmatrix} a_{i,i}^{(i-1)} \\ a_{i+1,i}^{(i-1)} \\ \vdots \\ a_{n,i}^{(i-1)} \end{pmatrix} \quad \tilde{y} = \begin{pmatrix} \|\tilde{x}\|_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad u_i = \frac{\tilde{x} - \tilde{y}}{\|\tilde{x} - \tilde{y}\|_2}$$
$$\implies H_i = \begin{bmatrix} I & 0 \\ 0 & I - 2u_i u_i^t \end{bmatrix}$$

Luego, si aplicamos esta matriz a $A^{(i-1)}$, obtenemos

$$H_i A^{(i-1)} = \begin{bmatrix} a_{12}^{(i-1)} & a_{12}^{(i-1)} & \cdots & a_{1i}^{(i-1)} & a_{1i+1}^{(i-1)} & \cdots & a_{1n}^{(i-1)} \\ 0 & a_{22}^{(i-1)} & \cdots & a_{2i}^{(i-1)} & a_{2i+1}^{(i-1)} & \cdots & a_{2n}^{(i-1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & ||\tilde{x}||_2 & a_{ii+1}^{i} & \cdots & a_{in}^{i} \\ 0 & 0 & 0 & 0 & a_{i+1i+1}^{i} & \cdots & a_{i+1n}^{i} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & 0 & a_{ni+1}^{i} & \cdots & a_{nn}^{i} \end{bmatrix} = A^i$$

Notemos que el único caso en el que este procedimiento se podría romper es en el caso $\tilde{x} = \tilde{y}$, al no poder definir u_i. Sin embargo, simplemente debemos saltear la columna, ya que esta ya tiene ceros donde se buscaba colocarlos.

Entonces, si aplicamos este procedimiento de forma iterativa, podemos concluir que la matriz $\mathbf{A}^{(n-1)}$ será triangular superior, y obteniendo la factorización QR:

$$\mathbf{H}_{n-1} \cdot \ldots \cdot \mathbf{H}_1 \cdot \mathbf{A} = \mathbf{R}$$
$$\mathbf{A} = \mathbf{H}_1^t \cdot \ldots \cdot \mathbf{H}_{n-1}^t \cdot \mathbf{R}$$
$$\boxed{\mathbf{A} = \mathbf{Q} \cdot \mathbf{R}}$$

Este algoritmo es conocido como el **método de Householder**, y siempre está definido para toda matriz $n \times n$. Además, tiene una complejidad de orden cúbica, donde la cantidad de operaciones de punto flotante necesarias es de alrededor de $\frac{2}{3} \cdot n^3$, sin embargo es ciega respecto a la cantidad de ceros en la columna, y por lo tanto es menos eficiente cuando se trabaja con matrices ralas, en comparación al método de rotaciones.

7.2. Unicidad de la Factorización QR

A esta altura tenemos dos metodologías para encontrar la factorización QR de una matriz de $n \times n$, y es natural preguntarse si dicha factorización es única. La factorización QR **no** es única, salvo que la matriz R tenga coeficientes de la diagonal positivos, es decir $r_{ii} > 0$ para todo $i = 1, \cdots, n$. Notemos que para que esto sea posible, A debe ser inversible.

7.3. Propiedades Varias

Identidades Trigonométricas

- $\operatorname{sen} -\theta = -\operatorname{sen}\theta = \operatorname{sen}(\theta + \pi)$.
- $\cos\theta = \cos\theta = -\cos(\theta + \pi)$.
- $\cos\theta = \operatorname{sen}\frac{\pi}{2} + \theta$.
- $\operatorname{sen}\theta = \cos(\frac{\pi}{2} - \theta)$.
- $\operatorname{sen}^2\theta + cos^2\theta = 1$.

Propiedades Equivalentes: $Q \in \mathbb{R}^{n \times n}$ es una matriz ortogonal sii:

- $QQ^t = Q^tQ = I$.
- $\|Qx\|_2 = \|x\|_2$, para todo x
- las filas (columnas) de Q forman un conjunto ortonormal.

Matrices Ortogonales

- Si Q es ortogonal y triangular, entonces Q es diagonal, y además $col_i(Q) = \pm e_i$.
- Las matrices ortogonales preservan la norma de Frobenius, es decir $\|A\|_F = \|QA\|_F$.
- Un algoritmo basado en reflexiones o en rotaciones para introducir cero es automáticamente estable.

Capítulo 8

Autovalores

Vamos a dejar de lado los sistemas de ecuaciones lineales y vamos a trabajar con otro concepto relacionado con las matrices, que es el tema de **autovalores**. Vamos a recordar lo que son los autovalores de una matriz:

Definición: Si A es una matriz $\in \mathbb{C}^{n\times n}$, entonces $x \in \mathbb{C}^n$ <u>**no nulo**</u> es una autovalor de A sii $\exists\lambda$ escalar tal que:

$$Ax = \lambda x$$

El escalar λ se denomina **autovalor** de A, y se dice que x es un **autovector asociado** a λ de A. Notemos que A debe ser una **matriz cuadrada**, ya que si A fuese una matriz $m \times n$, entonces $Ax \in \mathbb{C}^m$, mientras que $x \in col[C]n$, y por lo tanto A no podría existir ningún autovector.

Al autovalor de módulo máximo se lo conoce como **radio espectral** de A, y se nota como $\rho(A) = \text{máx}\{|\lambda| : \lambda \text{ autovalor de } A\}$. Este concepto nos va a resultar útil un poco más adelante cuando veamos algunos métodos iterativos para la resolución de sistemas de ecuaciones lineales.

¿Qué cosas podemos decir de los autovalores y los autovectores? Lo primero que vamos a decir es que si se tiene un autovalor λ de una matriz A, entonces la matriz $A - \lambda I$ es una matriz singular, pues

$$\begin{aligned} Ax &= \lambda x \\ Ax - \lambda x &= 0 \\ (A - \lambda I)x &= 0 \end{aligned}$$

por lo que $A - \lambda I$ es singular, por lo que $\det(A - \lambda I) = 0$.

Cuando se desarrolla este determinante, el resultado es siempre un polinomio $P(\lambda) = \det(A - \lambda I)$ denominado **polinomio característico** de A. Luego, podemos decir que λ es autovalor de A si y solo si λ es raíz del polinomio característico $P(\lambda)$, y por lo tanto toda matriz A de $n \times n$ tiene n autovalores contados con su multiplicidad, al ser estas las raíces del polinomio característico de grado n.

Propiedades de Autovalores :

- Si $Ax = \lambda x$, entonces $\alpha A + \beta I = (\alpha\lambda + \beta)x$, para todo $\alpha, \beta \in \mathbb{C}$.
- Si $Av = \lambda v$ con v autovector asociado a λ, $A^k v = \lambda^k v$.
- Si Q es una matriz ortogonal, entonces sus autovalores reales son 1 o -1 (las matriz ortogonales conservan la norma 2).

- Si λ^1, λ^2, $\cdots$, λ^k son autovalores distintos con autovectores asociados $v^1, v^2, \cdots, v^k$, entonces los autovectores son linealmente independientes. Por lo tanto, si A tiene n autovalores distintos, entonces tiene una base de autovectores.
- A y A^T tienen los mismos autovalores.
- Si A es triangular superior (inferior), entonces a_{ii} es autovalor de A.
- Si v es un autovector asociado a λ, entonces αv también es un autovector asociado a λ.

8.1. Discos de Gershgorin

Veamos, ahora, otro concepto relacionado con los autovectores, conocido como los **discos de Gershgorin**. Vamos a tomar una matriz, y para cada fila i se define el radio $r_i = \sum_{k=1, k\neq i}^{n} |a_{ik}|$, y con este radio se define el disco

$$D_i = \{x \in \mathbb{C} : |x - a_{ii}| \leq r_i, \text{ para } i = 1, \cdots, n$$

La propiedad que tienen estos discos es que si se tiene un λ autovalor de A, entonces $\lambda \in D_i$ para algún $i = 1, \cdots, n$. Esta propiedad nos da una idea de por donde andan los autovalores de una matriz.

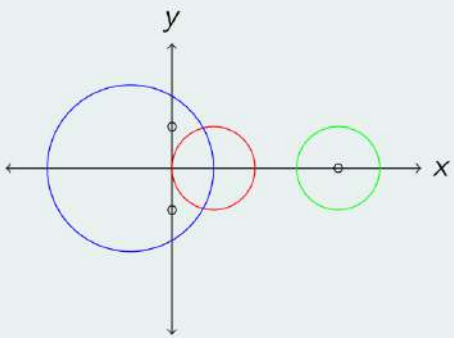
Además, si $M = D_{i_1} \cup D_{i_2} \cup \cdots \cup D_{i_m}$ es disjunto con la unión de los restantes discos D_i, entonces hay exactamente m autovalores de A (contados con su multiplicidad) en M. Veamos un ejemplo:

$$A = \begin{bmatrix} 1 & -1 & 0 \\ 2 & -1 & 0 \\ 0 & 1 & 4 \end{bmatrix} \qquad \lambda^1 = i \quad \lambda^2 = -i \quad \lambda^3 = 4$$

$$D_1 = \{x : |x - 1| \leq 1\}$$

$$D_2 = \{x : |x + 1| \leq 2\}$$

$$D_3 = \{x : |x - 4| \leq 1\}$$



Podemos observar que la unión M entre D_1 y D_2 es disjunta respecto a D_3, y por lo tanto podemos asegurar que en M hay dos autovalores de A ($\lambda^1 = i$, $\lambda^2 = -i$), y el restante se encuentra en D_3 ($\lambda^3 = 4$).

Como calcular los autovalores suele ser costoso, los discos de Gershgorin nos pueden servir para darnos una idea de por donde andan los autovalores, sin tener que calcularlos.

8.2. Diagonalización

Vamos a decir que las matrices A y B de $n \times n$ son **semejantes** si existe una matriz P de $n \times n$ inversible tal que

$$A = P^{-1}BP$$

El concepto de matrices semejantes es importante porque estas comparten sus autovalores. Es decir, si $Av = \lambda_i v$, entonces $B \cdot Pv = \lambda_i Pv$, siendo P una matriz inversible tal que $A = P^{-1}BP$ y v el autovector asociado a λ_i de A.

Hay matrices que tienen la propiedad de ser semejantes a una matriz diagonal. Es decir, dada una matriz A, existe una matriz D diagonal que es semejante a la matriz A. Este tipo de matrices se dice que son **diagonalizables** por semejanza, y tienen la propiedad de tener una base de autovectores.

$$A = P^{-1}DP$$
$$\Longleftrightarrow$$
Los autovectores de A forman una base.

Nota: La matriz P se puede construir tomando como columnas a los autovectores de A.

8.3. Matrices con Base de Autovectores

No toda matriz tiene base de autovectores, y comprobar que una matriz sea diagonalizable no resulta fácil, por lo que buscamos alguna propiedad más sencilla que nos permita afirmar que la matriz tiene base de autovectores.

Propiedades

- Una de la propiedades es que si tenemos una matriz $A \in \mathbb{R}^{n\times n}$ simétrica, podemos afirmar que sus autovalores son reales.
- Si A tiene un autovalor real, entonces existe un autovector asociado con coeficientes reales.
- Si A es simétrica y λ^1 y λ^2 son autovalores distintos con v^1 y v^2 autovectores asociados, entonces v^1 y v^2 no solo son linealmente independientes, sino que además son ortogonales.

Por otro lado, hay un resultado que nos dice que si A tiene todos sus autovalores reales, entonces existe $Q \in \mathbb{R}^{n\times n}$ ortogonal tal que $Q^tAQ = T$, con $T \in \mathbb{R}^{n\times n}$ triangular superior. Esto nos dice que A es semejante a una matriz triangular superior, y además la relación de semejanza es vía una matriz ortogonal:

$$Q^tAQ = T$$

Además, si A es simétrica, entonces T es diagonal, los elementos de la diagonal de T son los autovalores, y las columnas de Q los autovectores de A:

$$Q^tAQ = D$$

Esta propiedad nos dice que si A es simétrica, entonces tiene una base ortonormal de autovectores, y que por tanto es diagonalizable por semejanza vía una matriz ortogonal.

8.4. Método de la Potencia

En este punto tenemos una serie de propiedades acerca de los autovalores y autovectores de una matriz, hemos caracterizado cuándo podemos esperar tener una base de autovectores, y las implicaciones que tienen respecto a la diagonalización de la matriz. A continuación, vamos a ver cómo podemos calcular los autovalores de una matriz, aplicando el **método de la potencia**.

Sea $A \in \mathbb{R}^{n\times n}$, $\lambda^1, \cdots, \lambda^n$ sus n autovalores con $v^1, \cdots, v^n$ los autovectores asociados que conforman una base. Además, $|\lambda^1| > |\lambda^2| \geq \cdots \geq |\lambda^n|$. El objetivo del método de la potencia va a ser obtener el **autovalor principal**. Consideremos una base de autovectores de $\mathbf{A}$, $\{\mathbf{v}_1, \cdots, \mathbf{v}_n\}$, ordenados de forma tal que cada $\mathbf{v}_i$ está asociado al autovalor λ_i.

La idea va a ser aplicar de forma iterativa una sucesión que busca converger al autovector principal. Para ello, tomaremos un vector $\mathbf{x}^{(0)} \in \mathbb{R}^n$ que sea una combinación lineal de los autovectores, pero que el coeficiente α_1 asociada a v_1 sea distinto de 0:

$$\mathbf{x}^{(0)} = \alpha_1 \cdot \mathbf{v}_1 + \cdots + \alpha_n \cdot \mathbf{v}_n, \text{ con } \alpha_1 \neq 0.$$

En cada iteración, simplemente multiplicaremos a izquierda por $\mathbf{A}$. Es decir,

$$\begin{aligned}
\mathbf{x}^{(k)} = \mathbf{A} \cdot \mathbf{x}^{(k-1)} &= \mathbf{A}^k \cdot \mathbf{x}^{(0)} \\
&= \mathbf{A}^k \cdot (\alpha_1 \cdot \mathbf{v}_1 + \cdots + \alpha_n \cdot \mathbf{v}_n) \\
&= \alpha_1 \cdot \mathbf{A}^k \cdot \mathbf{v}_1 + \cdots + \alpha_n \cdot \mathbf{A}^k \cdot \mathbf{v}_n \\
&= \alpha_1 \cdot \lambda_1^k \cdot \mathbf{v}_1 + \cdots + \alpha_n \cdot \lambda_n^k \cdot \mathbf{v}_n \\
&= \lambda_1^k \cdot \left(\alpha_1 \cdot \mathbf{v}_1 + \alpha_2 \cdot \left(\frac{\lambda_2}{\lambda_1}\right)^k \cdot \mathbf{v}_2 + \cdots + \alpha_n \cdot \left(\frac{\lambda_n}{\lambda_1}\right)^k \cdot \mathbf{v}_n \right).
\end{aligned}$$

Ahora bien, como para todo $i \in \{1, \ldots, n\}$ se cumple que $|\lambda_1| > |\lambda_i|$, entonces

$$\lim_{k\to\infty} \left(\frac{\lambda_i}{\lambda_1}\right)^k = 0.$$

Si llamamos $\mathbf{r}^{(k)} = \dfrac{\mathbf{x}^{(k)}}{\lambda_1^k}$, tenemos que

$$\lim_{k\to\infty} \mathbf{r}^{(k)} = \lim_{k\to\infty} \left(\alpha_1 \cdot \mathbf{v}_1 + \alpha_2 \cdot \left(\frac{\lambda_2}{\lambda_1}\right)^k \cdot \mathbf{v}_2 + \cdots + \alpha_n \cdot \left(\frac{\lambda_n}{\lambda_1}\right)^k \cdot \mathbf{v}_n \right) = \alpha_1 \cdot \mathbf{v}_1.$$

Ahora bien, ¿cuál es la crítica a este procedimiento? El problema es que la sucesión definida, que converge a la dirección del autovector v_1, depende de λ_1 que es una de las cosas que queríamos encontrar, por lo que no parece muy útil.

Vamos a ver cómo podemos lograr el mismo resultado, sin tener que utilizar a λ_1 durante el procedimiento. Para ello, vamos a considerar una función $\Phi : \mathbb{R}^n \longrightarrow \mathbb{R}$ continua, que tenga la particularidad de que saque escalares afuera:

$$\Phi(\alpha x) = |\alpha| \Phi(x)$$

Luego, vamos a considerar

$$\begin{aligned}
\lim_{k\to\infty} \left|\frac{x^{(k)}}{\Phi(x^{(k)})}\right| &= \lim_{k\to\infty} \left|\frac{\lambda_1^k(\alpha_1 v_1 + \sum \cdots)}{\Phi(\lambda_1^k(\alpha_1 v_1 + \sum \cdots))}\right| \\
&= \lim_{k\to\infty} \left|\frac{\alpha_1 v_1 + \sum \cdots}{\Phi(\alpha_1 v_1 + \sum \cdots)}\right| \\
&= \lim_{k\to\infty} \left|\frac{\alpha_1 v_1}{\Phi(\alpha_1 v_1)}\right| \\
&= \lim_{k\to\infty} \left|\frac{v_1}{\Phi(v_1)}\right|
\end{aligned}$$

Notemos que es necesario que $\alpha_1 \neq 0$, y que además $\Phi(v_1) \neq 0$.

Mientras se cumpla con estas propiedades, Φ continua y saca escalares afuera, entonces se va a cumplir que

$$\lim_{k\to\infty} \left|\frac{x^{(k)}}{\Phi(x^{(k)})}\right| = \lim_{k\to\infty} \left|\frac{v_1}{\Phi(v_1)}\right|$$

con $\mathbf{x}^{(k)} = \mathbf{A}^k \cdot \mathbf{x}^{(0)}$

Un caso particular del método de la potencia es tomar a Φ como la norma 2, obteniendo

$$\lim_{k\to\infty} \left|\frac{x^{(k)}}{\|x^{(k)}\|_2}\right| = \lim_{k\to\infty} \left|\frac{v_1}{\|v_1\|_2}\right|$$

con $\mathbf{x}^{(k)} = \mathbf{A}^k \cdot \mathbf{x}^{(0)}$, y un pseudocódigo sería

Método de la Potencia con $\Phi = \|\bullet\|_2$

Entrada: $\mathbf{q}^{(0)} \in \mathbb{R}^n, \|q^{(0)}\|_2 = 1, \mathbf{A} \in \mathbb{R}^{n\times n}, lim \in \mathbb{N}$
Salida: q
for $k = 1, \ldots, lim$ **do**
  $z = Aq$
  $q = \dfrac{z}{\|z\|_2}$

Una vez hemos obtenido el autovector v_1, que ya se encuentra normalizado, podemos obtener al autovalor λ_1 mediante $q^t Aq$, pues

$$\begin{aligned} Av_1 &= \lambda v_1 \\ \Longrightarrow & \\ v_1^t A v_1 &= \lambda v_1^t v_1 \\ &= \lambda \|v_1\|_2^2 \\ &= \lambda \end{aligned}$$

Notemos que la única hipótesis que el método requiere sobre $\mathbf{x}^{(0)}$ es que α_1, su componente en la dirección del autovector principal, no sea nula. Esto suele ser difícil de garantizar, justamente porque no se conoce dicho autovector. La solución suele ser elegir $\mathbf{x}^{(0)}$ de manera aleatoria, y en caso de que el método no converja, volver a intentarlo nuevamente con otro $\mathbf{x}^{(0)}$.

8.5. Método de Deflación

Si trabajamos con una matriz que, además tiene un autovalor dominante, tiene un segundo autovalor que es mayor estricto que el resto:

$$|\lambda_1| > |\lambda_2| > \cdots \geq |\lambda_n|$$

podemos no solo obtener λ_1, sino que además podemos obtener λ_2, aplicando el **método de deflación**.

Este método consiste en considerar la matriz de reflexión H tal que $Hv_1 = e_1$, entonces

$$HAH^t = \begin{bmatrix} \lambda_1 & a^t \\ 0 & B \end{bmatrix}$$

donde HAH^t es semejante a A, a^t es algún vector fila de $\mathbb{R}^{n-1}$, y B es una matriz de $\mathbb{R}^{n\times n}$.

Veamos que la matriz HAH^t, efectivamente, tiene esta estructura. Para ello, basta con verificar que $col_1(HAH^t) = \lambda_1 e_1$:

$$\begin{aligned} Hv_1 &= e_1, \qquad \text{por definición de } H \\ v_1 &= H^t e_1 \\ \Longrightarrow & \\ HA(H^t e_1) &= HAv_1 \\ &= \lambda_1 H v_1 \\ &= \lambda_1 e_1 \end{aligned}$$

por lo que, efectivamente, $col_1(HAH^t) = HAH^t \cdot e_1 = \lambda_1 e_1$.

Por otro lado, veamos ahora que la submatriz B hereda los $n-1$ autovalores restantes de la matriz A:

Sabemos que

$$\begin{aligned} HA\cancel{H^t} \cdot \cancel{H}v_i &= HAv_i \\ &= \lambda_i H v_i \end{aligned}$$

Es decir, Hv_i es autovector asociado a λ_i de HAH^t para $i = 1, \ldots, n$. Luego, si reescribimos a Hv_i como $\begin{bmatrix} \beta_i \\ w_i \end{bmatrix}$, entonces:

$$\begin{aligned} \lambda_i H v_i &= HAH^t \cdot Hv_i \\ \lambda_i \begin{pmatrix} \beta_i \\ w_i \end{pmatrix} &= \begin{bmatrix} \lambda_1 & a^t \\ 0 & B \end{bmatrix} \begin{pmatrix} \beta_i \\ w_i \end{pmatrix} \\ &= \begin{pmatrix} \lambda_1 \beta_i + a^t w_i \\ B w_i \end{pmatrix} \\ \iff & \\ & \begin{cases} \lambda_i \beta_i = \lambda_1 \beta_i + a^t w_i \\ \lambda_i w_i = B w_i \end{cases} \end{aligned}$$

Notemos que en el caso $i = 1$, como habíamos visto antes, $HAH^t \cdot e_1 = \lambda_1 e_1$, y por lo tanto $\beta_1 = \lambda_1$ y $w_1 = 0$. Es decir, w_1 no es autovector de B. Además, como λ_1 es el autovalor dominante, $\lambda_1 \neq \lambda_i$ para todo $i = 2, \ldots, n$. Por lo tanto, Hv_i es linealmente independiente de $\lambda_1 e_1$ para todo $i = 2, \ldots, n$, es decir $w_i \neq 0$ para todo $i = 2, \ldots, n$. Luego, como $w_i \neq 0$ y $w_i = Bw_i$, podemos asegurar que w_i es autovector de B y λ_i su autovalor asociado.

Luego, esta matriz B va a tener como autovalor dominante a λ_2 de A, por lo que se le puede aplicar el método de la potencia para hallar λ_2. Notemos que si todos los autovalores de A son distintos en módulo, entonces podremos ir aplicando el método de la potencia, en combinación al método de deflación, de forma iterativa, para poder hallar todos los autovalores y autovectores de A.

Una variante del método de deflación consiste en definir

$$\mathbf{A}' = \mathbf{A} - \lambda_1 \cdot \mathbf{u}_1 \cdot \mathbf{u}_1^{\mathrm{T}},$$

donde $\mathbf{u}_1$ es un autovector unitario asociado a λ_1.

La matriz $\mathbf{A}'$ tiene autovalores $0, \lambda_2, \ldots, \lambda_n$, por lo que si $\lambda_2 > \lambda_3$, puede volver a aplicarse el método de la potencia.

8.6. Método de la potencia inversa

El **método de la potencia inversa** es una variante del método de la potencia que permite, dada una matriz inversible $\mathbf{A}$, encontrar su autovalor (y autovector asociado) de módulo mínimo, si el mismo existe y tiene multiplicidad simple.

Se basa en el hecho de que, si los autovalores de $\mathbf{A}$ son

$$\lambda_1, \ldots, \lambda_n,$$

con $|\lambda_1| < |\lambda_i|$ para todo $i \in \{2, \ldots, n\}$, entonces los autovalores de $\mathbf{A}^{-1}$ son

$$\lambda_1^{-1}, \ldots, \lambda_n^{-1},$$

con $|\lambda_1^{-1}| > |\lambda_i^{-1}|$ para todo $i \in \{2, \ldots, n\}$.

Por lo tanto, basta con aplicar el método de las potencias sobre $\mathbf{A}^{-1}$ para obtener $|\lambda_1^{-1}|$.

Una variante interesante del método de la potencia inversa permite, dado un valor $\mu \in \mathbb{R}$, encontrar el autovalor de $\mathbf{A}$ más cercano a μ. Consiste en aplicar el método de la potencia sobre la matriz $(\mathbf{A}-\mu\cdot\mathbf{I})^{-1}$, que tiene como autovalores a

$$(\lambda_1 - \mu)^{-1}, \ldots, (\lambda_n - \mu)^{-1};$$

el autovalor λ_i de $\mathbf{A}$ que minimiza la distancia con μ es también el que maximiza el valor de $(\lambda_i - \mu)^{-1}$.

8.7. Propiedades Varias

Números Complejos

- **Definición**: Un número complejo z es un par ordenado de números reales, denotado por $z = (a, b)$ o $z = a + bi$, donde $i^2 = -1$, la parte real es $Re(z) = a$, y la parte imaginaria es $Im(z) = b$.
- Si $z = a + bi$, entonces $\bar{z} = a - bi$.
- $\overline{z_1 + z_2} = \bar{z_1} + \bar{z_2}$.
- $\overline{z_1 \cdot z_2} = \bar{z_1} \cdot \bar{z_2}$.
- Si $z \in \mathbb{C}$, entonces $|z| = \sqrt{a^2 + b^2}$.
- Si $z \in \mathbb{C}$, entonces $z \cdot \bar{z} = |z|^2$.
- Si $u \in \mathbb{C}^n$, entonces $\|u\|_2 = \sum_{i=1}^{n} |u_i|^2$.
- Si $u, v \in \mathbb{C}^n$, entonces $u \cdot v = \sum_{i=1}^{n} u_i \bar{v_i}$.

Propiedades Equivalentes

- λ es autovalor de A.
- $(\lambda I - A)x = 0$.
- $\exists x \neq 0 \in \mathbb{R}^n$ tal que $Ax = \lambda x$.
- $\det(\lambda I - A) = 0$

Autovalores

- Si A es simétrica definida positiva y λ_i autovalor de A, entonces $\lambda_i > 0$.
- Si A es singular, entonces $\lambda = 0$ es autovalor de A.
- Un autovalor λ puede estar asociado a lo sumo m autovectores linealmente independientes asociados, donde m es la multiplicidad de λ en el polinomio característico.
- Las combinaciones lineales entre autovectores, asociados a un mismo autovalor λ, también son autovector de ese λ. Esto no vale para combinaciones lineales de autovectores asociados a distintos autovalores.
- Si λ es autovalor de AA^T, entonces λ es autovalor de A^TA.

Radio Espectral

Propiedad:El radio espectral de A $\rho(A) \leq \|A\|$ para cualquier norma matricial inducida $\|\bullet\|$.

Demostración: Sea v_i el autovalor unitario asociado al autovalor λ_i de A, y sea $\|\bullet\|$ una norma

vectorial / matricial inducida, entonces

$$
\begin{gathered}
Av_i = \lambda_i v_i \\
\Longrightarrow \\
\begin{cases} \|Av_i\| = \|\lambda_i v_i\| \\ \|Av_i\| \leq \|A\|\|v_i\| \end{cases} \\
\Longrightarrow \\
\|\lambda_i v_i\| \leq \|A\|\|v_i\| \\
|\lambda_i| \leq \|A\|
\end{gathered}
$$

Por lo tanto, $|\lambda_i| \leq \|A\|$ para todo $i = 1, \ldots, n$. Como $\rho(A) = \text{máx}_i\, |\lambda_i|$, entonces $\rho(A) \leq \|A\|$, para toda norma inducida.

Capítulo 9

Descomposición en valores singulares

Este capítulo está dedicado a encontrar una nueva factorización de una matriz. Hasta ahora, las factorizaciones que conocemos son las factorización LU, que proviene de la eliminación gaussiana, y la factorización QR, que proviene de aplicar rotaciones o reflexiones sobre una matriz, hasta obtener una matriz triangular superior. En todos estos casos, se descompone a la matriz como el producto de 2 matrices.

En la factorización en valores singular, vamos a escribir a la matriz $A \in \mathbb{R}^{m\times n}$, con $r = rango(A)$, como el producto de tres matrices:

$$A = U\Sigma V^t$$

con $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ matrices **ortogonales**, y $\Sigma \in \mathbb{R}^{m\times n}$ tal que

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots 0 & \\ 0 & \sigma_2 & \cdots & 0 & 0 & \cdots 0 & \\ \vdots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r & 0 & \cdots 0 & \\ \vdots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_1 & 0 & \cdots & 0 & 0 & \cdots 0 & \end{bmatrix}$$

con $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$, denominados **valores singulares**.

9.1. Buscando la Descomposición en Valores Singulares

Ahora veamos qué características tienen que tener las columnas de las matrices ortogonales U y V, en caso de que exista esta descomposición. Comenzamos planteando que queremos que

$$\begin{aligned} A &= U\Sigma V^t \\ AV &= U\Sigma \end{aligned}$$

$$A \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_r & \\ & & & \\ & & & \end{bmatrix}$$

$$\iff$$

$$\begin{cases} Av_i = \sigma_i u_i & \text{si } i = 1, \ldots, r \\ Av_i = 0 & \text{si } i = r+1, \ldots, n \end{cases}$$

Esto nos da una relación que debe existir entre las columnas de V y las columnas de U. Este mismo procedimiento lo podemos hacer para A^t:

$$
\begin{aligned}
A^t &= V\Sigma^t U^t \\
A^t U &= V\Sigma^t
\end{aligned}
$$

$$
A^t \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_r & \\ & & & \\ & & & \end{bmatrix}
$$

$$
\iff
$$

$$
\begin{cases} A^t u_i = \sigma_i v_i & \text{si } i = 1, \ldots, r \\ A^t u_i = 0 & \text{si } i = r+1, \ldots, m \end{cases}
$$

Entonces, obtenemos otras dos relaciones que debe haber entre los vectores columna de V y los vectores columna de U. Ahora, vamos a seguir buscando propiedades:

$$
\begin{cases} A^t A v_i = \sigma_i \cdot \underbrace{A^t u_i} = \sigma_i^2 v_i & \text{para } i = 1, \ldots, r \\ A^t A v_i = 0 & \text{para } i = r+1, \ldots, n \end{cases}
$$

Esto nos está diciendo que v_i es autovector de $A^t A$ correspondiente al autovalor σ_i^2, para $i = 1, \ldots, r$, y para $i = r+1, \ldots, n$, v_i es autovector relacionado al autovalor nulo.

Es decir, si esta U y esta V existen, sabemos entonces que $\boxed{v_1, \ldots, v_n}$ tiene que ser base ortonormal de autovectores de $A^t A$, al ser estos vectores las columnas de una matriz ortogonal. Por otro lado, sabemos que la base de autovectores de $A^t A$ existe, al ser esta una matriz simétrica. Luego, la base ortonormal de autovectores de $A^t A$ son candidatas a ser las columnas de la matriz ortogonal V.

Por otro lado, como $\sigma_i^2 = \lambda_i$, con $\sigma_i > 0$ y λ_i autovalor de $A^t A$ para $i = 1, \ldots, r$, entonces podemos definir a $\boxed{\sigma_i = \sqrt{\lambda_i}}$, que está bien definido al ser $A^t A$ una matriz semi-definida positiva, y por lo tanto $\lambda_i \geq 0$.

Veamos ahora de dónde sacamos el resto de las columnas de U, teniendo que cumplir las propiedades ya establecidas para que la factorización realmente exista. Por un lado tenemos la propiedad que nos dice que debe existir una relación entre las primeras r columnas de V y de U:

$$
A v_i = \sigma_i u_i \quad \text{para } i = 1, \ldots, r
$$

Como $\sigma_i > 0$, al ser los valores singulares que corresponden a las raíces cuadradas de los autovalores de $A^t A$ que no son nulos, entonces podemos dividir por σ_i obteniendo:

$$
\frac{A v_i}{\sigma_i} = u_i \text{ para } i = 1, \ldots, r
$$

Luego, como esta propiedad debe cumplirse, proponemos definir a u_i para $i = 1, \ldots, r$ tal que $u_i = \frac{A v_i}{\sigma_i}$. Tenemos que comprobar que esta definición resulte en vectores ortogonales entre sí y de norma 2 igual a 1, es decir que forman un conjunto ortonormal.

