Universidad de Buenos Aires - Facultad de Ciencias Exactas y Naturales

Álgebra Lineal Computacional

2do Cuatrimestre 2025

Laboratorio N° 02: Transformaciones Lineales.

1. Introducción

Sean (w, z) y (x, y) dos sistemas coordenados, denominados el espacio de entrada y espacio de salida respectivamente. Una transformación geométrica de coordenadas define al mapeo desde el espacio de entrada al de salida de la forma:

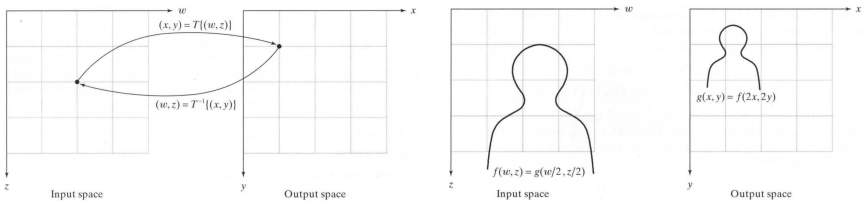
$$(x, y) = T\{(w, z)\}$$

donde $T\{\cdot\}$ se llama transformación o mapeo directo. Si $T\{\cdot\}$ posee una inversa, el mapeo inverso es aquel que traslada los puntos desde el espacio de salida al de entrada:

$$(x, y) = T^{-1}\{(w, z)\}$$

En la Fig. 1 se ilustra el mapeo directo e inverso para el ejemplo:

$$\begin{aligned}(x, y) &= T\{(w, z)\} = (w/2, z/2)\\ (w, z) &= T^{-1}\{(x, y)\} = (2x, 2y)\end{aligned}$$



(a) (b)

Figura 1: Transformación lineal

Las transformaciones geométricas de las imágenes se definen en términos de la transformación de coordenadas.

La Fig. 1 (b) muestra el efecto de aplicar la transformación lineal definida por $(x, y) = T\{(w, z)\} = (w/2, z/2)$ que escala la figura origina encogiéndola a la mitad.

De forma matricial, la proyección (y su inversa) se puede formalizar como:

$$\begin{pmatrix} x \\ y \end{pmatrix} = T \begin{pmatrix} w \\ z \end{pmatrix} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} w \\ z \end{pmatrix}$$

$$\begin{pmatrix} w \\ z \end{pmatrix} = T^{-1} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

donde *abusamos* la notación al llamar de la misma forma a la transformación lineal T y a su matriz asociada. En general, para una transformación lineal genérica $f(x, y) = (ax + by, cx + dy)$, sabemos que su matriz asociada es

$$T = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{1}$$

A lo largo de este notebook, nos enfocaremos en visualizar distintas características de las transformaciones lineales haciendo énfasis en los aspectos geométricos de las mismas.

2. Rotaciones, expansiones, traslaciones

En esta sección deberán programar diversas funciones que realizan deformaciones básicas a vectores de $\mathbb{R}^2$. En todos los casos, el objetivo es que exploren visualmente el resultado de aplicar la transformación usando las funciónes que se incluyen en el script `main.py`. La función `pointsGrid` genera una grilla de puntos como la de la Fig. 2. La función `proyectarPts` (a completar) aplicar una transformación lineal representada por un array de 2x2 a la grilla generada. Y `vistForm` se ocupa de graficar los valores transformados.

Ejercicio 1. Completen la función `proyectarPts` y grafiquen el efecto de aplicar la función $f(x, y) = (x/2, y/2)$ y su inversa.

