



Métodos Numéricos | Laboratorio 5

Ricardo Largaespada

24 de abril de 2025

1. Métodos Iterativos para Resolver Sistemas Lineales

Los métodos directos (como eliminación gaussiana) para resolver sistemas de ecuaciones lineales requieren memoria proporcional al cuadrado del orden de la matriz A y realizan un trabajo computacional proporcional a su cubo. Para sistemas grandes ($n \geq 50$) con matrices densas¹, estos métodos presentan limitaciones críticas:

- **Almacenamiento:** Demandan memoria excesiva.
- **Error numérico:** El alto número de operaciones amplifica los errores de redondeo.

Sin embargo, en problemas típicos de la **Ingeniería de Sistemas** —como el análisis de redes complejas, optimización de flujos de datos o procesamiento de grandes volúmenes de información— surgen sistemas de orden 1000 o superior. Estos sistemas suelen tener matrices **dispersas** (con mayoría de ceros) y propiedades que facilitan su resolución iterativa:

- **Estructura banda o diagonal dominante.**
- **Simetría** (common en modelos de grafos y redes).
- **Esparcidad** (típica en sistemas de comunicaciones y machine learning).

Los **métodos iterativos** (Jacobi, Gauss-Seidel) aprovechan estas propiedades para:

- Reducir el uso de memoria (almacenando solo elementos no nulos).
- Minimizar operaciones mediante aproximaciones sucesivas.
- Escalar eficientemente en sistemas de gran dimensión.

En esta sección analizaremos su implementación, convergencia y aplicaciones en problemas de sistemas computacionales.

¹Una matriz densa tiene pocos elementos nulos.



1.1. Métodos de Jacobi y Gauss–Seidel

Los métodos iterativos más sencillos y conocidos son una generalización del método de punto fijo, estudiado en la unidad 2. Se puede aplicar la misma técnica a fin de elaborar métodos para la solución de $A\mathbf{x} = \mathbf{b}$, de la siguiente manera:

Se parte de $A\mathbf{x} = \mathbf{b}$ para obtener la ecuación

$$A\mathbf{x} - \mathbf{b} = \mathbf{0} \quad (1)$$

ecuación vectorial correspondiente a $f(\mathbf{x}) = \mathbf{0}$. Se busca ahora una matriz B y un vector \mathbf{c} , de modo que la ecuación vectorial

$$\mathbf{x} = B\mathbf{x} + \mathbf{c} \quad (2)$$

sea sólo un arreglo de la ecuación (1); es decir, que la solución de una sea también la solución de la otra. La ecuación (2) correspondería a $\mathbf{x} = g(\mathbf{x})$. En seguida se propone un vector inicial $\mathbf{x}^{(0)}$, como primera aproximación al vector solución \mathbf{x} . Luego, se calcula con la ecuación (2) la sucesión vectorial $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots$ de la siguiente manera:

$$\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{c}, \quad k = 0, 1, 2, \dots \quad (3)$$

donde

$$\mathbf{x}^{(k)} = [x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}]^T \quad (4)$$

Para que la sucesión $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}, \dots$ converja al vector solución \mathbf{x} es necesario que eventualmente $x_j^{(m)}$, $1 \leq j \leq n$ (los componentes del vector $\mathbf{x}^{(m)}$) se aproximen tanto a x_j , $1 \leq j \leq n$ (los componentes correspondientes a \mathbf{x}), que todas las diferencias $|x_j^{(m)} - x_j|$, $1 \leq j \leq n$ sean menores que un valor pequeño previamente fijado, y que se conserven menores para todos los vectores siguientes de la iteración; es decir:

$$\lim_{m \rightarrow \infty} x_j^{(m)} = x_j, \quad 1 \leq j \leq n \quad (5)$$

La forma como se llega a la ecuación (2) define el algoritmo y su convergencia. Dado el sistema $A\mathbf{x} = \mathbf{b}$, la manera más sencilla es despejar x_i de la primera ecuación, x_2 de la segunda, y así sucesivamente. Para ello, es necesario que todos los elementos de la diagonal principal de A , por razones obvias, sean distintos de cero.

