Apuntes de teórico - Métodos Numéricos

Instituto de Matemáticas y Estadística "Rafael Laguardia"
Facultad de Ingeniería, Universidad de la República
2016 - Montevideo, Uruguay

Nota importante: El presente material forma parte de una versión **en proceso de revisión** de un texto teórico para la asignatura "Métodos Numéricos". Por lo tanto, para preparar los exámenes se debe utilizar el material de las clases teóricas y la **bibliografía recomendada**.

Índice general

Capítulo 2

Sistemas de Ecuaciones Lineales

2.1. Introducción

Un sistema de m ecuaciones lineales y n incógnitas consta de un conjunto de relaciones algebraicas de la forma:

$$\sum_{j=1}^{n} a_{ij}\, x_j = b_i \qquad x_j,\, a_{ij},\, b_i \in \mathbb{R} \quad \forall\, i = 1 \ldots m$$

el cual puede ser representado en notación matricial como $A\mathbf{x} = \mathbf{b}$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \in \mathcal{M}_{m\times n}(\mathbb{R}) \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

Este sistema tiene solución única si y solo si $m = n$ y $|A| \neq 0$, entonces decimos que es compatible determinado (CD). En este capítulo trabajaremos con sistemas de esta clase.

En el caso de $m > n$ existen más ecuaciones que variables, por lo tanto, si el vector $\mathbf{b}$ no pertenece al espacio generado por las columnas de A, no existe una solución que verifique todas las ecuaciones, decimos que el sistema es incompatible. Más adelante veremos técnicas para resolver este tipo de problemas.

Métodos directos e indirectos Los métodos de resolución de sistemas lineales pueden ser clasificados en las siguientes dos categorías:

- Directos: obtenemos solución luego de un número finito de iteraciones. Si tenemos precisión infinita la solución es exacta.
- Indirectos: obtenemos una aproximación de la solución $(\bar{x}_k)$, luego de k iteraciones. La solución se mejora sucesivamente con cada paso.

2.2. Métodos directos

Los métodos directos generan soluciones aproximadas luego de realizar una cantidad finita de pasos. Uno de los ejemplos más habituales es la escalerización gaussiana.

2.2.1. Solución de sistemas triangulares

Comenzaremos por abordar un caso particular de sistemas lineales. Sea el siguiente sistema 3×3 triangular inferior no singular:

$$\begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

Las matrices con bloques con ceros serán representadas esquemáticamente de la siguiente forma:

$$A\,\mathbf{x} = \mathbf{b} \qquad A = \begin{pmatrix} \diagdown & 0 \\ & \diagdown \end{pmatrix}$$

Dado que el sistema es no singular, las entradas de la diagonal a_{ii}, $i = 1, 2, 3$ son distintas a cero, por lo tanto lo podemos resolver de la siguiente forma:

$$x_1 = \frac{b_1}{a_{11}}, \tag{2.1}$$

$$x_2 = \frac{b_2 - a_{21}x_1}{a_{22}}, \tag{2.2}$$

$$x_3 = \frac{b_3 - a_{31}x_1 - a_{32}x_2}{a_{33}}. \tag{2.3}$$

Este algoritmo puede ser extendido para sistemas triangulares inferiores de orden n, de la forma $A\,\mathbf{x} = \mathbf{b}$. Este método tiene el nombre de sustitución hacia adelante:

$$\begin{aligned} x_1 &= \frac{b_1}{a_{11}} \\ x_i &= \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}\ x_j \right), \qquad i = 2, \ldots, n \end{aligned}$$

El pseudo-código del método es el siguiente:

- paso 1: $x_1 = \frac{b_1}{a_{11}}$
- paso i: $x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}\ x_j \right), \qquad i = 1, \ldots, i-1$

recordando la definición de *Flops* del capítulo 1, podemos calcular el costo computacional del método. Se realizan $n\,(n+1)/2$ multiplicaciones-divisiones mientras que el número de sumas-restas es $n\,(n-1)/2$, por lo tanto, el costo es n^2 *flops*.

En el caso que el sistema lineal sea triangular superior,

$$A\,\mathbf{x} = \mathbf{b} \qquad A = \begin{pmatrix} \diagdown & \\ 0 & \diagdown \end{pmatrix}$$

podemos utilizar un método análogo llamado sustitución hacia atrás (BS[1]):

$$\begin{aligned} x_n &= \frac{b_n}{a_{nn}} \\ x_i &= \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^{n} a_{ij}\, x_j \right), \qquad i = n-1, \ldots, 1 \end{aligned}$$

Pseudo-código:

- paso n: $x_n = b_n / a_{nn}$
- paso i, $i = n-1, \ldots, 1$: $x_i = \frac{b_i - \sum_{k=i+1}^{n} a_{ik}\, x_k}{a_{ii}}$

2.2.2. Escalerización Gaussiana (EG)

El Método de Escalerización Gaussiana (MEG) es un método directo para resolución de Sistemas de Ecuaciones Lineales.

Permite llevar un sistema general no singular a uno equivalente triangular superior, por medio de la aplicación de operaciones elementales (intercambio y combinación lineal de filas).

Algoritmo: Paso k: si $a_{kk}^{(k)} \neq 0$ será llamado *pivot*

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \ldots & a_{1k}^{(1)} & \ldots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \ldots & a_{2k}^{(2)} & \ldots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & a_{kk}^{(k)} & \ldots & a_{kn}^{(k)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & a_{nk}^{(k)} & \ldots & a_{nn}^{(k)} \end{bmatrix}$$

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

Las cantidades de flops de *EG* por cada bloque del código son las siguientes:

[1] BS por Backwards Substitution

Algoritmo 1 Pseudo-código: EG no eficiente

Sea A una matriz con entradas $A_{i,j} = a(i,j)$

```
for k = 1 → n − 1 do
  for i = k + 1 → n do
    l(i, k) ← a(i, k)/a(k, k)
    a(i, k) ← 0
    for j = k + 1 → n do
      a(i, j) ← a(i, j) − l(i, k) ∗ a(k, j)
    end for
  end for
end for
```

- loop j: $2(n-k)$
- loop i: $(2(n-k)+1)\ (n-k) = 2(n-k)^2 + (n-k)$
- loop k: $\sum_{k=1}^{n-1} 2(n-k)^2 + (n-k)$

Tomando el loop k y desarrollando obtenemos:

$$\begin{aligned} flops(EG) &= \sum_{k=1}^{n-1} 2n^2 - 4n\,k + 2k^2 + n - k \\ \ldots &= 2\sum_{k=1}^{n-1} k^2 - (4n+1)\sum_{k=1}^{n-1} k + 2n^2 + n \\ \ldots &= 2\,\frac{(n-1)n(2n-1)}{6} - (4n+1)\frac{(n-1)n}{2} + 2n^2 + n \end{aligned}$$

Por lo tanto, la cantidad de *flops* necesarios para resolver $A\,\mathbf{x} = \mathbf{b}$ usando EG y sustitución hacia adelante es la siguiente:

$$EG + BS = O(2/3\,n^3) + O(n^2) \approx O(2/3\,n^3)$$

Observación 2.2.1. Se recuerda que utilizando el MEG, sin pivoteo, el valor del determinante se mantiene inalterado, y una vez escalerizada la matriz, el determinante resulta en el producto de los valores de la diagonal, por tanto, acotamos la cantidad de *flops* para resolver $|A|$:

$$EG + \prod_{i=1}^{n} a_{ii}^{(i)} = O(2/3\,n^3) + O(n) \approx O(2/3\,n^3)$$

Se sugiere comparar con el costo computacional del cálculo utilizando la fórmula general o descomposición por filas (deberá observar que la cantidad de multiplicaciones son $n!(n-1)$ y luego se deben sumar $n!$ términos).

2.2.3. Descomposición LU (sin pivoteo)

Un segundo método directo de resolución de SL es mediante la llamada Descomposición LU. Esta descomposición se basa en el método de escalerización gaussiana pero permite ahorros computacionales en algunos casos como desarrollaremos luego de presentar el método.

Dada una matriz $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, tal que $|A| \neq 0$, existen dos matrices L y U tal que $A = LU$.

$$\begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ l_{21} & 1 & 0 & \vdots & \vdots \\ l_{31} & l_{32} & 1 & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & 0 \\ l_{n1} & \dots & \dots & l_{nn-1} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ \vdots & 0 & u_{33} & \dots & u_{3n} \\ \vdots & \dots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & u_{nn} \end{bmatrix} = A$$

por lo tanto:

$$a_{ij} = \sum_{k=1}^{r} l_{ik}\, u_{kj} \qquad r = \text{mín}\{i, j\}$$

tomando la convención de que $l_{ii} = 1$.

Obsérvese que los l_{ij} son los mismos coeficientes determinados por el algoritmo y los u_{ij} son las entradas de la matriz escalerizada, es decir que escribimos $u_{ij} = a_{ij}^{(i)}$.

