

# Computación Numérica

## Tema 2. Álgebra lineal numérica (III): métodos iterativos

**Irene Parada**

[irene.parada@upc.edu](mailto:irene.parada@upc.edu)

Departamento de Matemáticas

Universitat Politècnica de Catalunya · BarcelonaTech

11 de marzo de 2024

# Repaso

# Breve recordatorio del Tema 2.2

- ▶ Método de Cramer; número de operaciones.
- ▶ Método de Gauss; número de operaciones. Cálculo del rango y determinante.
- ▶ Estrategias de pivotamiento en el método de Gauss. Necesidad de pivotar.
- ▶ Condicionamiento de un sistema de ecuaciones lineales, vector residuo y acotamiento del error relativo en la solución calculada numéricamente.
- ▶ Métodos compactos de resolución de ecuaciones lineales: ejemplos y ventajas.
- ▶ Factorización LU; resolución de sistemas, determinante e inversa.
- ▶ Factorización LU: métodos, existencia y unicidad.
- ▶ Factorización LU de Cholesky.
- ▶ Factorización QR: propiedades, unicidad y comparativa de métodos.
- ▶ Método QR de Gram-Schmidt.
- ▶ Método QR de Householder.
- ▶ Método QR de Givens.
- ▶ Resolución de sistemas lineales sobredeterminados.

# **Sistemas de ecuaciones lineales**

## Métodos iterativos

Documentación de MATLAB - Iterative Methods for Linear Systems

# Métodos iterativos

Son métodos que construyen una **sucesión de vectores convergentes a la solución** exacta con un número finito de operaciones en cada iteración, si no fuera por los **errores** de redondeo acumulados y las posibles imprecisiones en el conocimiento inicial de la matriz  $A$  y el vector  $b$ .

# Métodos iterativos

Son métodos que construyen una **sucesión de vectores convergentes a la solución** exacta con un número finito de operaciones en cada iteración, si no fuera por los **errores** de redondeo acumulados y las posibles imprecisiones en el conocimiento inicial de la matriz  $A$  y el vector  $b$ .

- ▶ Se consideran adecuados para **sistemas lineales grandes** y, en particular, para sistemas con matrices dispersas (**sparse**).

# Métodos iterativos

Son métodos que construyen una **sucesión de vectores convergentes a la solución** exacta con un número finito de operaciones en cada iteración, si no fuera por los **errores** de redondeo acumulados y las posibles imprecisiones en el conocimiento inicial de la matriz  $A$  y el vector  $b$ .

- ▶ Se consideran adecuados para **sistemas lineales grandes** y, en particular, para sistemas con matrices dispersas (**sparse**).

Trabajaremos tres métodos:

- ▶ Método de **Jacobi**.
- ▶ Método de **Gauss-Seidel**.
- ▶ Métodos de **SOR**.

# Métodos iterativos: ejemplo

Este es un ejemplo introductorio de los **métodos iterativos de Jacobi y Gauss-Seidel**. Si se considera el sistema  $A\mathbf{x} = \mathbf{b}$ :

$$10x_1 - x_2 + 2x_3 = 6$$

$$-x_1 + 11x_2 - x_3 + 3x_4 = 25$$

$$2x_1 - x_2 + 10x_3 - x_4 = -11$$

$$3x_2 - x_3 + 8x_4 = 15$$



# Métodos iterativos: ejemplo

Este es un ejemplo introductorio de los **métodos iterativos de Jacobi y Gauss-Seidel**. Si se considera el sistema  $A\mathbf{x} = \mathbf{b}$ :

$$10x_1 - x_2 + 2x_3 = 6$$

$$-x_1 + 11x_2 - x_3 + 3x_4 = 25$$

$$2x_1 - x_2 + 10x_3 - x_4 = -11$$

$$3x_2 - x_3 + 8x_4 = 15$$

Tiene por solución  $\mathbf{x}^* = (1, 2, -1, 1)^t$ . Si se despeja  $x_i$  en cada ecuación  $i$ -ésima:

$$x_1 = \frac{1}{10}x_2 - \frac{1}{5}x_3 + \frac{3}{5}$$

$$x_2 = \frac{1}{11}x_1 + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11}$$

$$x_3 = -\frac{1}{5}x_1 + \frac{1}{10}x_2 + \frac{1}{10}x_4 - \frac{11}{10}$$

$$x_4 = -\frac{3}{8}x_2 + \frac{1}{8}x_3 + \frac{15}{8}$$

# Métodos iterativos: ejemplo

Este es un ejemplo introductorio de los **métodos iterativos de Jacobi y Gauss-Seidel**. Si se considera el sistema  $A\mathbf{x} = \mathbf{b}$ :

$$10x_1 - x_2 + 2x_3 = 6$$

$$-x_1 + 11x_2 - x_3 + 3x_4 = 25$$

$$2x_1 - x_2 + 10x_3 - x_4 = -11$$

$$3x_2 - x_3 + 8x_4 = 15$$

Tiene por solución  $\mathbf{x}^* = (1, 2, -1, 1)^t$ . Podemos iterar:

$$x_1^{(1)} = \frac{1}{10}x_2^{(0)} - \frac{1}{5}x_3^{(0)} + \frac{3}{5}$$

$$x_2^{(1)} = \frac{1}{11}x_1^{(0)} + \frac{1}{11}x_3^{(0)} - \frac{3}{11}x_4^{(0)} + \frac{25}{11}$$

$$x_3^{(1)} = -\frac{1}{5}x_1^{(0)} + \frac{1}{10}x_2^{(0)} + \frac{1}{10}x_4^{(0)} - \frac{11}{10}$$

$$x_4^{(1)} = -\frac{3}{8}x_2^{(0)} + \frac{1}{8}x_3^{(0)} + \frac{15}{8}$$

Se elige la aproximación inicial  $\mathbf{x}^{(0)} = \mathbf{0}$  y se genera una sucesión de vectores.

# Métodos iterativos: ejemplo

Este es un ejemplo introductorio de los **métodos iterativos de Jacobi y Gauss-Seidel**. Si se considera el sistema  $A\mathbf{x} = \mathbf{b}$ :

$$10x_1 - x_2 + 2x_3 = 6$$

$$-x_1 + 11x_2 - x_3 + 3x_4 = 25$$

$$2x_1 - x_2 + 10x_3 - x_4 = -11$$

$$3x_2 - x_3 + 8x_4 = 15$$

Tiene por solución  $\mathbf{x}^* = (1, 2, -1, 1)^t$ . El método iterativo para  $k \geq 0$  es:

$$x_1^{(k+1)} = \frac{1}{10}x_2^{(k)} - \frac{1}{5}x_3^{(k)} + \frac{3}{5}$$

$$x_2^{(k+1)} = \frac{1}{11}x_1^{(k)} + \frac{1}{11}x_3^{(k)} - \frac{3}{11}x_4^{(k)} + \frac{25}{11} \quad \text{Jacobi}$$

$$x_3^{(k+1)} = -\frac{1}{5}x_1^{(k)} + \frac{1}{10}x_2^{(k)} + \frac{1}{10}x_4^{(k)} - \frac{11}{10}$$

$$x_4^{(k+1)} = -\frac{3}{8}x_2^{(k)} + \frac{1}{8}x_3^{(k)} + \frac{15}{8}$$

Se elige la aproximación inicial  $\mathbf{x}^{(0)} = \mathbf{0}$  y se genera una sucesión de vectores.