Para ver esto en detalle considérense el sistema general de tres ecuaciones (naturalmente puede extenderse a cualquier número de ecuaciones). Sea entonces

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

con a_{ii} distintos de cero. Se despeja x_1 de la primera ecuación, x_2 de la segunda, y x_3 de la tercera, con lo que se obtiene:



$$\begin{aligned}x_1 &= -\frac{a_{12}}{a_{11}}x_2 - \frac{a_{13}}{a_{11}}x_3 + \frac{b_1}{a_{11}} \\x_2 &= -\frac{a_{21}}{a_{22}}x_1 - \frac{a_{23}}{a_{22}}x_3 + \frac{b_2}{a_{22}} \\x_3 &= -\frac{a_{31}}{a_{33}}x_1 - \frac{a_{32}}{a_{33}}x_2 + \frac{b_3}{a_{33}}\end{aligned}\tag{6}$$

que en notación matricial queda:

$$\begin{bmatrix}x_1 \\ x_2 \\ x_3\end{bmatrix} = \begin{bmatrix}0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} \\ -\frac{a_{31}}{a_{33}} & -\frac{a_{32}}{a_{33}} & 0\end{bmatrix} \begin{bmatrix}x_1 \\ x_2 \\ x_3\end{bmatrix} + \begin{bmatrix}\frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \frac{b_3}{a_{33}}\end{bmatrix}\tag{7}$$

y ésta es la ecuación (4) desarrollada, con

$$B = \begin{bmatrix}0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} \\ -\frac{a_{31}}{a_{33}} & -\frac{a_{32}}{a_{33}} & 0\end{bmatrix} \quad \text{y} \quad \mathbf{c} = \begin{bmatrix}\frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \frac{b_3}{a_{33}}\end{bmatrix}$$

Una vez que se tiene la forma (7), se propone un vector inicial $\mathbf{x}^{(0)}$ que puede ser $\mathbf{x}^{(0)} = \mathbf{0}$, o algún otro que sea aproximado al vector solución \mathbf{x} . Para iterar existen dos variantes:

1.1.1. Iteración de Jacobi (método de desplazamientos simultáneos)

Si

$$\mathbf{x}^{(k)} = \begin{bmatrix}x_1^k \\ x_2^k \\ x_3^k\end{bmatrix}\tag{8}$$

es el vector aproximación a la solución \mathbf{x} después de k iteraciones, entonces se tiene para la siguiente aproximación

$$\mathbf{x}^{(k+1)} = \begin{bmatrix}x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)}\end{bmatrix} = \begin{bmatrix}\frac{1}{a_{11}}(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}) \\ \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)}) \\ \frac{1}{a_{33}}(b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)})\end{bmatrix}\tag{9}$$

O bien, para un sistema de n ecuaciones con n incógnitas y usando notación más compacta y de mayor utilidad en programación, se tiene:

$$x_i^{(k+1)} = -\frac{1}{a_{ii}} \left[-b_i + \sum_{j=1}^n a_{ij}x_j^k \right], \quad \text{para } 1 \leq i \leq n\tag{10}$$

1.1.2. Iteración de Gauss–Seidel (método de desplazamientos sucesivos)

En este método, los valores que se van calculando en la $(k + 1)$ -ésima iteración se emplean para estimar los valores faltantes de esa misma iteración; es decir, con $\mathbf{x}^{(k)}$ se calcula $\mathbf{x}^{(k+1)}$ de acuerdo con



$$\mathbf{x}^{(k+1)} = \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \end{bmatrix} = \begin{bmatrix} \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}) \\ \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)}) \\ \frac{1}{a_{33}}(b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)}) \end{bmatrix} \quad (11)$$

O bien, para un sistema de n ecuaciones:

$$x_i^{(k+1)} = -\frac{1}{a_{ii}} \left[-b_i + \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} + \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right], \quad \text{para } 1 \leq i \leq n \quad (12)$$

Ejemplo 1.1. Resuelva el siguiente sistema por los métodos de Jacobi y Gauss-Seidel.

$$\begin{aligned} 4x_1 - x_2 &= 1 \\ -x_1 + 4x_2 - x_3 &= 1 \\ -x_2 + 4x_3 - x_4 &= 1 \\ -x_3 + 4x_4 &= 1 \end{aligned} \quad (13)$$

Solución. Despejando x_1 de la primera ecuación, x_2 de la segunda, etcétera, se obtiene:

$$\begin{aligned} x_1 &= \frac{x_2}{4} + \frac{1}{4} \\ x_2 &= \frac{x_1}{4} + \frac{x_3}{4} + \frac{1}{4} \\ x_3 &= \frac{x_2}{4} + \frac{x_4}{4} + \frac{1}{4} \\ x_4 &= \frac{x_3}{4} + \frac{1}{4} \end{aligned} \quad (14)$$

Vector inicial: Cuando no se tiene una aproximación al vector solución, generalmente se emplea como vector inicial el vector cero, esto es: $\mathbf{x}^{(0)} = [0 \ 0 \ 0 \ 0]^T$.

