Actividad_letra_A

November 24, 2024

1 Matriz para Letra A

1.0.1 Trabajo realizado por: Jessica Naomi Millan Sánchez

1.0.2 Graficación Computacional

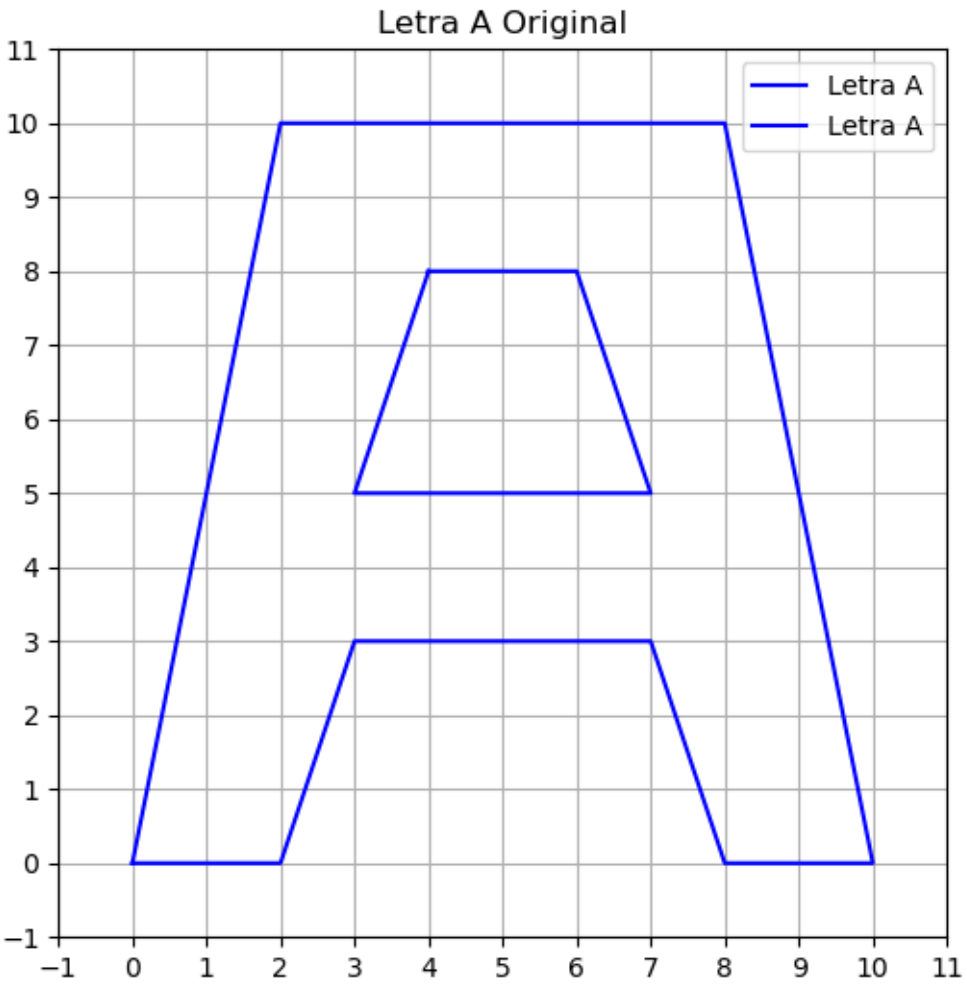
1.0.3 Profesora: Hazem Álvarez Rodríguez

