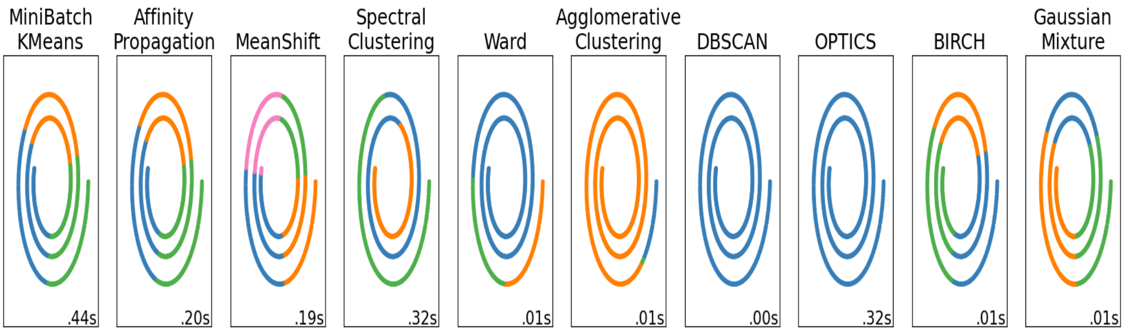
ODE indentification without Time Label

BY ZHIJUN ZENG

Date: 2023/4/2

1 Clustering Based segmentation

Thining about using Clustering method to segment the data into different piece:



The most important feature is that the clustering algorithm shoud avoid uncrossed time interval mixing.

1.1 AgglomerativeClustering

Agglomerative Clustering recursively merges pair of clusters of sample data; uses linkage distance.

The `AgglomerativeClustering` object performs a hierarchical clustering using a bottom up approach: each observation starts in its own cluster, and clusters are successively merged together. The linkage criteria determines the metric used for the merge strategy:

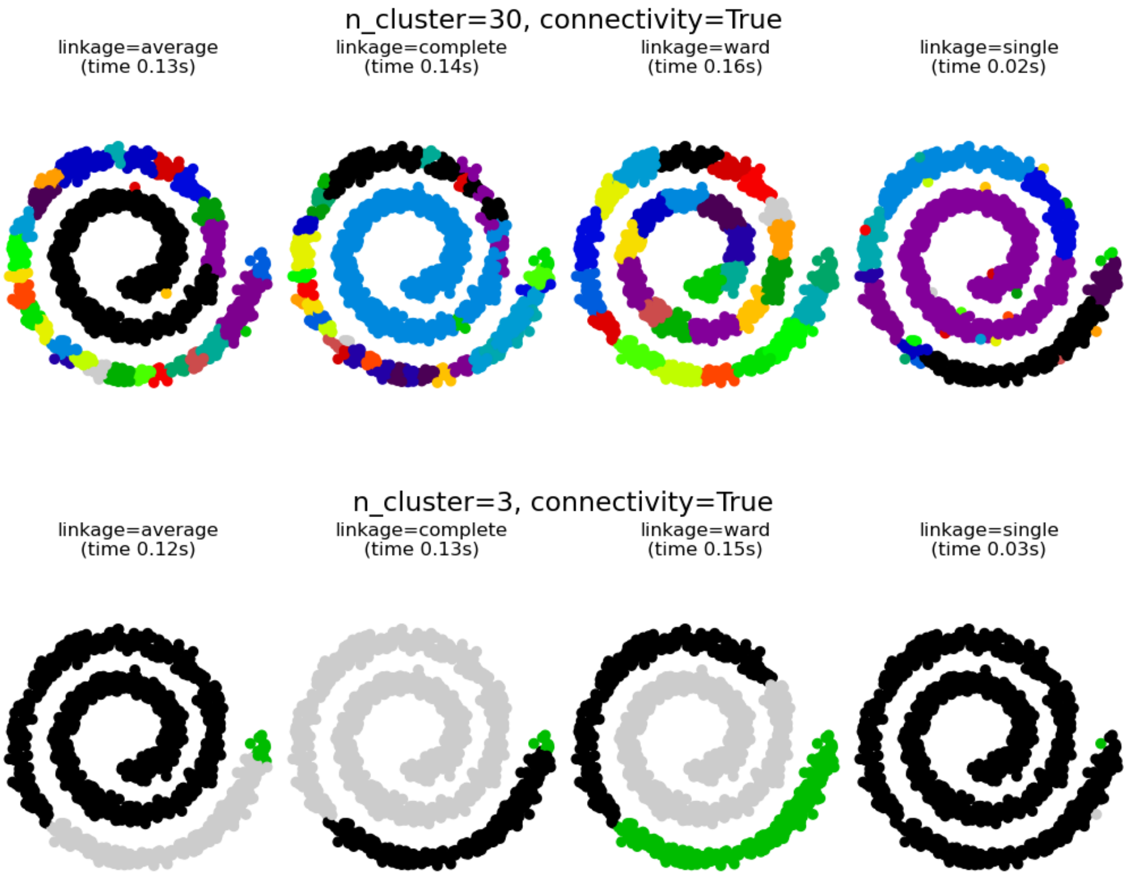
- **Ward** minimizes the sum of squared differences within all clusters. It is a variance-minimizing approach and in this sense is similar to the k-means objective function but tackled with an agglomerative hierarchical approach.

- **Maximum** or **complete linkage** minimizes the maximum distance between observations of pairs of clusters.

- **Average linkage** minimizes the average of the distances between all observations of pairs of clusters.

- **Single linkage** minimizes the distance between the closest observations of pairs of clusters.

AgglomerativeClustering can also scale to large number of samples when it is used jointly with a connectivity matrix, but is computationally expensive when no connectivity constraints are added between samples: it considers at each step all the possible merges.



1.2 DBSCAN

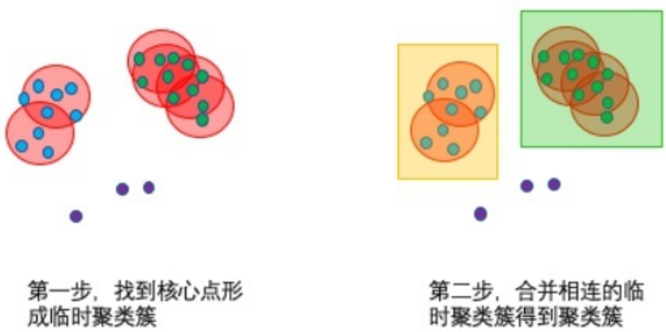
The DBSCAN algorithm views clusters as areas of high density separated by areas of low density. Due to this rather generic view, clusters found by DBSCAN can be any shape, as opposed to k-means which assumes that clusters are convex shaped. The central component to the DBSCAN is the concept of *core samples*, which are samples that are in areas of high density.

1. Finding core point and construct temp clusters

   Scan all the sample point, if the R radius neighbour of some point are larger than min_points, then these are assigned as core point, the density reachable points of a core point is formed as temp clusters.

2. Merge temp clusters to obtain final clusters

For each temp cluster, check if its points are core points, if so merge their temp clusters together to obtain new temp cluster.



1.3 Method

1. Fix the cluster number N,randomly select N point from the samples (these point construct a initial condition set).

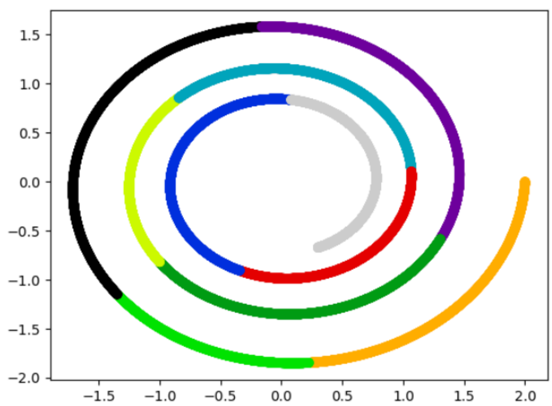
2. Remove a epsilon neighbour ball of these points to obtain segment point and intervals.

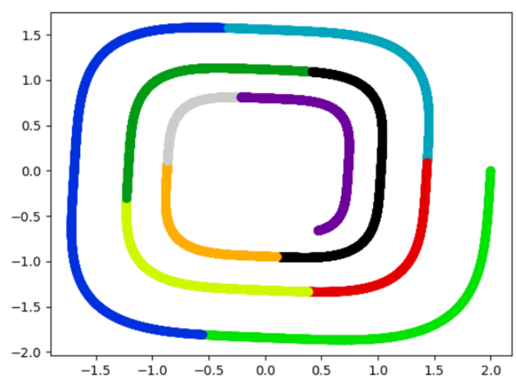
3. Using the DBSCAN method the segment the data into different pieces

   The time interval length is $\frac{N_{\text{interval}}}{N_{\text{all}}}T$, initial condition is the segment point. To see this ,we already know which interval the inital condion lies in, then we can find the other ends of this cluster, we compare the distance of all segment point to determined which points are the ends and becomes the start of the next interval, so on.