# Métodos iterativos: ejemplo

Este es un ejemplo introductorio de los [métodos iterativos de Jacobi y Gauss-Seidel](#). Si se considera el sistema  $A\mathbf{x} = \mathbf{b}$ :

$$\begin{aligned} 10x_1 - x_2 + 2x_3 &= 6 \\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25 \\ 2x_1 - x_2 + 10x_3 - x_4 &= -11 \\ 3x_2 - x_3 + 8x_4 &= 15 \end{aligned}$$

La siguiente tabla proporciona las diez primeras iteraciones:

$k$	0	1	2	3	...	8	9	10
$x_1^{(k)}$	0.0000	0.6000	1.0473	0.9326	...	1.0006	0.9997	1.0001
$x_2^{(k)}$	0.0000	2.2727	1.7159	2.0533	...	1.9987	2.0004	1.9998
$x_3^{(k)}$	0.0000	-1.1000	-0.8052	-1.0493	...	-0.9990	-1.0004	-0.9998
$x_4^{(k)}$	0.0000	1.8750	0.8852	1.1309	...	0.9989	1.0006	0.9998

# Métodos iterativos: ejemplo

Este es un ejemplo introductorio de los **métodos iterativos de Jacobi y Gauss-Seidel**. Si se considera el sistema  $A\mathbf{x} = \mathbf{b}$ :

$$\begin{aligned} 10x_1 - x_2 + 2x_3 &= 6 \\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25 \\ 2x_1 - x_2 + 10x_3 - x_4 &= -11 \\ 3x_2 - x_3 + 8x_4 &= 15 \end{aligned}$$

La siguiente tabla proporciona las diez primeras iteraciones:

$k$	0	1	2	3	...	8	9	10
$x_1^{(k)}$	0.0000	0.6000	1.0473	0.9326	...	1.0006	0.9997	1.0001
$x_2^{(k)}$	0.0000	2.2727	1.7159	2.0533	...	1.9987	2.0004	1.9998
$x_3^{(k)}$	0.0000	-1.1000	-0.8052	-1.0493	...	-0.9990	-1.0004	-0.9998
$x_4^{(k)}$	0.0000	1.8750	0.8852	1.1309	...	0.9989	1.0006	0.9998

Criterio para parar:  $\frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_\infty}{\|\mathbf{x}^{(k+1)}\|_\infty} < 10^{-3}$ ; obtenemos  $e_a = \|\mathbf{x}^{(10)} - \mathbf{x}^*\|_\infty = 0.0002$ .

# Métodos iterativos: ejemplo

Este es un ejemplo introductorio de los **métodos iterativos de Jacobi y Gauss-Seidel**. Si se considera el sistema  $A\mathbf{x} = \mathbf{b}$ :

$$10x_1 - x_2 + 2x_3 = 6$$

$$-x_1 + 11x_2 - x_3 + 3x_4 = 25$$

$$2x_1 - x_2 + 10x_3 - x_4 = -11$$

$$3x_2 - x_3 + 8x_4 = 15$$

Tiene por solución  $\mathbf{x}^* = (1, 2, -1, 1)^t$ . El método iterativo para  $k \geq 0$  es:

$$x_1^{(k+1)} = \frac{1}{10}x_2^{(k)} - \frac{1}{5}x_3^{(k)} + \frac{3}{5}$$

$$x_2^{(k+1)} = \frac{1}{11}x_1^{(k)} + \frac{1}{11}x_3^{(k)} - \frac{3}{11}x_4^{(k)} + \frac{25}{11} \quad \text{Jacobi}$$

$$x_3^{(k+1)} = -\frac{1}{5}x_1^{(k)} + \frac{1}{10}x_2^{(k)} + \frac{1}{10}x_4^{(k)} - \frac{11}{10}$$

$$x_4^{(k+1)} = -\frac{3}{8}x_2^{(k)} + \frac{1}{8}x_3^{(k)} + \frac{15}{8}$$

Se elige la aproximación inicial  $\mathbf{x}^{(0)} = \mathbf{0}$  y se genera una sucesión de vectores.

# Métodos iterativos: ejemplo

Este es un ejemplo introductorio de los **métodos iterativos de Jacobi y Gauss-Seidel**. Si se considera el sistema  $A\mathbf{x} = \mathbf{b}$ :

$$10x_1 - x_2 + 2x_3 = 6$$

$$-x_1 + 11x_2 - x_3 + 3x_4 = 25$$

$$2x_1 - x_2 + 10x_3 - x_4 = -11$$

$$3x_2 - x_3 + 8x_4 = 15$$

Tiene por solución  $\mathbf{x}^* = (1, 2, -1, 1)^t$ . Usando ahora toda la información nueva:

$$x_1^{(k+1)} = \frac{1}{10}x_2^{(k)} - \frac{1}{5}x_3^{(k)} + \frac{3}{5}$$

$$x_2^{(k+1)} = \frac{1}{11}x_1^{(k+1)} + \frac{1}{11}x_3^{(k)} - \frac{3}{11}x_4^{(k)} + \frac{25}{11}$$

**Gauss-Seidel**

$$x_3^{(k+1)} = -\frac{1}{5}x_1^{(k+1)} + \frac{1}{10}x_2^{(k+1)} + \frac{1}{10}x_4^{(k)} - \frac{11}{10}$$

$$x_4^{(k+1)} = -\frac{3}{8}x_2^{(k+1)} + \frac{1}{8}x_3^{(k+1)} + \frac{15}{8}$$

Se elige la aproximación inicial  $\mathbf{x}^{(0)} = \mathbf{0}$  y se genera una sucesión de vectores.

# Métodos iterativos: ejemplo

Este es un ejemplo introductorio de los [métodos iterativos de Jacobi y Gauss-Seidel](#). Si se considera el sistema  $A\mathbf{x} = \mathbf{b}$ :

$$\begin{aligned} 10x_1 - x_2 + 2x_3 &= 6 \\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25 \\ 2x_1 - x_2 + 10x_3 - x_4 &= -11 \\ 3x_2 - x_3 + 8x_4 &= 15 \end{aligned}$$

Tiene por solución  $\mathbf{x}^* = (1, 2, -1, 1)^t$ . Usando ahora toda la información nueva:

$k$	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.6000	1.0300	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.0370	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	-0.9873	-1.0140	-1.0025	-1.0003	-1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000



# Métodos iterativos: ejemplo

Este es un ejemplo introductorio de los [métodos iterativos de Jacobi y Gauss-Seidel](#). Si se considera el sistema  $A\mathbf{x} = \mathbf{b}$ :

$$\begin{aligned} 10x_1 - x_2 + 2x_3 &= 6 \\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25 \\ 2x_1 - x_2 + 10x_3 - x_4 &= -11 \\ 3x_2 - x_3 + 8x_4 &= 15 \end{aligned}$$

Tiene por solución  $\mathbf{x}^* = (1, 2, -1, 1)^t$ . Usando ahora toda la información nueva:

$k$	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.6000	1.0300	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.0370	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	-0.9873	-1.0140	-1.0025	-1.0003	-1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000

Se ha detenido el cálculo tras cinco iteraciones debido a:  $\frac{\|\mathbf{x}^{(5)} - \mathbf{x}^{(4)}\|_\infty}{\|\mathbf{x}^{(5)}\|_\infty} \leq 4 \cdot 10^{-4}$ .

