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

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

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 `-5.8634e-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:

```
# Matriz A
A = matrix(3,5,[1,1,-1,2,-3, -3,-3,4,-6,8, 2,2,1,4,-7])
# vector b, tambien b = matrix(3,1,[-1,2,-3])
b = vector([-1,2,-3])
# Matriz A ampliada con b
Aplus = A.augment(b)
# Calcula los rangos
print(rank(A))
print(rank(Aplus))

# Solución del sistema Ax=b
# Calcula una solución particular
print(A.solve_right(b))
# Los vectores v_j son
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.

1.3. Matriz inversa

Antes de entrar en el tema del título de esta sección, vamos a dejar de refunfuñar por la falta de unicidad de la eliminación de Gauss. Para ello, una vez que la hayamos

aplicado a nuestro gusto utilizaremos la segunda transformación elemental forzando que todos los pivotes sean unos y la primera transformación elemental para que todos los números encima de cada pivote sean ceros. De esta forma los unos que conforman los pivotes son los únicos elementos no nulos en su columna. Se dice que esta forma extendida de la eliminación de Gauss es la *eliminación de Gauss-Jordan*[3] o la *reducción de Gauss-Jordan*. La matriz resultante se dice que es la *forma escalonada reducida* de la matriz de partida.

Veamos un ejemplo. Al final de la primera sección al resolver un sistema habíamos aplicado eliminación de Gauss en la forma $f_1 \leftrightarrow f_2$, $f_3 \mapsto f_3 - f_1$, $f_3 \mapsto f_3 - f_2$ para conseguir

$$A^+ = \left(\begin{array}{ccc|c} 0 & 2 & 1 & 3 \\ 1 & 1 & -3 & -1 \\ 1 & 3 & -2 & 2 \end{array}\right) \qquad \longrightarrow \qquad E = \left(\begin{array}{ccc|c} 1 & 1 & -3 & -1 \\ 0 & 2 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{array}\right).$$

Esta es una matriz escalonada pero no la forma escalonada reducida de A^+. En primer lugar tenemos que dividir la segunda fila por 2 para que e_{22} pase a ser 1 y después cancelar el $e_{12} = 1$ que tiene encima. Los cálculos son:

$$E \underset{f_2 \mapsto f_2/2}{\longrightarrow} \left(\begin{array}{ccc|c} 1 & 1 & -3 & -1 \\ 0 & 1 & 1/2 & 3/2 \\ 0 & 0 & 0 & 0 \end{array}\right) \underset{f_1 \mapsto f_1 - f_2}{\longrightarrow} \left(\begin{array}{ccc|c} 1 & 0 & -7/2 & -5/2 \\ 0 & 1 & 1/2 & 3/2 \\ 0 & 0 & 0 & 0 \end{array}\right).$$

Esta última ya es la forma escalonada reducida.

Seguro que te estás preguntando qué tiene esto de particular. En primer lugar, da las soluciones de un sistema directamente, sin necesidad de sustituir, en segundo lugar permite resolver varios sistemas al tiempo si comparten la misma matriz y finalmente, es útil para el cálculo de la matriz inversa de la que habla el título.

Veamos la primera ventaja sobre el ejemplo anterior. Allí x_3 es un parámetro arbitrario, $x_3 = \lambda$, y la primera fila da directamente $x_1 = (7\lambda - 5)/2$, sin utilizar las otras ecuaciones. De la misma forma la segunda fila da $x_2 = (3 - \lambda)/2$. Nótese que la última columna es una solución particular, correspondiente a $\lambda = 0$, para las variables de las columnas pivote.

En general, en un sistema compatible $A\vec{x} = \vec{b}$, para $i \leq \mathrm{rg}(A)$ la i-ésima fila de la forma escalonada reducida de A^+ nos permite hallar la variable correspondiente a la columna del i-ésimo pivote. Pensándolo con cuidado esto se traduce en que solo hay una forma escalonada reducida aunque haya muchos modos de llegar a ella.

