ÉCOLE POLYTECHNIQUE DE LOUVAIN

LINMA1731 - Project 2019

Fish School Tracking

Année Académique 2018-2019
LINMA1731

Fievez Thibault - 2545-1500
Santacatterina Simon - 2156-1500

1 Introduction

The goal of this project is to study the movement of one or several fishes in a fish school in order to track them. To do so, the work will be divided into two parts. Firstly, the motion of fishes in the school will be describe by deriving parameters characterising their trajectories and analysing their behavior in term of mean, standard deviation and convergence of the covariance to a lower bound when the sample size increases. And in the second part, individual fishes will be tracked using filtering techniques.

2 Study of the fishes motion

The motion of fishes is unpredictable in a pool, their speed is therefore a random variable. We assume that this r.v. follows a known parametric data model which is a Gamma distribution. This distribution depends on two parameters : a shape parameter $k > 0$ and scale parameter $s > 0$.

Using N empirical observation v_i (i = 1,...,N) of the random variable following a Gamma distribution and supposing these are i.i.d. the goal is to find the parameters estimators which best fit our observation. In a first approach, k is assumed to be known.

The studied distribution for the speed of a fish is a gamma function defined as $X \sim \Gamma(k,s) = \frac{1}{s^k\Gamma(k)}v^{k-1}e^{-\frac{v}{s}}$. Where $\Gamma(k)$ is the Euler's Gamma function.
To find the maximum likelihood of the estimator $\theta := s$ let's consider the product of all the N Gamma distrition of the i.i.d. v_i. We then get

$$\prod_i \Gamma(k,s)_{v_i} = \frac{s^{-nk}}{\Gamma(k)^n} v_1^{k-1} v_2^{k-1} v_3^{k-1} ... v_n^{k-1} e^{-\frac{(v_1+v_2+v_3+...+v_n)}{s}} \tag{1}$$

Now let's use the natural logarithm of the function in order to get a sum, let's call this function log L. [1]

$$\log L = (k-1)\sum_i \log v_i - \frac{1}{s}(v_1 + v_2 + v_3 + ... + v_n) - nk\log(s) - n\log(\Gamma(k)) \tag{2}$$

The max of Log L for the parameter of interest will give us the maximum likelihood, which is obtain by $\frac{\delta logL}{\delta s} = 0$. This gives the following result

$$\hat{S} = \frac{v_1 + v_2 + v_3... + v_n}{nk} = \frac{\sum_{i=1}^N v_i}{nk} \tag{3}$$

From this estimator, several properties can be highlighted. Firstly, the estimator is asymptotically unbiased. An asymptotically unbiased estimator is such that $lim_{N->\infty}E(g(V),\theta) = \theta$, in this case $\theta := s$. Indeed, as N tends to infinity, the number of observation v_i (i.i.d.) increases and using the Central Limit Theorem [2] on the sum of the random variables one can see that

$$lim_{N->\infty}E\left[\frac{\sum_{i=1}^N v_i}{n}\right] = \mu. \tag{4}$$

Where μ is the mean of the distribution from which the samples come from. Here it is a Gamma distribution thus $\mu = ks$. [3] Hence,

$$lim_{N->\infty}E\left[\frac{\sum_{i=1}^N v_i}{kn}\right] = \frac{ks}{k} = s. \tag{5}$$

Which is the condition to have an asymptotically unbiased estimator. [4]

Secondly, $\hat{S}$ is also an efficient estimator. The pdf given by equation 1 satisfying the "regularity" condition, the proof is given in the appendix 6.1, we can state that the estimator will be efficient if its variance reaches the Cramér-Rao bound. [4] By computing the Fischer information matrix, a scalar in this case, one has

$$I(s) = -E\left[\frac{\partial^2 \log L}{\partial s^2}\right] = -E\left[\frac{-2\sum_{i=1}^{N} v_i}{s^3} + \frac{nk}{s^2}\right]. \tag{6}$$

The proof for the derivative of first and second order of log L are given in appendix 6.2 in equation 33 and 38. And again, using the central limit theorem we have $E[\sum_{i=1}^{N} v_i] = nks$ equation 6 become

$$I(s) = -E\left[\frac{\partial^2 \log L}{\partial s^2}\right] = \frac{nk}{s^2}. \tag{7}$$