Veamos que, efectivamente, $\mathbf{u}_1, \ldots, \mathbf{u}_r$ forman un conjunto ortonormal:

- $\|\mathbf{u}_i\|_2^2 = \left(\dfrac{\mathbf{A} \cdot \mathbf{v}_i}{\sigma_i}\right)^{\mathrm{T}} \cdot \left(\dfrac{\mathbf{A} \cdot \mathbf{v}_i}{\sigma_i}\right) = \dfrac{\mathbf{v}_i^{\mathrm{T}} \cdot \overbrace{\mathbf{A}^{\mathrm{T}} \cdot \mathbf{A} \cdot \mathbf{v}_i}^{=\lambda_i v}}{\sigma_i^2} = \dfrac{\lambda_i \ \overbrace{\mathbf{v}_i^t \mathbf{v}_i}^{\|v_i\|_2 = 1}}{\lambda_i} = 1.$

- Si $i \neq j$, entonces $\mathbf{u}_i^{\mathrm{T}} \cdot \mathbf{u}_j = \left(\dfrac{\mathbf{A} \cdot \mathbf{v}_i}{\sigma_i}\right)^{\mathrm{T}} \cdot \left(\dfrac{\mathbf{A} \cdot \mathbf{v}_j}{\sigma_j}\right) = \dfrac{\mathbf{v}_i^{\mathrm{T}} \cdot \mathbf{A}^{\mathrm{T}} \cdot \mathbf{A} \cdot \mathbf{v}_j}{\sigma_i \cdot \sigma_j}$

$$= \lambda_j \frac{\overbrace{\mathbf{v}_i^{\mathrm{T}} \cdot \mathbf{v}_j}^{v_i \perp v_j}}{\sigma_i \cdot \sigma_j} = 0.$$

Con esto hemos caracterizado las primeras r columnas de U. Veamos cómo podemos caracterizar las que nos faltan. Para ello, utilizaremos un resultado del álgebra lineal que nos dice

$$\mathrm{Im}(\mathbf{A}) \oplus \mathrm{Nu}(\mathbf{A}^t) = \mathbb{R}^m$$

donde la dimensión de la imagen de A es $dim(\mathrm{Im}(A)) = r$, y por lo tanto la dimensión del núcleo de A^t es $dim(\mathrm{Nu}(A)) = m - r$.

Como definimos a los $u_i = \frac{Av_i}{\sigma_i}$ para $i = 1, \ldots, r$, entonces $u_i \in \mathrm{Im}(A)$, y por tanto conforman una base de la imagen de A. Luego, como el núcleo de A^t está en suma directa con la imagen de A, podemos completar a la matriz U con una base ortonormal del espacio $\mathrm{Nu}(A^t)$, obteniendo así las $m - r$ columnas restantes de la matriz U ortogonal.

Notemos que en vez de analizar los autovectores de A^tA, pudimos haber analizado los autovectores de AA^t, caracterizando a las columnas de U como las base ortonormal de autovectores de AA^t.

En resumen,

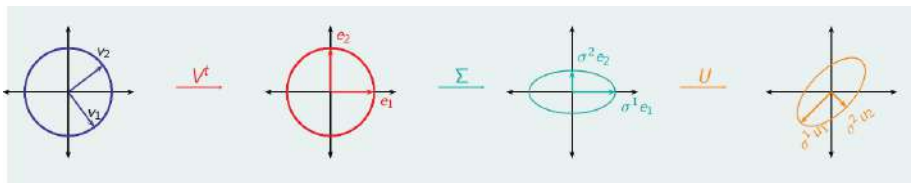
- $v_1, \ldots, v_n$ autovectores de A^tA, columnas de la matriz V.
- $u_q, \ldots, u_n$ autovectores de AA^t, columnas de U.
- $\sigma_i = \sqrt{\lambda_i}$ siendo λ_i el i-ésimo autovalor de A^tA ($\lambda_1 \geq \lambda_2 \geq \ldots \lambda_r$).

y la "receta" para obtener la descomposición en valores singulares de una matriz A sería:

- Hallar los autovectores y autovalores de A^tA.
- Calcular U según: $u_i = \frac{Av_i}{\sigma_i}$ para $i = 1, \ldots, r$.
- Completar el resto de las columnas de U con una base ortonormal del $Nu(A^t)$.

9.2. Interpretación geométrica

Vamos a considerar la circunferencia de radio 1, y vamos a identificar dentro de esa circunferencia a los vectores v_1, v_2 que conforman las columnas de V. Si tenemos $A = U\Sigma V^t$, ¿qué es hacer $Av_1 = U\Sigma V^t v_1$?



De esta manera, podemos observar que primero fue una rotación, luego un estiramiento o un achicamiento sobre los ejes obteniendo una elipse, y por último rotamos esta elipse.

9.3. Propiedades Importantes

Veamos ahora algunas propiedades de los valores singulares. La primera propiedad que vamos a ver es que la norma matricial inducida por la norma vectorial 2 es igual al valor singular más grande, es decir

$$\|A\|_2 = \sigma_1$$

Otra propiedad interesante es que podemos caracterizar al número de condición basado en la norma 2 a partir de los valores singulares de A, de la siguiente manera:

$$\kappa_2(A) = \frac{\sigma_1}{\sigma_n}$$

Por lo tanto, cuanto mayor sea la diferencia entre estos valores, peor condicionada estará la matriz.

Otra propiedad más es que la norma de Frobenius es igual a

$$\|A\|_F = \sqrt{(\sigma_1)^2 + \cdots + (\sigma_r)^2}$$

9.4. Propiedades Varias

- $A = U\Sigma V^T$.
- $AA^T = U\Sigma\Sigma^T U^T$.
- $A^T A = V\Sigma^T\Sigma V^T$.
- Si A inversible, entonces $\kappa_2(A) = \frac{\sigma_1}{\sigma_n}$.
- $\|A\|_2 = \sigma_1$.
- Si A es inversible, entonces los valores singulares de A^{-1} son $\frac{1}{\sigma_n} \geq \cdots \geq \frac{1}{\sigma_1}$.

Capítulo 10

Métodos Iterativos

Este capítulo está dedicado a presentar los **métodos iterativos** para resolver sistemas de ecuaciones lineales. Los métodos que conocemos para resolver un sistema de ecuaciones, hasta el momento, son la eliminación gaussiana, la factorización LU, y la factorización QR. Cualquiera de estos métodos nos asegura que, en una cantidad finita de pasos, obtenemos la solución del sistema, en un orden cúbico. Contrariamente a los métodos exactos o directos, existen los métodos iterativos que buscan generar una sucesión de vectores tal que, bajo ciertas hipótesis, converja a la solución del sistema.

Es decir, un método iterativo es un procedimiento para la resolución de un problema que se basa en construir una sucesión de elementos $\{x_k\}_{k\in\mathbb{N}}$ tal que $\lim_{k\to\infty} x_k = x^*$, donde x^* es una solución del problema que se quiere resolver. Esto permite obtener un algoritmo que se aproxime progresivamente a una solución calculando, en cada paso, el $k+1$-ésimo término de la sucesión a partir del k-ésimo. De esta forma, mediante sucesivas iteraciones, puede lograrse una aproximación cada vez mejor a una solución del problema.

En el caso de la resolución de un sistema de ecuaciones lineales $\mathbf{A}\cdot\mathbf{x} = \mathbf{b}$, se busca una sucesión de vectores $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}}$ que converja a una solución del sistema, es decir, a un valor $\mathbf{x}^*$ tal que $\mathbf{A}\cdot\mathbf{x}^* = \mathbf{b}$.

En particular, los métodos con los que trabajaremos consisten en reescribir el sistema como

$$\mathbf{x} = \mathbf{c} + \mathbf{T}\cdot\mathbf{x}$$

para ciertos $\mathbf{T}\in\mathbb{R}^{n\times n}$ y $\mathbf{c}\in\mathbb{R}^n$, y definir la iteración

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

partiendo de algún $\mathbf{x}^{(0)}$ arbitrario. Notemos que si una sucesión de este tipo converge a algún vector $\mathbf{x}^*$, dicho vector deberá ser solución del sistema.

Ya hemos visto cómo resolver sistemas lineales en forma directa y exacta. La pregunta lógica es ¿por qué buscar una solución iterativa al problema si ya tenemos una directa? Para sistemas de ecuaciones pequeños, los métodos iterativos resultan más lentos que los directos, pues demandan más tiempo para realizar las suficientes iteraciones de modo de aproximar con exactitud la solución. Sin embargo, en dos situaciones los métodos iterativos resultan una mejor opción:

- **Matrices Ralas**: Aquí los métodos iterativos son más eficientes tanto en términos temporales como espaciales. Si la matriz del sistema es rala, los métodos iterativos son compatibles con la utilización de representaciones adecuadas que reduzcan el espacio y tiempo de las operaciones, mientras que los métodos directos no. Por ejemplo, la eliminación gaussiana opera con filas completas, haciendo desaparecer los ceros presentes inicialmente en una fila. Otro ejemplo es la factorización LU que, aunque A sea rala, no asegura que L y U sean ralas.

- **Sistemas de ecuaciones muy grandes**: Dado que la solución se aproxima mediante iteraciones sucesivas, la cantidad de iteraciones necesarias para obtener una aproximación tan buena como se

desee depende de nuestro criterio. Para sistemas grandes, acotar la cantidad de iteraciones es un factor determinante en el costo temporal.

10.1. Método de Jacobi

Vamos a comenzar presentando el primer método iterativo, que es el **método de Jacobi**. El método de Jacobi va a poder ser aplicado únicamente a matrices que tengan los elementos de la diagonal distintos de 0, es decir $a_{ii} \neq 0$ para todo $i = 1, \ldots, n$.

Vamos a comenzar con un vector inicial $x^{(0)} \in \mathbb{R}^n$ y, para generar el siguiente vector, vamos a considerar la primer ecuación del sistema:

$$a_{1,1} \cdot x_1^{(0)} + \cdots + a_{1,n} \cdot x_n^{(0)} = b_1$$

Luego, vamos a fijar los valores de las variables $x_2, \ldots, x_n$ en los valores de $x^{(0)}$, y despejamos x_1 de tal manera que la primera ecuación se satisfaga por igualdad:

$$x_1^{(1)} = \frac{b_1 - a_{12}x_2^{(0)} - \cdots - a_{1n}x_n^{(0)}}{a_{11}}$$

Vamos a tomar ahora la segunda ecuación:

$$a_{2,1} \cdot x_1^{(0)} + \cdots + a_{2,n} \cdot x_n^{(0)} = b_1$$

Entonces, fijemos las variables $x_1, x_3, \ldots, x_n$ en los valores de $x^{(0)}$ para despejar a x_2, de tal manera que esta ecuación se satisfaga por igualdad:

$$x_2^{(1)} = \frac{b_2 - a_{21}x_1^{(0)} - \cdots - a_{2n}x_n^{(0)}}{a_{22}}$$

Nuevamente, fijadas ciertas variables, despejamos una para que la ecuación se satisfaga por igualdad.

Si vamos a la i-ésima ecuación

$$a_{i,1} \cdot x_1 + \cdots + a_{i,i} \cdot x_i + \cdots + a_{i,n} \cdot x_n = b_i$$

y, si despejamos como venimos haciendo, obtenemos

$$x_i^{(1)} = \frac{1}{a_{ii}} \cdot \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(0)} \right)$$

De esta manera, hemos obtenido las n coordenadas que vamos a utilizar para definir el vector $x^{(1)}$. Luego, la idea es proceder, para cada $k = 1, 2, \ldots$ considerar al vector $\mathbf{x}^{(k-1)} = (x_0^{(k-1)}, \ldots, x_n^{(k-1)})$ generado en la iteración anterior. Luego, se recorren en orden las ecuaciones del sistema

$$a_{i,1} \cdot x_1 + \cdots + a_{i,i} \cdot x_i + \cdots + a_{i,n} \cdot x_n = b_i$$

y, para cada una de ellas, se despeja $x_i^{(k)}$ reemplazando las demás variables por los valores correspondientes de $\mathbf{x}^{(k-1)}$. Es decir, se define $x_i^{(k)}$ de modo que

$$a_{i,1} \cdot x_1^{(k-1)} + \cdots + a_{i,i} \cdot x_i^{(k)} + \cdots + a_{i,n} \cdot x_n^{(k-1)} = b_i$$

y por tanto $x_i^{(k)}$:

$$x_i^{(k)} = \frac{1}{a_{i,i}} \cdot \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(a_{i,j} \cdot x_j^{(k-1)} \right) \right).$$

Notemos que para que la iteración esté bien definida, $\mathbf{A}$ no debe tener ceros en la diagonal.

Esta es una metodología que nos está generando una sucesión, y por tanto nos gustaría saber si esta sucesión converge a la solución del sistema

$$\{x^{(k)}\} \xrightarrow{k\to\infty} x^*$$

Para simplificar este análisis, primero veamos cómo podemos expresar esta metodología de forma matricial. Para ello, vamos a considerar escribir a la matriz A como $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$, donde

$$\mathbf{D} = \begin{bmatrix} a_{1,1} & & & \mathbf{0} \\ & a_{2,2} & & \\ & & \ddots & \\ \mathbf{0} & & & a_{n,n} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} & & & \\ -a_{2,1} & & \mathbf{0} & \\ \vdots & \ddots & & \\ -a_{n,1} & \cdots & -a_{n,n-1} & \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} & -a_{1,2} & \cdots & -a_{1,n} \\ & & \ddots & \vdots \\ & \mathbf{0} & & -a_{n-1,n} \\ & & & \end{bmatrix}$$

A partir de esta escritura, teniendo en cuenta que D es una matriz diagonal inversible, podemos reescribir al sistema $Ax = b$ de la siguiente manera:

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

Luego, llegamos a que la solución de $Ax = b$ es la solución de este sistema

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

pero esta expresión es la que nos caracteriza matricialmente las iteradas del método de Jacobi. Para comprobar esto, calculemos $x^{(k)}$ en términos de los elementos de A y B y veamos que coincide con el algoritmo de Jacobi. Se tiene

$$D^{-1}b = \begin{pmatrix} \frac{1}{a_{11}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{a_{nn}} \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix}$$

$$D^{-1}(L+U) = \underbrace{\begin{pmatrix} \frac{1}{a_{11}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{a_{nn}} \end{pmatrix}}_{\text{multiplica la fila } i \text{ por } 1/a_{ii}} \begin{pmatrix} 0 & & -a_{ij} \\ & \ddots & \\ -a_{ij} & & 0 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & \cdots & -\frac{a_{2n}}{a_{22}} \\ \vdots & & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \cdots & 0 \end{pmatrix}$$

Entonces

$$x^{(k)} = D^{-1}b + D^{-1}(L+U)x^{(k-1)} = \begin{pmatrix} \frac{1}{a_{11}}\left(b_1 - \sum_{j\neq 1}^{n} a_{1j}x_j^{(k-1)}\right) \\ \vdots \\ \frac{1}{a_{nn}}\left(b_n - \sum_{j\neq n}^{n} a_{nj}x_j^{(k-1)}\right) \end{pmatrix}$$

como queríamos ver.

Por lo tanto, podemos asegurar que si el método de Jacobi converge, lo hará a una solución de $\mathbf{A}\cdot\mathbf{x} = \mathbf{b}$. Notemos que todavía no sabemos si este método converge, solamente estamos diciendo que,

en caso de que converja, converge a la solución del sistema. Nuevamente, recordemos que es necesario que $a_{ii} \neq 0 \forall i = 1, \ldots, n$ para poder aplicar el método.

Escribiendo el algoritmo en forma de pseudocódigo, se tiene

Método de Jacobi

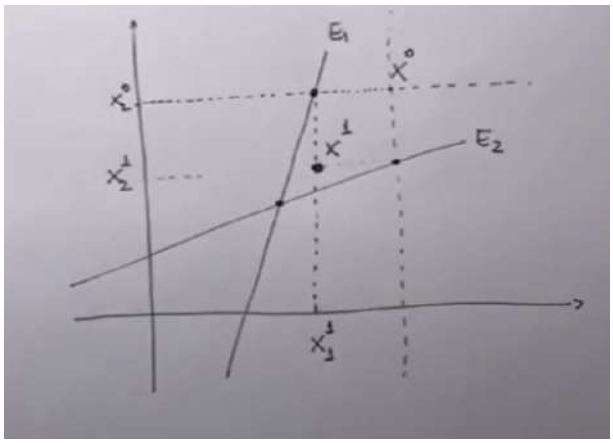
Entrada: $\mathbf{A} \in \mathbb{R}^{n \times n}$ sin ceros en la diagonal, $\mathbf{b}$, $\mathbf{x}^{(0)} \in \mathbb{R}^n$ arbitrarios, lim criterio de corte.
Salida: $\mathbf{x}^*$ solución del sistema $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$.

for $k = 1, 2, \ldots lim$ **do**
  for $i = 1, \ldots, n$ **do**
    $$x_i^{(k)} \leftarrow \frac{1}{a_{i,i}} \cdot \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(a_{i,j} \cdot x_j^{(k-1)} \right) \right)$$

Podemos observar que el costo de cada iteración es de orden cuadrático, por lo que el costo total del método nos queda $O(k \cdot n^2$, con k cantidad de iteraciones como criterio de corte.

10.1.1. Interpretación Geométrica

Vamos a dar una interpretación geométrica de lo que está haciendo el método en $\mathbb{R}^2$. Vamos a suponer que estamos intentando resolver un sistema de ecuaciones de 2×2, por lo que cada una de las ecuaciones E_1, E_2 caracteriza a una recta, y la solución del sistema que estamos buscando no es otra cosa que la intersección de estas rectas:



El método de Jacobi nos dice de considerar la primera ecuación, fijar la segunda coordenada $x_2^{(0)}$, y determinar la primer coordenada $x_1^{(1)}$ de tal manera que la primera ecuación se satisfaga. Luego, para determinar la segunda coordenada $x_2^{(1)}$, fijamos la primer coordenada $x_1^{(0)}$, y luego determinamos el valor de $x_2^{(1)}$ de tal manera que se satisfaga la segunda ecuación.

10.2. Método de Gauss-Seidel

El segundo método iterativo que vamos a ver es el método de **Gauss-Seidel**. El método de Gauss-Seidel es similar al de Jacobi en cuanto a que plantea partir de un vector inicial $x^{(0)}$ y va generando nuevos punto de una sucesión generada a partir de las ecuaciones del sistema, pero con una leve diferencia.

Al igual que en Jacobi, la primer coordenada la va a actualizar de tal manera que se cumpla la

primer ecuación, manteniendo fija el resto de las variables:

$$x_1^{(1)} = \frac{1}{a_{1,1}} \cdot \left(b_1 - \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(a_{1,j} \cdot x_j^{(0)} \right) \right).$$

Sin embargo, para la segunda coordenada, en lugar de utilizar todas las coordenadas de $x^{(0)}$, reemplaza el valor de la primera coordenada $x_1^{(0)}$ por la coordenada que acaba de determinar:

$$x_2^{(1)} = \frac{1}{a_{2,2}} \cdot \left(b_2 - a_{2,1} x_1^{(1)} - \sum_{\substack{j=3 \\ j \neq i}}^{n} \left(a_{2,j} \cdot x_j^{(0)} \right) \right).$$

De alguna manera, va a ir usando las coordenadas actualizadas, a medidas que las va obteniendo.

Cuando estemos en el caso general, y queramos determinar la i-ésima coordenada a partir de la i-ésima ecuación, vamos a dejar fijas las coordenadas de $i+1, \ldots, n$ respecto al $x(0)$, pero las coordenadas de $1, \ldots, i-1$, que ya han sido determinadas, van a tomar el valor actualizado:

$$x_i^{(1)} = \frac{1}{a_{i,i}} \cdot \left(b_i - \sum_{j=1}^{i-1} \left(a_{i,j} \cdot x_j^{(1)} \right) - \sum_{j=i+1}^{n} \left(a_{i,j} \cdot x_j^{(0)} \right) \right),$$

Luego, la diferencia esencial, con respecto al método de Jacobi, es que utiliza no solo las coordenadas del punto inicial, sino que a medida que actualiza coordenadas, utiliza estas últimas. Este es el caso inicial para pasar de $x^{(0)}$ a $x^{(1)}$, y el caso general para pasar de $x^{(k)}$ a $x^{(k+1)}$ nos queda

La iteración queda planteada como

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

Recordemos que este método asume que la matriz A tiene sus elementos de la diagonal no nulos, porque sino no podríamos dividir por a_{ii}.

Ahora, vamos a darle una expresión matricial al método de Gauss-Seidel, de la misma manera que hicimos con el método de Jacobi. Para ello, vamos a partir del sistema $Ax = b$, y vamos a reescribir a A como $A = D - L - U$, donde

$$\mathbf{D} = \begin{bmatrix} a_{1,1} & & & \mathbf{0} \\ & a_{2,2} & & \\ & & \ddots & \\ \mathbf{0} & & & a_{n,n} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} & & \mathbf{0} \\ -a_{2,1} & & \\ \vdots & \ddots & \\ -a_{n,1} & \cdots & -a_{n,n-1} \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} -a_{1,2} & \cdots & -a_{1,n} \\ & \ddots & \vdots \\ \mathbf{0} & & -a_{n-1,n} \\ & & \end{bmatrix}$$

Luego, a partir de esta reescritura, teniendo en cuenta que D es una matriz diagonal inversible, podemos reescribir al sistema de la siguiente manera:

$$\begin{gathered} Ax = b \\ (D - L - U)x = b \\ (D - L)x - Ux = b \\ (D - L)x = b + Ux \\ x = (D - L)^{-1} b + (D - L)^{-1} U x \end{gathered}$$

Esta última equivalencia vale debido a que $D - L$ es triangular inferior y $(D-L)_{ii} = a_{ii} \neq 0$, por lo que $(D-L)$ es inversible. Luego, decimos que las iteradas de Gauss-Seidel se pueden expresar de forma matricial como:

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

Veamos que esta definición coincide con el algoritmo de Gauss - Seidel. Vamos a despejar las componentes de $x^{(k)}$ a partir de la igualdad $(D-L)x^{(k)} = b + Ux^{(k-1)}$. Tenemos

$$b + Ux^{(k-1)} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} + \begin{pmatrix} 0 & & -a_{ij} \\ & \ddots & \\ 0 & & 0 \end{pmatrix} \begin{pmatrix} x_1^{(k-1)} \\ \vdots \\ x_n^{(k-1)} \end{pmatrix} = \begin{pmatrix} b_1 - \sum_{j=2}^{n} a_{1j}x_j^{(k-1)} \\ b_2 - \sum_{j=3}^{n} a_{2j}x_j^{(k-1)} \\ \vdots \\ b_n \end{pmatrix}$$

Entonces

$$(D-L)x^{(k)} = b + Ux^{(k-1)} \Leftrightarrow \begin{pmatrix} a_{11} & & 0 \\ \vdots & \ddots & \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} x^{(k)} = \begin{pmatrix} b_1 - \sum_{j=2}^{n} a_{1j}x_j^{(k-1)} \\ b_2 - \sum_{j=3}^{n} a_{2j}x_j^{(k-1)} \\ \vdots \\ b_n \end{pmatrix}$$

$$\Leftrightarrow x^{(k)} = \begin{pmatrix} \frac{1}{a_{11}}\left(b_1 - \sum_{j=2}^{n} a_{1j}x_j^{(k-1)}\right) \\ \frac{1}{a_{22}}\left(b_2 - a_{21}x_1^{(k)} - \sum_{j=3}^{n} a_{2j}x_j^{(k-1)}\right) \\ \vdots \\ \frac{1}{a_{nn}}\left(b_n - \sum_{j=1}^{n-1} a_{nj}x_j^{(k)}\right) \end{pmatrix}$$

que es lo que queríamos ver.

Escribiendo el algoritmo en forma de pseudocódigo, se tiene Podemos observar que el costo de cada

Método de Gauss - Seidel

```
Definir x^(0);
for k = 1 ... lim do
    for i = 1 ... n do
        x_i^(k) = 1/a_ii ( b_i - Σ_{j=1}^{i-1} a_ij x_j^(k) - Σ_{j=i+1}^{n} a_ij x_j^(k-1) );
```

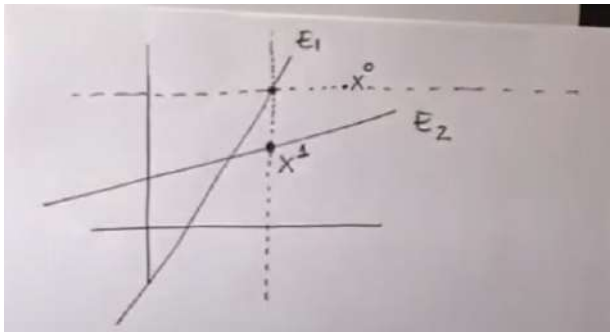
iterada de Gauss-Seidel es de orden $O(n^2)$, siendo el costo total de orden $O(k \cdot n^2)$, con k la cantidad de iteraciones producto del criterio de corte.

La convergencia del método de Gauss-Seidel está garantizada para matrices estrictamente diagonal-dominantes por filas, al igual que con el método de Jacobi. Además, también se puede asegurar su convergencia para matrices simétricas definidas positivas. Esto último no es cierto para el método de Jacobi.

Por último, otra ventaja que presenta el método de Gauss-Seidel sobre el de Jacobi es que, como solo es necesario tener al mismo tiempo parte de la iteración actual y parte de la anterior, el algoritmo puede realizarse utilizando un único arreglo, y por tanto requiere la mitad del espacio.

10.2.1. Interpretación Geométrica

Veamos cómo podemos interpretar geométricamente el método de Gauss-Seidel en $\mathbb{R}^2$. Vamos a suponer que estamos intentando resolver un sistema de ecuaciones de 2×2, por lo que cada una de las ecuaciones E_1, E_2 caracteriza a una recta, y la solución del sistema que estamos buscando no es otra cosa que la intersección de estas rectas:



Para actualizar la primer coordenada dejamos fija la segunda, y tomamos la primera coordenada de manera que se satisfaga la primera ecuación. Luego, vamos a determinar la segunda coordenada de tal manera que se satisfaga la segunda ecuación, pero ahora la que vamos a dejar fija es la primer coordenada $x_1^{(1)}$, y no $x_1^{(0)}$.

Considerando que el método de Gauss-Seidel trabaja con información actualizada, es razonable suponer que este converge más rápido que el método de Jacobi. Sin embargo, esto no es necesariamente cierto, incluso puede ocurrir que uno converja a la solución, mientras que el otro no lo haga. Veamos algunos ejemplos para notar que esto, efectivamente, depende del sistema:

En este ejemplo, el método de Jacobi converge a la solución del sistema, mientras que el método de Gauss-Seidel no converge:

$$\begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ 1 \end{bmatrix} \text{ Solución } x = (6, -4, -3)$$

Jacobi

$x^0 = (0, 0, 0)$
$x^1 = (4, -1, 1)$
$x^2 = (8, -6, -5)$
$x^3 = (6, -4, -3)$
$x^4 = (6, -4, -3)$
$x^5 = (6, -4, -3)$
$x^6 = (6, -4, -3)$

Gauss-Seidel

$x^0 = (0, 0, 0)$
$x^1 = (4, -5, 3)$
$x^2 = (20, -24, 9)$
$x^3 = (70, -80, 21)$
$x^4 = (206, -228, 45)$
$x^5 = (550, -596, 93)$
$x^6 = (1382, -1476, 189)$
$x^7 = (3334, -3524, 381)$
$x^8 = (7814, -8196, 765)$
$x^9 = (17926, -18692, 1533)$
$x^{10} = (40454, -41988, 3069)$

En este otro caso, podemos observar el método de Gauss-Seidel converge a la solución del sistema, mientras que el método de Jacobi no converge:

$$\begin{bmatrix} 1 & -0.5 & 0.5 \\ 1 & 1 & 1 \\ -0.5 & -0.5 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ 1 \end{bmatrix} \text{ Solución } x = \left(\tfrac{19}{9}, \tfrac{-31}{9}, \tfrac{1}{3}\right)$$

Jacobi

$x^0 = (0., 0, 0)$
$x^1 = (4, -1, 1)$
$x^2 = (3.00, -6.00, 2.50)$
$x^3 = (-0.25, -6.50, -0.50)$
$x^4 = (1.00, -0.25, -2.38)$
$x^5 = (5.06, 0.38, 1.38)$
$x^{10} = (4.28, -9.68, 5.62)$
$x^{20} = (-4.51, 15.60, -15.81)$
$x^{30} = (22.32, -61.55, 49.60)$
$x^{40} = (-59.57, 173.88, -150.01)$
$x^{45} = (258.10, 327.84, 90.68)$
$x^{50} = (190.34, -544.60, 459.14)$

Gauss-Seidel

$x^0 = (0, 0, 0)$
$x^1 = (4, -5, 0.5)$
$x^2 = (1.25, -2.75, 0.25)$
$x^3 = (2.5, -3.75, 0.375)$
$x^4 = (1.9375, -3.3125, 0.3125)$
$x^5 = (2.1875, -3.5, 0.34375)$
$x^{10} = (2.11035, -3.44433, 0.33300)$
$x^{15} = (2.11108, -3.44439, 0.33334)$
$x^{20} = (2.11111, -3.44444, 0.33333)$
$x^{25} = (2.11111, -3.44444, 0.33333)$
$x^{30} = (2.11111, -3.44444, 0.33333)$

En conclusión, no hay un método que supere al otro en cuanto a condiciones de convergencia. Hay sistemas de ecuaciones en los que ambos convergen, sistemas de ecuaciones en los cuales uno converge y el otro no converge.

10.3. Análisis de convergencia

Hasta ahora propusimos dos iteraciones distintas, aunque nunca probamos que efectivamente convergieran a una solución. Veamos cómo podemos determinar la convergencia de estos métodos. Para ello, primero observemos que tanto Jacobi como Gauss - Seidel, sus formas matriciales mantienen una cierta estructura más general:

$$\begin{aligned} \text{Gauss-Seidel: } \mathbf{x}^{(k+1)} &= (D-L)^{-1}U \cdot \mathbf{x}^{(k)} + (D-L)^{-1}b \\ \text{Jacobi: } \mathbf{x}^{(k+1)} &= D^{-1}(L+U) \cdot \mathbf{x}^{(k)} + D^{-1}b \\ \text{Esquema general: } \mathbf{x}^{(k+1)} &= \mathbf{T} \cdot \mathbf{x}^{(k)} + \mathbf{c} \end{aligned}$$

donde $T \in \mathbb{R}^{n \times n}$ y $c \in \mathbb{R}^n$ no dependen de $x^{(k)}$.

Entonces, bajo esta estructura, lo que queremos determinar qué propiedades tienen que cumplir para que las sucesiones de la forma

$$x^{(k+1)} = Tx^{(k)} + c$$

convergen a la solución del sistema

$$x^* = Tx^* + c$$

Primero, vamos a necesitar de algunos resultados del álgebra lineal:

- **Definición**: A es una matriz convergente si $\lim\limits_{k \to \infty} A^k = 0$.
- **Definición**: Llamamos **radio espectral** de una matriz a su autovalor de módulo máximo, es decir

$$\rho(\mathbf{A}) = \text{máx}\{|\lambda| : \lambda \text{ es un autovalor de } \mathbf{A}\}.$$

- Propiedad: A es convergente

$$\begin{aligned}
&\Leftrightarrow \rho(A) < 1 \\
&\Leftrightarrow \lim_{k\to\infty} \|A^k\| = 0 \\
&\Leftrightarrow \lim_{k\to\infty} A^k x = 0 \forall x \in \mathbb{R}^n
\end{aligned}$$

- Propiedad: Si $\rho(A) < 1$, entonces $I - A$ es inversible y $\displaystyle\sum_{k=0}^{\infty} A^k = (I-A)^{-1}$.

<u>Nota</u>: Buena fuente bibliográfica: *Analysis of Numerical Methods E. Isaacson.*

Luego, utilizaremos estas propiedades para demostrar que la sucesión $\{x^{(k)}\}$ definida por $x^{(k+1)} = Tx^{(k)} + c$ va a converger, sin importar el punto inicial $x^{(0)}$, si y solo si $\rho(T) < 1$.

$\Leftarrow$)

$$\begin{aligned}
x^{(k+1)} &= Tx^{(k)} + c \\
&= T(Tx^{(k-1)} + c) + c \\
&= T^2 x^{(k-1)} + Tc + c \\
&= T^2(Tx^{(k-2)} + c) + c \\
&= T^3 x^{(k-2)} + T^2 c + Tc + c \\
&\vdots \\
&= T^{k+1} x^{(0)} + T^k c + \cdots + Tc + c \\
\Longrightarrow
\end{aligned}$$

$$\begin{aligned}
\lim_{k\to\infty} x^{(k+1)} &= \lim_{k\to\infty} \underbrace{T^{k+1}}_{\to 0} x^{(0)} + T^k c + \cdots + Tc + c \\
&= \lim_{k\to\infty} 0 + \Big(\sum_{k=0}^{\infty} T^k\Big) \cdot c \\
&= (I-T)^{-1} \cdot c
\end{aligned}$$

Por lo tanto, el $\lim_{k\to\infty} x^{(k+1)}$ existe, y converge a $x^* = (I-T)^{-1} \cdot c$, que es la solución del sistema $x = Tx + c$, ya que

$$\begin{aligned}
x &= (I-T)^{-1} \cdot c \\
(I-T)x &= c \\
x &= Tx + c
\end{aligned}$$

Notemos que en todo durante este procedimiento no importa cuánto vale $x^{(0)}$, ya que la sucesión converge a x^* independientemente de su valor. Continuamos con la demostración para el otro lado.

$\Rightarrow$) Ahora queremos ver que si la sucesión $\{x^{(k)}\}$ converge para todo $x^{(0)}$ inicial, entonces $\rho(T) < 1$. Para comprobar esto, vamos a probar que

$$\lim_{k\to\infty} T^k z = 0 \forall z \in \mathbb{R}^n$$

al ser esta una propiedad equivalente.

