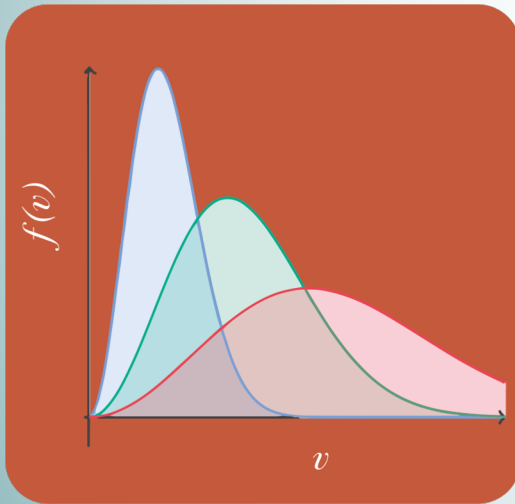


Métodos Matemáticos en la Ingeniería

BLOQUE II. MÉTODOS NUMÉRICOS

TEMA 8. RESOLUCIÓN NUMÉRICA DE SISTEMAS LINEALES



Vera Egorova

Sara Pérez Carabaza

DEPARTAMENTO DE MATEMÁTICA APLICADA
Y CIENCIAS DE LA COMPUTACIÓN

Este material se publica bajo la siguiente licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)



1 Introducción

- Antecedentes matemáticos
- Sistemas lineales
- Condicionamiento

2 Métodos directos

- Eliminación gaussiana
- Factorización LU

3 Métodos iterativos

- Método de Jacobi
- Método de Gauss-Seidel

4 Autovalores y autovectores



Hoy
vamos a hablar
de las matrices...

Expectativa



Realidad

Vamos a ver el desarrollo por Gauss de un sistema:

$$[A^{(1)}|b^{(1)}] = \begin{pmatrix} 6 & -2 & 2 & 4 & 12 \\ 12 & -8 & 6 & -10 & 34 \\ 3 & -13 & 9 & 3 & 27 \\ -6 & 4 & 1 & -18 & -38 \end{pmatrix}$$

Haciendo pivotaje parcial y operando con la primera fila (que ahora es la que era antes la segunda):

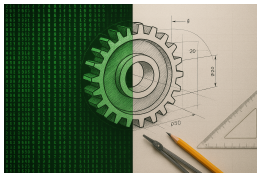
$$[A^{(2)}|b^{(2)}] = \begin{pmatrix} 12.0000 & -8.0000 & 6.0000 & 10.0000 & 34.0000 \\ 0.0000 & 2.0000 & -1.0000 & -1.0000 & -5.0000 \\ 0.0000 & -11.0000 & 7.5000 & 0.5000 & 18.5000 \\ 0.0000 & 0.0000 & 4.0000 & -13.0000 & -21.0000 \end{pmatrix}$$

Pivotando de nuevo la segunda por la tercera fila:

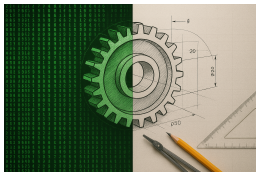
$$[A^{(3)}|b^{(3)}] = \begin{pmatrix} 12.0000 & -8.0000 & 6.0000 & 10.0000 & 34.0000 \\ 0.0000 & -11.0000 & 7.5000 & 0.5000 & 18.5000 \\ 0.0000 & 0.0000 & 0.3636 & -0.9091 & -1.6364 \\ 0.0000 & 0.0000 & 4.0000 & -13.0000 & -21.0000 \end{pmatrix}$$

$$[A^{(4)}|b^{(4)}] = \begin{pmatrix} 12.0000 & -8.0000 & 6.0000 & 10.0000 & 34.0000 \\ 0.0000 & -11.0000 & 7.5000 & 0.5000 & 18.5000 \\ 0.0000 & 0.0000 & 4.0000 & -13.0000 & -21.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.2727 & 0.2727 \end{pmatrix}$$

Y ya tenemos la matriz del sistema triangulizada.

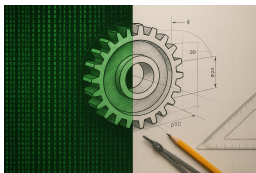


¿Para qué necesitamos matrices?



¿Para qué necesitamos matrices?

- Representación de datos estructurados (Data Science)
- Procesamiento de señales e imágenes
- Optimización y modelado de sistemas
- Otros métodos numéricos (ecuaciones diferenciales, interpolación)



¿Para qué necesitamos matrices?

- Representación de datos estructurados (Data Science)
- Procesamiento de señales e imágenes
- Optimización y modelado de sistemas
- Otros métodos numéricos (ecuaciones diferenciales, interpolación)

Este tema considera 2 tipos de problemas:

- Resolución de sistemas lineales $Ax = b$
- Cálculo de valores y vectores propios $Ap = \lambda p$, con $\lambda \in \mathbb{R}$ o $\mathbb{C}$

Matriz

Matriz es un conjunto bidimensional de números:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}_{m \times n}$$

- Los elementos individuales de una matriz $A_{m \times n}$, se denotan por a_{ij} , donde i indica la fila, y j - la columna.
- Si el número de filas y el de columnas coinciden, $m = n$, se dice que la matriz es **cuadrada** del orden n .
- La **matriz traspuesta** de una matriz A se obtiene al intercambiar sus filas por sus columnas. Se denota como A^T o A^t .
- Dada una matriz $M = (a_{ij})_{m \times n}$, y una matriz $N = (b_{hk})_{n \times p}$, llamamos **matriz producto** de M por N y lo representaremos por MN a la matriz $m \times p$ cuyo elemento c_{rs} es

$$c_{rs} = \sum_{k=1}^n a_{rk} b_{ks}.$$

- **Matriz nula:** Es una matriz de cualquier dimensión en la que todos sus elementos son cero. Se denota por $\mathbf{0}_{m \times n}$.
- **Matriz diagonal:** Es una matriz cuadrada en la que todos los elementos fuera de la diagonal principal son cero. Es decir, si $A = (a_{ij})$ es una matriz $n \times n$, es diagonal si $a_{ij} = 0$ para todo $i \neq j$.
- **Matriz identidad:** Es una matriz diagonal en la que todos los elementos de la diagonal principal son iguales a 1. Se denota por I o I_n , donde n es la dimensión de la matriz. Sus elementos se definen mediante la delta de Kronecker: $(I_n)_{ij} = \delta_{ij}$.
- **Matriz inversa:** Dada una matriz cuadrada A , su inversa, denotada como A^{-1} , es la única matriz que cumple la siguiente condición:

$$AA^{-1} = A^{-1}A = I_n$$

Una matriz A se dice

- **simétrica**, si $A = A^t$.
- **ortogonal**, si $A^{-1} = A^t$.
- **singular**, si $\det A = 0$ (no existe A^{-1}).