# Métodos iterativos estacionarios

Transformamos el sistema lineal  $A\mathbf{x} = \mathbf{b}$  en  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$ . Ambos sistemas deben ser consistentes (= que tienen una o más soluciones).

$$\left. \begin{array}{l} A\mathbf{x} = \mathbf{b} \\ A\mathbf{x}^* = \mathbf{b} \end{array} \right\} \Longleftrightarrow \left\{ \begin{array}{l} \mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{c} \\ \mathbf{x}^* = B\mathbf{x}^* + \mathbf{c} \end{array} \right.$$

# Métodos iterativos estacionarios

Transformamos el sistema lineal  $A\mathbf{x} = \mathbf{b}$  en  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$ . Ambos sistemas deben ser consistentes (= que tienen una o más soluciones).

$$\left. \begin{array}{l} A\mathbf{x} = \mathbf{b} \\ A\mathbf{x}^* = \mathbf{b} \end{array} \right\} \Longleftrightarrow \begin{cases} \mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{c} \\ \mathbf{x}^* = B\mathbf{x}^* + \mathbf{c} \end{cases}$$

- **Algoritmo:** Partimos de  $\mathbf{x}^{(0)}$  arbitrario, y generamos la sucesión de vectores  $\mathbf{x}^{(k)}$  a partir de la recurrencia  $\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{c}$ .

# Métodos iterativos estacionarios

Transformamos el sistema lineal  $A\mathbf{x} = \mathbf{b}$  en  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$ . Ambos sistemas deben ser consistentes (= que tienen una o más soluciones).

$$\left. \begin{array}{l} A\mathbf{x} = \mathbf{b} \\ A\mathbf{x}^* = \mathbf{b} \end{array} \right\} \Longleftrightarrow \begin{cases} \mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{c} \\ \mathbf{x}^* = B\mathbf{x}^* + \mathbf{c} \end{cases}$$

- ▶ **Algoritmo:** Partimos de  $\mathbf{x}^{(0)}$  arbitrario, y generamos la sucesión de vectores  $\mathbf{x}^{(k)}$  a partir de la recurrencia  $\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{c}$ .
- ▶ **Coste computacional:** En cada iteración hay  $n^2$  sumas y  $n^2$  productos; después de  $k$  iteraciones el total es  $2n^2k$ .

# Métodos iterativos estacionarios

Transformamos el sistema lineal  $A\mathbf{x} = \mathbf{b}$  en  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$ . Ambos sistemas deben ser consistentes (= que tienen una o más soluciones).

$$\left. \begin{array}{l} A\mathbf{x} = \mathbf{b} \\ A\mathbf{x}^* = \mathbf{b} \end{array} \right\} \Longleftrightarrow \begin{cases} \mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{c} \\ \mathbf{x}^* = B\mathbf{x}^* + \mathbf{c} \end{cases}$$

- ▶ **Algoritmo:** Partimos de  $\mathbf{x}^{(0)}$  arbitrario, y generamos la sucesión de vectores  $\mathbf{x}^{(k)}$  a partir de la recurrencia  $\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{c}$ .
- ▶ **Coste computacional:** En cada iteración hay  $n^2$  sumas y  $n^2$  productos; después de  $k$  iteraciones el total es  $2n^2k$ .
- ▶ **Convergencia:**  
¿La sucesión de vectores  $\mathbf{x}^{(k)}$  converge a  $\mathbf{x}^*$ , solución de  $A\mathbf{x} = \mathbf{b}$ ?

# Teoremas de convergencia

Sea  $\mathbf{x}^*$  la solución del problema original  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente.

# Teoremas de convergencia

Sea  $\mathbf{x}^*$  la solución del problema original  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente.

► **Teorema:** Si  $A$  es no singular, definimos el **vector residuo**  $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$ .  
Entonces:

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}^* \iff \lim_{k \rightarrow \infty} \mathbf{r}^{(k)} = \mathbf{0}.$$

# Teoremas de convergencia

Sea  $\mathbf{x}^*$  la solución del problema original  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente.

► **Teorema:** Si  $A$  es no singular, definimos el **vector residuo**  $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$ .  
Entonces:

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}^* \iff \lim_{k \rightarrow \infty} \mathbf{r}^{(k)} = \mathbf{0}.$$

► **Teorema:** El método iterativo  $\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{c}$  es **convergente** a la solución  $\mathbf{x}^*$  para cualquier  $\mathbf{x}^{(0)}$  si y solo si  $\rho(B) < 1$ .



# Teoremas de convergencia

Sea  $\mathbf{x}^*$  la solución del problema original  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente.

► **Teorema:** Si  $A$  es no singular, definimos el **vector residuo**  $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$ .  
Entonces:

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}^* \iff \lim_{k \rightarrow \infty} \mathbf{r}^{(k)} = \mathbf{0}.$$

► **Teorema:** El método iterativo  $\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{c}$  es **convergente** a la solución  $\mathbf{x}^*$  para cualquier  $\mathbf{x}^{(0)}$  si y solo si  $\rho(B) < 1$ .

↑  
**Radio espectral**

# Razón de convergencia

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

# Razón de convergencia

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

## Razón de convergencia

$$\begin{aligned} \text{► } \mathbf{x}^{(k)} - \mathbf{x}^* &= B (\mathbf{x}^{(k-1)} - \mathbf{x}^*) = \dots = B^k (\mathbf{x}^{(0)} - \mathbf{x}^*) \Rightarrow \\ \|\mathbf{x}^{(k)} - \mathbf{x}^*\| &\leq \|B\|^k \|\mathbf{x}^{(0)} - \mathbf{x}^*\| \end{aligned}$$

# Razón de convergencia

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

## Razón de convergencia

- ▶  $\mathbf{x}^{(k)} - \mathbf{x}^* = B (\mathbf{x}^{(k-1)} - \mathbf{x}^*) = \dots = B^k (\mathbf{x}^{(0)} - \mathbf{x}^*) \Rightarrow$   
 $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \|B\|^k \|\mathbf{x}^{(0)} - \mathbf{x}^*\|$
- ▶ Se define el **factor de convergencia asintótico**  $\alpha = \lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\|^{1/k}$ .

# Razón de convergencia

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

## Razón de convergencia

- ▶  $\mathbf{x}^{(k)} - \mathbf{x}^* = B (\mathbf{x}^{(k-1)} - \mathbf{x}^*) = \dots = B^k (\mathbf{x}^{(0)} - \mathbf{x}^*) \Rightarrow$   
 $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \|B\|^k \|\mathbf{x}^{(0)} - \mathbf{x}^*\|$
- ▶ Se define el **factor de convergencia asintótico**  $\alpha = \lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\|^{1/k}$ .  
Cuanto más pequeño sea  $\alpha$ , menores iteraciones necesarias.