Demostraremos la anterior relación para el cálculo de las entradas a_{ij} de A a partir de las entradas de L y U realizando los pasos de EG:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik}\, a_{kj}^{(k)} \qquad k = 1, \dots, p; \quad p = i - 1 \text{ si } i \leq j; \quad p = j \ \textit{si}\ i > j$$

Si sumamos la ecuación anterior para los distintos valores de $k = 1, \dots, p$, y reordenando:

$$\sum_{k=1}^{p} a_{ij}^{(k+1)} - \sum_{k=1}^{p} a_{ij}^{(k)} = -\sum_{k=1}^{p} l_{ik}\, a_{kj}^{(k)}$$

Ahora, los términos de la izquierda se cancelan entre las sumatorias:

$$a_{ij}^{(p+1)} - a_{ij} = -\sum_{k=1}^{p} l_{ik}\, a_{kj}^{(k)}$$

y como

$$a_{ij}^{(p+1)} = \begin{cases} a_{ij}^{(i)} & i \leq j \\ 0 & i > j \end{cases}$$

y además teníamos que $u_{ij} = a_{ij}^{(i)}$, se concluye reescribiendo las entradas que:

$$a_{ij} = \sum_{k=1}^{r} l_{ik}\, u_{kj} \qquad r = \text{mín}\{i, j\}$$

con lo que se termina la prueba.

El número de *flops* para realizar la descomposición L.U. es igual a las operaciones requeridas para aplicar la escalerización gaussiana. De esta forma, enumeremos las operaciones necesarias para resolver un sistema lineal aplicando L.U. :

$$\begin{aligned} A = LU \quad (EG) \quad &= O(2/3n^3) \\ L\,\mathbf{y} = \mathbf{b} \quad (BS) \quad &= O(n^2) \\ U\,\mathbf{x} = \mathbf{y} \quad (FS) \quad &= O(n^2) \\ \textbf{Total} \quad \textit{flops} \quad &\approx O(2/3n^3) \end{aligned} \tag{2.4}$$

Una característica importante de este método está en la resolución de múltiples sistemas lineales, es decir, varios sistemas con la misma matriz A. Por ejemplo, calculemos la cantidad de operaciones necesarias para resolver m sistemas lineales utilizando L.U.:

$$\begin{array}{c} A\,\mathbf{x}_i = \mathbf{b}_i \\ i = 1, \ldots, m \end{array} \Rightarrow \left\{ \begin{array}{l} A = L\,U \\ for\ i = 1, \ldots, m \\ \quad A\,\mathbf{y}_i = \mathbf{b}_i \\ \quad U\,\mathbf{x}_i = \mathbf{y}_i \\ end \end{array} \right. \Rightarrow flops = O(2/3\,n^3 + 2mn^2)$$

Número de *flops* para hallar A^{-1} con L.U.

$$A\,X = \mathrm{I} \Rightarrow \mathbf{b}_j = \mathrm{I}_j = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad j = 1, \ldots, n \quad \Rightarrow flops = O(2/3n^3 + 2n^3) = O(8/3n^3)$$

Observación 2.2.2. Dado un sistema lineal $A\,\mathbf{x} = \mathbf{b}$, no es económico hallar $\mathbf{x}$ calculando $A^{-1}\,\mathbf{b}$.

2.2.4. EG con pivotes

¿Qué pasa con el método de escalerización si en algún paso i, se llega a $a_{kk}^{(k)} = 0$? Es imposible realizar el algoritmo ya que este valor debería ser el denominador en los cocientes que modifican las entradas a partir de allí, en particular, se requiere para el cálculo de los $l(i,k)$.

Entonces, lo que podemos hacer es intercambiar filas (pivotear), ya que supusimos que A es invertible, entonces existe $a_{pk}^{(k)} \neq 0$, $k+1 \leq p \leq n$.

Lo anterior es cierto con aritmética exacta. Si la aritmética es PF entonces conviene pivotear si algún $a_{kk}^{(k)} << 1$.

Ejemplo 2.2.1 (Error sin pivoteo)**.** Arit. PF 5 dígitos con redondeo: $\pm a_1\,,\ a_2\,a_3\,a_4\,a_5\ \times 10^e$

Consideremos el siguiente sistema de ecuaciones lineales:

$$(S)\ \left[\begin{array}{ccc|c} 10 & -7 & 0 & 7 \\ -3 & 2,099 & 6 & 3,901 \\ 5 & -1 & 5 & 6 \end{array}\right]$$

y lo resolvemos aplicando EG sin utilizar pivoteo:

$$A^{(1)}\ \left[\begin{array}{ccc|c} 1,0000 \times 10^1 & -7,0000 \times 10^0 & 0,0000 \times 10^0 & 7,0000 \times 10^0 \\ -3,0000 \times 10^0 & 2,099 \times 10^0 & 6,0000 \times 10^0 & 3,9010 \times 10^0 \\ 5,0000 \times 10^0 & -1,0000 \times 10^0 & 5,0000 \times 10^0 & 6,0000 \times 10^0 \end{array}\right]$$

En el segundo paso obtenemos una entrada a_{22} con un valor muy bajo comparado con el resto de los valores de la matriz.

$$A^{(2)}\ \left[\begin{array}{ccc|c} 1,0000 \times 10^1 & -7,0000 \times 10^0 & 0,0000 \times 10^0 & 7,0000 \times 10^0 \\ 0 & -1,0000 \times 10^{-3} & 6,0000 \times 10^0 & 6,0010 \times 10^0 \\ 0 & 2,5000 \times 10^0 & 5,0000 \times 10^0 & 2,5000 \times 10^0 \end{array}\right]$$

en el tercer paso obtenemos un error en la componente b_3

$$A^{(3)} \left[\begin{array}{ccc|c} 1,0000 \times 10^1 & -7,0000 \times 10^0 & 0,0000 \times 10^0 & 7,0000 \times 10^0 \\ 0 & -1,0000 \times 10^{-3} & 6,0000 \times 10^0 & 6,0010 \times 10^0 \\ 0 & 0 & 1,5005 \times 10^4 & 1,5004 \times 10^4 \end{array}\right]$$

obtenemos la solución:

sol. numérica	sol. exacta	
$\hat{x}_1 = -2,8000 \times 10^{-1}$	$x_1 = 0$	$\times$ mal
$\hat{x}_2 = -1,4000 \times 10^0$	$x_2 = -1$	$\times$ mal
$\hat{x}_3 = 9,9993 \times 10^{-1}$	$x_3 = 1$	$\cong$ Ok

Observación 2.2.3. Si $1,5004 \times 10^4$ hubiera sido $1,5005 \times 10^4$ (valor exacto) entonces la solución numérica sería igual a la exacta, por lo tanto se concluye que el error está provocado por no haber pivoteado cuando el valor a_{22} era pequeño.

En general pivoteamos si $a_{ii}^{(i)}$ vale cero o cualquier valor que sea "mucho menor" que el resto de las entradas de $A^{(i)}_{i\ldots n,i\ldots n}$. En el ejemplo 2.2.1 un valor "mucho menor" que los otros corresponde a menor que un $1\,\%$ por ejemplo.

Si pivoteamos entonces EG es numericamente estable.

Estrategias de pivoteo Presentaremos dos formas de elegir la entrada de la matriz que utilizaremos como pivot.

Pivoteo parcial Se compara el valor del pivot actual con todas las entradas siguientes de esa columna (por debajo de k), eligiendo como pivot al de mayor magnitud.

$$\arg \max_{r \in \{k,\ldots,n\}} |a_{rk}^{(k)}| = p \Rightarrow \text{ pivot: } a_{pk}$$

Luego de elegir la fila p, debemos intercambiar la fila k por la p y continuar con EG. La cantidad de comparaciones totales que se realizan es del orden de n^2.

Pivoteo completo Se compara el valor del pivot actual con cualquier otra entrada por debajo de esa fila y a la derecha de esa columna, eligiendo como pivot al de mayor magnitud.

$$\arg \max_{r \in \{k,\ldots,n\}, s \in \{k,\ldots,n\}} |a_{rs}^{(k)}| = p, q \Rightarrow \text{ pivot: } a_{pq}$$

En este caso, además de intercambiar la fila k por la fila p, también debemos intercambiar la columna k por la q. En este caso la cantidad de comparaciones totales es del orden de n^3.

2.2.5. Descomposición LU con intercambio de filas

Teorema 2.2.1. *Sea $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ no singular. Se puede descomponer A como $PA = LU$, con P una matriz de permutación.*

Observación 2.2.4. PA intercambia las filas de A, mientras que AP intercambia sus columnas. Compruébelo tomando $A \in \mathcal{M}_{2\times 2}(\mathbb{R})$ genérica y $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Haciendo EG con pivoteo formamos $P \in \mathcal{M}_{n\times n}(\mathbb{R})$ matriz de permutaciones de filas

$$P\,A = \tilde{A} = L\,U \Rightarrow PA = LU$$

Solución de un SL con descomposición PLU

Hallar la descomposición PLU. Luego $PA\mathbf{x} = P\mathbf{b}$. Llamamos $P\mathbf{b} = \mathbf{b}'$.