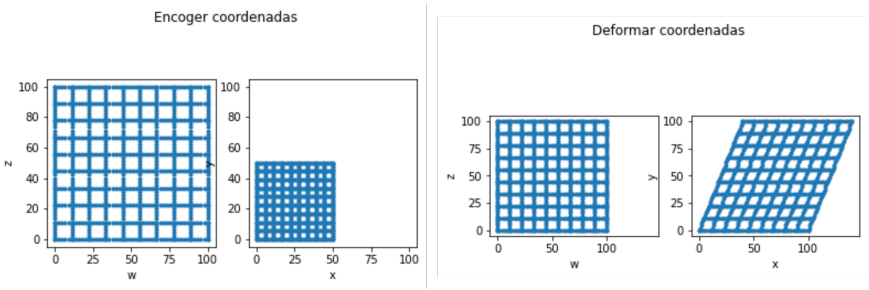


Figura 2: Salida del script de Python.

Ejercicio 2 (Reescalamiento)**.** Consideren la transformación lineal

$$(x, y) = T\{(w, z)\} = (aw, bz),$$

con $a, b \neq 0$. Se pide:

1. Encontrar la expresión de la matriz T y T^{-1}.
2. Tomando $a = 2$ y $b = 3$, usar estas transformaciones para aplicarlas en los vectores canónicos, a un punto arbitrario y a una circunferencia. ¿Qué sucede en cada caso?

Ejercicio 3 (Deformación de cizalla)**.** En este caso se trasladan las coordenadas horizontales un factor que depende de las verticales, y las coordenadas verticales quedan sin modificación, provocando una deformación de la salida:

$$(x, y) = T\{(w, z)\} = (w + cz, z + dw)$$

Se pide:

(a) Encontrar la expresión de la matriz T y T^{-1}.

(b) Considere el efecto de aplicar transformaciones lineales como en la Ec. 1, fijando $a = d = 1$, $c = 0$ y $b \in [0, 1]$. Describa cualitativamente el efecto de b en la transformación.

(c) Ahora varíe b manteniendo $b = c$, con $b \in [0, 1]$. Describa cualitativamente el efecto de la transformación.

Ejercicio 4 (Rotación)**.** Sea $R_\theta : \mathbb{R}^2 \to \mathbb{R}^2$ denota la rotación de un ángulo θ alrededor del origen, como muestra la Fig. 3.

$$\begin{pmatrix} X_\theta \\ Y_\theta \end{pmatrix} = R \begin{pmatrix} X \\ Y \end{pmatrix}$$

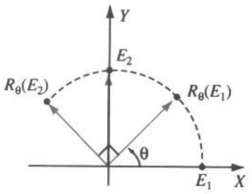


Figura 3: Transformación de rotación

Se pide:

(a) Encontrar las funciones $a(\theta)$, $b(\theta)$, $c(\theta)d(\theta)$ que dan lugar a la matriz $R(\theta)$.

(b) Validar $R(\theta)$ la función usando `pointsGrid`. En particular, prueben para $\theta = 0, \pi/2, \pi, 3\pi/2 2\pi$. ¿Qué distingue las deformaciones de cizalla de las rotaciones?

Ejercicio 5 (Rotación y reescalamiento)**.** Supongamos que quiero construir la siguiente transformación. En primer lugar dado un (x, y) que lo rote 45^{o}, luego haga un reescale de la forma $T(u, v) = (2u, 3v)$ y vuelva a rotar en sentido contrario a la primer rotación.

(a) Encontrar la expresión de la matriz que haga las tres transformaciones. *Sugerencia: Considerar una composición de transformaciones y la matriz hallada en el ejercicio anterior.*

(b) Usar dicha transformación para una circunferencia.

Ejercicio 6 (Transformación afín)**.** En los ejercicios precedentes se estudió el caso de matrices de transformación que encogían, dilataban o rotaban la posición de puntos iniciales, en un nuevo espacio a través del álgebra lineal.

Estas transformaciones son un caso particular de la transformación denominada *afín* . En general se puede formalizar las operaciones a través de:

$$\begin{pmatrix} x \\ y \end{pmatrix} = T \begin{pmatrix} w \\ z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

al cual hemos agregado la posibilidad de hacer una traslación lineal de las coordenadas origen en los dos ejes utilizando los valores constantes (b_1, b_2). A través de estas transformaciones se puede: escalar, rotar, trasladar o deformar. Como una conveniencia matemática y computacional, todas estas transformaciones pueden ser efectuadas utilizando una matriz de 3×3:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w \\ z \\ 1 \end{pmatrix}$$

agregarle el 1 a las coordenadas, nos permite seguir trabajando en coordenadas cartesianas (2D) con una matriz cuadrada de 3×3.

