

LINMA1731 - Stochastic processes: estimation and prediction

Project : Tracking a Lorenz particle

1 - Theoretical Questions

If we have the Lorenz system defined by the following equations:

$$\begin{cases} \dot{x} = \sigma(y-x) \\ \dot{y} = x(\rho-z)-y \\ \dot{z} = xy-\beta z \end{cases}$$

where x , y and z stand for the position coordinates of the point in the system and σ , ρ and β are positive real parameters. Lorenz used the values $\sigma = 10$, $\rho = 28$ and $\beta = \frac{8}{3}$ for which the system shows a chaotic behaviour.

1.1 - Empirical probability density function

Starting point at $(x, y, z) = (1, 1, 1)$ at time $t = 0$. The total time is $100s$ and we make step of time of $0.02s$. We have the following restriction for the domain of respectively x, y, z : $[-20, 20] \times [-30, 30] \times [0, 50]$.

a) To be able to compute the empirical PDF, we are going to divide the spatial domain into 3D cubic box of length $l = 5$. For exemple the interval x is then going to be : $x \in [-20, -15] \times [-15, -10] \times [-10, -5] \times [-5, 0] \times [0, 5] \times [5, 10] \times [10, 15] \times [15, 20]$. We will then have 8 intervals for x . We will then start from the point $(x_0, y_0, z_0) = (1, 1, 1)$ and for each step of time

$$t_k = k \times 0.02$$

with $k = 0, 1, 2, \dots, 5000$. We have the point

$$(x_k, y_k, z_k)$$

. The next step is computing in wich box is the point and increment the counter of the box. We will then compute the empirical PDF by dividing the number of points in each box ($N_{i,j,k}$) by the total number of points (N_{total}) iid :

$$P_{i,j,k} = \left(\frac{N_{i,j,k}}{N_{\text{total}}} \right)$$

If we project the empirical PDF on the different plan we obtain the following result :

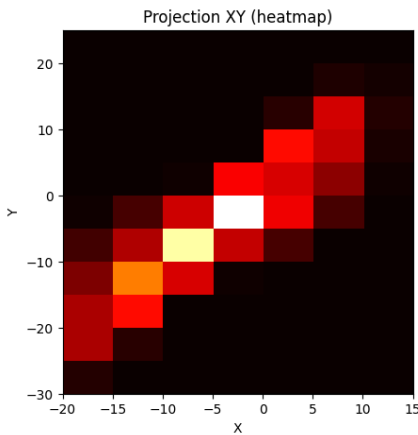


Fig. 1. – Plan XY

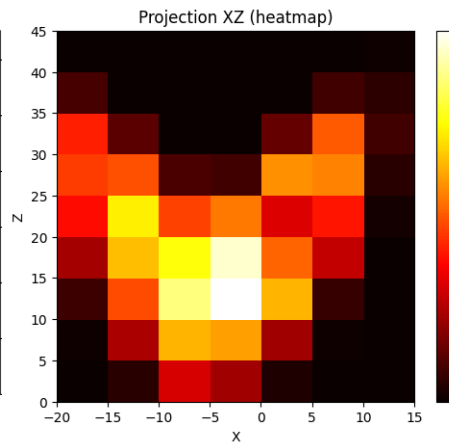


Fig. 2. – Plan XZ

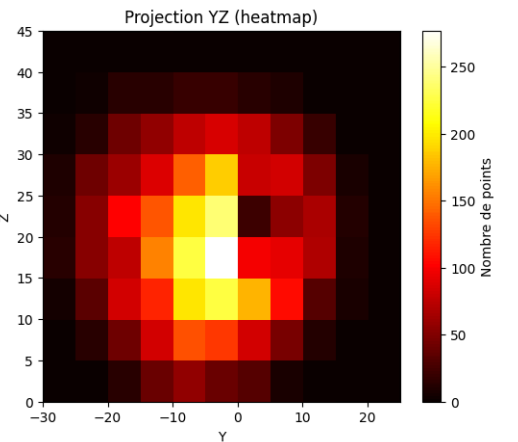


Fig. 3. – Plan XY

What we can see is that there is a majority of point in the middle of the domain. In the plan XY it's seems like we have a normal distribution. We can clearly observe the general shape of the system just by looking at the different plan.

b) To measure the distance between two empirical PDF, we made the choice to measure it by using the Kullback-Leibler divergence. The Kullback-Leibler divergence is a measure of how one probability distribution diverges from a second expected probability distribution. The Kullback-Leibler divergence is defined as follows¹ :

For two discrete probability distributions P and Q defined on the space X , the Kullback-Leibler divergence of P from Q is defined as:

$$D_{KL}(P\|Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}$$

where $P(x)$ and $Q(x)$ are the probabilities of the event x according to the distributions P and Q , respectively.

The second choice of distance is the distance of Bhattacharyya. The Bhattacharyya distance is a measure of the similarity of two probability distributions. It is defined as the negative logarithm of the Bhattacharyya coefficient, which is defined as follows² :

For two discrete probability distributions P and Q defined on the same probability space X , the Bhattacharyya distance between P and Q is defined as:

$$D_B(P, Q) = -\log \left(\sum_{x \in X} \sqrt{P(x)Q(x)} \right)$$

where $P(x)$ and $Q(x)$ are the probabilities of the event x according to the distributions P and Q , respectively.

c) To measure the impact of the parameters σ, ρ, β on the distribution. We have made a function where we can observe the evolution of the system of Lorenz by adjusting the parameters.

