



INFORMÁTICA

Sistemas lineales

$$Ax = b$$



Clase

1. Resolución de sistemas de ecuaciones lineales: preliminares
2. Método directo y exacto: Gauss
3. Método directo y exacto (II): descomposición LU
4. Métodos indirectos: Jacobi, Gauss-Seidel

Sistemas lineales **PRELIMINARES**

Matriz
de coeficientes

$$\mathbf{Ax} = \mathbf{b}$$

Vector de incógnitas

Sistema de ecuaciones lineales

Recordatorio de álgebra lineal...

Si $A \in \mathbb{R}^{n \times n}$, entonces las siguientes afirmaciones son equivalentes:

1. Para cualquier $b \in \mathbb{R}^n$, el sistema $Ax = b$ tiene solución.
2. Si $Ax = b$ tiene solución, ésta es única.
3. Para cualquier $x \in \mathbb{R}^n$, $Ax = 0 \implies x = 0$.
4. Las columnas (filas) de A son linealmente independientes.
5. Existe A^{-1} matriz inversa de A tal que $AA^{-1} = A^{-1}A = I$ (I matriz identidad de orden n).
6. $\det(A) = |A| \neq 0$.

La primera opción que uno se plantea es

$$x = A^{-1}b$$

- ✓ No es eficiente (demasiadas operaciones)
- ✓ Si el determinante de A es próximo a cero, el error de redondeo puede ser muy grande, y esto es difícil de estimar numéricamente ($\det(\gamma A) = \gamma^n \det(A)$)

→ Se requieren métodos numéricos alternativos:

La primera opción que uno se plantea es

$$\mathbf{x}_0 = A^{-1}\mathbf{b}$$

- ✓ No es eficiente (demasiadas operaciones)
- ✓ Si el determinante de A es próximo a cero, el error de redondeo puede ser muy grande.

→ Se requieren métodos numéricos alternativos:

❑ métodos **directos** (son exactos (no tienen asociado error de truncamiento), y son usados cuando la mayoría de los coeficientes de A son distintos de cero y las matrices no son demasiado grandes). Suelen ser algoritmos 'complicados de implementar'

La primera opción que uno se plantea es

$$\mathbf{x}_0 = A^{-1}\mathbf{b}$$

- ✓ No es eficiente (demasiadas operaciones)
- ✓ Si el determinante de A es próximo a cero, el error de redondeo puede ser muy grande.

→ Se requieren métodos numéricos alternativos:

❑ métodos **directos** (son exactos (no tienen asociado error de truncamiento), y son usados cuando la mayoría de los coeficientes de A son distintos de cero y las matrices no son demasiado grandes). Suelen ser algoritmos 'complicados de implementar'

❑ métodos **indirectos o iterativos** (tienen asociado un error de truncamiento y se usan preferiblemente para matrices grandes ($n \gg 1000$) cuando los coeficientes de A son la mayoría nulos – matrices *sparse*–). Algoritmos sencillos de implementar que requiere aproximación inicial y que en general no tiene por qué converger (requieren análisis de convergencia previo).

Ejemplos

Métodos directos

- ☐ método de eliminación de Gauss (*llevar A a triangular*)
- ☐ método de descomposición LU ($A=LU$, donde L triangular inferior y U triangular superior)
- ☐ método de Cholesky (LU para matrices simétricas)

Métodos iterativos

- ☐ método de Jacobi
- ☐ método de Gauss-Seidel
- ☐ método SOR

METODOS DIRECTOS

Los *métodos directos* de resolución de sistemas lineales de ecuaciones son aquellos que permiten obtener la solución después de un número finito de operaciones aritméticas. Este número de operaciones es función del tamaño de la matriz.

