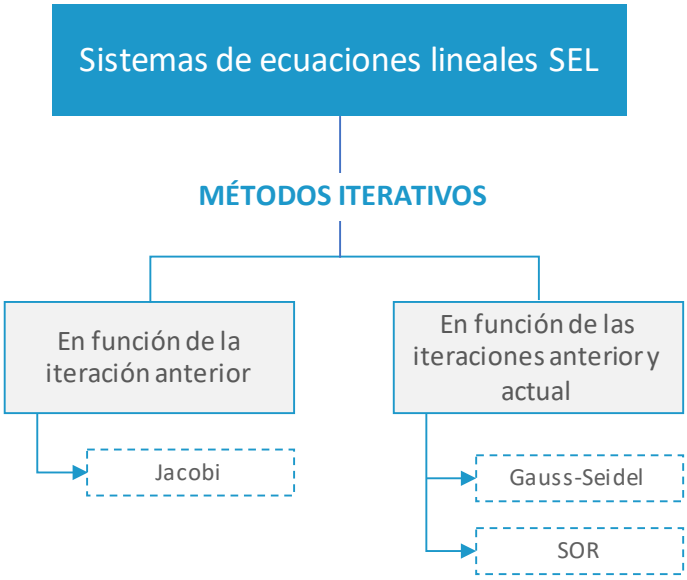
Métodos Numéricos Avanzados en Ingeniería

Sistemas de ecuaciones lineales

Índice

Esquema



Ideas clave

10.1. ¿Cómo estudiar este tema?

Para estudiar este tema lee las Ideas clave que encontrarás a continuación.

La resolución de sistemas de ecuaciones lineales es uno de los problemas principales de los que trata el álgebra lineal. En los primeros cursos de cualquier Grado se imparten los conceptos relacionados con esta área de las matemáticas, entre los que se encuentran diferentes algoritmos para resolver los sistemas, métodos para factorizar la matriz, etc.

El objetivo principal de la resolución de sistemas de ecuaciones lineales es, una vez obtenida su expresión matricial:

$$Ax = b \leftrightarrow \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \vdots & a_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

conseguir alcanzar su solución x, donde A es una matriz de tamaño $n \times m$, x es un vector de m componentes, y b es un vector de n componentes.

Este tipo de problemas los hemos visto, por ejemplo, cada vez que hemos tenido que resolver un sistema por el método de Crout, dentro de los problemas de contorno multidimensional.

Volvamos al sistema sobre el que vamos a trabajar. Debemos tener en cuenta los resultados del Teorema 1.

Teorema 1 (Rouché-Frobenius)

Sea $Ax = b$ un sistema lineal de n ecuaciones con m incógnitas, donde $A \in Mat_{n\times m}$ y $b \in \mathbb{R}^n$.

Si $rg(A) = rg(A|b) = m$ entonces el sistema es compatible determinado, es decir, tiene solución única.

Si $rg(A) = rg(A|b) < m$ entonces el sistema es compatible indeterminado, es decir, tiene infinitas soluciones.

Si $rg(A) < rg(A|b)$ entonces el sistema es incompatible, es decir, no tiene solución.

Existen dos tipos de métodos para obtener las soluciones del sistema lineal. Están los métodos directos y los métodos iterativos.

Los métodos directos consisten en transformar el sistema en otro equivalente cuya resolución sea prácticamente inmediata. Estas transformaciones se hacen mediante las llamadas operaciones elementales. A modo de recordatorio, vamos a indicar los tres métodos directos más conocidos.

Por un lado, el primer método que se aprende es el método de Cramer. En este método, la solución x_i se obtiene a partir de los determinantes de la matriz A y de la matriz modificada $\tilde{A}_i$, donde se sustituye la columna i de la matriz A por el vector b. De este modo, los diferentes valores de la solución serán:

$$x_i = \frac{1}{|A|}\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1,i-1} & b_1 & a_{1,i+1} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{n,i-1} & b_n & a_{n,i+1} & \cdots & a_{nn} \end{vmatrix}, i = 1, \dots, n$$

El mayor inconveniente de este método es la cantidad de operaciones que hay que realizar para alcanzar la solución.

Otro método de resolución de sistemas lineales es el método de Gauss-Jordan. En este método, el primer paso consiste en obtener una matriz ampliada que sea triangular superior, de modo que:

$$\left[\begin{array}{ccc|c} a_{11} & \cdots & a_{1m} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nm} & b_n \end{array}\right] \sim \left[\begin{array}{ccc|c} a'_{11} & \cdots & a'_{1m} & b'_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & a'_{nm} & b'_n \end{array}\right]$$

El siguiente paso del algoritmo de Gauss-Jordan consiste en resolver el sistema mediante sustitución regresiva. El problema que presenta este tipo de resolución es que los **errores de redondeo pueden dispararse** si no se elige el pivote adecuado.
La otra técnica de resolución de sistemas lineales son los **métodos iterativos**. A lo largo de este tema se van a desarrollar técnicas iterativas que permitan obtener la solución de un sistema de una forma diferente a como se realiza en los métodos directos.