Influence of parameter σ :

- **Small** : The system tends to have a more regular behaviour. After a certain value of σ the system starts to have stable behaviour.
- **Big** : The general behaviour of the system is chaotic. After a certain value of σ the system tends to have a special behaviour

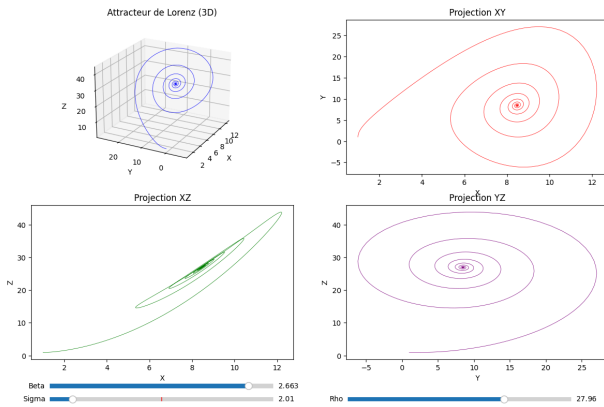


Fig. 4. – Small σ

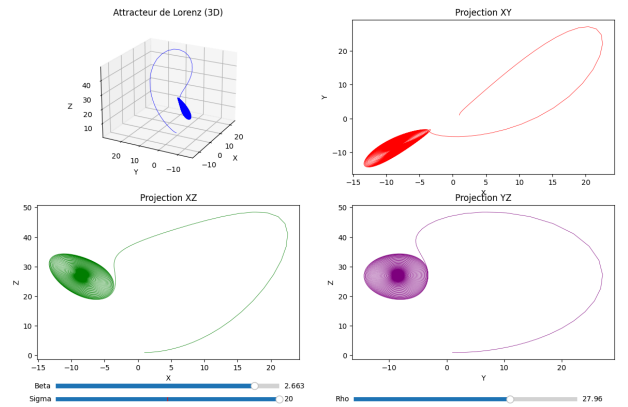


Fig. 5. – Big σ

Influence of parameter ρ :

- **Small** :

The system converges to a stable point. The system is not chaotic.

- **Middle** :

The system is oscillating between two points. The system is not chaotic.

¹https://fr.wikipedia.org/wiki/Divergence_de_Kullback-Leibler

²https://fr.wikipedia.org/wiki/Distance_de_Bhattacharyya

- **Big :**

The system is chaotic. We can not predict the future position of the system.

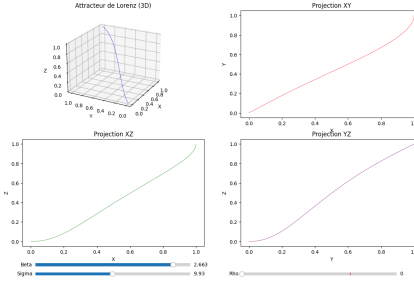


Fig. 6. – Small ρ

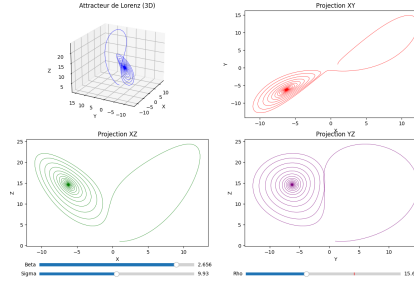


Fig. 7. – Middle ρ

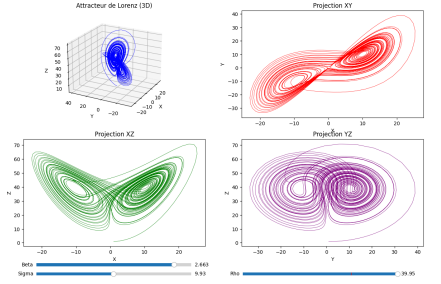


Fig. 8. – Big ρ

Influence of parameter β :

- **Small :**

The system tend to be less chaotic.

- **Big :**

The chaos can be amplified. The system is more chaotic.

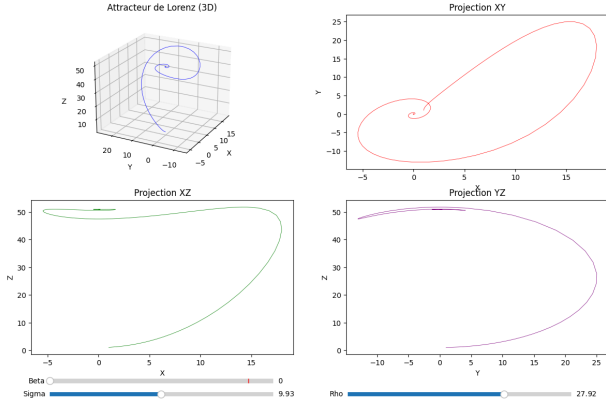


Fig. 9. – Small β

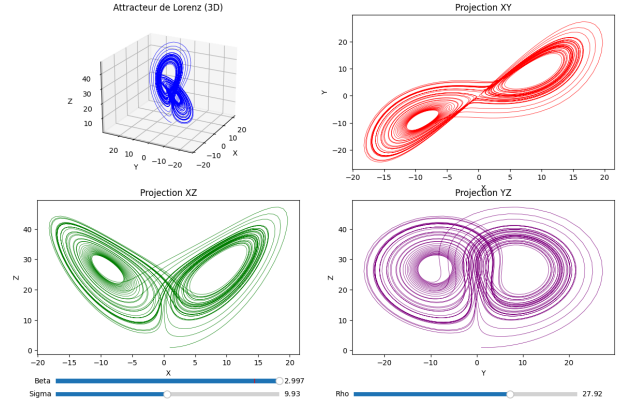


Fig. 10. – Big β

If we mesure the distance between our empirial PDF and a new one calculated with different paramters ($\sigma = 5, \rho = 20, \beta = 2$) we can see that the distances are not the same. In term of number we can see that the distance of KL divergence is 18.13 and the distance of Bhattacharyya is 4.51. The distance of KL divergence is bigger than the distance of Bhattacharyya. The distance of Bhattacharyya is smaller because she is not impacted as mush as the KL divergence by the presence of 0 in one distribution and not in the other one. As the twee distance are big with respect to the value that they can take, we can say that the two empirial PDF are not really close.

d) If we mesure the distance between our empirial PDF and the same one but with an other start point $((x_0, y_0, z_0) = (10, 10, 10))$ The distance is bigger for the Kullback-Leibler divergence than for the Bhattacharyya distance. In term of number we have a distance of 0.896 for the KL Divergence and a distance of 0.0448 for the Bhattacharyya distance. As the twee distance are small with respect to the value that they can take, we can say that the two empirial PDF are really close.

In conclusion for this part we can see that the initial state does not impact as mush as the paramter of the system. The distance between the two empirial PDF is smaller when we change the initial state than when we change the parameters of the system.

1.2 - Particle Filter

If we have a dynamic state space model :

$$\begin{cases} x_t = f(x_{t-1}, v_t) \\ y_t = g(x_t, w_t) \end{cases}$$

where $x_t \in \mathbb{R}, y_t \in \mathbb{R}$ and v_t, w_t are the noise. Our goal is to estimate x_t using observations $y_{1:t} = \{y_1, y_2, \dots, y_t\}$. Since we don't know exactly $p(x_{0:t}|y_{1:t})$, we are going to approximate it using a set of weighted samples. That bring some problems as over time, some particles have very low weights, while a few dominate. This leads to weight degeneracy, where most particles contribute very little to the estimate. To fix this, we perform resampling, which eliminates low-weight particles and duplicates high-weight ones. Here we are going to compare the performance of three type of resampling based on there complexity and variance. We are then going to compute $var(N_t^i)$

- Multinomial Resampling :

As the N_t^i are computed from a multinomial distribution with paramters N and \tilde{w}_t^i we have directly that the variance is given by : $var(N_t^i) = N * \tilde{w}_t^i * (1 - \tilde{w}_t^i)$.

- Residual Resampling :

The first syep is to compute the deterministic part, so for each particle i , we are going to compute a integer $\tilde{N}_t^i = \lfloor N \tilde{w}_t^i \rfloor$. By doing this we ensure that particle with a high weight gets a certain number of copies without any randomness. After having computed the deterministic part, there are still $\bar{N}_t = N - \sum_{i=1}^N \tilde{N}_t^i$ particles that don't have a weight. For these particles we are going to apply a procedure called SIR (Sampling Importance Resampling) wich give us the following new weights : $w_t'^{(i)} = \bar{N}_t^{-1} (\tilde{w}_t^{(i)} N - \tilde{N}_t^i)$ The total number N_t is then given by the sum of the deterministic part and the random part. As the deterministic part is deterministic, we have that the variance of the deterministic part is 0. The variance is then given by the variance of the random part which is given by : $var(N_t^{(i)}) = \bar{N}_t w_t'^{(i)} (1 - w_t'^{(i)})$ By doing some more computation we can see that the variance is smaller than the one of the multinomial resampling : As $var(N_t^{(i)}) = \bar{N}_t w_t'^{(i)} (1 - w_t'^{(i)}) = \bar{N}_t \bar{N}_t^{-1} (\tilde{w}_t^{(i)} N - \tilde{N}_t^i) (1 - \bar{N}_t^{-1} (\tilde{w}_t^{(i)} N - \tilde{N}_t^i))$.

If we compare to the multinomial resampling. The term $\tilde{w}_t^{(i)} N - \tilde{N}_t^i$ is smaller than $N * \tilde{w}_t^{(i)}$ because we remove \tilde{N}_t^i . The part $1 - \bar{N}_t^{-1} (\tilde{w}_t^{(i)} N - \tilde{N}_t^i)$ is also smaller than $1 - \tilde{w}_t^{(i)}$ because we remove \tilde{N}_t^i and we divide by \bar{N}_t . So we can conclude that the variance of the residual resampling is smaller than the one of the multinomial resampling.

- Systematic Resampling :

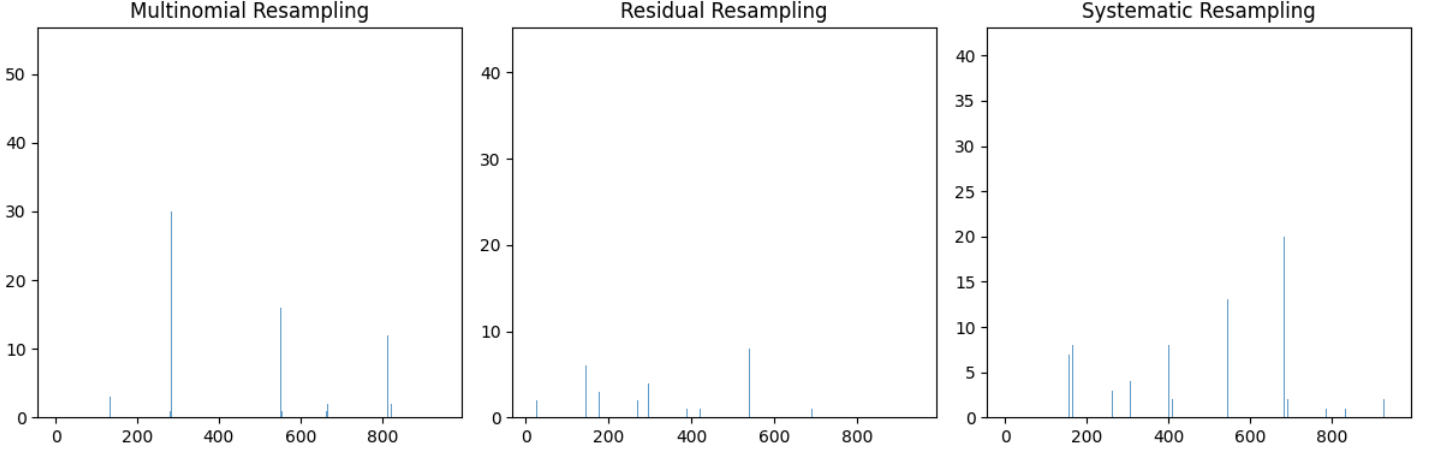
If we define the cumulative sequence : $C_i = \sum_{j=1}^i \tilde{w}_t^{(j)}$ with $C_0 = 0$ and $C_M = 1$. We have that the cumulative sums partition the interval $[0, 1]$ into M subintervals: $(C_0, C_1], (C_1, C_2], \dots, (C_{M-1}, C_M]$. Each of these interval have a length of $\tilde{w}_t^{(i)}$. As we sample a set $U = \{u_1, u_2, \dots, u_N\}$ of N points in $[0, 1]$, if we count the number of points $N_t^{(i)}$ iid : $N_t^{(i)} = \#\{u_k \in U : C_{i-1} < u_k \leq C_i\}$ In a pure multinomial resampling scheme, the variance in the number of offspring $N_t^{(i)}$ would be : $var(N_t^{(i)}) = N * \tilde{w}_t^{(i)} (1 - \tilde{w}_t^{(i)})$.

However, because systematic or stratified sampling forces an even spread of points, the variance is further reduced. For this strategy, the variance becomes $var(N_t^{(i)}) = \bar{N}_t w_t'^{(i)} (1 - \bar{N}_t w_t'^{(i)})$.

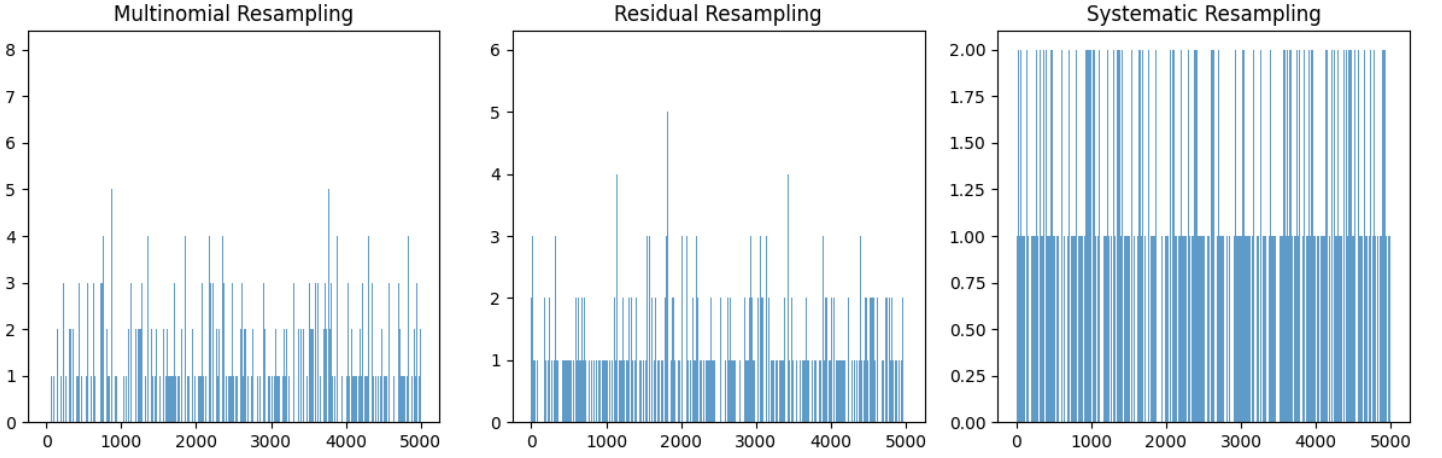
As we can see it is smaller than the one of the Systematic resampling as we substract $\bar{N}_t w_t'^{(i)}$ wich is greater than $\tilde{w}_t^{(i)}$.

Type of sampling	Variance	Time
Multinomial Resampling	3.6303	0.000070s
Residual Resampling	3.1031	0.000060s
Systematic Resampling	3.0735	0.000040s

By applying the three differents method of resampling we can see that the systematic resampling is clearly the best one in term of variance and time. To compute these data we have used our sample from point 1.2. We can also plot the different repartion of the particles after resampling.



It is much clear on a random set of weights :



2 - Particle Filter Implementation for the Lorenz System

For this second part we have implemented the particle filter for the Lorenz system. Our goal is to estimate the position of the particle in the system for each time step. To do such we have used the following discretization of the Lorenz system :

$$\begin{cases} x_t = x_{t-1} + \frac{h}{6}(f_1(x_{t-1}) + 2f_2(x_{t-1}) + 2f_3(x_{t-1}) + f_4(x_{t-1})) + v_t \\ y_t = x_t + w_t \end{cases}$$

where we assume that $x_t = [x(t), y(t), z(t)]^T$, and

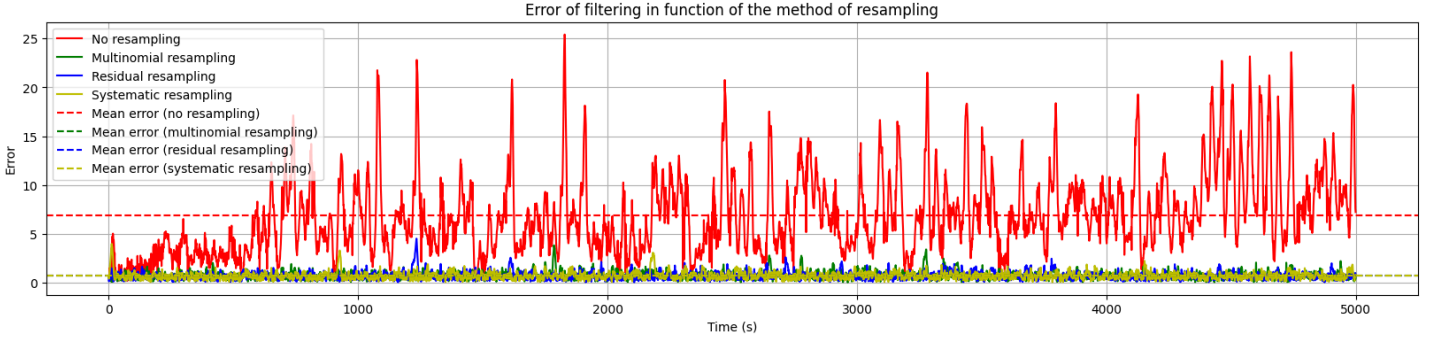
$$\begin{pmatrix} \sigma(y(t) - x(t)) \\ x(t)(\rho - z(t)) - y(t) \\ y(t)x(t) - \beta z(t) \end{pmatrix} = f(x_t)$$

where $f_1 = f(x_t)$, $f_2 = f(x_t + h\frac{f_1}{2})$, $f_3 = f(x_t + h\frac{f_2}{2})$, $f_4 = f(x_t + hf_3)$ and v_t, w_t are respectively the disturbances and the noise.