1.0.4 Clase del 20 de noviembre de 2024

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
```

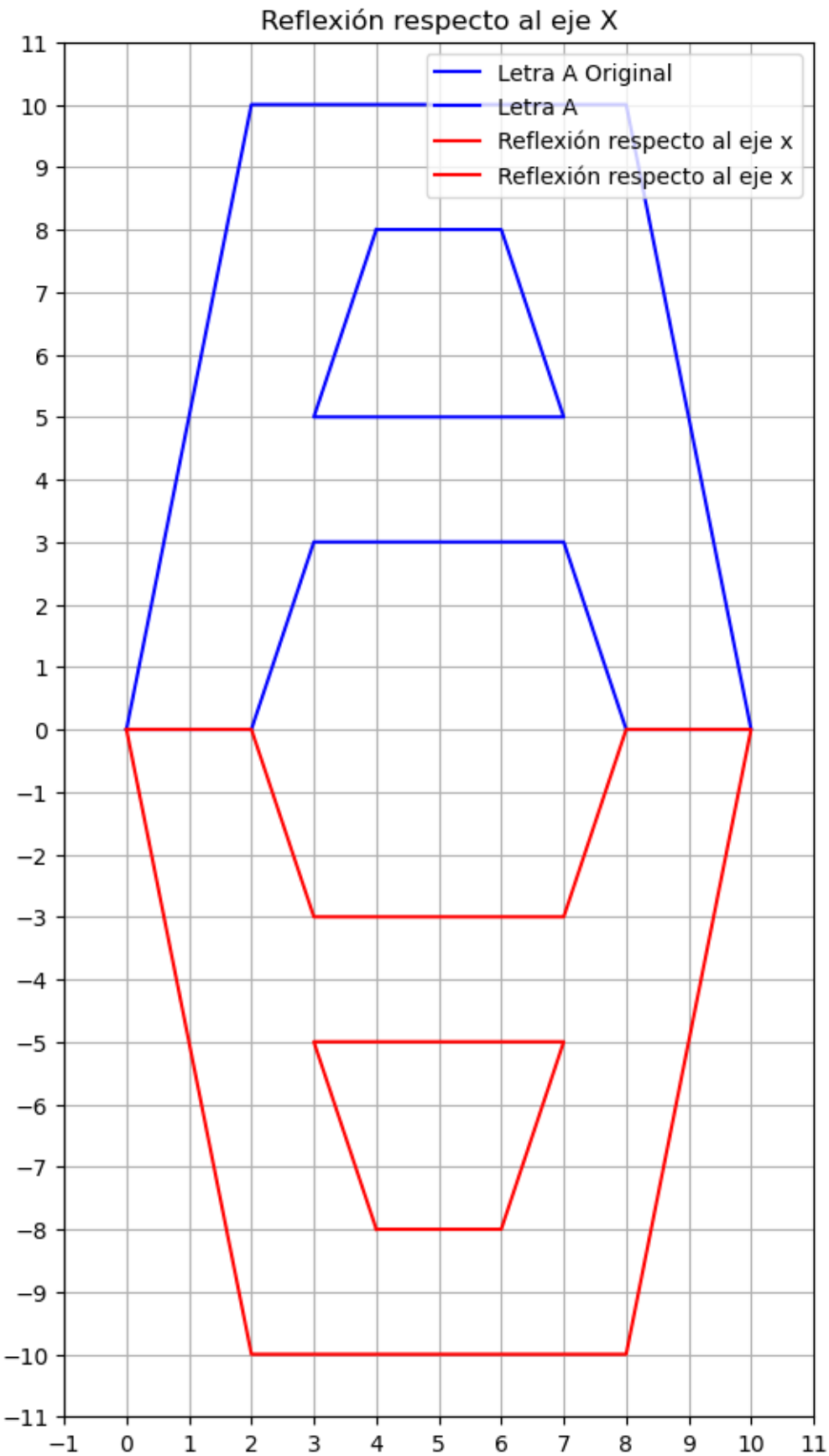
[56]:
```
# Definimos la matriz E con los puntos de la letra "E" corregida
E = np.array([
    (0, 0, 0), (2,0,0), (3,3,0), (7,3,0), (8,0,0), (10,0,0), (8,10,0),␣
 ↪(2,10,0), (0,0,0)  ])

E1 = np.array([
    (4, 8, 0), (6,8,0), (7,5,0), (3,5,0), (4,8,0)  ])
```

[3]:
```
# Matriz Identidad
Ic = np.eye(3)

