

```
In [102]: import matplotlib.pyplot as plt
          from matplotlib import cm
          import numpy as np
          import string
```

```
In [103]: # points a, b, c, d
          a, b, c, d = (-1, 1), (1, 1), (1, -1), (-1, -1)

          # matrix with row vectors of points
          A = np.array([a, b, c, d])

          # 2x2 Identity transformation matrix
          I = np.eye(2)
```

```

In [109]: T_rotate = np.array([[0.707, 0.707], [-0.707, 0.707]])
T_scale = np.array([[1.5, 0], [0, 1]])
T_inverse_rotate = T_rotate.T
compose = T_inverse_rotate @ T_scale @ T_rotate

p_diag = 1.5
derived = np.array([[0.5*p_diag + 0.5, 0.5*p_diag - 0.5], [0.5*p_diag - 0.5, 0.5*p_diag + 0.5]])

fig = plt.figure()
ax = plt.gca()
xs_s = []
ys_s = []

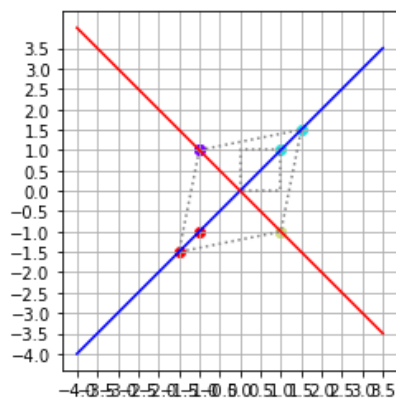
colors = cm.rainbow(np.linspace(0, 1, 4))
for index, row in enumerate(A):
    output_row = derived @ row
    x, y = row
    x_s, y_s = output_row
    xs_s.append(x_s)
    ys_s.append(y_s)
    plt.scatter(x, y, color=colors[index])
    plt.scatter(x_s, y_s, color=colors[index])

xs_s.append(xs_s[0])
ys_s.append(ys_s[0])
plt.plot(xs, ys, color="gray", linestyle='dotted')
plt.plot(xs_s, ys_s, color="gray", linestyle='dotted')

x_range = np.arange(-4, 4, 0.5)
plt.plot(x_range, x_range, color="blue")
plt.plot(x_range, -x_range, color="red")
ax.set_xticks(x_range)
ax.set_yticks(x_range)

plt.gca().set_aspect('equal', adjustable='box')
plt.grid()
plt.show()

```



In [ ]:

In [ ]:

In [ ]:

In [ ]: