Apunte Único: Álgebra Lineal Computacional - Práctica 3

Por alumnos de ALC
Facultad de Ciencias Exactas y Naturales
UBA

última actualización 17/04/25 @ 02:42

Choose your destiny:

(dobleclick en el ejercicio para saltar)

Esta Guía 3 que tenés se actualizó por última vez:
17/04/25 @ 02:42

Escaneá el QR para bajarte (quizás) una versión más nueva:

Guía 3

El resto de las guías repo en github para descargar las guías con los últimos updates.

Si querés mandar un ejercicio o avisar de algún error, lo más fácil es por Telegram .

Notas teóricas:

- ... hay que hacerlo!

  Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al.

Ejercicios de la guía:

Ejercicio 1. Sean A y $B \in K^{n\times n}$. Probar que:

(a) Si A y B son triangulares superiores, AB es triangular superior.

(b) Si A y B son diagonales, AB es diagonal.

(c) Si A es estrictamente triangular superior (es decir, $a_{ij} = 0$ si $i \geq j$), $A^n = 0$.

(a) Una matriz A va a ser triangular superior si todos los número debajo de la diagonal son cero:

$$A_{ij} \overset{\star^1}{=} \begin{cases} 0 & \text{si} \quad i > j \\ a_{ij} & \text{si} \quad i \leq j \end{cases} \qquad \begin{pmatrix} & a_{ij} \\ 0 & \end{pmatrix}$$

Los a_{ij} no tienen que ser necesariamente distinto a cero. Ahora multiplico dos matrices triangulares superiores:

$$[A \cdot B]_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \cdots + a_{in} \cdot b_{nj} \overset{\star^1}{=} \begin{cases} \sum\limits_{i \leq k \leq j} a_{ik} \cdot b_{kj} & \star^2 \\ 0 & \text{en otro caso} \end{cases}$$

Se cumple $\star^2$ son los que tiene las *filas menores o iguales columnas* y *filas menores o iguales columnas*, si no son cero. Básicamente la definición de matriz triangular superior.

(b) Esta es un poco más fácil. Una matriz es diagonal si:

$$A_{ij} \overset{\star^1}{=} \begin{cases} 0 & \text{si} \quad i \neq j \\ a_{ij} & \text{si} \quad i = j \end{cases} \qquad \begin{pmatrix} & 0 \\ 0 & \end{pmatrix}$$

Nuevamente, los elementos diagnonales no tienen que ser necesariamente distintos de cero. Ahora multiplico dos matrices diagonales:

$$[A \cdot B]_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \cdots + a_{in} \cdot b_{nj} \overset{\star^1}{=} \begin{cases} a_{ii} \cdot b_{ii} & \star^2 \\ 0 & \text{en otro caso} \end{cases}$$

En la sumatoria las columnas de los elementos de A coinciden con las filas de los elementos de B, pero solo cuando estemos multiplicando la fila i con la columna i es que ambos elementos podrían ser no nulos.

(c) Una matriz A va a ser triangular superior estricta si todos los número debajo y de la diagonal son cero:

$$A_{ij} \overset{\star^1}{=} \begin{cases} 0 & \text{si} \quad i \geq j \\ a_{ij} & \text{si} \quad i < j \end{cases} \qquad \begin{pmatrix} 0 & a_{ij} \\ 0 & 0 \end{pmatrix}$$

Meto inducción porque es un viaje. Quiero probar que:

$$p(n) : A \in K^{n\times n} \text{ estrictamente triangular superior } \implies A^n = 0.$$

Caso base:

$$p(2) : A \in K^{2\times 2} \text{ estrictamente triangular superior } \implies A^2 = 0$$

Cálculo directo

$$A \cdot A \begin{pmatrix} 0 & a_{12} \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & a_{12} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

por lo tanto $p(2)$ es verdadera.

Paso inductivo: Voy a asumir que para algún $k \in \mathbb{Z}$

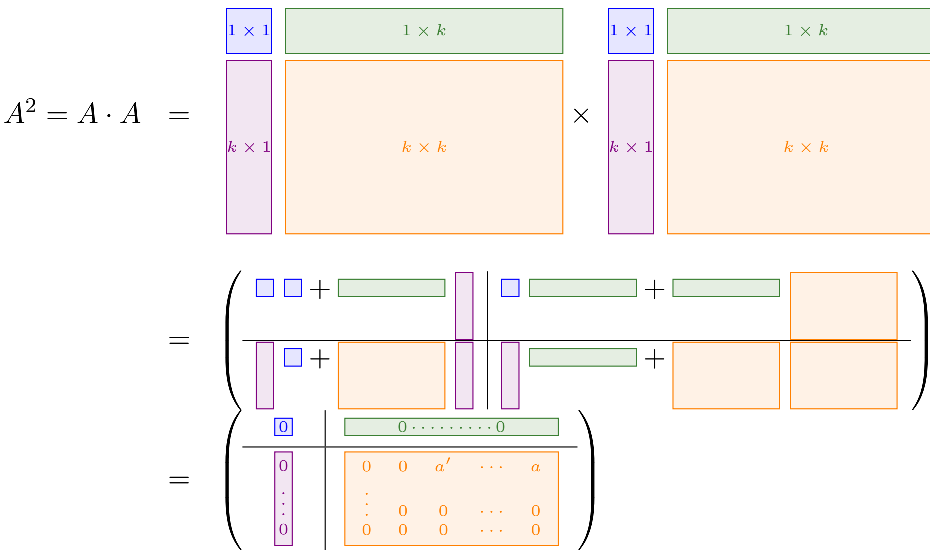
$$p(k) : \underbrace{A \in K^{k\times k} \text{ estrictamente triangular superior } \Longrightarrow A^k = 0}_{\text{hipótesis inductiva}}$$

es verdadera. Por lo tanto ahora quiero probar que:

$$p(k+1) : A \in K^{(k+1)\times(k+1)} \text{ estrictamente triangular superior } \Longrightarrow A^{k+1} = 0$$