Con $\mathbf{b}'$ planteamos el sistema $LU\mathbf{x} = \mathbf{b}'$, el cual se resuelve mediante sustitución hacia atrás y hacia adelante, primero resolviendo un sistema auxiliar $L\mathbf{y} = \mathbf{b}'$, donde $\mathbf{y} = U\mathbf{x}$, y luego obtenemos $\mathbf{x}$ resolviendo $U\mathbf{x} = \mathbf{y}$.

$$L\,U\,\mathbf{x} = \mathbf{b}' \;\Rightarrow \left\{ \begin{array}{l} P,\, L,\, U \\ \mathbf{b}' = P\,\mathbf{b} \\ L\,\mathbf{y} = \mathbf{b}' \\ U\,\mathbf{x} = \mathbf{y} \end{array} \right. \Rightarrow \mathbf{x}$$

2.2.6. Almacenamiento económico

Definición 2.2.1 (Matrices Esparsas)**.** $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ es esparsa si la mayoría de sus entradas son nulas.

Las matrices esparsas aparecen en muchísimos problemas tanto de ingeniería como de otras ramas y su manipulación son un vivo tema de investigación.

Sea A una matriz esparsa con elementos no nulos, una técnica de almacenamiento podría ser guardar el valor de la entrada junto con sus índices de fila y columna:

$$\left.\begin{array}{ccc} (i_1 & j_1 & a_{i_1,j_1}) \\ (i_2 & j_2 & a_{i_2,j_2}) \\ & \vdots & \\ (i_m & j_m & a_{i_m,j_m}) \end{array}\right\} \text{ si una entrada } a_{ij} \text{ no está en la lista, entonces es nula.}$$

¿Cómo puede ser esto eficiente si por cada entrada debo almacenar tres valores en lugar de uno? Veamos, si se guardan las ternas se requieren $3m$ NPF (números en punto flotante). Por otra parte, si se guarda llena, es decir con todos los ceros que corresponden se necesitarán n^2 NPF. Por tanto, si la matriz es esparsa, $m << n$, entonces entonces $3m << n^2$, lográndose una interesante reducción de espacio de almacenamiento.

Observación 2.2.5. En general A^{-1} no es esparsa aunque A lo sea.

2.2.7. Estructura de banda

Una matriz $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ es una *matriz banda* si los valores no nulos de la matriz se presentan solamente en una banda entorno a la diagonal.

Formalmente, la matriz tiene una estructura de banda si:

$$a_{i,j} = 0 \text{ si } \begin{cases} i - j > k_1 \\ j - i > k_2 \end{cases} \quad \text{con } k_1, k_2 \geq 0.$$

Definimos entonces el *ancho de banda* de una matriz como $k_1 + k_2 + 1$.

Ejemplo 2.2.2. Algunos casos particulares:

- Una matriz diagonal es una matriz banda con $k_1 = k_2 = 0$ y su ancho de banda es 1.
- Una matriz banda es tridiagonal cuando $k_1 = k_2 = 1$.
- Una matriz es triangular superior si $k_1 = n - 1$ y $k_2 = 0$.

Para este tipo de matrices, muchos algoritmos de resolución pueden optimizarse reduciendo considerablemente tanto la complejidad como el costo computacional y así el tiempo de ejecución.

Un ejemplo de esto es el Algoritmo de Thomas para matrices tridiagonales, en el cual la descomposición LU pasa a tener un orden lineal.

2.2.8. Otros métodos directos

Antes de pasar a los métodos indirectos de resolución, mencionaremos algunos otros métodos directos para que el lector amplíe su visión sobre los mismos e indague sus ventajas y desventajas.

Asumiremos que los sistemas son compatibles determinados.

Cálculo de matriz inversa

Dado el sistema $A\mathbf{x} = \mathbf{b}$, es posible determinar $\mathbf{x}$ operando algebraicamente en la ecuación: $A^{-1}A\mathbf{x} = A^{-1}\mathbf{b} \Rightarrow \mathbf{x} = A^{-1}\mathbf{b}$.

Por tanto, hallar la matriz inversa de A es otro mecanismo para resolver un sistema de ecuaciones.

Notamos sin embargo que el método para el cálculo de la matriz inversa en el que se construye la matriz ampliada $[A|I]$ y se reduce hasta obtener $[I|A^{-1}]$ involucra doblemente escalerización Gaussiana (ya que se debe escalerizar hacia "abajo" y luego hacia "arriba"), para luego realizar una multiplicación matriz por vector. Se sugiere comparar este costo con respecto a $EG + BS$.

Regla de Cramer

Dado el sistema $A\mathbf{x} = \mathbf{b}$, la Regla de Cramer devuelve cada entrada del vector solución solamente como un cociente de determinantes que dependen de la matriz A y del vector $\mathbf{b}$. En efecto:

$$x_i = \frac{\det(A_i)}{\det(A)}$$

donde A_i es la matriz resultante de reemplazar la columna i-ésima de la matriz A por el vector $\mathbf{b}$.

Ejemplo 2.2.3. Sea $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$, y $\mathbf{b} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$.

Entonces $\det(A) = 1$, $\det(A_1) = \begin{vmatrix} 3 & 1 \\ 2 & 1 \end{vmatrix} = 1$, $\det(A_2) = \begin{vmatrix} 2 & 3 \\ 1 & 2 \end{vmatrix} = 1$.

Resultando en $\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Se deja como ejercicio calcular el costo computacional de este método para su comparación con los anteriores.

2.3. Estabilidad de sistemas lineales

La resolución de sistemas lineales mediante métodos numéricos involucra tanto errores de redondeo por el pasaje de números a una representación finita, como en el caso de la resolución de estos sistemas utilizando métodos iterativos, el truncamiento de una sucesión que converge a la solución real del sistema. Es por esta razón que es necesario definir una noción de cercanía en los espacios con los que estaremos trabajando. Es así que comenzamos la sección introduciendo algunos conceptos y resultados referentes a normas vectoriales y matriciales.

2.3.1. Norma de vectores

Trabajaremos en el espacio vectorial $\mathbb{R}^n$ con las operaciones suma y producto interno habituales entre vectores. Una norma es una función $\|\cdot\|$ que verifica las siguientes propiedades:

$$(\mathbb{R}^n, \mathbb{R}, +, \cdot) \text{ } e.v., \qquad \|\cdot\| : \mathbb{R}^n \longrightarrow \mathbb{R}$$

1. $\|\mathbf{u}\| \geq 0 \quad \forall \mathbf{u} \in \mathbb{R}^n \quad$ y $\quad \|\mathbf{u}\| = 0 \quad \Leftrightarrow \quad \mathbf{u} = \vec{0}$
2. $\|\lambda\, \mathbf{u}\| = |\lambda|\, \|\mathbf{u}\| \qquad \forall \lambda \in \mathbb{R}, \quad \forall \mathbf{u} \in \mathbb{R}^n$
3. $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\| \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^n$

Norma-p Dado un vector de $\mathbb{R}^n$, $\mathbf{x} = (x_1, \ldots, x_n)^t$. La Norma-$p$ del vector será:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_1|^p \right)^{1/p} \qquad 1 \leq p < \infty$$

Para distintos valores de p se obtienen distintas variantes de la norma:

$$\begin{aligned} \|\mathbf{x}\|_1 = & \quad \textstyle\sum_{i=1}^{n} |x_i| & p = 1 \\ \|\mathbf{x}\|_2 = & \quad \sqrt{\textstyle\sum_{i=1}^{n} x_i^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}} & p = 2 \quad \text{(Euclidiana)} \end{aligned}$$

Norma Infinito En el caso de que $p = \infty$ podemos tomar el límite a partir de la definición:

$$\begin{aligned} \|\mathbf{x}\|_\infty &= \lim_{p \to \infty} \|\mathbf{x}\|_p \\ \|\mathbf{x}\|_\infty &= \max_i |x_i| \underbrace{\lim_{p \to \infty} \left(\sum_{j=1}^{n} \alpha_j^p \right)^{1/p}}_{1} \qquad \alpha_j = \frac{x_j}{\max_i |x_i|} \in [0,1] \; j = 1, \ldots, n \\ \|\mathbf{x}\|_\infty &= \max_i |x_i| \end{aligned}$$

2.3.2. Norma de matrices

La norma de matrices, en este apuntes será una función definida en el espacio vectorial de las matrices con entradas reales y las operaciones habituales de suma y producto de matrices.

$$(\mathcal{M}_{n\times n}(\mathbb{R}), \mathbb{R}, +, *) \; e.v., \qquad \|\cdot\| : \mathcal{M}_{n\times n}(\mathbb{R}) \longrightarrow \mathbb{R}$$

1. $\|A\| \geq 0 \quad \forall A \in \mathcal{M}_{n\times n}(\mathbb{R}) \quad$ y $\quad \|A\| = 0 \quad \Leftrightarrow \quad A = \vec{0}$
2. $\|\lambda\, A\| = |\lambda|\, \|A\| \qquad \forall \lambda \in \mathbb{R}, \quad \forall A \in \mathcal{M}_{n\times n}(\mathbb{R})$
3. $\|A + B\| \leq \|A\| + \|B\| \quad \forall A, B \in \mathcal{M}_{n\times n}(\mathbb{R})$

Diremos además que una norma de matrices es *submultiplicativa* si $\|AB\| \leq \|A\|\|B\| \quad \forall A, B \in \mathcal{M}_{n\times n}(\mathbb{R})$.

