

Los punto y coma tras las operaciones están situados de manera que se vean las tres matrices del ejemplo y la matriz final. Así la ausencia de punto y coma en la línea 7 muestra toda la matriz a pesar de que solo estamos asignado la segunda fila.

Como contrapunto, veamos cómo proceder con `sagemath` usando comandos compactos. Este *software* usa las convenciones informáticas en vez de las de la gente de la calle: las filas y columnas empiezan con el índice cero y la matriz es una lista de listas (las filas). Así `A[0][:]` es la primera fila de `A` y `A[1][2]` indica el elemento a_{23} .

```

1 # Matriz ampliada de un ejemplo anterior
2 A = matrix(3,4, [0,2,1,3, 1,1,-3,-1, 1,3,-2,2])
3
4 # Intercambia las dos primeras filas (tr. el. 3)
5 A.swap_rows(0,1)
6 print(A)
7
8 # f_3 -> f_3-f_1 (tr. el. 1)
9 A.add_multiple_of_row (2,0,-1)
10 print(A)
11 # f_3 -> f_3-f_2 (tr. el. 1)
12 A.add_multiple_of_row (2,1,-1)
13 print(A)
14
15 # Multiplica la segunda fila por 2 (tr. el. 2)
16 A.rescale_row(1, 2)
17 print(A)

```

Los comandos `print(A)` se han situado de forma que, de nuevo, se vean las matrices mostradas en el ejemplo y el resultado final.

1.2. Solución general

Una vez que conocemos las cuentas que hay que hacer para resolver un sistema de ecuaciones lineales, vamos a estudiar la estructura del conjunto de soluciones. No hay grandes ideas, solo ordenar las conclusiones que sacaríamos tras practicar con unos cuantos ejemplos de eliminación de Gauss. ¿Lo has hecho ya?

En el estudio de esta estructura desempeña un papel importante el número de pivotes (escalones) al aplicar eliminación de Gauss a una matriz A . Llamaremos a este número *rango* de A , y escribiremos $\text{rg}(A)$. Aparte de la queja de “así no es como me lo explicaron a mí”, que más adelante solventaremos mostrando la equivalencia, alguien muy perspicaz podría protestar diciendo que no hay una única forma de aplicar la eliminación de Gauss y por tanto no está claro que a mí y a mi compañero nos salga el mismo rango sin habernos equivocado ninguno. A sabiendas de que esto es solo alimento para filomatemáticos, haremos de ello el primer resultado teórico del curso.

Lema 1.2.1. *Al reducir una matriz a su forma escalonada, los pivotes aparecen siempre en las mismas posiciones independientemente de como se lleve a cabo la eliminación de Gauss. En particular el rango está bien definido.*

Demostración. Sean E y \tilde{E} dos matrices escalonadas obtenidas por eliminación de Gauss sobre $A \in \mathcal{M}_{m \times n}$. Si la situación de algún pivote no coincidiera, digamos que $1 \leq N \leq n$ es tal que los $N - 1$ primeros pivotes tienen la misma localización pero E tiene el N -ésimo pivote en la columna J y \tilde{E} en la columna $J' > J$. Consideremos

el sistema $N \times J$, $(a_{ij})_{i,j=1}^{N,J} \vec{x} = \vec{0}$. Al resolverlo con la forma escalonada $(e_{ij})_{i,j=1}^{N,J}$ se sigue $x_J = 0$, ya que el último elemento e_{NJ} es un pivote, mientras que con $(\tilde{e}_{ij})_{i,j=1}^{N,J}$ la incógnita x_J resulta arbitraria. Esto contradice que en la eliminación de Gauss no se pierden ni ganan soluciones. \square

Destacaremos los *sistemas homogéneos* $A\vec{x} = \vec{0}$, que ya han aparecido en la demostración anterior. Lo que tienen de peculiar es que siempre admiten al menos la solución trivial $\vec{x} = \vec{0}$ y que la última columna de A^+ siempre va a conservar los ceros a lo largo del proceso de eliminación de Gauss, por ello a menudo no se escribe.