Para probar esta tremenda garompa, voy a usar el producto en bloques. Tengo una matriz $A \in K^{(k+1)\times(k+1)}$ estrictamente triangular superior y la parto en bloques así:

$$A = \left(\begin{array}{c|ccc} \boxed{0} & a_{12} & \cdots & a_{1k+1} \\ \hline 0 & 0 & \ddots & \vdots \\ \vdots & 0 & \ddots & a_{kk+1} \\ 0 & 0 & 0 & 0 \end{array}\right)$$



Oka, esto se fue al carajo. Pero está demostrado. La última matriz, es el resultado de A^2. Tiene todos ceros excepto en el bloque naranja, que está en $K^{k\times k}$ y es el producto de hacer el bloque naranja por el bloque naranja dado que el bloque violeta por el bloque verde dio 0. Por lo tanto el bloque naranja es la hipótesis inductiva!!! Multiplicar $k+1$ veces A por si misma dará 0, porque el producto, será el (bloque naranja)^2 con cada vez más ceros.

Dado que $p(2), p(k)$ y $p(k+1)$ resultaron verdaderas, por el principio de inducción en $p(n)$ también será verdadera $\forall n \in \mathbb{N}$

El caso con $n = 1$ es trivial, dejame en paz.

`Dale las gracias y un poco de amor ♥ a los que contribuyeron! Gracias por tu aporte:`
`♉ naD GarRaz`

Ejercicio 2. Sea $A = \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 2 & -1 & 0 & -2 \\ -3 & 3 & 0 & -1 \end{pmatrix} \in \mathbb{R}\, 4 \times 4$

(a) Escalonar la matriz A multiplicándola a izquierda por matrices elementales $T^{ij}(a), a \in \mathbb{R}, 1 \leq i, j \leq 4$, con $i \neq j$.

Recordar que $T^{ij}(a) \in K^{n \times n}$ *se define como:*

$$T^{ij}(a) = I_n + aE^{ij}, \quad 1 \leq i, j \leq n, \quad i \neq j, a \in K,$$

siendo E^{ij} las matrices canónicas de $K^{n \times n}$

(b) Hallar la descomposición LU de A.

(c) Usando la descomposición del ítem anterior resolver el sistema $Ax = b$, para $b = \begin{pmatrix} 1 \\ -7 \\ -5 \\ 1 \end{pmatrix}$.

... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 3. Escribir funciones de `Python` que calculen la solución de un sistema:

(a) $Ly = b$, siendo L triangular inferior.

(b) $Ux = y$, siendo U triangular inferior.

... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 4. Escribir funciones de `Python` que realicen las siguientes tareas:

(a) Calcular la descomposición LU de una matriz dada A, asumiendo que no es necesario realizar pivoteos.

(b) Resolver un sistema $Ax = b$, utilizando la función del ítem anterior y las del ejercicio 3. Aplicar esta función para resolver el ítem (c) del ejercicio 2.

(a) El siguiente *snippet* es en gran parte código para generar la matriz y después del cálculo de la triangulación formar las matrices L y U.

Si hacés un `copy paste` de este código debería funcionar lo más bien

```
"""
Eliminacion Gausianna
"""

import numpy as np


def elim_gaussiana(A):
```

```
    m = A.shape[0]
    n = A.shape[1]
    Ac = A.copy()

    if m != n:
        print("Matriz no cuadrada")
        return


    for i in range(0, n - 1):
        divisor = Ac[i][i]
        for j in range(i, n - 1):
            coef = Ac[j + 1][i] / divisor
            Ac[j + 1][i:] = np.subtract(Ac[j + 1][i:], coef * Ac[i][i:])
            Ac[j + 1][i] = coef


    L = np.tril(Ac, -1) + np.eye(A.shape[0])
    U = np.triu(Ac)

    return L, U

def main():
    n = 7
    B = np.eye(n) - np.tril(np.ones((n, n)), -1)
    B[:n, n - 1] = 1
    print(f"Matriz B = \n{B}\n")

    L, U = elim_gaussiana(B)

    print(f"Matriz L = \n{L}\n")
    print(f"Matriz U = \n{U}\n")
    print("B = LU? ", "Sí!" if np.allclose(np.linalg.norm(B - L @ U, 1), 0)
   else "No!")
    print("Norma infinito de U: ", np.max(np.sum(np.abs(U), axis=1)))


if __name__ == "__main__":
    main()
```

Ejercicio 5. ... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 6. ... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 7. ... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 8.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 9.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 10.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 11.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 12.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 13.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 14.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 15.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 16.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 17.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 18.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 19.
... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 20.

... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 21.

... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 22.

... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 23.

... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicio 24.

... hay que hacerlo!

Si querés mandá la solución → al grupo de Telegram, o mejor aún si querés subirlo en LaTeX→ una *pull request* al .

Ejercicios de parciales:

1.