Clase 16 - Resolución de sistemas lineales

El problema

Se desea encontrar $x_1, \ldots, x_n$ solución del siguiente sistema de ecuaciones lineales:

$$\begin{cases} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + & \ldots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & + & \ldots & + & a_{2n}x_n & = & b_2 \\ a_{31}x_1 & + & a_{32}x_2 & + & a_{33}x_3 & + & \ldots & + & a_{3n}x_n & = & b_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & a_{n3}x_3 & + & \ldots & + & a_{nn}x_n & = & b_n \end{cases}$$

Es conveniente escribir este sistema en forma matricial:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \ldots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \ldots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots\vdots & \\ a_{n1} & a_{n2} & a_{n3} & \ldots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix},$$

es decir $Ax = b$, con $A \in \mathbb{R}^{n\times n}$ y $b \in \mathbb{R}^n$. Así, el problema de esta unidad puede formularse de la siguiente manera:

$$\text{Dada } A \in \mathbb{R}^{n\times n} \text{ y } b \in \mathbb{R}^n \text{ hallar } x_* \text{ solución de } Ax = b.$$

Este es uno de los problemas más importantes en Análisis Numérico en general porque la resolución de muchos problemas de diferentes disciplinas o aplicaciones termina en la resolución de sistemas lineales. Para lo cual es muy importante disponer de métodos numéricos y computacionales que sean rápidos y eficientes.

A continuación enunciamos un teorema básico de Álgebra lineal que será de utilidad en lo que sigue.

Teorema 1. *para toda matriz $A \in \mathbb{R}^{n\times n}$, las siguientes propiedades son equivalentes:*

1. *existe A^{-1}, la inversa de A. ($A^{-1}A = AA^{-1} = I$);*
2. *la matriz A es no singular, es decir, si $Ax = 0$ entonces $x = 0$;*
3. $\det(A) \neq 0$*;*
4. *las columnas de A forman una base de $\mathbb{R}^n$;*
5. *las filas de A forman una base de $\mathbb{R}^n$;*
6. *para cada $b \in \mathbb{R}^n$ existe un único $x \in \mathbb{R}^n$ solución de $Ax = b$.*

Si bien se conocen métodos matemáticos de Álgebra lineal para resolver ese sistema lineal, nuestro objetivo será estudiar métodos y algoritmos numéricos para resolver este problema de una forma sistemática y eficiente. Esta eficiencia estará asociada al costo computacional de un algoritmo. Este costo se mide en términos de cantidad de operaciones y en espacio de almacenamiento. Dado que los problemas que estudiaremos son simples y pequeños nos ocuparemos sólo de contabilizar la cantidad de operaciones que realiza cada algoritmo para obtener la solución buscada.

Antes de estudiar el caso de una matriz general, consideraremos la resolución de sistemas lineales de dos casos muy simples.

Sistemas diagonales

Definición 1. *Una matriz A es diagonal si y sólo si $a_{ij} = 0$ para $i \neq j$.*

Sea $A \in \mathbb{R}^{n\times n}$ una matriz diagonal

$$A = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix}.$$

Luego A es no singular si y sólo sí $a_{ii} \neq 0$, $i = 1,\dots,n$ si y sólo si $\det(A) = a_{11}a_{22}\dots a_{nn} \neq 0$.

Entonces $Ax = b$ si y sólo si $a_{ii}x_i = b_i$, para $i = 1,\dots,n$, si y sólo si $x_i = \dfrac{b_i}{a_{ii}}$, para $i = 1,\dots,n$.

Algoritmo 1: resolución de sistema lineal con matriz diagonal.

input n, A, b

for $i = 1,\dots,n$ **do**

$x_i \leftarrow b_i/a_{ii}$

output i, x_i

end for

end

Costo computacional: n operaciones (productos) en punto flotante, y así se tiene $\mathscr{O}(n)$ flops.

Observación: notar que como la matriz A es diagonal y no singular, los únicos elementos no nulos están precisamente en la diagonal. Entonces, sería suficiente almacenar los elementos diagonales de A en un vector $d = diag(A)$, y así calcular $x(i) = b(i)/d(i)$ para $i = 1,\dots,n$.

Sistemas triangulares

Definición 2. *Una matriz A es triangular inferior (superior) si y sólo si $a_{ij} = 0$ para $i < j$ ($a_{ij} = 0$ para $i > j$).*

$A \in \mathbb{R}^{n\times n}$ es triangular inferior y no singular si y sólo si $\det(A) = a_{11}a_{22}\dots a_{nn} \neq 0$, con

$$A = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}.$$

Entonces resolver $Ax = b$ es equivalente a

$$\begin{cases} a_{11}x_1 & = & b_1 \\ a_{21}x_1 + a_{22}x_2 & = & b_2 \\ \vdots & & \vdots \\ a_{i1}x_1 + \cdots + a_{ii}x_i & = & b_i \\ \vdots & & \vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n & = & b_n \end{cases} \Longrightarrow \begin{cases} x_1 & = & b_1/a_{11} \\ x_2 & = & \frac{1}{a_{22}}(b_2 - a_{21}x_1) \\ \vdots & & \vdots \\ x_i & = & \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j) \\ \vdots & & \vdots \\ x_n & = & \frac{1}{a_{nn}}(b_n - \sum_{j=1}^{n-1} a_{nj}x_j) \end{cases}$$

Algoritmo 2: resolución de sistema lineal con matriz triangular inferior.

input n, A, b

for $i = 1, \ldots, n$ **do**

$$x_i \leftarrow \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j\right)/a_{ii} \qquad \text{(no sumar si } i = 1\text{)}$$

output i, x_i

end for

end

Costo computacional: vamos a contar las sumas/restas y productos/ divisiones que se realizan para cada i y luego sumamos:

i	sumas	productos
1	0	1
2	1	2
3	2	3
$\vdots$	$\vdots$	$\vdots$
n	n-1	n
Total	$\frac{(n-1)n}{2}$	$\frac{n(n+1)}{2}$

Así, la cantidad total de operaciones es: $\frac{(n-1)n}{2} + \frac{n(n+1)}{2} = n^2$, es decir $\mathcal{O}(n^2)$ flops.

(Recordar que $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$).

Análogamente, Si $A \in \mathbb{R}^{n \times n}$ es triangular superior y no singular, es decir,

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ 0 & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & a_{nn} \end{bmatrix},$$

con $\det(A) = a_{11}a_{22}\ldots a_{nn} \neq 0$.

Entonces resolver $Ax = b$ es equivalente a

$$\left\{\begin{array}{lcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ \vdots & & \vdots \\ a_{ii}x_i + \cdots + a_{in}x_n & = & b_i \\ \vdots & & \vdots \\ a_{nn}x_n & = & b_n \end{array}\right. \implies \left\{\begin{array}{rcl} x_n & = & b_n/a_{nn} \\ x_{n-1} & = & \dfrac{1}{a_{n-1,n-1}}(b_{n-1} - a_{n-1,n}x_n) \\ \vdots & & \vdots \\ x_i & = & \dfrac{1}{a_{ii}}(b_i - \sum\limits_{j=i+1}^{n} a_{ij}x_j) \\ \vdots & & \vdots \\ x_1 & = & \dfrac{1}{a_{11}}(b_1 - \sum\limits_{j=2}^{n} a_{1j}x_j) \end{array}\right.$$

Algoritmo 3: resolución de sistema lineal con matriz triangular superior.

input n, A, b

for $i = n, \ldots, 1$ **do**

$$x_i \leftarrow \left(b_i - \sum_{j=i+1}^{n} a_{ij}x_j\right)/a_{ii} \qquad \text{(no sumar si } i = n\text{)}$$

output i, x_i

end for

end

Costo computacional: de manera análoga al caso anterior se puede ver que la cantidad total de operaciones es: n^2, es decir $\mathcal{O}(n^2)$ flops.