Como estamos suponiendo que la sucesión converge a la solución del sistema x^*, para cualquier $x^{(0)}$ inicial. En particular, tomaremos $x^{(0)} = x^* - z$, siendo z un vector en $\mathbb{R}^n$. Luego

$$\begin{aligned}
\lim_{k\to\infty} T^k z &= \lim_{k\to\infty} T^{k-1}Tz \\
&= \lim_{k\to\infty} T^{k-1}T(x^* - x^{(0)} \\
&= \lim_{k\to\infty} T^{k-1}(Tx^* - Tx^{(0)} \\
&= \lim_{k\to\infty} T^{k-1}(x^* \cancel{-c} - (x^{(1)} \cancel{-c}) \\
&= \lim_{k\to\infty} T^{k-1}(x^* - x^{(1)} \\
&= \lim_{k\to\infty} T^{k-2}(Tx^* - Tx^{(1)} \\
&= \lim_{k\to\infty} T^{k-2}(x^* - c - (x^{(2)} - c) \\
&= \lim_{k\to\infty} T^{k-1}(x^* - x^{(2)} \\
&\vdots \\
&= \lim_{k\to\infty} x^* - x^{(k)} = 0
\end{aligned}$$

Luego, $\lim_{k\to\infty} T^k z = 0$ para todo z, por lo que T es una matriz convergente.

Por lo tanto, podemos concluir que, efectivamente, la matriz T es una matriz convergente, es decir $\rho(T) < 1$, si y solo si la sucesión $\{x^{(k)}\}$ converge a la solución del sistema x^*, para todo $x^{(0)}$ inicial.

Este resultado nos brinda un criterio útil para determinar la convergencia de una iteración. Recordemos que Jacobi usaba $T_J = D^{-1}(L+U)$ mientras que Gauss - Seidel tomaba $T_{GS} = (D-L)^{-1}U$, de modo tal que, fijada A, basta determinar si $\rho(T) < 1$. Lo que nos falta ver es cómo podemos asegurar que el radio espectral es menor a 1, y de este modo afirmar su convergencia a la solución del sistema.

10.3.1. Matrices particulares

Hay familias de matrices para las cuales vamos a poder asegurar que el método de Jacobi o el método de Gauss-Seidel convergen.

Matrices EDD

Vamos a comenzar con las matrices diagonal dominante. Recordemos que una matriz A es *edd* si y solo su $|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$ para todo $i = 1, \ldots, n$.

Entonces, si A es *edd*, entonces la matriz de iteración de Jacobi $T_J = D^{-1}(L+U)$ es una matriz convergente, es decir $\rho(T_J) < 1$, y por tanto el método de Jacobi va a converger. Es un error común mezclar a la matriz de iteración T con la matriz original A al momento de analizar la convergencia del método iterativo. Necesitamos que $\rho(T) < 1$, pero estamos pidiendo que la matriz A sea *edd*. Veamos que esto es cierto.

Recordemos que $\rho(A) \leq \|A\|$ para toda norma inducida $\|\bullet\|$. Por lo tanto, para demostrar que $\rho(T_J) < 1$ basta con encontrar una norma inducida tal que $\|T\| < 1$. La norma que vamos a utilizar es la norma infinito:

$$T_J = D^{-1}(L+U)$$

$$\|T_J\|_\infty = \max_{i=1,\cdots,n} \|(T_J)_i^t\|_1$$

, siendo $(T_J)_i^t$ la fila i-ésima de T_J.

Pero la fila i-ésima de T_J tiene la siguiente pinta:

$$(T_J)_i^t = \frac{1}{a_{i,i}} \cdot [-a_{i,1}, \dots, -a_{i,i-1}, 0, -a_{i,i+1}, \cdots, -a_{i.n}]$$

Por lo tanto, si le estamos tomando la norma 1 a esta fila, obtenemos:

$$\|(T_J)_i^t\|_1 = \underbrace{\sum_{\substack{j=1 \\ j\neq i}}^{n} \left|\frac{a_{ij}}{a_{ii}}\right|}_{<1}$$

al ser A *edd*.

Por lo tanto, $\|T_J\|_\infty < 1$, y por tanto el radio espectral $\rho(T_J) < 1$.

Para el método de Gauss-Seidel tenemos un resultado similar. Si la matriz A es *edd*, entonces el método también converge. Para ello, nuevamente vamos a comprobar que $\rho(T_{GS} < 1$, con

$$T_{GS} = (D-L)^{-1} \cdot U$$

Sea λ autovalor de T_{GS}, y x el autovalor asociado tal que $\|x\|_\infty = 1$. Veamos que $|\lambda| < 1$:

$$\begin{gathered}
T_{GS}x = \lambda x \\
(D-L)^{-1} \cdot U = \lambda x \\
Ux = \lambda(D-L)x \\
\Longrightarrow \\
-\sum_{j=i+1}^{n} a_{ij}x_j = \lambda \sum_{j=1}^{i} a_{ij}x_j \qquad \text{(Mirando la Fila i)} \\
-\sum_{j=i+1}^{n} a_{ij}x_j = \lambda \sum_{j=1}^{i-1} a_{ij}x_j + \lambda a_{ii}x_i \\
\lambda a_{ii}x_i = -\sum_{j=i+1}^{n} a_{ij}x_j - \lambda \sum_{j=1}^{i-1} a_{ij}x_j \\
\Longrightarrow \\
|\lambda||a_{ii}||x_i| \leq \sum_{j=i+1}^{n} |a_{ij}||x_j| + |\lambda| \cdot \sum_{j=1}^{i-1} |a_{ij}||x_j|
\end{gathered}$$

Como $\|x\|_\infty = 1$, entonces existe $1 = |x_{i_0}| \geq |x_i|$ para todo i. Como la desigualdad a la que llegamos vale para todo i, en particular vale para i_0. Entonces

$$\begin{gathered}
|\lambda||a_{i_0 i_0}| \underbrace{|x_{i_0}|}_{=1} \leq \sum_{j=i_0+1}^{n} |a_{i_0 j}| \underbrace{|x_j|}_{\leq 1} + |\lambda| \cdot \sum_{j=1}^{i_0-1} |a_{i_0 j}| \underbrace{|x_j|}_{\leq 1} \\
\Longrightarrow \\
|\lambda||a_{i_0 i_0}| \leq \sum_{j=i_0+1}^{n} |a_{i_0 j}| + |\lambda| \sum_{j=1}^{i_0-1} |a_{i_0 j}| \\
\underbrace{|\lambda| \cdot \left(|a_{i_0 i_0}| - \sum_{j=1}^{i_0-1} |a_{i_0 j}|\right)}_{>0 \text{ por ser } A \text{ edd}} \leq \sum_{j=i_0+1}^{n} |a_{i_0 j}| \\
|\lambda| \leq \frac{\sum_{j=i_0+1}^{n} |a_{i_0 j}|}{|a_{i_0 i_0}| - \sum_{j=1}^{i_0-1} |a_{i_0 j}|}
\end{gathered}$$

Por otro lado,

$$
\begin{aligned}
|a_{i_0,i_0}| &> \sum_{j=1}^{i_0-1} |a_{i_0 j}| + \sum_{j=i_0+1}^{n} |a_{i_0 j}| \\
|a_{i_0,i_0}| - \sum_{j=1}^{i_0-1} |a_{i_0 j}| &> \sum_{j=i_0+1}^{n} |a_{i_0 j}| \\
1 &> \frac{\sum_{j=i_0+1}^{n} |a_{i_0 j}|}{|a_{i_0 i_0}| - \sum_{j=1}^{i_0-1} |a_{i_0 j}|}
\end{aligned}
$$

Entonces, $|\lambda| < \dfrac{\sum_{j=i_0+1}^{n} |a_{i_0 j}|}{|a_{i_0 i_0}| - \sum_{j=1}^{i_0-1} |a_{i_0 j}|} < 1$, para todo λ autovalor de T_{GS}.

Luego, $\rho(T_{GS}) < 1$, por lo que podemos concluir que si A es *edd*, entonces el método de Gauss-Seidel converge.

Matrices SDP

También se puede demostrar que si A es una matriz simétrica definida positiva, entonces el método de Gauss-Seidel converge.

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Métodos Numéricos modo virtual
(pandemia COVID-19)
Material Complementario

Métodos iterativos - Gauss Seidel en SDP - versión 1.0

Este es material complementario de las diapos de la clase de métodos iterativos usadas durante el dictado virtual (pandemia COVID-19).
Este documento incluye la demostración de convergencia del método de Gauss Seidel para el caso de matrices simétricas definidas positivas.

Como paso previo al análisis de convergencia de Gauss Seidel vamos a probar una propiedad que nos resultará útil

Lema: *Sea $A \in \mathbb{R}^{n\times n}$ una matriz simétrica definida positiva y $B \in \mathbb{R}^{n\times n}$ tal que $A - B - B^t$ es simétrica definida positiva. Entonces*

1. *$A - B$ es inversible*
2. *$\rho(-(A-B)^{-1}B) < 1$*

Demostración:

1. Supongamos que $A-B$ es singular. Entonces existe $v \in \mathbb{R}^n, v \neq 0$ tal que $(A-B)v = 0$. Si multiplicamos por v^t, obtenemos
$$v^t(A-B)v = 0$$
$$v^tAv = v^tBv$$
Como A y $A - B - B^t$ son simétricas definidas positivas y $v \neq 0$
$$v^t(A)v = v^tBv > 0$$
$$v^t(A-B)v - v^tB^tv = v^t(A-B-B^t)v > 0$$
Pero como $v^t(A-B)v = 0$, entonces $v^tB^tv < 0$ lo cual nos lleva a una contradicción. Entonces $A - B$ es inversible.

2. Sea λ autovalor de $-(A-B)^{-1}B$ y w el autovector asociado. Queremos ver que $|\lambda| < 1$. Por definición

$$-(A-B)^{-1}Bw = \lambda w$$

Multiplicando por $(A-B)$

$$-Bw = \lambda(A-B)w$$

Multiplicando por w^t

$$-w^tBw = \lambda w^t(A-B)w = \lambda w^tAw - \lambda w^tBw$$

Agrupando los términos

$$(\lambda-1)w^tBw = \lambda w^tAw$$

Como A es definida positiva entonces $w^tAw > 0$ y por lo tanto podemos afirmar que $\lambda \neq 1$. Entonces podemos dividir por $\lambda - 1$

$$w^tBw = \frac{\lambda}{(\lambda-1)}w^tAw$$

Por otro lado tenemos que $A - B - B^t$ es simétrica definida positiva, entonces

$$w^t(A-B-B^t)w > 0$$

Distribuyendo

$$w^tAw - w^tBw - w^tB^tw > 0$$

Usando que $w^tBw = w^tB^tw$

$$w^tAw - 2w^tBw > 0$$

Usando que $w^tBw = \frac{\lambda}{(\lambda-1)}w^tAw$

$$w^tAw - 2\frac{\lambda}{(\lambda-1)}w^tAw > 0$$

$$(1 - 2\frac{\lambda}{(\lambda-1)})w^tAw > 0$$

Como A es definida positiva, entonces $(1 - 2\frac{\lambda}{(\lambda-1)})$ debe ser > 0.

Esto es válido sii $2\frac{\lambda}{(\lambda-1)} < 1$.

Si $\lambda > 1 \Rightarrow 2\lambda < \lambda - 1 \Rightarrow \lambda < 1$ lo cual es una contradicción.

Si $\lambda \leq -1 \Rightarrow 2\lambda > \lambda - 1 \Rightarrow \lambda > -1$ lo cual es una contradicción.

Como ya teniamos que $\lambda \neq 1$, entonces podemos afirmar que $|\lambda| < 1$.

Concluimos entonces que $\rho(-(A-b)^{-1}B) < 1$

■

Proposición: *Sea $A \in \mathbb{R}^{n\times n}$ una matriz simétrica definida positiva. El método de Gauss Seidel converge independientemente del x^0 inicial.*

Demostración:
Sabemos que $A = D - L - U$. Como A es simétrica $\Rightarrow U = L^t$.
De la expresión de A

$$A = D - L - U$$

Usando que $U = L^t$ y pasando términos

$$A + L^t + L = D$$
$$A - (-L^t) - (-L) = D$$

Como A es definida positiva $\Rightarrow d_{ii} > 0$ y por lo tanto D es una matriz diagonal definida positiva. Si llamamos $B = -L^t$, entonces $A - B - B^t$ es simétrica definida positiva y podemos aplicar el lema anterior y afirmar que $\rho(-(A-B)^{-1}B) < 1$

¿Qué matriz es $-(A-B)^{-1}B$?

$$-(A-B)^{-1}B = -(A+L^t)^{-1}(-L^t) = (A+L^t)^{-1}L^t = (D-L)^{-1}U$$

que no es otra cosa que la matriz T_{GS}. Concluimos que $\rho(T_{GS}) < 1$ y el método de Gauss Seidel converge independientemente del x^0 inicial.

■

Hay otro resultado, más general, que nos habla de matrices tales que $a_{ij} \leq 0$ para todo $i \neq j$ y $a_{ii} > 0$, que nos dice que, para estas matrices, se satisface una sola de las siguientes propiedades:

- O bien $\rho(T_{GS} < \rho(T_J) < 1$. Es decir, ambos métodos convergen, pero como el radio espectral de la Gauss-Seidel es más chico este último converge más rápido.
- O bien $\rho(T_{GS} > \rho(T_J) > 1$. Es decir, ambos divergen.

Estas propiedades están sacadas de el paper de *P. Stein, R:L: Rosenberg, On the solution of lineal sumultaneous equations by iteration.*

10.4. Cota del Error

Por último, vamos a ver un resultado acerca de la cota del error para un esquema general $x = Tx + c$.

Sea $T \in \mathbb{R}^{n \times n}$ tal que $\|T\| < 1$ para una norma inducida, y x^* la solución del sistema. Entonces:

- $x^{(k+1)} = Tx^{(k)} + c$ converge independientemente del $x^{(0)}$ inicial, al ser $\rho(T) < \|(\|T) < 1$.
- $\|x^* - x^{(k)}\| \leq \|T\|^k \|X^{(0)} - x^*\|$, lo cual nos da una cota sobre el error con respecto a la solución del sistema en función del error inicial. Sin embargo, este resultado no resulta demasiado útil ya que tendríamos que conocer de antemano qué tan lejos está el vector inicial $x^{(0)}$ respecto de la solución del sistema.
- $\|x^* - x^{(k)}\| \leq \frac{\|T\|^k}{1-\|T\|} \cdot \|x^1 - x^{(0)}\|$. Ahora sí hemos obtenido una cota del error del paso k-ésimo, en función de las dos primeras iteradas, que sí podemos calcular.

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Métodos Numéricos modo virtual
(pandemia COVID-19)
Material Complementario

Métodos iterativos - Error - versión 1.0

Este es material complementario de las diapos de la clase de métodos iterativos usadas durante el dictado virtual (pandemia COVID-19).
Este documento incluye la demostración de una cota del error de un método iterativo para resolver sistemas de ecuaciones lineales.

Proposición: *Sean $T \in \mathbb{R}^{n\times n}$. Si $||T|| < 1$ para una norma inducida entonces*

1. *La sucesión $x^k = Tx^{k-1} + c$ converge a $x^* = (I - T)^{-1}c$ para cualquier x^0 inicial*
2. $||x^* - x^k|| \leq ||T||^k ||x^* - x^0||$
3. $||x^* - x^k|| \leq \frac{||T||^k}{1-||T||} ||x^1 - x^0||$

Demostración:

1. Hay una propiedad que establece que $|\rho(A)| \leq ||A||$ para toda norma inducida. Entonces $|\rho(T)| \leq ||T|| < 1$. Por lo tanto estamos en condiciones del teorema general de convergencia y podemos concluir que la sucesión converge para cualquier x^0.
2. Partimos del error y veamos como ir acotando. Sabemos que $x^* = Tx^* + c$ y que $x^k = Tx^{k-1} + c$.

$$||x^* - x^k|| = ||Tx^* + c - Tx^{k-1} - c|| = ||Tx^* - Tx^{k-1}|| = ||T(x^* - x^{k-1})|| \underset{\substack{\downarrow \\ \text{por norma inducida}}}{\leq} ||T||||x^* - x^{k-1}||$$

Volviendo a aplicar el mismo reeemplazo

$$||x^* - x^k|| \leq ||T||||Tx^* + c - Tx^{k-2} - c|| = ||T||||Tx^* - Tx^{k-2}|| = ||T||||T(x^* - x^{k-2})|| \underset{\substack{\downarrow \\ \text{por norma inducida}}}{\leq} ||T||^2||x^* - x^{k-2}||$$

Repitiendo el proceso, llegamos a:

$$||x^* - x^k|| \leq ||T||^k ||x^* - x^0||$$

3. Veamos primero la diferencia entre dos iteradas sucesivas

$$||x^{k+1} - x^k|| = ||Tx^k + c - Tx^{k-1} - c|| = ||T(x^k - x^{k-1})|| \leq ||T|| ||x^k - x^{k-1}||$$

Si seguimos reemplazado llegamos a que

$$||x^{k+1} - x^k|| \leq ||T||^k ||x^1 - x^0||$$

Consideremos ahora dos iteradas j y k con $j > k$:

$$||x^j - x^k|| = ||x^j - x^{j-1} + x^{j-1} - x^{j-2} + x^{j-2} - \ldots - x^{k+1} + x^{k+1} - x^k|| \leq ||x^j - x^{j-1}|| + ||x^{j-1} - x^{j-2}|| + \ldots + ||x^{k+1} - x^k||$$

Usando la cota anterior:

$$||x^j - x^k|| \leq (||T||^{j-1} + ||T||^{j-2} + \ldots + ||T||^k)||x^1 - x^0|| = ||T||^k (\sum_{i=0}^{j-1-k} ||T||^i)||x^1 - x^0||$$

Si ahora tomamos límite con $j \to \infty$, como $\{x^j\}_{j=0}^{\infty}$ converge a x^* y $\sum_{i=0}^{j-1-k} ||T||^i = \frac{1}{1-||T||}$ ya que $||T|| < 1$, obtenemos

$$||x^* - x^k|| \leq \frac{||T||^k}{(1 - ||T||)} ||x^1 - x^0||$$

■

Capítulo 11

Cuadrados Mínimos Lineales

Este capítulo está dedicado al problema de **Cuadrados Mínimos Lineales**. Vamos a comenzar por definir cuál es el problema, vamos a estudiar algunas propiedades teóricas, y finalmente vamos a proponer algoritmos para resolver el problema.

¿En qué consiste el problema? Vamos a tener un conjunto de pares ordenados de valores (x_i, y_i) para $i = 1, \ldots, n$, donde x es la variable independiente e y es la variable dependiente, para el cual buscamos una función $f(x)$, perteneciente a cierta familia $\mathcal{F}$, tal que "mejor aproxime" o mejor describa a los datos.

En esta definición del problema hay algunos términos ambiguos, por ejemplo ¿qué quiere decir que "mejor aproxime" a los datos? Entonces, podemos tener distintas propuestas

- La primer propuesta es considerar el error en los valores de la función con respecto a los valores de la variable dependiente, buscar el máximo error, y queremos determinar aquella función de la familia de funciones tal que minimice el *máximo error* (criterio *minimax*) entre cada uno de los puntos y el gráfico de la función, es decir, considerar como métrica

$$\min_{f \in \mathcal{F}} \left(\max_{i=1,\ldots,m} |f(x_i) - y_i| \right)$$

  La crítica que se le hace a esta expresión es que si la muestra llega a tener valores atípicos, estos valores podrían dominar a la muestra e inclinar la elección de la función, llevando a una peor aproximación para el caso general.

- Para evitar este problema, hay una segunda propuesta para expresar el concepto de "mejor aproxima" a los datos, que es considerar la suma de los errores en módulo, y buscar aquella función que minimice el *error absoluto* entre los puntos y el gráfico de la función:

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} |f(x_i) - y_i|$$

  De esta manera, los valores atípicos no dominan la muestra, obteniendo así una función tal que, dentro de la familia de funciones, describa mejor los datos. La crítica que se le suele hacer a este criterio es que está tomando en cuenta una función que no es derivable.

- Luego, se propone buscar aquella función, dentro de la familia de funciones, que minimice la suma del *error cuadrático*

$$\boxed{\min_{f \in \mathcal{F}} \sum_{i=1}^{m} (f(x_i) - y_i)^2} \quad \leftarrow \text{Método de Cuadrados Mínimos}$$

  Este criterio es el más usado en el contexto de aproximación ya que, bajo ciertos escenarios, tiene propiedades teóricas y prácticas que facilitan obtener la solución.

Nos vamos a centrar en resolver el problema de Cuadrados Mínimos para familias de funciones donde los parámetros a determinar estén relacionados de forma lineal. Por eso es que hablamos del problema de Cuadrados Mínimos Lineales.

Este problema consiste en, dado un conjunto de funciones $\{\phi_1, \ldots, \phi_n\}$ linealmente independientes, vamos a definir a la familia de funciones como la combinación lineal de esas funciones:

$$\mathcal{F} = \{f(x) = \sum_{j=1}^{n} c_j \phi_i\}$$

donde $\varphi_1, \ldots, \varphi_n$ son funciones reales fijas. A modo de ejemplo, podemos considerar la familia de funciones lineales (en cuyo caso tendremos dos parámetros), de funciones cuadráticas (donde habrá tres parámetros), etc.

Por lo tanto, el problema de CML va a consistir en hallar estos coeficientes $c_1, \ldots, c_n$ tal que:

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} (f(x_i) - y_i)^2 = \min_{c_1, \ldots, c_n} \sum_{i=1}^{m} \Big(\sum_{j=1}^{n} (c_j \phi_j(x_i)) - y_i \Big)^2$$

Veamos ahora una forma de expresar el problema de CML de forma matricial. Para ello, vamos a considerar una matriz $A \in \mathbb{R}^{m \times n}$, donde colocamos los valores de las funciones ϕ_i evaluadas en las variables independientes $x_1, \ldots, x_m$ de la muestra, un vector $b \in \mathbb{R}^m$ con los valores de la variable dependiente de la muestra, y un vector x correspondiente a los coeficientes a determinar:

$$A = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \ldots & \phi_n(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \ldots & \phi_n(x_2) \\ \vdots & \vdots & \ldots & \vdots \\ \phi_1(x_m) & \phi_2(x_m) & \ldots & \phi_n(x_m) \end{bmatrix}, \quad b = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}, \quad x = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$$

Nota: Acá se le llama a x tanto a los coeficientes a determinar, como a los datos del par ordenado (x_i, y_i), pero no tienen relación alguna.

Luego, con estas definiciones de A, b, x, decimos que de CML se formula como

$$\boxed{\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2}$$

En el contexto de la estadística, el problema de CML es conocido como *regresión lineal*. Veamos una motivación, desde el punto de vista estadístico, para CML. Asumamos que los datos a_{ii} son conocidos de forma exacta, de manera tal que solo b tenga ruido, y que el ruido presente en cada b_i es independiente y tiene distribución normal con media $= 0$ y un desvío estándar σ. Sea x la solución de CML y x_T el verdadero valor de los parámetros. Entonces, x es conocido como el estimador de *máxima verosimilitud* de x_T, y el error $x - x_T$ tiene distribución normal, con media $= 0$ en cada componente, y la matriz de covarianza es $\sigma^2(A^t A)^{-1}$. Además, x resulta ser un estimador insesgado y de varianza mínima. Para más detalles respecto a la conexión con la estadística, ver *A. Bjorck, Numerical Methods for Least Squares Problems. SIAM, Philadelphia, PA, 1996 (Pág. 259).*

11.1. Solución de CML

Una vez planteado el problema, veamos si este tiene solución. Para ver esto, vamos a hacer uso de un resultado del álgebra lineal que nos dice que

$$\mathrm{Im}(A) \oplus \mathrm{Nu}(A^t) = \mathbb{R}^n$$

$$\mathrm{Nu}(A^t) = \mathrm{Im}(A)^{\perp}$$

con este resultado del álgebra lineal, podemos afirmar que, como $b \in \mathbb{R}^m$, entonces podemos escribir a b como

$$b^{(1)} + b^{(2)} = b$$

donde $b^{(1)} \in \mathrm{Im}(A)$ es la proyección ortogonal de b sobre $\mathrm{Im}(A)$, y $b^{(2)} \in \mathrm{Nu}(A^t)$ es la proyección ortogonal de b sobre $\mathrm{Nu}(A^t)$.

Entonces, el problema de cuadrados mínimos consiste en hallar el x tal que minimice

$$\begin{aligned}
\min_{x\in\mathbb{R}^m} \|Ax-b\|_2^2 &= \min_{y\in\mathrm{Im}(A)} \|y-b\|_2^2 \\
&= \min_{y\in\mathrm{Im}(A)} \|y-(b^{(1)}+b^{(2)}\|_2^2 \\
&= \min_{y\in\mathrm{Im}(A)} \|(y-b^{(1)})-b^{(2)}\|_2^2 \\
&= \min_{y\in\mathrm{Im}(A)} \|(y-b^{(1)})\|_2^2 + \|b^{(2)}\|_2^2 - \underbrace{2(y-b^{(1)})^t\cdot b^{(2)}}_{=0,\text{ pues } y-b^{(1)}\perp b^{(2)}} \\
&= \min_{y\in\mathrm{Im}(A)} \|(y-b^{(1)})\|_2^2 + \|b^{(2)}\|_2^2 \\
&= \min_{y\in\mathrm{Im}(A)} \|(y-b^{(1)})\|_2^2 \quad (b \text{ es una constante})
\end{aligned}$$

Como $b^{(1)}, y \in \mathrm{Im}(A)$, entonces si tomamos $\boxed{y = b^{(1)}}$ se alcanza el mínimo, y por lo tanto $x^* \in \mathbb{R}^n$ es solución de cuadrados mínimos lineales si:

$$Ax^* = b^{(1)}$$

Como $b^{(1)} \in \mathrm{Im}(A)$, esto quiere decir que el problema de cuadrados mínimos siempre tiene solución. Asegurada la existencia de la solución, la próxima pregunta a hacer es si la solución es única.

Notemos que si el sistema $\mathbf{A}\cdot\mathbf{x} = \mathbf{b}$ tiene alguna solución $\mathbf{x}^*$, entonces $\mathbf{b} \in \mathrm{Im}(\mathbf{A})$, por lo que $\mathbf{b^1} = \mathbf{b}$, y por lo tanto $\mathbf{x}^*$ también será solución del problema de cuadrados mínimos lineales.

Sin embargo, lo interesante de este problema es que tiene solución *incluso* cuando el sistema $\mathbf{A}\cdot\mathbf{x} = \mathbf{b}$ no la tiene, es decir, cuando está *sobre-determinado* (tiene ecuaciones que "se contradicen"). En estos casos, obviamente, el resultado hallado no será una solución de $\mathbf{A}\cdot\mathbf{x} = \mathbf{b}$, pero sí será la mejor aproximación posible según el criterio de minimizar el error cuadrático.

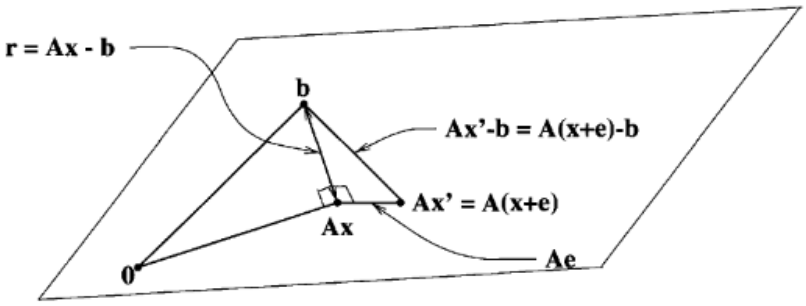
Unicidad

Como la solución de cuadrados mínimos está caracterizada por $Ax^* = b^{(1)}$, y Ax no es otra cosa que una combinación lineal de las columnas de A, podemos afirmar que la solución es única si y solo si hay una única forma de escribir a $\mathbf{b_1}$ como combinación lineal de las columnas de $\mathbf{A}$. Esto equivale a pedir que las columnas de $\mathbf{A}$ sean linealmente independientes, o que $\mathbf{A}$ sea de rango columna completo ($\mathrm{rg}(\mathbf{A}) = n$).

11.1.1. Interpretación geométrica

En primer lugar, observemos que si el sistema $Ax = b$ tiene solución, entonces cualquiera de ellas realiza el mínimo, que es 0.

Si $Ax = b$ no tiene solución, entonces es evidente que el mínimo es mayor que 0. Para entender cómo elegir un vector x que lo realice, pensemos en Ax y b como vectores en $\mathbb{R}^m$. El mínimo se alcanza cuando la distancia euclídea entre estos dos vectores es mínima. Pero el único de estos dos vectores que se mueve es Ax, con lo cual hay que elegirlo de modo tal que esté lo más cerca posible de b. Recordemos que el conjunto de valores que puede tomar Ax es el subespacio $\mathrm{Im}(A) = \{Ax : x \in \mathbb{R}^n\}$. Luego, queremos encontrar la distancia del punto b al subespacio $\mathrm{Im}(A)$, y es sabido que el punto sobre el subespacio que realiza la distancia es la proyección ortogonal de b sobre $\mathrm{Im}(A)$.



11.2. Formas Explícitas para la solución de CML

Si bien tenemos caracterizadas las soluciones del problema de CML vía $Ax = b^{(1)}$, esta no está escrita en función de los datos originales, sino que en función de $b^{(1)}$, el cual no conocemos. Además, aún si conociéramos el valor de $b^{(1)}$ contamos con el problema de que la matriz $A \in \mathbb{R}^{m\times n}$ no es cuadrada.

El problema de CML tiene varias soluciones explícitas que vamos a estudiar:

1. Ecuaciones Normales,
2. Descomposición QR,
3. Descomposición en valores singulares

El primer método es el más rápido pero el menos preciso, ideal cuando el número de condición es pequeño. El segundo método es el estándar y puede llegar a costar el doble que el primero. El tercero es uno de los más usados para resolver sistemas mal condicionados, cuando A no tiene rango completo, pero es varias veces más costoso. El primer y segundo método tienen la ventaja de que pueden ser adaptados para trabajar de forma eficiente con matrices ralas.

11.2.1. Ecuaciones Normales

Este método consiste en resolver el siguiente sistema de ecuaciones, que recibe el nombre de **ecuaciones normales**:

$$\mathbf{A}^{\mathrm{T}} \cdot \mathbf{A} \cdot \mathbf{x} = \mathbf{A}^{\mathrm{T}} \cdot \mathbf{b}.$$

La ventaja de este sistema es que está expresado únicamente en función de datos originales, y además es un sistema cuadrado, donde $A^tA \in \mathbb{R}^{n\times n}$.

Para verificar que estas ecuaciones resuelven el problema, veamos primero que si $Ax = b^{(1)}$, entonces x es solución de las ecuaciones normales, luego probaremos la vuelta. Si x es solución de CML, entonces

$$\begin{aligned}
Ax &= b^{(1)} \\
b - Ax &= b - b^{(1)} \\
&= b^{(2)} \\
&\Longrightarrow \\
A^t(b - Ax) &= A^t b^{(2)} \\
&= 0 \quad \text{pues } b^{(2)} \in \mathrm{Nu}(A^t) \\
&\Longrightarrow \\
A^t b - A^t Ax &= 0 \\
&\boxed{A^t Ax = A^t b}
\end{aligned}$$

Esto quiere decir que cualquier solución de cuadrados mínimos también será solución de las ecuaciones normales.

Para ver la vuelta, es decir que cualquier solución de las ecuaciones normales es una solución de cuadrados mínimos, consideremos $\mathbf{x}$ tal que $\mathbf{A}^{\mathrm{T}} \cdot \mathbf{A} \cdot \mathbf{x} = \mathbf{A}^{\mathrm{T}} \cdot \mathbf{b_1}$; en tal caso $\mathbf{A}^{\mathrm{T}} \cdot (\mathbf{A} \cdot \mathbf{x} - \mathbf{b_1}) = 0$, por lo que

$$\mathbf{A} \cdot \mathbf{x} - \mathbf{b_1} \in \mathrm{Nu}(\mathbf{A}^{\mathrm{T}}) = \mathrm{Im}(\mathbf{A})^{\perp}.$$

Por otra parte, como $\mathbf{A} \cdot \mathbf{x} \in \mathrm{Im}(\mathbf{A})$ y $\mathbf{b_1} \in \mathrm{Im}(\mathbf{A})$, necesariamente

$$\mathbf{A} \cdot \mathbf{x} - \mathbf{b_1} \in \mathrm{Im}(\mathbf{A}).$$

El único vector que está simultáneamente en $\mathrm{Im}(\mathbf{A})$ y en $\mathrm{Im}(\mathbf{A})^{\perp}$ es 0. Luego $\mathbf{A} \cdot \mathbf{x} - \mathbf{b_1} = 0$, por lo que $\mathbf{A} \cdot \mathbf{x} = \mathbf{b_1}$ y entonces $\mathbf{x}$ es una solución de cuadrados mínimos.

En el caso de que la matriz $A\mathbb{R}^{m \times n}$ tenga columnas linealmente independientes, es decir $rango(A) = n$, entonces el $rango(A^t A) = rango(A) = n$, entonces $A^t A \in \mathbb{R}^{n \times n}$ es inversible, y la solución de CML es $x^* = (A^t A)^{-1} A^t b$.

El sistema de las ecuaciones normales $\mathbf{A}^{\mathrm{T}} \cdot \mathbf{A}$ tiene las buenas propiedades de ser simétrica, semidefinida positiva y, en caso de que las columnas de A sean linealmente independientes, definida positiva. Luego, si A tiene rango completo, se puede utilizar la factorización de Cholesky para resolver las ecuaciones normales, en $\approx n^2 m + \frac{1}{3} n^3$ operaciones de punto flotante. En general, $m \gg n$, por lo que el costo $n^2 m$ de calcular $A^t A$ es el costo dominante.

Generalización del Número de Condición

Cuando trabajamos con sistemas de ecuaciones, una de los temas a analizar es la estabilidad numérica. En particular, nos interesa estudiar cómo varía la solución del sistema cuando se realizan modificaciones sobre el término independiente. Para las matrices cuadradas e inversibles teníamos el concepto del número de condición, el cual si es grande, entonces pequeños cambios sobre el término independiente pueden generar grandes cambios sobre el vector solución, lo cual no es deseable.

Para el problema de CML tenemos un resultado similar. Estamos bajo el caso de una matriz $A \in \mathbb{R}^{m \times n}$ con rango completo, es decir $rango(A) = n$. Entonces, sabemos que la solución de CML es solución de las ecuaciones normales

$$A^t A x = A^t b$$

y como estamos trabajando con una matriz A con $rango(A) = n$, $A^t A$ es inversible. Luego, la única solución de CML es $x = (A^t A)^{-1} A^t b$.

Para analizar la estabilidad numérica, consideremos que en lugar de b tenemos $\bar{b}$, luego nos queda el sistema $A^t A x = A^t \bar{b}$, y la solución de CML es $\bar{x} = (A^t A)^{-1} A^t \bar{b}$. Entonces, la propiedad nos dice que tenemos una cota del error

$$\frac{\|x - \bar{x}\|_2}{\|x\|_2} \leq \mathcal{X}(A) \cdot \frac{\|b^{(1)} - \bar{b}^{(1)}\|_2}{\|b^{(1)}\|_2}$$

donde

- $\mathcal{X}(A) = \|A\|_2 \cdot \|A^+\|_2$ es la generalización del número de condición, donde $A^+ = (A^t A)^{-1} A^t$ se conoce como la pseudo-inversa (Moore-Penrose) de A, con $m \geq n$. Si $m < n$, entonces $A^+ = A^t (A A^t)^{-1}$. Notemos que en el caso de que A sea inversible, $\mathcal{X}(A) = \kappa(A)$.
- $b = b^{(1)} + b^{(2)}$.
- $\bar{b} = \bar{b}^{(1)} + \bar{b}^{(2)}$.

Además, se puede probar que $\mathcal{X}_2(A)^2 = \mathcal{X}_2(A^t A)$ con X_2 el número de condición inducido por la norma 2.

Métodos Numéricos modo virtual
(pandemia COVID-19)
Material Complementario

CML - Error - versión 1.0

Este es material complementario de las diapos de la clase de cuadrados mínimos usadas durante el dictado virtual (pandemia COVID-19). En este documento hacemos un análisis del error.

Vamos a analizar la sensibilidad de la solución cuando variamos el término independiente. Queremos determinar la relación entre pequeños cambios en el vector b con los cambios en la solución. La idea es muy similar a la que vimos para sistemas lineales donde el número de condición de la matriz nos permitía establecer esta relación. En este caso vamos a tener una generalización del número de condición.

Proposición: *Sea* $A \in \mathbb{R}^{m\times n}$ *con* $rango(A) = n$. *Sean* $b, \bar{b} \in \mathbb{R}^m$ *y* $b = b^1 + b^2$, $\bar{b} = \bar{b}^1 + \bar{b}^2$ *con* $b^1, \bar{b}^1 \in Im(A)$ *y* $b^2, \bar{b}^2 \in Nu(A^t)$. *Si* $b^1 \neq 0$ *entonces*

$$\frac{||x^* - \bar{x}^*||_2}{||x^*||_2} = \frac{||(A^tA)^{-1}A^tb - (A^tA)^{-1}A^t\bar{b}||_2}{||(A^tA)^{-1}A^tb||_2} \leq \chi(A)\frac{||b^1 - \bar{b}^1||_2}{||b^1||_2}$$

donde $\chi(A) = ||A||_2||(A^tA)^{-1}A^t||_2$

Demostración:
Como el $rango(A) = n$, la solución del problema de cuadrados mínimos lineales es única y basado en las propiedades que vimos, sabemos que verifica:

$$x^* = (A^tA)^{-1}A^tb \qquad Ax^* = b^1$$
$$\bar{x}^* = (A^tA)^{-1}A^t\bar{b} \qquad A\bar{x}^* = \bar{b}^1$$

$$||(A^tA)^{-1}A^tb-(A^tA)^{-1}A^t\bar{b}||_2 = ||(A^tA)^{-1}A^t(b^1+b^2)-(A^tA)^{-1}A^t(\bar{b}^1+\bar{b}^2)||_2 \underset{\substack{\downarrow \\ b^2, \bar{b}^2 \in Nu(A^t)}}{=} ||(A^tA)^{-1}A^tb^1-(A^tA)^{-1}A^t\bar{b}^1||_2$$

$$||(A^tA)^{-1}A^tb-(A^tA)^{-1}A^t\bar{b}||_2 = ||(A^tA)^{-1}A^tb^1-(A^tA)^{-1}A^t\bar{b}^1||_2 = ||(A^tA)^{-1}A^t(b^1-\bar{b}^1)||_2 \underset{\substack{\downarrow \\ \text{por ser norma inducida}}}{\leq} ||(A^tA)^{-1}A^t||_2||b^1-\bar{b}^1||_2$$

Por otro lado, $Ax^* = b^1$, entonces $||b^1||_2 = ||Ax^*||_2 \underset{\substack{\downarrow \\ \text{por ser norma inducida}}}{\leq} ||A||_2||x^*||_2$

$$\frac{1}{||x^*||_2} \leq \frac{||A||_2}{||b^1||_2}$$

En conclusión tenemos las dos siguientes desigualdades:

$$||(A^tA)^{-1}A^tb - (A^tA)^{-1}A^t\bar{b}||_2 \leq ||(A^tA)^{-1}A^t||_2||b^1 - \bar{b}^1||_2$$

$$\frac{1}{||x^*||_2} \leq \frac{||A||_2}{||b^1||_2}$$

Multiplicando los términos (son todos positivos) del mismo lado de las desigualdades obtenemos:

$$\frac{||(A^tA)^{-1}A^tb - (A^tA)^{-1}A^t\bar{b}||_2}{||x^*||_2} \leq \frac{||A||_2||(A^tA)^{-1}A^t||_2||b^1 - \bar{b}^1||_2}{||b^1||_2}$$

■

Hasta aquí tenemos todos los resultados teóricos respecto a la existencia de solución, unicidad, una cierta caracterización de la solución, y una metodología para resolverlo. Sin embargo, las ecuaciones normales no siempre son numéricamente estables. Si el número de condición $\mathcal{X}(A)$, que de por sí puede no ser bueno, al computar $\mathbf{A}^{\mathrm{T}} \cdot \mathbf{A}$ este se eleva al cuadrado. Esto motiva la utilización de otros métodos más estables, que aprovechan algunas de las factorizaciones matriciales estudiadas anteriormente.

11.2.2. Factorización QR

Un método para resolver el problema de cuadrados mínimos es utilizar la factorización QR. Recordemos que toda matriz admite una factorización QR, es decir siempre podemos escribir a A como $A = QR$, con Q ortogonal y R triangular superior. Habíamos visto dos manera de obtener la factorización QR, una basada en reflexiones, y la otra basada en rotaciones. Vamos a recordar el método de reflexiones.

La idea era considerar a una matriz $A \in \mathbb{R}^{n\times n}$ y, para la primera columna de A, nos construíamos una reflexión $\in \mathbb{R}^{n\times n}$ tal que

$$Q_1 a_1 = \|a_1\|_2 e_1$$

Una vez teníamos esta matriz $Q_1 A$, considerábamos la siguiente columna de A, y nos construíamos una reflexión tal que la segunda columna de $(Q_2 \cdot Q_1 A)$ tenga elementos nulos a partir de la tercer posición en adelante. Si continuábamos con este procedimiento, obteníamos

$$Q_{n-1} \dots Q_2 \underbrace{Q_1 A}_{n\times n} = \underbrace{R}_{n\times n}$$

Si ahora trabajamos con una matriz $A \in \mathbb{R}^{m\times n}$, si consideramos la primer columna de esta matriz, siempre vamos a poder construirnos una reflexión $Q_1 \in \mathbb{R}^{m\times m}$ tal que

$$Q_1 a_1 = \|a_1\|_2 e_1$$

Si ahora consideramos la segunda columna de $Q_1 A$ y nos construimos una reflexión tal que la segunda columna de $(Q_2 \cdot Q_1 A)$ tenga elementos nulos a partir de la tercer posición en adelante. Si continuamos con este procedimiento, obtenemos

$$Q_{n-1} \dots Q_2 \underbrace{Q_1 A}_{m\times n} = \underbrace{R}_{m\times n}$$

Por lo tanto, este proceso queda bien definido, independientemente de si la matriz A es o no cuadrada.

Ahora veamos qué estructura puede llegar a tener $R \in \mathbb{R}^{m\times n}$ asociada a la factorización QR de la matriz $A \in \mathbb{R}^{m\times n}$. Para ello, vamos a analizar el análisis en dos casos: $rg(A) = n$, y $rg(A) < n$.

En el caso de que $rg(A) = n$, es decir que las columnas de A son linealmente independientes, la matriz R también debe tener rango completo, por lo que R tiene que tener la siguiente estructura

$$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

con $R_1 \in \mathbb{R}^{n\times n}$ triangular superior

En el caso de que $rg(A) < n$, entonces puede ocurrir que, durante el proceso de aplicar reflexiones, lleguemos a tener todos los elementos por debajo del elemento de la diagonal nulos, es decir

$$\begin{bmatrix} * & * & \cdots & \cdots & * \\ & \ddots & \cdots & \cdots & * \\ & & 0 & \vdots & * \\ & & \vdots & \ddots & \vdots \\ & & 0 & \vdots & * \end{bmatrix}$$

Cuando hacíamos la factorización QR, nuestro objetivo era colocar ceros por debajo de la diagonal, y como este objetivo estaba cumplido, podíamos continuar con el siguiente paso. Sin embargo, en el

contexto de CML, vamos a buscar entre las columnas $i+1$ en adelante si hay alguna que tenga elementos no nulos desde la posición i hacia abajo. Si esa columna existe, la permutamos con la columna i, y ahora hacemos la reflexión respectiva.

Entonces, lo que nos va a estar pasando es que, mediante permutaciones de las columnas, vamos a ir teniendo elementos no nulos en la diagonal, de manera que nos quedan todas las columnas incompletas de la matriz R se encuentren todas al final. En general, se permutan las columnas de $\mathbf{R}$ de forma tal que

$$R = \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix}$$

con $R_1 \in \mathbb{R}^{r\times r}$ triangular superior, $R_2 \in \mathbb{R}^{n-r\times n-r}$, con $r = rg(A)$.

Luego, nos queda la siguiente igualdad

$$\mathbf{A}\cdot\mathbf{P} = \mathbf{Q}\cdot\mathbf{R}$$

con P la matriz que permuta a las columnas de A.

Ahora que conocemos la estructura de la factorización QR para matrices de $m\times n$, veamos cómo nos queda la solución de CML. Si escribimos $\mathbf{A} = \mathbf{Q}\cdot\mathbf{R}$, la expresión que buscamos minimizar se puede reescribir como

$$\|\mathbf{A}\cdot\mathbf{x}-\mathbf{b}\|_2^2 = \|\mathbf{Q}\cdot\mathbf{R}\cdot\mathbf{x}-\mathbf{b}\|_2^2 .$$

Pero como $\mathbf{Q}$ es una matriz ortogonal, multiplicar por $\mathbf{Q}^{\mathrm{T}}$ no altera la norma 2, por lo que resulta que

$$\|\mathbf{A}\cdot\mathbf{x}-\mathbf{b}\|_2^2 = \left\|\mathbf{Q}^{\mathrm{T}}\cdot(\mathbf{Q}\cdot\mathbf{R}\cdot\mathbf{x}-\mathbf{b})\right\|_2^2 . = \left\|\mathbf{R}\cdot\mathbf{x}-\mathbf{Q}^{\mathrm{T}}\cdot\mathbf{b}\right\|_2^2 .$$

En definitiva, nuestro problema se convierte en hallar $\mathbf{x}$ que realice el mínimo

$$\min_{\mathbf{x}} \left\|\mathbf{R}\cdot\mathbf{x}-\mathbf{Q}^{\mathrm{T}}\cdot\mathbf{b}\right\|_2^2 .$$

Para hallar este mínimo hay que proceder de una forma ligeramente distinta en base a dos casos, que dependen de $k = \mathrm{rg}(\mathbf{A})$. En ambos casos es necesario tener en cuenta que, como $\mathbf{Q}$ es inversible, entonces $\mathrm{rg}(\mathbf{R}) = \mathrm{rg}(\mathbf{A}) = k$.

(I) Si $\mathbf{A}$ es de rango columna completo, podemos escribir:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R_1} \\ \mathbf{0} \end{bmatrix} \qquad \mathbf{Q}^{\mathrm{T}}\cdot\mathbf{b} = \begin{pmatrix} c \\ d \end{pmatrix}$$

donde $\mathbf{R_1} \in \mathbb{R}^{n\times n}$ es triangular superior, $\mathbf{c} \in \mathbb{R}^n$ y $\mathbf{d} \in \mathbb{R}^{m-n}$.

Entonces, resulta que:

$$\begin{aligned}
\min_{\mathbf{x}} \left|\left|\mathbf{R}\cdot\mathbf{x}-\mathbf{Q}^{\mathrm{T}}\cdot\mathbf{b}\right|\right|_2^2 &= \min_{\mathbf{x}} \left|\left|\begin{pmatrix} R_1\cdot x \\ 0\cdot x \end{pmatrix} - \begin{pmatrix} c \\ d \end{pmatrix}\right|\right|_2^2 \\
&= \min_{\mathbf{x}} \left|\left|\begin{pmatrix} R_1\cdot x - c \\ -d \end{pmatrix}\right|\right|_2^2 \\
&= \min_{\mathbf{x}} \|\mathbf{R_1}\cdot\mathbf{x}-\mathbf{c}\|_2^2 + \|\mathbf{d}\|_2^2
\end{aligned}$$

Basta con minimizar el primer término de la expresión, ya que es el único que depende de $\mathbf{x}$. En efecto, el mínimo se alcanza si dicho término se anula, es decir, si $\mathbf{x}$ es solución del sistema

$$\mathbf{R_1}\cdot\mathbf{x} = \mathbf{c},$$

que siempre tiene solución existe y es única porque R_1 es inversible, al ser cuadrada y de rango completo, y la solución nos queda

$$\boxed{\mathbf{x} = \mathbf{R_1^{-1}}\mathbf{c}}$$

(II) Si $\mathbf{A}$ no es de rango columna completo, es decir, $k < n$, necesitaremos que la factorización QR haya sido obtenida con *pivoteo de columnas*; esto quiere decir, que haya sido construida de forma tal que las columnas incompletas de $\mathbf{R}$ se encuentren todas al final. En general, se permutan las columnas de $\mathbf{R}$ de forma tal que $|r_{1,1}| \geq |r_{2,2}| \geq \cdots \geq |r_{n,n}|$. Así,

$$\mathbf{A} = \mathbf{Q} \cdot \mathbf{R} \cdot \mathbf{P},$$

donde $\mathbf{P}$ es la matriz de permutación correspondiente al pivoteo. Por lo tanto, el mínimo buscado será

$$\min_{\mathbf{x}} \left\|\mathbf{R} \cdot \mathbf{P} \cdot \mathbf{x} - \mathbf{Q}^{\mathrm{T}} \cdot \mathbf{b}\right\|_2^2 = \min_{\mathbf{x}} \left\|\mathbf{R} \cdot \tilde{\mathbf{x}} - \mathbf{Q}^{\mathrm{T}} \cdot \mathbf{b}\right\|_2^2$$

donde $\tilde{\mathbf{x}} = \mathbf{P} \cdot \mathbf{x}$. Resolveremos el sistema para $\tilde{\mathbf{x}}$; terminado el proceso, deberemos tener en cuenta que las soluciones halladas tendrán permutadas sus componentes.

Podemos escribir, entonces:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R_1} & \mathbf{R_2} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad \mathbf{Q^t b} = \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix} \qquad \tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{pmatrix}$$

donde $\mathbf{R_1} \in \mathbb{R}^{k \times k}$ es triangular superior, $\mathbf{R_2} \in \mathbb{R}^{k \times n-k}$, $\mathbf{c}, \mathbf{x_1} \in \mathbb{R}^k$ y $\mathbf{d}, \mathbf{x_2} \in \mathbb{R}^{n-k}$.

De lo anterior,

$$\begin{aligned} \min_{\mathbf{x}} \left|\left|\mathbf{R} \cdot \mathbf{x} - \mathbf{Q}^{\mathrm{T}} \cdot \mathbf{b}\right|\right|_2^2 &= \min_{\mathbf{x}} \left|\left| \begin{pmatrix} \mathbf{R_1} \cdot \mathbf{x_1} + \mathbf{R_2} \cdot \mathbf{x_2} - \mathbf{c} \\ -\mathbf{d} \end{pmatrix} \right|\right|_2^2 \\ &= \min_{\mathbf{x}} \|\mathbf{R_1} \cdot \mathbf{x_1} + \mathbf{R_2} \cdot \mathbf{x_2} - \mathbf{c}\|_2^2 + \|\mathbf{d}\|_2^2 \end{aligned}$$

De nuevo, esta expresión alcanza el mínimo cuando $R_1 x_1 + R_2 x_2 = c$, el cual es un sistema de r ecuaciones con n incógnitas, por lo que tenemos infinitas soluciones. Luego, para obtener un sistema de r ecuaciones y r incógnitas podemos fijar al $\mathbf{x_2} \in \mathbb{R}^{n-r}$ según nos convenga en un $\mathbf{x_2^*}$, para luego determinar $\mathbf{x_1}$ como la única solución del sistema

$$\mathbf{R_1} \cdot \mathbf{x_1} = \mathbf{c} - \mathbf{R_2} \cdot \mathbf{x_2^*}.$$

Como ya se mencionó cuando se habló de la factorización QR como método para resolver sistemas de ecuaciones lineales, se trata de un método numéricamente muy estable, debido a que la matriz $\mathbf{Q}$ es ortogonal. En particular, podemos afirmar que si el problema de cuadrados mínimos lineales

$$\min_{x} \|b - Ax\|_2$$

es resultado utilizando la factorización QR, entonces la solución computada $\hat{x}$ es la solución exacta para

$$\min_{x} \|(b + \Delta_b) - (A + \Delta_A) \cdot \hat{x}\|_2$$

Además, el costo para la resolución de CML vía factorización QR es $\approx 2n^2m - \frac{2}{3}n^3$, cuando A tiene rango completo, es alrededor del doble de costo que la resolución vía ecuaciones normales para $m \gg n$, y alrededor del mismo costo si $m = n$.

Sin embargo, al utilizarlo con matrices de rango incompleto, se vuelve necesario incorporar el pivoteo, lo cual aumenta la complejidad del algoritmo y lo vuelve menos estable. Además, hallar el rango de $\mathbf{A}$ puede resultar difícil debido a errores de redondeo.

11.2.3. Descomposición en Valores Singulares

Otra posibilidad para resolver el problema de cuadrados mínimos es utilizar la descomposición en valores singulares de

$$\underbrace{\mathbf{A}}_{m \times n} = \underbrace{\mathbf{U}}_{m \times m} \cdot \underbrace{\mathbf{\Sigma}}_{m \times n} \cdot \underbrace{\mathbf{V}^{\mathrm{T}}}_{n \times n}$$

En este caso, como $\mathbf{U}$ es ortogonal, multiplicar por $\mathbf{U}^{\mathrm{T}}$ no modifica la norma 2, y tenemos que:

$$\begin{aligned}\|\mathbf{A}\cdot\mathbf{x}-\mathbf{b}\|_2^2 &= \left\|\mathbf{U}\cdot\mathbf{\Sigma}\cdot\mathbf{V}^{\mathrm{T}}\cdot\mathbf{x}-\mathbf{b}\right\|_2^2\\ &= \left\|\mathbf{\Sigma}\cdot\mathbf{V}^{\mathrm{T}}\cdot\mathbf{x}-\mathbf{U}^{\mathrm{T}}\cdot\mathbf{b}\right\|_2^2.\end{aligned}$$

Si llamamos $\mathbf{V}^{\mathrm{T}}\cdot\mathbf{x}=\mathbf{y}$, como $\mathbf{V}^{\mathrm{T}}$ es inversible, tenemos un sistema de ecuaciones determinado: si encontramos un valor que nos sirva para $\mathbf{y}$, podemos determinar cuál debe ser el valor de $\mathbf{x}$. Entonces, sustituyendo, obtenemos que

$$\|\mathbf{A}\cdot\mathbf{x}-\mathbf{b}\|_2^2 = \left\|\mathbf{\Sigma}\cdot\mathbf{y}-\mathbf{U}^{\mathrm{T}}\cdot\mathbf{b}\right\|_2^2,$$

y, por lo tanto, el problema de minimización a resolver es

$$\min_{\mathbf{y}} \left\|\mathbf{\Sigma}\cdot\mathbf{y}-\mathbf{U}^{\mathrm{T}}\cdot\mathbf{b}\right\|_2^2.$$

Volvemos a separar en dos casos según $k=\mathrm{rg}(\mathbf{A})=\mathrm{rg}(\mathbf{\Sigma})$.

(I) Si $\mathbf{A}$ es de rango columna completo, podemos escribir:

$$\mathbf{\Sigma} = \left[\begin{array}{ccc} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ \hline & \mathbf{0} & \end{array}\right] \qquad \mathbf{U}^{\mathrm{T}}\cdot\mathbf{b} = \left[\begin{array}{c} \mathbf{c} \\ \hline \mathbf{d} \end{array}\right]$$

donde $\mathbf{c}\in\mathbb{R}^n$ y $\mathbf{d}\in\mathbb{R}^{m-n}$. Así,

$$\begin{aligned}\min_{\mathbf{y}} \left\|\mathbf{\Sigma}\cdot\mathbf{y}-\mathbf{U}^{\mathrm{T}}\cdot\mathbf{b}\right\|_2^2 &= \min_{\mathbf{y}} \left\| \left[\begin{array}{c} \sigma_1\cdot y_1 \\ \vdots \\ \sigma_n\cdot y_n \\ \hline \mathbf{0} \end{array}\right] - \left[\begin{array}{c} \mathbf{c} \\ \hline \mathbf{d} \end{array}\right] \right\|_2^2\\ &= \min_{\mathbf{y}} \left\| \begin{bmatrix} \sigma_1\cdot y_1 - c_1 \\ \vdots \\ \sigma_n\cdot y_n - c_n \end{bmatrix} \right\|_2^2 + \|\mathbf{d}\|_2^2\end{aligned}$$

Para alcanzar el mínimo basta con anular el primer término, lo cual sucede si y solo si se toma $\mathbf{y}=\left(\frac{c_1}{\sigma_1},\ldots,\frac{c_n}{\sigma_n}\right)$.

(II) Si $\mathbf{A}$ no es de rango columna completo ($k<n$), solo las primeras k entradas de la diagonal de $\mathbf{\Sigma}$ son no nulas. Escribimos entonces:

$$\mathbf{U}^{\mathrm{T}}\cdot\mathbf{b} = \left[\begin{array}{c} \mathbf{c} \\ \hline \mathbf{d} \end{array}\right]$$

con $\mathbf{c} \in \mathbb{R}^k$ y $\mathbf{d} \in \mathbb{R}^{m-k}$. Ahora,

$$\begin{aligned}
\min_{\mathbf{y}} \left\|\mathbf{\Sigma} \cdot \mathbf{y} - \mathbf{U}^{\mathrm{T}} \cdot \mathbf{b}\right\|_2^2 &= \min_{\mathbf{y}} \left\| \begin{bmatrix} \sigma_1 \cdot y_1 \\ \vdots \\ \sigma_k \cdot y_k \\ \hline \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{c} \\ \hline \mathbf{d} \end{bmatrix} \right\|_2^2 \\
&= \min_{\mathbf{y}} \left\| \begin{bmatrix} \sigma_1 \cdot y_1 - c_1 \\ \vdots \\ \sigma_k \cdot y_k - c_k \end{bmatrix} \right\|_2^2 + \|\mathbf{d}\|_2^2
\end{aligned}$$

De nuevo, para alcanzar el mínimo, debe anularse el primer término. Existen infinitas soluciones, las cuales se logran tomando $\mathbf{y} = \left(\frac{c_1}{\sigma_1}, \ldots, \frac{c_k}{\sigma_k}, y_{k+1}, \ldots, y_n\right)$, con $y_{k+1}, \ldots, y_n \in \mathbb{R}$ cualesquiera. Una posibilidad es tomarlos a todos iguales a 0, lo cual minimiza la norma de la solución $\mathbf{x}$ que se encontrará luego.

En ambos casos, una vez hallado $\mathbf{y}$, resta resolver el sistema $\mathbf{V}^{\mathrm{T}}\mathbf{x} = \mathbf{y}$ para hallar la solución $\mathbf{x}$ al problema de cuadrados mínimos.

Utilizar SVD para resolver el problema es más costoso que hacerlo mediante QR. En particular, resolver CML vía la factorización en valores singulares tiene un costo cercano al de QR cuando $m \gg n$, y cercano a $4n^2m - \frac{4}{3}$ para m más chico. Una comparación precisa entre los costos de usar QR y SVD también depende de la máquina que esté siendo usada. Más allá de las consideraciones de eficiencia, en casos de matrices de rango incompleto, su estabilidad numérica hace preferible emplear SVD por encima de QR.

11.3. Propiedades Varias

- $\|u + v\|_2^2 = \|u\|_2^2 + \|v\|_2^2 + 2u^t v$.
- Si $s \in S$, $t \in S^{\perp}$, entonces existe una única $s + t = w$, para todo $w \in \mathbb{R}^n$.
- **Teorema rango-nulidad**: Si A es una matriz $m \times n$, entonces

$$dim(\mathrm{Im}(A)) + dim(\mathrm{Nu}(A)) = n$$

- $dim(\mathrm{Im}(A)) = rango(A)$.

Capítulo 12

Interpolación

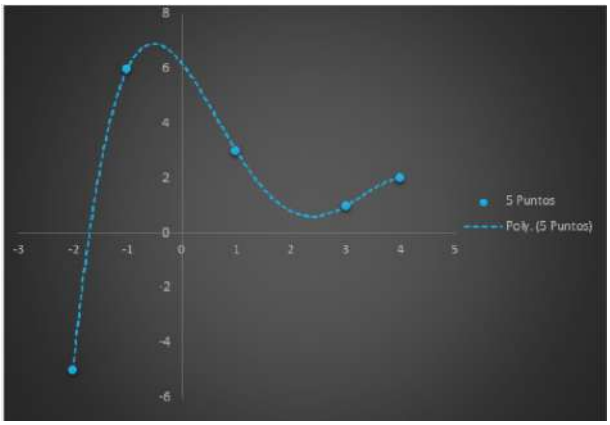
Este capítulo está dedicado al tema de **Interpolación**. Vamos a comenzar definiendo cuál es el problema matemático que queremos resolver. En este caso, tenemos un conjunto de $n+1$ pares ordenados $(x_0, y_0), \ldots, (x_n, y_n)$, donde la primera variable es la variable independiente y la segunda es la variable dependiente, y buscamos una función $f(x)$ tal que *interpole* a los datos. Es decir, buscamos $f(x)$ tal que

$$f(x_i) = y_i \forall\, i = 0, \ldots, n$$

Este problema tiene múltiples aplicaciones; resulta útil, por ejemplo

- Para derivar o integrar una versión más simple de una función complicada.
- Cuando se tiene un conjunto de pares de datos (x, y) que provienen de una función $y = f(x)$, o una medición cualquiera. Encontrar un polinomio que interpola los datos significa reemplazar la información con una regla que puede ser evaluada en una cantidad finita de pasos. Si bien es poco realista esperar a que el polinomio represente exactamente a la función verdadera f en nuevos datos de entrada, es posible que aproxime lo suficientemente cerca para resolver problemas prácticos. Por ejemplo, para calcular funciones como el seno, se elige puntos sobre la curva sinusoidal, y se guarda el polinomio interpolante en la calculadora como si fuese una versión comprimida de la función seno.
- Las CPU suelen tener métodos rápidos en hardware para sumar y multiplicar números de punto flotante, que son las únicas operaciones necesarias para evaluar un polinomio. Luego, es posible aproximar funciones complicadas interpolando polinomios para hacerlas computables con estas dos operaciones de hardware.

Recordemos que cuando estudiamos el problema de CML, también teníamos un conjunto de pares ordenados, pero en ese caso buscábamos una función que mejor *aproxime* al conjunto de datos, bajo algún criterio. En este caso, la interpolación nos **exige** que el valor de la función sea exactamente igual al valor de la variable dependiente. Esto es la diferencia sustancial entre un problema de aproximación y un problema de interpolación.



En particular, dentro del tema de interpolación, nos vamos a restringir a trabajar con polinomios. Es decir, vamos a tener un conjunto de $n+1$ puntos, y vamos a buscar un polinomio de grado a lo sumo n tal que interpole al conjunto de datos. Esto se debe principalmente a que los polinomios son una clase de funciones muy estudiada, y tienen múltiples propiedades deseables; por ejemplo, son sencillos de evaluar, derivar e integrar.

La primera pregunta que nos vamos a hacer es si existe un polinomio de tales condiciones, para luego, en caso de que exista, preguntarnos si ese polinomio es único.

12.1. Polinomio Interpolante de Lagrange

12.1.1. Existencia

Vamos a comenzar por la existencia, y para eso vamos a considerar unos polinomios muy particulares

$$L_{nk}(x) = \prod_{\substack{j=0 \\ j\neq k}}^{n} \frac{(x-x_j)}{(x_k-x_j)}$$

donde n hace referencia al grado del polinomio, y k va a hacer referencia al dato omitido del conjunto de pares ordenados.

¿Qué particularidad tiene este polinomio? En principio, es un polinomio de grado n, y si lo evaluamos en alguno de los puntos de nuestro conjunto de datos x_i, $i \neq k$, obtenemos

$$L_{nk}(x_i) = 0, \qquad i \neq k$$

En el caso en el que se lo evalúe en x_k, obtenemos

$$L_{nk}(x_k) = 1$$

Si al polinomio L_{nk} lo multiplicamos por y_k, entonces vamos a obtener nuevamente un polinomio de grado n, va a seguir valiendo que $y_k L_{nk}(x_i) = 0$, para $i \neq k$, pero cuando lo evaluamos en x_k, el resultado es $y_k L_{nk}(x_i) = y_k$.

Entonces, este es un polinomio que se anula en todos los puntos de la muestra, salvo en el x_k, donde vale y_k, que es lo que estábamos buscando. Por lo tanto, si sumamos todos esos polinomios, obtenemos un polinomio de grado a lo sumo n

$$P(x) := \sum_{k=0}^{n} f(x_k) \cdot L_{nk}(x)$$

¿Qué particularidad tiene este polinomio? Si lo evaluamos en x_i, obtenemos $P(x_i) = y_i \forall i = 0, \cdots, n$. Entonces, este es un polinomio que interpola en todos los puntos de la muestra. Por lo tanto, el polinomio interpolante existe para todo conjunto de datos que cumpla $x_i \neq x_j \forall i \neq j$.

12.1.2. Fórmula del Error

Recordemos del análisis el polinomio de Taylor, construido a partir del conocimiento en un punto de el valor de la función y de las n primeras derivadas, y había una fórmula que relacionaba a la función con el polinomio de Taylor.

Para el caso de interpolación vamos a tener algo similar, y lo que vamos a decir es que si $x_0, \ldots, x_n \in [a, b]$ y $f \in \mathcal{C}^{n+1}([a, b])$, es decir que tiene derivada continua hasta orden $n+1$, entonces podemos afirmar el valor de la función en un punto $x \in [a, b]$ cualquiera es igual a

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \cdot \prod_{i=0}^{n}(x - x_i)$$

donde $\xi_{\mathbf{x}}$ es algún punto intermedio del intervalo $[a, b]$

Esta expresión se parece bastante a la expresión que teníamos en el polinomio de Taylor. También aparecía la derivada de orden $n + 1$ en un punto intermedio del intervalo, pero simplemente estaba multiplicado por $(x - x_0)^{n+1}$, si x_0 era el punto sobre el cual estaba desarrollado el polinomio de Taylor.

En el caso del Polinomio Interpolador, como este polinomio no está construido sobre datos de un único punto, sino que a partir de un $n + 1$ datos, entonces es que nos aparece esta productoria.

Métodos Numéricos modo virtual
(pandemia COVID-19)
Material Complementario

Interpolación - Error - versión 1.0

Este es material complementario de las diapos de la clase de interpolación usadas durante el dictado virtual (pandemia COVID-19). En este documento deducimos la expresión del error del polinomio interpolante.