Si los ordenadores pudieran almacenar y operar con todas las cifras de los números reales, es decir, si emplearan una aritmética exacta, con los métodos directos se obtendría la solución exacta del sistema en un número finito de pasos. Puesto que los ordenadores tienen una precisión finita, los errores de redondeo se propagan y la solución numérica obtenida siempre difiere de la solución exacta. La cota del error, para una matriz y término independiente dados, se asocia por lo general al número de operaciones de cada método. Se pretende, por lo tanto, obtener métodos con el mínimo número de operaciones posible.

método de eliminación de Gauss

$$\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \dots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & & & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} & b_n^{(1)} \end{pmatrix}$$



$$\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & a_{nn}^{(n)} & b_n^{(n)} \end{pmatrix}$$

$(A|\mathbf{b})$

LA IDEA

- Primero, mediante operaciones elementales por filas, se transforma la matriz ampliada en una matriz triangular superior (equivalente a la matriz de partida) (PASO DE ELIMINACION)
- Segundo, se resuelve dicho sistema obteniendo las incógnitas, empezando por la n-esima –la última- y acabando con la primera (PASO DE SUSTITUCION).

$$(A|\mathbf{b})$$
$$\left(\begin{array}{ccccc} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \dots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & & & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} & b_n^{(1)} \end{array} \right)$$

suponiendo que sea $a_{11}^{(1)} \neq 0$, tomamos ese elemento como *pivote* y “hacemos ceros” por debajo de él. Para ello, modificaremos las filas de la matriz en la forma

$$F_i \rightarrow F_i - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} F_1, \quad i = 2, \dots, n$$

$a_{11}^{(1)} = 0$, al no estar definidos los multiplicadores, se busca la primera fila para la cual $a_{p1}^{(1)} \neq 0$ y se hace el cambio $F_1 \leftrightarrow F_p$ (a este intercambio de filas se le conoce como *pivoteo*), para después realizar las operaciones (2.3). Con todo esto, la matriz del sistema se transforma en

$$\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & & & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \dots & a_{nn}^{(2)} & b_n^{(2)} \end{pmatrix} \quad (2.4)$$

donde todos los elementos a partir de la segunda fila han cambiado (de ahí el superíndice $^{(2)}$). En la siguiente etapa, se toma $a_{22}^{(2)}$ como pivote en la segunda columna, o se pivota si $a_{22}^{(2)} = 0$ (la condición $\det A \neq 0$ garantiza que podemos encontrar pivotes en todas las columnas de la matriz), para después realizar las operaciones $F_i \rightarrow F_i - m_{i2}F_2$, con $m_{i2} = a_{i2}^{(2)}/a_{22}^{(2)}$, para $i = 3, \dots, n$, consiguiendo de esta forma ceros por debajo del segundo pivote.

REPETIMOS ESTE PROCESO N-1 VECES HASTA LLEGAR A

$$\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & a_{nn}^{(n)} & b_n^{(n)} \end{pmatrix}$$

que se corresponde con el sistema de ecuaciones

$$\left. \begin{aligned} a_{11}^{(1)} x_1 + a_{12}^{(1)} x_2 + \dots + a_{1n-1}^{(1)} x_{n-1} + a_{1n}^{(1)} x_n &= b_1^{(1)} \\ a_{22}^{(2)} x_2 + \dots + a_{2n-1}^{(2)} x_{n-1} + a_{2n}^{(2)} x_n &= b_2^{(2)} \\ &\vdots \\ a_{n-1n-1}^{(n-1)} x_{n-1} + a_{n-1n}^{(n-1)} x_n &= b_{n-1}^{(n-1)} \\ a_{nn}^{(n)} x_n &= b_n^{(n)} \end{aligned} \right\}$$

Finalmente, por sustitución regresiva resolvemos el sistema.

$$\mathbf{A} = \mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & \cdots & \cdots & u_{1n} \\ 0 & u_{22} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & u_{n-1,n-1} & u_{n-1,n} \\ 0 & \cdots & \cdots & 0 & u_{nn} \end{pmatrix}$$

$$x_n = b_n / u_{nn}$$

$$x_i = \left(b_i - \sum_{j=i+1}^n u_{ij} x_j \right) / u_{ii} \quad i = n-1, n-2, \dots, 1$$