Eliminación gaussiana

Dado un sistema lineal $Ax = b$, la idea consiste en transformarlo en otro sistema equivalente $Ux = y$, que sea más fácil de resolver. Recordar que sistemas equivalentes tienen la misma solución x. La matriz U del nuevo sistema será triangular superior, cuya resolución ya vimos que es muy sencilla. Recordemos también que existen 3 tipos de operaciones elementales por filas que permiten transformar un sistema lineal ampliado en otro sistema lineal ampliado equivalente:

- reemplazar una fila por un múltiplo escalar (no nulo) de ella misma ($f_i \leftarrow \lambda f_i$, $\lambda \neq 0$).
- restar a una fila un múltiplo escalar (no nulo) de otra fila ($f_i \leftarrow f_i - \lambda f_j$, $\lambda \neq 0$).
- intercambiar dos filas ($f_i \leftrightarrow f_j$).

El método de eliminación gaussiana se basa fundamentalmente en las dos primeras operaciones elementales por filas. Es decir, se realizan $(n-1)$ pasos para obtener la matriz triangular U partiendo de A:

$$A \xrightarrow{\text{Paso 1}} A^{(1)} \xrightarrow{\text{Paso 2}} A^{(2)} \xrightarrow{\text{Paso 3}} \ldots \xrightarrow{\text{Paso } n-1} A^{(n-1)} = U$$

y las mismas operaciones que se realizan en A se deben realizar en el vector b.

Para fijar ideas, veamos con detalles cómo se hacen estos pasos en un ejemplo numérico. Consideremos el sistema lineal:

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 34 \\ 27 \\ -38 \end{bmatrix}$$

Paso 1: para pasar de la matriz A a la matriz $A^{(1)}$ se deben realizar operaciones elementales por filas para obtener ceros en las posiciones de la primera columna a partir de la segunda fila hasta la última.

Para esto, denotemos f_i a la i-ésima fila de A. Llamamos a f_1 como la fila pivote y a $a_{11} = 6$ será llamado el elemento pivote. Entonces haremos las siguientes operaciones:

$$\begin{aligned} f_2 &\leftarrow f_2 - 2f_1 \\ f_3 &\leftarrow f_3 - (1/2)f_1 \\ f_4 &\leftarrow f_4 - (-1)f_1 \end{aligned}$$

Así obtenemos,

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ 21 \\ -26 \end{bmatrix}$$

Paso 2: para pasar de $A^{(1)}$ a $A^{(2)}$ se deben realizar operaciones elementales por filas para obtener ceros en las posiciones de la segunda columna a partir de la tercera fila hasta la última. La fila pivote será la nueva f_2 y el elemento pivote será $a_{22}^{(1)} = -4$. Se realizan las siguientes operaciones:

$$\begin{aligned} f_3 &\leftarrow f_3 - 3f_2 \\ f_4 &\leftarrow f_4 - (-1/2)f_2 \end{aligned}$$

Así obtenemos,

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 4 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ -9 \\ -21 \end{bmatrix}$$

Paso 3: por último, para pasar de $A^{(2)}$ a $A^{(3)} = U$ se deben realizar operaciones elementales por filas para modificar la última fila:

$$f_4 \leftarrow f_4 - 2f_3$$

Así obtenemos,

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ -9 \\ -3 \end{bmatrix}$$

cuya solución es: $x = (1, -3, -2, 1)$.

Observación 1: notar que, en general, al pasar de A a $A^{(1)}$ en el Paso 1, se obtiene una matriz de la forma

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \xrightarrow{\text{Paso 1}} \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & & \dots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{bmatrix}$$

Más específicamente, se calcula

$$a_{ij}^{(1)} = \begin{cases} a_{ij} & \text{si } i=1,\ j=1,\dots,n \\ 0 & \text{si } i=2,\dots,n,\ j=1 \\ a_{ij} - \dfrac{a_{i1}}{a_{11}} a_{1j} & \text{si } i=2,\dots,n,\ j=2,\dots,n \end{cases} \quad \text{y} \quad b_i^{(1)} = \begin{cases} b_i & \text{si } i=1 \\ b_i - \dfrac{a_{i1}}{a_{11}} b_1 & \text{si } i=2,\dots,n. \end{cases}$$

Los coeficientes $m_{i,1} = a_{i1}/a_{11}$ para $i = 2,\dots,n$ se denominan multiplicadores.

Costo computacional: calcularemos por separado la cantidad de sumas/restas y de productos/divisiones.