Los apartados de los que consta este tema son:

- Conceptos básicos.
- Método de Jacobi.
- Método de Gauss-Seidel.
- Método de sobrerrelajación.
- Convergencia de los métodos iterativos.

10.2. Conceptos básicos

Un método iterativo va a obtener una solución aproximada del sistema $Ax = b$ construyendo una sucesión de vectores:

$$\{x^{(1)}, x^{(2)}, \ldots, x^{(k)}, \ldots\} \in \mathbb{R}^n$$

A partir de un vector arbitrario $x^{(0)}$ denominado aproximación inicial.

Convergencia de métodos iterativos

Un método iterativo es convergente si:

$$\lim_{k\to\infty} x^{(k)} = \bar{x}$$

El vector error, en cada iteración, se define como:

$$e^{(k)} = \bar{x} - x_k$$

El vector residuo, en cada iteración, se define como:

$$r^{(k)} = b - Ax^{(k)}$$

Estos tres conceptos están relacionados de la siguiente manera:

$$\lim_{k\to\infty} x^{(k)} = \bar{x} \Leftrightarrow \lim_{k\to\infty} \left\|e^{(k)}\right\| = 0 \Leftrightarrow \lim_{k\to\infty} \left\|r^{(k)}\right\| = 0$$

Criterio de parada

Como indicamos al principio de la asignatura, las soluciones que vamos a obtener serán aproximadas, de forma que los límites anteriores nunca alcanzarán el valor 0, por lo que hay que determinar un valor muy pequeño para el cual consideraremos que hemos alcanzado la solución. A ese valor lo denominaremos **«Tol»,** de modo que cuando el error absoluto $\left\|e^{(k)}\right\| < Tol$ o cuando el error relativo $\frac{\left\|e^{(k)}\right\|}{\bar{x}} < Tol$ habremos encontrado la solución. Pero debemos tener en cuenta que no disponemos de los valores de $\left\|e^{(k)}\right\|$ y de $\bar{x}$.

El criterio de parada lo tomaremos a partir del residuo, de modo que cuando el error absoluto $\left\|r^{(k)}\right\| = \left\|b - Ax^{(k)}\right\| < Tol$ o el error relativo $\frac{\left\|r^{(k)}\right\|}{\|b\|} = \frac{\left\|b-Ax^{(k)}\right\|}{\|b\|} < Tol$, habremos alcanzado la solución.

Analicemos si el criterio de parada basado en el residuo es conveniente o no:

$$r^{(k)} = b - Ax^{(k)} = A\bar{x} - Ax^{(k)} = A\left(\bar{x} - x^{(k)}\right) = Ae^{(k)}$$

Usando normas matriciales:

$$\left\|r^{(k)}\right\| \leq \|A\|\left\|e^{(k)}\right\| \leftrightarrow \left\|e^{(k)}\right\| \leq \|A^{-1}\|\left\|r^{(k)}\right\|$$

Además:

$$\|\bar{x}\| \leq \|A^{-1}\|\|b\| \leftrightarrow \|b\| \leq \|A\|\|\bar{x}\|$$

Combinando estas desigualdades:

$$\frac{1}{\|A\|\|A^{-1}\|}\frac{\left\|r^{(k)}\right\|}{\|b\|} \leq \frac{\left\|e^{(k)}\right\|}{\|\bar{x}\|} \leq \|A\|\|A^{-1}\|\frac{\left\|r^{(k)}\right\|}{\|b\|}$$

Teniendo en cuenta que el número de condición de la matriz $A, \mathcal{K}(A) = \|A\|\|A^{-1}\|$, podemos concluir que si el valor de $\mathcal{K}(A)$ no es muy grande, es decir, A es una matriz estable, entonces el criterio del residuo es fiable.

Ejemplo 1. Vamos a resolver dos sistemas de ecuaciones muy similares.

Sea el sistema de ecuaciones lineales:

$$\left.\begin{array}{rcrcr} 3x & + & y & = & 7 \\ 3x & + & 1.0001y & = & 7.0001 \end{array}\right\}$$

Obtén una solución del sistema y representa las rectas del sistema de ecuaciones.

b) Resuelve el mismo sistema cambiando el valor 7.0001 por 6.9999 y representa las nuevas rectas del sistema de ecuaciones.
c) ¿Cuál es el número de condición de la matriz?

Apartado a. Resolvamos este sistema por el método de Gauss-Jordan utilizando Matlab. Usaremos el comando `rref`.

```
>> A=[3 1;3 1.0001]; b=[7;7.0001];
>> xy1=rref([A b])
```

La solución es $(x, y) = (2,1)$. Representemos las ecuaciones lineales:

```
>> x=linspace(1,3); y1=7-3*x; y2=(7.0001-
3*x)/1.0001;
>> plot(x,[y1;y2])
```

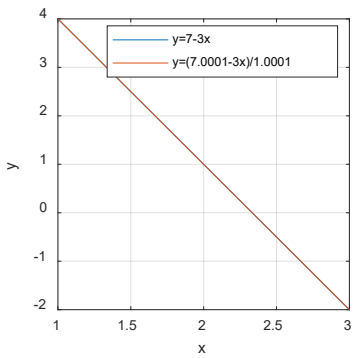


Figura 1. Representación de la solución del sistema $(x, y) = (2,1)$.