Se puede comprobar que el número de operaciones aritméticas elementales que se realizan siguiendo el método de eliminación de Gauss en un sistema $n \times n$ es $O(n^3)$.^a

ELIMINACIÓN DE GAUSS: ALGORITMO COMPLETO

Paso de eliminación

$$\begin{aligned} a_{ij}^{(k)} &= a_{ij}^{(k-1)} - m_{ik} a_{kj}^{(k-1)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)} \\ b_i^{(k)} &= b_i^{(k-1)} - m_{ik} b_k^{(k-1)} = b_i^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} b_k^{(k-1)} \end{aligned} \quad \begin{cases} k = 1, \dots, n-1 \\ i = k+1, \dots, n \\ j = k+1, \dots, n \end{cases}$$

Paso de sustitución (cambio de notación $a \rightarrow u$)

$$x_n = b_n / u_{nn}$$

$$x_i = \left(b_i - \sum_{j=i+1}^n u_{ij} x_j \right) / u_{ii} \quad i = n-1, n-2, \dots, 1$$

método de descomposición *LU*

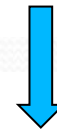
$$A = LU$$



$$Ax = LUx = b$$

$$Ux = y$$

$$Ly = b$$



LA IDEA

- Primero, a partir de la matriz A se calcula aquella matriz triangular inferior L y aquella matriz triangular superior U (con 1's en la diagonal) tal que $A=LU$ (PASO DE DESCOMPOSICION)
- Así, el sistema $A\mathbf{x}=\mathbf{b}$ pasa a ser $LU\mathbf{x}=\mathbf{b}$. Primero hacemos el cambio $U\mathbf{x}=\mathbf{y}$, que introducimos y el sistema resulta $L\mathbf{y}=\mathbf{b}$. Como L es triangular, fácilmente calculamos \mathbf{y} . Finalmente, introducimos este resultado en $U\mathbf{x}=\mathbf{y}$, y como U es triangular, fácilmente calculamos \mathbf{x} . (PASO DE SUSTITUCION).

PASO DE DESCOMPOSICION : L, U \ A=LU

$$l_{11} = a_{11} \quad u_{11} = 1$$

$$k = 1, \dots, n - 1$$

$$\left\{ \begin{array}{l} u_{1,k+1} = a_{1,k+1} / l_{11} \\ u_{i,k+1} = \left(a_{i,k+1} - \sum_{j=1}^{i-1} l_{ij} u_{j,k+1} \right) / l_{ii} \quad i = 2, \dots, k \\ l_{k+1,1} = a_{k+1,1} \\ l_{k+1,i} = a_{k+1,i} - \sum_{j=1}^{i-1} u_{ji} l_{k+1,j} \quad i = 2, \dots, k \\ u_{k+1,k+1} = 1 \\ l_{k+1,k+1} = a_{k+1,k+1} - \sum_{i=1}^k l_{k+1,i} u_{i,k+1} \end{array} \right.$$

PASO DE SUSTITUCION : $Ly=b$ (cambio de notación $y \rightarrow x$)

MATRIZ TRIANGULAR INFERIOR

$$A = L = \begin{pmatrix} l_{11} & 0 & \cdots & \cdots & 0 \\ l_{21} & l_{22} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & l_{n-1,n-1} & 0 \\ l_{n1} & \cdots & \cdots & l_{n,n-1} & l_{nn} \end{pmatrix}$$

Se aplica un algoritmo similar al anterior que se denomina de *sustitución hacia adelante*:

$$x_1 = b_1 / l_{11}$$

$$x_i = \left(b_i - \sum_{j=1}^{i-1} l_{ij} x_j \right) / l_{ii} \quad i = 2, \dots, n$$

PASO DE SUSTITUCION : $Ux=y$ (cambio de notación $y \rightarrow b$)

MATRIZ TRIANGULAR SUPERIOR

$$A = U = \begin{pmatrix} u_{11} & u_{12} & \cdots & \cdots & u_{1n} \\ 0 & u_{22} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & u_{n-1,n-1} & u_{n-1,n} \\ 0 & \cdots & \cdots & 0 & u_{nn} \end{pmatrix} \quad (6.6)$$

