

Apunte Único: Álgebra Lineal Computacional - Práctica 3

Por alumnos de ALC
Facultad de Ciencias Exactas y Naturales
UBA

última actualización 06/05/25 @ 13:15

Choose your destiny:

(doubleclick en el ejercicio para saltar)

☉ [Notas teóricas](#)

☉ Ejercicios de la guía:

1.	4.	7.	10.	13.	16.	19.	22.
2.	5.	8.	11.	14.	17.	20.	23.
3.	6.	9.	12.	15.	18.	21.	24.

☉ Ejercicios de Parciales

 [1.](#)

Esta Guía 3 que tenés se actualizó por última vez:

06/05/25 @ 13:15

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:✿ *Matriz definida positiva:*

Sea $A \in \mathbb{R}^{n \times n}$ definida positiva:

$$\forall x \neq 0 \quad x^t A x > 0 \text{ con } x \in \mathbb{R}^n$$

Algunas propiedades de las matrices definidas positivas:

- $A \in \mathbb{R}^{n \times n} \implies \exists A^{-1}$.
- Los elementos diagonales son positivos.
- Las submatrices principales también son matrices definidas positivas

✿ *Descomposición de Cholesky:*

La descomposición de Cholesky para una matriz A simétrica y definida positiva:

$$A = LL^t$$

con L triangular inferior. A partir de la descomposición:

$$A = \overbrace{\tilde{L}U}^{\text{la de siempre}} \xrightarrow{\text{✿}} A = \tilde{L} \overbrace{D\tilde{L}^t} \downarrow \begin{array}{l} \text{matriz diagonal} \\ \text{con los elementos} \\ \text{diagonales de } U \end{array}$$



$A = \tilde{L}D\tilde{L}^t$, es definida positiva si y solo si D lo es. Como D es diagonal, solo es cuestión de ver que $[D]_{ii} > 0$.



Finalmente:

$$A = \tilde{L}D\tilde{L}^t \Leftrightarrow A = \tilde{L}\sqrt{D}\sqrt{D}\tilde{L}^t \Leftrightarrow A = LL^t$$

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:

$$A_{ij} \stackrel{\star^1}{=} \begin{cases} 0 & \text{si } i > j \\ a_{ij} & \text{si } i \leq j \end{cases} \quad \left(\begin{array}{c} \triangle \\ 0 \end{array} \right)$$

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

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

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:

$$A_{ij} \stackrel{\star^1}{=} \begin{cases} 0 & \text{si } i \neq j \\ a_{ij} & \text{si } i = j \end{cases} \quad \left(\begin{array}{c} 0 \\ \diagdown \\ 0 \end{array} \right)$$

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

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

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:

$$A_{ij} \stackrel{\star^1}{=} \begin{cases} 0 & \text{si } i \geq j \\ a_{ij} & \text{si } i < j \end{cases} \quad \left(\begin{array}{c} 0 \quad a_{ij} \\ 0 \quad 0 \end{array} \right)$$

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

$$A \cdot A \left(\begin{array}{cc} 0 & a_{12} \\ 0 & 0 \end{array} \right) \cdot \left(\begin{array}{cc} 0 & a_{12} \\ 0 & 0 \end{array} \right) = \left(\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \right)$$

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

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

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

es verdadera. Por lo tanto ahora quiero probar que:

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

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:

$$A = \begin{pmatrix} \boxed{0} & \boxed{a_{12} \quad \cdots \quad a_{1k+1}} \\ \boxed{0} & \boxed{\begin{matrix} 0 & \ddots & \vdots \\ 0 & \ddots & a_{kk+1} \\ 0 & 0 & 0 \end{matrix}} \end{pmatrix}$$