Therefore, the Cramér-Rao lower bound is I^{-1}(s) $= \frac{s^2}{nk}$. The variance of the estimator, Var($\hat{S}$) is such that

$$Var(\hat{S}) = Var\left(\frac{\sum_{i=1}^{N} v_i}{nk}\right) = Var\left(\frac{\sum_{i=1}^{N} \frac{v_i}{k}}{n}\right). \tag{8}$$

Using the central limit theorem on the variance of a sample this leads to

$$Var\left(\frac{\sum_{i=1}^{N} \frac{v_i}{k}}{n}\right) = \frac{1}{n}Var\left(\frac{v_i}{k}\right) = \frac{1}{nk^2}ks^2 = \frac{s^2}{nk}. \tag{9}$$

As the variance in equation 9 is equal to the Cramér-Rao lower bound I^{-1}(s) $= \frac{s^2}{nk}$, our estimator is termed efficient.

Thirdly, it can be shown that the estimator is asymptotically normal. Mathematically, [4]

$$\sqrt{N}(\hat{S}_N - s) \xrightarrow[N\to\infty]{D} \mathcal{N}(0, \Sigma). \tag{10}$$

In this equation, the only random quantity is $\hat{S}_N$, its behavior while N tends to infinity is what is relevant.The Lindeberg–Lévy CLT [5] could be directly used because E[$\hat{s}_N$] is equal to s and Var($\hat{s}_N$) $< \infty$ this would fulfill the conditions for equation 10 to happen while n tends to infinity. In another way, it can be said that when N become large, by the CLT, $\hat{S}_N$, made up of a normalized sum of random variable, converges into a normal law. This is also true for $\sqrt{N}(\hat{S}_N - s)$ which is just a linear transformation of $\hat{S}_N$ as linear transformation of a gaussian r.v. is also a gaussian. [6] The mean E[$\hat{s}_N$] is equal to s. Hence, subtracting it to $\hat{s}_N$ gives a new r.v. with 0 mean. And $Var(\sqrt{N}(\hat{S}_N - s)) = nVar(\hat{S}_N) = \frac{s^2}{k} > 0$. Thereby, we have equation (10) which is verified as the left quantity tends to converges to a normal $\mathcal{N}(0, \frac{s^2}{k})$.

And finally, this estimator is consistent. The condition for $\hat{s}$ to be consistent is $plim_{N->\infty}\hat{S} = s$. This can be rewritten as [4]

$$\forall \varepsilon > 0 : lim_{N->\infty}Pr[|\hat{s} - s| > \varepsilon] = 0. \tag{11}$$

Here, as in the previous paragraph, by using the CLT we can state that while N become large, $\hat{S}$ converge to a normal law $\mathcal{N}(s, \frac{s^2}{nk})$. Following this, we can do a standardized the nomal and $Z = \frac{\sqrt{nk}|\hat{S}-s|}{s}$ has a standard normal distribution and

$$lim_{N->\infty}Pr\left[\frac{\sqrt{nk}|\hat{S} - s|}{s} > \frac{\sqrt{nk}\varepsilon}{s}\right] = lim_{N->\infty}Pr\left[Z > \frac{\sqrt{nk}\varepsilon}{s}\right] \to 0 \tag{12}$$

$\forall \varepsilon > 0$, as N tends to infinity this tends to 0. Therefore, $\hat{s}$ is consistent for s.

In the following part of this report the k is not anymore considered as a constant and a maximum likelihood estimator is also developed for k as well as a method of moment for both k and s.

To find the maximum likelihood for the k parameter, it is necessary to use the same method as for s, by starting of with equation 2, and derivating it with respect to k in order to find the maximum. This gives

$$\frac{\delta logL}{\delta k} = \sum_i \log v_i - n\log s - n\frac{\Gamma(k)'}{\Gamma(k)} = 0 \tag{13}$$

as with $\Gamma(k)$ the gamma function. The fraction $\frac{\Gamma(k)'}{\Gamma(k)}$ is called a Digamma function noted $\psi(k) = \frac{\Gamma(k)'}{\Gamma(k)}$, to isolate k the reciprocal of this function would be needed. Also s can be replaced by its estimation in equation 3.
Solving this system would require numerical method such as Newton-Raphson and as such would require a starting point often found by using the moment method, however the result can be approximated by the following function [7] :

$$\hat{K} = \frac{n \sum v_i}{n \sum v_i lnv_i - \sum lnv_i \sum v_i} \tag{14}$$

The equation 14 will be use for simulation purposes in the some additional graphs that are in the appendix, section 6.4.