# Graficamos la letra "E" original
fig, ax = plt.subplots(figsize=(6, 6))
ax.plot(E[:, 0], E[:, 1], color="blue", label="Letra A")
ax.plot(E1[:, 0], E1[:, 1], color="blue", label="Letra A")
ax.set_title("Letra A Original")
ax.set_xticks(np.arange(-1, 12, 1))
ax.set_yticks(np.arange(-1, 12, 1))
ax.set_aspect('equal', adjustable='box')
ax.grid()
ax.legend()
plt.show()
```

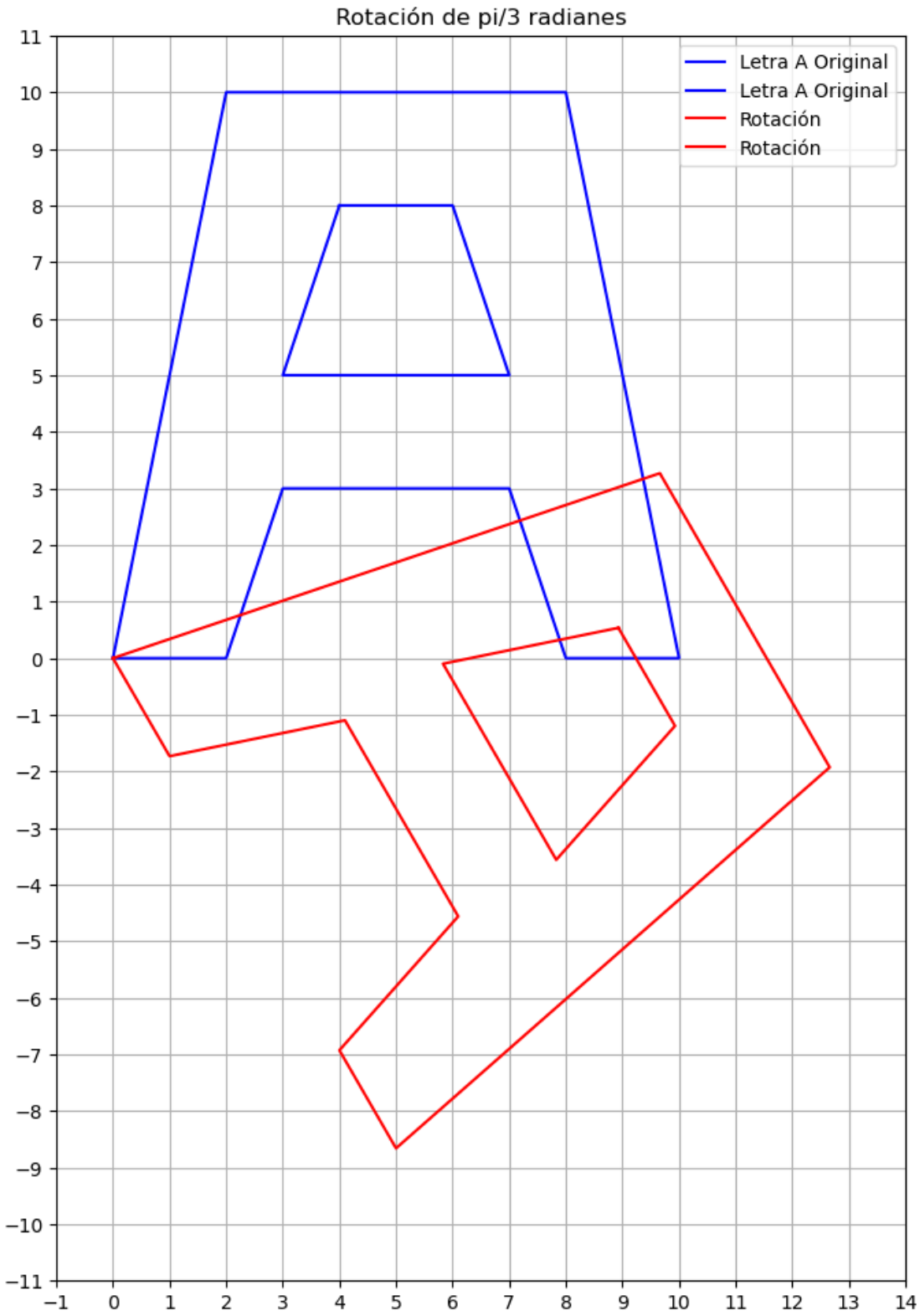


[17]:
```python
# Matriz de Reflexión respecto al eje x
Refx = np.array([
    [1., 0, 0],
    [0, -1., 0],
    [0., 0., 1.]
])