$$\begin{aligned} A^2 = A \cdot A &= \begin{pmatrix} \boxed{1 \times 1} & \boxed{1 \times k} \\ \boxed{k \times 1} & \boxed{k \times k} \end{pmatrix} \times \begin{pmatrix} \boxed{1 \times 1} & \boxed{1 \times k} \\ \boxed{k \times 1} & \boxed{k \times k} \end{pmatrix} \\ &= \begin{pmatrix} \boxed{} \boxed{} + \boxed{} & \boxed{} \boxed{} + \boxed{} & \boxed{} \boxed{} + \boxed{} \\ \boxed{} \boxed{} + \boxed{} & \boxed{} \boxed{} + \boxed{} & \boxed{} \boxed{} + \boxed{} \end{pmatrix} \\ &= \begin{pmatrix} \boxed{0} & \boxed{0 \cdots \cdots 0} \\ \boxed{0} & \boxed{\begin{matrix} 0 & 0 & a' & \cdots & a \\ \vdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \end{matrix}} \end{pmatrix} \end{aligned}$$

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 🍷

🔗 ¿Errores? **Avisá acá** así se corrige y ganamos todos.

Compilado: 06/05/25 @ 13:15 . Chequeá si hay una **versión nueva** → **acá**.

[Ir a índice ↑](#)

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 $b = \begin{pmatrix} 1 \\ -7 \\ -5 \\ 1 \end{pmatrix}$.

- (a) Hacer una operación entre *filas* es multiplicar por esas matrices T^{ij} , pero dado que el *me da tremenda pajómetro* explota, escribo las T^{ij} para la primera *columna* de ceros no más.

$$\begin{aligned} A = \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 2 & -1 & 0 & -2 \\ -3 & 3 & 0 & -1 \end{pmatrix} &\rightsquigarrow \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{T^{31}(-\frac{2}{1})=I_4+(-\frac{2}{1})E^{31}} \cdot \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 2 & -1 & 0 & -2 \\ -3 & 3 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 0 & 1 & 0 & -4 \\ -3 & 3 & 0 & -1 \end{pmatrix} \\ &\rightsquigarrow \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{pmatrix}}_{T^{41}(\frac{3}{1})=I_4+(\frac{3}{1})E^{41}} \cdot \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 0 & 1 & 0 & -2 \\ -3 & 3 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 0 & 2 \end{pmatrix} \star^1 \end{aligned}$$

Ahí entonces están las T^{ij} para hacer ceros en la primera *columna*. Y como la *matemagia* en esta materia parece no tener parangón, cuando multiplicás esas matrices T^{ij} da lo mismo que sumar los elementos fuera de la diagonal componente a componente:

$$T^{31} \cdot T^{41} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{pmatrix}$$

Es gracias a ese resultado que en el próximo paso podría armar solo una matriz con la info para triangular toda la *segunda columna*. solo un producto matricial. Continúo la triangulación de \star^1 :

$$\begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 0 & 2 \end{pmatrix} \rightsquigarrow \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{T^{32}(-1)=I_4+(-\frac{1}{1})E^{32}} \cdot \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & -4 & -4 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

Por lo tanto para que la matriz A quede triangulada superiormente:

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{T^{32} \cdot T^{41} \cdot T^{31}} \cdot \underbrace{\begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 2 & -1 & 0 & -2 \\ -3 & 3 & 0 & -1 \end{pmatrix}}_A = \underbrace{\begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & -4 & -4 \\ 0 & 0 & 0 & 2 \end{pmatrix}}_U$$

$$T^{32} \cdot T^{41} \cdot T^{31} \cdot A = U$$

- (b) La U está una vez triangulada la matriz A . Encontrar la L sale con las matrices que multiplicamos para obtener la matriz triangulada:

$$L^{-1} \cdot A = U \xrightarrow[L]{\times \text{izquierda}} L \cdot L^{-1} \cdot A = L \cdot U \Leftrightarrow A = L \cdot U$$

El producto de las matrices elementales me forma la inversa de $L : L^{-1}$. Por suerte encontrar la inversa de $(L^{-1})^{-1}$ es sencillo:

$$L^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -2 & -1 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{pmatrix} \Rightarrow L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ +2 & +1 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{pmatrix}$$