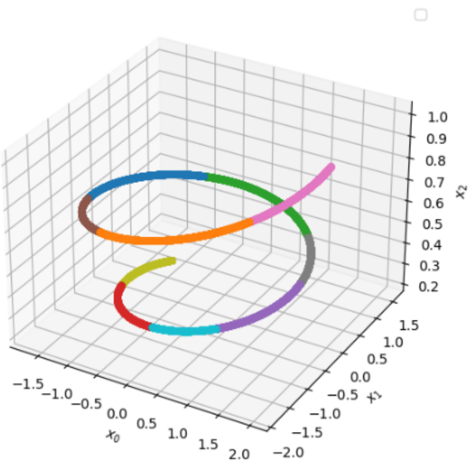
Here is the clustering result of different task:

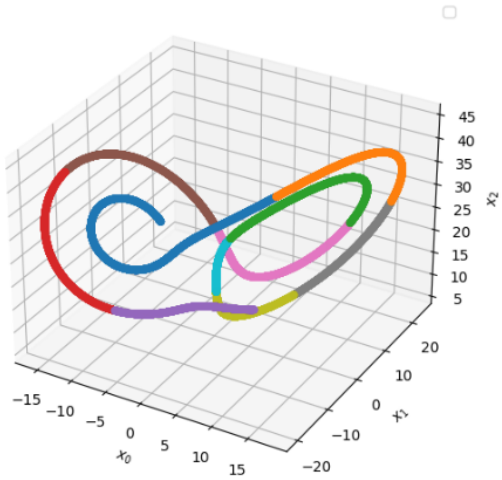
1.3.1 Linear 2D and Cubic2D





1.3.2 Linear3D and Lorentz





2 NN representation of Random Variable Transform and PINN/LSQ indentification

Problem: Directly learning the representation of ODE using ode solver is too slow when the initial parameter are far from gound truth and even not converges.

Target: Using NN to obtain fast training speed in fitting the transition dynamics(a random variable transform of $U[0,T]$)

Algorithm:

算法 1

Using clustering method to segment the dataset into N subset, the inital condition and time interval length is $\{x_i\}_{i=1}^N, \{T_i\}_{i=1}^N$.

For each epoch,for each interval, sample training time data from $\{t_i\} \sim U[0, T_i]$, and compute SWD between $\text{NN}(\{t_i\})$ and $\{y_i\} \sim U[\text{Cluster } N]$

PINN:

For each epoch ,sample a small batch of $\{t_i\} \sim U[0, T]$, compute autograd to obtain $\text{NN}_t(\{t_i\})$, $\text{loss} = \|A\zeta_{\text{basis}}(t_i) - \text{NN}_t(t_i)\|_2$ to learn learn the ode parameters ζ_{basis}

LSQ

Sample a large batch of $\{t_i\} \sim U[0, T]$, compute autograd to obtain $\text{NN}_t(\{t_i\})$, solving the following least square problem :$\|A\zeta_{\text{basis}}(t_i) - \text{NN}_t(t_i)\|_2$

3 Result

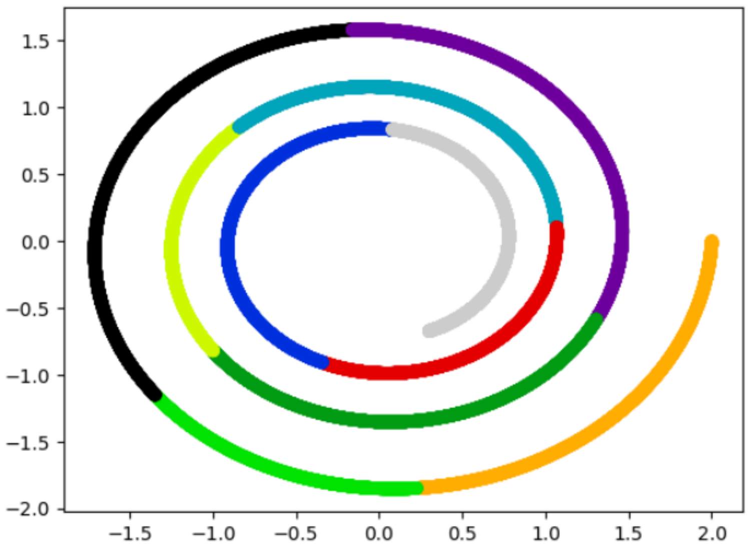
3.1 Linear 2D

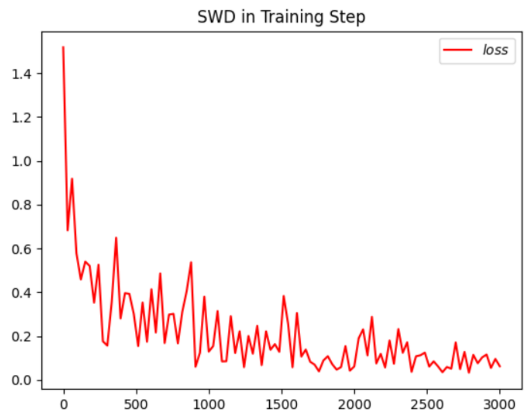
In this example , we consider the two-dimensional damped hamonic oscillator with linear, as in

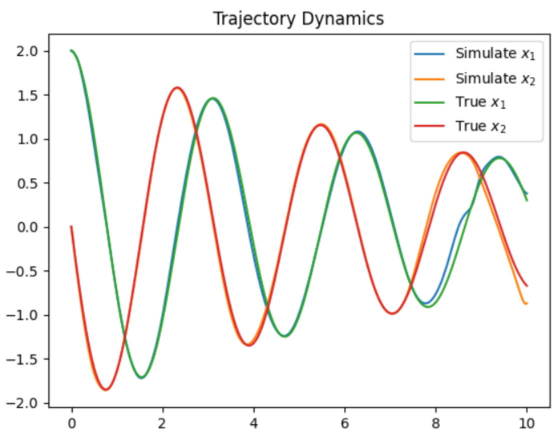
$$\frac{d}{\text{dt}}\begin{bmatrix} q \\ p \end{bmatrix} = \begin{bmatrix} -0.1 & 2 \\ -2 & -0.1 \end{bmatrix}\begin{bmatrix} q \\ p \end{bmatrix} \tag{1}$$

Here we consider T =10, and the initial condition is $[2,0]$,We generate 30000 samples using RK4 method and sorted uniform time label $\{t_i\}$.

Using clustering method we can obtaim segmentations (N_cluster = 10)







LSQ result :

degree 1 Polynomial Library

	1	x1	x2
x1	0.	-0.1196	1..95
x2	0.	-2.0462	-0.0644

degree 2 Polynomial Library

	1	x1	x2	x1^2	x1 x2	x2^2
x1	0.	-0.1196	1..95	0	0	0
x2	0.	-2.0462	-0.0644	0	0	0

degree 3 Polynomial Library

	1	x1	x2	x1^2	x1 x2	x2^2	x1 ^3	x1^2x2	x1x2^2	x2^3
x1	0.		1..90	0	0	0	0	0.1119	0	0
x2	0.	-2.0398	-0.101	0	0	0	0	0	0	0