Para el Paso 1:

- **Sumas/restas:** $(n-1)^2$ (submatriz $(n-1)\times(n-1)$) y $(n-1)$ (vector b).
  En total son: $(n-1)^2 + (n-1) = (n-1)(n-1+1) = n(n-1) = n^2 - n$.

- **Productos/divisiones:** $(n-1)$ (multiplicadores), $(n-1)^2$ (submatriz $(n-1)\times(n-1)$) y $(n-1)$ (vector b).
  En total son: $(n-1)^2 + 2(n-1) = (n-1)(n-1+2) = (n-1)(n+1) = n^2 - 1$.

Observación 2: para el Paso 2, es análogo a lo anterior, pero se trabaja con una matriz de un orden menor, es decir $(n-1)\times(n-1)$, y por lo tanto se requieren: $(n-1)^2 - (n-1)$ sumas/restas y $(n-1)^2 - 1$ productos/divisiones.

Observación 3: en general, para pasar de la matriz $A^{(k-1)}$ a la matriz $A^{(k)}$ en el Paso k, se calcula

$$a_{ij}^{(k)} = \begin{cases} a_{ij}^{(k-1)} & \text{si } i=1,\dots,k,\ j=1,\dots,n \\ 0 & \text{si } i=k+1,\dots,n,\ j=k \\ a_{ij}^{(k-1)} - \dfrac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)} & \text{si } i=k+1,\dots,n,\ j=k+1,\dots,n \end{cases}$$

y

$$b_i^{(k)} = \begin{cases} b_i^{(k-1)} & \text{si } i=1,\dots,k \\ b_i^{(k-1)} - \dfrac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} b_k^{(k-1)} & \text{si } i=k+1,\dots,n. \end{cases}$$

Ejercicio: mostrar que el costo total de operaciones del método de eliminación gaussiana para resolver un sistema $Ax=b$, con $A\in\mathbb{R}^{n\times n}$ es $\mathcal{O}(\frac{2}{3}n^3)$ flops.

Ayuda: recordar que $\displaystyle\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$ y $\displaystyle\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$.

Algoritmo 4: eliminación gaussiana para obtener un sistema lineal con matriz triangular superior.

input n, A, b

for $k = 1, \dots, n-1$ **do**

    for $i = k+1, \dots, n$ **do**

        if $(a_{kk} = 0)$ STOP!

        $m \leftarrow a_{ik}/a_{kk}$

        for $j = k+1, \dots, n$ **do**

            $a_{ij} \leftarrow a_{ij} - m a_{kj}$

        endfor (j)

        $b_i \leftarrow b_i - m b_k$

    endfor (i)

endfor (k)

output A, b

end

Observación 1: notar que estamos almacenando la matriz triangular superior U en la misma matriz A y el vector modificado y se sobreescribe en el mismo vector b. No es necesario anular los elementos que están debajo de la diagonal, sólo se tiene en cuenta que se utilizarán los elementos de la parte triangular superior.

Observación 2: una vez obtenida la matriz triangular superior U y el vector y, se puede aplicar el algoritmo 3 para resolver sistemas con matrices triangulares superiores.

Observación 3: si para algún k se tiene que $a_{kk} = 0$, se podría intercambiar la fila k por otra fila j, con $j = k+1, \dots, n$ tal que $a_{jk} \neq 0$. Esto se llama estrategia de pivoteo y no se estudia en este curso.

Observación 4: notar que el ciclo (j) del algoritmo, donde se van recorriendo las columnas de la matriz, comienza desde $k+1$ y no desde k. Esto se debe a que, en cada fila, el coeficiente que multiplica a la fila pivote (multiplicador) se define de manera de anular el coeficiente de la columna k. Entonces no tiene sentido hacer las operaciones en la columna k pues ya se sabe que el resultado va a dar cero.

Por último enunciamos un resultado que da condiciones suficientes para poder aplicar eliminación gaussiana para resolver un sistema lineal.

Teorema 2. *Sean $A_k = A(1:k, 1:k)$ es la submatriz de A formada por las primeras k filas y k columnas de A para todo $k = 1, \dots, n$. Si $\det(A_k) \neq 0$ para $k = 1, \dots, n$, entonces es posible realizar el proceso completo de eliminación gaussiana y, por lo tanto, el sistema $Ax = b$ tiene única solución.*