# Aplicamos la reflexión con respecto al eje x
Ex, Ey = [], []
for row in E:
    output_row = Refx.dot(row)  # Multiplicación matriz-vector
    x, y = output_row[:2]    # Extraemos las coordenadas x, y
    Ex.append(x)
    Ey.append(y)

Ex1, Ey1 = [], []
```

```
for row in E1:
    output_row = Refx.dot(row)  # Multiplicación matriz-vector
    x, y = output_row[:2]    # Extraemos las coordenadas x, y
    Ex1.append(x)
    Ey1.append(y)

# Graficamos la transformación
fig, ax = plt.subplots(figsize=(6, 12))
ax.plot(E[:, 0], E[:, 1], color="blue", label="Letra A Original")
ax.plot(E1[:, 0], E1[:, 1], color="blue", label="Letra A")
ax.plot(Ex, Ey, color="red", label="Reflexión respecto al eje x")
ax.plot(Ex1, Ey1, color="red", label="Reflexión respecto al eje x")
ax.set_title("Reflexión respecto al eje X")
ax.set_xticks(np.arange(-1, 12, 1))
ax.set_yticks(np.arange(-11, 12, 1))
ax.set_aspect('equal', adjustable='box')
ax.grid()
ax.legend()
plt.show()
```


Reflexión respecto al eje X
Letra A Original
Letra A
Reflexión respecto al eje x
Reflexión respecto al eje x
11
10
9
8
7
6
5
4
3
2
1
0
−1
−2
−3
−4
−5
−6
−7
−8
−9
−10
−11
−1
0
1
2
3
4
5
6
7
8
9
10
11

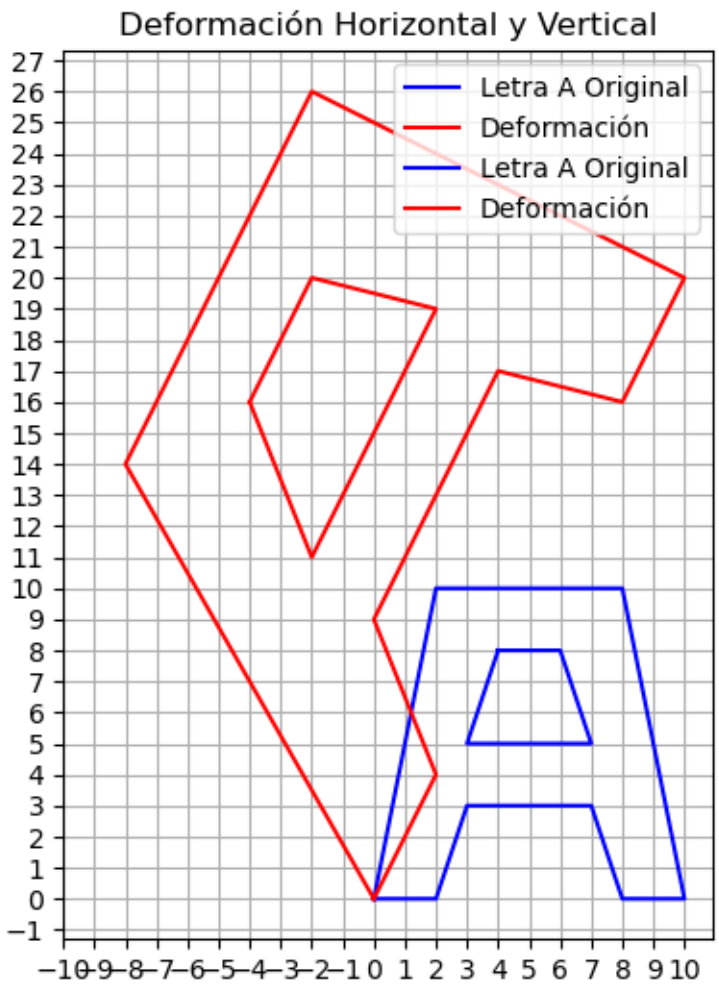
[65]:
```
# Ángulo de rotación deseado (en este caso pi/3)
theta = np.pi / 3

# Matriz de Rotación
R = np.array([
    [np.cos(theta), np.sin(theta), 0.],
    [-np.sin(theta), np.cos(theta), 0.],
    [0., 0., 1.]
])