2.1 - Analyze of the result

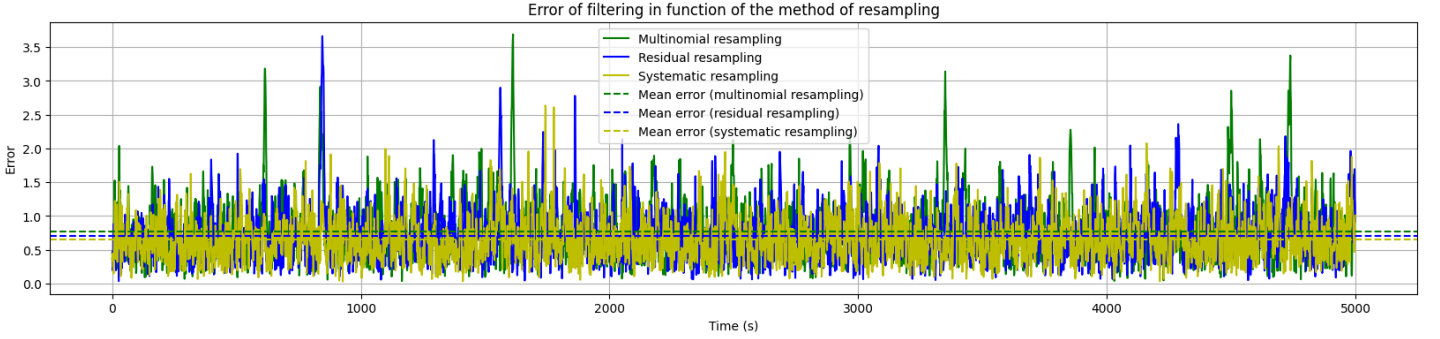
Impact of resampling

To Analyze our result we have begin by analyzing the impact of the resampling with $v_t = 0.1$, $h = 0.02$ and $N = 10$ with N the number of particles. We have plotted here the **error of the filtering** i.e. $(\|x_{\text{true}}(t) - x_{\text{predicted}}(t)\|)$ for the three resampling methods and for the case without resampling.

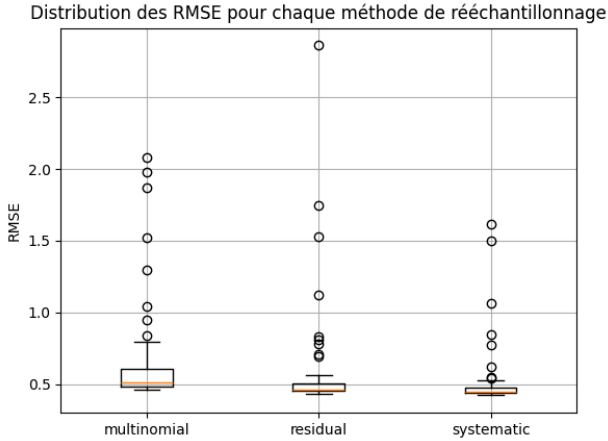


From the top graph, we clearly observe that the **mean error** is significantly lower when using any resampling method compared to not resampling. This confirms that applying resampling improves the accuracy of the particle filter by mitigating weight degeneracy.

To compare the resampling methods, we excluded the « no resampling » case in the second graph. This allowed us to better distinguish the relative performance of the three resampling strategies. Among them, systematic resampling provides the lowest mean error, indicating better long-term tracking accuracy.



Advance analyse



To assess the impact of resampling strategies on particle filter performance and corobore our previous analyse, we ran 50 independent simulations for each of the three methods: **multinomial**, **residual**, and **systematic**. The figure below presents the distribution of RMSE (Root Mean Squared Error) for each method. From the boxplot, we observe that systematic resampling consistently yields the lowest RMSE and exhibits the smallest spread, indicating both high accuracy and strong stability. Residual resampling performs similarly, with slightly higher variability and a few more outliers. In contrast, multinomial resampling shows a broader RMSE distribution and several high-error outliers, confirming its known tendency to introduce higher variance due to random sampling.

The visual evidence and lower median RMSE suggest that systematic resampling is the most reliable method, especially in a chaotic system like the Lorenz attractor. Based on these results, we adopted systematic resampling for the rest of our experiments.

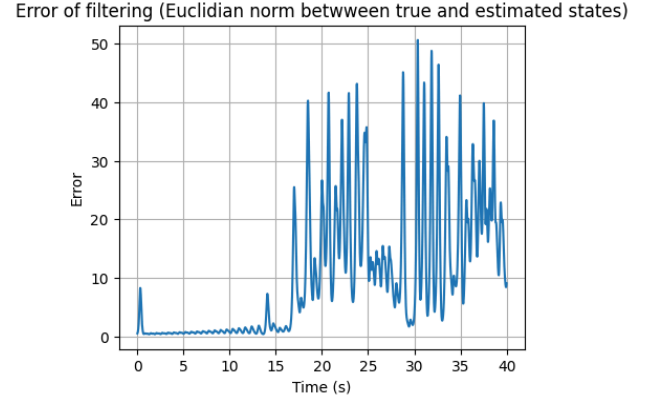
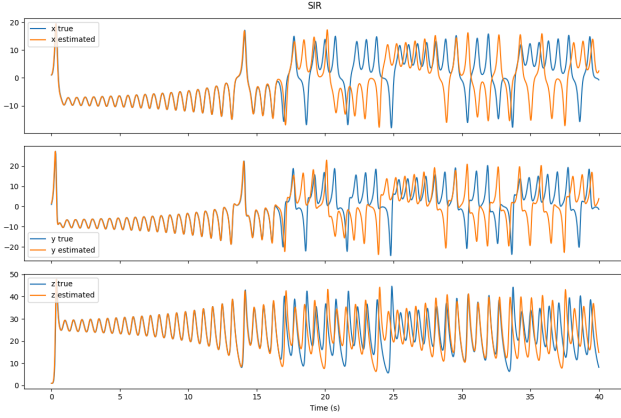
Impact of the process noise (v_t)

This set of figures illustrates the influence of process noise (v_t) on the performance of the particle filter when estimating the Lorenz system states. The true states (x, y, z) are compared to the filter's estimates across three different values of v_t : 0, 0.1, 10. In each case, the error is computed as the Euclidean distance between

the estimated and true states. The parameter that we have for all of our analyse is $w_t = 1$, $h = 0.02$ and the systematic resampling method as it is the better one.