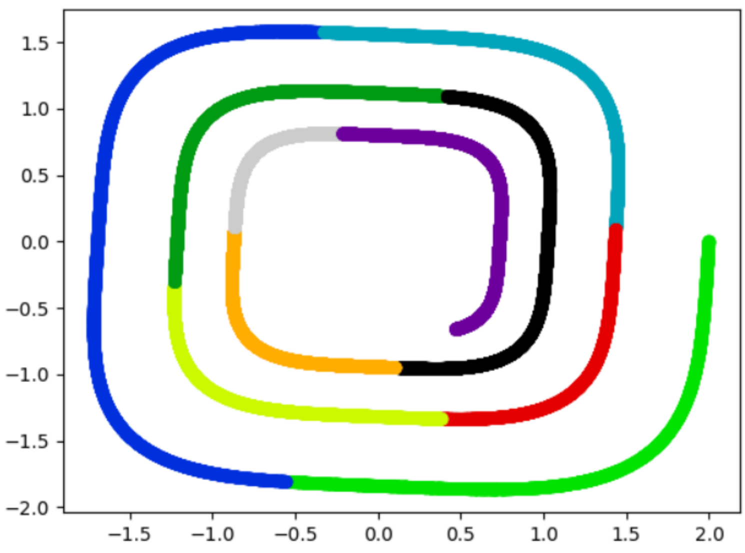
3.2 Cubic 2D

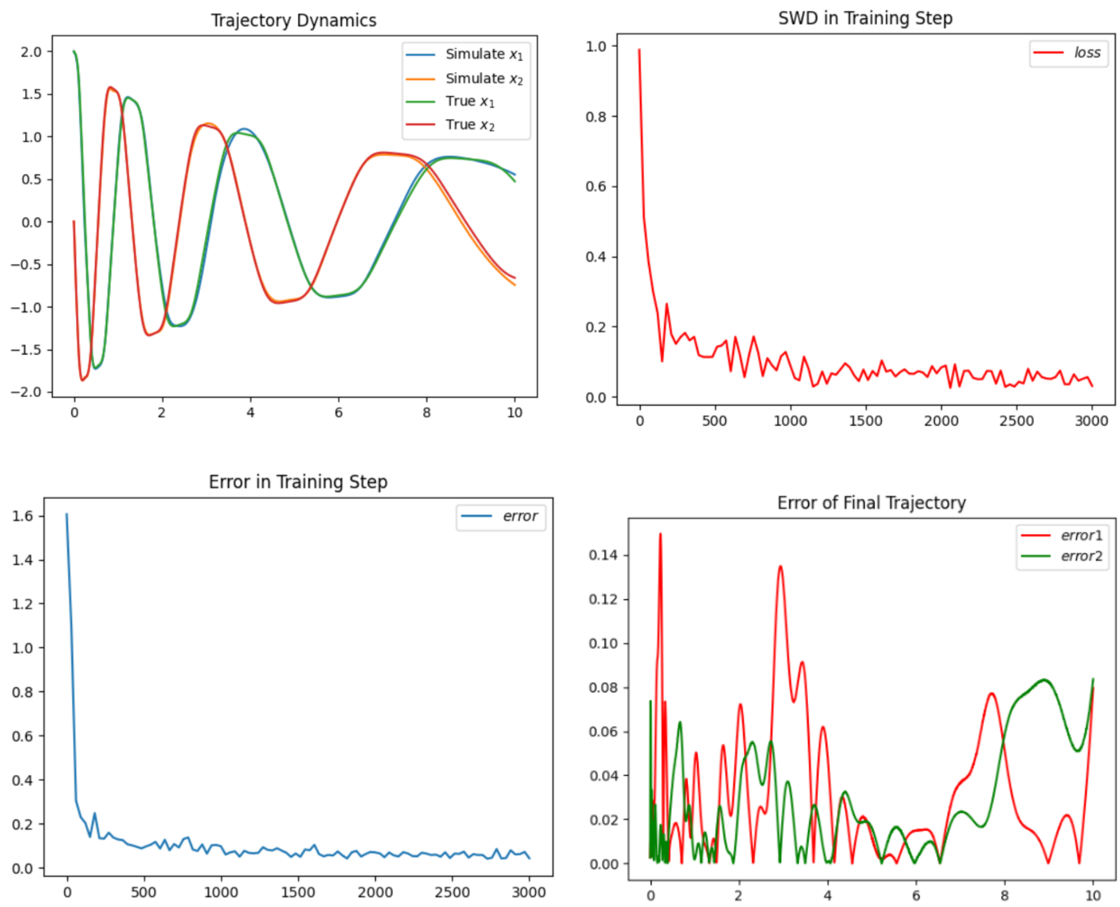
In this example , we consider the two-dimensional damped hamonic oscillator with linear, as in

$$\frac{d}{\mathrm{dt}}\begin{bmatrix} q \\ p \end{bmatrix} = \begin{bmatrix} -0.1 & 2 \\ -2 & -0.1 \end{bmatrix}\begin{bmatrix} q^3 \\ p^3 \end{bmatrix} \tag{2}$$

Here we consider T $=$10, and the initial condition is $[2,0]$,We generate 30000 samples using RK4 method and sorted uniform time label $\{t_i\}$.

Using clustering method we can obtaim segmentations (N_cluster $=$ 10)





degree 3 Polynomial Library

	1	x1	x2	x1^2	x1 x2	x2^2	x1 ^3	x1^2x2	x1x2^2	x2^3
x1	0.	0.07	0.	-0.0349	0.	0.0963	-0.101	0.14439	-0.14516	1.9607
x2	0.	0.0346	-0.1022	0	0.053	0	-1.9586	0.08993	-0.083255	-0.0536