Apartado b. Al modificar el valor, modificamos el vector b. Calculemos la solución:

```
>> b2=[7;6.9999];
>> xy2=rref([A b2]);
```

La solución ahora es $(x, y) = \left(\frac{8}{3}, -1\right)$. Representemos las ecuaciones lineales:

```
>> y3=(6.9999-3*x)/1.0001
>> plot(x,[y1;y3])
```

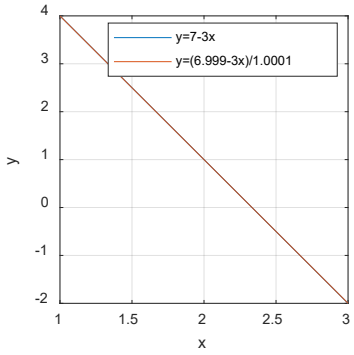


Figura 2. Representación de la solución del sistema $(x, y) = \left(\frac{8}{3}, -1\right)$.

Apartado c. Para obtener el número de condición de la matriz, utilizamos el comando cond.

```
>> cond(A)
```

El número de condición es 66667, que es un número bastante grande. Esta cantidad justifica la inestabilidad de la solución, puesto que ante pequeñas variaciones de algunos de los parámetros, la solución ha sufrido un cambio considerable.

Métodos iterativos para sistemas lineales

Vamos a transformar el sistema:

$$Ax = b$$

en un sistema de punto fijo equivalente, cuya solución se aproxima paso a paso.

Consideremos la partición de la matriz:

$$A = M + A - M$$

donde $M \neq A$ es una matriz invertible. Reemplazando en el sistema original:

$$Ax = b \leftrightarrow (M + A - M)x = b \leftrightarrow Mx + (A - M)x = b \leftrightarrow Mx = \underbrace{(M - A)}_{N} x + b$$

$$\leftrightarrow x = \underbrace{M^{-1}N}_{H} x + \underbrace{M^{-1}b}_{q}$$

Construimos el proceso iterativo y obtenemos:

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b = Hx^{(k)} + q, k = 0,1, \dots$$

Un método es estacionario si la matriz H es constante en todo el proceso.

10.3. Método de Jacobi

Tomemos la partición de matriz A en la suma de tres matrices:

$$A = L + D + U$$

donde L es la parte triangular estrictamente inferior de A, D es la diagonal principal de A y U es la parte triangular estrictamente superior de A.

El método de Jacobi es un método estacionario en que $M = D$ y $N = -(L + U)$, de forma que su expresión iterativa es:

$$x^{(k+1)} = -D^{-1}(L + U)x^{(k)} + D^{-1}b, k = 0,1, \dots$$

Obtención del esquema iterativo

Sea el sistema lineal:

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

en el que todos los coeficientes de la diagonal $a_{ii} \neq 0$. Despejando x_i de la i-ésima ecuación:

$$\begin{array}{rcl} x_1 & = & \dfrac{1}{a_{11}}(b_1 - a_{12}x_2 - \cdots - a_{1n}x_n) \\ x_2 & = & \dfrac{1}{a_{22}}(b_2 - a_{21}x_1 - \cdots - a_{2n}x_n) \\ & \vdots & \\ x_n & = & \dfrac{1}{a_{nn}}(b_n - a_{n1}x_1 - \cdots - a_{n,n-1}x_{n-1}) \end{array}$$

Matricialmente:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \frac{1}{a_{11}} & 0 & \cdots & 0 \\ 0 & \frac{1}{a_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{a_{nn}} \end{bmatrix} \left(- \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \right)$$

De modo que el esquema iterativo queda como:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}^{(k+1)} = \begin{bmatrix} \frac{1}{a_{11}} & 0 & \cdots & 0 \\ 0 & \frac{1}{a_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{a_{nn}} \end{bmatrix} \left(- \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}^{(k)} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \right)$$

Implementación en Matlab

En el **¡Error! No se encuentra el origen de la referencia.** se muestran los pasos para implementar en Matlab el método de Jacobi:

ENTRADA matriz A, vector b, vector estimación inicial `x0`, criterio de parada `tol`, número máximo de iteraciones `maxiter`.

1. Inicializar variables de control
2. Obtener las matrices L, D, D^{-1} y U a partir de A.
3. Obtener $M^{-1}=D^{-1}$ y `N=-(L+U)`.
4. Mientras no se cumpla el criterio de parada ni se alcance el número máximo de iteraciones permitido.
   - Calcular x = M-1 N x0 + M-1 b.
   - Actualizar criterios de parada y número de iteraciones.
   - Actualizar variables.

SALIDA vector solución x

Resolución de sistemas lineales con el método de Jacobi

Ejemplo 2. Sea el sistema:

$$\left.\begin{aligned} T_1 &= \frac{1}{4}(150 + T_2 + T_3) \\ T_2 &= \frac{1}{4}(50 + T_1 + T_4) \\ T_3 &= \frac{1}{4}(50 + T_1 + T_4 + T_5) \\ T_4 &= \frac{1}{4}(50 + T_2 + T_3 + T_6) \\ T_5 &= \frac{1}{4}(150 + T_3 + T_6) \\ T_6 &= \frac{1}{4}(50 + T_4 + T_5) \end{aligned}\right\}$$

Resuelve el sistema por los métodos de Gauss-Jordan y Jacobi, utilizando como estimación inicial $T^{(0)} = (50,50,50,50,50,50)$

y una tolerancia de 1e-3. Indica qué método emplea más tiempo en obtener la solución.

En primer lugar, tenemos que escribir el sistema como $Ax = b$.

$$\left.\begin{array}{c} 4T_1 - T_2 - T_3 = 150 \\ -T_1 + 4T_2 - T_4 = 50 \\ -T_1 + 4T_3 - T_4 - T_5 = 50 \\ -T_2 - T_3 + 4T_4 - T_6 = 50 \\ -T_3 + 4T_5 - T_6 = 150 \\ -T_4 - T_5 + 4T_6 = 50 \end{array}\right\}$$

$$\leftrightarrow \begin{bmatrix} 4 & -1 & -1 & 0 & 0 & 0 \\ -1 & 4 & 0 & -1 & 0 & 0 \\ -1 & 0 & 4 & -1 & -1 & 0 \\ 0 & -1 & -1 & 4 & 0 & -1 \\ 0 & 0 & -1 & 0 & 4 & -1 \\ 0 & 0 & 0 & -1 & -1 & 4 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}$$

$$= \begin{bmatrix} 150 \\ 50 \\ 50 \\ 50 \\ 150 \\ 50 \end{bmatrix}$$

Introducimos las matrices en Matlab por bloques:

```
>> m4=[4 -1;-1 4]; m1=[-1 0;0 -1]; m0=[0 0;0 0];
>> A=[m4 m1 m0; m1 m4 m1; m0 m1 m4];
>> b=[150 50 50 50 150 50]';
```

Ejecutamos los métodos de Gauss-Jordan y Jacobi:

```
>> solGJ=rref([A b])
>> solJac=Jacobi(A,b,50*ones(6,1),1e-3,100);
```

Las soluciones por ambos métodos son:

T_i	Gauss-Jordan	Jacobi
T_1	60.8696	60.8697
T_2	39.1304	39.1303
T_3	54.3478	54.3477
T_4	45.6522	45.6523
T_5	60.8696	60.8697
T_6	39.1304	39.1303

Tabla 1. Soluciones sistema de ecuaciones. Métodos de Gauss-Jordan y Jacobi.

Para comparar los tiempos de ejecución, generamos un programa que ejecute cada uno de los métodos 1.000 veces:

```
function [tJ,tGJ]=tiempos(A,b,x0,tol,maxiter)
tGJ=0; tJ=0;
for i=1:1000
   t0=tic;
   rref([A b]);
   tGJ=tGJ+toc(t0);
end
for i=1:1000
   t0=tic;
   Jacobi (A,b,x0,tol,maxiter);
   tJ=tJ+toc(t0);
End
```

El tiempo que tarda en ejecutarse Jacobi es 62.6 μs, mientras que Gauss-Jordan tarda 345.3 ms. Destacar que en cada ejecución del programa los tiempos pueden variar, pero la relación entre ambos tiempos debe ser del mismo orden.

10.4. Método de Gauss-Seidel

El método de Gauss-Seidel propone **utilizar los nuevos valores de las incógnitas a medida que se van calculando en cada iteración**. Así, en el paso k, una vez calculado el valor de $x_1^{(k+1)}$, este se utiliza para actualizar el valor de $x_2^{(k+1)}$, y así sucesivamente.

Tomemos la misma partición de la matriz A que para el método de Jacobi:

$$A = L + D + U$$

En el método de Gauss-Seidel tomamos $M = D + L$ y $N = -U$, de forma que su expresión iterativa es:

$$x^{(k+1)} = -(D+L)^{-1}Ux^{(k)} + (D+L)^{-1}b, k = 0,1,\ldots$$

Obtención del esquema iterativo

Sea el sistema lineal:

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

en el que todos los coeficientes de la diagonal $a_{ii} \neq 0$. Despejando x_i de la i-ésima ecuación, obtenemos el esquema iterativo:

$$\begin{array}{rcl} x_1^{(k+1)} & = & \dfrac{1}{a_{11}}\left(-a_{12}x_2^{(k)} - \cdots - a_{1n}x_n^{(k)} + b_1\right) \\ x_2^{(k+1)} & = & \dfrac{1}{a_{22}}\left(a_{21}x_1^{(k+1)} - \cdots - a_{2n}x_n^{(k)} + b_2\right) \\ & \vdots & \\ x_n^{(k+1)} & = & \dfrac{1}{a_{nn}}\left(b_n - a_{n1}x_1^{(k+1)} - \cdots - a_{n,n-1}x_{n-1}^{(k+1)}\right) \end{array}$$

De este modo, en cada iteración debemos resolver el sistema:

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

Implementación en Matlab

En el siguiente algoritmo se muestran los pasos para implementar en Matlab el método de Gauss-Seidel.