a) Método de Jacobi El cálculo de $\mathbf{x}^{(1)}$ en el método de Jacobi se obtiene reemplazando $\mathbf{x}^{(0)}$ en cada una de las ecuaciones de la ecuación (14) y entonces $\mathbf{x}^{(1)} = [1/4 \ 1/4 \ 1/4 \ 1/4]^T$. Para calcular $\mathbf{x}^{(2)}$ se sustituye $\mathbf{x}^{(1)}$ en cada una de las ecuaciones de la ecuación (14). Para simplificar la notación, se han omitido los superíndices. Así:

$$\begin{aligned} x_1 &= \frac{1}{16} + \frac{1}{4} = 0.3125 \\ x_2 &= \frac{1}{16} + \frac{1}{16} + \frac{1}{4} = 0.3750 \\ x_3 &= \frac{1}{16} + \frac{1}{16} + \frac{1}{4} = 0.3750 \\ x_4 &= \frac{1}{16} + \frac{1}{4} = 0.3125 \end{aligned}$$

A continuación se presentan los resultados de subsecuentes iteraciones, en forma tabular: Para realizar los cálculos puede usarse **Matlab**.

Cuadro 1: Solución del sistema (13) por el método de Jacobi.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$
0	0.0000	0.0000	0.0000	0.0000
1	0.2500	0.2500	0.2500	0.2500
2	0.3125	0.3750	0.3750	0.3125
3	0.3438	0.4219	0.4219	0.3438
4	0.3555	0.4414	0.4414	0.3555
5	0.3604	0.4492	0.4492	0.3604
6	0.3623	0.4524	0.4524	0.3623
7	0.3631	0.4537	0.4537	0.3631
8	0.3634	0.4542	0.4542	0.3634
9	0.3635	0.4544	0.4544	0.3635
10	0.3636	0.4545	0.4545	0.3636

```

1  % Método de Jacobi
2  clear
3  A = [4 -1 0 0; -1 4 -1 0; 0 -1 4 -1; 0 0 -1 4];
4  b = [1 1 1 1];
5  X0 = zeros(1,4);
6  K = 0;
7  Norma = 1;
8  fprintf(' K      X(1)      X(2)      X(3)      X(4)      Norma\n');
9  while Norma > 0.0001
10     K = K + 1;
11     fprintf('%2d ', K)
12     for i = 1:4
13         suma = 0;
14         for j = 1:4
15             if i ~= j
16                 suma = suma + A(i,j) * X0(j);
17             end
18         end
19         X(i) = (b(i) - suma) / A(i,i);
20         fprintf('%10.4f ', X(i))
21     end
22     Norma = norm(X0 - X);
23     fprintf('%10.4f\n', Norma)
24     X0 = X;
25     if K > 25
26         disp('No se alcanzó la convergencia')
27         break
28     end
29 end

```

b) Método de Gauss–Seidel

Para el cálculo del primer elemento del vector $\mathbf{x}^{(1)}$, se sustituye $\mathbf{x}^{(0)}$ en la primera ecuación de la ecuación (14); con el fin de simplificar la notación se han omitido los superíndices:

$$x_1 = \frac{0}{4} + \frac{1}{4} = \frac{1}{4}$$

Para el cálculo de x_2 de $\mathbf{x}^{(1)}$ se emplea el valor de x_1 ya obtenido (1/4) y los valores x_2 , x_3 , y x_4 de $\mathbf{x}^{(0)}$. Así:

$$x_2 = \frac{1}{4} \left(\frac{1}{4} + 0 + \frac{1}{4} \right) = 0.3125$$

Con los valores de x_1 , x_2 ya obtenidos, y con x_3 , x_4 de $\mathbf{x}^{(0)}$, se evalúa x_3 de $\mathbf{x}^{(1)}$:

$$x_3 = \frac{1}{4} \left(0.3125 + 0 + \frac{1}{4} \right) = 0.3281$$

Finalmente, con los valores de x_1 , x_2 , y x_3 calculados previamente, y con x_4 de $\mathbf{x}^{(0)}$, se obtiene la última componente de $\mathbf{x}^{(1)}$:

$$x_4 = \frac{1}{4} \left(0.3281 + \frac{1}{4} \right) = 0.3320$$

Entonces $\mathbf{x}^{(1)} = [0.25 \quad 0.3125 \quad 0.3281 \quad 0.3320]^T$.

Para la segunda iteración (cálculo de $\mathbf{x}^{(2)}$) se procede de igual manera:

$$x_1 = \frac{1}{4} \left(0.3125 + \frac{1}{4} \right) = 0.3281$$

$$x_2 = \frac{1}{4} \left(0.3281 + 0.3281 + \frac{1}{4} \right) = 0.4141$$

$$x_3 = \frac{1}{4} \left(0.4141 + 0.3320 + \frac{1}{4} \right) = 0.4365$$

$$x_4 = \frac{1}{4} \left(0.4365 + \frac{1}{4} \right) = 0.3591$$

Con lo que $\mathbf{x}^{(2)} = [0.3281 \quad 0.4141 \quad 0.4365 \quad 0.3591]^T$.