Proposición 1.3.1. *Cada matriz A tiene solo una forma escalonada reducida.*

Demostración. Si $A \in \mathcal{M}_{m\times n}$ y $\mathrm{rg}(A) = n$ todas las columnas tienen pivotes y la única matriz escalonada posible E es la que tiene $e_{ii} = 1$ para $1 \leq i \leq n$ y el resto

[3]Este Jordan es W. Jordan que trabajaba en geodesia, no tiene que ver con el matemático famoso C. Jordan cuyo nombre aparecerá en el último capítulo [1].

de los elementos cero. Si $\mathrm{rg}(A) < n$, consideremos el sistema homogéneo $A\vec{x} = \vec{0}$ y digamos que E y E' son dos formas escalonadas reducidas de A. Por el Lema 1.2.1, si $e_{ij} = 1$ es un pivote también lo es $e'_{ij} = 1$ y las columnas con pivote coinciden. Al usar E y E' para resolver el sistema, la i-ésima fila implica la relación

$$x_j = -\sum_l e_{iq_l} x_{ql} = -\sum_l e'_{iq_l} x_{ql} \quad \text{con } q_l \text{ las columnas sin pivote.}$$

Ahora bien, como x_{q_l} son parámetros arbitrarios, podemos elegirlos todos nulos excepto uno, con lo que se deduce $e_{iq_l} = e'_{iq_l}$ cualquiera que sea $1 \leq l \leq n - r$. □

La segunda ventaja, acerca de la resolución simultánea de varios sistemas, está relacionada con la primera. Como en la eliminación de Gauss-Jordan no hay que sustituir, al aplicarla a una matriz $(A|\vec{b}_1\vec{b}_2 \dots \vec{b}_k)$, ampliada con el segundo miembro de varios sistemas, podemos leer en las k últimas columnas de la forma escalonada reducida soluciones particulares para las variables de las columnas pivote.

Supongamos que extendemos el ejemplo anterior a resolver los sistemas $A\vec{x} = \vec{b}_1$, $A\vec{x} = \vec{b}_2$ y $A\vec{x} = \vec{b}_3$ con

$$A = \begin{pmatrix} 0 & 2 & 1 \\ 1 & 1 & -3 \\ 1 & 3 & -2 \end{pmatrix}, \quad \vec{b}_1 = \begin{pmatrix} 3 \\ -1 \\ 2 \end{pmatrix}, \quad \vec{b}_2 = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}, \quad \text{y} \quad \vec{b}_3 = \begin{pmatrix} 2 \\ 2 \\ 4 \end{pmatrix}.$$

Ya habíamos mencionado que se puede llevar a cabo la reducción de Gauss sobre A con $f_1 \leftrightarrow f_2$, $f_3 \mapsto f_3 - f_1$, $f_3 \mapsto f_3 - f_2$. Haciendo los cálculos (aprovechando lo anterior, basta hacerlos para $\vec{b}_2$ y $\vec{b}_3$), se obtiene

$$(A|\vec{b}_1\vec{b}_2\vec{b}_3) \qquad \longrightarrow \qquad E = \left(\begin{array}{ccc|ccc} 1 & 1 & -3 & -1 & 0 & 2 \\ 0 & 2 & 1 & 3 & -2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right).$$

Siguiendo con la reducción de Gauss-Jordan, como antes, con $f_2 \mapsto f_2/2$ y $f_1 \mapsto f_1 - f_2$ se llega a la forma escalonada reducida. De nuevo, la única diferencia es que tenemos dos columnas nuevas y basta trabajar sobre ellas.