After having found the estimators using MLE method, let's find the estimate of the parameters by using the moments methods. Let's quickly remind ourself of how it works : let's consider d unknown parameters $\theta_1, \theta_2 ... \theta_d$ of a distribution $f_W(\theta_1, \theta_2 ... \theta_d)$ with W the random variable associated to that distribution. In our case, we have V associated with the gamma distribution of parameters k and s. We can express the x th order moment as a function of the parameters $\theta_1, \theta_2 ... \theta_d$ by applying the definiton :

$\mu_1 = E[X]$ order 1

$\mu_2 = E[X^2]$ order 2

$\mu_3 = E[X^3]$ order 3

...and so on .
And for n values drawn , which are $v_1, v_2, ..., v_n$ we then let (empirically) $\hat{\mu}_j = \frac{1}{n} \sum_{j=1}^{n} v_i^j$ with the goal being to find a solution to the system of equation above.
So when it come to our gamma distribution, for the method of moment of second order, one could find : theoretically the first moment is for a gamma given by [8]

$$E[X] = \mu_1 = ks \tag{15}$$

estimated empirically by

$$\hat{\mu_1} = \frac{v_1 + v_2 + ... + v_n}{n} \tag{16}$$

Theoretically the second moment is for a gamma given by

$$E[X^2] = \mu_2 = s^2 k(k+1). \tag{17}$$

estimated empirically by

$$\hat{\mu_2} = \frac{v_1^2 + v_2^2 + ... + v_n^2}{n}. \tag{18}$$

We therefore end up by solving the system (2 equations 7=8 and 9=10) with :

$$\hat{K} = \frac{\hat{\mu_1}^2}{\hat{\mu_2} - \hat{\mu_1}^2} \tag{19}$$

$$\hat{S} = \frac{\hat{\mu_2} - \hat{\mu_1}^2}{\hat{\mu_1}} \tag{20}$$

In general, those estimators are consistent but often biased, it is therefore better to use a maximum likelihood method, however in the case of the gamma function, the calculation for the likelihood function is difficult and requires the use of algorithms or approximations whereas the method of moment is straightforward. Gamma functions require numerical methods and algorithms in order to be solved such as Newton-Raphson, however this method requires a starting point that could be provided by using the moment method. Whereas, in the case of the moment method, when the sample is not large enough the CLT does not apply and the empirical moments do not get close enough from the actual theoretical moment which lead to wrong answers (figure 10 in the appendix).

The comparison between the MLE and moment method is showcased on the following figures $(1, 2)$. Those graphs depict the behaviours of the computed values of $\hat{S}$ and $\hat{K}$ when n increases for both method. The MLE uses equations 3 as well as 13[1] whereas the moment method uses equations 19 and 20. One could notice on figure 1 that the MLE gives a closer estimate of $\hat{S}$ with respect to it's actual value which is 2, as the relative distance between the estimated value and the real value is smaller for the red curve than for the blue curve. In other words , the closer it gets to 0 the more accurate it is, as it mean the estimate is getting closer to the real value. Also a smaller standard deviation can be observed on figure 2 for the MLE method compared to the Moments method and naturally decreases as N increases. Clearly it is noticeable on figure 3 that the distribution of the (500) estimate of $\hat{S}$ for a big sample size (1000 values) tend to be a Gaussian distribution. The same conclusion can also be made by analysing the graphs provided in section 6.3, displaying the mean of each estimated parameter and its standard deviation for both methods.

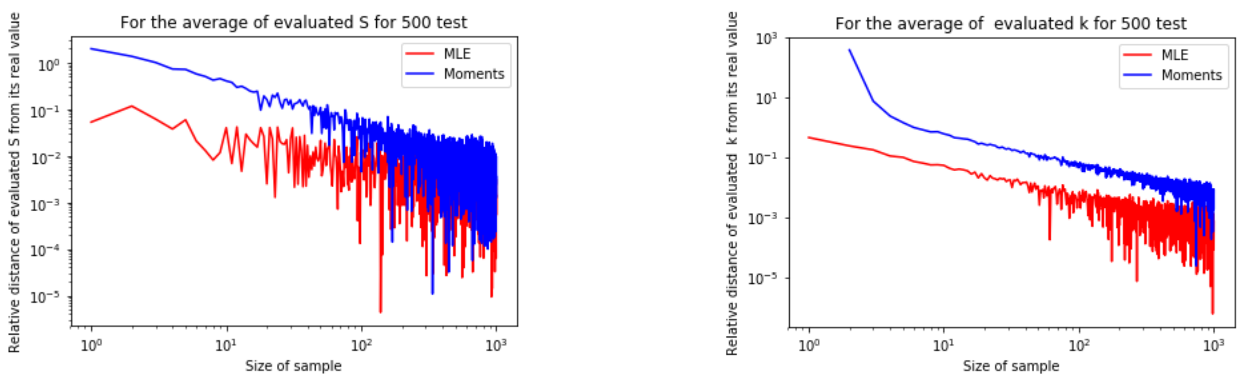

Figure 1 – Comparison of the average on 500 test evaluation of $\hat{S}$ on the left and $\hat{K}$ on the right with its real value. In red the MLE evaluation and in blue the Moment method