En la tabla siguiente se presentan los resultados de las iteraciones subsecuentes.

Cuadro 2: Solución del sistema (13) por el método de Gauss–Seidel.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$
0	0.0000	0.0000	0.0000	0.0000
1	0.2500	0.3125	0.3281	0.3320
2	0.3281	0.4141	0.4365	0.3591
3	0.3535	0.4475	0.4517	0.3629
4	0.3619	0.4534	0.4541	0.3635
5	0.3633	0.4544	0.4545	0.3636
6	0.3636	0.4545	0.4545	0.3636

Para realizar los cálculos puede usarse **Matlab**.



```
1 %Método de Gauss–Seidel
2 clear
3 A = [4 -1 0 0;-1 4 -1 0; 0 -1 4 -1;0 0 -1 4];
4 b = [1 1 1 1]';
5 X0 = zeros(1, 4); % Vector inicial
6 X = X0;
7 K = 0;
8 Norma = 1;
9 fprintf(' K      X(1)      X(2)      X(3)      X(4)      Norma\n');
10 while Norma > 0.0001
11     K = K + 1;
12     fprintf('%2d ', K)
13     for i = 1:4
14         suma = 0;
15         for j = 1:4
16             if i ~= j
17                 suma = suma + A(i, j) * X(j); % Usa valores nue. o
18                                     act.
19             end
20         end
21         X(i) = (b(i) - suma) / A(i, i);
22         fprintf('%10.4f ', X(i))
23     end
24     Norma = norm(X0 - X);
25     fprintf('%10.4f\n', Norma)
26     X0 = X;
27     if K > 17
28         disp('No se alcanzó la convergencia')
29         break
30     end
31 end
```

En la aplicación de estas dos variantes son válidas las preguntas siguientes:

1. ¿La sucesión de vectores $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \dots$ converge o se aleja del vector solución $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$?
2. ¿Cuándo detener el proceso iterativo?

Las respuestas correspondientes, conocidas como criterio de convergencia, se dan a continuación:

1. Si la sucesión converge a \mathbf{x} , cabe esperar que los elementos de $\mathbf{x}^{(k)}$ se vayan acercando a los elementos correspondientes de \mathbf{x} ; es decir, $x_1^k \rightarrow x_1$, $x_2^k \rightarrow x_2$, etcétera, o que se alejen en caso contrario.
2. Cuando:



- a) Los valores absolutos $|x_1^{(k+1)} - x_1^{(k)}|$, $|x_2^{(k+1)} - x_2^{(k)}|$, etc., sean todos menores de un número pequeño ε , cuyo valor será dado por el programador.
- b) Si el número de iteraciones ha excedido un máximo predeterminado MAXIT.

Por otro lado, es natural pensar que si la sucesión $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots$ converge a \mathbf{x} , la distancia (véase sección 3.2) de $\mathbf{x}^{(0)}$ a \mathbf{x} , de $\mathbf{x}^{(1)}$ a \mathbf{x} , etc., se va reduciendo; también es cierto que la distancia entre dos vectores consecutivos $\mathbf{x}^{(0)}$ y $\mathbf{x}^{(1)}$, $\mathbf{x}^{(1)}$ y $\mathbf{x}^{(2)}$, etc., se decrementa conforme el proceso iterativo avanza; esto es, la sucesión de números reales

$$\begin{aligned} &|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}| \\ &|\mathbf{x}^{(2)} - \mathbf{x}^{(1)}| \\ &\vdots \\ &|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}| \end{aligned} \tag{15}$$

convergirán a cero.

Si, por el contrario, esta sucesión de números diverge, entonces puede pensarse que el proceso diverge. Con esto, un criterio más es:

- c) Detener el proceso una vez que $|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}| < \varepsilon$.

Al elaborar un programa de cómputo para resolver sistemas de ecuaciones lineales, generalmente se utilizan los criterios a), b) y c), o la combinación de a) y b), o la de b) y c).

1.2. Rearreglo de ecuaciones

Para motivar el rearreglo de ecuaciones, se propone resolver el siguiente sistema con el método de Gauss-Seidel y con $\varepsilon = 10^{-2}$ aplicado a $|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}|$:

$$\begin{aligned} -x_1 + 3x_2 + 5x_3 + 2x_4 &= 10 \\ x_1 + 9x_2 + 8x_3 + 4x_4 &= 15 \\ x_2 + x_4 &= 2 \\ 2x_1 + x_2 + x_3 - x_4 &= -3 \end{aligned} \tag{16}$$

Al resolver para x_1 de la primera ecuación, para x_2 de la segunda, para x_3 de la cuarta, y para x_4 de la tercera, se obtiene:

$$\begin{aligned} x_1 &= 3x_2 + 5x_3 + 2x_4 - 10 \\ x_2 &= -x_1/9 - \frac{8}{9}x_3 - \frac{4}{9}x_4 + \frac{15}{9} \\ x_3 &= -2x_1 - x_2 + x_4 - 3 \\ x_4 &= -x_2 + 2 \end{aligned}$$

Con el vector cero como vector inicial, se tiene la siguiente sucesión de vectores. Nótese que el proceso diverge.