ENTRADA matriz A, vector b, vector estimación inicial `x0`, criterio de parada `tol`, número máximo de iteraciones `maxiter`.

1. Inicializar variables de control.

2. Obtener las matrices L, D, D^{-1} y U a partir de A.
3. Obtener $M^{-1}=(D+L)^{-1}$ y N=-U.
4. Mientras no se cumpla el criterio de parada ni se alcance el número máximo de iteraciones permitido.
   - Resolver $(D+L)\, x = b - U\, x0$.
   - Actualizar criterios de parada y número de iteraciones.
   - Actualizar variables.

SALIDA vector solución x

Resolución de sistemas lineales con el método de Gauss-Seidel

Ejemplo 3. Sea el sistema:

$$\left.\begin{aligned} T_1 &= \frac{1}{4}(150 + T_2 + T_3) \\ T_2 &= \frac{1}{4}(50 + T_1 + T_4) \\ T_3 &= \frac{1}{4}(50 + T_1 + T_4 + T_5) \\ T_4 &= \frac{1}{4}(50 + T_2 + T_3 + T_6) \\ T_5 &= \frac{1}{4}(150 + T_3 + T_6) \\ T_6 &= \frac{1}{4}(50 + T_4 + T_5) \end{aligned}\right\}$$

Resuelve el sistema por los métodos de Gauss-Jordan y Gauss-Seidel, utilizando como estimación inicial $T^{(0)} = (50,50,50,50,50,50)$ y una tolerancia de 1e-3. Indica qué método emplea más tiempo en obtener la solución.

El planteamiento del sistema y la introducción de las matrices ya fue detallado en el Ejemplo 2. Ejecutamos los métodos de Gauss-Jordan y Gauss-Seidel.

```
>> solGJ=rref([A b])
>>       solGS=GaussSeidel(A,b,50*ones(6,1),1e-
3,100);
```

Las soluciones por ambos métodos son:

T_i	Gauss-Jordan	Gauss-Seidel
T_1	60.8696	60.8697
T_2	39.1304	39.1305
T_3	54.3478	54.3480
T_4	45.6522	45.6523
T_5	60.8696	60.8696
T_6	39.1304	39.1305

Tabla 2. Soluciones del sistema. Métodos de Gauss-Jordan y Gauss-Seidel.

Para comparar los tiempos de ejecución, generamos un programa similar al del Ejemplo 2. El tiempo que tarda en ejecutarse Gauss-Seidel es 102.7 ms, mientras que Gauss-Jordan tarda 335.2 ms.

10.5. Métodos de sobrerrelajación

Para introducir los métodos de relajación, necesitamos **hacer que los errores sean cada vez más pequeños en cada iteración**. Así que sea $\bar{x} \in \mathbb{R}^n$ una aproximación a la solución del sistema $Ax = b$, el vector residual para $\bar{x}$ es:

$$r = b - A\,\bar{x}$$

Obtención del esquema iterativo

El método de Jacobi relajado, denotado por JSOR, tiene la expresión iterativa:

$$x^{(k+1)} = x^{(k)} + \omega D^{-1} r^{(k)}$$

Si el método de Jacobi converge, entonces el método JSOR converge para $0 \leq \omega \leq 1$.

A continuación, vamos a presentar un método de sobrerrelajación. Denotado por SOR1, tiene la expresión:

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

Desarrollando escalarmente la expresión anterior, podemos llegar a:

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

Si $\bar{x}^{(k+1)}$ denota el iterado $k+1$ del método de Gauss-Seidel, la expresión iterativa vectorial del método SOR1 se puede escribir como:

$$x^{(k+1)} = (1 - \omega)x^{(k)} + \omega\, \bar{x}^{(k+1)}, k = 0,1,2, \ldots$$

Teorema 2 (Ostrowski-Reich). Si A es una matriz definida positiva y $0 < \omega < 2$, entonces el método SOR1 converge para cualquier elección de la aproximación inicial $x^{(0)}$.