En este caso la solución de la última ecuación es trivial $x_n = b_n / u_{nn}$. Una vez conocido x_n , la penúltima ecuación (la $n - 1$) sólo tiene una incógnita que se deduce de forma sencilla. Conocidos ahora x_n y x_{n-1} , se pasa a la ecuación anterior (la $n - 2$) y se resuelve para su única incógnita, x_{n-2} . Retrocediendo progresivamente se obtiene el algoritmo de *sustitución hacia atrás* que se escribe de la siguiente forma

$$x_n = b_n / u_{nn}$$
$$x_i = \left(b_i - \sum_{j=i+1}^n u_{ij}x_j \right) / u_{ii} \quad i = n - 1, n - 2, \dots, 1 \quad (6.7)$$

Comentarios práctica (MÉTODOS DIRECTOS)

- Cuidado con la información de entrada en las subrutinas!!
- Subrutina LU: algoritmo se resuelve secuencialmente!!
- Cálculo de determinantes con LU: **trivial !!!!!**

If B is also an $n \times n$ matrix, then $\det AB = \det A \det B$.

If A is an upper triangular, lower triangular, or diagonal matrix, then

$$\det A = a_{11} \cdot a_{22} \cdots a_{nn}.$$

$$\rightarrow \det(A) = \det(LU) = \det(L)\det(U)$$

Matrices estrictamente diagonal dominantes: propiedad suficiente para Gauss sin pivote y para LU (aunque no necesarias)

The $n \times n$ matrix A is said to be strictly diagonally dominant when

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

holds for each $i = 1, 2, \dots, n$.

[Strictly Diagonally Dominant Matrices] A strictly diagonally dominant matrix A **has an inverse**. Moreover, in this case, Gaussian elimination can be performed on any linear system of the form $Ax = b$ to obtain its unique solution without row or column interchanges, and the computations are stable with respect to the growth of round-off error.

METODOS **INDIRECTOS**

GENERALIDADES

-¿Qué son los métodos indirectos/iterativos?

Esquemas numéricos para resolver sistemas de ecuaciones lineales, basados en la aplicación de un algoritmo a partir de una solución inicial, que se repite iterativamente hasta que un criterio de parada (convergencia) detiene el algoritmo (número de pasos desconocido a priori).

- ¿Para qué se usan los métodos indirectos/iterativos?

Eficientes para sistemas grandes con matrices con un elevado número de ceros (matrices sparse), que aparecen en problemas de física e ingeniería, asociados a la resolución de ecuaciones diferenciales en derivadas parciales.

- Convergencia y problemas mal condicionados

Estos métodos no siempre convergen, y además, en ocasiones pequeñas variaciones pueden introducir grandes errores (problema de mal condicionamiento).

GENERALIDADES- esquema general

Sea el problema $\mathbf{Ax}=\mathbf{b}$

En la iteración k-ésima, el esquema numérico es

$$\mathbf{x}^{(k)} = \mathbf{T} \mathbf{x}^{(k-1)} + \mathbf{c}$$

Donde \mathbf{x} es el vector solución, \mathbf{T} una matriz $n \times n$ y \mathbf{c} un vector, que dependen todos ellos de \mathbf{A} y \mathbf{b}

GENERALIDADES- esquema general

- Si el esquema converge, el orden de convergencia es LINEAL (se necesitarán bastantes pasos para llegar a convergencia)
- Por eso, se aplican estos métodos cuando las matrices tienen muchos ceros (cada paso tiene poco coste computacional)

GENERALIDADES- esquema general

-criterio de parada: ¿noción de distancia entre vectores?

norma vectorial

$$\|x\|_2 = \left\{ \sum_{i=1}^n x_i^2 \right\}^{1/2} \quad \text{and} \quad \|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

y matricial

$[l_\infty$ Norm of a Matrix]