$$E \underset{f_2 \mapsto f_2/2}{\longrightarrow} \left(\begin{array}{ccc|ccc} 1 & 1 & -3 & -1 & 0 & 2 \\ 0 & 1 & 1/2 & 3/2 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \underset{f_1 \mapsto f_1 - f_2}{\longrightarrow} \left(\begin{array}{ccc|ccc} 1 & 0 & -7/2 & -5/2 & 1 & 1 \\ 0 & 1 & 1/2 & 3/2 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right).$$

Las tras últimas columnas dan soluciones particulares para x_1 y x_2 (correspondientes a los pivotes) de los sistemas con $x_3 = 0$. Tomando $x_3 = \lambda$ se deduce que las soluciones generales de los tres sistemas son, respectivamente,

$$\begin{cases} \vec{x} = (-5/2, 3/2, 0)^t + \lambda\vec{v}, \\ \vec{x} = (1, -1, 0)^t + \lambda\vec{v}, \\ \vec{x} = (1, 1, 0)^t + \lambda\vec{v}, \end{cases} \qquad \text{con} \quad \vec{v} = \begin{pmatrix} 7/2 \\ -1/2 \\ 1 \end{pmatrix}.$$

Pasamos por fin al tema principal de esta sección. A estas alturas todos sabemos que solo se calculan inversas de matrices cuadradas pero es el momento de echarle un poco de literatura y entender por qué. Si no eres *filomatemático*, todo lo que necesitas saber está en el enunciado de la Proposición 1.3.2 y tienes permiso para saltarte lo que media hasta allí y su demostración si te resulta aburrido.

El inverso de un número x es otro número y tal que $xy = 1$, como la multiplicación es *conmutativa*[4], no importa dónde poner la y pero en el caso de matrices el producto puede depender del orden e incluso solo tener sentido en uno de los órdenes posibles. Como unas veces necesitamos un orden y otras veces otro, lo lógico es decir que una matriz $A \in \mathcal{M}_{m\times n}$ es *invertible* si existen matrices B y C, llamadas provisionalmente *inversa por la izquierda* e *inversa por la derecha* tales que BA y AC son matrices identidad. Basta pensar un momento en las dimensiones de las matrices para deducir que esto requiere $B, C \in \mathcal{M}_{n\times m}$. A pesar de que la definición no da ningún indicio de ello, se prueba que la única posibilidad es $B = C$. La demostración es simple pero ingeniosa, difícil que a uno se le ocurra. Se reduce a la siguiente línea[5]

$$B = BI_n = B(AC) = (BA)C = I_m C = C. \tag{1.3}$$

Con esto, la distinción de inversa por la derecha o por la izquierda se revela innecesaria. En definitiva, tenemos la definición equivalente de que A es invertible si existe una matriz C tal que $CA = AC = I$. Todavía no sabemos que tal matriz solo pueda existir para matrices cuadradas, ni si para una A puede haber varias posibles C. En fin, somos un mar de dudas, que quedan aclaradas en el siguiente resultado que da una caracterización sencilla de las matrices invertibles.

Proposición 1.3.2. *Una matriz A es invertible si y solo si $A \in \mathcal{M}_n$ y $\operatorname{rg}(A) = n$. En ese caso, existe una sola matriz que llamaremos* matriz inversa *de A y denotaremos con A^{-1}, tal que $A^{-1}A = AA^{-1} = I$.*

Demostración. Si $A \in \mathcal{M}_n$ y $\operatorname{rg}(A) = n$, el Corolario 1.2.4 asegura que los sistemas que resultan al comparar columnas en la ecuación matricial $AX = I_n$ con $X \in \mathcal{M}_n$ son compatibles determinados. Con ello existe una inversa a la derecha C, esto es, $AC = I$. Necesariamente $\operatorname{rg}(C) = n$ porque en otro caso, de nuevo por el Corolario 1.2.4, existiría $\vec{x} \neq \vec{0}$ con $C\vec{x} = \vec{0}$, lo que contradiría $AC\vec{x} = I\vec{x}$. Entonces C también tiene inversa a la derecha $CC' = I$ y con un razonamiento que recuerda a (1.3),

$$CA = CA(CC') = C(AC)C' = CC' = I.$$

[4]Recuerda, una operación $*$ entre los elementos de un conjunto se dice que tiene la propiedad conmutativa si $A * B = B * A$. Para números reales o en general complejos, la suma y el producto son conmutativos pero para matrices solo lo es la suma (de matrices de las mismas dimensiones).