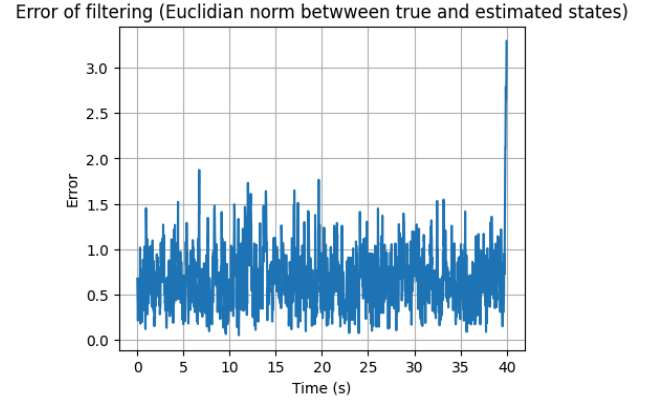
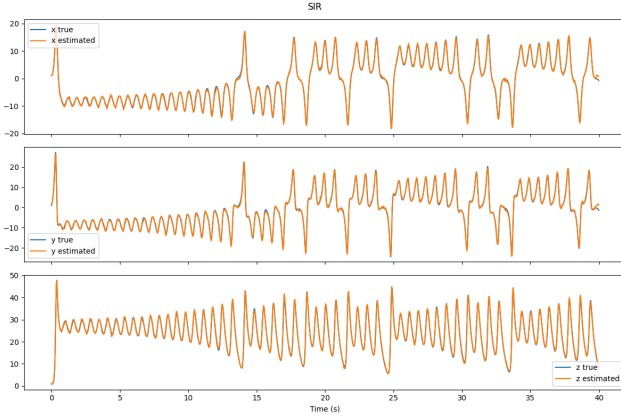
- $v_t = 0$

The predicted trajectories (orange) visibly diverge from the true ones (blue) after around 15 seconds. The estimation error rises steeply and reaches high values (up to 50), especially from 20 to 40 seconds. With no process noise, the filter assumes the dynamics are perfectly known. In a chaotic system like Lorenz, even a small modeling or integration error accumulates rapidly. As a result, the particles fail to stay near the true trajectory, causing a significant loss of accuracy. We can then conclude that a complete absence of process noise makes the filter too confident, which leads to an inability to recover once divergence begins.



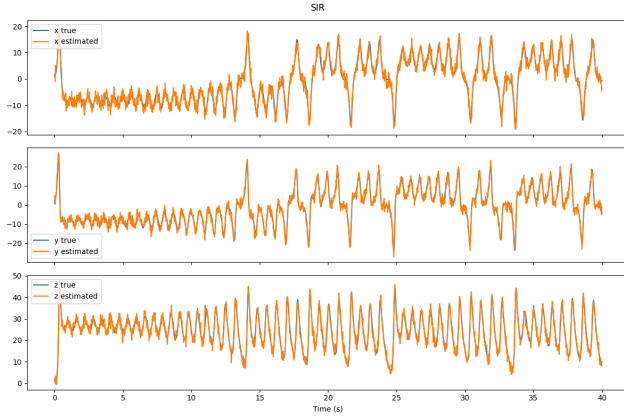
- $v_t = 0.1$

The estimated trajectories now closely follow the true ones across the whole time window. The corresponding error plot shows that the estimation remains consistently low, around 1 to 2, without major spikes. Adding moderate process noise introduces variability among particles, helping them explore the state space better and remain close to the true trajectory even when uncertainty or nonlinearity increases. This is a well-calibrated value of v_t , balancing model trust and adaptability, which enables the filter to accurately track the system.

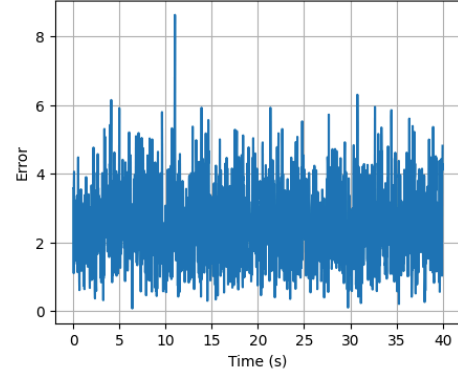


- $v_t = 10$

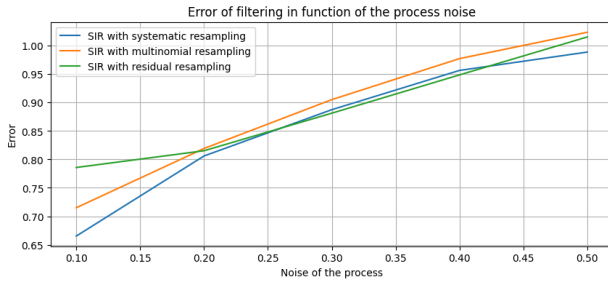
The estimated trajectories still follow the general pattern of the true states, but the error increases slightly and becomes more variable compared to the $v_t = 0.1$ case. The Euclidean error oscillates between 2 and 8. While a high process noise allows for greater flexibility, it also introduces excessive uncertainty, making the filter less confident and more reactive to noise. The estimates become noisier and less stable. With excessive process noise, the filter remains adaptive but loses precision. This results in noisy but still bounded tracking, which may be acceptable depending on the application.



Error of filtering (Euclidian norm between true and estimated states)



General error

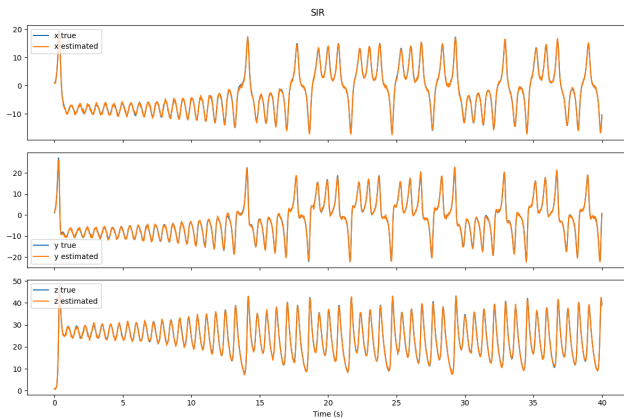


To better understand how the process noise affects the performance of the particle filter, we plotted the filtering error as a function of the process noise magnitude for different resampling methods. As expected, the error increases with the amount of noise added, since higher process noise introduces more uncertainty but also increases the flexibility of the model. This trade-off affects the tracking performance depending on the resampling strategy used.