Cuadro 3: Aplicación del método de Gauss-Seidel al sistema (16).

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$	$ \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} $
0	0.0000	0.0000	0.0000	0.0000	
1	-10.0000	2.7778	14.2220	-0.7778	17.62
2	67.8889	-18.1720	-121.2	20.17	159.0
3	-631.1	170.7	1108.0	-168.71	1439.05

Si el proceso iterativo diverge, como es el caso, un rearrreglo de las ecuaciones puede originar convergencia; por ejemplo, en lugar de despejar x_1 de la primera ecuación, x_2 de la segunda, etc., cabe despejar las diferentes x_i de diferentes ecuaciones, teniendo cuidado de que los coeficientes de las x_i despejadas sean distintos de cero.

Esta sugerencia presenta, para un sistema de n ecuaciones, $n!$ distintas formas de rearrreglar dicho sistema. A fin de simplificar este procedimiento, se utilizará el siguiente teorema.

Teorema 1. Convergencia de Método Iterativo

Los procesos de Jacobi y Gauss-Seidel convergirán si, en la matriz coeficiente, cada elemento de la diagonal principal es mayor (en valor absoluto) que la suma de los valores absolutos de todos los demás elementos de la misma fila o columna (matriz diagonal dominante). Es decir, se asegura la convergencia si:

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad \text{para } 1 \leq i \leq n \quad \text{y} \quad |a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ji}| \quad \text{para } 1 \leq i \leq n \quad (17)$$

Debemos advertir que este teorema no será de mucha utilidad si se toma al pie de la letra, ya que contados sistemas de ecuaciones lineales poseen matrices coeficientes diagonalmente dominantes; sin embargo, si se arreglan las ecuaciones para tener el sistema lo más cercano posible a las condiciones del teorema, algún beneficio se puede obtener. Ésta es la pauta para reordenar las ecuaciones y obtener o mejorar la convergencia, en el mejor de los casos. A continuación se ilustra esto, reagrupando el sistema (16) y despejando x_1 de la ecuación 4, x_2 de la ecuación 2, x_3 de la ecuación 1 y x_4 de la ecuación 3, para llegar a:

$$\begin{aligned} x_1 &= -\frac{x_2}{2} - \frac{x_3}{2} + \frac{x_4}{2} - \frac{3}{2} \\ x_2 &= -\frac{x_1}{9} - \frac{8x_3}{9} - \frac{4x_4}{9} + \frac{15}{9} \\ x_3 &= \frac{x_1}{5} - \frac{3x_2}{5} - \frac{2x_4}{5} + \frac{10}{5} \\ x_4 &= -x_2 + 2 \end{aligned}$$

Los resultados para las primeras 18 iteraciones con el vector cero como vector inicial se muestran en la tabla 4.

Antes de continuar las iteraciones, puede observarse en la tabla 4 que los valores de $\mathbf{x}^{(18)}$ parecen converger al vector

$$\mathbf{x} = [-1 \quad 0 \quad 1 \quad 2]^T$$

Con la sustitución de estos valores en el sistema (16), se comprueba que $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, y $x_4 = 2$ es el vector solución, y por razones obvias se detiene el proceso.

Finalmente, las ecuaciones (17) son equivalentes (en sistemas de ecuaciones) a la expresión de convergencia del método de punto fijo de la unidad 2.

Cuadro 4: Aplicación del método de Gauss-Seidel al sistema (16), reorganizando las ecuaciones para obtener una aproximación a un sistema diagonal dominante.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$	$ \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} $
0	0.0000	0.0000	0.0000	0.0000	0.0000
1	-1.5000	1.8333	0.6000	0.1667	2.4400
2	-2.6333	1.3519	0.5956	0.6481	1.3200
3	-2.1496	1.0881	0.6580	0.9119	0.6140
4	-1.9171	0.8895	0.7181	1.1105	0.3695
5	-1.7486	0.7291	0.7686	1.2704	0.2867
6	-1.6134	0.5978	0.8102	1.4022	0.2337
7	-1.5030	0.4903	0.8444	1.5097	0.1907
8	-1.4125	0.4020	0.8724	1.5980	0.1567
9	-1.3382	0.3297	0.8953	1.6703	0.1285
10	-1.2774	0.2704	0.9142	1.7296	0.1052
11	-1.2275	0.2217	0.9296	1.7783	0.08643
12	-1.1865	0.1818	0.9423	1.8182	0.07089
13	-1.1530	0.1491	0.9527	1.8509	0.06162
14	-1.1254	0.1223	0.9612	1.8777	0.04764
15	-1.1029	0.1003	0.9682	1.8997	0.03903
16	-1.0844	0.0822	0.9739	1.9178	0.03209
17	-1.0692	0.0674	0.9786	1.9326	0.02629
18	-1.0567	0.0553	0.9824	1.9447	0.02152