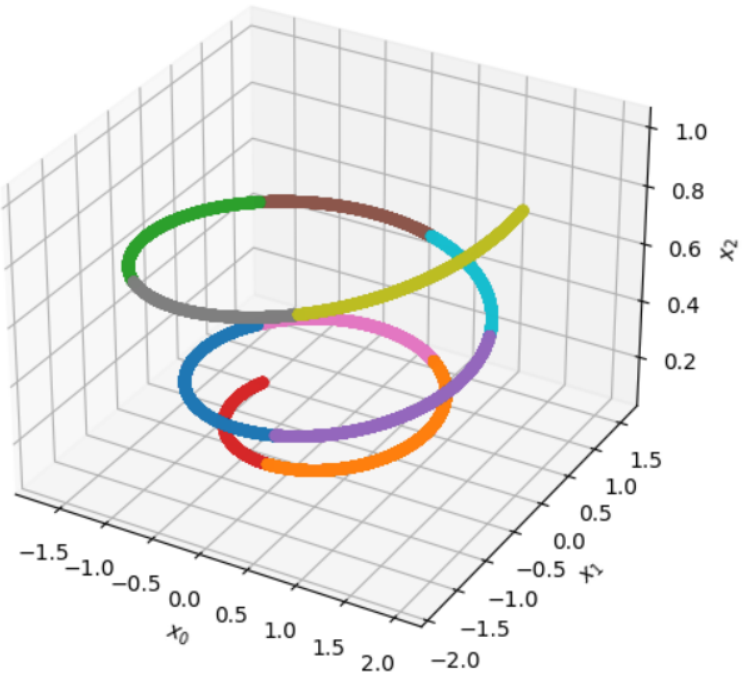
3.3 Linear 3D

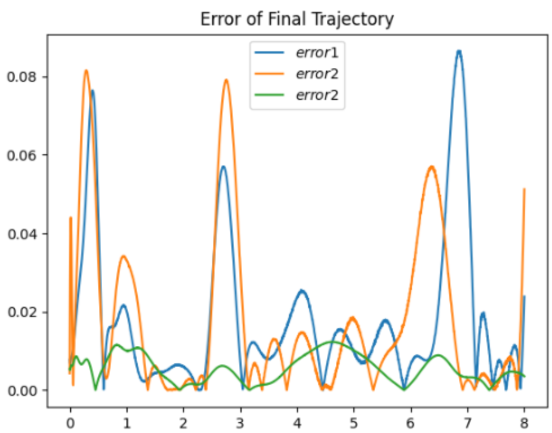
A linear system with three variables and the sparse approximation are shown below. In this case, the dynamics are given by

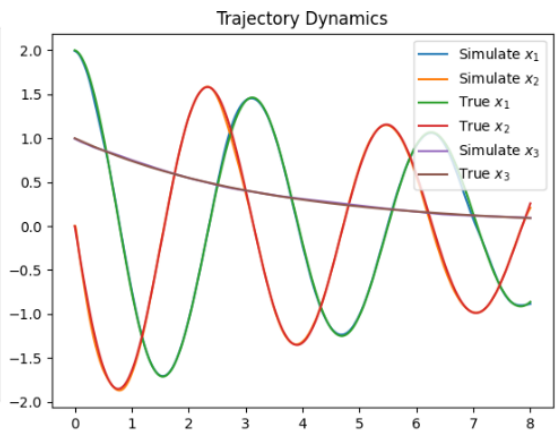
$$\frac{d}{\mathrm{dt}}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -0.1 & -2 & 0 \\ 2 & -0.1 & 0 \\ 0 & 0 & -0.3 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3}$$

Here we consider T $=$10, and the initial condition is $[2, 0, 1]$,We generate 50000 samples using RK4 method and sorted uniform time label $\{t_i\}$.

Using clustering method we can obtaim segmentations (N_cluster = 10)