# Razón de convergencia

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

## Razón de convergencia

- ▶  $\mathbf{x}^{(k)} - \mathbf{x}^* = B (\mathbf{x}^{(k-1)} - \mathbf{x}^*) = \dots = B^k (\mathbf{x}^{(0)} - \mathbf{x}^*) \Rightarrow$   
 $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \|B\|^k \|\mathbf{x}^{(0)} - \mathbf{x}^*\|$
- ▶ Se define el **factor de convergencia asintótico**  $\alpha = \lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\|^{1/k}$ .  
Cuanto más pequeño sea  $\alpha$ , menores iteraciones necesarias.
- ▶  $\alpha \leq \rho(B)$ .

# Razón de convergencia

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

## Razón de convergencia

- ▶  $\mathbf{x}^{(k)} - \mathbf{x}^* = B (\mathbf{x}^{(k-1)} - \mathbf{x}^*) = \dots = B^k (\mathbf{x}^{(0)} - \mathbf{x}^*) \Rightarrow$   
 $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \|B\|^k \|\mathbf{x}^{(0)} - \mathbf{x}^*\|$
- ▶ Se define el **factor de convergencia asintótico**  $\alpha = \lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\|^{1/k}$ .  
Cuanto más pequeño sea  $\alpha$ , menores iteraciones necesarias.
- ▶  $\alpha \leq \rho(B)$ .
- ▶ La **velocidad de convergencia** es  $R = -\log(\rho(B))$ .

# Cotas del error

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.



# Cotas del error

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

Cotas del error: Sea  $\beta = \|B\| < 1$ .

# Cotas del error

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

Cotas del error: Sea  $\beta = \|B\| < 1$ .

► 
$$\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \frac{\beta}{1-\beta} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|.$$

# Cotas del error

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

Cotas del error: Sea  $\beta = \|B\| < 1$ .

►  $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \frac{\beta}{1-\beta} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|.$

Si nuestro **criterio de parada** es  $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| \leq \varepsilon$ , el error absoluto está acotado por:  $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \frac{\varepsilon\beta}{1-\beta}.$

# Cotas del error

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

Cotas del error: Sea  $\beta = \|B\| < 1$ .

►  $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \frac{\beta}{1-\beta} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|.$

⚠ En general, para métodos iterativos este criterio de parada **no** es una buena idea.

Si nuestro **criterio de parada** es  $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| \leq \varepsilon$ , el error absoluto está acotado por:  $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \frac{\varepsilon\beta}{1-\beta}.$

# Cotas del error

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

Cotas del error: Sea  $\beta = \|B\| < 1$ .

►  $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \frac{\beta}{1-\beta} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|.$

Si nuestro **criterio de parada** es  $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| \leq \varepsilon$ , el error absoluto está acotado por:  $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \frac{\varepsilon\beta}{1-\beta}.$

►  $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \frac{\beta^k}{1-\beta} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|.$

⚠ En general, para métodos iterativos este criterio de parada **no** es una buena idea.

# Cotas del error

Sea  $\mathbf{x}^*$  la solución del problema  $A\mathbf{x} = \mathbf{b}$  y sea  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$  una formulación equivalente. Consideramos una norma vectorial y la norma matricial asociada.

Cotas del error: Sea  $\beta = \|B\| < 1$ .

►  $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \frac{\beta}{1-\beta} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|.$

⚠ En general, para métodos iterativos este criterio de parada **no** es una buena idea.

Si nuestro **criterio de parada** es  $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| \leq \varepsilon$ , el error absoluto está acotado por:  $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \frac{\varepsilon\beta}{1-\beta}.$

►  $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \frac{\beta^k}{1-\beta} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|.$

Nos permite **estimar el número de iteraciones** para un error absoluto prefijado.

# Métodos iterativos de Jacobi y Gauss-Seidel

Para convertir  $A\mathbf{x} = \mathbf{b}$  en un sistema de la forma  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$ , expresamos la matriz  $A = (a_{ij})$  como la suma de tres matrices:  $A = L + D + U$  tal que

$$\underbrace{\begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn-1} & 0 \end{pmatrix}}_L + \underbrace{\begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{nn} \end{pmatrix}}_D + \underbrace{\begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1n} \\ 0 & \dots & 0 & 0 \end{pmatrix}}_U$$

# Métodos iterativos de Jacobi y Gauss-Seidel

Para convertir  $A\mathbf{x} = \mathbf{b}$  en un sistema de la forma  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$ , expresamos la matriz  $A = (a_{ij})$  como la suma de tres matrices:  $A = L + D + U$  tal que

$$\underbrace{\begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn-1} & 0 \end{pmatrix}}_L + \underbrace{\begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{nn} \end{pmatrix}}_D + \underbrace{\begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1n} \\ 0 & \dots & 0 & 0 \end{pmatrix}}_U$$

► Método de Jacobi:



# Métodos iterativos de Jacobi y Gauss-Seidel

Para convertir  $A\mathbf{x} = \mathbf{b}$  en un sistema de la forma  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$ , expresamos la matriz  $A = (a_{ij})$  como la suma de tres matrices:  $A = L + D + U$  tal que

$$\underbrace{\begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn-1} & 0 \end{pmatrix}}_L + \underbrace{\begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{nn} \end{pmatrix}}_D + \underbrace{\begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1n} \\ 0 & \dots & 0 & 0 \end{pmatrix}}_U$$

► **Método de Jacobi:**  $D\mathbf{x} = \mathbf{b} - (L + U)\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_J\mathbf{x}^{(k)} + \mathbf{c}_J$ ,  
con **matriz de iteración**  $B_J = -D^{-1}(L + U)$  y vector  $\mathbf{c}_J = D^{-1}\mathbf{b}$ .

# Métodos iterativos de Jacobi y Gauss-Seidel

Para convertir  $A\mathbf{x} = \mathbf{b}$  en un sistema de la forma  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$ , expresamos la matriz  $A = (a_{ij})$  como la suma de tres matrices:  $A = L + D + U$  tal que

$$\underbrace{\begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn-1} & 0 \end{pmatrix}}_L + \underbrace{\begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{nn} \end{pmatrix}}_D + \underbrace{\begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1n} \\ 0 & \dots & 0 & 0 \end{pmatrix}}_U$$

► **Método de Jacobi:**  $D\mathbf{x} = \mathbf{b} - (L + U)\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_J\mathbf{x}^{(k)} + \mathbf{c}_J$ ,  
con **matriz de iteración**  $B_J = -D^{-1}(L + U)$  y vector  $\mathbf{c}_J = D^{-1}\mathbf{b}$ .

Converge  $\forall \mathbf{x}^{(0)}$  si  $A$  es estrictamente diagonal dominante.

# Métodos iterativos de Jacobi y Gauss-Seidel

Para convertir  $A\mathbf{x} = \mathbf{b}$  en un sistema de la forma  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$ , expresamos la matriz  $A = (a_{ij})$  como la suma de tres matrices:  $A = L + D + U$  tal que

$$\underbrace{\begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn-1} & 0 \end{pmatrix}}_L + \underbrace{\begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{nn} \end{pmatrix}}_D + \underbrace{\begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1n} \\ 0 & \dots & 0 & 0 \end{pmatrix}}_U$$

- **Método de Jacobi:**  $D\mathbf{x} = \mathbf{b} - (L + U)\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_J\mathbf{x}^{(k)} + \mathbf{c}_J$ ,  
con **matriz de iteración**  $B_J = -D^{-1}(L + U)$  y vector  $\mathbf{c}_J = D^{-1}\mathbf{b}$ .

Converge  $\forall \mathbf{x}^{(0)}$  si  $A$  es estrictamente diagonal dominante.

- **Método de Gauss-Seidel:**  $(L + D)\mathbf{x} = \mathbf{b} - U\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_G\mathbf{x}^{(k)} + \mathbf{c}_G$ ,  
con **matriz de iteración**  $B_G = -(L + D)^{-1}U$  y vector  $\mathbf{c}_G = (L + D)^{-1}\mathbf{b}$ .

# Métodos iterativos de Jacobi y Gauss-Seidel

Para convertir  $A\mathbf{x} = \mathbf{b}$  en un sistema de la forma  $\mathbf{x} = B\mathbf{x} + \mathbf{c}$ , expresamos la matriz  $A = (a_{ij})$  como la suma de tres matrices:  $A = L + D + U$  tal que

$$\underbrace{\begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn-1} & 0 \end{pmatrix}}_L + \underbrace{\begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{nn} \end{pmatrix}}_D + \underbrace{\begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1n} \\ 0 & \dots & 0 & 0 \end{pmatrix}}_U$$

► **Método de Jacobi:**  $D\mathbf{x} = \mathbf{b} - (L + U)\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_J\mathbf{x}^{(k)} + \mathbf{c}_J$ ,  
con **matriz de iteración**  $B_J = -D^{-1}(L + U)$  y vector  $\mathbf{c}_J = D^{-1}\mathbf{b}$ .

Converge  $\forall \mathbf{x}^{(0)}$  si  $A$  es estrictamente diagonal dominante.

► **Método de Gauss-Seidel:**  $(L + D)\mathbf{x} = \mathbf{b} - U\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_G\mathbf{x}^{(k)} + \mathbf{c}_G$ ,  
con **matriz de iteración**  $B_G = -(L + D)^{-1}U$  y vector  $\mathbf{c}_G = (L + D)^{-1}\mathbf{b}$ .

Converge  $\forall \mathbf{x}^{(0)}$  si  $A$  es estrictamente diagonal dominante o sim. definida positiva.

# Métodos de sobrerrelajación sucesiva (SOR)

Son una generalización de los dos métodos estudiados. Si sumamos y restamos  $x_i^{(k)}$  en la expresión del método de Jacobi:

$$x_i^{(k+1)} = x_i^{(k)} + \underbrace{\frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^n a_{ij} x_j^{(k)} \right)}_{\text{corrección}}, k \geq 0.$$

# Métodos de sobrerelajación sucesiva (SOR)

Son una generalización de los dos métodos estudiados. Si sumamos y restamos  $x_i^{(k)}$  en la expresión del método de Jacobi:

$$x_i^{(k+1)} = x_i^{(k)} + \underbrace{\frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^n a_{ij} x_j^{(k)} \right)}_{\text{corrección}}, k \geq 0.$$

El método de relajación consiste en **multiplicar la corrección por** un parámetro  $\omega$ , **parámetro de relajación**, para acelerar la convergencia.

# Métodos de sobrerrelajación sucesiva (SOR)

Son una generalización de los dos métodos estudiados. Si sumamos y restamos  $x_i^{(k)}$  en la expresión del método de Jacobi:

$$x_i^{(k+1)} = x_i^{(k)} + \underbrace{\frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^n a_{ij} x_j^{(k)} \right)}_{\text{corrección}}, k \geq 0.$$

El método de relajación consiste en **multiplicar la corrección por un parámetro  $\omega$ , parámetro de relajación**, para acelerar la convergencia.

- ▶  $\omega > 1$ : **sobrerrelajación**.
- ▶  $\omega < 1$ : **subrelajación**.

# Métodos de sobrerrelajación sucesiva (SOR)

Son una generalización de los dos métodos estudiados. Si sumamos y restamos  $x_i^{(k)}$  en la expresión del método de Jacobi:

$$x_i^{(k+1)} = x_i^{(k)} + \underbrace{\frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^n a_{ij} x_j^{(k)} \right)}_{\text{corrección}}, k \geq 0.$$

El método de relajación consiste en **multiplicar la corrección por un parámetro  $\omega$ , parámetro de relajación**, para acelerar la convergencia.

►  $\omega > 1$ : **sobrerrelajación**.

►  $\omega < 1$ : **subrelajación**.

$\omega = 1$ : **método original**



# Métodos de sobrerrelajación sucesiva (SOR)

Son una generalización de los dos métodos estudiados. Si sumamos y restamos  $x_i^{(k)}$  en la expresión del método de Jacobi:

$$x_i^{(k+1)} = x_i^{(k)} + \underbrace{\frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^n a_{ij} x_j^{(k)} \right)}_{\text{corrección}}, k \geq 0.$$

El método de relajación consiste en **multiplicar la corrección por un parámetro  $\omega$ , parámetro de relajación**, para acelerar la convergencia.

►  $\omega > 1$ : **sobrerrelajación**.

$\omega = 1$ : **método original**

►  $\omega < 1$ : **subrelajación**.

El nuevo  $x_i^{(k+1)}$  está generado por  $x_i^{(k)}$  y por el calculado en este paso  $\tilde{x}_i^{(k+1)}$ :

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \omega\tilde{x}_i^{(k+1)}.$$

# Métodos de sobrerrelajación: variante Jacobi

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :

# Métodos de sobrerrelajación: variante Jacobi

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :

- **Método de Jacobi:**  $D\mathbf{x} = \mathbf{b} - (L + U)\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_J\mathbf{x}^{(k)} + \mathbf{c}_J$ ,  
con matriz de iteración  $B_J = -D^{-1}(L + U)$  y vector  $\mathbf{c}_J = D^{-1}\mathbf{b}$ .

# Métodos de sobrerrelajación: variante Jacobi

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :

- **Método de Jacobi:**  $D\mathbf{x} = \mathbf{b} - (L + U)\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_J\mathbf{x}^{(k)} + \mathbf{c}_J$ ,  
con matriz de iteración  $B_J = -D^{-1}(L + U)$  y vector  $\mathbf{c}_J = D^{-1}\mathbf{b}$ .

$$\tilde{\mathbf{x}}^{(k+1)} = -D^{-1}(L + U)\mathbf{x}^k + D^{-1}\mathbf{b}.$$

# Métodos de sobrerrelajación: variante Jacobi

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :

- **Método de Jacobi:**  $D\mathbf{x} = \mathbf{b} - (L + U)\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_J\mathbf{x}^{(k)} + \mathbf{c}_J$ ,  
con matriz de iteración  $B_J = -D^{-1}(L + U)$  y vector  $\mathbf{c}_J = D^{-1}\mathbf{b}$ .

$$\tilde{\mathbf{x}}^{(k+1)} = -D^{-1}(L + U)\mathbf{x}^k + D^{-1}\mathbf{b}.$$

- **SOR:** El nuevo  $\mathbf{x}^{(k+1)}$  está generado por el anterior  $\mathbf{x}^{(k)}$  y el calculado  $\tilde{\mathbf{x}}^{(k+1)}$ :  
$$\mathbf{x}^{(k+1)} = (1 - \omega)\mathbf{x}^{(k)} + \omega\tilde{\mathbf{x}}^{(k+1)}.$$

# Métodos de sobrerrelajación: variante Jacobi

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :

- **Método de Jacobi:**  $D\mathbf{x} = \mathbf{b} - (L + U)\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_J\mathbf{x}^{(k)} + \mathbf{c}_J$ ,  
con matriz de iteración  $B_J = -D^{-1}(L + U)$  y vector  $\mathbf{c}_J = D^{-1}\mathbf{b}$ .

$$\tilde{\mathbf{x}}^{(k+1)} = -D^{-1}(L + U)\mathbf{x}^k + D^{-1}\mathbf{b}.$$

- **SOR:** El nuevo  $\mathbf{x}^{(k+1)}$  está generado por el anterior  $\mathbf{x}^{(k)}$  y el calculado  $\tilde{\mathbf{x}}^{(k+1)}$ :

$$\mathbf{x}^{(k+1)} = (1 - \omega)\mathbf{x}^{(k)} + \omega\tilde{\mathbf{x}}^{(k+1)}.$$

$$\mathbf{x}^{(k+1)} = B_\omega\mathbf{x}^{(k)} + \mathbf{c}_\omega$$

# Métodos de sobrerrelajación: variante Jacobi

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :

- **Método de Jacobi:**  $D\mathbf{x} = \mathbf{b} - (L + U)\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_J\mathbf{x}^{(k)} + \mathbf{c}_J$ ,  
con matriz de iteración  $B_J = -D^{-1}(L + U)$  y vector  $\mathbf{c}_J = D^{-1}\mathbf{b}$ .

$$\tilde{\mathbf{x}}^{(k+1)} = -D^{-1}(L + U)\mathbf{x}^k + D^{-1}\mathbf{b}.$$

- **SOR:** El nuevo  $\mathbf{x}^{(k+1)}$  está generado por el anterior  $\mathbf{x}^{(k)}$  y el calculado  $\tilde{\mathbf{x}}^{(k+1)}$ :  
$$\mathbf{x}^{(k+1)} = (1 - \omega)\mathbf{x}^{(k)} + \omega\tilde{\mathbf{x}}^{(k+1)}.$$

$$\mathbf{x}^{(k+1)} = B_\omega\mathbf{x}^{(k)} + \mathbf{c}_\omega$$

- $C = D^{-1}$

- $B_\omega = C((1 - \omega)D - \omega(L + U))$

- $\mathbf{c}_\omega = \omega C\mathbf{b}$

Matriz auxiliar.

Matriz de iteración.

Vector de iteración.

# Métodos de sobrerrelajación: variante Gauss-Seidel

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :



# Métodos de sobrerrelajación: variante Gauss-Seidel

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :

- **Método de Gauss-Seidel:**  $(L + D)\mathbf{x} = \mathbf{b} - U\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_G\mathbf{x}^{(k)} + \mathbf{c}_G$ ,  
con matriz de iteración  $B_G = -(L + D)^{-1}U$  y vector  $\mathbf{c}_G = (L + D)^{-1}\mathbf{b}$ .

# Métodos de sobrerelajación: variante Gauss-Seidel

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :

- **Método de Gauss-Seidel:**  $(L + D)\mathbf{x} = \mathbf{b} - U\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_G\mathbf{x}^{(k)} + \mathbf{c}_G$ ,  
con matriz de iteración  $B_G = -(L + D)^{-1}U$  y vector  $\mathbf{c}_G = (L + D)^{-1}\mathbf{b}$ .

$$\tilde{\mathbf{x}}^{(k+1)} = -(L + D)^{-1}U\mathbf{x}^k + (L + D)^{-1}\mathbf{b}.$$

# Métodos de sobrerrelajación: variante Gauss-Seidel

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :

- **Método de Gauss-Seidel:**  $(L + D)\mathbf{x} = \mathbf{b} - U\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_G\mathbf{x}^{(k)} + \mathbf{c}_G$ ,  
con matriz de iteración  $B_G = -(L + D)^{-1}U$  y vector  $\mathbf{c}_G = (L + D)^{-1}\mathbf{b}$ .

$$\tilde{\mathbf{x}}^{(k+1)} = -(L + D)^{-1}U\mathbf{x}^k + (L + D)^{-1}\mathbf{b}.$$

- **SOR:** El nuevo  $\mathbf{x}^{(k+1)}$  está generado por el anterior  $\mathbf{x}^{(k)}$  y el calculado  $\tilde{\mathbf{x}}^{(k+1)}$ :  
$$\mathbf{x}^{(k+1)} = (1 - \omega)\mathbf{x}^{(k)} + \omega\tilde{\mathbf{x}}^{(k+1)}.$$

# Métodos de sobrerrelajación: variante Gauss-Seidel

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :

- **Método de Gauss-Seidel:**  $(L + D)\mathbf{x} = \mathbf{b} - U\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_G\mathbf{x}^{(k)} + \mathbf{c}_G$ ,  
con matriz de iteración  $B_G = -(L + D)^{-1}U$  y vector  $\mathbf{c}_G = (L + D)^{-1}\mathbf{b}$ .

$$\tilde{\mathbf{x}}^{(k+1)} = -(L + D)^{-1}U\mathbf{x}^{(k)} + (L + D)^{-1}\mathbf{b}.$$

- **SOR:** El nuevo  $\mathbf{x}^{(k+1)}$  está generado por el anterior  $\mathbf{x}^{(k)}$  y el calculado  $\tilde{\mathbf{x}}^{(k+1)}$ :

$$\mathbf{x}^{(k+1)} = (1 - \omega)\mathbf{x}^{(k)} + \omega\tilde{\mathbf{x}}^{(k+1)}.$$

$$\mathbf{x}^{(k+1)} = B_\omega\mathbf{x}^{(k)} + \mathbf{c}_\omega$$

# Métodos de sobrerrelajación: variante Gauss-Seidel

En términos matriciales para el sistema  $A\mathbf{x} = \mathbf{b}$  y  $A = L + D + U$ :

- **Método de Gauss-Seidel:**  $(L + D)\mathbf{x} = \mathbf{b} - U\mathbf{x} \implies \mathbf{x}^{(k+1)} = B_G\mathbf{x}^{(k)} + \mathbf{c}_G$ ,  
con matriz de iteración  $B_G = -(L + D)^{-1}U$  y vector  $\mathbf{c}_G = (L + D)^{-1}\mathbf{b}$ .

$$\tilde{\mathbf{x}}^{(k+1)} = -(L + D)^{-1}U\mathbf{x}^k + (L + D)^{-1}\mathbf{b}.$$

- **SOR:** El nuevo  $\mathbf{x}^{(k+1)}$  está generado por el anterior  $\mathbf{x}^{(k)}$  y el calculado  $\tilde{\mathbf{x}}^{(k+1)}$ :  
$$\mathbf{x}^{(k+1)} = (1 - \omega)\mathbf{x}^{(k)} + \omega\tilde{\mathbf{x}}^{(k+1)}.$$

$$\mathbf{x}^{(k+1)} = B_\omega\mathbf{x}^{(k)} + \mathbf{c}_\omega$$

- $C = (D + \omega L)^{-1}$

- $B_\omega = C((1 - \omega)D - \omega U)$

- $\mathbf{c}_\omega = \omega C\mathbf{b}$

Matriz auxiliar.