Solo hay que cambiarle los signos a los elementos que estás por debajo de la diagonal.

$$A = LU \Leftrightarrow \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 2 & -1 & 0 & -2 \\ -3 & 3 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & -4 & -4 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

(c)

$$A \cdot x = b \xLeftrightarrow{A=LU} LU \cdot x = b \Leftrightarrow L \underbrace{(U \cdot x)}_y = b \Leftrightarrow \begin{cases} L \cdot y = b & \star^1 \text{ Arranco por acá.} \\ U \cdot x = y & \star^2 \text{ Sigo por acá una vez encontrado } y. \end{cases}$$

Entonces resuelvo primero \star^1 :

$$Ly = b \xrightarrow{\text{armo sistema}} \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & -7 \\ 2 & 1 & 1 & 0 & -5 \\ -3 & 0 & 0 & 1 & 1 \end{array} \right) \Rightarrow y \stackrel{\star^3}{=} \begin{pmatrix} 1 \\ -7 \\ 0 \\ 4 \end{pmatrix}$$

Con la \star^3 resuelvo \star^2 :

$$Ux = y \xrightarrow[\text{con } \star^3]{\text{armo sistema}} \left(\begin{array}{cccc|c} 1 & -1 & 0 & 1 & 1 \\ 0 & 1 & 4 & 0 & -7 \\ 0 & 0 & -4 & -4 & 0 \\ 0 & 0 & 0 & 2 & 4 \end{array} \right) \Rightarrow x = \begin{pmatrix} 0 \\ 1 \\ -2 \\ 2 \end{pmatrix}$$

Y porque soy un tipazo (y le pifí 1000 veces a las cuentas) acá tenés el código para corroborar:


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

```
import numpy as np
import scipy

# Matriz A
A = np.array([[1, -1, 0, 1], [0, 1, 4, 0], [2, -1, 0, -2], [-3, 3, 0, -1]])
L = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [2, 1, 1, 0], [-3, 0, 0, 1]])
U = np.array([[1, -1, 0, 1], [0, 1, 4, 0], [0, 0, -4, -4], [0, 0, 0, 2]])
b = np.array([[1], [-7], [-5], [1]])


print(f"A =\n {A}")
print(f"L =\n {L}")
print(f"U =\n {U}")

print(f"\nA == LU --> {np.array_equal(A, L @ U)}")
print(f"Ax = b --> x = {np.transpose(np.linalg.solve(A,b))}")
```

Dale las gracias y un poco de amor  a los que contribuyeron! Gracias por tu aporte:

 naD GarRaz 


 Ale S. 

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 L^AT_EX → [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 L^AT_EX → [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 L^AT_EX → [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 L^AT_EX → [una pull request](#) al 🐙.

Ejercicio 9. Considerar la matriz

$$\begin{pmatrix} 4 & 2 & -2 \\ 2 & 5 & 5 \\ -2 & 5 & 11 \end{pmatrix}$$

Mostrar que es definida positiva y calcular su descomposición de Cholesky.

según la definición de matriz definida positiva:

$$\begin{aligned} \mathbf{x}^t \begin{pmatrix} 4 & 2 & -2 \\ 2 & 5 & 5 \\ -2 & 5 & 11 \end{pmatrix} \mathbf{x} &= 4x^2 - 4xz + 5y^2 + 10yz + 11z^2 \stackrel{!!}{=} (4x^2 - 4xz + z^2) + 5(y^2 + 2yz + z^2) + 5z^2 \\ &= 5z^2 + 5(y+z)^2 + (2x-z)^2 > 0 \end{aligned}$$

La matriz cumple la definición de *matriz definida positiva* $\forall \mathbf{x} \neq \mathbf{0} \in \mathbb{R}^3$. Sí, oka, hacer eso es una locura, más fácil es hacer lo que sigue y mirar los elementos de la matriz D :

Hay un teorema que dice algo así:

Sea $A \in \mathbb{R}^{n \times n}$ una matriz simétrica y definida positiva si y solo si existe $L \in \mathbb{R}^{n \times n}$ triangular inferior con diagonal positiva tal que $A = LL^t$.