Ejercicios:

1. ¿Toda matriz diagonal es también una matriz simétrica? ¿Por qué?
2. Si una matriz A es ortogonal, ¿puede ser también una matriz singular?
3. Resuelve Ejercicio (4.1).

Norma matricial

Una **norma matricial** en el espacio $\mathbb{R}^{n \times m}$ se define como una función $\|\cdot\| : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$ que cumple las siguientes propiedades:

- $\|A\| \geq 0$ para todo $A \in \mathbb{R}^{n \times m}$.
- $\|A\| = 0$ si, y sólo si, A es la matriz nula.
- $\|\alpha A\| = |\alpha| \|A\|$ para todo $\alpha \in \mathbb{R}$ y $A \in \mathbb{R}^{n \times m}$.
- $\|A + B\| \leq \|A\| + \|B\|$ para todo $A, B \in \mathbb{R}^{n \times m}$.

Si el conjunto de matrices contiene sólo matrices cuadradas, se suele añadir la propiedad:

$$\|A \cdot B\| \leq \|A\| \cdot \|B\| \text{ para todo } A, B \in \mathbb{R}^{n \times n}$$

- **Norma 1:** $\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|$
- **Norma infinito:** $\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|$

Ejercicios (4.2) y (4.3)

Ejercicio (4.2)

Ejercicio (4.2)

Para cada una de las siguientes matrices, calcula $\|\cdot\|_1$ y $\|\cdot\|_\infty$:

$$A = \begin{pmatrix} 1 & -7 & 3 \\ -5 & 2 & -1 \end{pmatrix}, \quad B = \begin{pmatrix} 3 & 0 & -2 \\ 1 & 5 & 4 \\ -6 & 1 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 8 \\ -1 \\ 3 \end{pmatrix} \quad D = (4 \quad -2 \quad 0 \quad -5)$$

- Para la **norma 1**, suma el valor absoluto de los elementos en cada **columna** y quédate con la suma más grande.
- Para la **norma infinito**, suma el valor absoluto de los elementos en cada **fila** y quédate con la suma más grande.

Ejercicio (4.2)

Ejercicio (4.2)

Para cada una de las siguientes matrices, calcula $\|\cdot\|_1$ y $\|\cdot\|_\infty$:

$$A = \begin{pmatrix} 1 & -7 & 3 \\ -5 & 2 & -1 \end{pmatrix}, \quad B = \begin{pmatrix} 3 & 0 & -2 \\ 1 & 5 & 4 \\ -6 & 1 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 8 \\ -1 \\ 3 \end{pmatrix} \quad D = (4 \quad -2 \quad 0 \quad -5)$$

- Para la **norma 1**, suma el valor absoluto de los elementos en cada **columna** y quédate con la suma más grande.
- Para la **norma infinito**, suma el valor absoluto de los elementos en cada **fila** y quédate con la suma más grande.

Matriz A:

- $\|A\|_1 = \max(|1| + |-5|, |-7| + |2|, |3| + |-1|) = \max(6, 9, 4) = 9$
- $\|A\|_\infty = \max(|1| + |-7| + |3|, |-5| + |2| + |-1|) = \max(11, 8) = 11$

Ejercicio (4.2)

Ejercicio (4.2)

Para cada una de las siguientes matrices, calcula $\|\cdot\|_1$ y $\|\cdot\|_\infty$:

$$A = \begin{pmatrix} 1 & -7 & 3 \\ -5 & 2 & -1 \end{pmatrix}, \quad B = \begin{pmatrix} 3 & 0 & -2 \\ 1 & 5 & 4 \\ -6 & 1 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 8 \\ -1 \\ 3 \end{pmatrix} \quad D = (4 \quad -2 \quad 0 \quad -5)$$

- Para la **norma 1**, suma el valor absoluto de los elementos en cada **columna** y quédate con la suma más grande.
- Para la **norma infinito**, suma el valor absoluto de los elementos en cada **fila** y quédate con la suma más grande.

Matriz A:

- $\|A\|_1 = \max(|1| + |-5|, |-7| + |2|, |3| + |-1|) = \max(6, 9, 4) = 9$
- $\|A\|_\infty = \max(|1| + |-7| + |3|, |-5| + |2| + |-1|) = \max(11, 8) = 11$

Matriz B: $\|B\|_1 = 10$, $\|B\|_\infty = 10$

Matriz C: $\|C\|_1 = 12$, $\|C\|_\infty = 8$

Matriz D: $\|D\|_1 = 5$, $\|D\|_\infty = 11$

Autovalores (Eigenvalues, Valores Propios)

Para una matriz cuadrada A , un **autovalor** λ y su **autovector** asociado $\mathbf{v}$ ($\mathbf{v} \neq 0$) satisfacen:

$$A\mathbf{v} = \lambda\mathbf{v}$$

El autovalor λ nos dice cómo la matriz A "estira" o "contrae" el vector $\mathbf{v}$ sin cambiar su dirección.

Radio espectral $\rho(A)$

El **radio espectral** de A es el máximo de los valores absolutos de todos sus autovalores:

$$\rho(A) = \max_i |\lambda_i|$$

Visualmente, si dibujamos todos los autovalores en el plano complejo, $\rho(A)$ es el radio del círculo más pequeño centrado en el origen que los contiene a todos.

Si $A_{n \times n}$ es una matriz cuadrada, entonces

- $\rho(A) \leq \|A\|$
- $\|A\|_2 = [\rho(A^t A)]^{1/2}$
- Si A es una matriz real y simétrica, entonces $\rho(A) = \|A\|_2$.

Sistema de ecuaciones lineales

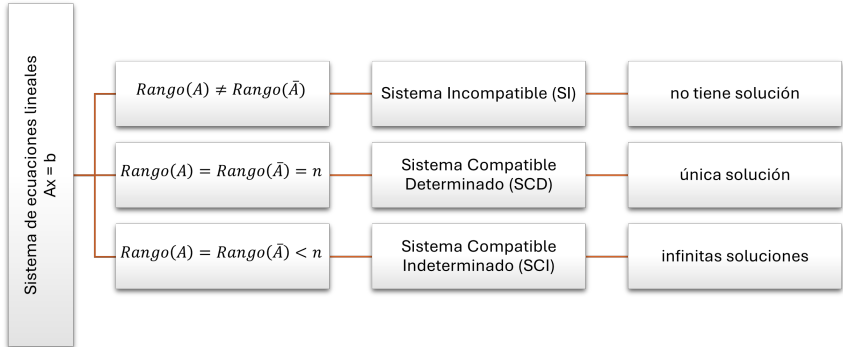
$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

Estos sistemas de ecuaciones pueden ser expresados de forma matricial

$$Ax = b$$

donde $A = \{a_{ij}\}_{j=1, i=1}^{n, m}$, $b = \{b_i\}_{i=1}^m$ y $x = \{x_j\}_{j=1}^n$.

- $A = \{a_{ij}\}_{j=1, i=1}^{n, m}$ es la **matriz de coeficientes**
- $b = \{b_i\}_{i=1}^m$ se llama **columna de términos independientes**
- $x = \{x_j\}_{j=1}^n$ es el **vector de incógnitas**
- La matriz completa $\bar{A} = (A|b)$ se le llama **matriz ampliada**



El número de condición es, de manera informal, una cantidad que nos permite medir cómo de sensible es la solución de un problema a perturbaciones en sus datos iniciales.

Consideramos el sistema

$$Ax = b$$

pero existe una pequeña perturbación δb en los datos del vector b . Esto nos lleva a que el vector solución x también será perturbado por una cantidad δx y el sistema será ahora

$$A(x + \delta x) = b + \delta b \implies Ax + A\delta x = b + \delta b \implies A\delta x = \delta b$$

Del sistema original, sabemos que

$$\|b\| = \|Ax\| \leq \|A\|\|x\|$$

y del sistema perturbado sacamos que

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

Luego combinando ambas desigualdades, tenemos que

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

Llamamos **número de condición** a la cantidad

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

Número de condición

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

- Para toda matriz A invertible se cumple:
 1. $1 \leq \kappa(A) < \infty$, $\kappa(A) = \kappa(A^{-1})$,
 2. $\kappa(\alpha A) = \kappa(A)$ para todo escalar α .
- Cuanto más cercano a 1 sea el número de condición, más estable es la resolución numérica del sistema de ecuaciones.
- Si el número de condición es 10^k con $k \geq 1$, entonces se espera que la solución del problema pierda k dígitos de precisión.
- Decimos que el problema está bien (o mal) condicionado si el número de condición es pequeño (o grande).

Ejercicio (4.5)

Se tienen dos sistemas de ecuaciones, uno con la matriz P y otro con Q :

$$P = \begin{pmatrix} 5 & 0 \\ 0 & 4 \end{pmatrix} \quad Q = \begin{pmatrix} 1 & 1 \\ 1 & 1.0001 \end{pmatrix}$$

Calcula el número de condición $\kappa(P)$ y $\kappa(Q)$ usando la norma infinito. ¿Cuál de los dos sistemas es numéricamente más estable (mejor condicionado) para resolver?

Criterio	Métodos Directos	Métodos Iterativos
Resultado	Solución exacta	Solución aproximada
Coste	Alto, típicamente $O(n^3)$	El coste total depende del número de iteraciones.
Memoria	Puede ser muy alto	Usualmente bajo, ideal para matrices esparsas.
Garantía de Solución	Pueden ser sensibles a la acumulación de errores de redondeo	La convergencia depende de las propiedades de la matriz
Ideal para...	Sistemas pequeños y/o densos , donde se requiere alta precisión.	Sistemas muy grandes y esparsos (ej. de Ecuaciones Diferenciales).
Ejemplos	Eliminación de Gauss, Factorización LU, Método de Cholesky	Método de Jacobi, Método de Gauss-Seidel, Sobrerelajación Sucesiva (SOR)

Idea: Transformar un sistema de ecuaciones lineales $Ax = b$ en un sistema equivalente y fácil de resolver, de forma triangular superior $Ux = c$.

1. **Eliminación hacia adelante (Triangularización):** Introducir ceros sistemáticamente debajo de la diagonal principal.

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array} \right] \Rightarrow \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ & a'_{22} & a'_{23} & b'_2 \\ & & a''_{33} & b''_3 \end{array} \right]$$

2. **Sustitución regresiva:** Resolver el sistema triangular superior resultante.

$$x_3 = \frac{b''_3}{a''_{33}}$$

$$x_2 = \frac{b'_2 - a'_{23}x_3}{a'_{22}}$$

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3}{a_{11}}$$

Fase 1: Eliminación hacia adelante

- Se trabaja con la **matriz aumentada** $[A|\mathbf{b}]$.
- El proceso se realiza en $n - 1$ etapas.
- En cada etapa k (para $k = 1, \dots, n - 1$):

Paso clave en la etapa k

1. Se elige el elemento $a_{kk}^{(k)}$ como **pivote**.
2. Para cada fila i por debajo del pivote ($i = k + 1, \dots, n$):
 - Se calcula el **multiplicador**: $m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$
 - Se actualiza la fila i con la operación: $F_i \leftarrow F_i - m_{ik} \cdot F_k$
3. El resultado es que todos los elementos $a_{ik}^{(k+1)}$ con $i > k$ se hacen cero.

Consideremos el sistema y su matriz aumentada:

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array} \right]$$

1. Pivote: $a_{11} = 2$.

Consideremos el sistema y su matriz aumentada:

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array} \right]$$

1. Pivote: $a_{11} = 2$.

Resultado de la Etapa 1:

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ 0 & 0.5 & 0.5 & 1 \\ 0 & 2 & 1 & 5 \end{array} \right]$$

Consideremos el sistema y su matriz aumentada:

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array} \right]$$

1. Pivote: $a_{11} = 2$.

Resultado de la Etapa 1:

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ 0 & 0.5 & 0.5 & 1 \\ 0 & 2 & 1 & 5 \end{array} \right]$$

2. Pivote: $a_{22} = 0.5$.

Consideremos el sistema y su matriz aumentada:

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array} \right]$$

1. Pivote: $a_{11} = 2$.

Resultado de la Etapa 1:

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ 0 & 0.5 & 0.5 & 1 \\ 0 & 2 & 1 & 5 \end{array} \right]$$

2. Pivote: $a_{22} = 0.5$.

Resultado de la Etapa 2:

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ 0 & 0.5 & 0.5 & 1 \\ 0 & 0 & -1 & 1 \end{array} \right]$$

El sistema triangular superior obtenido es:

$$\begin{array}{rcl} 2x_1 + x_2 - x_3 & = & 8 \\ 0.5x_2 + 0.5x_3 & = & 1 \\ -x_3 & = & 1 \end{array}$$

Se resuelve de abajo hacia arriba:

1. De la última ecuación:

$$-x_3 = 1 \implies x_3 = -1$$

2. Sustituimos x_3 en la segunda ecuación:

$$0.5x_2 + 0.5(-1) = 1 \implies 0.5x_2 = 1.5 \implies x_2 = 3$$

3. Sustituimos x_2 y x_3 en la primera ecuación:

$$2x_1 + (3) - (-1) = 8 \implies 2x_1 = 4 \implies x_1 = 2$$

Solución Final: $\mathbf{x} = [2, 3, -1]^T$

¿Qué ocurre si el elemento pivote $a_{kk}^{(k)}$ es cero o muy cercano a cero?

¿Qué ocurre si el elemento pivote $a_{kk}^{(k)}$ es cero o muy cercano a cero?

¡Un pivote nulo o muy pequeño es un desastre!

- **Pivote Cero:** El cálculo del multiplicador $m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ implica una **división por cero**.
El algoritmo falla.
- **Pivote Casi Cero:** La división por un número muy pequeño produce un multiplicador **muy grande**. Esto amplifica los errores de redondeo existentes y puede destruir la precisión de la solución.

Ejercicio (4.6), apartado b

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

La matriz es no singular, pero Gauss falla en el primer paso porque $a_{11} = 0$.

Estrategia de pivoteo parcial

En cada etapa k , antes de calcular los multiplicadores:

1. **Buscar:** Se busca el elemento de mayor valor absoluto en la columna k , desde la fila k hacia abajo.

$$|a_{pk}^{(k)}| = \max_{i=k, \dots, n} |a_{ik}^{(k)}|$$

2. **Intercambiar:** Se intercambia la fila k con la fila p (donde se encontró el máximo).

$$F_k \leftrightarrow F_p$$

3. **Proceder:** Se realiza la eliminación como antes, usando el nuevo pivote (el más grande posible).

- **Evita la división por cero** si la matriz es no singular.
- **Mejora la estabilidad numérica** al mantener los multiplicadores $|m_{ik}| \leq 1$, lo que controla la propagación de errores.

Importante

Resuelve Ejercicio (4.7)

Método de Gauss con pivoteo parcial

```
Para k = 1 hasta n-1:
// Pivoteo
p = encontrar fila con max(|A(i,k)|) para i >= k
Intercambiar Fila(k) con Fila(p)

/ Eliminación
Para i = k+1 hasta n:
multiplicador = A(i,k) / A(k,k)
Para j = k hasta n:
A(i,j) = A(i,j) - multiplicador * A(k,j)
b(i) = b(i) - multiplicador * b(k)

// Sustitución Regresiva
Resolver sistema triangular Ux = c
```

Coste computacional

El número de operaciones de punto flotante (multiplicaciones y sumas) es:

$$\text{Coste} \approx \frac{2}{3}n^3 + O(n^2) \approx O(n^3)$$

Esto lo hace muy costoso para sistemas grandes.

- Transforma el sistema en uno triangular superior, que se resuelve fácilmente por sustitución regresiva.
- El pivoteo parcial es esencial para garantizar la estabilidad numérica y evitar fallos por división entre cero.
- Su alto coste computacional ($O(n^3)$) lo hace impráctico para sistemas muy grandes y esparsos.
- Sirve como base conceptual para otros métodos directos más avanzados, como la **factorización LU**.

Idea: Factorizar la matriz A como el producto de dos matrices triangulares: una inferior (L) y una superior (U): $A = LU$.

Una vez que tenemos $A = LU$, el sistema se convierte en $LU\mathbf{x} = \mathbf{b}$:

1. Definimos un vector intermedio $\mathbf{y}$ tal que $U\mathbf{x} = \mathbf{y}$
2. Resolvemos el sistema triangular inferior para $\mathbf{y}$: $L\mathbf{y} = \mathbf{b}$, esto es rápido ($O(n^2)$).
3. Resolvemos el sistema triangular superior para $\mathbf{x}$: $U\mathbf{x} = \mathbf{y}$, esto también es rápido ($O(n^2)$).

$$\mathbf{b} \xrightarrow{\text{Sust. Adelante}} \mathbf{y} \xrightarrow{\text{Sust. Regresiva}} \mathbf{x}$$

¿De dónde vienen **L** y **U**?

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{pmatrix}, \quad U = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

- **La matriz U** es simplemente la matriz triangular superior que obtenemos al final de la fase de eliminación hacia adelante.
- **La matriz L** es una matriz triangular inferior con unos en la diagonal, que almacena en su parte inferior los **multiplicadores** m_{ik} que usamos en cada paso de la eliminación.

¡No hay magia!

La factorización LU no es un método nuevo, sino una reorganización inteligente de la eliminación gaussiana que guarda los resultados de forma estructurada.

Ejemplo: Construcción de L y U

Recordemos nuestro ejemplo de la eliminación gaussiana (Ejercicio (4.6)):

$$A = \begin{pmatrix} 2 & 1 & -1 \\ -3 & -1 & 2 \\ -2 & 1 & 2 \end{pmatrix} \xrightarrow{\text{Eliminación}} U = \begin{pmatrix} 2 & 1 & -1 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & -1 \end{pmatrix}$$

Los multiplicadores que calculamos fueron:

- $m_{21} = \frac{-3}{2} = -1.5$
- $m_{31} = \frac{-2}{2} = -1$
- $m_{32} = \frac{2}{0.5} = 4$

La matriz L se construye directamente con ellos:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -1.5 & 1 & 0 \\ -1 & 4 & 1 \end{pmatrix}$$

Y se puede comprobar que, efectivamente, $A = LU$.

La factorización $A = LU$ simple puede fallar o ser numéricamente inestable por la misma razón que la eliminación gaussiana básica: un pivote nulo o pequeño.

Factorización $PA = LU$

En la práctica, casi siempre se usa el pivoteo parcial. Esto es equivalente a reordenar las filas de A antes de la factorización.

- Se introduce una **matriz de permutación** P que registra todos los intercambios de filas.
- La descomposición robusta es: $PA = LU$.

Proceso de solución con pivoteo

$$Ax = \mathbf{b} \implies PAx = P\mathbf{b} \implies LUx = P\mathbf{b}$$

1. Resolver $Ly = P\mathbf{b}$.
2. Resolver $Ux = \mathbf{y}$.

Para resolver $Ax = b$, el comando más eficiente y recomendado es:

$$x = A \setminus b$$

¡Importante! Esto **NO** es lo mismo que $\text{inv}(A)*b$. Calcular la inversa explícitamente es más costoso y numéricamente menos estable.

MATLAB analiza la matriz A y elige el mejor método:

- Si A es triangular, usa sustitución.
- Si A es simétrica y definida positiva, usa factorización de Cholesky.
- Si A es cuadrada general, usa factorización LU con pivoteo parcial.

Factorización LU

$$[L, U, P] = \text{lu}(A)$$

Esto calcula la descomposición $PA = LU$. Luego, podrías resolver el sistema manualmente:

$$\begin{aligned} y &= L \setminus (P*b); \quad \% \text{ Resuelve } Ly = Pb \\ x &= U \setminus y; \quad \% \text{ Resuelve } Ux = y \end{aligned}$$

- La convergencia de ambos métodos está garantizada.
- La factorización LU es una representación matricial de la eliminación gaussiana.
- Su principal ventaja es la eficiencia para resolver sistemas $A\mathbf{x} = \mathbf{b}$ con la misma matriz A y múltiples vectores $\mathbf{b}$.
- Otra ventaja de la factorización LU es que permite calcular fácilmente el determinante de una matriz dada. Si consideramos la factorización LU de la matriz A

$$A = LU$$

por propiedades de determinantes, sabemos que

$$\det(A) = \det(L) \det(U) = \det(U) = u_{11} \cdots u_{nn}$$

(ya que $\det(L) = 1$ por construcción).

Importante

Resuelve Ejercicios (4.8) – (4.9)

Limitaciones de los Métodos Directos (Gauss, LU)

- **Coste Computacional:** La complejidad de $O(n^3)$ los hace inviables para sistemas muy grandes ($n > 10^5$).
- **Almacenamiento y "Fill-in":** Si la matriz A es *esparsa* (tiene muchos ceros), la eliminación gaussiana puede destruir esa estructura llenando la matriz de elementos no nulos ("fill-in"), lo que consume mucha memoria.

Limitaciones de los Métodos Directos (Gauss, LU)

- **Coste Computacional:** La complejidad de $O(n^3)$ los hace inviables para sistemas muy grandes ($n > 10^5$).
- **Almacenamiento y "Fill-in":** Si la matriz A es *esparsa* (tiene muchos ceros), la eliminación gaussiana puede destruir esa estructura llenando la matriz de elementos no nulos ("fill-in"), lo que consume mucha memoria.

Idea: Transformar el sistema $Ax = b$ en una forma equivalente de punto fijo:

$$x = Tx + c$$

Donde T es la **matriz de iteración** y c es un vector.

Proceso iterativo:

$$x^{(0)} \xrightarrow{\text{Iterar}} x^{(1)} \xrightarrow{\text{Iterar}} x^{(2)} \xrightarrow{\text{Iterar}} \dots \rightarrow x_{\text{solución}}$$

Condición necesaria y suficiente de convergencia

La iteración $\mathbf{x}^{(k+1)} = T\mathbf{x}^{(k)} + \mathbf{c}$ converge a la solución única para cualquier vector inicial $\mathbf{x}^{(0)}$ si y solo si el radio espectral de T es menor que 1:

$$\rho(T) < 1$$

donde $\rho(T) = \max_i |\lambda_i|$ es el valor propio (autovalor) de mayor magnitud de T .

Condición suficiente (más fácil de verificar)

Si la matriz original $A_{n \times n}$ es **estrictamente diagonal dominante**,

$$|a_{kk}| > \sum_{j=1, j \neq k}^n |a_{kj}|, \quad k = 1, 2, \dots, n,$$

entonces los métodos de Jacobi y Gauss-Seidel convergen.

Idea: En la ecuación i -ésima del sistema, $\sum_{j=1}^n a_{ij}x_j = b_i$, se despeja la variable x_i y se usan los valores del paso anterior para las demás variables.

Sea el sistema

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

Idea: En la ecuación i -ésima del sistema, $\sum_{j=1}^n a_{ij}x_j = b_i$, se despeja la variable x_i y se usan los valores del paso anterior para las demás variables.

Sea el sistema

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

consideramos las fórmulas de iteración:

$$\begin{cases} x_1^{(k+1)} = \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(k)} - \cdots - a_{1n}x_n^{(k)}) \\ x_2^{(k+1)} = \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(k)} - \cdots - a_{2n}x_n^{(k)}) \\ \vdots \\ x_n^{(k+1)} = \frac{1}{a_{nn}}(b_n - a_{n1}x_1^{(k)} - \cdots - a_{n(n-1)}x_{n-1}^{(k)}) \end{cases}$$

Método de Jacobi

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

Se necesitan dos vectores en memoria: uno para $\mathbf{x}^{(k)}$ y otro para $\mathbf{x}^{(k+1)}$.

Forma Matricial

El método de Jacobi también se puede interpretar de forma matricial:

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

donde D es la matriz diagonal de A , y $J = -D^{-1}(A - D)$:

$$J = \begin{pmatrix} 0 & -a_{12}/a_{11} & \dots & -a_{1n}/a_{11} \\ -a_{21}/a_{22} & 0 & \dots & -a_{2n}/a_{22} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1}/a_{nn} & -a_{n2}/a_{nn} & \dots & 0 \end{pmatrix}$$

Ejercicio (4.10)

Ejercicio (4.10)

Utilizando el método de Jacobi, calcula tres iteraciones para el siguiente sistema, empezando desde el vector nulo $\mathbf{x}^{(0)} = (0, 0, 0)^T$. ¿Converge la iteración?

$$\begin{cases} 4x - y + z = 7 \\ 4x - 8y + z = -21 \\ -2x + y + 5 = 15 \end{cases}$$

Ejercicio (4.10)

Ejercicio (4.10)

Utilizando el método de Jacobi, calcula tres iteraciones para el siguiente sistema, empezando desde el vector nulo $\mathbf{x}^{(0)} = (0, 0, 0)^T$. ¿Converge la iteración?

$$\begin{cases} 4x - y + z = 7 \\ 4x - 8y + z = -21 \\ -2x + y + 5 = 15 \end{cases}$$

1. Convergencia

- **Fila 1:** $|4| > |-1| + |1| \implies 4 > 2$ (Se cumple)
- **Fila 2:** $|-8| > |4| + |1| \implies 8 > 5$ (Se cumple)
- **Fila 3:** $|5| > |-2| + |1| \implies 5 > 3$ (Se cumple)

Como la condición se cumple para todas las filas, la matriz es estrictamente diagonal dominante. Por lo tanto, el método de Jacobi convergerá a la solución única del sistema, sin importar el punto de partida.

2. Fórmulas de Iteración de Jacobi

Despejamos cada variable de una ecuación distinta:

$$\begin{cases} 4x - y + z = 7 \\ 4x - 8y + z = -21 \\ -2x + y + 5 = 15 \end{cases} \implies \begin{cases} x^{(k+1)} = \frac{1}{4}(7 + y^{(k)} - z^{(k)}) \\ y^{(k+1)} = \frac{1}{8}(21 + 4x^{(k)} + z^{(k)}) \\ z^{(k+1)} = \frac{1}{5}(15 + 2x^{(k)} - y^{(k)}) \end{cases}$$

2. Fórmulas de Iteración de Jacobi

Despejamos cada variable de una ecuación distinta:

$$\begin{cases} 4x - y + z = 7 \\ 4x - 8y + z = -21 \\ -2x + y + 5 = 15 \end{cases} \implies \begin{cases} x^{(k+1)} = \frac{1}{4}(7 + y^{(k)} - z^{(k)}) \\ y^{(k+1)} = \frac{1}{8}(21 + 4x^{(k)} + z^{(k)}) \\ z^{(k+1)} = \frac{1}{5}(15 + 2x^{(k)} - y^{(k)}) \end{cases}$$

3. Cálculo de iteraciones:

Comenzamos con el vector nulo: $\mathbf{x}^{(0)} = (x^{(0)}, y^{(0)}, z^{(0)})^T = (0, 0, 0)^T$.

Iteración 1 ($k=0$): $\mathbf{x}^{(1)} = (1.75, 2.625, 3)^T$.

Iteración 2 ($k=1$): $\mathbf{x}^{(2)} = (1.65625, 3.875, 3.175)^T$.

Iteración 3 ($k=2$): $\mathbf{x}^{(3)} = (1.925, 3.85, 2.8875)^T$.

Como la solución es aproximada, debemos definir una tolerancia ε y parar cuando se cumpla una condición.

Se detiene cuando la diferencia entre dos iteraciones sucesivas es suficientemente pequeña:

$$\frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}^{(k+1)}\|} < \varepsilon \quad (\text{Error relativo})$$

o

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \varepsilon \quad (\text{Error absoluto})$$

¡Importante!

Siempre se debe incluir un **número máximo de iteraciones** para evitar bucles infinitos si el método no converge.

Ejercicio (4.11)

Expresa los sistemas planteados en el ejercicio (4.10) en su forma matricial. A continuación, utiliza la formulación matricial del método de Jacobi para implementar el algoritmo y hallar la solución numérica.

Cuando calculamos $x_i^{(k+1)}$, ya hemos calculado los nuevos valores $x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}$.
¿Por qué no usarlos inmediatamente?

Método de Gauss-Seidel

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

- Solo se necesita un vector en memoria, ya que los valores se sobrescriben.
- Generalmente, si converge, lo hace más rápido que Jacobi.