La estructura de las soluciones de un sistema homogéneo se describe en el siguiente resultado fundamental en la teoría del curso:

Teorema 1.2.2. *Consideremos el sistema homogéneo $A\vec{x} = \vec{0}$ con $A \in \mathcal{M}_{m \times n}$ y sea $r = \text{rg}(A)$. Si $r = n$ la única solución es $\vec{x} = \vec{0}$ y si $r < n$ todas las soluciones son*

$$\vec{x} = \lambda_1 \vec{v}_1 + \lambda_2 \vec{v}_2 + \cdots + \lambda_{n-r} \vec{v}_{n-r}$$

con λ_j arbitrarios y \vec{v}_j ciertos vectores. Además elecciones distintas de los λ_j dan lugar a soluciones distintas.

La estructura de las soluciones de un sistema general se relaciona con el caso homogéneo.

Teorema 1.2.3. *Un sistema de ecuaciones lineales $A\vec{x} = \vec{b}$ tiene solución si y solo si $\text{rg}(A) = \text{rg}(A^+)$. En ese caso todas las soluciones son $\vec{x} = \vec{x}_0 + \vec{y}$ con \vec{y} la solución general del sistema homogéneo $A\vec{y} = \vec{0}$.*

La siguiente consecuencia se llama en el mundo hispano *teorema de Rouché-Frobenius* aunque el matemático F.G. Frobenius no tenga tanto que ver con este resultado². Por alguna razón, ha sido tradicionalmente material ineludible en los exámenes de matemáticas de acceso a la universidad española.

Corolario 1.2.4. *Dado un sistema de ecuaciones lineales $A\vec{x} = \vec{b}$ con $A \in \mathcal{M}_{m \times n}$ se cumple:*

- 1) *Tiene solución única, y se dice que es compatible determinado, si y solo si $n = \text{rg}(A) = \text{rg}(A^+)$.*
- 2) *Tiene infinitas soluciones, y se dice que es compatible indeterminado, si y solo si $n > \text{rg}(A) = \text{rg}(A^+)$.*
- 3) *No tiene solución, y se dice que es incompatible, si y solo si $\text{rg}(A) \neq \text{rg}(A^+)$.*

Antes de ir a las demostraciones de los teoremas, veamos unos ejemplos.

²Parece que el nombre fue introducido por el matemático hispanoargentino J. Rey Pastor. En el mundo anglosajón se llama *teorema de Rouché-Capelli*, lo que es mucho más lógico porque tanto E. Rouché como A. Capelli lo enunciaron y demostraron, aunque no fueran los primeros en hacerlo.

En primer lugar, hallemos vectores \vec{v}_j que dan la solución en el Teorema 1.2.2 para el sistema homogéneo $A\vec{x} = \vec{0}$ con

$$(1.2) \quad A = \begin{pmatrix} 1 & 1 & -1 & 2 & -3 \\ -3 & -3 & 4 & -6 & 8 \\ 2 & 2 & 1 & 4 & -7 \end{pmatrix}.$$

La forma escalonada se consigue con los siguientes pasos:

$$A \xrightarrow[\substack{f_2 \mapsto f_2 + 3f_1 \\ f_3 \mapsto f_3 - 2f_1}]{\begin{pmatrix} 1 & 1 & -1 & 2 & -3 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 3 & 0 & -1 \end{pmatrix}} \xrightarrow{f_3 \mapsto f_3 - 3f_2} \begin{pmatrix} 1 & 1 & -1 & 2 & -3 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

Las columnas que no contienen pivotes son la segunda y la cuarta y corresponden a parámetros libres, así $x_2 = \lambda_1$, $x_4 = \lambda_2$, mientras que despejando x_1 , x_3 y x_5 utilizando las ecuaciones de abajo a arriba, $x_5 = 0$, $x_3 = 0$, $x_1 = -\lambda_1 - 2\lambda_2$. Con ello

$$\vec{x} = \begin{pmatrix} -\lambda_1 - 2\lambda_2 \\ \lambda_1 \\ 0 \\ \lambda_2 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 & -2 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \lambda_1 \begin{pmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} -2 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

El paso intermedio es solo para practicar con el producto de matrices, está claro que las coordenadas de \vec{v}_1 y \vec{v}_2 vienen dadas por los coeficientes de λ_1 y λ_2 en la expresión inicial para \vec{x} .