[5]Aquí se está usando la propiedad *asociativa* del producto de matrices, es decir, que $B(AC) = (BA)C$. Esto nos lo creemos todos sin dudarlo. Si nos dejaran sin comer hasta que no diéramos una prueba matemática, romperíamos el ayuno utilizando que el elemento ij en estos productos es $\sum_k b_{ik} \sum_l a_{kl}c_{lj}$ y $\sum_k b_{ik}a_{kl} \sum_l c_{lj}$, por tanto la asociativa del producto de matrices se sigue de la asociativa de los números reales o complejos que ya nadie pedirá que demostremos [10].

Por tanto $CA = AC = I$ y A es invertible. Además como los sistemas eran compatibles determinados, esta C es única.

Falta por probar que si A es invertible entonces es cuadrada de rango máximo. Si $A \in \mathcal{M}_{m\times n}$ y $\operatorname{rg}(A) < n$ entonces, como antes, $A\vec{x} = \vec{0}$ tiene una solución $\vec{x} \neq \vec{0}$ por el Corolario 1.2.4 y esto contradice $CA\vec{x} = I\vec{x}$. De la misma forma, $C^tA^t\vec{x} = I\vec{x}$, que se deduce trasponiendo $CA = I$, lleva a contradicción si $\operatorname{rg}(A^t) < m$. Entonces si A es invertible $\operatorname{rg}(A) = n$ y $\operatorname{rg}(A^t) = m$. Obviamente ambos rangos son como mucho el mínimo de m y n (a lo más un pivote por fila o columna), por tanto $n = m$. □

Para practicar, calculemos la inversa de la siguiente matriz utilizando la eliminación de Gauss-Jordan:

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 5 & 1 \\ -1 & 1 & 11 \end{pmatrix}.$$

Para ello debemos aplicar la reducción de Gauss a $(A|I_3)$,

$$(A|I_3) \underset{\substack{f_2\mapsto f_2-2f_1\\ f_3\mapsto f_3+f_1}}{\longrightarrow} \left(\begin{array}{ccc|ccc} 1 & 2 & -1 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & 1 & 0 \\ 0 & 3 & 10 & 1 & 0 & 1 \end{array}\right) \underset{f_3\mapsto f_3-3f_2}{\longrightarrow} \left(\begin{array}{ccc|ccc} 1 & 2 & -1 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 7 & -3 & 1 \end{array}\right).$$

Los pivotes ya son unos y solo resta crear los ceros encima de ellos:

$$\underset{f_1\mapsto f_1-2f_2}{\longrightarrow} \left(\begin{array}{ccc|ccc} 1 & 0 & -7 & 5 & -2 & 0 \\ 0 & 1 & 3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 7 & -3 & 1 \end{array}\right) \underset{\substack{f_1\mapsto f_1+7f_3\\ f_2\mapsto f_2-3f_3}}{\longrightarrow} \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 54 & -23 & 7 \\ 0 & 1 & 0 & -23 & 10 & -3 \\ 0 & 0 & 1 & 7 & -3 & 1 \end{array}\right).$$

En definitiva,

$$A^{-1} = \begin{pmatrix} 54 & -23 & 7 \\ -23 & 10 & -3 \\ 7 & -3 & 1 \end{pmatrix}$$

Quizá a alguien no se le haya pasado por alto que la matriz de partida es invariante por trasposición $A = A^t$, es simétrica, y la inversa también tiene esa propiedad. ¿Es una casualidad? No, porque que una matriz A sea invertible es equivalente a A^t sea invertible y en ese caso

$$\left(A^{-1}\right)^t = \left(A^t\right)^{-1}. \tag{1.4}$$

Esto se sigue notando que al trasponer $A^{-1}A = I$ y $AA^{-1} = I$ se obtienen $A^t\left(A^{-1}\right)^t = I$ y $\left(A^{-1}\right)^tA^t = I$ y esto dice que la inversa de A^t es $\left(A^{-1}\right)^t$. Con respecto a los productos, la inversa se comporta como la trasposición invirtiendo el orden. Concretamente, es fácil probar que si $A_1, \dots A_k$ son matrices invertibles entonces $A_1A_2\cdots A_k$ también lo es y

$$(A_1A_2\cdots A_k)^{-1} = A_k^{-1}A_{k-1}^{-1}\cdots A_1^{-1}.$$

Seguro que tras el último ejemplo del cálculo de una inversa con reducción de Gauss-Jordan más de uno está murmurando "eso lo sé hacer yo más rápido con determinantes". No lo niego pero estoy seguro de que en un ejemplo típico 5×5, Gauss-Jordan te ganaría por goleada. En breve, cuando la dimensión es grande los determinantes se vuelven impracticables. Volveremos sobre ello en un capítulo posterior.

