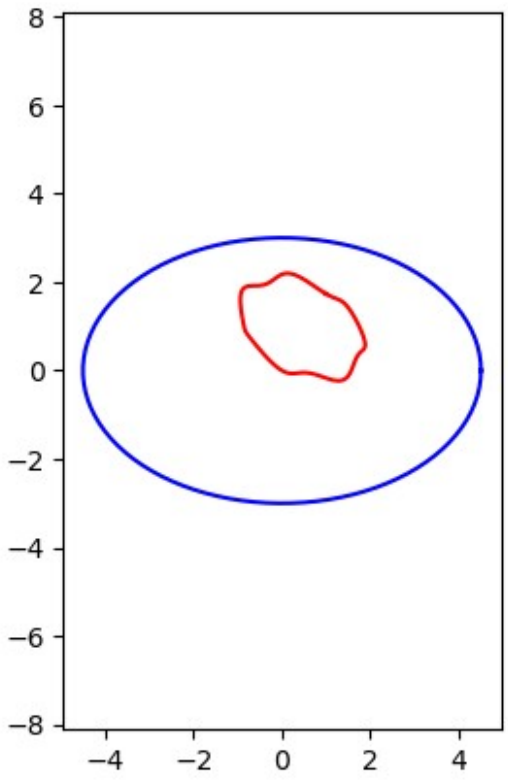
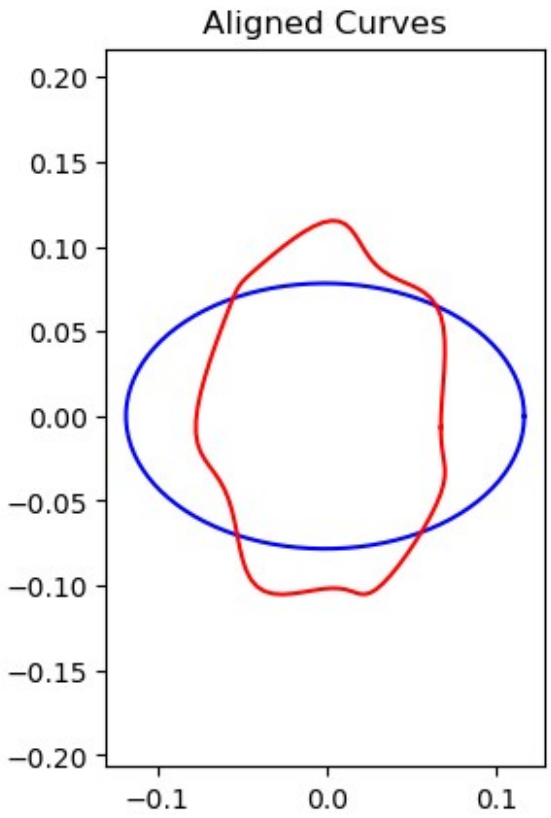
```python
import numpy as np
from scipy.spatial import procrustes

# I defined t as a range provided by the exercises text through the
linspace function
t = np.linspace(0, 2*np.pi, 100)

# I defined the two functions as python functions in a raw way as
written in the exercise text
def gamma1(t):
    return 3*(1.5*np.cos(t) + 1j*np.sin(t))

def gamma2(t):
    return np.exp(1j*np.pi/3) * (1 + np.cos(t) + 1.5j*np.sin(t) +
0.125*(1 + 1.5j)*np.sin(3*t)**2)

# save the points as real-valued 2D landmark points
g1p = np.array([[real, imag] for real, imag in zip(gamma1(t).real,
gamma1(t).imag)])
g2p = np.array([[real, imag] for real, imag in zip(gamma2(t).real,
gamma2(t).imag)])

# solve the Procrustes problem through this simple function from the
procrustes library
gamma1_aligned, gamma2_aligned, disparity = procrustes(g1p, g2p)

import matplotlib.pyplot as plt

# plot for the 'raw' curves
plt.subplot(1, 2, 1)
plt.plot(g1p[:,0], g1p[:,1], 'b-', label='$\gamma_1$ Original')
plt.plot(g2p[:,0], g2p[:,1], 'r-', label='$\gamma_2$ Original')

# equalize the scales of the axes
plt.axis('equal')

plt.show()
```




```
# plot for the aligned curves
plt.subplot(1, 2, 2)
plt.plot(gamma1_aligned[:,0], gamma1_aligned[:,1], 'b-', label='$\
gamma_1$ Aligned')
plt.plot(gamma2_aligned[:,0], gamma2_aligned[:,1], 'r-', label='$\
gamma_2$ Aligned')
plt.title('Aligned Curves')

# equalize the scales of the axes
plt.axis('equal')

plt.show()
```


Aligned Curves
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.1
0.0
0.1