Forma Matricial

Se descompone $A = D - L - U$ (Diagonal, Triangular Inferior, Triangular Superior). La matriz de iteración de Gauss-Seidel es:

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

Ejercicio (4.12)

Ejercicio (4.12)

Utilizando el método de Gauss-Seidel, calcula tres iteraciones para el siguiente sistema, empezando desde el vector nulo $\mathbf{x}^{(0)} = (0, 0, 0)^T$. ¿Converge la iteración?

$$\begin{cases} 4x - y + z = 7 \\ 4x - 8y + z = -21 \\ -2x + y + 5z = 15 \end{cases}$$

Ejercicio (4.12)

Utilizando el método de Gauss-Seidel, calcula tres iteraciones para el siguiente sistema, empezando desde el vector nulo $\mathbf{x}^{(0)} = (0, 0, 0)^T$. ¿Converge la iteración?

$$\begin{cases} 4x - y + z = 7 \\ 4x - 8y + z = -21 \\ -2x + y + 5 = 15 \end{cases}$$

1. Convergencia: El método converge ya que la matriz es estrictamente diagonal dominante.

Ejercicio (4.12)

Ejercicio (4.12)

Utilizando el método de Gauss-Seidel, calcula tres iteraciones para el siguiente sistema, empezando desde el vector nulo $\mathbf{x}^{(0)} = (0, 0, 0)^T$. ¿Converge la iteración?

$$\begin{cases} 4x - y + z = 7 \\ 4x - 8y + z = -21 \\ -2x + y + 5 = 15 \end{cases}$$

1. Convergencia: El método converge ya que la matriz es estrictamente diagonal dominante.

2. Fórmulas de Iteración

$$x^{(k+1)} = \frac{1}{4}(7 + y^{(k)} - z^{(k)})$$

$$y^{(k+1)} = \frac{1}{8}(21 + 4x^{(k+1)} + z^{(k)})$$

$$z^{(k+1)} = \frac{1}{5}(15 + 2x^{(k+1)} - y^{(k+1)})$$

3. Cálculo de iteraciones:

Comenzamos con el vector nulo: $\mathbf{x}^{(0)} = (x^{(0)}, y^{(0)}, z^{(0)})^T = (0, 0, 0)^T$.

Iteración 1 ($k=0$): $\mathbf{x}^{(1)} = (1.75, 3.5, 3)^T$.

Iteración 2 ($k=1$): $\mathbf{x}^{(2)} = (1.875, 3.9375, 2.9625)^T$.

Iteración 3 ($k=2$): $\mathbf{x}^{(3)} = (1.99375, 3.9921875, 2.9990625)^T$.

Método	$x^{(3)}$	$y^{(3)}$	$z^{(3)}$
Jacobi	1.925	3.85	2.8875
Gauss-Seidel	1.99375	3.9921875	2.9990625
Solución Exacta	2.0	4.0	3.0

- Ambos métodos convergen hacia la solución, como se esperaba, ya que la matriz es estrictamente diagonal dominante.
- Tras solo tres iteraciones, se observa claramente que la aproximación obtenida con el **método de Gauss-Seidel está significativamente más cerca** de la solución exacta que la obtenida con el método de Jacobi.

Métodos clásicos (Jacobi y Gauss-Seidel) son excelentes para aprender y muy fáciles de implementar, pero a menudo son demasiado lentos o su convergencia es muy limitada para ser usados en software profesional.

Función	Nombre Completo	Tipo de Matriz	Descripción
pcg	Gradiente Conjugado Precond.	Simétrica, def. positiva	El método más rápido para esta clase.
gmres	Residuo Mínimo Generalizado	Cuadrada, no simétrica	Uno de los más populares y robustos.
minres	Residuo Mínimo	Simétrica, indefinida	Para matrices simétricas no def. positivas.
lsqr	Mínimos Cuadrados	Rectangular o cuadrada	Resuelve sistemas de mínimos cuadrados.

La sintaxis para usar estos métodos es muy similar entre ellos:

```
x = gmres(A, b, tol, maxit, x0);
```

- A: La matriz de coeficientes
- b: El vector de términos independientes.
- tol: (Opcional) La tolerancia del error. Por defecto es $1e - 6$.
- maxit: (Opcional) El número máximo de iteraciones.
- x0: (Opcional) La estimación inicial de la solución.

	Métodos Directos (Gauss, LU)	Métodos Iterativos (Jacobi, Gauss-Seidel)
Pros	Robustos, predecibles, dan la solución "exacta" (sin errores de redondeo)	Mucho más rápidos y eficientes en memoria para sistemas grandes y esparsos
Contras	Muy costosos ($O(n^3)$) para sistemas grandes	La convergencia no está garantizada y depende de la matriz. La solución es una aproximación
Usar cuando	El sistema es pequeño, denso o se necesita alta precisión garantizada.	El sistema es grande, esparso y la matriz tiene propiedades que aseguran la convergencia

Autovalores y Autovectores

Dado una matriz cuadrada $A \in \mathbb{C}^{n \times n}$, un escalar λ es un **autovalor** de A si existe un vector no nulo $\mathbf{v}$ (su **autovector** asociado) tal que:

$$A\mathbf{v} = \lambda\mathbf{v}$$

Aplicaciones:

- **Física:** Frecuencias de vibración de un sistema mecánico.
- **Estabilidad:** Análisis de sistemas dinámicos y ecuaciones diferenciales.
- **Ciencia de Datos:** Componentes Principales (PCA), PageRank de Google.

Autovalores y Autovectores

Dado una matriz cuadrada $A \in \mathbb{C}^{n \times n}$, un escalar λ es un **autovalor** de A si existe un vector no nulo $\mathbf{v}$ (su **autovector** asociado) tal que:

$$A\mathbf{v} = \lambda\mathbf{v}$$

Aplicaciones:

- **Física:** Frecuencias de vibración de un sistema mecánico.
- **Estabilidad:** Análisis de sistemas dinámicos y ecuaciones diferenciales.
- **Ciencia de Datos:** Componentes Principales (PCA), PageRank de Google.

Los autovalores son las raíces de la ecuación característica: $\det(A - \lambda I) = 0$.

- Para matrices grandes, calcular el determinante y encontrar las raíces del polinomio es computacionalmente prohibitivo y numéricamente inestable.
- No hay fórmulas generales para raíces de polinomios de grado ≥ 5 .

Autovalores y Autovectores

Dado una matriz cuadrada $A \in \mathbb{C}^{n \times n}$, un escalar λ es un **autovalor** de A si existe un vector no nulo $\mathbf{v}$ (su **autovector** asociado) tal que:

$$A\mathbf{v} = \lambda\mathbf{v}$$

Aplicaciones:

- **Física:** Frecuencias de vibración de un sistema mecánico.
- **Estabilidad:** Análisis de sistemas dinámicos y ecuaciones diferenciales.
- **Ciencia de Datos:** Componentes Principales (PCA), PageRank de Google.

Los autovalores son las raíces de la ecuación característica: $\det(A - \lambda I) = 0$.

- Para matrices grandes, calcular el determinante y encontrar las raíces del polinomio es computacionalmente prohibitivo y numéricamente inestable.
- No hay fórmulas generales para raíces de polinomios de grado ≥ 5 .

Conclusión: Necesitamos métodos numéricos iterativos.

El cálculo de autovalores puede ser un problema **mal condicionado**. Pequeños cambios en la matriz pueden causar grandes cambios en los autovalores:

Ejemplo

Sean dos matrices A y A' :

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad A' = \begin{pmatrix} 0 & 1 \\ \epsilon & 0 \end{pmatrix}$$

El cálculo de autovalores puede ser un problema **mal condicionado**. Pequeños cambios en la matriz pueden causar grandes cambios en los autovalores:

Ejemplo

Sean dos matrices A y A' :

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad A' = \begin{pmatrix} 0 & 1 \\ \epsilon & 0 \end{pmatrix}$$

- Los autovalores de A son $\lambda_1 = \lambda_2 = 0$.
- Los autovalores de A' son $\lambda' = \pm\sqrt{\epsilon}$.

El cálculo de autovalores puede ser un problema **mal condicionado**. Pequeños cambios en la matriz pueden causar grandes cambios en los autovalores:

Ejemplo

Sean dos matrices A y A' :

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad A' = \begin{pmatrix} 0 & 1 \\ \epsilon & 0 \end{pmatrix}$$

- Los autovalores de A son $\lambda_1 = \lambda_2 = 0$.
- Los autovalores de A' son $\lambda' = \pm\sqrt{\epsilon}$.

Imaginemos que $\epsilon = 10^{-16}$

- cambio en la matriz: 10^{-16} .
- Los nuevos autovalores: $\lambda' = \pm\sqrt{10^{-16}} = \pm 10^{-8}$.

El cálculo de autovalores puede ser un problema **mal condicionado**. Pequeños cambios en la matriz pueden causar grandes cambios en los autovalores:

Ejemplo

Sean dos matrices A y A' :

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad A' = \begin{pmatrix} 0 & 1 \\ \epsilon & 0 \end{pmatrix}$$

- Los autovalores de A son $\lambda_1 = \lambda_2 = 0$.
- Los autovalores de A' son $\lambda' = \pm\sqrt{\epsilon}$.

Imaginemos que $\epsilon = 10^{-16}$

- cambio en la matriz: 10^{-16} .
- Los nuevos autovalores: $\lambda' = \pm\sqrt{10^{-16}} = \pm 10^{-8}$.

Este es un problema **extremadamente mal condicionado**. La causa es la estructura no simétrica con autovalores repetidos. Las matrices simétricas no sufren de este problema tan severo.

Teorema de Gerschgorin

Todos los autovalores de una matriz A se encuentran en la unión de n discos en el plano complejo, definidos por:

$$D_i = \left\{ z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}| \right\}$$

Para cada fila i :

- El **centro** del círculo es el elemento de la diagonal a_{ii} .
- El **radio** es la suma de los valores absolutos de los demás elementos de esa fila.

Ejercicio (4.13)

Utiliza el Teorema de Gerschgorin para estimar la localización de los autovalores de la matriz

$$A = \begin{pmatrix} 4 & -1 & 0 \\ 1 & 1 & 1 \\ 0 & 2 & -2 \end{pmatrix}$$

Teorema de Gerschgorin

Todos los autovalores de una matriz A se encuentran en la unión de n discos en el plano complejo, definidos por:

$$D_i = \left\{ z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}| \right\}$$

Para cada fila i :

- El **centro** del círculo es el elemento de la diagonal a_{ii} .
- El **radio** es la suma de los valores absolutos de los demás elementos de esa fila.

Ejercicio (4.13)

Utiliza el Teorema de Gerschgorin para estimar la localización de los autovalores de la matriz

$$A = \begin{pmatrix} 4 & -1 & 0 \\ 1 & 1 & 1 \\ 0 & 2 & -2 \end{pmatrix}$$

Todos los autovalores están en $[-4, 5]$.

MATLAB ofrece la función `eig` para calcular autovalores y autovectores. Tiene dos formas de uso principales:

1. `lambda = eig(A)`

Devuelve un vector columna con todos los autovalores de la matriz `A`.

2. `[V, D] = eig(A)`

Devuelve dos matrices:

- `D`: Es una matriz **diagonal** que contiene los autovalores λ_i en su diagonal.
- `V`: Es una matriz cuyas **columnas** son los autovectores correspondientes a cada autovalor en `D`.

MATLAB ofrece la función `eig` para calcular autovalores y autovectores. Tiene dos formas de uso principales:

1. `lambda = eig(A)`

Devuelve un vector columna con todos los autovalores de la matriz `A`.

2. `[V, D] = eig(A)`

Devuelve dos matrices:

- `D`: Es una matriz **diagonal** que contiene los autovalores λ_i en su diagonal.
- `V`: Es una matriz cuyas **columnas** son los autovectores correspondientes a cada autovalor en `D`.

Para matrices grandes y/o esparsas, se usa `eigs`:

`[V, D] = eigs(A, k)`

(Devuelve los `k` autovalores de mayor magnitud)

Ejercicios

Resuelve Ejercicios (4.14)–(4.15)

Bibliografía

1. **Capítulos 9–11** de
Chapra S.C.; Canale R. (2005) *Métodos Numéricos para Ingenieros*. Ed. McGraw-Hill
2. **Capítulos 3 y 6** de
Mathews J.H., Fink K.D. (2000) *Métodos Numéricos con MATLAB*. Ed. Prentice Hall.