Teorema 3. Si A es definida positiva y tridiagonal, entonces:

$$\rho(H_{GS}) = \left[\rho(H_J)\right]^2 < 1$$

Además, la elección óptima de ω para el método SOR1 es:

$$\omega = \frac{2}{1 + \sqrt{1 - \left[\rho(H_J)\right]^2}}.$$

Ejemplo 4. Encuentra el valor óptimo de ω para el método SOR1 de la matriz:

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}$$

La matriz A es positiva ya que para cualquier $x \neq 0, x^T Ax > 0$. Además, se observa claramente que es tridiagonal, por lo que podemos aplicar el Teorema 3. Obtengamos la matriz $H_J = -D^{-1}(L + U)$.

$$D = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix} \rightarrow D^{-1} = \begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{bmatrix}, L + U = \begin{bmatrix} 0 & 3 & 0 \\ 3 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$H_J = -D^{-1}(L + U) = \begin{bmatrix} 0 & -3/4 & 0 \\ -3/4 & 0 & 1/4 \\ 0 & 1/4 & 0 \end{bmatrix}$$

Calculemos el mayor valor absoluto de los valores propios de H_J:

```
>>[~,l]=eig(H);
>> rho=max(abs(l(:)));
```

El valor de $\rho(H_J)$ es 0.7906. De modo que el valor óptimo de ω es:

$$\omega = \frac{2}{1+\sqrt{1-[0.7906]^2}} \approx 1.24$$

Resolución de sistemas lineales con el método SOR1

Ejemplo 5. Sea el sistema lineal $Ax = b$, cuyas matrices son:

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}, b = \begin{bmatrix} 24 \\ 30 \\ -24 \end{bmatrix}$$

Obtén la solución aplicando los métodos de Gauss-Seidel y SOR1 con $\omega = 1.25$ utilizando $x^{(0)} = (1,1,1)$ para ambos casos y utilizando como criterio de parada:

$$\left\| \bar{x} - x^{(k)} \right\|_\infty < 10^{-7}$$

siendo $\bar{x} = (3{,}4, -5)$ la solución del sistema. Indica el número de iteraciones requeridas para alcanzar la solución por ambos métodos. Indica los resultados con 7 decimales para las 4 primeras iteraciones.

Introducimos las matrices en Matlab y aplicamos los métodos de Gauss-Seidel y SOR1. Los resultados de Gauss-Seidel se muestran en la siguiente tabla.

k	1	2	3	4
$x_1^{(k)}$	5.250000	3.1406250	3.0878906	3.0549316
$x_2^{(k)}$	3.812500	3.8828125	3.9267578	3.9542236
$x_3^{(k)}$	-5.046875	-5.0292969	-5.0183105	-5.0114441

Tabla 3. Resultados del ejemplo 5 con el método de Gauss-Seidel.

Los resultados de SOR1 se muestran en la siguiente tabla:

k	1	2	3	4
$x_1^{(k)}$	6.312500	2.6223145	3.1333027	2.9570512
$x_2^{(k)}$	3.5195313	3.9585266	4.0102646	4.0074838

$x_3^{(k)}$	-6.6501465	-4.6004238	-5.0966863	-4.9734897

Tabla 4. Resultados del ejemplo 5 con el método SOR1.

El método de Gauss-Seidel ha tardado 34 iteraciones en converger, mientras que el método SOR1 ha tardado 14 iteraciones.

10.6. Convergencia de los métodos iterativos

En este último apartado del tema vamos a estudiar algunos aspectos acerca de la convergencia de los métodos iterativos que hemos utilizado para resolver los sistemas de ecuaciones lineales.

Teorema 4. Sea A una matriz invertible. Un método iterativo estacionario converge, sea cual sea la estimación inicial $x^{(0)} \in \mathbb{R}^n$, a la solución exacta del sistema lineal si y solo si:

$$\rho(H) < 1$$

donde ρ representa el mayor valor propio en valor absoluto de la matriz de iteración.

Ejemplo 6. Sea el sistema de ecuaciones lineales:

$$\left.\begin{array}{c} x + 2y - z = -1 \\ 2x - 3y + z = -6 \\ x + 4z = 2 \end{array}\right\}$$

En base al Teorema 4, si aplicamos el método de Jacobi, ¿vamos a obtener la solución exacta del sistema, independientemente de la estimación inicial?

La expresión iterativa del método de Jacobi es:

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

El método iterativo lo podemos expresar como:

$$x^{(k+1)} = Hx^{(k)} + q$$

Identificando términos, $H = -D^{-1}(L+U)$. Por tanto, obtengamos la matriz H del sistema. Por un lado, tenemos que:

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 2 & -3 & 1 \\ 1 & 0 & 4 \end{bmatrix}$$

de forma que:

$$L+U = \begin{bmatrix} 0 & 2 & -1 \\ 2 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 4 \end{bmatrix} \rightarrow D^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1/3 & 0 \\ 0 & 0 & 1/4 \end{bmatrix}$$

Por tanto:

$$H = -D^{-1}(L+U) = \begin{bmatrix} 0 & -2 & 1 \\ 2/3 & 0 & 1/3 \\ -1/4 & 0 & 0 \end{bmatrix}$$

Obtenemos los valores absolutos de valores propios de H como:

```
>> [~,l]=eig(H);
>> rho=abs(l);
```

Vemos que $\rho = (1.2626, 1.2626, 0.1045)$. Como el mayor de estos valores propios es superior a la unidad, no podemos garantizar que el método de Jacobi converja para cualquier estimación inicial.