1.3. Aceleración de convergencia

Si aún después de arreglado el sistema por resolver $A\mathbf{x} = \mathbf{b}$, conforme la pauta del Teorema 1, no se obtiene convergencia por los métodos de Jacobi y Gauss-Seidel, o ésta es muy lenta (como sucedió con el sistema (16) de la sección anterior), puede recurrirse a los métodos de **relajación** que, como se hará notar posteriormente, son los métodos de Jacobi y Gauss-Seidel afectados por un factor de peso ω que, elegido adecuadamente, puede producir convergencia o acelerarla, si ya existe. A continuación se describen estos métodos para un sistema de n ecuaciones en n incógnitas.



Llámesse N a la matriz coeficiente del sistema por resolver, una vez que haya sido llevada a la forma más cercana posible a diagonal dominante, y después de dividir la primera fila entre a_{11} , la segunda entre a_{22}, \dots , y la n -ésima entre a_{nn} . N es una matriz con unos en la diagonal principal. A continuación descompóngase N en la siguiente forma:

$$N = L + I + U \quad (1)$$

donde L es una matriz cuyos elementos por debajo de su diagonal principal son idénticos a los correspondientes de N y ceros en cualquier otro sitio, I es la matriz identidad, y U una matriz cuyos elementos arriba de la diagonal principal son idénticos a los correspondientes de N y cero en cualquier otro sitio. Sustituyendo esta descomposición de N , el sistema que se quiere resolver quedaría

$$(L + I + U)\mathbf{x} = \mathbf{b} \quad (18)$$

Si ahora se suma \mathbf{x} a cada miembro de la ecuación (18) se obtiene:

$$(L + I + U)\mathbf{x} + \mathbf{x} = \mathbf{b} + \mathbf{x}$$

Despejando \mathbf{x} del lado izquierdo, se llega al esquema siguiente:

$$\mathbf{x} = \mathbf{x} + [\mathbf{b} - L\mathbf{x} - \mathbf{x} - U\mathbf{x}] \quad (19)$$

que puede utilizarse para iterar a partir de un vector inicial $\mathbf{x}^{(0)}$. Nótese que la ecuación (19), puede al igual que la ecuación (7), ya que sólo es un rearrreglo de ésta.

Al aplicar la ecuación (19), pueden presentarse de nuevo las dos variantes que dieron lugar a los métodos de Jacobi y Gauss-Seidel, con lo que el esquema de desplazamientos simultáneos quedaría:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + [\mathbf{b} - L\mathbf{x}^{(k)} - \mathbf{x}^{(k)} - U\mathbf{x}^{(k)}] \quad (20)$$

y el de desplazamientos sucesivos así:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + [\mathbf{b} - L\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} - U\mathbf{x}^{(k)}] \quad (21)$$

Llegar a los esquemas (20) y (21) no se trata simplemente de tener una versión distinta de las ecuaciones (7), sino para someterlos a un análisis que permita proponer “nuevos métodos” o mejoras en los que ya se tienen. Por ejemplo, factorizando $\mathbf{x}^{(k)}$ dentro del paréntesis rectangular de la ecuación (20), se tiene:

$$\mathbf{b} - (L + I + U)\mathbf{x}^{(k)} = \mathbf{b} - N\mathbf{x}^{(k)} = \mathbf{r}^{(k)} \quad (22)$$

vector que se denota como $\mathbf{r}^{(k)}$ y se llama **vector residuo** de la k -ésima iteración y puede tomarse como una medida de la cercanía de $\mathbf{x}^{(k)}$ al vector solución \mathbf{x} ; si las componentes de $\mathbf{r}^{(k)}$ son pequeñas, $\mathbf{x}^{(k)}$ suele ser una buena aproximación a \mathbf{x} , pero si los elementos de $\mathbf{r}^{(k)}$ son grandes, puede pensarse que $\mathbf{x}^{(k)}$ no es muy cercana a \mathbf{x} . Aunque hay circunstancias donde esto no se cumple, por ejemplo, cuando el sistema por resolver está mal condicionado, es práctico tomar estos criterios como *válidos*.

Al sustituir la ecuación (22), en la (20) queda:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{r}^{(k)} \quad (23)$$

que puede verse como un esquema iterativo donde el vector de la $(k + 1)$ -ésima iteración se obtiene a partir del vector de la k -ésima iteración y el residuo correspondiente.