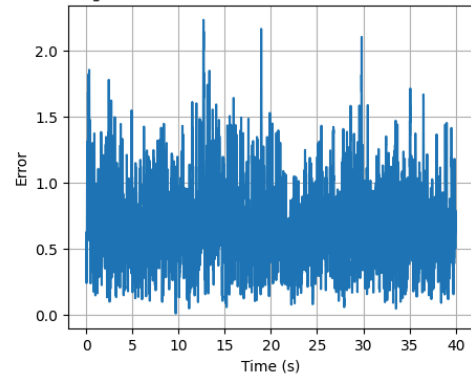
Impact of the time step (h)

As for the process noise this set of figures illustrates the influence of time step (h) on the performance of the particle filter when estimating the Lorenz system states. The time step h in the Runge-Kutta discretization of the Lorenz system determines the precision of the numerical integration. The true states (x, y, z) are compared to the filter's estimates across three different values of h : 0.01, 0.02, 0.03. In each case, the error is computed as the Euclidean distance between the estimated and true states. The parameter that we have for all of our analyse is $w_t = 1$, $v_t = 0.1$ and the systematic resampling method as it is the better one.

• $h = 0.01$

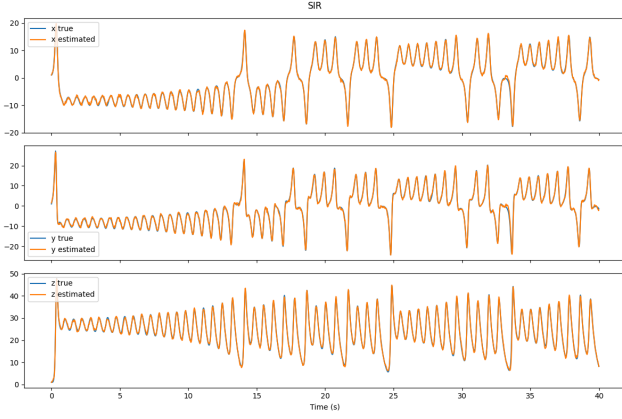


Error of filtering (Euclidian norm between true and estimated states)

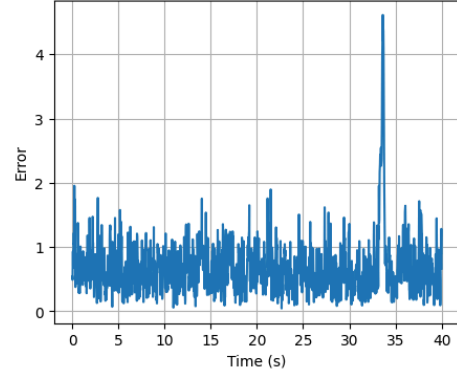


The estimated trajectory follows the true trajectory very closely throughout the simulation. The estimation error remains extremely low and stable. This result is expected as a smaller time step yields a more accurate numerical approximation of the continuous dynamics.

• $h = 0.02$

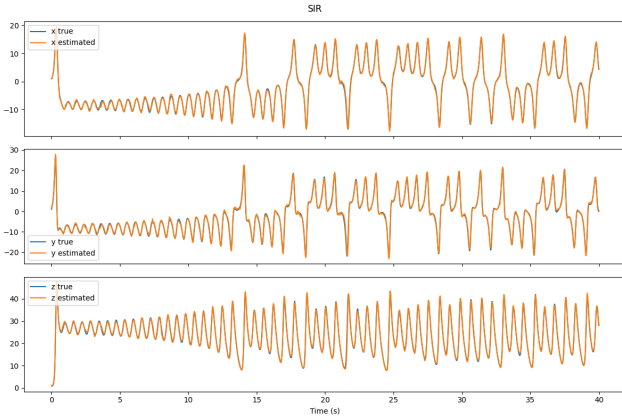


Error of filtering (Euclidian norm between true and estimated states)

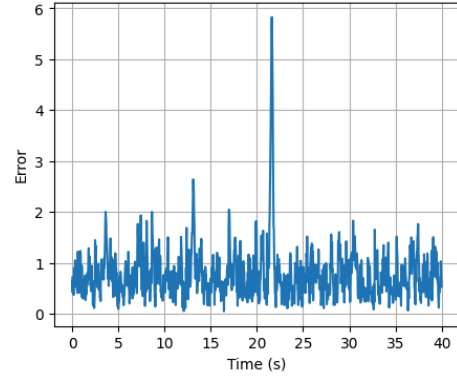


The performance remains good, with only a slight increase in estimation error. The filter is still able to track the true trajectory well. This value represents a good trade-off between integration accuracy and computational efficiency.

- $h = 0.03$

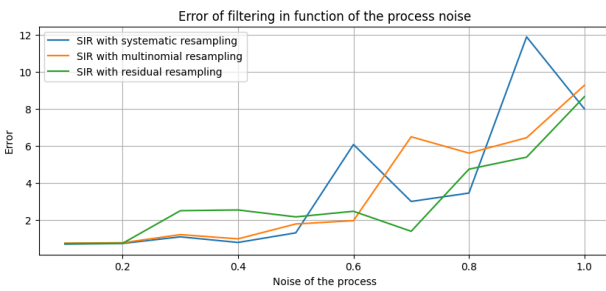


Error of filtering (Euclidian norm between true and estimated states)



The error increases more noticeably. The predicted trajectory begins to drift from the true one, especially during rapid transitions. The error becomes less stable and reflects the impact of coarser discretization on prediction quality.