Se pide:

(a) Modificar la función proyectarPts para que sea capaz de realizar la operación de transformación afín, o sea con matrices de 3×3.

(b) Encontrar la expresión para hacer cada una de estas operaciones con las coordenadas **{escalar, trasladar, rotar, deformar}**, usando la matriz afín. Chequear con el script de Python.

Ejercicios extra

Ejercicio 7 (Núcleo e imagen)**.** En los siguientes puntos, el objetivo es visualizar qué significa que una dirección se encuentre en el núcleo y la imagen de una transformación lineal.

Escriban la matriz de las siguientes transformaciones lineales en base canónica. Identifiquen analíticamente el núcleo y la imagen de cada transformación lineal y describan geométricamente cada una de ellas usando `pointsGrid`.

(a) $f_1(x, y) = (x, x + y)$

(b) $f_2(x, y) = (\frac{x-y}{2}, \frac{y-x}{2})$

(c) $f_3(x, y) = (\frac{x+y}{2}, \frac{x-y}{2})$

(d) $f_4(x, y) = (x, x)$

(e) $f_5(x, y) = (\frac{x-y}{2}, \frac{x-y}{2})$

(f) $f_2 \circ f_3(x, y))$, donde $f \circ g(x, y) = f(g(x, y))$. Interpretar en términos de las gráficas de f_2 y f_3 la estructura observada.

(g) Comparar $f_4 \circ f_2(x, y))$ y $f_2 \circ f_4(x, y)$ y explicar por qué dan resultados distintos.

(h) Comparar f_3 y f_5 y explicar en términos geométricos por qué $f_3 \circ f_5(x, y) = (0, 0)$.

En todos los casos, identificar geométricamente núcleo, imagen, preimagen y coimagen de cada transformación lineal.

Ejercicio 8 (Transformaciones como flujos)**.** En el contexto de dinámica de fluidos, el flujo en un dado punto puede representarse por una transformación lineal T. En el caso más sencillo esta transformación lineal es constante, de forma

tal que el flujo en el punto (x, y) está definido por $T \begin{pmatrix} x \\ y \end{pmatrix}$. Graficamente, este flujo se representa con flechas en cada punto, que indican hacia donde se moverá el fluido que se encuentra en cada punto. Se pide:

a Utilizando la función `quiver` de matplotlib, y apoyandose en las funciones `pointsGrid` y `vistform`, construya una función `graficaFluido(T,wz)` que realice un gráfico de flujo como el de la Fig. 4.

b Visualicen las funciones del ejercicio anterior empleando esta metodología. ¿Cómo se manifiesta el núcleo en este caso? ¿Qué pasa con las velocidades conforme nos alejamos del origen?

c Si pensamos que la particula ubicada en $X = \begin{pmatrix} x \\ y \end{pmatrix}$ se mueve a $X + TX$, entonces podemos definir una trayectoria a partir de la sucesiön:

$$X_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$
$$X_{n+1} = TX_n + X_n$$

d Construya una función `trayectoria(T,X0,pasos)` que reciba una matriz `T` y un punto inicial `X0` y retorne `pasos` pasos de la trayectoria. Grafique la trayectoria como una línea sobre las líneas de flujo construidas con `graficaFluido` para las transformaciones lineales

- $h_1(x, y) = \frac{1}{2 \cdot 10^3}(-y, x)$ por 10^5 pasos partiendo del $(0, 1)$
- $h_2(x, y) = \frac{1}{2 \cdot 10^3}(x, x - y)$ por 5×10^3 pasos, partiendo del $(1/10, 1)$
- $h_3(x, y) = \frac{1}{2 \cdot 10^3}(-x, x - y)$ por 10^4 pasos, partiendo del $(1, 1)$

Se dice que un flujo representado por una transformación lineal cuya traza es cero es *incompresible*, y uno cuya traza es positiva tiene un *sumidero*, donde por donde se escapa el flujo. Vincule estos conceptos con lo que observa gráficamente.