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|.$$

If

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 3 & -1 \\ 5 & -1 & 1 \end{bmatrix},$$

then

$$\sum_{j=1}^3 |a_{1j}| = |1| + |2| + |-1| = 4, \quad \sum_{j=1}^3 |a_{2j}| = |0| + |3| + |-1| = 4,$$

and

$$\sum_{j=1}^3 |a_{3j}| = |5| + |-1| + |1| = 7.$$

So

$$\|A\|_{\infty} = \max\{4, 4, 7\} = 7. \quad \square$$

GENERALIDADES- esquema general

Distancia entre vectores

[Distance **Between Vectors**] If $\mathbf{x} = (x_1, x_2, \dots, x_n)^t$ and $\mathbf{y} = (y_1, y_2, \dots, y_n)^t$ are vectors in \mathbb{R}^n , the l_2 and l_∞ distances between \mathbf{x} and \mathbf{y} are defined by

$$\|\mathbf{x} - \mathbf{y}\|_2 = \left\{ \sum_{i=1}^n (x_i - y_i)^2 \right\}^{1/2} \quad \text{and} \quad \|\mathbf{x} - \mathbf{y}\|_\infty = \max_{1 \leq i \leq n} |x_i - y_i|.$$

Criterio de parada



$$\frac{\|\mathbf{x}^{(N)} - \mathbf{x}^{(N-1)}\|}{\|\mathbf{x}^{(N)}\|} < \text{TOL}$$

MÉTODO DE JACOBI

Considérese el sistema de ecuaciones

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \quad (2.24)$$

donde, supuesto que $a_{ii} \neq 0 \ \forall i$ podemos despejar de cada ecuación la incógnita correspondiente

$$\begin{cases} x_1 = (b_1 - a_{12}x_2 - \cdots - a_{1n}x_n)/a_{11} \\ x_2 = (b_2 - a_{21}x_1 - \cdots - a_{2n}x_n)/a_{22} \\ \vdots \\ x_n = (b_n - a_{n1}x_1 - \cdots - a_{nn}x_n)/a_{nn} \end{cases} \quad (2.25)$$

MÉTODO DE JACOBI

¡ Planteamos el siguiente esquema iterativo ! (si alguno de los elementos diagonales es nulo, reordenamos la matriz)

$$\left\{ \begin{array}{l} x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - \cdots - a_{1n}x_n^{(k)})/a_{11} \\ x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k)} - \cdots - a_{2n}x_n^{(k)})/a_{22} \\ \vdots \\ x_n^{(k+1)} = (b_n - a_{n1}x_1^{(k)} - \cdots - a_{n,n-1}x_{n-1}^{(k)})/a_{nn} \end{array} \right. \quad (2.26)$$

MÉTODO DE JACOBI

Podemos reexpresar matricialmente el proceso (2.26) reescribiendo la matriz del sistema como $A = T_I + D + T_S$, donde $D = \text{diag}(a_{11}, a_{22}, \dots, a_{nn})$, y

$$T_I = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ a_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn-1} & 0 \end{pmatrix}, \quad T_S = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a_{n-1n} \\ 0 & \cdots & \cdots & 0 \end{pmatrix}$$

De esta forma, la ecuación (2.26) podría expresarse como $\mathbf{x}^{(k+1)} = D^{-1}(\mathbf{b} - (T_I + T_S)\mathbf{x}^{(k)})$ que, haciendo $T_J \equiv -D^{-1}(T_I + T_S)$ y $\mathbf{c}_J \equiv D^{-1}\mathbf{b}$ resulta

$$\boxed{\mathbf{x}^{(k+1)} = T_J \mathbf{x}^{(k)} + \mathbf{c}_J} \quad (2.27)$$

MÉTODO DE JACOBI: criterio de parada

Suponiendo que la sucesión de $\mathbf{x}^{(k)}$ es convergente (más tarde veremos cuándo sucede esto), el algoritmo parará cuando la solución varíe poco, por ejemplo:

$$\frac{||\mathbf{x}^{(N)} - \mathbf{x}^{(N-1)}||}{||\mathbf{x}^{(N)}||} < \text{TOL}$$

MÉTODO DE GAUSS-SEIDEL

Partiendo del método de Jacobi...