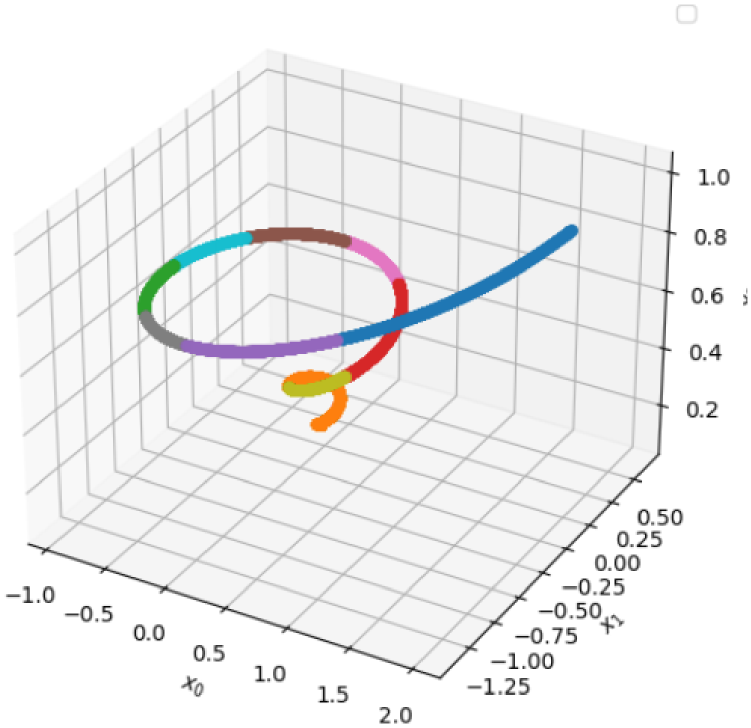


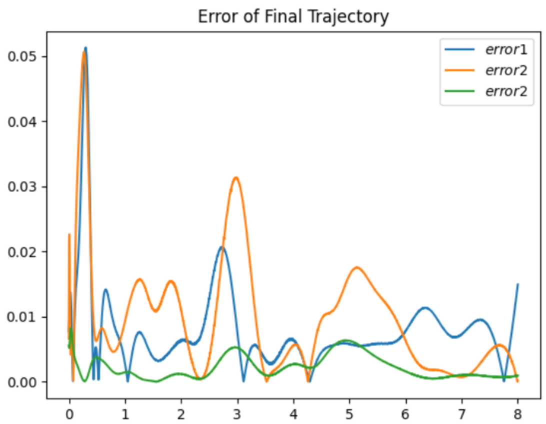


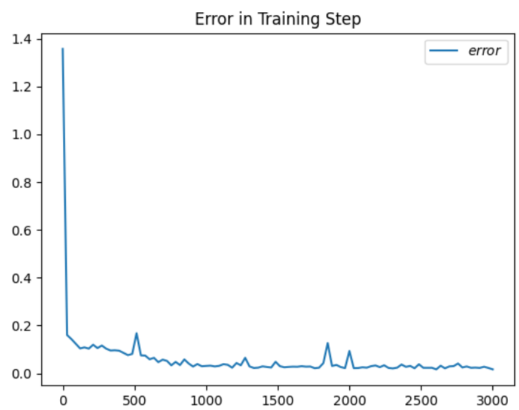
	1	x1	x2	x3
x1	0.	-0.0951	1.982	0
x2	0.	-2.0187	-0.1044	0
x3	0.	0.	0.	-0.2942

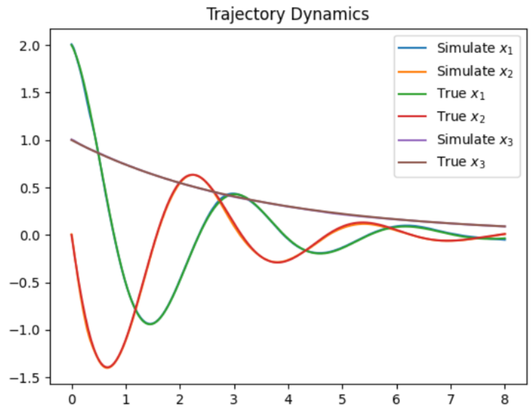
	1	x1	x2	x3	x1^2	x1x2	x1x3	x2x2	x2x3	x3^2
x1	0.	0.	2.0065	0						
x2	0.	-1.9919	0.	0						
x3	0.	0.	0.	-0.2919						

$$\frac{d}{\mathrm{dt}}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -0.5 & -2 & 0 \\ 2 & -0.5 & 0 \\ 0 & 0 & -0.3 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4}$$









	1	x1	x2	x3	x1^2	x1x2	x1x3	x2x2	x2x3	x3^2
x1	0.	-0.52	1.97	0						
x2	0.	-2.03	-0.493	0						
x3	1.98	0.	0.	-0.304						