Arranco como buscando la descomposición LU:

$$\begin{pmatrix} 4 & 2 & -2 \\ 2 & 5 & 5 \\ -2 & 5 & 11 \end{pmatrix} \xrightarrow[\substack{F_3 + \frac{1}{2}F_1}]{\substack{F_2 - \frac{1}{2}F_1}} \begin{pmatrix} 4 & 2 & -2 \\ 0 & 4 & 6 \\ 0 & 6 & 10 \end{pmatrix} \xrightarrow{F_3 - \frac{3}{2}F_2} \begin{pmatrix} 4 & 2 & -2 \\ 0 & 4 & 6 \\ 0 & 0 & 1 \end{pmatrix} = U$$



¡Sí! Ver los valores diagonales de U alcanza para ver que la matriz era efectivamente definida positiva. ¿Pero quién puede quitarnos el placer de haberlo comprobado de ambas formas?



Y ahora me formo la \tilde{L} a partir de la *eliminación gaussiana*:

$$\tilde{L} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -\frac{1}{2} & \frac{3}{2} & 1 \end{pmatrix}$$

Con esto ya casi estamos:

$$A = \tilde{L}U = \tilde{L}D\tilde{L}^t = \tilde{L}\sqrt{D}\sqrt{D}\tilde{L}^t = \tilde{L}\sqrt{D}(\tilde{L}\sqrt{D})^t = LL^t \text{ con } L = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ -1 & 3 & 1 \end{pmatrix}$$

Pequeña verificación:

```
import numpy as np

L_tilde = np.array([[1,0,0],[0.5,1,0],[-.5, 1.5, 1]])
U = np.array([[4,2,-2],[0,4,6],[0, 0, 1]])

print(f"L_tilde U=\n{L_tilde@U}")
#   [ 4 2 -2]
#   [ 2 5 5]
#   [-2 5 11]
```

```
import numpy as np

L_tilde = np.array([[1,0,0],[0.5,1,0],[-.5, 1.5, 1]])
D = np.array([[4,0,0],[0,4,0],[0, 0, 1]])

print(f"D L_tilde_traspuesta=\n{D@np.transpose(L_tilde)}")
#   [ 4 2 -2]
#   [ 0 4 6]
#   [ 0 0 1]
```

```
import numpy as np

L_tilde = np.array([[1,0,0],[0.5,1,0],[-.5, 1.5, 1]])
D_sqrt = np.array([[2,0,0],[0,2,0],[0, 0, 1]])
L_chole = L_tilde @ D_sqrt

print(f"A=LL_traspuesta=\n{L_chole@np.transpose(L_chole)}")
#   [ 4 2 -2]
#   [ 2 5 5]
#   [-2 5 11]
```

Dale las gracias y un poco de amor ❤️ a los que contribuyeron! Gracias por tu aporte:

👤 naD GarRaz 🐙

Ejercicio 10. Sea $A \in \mathbb{R}^{n \times n}$ una matriz simétrica. Probar que A es definida positiva si y solo si existe un conjunto de vectores linealmente independientes $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subseteq \mathbb{R}^n$ tal que $a_{ij} = \mathbf{x}_i^t \mathbf{x}_j$

(\Rightarrow) Si A es una matriz simétrica y definida positiva va a admitir la descomposición de Cholesky ([mirá acá](#)).

$$A \xleftrightarrow[A \text{ def. pos.}]{A = A^t} A = \tilde{L} \overset{d_{ij} > 0}{\underset{\substack{\uparrow \\ \text{la de LU}}}{D}} \tilde{L}^t \Leftrightarrow A = \tilde{L} \sqrt{D} \sqrt{D} \tilde{L}^t \Leftrightarrow A = LL^t \xleftrightarrow[\mathbf{x} \neq \mathbf{0}]{\times} 0 < \mathbf{x}^t A \mathbf{x} = \mathbf{x}^t L L^t \mathbf{x} = (L^t \cdot \mathbf{x})^t (L^t \mathbf{x}) = \mathbf{y}^t \mathbf{y}$$