Lo que sí es cierto es que si queremos escribir la fórmula general que da la inversa de una matriz 2×2 y si hemos memorizado la fórmula con determinantes no hay que hacer prácticamente nada. Para el que tenga flaca la memoria y para el que quiera practicar, veamos cómo se haría con Gauss-Jordan.

Partimos de

$$(A|I) = \left(\begin{array}{cc|cc} a & b & 1 & 0 \\ c & d & 0 & 1 \end{array}\right).$$

Supongamos primero $a \neq 0$ de modo que a_{11} es un pivote y llevamos a cabo la eliminación de Gauss con

$$(A|I) = \underset{f_1 \mapsto f_1/a}{\longrightarrow} \left(\begin{array}{cc|cc} 1 & b/a & 1/a & 0 \\ c & d & 0 & 1 \end{array}\right) \underset{f_2 \mapsto f_2 - cf_1}{\longrightarrow} \left(\begin{array}{cc|cc} 1 & b/a & 1/a & 0 \\ 0 & \Delta/a & -c/a & 1 \end{array}\right)$$

donde se ha abreviado $\Delta = ad - bc$. Necesitamos $\Delta \neq 0$ para que $\mathrm{rg}(A) = 2$ y por tanto que sea invertible. Terminamos la reducción de Gauss-Jordan con

$$\underset{f_2 \mapsto af_2/\Delta}{\longrightarrow} \left(\begin{array}{cc|cc} 1 & b/a & 1/a & 0 \\ 0 & 1 & -c/\Delta & a/\Delta \end{array}\right) \underset{f_1 \mapsto f_1 - bf_2/a}{\longrightarrow} \left(\begin{array}{cc|cc} 1 & 0 & d/\Delta & -b/\Delta \\ 0 & 1 & -c/\Delta & a/\Delta \end{array}\right)$$

Entonces cuando $a \neq 0$ la matriz A es invertible si y solo si $\Delta \neq 0$ y se tiene

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

¿Y qué ocurre si $a = 0$? En ese caso si A es invertible $c \neq 0$ porque $\mathrm{rg}(A) = 2$ y se podría repetir el razonamiento intercambiando las dos primeras filas en el primer paso de la eliminación de Gauss. Más profundo y más rápido es notar que hemos probado que

$$\frac{1}{\Delta} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} A = A \frac{1}{\Delta} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = I_2 \qquad \text{para } a \neq 0.$$

Nada nos impide tomar límites $a \to 0$ siempre que $\Delta \neq 0$ y la continuidad de la expresión prueba que también es válida para $a = 0$. Un poco más pedestre es ver que a mano que esta identidad es cierta para $a = 0$ operando los productos.

La fórmula por supuesto funciona igual con reales y complejos. Un ejemplo es:

$$A = \frac{1}{3} \begin{pmatrix} 2i & 1 + 2i \\ -1 + 2i & -2i \end{pmatrix} \quad \Longrightarrow \quad A^{-1} = \frac{1}{3} \begin{pmatrix} -2i & -1 - 2i \\ 1 - 2i & 2i \end{pmatrix}.$$

Por razones que es fácil que justifiques, es correcto invertir 1/3 a 3 y multiplicar por la inversa del resto. Una particularidad de este ejemplo es que $A^{-1} = A^{\dagger}$. En un capítulo posterior daremos un nombre a esta situación. Por ahora, para captar tu atención solo diré que el caso 2×2, como este, corresponde a puertas lógicas actuando sobre un solo *qubit* en ordenadores cuánticos.