Definición 2.3.1 (Norma compatible)**.** Una norma matricial $\|\cdot\|_M$ es compatible con una vectorial $\|\cdot\|_v$ si

$$\|A\mathbf{x}\|_v \leq \|A\|_M \|\mathbf{x}\|_v \quad \forall A \in \mathcal{M}_{n\times n}(\mathbb{R}), \forall \mathbf{x} \in \mathbb{R}^n$$

A partir de aquí no haremos referencia explícita a qué norma estamos considerando lo que se deducirá del argumento que tome la norma según corresponda.

Definición 2.3.2 (Norma inducida, o norma operador)**.**

$$\|A\| = \max_{\mathbf{x} \neq \vec{0}} \frac{\|A\,\mathbf{x}\|}{\|\mathbf{x}\|}$$

Esta norma es inducida a partir de alguna norma de vectores.

Ejercicio 2.3.1. Verificar que la norma inducida cumple las propiedades de norma.

Podemos ver que esta norma cumple las siguientes propiedades:

Proposición 2.3.1.

$$\|A\| = \max_{\|\mathbf{x}\|=1} \|A\,\mathbf{x}\|$$

Demostración.

$$\|A\| = \max_{\mathbf{x}\neq\vec{0}} \frac{\|A\,\mathbf{x}\|}{\|\mathbf{x}\|} = \max_{\mathbf{x}\neq\vec{0}} \|A\,\frac{\mathbf{x}}{\|\mathbf{x}\|}\| = \max_{\|\mathbf{y}\|=1} \|A\,\mathbf{y}\|$$

□

Comentamos que la propiedad anterior ilustra que el valor de la norma de la matriz es igual al valor de la norma del vector más deformado por la transformación $A\mathbf{x}$.

Proposición 2.3.2. *La norma operador es compatible, es decir,*

$$\|A\,\mathbf{x}\| \leq \|A\|\,\|\mathbf{x}\| \quad \forall A \in \mathcal{M}_{n\times n}(\mathbb{R}), \forall \mathbf{x} \in \mathbb{R}^n$$

Demostración.

$$\|A\,\mathbf{x}\| = \frac{\|A\,\mathbf{x}\|}{\|\mathbf{x}\|}\|\mathbf{x}\| \leq \left(\max_{\mathbf{z}\neq\vec{0}} \frac{\|A\,\mathbf{z}\|}{\|\mathbf{z}\|}\right)\|\mathbf{x}\| = \|A\|\,\|\mathbf{x}\|$$

□

Ejercicio 2.3.2. Demostrar que dadas dos matrices $n \times n$ y la norma inducida dada en 2.3.2, se cumple:

$$\|A\,B\| \leq \|A\|\,\|B\|$$

Ejemplos de normas Las siguientes son algunos ejemplos normas usualmente utilizadas:

- Norma 1:

$$\|A\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$$

- Norma 2:

$$\|A\|_2 = \max_{\|\mathbf{x}\|=1} \sqrt{\mathbf{x}^t A^t A \mathbf{x}} = \sigma_1(A)$$

- Norma Infinito:

$$\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$$

Aquí notamos $\sigma_1(A) = \sqrt{\lambda_1}$, con λ_1 el mayor valor propio de A^tA (que existe y es positivo).

Veamos que $\|A\|_\infty = \text{máx}_i \sum_{j=1}^n |a_{ij}|$.

$$\|A\mathbf{x}\|_\infty = \max_i |\sum_{j=1}^n a_{ij}x_j| \leq \max_i \sum_{j=1}^n |a_{ij}x_j| \leq \max_i \sum_{j=1}^n |a_{ij}|$$

Donde hemos usado en la última inecuación que $\|\mathbf{x}\|_\infty = 1 \Rightarrow \text{máx}_i |x_i| = 1$.

Por tanto, sabemos que $\|A\mathbf{x}\|_\infty \leq \text{máx}_i \sum_{j=1}^n |a_{ij}|$ con $\|\mathbf{x}\|_\infty = 1$. Ahora bien, debemos ver que se cumple la igualdad, es decir, probar que la cota se alcanza.

Esto es, $\exists \mathbf{y} \in \mathbb{R}^n$ tal que $\|A\mathbf{y}\|_\infty = \text{máx}_i \sum_{j=1}^n |a_{ij}|$ con $\|\mathbf{y}\|_\infty = 1$.

Sea i_0 el índice de la fila en el que se da el $\text{máx}_i \sum_{j=1}^n |a_{ij}| = \sum_{j=1}^n |a_{i_0 j}|$, entonces tomando $\mathbf{y}^t = (sg(a_{i_0 1}), sg(a_{i_0 2}), \ldots, sg(a_{i_0 n}))^t$. En tal caso, $\|\mathbf{y}\|_\infty = 1$ y además $\|A\mathbf{y}\|_\infty = \text{máx}_i \sum_{j=1}^n |a_{ij}|$. La cota se alcanza y por lo tanto es el máximo.

Es así que se concluye que $\|A\|_\infty = \text{máx}_i \sum_{j=1}^n |a_{ij}|$.

Ejercicio 2.3.3. Probar que $\|A\|_2 = \sigma_1(A) = \sqrt{\lambda_1}$, con λ_1 el mayor valor propio de A^tA. Puede ser de utilidad para completar la demostración formalizar:

1. $\|A\mathbf{x}\|_2^2 = (A\mathbf{x})^t(A\mathbf{x}) = \mathbf{x}^tA^tA\mathbf{x}$.
2. A^tA es simétrica y definida positiva ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$).
3. Si v_i son los vectores propios tal que $\|v_i\| = 1$, $v_i \perp v_j \quad \forall i \neq j$, $\mathbf{x} = \sum_{i=1}^n \alpha_i v_i$, con $\|\mathbf{x}\|_2 = 1 \Rightarrow \sum_{i=1}^n \alpha_i^2 = 1$.

Definición 2.3.3 (Radio espectral)**.** Sea A una matriz cuadrada $A \in \mathcal{M}_{n\times n}(\mathbb{R})$. Su radio espectral ρ está definido como el máximo de los valores absolutos de sus valores propios.

$$\rho(A) = \max_i |\lambda_i| \qquad A\mathbf{v}_i = \lambda_i \mathbf{v}_i \quad \forall i = 1, \ldots, n$$

Proposición 2.3.3. *La norma operador está acotada inferiormente por su radio espectral:*

$$\rho(A) \leq \|A\|$$

Demostración. Sea λ valor propio de A tal que $|\lambda| = \rho(A)$, y $\mathbf{v}$ correspondiente vector propio de norma 1. Entonces:

$$\|A\| = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\| \geq \|A\mathbf{v}\| = \|\lambda\mathbf{v}\| = |\lambda| = \rho(A). \tag{2.5}$$

□

Teorema 2.3.4 (Teorema del Radio Espectral)**.** *Sea A una matriz cuadrada $A \in \mathcal{M}_{n\times n}(\mathbb{R})$. Para todo $\varepsilon > 0$ existe alguna norma consistente $\|\cdot\|$ tal que*

$$\|A\| < \rho(A) + \varepsilon$$

Observación 2.3.1. El teorema 2.3.4 es equivalente a decir que el radio espectral es el ínfimo de todas las normas consistentes de una matriz.

2.3.3. Número de condición

Definición 2.3.4 (Número de condición de una matriz)**.** Dada una matriz cuadrada $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ decimos que su número de condición k está dado por la siguiente expresión

$$k(A) = \|A\| \left\|A^{-1}\right\|$$

Observación 2.3.2. $k(A) \geq 1$: $k(A) = \|A\|\|A^{-1}\| \geq \|AA^{-1}\| = \|Id\| = 1$

2.3.4. Análisis de perturbaciones

Debido a los errores de representación, en vez de resolver un sistema $A\mathbf{x} = \mathbf{b}$ estaremos resolviendo $(A + \delta_A)\mathbf{x} = (\mathbf{b} + \delta_\mathbf{b})$. Esto significa que existirán perturbaciones tanto en A como en $\mathbf{b}$, y la solución hallada se apartará de la original en $\mathbf{x} + \delta\mathbf{x}$. Analizaremos el error relativo de estas perturbaciones.