El método de Gauss-Seidel supone una aparente mejora del método de Jacobi ya que no espera al final de cada iteración para actualizar el valor de las variables ya calculadas. En este método, una vez introducidos los valores $(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$ en la primera de las ecuaciones (2.26), y obtenido el valor de $x_1^{(1)}$, introduciremos dicho valor junto con $x_2^{(0)}, \dots, x_n^{(0)}$ en la segunda ecuación para obtener $x_2^{(1)}$. A su vez, en la tercera ecuación introducimos $x_1^{(1)}$ y $x_2^{(1)}$ junto con $x_3^{(0)}, \dots, x_n^{(0)}$ para obtener $x_3^{(1)}$ y así sucesivamente.

MÉTODO DE GAUSS-SEIDEL

jacobi

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

↓

$$\begin{cases} a_{11}x_1^{(k+1)} & = b_1 - a_{12}x_2^{(k)} - \dots - a_{1n}x_n^{(k)} \\ a_{21}x_1^{(k+1)} + a_{22}x_2^{(k+1)} & = b_2 - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)} \\ & \vdots \\ a_{n1}x_1^{(k+1)} + \dots + a_{n,n-1}x_{n-1}^{(k+1)} + a_{nn}x_n^{(k+1)} & = b_n \end{cases}$$

(2.29)

MÉTODO DE GAUSS-SEIDEL

$$\begin{cases} a_{11}x_1^{(k+1)} &= b_1 - a_{12}x_2^{(k)} - \dots - a_{1n}x_n^{(k)} \\ a_{21}x_1^{(k+1)} + a_{22}x_2^{(k+1)} &= b_2 - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)} \\ &\vdots \\ a_{n1}x_1^{(k+1)} + \dots + a_{n,n-1}x_{n-1}^{(k+1)} + a_{nn}x_n^{(k+1)} &= b_n \end{cases} \quad (2.29)$$

o, equivalentemente: $(T_I + D)\mathbf{x}^{(k+1)} = \mathbf{b} - T_S\mathbf{x}^{(k)}$, de donde haciendo $T_{G-S} \equiv -(T_I + D)^{-1}T_S$ y $\mathbf{c}_{G-S} \equiv (T_I + D)^{-1}\mathbf{b}$ resulta

$$\boxed{\mathbf{x}^{(k+1)} = T_{G-S}\mathbf{x}^{(k)} + \mathbf{c}_{G-S}} \quad (2.30)$$

MÉTODO DE GAUSS-SEIDEL

En notación matricial, requiere calcular inversas, luego es preferible un algoritmo que resuelva, en cada iteración, secuencialmente cada elemento del vector solución:

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

Donde $i=1,2,\dots,n$.

MÉTODO DE GAUSS-SEIDEL: criterio de parada

Suponiendo que la sucesión de $\mathbf{x}^{(k)}$ es convergente (más tarde veremos cuándo sucede esto), el algoritmo parará cuando la solución varíe poco, por ejemplo:

$$\frac{||\mathbf{x}^{(N)} - \mathbf{x}^{(N-1)}||}{||\mathbf{x}^{(N)}||} < \text{TOL}$$

COMPARACION – EJEMPLO

$$\begin{cases} 10x_1 - x_2 + 2x_3 = 6 \\ -x_1 + 11x_2 - x_3 + 3x_4 = 25 \\ 2x_1 - x_2 + 10x_3 - x_4 = -11 \\ 3x_2 - x_3 + 8x_4 = 15 \end{cases} \quad (2.28)$$

cuya solución única es $\mathbf{x} = (1, 2, -1, 1)^t$, a partir de la aproximación inicial $\mathbf{x}^{(0)} = (0, 0, 0, 0)^t$ y con una tolerancia $\text{TOL} = 5 \times 10^{-4}$.