Supongamos ahora que deseamos hallar todas las soluciones de $A\vec{x} = \vec{b}$ con A como antes y $\vec{b} = (-1, 2, -3)^t$. Para reducir $A^+ = (A|\vec{b})$ a una matriz escalonada solo tenemos que aplicar los pasos anteriores a la última columna, el resto será igual. Con esas breves cuentas se obtiene:

$$\left(\begin{array}{ccccc|c} 1 & 1 & -1 & 2 & -3 & -1 \\ 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{array} \right).$$

Si no tuviéramos el ejemplo anterior, despejaríamos aquí para hallar la solución general pero podemos aprovechar un poco lo hecho tomando las incógnitas libres x_2 y x_4 como nulas llegando al sistema escalonado más simple $2x_5 = 2$, $x_3 - x_5 = -1$, $x_1 - x_3 - 3x_5 = 1$ que conduce a la solución $\vec{x}_0 = (2, 0, 0, 0, 1)^t$ y según el Teorema 1.2.3 todas las soluciones son $\vec{x} = \vec{x}_0 + \lambda_1 \vec{v}_1 + \lambda_2 \vec{v}_2$ con \vec{v}_1 y \vec{v}_2 como antes.

Como último ejemplo resolveremos con eliminación de Gauss el primer problema del modelo de la opción A de la EVAU 2011, ¡seguramente nadie lo hizo así! El enunciado pide discutir el siguiente sistema en función del parámetro λ y hallar la solución para $\lambda = 1$.

$$\begin{array}{rcl} \lambda x & + \lambda z & = 2, \\ x & + \lambda y & - z = 1, \\ x & + 3y & + z = 2\lambda. \end{array}$$

Para estar seguros de tener un pivote en a_{11} , ordenamos las ecuaciones en orden contrario al presentado ($f_1 \leftrightarrow f_3$). En esta situación conseguimos los ceros de la primera columna con $f_2 \mapsto f_2 - f_1$ y $f_3 \mapsto f_3 - \lambda f_1$

$$\left(\begin{array}{ccc|c} 1 & 3 & 1 & 2\lambda \\ 1 & \lambda & -1 & 1 \\ \lambda & 0 & \lambda & 2 \end{array} \right) \xrightarrow[\substack{f_2 \mapsto f_2 - f_1 \\ f_3 \mapsto f_3 - \lambda f_1}]{\quad} \left(\begin{array}{ccc|c} 1 & 3 & 1 & 2\lambda \\ 0 & \lambda - 3 & -2 & 1 - 2\lambda \\ 0 & -3\lambda & 0 & 2 - 2\lambda^2 \end{array} \right).$$

¿Hemos acabado ya? Solo tenemos una matriz escalonada si $\lambda = 0$ y en ese caso el sistema resulta incompatible por el Corolario 1.2.4 (A con dos pivotes y A^+ con tres). Si $\lambda \neq 0$ podríamos intercambiar f_2 y f_3 (con ello tenemos un pivote bajo el 3) y después aplicar $f_3 \mapsto f_3 + (\lambda - 3)f_2/(3\lambda)$. Da igual lo que salgan exactamente las cuentas, está claro que hay tres pivotes y por tanto solución única: el sistema es compatible determinado para $\lambda \neq 0$. Para la segunda parte del problema sustituyendo $\lambda = 1$ en la última matriz y aplicando $f_3 \mapsto f_3 - 3f_2/2$ tenemos

$$\left(\begin{array}{ccc|c} 1 & 3 & 1 & 2\lambda \\ 0 & -2 & -2 & -1 \\ 0 & -3 & 0 & 0 \end{array} \right) \xrightarrow{f_3 \mapsto f_3 - 3f_2/2} \left(\begin{array}{ccc|c} 1 & 3 & 1 & 2 \\ 0 & -2 & -2 & -1 \\ 0 & 0 & 3 & 3/2 \end{array} \right)$$

que da $z = 1/2$, $y = 0$, $x = 3/2$.

Ahora vamos con las demostraciones de los teoremas. Aunque la primera parezca un poco farragosa, solo pone por escrito conclusiones naturales al aplicar eliminación de Gauss.

Demostración del Teorema 1.2.2. Si $r = n$ entonces todas las columnas contienen pivotes, es decir, ocupan toda la diagonal de la forma escalonada E . La última ecuación es $e_{nn}x_n = 0$ que implica $x_n = 0$, la anterior es $e_{n-1n-1}x_{n-1} + e_{n-1n}x_n = 0$ que implica $x_{n-1} = 0$ y así sucesivamente.

