Apunte Único: Álgebra Lineal Computacional - Práctica 3

Por alumnos de ALC
Facultad de Ciencias Exactas y Naturales
UBA

última actualización 06/05/25 @ 11:52

Choose your destiny:

(dobleclick en el ejercicio para saltar)

Esta Guía 3 que tenés se actualizó por última vez:
06/05/25 @ 11:52
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:
  $$A = \overbrace{\tilde{L}U}^{\text{la de siempre}} \longrightarrow A = \tilde{L}\,\overbrace{\underset{\substack{\downarrow \\ \text{matriz diagonal} \\ \text{con los elementos} \\ \text{diagonales de } U}}{D}\tilde{L}^t}.$$

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

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

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:

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

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

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

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:

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

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 $k \in \mathbb{Z}$

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

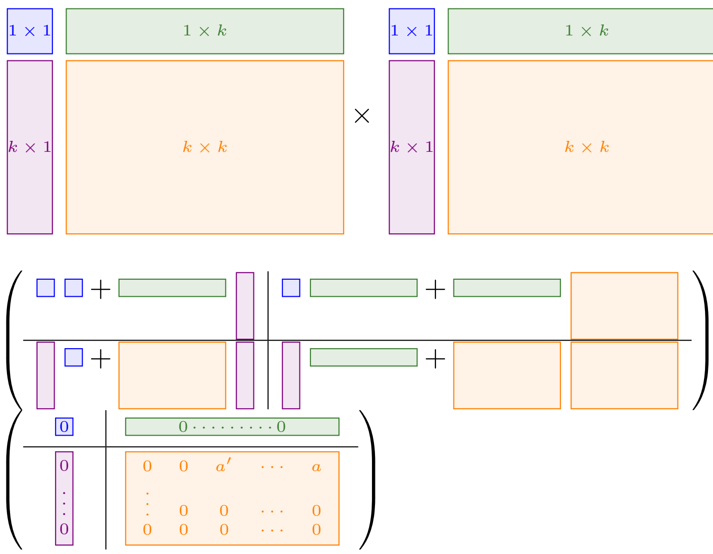
es verdadera. Por lo tanto ahora quiero probar que:

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

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:

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

$A^2 = A \cdot A =$



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

(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.

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

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

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 $\star^1$:

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

Ir al índice ↑

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

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

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

(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:

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

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

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

(c)

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

Entonces resuelvo primero $\bigstar^{1}$:

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

Con la $\bigstar^{3}$ resuelvo $\bigstar^{2}$:

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

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

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

```python
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

```python
"""
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. 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:

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

La matriz cumple la defición de *matriz definida positiva* $\forall \boldsymbol{x} \neq \boldsymbol{0} \in \mathbb{R}^3$. 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 sí existe $L \in \mathbb{R}^{n\times n}$ triangular inferior con diagonal positiva tal que $A = LL^t$.

Arranco como buscando la descomposición LU:

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

¡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:

```python
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]
```

```python
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]
```

```python
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 indepedientes $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subseteq \mathbb{R}^n$ tal que $a_{ij} = \boldsymbol{x}_i^t \boldsymbol{x}_j$

($\Longrightarrow$) Si A es una matriz simétrica y definida positiva:

$$A = A^t \quad \text{y} \quad \boldsymbol{x}^t A \boldsymbol{x} > 0 \qquad \forall \boldsymbol{x} \neq \boldsymbol{0} \in \mathbb{R}^n$$

($\Longleftarrow$)

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. ____________________