COMPARACION – EJEMPLO: USANDO JACOBI...

Las ecuaciones del proceso iterativo son

$$\left\{ \begin{array}{l} x_1^{(k+1)} = \frac{1}{10}x_2^{(k)} - \frac{1}{5}x_3^{(k)} + \frac{3}{5} \\ x_2^{(k+1)} = \frac{1}{11}x_1^{(k)} + \frac{1}{11}x_3^{(k)} - \frac{3}{11}x_4^{(k)} + \frac{25}{11} \\ x_3^{(k+1)} = -\frac{1}{5}x_1^{(k)} + \frac{1}{10}x_2^{(k)} + \frac{1}{10}x_4^{(k)} - \frac{11}{10} \\ x_4^{(k+1)} = -\frac{3}{8}x_2^{(k)} + \frac{1}{8}x_3^{(k)} + \frac{15}{8} \end{array} \right.$$

y producen el resultado:

k	0	1	2	3	4	5	6	7	8	9	10
$x_1^{(k)}$	0	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
$x_2^{(k)}$	0	2.2727	1.7159	2.0530	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$x_3^{(k)}$	0	-1.1000	-0.8052	-1.0493	-0.9681	-1.0103	-0.9945	-1.0020	-0.9990	-1.0004	-0.9998
$x_4^{(k)}$	0	1.8750	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

donde hemos parado la iteración en $k = 10$ al ser

$$\frac{||\mathbf{x}^{(10)} - \mathbf{x}^{(9)}||}{||\mathbf{x}^{(10)}||} = 0.327 \times 10^{-4} < \text{TOL}$$

COMPARACION – EJEMPLO: USANDO GAUSS-SEIDEL...

En este caso, las ecuaciones del proceso iterativo son

$$\begin{cases} x_1^{(k+1)} = & \frac{1}{10}x_2^{(k)} - \frac{1}{5}x_3^{(k)} + \frac{3}{5} \\ x_2^{(k+1)} = & \frac{1}{11}x_1^{(k+1)} + \frac{1}{11}x_3^{(k)} - \frac{3}{11}x_4^{(k)} + \frac{25}{11} \\ x_3^{(k+1)} = & -\frac{1}{5}x_1^{(k+1)} + \frac{1}{10}x_2^{(k+1)} + \frac{1}{10}x_4^{(k)} - \frac{11}{10} \\ x_4^{(k+1)} = & -\frac{3}{8}x_2^{(k+1)} + \frac{1}{8}x_3^{(k+1)} + \frac{15}{8} \end{cases}$$

y producen el resultado:

k	0	1	2	3	4	5
$x_1^{(k)}$	0	0.6000	1.0302	1.0066	1.0009	1.0001
$x_2^{(k)}$	0	2.3273	2.0369	2.0036	2.0003	2.000
$x_3^{(k)}$	0	-0.9873	-1.0145	-1.0025	-1.0003	-1.000
$x_4^{(k)}$	0	0.8789	0.9843	0.9984	0.9998	1.000

donde hemos parado la iteración en $k = 5$ al ser

$$\frac{||\mathbf{x}^{(5)} - \mathbf{x}^{(4)}||}{||\mathbf{x}^{(5)}||} = 2.09 \times 10^{-4} < \text{TOL}$$

COMPARACION – EJEMPLO

USANDO JACOBI...

k	0	1	2	3	4	5	6	7	8	9	10
$x_1^{(k)}$	0	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
$x_2^{(k)}$	0	2.2727	1.7159	2.0530	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$x_3^{(k)}$	0	-1.1000	-0.8052	-1.0493	-0.9681	-1.0103	-0.9945	-1.0020	-0.9990	-1.0004	-0.9998
$x_4^{(k)}$	0	1.8750	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

USANDO GAUSS-SEIDEL...

k	0	1	2	3	4	5
$x_1^{(k)}$	0	0.6000	1.0302	1.0066	1.0009	1.0001
$x_2^{(k)}$	0	2.3273	2.0369	2.0036	2.0003	2.000
$x_3^{(k)}$	0	-0.9873	-1.0145	-1.0025	-1.0003	-1.000
$x_4^{(k)}$	0	0.8789	0.9843	0.9984	0.9998	1.000

ES SIEMPRE GS MÁS RÁPIDO QUE JACOBI??