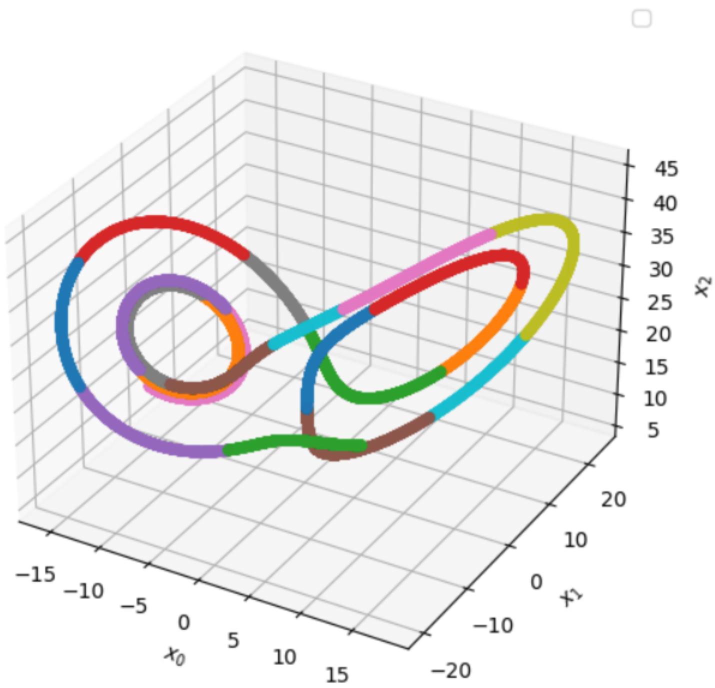
3.4 Lorentz

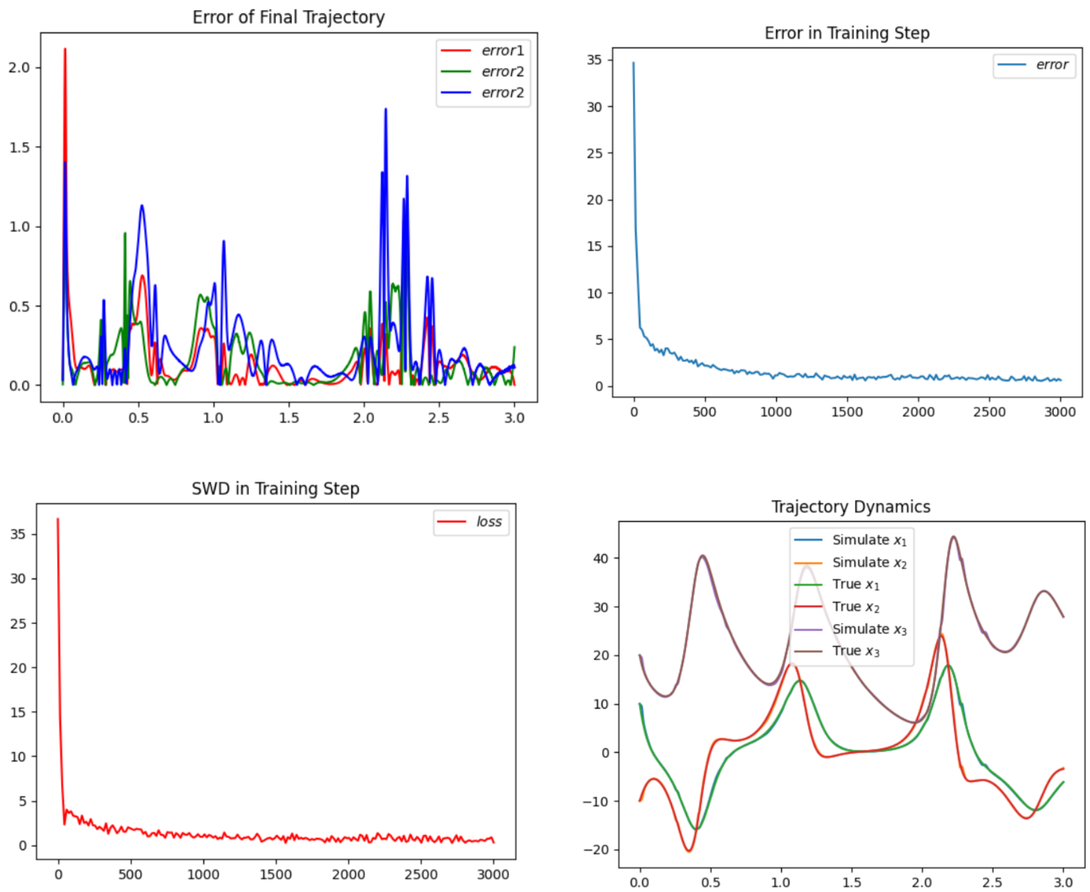
Here we consider the nonlinear Lorenz system to explore the identification of chaotic dynamics evolving on an attractor, shown in

$$\begin{aligned} \dot{x} &= \sigma(y-x) \\ \dot{y} &= x(\rho-z)-y \\ \dot{z} &= \mathrm{xy}-\beta z \end{aligned} \tag{5}$$

Here we consider T =3, and the initial condition is $[-8,7,27]^{\mathrm{T}}$,We generate 50000 samples using RK4 method and sorted uniform time label $\{t_i\}$.For this example, we use the standard parameters $\sigma=10$, $\beta=\frac{8}{3}$, $\rho=28$.

Using clustering method we can obtaim segmentations (N_cluster = 10)





	1	x1	x2	x3	x1^2	x1x2	x1x3	x2x2	x2x3	x3^2
x1	0.	-9.80	9.78	0						
x2	0.	27.88	-0.67	0			-1.01			
x3	1.98	0.	0.	-2.689		0.9721				