Una matriz A de tamaño $n \times n$ se dice estrictamente diagonal dominante si:

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|, \forall i = 1, 2, \ldots, n$$

Teorema 5. Si la matriz A de un sistema lineal $Ax = b$ es estrictamente diagonal dominante, entonces los métodos de Jacobi y Gauss-Seidel son convergentes independientemente de la estimación inicial $x^{(0)}$.

Ejemplo 7. Sea el sistema de ecuaciones lineales:

$$\left.\begin{array}{c} x + 2y - z = -1 \\ 2x - 3y + z = -6 \\ x + 4z = 2 \end{array}\right\}$$

En base al Teorema 5, si aplicamos los métodos de Jacobi o Gauss-Seidel, ¿vamos a obtener la solución exacta del sistema, independientemente de la estimación inicial?

La matriz A del sistema $Ax = b$ es:

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 2 & -3 & 1 \\ 1 & 0 & 4 \end{bmatrix}$$

Obtengamos la desigualdad para $i = 1,2,3$.

$$i = 1: \sum_{j=2}^{3} |a_{1j}| = 2 + 1 = 3$$

Como $|a_{11}| = 1 < 3$, no podemos garantizar la convergencia de los métodos sobre este problema.

Teorema 6. Si $\|H\| < 1$ para cualquier norma matricial, entonces la sucesión $\{x^{(k)}\}$ obtenida por un método iterativo estacionario converge a la solución del sistema para cualquier estimación inicial $x^{(0)}$, y se satisfacen las cotas de error:

$$\|\bar{x} - x^{(k)}\| \leq \|H\|^{(k)} \|\bar{x} - x^{(0)}\|$$

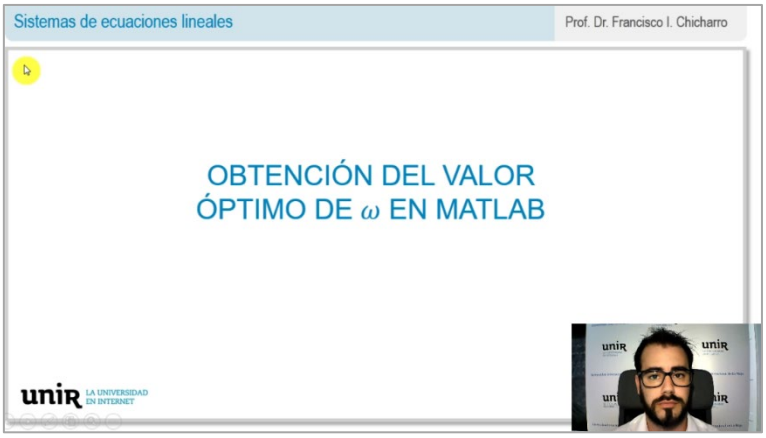
$$\|\bar{x} - x^{(k)}\| \leq \frac{\|H\|^{(k)}}{1 - \|H\|} \|x^{(1)} - x^{(0)}\|$$

Lo + recomendado

Lecciones magistrales

Obtención del valor óptimo de ω en Matlab

En esta lección magistral vamos a generar un código que nos permita obtener el valor óptimo del parámetro ω que caracteriza a los métodos SOR. Para ello utilizaremos el software Matlab.

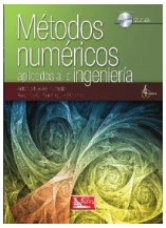


Accede a la lección magistral a través del aula virtual

No dejes de leer

Matrices y sistemas de ecuaciones lineales

Nieves, A. (2014). «Matrices y sistemas de ecuaciones lineales». En Autores, *Métodos numéricos: aplicados a la ingeniería*. México D. F.: Editorial Patria.



En el capítulo 3 de este libro puedes encontrar más teoría acerca de los **métodos** que hemos visto a lo largo del tema. Además, muestra **ejemplos de implementación de diferentes programas**, y ejercicios resueltos

Accede al libro a través de la Biblioteca Virtual de UNIR

Sistemas de ecuaciones lineales

Cordero, A., Hueso, J. L., Martínez, E. y Torregrosa, J. R. (2006). «Sistemas de ecuaciones lineales». En Autores, *Problemas resueltos de métodos numéricos*. Madrid: Thomson.

En el capítulo 6 de este libro se desarrollan los **métodos de Jacobi, Gauss-Seidel y sobrerrelajación**. Este documento puede servirte como referencia para la realización de ejercicios y para la revisión de la teoría.

Matrices especiales y el método de Gauss-Seidel

Chapra, S. C. y Canale, R. P. (2007). «Matrices especiales y el método de Gauss-Seidel». En Autores, *Métodos numéricos para ingenieros* (5ª. Ed.). McGraw-Hill.