Sean $f(x)$ una función definida en un intervalo $[a, b]$ y pares ordenados $(x_i, f(x_i))$, $x_i \in [a, b]$ para $i = 0, \ldots, n$. Sabemos que existe un polinomio $P(x)$ de grado $\leq n$ tal que $P(x_i) = f(x_i)$ para todo $i = 0, \ldots, n$.
Dado $\bar{x} \in [a, b]$, $\bar{x} \neq x_i$ para todo $i = 0, \ldots, n$ estamos interesados en saber que error cometemos si aproximamos el valor de $f(\bar{x})$ por $P(\bar{x})$. En la próxima propiedad daremos respuesta a esto.

Proposición: Sea $f(x) \in C^{n+1}[a, b]$, $(x_i, f(x_i))$, $x_i \in [a, b]$ para $i = 0, \ldots, n$. Consideremos $P(x)$ el polinomio interpolante de grado $\leq n$ y $\bar{x} \in [a, b]$. Existe $\xi(\bar{x})$ tal que

$$f(\bar{x}) = P(\bar{x}) + \frac{f^{n+1}(\xi(\bar{x}))}{(n+1)!}(\bar{x} - x_0)(\bar{x} - x_1)\ldots(\bar{x} - x_n)$$

Demostración:

- Caso a: $\bar{x} = x_k$ para algún $k \in \{0, \ldots, n\}$.
  Sabemos que $P(x_k) = f(x_k)$ porque $P(x)$ es el polinomio interpolante en los puntos x_i para $i = 0, \ldots, n$. Por otro lado $(x_k - x_0)(x_k - x_1)\ldots(x_k - x_n)$ se anula. Entonces $\xi(\bar{x})$ puede elegirse en forma arbitraria y la identidad es verdadera.

- Caso b: $\bar{x} \neq x_k$ para todo $k \in \{0, \ldots, n\}$.
  Definimos una función $g(t) = f(t) - P(t) - \big(f(\bar{x}) - P(\bar{x})\big) \prod_{i=0}^{n} \frac{(t-x_i)}{(\bar{x}-x_i)}$ para $t \in [a, b]$.

  Veamos que propiedades podemos deducir que cumple la función $g(t)$. Sabemos que:

  1. $f(t) \in C^{n+1}[a, b]$ por hipótesis.
  2. $P(t) \in C^{n+1}[a, b]$ porque es un polinomio.
  3. $\prod_{i=0}^{n} \frac{(t-x_i)}{(\bar{x}-x_i)} \in C^{n+1}[a, b]$ porque es un polinomio.

entonces podemos concluir que $g(t) \in C^{n+1}[a,b]$.

¿Qué más podemos deducir? Evaluemos a $g(t)$ en los puntos de interpolación:

$g(x_k) = f(x_k) - P(x_k) - \big(f(\bar{x}) - P(\bar{x})\big) \prod_{i=0}^{n} \frac{(x_k - x_i)}{(\bar{x} - x_i)}$

La última productoria se anula ya que $k \in \{0, \ldots, n\}$. Además $f(x_k) = P(x_k)$. Por lo tanto

$$g(x_k) = 0 \text{ para todo } k \in \{0, \ldots, n\}$$

Ahora evaluemos a $g(t)$ en $\bar{x}$:

$g(\bar{x}) = f(\bar{x}) - P(\bar{x}) - \big(f(\bar{x}) - P(\bar{x})\big) \prod_{i=0}^{n} \frac{(\bar{x} - x_i)}{(\bar{x} - x_i)}$.

La última productoria vale 1, entonces $g(\bar{x}) = f(\bar{x}) - P(\bar{x}) - \big(f(\bar{x}) - P(\bar{x})\big)$, lo que implica que $g(\bar{x}) = 0$.

Resumiendo lo que sabemos de $g(t)$ es que:

- $g(t) \in C^{n+1}[a,b]$
- $g(t)$ se anula en $x_0, \ldots, x_n$ y $\bar{x}$.

Recordamos un resultado clásico del análisis (teorema de Rolle) que nos dice que si tenemos una función h continua en $[c,d]$ y diferenciable en (c,d) tal que $h(c) = h(d)$, entonces existe $\xi \in (a,b)$ tal que $h'(\xi) = 0$.
La función $g(t)$ tiene al menos $n+2$ puntos donde se anula. Si ordenamos $x_0, x_1, \ldots, x_n, \bar{x}$ de menor a mayor, podemos aplicar el teorema de Rolle a la función $g(t)$ en cada intervalo definido por dos puntos sucesivos (la función $g(t)$ coincide en valor en los extremos de cada intervalo ya que vale cero en ambos puntos). Entonces, podemos afirmar que $g'(t)$ se anula en al menos un punto en cada intervalo. Por lo tanto podemos afirmar que $g'(t)$ se anula en al menos $n+1$ puntos.

Si este mismo razonamiento lo aplicamos ahora a la función $g'(t)$ en los intervalos definidos por los $n+1$ puntos donde se anula , llegaremos a la conclusión que $g''(t)$ se anula en al menos n puntos.

Repitiendo el proceso, llegaremos a que $g^{n+1}(t)$ se anula en al menos 1 punto. Este punto depende de los valores $x_0, x_1, \ldots, x_n, \bar{x}$. Llamemos $\xi(\bar{x})$ a dicho punto.

Volvamos ahora a la expresión de $g(t)$:

$$g(t) = f(t) - P(t) - \big(f(\bar{x}) - P(\bar{x})\big) \prod_{i=0}^{n} \frac{(t - x_i)}{(\bar{x} - x_i)}$$

Desde aqui, derivando término a término, podemos obtener la expresión de la derivada de orden $n+1$:

$$g^{n+1}(t) = f^{n+1}(t) - P^{n+1}(t) - \big(f(\bar{x}) - P(\bar{x})\big)\Big(\prod_{i=0}^{n} \frac{(t - x_i)}{(\bar{x} - x_i)}\Big)^{n+1}$$

Sabemos que $P(t)$ es un polinomio de grado $\leq n$, por lo tanto la deriva de orden $n+1$ es cero. Además $\prod_{i=0}^{n} \frac{(t - x_i)}{(\bar{x} - x_i)}$ es un polinomio de grado $n+1$, por lo tanto la deriva de orden $n+1$ es igual al coeficiente que acompaña a la potencia de orden $n+1$ (que vale $\prod_{i=0}^{n} \frac{1}{(\bar{x} - x_i)}$), multiplicada por $(n+1)!$

De estas observaciones, deducimos que:

$$g^{n+1}(t) = f^{n+1}(t) - \big(f(\bar{x}) - P(\bar{x})\big)(n+1)!\Big(\prod_{i=0}^{n} \frac{1}{(\bar{x} - x_i)}\Big)$$

Si ahora evaluamos la expresión anterior en $\xi(\bar{x})$, tendremos que

$$g^{n+1}(\xi(\bar{x})) = 0 = f^{n+1}(\xi(\bar{x})) - \big(f(\bar{x}) - P(\bar{x})\big)(n+1)!(\prod_{i=0}^{n} \frac{1}{(\bar{x} - x_i)})$$

$$f^{n+1}(\xi(\bar{x})) = \big(f(\bar{x}) - P(\bar{x})\big)(n+1)!(\prod_{i=0}^{n} \frac{1}{(\bar{x} - x_i)})$$

$$\frac{f^{n+1}(\xi(\bar{x}))}{(n+1)!} \prod_{i=0}^{n} (\bar{x} - x_i) = \big(f(\bar{x}) - P(\bar{x})\big)$$

$$P(\bar{x}) + \frac{f^{n+1}(\xi(\bar{x}))}{(n+1)!} \prod_{i=0}^{n} (\bar{x} - x_i) = f(\bar{x})$$

■

12.1.3. Unicidad

Otra propiedad destacada del polinomio de Lagrange es su **unicidad**.

Teorema 12.1.1. *Dados (x_i, y_i) para $i = 0, \ldots, n$, el polinomio interpolante de grado menor o igual a n existe y es único.*

Demostración. Sea P_1 el polinomio interpolador de Lagrange en los puntos $x_0, \ldots, x_n$, y supongamos que existe otro polinomio P_2, de grado menor o igual que n, tal que, para todo $i \in \{0, \ldots, n\}$, $P_2(x_i) = f(x_i)$.

Ahora bien, si pensamos a $P_2(x)$ como un polinomio interpolante de la función $f(x) = P_1(x)$, pues $P_2(x_i) = P_1(x_i)$ para $i = 1, \ldots, n$, y por tanto P_2 interpola a P_1. Luego, podemos escribir a P_1 como

$$P_1(x) = P_2(x) + \frac{P_1^{(n+1)}(\xi_x)}{(n+1)!} \cdot \prod_{i=0}^{n}(x - x_i) \quad \text{para algún } \xi_x \in [x_0, x_n]$$

Pero como P_1 es un polinomio de grado n, $P_1^{(n+1)}(x) = 0$, entonces

$$\begin{aligned} P_1(x) &= P_2(x) + \overbrace{\frac{P_1^{(n+1)}(\xi_x)}{(n+1)!}}^{=0} \cdot \prod_{i=0}^{n}(x - x_i) \\ &= P_2(x) \end{aligned}$$

Entonces, el polinomio interpolante existe y es único. ■

Notemos que este resultado nos habla de los polinomios de grado menor o igual a n. Sin embargo, ¿qué pasa con los polinomios de grado mayor o igual a $n+1$? Una manera de construir un polinomio de grado $n+1$ que interpola a los n pares ordenados consiste en agregar un nuevo punto por el que no pase el polinomio de grado n, e interpolar nuevamente. Por lo tanto, hay infinitos polinomios de grado $n+1$ que interpolan al conjunto de datos. Otra manera de construirnos un polinomio de grado $n+1$ que interpola al conjunto de datos consiste en sumarle al polinomio interpolante de grado menor o igual a n un polinomio de grado $n+1$ tal que cada una de sus raíces $r_i = x_i$, es decir

$$P_{n+1}(x) = P_n(x) + \alpha \cdot (x - x_1)(x - x_2) \cdots (x - x_n), \qquad \alpha \neq 0$$

12.2. Diferencias divididas

Hasta el momento tenemos que, dado un conjunto de $n+1$ pares ordenados, podemos construir un polinomio interpolante, el cual es único, y está relacionado con la función que estamos tratando de interpolar mediante la fórmula del error. Sin embargo, el método de interpolación de Lagrange es raramente utilizado para el cómputo del polinomio interpolante, ya que hay otros métodos alternativos que resultan en formas más fáciles de manejar y menos costosas. Con esta idea en mente, vamos a volver un poco para atrás, y vamos a analizar la expresión del polinomio interpolante.

$$P(x) := \sum_{k=0}^{n} f(x_k) \cdot L_{nk}(x) \qquad \text{, donde}$$

$$L_{nk}(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{(x - x_j)}{(x_k - x_j)}$$

Supongamos que tenemos construido este polinomio interpolante, y luego se nos pide añadir un nuevo par ordenado al conjunto de datos. Entonces, cada uno de los polinomios L_{nk} debe ser reconstruido,

porque la productoria va a ir hasta $n+1$, y además vamos a tener un nuevo polinomio el cual tenemos que sumarlo a $P(x)$. Entonces, pareciera que tenemos que hacer un trabajo casi desde cero por el solo hecho de añadir un nuevo dato al conjunto de pares ordenados.

Sabemos que tenemos distintas maneras de expresar a un mismo polinomio, entonces vamos a ver si al cambiar esta forma de expresarlo, podemos obtener una forma conveniente de añadir nuevos datos al conjunto de pares ordenados, y para eso vamos a hacer uso de las **diferencias divididas**.

Partimos de la siguiente definición recursiva:

- La **diferencia dividida de orden 0** en x_j es, para $j = 0, \ldots, n$:

$$f[x_j] := f(x_j).$$

- La **diferencia dividida de orden 1** en x_j es, para $j = 0, \ldots, n-1$:

$$f[x_j, x_{j+1}] := \frac{f[x_{j+1}] - f[x_j]}{x_{j+1} - x_j}.$$

- La **diferencia dividida de orden $\boldsymbol{k}$** en x_j es, para $j = 0, \ldots, n-k$:

$$f[x_j, \ldots, x_{j+k}] := \frac{f[x_{j+1}, \ldots, x_{j+k}] - f[x_j, \ldots, x_{j+k-1}]}{x_{j+k} - x_j}.$$

Afirmamos que, si P es el polinomio interpolador para los puntos $x_0, \ldots, x_n$, entonces

$$P(x) = \sum_{i=0}^{n} \left(f[x_0, \ldots, x_i] \prod_{j=0}^{i-1} (x - x_j) \right)$$

es decir,

$$\begin{aligned} P(x) = f[x_0] &+ f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &+ \cdots + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1) \ldots (x - x_n). \end{aligned}$$

expresión que se conoce como **diferencias divididas**, y a veces como **forma de Newton**, del polinomio interpolador.

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Métodos Numéricos modo virtual
(pandemia COVID-19)
Material Complementario

Interpolación - Diferencias Divididas - versión 1.0

Este es material complementario de las diapos de la clase de interpolación usadas durante el dictado virtual (pandemia COVID-19). En este documento deducimos la expresión del polinomio interpolante mediante el uso de diferencias divididas.

Sean $f(x)$ una función definida en un intervalo $[a, b]$ y pares ordenados $(x_i, f(x_i))$, $x_i \in [a, b]$ para $i = 0, \ldots, n$. Sabemos que existe un polinomio $P(x)$ de grado $\leq n$ tal que $P(x_i) = f(x_i)$ para todo $i = 0, \ldots, n$.
La expresion para este (único!) polinomio es $P(x) = \sum_{k=0}^{n} y_k L_{nk}(x)$ donde $L_{nk} = \prod_{i=0, i\neq k}^{n} \frac{(x-x_i)}{(x_k-x_i)}$.
En el caso que se agregara un punto más al conjunto de los puntos de interpolación, deberíamos rehacer la expresión de cada término. ¿Cómo podremos evitar este trabajo?
Definimos las Diferencias Divididas como

- Orden 0 : $f[x_i] = f(x_i)$
- Orden 1 : $f[x_i, x_{i+1}] = \frac{f[x_{i+1}]-f[x_i]}{x_{i+1}-x_i}$
- Orden k : $f[x_i, x_{i+1}, \ldots, x_{i+k}] = \frac{f[x_{i+1},\ldots,x_{i+k}]-f[x_i,\ldots x_{i+k-1}]}{x_{i+k}-xi}$

Veamos que el polinomio interpolante se puede expresar en función de estas diferencias

Proposición: *Dada $f(x)$ un función definida en $[a, b]$ y pares ordenados $(x_i, f(x_i))$, $x_i \in [a, b]$ para $i = 0, \ldots, n$, el polinomio interpolante se puede expresar como*

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

Demostración:
Haremos la demostración por inducción en n.

- Caso base: n=1 Los puntos de interpolación son x_0 y x_1 y el polinomio interpolante tiene grado ≤ 1. Por la expresión del polinomio en función de los L_{nk}, tenemos que

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

Sumando y restando x_0 en el primer término

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

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

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

Simplificando en el primer término y sacando factor común $(x - x_0)$ entre los dos últimos, obtenemos:

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

Usando las definiciones de las diferencias divididas obtenemos la expresión de $P(x)$ en función de ellas:

$$P(x) = f[x_0] + f[x_0, x_1](x_1 - x_0)$$

- Paso inductivo

  Sea $P_n(x)$ el polinomio interpolante en los puntos $x_0, \ldots, x_n$, es decir $P_n(x_i) = f(x_i)$. Por hipótesis inductiva,

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

  Sea $Q_n(x)$ el polinomio interpolante en los puntos $x_1, \ldots, x_{n+1}$, es decir $Q_n(x_i) = f(x_i)$. Por hipótesis inductiva,

$$Q_n(x) = f\left[x_1\right] + f\left[x_1, x_2\right](x - x_1) + f\left[x_1, x_2, x_3\right](x - x_1)(x - x_1) + \ldots + f\left[x_1, \ldots, x_{n+1}\right](x - x_1)\ldots(x - x_n)$$

  Sea $P_{n+1}(x)$ el polinomio interpolante en los puntos $x_0, \ldots, x_{n+1}$. Queremos ver que

$$P_{n+1}(x) = f\left[x_0\right] + \ldots + f\left[x_0, \ldots, x_n\right](x - x_0)\ldots(x - x_{n-1}) + f\left[x_0, \ldots, x_{n+1}\right](x - x_0)\ldots(x - x_n)$$

  Nos construimos el polinomio $P(x) = P_n(x) + a(x - x_0)\ldots(x - x_n)$. Veamos que propiedades tiene $P(x)$.

  Claramente $P(x)$ es un polinomio de grado $\leq n + 1$ y además es fácil ver que $P(x_i) = P_n(x_i) = f(x_i)$ para $i = 0, \ldots, n$.

  Por otro lado, eligiendo convenientemente a podemos conseguir que $P(x_{n+1}) = f(x_{n+1})$. ¿Cómo hacemos esto? Si queremos que $P(x_{n+1}) = f(x_{n+1})$, entonces debe cumplirse que $P_n(x_{n+1}) + a(x_{n+1} - x_0)\ldots(x_{n+1} - x_n) = f(x_{n+1})$. Basta tomar $a = \frac{f(x_{n+1}) - P_n(x_{n+1})}{(x_{n+1} - x_0)\ldots(x_{n+1} - x_n)}$ que siempre está definido.

  En conclusión, $P(x)$ es un polinomio de grado $\leq n+1$ que interpola en los puntos $x_0, \ldots, x_n, x_{n+1}$. Como ya sabemos que el polinomio interpolante en un conjunto de puntos es único, entonces $P(x) = P_{n+1}(x)$.

  Si demostramos que $a = f[x_0, \ldots, x_{n+1}]$ entonces tendremos la propiedad requerida.

  Consideremos un nuevo polinonio $Q(x) = Q_n(x) + \frac{(x - x_{n+1})}{(x_{n+1} - x_0)}(Q_n(x) - P_n(x))$. Por la expresión de $Q(x)$ deducimos que es un polinomio de grado $\leq n + 1$

  Vamos a evaluar a $Q(x)$ en los puntos x_i para todo $i = 0, \ldots, n + 1$.

$$Q(x_i) = Q_n(x_i) + \frac{(x_i - x_{n+1})}{(x_{n+1} - x_0)}(Q_n(x_i) - P_n(x_i))$$

Si $i = 1, \ldots, n$, sabemos que x_i es un punto de interpolación tanto para $Q_n(x)$ como para $P_n(x)$. Entonces $Q_n(x_i) - P_n(x_i) = 0$, de donde se deduce que $Q(x_i) = Q_n(x_i) = f(x_i)$.

Si evaluamos en $x_i = x_{n+1}$, el segundo término se anula y resulta que $Q(x_{n+1}) = Q_n(x_{n+1}) = f(x_{n+1})$ ya que x_{n+1} es un punto de interpolación para $Q_n(x)$.

Finalmente, si evaluamos en x_0, $Q(x_0) = Q_n(x_0) + \frac{(x_0 - x_{n+1})}{(x_{n+1} - x_0)}(Q_n(x_0) - P_n(x_0)) = Q_n(x_0) - (Q_n(x_0) - P_n(x_0)) = P_n(x_0)$. Como x_0 es punto de interpolación para $P_n(x)$, sabemos que $P_n(x_0) = f(x_0)$ por lo tanto resulta $Q(x_0) = f(x_0)$

En resumen: $Q(x_i) = f(x_i)$ para todo $i = 0, \ldots, n+1$. Pero entonces $Q(x) = P_{n+1}(x)$ ya que sabemos que el polinomio interpolante es único.

Si dos polinomios son iguales, entonces los coeficientes que acompañan a cada potencia deben coincidir. Recordemos la expresion de $P_{n+1}(x)$ y $Q(x)$:

$$P_{n+1}(x) = P_n(x) + a(x - x_0)\ldots(x - x_n)$$
$$Q(x) = Q_n(x) + \frac{(x - x_{n+1})}{(x_{n+1} - x_0)}(Q_n(x) - P_n(x))$$

El coeficiente que acompaña a la potencia $n+1$ de $P_{n+1}(x)$ es a que es el que queremos demostrar que vale $f[x_0, \ldots, x_{n+1}]$.

El coeficiente que acompaña a la potencia $n+1$ de $Q(x)$ es el coeficiente de la potencia n de $Q_n(x)$, menos el coeficiente de la potencia n de $P_n(x)$, dividido por $(x_{n+1} - x_0)$.

El coeficiente de la potencia n de $Q_n(x)$, por hipótesis inductiva es $f[x_1, \ldots, x_{n+1}]$.

El coeficiente de la potencia n de $P_n(x)$, por hipótesis inductiva es $f[x_0, \ldots, x_n]$.

Entonces $a = \frac{f[x_1,\ldots,x_{n+1}] - f[x_0,\ldots,x_n]}{(x_{n+1} - x_0)}$ que es la definición de $f[x_0, \ldots, x_{n+1}]$.

■

¿Para qué nos resulta útil esta expresión? Si tenemos un conjunto de datos, entonces podemos expresar al polinomio interpolante de grado n como

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

de manera que si agregamos un nuevo punto, lo que nos va a quedar es que el nuevo polinomio interpolante es igual a

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

Entonces, P_n nos sirve para calcular el nuevo polinomio interpolante, y no tenemos que rehacer todos los cálculos devuelta. ¿Cómo nos conviene calcular este nuevo término? Este se puede calcular de forma eficiente, ya que su coeficiente es la diferencia dividida $f[x_0, \ldots, x_{n+1}]$, que, gracias a la estructura recursiva de las diferencias divididas, se puede computar reutilizando resultados anteriores

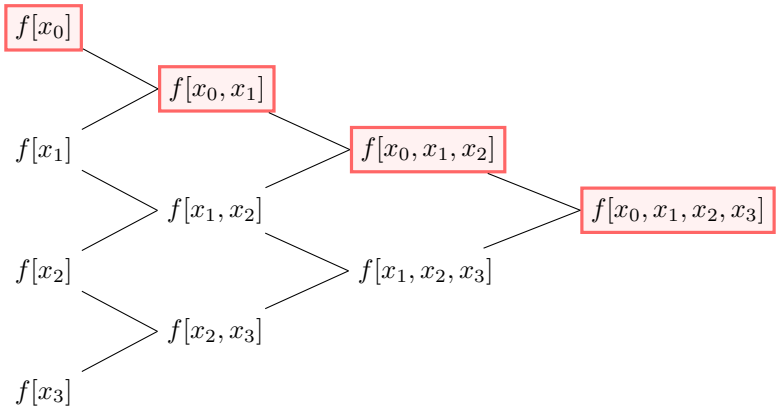


Figura 12.1: Diferencias divididas que es necesario computar para agregar un cuarto punto a un conjunto de tres puntos ya interpolados.

Por lo tanto, si quisiéramos construirnos el polinomio interpolante en este conjunto de datos, simplemente hacemos

$$\begin{aligned} P(x) = f(x_0) &+ f[x_0, x_1](x - x_0) + f[x_0.x_1.x_2](x - x_0)(x - x_1) \\ &+ f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) \end{aligned}$$

12.3. Método de Neville

Vamos a ver ahora otra manera de expresar el polinomio interpolante. El polinomio interpolador para los puntos $x_1, \ldots, x_n$ admite también se puede expresar en función de dos polinomios que interpolan $n - 1$ puntos. Luego, para cualesquiera $0 \leq i, j \leq n$, con $i \neq j$, se tiene que

$$P(x) = \frac{(x - x_j) \cdot P_j(x) - (x - x_i) \cdot P_i(x)}{x_i - x_j},$$

donde P_i denota al polinomio interpolador para los puntos $x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$, y P_j al polinomio interpolador para $x_0, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n$.

Veamos que esto es cierto. Por la unicidad del polinomio interpolante, nos basta ver que esta expresión interpola a todos los puntos desde $x_1, \ldots, x_n$

- En x_i, tenemos que

$$\begin{aligned} P(x_i) &= \frac{(x_i - x_j)\cdot P_j(x_i) - \overbrace{(x_i - x_i)}^{=0}\cdot P_i(x_i)}{x_i - x_j} \\ &= \frac{\cancel{(x_i - x_j)}\cdot P_j(x_i)}{\cancel{x_i - x_j}} \\ &= P_j(x_i) = f(x_i) \qquad \text{pues } P_j \text{ interpola a } x_i \end{aligned}$$

- Análogamente, en x_j, tenemos que

$$\begin{aligned} P(x_j) &= \frac{\overbrace{(x_j - x_j)}^{=0}\cdot P_j(x_j) - (x_j - x_i)\cdot P_i(x_j)}{x_i - x_j} \\ &= \frac{\cancel{-(x_j - x_i)}\cdot P_i(x_j)}{\cancel{x_i - x_j}} \\ &= P_i(x_i) = f(x_i) \qquad \text{pues } P_i \text{ interpola a } x_j \end{aligned}$$

- En x_k, con $k = 1, \ldots, n, k \neq i, j$, tenemos que

$$\begin{aligned} P(x_k) &= \frac{(x_k - x_j)\cdot P_j(x_k) - (x_k - x_i)\cdot P_i(x_k)}{x_i - x_j} \\ &= \frac{(x_k - x_j)\cdot f(x_k) - (x_k - x_i)\cdot f(x_k)}{x_i - x_j} \qquad \text{pues } P_i, P_j \text{ interpolan a } x_k \\ &= \frac{(x_i - x_j)\cdot f(x_k)}{x_i - x_j} \\ &= f(x_k) \end{aligned}$$

Por lo tanto, esta expresión para el polinomio interpolante es válida.

Esta escritura recursiva da origen al **método de Neville**, que es otra manera de construir polinomios interpoladores de forma incremental, permitiendo obtener un polinomio interpolador para $n+1$ puntos a partir de uno que interpola un subconjunto de n de estos puntos. Por ejemplo, si buscamos al polinomio interpolante del conjunto x_1, x_2, x_3, x_4, lo obtenemos aplicando

$$P(x) = \frac{(x - x_j)\cdot P_j(x) - (x - x_i)\cdot P_i(x)}{x_i - x_j},$$

en el siguiente orden

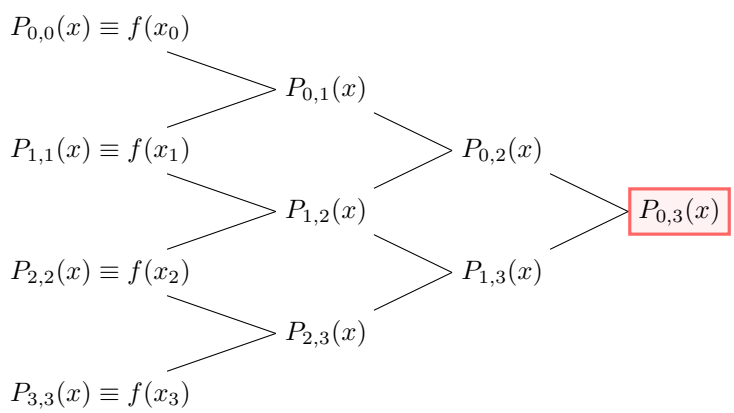


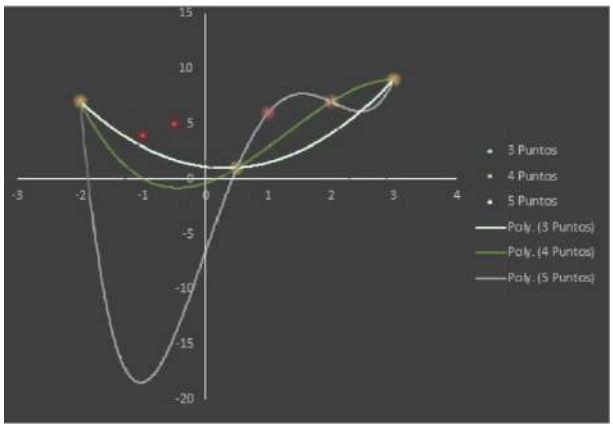
Figura 12.2: Extensión de un polinomio que interpola los puntos $x_0, \ldots, x_3$ para interpolar también el punto x_4. La notación $P_{i,j}$ indica, para $0 \leq i \leq j \leq 4$, el polinomio interpolador en los puntos $x_i, x_{i+1}, \ldots, x_j$.

La diferencia con el método de diferencias divididas es que antes teníamos valores, que terminaban siendo los coeficientes que iban acompañando a las distintas potencias. En este caso, todos estos elementos son polinomios.

12.4. Interpolación fragmentaria

12.4.1. Variando el grado

Cuando tenemos un conjunto de datos, podemos ir viendo qué es lo que ocurre con el polinomio interpolante a partir de ir considerando cada vez más datos, y por tanto va a ir variando el grado. Supongamos que tenemos este conjunto de 7 puntos, y consideramos el polinomio interpolante de $3, 4$ y 5 puntos.



Notemos que a medida que aumentamos la cantidad de puntos, a mayor grado del polinomio, mayor oscilaciones tiene el polinomio. Esto es algo que nos va a ocurrir siempre que aumentemos el grado, pero no es un comportamiento deseable, ya que si la interpolación varía demasiado, entonces la aproximación a un dato fuera de la tabla no será tan buena.

Entonces, queremos poder aumentar la cantidad de puntos a considerar, pero tener más puntos implica aumentar el grado polinomio, y en consecuencia una mayor cantidad de oscilaciones. Frente a este problema, aparece lo que se conoce como **interpolación fragmentaria**.

La idea de la interpolación fragmentaria es que, en vez de utilizara todos los puntos a la vez, se consideren varios intervalos más pequeños, interpolar en cada uno de esos intervalos, y la función de interpolación va a ser la función definida en cada segmento como la obtenida en cada uno de esos intervalos. El resultado ya no será un polinomio, sino una función compuesta de muchos polinomios "pegados" en los extremos.

En general, buscaremos definir n polinomios distintos, $S_1, \ldots, S_n$, uno para cada par de puntos consecutivos entre los valores a interpolar. La idea es que estos polinomios cumplan $S_i(x_{i-1}) = f(x_{i-1})$ y $S_i(x_i) = f(x_i)$, para $i = 1, \ldots, n$. Luego, podremos construir la función

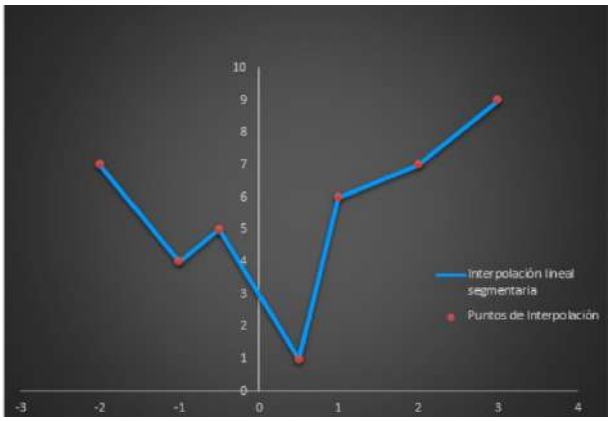
$$S(x) := \begin{cases} S_1(x) & \text{si } x \in [x_0, x_1) \\ S_2(x) & \text{si } x \in [x_1, x_2) \\ \quad\vdots & \\ S_n(x) & \text{si } x \in [x_{n-1}, x_n]. \end{cases}$$

12.4.2. Interpolación fragmentaria lineal

La forma más sencilla de interpolación fragmentaria es la **interpolación lineal**, donde cada uno de los S_i es un polinomio de grado menor o igual que 1, que interpola correctamente en los puntos x_{i-1} y

x_i[1]. Los S_i se definen, para $i = 1, \ldots, n$, como

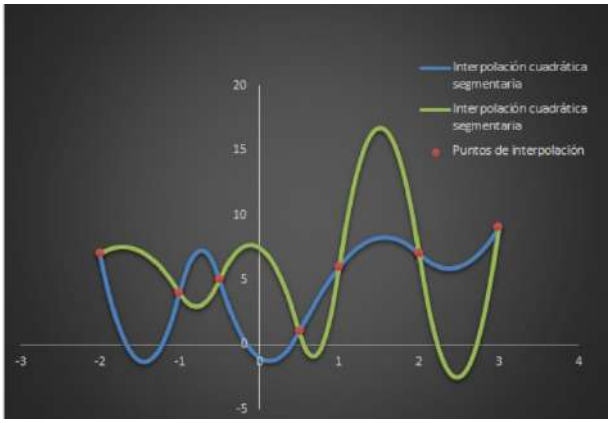
$$S_i(x) := f(x_{i-1}) + \frac{f(x_i) - f(x_{i-1})}{x_i - x_{x-1}} \cdot (x - x_{i-1}).$$



La interpolación lineal es sencilla, fácil de calcular (incluso en forma manual) y en muchas ocasiones resulta suficiente para el problema que se busca resolver. Sin embargo, la función que se obtiene, si bien es continua, no es derivable en los puntos x_i. Este problema puede resolverse utilizando polinomios de mayor grado.

12.4.3. Interpolación fragmentaria cuadrática

Si cada S_i se define para ser un polinomio **cuadrático**, se tienen más parámetros para definir, que se pueden aprovechar para que la función S obtenida sea derivable en todo su dominio.



Sean (x_i, y_i) con $x_i < x_{i+1}$ para $i = 0, \ldots, n$. Por cada par de puntos (x_i, y_i) y (x_{i+1}, y_{i+1}) para $i = 0, \ldots, n-1$ realizamos una interpolación cuadrática S_i

$$S_i(x) := a_i(x - x_i)^2 + b_i(x - x_i) + c_i$$

para lo cual necesitamos determinar los valores para los a_i, b_i y c_i de modo que se cumplan las siguientes condiciones:

(I) S es interpoladora: para $i = 0, \ldots, n-1$

$$\left.\begin{aligned} S_i(x_i) &= f(x_i) \\ S_i(x_{i+1}) &= f(x_{i+1}) \end{aligned}\right\} 2n \text{ ecuaciones}$$

[1] Cada S_i termina siendo el polinomio interpolador de Lagrange para los puntos x_{i-1}, x_i.