COMPARACION – EJEMPLO

USANDO JACOBI...

k	0	1	2	3	4	5	6	7	8	9	10
$x_1^{(k)}$	0	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
$x_2^{(k)}$	0	2.2727	1.7159	2.0530	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$x_3^{(k)}$	0	-1.1000	-0.8052	-1.0493	-0.9681	-1.0103	-0.9945	-1.0020	-0.9990	-1.0004	-0.9998
$x_4^{(k)}$	0	1.8750	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

USANDO GAUSS-SEIDEL...

k	0	1	2	3	4	5
$x_1^{(k)}$	0	0.6000	1.0302	1.0066	1.0009	1.0001
$x_2^{(k)}$	0	2.3273	2.0369	2.0036	2.0003	2.000
$x_3^{(k)}$	0	-0.9873	-1.0145	-1.0025	-1.0003	-1.000
$x_4^{(k)}$	0	0.8789	0.9843	0.9984	0.9998	1.000

ES SIEMPRE GS MÁS RÁPIDO QUE JACOBI??

¡ NO SIEMPRE ! ES NECESARIO REALIZAR UN ESTUDIO DE CONVERGENCIA PREVIO...

CONVERGENCIA

Los métodos anteriores pueden converger/divergir independientemente (cada uno puede converger o divergir para el mismo problema).

CONDICION SUFICIENTE DE CONVERGENCIA

Teorema 2.1.2 Si A es estrictamente diagonal dominante, entonces con cualquier elección de $\mathbf{x}^{(0)}$ tanto el método de Jacobi como el de Gauss-Seidel generan sucesiones $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ que convergen a la solución única del sistema $A\mathbf{x} = \mathbf{b}$. □

CONVERGENCIA: radio espectral

¿Cómo crece el error de truncamiento absoluto del esquema?

Definición 2.1.2 El *radio espectral* $\rho(A)$ de una matriz A de autovalores $\lambda_1, \dots, \lambda_p$ se define como

$$\rho(A) = \max_{1 \leq i \leq p} |\lambda_i|$$



Pues bien, aunque no entraremos en detalle en este punto, puede comprobarse que una estimación para el error absoluto en el paso k del proceso iterativo $\mathbf{x}^{(k+1)} = T\mathbf{x}^{(k)} + \mathbf{c}$ viene dada por

$$\|\mathbf{x}^{(k)} - \mathbf{x}\| \simeq \rho(T)^k \|\mathbf{x}^{(0)} - \mathbf{x}\| \quad (2.31)$$

CONVERGENCIA

es decir, la propagación del error inicial depende del radio espectral. Si queremos que el proceso converja, éste tendrá que ser $\rho(T) < 1$. El método —Jacobi o Gauss-Seidel— que más rápidamente convergerá en cada caso particular será aquel cuya matriz asociada tenga un menor radio espectral.

[Convergence and the Spectral Radius] The sequence

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$$

converges to the unique solution of $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ for any $\mathbf{x}^{(0)}$ in \mathbb{R}^n if and only if $\rho(T) < 1$.

CONVERGENCIA

$[l_\infty$ Norm of a Matrix]

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|.$$

Ojo

Calcular el radio espectral puede ser costoso, pero cualquier norma natural de una matriz es cota superior de su radio espectral → basta con calcular una norma natural (la más sencilla de calcular es l infinito) y verificar que es menor que 1 para que el método sea convergente.

$[l_2$ Matrix Norm Characterization] If A is an $n \times n$ matrix, then

- (i) $\|A\|_2 = [\rho(A^t A)]^{1/2};$
- (ii) $\rho(A) \leq \|A\|$ for any natural norm.