Si la aplicación de la ecuación (23) a un sistema particular da convergencia lenta, entonces $\mathbf{x}^{(k+1)}$ y $\mathbf{x}^{(k)}$ están muy cercanas entre sí, y para que la convergencia se acelere puede intentarse afectar $\mathbf{r}^{(k)}$ con un peso $w > 1$ (*sobrerrelajar* el proceso); si, en cambio, el proceso diverge y $\mathbf{r}^{(k)}$ es grande, convendría afectar $\mathbf{r}^{(k)}$ con un factor $w < 1$ (*subrelajar* el proceso), para provocar la convergencia. El esquema (23) quedaría en general así:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + w \mathbf{r}^{(k)} \quad (24)$$

o

$$x_i^{(k+1)} = x_i^{(k)} + w \left[b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(k)} \right], \quad 1 \leq i \leq n \quad (25)$$

para desplazamientos simultáneos.

Para desplazamientos sucesivos, en cambio, quedaría

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + w [\mathbf{b} - L\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} - U\mathbf{x}^{(k)}] \quad (26)$$

o

$$x_i^{(k+1)} = x_i^{(k)} + w \left[b_i - \sum_{j=1}^{i-1} l_{ij} x_j^{(k+1)} - x_i^{(k)} - \sum_{j=i+1}^n u_{ij} x_j^{(k)} \right], \quad 1 \leq i \leq n \quad (27)$$

Estos métodos se abrevian frecuentemente como **SOR** (del inglés, *Successive Over-Relaxation*). En general, el cálculo de w es complicado y sólo para sistemas especiales (matriz coeficiente positivamente definida y tridiagonal) se tiene una fórmula.²

Ejemplo 1.2. Resuelva el sistema (16):

$$-x_1 + 3x_2 + 5x_3 + 2x_4 = 10$$

$$x_1 + 9x_2 + 8x_3 + 4x_4 = 15$$

$$x_2 + x_4 = 2$$

$$2x_1 + x_2 + x_3 - x_4 = -3$$

con desplazamientos sucesivos, $w = 1.3$ y con $\varepsilon = 10^{-2}$ aplicado a $|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}|$.

Solución. La matriz N y el vector de términos independientes correspondiente son:

²R. L. Burden y J. D. Faires, *Análisis numérico*, Grupo Editorial Iberoamericana, 1985, p. 475.



$$N = \begin{bmatrix} 1 & 1/2 & 1/2 & -1/2 \\ 1/9 & 1 & 8/9 & 4/9 \\ -1/5 & 3/5 & 1 & 2/5 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -3/2 \\ 15/9 \\ 10/5 \\ 2 \end{bmatrix}$$

Descomposición de N

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1/9 & 0 & 0 & 0 \\ -1/5 & 3/5 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & 1/2 & 1/2 & -1/2 \\ 0 & 0 & 8/9 & 4/9 \\ 0 & 0 & 0 & 2/5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Primera iteración. Obtención de $\mathbf{x}^{(1)}$ a partir del vector inicial $\mathbf{x}^{(0)} = [0 \ 0 \ 0 \ 0]^T$ y empleando la ecuación (26).

Cálculo de $x_1^{(1)}$, esto es, $i = 1$ y $k + 1 = 1$:

$$x_1^{(1)} = x_1^{(0)} + 1.3 \left[b_1 - \sum_{j=1}^0 l_{1j} x_j^{(1)} - x_1^{(0)} - \sum_{j=2}^4 u_{1j} x_j^{(0)} \right]$$

Obsérvese que en la primera sumatoria el valor inicial ($j = 1$) es mayor que el valor final (0); la convención en estos casos es que la sumatoria no se realiza. Por lo tanto,

$$x_1^{(1)} = 0 + 1.3 \left[-\frac{3}{2} - 0 - \frac{1}{2}(0) - \frac{1}{2}(0) + \frac{1}{2}(0) \right] = -1.95$$

Cálculo de $x_2^{(1)}$, $i = 2$:

$$x_2^{(1)} = 0 + 1.3 \left[\frac{15}{9} - \frac{1}{9}(-1.95) - 0 - \frac{8}{9}(0) - \frac{4}{9}(0) \right] = 2.4483$$

Cálculo de $x_3^{(1)}$, $i = 3$:

$$x_3^{(1)} = 0 + 1.3 \left[\frac{10}{5} - (-\frac{1}{5})(-1.95) - \frac{3}{5}(2.4483) - 0 - \frac{2}{5}(0) \right] = 0.1833$$

Cálculo de $x_4^{(1)}$, $i = 4$:

$$x_4^{(1)} = 0 + 1.3 [2 - 0 - 1(2.4483) - 0(0.1833) - 0] = -0.5828$$

Cálculo de $|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}| = d_1$:

$$\begin{aligned} d_1 &= \sqrt{(x_1^{(1)} - x_1^{(0)})^2 + (x_2^{(1)} - x_2^{(0)})^2 + (x_3^{(1)} - x_3^{(0)})^2 + (x_4^{(1)} - x_4^{(0)})^2} \\ &= \sqrt{(-1.95)^2 + (2.4483)^2 + (0.1833)^2 + (-0.5828)^2} = 3.1891 \end{aligned}$$

Los valores mostrados en la tabla 5 se encuentran continuando las iteraciones.