Exprimiendo el silicio [opcional]**.** En `matlab/octave` el comando `rref` (abreviatura de *reduced row echelon form*) produce la forma escalonada reducida de una matriz. Ya sabemos que la inversa se podría obtener con ello, ampliando con la matriz identidad pero hay un comando directo `inv` y también reconoce `A^-1`.

El siguiente código comprueba la forma escalonada reducida usada para el triple sistema en un ejemplo anterior y el cálculo de la inversa 3×3 que habíamos hecho, tanto a través de `rref` como con el comando directo `inv`.

```
1  % Matriz y vectores
2  A = [0,2,1; 1, 1, -3; 1,3,-2 ];
3  b1 = [3;-1;2]
4  b2 = [-2;0;-2]
5  b3 = [2;2;4]
6  % Forma escalonada reducida
7  rref( [A, b1, b2, b3]  )
8
9  % Matriz
10 A = [1,2,-1; 2, 5, 1; -1,1,11 ];
11 % Inversa con Gauss-Jordan sobre [A I]
12 E = rref( [A, eye(3)] );
13 % Esto muestra las últimas tres columnas
14 E(:,4:6)
15
16 % Inversa con un comando directo (también A^-1)
17 inv(A)
```

Si se usan dos variables para capturar la salida de `rref`, la segunda contendrá las columnas pivote. Esto es, si cambiamos la línea 7 por `[X, p] = rref( [A, b1, b2, b3] )` entonces `X` será la forma escalonada y `p` será la matriz `[1 2]` que indica que los pivotes están en las dos primeras columnas.

Se puede demostrar que el número de operaciones para calcular la inversa de una matriz $n \times n$ con eliminación de Gauss-Jordan es similar a $2n^3/3$ en el sentido de que el cociente entre ambas cantidades tiende a 1 cuando $n \to \infty$ [11, §2.1]. Despreciando el tiempo de acceso a la memoria, esto implica que a la larga el tiempo de computación se multiplica por 8 cuando el tamaño de la matriz se duplica. Para comprobarlo, he ejecutado `rref` para matrices de dimensiones muy grandes con elementos aleatorios en $[0, 1]$.

En mi portátil (viejo y lento) los tiempos presentaban cierta variabilidad por ello consideré el siguiente programa que incluye un bucle `for` que repite el cálculo 20 veces y toma la mediana, `median`, de los tiempos de computación. Esta es menos sensible que la media a valores extremos. Los comandos `tic` y `toc` permiten medir intervalos de tiempo real (no de la CPU).

```
1  % Dimensión
2  N = 500
3  time_val = zeros(1,20);
4  for k = 1:20
5          % Random matrix N x N
```

```
6           A = [rand(N) eye(N)];
7           t0 = tic;
8           E = rref(A);
9           time_val(k) = toc(t0);
10  end
11  median(time_val)
```

Al ejecutarlo tres veces el valor medio fue 2.80 mientras que al cambiar en la segunda línea 500 por 1000, pasó a ser 21.96. El cociente es alrededor de 7.84 que se parece lo suficiente a 8 como para no extrañarse del resultado.

El análogo para `sagemath` del primer programa, sería:

```
1  # Matriz A
2  A = matrix(3,3,[0,2,1, 1, 1, -3, 1,3,-2])
3  # Ampliada con los tres vectores
4  A = A.augment(vector([3,-1,2]))
5  A = A.augment(vector([-2,0,-2]))
6  A = A.augment(vector([2,2,4]))
7  # Forma escalonada reducida
8  print( A.echelon_form() )
9
10 # Matriz
11 A = matrix(3,3, [1,2,-1, 2, 5, 1, -1,1,11 ])
12 B = (A.augment( identity_matrix(3) )).echelon_form()
13 print( B[:,3:] )
14
15 # Inversa con un comando directo (también A^-1)
16 print( A.inverse() )
```

Para los no iniciados, `B[:,3:]` es lo mismo que `B[0:3,3:6]`.