Aplicación a estimación de error ($\mathbf{b} + \delta_\mathbf{b}$): Si consideramos un sistema de ecuaciones lineales y adicionamos solamente un vector de errores $\delta_\mathbf{b}$ al término independiente, obtendremos una solución que consistirá en la solución sin error, $\mathbf{x}$ con un vector de errores adicionado $\delta_\mathbf{x}$

$$A\left(\mathbf{x} + \delta_\mathbf{x}\right) = \mathbf{b} + \delta_\mathbf{b}$$

dado que suponemos que A es invertible podemos escribir

$$\mathbf{x} + \delta_\mathbf{x} = A^{-1}(\mathbf{b} + \delta_\mathbf{b}) = A^{-1}\,\mathbf{b} + A^{-1}\,\delta_\mathbf{b}$$

dado que $A^{-1}\,\mathbf{b} = \mathbf{x}$ anulamos el x de ambos lados y aplicamos norma, obteniendo

$$\|\delta_\mathbf{x}\| = \|A^{-1}\,\delta_\mathbf{b}\| \leq \|A^{-1}\|\,\|\delta_\mathbf{b}\|$$

desigualdad que puede ser dividida por la norma de x para obtener

$$\frac{\|\delta_\mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{\|A^{-1}\|\,\|\delta_\mathbf{b}\|}{\|\mathbf{x}\|}$$

multiplicamos y dividimos por $\|A\|$ y obtenemos el número de condición en el numerador

$$\frac{\|\delta_\mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{k(A)\,\|\delta_\mathbf{b}\|}{\|A\|\,\|\mathbf{x}\|}$$

por otra parte es simple ver que se cumple

$$\frac{1}{\|A\|\,\|\mathbf{x}\|} \leq \frac{1}{\|\mathbf{b}\|}$$

por lo tanto obtenemos

$$\boxed{\frac{\|\delta_\mathbf{x}\|}{\|\mathbf{x}\|} \leq k(A)\,\frac{\|\delta_\mathbf{b}\|}{\|\mathbf{b}\|}}$$

Aplicación a estimación de error $(A+\delta_A)$: Si ahora consideramos una perturbación en A:

$$(A+\delta_A)(\mathbf{x}+\delta_\mathbf{x}) = \mathbf{b} \Rightarrow (A+\delta_A)\delta_\mathbf{x} + A\mathbf{x} + \delta_A\mathbf{x} = \mathbf{b}$$

$$A\delta\mathbf{x} + \delta_A(\mathbf{x}+\delta_\mathbf{x}) = 0 \Rightarrow \delta_\mathbf{x} = -A^{-1}\delta_A(\mathbf{x}+\delta_\mathbf{x})$$

Aplicando normas:

$$\|\delta_\mathbf{x}\| \leq \|A^{-1}\|\|\delta_A\|\|\mathbf{x}+\delta_\mathbf{x}\| \Rightarrow \boxed{\frac{\|\delta_\mathbf{x}\|}{\|\mathbf{x}+\delta_\mathbf{x}\|} \leq k(A)\frac{\|\delta_A\|}{\|A\|}}$$

En general el número de condición de A se estima por otro método (si el problema está mal condicionado díficilmente pueda conocer A^{-1}).

Ejemplo 2.3.1. Si el error relativo en $\mathbf{b}$ es bajo, $\frac{\|\delta_\mathbf{b}\|}{\|\mathbf{b}\|} = 10^{-3}$, pero el número de condición es alto, $k(A) = 10^6$, entonces, la perturbación a la salida puede llegar a ser grande: $\frac{\|\delta_\mathbf{x}\|}{\|\mathbf{x}\|} \leq 10^3$.

Error residual: Sea $\mathbf{x}^*$ una aproximación a la solución de $A\mathbf{x} = \mathbf{b}$, definimos el residuo $\mathbf{r} = b - A\mathbf{x}^*$.

Nos preguntamos, ¿si el residuo es "pequeño", se cumplirá qué el error en la solución es también "pequeño"?

Veamos:

$$\mathbf{r} = b - A\mathbf{x}^* = A\mathbf{x} - A\mathbf{x}^* = A(\mathbf{x}-\mathbf{x}^*) \Rightarrow \mathbf{x}-\mathbf{x}^* = A^{-1}\mathbf{r}$$

Por otra parte:

$$\|\mathbf{r}\| = \|A(\mathbf{x}-\mathbf{x}^*)\| \leq \|A\|\|\mathbf{x}-\mathbf{x}^*\|$$

Por lo que combinando ambas ecuaciones:

$$\frac{\|\mathbf{r}\|}{\|A\|} \leq \|\mathbf{x}-\mathbf{x}^*\| \leq \|A^{-1}\|\|\mathbf{r}\|$$

Por otra parte:

$$A\mathbf{x} = \mathbf{b} \Rightarrow \|\mathbf{b}\| \leq \|A\|\|\mathbf{x}\|$$
$$\mathbf{x} = A^{-1}\mathbf{b} \Rightarrow \|\mathbf{x}\| \leq \|A^{-1}\|\|\mathbf{b}\|$$

Por lo que combinando ambas ecuaciones:

$$\frac{\|\mathbf{b}\|}{\|A\|} \leq \|\mathbf{x}\| \leq \|A^{-1}\|\|\mathbf{b}\|$$

Finalmente, juntando todo se llega a:

$$\frac{\|\mathbf{r}\|}{\|b\|}\frac{1}{\|A\|\|A^{-}1\|} \leq \frac{\|\mathbf{x}-\mathbf{x}^*\|}{\|\mathbf{x}\|} \leq \|A^{-1}\|\|A\|\frac{\|\mathbf{r}\|}{\|b\|}$$

$$\frac{\|\mathbf{r}\|}{\|b\|}\frac{1}{k(A)} \leq \frac{\|\mathbf{x}-\mathbf{x}^*\|}{\|\mathbf{x}\|} \leq k(A)\frac{\|\mathbf{r}\|}{\|b\|}$$

Mencionamos como conclusiones que si $\mathbf{r}$ es pequeño y $k(A)$ es elevado (matriz mal condicionada), entonces el residuo no da información sobre la calidad de la solución x^*.

Por otra parte, si $\mathbf{r}$ es pequeño y $k(A)$ es bajo (matriz bien condicionada), entonces el residuo da información sobre la calidad de la solución x^*.

Ejemplo 2.3.2. Sea $A = \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}$, y el sistema $A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Entonces $A^{-1} = \begin{bmatrix} \frac{-1}{1-\epsilon} & \frac{1}{1-\epsilon} \\ \frac{1}{1-\epsilon} & \frac{-\epsilon}{1-\epsilon} \end{bmatrix}$.

Calculando $k(A)$ usando $\|\cdot\|_\infty$ obtenemos que $\|A\|_\infty = \text{máx}\{1+|\epsilon|, 2\}$ y que $\|A^{-1}\|_\infty = \text{máx}\{\frac{2}{|1-\epsilon|}, \frac{1-|\epsilon|}{|1-\epsilon|}\}$.

Entonces, para $\epsilon \approx 1$, tenemos que $\|A\|_\infty \approx 2$ y $\|A^{-1}\|_\infty \approx \frac{2}{|1-\epsilon|}$.

Así, $k(A) \approx \frac{4}{|1-\epsilon|}$ que va a ser muy grande (ya que $\epsilon \approx 1$).

Ejemplo 2.3.3. Sea $A = \begin{bmatrix} 1,2969 & 0,8648 \\ 0,2161 & 0,1441 \end{bmatrix}$, y $\mathbf{b} = \begin{bmatrix} 0,8642 \\ 0,1440 \end{bmatrix}$.

Supongamos que $\mathbf{x}^* = \begin{bmatrix} 0,9911 \\ -0,4870 \end{bmatrix}$.

Entonces $\mathbf{r} = \mathbf{b} - A\mathbf{x}^* = \begin{bmatrix} -10^{-8} \\ 10^{-8} \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Sin embargo la solución real es $\mathbf{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

En este caso lo que ocurre es que $k(A) \approx 3,3 \times 10^8$, por lo que $3,35 \times 10^{-17} \leq \frac{\|\mathbf{x}-\mathbf{x}^*\|}{\|\mathbf{x}\|} \leq 3,78$.

2.4. Métodos indirectos

Los métodos indirectos como mencionamos al inicio del capítulo se basan en aproximar sucesivamente la solución a un sistema. Ya tenemos una noción de cercanía de acuerdo a lo trabajado en la sección anterior, por lo que tenemos los ingredientes para establecer si una sucesión es convergente o no.

2.4.1. Método de Jacobi

Consideremos un sistema $A\mathbf{x} = \mathbf{b}$, con $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ invertible, $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Sabemos que para la solución real se cumple: $b_i = \sum_{j=1}^{n} a_{ij}x_j$.

$b_i = a_{ii}x_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j = b_i$

De donde, asumiendo $a_{ii} \neq 0$: $x_i = \frac{b_i - \sum_{j=1 j\neq i}^{n} a_{ij}x_j}{a_{ii}} \quad \forall i = 1, \ldots, n.$

Esta igualdad no tiene sentido, ya que requiere conocer el vector $\mathbf{x}$ para hallar el vector $\mathbf{x}$.