Para realizar los cálculos puede usarse el siguiente guión de Matlab:

Cuadro 5: Resultados obtenidos con $w = 1.3$.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$	$ \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} $
0	0.0000	0.0000	0.0000	0.0000	—
1	-1.9500	2.4483	0.1833	-0.5828	3.1891
2	-3.4544	2.0561	0.3462	0.1020	1.7066
3	-2.4089	1.4388	0.6945	0.6989	1.3971
4	-2.1597	0.8406	0.8110	1.2976	0.8898
5	-1.5322	0.4489	0.9334	1.6271	0.8190
6	-1.3312	0.2055	0.9674	1.8447	0.3848
7	-1.1140	0.0822	0.9968	1.9397	0.2689
8	-1.0563	0.0220	1.0005	1.9895	0.0972
9	-1.0046	-0.0004	1.0045	2.0037	0.0583
10	-0.9988	-0.0074	1.0028	2.0084	0.0103
11	-0.9919	-0.0070	1.0024	2.0066	0.0072

```

1  % Método SOR
2  clear
3  A = [2 1 1 -1; 1 9 8 4; -1 3 5 2; 0 1 0 1];
4  b = [-3 15 10 2]';
5  for i = 1:4
6      N(i,:) = A(i,:) / A(i,i);
7      b(i) = b(i) / A(i,i);
8  end
9  U = triu(N,1);
10 L = tril(N,1);
11 X0 = zeros(1,4);
12 X = X0; w = 1.3;
13 K = 0; Norma = 1;
14 fprintf(' K X(1) X(2) X(3) X(4) Norma\n')
15 fprintf('%d %8.4f %8.4f %8.4f %8.4f\n',K,X(1),X(2),X(3),X(4))
16 while Norma > 0.01
17     K = K + 1;
18     fprintf('%d ', K)
19     for i = 1:4
20         sumaL = 0; sumaU = 0;
21         for j = 1:4
22             if j < i
23                 sumaL = sumaL + L(i,j) * X(j);
24             elseif j > i
25                 sumaU = sumaU + U(i,j) * X0(j);
26             end
27         end
28         X(i) = X0(i) + w * (b(i) - sumaL - X0(i) - sumaU);

```



```
29         fprintf( ' %8.4f ', X(i) )
30     end
31     Norma = norm(X - X0);
32     fprintf( ' %8.4f\n', Norma)
33     X0 = X;
34 end
```

Al comparar estos resultados con los obtenidos en la tabla 4 (método de Gauss-Seidel aplicado al sistema que aquí se resuelve), se observa que la convergencia es acelerada y los cálculos se reducen a la mitad.

1.4. Comparación de los métodos directos e iterativos

Una parte importante del análisis numérico consiste en conocer las características (ventajas y desventajas) de los métodos numéricos básicos que resuelven una familia de problemas (en este caso $A\mathbf{x} = \mathbf{b}$), para así elegir el algoritmo más adecuado para cada problema.

A continuación se presentan las circunstancias donde pudiera considerarse como ventajosa la elección de un método iterativo y también a qué se renuncia con esta decisión (desventajas).

Cuadro 6: Ventajas y desventajas comparativas entre métodos iterativos y directos

Ventajas de métodos iterativos	Desventajas de métodos iterativos
1. Eficientes para sistemas de orden muy alto ($n \geq 1000$)	1. Ineficiente para múltiples sistemas con misma matriz: requiere recalcular para cada vector independiente
2. Implementación computacional más simple	2. Convergencia potencialmente lenta (no siempre predecible)
3. Permite usar aproximaciones iniciales cercanas a la solución	3. Exactitud depende críticamente del criterio de convergencia elegido
4. Genera aproximaciones útiles en pocas iteraciones	4. Requiere verificación cuidadosa cuando la convergencia es lenta
5. Robustez ante errores de redondeo (ideal para matrices mal condicionadas)	5. No aprovecha simetrías matriciales para acelerar cómputo
6. Requiere memoria $\mathcal{O}(n)$ (vs. $\mathcal{O}(n^2)$ en métodos directos)	6. No calcula inversas ni determinantes