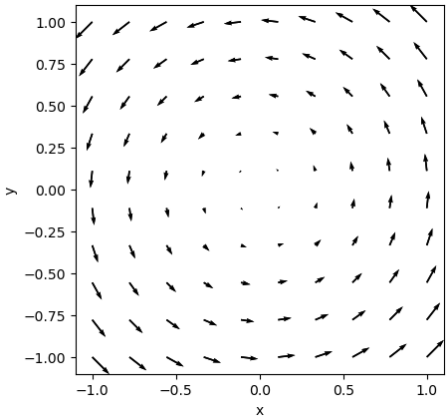


Figura 4: Flujo de ejemplo

Módulo ALC

Para el módulo ALC, deben programar:

```
def rota(theta):
"""
Recibe un angulo theta y retorna una matriz de 2x2
que rota un vector dado en un angulo theta
"""

def escala(s):
"""
Recibe una tira de numeros s y retorna una matriz cuadrada de
n x n, donde n es el tamano de s.
La matriz escala la componente i de un vector de Rn
en un factor s[i]
"""

def rota_y_escala(theta,s):
"""
Recibe un angulo theta y una tira de numeros s,
y retorna una matriz de 2x2 que rota el vector en un angulo theta
y luego lo escala en un factor s
"""

def afin(theta,s,b):
"""
Recibe un angulo theta, una tira de numeros s (en R2), y un vector
    b en R2.
Retorna una matriz de 3x3 que rota el vector en un angulo theta,
luego lo escala en un factor s y por ultimo lo mueve en un valor
    fijo b
"""

def trans_afin(v,theta,s,b):
"""
Recibe un vector v (en R2), un angulo theta,
una tira de numeros s (en R2), y un vector b en R2.
Retorna el vector w resultante de aplicar la transformacion afin a
    v
"""
```

Tests utilizando la función `assert`:

```
# Tests para rota
assert(np.allclose(rota(0), np.eye(2)))
assert(np.allclose(rota(np.pi/2), np.array([[0, -1],[1, 0]])))
assert(np.allclose(rota(np.pi), np.array([[-1, 0],[0, -1]])))

# Tests para escala
assert(np.allclose(escala([2,3]), np.array([[2,0],[0,3]])))
assert(np.allclose(escala([1,1,1]), np.eye(3)))
assert(
    np.allclose(escala([0.5,0.25]), np.array([[0.5,0],[0,0.25]]))
)

# Tests para rota_y_escala
assert(
    np.allclose(rota_y_escala(0,[2,3]), np.array([[2,0],[0,3]]))
)
assert(np.allclose(
    rota_y_escala(np.pi/2,[1,1]), np.array([[0,-1],[1,0]])
    ))
assert(np.allclose(
    rota_y_escala(np.pi,[2,2]), np.array([[-2,0],[0,-2]])
))

# Tests para afin
assert(np.allclose(
    afin(0,[1,1],[1,2]),
    np.array([[1,0,1],
              [0,1,2],
              [0,0,1]])))

assert(np.allclose(afin(np.pi/2,[1,1],[0,0]),
                   np.array([[0,-1,0],
                             [1, 0,0],
                             [0, 0,1]])))

assert(np.allclose(afin(0,[2,3],[1,1]),
                   np.array([[2,0,1],
                             [0,3,1],
                             [0,0,1]])))

# Tests para trans_afin
assert(np.allclose(
    trans_afin(np.array([1,0]), np.pi/2,[1,1],[0,0]),
    np.array([0,1])
))
assert(np.allclose(
    trans_afin(np.array([1,1]), 0,[2,3],[0,0]),
    np.array([2,3])
))
assert(np.allclose(
    trans_afin(np.array([1,0]), np.pi/2,[3,2],[4,5]),
    np.array([4,7])
))
```