Sin embargo podemos generar un método iterativo de la forma:

$$\text{(Jacobi):} \begin{cases} x_i^{k+1} = \frac{b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j^k}{a_{ii}} & \forall i = 1, \ldots, n \\ \mathbf{x}^0 \text{ punto inicial} & \mathbf{x}^0 = \begin{bmatrix} x_1^0 \\ \vdots \\ x_n^0 \end{bmatrix}. \end{cases}$$

Este método se llama Método de Jacobi y el punto $\mathbf{x}^0$ es el punto inicial. Ahora bien, podríamos preguntarnos ¿será que esta sucesión así definida converge a la solución del sistema $A\mathbf{x} = \mathbf{b}$? Veremos más adelante que, efectivamente, si se cumplen ciertas condiciones sobre la matriz A el método converge. Estudiaremos también cómo se debe elegir el punto $\mathbf{x}^0$, si es necesario que esté en alguna región particular, cuál es la velocidad de convergencia del método, etc.

Antes de continuar observemos otra forma de escribir la iteración del método:

$$x_i^{k+1} = x_i^k + \frac{b_i - \sum_{j=1}^{n} a_{ij}x_j^k}{a_{ii}} \quad \forall i = 1, \ldots, n$$

Implementación de Jacobi:

Algoritmo 2 Pseudo-código: Jacobi

Sea A una matriz con entradas $A_{i,j} = a(i,j)$, y vectores $\mathbf{b}$ y $\mathbf{x}^0$

```
k = 1
error = ínf
while error > tolerancia & k < max_iteraciones do
  for i = 1 → n do
    x_i^{k+1} = x_i^k + (b_i − Σ_{j=1}^{n} a_ij x_j^k) / a_ii
  end for
  error = norm(x^{k+1} − x^k)
  k = k + 1
end while
```

Los valores de tolerancia del error y máximo de iteraciones se determinarán de acuerdo a la experiencia del usuario y requerimientos del problema.

2.4.2. Método de Gauss-Seidel (GS)

El método de Gauss-Seidel introduce una variante del método anterior. Para ello, es importante observar que el método de Jacobi, requiere conocer completamente el vector $\mathbf{x}^k$ para calcular

$\mathbf{x}^{k+1}$. Sin embargo, las entradas del vector $\mathbf{x}^{k+1}$ se van calculando una a una, y podrían utilizarse para para calcular la entrada siguiente, en lugar de utilizar las del vector anterior. Es decir, para calcular x_i^{k+1} se precisan $x_1^k, x_2^k, \ldots, x_{i-1}^k, x_{i+1}^k, \ldots, x_n^k$. Pero las primeras $x_1^{k+1}, x_2^{k+1}, \ldots, x_{i-1}^{k+1}$ ya están calculadas. Por tanto, el método de Gauss-Seidel hace uso de esa información y en general veremos que se mejora el método.

Explícitamente, la iteración queda expresada:

$$(Gauss-Seidel): \begin{cases} x_i^{k+1} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij} x_j^k}{a_{ii}} & \forall i = 1, \ldots, n \\ \\ \mathbf{x}^0 \text{ punto inicial} & \mathbf{x}^0 = \begin{bmatrix} x_1^0 \\ \vdots \\ x_n^0 \end{bmatrix}. \end{cases}$$

Observación 2.4.1. Nuevamente debemos pedir $a_{ii} \neq 0$.

Ejercicio 2.4.1. Modificar el código de Jacobi para implementar Gauss-Seidel.

2.4.3. Expresión Matricial de Jacobi y Gauss-Seidel

Dado un sistema de ecuaciones lineales

$$A\,\mathbf{x} = \mathbf{b}, \qquad A = \begin{pmatrix} \searrow & -F \\ D & \\ -E & \searrow \end{pmatrix}$$

aplicamos la descomposición de la matriz A en sus componentes triangular inferior $-E$, diagonal D y triangular superior $-F$.

Veamos cómo es posible escribir los métodos vistos hasta el momento en forma matricial.

Jacobi:

$$A = D - E - F \qquad \Rightarrow \qquad D\mathbf{x} = (E+F)\mathbf{x} + \mathbf{b}$$

$$\mathbf{x}^{(k+1)} = D^{-1}(E+F)\mathbf{x}^{(k)} + D^{-1}\mathbf{b}$$

lo podemos escribir como

$$\begin{cases} \mathbf{x}^{(k+1)} = Q_J\,\mathbf{x}^{(k)} + \mathbf{r}_J & Q_J = D^{-1}(E+F) \\ \mathbf{x}^{(0)} = \mathbf{x}_0 & \mathbf{r}_J = D^{-1}\,\mathbf{b} \end{cases}$$

Gauss-Seidel:

$$A = D - E - F \qquad \Rightarrow \qquad (D-E)\mathbf{x} = F\mathbf{x} + \mathbf{b}$$

$$\mathbf{x}^{(k+1)} = (D-E)^{-1}F\mathbf{x}^{(k)} + (D-E)^{-1}\mathbf{b}$$

lo podemos escribir como

$$\begin{cases} \mathbf{x}^{(k+1)} = Q_{GS}\,\mathbf{x}^{(k)} + \mathbf{r}_{GS} & Q_{GS} = (D-E)^{-1}F \\ \mathbf{x}^{(0)} = \mathbf{x}_0 & \mathbf{r}_{GS} = (D-E)^{-1}\,\mathbf{b} \end{cases}$$

2.4.4. Método Iterativo Matricial

Los métodos indirectos para resolver sistemas lineales son iterativos. Por lo tanto resulta útil expresar las operaciones que realizan de forma $\mathbf{x}^{(k+1)} = g(\mathbf{x}^{(k)})$. Dado que $\mathbf{x}^{(k)}$ es un vector en el caso general, se trabaja con matrices, quedando expresado de la siguiente forma:

$$(M): \begin{cases} \mathbf{x}^{(k+1)} = Q\mathbf{x}^{(k)} + \mathbf{r} & \mathbf{x}^{(k)} \in \mathbb{R}^n \quad k = 0, 1, \dots \\ \mathbf{x}^{(0)} = \mathbf{x}_0 & Q \in \mathcal{M}_{n\times n}(\mathbb{R}),\ \mathbf{r} \in \mathbb{R}^n \end{cases}$$

En general, dado el sistema $A\mathbf{x} = \mathbf{b}$, elegimos una matriz $M \in \mathcal{M}_{n\times n}(\mathbb{R})$ invertible y planteamos:

$$A\mathbf{x} = \mathbf{b} \Leftrightarrow M\mathbf{x} = (M - A)\mathbf{x} + b$$

En los métodos iterativos se escoge una matriz M relacionada con A (observar Jacobi y G-S) y se genera una sucesión de vectores $\{\mathbf{x}^{(k)}\}_{k\geq 0}$ a partir de la ecuación

$$M\mathbf{x}^{(k+1)} = (M - A)\mathbf{x}^{(k)} + b$$

Si $\{\mathbf{x}^{(k)}\}_{k\geq 0}$ resulta convergente, la convergencia será hacia la solución de $A\mathbf{x} = \mathbf{b}$. Veremos que muchas veces es posible elegir un $\mathbf{x}^{(0)}$ inicial arbitrario (cualquiera).

Por tanto, si escribimos el sistema como $M\mathbf{x}^{(k+1)} = (M - A)\mathbf{x}^{(k)} + b$, la iteración estacionaria será:

$$\begin{cases} \mathbf{x}^{(k+1)} = Q\,\mathbf{x}^{(k)} + \mathbf{r} & Q = M^{-1}(M - A) \\ \mathbf{x}^{(0)} = \mathbf{x}_0 & \mathbf{r} = M^{-1}\,\mathbf{b} \end{cases}$$

Observación 2.4.2. Es una iteración estacionaria porque tanto Q como r no dependen del paso k, y es de orden 1 porque $\mathbf{x}^{(k+1)}$ depende solamente del valor anterior $\mathbf{x}^{(k)}$ (y no de pasos anteriores).

Observación 2.4.3. $\lim_{k\to\infty} M\mathbf{x}^{(k+1)} = \lim_{k\to\infty}(M - A)\mathbf{x}^{(k)} + b$. Si $\lim_{k\to\infty} \mathbf{x}^{(k+1)} = \mathbf{x}^*$ tenemos que: $\lim_{k\to\infty} M\mathbf{x}^{(k+1)} = M\mathbf{x}^* = (M - A)\mathbf{x}^* + b = \lim_{k\to\infty}(M - A)\mathbf{x}^{(k)} + b$.

Definición 2.4.1 (Punto Fijo)**.** Dado un método iterativo matricial (M). Diremos que $\mathbf{x}^*$ es un punto fijo del mismo si y solo si

$$\mathbf{x}^* = Q\mathbf{x}^* + r$$

En este caso decimos que el método iterativo es consistente.