Supongamos ahora $r < n$. Según hemos visto, en la eliminación de Gauss despejamos x_j si en la columna j de la matriz escalonada hay un pivote, el resto de las incógnitas son libres. Digamos que los pivotes están en las columnas numeradas con $\{p_1, \dots, p_r\}$ y el resto de números de columnas son $\{q_1, \dots, q_{n-r}\}$, entonces tomamos $x_{q_l} = \lambda_l$ con λ_l arbitrario $1 \leq l \leq n - r$ y al despejar las incógnitas correspondientes a las columnas pivote se tendrá algo del tipo $x_{p_k} = c_{k1}\lambda_1 + \dots + c_{kn-r}\lambda_{n-r}$ con $1 \leq k \leq r$. En forma matricial, $\vec{x} = F\vec{\lambda}$ donde $f_{p_k j} = c_{kj}$ mientras que $f_{q_l l} = 1$ y $f_{q_l j} = 0$ si $j \neq l$. Si $\vec{v}_1, \dots, \vec{v}_{n-r}$ son las columnas de F esto es lo mismo que $\vec{x} = \lambda_1 \vec{v}_1 + \dots + \lambda_{n-r} \vec{v}_{n-r}$ por la regla para multiplicar matrices. Finalmente, si dos vectores de parámetros $\vec{\lambda} = \vec{\mu}$ y $\vec{\lambda} = \vec{\nu}$ dieran la misma solución \vec{x} entonces como todos los vectores tienen la q_l -ésima coordenada nula excepto \vec{v}_l que la tiene igual a 1, se deduce $\mu_l = \nu_l$ y esto es válido para cualquier $1 \leq l \leq n - r$. \square

Demostración del Teorema 1.2.3. Para la primera parte pensemos en que hallamos una forma escalonada de A^+ , que es una de A extendida con una columna. Si la forma escalonada de A^+ tiene un pivote p más que la de A , la ecuación correspondiente a

la última fila no nula será $0 = p$ lo que indica que no hay solución. Si tuvieran los mismos pivotes, esta última ecuación sería de la forma $e_{ij}x_j + \dots + e_{in}x_n = c$ con $e_{ij} \neq 0$ el pivote y se puede llevar a cabo la sustitución regresiva de la eliminación de Gauss.

Para la segunda parte basta observar que $A\vec{x} = \vec{b}$ equivale a $A\vec{y} = \vec{0}$ con $\vec{y} = \vec{x} - \vec{x}_0$ porque $A\vec{x}_0 = \vec{b}$. \square

Exprimiendo el silicio [opcional]. Sin cargar paquetes especiales, `matlab/octave` no hace manipulación simbólica por ello hay que tener mucha precaución con cantidades como el rango que no varían continuamente. Si pensamos en la matriz 2×2 de rango 1 que tiene como primera fila $(\sqrt{3}, 3)$ y como segunda fila $(1, \sqrt{3})$, cualquier aproximación interna por la cual $\sqrt{3}$ no sea un número que al cuadrado dé tres producirá el valor erróneo 2 para el rango. Las calculadoras y ordenadores tienen trucos para cambiar un dos coma muchos nueves por 3 basándose en que los humanos tratamos la mayor parte del tiempo con números redondos pero eso no siempre funciona. Aquí hay un ejemplo basado en la documentación de `matlab`. Incluye el comando `diag` que permite teclear menos al introducir una matriz diagonal, por otro lado, `rank` es el comando para hallar el rango.

```
1 % Variante la documentación de matlab
2 A = diag([20,21,2021,1e-15])
3 rank(A)
```

El resultado que da `matlab/octave` es 3 cuando está claro que el rango correcto es 4. Hay una variante de `rank` que permite especificar una tolerancia y tener algún control sobre estos errores.

La instrucción `A\b` da una solución particular de $Ax=b$ (como curiosidad, el caso incompatible no produce un error sino que devuelve una solución aproximada). Si el sistema es compatible indeterminado, por supuesto, es raro que coincida con la que nosotros hayamos imaginado. Lo mismo ocurre con los vectores del Teorema 1.2.2, que forman una matriz que se calcula con `null(A)` y que tampoco están únicamente determinada. En `matlab/octave` es posible definir matrices por bloques y por ello la ampliada se obtiene simplemente con `[A b]`.

