1 Introduction

In the previous example you plotted a curve using some code like below.

```
import scipy.io as sio
import numpy as np
import matplotlib.pyplot as plt

# load the data
data = sio.loadmat('closed_curves_2D/toydata.mat')

# Examine the data
data.keys()
# ignore the keys beginning and ending with `__`. There is only one variable called `C` (C for curves).

# Examine the `C` data
d,n,N=data['C'].shape

# d is the dimension of space the curve is in (2D plane)
# n is the number of points in the curve
# N is the number of curves
# so you have 1300 2D curves with 100 points each.

# print the first curve to screen
print(data['C'][:,:,0] )

# put your code here to plot the curve

plt.plot(data['C'][0,:,0],data['C'][1,:,0] )
```

Next try to apply each of the 4 shape similarity transformations to the curve.

Let $\beta : [0,1] \to \mathbb{R}^2$ denote your parametrized shape curve. Let $0 = t_1 < t_2 < t_n = 1$ denote the time points you have observed values $\beta(t_1), \ldots, \beta(t_n)$.

1.1 Translate the curve

Let $p \in \mathbb{R}^2$ be some point (say $p = (1,1)$). Compute the translated curve $\beta(t_i) + p$ for each $i \in \{1, \ldots, n\}$. Plot the translated curve.

1.2 Scale the curve

Let $s \in [0, \infty)$ be a scalar. Compute the scaled curve $s\beta(t_i)$ for each $i \in \{1, \ldots, n\}$. Plot the scaled curve.

1.3 Rotate the curve

Let $R \in SO(2)$ be a 2×2 special orthogonal matrix. Compute the rotated curve $R\beta(t_i)$ for each $i \in \{1, \ldots, n\}$. Note that right matrix multiplication is applied to a 2D vector $\mathbb{R}^{2\times 1}$. Plot the rotated scaled curve.

1.4 Reflect/Rotate the curve

Let $R \in O(2)$ be a 2×2 orthogonal matrix.
Make sure it has determinant -1 to that a reflection occurs. Compute the reflected curve $R\beta(t_i)$ for each $i \in \{1, \ldots, n\}$. Note that right matrix multiplication is applied to a 2D vector $\mathbb{R}^{2\times 1}$. Plot the reflected curve.