

# **BLOQUE II: RESOLUCIÓN DE SISTEMAS DE ECUACIONES.**

## **II.2 – Sistemas lineales: Métodos iterativos.**

→ Sistemas lineales: Métodos iterativos.

- Método de Jacobi.
- Método de Gauss-Seidel.
- Método de sobrerelajación.

→ Métodos preferibles a los directos cuando la matriz de coeficientes tiene muchos ceros (ecuaciones diferenciales en derivadas parciales).

→ Se pueden usar para reducir el error de redondeo en las soluciones calculadas por los métodos directos.

→ Aplicables también a sistemas de ecuaciones no lineales.

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

→ Sea el sistema de  $n$  ecuaciones:

→ Despejando de cada una de las ecuaciones una incognita,

$$\begin{aligned}
 x_1 &= \left( \frac{1}{A_{11}} \right) [b_1 - A_{12}x_2 - A_{13}x_3 - \dots - A_{1n}x_n] \\
 x_2 &= \left( \frac{1}{A_{22}} \right) [b_2 - A_{21}x_1 - A_{23}x_3 - \dots - A_{2n}x_n] \\
 &\dots\dots\dots \\
 x_n &= \left( \frac{1}{A_{nn}} \right) [b_n - A_{n1}x_1 - A_{n2}x_2 - \dots - A_{nn-1}x_{n-1}]
 \end{aligned}$$

(\*) reordenamiento similar al método de la iteración del punto fijo,  $x=g(x)$ ,

o en forma más compacta:

$$x_i = \frac{b_i}{A_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{A_{ij}}{A_{ii}} x_j \quad \leftarrow (A_{ii} \neq 0)$$

## MÉTODO DE JACOBI ó de desplazamientos simultáneos.

→ Se asumen valores iniciales para la solución:  $[x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}]$

Se puede tomar cualquier valor. Sugerencia:  $x_i^{(0)} = \frac{b_i}{A_{ii}}$  (si  $A_{ii} \neq 0$ )

→ Esta primera aproximación de orden (0) se emplea para calcular  $[x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}]$ .

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

→ Estos nuevos valores se emplean simultáneamente para calcular un nuevo conjunto de

valores de la solución  $[x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)}]$ . →  $x_i^{(2)} = \frac{b_i}{A_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{A_{ij}}{A_{ii}} x_j^{(1)}$

→ Se continua con el proceso hasta encontrar la convergencia.

→ La ecuación genérica, para la iteración  $k+1$ , viene dada por:


$$x_i^{(k+1)} = \frac{b_i}{A_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{A_{ij}}{A_{ii}} x_j^{(k)}$$

## MÉTODO DE GAUSS-SEIDEL ó de desplazamientos sucesivos.

→ Variante del método de Jacobi.

→ Para acelerar la convergencia, el proceso de sustitución no se hace en bloque simultáneamente, sino que cada elemento recién calculado se utiliza inmediatamente en el cálculo siguiente, continuando de forma iterativa hasta encontrar la convergencia.

→ La ecuación genérica viene dada por:

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


**Ejemplo:** notación para la primera iteración.

$$x_1^{(1)} = \frac{1}{A_{11}} \left( b_1 - \sum_{j=2}^n A_{1j} x_j^{(0)} \right) \rightarrow x_2^{(1)} = \frac{1}{A_{22}} \left( b_2 - A_{21} x_1^{(1)} - \sum_{j=3}^n A_{2j} x_j^{(0)} \right) \rightarrow$$

$$x_3^{(1)} = \frac{1}{A_{33}} \left( b_3 - A_{31} x_1^{(1)} - A_{32} x_2^{(1)} - \sum_{j=4}^n A_{3j} x_j^{(0)} \right) \rightarrow \dots \rightarrow x_{n-1}^{(1)} = \frac{1}{A_{n-1,n-1}} \left( b_{n-1} - \sum_{j=1}^{n-2} A_{n-1,j} x_j^{(1)} - A_{n-1,n} x_n^{(0)} \right)$$

$$\rightarrow x_n^{(1)} = \frac{1}{A_{nn}} \left( b_n - \sum_{j=1}^{n-1} A_{nj} x_j^{(1)} \right)$$

## **Convergencia:**

→ Si  $|a_{ii}| \geq \sum_{j=1; j \neq i}^n |a_{ij}|$  los métodos de Jacobi y Gauss-Seidel convergen siempre con independencia de los valores iniciales.

→ Es una **condición suficiente pero no necesaria**.

→ El sistema de ecuaciones se puede reordenar con objeto de que se cumpla esta condición.

→ Los sistemas que cumplen esta condición se denominan **diagonalmente dominantes**.

→ Hay muchos problemas en Física que se pueden plantear como un sistema de ecuaciones, y que cumplen esta condición (transferencia de calor).

→ En contraste con los métodos directos, no se acumulan los errores de redondeo de una iteración a otra. Después de cada iteración ( $k$ ), el nuevo resultado es tomado como una nueva estimación de la aproximación inicial a la raíz real.

### Ejemplo:

→ Sea el sistema:

$$6x_1 - 2x_2 + x_3 = 11$$

$$x_1 + 2x_2 - 5x_3 = -1$$

$$-2x_1 + 7x_2 + 2x_3 = 5$$

(la solución exacta es  $x_1=2, x_2=1, x_3=1$ )

→ Intercambiamos las filas 2 y 3 para que el sistema sea diagonalmente dominante:

$$6x_1 - 2x_2 + x_3 = 11$$

$$-2x_1 + 7x_2 + 2x_3 = 5$$

$$x_1 + 2x_2 - 5x_3 = -1$$

En los métodos iterativos tenemos que reordenar la ecuación de la forma:

$$x_i = \left( \frac{b_i}{A_{ii}} - \sum_{j=1; j \neq i}^n \frac{A_{ij}}{A_{ii}} x_j \right); i=1, 2, 3$$

lo que nos da:

$$x_1 = 1.8333 + 0.3333x_2 - 0.1667x_3$$

$$x_2 = 0.7143 + 0.2857x_1 - 0.2857x_3$$

$$x_3 = 0.2000 + 0.2000x_1 + 0.4000x_2$$

**Método de Jacobi:** (o de desplazamientos simultáneos)

$$x_1^{(k+1)} = 1.8333 + 0.3333x_2^{(k)} - 0.1667x_3^{(k)}$$

$$x_2^{(k+1)} = 0.7143 + 0.2857x_1^{(k)} - 0.2857x_3^{(k)}$$

$$x_3^{(k+1)} = 0.2000 + 0.2000x_1^{(k)} + 0.4000x_2^{(k)}$$

Empezando con un valor inicial  $[x_1^{(0)}, x_2^{(0)}, x_3^{(0)}] = (0,0,0)$ ,

obtenemos las siguientes aproximaciones:

	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	.....	<b><math>k=8</math></b>
$x_1$	0	1.833	2.038	2.085	2.004	1.994	.....	2.000
$x_2$	0	0.714	1.181	1.053	1.001	0.990	.....	1.000
$x_3$	0	0.200	0.852	1.080	1.038	1.001	.....	1.000

## Método de Gauss-Seidel: (o de desplazamientos sucesivos)

$$\begin{aligned}x_1^{(k+1)} &= 1.8333 + 0.3333x_2^{(k)} - 0.1667x_3^{(k)} \\x_2^{(k+1)} &= 0.7143 + 0.2857x_1^{(k+1)} - 0.2857x_3^{(k)} \\x_3^{(k+1)} &= 0.2000 + 0.2000x_1^{(k+1)} + 0.4000x_2^{(k+1)}\end{aligned}$$

Empezando con un valor inicial  $[x_1^{(0)}, x_2^{(0)}, x_3^{(0)}] = (0, 0, 0)$

obtenemos las siguientes aproximaciones:

	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
$x_1$	0	1.833	2.069	1.998	1.999	2.000
$x_2$	0	1.238	1.002	0.995	1.000	1.000
$x_3$	0	1.062	1.015	0.998	1.000	1.000

**$\Rightarrow$  Convergencia más rápida  $\Leftarrow$**



**Algoritmo:** Método iterativo de Jacobi.

// La aproximación inicial se almacena en el vector oldx. Sugerencia oldx[i]= b[i]/a[i][i]

```
for (int i=0; i<=n-1; i++) {  
    newx[i]=oldx[i];  
}  
do {  
    for (int i=0; i<=n-1; i++) {  
        oldx[i]=newx[i];  
    } // final bucle i  
    for (int i=0; i<=n-1; i++) {  
        newx[i]=b[i]/ a[i][i];  
        for (int j=0; j<=n-1; j++) {  
  
            if (j != i) { newx[i]=newx[i] - (a[i][j]/ a[i][i])*oldx[j];}  
  
        } // final bucle j  
    } // final bucle i  
}
```

// →

$$x_i^{(k+1)} = \frac{b_i}{A_{ii}} - \sum_{\substack{j=0 \\ j \neq i}}^{n-1} \frac{A_{ij}}{A_{ii}} x_j^{(k)}$$

```
} while (fabs(newx - oldx)) > tolerancia; // ← hay que definir la norma de un vector
```

// Conviene introducir un “contador” para obtener el número de iteraciones

**Algoritmo:** Método iterativo de Gauss-Seidel.

```
// La aproximación inicial se almacena en el vector oldx. Sugerencia oldx[i]= b[i]/a[i][i]
for (int i=0; i<=n-1; i++) {
    newx[i]=oldx[i];
}
do {
    for (int i=0; i<=n-1; i++) {
        oldx[i]=newx[i];
    } // final bucle i
    for (int i=0; i<=n-1; i++) {
        newx[i]=b[i]/a[i][i];
        for (int j=0; j<=n-1; j++) {
            if (j != i) { newx[i]=newx[i] - (a[i][j]/ a[i][i])*newx[j];} // → se utiliza el valor actual
        } // final bucle j
    } // final bucle i
} while (fabs(newx - oldx)) > tolerancia; // ← hay que definir la norma de un vector
// Conviene introducir un “contador” para obtener el número de iteraciones
```

→ Un problema en el caso de los métodos iterativos es cómo determinar cuando ha ocurrido la convergencia. Podemos considerar la solución  $\mathbf{x}=[x_1, x_2, \dots, x_n]$  como un vector cuya **norma**

**euclidiana** viene dada por:  $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^n x_i^2}$ .

De tal forma que el criterio de convergencia se pueda expresar como:

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \text{tolerancia}$$

→ Otra opción es:  $\sqrt{\sum_{i=1}^n (x_i^{(k+1)} - x_i^{(k)})^2} < \text{tolerancia}$

→ Otra opción es comparar componente a componente: if (fabs(newx[i]-oldx[i]) < tolerancia, haciendo un bucle para todas las componentes.

(\*)→ **Enfoque matricial en los métodos iterativos:**

→ Existe una clara analogía entre los métodos de Jacobi y Gauss-Seidel, y el método del punto fijo. Podemos expresar el proceso iterativo en notación matricial:  $\mathbf{x}^{(k+1)} = G \cdot \mathbf{x}^{(k)}$ , donde  $\mathbf{x}^{(k+1)}$  y  $\mathbf{x}^{(k)}$  son vectores de dimensión  $n$ , y  $G$  es una transformación lineal en lugar de una función.

→ De acuerdo a las expresiones que hemos visto,  $G$  debe contener un término independiente y otro lineal en  $\mathbf{x}^{(k)}$  para el método de Jacobi:  $\mathbf{x}^{(k+1)} = \mathbf{b}' - B \cdot \mathbf{x}^{(k)}$

→ Para deducir la expresión de  $\mathbf{b}'$  y  $B$ , separamos la matriz  $A$  en 3 partes:  $A = L + D + U$ ; donde  $L$  es una matriz con los coeficientes de  $A$  que están por debajo de la diagonal,  $U$  es una matriz con los coeficientes que están por encima de la diagonal, y  $D$  una matriz que contiene sólo los elementos de la diagonal de  $A$  y el resto son ceros.

$$L = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ A_{21} & 0 & \cdots & 0 \\ \vdots & \cdots & \cdots & 0 \\ A_{n1} & A_{n2} & \cdots & 0 \end{bmatrix}; D = \begin{bmatrix} A_{11} & 0 & \cdots & 0 \\ 0 & A_{22} & \cdots & 0 \\ \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & A_{nn} \end{bmatrix}; U = \begin{bmatrix} 0 & A_{12} & \cdots & A_{1n} \\ 0 & 0 & \cdots & A_{2n} \\ \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

→ Con esta descomposición de  $A$  tenemos:  $A\mathbf{x} = \mathbf{b} \rightarrow (L + D + U)\mathbf{x} = \mathbf{b}$

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

→ Por tanto tenemos:  $\boxed{\mathbf{x}^{(k+1)} = D^{-1} \cdot \mathbf{b} - D^{-1} \cdot (L + U) \mathbf{x}^{(k)} = \mathbf{b}' - B \cdot \mathbf{x}^{(k)}}$  →  $\begin{cases} \mathbf{b}' = D^{-1} \cdot \mathbf{b} \\ B = D^{-1} \cdot (L + U) \end{cases}$

$$\text{donde } D^{-1} = \begin{bmatrix} \frac{1}{A_{11}} & 0 & \cdots & 0 \\ 0 & \frac{1}{A_{22}} & \cdots & 0 \\ \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & \frac{1}{A_{nn}} \end{bmatrix}; \quad L + U = \begin{bmatrix} 0 & A_{12} & \cdots & A_{1n} \\ A_{21} & 0 & \cdots & A_{2n} \\ \vdots & \cdots & \cdots & \vdots \\ A_{n1} & A_{n2} & \cdots & 0 \end{bmatrix}$$

→ Para el método de Gauss-Seidel también podemos llegar a una expresión matricial:

$$A\mathbf{x} = \mathbf{b} \rightarrow (L + D + U)\mathbf{x} = \mathbf{b}$$

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

**Ejemplo:** aproximación matricial del método de Jacobi.

$$\begin{aligned}6x_1 - 2x_2 + x_3 &= 11 \\ -2x_1 + 7x_2 + 2x_3 &= 5 \\ x_1 + 2x_2 - 5x_3 &= -1\end{aligned}$$

$$L = \begin{bmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 2 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & -5 \end{bmatrix} \quad U = \begin{bmatrix} 0 & -2 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

$$D^{-1} = \begin{bmatrix} 0.1667 & 0 & 0 \\ 0 & 0.1429 & 0 \\ 0 & 0 & -0.2000 \end{bmatrix}$$

$$\mathbf{b}' = D^{-1} \cdot \mathbf{b} = \begin{bmatrix} 1.8333 \\ 0.7143 \\ 0.2000 \end{bmatrix} \quad B = D^{-1} \cdot (L + U) = \begin{bmatrix} 0 & -0.3333 & 0.1667 \\ -0.2857 & 0 & 0.2857 \\ -0.2000 & -0.4000 & 0 \end{bmatrix}$$

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

## MÉTODO DE SOBRRERRELAJACIÓN.


→ Objetivo: Acelerar la convergencia del método de Gauss-Seidel.

→ La ecuación genérica del método de Gauss-Seidel viene dada por:

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

→ Si sumamos y restamos  $x_i^{(k)}$  en el lado derecho de esta ecuación obtenemos la forma

equivalente:

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


donde para  $k$  suficientemente grande el *residuo* tenderá a cero.

→ El objetivo es hacer el residuo cero en el menor número de iteraciones posible.

Esto se puede conseguir utilizando técnicas de relajación, en las cuales la ecuación anterior se modifica multiplicando el *residuo* por un factor,  $\omega$ , tal que:

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

→  $\omega$ , se denomina factor de relajación.

### Valores que puede tomar $\omega$ :

→ Si  $0 < \omega < 1$ : tenemos un método de subrelajación. Puede servir para obtener la convergencia de algunos sistemas que no son convergentes con el método de Gauss-Seidel.

→  $1 < \omega < 2$  : tenemos un **método de sobrerrelajación**. Sirve para acelerar la convergencia de sistemas que son convergentes con el método de Gauss-Seidel.

→ En las técnicas de sobrerrelajación el valor óptimo de  $\omega$  está entre 1 y 2. Nunca puede ser mayor que 2 para evitar divergencias.

→ La técnica de sobrerrelajación es muy útil para sistemas de ecuaciones derivadas del análisis de ecuaciones diferenciales en derivadas parciales.



**Ejemplo:** Influencia de  $\omega$ .

$$\begin{bmatrix} -4 & 1 & 1 & 1 \\ 1 & -4 & 1 & 1 \\ 1 & 1 & -4 & 1 \\ 1 & 1 & 1 & -4 \end{bmatrix} x = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

cuya solución exacta es  $x_1=x_2=x_3=x_4=-1$ .

→ Empezando con un valor inicial  $[x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)}] = (0,0,0,0)$  obtenemos el siguiente número de iteraciones,  $k$ , para alcanzar un error  $< 1 \times 10^{-5}$ , en función de  $\omega$ .

$\omega \rightarrow$	1.0	1.1	1.2	<b>1.3</b>	1.4	1.5	1.6	1.7	1.8	1.9
$k \rightarrow$	24	18	13	<b>11</b>	14	18	24	35	55	>100

→ Para  $\omega = 1$  se recupera el método de Gauss-Seidel.