Lema 2.4.1. *Dada una matriz cuadrada* $A \in \mathcal{M}_{n\times n}(\mathbb{R})$

$$\lim_{k\to\infty} A^k = 0 \quad \Leftrightarrow \quad \rho(A) < 1$$

Observación 2.4.4. Notamoms que aquí 0 representa la matriz nula.

Demostración. ($\Rightarrow$) Por absurdo supongamos que $\rho(A) \geq 1$, entonces existe λ valor propio de A tal que $|\lambda| \geq 1$ y $Av = \lambda v$, con v vector propio asociado a λ.

$$\Rightarrow \|A^k\| \geq \frac{\|A^k v\|}{\|v\|} = |\lambda|^k \xrightarrow[k\to+\infty]{} +\infty \Rightarrow \lim_{k\to\infty} A^k \neq 0$$

($\Leftarrow$) Como $\rho(A) < 1$, entonces existe $\varepsilon > 0$ tal que $\rho(A) < 1 - \varepsilon$ y además existe una norma inducida $\|\cdot\|_\varepsilon$ que cumple $\|A\|_\varepsilon \leq \rho(A) + \varepsilon < 1$. Ahora, $\|A^k\| \leq \|A\|^k < 1$ y entonces $\|A^k\|$ está acotada por $\|A\|^k$ que tiende a cero con k. Así $\lim_{k\to\infty} A^k = 0$. □

Observación 2.4.5. Hay métodos eficientes para estimar el radio espectral.

Denotaremos al vector de error en el paso k-ésimo como $e^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$.

Proposición 2.4.2. $\lim_k \mathbf{x}^{(k)} = \mathbf{x}^*$ *sii* $\lim_k e^{(k)} = \vec{0}$ *sii* $\lim_k \|e^{(k)}\| = 0$

Teorema 2.4.3. *La iteración estacionaria* $\begin{cases} \mathbf{x}^{(k+1)} = Q\mathbf{x}^{(k)} + \mathbf{r} \\ \mathbf{x}^{(0)} \in \mathbb{R}^n \end{cases}$ *es convergente sii* $\rho(Q) < 1$.

Demostración. Sea la solución del sistema $\mathbf{x}^* \in \mathbb{R}^n$ tal que $\mathbf{x}^* = Q\mathbf{x}^* + \mathbf{r}$ y $e^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$ el error en el paso k. Utilizando que $\mathbf{x}^*$ es punto fijo de la iteración tenemos que:

$$e^{(k+1)} = \mathbf{x}^{(k+1)} - \mathbf{x}^* = Q\mathbf{x}^{(k)} + r - (Q\mathbf{x}^* + \mathbf{r}) = Q(\mathbf{x}^{(k)} - \mathbf{x}^*) = Qe^{(k)}.$$

Luego, por inducción en los naturales tenemos que $e^{(k)} = Q^k e^{(0)}$. Vamos ahora a probar el directo y el recíproco en partes:

($\Rightarrow$) Supongamos por absurdo que $\rho(Q) \geq 1$. En tal caso, existe un valor propio λ de Q, con $|\lambda| \geq 1$, y un vector propio $v \neq 0$ tal que $Qv = \lambda v$. Elijamos $\mathbf{x}^{(0)}$ de modo que $e^{(0)} = v$. Esto es posible tomando $\mathbf{x}^{(0)} = \mathbf{x}^* + e^{(0)}$. Por lo anteriormente observado, tenemos que:

$$e^{(k)} = Q^k e^{(0)} = Q^k v = \lambda^k v,$$

y tomando normas, tenemos que $\|e^{(k)}\| = |\lambda|^k \|v\|$. Tomando límites en ambos miembros, se consigue que $\lim_k \|e^{(k)}\| \neq 0$, pues $|\lambda| \geq 1$ y $\|v\| \neq 0$. Esto es decir que el error no tiende al vector nulo, o equivalentemente, que la sucesión $\{x^k\}_{k\in\mathbb{N}}$ no converge a $\mathbf{x}^*$, en contradicción con la hipótesis.

($\Leftarrow$) Por el Teorema del Radio Espectral, el radio espectral es el ínfimo de las normas operadores. Sea ε igual a la mitad de la distancia entre $\rho(Q)$ y 1, es decir, $\varepsilon = \frac{1-\rho(Q)}{2}$. Por definición de ínfimo, existe una norma operador $\|\cdot\|_\varepsilon$ tal que $\|Q\|_\varepsilon - \rho(Q) < \varepsilon$. Pero entonces:

$$\|Q\|_\varepsilon < \rho(Q) + \varepsilon = \frac{2\rho(Q) + (1 - \rho(Q))}{2} = \frac{1 + \rho(Q)}{2} < 1.$$

Hemos conseguido así una norma $\|\cdot\|_\varepsilon$ compatible con una vectorial tal que $\|Q\|_\varepsilon < 1$. Como $e^{(k)} = Q^k e^{(0)}$, tomando normas en cada miembro tenemos que:

$$\|e^{(k)}\| = \|Q^k e^{(0)}\| \leq (\|Q\|_\varepsilon)^k \|e^{(0)}\|,$$

y tomando límite con k tenemos que $0 \leq \lim_k \|e^{(k)}\| \leq (\|Q\|_\varepsilon)^k \|e^{(0)}\| = 0$. La única opción válida es que $\lim_k \|e^{(k)}\| = 0$, y por la primera propiedad de una norma tenemos que la sucesión de vectores $\{e^{(k)}\}_{k\in\mathbb{N}}$ converge al vector nulo. Esto significa que $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}}$ converge a $\mathbf{x}^*$.

□

Observación 2.4.6. Recalcamos que esto significa que $\mathbf{x}^{(k)}$ converge independientemente del dato inicial $\mathbf{x}^{(0)}$.

Podemos ver como corolario que si en alguna norma $\|Q\| < 1 \Rightarrow \rho(Q) \leq \|Q\| < 1$, la iteración es convergente.

El teorema anterior es de gran utilidad ya que permite establecer si un método será convergente o no dependiendo de la matriz Q, para cualquier método que pueda escribirse como una ecuación estacionaria. Sin embargo debemos encontrar la matriz Q asociada al sistema $A\mathbf{x} = \mathbf{b}$. A continuación veremos algunos criterios para determinar la convergencia a partir de características de la matriz A.

Definición 2.4.2 (Matriz diagonal dominante)**.** Sea A una matriz $n \times n$, decimos que es diagonal dominante por filas si y solo si

$$\sum_{j=1, j\neq i}^{n} |a_{ij}| < |a_{ii}| \qquad i = 1, \ldots, n$$

Proposición 2.4.4 (Convergencia de Jacobi)**.** *A es diagonal dominante por filas (o por columnas) $\Rightarrow$ la sucesión generada por Jacobi converge.*

Demostración. Caso filas:

$$\sum_{j=1, j\neq i}^{n} |a_{ij}| < |a_{ii}| \Rightarrow \sum_{j=1, j\neq i}^{n} \frac{|a_{ij}|}{|a_{ii}|} < 1 \Rightarrow \|Q_J\|_\infty < 1 \Rightarrow \text{ Jacobi converge}$$

Caso columnas: Ejercicio. □

Proposición 2.4.5 (Convergencia de Gauss-Seidel)**.** *A es diagonal dominante por filas (o por columnas) $\Rightarrow$ la sucesión generada por Gauss-Seidel converge.*

Observación 2.4.7. Las proposiciones anteriores indican que si la matriz A es estrictamente diagonal dominante por filas (o por columnas), entonces tanto Jacobi como Gauss-Seidel son convergentes, para toda condición inicial x_0. Esto es una condición suficiente. Es decir que el método podría ser convergente aunque A no sea estrictamente diagonal dominante.

Ejercicio 2.4.2. Se considera $A = \begin{pmatrix} 3 & -1 \\ 1 & \beta \end{pmatrix}$. Sin calcular $\rho(Q)$, indicar un rango de valores de β que asegure convergencia de Jacobi y Gauss-Seidel.

Observación 2.4.8.

- Para matrices esparsas, los métodos iterativos permiten encontrar la solución en forma rápida y eficiente.
- No se usan para matrices mal condicionadas.
- Hay variantes para acelerar la convergencia.
- Típicamente Gauss-Seidel es más rápido que Jacobi.
- El radio espectral determina la velocidad de convergencia.

Velocidad de convergencia de MIG

Dado un método iterativo, de la forma $\begin{cases}\mathbf{x}^{(k+1)} = Q\mathbf{x}^{(k)} + \mathbf{r} \\ \mathbf{x}^{(0)} = \mathbf{x}_0\end{cases}$ con $\mathbf{x}^{(k)}, \mathbf{r} \in \mathbb{R}^n$, $Q \in \mathcal{M}_{n\times n}(\mathbb{R})$.