En el capítulo 11 del libro Chapra y Canale puedes mejorar tus conocimientos acerca del **método de Gauss-Seidel**. Este libro, caracterizado por su simplicidad a la hora de presentar los conceptos, te permitirá obtener una mejor comprensión del apartado correspondiente a Gauss-Seidel.

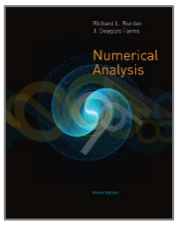
Accede al libro a través de la Biblioteca Virtual de UNIR

+ Información

A fondo

Técnicas iterativas en Álgebra Matricial

Burden, R. L. y Faires, J. D. (2011). « Técnicas iterativas en Álgebra Matricial». En Autores, *Numerical analysis* (9ª Ed). Boston: Brooks/Cole CENGAGE learning.



En el capítulo 7 de *Numerical Analysis* encontrarás toda la base teórica necesaria para obtener los métodos numéricos que hemos presentado en este tema.

The «Gauss-Seidelization» of iterative methods for solving nonlinear equations in the complex plane

Gutiérrez, J. M., Magreñán, Á. A. y Varona, J. L. (2011). The «Gauss-Seidelization» of iterative methods for solving nonlinear equations in the complex plane. *Applied Mathematics and Computation* 218, 2467—2479.

De forma similar al **procedimiento de Gauss-Seidel** para sistemas lineales, en este artículo de investigación se estudia la aplicación de dicho proceso para aplicarlo sobre **ecuaciones no lineales**.

Webgrafía

Método de Crout

En anteriores temas hemos utilizado el método de Crout para la resolución de sistemas lineales con algunas condiciones particulares de las matrices implicadas en el sistema $Ax = b$. En este recurso web tienes un ejemplo del uso del método de Crout para resolver un sistema de ecuaciones lineales.

Accede a la página web a través del aula virtual o desde la siguiente dirección web:

https://numericalmethodsece101.weebly.com/croutrsquos-method.html

Test

1. El criterio del residuo es fiable si el número de condición de la matriz A:

A. Es cero.

B. Es mayor que 1.

C. No es muy grande.

2. Si una matriz A tiene un número de condición grande:

A. Pequeños cambios en b provocan grandes cambios en la solución.

B. Pequeños cambios en b provocan pequeños cambios en la solución.

C. Todas las anteriores son falsas.

3. Un método es estacionario si:

A. H es constante en todo el proceso.

B. M es constante en todo el proceso.

C. A es constante en todo el proceso.

4. ¿En qué método se descompone la matriz A como $L + D + U$?

A. Jacobi.

B. Gauss-Seidel.

C. Todas las anteriores son correctas.

5. ¿Qué método tarda más en ejecutarse: Gauss-Jordan o Jacobi?

A. Gauss-Jordan.

B. Jacobi.

C. No es posible determinarlo.

6. En el método de Jacobi:

A. se calcula el valor de $x_{i+1}^{(k+1)}$ a partir de $x_1^{(k)}, \dots, x_i^{(k)}, x_{i+2}^{(k)}, \dots x_n^{(k)}$

B. se calcula el valor de $x_{i+1}^{(k+1)}$ a partir de $x_1^{(k+1)}, \dots, x_i^{(k+1)}, x_{i+2}^{(k)}, \dots x_n^{(k)}$

C. se calcula el valor de $x_{i+1}^{(k+1)}$ a partir de $x_1^{(k-1)}, \dots, x_i^{(k-1)}, x_{i+2}^{(k)}, \dots x_n^{(k)}$

7. En el método de Gauss-Seidel:

A. se calcula el valor de $x_{i+1}^{(k+1)}$ a partir de $x_1^{(k)}, \dots, x_i^{(k)}, x_{i+2}^{(k)}, \dots x_n^{(k)}$

B. se calcula el valor de $x_{i+1}^{(k+1)}$ a partir de $x_1^{(k+1)}, \dots, x_i^{(k+1)}, x_{i+2}^{(k)}, \dots x_n^{(k)}$

C. se calcula el valor de $x_{i+1}^{(k+1)}$ a partir de $x_1^{(k-1)}, \dots, x_i^{(k-1)}, x_{i+2}^{(k)}, \dots x_n^{(k)}$

8. El método SOR1 converge para cualquier estimación inicial $x^{(0)}$ si A es definida positiva y:

A. $\omega \in [0,1]$

B. $\omega \in [1,2]$

C. $\omega \in [0,2]$

9. El método de Jacobi converge para cualquier estimación inicial $x^{(0)}$ si

A. $\rho(H_J) > 0$

B. $\rho(H_J) < 1$

C. No se puede garantizar la convergencia en ningún caso.

10. El método de Gauss-Seidel converge para cualquier estimación inicial $x^{(0)}$ si A es:

A. Estrictamente diagonal dominante.

B. Semidefinida positiva.

C. Tridiagonal superior.