Las siguientes líneas de código hallan primero los rangos de A y A^+ para el sistema del segundo ejemplo y después calcula \vec{x}_0 y la matriz F formada por los vectores \vec{v}_j .

```
1 % Matriz A
2 A = [1,1,-1,2,-3; -3,-3,4,-6,8; 2,2,1,4,-7];
3 % vector b
4 b = [-1;2;-3];
5 % Matriz A ampliada con b
6 Aplus = [A b];
7 % Calcula los rangos
8 rank(A)
9 rank(Aplus)
10
11 % Solución del sistema Ax=b
12 % Calcula una solución particular
13 x_0 = A\b
14 % Los vectores v_j forman la matriz
15 F = null(A)
```

El rango de A y A^+ se obtiene correctamente como 3. La solución particular y la matriz formada por los vectores \vec{v}_j que ofrece el programa es

$$\mathbf{x}_0 \rightarrow \begin{pmatrix} 3.3333\text{e-}01 \\ 3.3333\text{e-}01 \\ -5.8634\text{e-}16 \\ 6.6667\text{e-}01 \\ 1.0000\text{e+}00 \end{pmatrix}, \quad \mathbf{F} \rightarrow \begin{pmatrix} -5.9545\text{e-}01 & 6.9193\text{e-}01 \\ -5.5886\text{e-}01 & -7.2181\text{e-}01 \\ 1.7347\text{e-}15 & -1.8041\text{e-}15 \\ 5.7716\text{e-}01 & 1.4940\text{e-}02 \\ 3.8303\text{e-}15 & -4.3854\text{e-}15 \end{pmatrix}$$

Aparte de que no coincida con lo que hemos obtenido, vemos el efecto de la computación no simbólica con algunos números infinitesimales. Por ejemplo, en \vec{x}_0 es de suponer que $-5.8634\text{e-}16$ lo podríamos cambiar por cero y el resto por las fracciones de denominador tres a las que se parecen. Concretamente, es fácil comprobar que $\frac{1}{3}(1, 1, 0, 2, 3)^t$ es solución.

Quizá te intrigue que tanto en el ejemplo del rango como aquí se confunda algo del orden de 10^{-15} con cero. En las calculadoras y el *software* numérico existe un $\epsilon > 0$, llamado *épsilon máquina*, tal que 1 y $1 + \delta$ son indistinguibles para $0 < \delta < \epsilon$. Con los 64 bits empleados en doble precisión se suelen dedicar 1 bit al signo, 52 bits a la mantisa y 11 bits al exponente (si no estás al tanto de la terminología, para $-0.8125 = -13 \cdot 2^{-4}$ el signo es $-$, el exponente -4 y la mantisa 13) por tanto $\epsilon = 2^{-52} \approx 2.2 \cdot 10^{-16}$ es un valor habitual del *épsilon máquina*, en particular el de `matlab/octave`.

La computación simbólica de serie en `sagemath` elimina estos problemas. Obviamente hay variaciones en el nombre de los comandos. El análogo del programa anterior es:

```

1 # Matriz A
2 A = matrix(3,5,[1,1,-1,2,-3, -3,-3,4,-6,8, 2,2,1,4,-7])
3 # vector b, tambien b = matrix(3,1,[-1,2,-3])
4 b = vector([-1,2,-3])
5 # Matriz A ampliada con b
6 Aplus = A.augment(b)
7 # Calcula los rangos
8 print(rank(A))
9 print(rank(Aplus))
10
11 # Solución del sistema Ax=b
12 # Calcula una solución particular
13 print(A.solve_right(b))
14 # Los vectores v_j son
15 print(A.right_kernel())

```

Para ilustrar el constructor de vectores, se ha usado para `b` en vez del de matrices, que también funcionaría. La solución particular dada por el programa es $(2, 0, 0, 0, 1)$, la misma que habíamos calculado nosotros (aunque no se ha usado, `A\b` también funciona en `sagemath`), y los vectores son $\vec{v}_1 = (1, 1, 0, -1, 0)$ y $\vec{v}_2 = (0, 2, 0, -1, 0)$ que no coinciden con los nuestros pero no es difícil relacionarlos con ellos.