Sean $\{\lambda_1, \ldots, \lambda_n\}$ los valores propios de Q que asumiremos reales y satisfaciendo la relación: $\rho(Q) = |\lambda_1| > |\lambda_2| > \cdots > |\lambda_{n-1}| > |\lambda_n| \geq 0$. Sean $\{v_1, \ldots, v_n\}$ los vectores propios de Q asociados a dichos valores propios. Entonces, si

$$\left.\begin{array}{l} e^{(k+1)} = \mathbf{x}^{(k+1)} - \mathbf{x}^* \\ \rho(Q) < 1 \end{array}\right\} \Rightarrow \frac{\|e^{(k+1)}\|}{\|e^{(k)}\|} \xrightarrow[k\to\infty]{} \rho(Q)$$

$e^{(k)}$ es el error en la aproximación en el paso k-ésimo.

Supongamos que $e^{(0)} = \sum_{i=1}^n \alpha_i v_i$, sabemos que el error en el k-ésimo paso satisface $e^{(k)} = Q^k e^{(0)}$. Con lo cual:

$$e^{(k)} = \sum_{i=1}^n Q^k \alpha_i v_i = \sum_{i=1}^n (\lambda_i)^k \alpha_i v_i = (\lambda_1)^k \alpha_1 v_1 + \sum_{i=2}^n (\lambda_i)^k \alpha_i v_i,$$

de la misma manera tenemos que:

$$e^{(k+1)} = (\lambda_1)^{k+1} \alpha_1 v_1 + \sum_{i=2}^n (\lambda_i)^{k+1} \alpha_i v_i.$$

Entonces: $\lim_{k\to+\infty} \frac{\|e^{(k+1)}\|}{\|e^{(k)}\|} = \lim_{k\to+\infty} \frac{\|(\lambda_1)^{k+1}\alpha_1 v_1 + (\lambda_1)^{k+1}\sum_{i=2}^n \frac{(\lambda_i)^{k+1}}{(\lambda_1)^{k+1}}\alpha_i v_i\|}{\|(\lambda_1)^k \alpha_1 v_1 + (\lambda_1)^k \sum_{i=2}^n \frac{(\lambda_i)^k}{(\lambda_1)^k}\alpha_i v_i\|} =$

$$= \lim_{k\to+\infty} \frac{\|(\lambda_1)^{k+1}\alpha_1 v_1\|}{\|(\lambda_1)^k \alpha_1 v_1\|} = |\lambda_1|.$$

Ejemplo 2.4.1. Si $\rho(Q_1) = 0,4$ y $\rho(Q_2) = 0,75$ el método convergerá más rápido para Q_1.

Ejemplo 2.4.2. Calcular y comparar los valores de $\rho(Q_J)$ y $\rho(Q_{GS})$ para $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$.

En primer lugar, sabemos que ambos métodos son convergentes por ser la matriz A diagonal dominante.

$Q = M^{-1}(M - A)$. Denotando $A = D - E - F$, donde $-E$ es la matriz subdiagonal inferior y $-F$ es la matriz por encima de la diagonal D; en Jacobi se tiene: $Q_J = D^{-1}(E + F)$ (se toma $M = D$). En Gauss-Seidel se tiene: $Q_{GS} = (D - E)^{-1}(F)$ (se toma $M = D - E$).

Tendremos que: $Q_J = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}$, y sus valores propios son $1/2$ y $-1/2$ con lo cual $\rho(Q_J) = \frac{1}{2}$.

Además $Q_{GS} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix}$, y sus valores propios son 0 y $1/4$ con lo cual $\rho(Q_{GS}) = \frac{1}{4}$.

Gauss-Seidel convergerá el doble de rápido que Jacobi en este caso.

Condición de parada de MIG

Nos preguntamos a continuación cómo saber cuándo detener la iteración, ya que no conocemos el valor real de la solución, es decir, no podemos imponer $\|e^{(k)}\| = \|\mathbf{x}^{(k)} - \mathbf{x}^*\| < \varepsilon$. Podríamos imponer $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \varepsilon$, pero qué relación tendría esto con la condición usando la solución verdadera:

$$\begin{aligned}\mathbf{x}^{(k+1)} - \mathbf{x}^* &= Q(\mathbf{x}^{(k)} - \mathbf{x}^*) \\ &= -Q(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) + Q(\mathbf{x}^{(k+1)} - \mathbf{x}^*)\end{aligned}$$

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| \leq \|Q\|\|(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})\| + \|Q\|\|(\mathbf{x}^{(k+1)} - \mathbf{x}^*)\|$$

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|(1 - \|Q\|) \leq \|Q\|\|(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})\|$$

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| \leq \frac{\|Q\|}{1 - \|Q\|}\|(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})\|$$

Es así que como decíamos, imponiendo $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \varepsilon$ obtendríamos una cota para el error en el paso $k+1$ que está relacionado con la diferencia entre los valores calculados en pasos anteriores:

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| \leq \frac{\|Q\|}{1 - \|Q\|}\,\varepsilon$$

Como caso particular, si $\|Q\| < \frac{1}{2}$, entonces $\frac{\|Q\|}{1-\|Q\|} < 1$, por lo que para alcanzar una cota de error ε alcanza que $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \approx \varepsilon$.

2.5. Métodos de Sobrerrelajación

Las técnicas de sobrerrelajación para resolución de sistemas lineales usando métodos iterativos tienen el cometido es lograr convergencia partiendo de métodos iterativos que no resultan convergentes, o acelerar la velocidad de convergencia de los que sí son convergentes. Para ello, se realiza una especie relajación convexa entre los puntos hallados en los pasos k y el punto $k+1$ hallado por el método elegido:

$$\mathbf{x}^{k+1} = \omega \mathbf{x}_{(M)}^{(k+1)} + (1-\omega)\mathbf{x}^{(k)} \quad \forall i = 1, \ldots, n$$

El valor de ω se escoge optimizando la velocidad de convergencia.

Observación 2.5.1. Si $\omega > 1$ se llama sobrerrelajación (aceleran convergencia de Jacobi y G-S). Si $\omega < 1$ se llama subrelajación (Jacobi y G-S no convergen).

2.5.1. Sobrerrelajación de Jacobi

La iteración queda:

$$x_i^{k+1} = \frac{\omega}{a_{ii}} \left[b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^k \right] + (1-\omega) x_i^k \quad \forall i = 1, \ldots, n$$

Y la matriz de iteración:

$$Q = \omega Q_j + (1-\omega) Id$$

Ejemplo 2.5.1. El método de Jacobi no converge para un sistema $Ax = b$ con matriz:

$$A = \begin{pmatrix} 1 & -6 \\ 2 & 3 \end{pmatrix}.$$

y vector $b = (1, 0)^t$.

Esto puede deducirse ya que aunque en este caso la matriz A no es diagonal dominante, se puede utilizar el criterio de $\rho(Q_J)$. La matriz de iteración es:

$$Q_J = \begin{pmatrix} 0 & 6 \\ -\frac{2}{3} & 0 \end{pmatrix}$$

Sus valores propios son $\lambda = \pm 2i$. Por lo tanto $\rho(Q_J) = 2 > 1$ y se concluye que el método no es convergente.

Sin embargo, consideremos la relajación del método de Jacobi donde $Q = \omega Q_j + (1-\omega) Id$ es la nueva matriz de iteración. Analicemos si existe $\omega > 0$ que garantice convergencia de este método para el sistema lineal anterior. Ahora:

$$Q = \begin{pmatrix} 1-\omega & 6\omega \\ -\frac{2}{3}\omega & 1-\omega \end{pmatrix}$$

Sus valores propios son $\lambda = (1-\omega) \pm 2i\omega$. Busquemos $\omega > 0$ para que ambos valores propios tengan magnitud inferior a 1. Tenemos que $|\lambda|^2 = (1-\omega)^2 + 4\omega^2 = 5\omega^2 - 2\omega + 1 < 1$, o equivalentemente, $5\omega^2 - 2\omega < 0$. Usando que $\omega > 0$ y factorizando, tenemos que si $\omega \in (0, 2/5)$ se asegura convergencia. Conseguimos así relajar el método de Jacobi en un caso que no era convergente y traducirlo a otro método que sí es convergente.

2.5.2. Sobrerrelajación sucesiva

El método SOR (por *Successive Over-Relaxation*) es la aplicación a Gauss-Seidel de una relajación análoga a la realizada en la sección anterior.

La iteración queda:

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

Se toma $M = \frac{1}{\omega} D - E$, $\omega \in (1,2)$ ($\omega = 1$ es GS).

Ejemplo 2.5.2.

$$A = \begin{bmatrix} 2 & 2 & 2 \\ -3 & 3 & 5 \\ -2 & 4 & -2 \end{bmatrix} \Rightarrow M = \begin{bmatrix} \frac{2}{\omega} & 0 & 0 \\ -3 & \frac{3}{\omega} & 0 \\ -2 & 4 & \frac{-2}{\omega} \end{bmatrix}$$

Observación 2.5.2.

- El ω óptimo es aquel que minimiza $\rho(Q_{SOR})$.

$$\omega_{opt} = \min_{\omega \in (1,2)} \rho(Q_{SOR})$$

- En general $\rho(Q_{SOR})$ es no lineal en ω.