Ahora esos \mathbf{y} tienen que ser *linealmente independientes*, más fácil, tengo que encontrar un solo conjunto:

$$\mathbf{y} = L^t \mathbf{x} \xrightarrow{\text{elijo}} \{\mathbf{x}_1, \dots, \mathbf{x}_n\} = \{\mathbf{e}_1, \dots, \mathbf{e}_n\} \Rightarrow \begin{cases} \mathbf{y}_1 = L^t \mathbf{e}_1 = \text{Col}(L^t)_1 \\ \vdots \\ \mathbf{y}_n = L^t \mathbf{e}_n = \text{Col}(L^t)_n \end{cases}$$

La matriz L es triangular inferior $\begin{pmatrix} & & 0 \\ & \text{triángulo azul} & \\ a_{ij} & & \end{pmatrix}$, con unos en la diagonal por lo que sus columnas son *mega linealmente independientes*. Y dado que

$$\mathbf{x}_i^t A \mathbf{x}_j = \mathbf{e}_i^t A \mathbf{e}_j \stackrel{!}{=} a_{ij} = \mathbf{y}_i^t \mathbf{y}_j$$

tenemos que el conjunto que verifica lo pedido:

$$\{\text{Col}(L^T)_1, \dots, \text{Col}(L^T)_n\} \subseteq \mathbb{R}^n$$

(\Leftarrow) La hipótesis ahora me dice que tengo vectores *linealmente independientes* y además que con esos vectores me formo la matriz A . Es decir que $A = X^t X$, con X :

$$X = \left(\mathbf{x}_1 \mid \dots \mid \mathbf{x}_n \right) \Rightarrow A = X^t X \xleftrightarrow[\mathbf{y} \neq \mathbf{0}]{\times} \mathbf{y}^t A \mathbf{y} = \mathbf{y}^t X^t X \mathbf{y} \Leftrightarrow \mathbf{y}^t A \mathbf{y} = (X \mathbf{y})^t X \mathbf{y} = \|X \mathbf{y}\|_2^2 > 0$$

Queda demostrado.

Dale las gracias y un poco de amor ❤️ a los que contribuyeron! Gracias por tu aporte:

👤 naD GarRaz 🐙

Ejercicio 11. 🤖... hay que hacerlo! 🐙

Si querés mandá la solución \rightarrow [al grupo de Telegram](#) 🗣️, o mejor aún si querés subirlo en L^AT_EX \rightarrow [una pull request](#) al 🐙.

Ejercicio 12. 🤖... hay que hacerlo! 🐙

Si querés mandá la solución \rightarrow [al grupo de Telegram](#) 🗣️, o mejor aún si querés subirlo en L^AT_EX \rightarrow [una pull request](#) al 🐙.

Ejercicio 13. 🤖... hay que hacerlo! 🐙

Si querés mandá la solución \rightarrow [al grupo de Telegram](#) 🗣️, o mejor aún si querés subirlo en L^AT_EX \rightarrow [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 $\text{IAT}_{\text{E}}\text{X}$ → *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 $\text{IAT}_{\text{E}}\text{X}$ → *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 $\text{IAT}_{\text{E}}\text{X}$ → *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 $\text{IAT}_{\text{E}}\text{X}$ → *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 $\text{IAT}_{\text{E}}\text{X}$ → *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 $\text{IAT}_{\text{E}}\text{X}$ → *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 $\text{IAT}_{\text{E}}\text{X}$ → *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 $\text{IAT}_{\text{E}}\text{X}$ → *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 $\text{IAT}_{\text{E}}\text{X}$ → *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 $\text{IAT}_{\text{E}}\text{X}$ → *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 $\text{IAT}_{\text{E}}\text{X}$ → *una pull request* al 🐙.

Ejercicios de parciales:

 1. _____