# Aplicamos la rotación
Ex, Ey = [], []
for row in E:
    output_row = R.dot(row)  # Multiplicación matriz-vector
    x, y = output_row[:2]    # Extraemos las coordenadas x, y
    Ex.append(x)
    Ey.append(y)

# Aplicamos la rotación
Ex1, Ey1 = [], []
for row in E1:
    output_row = R.dot(row)  # Multiplicación matriz-vector
    x, y = output_row[:2]    # Extraemos las coordenadas x, y
    Ex1.append(x)
    Ey1.append(y)


# Graficamos la transformación
fig, ax = plt.subplots(figsize=(8, 12))
ax.plot(E[:, 0], E[:, 1], color="blue", label="Letra A Original")
ax.plot(E1[:, 0], E1[:, 1], color="blue", label="Letra A Original")

ax.plot(Ex, Ey, color="red", label="Rotación")
ax.plot(Ex1, Ey1, color="red", label="Rotación")

ax.set_title("Rotación de pi/3 radianes")
ax.set_xticks(np.arange(-1, 15, 1))
ax.set_yticks(np.arange(-11, 12, 1))
ax.set_aspect('equal', adjustable='box')
ax.grid()
ax.legend()
plt.show()
```


Rotación de pi/3 radianes
Letra A Original
Letra A Original
Rotación
Rotación

[64]:
```
# Escalar
s = 2

# Matriz de cambio de escala
S = np.array([
    [s, 0., 0.],
    [0., s, 0.],
    [0., 0., 1.]
])

# Aplicamos el cambio de escala
Ex, Ey = [], []
for row in E:
    output_row = S.dot(row)  # Multiplicación matriz-vector
    x, y = output_row[:2]    # Extraemos las coordenadas x, y
    Ex.append(x)
    Ey.append(y)

Ex1, Ey1 = [], []
for row in E1:
    output_row = S.dot(row)  # Multiplicación matriz-vector
    x, y = output_row[:2]    # Extraemos las coordenadas x, y
    Ex1.append(x)
    Ey1.append(y)

# Graficamos la transformación
fig, ax = plt.subplots(figsize=(6, 6))
ax.plot(E[:, 0], E[:, 1], color="blue", label="Letra A Original")
ax.plot(Ex, Ey, color="red", label="Cambio de escala")
ax.plot(E1[:, 0], E1[:, 1], color="blue", label="Letra A Original")
ax.plot(Ex1, Ey1, color="red", label="Cambio de escala")

ax.set_title("Escala por factor 2")
ax.set_xticks(np.arange(-1, 23, 1))
ax.set_yticks(np.arange(-1, 23, 1))
ax.set_aspect('equal', adjustable='box')
ax.grid()
ax.legend()
plt.show()
```

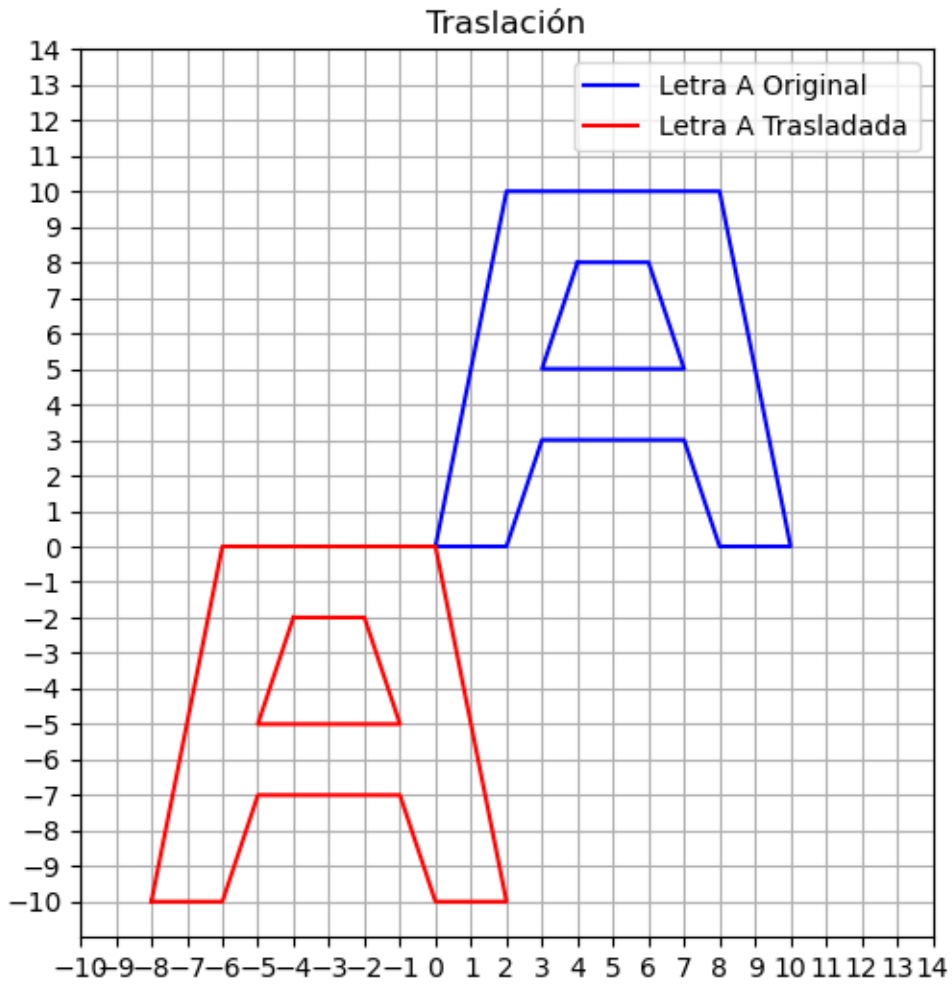
[63]:
```
# Factores de deformación
h = -1  # Deformación horizontal
v = 2   # Deformación vertical

# Matriz de deformación
D = np.array([
    [1., h, 0.],
    [v, 1., 0.],
    [0., 0., 1.]
])

# Aplicamos la deformación
Ex, Ey = [], []
for row in E:
    output_row = D.dot(row)  # Multiplicación matriz-vector
    x, y = output_row[:2]    # Extraemos las coordenadas x, y
```

```
    Ex.append(x)
    Ey.append(y)

# Aplicamos la deformación
Ex1, Ey1 = [], []
for row in E1:
    output_row = D.dot(row)  # Multiplicación matriz-vector
    x, y = output_row[:2]    # Extraemos las coordenadas x, y
    Ex1.append(x)
    Ey1.append(y)

# Graficamos la transformación
fig, ax = plt.subplots(figsize=(6, 6))
ax.plot(E[:, 0], E[:, 1], color="blue", label="Letra A Original")
ax.plot(Ex, Ey, color="red", label="Deformación")
ax.plot(E1[:, 0], E1[:, 1], color="blue", label="Letra A Original")
ax.plot(Ex1, Ey1, color="red", label="Deformación")
ax.set_title("Deformación Horizontal y Vertical")
ax.set_xticks(np.arange(-10, 11, 1))
ax.set_yticks(np.arange(-1, 28, 1))
ax.set_aspect('equal', adjustable='box')
ax.grid()
ax.legend()
plt.show()
```



[62]:
```
import numpy as np
import matplotlib.pyplot as plt

# Valores de traslación
tx = -8
ty = -10

# Matriz de traslación
T = np.array([
    [1., 0., tx],
    [0., 1., ty],
    [0., 0., 1.]
])


# Aplicamos la traslación usando multiplicación de matrices
```

```
E_translated = (T @ E.T).T  # (n x 3)
E1_translated = (T @ E1.T).T

# Graficamos la transformación
fig, ax = plt.subplots(figsize=(6, 6))
ax.plot(E[:, 0], E[:, 1], color="blue", label="Letra A Original")
ax.plot(E1[:, 0], E1[:, 1], color="blue")

ax.plot(E_translated[:, 0], E_translated[:, 1], color="red", label="Letra A␣
↪Trasladada")
ax.plot(E1_translated[:, 0], E1_translated[:, 1], color="red")

ax.set_title("Traslación")
ax.set_xticks(np.arange(-10, 15, 1))
ax.set_yticks(np.arange(-10, 15, 1))
ax.set_aspect('equal', adjustable='box')
ax.grid()
ax.legend()
plt.show()
```


Traslación
Letra A Original
Letra A Trasladada