Matriz de iteración.

Vector de iteración.

# Métodos de sobrerrelajación: convergencia

No hay una regla general para elegir el valor óptimo del factor de relajación  $\omega$ .

# Métodos de sobrerrelajación: convergencia

**Teorema:** Sea  $A$  **simétrica definida positiva y tridiagonal en bloques:**

$$A = \begin{pmatrix} D_1 & U_1 & 0 & \dots & 0 \\ L_2 & D_2 & U_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & L_{n-1} & D_{n-1} & U_{n-1} \\ 0 & \dots & 0 & L_n & D_n \end{pmatrix}$$

donde  $D_i$ ,  $i = 1, \dots, n$  son submatrices diagonales,  $U_i$ ,  $L_i$ , submatrices cualesquiera que satisfacen  $L_{i+1} = U_i^T$ ,  $i = 1, \dots, n - 1$ .

# Métodos de sobrerrelajación: convergencia

**Teorema:** Sea  $A$  **simétrica definida positiva y tridiagonal en bloques:**

$$A = \begin{pmatrix} D_1 & U_1 & 0 & \dots & 0 \\ L_2 & D_2 & U_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & L_{n-1} & D_{n-1} & U_{n-1} \\ 0 & \dots & 0 & L_n & D_n \end{pmatrix}$$

donde  $D_i$ ,  $i = 1, \dots, n$  son submatrices diagonales,  $U_i$ ,  $L_i$ , submatrices cualesquiera que satisfacen  $L_{i+1} = U_i^T$ ,  $i = 1, \dots, n-1$ .

Entonces:

- Los radios espectrales de las matrices de iteración cumplen  $\rho(B_{GS}) = \rho^2(B_J)$ . **SOR Jacobi convergente  $\Rightarrow$  SOR Gauss-Seidel convergente y con factor de convergencia al cuadrado.**



# Métodos de sobrerrelajación: convergencia

**Teorema:** Sea  $A$  **simétrica definida positiva y tridiagonal en bloques:**

$$A = \begin{pmatrix} D_1 & U_1 & 0 & \dots & 0 \\ L_2 & D_2 & U_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & L_{n-1} & D_{n-1} & U_{n-1} \\ 0 & \dots & 0 & L_n & D_n \end{pmatrix}$$

donde  $D_i$ ,  $i = 1, \dots, n$  son submatrices diagonales,  $U_i$ ,  $L_i$ , submatrices cualesquiera que satisfacen  $L_{i+1} = U_i^T$ ,  $i = 1, \dots, n-1$ .

Entonces:

- Los radios espectrales de las matrices de iteración cumplen  $\rho(B_{GS}) = \rho^2(B_J)$ . **SOR Jacobi convergente  $\Rightarrow$  SOR Gauss-Seidel convergente y con factor de convergencia al cuadrado.**
- El **parámetro de relajación óptimo  $\bar{w}$** , que minimiza el radio espectral para  $0 < \bar{w} < 2$  es:  $\bar{w} = \frac{2}{1 + \sqrt{1 - \rho(B_{GS})}}$ ,  $\rho(B_{GS}) < 1$ .  
El valor óptimo de  $\rho(B_w)$  (para SOR Gauss-Seidel) es  $\rho(B_{\bar{w}}) = \bar{w} - 1$ .

# **Sistemas de ecuaciones lineales**

## Precondicionadores

# Precondicionamiento

- ▶ Transformar el sistema para **mejorar la convergencia y la estabilidad** de los métodos iterativos para matrices grandes y dispersas.

# Precondicionamiento

- ▶ Transformar el sistema para **mejorar la convergencia y la estabilidad** de los métodos iterativos para matrices grandes y dispersas.
- ▶ Un **precondicionador**  $P$  de una matriz  $A$  es una matriz no singular tal que  $P^{-1}A$  tiene un número de condición bajo.

# Precondicionamiento

- ▶ Transformar el sistema para **mejorar la convergencia y la estabilidad** de los métodos iterativos para matrices grandes y dispersas.
- ▶ Un **precondicionador**  $P$  de una matriz  $A$  es una matriz no singular tal que  $P^{-1}A$  tiene un número de condición bajo.
- ▶ Precondicionador ideal:  $A^{-1}$ .

# Precondicionamiento

- ▶ Transformar el sistema para **mejorar la convergencia y la estabilidad** de los métodos iterativos para matrices grandes y dispersas.
- ▶ Un **precondicionador**  $P$  de una matriz  $A$  es una matriz no singular tal que  $P^{-1}A$  tiene un número de condición bajo.
- ▶ Precondicionador ideal:  $A^{-1}$ .
- ▶ Precondicionador  $P$ :
  - ▷ **Por la izquierda** (más comunes):  $P^{-1}A\mathbf{x} = P^{-1}\mathbf{b}$ .
  - ▷ **Por la derecha**:  $AP^{-1}(P\mathbf{x}) = \mathbf{b}$ .

# Precondicionamiento

- ▶ Transformar el sistema para **mejorar la convergencia y la estabilidad** de los métodos iterativos para matrices grandes y dispersas.
- ▶ Un **precondicionador**  $P$  de una matriz  $A$  es una matriz no singular tal que  $P^{-1}A$  tiene un número de condición bajo.
- ▶ Precondicionador ideal:  $A^{-1}$ .
- ▶ Precondicionador  $P$ :
  - ▷ **Por la izquierda** (más comunes):  $P^{-1}A\mathbf{x} = P^{-1}\mathbf{b}$ .
  - ▷ **Por la derecha**:  $AP^{-1}(P\mathbf{x}) = \mathbf{b}$ .
- ▶ Ejemplo: **precondicionador de Jacobi**  $P = \text{diag}(A)$ .

# Precondicionamiento en perspectiva

## Métodos directos:

Gauss, LU, Cholesky

+ robustos

- secuenciales

- pierden dispersión (sparsity)



# Precondicionamiento en perspectiva

## Métodos directos:

Gauss, LU, Cholesky

+ robustos

- secuenciales

- pierden dispersión (sparsity)

## Métodos iterativos:

Jacobi, Gauss-Seidel, SOR

+ paralelizables

+ mantienen dispersión (sparsity)

- convergencia a veces lenta

# Precondicionamiento en perspectiva

## Métodos directos:

Gauss, LU, Cholesky

+ robustos

- secuenciales

- pierden dispersión (sparsity)

## Métodos iterativos:

Jacobi, Gauss-Seidel, SOR

+ paralelizables

+ mantienen dispersión (sparsity)

- convergencia a veces lenta

## Precondicionamiento:

Precondicionador invertible  $P$  tal que al usar un método iterativo:

+ El sistema esté mejor condicionado.

+ La matriz del sistema resultante sea fácil de manejar en paralelo.

+ El espectro de la matriz del sistema resultante esté mejor agrupado.

# **Sistemas de ecuaciones lineales**

## Métodos iterativos no estacionarios

# Métodos iterativos no estacionarios

- ▶ Los métodos **no estacionarios** difieren de los métodos estacionarios en que los cálculos involucran información que cambia en cada iteración.

# Métodos iterativos no estacionarios

- ▶ Los métodos **no estacionarios** difieren de los métodos estacionarios en que los cálculos involucran información que cambia en cada iteración.
- ▶ Normalmente, las constantes se calculan tomando productos internos de residuos u otros vectores derivados del método iterativo.

# Métodos iterativos no estacionarios

- ▶ Los métodos **no estacionarios** difieren de los métodos estacionarios en que los cálculos involucran información que cambia en cada iteración.
- ▶ Normalmente, las constantes se calculan tomando productos internos de residuos u otros vectores derivados del método iterativo.
- ▶ A veces hace falta **precondicionar** para asegurar la convergencia.

# Métodos iterativos no estacionarios

- ▶ Los métodos **no estacionarios** difieren de los métodos estacionarios en que los cálculos involucran información que cambia en cada iteración.
- ▶ Normalmente, las constantes se calculan tomando productos internos de residuos u otros vectores derivados del método iterativo.
- ▶ A veces hace falta **precondicionar** para asegurar la convergencia.
- ▶ Algunos de estos métodos son: Método del gradiente conjugado (CG) y variantes: MINRES, SYMMLQ, CGNE, GMRES, BiCG, QMR, Bi-CGSTAB.

# Métodos iterativos no estacionarios

- ▶ Los métodos **no estacionarios** difieren de los métodos estacionarios en que los cálculos involucran información que cambia en cada iteración.
- ▶ Normalmente, las constantes se calculan tomando productos internos de residuos u otros vectores derivados del método iterativo.
- ▶ A veces hace falta **precondicionar** para asegurar la convergencia.
- ▶ Algunos de estos métodos son: Método del gradiente conjugado (CG) y variantes: MINRES, SYMMLQ, CGNE, GMRES, BiCG, QMR, Bi-CGSTAB.

**Gradiente conjugado (CG):** para  $A\mathbf{x} = \mathbf{b}$  con  $A$  simétrica definida positiva.

- ▷ Equivalentemente, para **optimizar un función cuadrática convexa**.
- ▷ Para sistemas dispersos demasiado grandes para los métodos directos.
- ▷ Surgen a menudo al resolver ecuaciones en derivadas parciales (EDPs).



# Métodos iterativos no estacionarios

- ▶ Los métodos **no estacionarios** difieren de los métodos estacionarios en que los cálculos involucran información que cambia en cada iteración.
- ▶ Normalmente, las constantes se calculan tomando productos internos de residuos u otros vectores derivados del método iterativo.
- ▶ A veces hace falta **precondicionar** para asegurar la convergencia.
- ▶ Algunos de estos métodos son: Método del gradiente conjugado (CG) y variantes: MINRES, SYMMLQ, CGNE, GMRES, BiCG, QMR, Bi-CGSTAB.


**Gradiente conjugado (CG):** para  $Ax = b$  con  $A$  simétrica definida positiva.

- ▷ Equivalentemente, para **optimizar un función cuadrática convexa.** **siguiente** →
- ▷ Para sistemas dispersos demasiado grandes para los métodos directos.
- ▷ Surgen a menudo al resolver ecuaciones en derivadas parciales (EDPs).

# Métodos iterativos no estacionarios

- ▶ Los métodos **no estacionarios** difieren de los métodos estacionarios en que los cálculos involucran información que cambia en cada iteración.
- ▶ Normalmente, las constantes se calculan tomando productos internos de residuos u otros vectores derivados del método iterativo.
- ▶ A veces hace falta **precondicionar** para asegurar la convergencia.
- ▶ Algunos de estos métodos son: Método del gradiente conjugado (CG) y variantes: MINRES, SYMMLQ, CGNE, GMRES, BiCG, QMR, Bi-CGSTAB.

**Gradiente conjugado (CG):** para  $Ax = b$  con  $A$  simétrica definida positiva.

- ▷ Equivalentemente, para **optimizar un función cuadrática convexa.** siguiente 
- ▷ Para sistemas dispersos demasiado grandes para los métodos directos.
- ▷ Surgen a menudo al resolver ecuaciones en derivadas parciales (EDPs).

*An introduction to the conjugate gradient method without the agonizing pain.*

# Vector residuo y vector gradiente

Sistema  $A\mathbf{x} = \mathbf{b}$ , con  $A$  simétrica definida positiva.

# Vector residuo y vector gradiente

Sistema  $A\mathbf{x} = \mathbf{b}$ , con  $A$  simétrica definida positiva.

Resolver el sistema lineal  $A\mathbf{x} = \mathbf{b}$  es equivalente al problema de minimizar la función definida por:

# Vector residuo y vector gradiente

Sistema  $A\mathbf{x} = \mathbf{b}$ , con  $A$  simétrica definida positiva.

Resolver el sistema lineal  $A\mathbf{x} = \mathbf{b}$  es equivalente al problema de minimizar la función definida por:

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^t A\mathbf{x} - \mathbf{x}^t \mathbf{b}$$

# Vector residuo y vector gradiente

Sistema  $A\mathbf{x} = \mathbf{b}$ , con  $A$  simétrica definida positiva.

Resolver el sistema lineal  $A\mathbf{x} = \mathbf{b}$  es equivalente al problema de minimizar la función definida por:

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^t A\mathbf{x} - \mathbf{x}^t \mathbf{b}$$

► El gradiente de esta función es  $\nabla\phi(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ .

# Vector residuo y vector gradiente

Sistema  $A\mathbf{x} = \mathbf{b}$ , con  $A$  simétrica definida positiva.

Resolver el sistema lineal  $A\mathbf{x} = \mathbf{b}$  es equivalente al problema de minimizar la función definida por:

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^t A\mathbf{x} - \mathbf{x}^t \mathbf{b}$$

- ▶ El gradiente de esta función es  $\nabla\phi(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ .
- ▶ La función  $\phi(x)$  es estrictamente **convexa** si y solo si  $A$  es definida positiva.

# Vector residuo y vector gradiente

Sistema  $A\mathbf{x} = \mathbf{b}$ , con  $A$  simétrica definida positiva.

Resolver el sistema lineal  $A\mathbf{x} = \mathbf{b}$  es equivalente al problema de minimizar la función definida por:

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^t A\mathbf{x} - \mathbf{x}^t \mathbf{b}$$

- ▶ El gradiente de esta función es  $\nabla\phi(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ .
- ▶ La función  $\phi(x)$  es estrictamente **convexa** si y solo si  $A$  es definida positiva.
- ▶ El mínimo es la solución  $\mathbf{x}$  de  $A\mathbf{x} - \mathbf{b}$ .



# Guia de estudio

Libro *Càlcul numèric: teoria i pràctica* de M. Grau Sánchez y M. Noguera Batlle.

- ▶ Conceptos y ejercicios resueltos: Capítulo 4, páginas restantes.
- ▶ Problemas propuestos: restantes.
- ▶ Prácticas propuestas: Secciones 4.6.1 y 4.6.2.

Libro *Cálculo numérico* de M. Grau Sánchez y M. Noguera Batlle.

- ▶ Conceptos y ejercicios resueltos: Capítulo 4, páginas restantes.
- ▶ Problemas propuestos: restantes.
- ▶ Prácticas propuestas: Secciones 4.6.1 y 4.6.2.