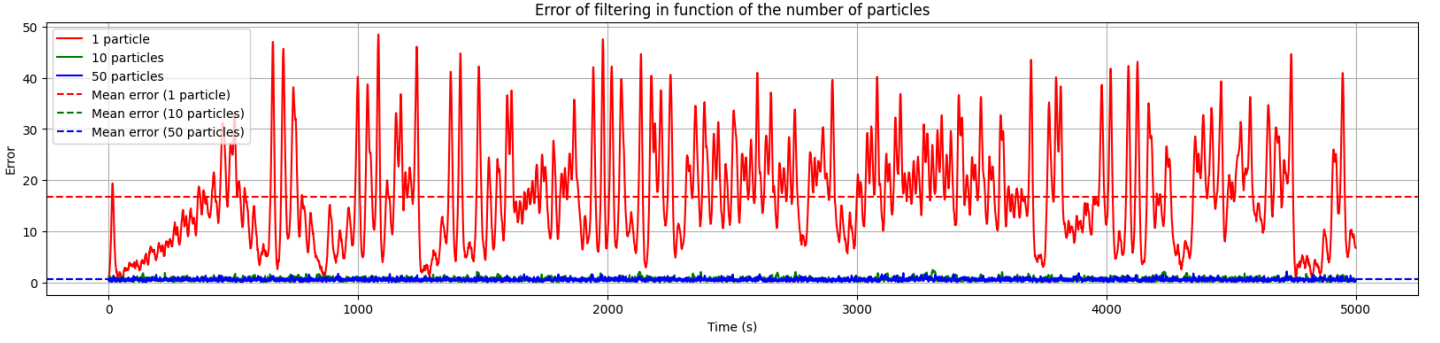
- **General error**



In conclusion, smaller values of h improve filter accuracy but also increase computational cost. The value $h = 0.02$ appears to offer a good balance for accurate and efficient tracking of the Lorenz system.

Impact of the number of particles N :

The number of particles N is critical in the particle filter, as it determines the granularity of the approximation to the posterior distribution. We examined $N : 1, 10, 50$, using fixed parameters: $v_t = 0.1$, $w_t = 1$, $h = 0.02$, and systematic resampling.



When $N = 1$, the filter fails to track the system properly. A single particle cannot represent uncertainty or adapt to the system's stochastic nature. As a result, the estimated trajectory diverges quickly from the true state, and the error becomes very large.

At $N = 10$, the filter performs significantly better. The trajectory estimation becomes much more accurate, although some fluctuations in error are still observed due to the limited number of samples.

When increasing to $N = 50$, the filter's performance improves further. The estimated trajectory closely matches the true trajectory, and the estimation error remains consistently low and stable over time. The higher particle count allows better representation of the system's uncertainty and more reliable state estimation.

In summary, increasing the number of particles leads to better accuracy and stability. However, this comes at the cost of increased computation time. In our experiments, $N = 50$ provided excellent results with a manageable computational load, making it a suitable choice for chaotic systems like Lorenz.

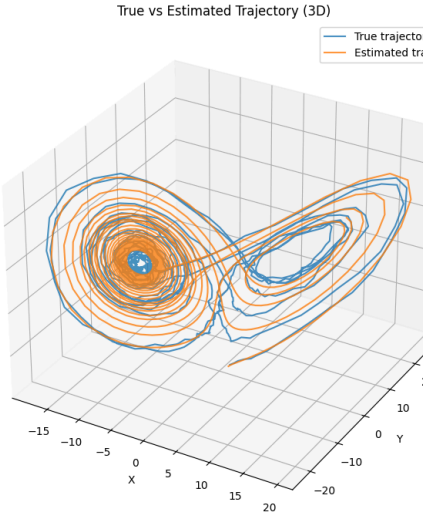


Fig. 26. – $N = 1$

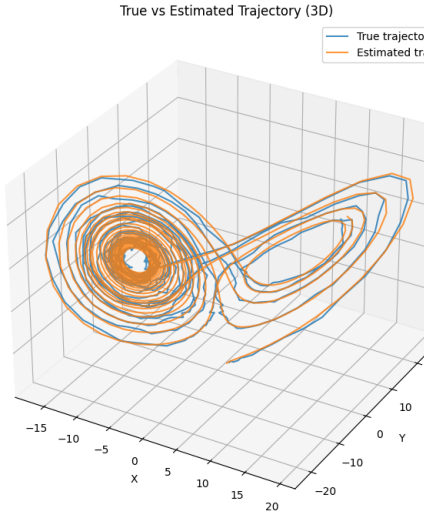


Fig. 27. – $N = 10$

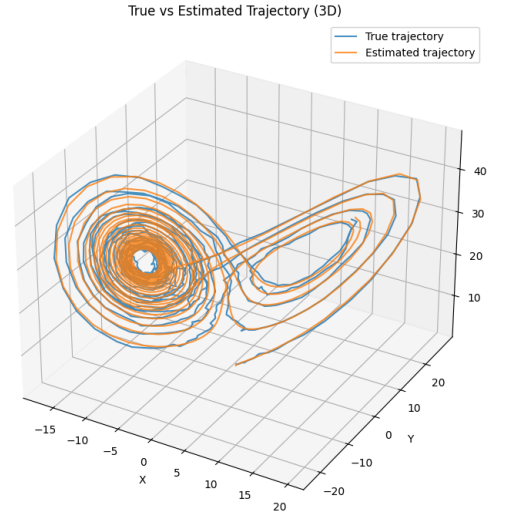


Fig. 28. – $N = 50$

3 - Conclusion

Our results clearly show that resampling is essential to prevent weight degeneracy and maintain estimation accuracy over time. Among the tested methods, systematic resampling consistently provided the lowest error and highest stability, making it the most effective choice.

We also demonstrated that a moderate level of process noise helps the filter adapt to uncertainty in the system, whereas the absence of process noise can lead to divergence. Similarly, the time step h significantly affects the integration accuracy of the dynamic model. A smaller step improves precision but increases computational cost, with $h = 0.02$ offering a good balance.

Finally, we observed that increasing the number of particles N leads to better state estimation, reducing the variance and error of the filter. However, this improvement must be weighed against computational efficiency.