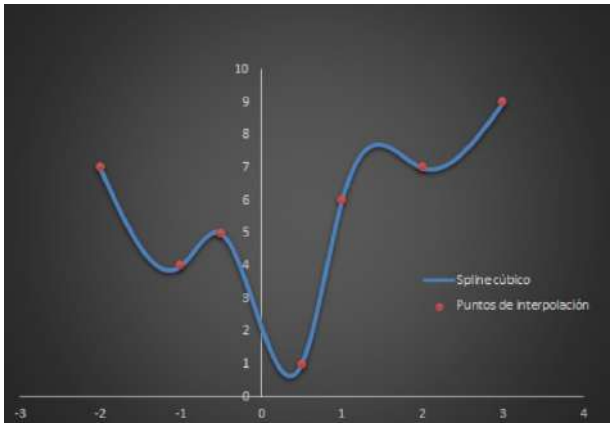
(II) S es derivable: para $i = 1, \ldots, n-1$

$$S_i'(x_i) = S_{i+1}'(x_i)\Big\} \; n-1 \text{ ecuaciones}$$

Entonces, se tiene un sistema de $3n$ incógnitas y $3n-1$ ecuaciones, lo cual deja una única ecuación libre para pedir alguna propiedad adicional. En general, esta se utiliza para controlar el comportamiento de la derivada en alguno de los extremos x_0 o x_n, con el inconveniente de que se obtiene una solución asimétrica, ya que es imposible pedir condiciones simultáneamente sobre los dos extremos y mantener la certeza de que el sistema resulta compatible.

12.4.4. Interpolación fragmentaria cúbica

Dentro de las interpolaciones fragmentarias, una de las que más se usa es la interpolación cúbica. La interpolación cúbica considera, para cada intervalo, polinomios de grado 3, y lo que le vamos a pedir a esos polinomios de grado 3 es que sean interpolantes, que la primera derivada esté bien definida, y que las segundas derivadas estén bien definidas.



Para lograrlo, los S_i se definen en la forma

$$S_i(x) := a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i$$

y los valores para cada a_i, b_i, c_i y d_i se determinan de modo tal que:

(I) S es interpoladora: para $i = 0, \ldots, n-1$

$$\left.\begin{aligned} S_i(x_i) &= f(x_i) \\ S_i(x_{i+1}) &= f(x_{i+1}) \end{aligned}\right\} 2n \text{ ecuaciones}$$

(II) S es derivable: para $i = 1, \ldots, n-1$

$$S_i'(x_i) = S_{i+1}'(x_i)\Big\} \; n-1 \text{ ecuaciones}$$

(III) S es dos veces derivable: para $i = 1, \ldots, n-1$

$$S_i''(x_i) = S_{i+1}''(x_i)\Big\} \; n-1 \text{ ecuaciones}$$

Entonces, el sistema que resulta tiene $4n$ incógnitas y $4n-2$ ecuaciones, con lo que pueden agregarse dos nuevas ecuaciones. Hay diferentes alternativas en la literatura, muchas de ellas basadas en imponer *condiciones de frontera*, es decir condicionar el comportamiento en los puntos frontera x_0 y x_n. Típicamente, estas alternativas son:

(IV) (a) Podemos pedir que la derivada segunda en los puntos frontera tomen el mismo valor que la derivada primera del polinomio interpolante. Esta condición es conocida como *Frontera sujeta*:

- $S_1'(x_0) = f'(x_0)$.
- $S_n'(x_n) = f'(x_n)$.

(b) Podemos pedir que la derivada segunda se anule en los puntos frontera, condición conocida como *Frontera natural*:

- $S_1''(x_0) = 0$
- $= S_n''(x_n) = 0$

En ambos casos, se puede demostrar que el sistema de ecuaciones que se obtiene es estrictamente diagonal dominante, lo cual asegura que siempre existe solución, y además que esta es única.

Métodos Numéricos modo virtual
(pandemia COVID-19)
Material Complementario

Interpolación segmentaria usando trazadores cúbicos - versión 1.0

Este es material complementario de las diapos de la clase de interpolación usadas durante el dictado virtual (pandemia COVID-19). En este documento deducimos la existencia y unicidad de un trazador cúbico.

Sean $f(x)$ una función definida en un intervalo $[a, b]$ y pares ordenados $(x_i, f(x_i))$, $x_i \in [a, b]$ para $i = 0, \ldots, n$. Una trazador cúbico es un función $S(x)$ tal que verifica la siguientes propiedades:

1. $S(x) = S_i(x)$ para $x \in [x_i, x_{i+1}]$ con $S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$ para $i = 0, \ldots, n-1$
2. $S(x_i) = f(x_i)$ para $i = 0, \ldots, n$
3. $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$ para $i = 0, \ldots, n-2$
4. $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$ para $i = 0, \ldots, n-2$
5. $S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$ para $i = 0, \ldots, n-2$
6. $S''(x_0) = S''(x_n) = 0$ ó $S'(x_0) = f'(x_0)$, $S'(x_n) = f'(x_n)$

El objetivo del desarrollo que haremos a continuación es mostrar porque podemos asegurar que existe una función que cumple con todas estas condiciones.
Notemos en primer lugar que tenemos 4 coeficientes a determinar para cada $S_i(x)$, lo que nos da un total de $4n$ coeficientes. La segunda propiedad nos impone $n + 1$ condiciones. La tercera, cuarta y quinta propiedad imponen $n - 1$ condiciones cada una. Tenemos entonces un total de $n + 1 + n - 1 + n - 1 + n - 1 = 4n - 2$ condiciones. La última propiedad aporta 2 condiciones. Por lo tanto, tenemos tantas condiciones como coeficientes a determinar. Debemos ver que existen coeficientes que satisfacen todas estas condiciones.
Analicemos cada una de estas condiciones. Comenzamos con $S(x_i) = f(x_i)$ para $i = 0, \ldots, n$. Como $S(x) = S_i(x)$ para $x \in [x_i, x_{i+1}]$, entonces tendremos que:

$$S(x_i) = a_i + b_i(x_i - x_i) + c_i(x_i - x_i)^2 + d_i(x_i - x_i)^3 = f(x_i) \quad \forall i = 0, \ldots, n-1$$
$$S(x_n) = a_{n-1} + b_{n-1}(x_n - x_{n-1}) + c_{n-1}(x_n - x_{n-1})^2 + d_{n-1}(x_n - x_{n-1})^3 = f(x_n)$$

De aqui derivamos que

$$\boldsymbol{a_i = f(x_i) \quad \forall i = 0, \ldots, n-1}$$
$$a_{n-1} + b_{n-1}(x_n - x_{n-1}) + c_{n-1}(x_n - x_{n-1})^2 + d_{n-1}(x_n - x_{n-1})^3 = f(x_n)$$

$$\boldsymbol{b_{n-1}(x_n - x_{n-1}) + c_{n-1}(x_n - x_{n-1})^2 + d_{n-1}(x_n - x_{n-1})^3 = f(x_n) - f(x_{n-1})}$$

La próxima condición es $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$ para $i = 0, \ldots, n-2$. Considerando la expresión de cada $S_i(x)$, tenemos la siguiente relación:

$$a_i+b_i(x_{i+1}-x_i)+c_i(x_{i+1}-x_i)^2+d_i(x_{i+1}-x_i)^3 = a_{i+1}+b_{i+1}(x_{i+1}-x_{i+1})+c_{i+1}(x_{i+1}-x_{i+1})^2+d_{i+1}(x_{i+1}-x_{i+1})^3$$

$$a_i + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 + d_i(x_{i+1} - x_i)^3 = a_{i+1} \text{ para } i = 0, \ldots, n-2$$

$$\boldsymbol{f(x_i) + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 + d_i(x_{i+1} - x_i)^3 = f(x_{i+1})} \text{ para } \boldsymbol{i = 0, \ldots, n-2}$$

La cuarta condición es $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$ para $i = 0, \ldots, n-2$

$$b_i + 2c_i(x_{i+1} - x_i) + 3d_i(x_{i+1} - x_i)^2 = b_{i+1} + 2c_{i+1}(x_{i+1} - x_{i+1}) + 3d_{i+1}(x_{i+1} - x_{i+1})^2$$

$$\boldsymbol{b_i + 2c_i(x_{i+1} - x_i) + 3d_i(x_{i+1} - x_i)^2 = b_{i+1}}$$

La quinta condición es $S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$ para $i = 0, \ldots, n-2$

$$2c_i + 6d_i(x_{i+1} - x_i) = 2c_{i+1} + 6d_{i+1}(x_{i+1} - x_{i+1})$$

$$\boldsymbol{2c_i + 6d_i(x_{i+1} - x_i) = 2c_{i+1}}$$

Finalmente, analicemos una de las dos últimas alternativas: $S''(x_0) = S''(x_n) = 0$ (la otra alternativa es similar)

$$\boldsymbol{S''(x_0) = S_0''(x_0) = 2c_0 = 0}$$

$$\boldsymbol{S''(x_n) = S_{n-1}''(x_n) = 2c_{n-1} + 6d_{n-1}(x_n - x_{n-1}) = 0}$$

Veamos entonces todas las condiciones que nos quedaron:

1. $a_i = f(x_i)$ para $i = 0, \ldots, n-1$
2. $b_{n-1}(x_n - x_{n-1}) + c_{n-1}(x_n - x_{n-1})^2 + d_{n-1}(x_n - x_{n-1})^3 = f(x_n) - f(x_{n-1})$
3. $f(x_i) + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 + d_i(x_{i+1} - x_i)^3 = f(x_{i+1})$ para $i = 0, \ldots, n-2$
4. $b_i + 2c_i(x_{i+1} - x_i) + 3d_i(x_{i+1} - x_i)^2 = b_{i+1}$ para $i = 0, \ldots, n-2$
5. $2c_i + 6d_i(x_{i+1} - x_i) = 2c_{i+1}$ para $i = 0, \ldots, n-2$
6. $c_0 = 0$
7. $2c_{n-1} + 6d_{n-1}(x_n - x_{n-1}) = 0$

La idea de lo que vamos a hacer a continuación es tratar de poner a todas las variables en función de los coeficientes a_i que ya tenemos determinados y de los c_i. Notamos $h_i = (x_i - x_{i-1})$ para $i = 1, \ldots, n$.

De (7) podemos despejar $d_{n-1} \to \boldsymbol{d_{n-1} = -\frac{c_{n-1}}{3h_n}}$.

De (2) podemos despejar $b_{n-1} \to b_{n-1} = \frac{(f(x_n)-f(x_{n-1})-c_{n-1}h_n^2-d_{n-1}h_n^3)}{h_n}$. Reemplazando la expresión que ya tenemos de d_{n-1}, obtenemos $b_{n-1} = \frac{(f(x_n)-f(x_{n-1})-c_{n-1}h_n^2+\frac{c_{n-1}}{3h_n}h_n^3)}{h_n}$, $\boldsymbol{b_{n-1} = \frac{(f(x_n)-f(x_{n-1}))}{h_n} - \frac{2}{3}c_{n-1}h_n}$

De (5) podemos despejar $d_i \to \boldsymbol{d_i = \frac{(2c_{i+1}-2c_i)}{6h_{i+1}}}$ para $i = 0, \ldots, n-2$

De (3) podemos despejar $b_i \to b_i = \frac{(f(x_{i+1})-f(x_i)-c_i h_{i+1}^2-d_i h_{i+1}^3)}{h_{i+1}}$ para $i = 0, \ldots, n-2$. Reemplazando la expresión de d_i, obtenemos

$$b_i = \frac{(f(x_{i+1}) - f(x_i) - c_i h_{i+1}^2 - \frac{(2c_{i+1}-2c_i)}{6h_{i+1}} h_{i+1}^3)}{h_{i+1}}$$

$$b_i = \frac{(f(x_{i+1}) - f(x_i))}{h_{i+1}} - c_i h_{i+1} - \frac{(2c_{i+1} - 2c_i)}{6} h_{i+1}$$

$$\boldsymbol{b_i = \frac{(f(x_{i+1}) - f(x_i))}{h_{i+1}} - \frac{2}{3} c_i h_{i+1} - \frac{c_{i+1}}{3} h_{i+1}}$$

Finalmente, vamos a usar (4). Por un lado lo hacemos para $i = 0, \ldots, n-3$

$$b_i + 2c_i(x_{i+1} - x_i) + 3d_i(x_{i+1} - x_i)^2 = b_{i+1}$$

Reemplazamos la expresión que tenemos de b_i, b_{i+1} y d_i

$$\frac{(f(x_{i+1}) - f(x_i))}{h_{i+1}} - \frac{2}{3} c_i h_{i+1} - \frac{c_{i+1}}{3} h_{i+1} + 2c_i h_{i+1} + 3h_{i+1}^2 \frac{(2c_{i+1} - 2c_i)}{6h_{i+1}} = \frac{(f(x_{i+2}) - f(x_{i+1}))}{h_{i+2}} - \frac{2}{3} c_{i+1} h_{i+2} - \frac{c_{i+2}}{3} h_{i+2}$$

$$c_i(-\frac{2}{3} h_{i+1} + 2h_{i+1} - h_{i+1}) + c_{i+1}(-\frac{1}{3} h_{i+1} + h_{i+1} + \frac{2}{3} h_{i+2}) + c_{i+2}(\frac{1}{3} h_{i+2}) = \frac{(f(x_{i+2}) - f(x_{i+1}))}{h_{i+2}} - \frac{(f(x_{i+1}) - f(x_i))}{h_{i+1}}$$

$$c_i(\frac{1}{3} h_{i+1}) + c_{i+1}(\frac{2}{3}(h_{i+1} + h_{i+2})) + c_{i+2}(\frac{1}{3} h_{i+2}) = \frac{(f(x_{i+2}) - f(x_{i+1}))}{h_{i+2}} - \frac{(f(x_{i+1}) - f(x_i))}{h_{i+1}}$$

Nos queda el caso $i = n-2$:

$$b_{n-2} + 2c_{n-2}(x_{n-1} - x_{n-2}) + 3d_{n-2}(x_{n-1} - x_{n-2})^2 = b_{n-1}$$

Reemplazamos la expresión que tenemos de b_{n-2},b_{n-1} y d_{n-2}

$$\frac{(f(x_{n-1}) - f(x_{n-2}))}{h_{n-1}} - \frac{2}{3} c_{n-2} h_{n-1} - \frac{c_{n-1}}{3} h_{n-1} + 2c_{n-2} h_{n-1} + 3\frac{(2c_{n-1} - 2c_{n-2})}{6h_{n-1}} h_{n-1}^2 = \frac{(f(x_n) - f(x_{n-1}))}{h_n} - \frac{2}{3} c_{n-1} h_n$$

$$c_{n-2}(-\frac{2}{3} h_{n-1} + 2h_{n-1} - h_{n-1}) + c_{n-1}(-\frac{1}{3} h_{n-1} + h_{n-1} + \frac{2}{3} h_n) = \frac{(f(x_n) - f(x_{n-1}))}{h_n} - \frac{(f(x_{n-1}) - f(x_{n-2}))}{h_{n-1}}$$

$$c_{n-2}(\frac{1}{3} h_{n-1}) + c_{n-1}\frac{2}{3}(h_n + h_{n-1}) = \frac{(f(x_n) - f(x_{n-1}))}{h_n} - \frac{(f(x_{n-1}) - f(x_{n-2}))}{h_{n-1}}$$

En definitiva tenemos las siguientes n ecuaciones que involucran a los n coeficientes $c_0, \ldots, c_{n-1}$

$\boldsymbol{c_0 = 0}$

$\boldsymbol{c_i(\frac{1}{3} h_{i+1}) + c_{i+1}\frac{2}{3}(h_{i+1} + h_{i+2}) + c_{i+2}(\frac{1}{3} h_{i+2}) = \frac{(f(x_{i+2}) - f(x_{i+1}))}{h_{i+2}} - \frac{(f(x_{i+1}) - f(x_i))}{h_{i+1}}}$ para i=0,...,n-3

$\boldsymbol{c_{n-2}(\frac{1}{3} h_{n-1}) + c_{n-1}\frac{2}{3}(h_n + h_{n-1}) = \frac{(f(x_n) - f(x_{n-1}))}{h_n} - \frac{(f(x_{n-1}) - f(x_{n-2}))}{h_{n-1}}}$

La matriz asociada al sistema es

$$\begin{bmatrix} c_0 & c_1 & c_2 & c_3 & \dots & c_i & c_{i+1} & c_{i+2} & \dots & c_{n-2} & c_{n-1} \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \frac{1}{3}h_1 & \frac{2}{3}(h_1+h_2) & \frac{1}{3}h_2 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \frac{1}{3}h_2 & \frac{2}{3}(h_2+h_3) & \frac{1}{3}h_3 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{3}h_{i+1} & \frac{2}{3}(h_{i+1}+h_{i+2}) & \frac{1}{3}h_{i+2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & \frac{1}{3}h_{n-1} & \frac{2}{3}(h_n+h_{n-1}) \end{bmatrix}$$

que resulta ser estrictamente diagonal dominante, por lo cual existe solución del sistema y la solución es única. De esta manera obtenemos en forma única los coeficientes $c_0, \dots, c_{n-1}$. Dado que el resto de los coeficientes se encuentran expresados en función de $c_0, \dots, c_{n-1}$, podemos afirmar que el trazador cúbico existe y es único.

Capítulo 13

Integración

En este capítulo vamos a presentar métodos numéricos para el cálculo de integrales. Este es un problema matemático muy común, que tiene multitud de aplicaciones en diversos campos de la ciencia y de la ingeniería: cálculo de áreas y volúmenes, obtención de ciertas magnitudes físicas a partir de otras (desplazamiento a partir de la velocidad, por ejemplo), etc. La forma clásica de calcular integrales es analítica: se busca una primitiva de la función a integrar, que luego se evalúa en sus dos extremos. Sin embargo, esta solución no siempre es factible. Por ejemplo, existen funciones cuya integral analítica es una expresión complicada de obtener o de evaluar computacionalmente, o incluso algunas para las que no existe una expresión analítica conocida. Peor aún, a veces es necesario integrar funciones de las que se conocen únicamente algunos valores (por ejemplo, muestras obtenidos empíricamente), volviendo imposible la integración analítica.

En estos casos, es útil contar con métodos que permitan aproximar el valor de una integral de manera numérica. En particular, nos vamos a enfocar en los métodos de **cuadratura numérica**, donde la integral de una función se va a aproximar mediante una combinación de valores de una función, evaluada en un conjunto de datos.

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i)$$

Dentro de los métodos de cuadratura numérica, vamos a destacar a aquellos que utilizan el polinomio interpolante, ya que constituyen una buena aproximación de la función a integrar, sus integrales son sencillas de calcular analíticamente y pueden ser evaluadas en forma eficiente, y además permiten calcular una cota para el error cometido en la aproximación.

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \cdot \prod_{i=0}^{n}(x - x_i)$$

$$P(x) = \sum_{k=0}^{n} f(x_k) \cdot L_{nk}(x)$$

donde P_n es el polinomio interpolador de Lagrange en $n+1$ puntos en el intervalo $[a, b]$ y E_n es el error de la aproximación. Por lo tanto, utilizando esta expresión, podemos decir que la integral, en el intervalo $[a, b]$ de la función $f(x)$ nos queda

$$
\begin{aligned}
\int_a^b f(x)dx &= \int_a^b P_n(x)dx + \int_a^b E_n(x)dx \\
&= \int_a^b \left(\sum_{k=0}^{n} f(x_k) \cdot L_{nk}(x)dx \right) + \int_a^b E_n(x)dx \\
&= \sum_{k=0}^{n} \left(f(x_k) \cdot \underbrace{\int_a^b L_{nk}(x)dx}_{=a_i} \right) + \underbrace{\int_a^b E_n(x)dx}_{\text{Error}}
\end{aligned}
$$

Entonces, efectivamente, utilizar a la integral del polinomio interpolante como una aproximación de la función entra dentro del esquema de cuadratura numérica.

Los distintos métodos, que vamos a estudiar, van a variar en cuanto al grado del polinomio interpolante que utilizan para la aproximación.

13.1. Regla de trapecios

Si consideramos una función $f \in \mathcal{C}^2([a,b])$, para la cual queremos calcular

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

Entonces, podemos derivar una fórmula de cuadratura a partir del polinomio interpolador de Lagrange de grado 1, en los puntos $x_0 = a$ y $x_1 = b$. Para ello, escribimos $f(x) = P(x) + E(x)$, donde $P(x)$ es este polinomio y $E(x)$ representa el error de aproximación para cada punto. Así,

$$
P(x) = f(x_0) \cdot \frac{x - x_1}{x_0 - x_1} + f(x_1) \cdot \frac{x - x_0}{x_1 - x_0} \qquad \text{y} \qquad E(x) = \frac{f''(\xi_x)}{2} \cdot (x - x_0) \cdot (x - x_1)
$$

Por un lado,

$$
\begin{aligned}
\int_{x_0}^{x_1} P(x)\,dx &= \int_{x_0}^{x_1} f(x_0) \cdot \frac{x - x_1}{x_0 - x_1} + f(x_1) \cdot \frac{x - x_0}{x_1 - x_0}\,dx \\
&= \left[f(x_0) \cdot \frac{(x - x_1)^2}{2 \cdot (x_0 - x_1)} + f(x_1) \cdot \frac{(x - x_0)^2}{2 \cdot (x_1 - x_0)} \right]\Bigg|_{x_0}^{x_1} \\
&= \frac{x_1 - x_0}{2} \cdot [f(x_1) + f(x_0)]
\end{aligned}
$$

Por otra parte,

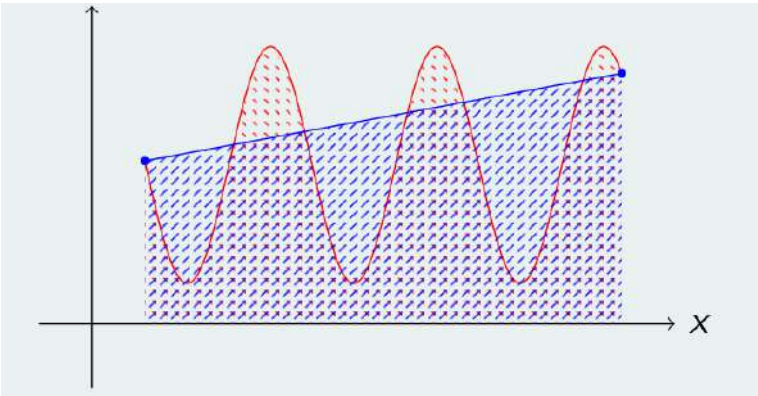
$$
\begin{aligned}
\int_{x_0}^{x_1} E(x)\,dx &= \int_{x_0}^{x_1} \frac{f''(\xi_x)}{2} \cdot (x - x_0) \cdot (x - x_1)\,dx && \text{para algún } \xi_x \in (x_0, x_1) \\
&= \frac{f''(\xi)}{2} \cdot \int_{x_0}^{x_1} (x - x_0) \cdot (x - x_1)\,dx && \text{para algún } \xi \in (x_0, x_1) \\
&= \frac{f''(\xi)}{2} \cdot \left[\frac{x^3}{3} - \frac{x_1 + x_0}{2} \cdot x^2 + x_0 \cdot x_1 \cdot x \right]\Bigg|_{x_0}^{x_1} \\
&= -\frac{(x_1 - x_0)^3}{12} \cdot f''(\xi)
\end{aligned}
$$

Entonces, llamando $h = x_1 - x_0$, tenemos

$$\int_a^b f(x)\,dx = \int_{x_0}^{x_1} P(x)\,dx + \int_{x_0}^{x_1} E(x)\,dx$$
$$= \frac{h}{2}\cdot[f(x_1)+f(x_0)]\underbrace{-\frac{h^3}{12}\cdot f''(\xi)}_{Error}$$

Esta formulación de la integral de f recibe la denominación de **regla del trapecio**. Notemos que el término del error está multiplicado por la segunda derivada de f, y por lo tanto si podemos encontrar una cota de la derivada segunda en el intervalo $[a,b]$, entonces podemos acotar el error cometido.

Veamos gráficamente lo que estamos diciendo. Supongamos que tenemos la función $f(x) = sen(2x) + 2cos(2x) + 3$, tomemos como punto de interpolación a los extremos del intervalo, por lo que nos queda



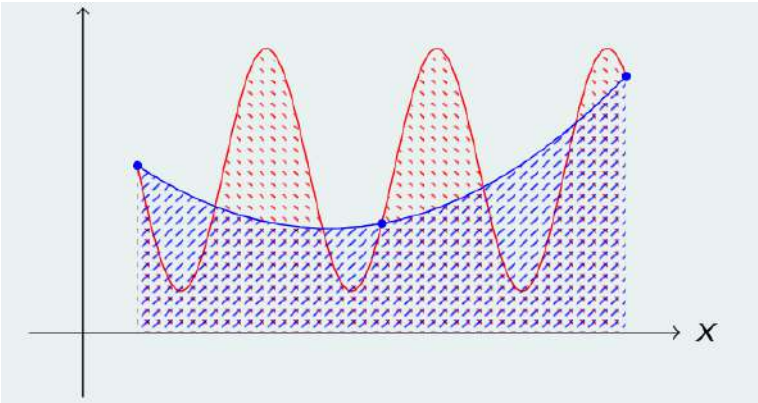
donde el área azul es la aproximación del área roja. En este gráfico podemos ver que a este método se le llama la regla del trapecio justamente porque estamos aproximando el área de la función $f(x)$ mediante el área de un trapecio.

13.2. Regla de Simpson

De la misma manera que hemos considerado el polinomio interpolante de grado 1, podemos considerar el polinomio interpolante de grado 2, que es lo que se conoce como **regla de Simpson**. Nuevamente, los puntos de interpolación son los extremos del intervalo, pero además tomamos como punto adicional al punto del medio, por lo que nos queda $x_0 = a$, $x_1 = \frac{a+b}{2}$, $x_2 = b$. Entonces, la integral de la función se va a aproximar por la integral del polinomio interpolante de grado 2. Si calculamos las integrales correspondientes, nos queda

$$\int_a^b f(x)\,dx \approx \int_a^b P(x)dx = \frac{(x_2-x_0)}{6}\cdot(f(x_0)+4f(x_1)+f(x_2))$$
$$\text{Error } = \int_a^b E(x)dx = -\frac{h^5}{90}f^{(4)}(\mu) \quad \text{con } \mu\in(a,b)$$

Por lo tanto, si conocemos una cota de la derivada cuarta en el intervalo $[a,b]$, podremos tener una cota del error máximo que estamos cometiendo al aproximar el valor de la integral por este método. Gráficamente, el área determinada por ese polinomio va a tener la siguiente pinta



13.3. Regla compuesta

Hasta ahora, hemos considerado con un único intervalo tanto para el caso de la Regla de trapecios como para la Regla de Simpson. Tratando de utilizar esta idea, la propuesta es dividir al intervalo de integración en intervalos más pequeños, y en cada uno de ellos aplicar alguno de los métodos conocidos. De esta manera, la aproximación por regla compuesta consiste en sumar las distintas aproximaciones de las integrales en los distintos intervalos, aprovechando el hecho que

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} \left(\int_{x_i}^{x_{i+1}} f(x)dx \right)$$

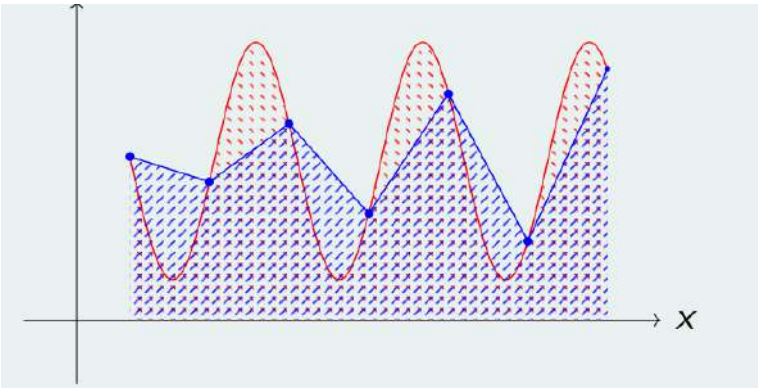
13.3.1. Regla compuesta de trapecios

En el caso de la regla compuesta de trapecios, para aproximar la integral de una función $f \in \mathcal{C}^2[a,b]$, con $x_0, \ldots, x_n \in [a,b]$, dividimos al intervalo $[a,b]$ en n intervalos más pequeños, cada uno de longitud $h = \frac{b-a}{n}$. Luego, si aplicamos la regla de trapecios para cada uno de estos n intervalos, surge la siguiente fórmula

$$\int_a^b f(x)\,dx \approx \frac{h}{2}\left(f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n) \right)$$

$$\text{Error } = -\frac{b-a}{12} \cdot h^2 f''(\mu) \quad \text{con } \mu \in (a,b)$$

Gráficamente nos queda



13.3.2. Regla compuesta de Simpson

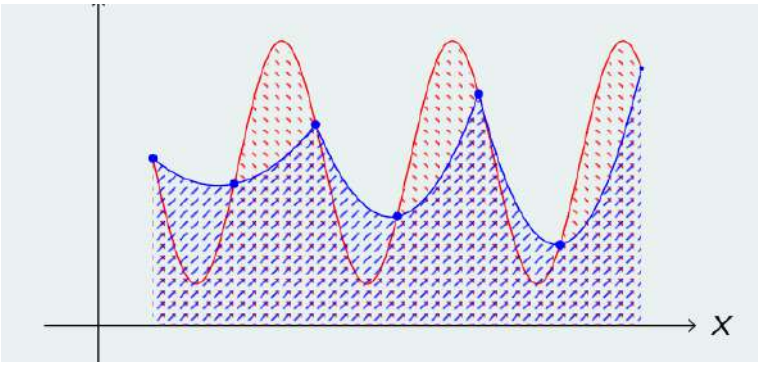
Lo mismo podemos hacer con la regla de Simpson. En este caso, en con la diferencia que la regla de Simpson utiliza tres puntos del intervalo para aproximar, por lo que vamos a necesitar una cantidad par de intervalos.

Entonces, tenemos una función $f \in \mathcal{C}^2[a.b]$, con $x_0, \ldots, x_{2n} \in [a, b]$, $x_0 = a, x_{2n} = b$ y $h = \frac{b-a}{2n}$, y vamos a aplicar la regla de Simpson sobre cada par consecutivo de intervalos. Si hacemos las integrales correspondientes, la aproximación nos queda

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\left[f(x_0) + 2\sum_{j=1}^{(n)-1} f(x_{2j}) + 4\sum_{j=1}^{n} f(x_{2j-1}) + f(x_n)\right]$$

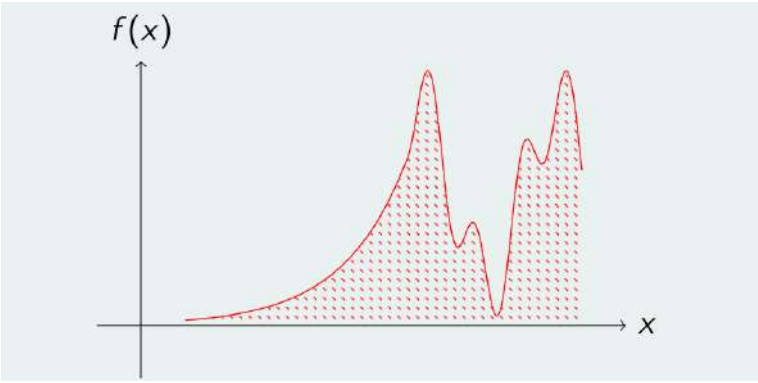
$$\text{Error } = -\frac{b-a}{180}h^4 f^{(4)}(\mu) \quad \text{con } \mu \in (a, b)$$

Gráficamente nos queda



13.4. Métodos adaptativos

La idea de dividir al intervalo en intervalos más chicos parece muy apropiada, y también parece sugerirnos que a mayor cantidad de sub-intervalos consideremos, mejor va a ser la aproximación de la integral que queremos calcular. Sin embargo, aumentar la cantidad de sub-intervalos resulta en un mayor costo computacional. Si pensamos en optimizar la cantidad de intervalos utilizados nos surge la siguiente pregunta: ¿es necesario utilizar el mismo espaciado para todo el dominio? Para responder esta pregunta, supongamos que queremos integrar una función cuyo comportamiento es irregular.



Podemos observar que en algunos sub-intervalos la función tiene una gran variación, lo cual obliga a utilizar una aproximación con una partición fina del sub-intervalo sobre la cual utilizar una regla compuesta. Sin embargo en otros sub-intervalo disjuntos, la función tiene una variación muy pequeña, haciéndola apta para un método de aproximación sin demasiado refinamiento.

En este tipo de situaciones se utilizan **métodos adaptativos**, que analizan en cada sub-intervalo cuál es la precisión de una aproximación de la integral y en caso de no ser suficiente, utilizan una aproximación más fina partiendo en otros sub-intervalos.

En particular, vamos a enfocarnos en el caso de la regla de Simpson compuesta para ver cómo podemos decidir si en alguna parte del intervalo no hace falta particionar más, y en otra parte del intervalo sería conveniente considerar más puntos.

Llamemos $S(x, y)$ a la aproximación de Simpson del intervalo $[x, y]$ para la función f. Supongamos que queremos integrar el intervalo $[a, b]$.

`Paso 1:` Tomamos dos sub-intervalos, cada uno de tamaño $h = \frac{b-a}{2}$, aplicando Simpson, obteniéndose

$$\int_a^b f(x)dx = S(a, b) - \frac{h^5}{90} f^{(4)}(\mu)$$

`Paso 2:` Partimos cada sub-intervalo en otros dos de tamaño $\frac{h}{2}$. Aplicamos la regla compuesta de Simpson en $\left[a, \frac{a+b}{2}\right]$ y $\left[\frac{a+b}{2}, b\right]$, obteniéndose

$$\int_a^b f(x)dx = S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right) - \frac{1}{16}\frac{h^5}{90} f^{(4)}(\tilde{\mu})$$

Supongamos que $\boxed{f^{(4)}(\mu) \approx f^{(4)}(\tilde{\mu})}$, entonces si igualamos las expresiones obtenidas en los pasos 1 y 2:

$$\begin{aligned}
&S(a, b) - \frac{h^5}{90} f^{(4)}(\mu) \approx S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right) - \frac{1}{16}\frac{h^5}{90} f^{(4)}(\mu) \\
&\Leftrightarrow -\frac{15}{16}\frac{h^5}{90} f^{(4)}(\mu) \approx S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right) - S(a, b) \\
&\Leftrightarrow -\frac{1}{16}\frac{h^5}{90} f^{(4)}(\mu) \approx \frac{1}{15}\left(S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right) - S(a, b)\right)
\end{aligned}$$

Por lo tanto, si volvemos a la expresión que obtuvimos de la integral al momento de subdividir los intervalos

$$\int_a^b f(x)dx = S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right) - \frac{1}{16}\frac{h^5}{90} f^{(4)}(\tilde{\mu})$$

$$\Longrightarrow$$

$$\left|\frac{1}{16}\frac{h^5}{90} f^{(4)}(\tilde{\mu})\right| = \left|\int_a^b f(x)dx - (S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right))\right|$$

$$\Longrightarrow$$

$$\left|\frac{1}{16}\frac{h^5}{90} f^{(4)}(\mu)\right| \approx \left|\int_a^b f(x)dx - (S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right))\right|$$

Pero, si volvemos a la expresión que obtuvimos al suponer $f^{(4)}(\mu) \approx f^{(4)}(\tilde{\mu}$

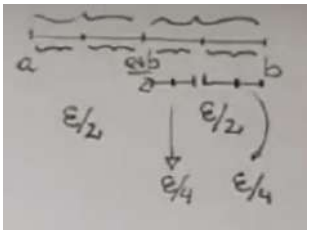
$$\left|\frac{1}{16}\frac{h^5}{90} f^{(4)}(\mu)\right| \approx \left|\frac{1}{15}\left(S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right) - S(a, b)\right)\right|$$

Por lo tanto, el error cometido al momento de subdividir los intervalos es "parecido" a

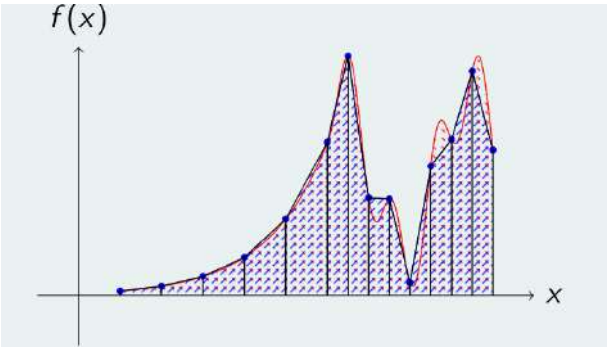
$$\frac{1}{15}\left(S\left(a,\frac{a+b}{2}\right)+S\left(\frac{a+b}{2},b\right)-S(a,b)\right)$$

Luego, si pedimos que esta diferencia sea menor que un ϵ, podemos asumir que a la aproximación vía Simpson compuesta va a tener un error menor que ϵ. ¿Para qué nos puede servir este resultado?

Tenemos un intervalo $[a,b]$, el cual hemos divido en dos sub-intervalos, en los cuales hemos aplicado Simpson. Si el error no es menor que ϵ, vamos a razonar de la siguiente manera. Para cada sub-intervalo, aplicamos nuevamente Simpson, pero vamos a pedir que la diferencia sea menor que $\epsilon/2$. Si lo logramos, entonces quiere decir que hemos conseguido una buena aproximación de la integral. Si en alguno de los dos sub-intervalos el error nos da mayor que $\epsilon/2$, entonces volvemos a dividir a ese sub-intervalo en dos intervalos más chicos, y volvemos a aplicar este procedimiento de manera recursiva, pero esta vez con $\epsilon/4$.



Entonces, lo que estamos logrando con esta metodología es particionar en sub-intervalos más chicos en las zonas que sea necesaria. En las zonas que ya hemos obtenido el error que buscábamos, no seguimos particionando, por lo que no pagamos el costo adicional de realizar evaluaciones innecesarias. En los lugares donde se necesite refinar para obtener el error buscado, realizamos la partición. Si aún no conseguimos el error buscado, volvemos a refinar el intervalo.



De esta manera, terminamos adaptando la medida del sub-intervalo a medidas más chicas en las zonas que sean necesarias, y en otras zonas donde la aproximación ya sea buena, no particionamos los sub-intervalos.

Capítulo 14

Ceros de funciones

Este capítulo está dedicado al tema **ceros de funciones**. Como lo hemos hecho habitualmente, vamos a comenzar definiendo cuál es el problema matemático que queremos resolver, para luego proponer métodos numéricos que nos permitan encontrar la solución del problema.

Dada una función $f : \mathbb{R} \to \mathbb{R}$, buscamos identificar a aquellos valores x^* tal que $f(x^*) = 0$. Dependiendo de las características de la función, este puede ser un problema sencillo o un problema más complicado. Las raíces de una ecuación no lineal $f(x) = 0$, en general, no tienen una fórmula cerrada. Incluso cuando las tienen, la expresión a menudo es tan complicada que no es resulta práctica . Por lo tanto, para resolver un sistema no lineal de ecuaciones estamos obligados a utilizar métodos de aproximación.

Estos métodos suelen estar basados en la idea de una aproximación sucesiva. Estos métodos son *iterativos*, es decir, comienzan con uno o más puntos iniciales, y generan una sucesión $x_0, x_1, \ldots,$ que converja a la raíz de f. Algunos métodos requieren de conocer un intervalo $[a, b]$ que contenga a la raíz, mientras que otros necesitan de una posición inicial cercana a la raíz (con la ventaja de que convergen más rápido). Por lo tanto, suele ser conveniente comenzar con un método más fuerte (en el sentido de condiciones de convergencia), para luego cambiar a uno que converja más rápido. Hay relativamente poco conocimiento acerca de cómo atacar este problema si no se conoce a priori ninguna información sobre la ubicación de las raíces, por lo que es de esperar que necesitemos condiciones relativamente exigentes para la convergencia de los distintos métodos.

Todos los algoritmos que vamos a ver se encuadran dentro de un esquema general, donde se genera una sucesión $\{x_k\}_0^\infty$ que, bajo ciertas condiciones, va a converger, y el límite de esta sucesión es una raíz (o cero) de la función

$$\lim_{k\to\infty} x_k = x^* \quad \text{con } f(x^*) = 0$$

Los diferentes métodos que vamos a estudiar van a diferir en cómo generan esta sucesión y en las condiciones de convergencia.

Como vamos a tener varios métodos para aproximarnos a la raíz, necesitamos de criterios que nos permitan identificar cuál de ellos nos conviene utilizar, dado un caso particular. Entre los criterios que existen para comparar, vamos a utilizar

- El costo computacional.
- Las condiciones de convergencia. Es decir, qué propiedades debe cumplir la función f para que la sucesión generada converja.
- El orden de convergencia de la sucesión generada. El orden de convergencia tiene que ver con la velocidad con la cual la sucesión se acerca a su límite. Hay diferentes maneras para definir esta velocidad.

14.1. Orden de convergencia

Una de las posibles maneras de definir este concepto, viene dada a partir de un límite que nos relaciona el error del paso $k+1$ con el error del paso k elevado a una cierta potencia p. Es decir

Definición 14.1.1. Sea $\{x_k\}_{k\in\mathbb{N}}$ una sucesión tal que $\lim_{k\to\infty} x_k = x^*$. Decimos que $\{x_k\}_{k\in\mathbb{N}}$ tiene **orden de convergencia** $p \in \mathbb{R}_{>0}$ si

$$\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{(|x_k - x^*|)^p} = c > 0.$$

Por lo tanto, si este límite existe, se cumple que $|x_{k+1} - x^*| \approx c \cdot (|x_k - x^*|)^p$ para algún $c > 0$. Es decir, cuanto mayor sea este número p, mayor será la velocidad con la que la sucesión se acerca a su límite.

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

Otra manera de definir el orden de convergencia es a partir de comparar la sucesión que tenemos con una sucesión que tienda a 0. Consideremos la sucesión $\{x_k\}_{k\in\mathbb{N}}$ que converge a x^*, y la sucesión $\{\beta_k\}_{k\in\mathbb{N}}$ que converge a 0.

Definición 14.1.2. Si existe algún $k_0 \in \mathbb{N}$ tal que, para todo $k \geq k_0$, el error de la sucesión original esté acotado por una constante multiplicada por el error del paso k-ésimo de la sucesión $\{\beta_k\}$, es decir

$$|\alpha_k - \alpha| \leq |\beta_k|$$

entonces, vamos a poder afirmar que la sucesión $\{x_k\}$ se acerca al límite al menos tan rápidamente como la sucesión $\{\beta_k\}$ se acerca a 0.

En este caso, el orden de convergencia viene dado por la comparación con otra sucesión. Entonces, si conocemos de antemano que la sucesión$\{\beta_k\}_{k\in\mathbb{N}}$ tiene orden de convergencia p, entonces podremos afirmar que $\{\alpha_k\}_{k\in\mathbb{N}}$ tiene orden de convergencia mayor o igual a p.

¿Qué significa que $\{x_n\}_n$ converja a x^* con orden p? Llamemos $e_n = x_n - x^*$. Según la definición de antes, esto significa que $\lim\limits_{n\to\infty} \dfrac{|e_{n+1}|}{|e_n|^p} = c$ para cierta constante $c \neq 0$.

- Que c no sea infinito significa que $|e_n|^p$ no tiende a 0 más rápido de lo que lo hace $|e_{n+1}|$.
- Que c sea no nulo, significa que $|e_{n+1}|$ tampoco lo hace más rápido que $|e_n|^p$.

Por lo tanto, $|e_{n+1}|$ y $|e_n|^p$ convergen a 0 con la misma velocidad.

A su vez, es posible interpretar el significado de esta velocidad en términos prácticos. Dada la equivalencia asintótica de $|e_{n+1}|$ y $|e_n|^p$, vamos a suponer que para n suficientemente grande $|e_{n+1}| \approx |e_n|^p$.

Luego, supongamos que hasta el término n-ésimo llevamos calculados k dígitos decimales del valor x^*, es decir $|e_n| \approx 10^{-k}$. Entonces, $|e_{n+1}| \approx (10^{-k})^p = 10^{-kp}$, por lo que podemos concluir que, por cada iteración, la cantidad de decimales calculados se multiplica por p. A modo de ejemplo, que una sucesión converja de forma cuadrática significa, a nivel práctico, que la cantidad de dígitos decimales calculados se duplica a cada paso.

Con este concepto en mente, continuemos con los métodos y algoritmos para encontrar los ceros de funciones.

14.2. Método de la bisección

El primer método que vamos a ver es el **método de bisección**, y está basado en el teorema de Bolzano.

Teorema 14.2.1. *Consideremos una función continua $f : [a, b] \to \mathbb{R}$ tal que en los extremos tienen distinto signo ($f(a) \cdot f(b) < 0$). Entonces, por el teorema de Bolzano, la función tiene algún cero dentro del intervalo, es decir, existe $x^* \in (a, b)$ tal que $f(x^*) = 0$. Luego, bajo estas condiciones, podemos definir un proceso para poder encontrar un cero (o raíz) de la función.*

Comenzamos el proceso tomando el punto medio del intervalo $c = \dfrac{a+b}{2}$, dividiendo el intervalo (a, b) en dos mitades. Si $f(c) = 0$, entonces hemos encontrado el cero de la función que estábamos buscando. Si no es el caso, ese punto $f(c)$ va a diferir en signo con alguno de los dos extremos del intervalo, es decir

$$\text{o bien } f(a)f(c) < 0 \qquad \text{o bien} \qquad f(b)f(c) < 0$$

Por lo tanto, vamos a poder asegurar, por el teorema de Bolzano, o bien que hay una raíz en el sub-intervalo (a, c), o bien que hay una raíz en el sub-intervalo (c, b), dependiendo con cuál de los dos extremos la función difiere en signo con $f(c)$.

Luego, hemos partido del intervalo (a, b), donde sabíamos que había al menos una raíz, a un sub-intervalo, ya sea o bien (a, c), o bien (c, b), cuya medida es exactamente la mitad del intervalo anterior, donde podemos asegurar que existe una raíz. Luego, podemos definir una sucesión

$$c_k = \frac{1}{2} \cdot (a_k + b_k)$$
$$(a_{k+1}, b_{k+1}) = \begin{cases} (c_k, b_k) \text{ si } f(c_k)f(b_k) < 0 \\ (a_k, c_k) \text{ si } f(c_k)f(a_k) < 0 \end{cases}$$

Si aplicamos este procedimiento de forma iterativa, se obtiene un intervalo de menor longitud que contiene a la raíz buscada, lo cual permite aproximarla con precisión arbitraria.

Demostración. Si consideramos la sucesión que está definida por los puntos intermedios de cada uno de los intervalos $\{c_k\}_0^\infty$, entonces vamos a poder demostrar que esta sucesión converge a una raíz de la función. Por un lado, por la forma que fuimos construyendo la sucesión, sabemos que partimos de un intervalo inicial (a_0, b_0), y vamos construyendo intervalos donde siempre

$$a_0 \leq a_1 \leq \cdots \leq a_k \leq c_k \leq b_k \leq \cdots \leq b_1 \leq b_0$$

Por lo tanto, la sucesión $\{a_k\}$ es una sucesión monótona creciente acotada, y la sucesión $\{b_k\}$ es una sucesión monótona decreciente acotada. Por lo tanto, existen los límites

$$\lim_{k\to\infty} a_k = \alpha_1$$
$$\lim_{k\to\infty} b_k = \alpha_2$$

Por otro lado, si consideramos la sucesión $\{b_k - a_k\}_0^\infty$, donde $b_{k+1} - a_{k+1} = \dfrac{b_k - a_k}{2}$, entonces sabemos que su límite es

$$\lim_{k\to\infty} b_k - a_k = \lim_{k\to\infty} \frac{b_0 - a_0}{2^k} = 0$$

Por lo tanto, podemos deducir que $\alpha_1 = \alpha_2$. Luego, considerando que la sucesión $\{a_k\}$ monótona creciente converge, la sucesión $\{b_k\}$ monótona decreciente converge, y ambas tienen el mismo límite, y como la sucesión $\{c_k\}$ cumple que $a_k \leq c_k \leq b_k \forall k$, entonces podemos deducir que el límite de la sucesión $\{c_k\}$ existe, y es

$$\lim_{k\to\infty} c_k = \alpha_1 = \alpha_2$$

Por lo tanto, la sucesión $\{c_k\}$ es convergente. Lo que nos falta ver es que esta converge a una raíz (cero) de la función. Por la forma en la que fuimos construyendo los intervalos, sabemos que la evaluación de los extremos del intervalo k-ésimo difieren en signo, es decir $f(a_k)f(b_k) < 0$, por lo tanto si consideramos el siguiente límite

$$\lim_{k\to\infty} f(a_k)f(b_k) \leq 0$$

pero, por otro lado, sabemos que

$$\begin{aligned}\lim_{k\to\infty} f(a_k)f(b_k) &= \lim_{k\to\infty} \overbrace{f(\alpha_1)^2}^{\geq 0} \quad \text{al ser } f \text{ continua} \\ &= \lim_{k\to\infty} f(c_k)^2\end{aligned}$$

Por lo tanto, $\lim_{k\to\infty} f(a_k)f(b_k) = \lim_{k\to\infty} f(c_k)^2 = 0$, por lo que $\{c_k\} \to 0$. Luego, podemos concluir que la sucesión $\{c_k\}$ generada por el método de bisección, efectivamente, es una sucesión convergente, y además converge a una raíz de f.

■

¿Qué podemos decir respecto al orden de convergencia? La sucesión que estamos generando está determinada por los puntos intermedios de los intervalos que nos vamos construyendo. Entonces, el error del paso k-ésimo

$$|c_k - \alpha| \leq b_k - a_k = \frac{b_0 - a_0}{2^k}$$

siendo α el límite de $\{c_k\}$. Luego,

$$|c_k - \alpha| \leq (b_0 - a_0) \cdot \underbrace{\frac{1}{2^k}}_{\{\beta_k\} \xrightarrow[k\to\infty]{} 0}$$

Es decir, si recordamos la segunda definición que dimos para el orden de convergencia, notamos que nos encontramos en una situación donde $|x_k - x^*| \leq M|\beta_k|$, por lo que podemos afirmar que el método de la bisección converge a x^* al menos tan rápidamente como la sucesión $\left\{\frac{1}{2^k}\right\}_{k\in\mathbb{N}}$ converge a 0.

Si aplicamos la primera definición que dimos para el orden de convergencia, se puede demostrar que el orden de esta última tiene es de orden de convergencia lineal, lo cual nos permite afirmar que el método de
la bisección se aproxima, al menos, linealmente a una raíz de f.

El pseudocódigo del algoritmo es el siguiente (debe ser completado con algún criterio de parada

lim).

Algoritmo de la bisección

Entrada: $a, b \in \mathbb{R}$, y $f : [a, b] \to \mathbb{R}$ tal que $f(a) \cdot f(b) < 0$
Salida: una aproximación de una raíz $x^* \in (a, b)$ de f

```
a_0 ← a
b_0 ← b
for k = 0, ..., lim do
    c_k ← (a_k + b_k)/2
    if f(c_k) = 0 then
        return c_k
    if f(c_k) · f(a_k) < 0 then
        a_{k+1} ← a_k
        b_{k+1} ← c_k
    else
        a_{k+1} ← c_k
        b_{k+1} ← b_k
return c_k
```

En resumen, el método de bisección genera una sucesión convergente a una raíz de la función, y que el orden de convergencia es lineal.

Ventajas

- Para cada a_k y b_k, nos alcanza con conocer el signo de $f(a_k)$ y el de $f(b_k)$ con lo cual podría no ser necesario evaluar la función f en esos puntos. Esto es conveniente en contextos en los cuales la evaluación es una operación costosa y es posible conocer el signo por alguna vía sencilla.
- Tenemos una cota para el error absoluto.
- Es fácil encontrar puntos iniciales a_0 y b_0 factibles.
- Funciona bien para obtener aproximaciones iniciales.

Desventajas

- La convergencia del método de bisección (lineal) es lenta.

14.3. Criterios de parada

Notemos que, cuando vimos el algoritmo, hemos determinado que este realiza una cantidad lim de iteraciones. En realidad, el criterio de parada podría ser otro, pero es necesario contar con un criterio que permita decidir cuándo la aproximación ya es lo suficientemente buena. Estos se conocen como **criterios de parada**. Algunos de los criterios más comúnmente utilizados son:

(I) Establecer una cantidad fija de iteraciones. Es el criterio más sencillo, pero es insensible a las características del método usado y no permite decidir *a priori* la precisión de los resultados.

(II) Fijar un valor $\varepsilon > 0$ y parar cuando $|x_{k+1} - x_k| < \varepsilon$, es decir, cuando el ritmo de convergencia sea lo suficientemente lento. Si bien es un criterio más sofisticado, puede dar resultados erróneos. Por ejemplo, considerando la sucesión

$$x_k = \sum_{i=0}^{k} \frac{1}{k} = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$$

se tiene que $|x_{k+1} - x_k| = \frac{1}{k+1} \xrightarrow{k\to\infty} 0$. Luego, para cualquier valor de ε, el criterio terminará y

arrojará un resultado supuestamente cercano al límite de la sucesión, que en realidad es divergente.

(III) Fijar un valor $\varepsilon > 0$ y parar cuando $\dfrac{|x_{k+1} - x_k|}{|x_k|} < \varepsilon$. La idea, en este caso, es testear el error relativo de la aproximación. Sufre de problemas similares al criterio anterior, pero es un buen candidato a utilizar en la ausencia de información adicional.

(IV) Fijar un valor $\varepsilon > 0$ y parar cuando $f(x_k) < \varepsilon$. También puede dar falsos positivos, ya que f puede tomar valores arbitrariamente cercanos a 0 sin que esto indique la cercanía de una raíz.

(V) Fijar un valor $\varepsilon > 0$ y parar cuando $|f(x_{k+1}) - f(x_k)| < \varepsilon$.

(VI) Fijar un valor $\varepsilon > 0$ y parar cuando $\dfrac{|f(x_{k+1}) - f(x_k)|}{|f(x_k)|} < \varepsilon$.

Como puede verse, todos estos criterios tienen casos en los que arrojan resultados falsos. Por este motivo, la elección del criterio de parada debe hacerse teniendo en cuenta las características del problema a resolver. Además, es posible emplearlos de forma combinada; por ejemplo, es común establecer un límite fijo de iteraciones incluso aunque se use un criterio de parada distinto, y así evitar la posibilidad que el programa no termine.

En el caso del método de bisección, una de las ventajas que podemos aprovechar es que siempre tenemos encerrada una raíz en el intervalo del paso k-ésimo, es decir $|c_k - \alpha \leq |b_k - a_k$. Por lo tanto, podemos tomar como criterio de parada la condición de que $|b_k - a_k| < \epsilon$, lo cual nos asegura que el valor obtenido se encuentra a una distancia menor que ϵ de una raíz de la función.

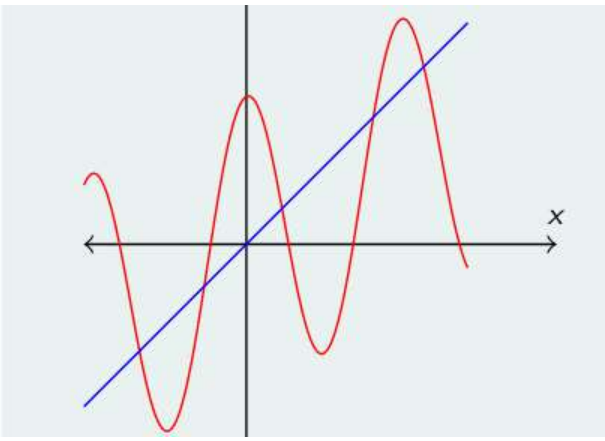
14.4. Puntos Fijos

A veces, al momento de resolver un problema, es conveniente transformarlo en otro, con la esperanza de que para ese otro problema tengamos una metodología para resolverlo. La idea es que ambos problemas van a ser equivalentes en el sentido de que si encontramos solución para uno, habremos encontrado solución para el otro, y viceversa. En el caso de encontrar ceros de funciones, este problema está muy relacionado con los **puntos fijos** de una función. Primero vamos a comenzar por definir qué es un punto fijo.

Definición 14.4.1. Dada una función $g : [a, b] \to \mathbb{R}$, se llama **punto fijo** de g a un valor $p \in [a, b]$ tal que

$$g(p) = p$$

Gráficamente, los puntos fijos de la función $g(x)$ no son otra cosa que las intersecciones de $g(x)$ con la función $f(x) = x$



¿Por qué es de interés determinar los puntos fijos de una función? Es posible establecer una relación entre los puntos fijos de una función y los ceros (o raíces) de otra. Si definimos una función $f(x) = g(x) - x$,

entonces un punto fijo de g es un cero de f, y un cero de f es un punto fijo de g. Entonces, al haber una correspondencia unívoca entre los puntos fijos de una función y los ceros de otra, entonces podemos utilizar cualquier algoritmos que nos permita resolver alguno de los problemas, con el objetivo de resolver el otro.

En este caso, vamos a tratar de ver si podemos utilizar algoritmos que resuelven el problema de puntos fijos de una función g, con el objetivo de encontrar los ceros de una función f. Así como existe el teorema de Bolzano para asegurar la existencia de un cero dentro de un intervalo, tenemos un resultado similar para los puntos fijos.

Teorema 14.4.1. *Dada una función $g : [a, b] \to [a, b]$ continua, entonces vamos a poder afirmar que la función g tiene un punto fijo dentro del intervalo $[a, b]$. Además, si esta función es derivable dentro del intervalo (a, b) y la derivada está acotada por una constante M tal que $g'(x) \leq M < 1$, entonces el punto fijo es único.*

(I) Si g es continua, entonces g tiene al menos un punto fijo en $[a, b]$.

Demostración.

- Si $g(a) = a$ o $g(b) = b$, entonces a o b es un punto fijo.
- En caso contrario, consideremos la función

$$h(x) = g(x) - x \qquad \text{continua en } [a, b]$$

$$h(a) = \underbrace{g(a)}_{\in[a,b]} - a > 0$$

$$h(b) = \underbrace{g(b)}_{\in[a,b]} - b < 0$$

Luego, aplicando el teorema de Bolzano, existe $c \in (a, b)$ tal que $h(c) = g(c) - c = 0$, y por lo tanto c es punto fijo de g

$$\boxed{g(c) = c}$$

■

(II) Si g es derivable en (a, b) y existe alguna constante M tal que $|g'(x)| \leq c < 1$, entonces el punto fijo es único.

Demostración. Supongamos que existen dos puntos fijos distintos c_1 y $c_2 \in [a, b]$. Por el Teorema del Valor Medio, existe $\xi \in (a, b)$ tal que

$$\begin{aligned} |g'(\xi)| &= \left| \frac{g(c_1) - g(c_2)}{c_1 - c_2} \right| \\ &= \left| \frac{\cancel{c_1 - c_2}}{\cancel{c_1 - c_2}} \right| \\ &= 1 \end{aligned}$$

Es decir, hay algún punto $\xi \in (a, b)$ tal que $g'(\xi) = 1$, lo cual nos contradice la hipótesis inicial que nos decía que $g'(x)| \leq M < 1 \forall x \in (a, b)$. Luego, hemos llegado a un absurdo que provino de suponer que existían dos puntos fijos.

En conclusión, la función g, bajo estas condiciones, tiene un punto fijo y, además, este es único. ■

Notemos que estas condiciones son condiciones **suficientes**, pero no necesarias para la existencia y unicidad de puntos fijos.

14.5. Algoritmo de Punto fijo

Entonces, ya hemos caracterizado condiciones suficientes para la existencia y unicidad de un punto fijo, por lo que ahora vamos a proponer un algoritmo para encontrar un punto fijo.

Teorema 14.5.1. *Sea $g : [a, b] \to [a, b]$ continua y derivable en (a, b), y sea una constante M tal que $|g'(x)| \leq M < 1$ para todo $x \in (a, b)$. Sea $\{x_k\}_{k\in\mathbb{N}}$ una sucesión*

$$x_{k+1} = g(x_k)$$

, con $x_0 \in [a, b]$. Entonces $\{x_k\}_k$ converge al único punto fijo de g.

Esta es una metodología que, en caso de converger, converge a $x^* = g(x^*)$, pues

$$\lim_{k\to\infty} x_{k+1} = \lim_{k\to\infty} g(x_k)$$

$$x^* = g(x^*)$$

Es decir, en caso de que converja, converge a un punto fijo de g. Entonces, lo que necesitamos determinar si la sucesión $\{x_k\}$ converge o no. Veamos que esta sucesión converge.

Demostración.

(I) En primer lugar, vamos a poder afirmar que la sucesión que estamos generando pertenece al intervalo $[a, b]$. Esto se debe a que estamos partiendo de un $x_0 \in [a, b]$, luego $x_1 = g(x_0)$ está bien definida. Además, como la función g está definida en $[a, b] \to [a, b]$, entonces $x_1 \in [a, b]$. Si aplicamos este razonamiento de forma inductiva, entonces podemos concluir que $x_k \in [a, b]$ para todo $k = 0, 1, \ldots$.

(II) Por otro lado, dado que se cumplen las hipótesis que nos permitían afirmar la existencia de un único punto fijo, podemos llamar a $x^* \in [a, b]$ al punto fijo de g. Luego, si consideramos

$$\begin{aligned} |x_k - x^*| &= |g(x_{k-1}) - g(x^*)| && \text{(por definición)} \\ &= |g'(\epsilon_{k-1})| \cdot |x_{k-1} - x^*| && \epsilon \in (a, b), \text{ por el TVM} \\ \text{como } g'(x) < M \text{ por hipótesis} &\Longrightarrow \\ |x_k - x^*| &\leq M||x_{k-1} - x^*| \\ &\vdots \\ &\leq M^k|x_0 - x^*| \end{aligned}$$

Si tomamos límite cuando $k \to \infty$, entonces

$$0 \leq \lim_{k\to\infty} |x_k - x^*| \leq \lim_{k\to\infty} M^k|x_0 - x^*| = 0$$

Por lo tanto, podemos concluir que la sucesión generada converge al punto fijo de la función g.

■

Entonces, tenemos aquí un algoritmo que, bajo ciertas condiciones, encontrar el punto fijo de la función g.

Cotas del Error

También se puede demostrar que

- $|x_k - x^*| \leq M^k \cdot \text{máx}\,(x_0 - a, b - x_0)$. Para ver esto, recordemos que

$$|x_k - x^*| \leq M^k \cdot |x_0 - x^*|$$

Por otro lado, supongamos $x_0 \leq x^*$

$$x_0 \leq x^* \leq b$$

$$0 \leq x^* - x_0 \leq b - x_0$$

Ahora supongamos que $x_0 \geq x^*$, entonces

$$a \leq x^* \leq x_0$$

$$0 \leq x^* - a \leq x_0 - a$$

Por lo tanto, $|x_0 - x^*| \leq \text{máx}\,(b - x_0, x_0 - a)$. Luego,

$$\boxed{|x_k - x^*| \leq M^k \,\text{máx}\,(b - x_0, x_0 - a)}$$

- $|x_k - x^*| \leq \frac{M^k}{1-M} \cdot |x_1 - x_0|$. Para ver esto, consideremos la diferencia entre dos iteradas sucesivas

$$\begin{aligned}
|x_{k+1} - x_k| &= |g(x_k) - g(x_{k-1}| \\
&= |x_k - x_{k-1}| \cdot g'(\epsilon) \qquad \text{para algún } \epsilon \in (a, b) \\
&\leq M|x_k - x_{k-1}| \\
&\vdots \\
&\leq M^k|x_1 - x_0|
\end{aligned}$$

Si ahora consideramos la diferencia entre dos iteradas x_i, x_k, con $i > k$, entonces

$$\begin{aligned}
|x_i - x_k| &= |x_i - x_k + x_{i-1} - x_{i-1}| \\
&\leq |x_i - x_{i-1}| + |x_{i-1} - x_k| \\
&\vdots \\
&\leq \sum_{j=k}^{i-1} |x_{j+1} - x_j|
\end{aligned}$$

Por lo tanto, si consideramos ahora el límite de $i \to \infty$, es decir $x_i = x^*$ nos queda

$$\begin{aligned}
|x^* - x_k| &\leq M^k \cdot |x^* - x_0| \\
&\leq M^k \cdot \sum_{i=0}^{\infty} |x_{i+1} - x_i| \\
&\leq M^k \cdot \sum_{i=0}^{\infty} M^i \cdot |x_1 - x_0| \\
&\leq M^k|x_1 - x_0| \cdot \sum_{i=0}^{\infty} M^i \\
&\leq M^k|x_1 - x_0| \cdot \frac{1}{1-M}
\end{aligned}$$

Por lo tanto,

$$\boxed{|x^* - x_k| \leq \frac{M^k}{1-M} \cdot |x_1 - x_0|}$$

Estas cotas nos permiten determinar la cantidad de iteraciones necesarias para obtener un error menor a un ϵ cualquiera.

Interpretación geométrica

Ahora veamos gráficamente qué es lo que hace el algoritmo. Vamos a considerar está función $f(x)$ (en rojo), y en azul la función $y = x$, con lo cual el punto fijo de la función está determinado por la intersección entre ambas funciones.

Aplicamos el algoritmo a partir de un x_0 cualquiera, y calculamos $g(x_0)$. Como el valor de $x = g(x_0)$, pero está definido sobre el eje x, por lo que tenemos que trasladar este valor que observamos sobre el eje y al eje x. Luego, continuamos aplicando este procedimiento de manera iterativa.

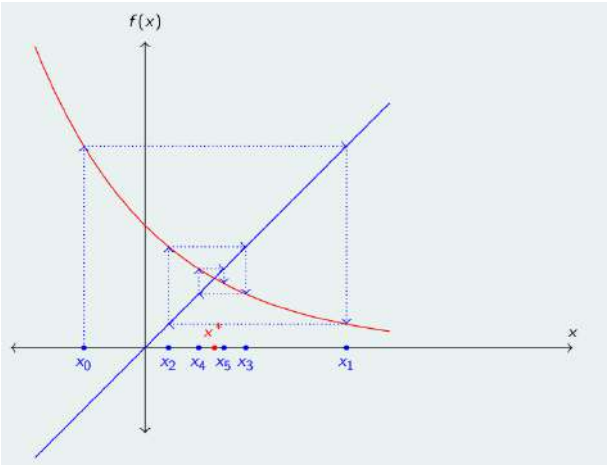
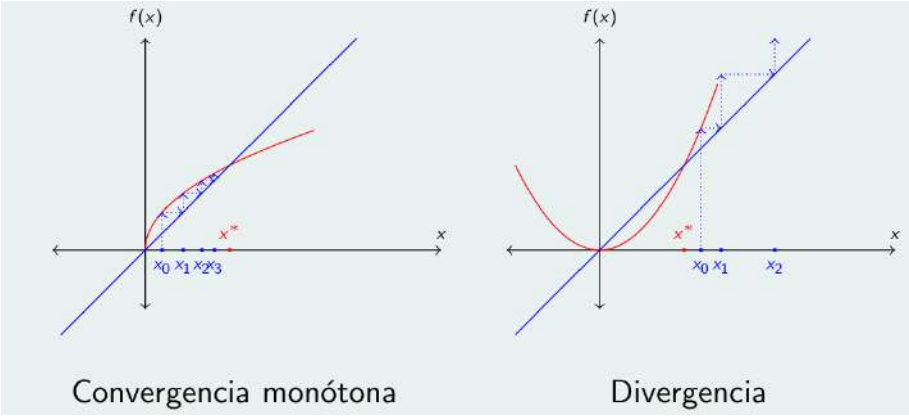


Figura 14.1: Convergencia alterante

En este caso, lo que tenemos es una convergencia alternante, en el sentido de que en una iterada $x_k \leq x^*$, pero en la próxima iterada $x_{k+1} \geq x^*$.

También puede darse una convergencia monótona o puede que no converja (si no se cumplen alguna de las condiciones de convergencia).



Orden de convergencia

Hasta ahora, tenemos un algoritmo que, bajo ciertas condiciones, es convergente, y una cota para el error en el paso k-ésimo. Ahora, vamos a analizar qué es lo que ocurre con el orden de convergencia, cuando la sucesión converge. Para determinar el orden de convergencia, vamos a utilizar una propiedad que nos dice

Teorema 14.5.2. *Sea $g \in C^r[a,b]$, $x^* \in (a,b)$ punto fijo de g tal que*

$$g'(x^*) = g''(x^*) = \cdots = g^{(r-1)}(x^*) = 0,\ g^{(r)}(x^*) \neq 0$$

Entonces, dado $x_0 \in [a,b]$ si la sucesión definida por $x_{k+1} = g(x_k)$ converge a x^*, entonces el orden de convergencia es r.

Demostración. Como $g \in C^r([a,b])$, podemos considerar el polinomio de Taylor de la función g de orden $r-1$, alrededor del punto fijo x^*, y sea ξ_x algún valor en x y x^* tal que

$$\begin{aligned} g(x) &= g(x^*) + g'(x^*)(x-x^*) + \cdots + \frac{g^{(r-1)}(x^*)}{(r-1)!}(x-x^*)^{r-1} + \frac{g^{(r)}(\xi_x)}{r!}(x-x^*)^r \\ &= g(x^*) + \frac{g^{(r)}(\xi_x)}{r!}(x-x^*)^r \\ g(x) - g(x^*) &= \frac{g^{(r)}(\xi_x)(x-x^*)^r}{r!} \end{aligned}$$

Si consideramos el error en el paso $k+1$, obtenemos

$$\begin{aligned} |x_{k+1} - x^*| &= |g(x_k) - g(x^*)| \\ &= \frac{|g^{(r)}(\xi_k)| \cdot |(x_k - x^*)^r|}{r!} \end{aligned}$$

Si ahora tomamos límite para $k \to \infty$, entonces

$$\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|(x_k - x^*)^r|} = \lim_{k\to\infty} \frac{|g^{(r)}(\xi_k)|}{r!}$$

Ahora bien, ξ_k es un punto intermedio entre x_k y x^*, con lo cual si x_k está convergiendo a x^*, entonces ξ_k también está convergiendo a x^*. Por lo tanto,

$$\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|(x_k - x^*)^r|} = \frac{|g^{(r)}(x^*)|}{r!} \neq 0$$

Ahora, si consideramos la definición primera definición del orden de convergencia, hemos llegado a que el límite nos da una constante no nula, y por tanto el orden de convergencia de $\{x_k\}_{k\in\mathbb{N}}$ es r. ■

Por lo tanto, el orden de convergencia del método de punto fijo está relacionado con la cantidad de derivadas que se anulan en el punto fijo.

14.6. Método de Newton

Ahora, recordemos cómo es que habíamos llegado a este problema de punto fijo. Llegamos bajo la idea de que queríamos buscar los ceros de una función f, donde habíamos planteado que encontrar el cero de una función f era equivalente a encontrar un punto fijo de una función $g(x) = f(x) + x$.

Sin embargo, esta no es la única manera de relacionar una función, a la cual estamos buscando un cero, con una función a la cual estamos buscando un punto fijo. Hay otra manera de encontrar funciones. Por ejemplo, consideremos la función $f(x) = 4x^3 - 10x^2 + 5x - 17$ a la cual queremos encontrar sus ceros. Podríamos definir una función

$$\begin{aligned} g_1(x) &= \frac{17}{4x^2 - 10x + 5} \\ g_2(x) &= \frac{-4x^3 + 10x^2 + 17}{5} \\ g_3(x) &= \frac{4x^3 + 5x - 17}{10x} \end{aligned}$$

Todas estas funciones tienen la propiedad de que sus puntos fijos son los ceros de f, y los ceros de f son puntos fijos de g. Nuevamente, no hay una única manera de relacionar una función con puntos fijos y una función a la que le queramos encontrar sus ceros. Ahora bien, sabemos que el orden de convergencia del algoritmo de punto fijo depende de la cantidad de derivada que se anulen. Entonces, si tenemos varias candidatas a las cuales queremos encontrarle los puntos fijos, entonces sería conveniente elegir a aquella función con la mayor cantidad de derivadas nulas. Por lo tanto, es de interés obtener una estructura general que nos sirva para identificar a aquella con un alto orden de convergencia.

El **método de Newton** es un algoritmo para la búsqueda de raíces de funciones, se basa en plantear una iteración de punto fijo que, bajo ciertas hipótesis, converge con orden al menos cuadrático.

Con este objetivo en mente, vamos a definir una función

$$g(x) = x - h(x)f(x)$$

donde la función h tiene que cumplir que $h(x^*) \neq 0$, con x^* raíz de f (y punto fijo de g).

Supongamos que vamos a aplicar el algoritmo de punto fijo a esta función g, y queremos que el algoritmo resulte con convergencia cuadrática. Para ello, necesitamos pedir que $g'(x^*) = 0$. Si ahora calculamos la derivada primera nos queda

$$\begin{aligned} g'(x) &= 1 - h'(x)f(x) - h(x)f'(x) \\ g'(x^*) &= 1 - h'(x^*)\underbrace{f(x^*)}_{=0} - h(x^*)f'(x^*) = 0 \\ 1 - h(x^*)f'(x^*) &= 0 \\ h(x^*) &= \frac{1}{f'(x^*)} \end{aligned}$$

Entonces, necesitamos que

1. f derivable.
2. $f'(x^*) \neq 0$.
3. $h(x^*) \neq 0$.
4. $h(x^*) = \frac{1}{f'(x^*)}$

El problema que tenemos con estas condiciones es que la h que nos queremos construir tiene que cumplir condiciones que están definidas en función de x^*, que no conocemos. Entonces, la candidata natural para tomar como h es

$$h(x) = \frac{1}{f'(x)}$$

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

donde g es la función candidata a que. en caso de que el algoritmo de punto fijo converja, converja con convergencia cuadrática. Entonces, el algoritmo nos queda definido como

Método de Newton

Entrada: $\mathbf{x_0} \in [\mathbf{a}, \mathbf{b}]$
Salida: $\mathbf{x_k}$ tal que $x_k \approx x^*$.

```
for k = 1, ... lim do
    x_k ← x_k − f(x_k)/f'(x_k)
return x_k
```

Notemos que lo que hicimos fue buscar condiciones para las cuales, en caso de que el algoritmo converja, tengamos un orden de convergencia cuadrático, pero nadie nos asegura que este converja. Luego, nos falta analizar qué condiciones debería cumplir la f para que este algoritmo converja.

Si lo que necesitamos es ver condiciones bajo las cuales el algoritmo converja, vamos a hacer uso de una propiedad que nos dice

Teorema 14.6.1. *Sean $f(x) \in \mathcal{C}^2[a,b]$ y $x^* \in [a,b]$ tal que $f(x^*) = 0$ y $f'(x) \neq 0$, entonces existe $\delta > 0$ tal que la sucesión*

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

converge a x^* si $x_0 \in [x^* - \delta, x^* + \delta]$.

Es decir, esta propiedad nos asegura la convergencia de la sucesión de Newton, dentro de un entorno del x_0, de x^*. Esto quiere decir que el valor inicial x_0 no puede ser cualquiera, tiene que estar lo suficientemente cercano a la raíz que estamos buscando para poder asegurar que el algoritmo va a converger.

El problema que queda aquí abierto es qué significa el δ, es decir, qué tan cercano tiene que estar x_0 de la raíz. Eso no lo vamos a poder definir en forma exacta, solo podemos hablar de la existencia de un δ tal que la sucesión converja. Una solución común a este problema, cuando no se cuenta con dicha aproximación, es obtenerla en primer lugar ejecutando algunas iteraciones del método de la bisección. Más allá de este problema, vamos a demostrar que esta propiedad es cierta.

Demostración. Para eso vamos a recordar que el algoritmo de Newton surge de haber considerado la función para punto fijo

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

Por lo tanto, si el algoritmo de Newton nos es otra cosa que aplicar el algoritmo de punto fijo sobre esta función g, nos basta con demostrar que podemos construirnos un intervalo en el cual la función esté definida sobre ese intervalo y su imagen también caiga dentro del intervalo, y además que la derivada de la función g esté acotada por una constante en este intervalo.

Es decir, queremos determinar un intervalo tal que

$$\begin{cases} g : [x^* - \delta, x^* + \delta] \to [x^* - \delta, x^* + \delta] \\ |g'(x)| \leq M < 1 \quad \forall x \in (x^* - \delta, x^* + \delta) \end{cases}$$

al ser estas las condiciones suficientes para asegurar la convergencia del algoritmo de punto fijo.

Entonces, bajo las hipótesis de la función f, veamos que podemos construirnos este intervalo. Por hipótesis sabemos que

1. $f'(x^*) \neq 0$.
2. La derivada primera es continua al ser $f \in \mathcal{C}^2[a,b]$.

Entonces, podemos afirmar que existe un δ_1 tal que para todo $x \in [x^* - \delta_1, x^* + \delta_1]$ se cumple que $f'(x) \neq 0$. Además, la función g es una función que va a estar bien definida en este intervalo.

Ahora, calculemos la derivada primera de g.

$$\begin{aligned} g'(x) &= \left(x - \frac{f(x)}{f'(x)} \right) \\ &= 1 - \left[\frac{f'(x)f'(x) - f(x)f''(x)}{f'(x)^2} \right] \\ &= \frac{f(x)f''(x)}{f'(x)^2} \end{aligned}$$

Como estamos trabajando dentro del intervalo $[x^* - \delta_1 x^* + \delta_1]$, donde la $f'(x) \neq 0$, entonces, en este

intervalo, $g'(x)$ está bien definida, y no solo está bien definida, sino que además

$$g'(x^*) = \frac{\overbrace{f(x^*)}^{=0} f''(x^*)}{f'(x^*)^2} = 0$$

Entonces, la derivada primera de g está bien definida en el intervalo $[x^* - \delta_1, x^* + \delta_1]$, y además $g'(x^*) = 0$. Además, g' es una función continua al ser f' y f'' continuas, y $f'(x) \neq 0$. Luego, al ser g' continua y $g'(x^*) = 0$, existe un intervalo tal que para todo $x \in [x^* - \delta_2, x^* + \delta_2]$

$$|g'(x)| \leq M < 1$$

Entonces, teníamos un intervalo $[x^* - \delta_1 x^* + \delta_1]$ donde la función g estaba bien definida, y además tenemos un intervalo $[x^* - \delta_2, x^* + \delta_2]$ donde la función g' también existe, está bien definida, y está acotada por una constante más chica que 1.

Por lo tanto, si nos quedamos con el intervalo más chico entre ambos, eso nos define un δ tal que para todo $x \in [x^* - \delta, x^* + \delta]$ se cumple que

- g está bien definida.
- $|g'(x)| \leq M < 1$ para todo x.

Entonces, estamos bastante cerca para tener las condiciones suficientes para asegura la convergencia. Nos está faltando que la función g esté definida en $[x^* - \delta, x^* + \delta] \to [x^* - \delta, x^* + \delta]$. Veamos que esto es verdad.

Tomemos un $x \in [x^* - \delta, x^* + \delta]$, y queremos ver que $g(x) \in [x^* - \delta, x^* + \delta]$, es decir que $|g(x) - x^*| \leq \delta$. Veamos que esto es cierto.

$$\begin{aligned}
|g(x) - x^*| &= |g(x) - g(x^*)| && (x^* \text{ es punto fijo}) \\
&= |g'(\xi)||x - x^*| && (\text{por teorema de Valor Medio}) \\
&\leq M|x - x^*| \\
&\leq M \cdot \delta \\
&\leq \delta
\end{aligned}$$

Por lo tanto, podemos concluir que $g : [x^* - \delta, x^* + \delta] \to [x^* - \delta, x^* + \delta]$. Luego, estamos cumpliendo todas las condiciones suficientes para que el algoritmo de punto fijo converja. ■

14.6.1. Interpretación geométrica

Al algoritmo de Newton se le puede dar una interpretación geométrica, que puede observarse en la Figura 14.2. Partiendo de un punto x_0, se considera la recta tangente a f en el punto $(x_0, f(x_0))$. El punto x_1 se define como la intersección entre esta recta y el eje x, es decir $f(x) = 0$. El proceso se repite con cada iteración, arrojando cada vez una aproximación más cercana a la raíz x^* buscada.

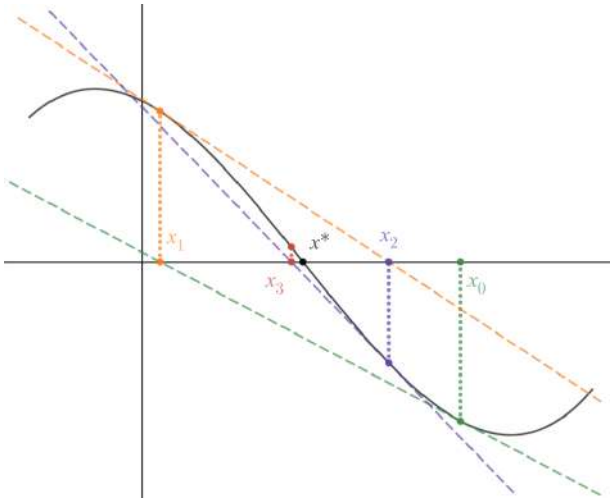


Figura 14.2: Interpretación geométrica del método de Newton.

Si consideramos el polinomio de Taylor de grado 1, alrededor de un punto $\bar{x}$:

$$f(x) + f(\bar{x}) + f'(\bar{x})(x - \bar{x}) + \frac{f''(\xi(x))}{2} \cdot (x - \bar{x})^2$$

Si pensamos que la raíz de f se encuentra cercana a $\bar{x}$, entonces podríamos despreciar el término del error, por lo que

$$0 \approx f(\bar{x}) + f'(\bar{x})(x^* - \bar{x})$$
$$x^* \approx \bar{x} - \frac{f(\bar{x})}{f'(\bar{x})}$$

Entonces, de alguna manera estamos diciendo que x^* se parece bastante a $\bar{x} - \dfrac{f(\bar{x})}{f'(\bar{x})}$, que es parecida a la fórmula que estábamos considerando para el método de Newton. De aquí podemos plantear la sucesión

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

que es justamente la sucesión de Newton.

Por lo tanto, lo que estaría haciendo el método de Newton es, tomando el polinomio de Taylor de orden 1, encontrar dónde ese polinomio se anula como una aproximación del cero de la función f.

14.6.2. Casos particulares

Hay casos particulares para los cuales se puede asegurar que el algoritmo de Newton va a converger, desde cualquier punto inicial.

Teorema 14.6.2. *Sea $f(x) \in \mathcal{C}^2[a,b]$ creciente y convexa ($f''(x) \geq 0$). Entonces, si existe $x^* \in [a,b]$ tal que $f(x^*) = 0$, la raíz es única, y el algoritmo de Newton converge desde cualquier $x_0 \in [a,b]$ inicial.*

Demostración. Supongamos que existe x_1^* y x_2^* raíces de $f(x)$, con $x_1^* < x_2^*$. Como $f(x)$ es estrictamente creciente, entonces $0 = f(x_1^*) < f(x_2^*) = 0$, lo cual nos lleva a una contradicción. Por lo tanto, si f tiene una raíz, esta es única.

Veamos ahora la convergencia del método de Newton. Como $f(x)$ es estrictamente creciente y convexa, podemos afirmar que $f'(x) > 0$ y $f''(x) \leq 0$. Si consideramos el polinomio de Taylor de grado 1 alrededor de x_k para $k \leq 0$, nos queda

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{f''(\xi_x)}{2} \cdot (x - x_k)^2$$

Si evaluamos a este polinomio en x^* (raíz única de f), obtenemos

$$0 = f(x_k) + f'(x_k)(x^* - x_k) + \frac{f''(\xi_{x^*})}{2} \cdot (x^* - x_k)^2$$

$$\text{Si dividimos por } f'(x_k) > 0 \Longrightarrow$$

$$0 = \frac{f(x_k)}{f'(x_k)} + (x^* - x_k) + \frac{f''(\xi_{x^*})}{2f'(x_k)} \cdot (x^* - x_k)^2$$

$$\text{Como } x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \text{ entonces}$$

$$0 = \cancel{x_k} - x_{k+1} + (x^* \cancel{- x_k}) + \frac{f''(\xi_{x^*})}{2f'(x_k)} \cdot (x^* - x_k)^2$$

$$x_{k+1} - x^* = \underbrace{\frac{f''(\xi_{x^*})}{2f'(x_k)} \cdot (x^* - x_k)^2}_{\geq 0}$$

al ser $f'(x) > 0$ y $f''(x) \geq 0$ para todo x. Entonces, podemos asegurar que $x_{k+1} - x^* \geq 0$.

Luego, podemos deducir que o bien $x_{k+1} = x^*$, o bien $x_{k+1} > x^*$. Si $x_{k+1} = x^*$, la sucesión se estabiliza en x^*, y por lo tanto converge a la raíz. Ahora, asumamos que $x_{k+1} > x^*$. Como f es estrictamente creciente, entonces $f(x_{k+1}) > f(x^*) = 0$. Además, por hipótesis, $f'(x) > 0$, luego

$$x_{k+1} = x_k - \underbrace{\frac{f(x_k)}{f'(x_k)}}_{> 0}$$

$$\Longrightarrow$$

$$x_{k+1} < x_k$$

Es decir, la sucesión $\{x_k\}_{k=1}^{\infty}$ es estrictamente decreciente, y además está acotada inferiormente por x^*. Por lo tanto, por el teorema de Weierstrass, $\{x_k\}$ es una sucesión convergente .

Por otro lado, consideremos el límite $\lim_{k\to\infty} x_k = p$, que sabemos que existe al ser $\{x_k\}$ una sucesión convergente. Entonces,

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

$$\lim_{k\to\infty} x_{k+1} = \lim_{k\to\infty} x_k - \frac{f(x_k)}{f'(x_k)}$$

$$p = p - \frac{f(p)}{f'(p)}$$

$$0 = \frac{f(p)}{f'(p)}$$

y como $f'(x) > 0$ para todo x, en particular $f'(p) > 0$, por lo que, necesariamente, $f(p) = 0$, por lo que $p = x^*$. Luego, podemos concluir que la sucesión de Newton $\{x_k\}$ converge a la raíz única de f para todo $x_0 \in [a, b]$ inicial. ■

Teorema 14.6.3. *Sea $f \in \mathcal{C}^2[1, b]$, $f'(x) \neq 0$, y $f''(x)$ no cambia de signo en el intervalo $[a, b]$, con $f(a) \cdot f(b) < 0$. Si*

$$\left|\frac{f(a)}{f'(a)}\right| < b - a, \qquad \left|\frac{f(b)}{f'(b)}\right| < b - a,$$

Entonces el método de Newton converge para cualquier punto inicial $x_0 \in [a, b]$.

En conclusión, el método de Newton se trata de un método que, bajo las condiciones adecuadas, converge a una buena velocidad. Sin embargo, tiene dos principales desventajas:

- Una de ellas ya se mencionó anteriormente, y es la necesidad de conocer de antemano una aproximación relativamente buena de la raíz buscada para poder asegurar la convergencia. Sin embargo, en la práctica, esta restricción rara vez es de importancia.

- La otra desventaja es la necesidad de computar, en cada paso, el valor de la derivada de f. Por ejemplo, si $f(x)$ se conoce solo implícitamente (digamos, como la solución de alguna ecuación diferencial en la que x es un parámetro en los datos iniciales), puede ser poco práctico evaluar $f'(x_k)$ en cada iteración, y en algunos casos hasta imposible.

  Sin embargo, para funciones suficientemente simples, que se dan explícitamente, esto puede no ofrecer ninguna dificultad seria. Esto es especialmente cierto para los polinomios cuyas derivadas se evalúan fácilmente mediante división sintética. Además, calcular $f'(x)$ solo es un medio para obtener
  $$-\frac{f(x_k)}{f'(x_k)}$$
  Por lo tanto, resulta innecesario computar $f'(x_k)$ con un error relativo mucho menor que $f(x_k)$, y como el error relativo de $f(x_k)$ aumenta a medida que x_k se aproxima a la raíz, podríamos utilizar $f'(x_i)$ para $k = i + 1$, e incluso para $k = i + 2$. Es decir, actualizar la derivada de vez en cuando, en vez de tener que calcularla en cada iterada.

14.7. Método de la secante

La principal crítica que se le hace al algoritmo de Newton es la necesidad de contar con la derivada primera, lo cual puede no ser posible (si la función no es derivable), o porque puede ser demasiado costoso calcularla en cada paso.

La alternativa del algoritmo de Newton es el **método de la secante**. Este método puede se construye a partir del método de Newton, aproximando a la derivada $f'(x_k)$ por el coeficiente $\dfrac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$. Esta idea se basa en el hecho de que la derivada de una función f en un punto no es otra cosa que

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

luego, es razonable utilizar la siguiente aproximación

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

La idea es que la pendiente de la secante entre dos puntos x_k, x_{k-1} cercanos es una buena aproximación para la pendiente de la tangente en x_k.

Entonces, la sucesión del método de la secante nos queda

$$\begin{aligned} x_{k+1} &= x_k - \frac{f(x_k)}{\frac{f(x_k)-f(x_{k-1})}{x_k - x_{k-1}}} \\ &= x_k - f(x_k) \cdot \frac{(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})} \end{aligned}$$

Notemos que necesitamos dos puntos iniciales x_0, x_1, pero solo se evalúa **una** función en cada paso.

Además, notemos que si reescribimos esta ecuación en la forma

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

entonces se podrían generar errores numéricos asociados a la cancelación cuando $x_k \approx x_{k-1}$ y $f(x_k)f(x_{k-1} > 0$. Por lo tanto, **no** se debería de reescribir a la sucesión de la secante de esta forma.

<u>Nota</u>: El error dominante en $\dfrac{f(x_k) \cdot (x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$ viene dado por el error de $f(x_k)$. El error en el resto de los factores es de menor importancia.

La Figura 14.3 ilustra la interpretación geométrica de este método.

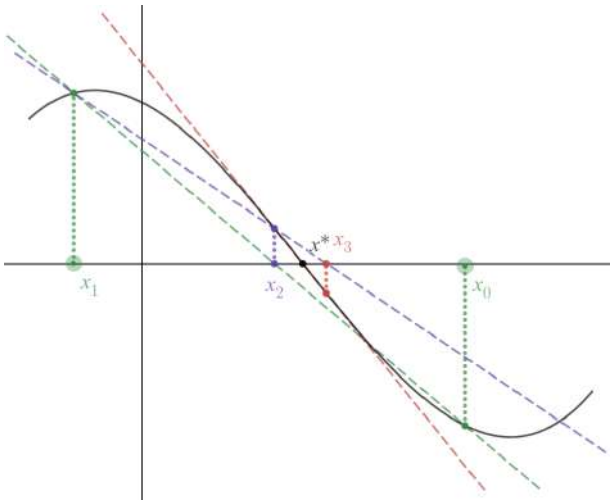


Figura 14.3: Interpretación geométrica del método de la secante.

Podemos observar que x_{k+1} queda determinado como al abscisa de la intersección entre la secante de $(x_{k-1}, f(x_{k-1})$ y $(x_k, f(x_k)$ y el eje x.

La elección entre el método de la secante y el método de Newton va a depender de la cantidad de trabajo necesario para computar $f'(x)$. Supongamos que la cantidad de trabajo necesaria para computar $f'(x)$ es α veces la cantidad de trabajo necesario para computar $f(x)$. Entonces, un análisis asintótico puede ser utilizado para motivar la siguiente regla: si $\alpha > 0{,}44$, entonces utilizar el método de la secante; en caso contrario, utilizar el método de Newton.

El algoritmo de la secante no necesariamente converge, se exigen una serie de condiciones sobre la función f para que converja, y además perdemos la convergencia cuadrática del método de Newton. En particular, el método de la secante converge, para puntos x_0, x_1 suficientemente cercanos, si $f'(x) \neq 0$ y f tiene derivada segunda continua.

Sin embargo, se puede demostrar que tiene una convergencia **super-lineal** $(1 < p < 2)$, más específicamente, tiene un orden de convergencia $\varphi = \frac{1+\sqrt{5}}{2} \approx 1{,}6$. Además, a diferencia del método de Newton, solo se evalúa una función en cada iterada (en lugar de las dos de Newton $f(x_k), f'(x_k)$). Por lo tanto, si consideramos tener dos evaluaciones en cada paso, el método de la secante tendría un orden de convergencia $\approx (1{,}6)^2 > 2{,}5$

Método de la secante

Entrada: $\mathbf{a}, \mathbf{b} \in [\mathbf{a}, \mathbf{b}]$
Salida: $\mathbf{x_k}$ tal que $x_k \approx x^*$.

```
x_0 ← a
x_1 ← b
for k = 1, ... lim do
    x_{k+1} ← x_k − (f(x_k)·(x_k − x_{k−1})) / (f(x_k) − f(x_{k−1}))

return x_k
```

14.8. Método *regula falsi*

Por último, el algoritmo de *regla falsa* o *regula falsi* o *False Position*, es una variante del método de la bisección, que incorpora la idea principal del método de la secante. Al igual que en el método de la bisección, se comienza con dos puntos iniciales donde f tiene distinto signo, y en cada paso se divide el intervalo en dos y se pasa a trabajar con la parte en cuyos extremos f tiene diferente signo. Sin embargo,

en lugar de dividir al intervalo por su punto medio, se utiliza la intersección entre la recta secante y el eje x, es decir donde se anula la recta secante, para luego aplicar la regla de bisección para determinar con qué sub-intervalo nos quedamos.

$$c_k = a_k - \frac{f(a_k) \cdot (a_k - b_k)}{f(a_k) - f(b_k)}$$

$$(a_{k+1}, b_{k+1}) = \begin{cases} (c_k, b_k) \text{ si } f(c_k)f(b_k) < 0 \\ (a_k, c_k) \text{ si } f(c_k)f(a_k) < 0 \end{cases}$$

Notemos que se asegura que

La ventaja que tiene respecto del método de la secante es que siempre converge para funciones continuas (que era una de las buenas propiedades del método de bisección), pero tiene un orden de convergencia lineal. Además, a diferencia del método de bisección, como los sub-intervalos no son iguales, no se garantiza reducirlos en cada paso a la mitad. Por lo tanto, hay casos en los que converge muy lentamente.

Por ejemplo, si consideramos aplicar el método de regla falsa en un intervalo inicial $[-1, 1]$ para encontrar la raíz $r = 0$ de $f(x) = x^3 - 2x^2 + \frac{3}{2}x$, dados los puntos iniciales $x_0 = -1, x_1 = 1$, obtenemos

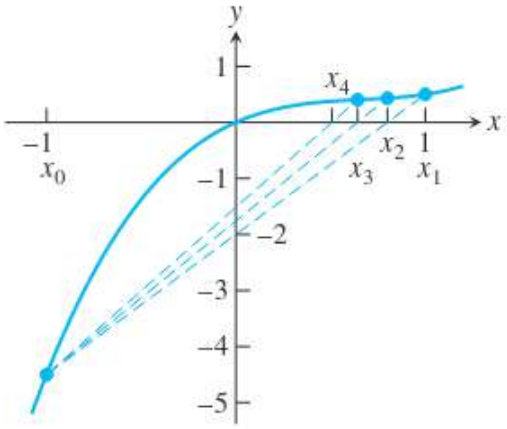


Figura 14.4: Convergencia lenta para regla falsa

Como $f(-1)f(4/5) < 0$, el nuevo sub-intervalo es $[x_0, x_2] = [-1, 0{,}8]$. Notemos que el tamaño del intervalo ser redujo mucho menos que por un factor de $1/2$. Esto también ocurre para las siguientes iteradas como se puede ver en la figura 14.4.

El método de regla falsa es un buen método para empezar a aproximarnos a la raíz, pero no debe ser utilizado cerca de la misma. Por lo tanto, suele ser usado como parte de un método "híbrido" que tenga buenas propiedades de convergencia cuando estamos cerca de la raíz, como por ejemplo el método de **Steffensen**:

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

el cual tiene un orden de convergencia cuadrático, o el algoritmo de **Illinois**, que tiene un orden de

convergencia cúbico.

Algoritmo de regla falsa

Entrada: $a, b \in \mathbb{R}$, y $f : [a, b] \to \mathbb{R}$ tal que $f(a) \cdot f(b) < 0$
Salida: una aproximación de una raíz $x^* \in (a, b)$ de f

```
a_0 ← a
b_0 ← b
for k = 0, ..., lim do
    c_k ← a_k − (f(a_k) · (a_k − b_k)) / (f(a_k) − f(b_k))
    if f(c_k) = 0 then
        return c_k
    if f(c_k) · f(a_k) < 0 then
        a_{k+1} ← a_k
        b_{k+1} ← c_k
    else
        a_{k+1} ← c_k
        b_{k+1} ← b_k
return c_k
```

Capítulo 15

Preguntas de Final

Factorización LU

- ¿Toda matriz tiene LU? ¿De qué depende?
- ¿Conoces alguna condición si y sólo si para que tenga LU (aparte de la de eliminación Gaussiana)?

Número de Condición y Normas

- ¿Qué es el número de condición? Intuición y definición.

Factorización de Cholesky(?)

- Si una matriz es simétrica definida positiva (s.d.p.) ¿cómo te conviene resolver un sistema lineal?

Factorización QR

- ¿Toda matriz tiene factorización QR?
- ¿Es única?
- ¿Bajo qué condiciones lo es?

Autovalores

- Dar alguna condición para afirmar que tenemos una base de autovectores.
- Algún método para encontrar el autovalor más grande de una matriz.
- ¿Qué condiciones son necesarias para que el método de la potencia converja?

Descomposición en valores singulares

- ¿Qué son los valores singulares?

Métodos iterativos

- ¿Cuándo convergen? Condiciones necesarias y suficientes.
- ¿Qué es el radio espectral?

Cuadrados mínimos

- ¿Por qué está bueno cuadrados mínimos lineales en relación a cuadrados mínimos no lineales?
- Interpretación geométrica.
- Tengo un problema de cuadrados mínimos, siempre tiene solución?
- ¿Qué métodos conoces para resolver cuadrados mínimos?

Interpolación

- Quiero resolver un sistema con splines cúbicos con frontera sujeta. ¿Siempre tengo solución? ¿Por qué?
- Quiero dar un polinomio interpolador. ¿Siempre existe? ¿Qué algoritmos conoces para calcularlo?

Ceros de funciones

- Método de Newton, ¿qué condiciones necesito para converger?
- ¿Cuál es la idea intuitiva del método de Newton?
- Método de Newton, ¿alguna crítica?
- Método de bisección, ¿alguna crítica?
- Comparar Newton con secante.
- Orden de convergencia del método de la secante.

Aritmética Finita

- ¿Qué cosas debería tener en cuenta o errores que puedo tener?
- ¿Qué es el epsilon de la máquina?

Bibliografía

[1] Ake Bjorck. *Numerical Methods*. Dover Publications, 2003.
[2] Richard Burden. *Análisis Numérico*. Cengage Learning Editores, 2017.
[3] Richard Burden. *Applied Linear Algebra*. Springer, 2018.
[4] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.
[5] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms-Society for Industrial Mathematics*. SIAM, 2009.
[6] Carl D. Meyer. *Matrix Analysis & Applied Linear Algebra*. SIAM, 2001.
[7] Timothy Sauer. *Numerical Analysis*. Pearson, 2017.