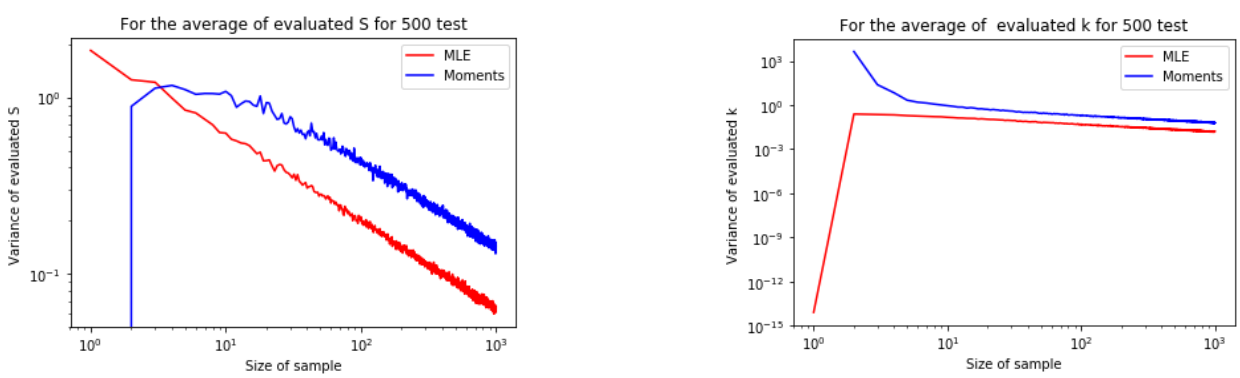

Figure 2 – Standard deviation of determined $\hat{S}$ on the left and $\hat{K}$ on the right. In red the MLE evaluation and in blue the Moment method

1. Results using the approximating formula 14 can be found in the appendix

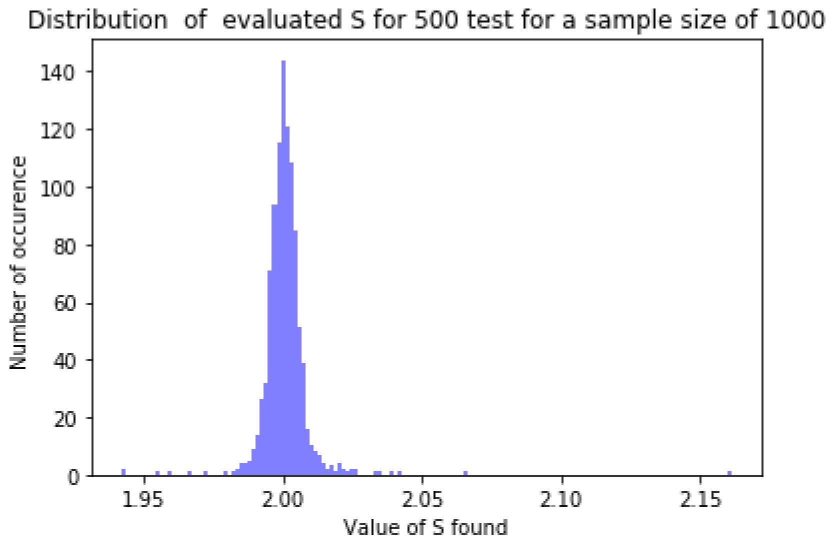

FIGURE 3 – Distribution of different values obtained for $\hat{S}$ and for $\hat{K}$ are clearly a gaussian distribution when the sample size is big enough (1000 here)

Finally, based on the estimand k and s, it is possible to check if the variance of the ML estimator tends to the Cramér-Rao lower bound when N increases. To do so, it is needed to derive the Fischer information matrix $I(\theta := (k\ s)^T)$. [9] This matrix is made up of elements $I_{j,k}(\theta) = -E\left[\frac{\partial^2 \log L}{\partial\theta_k \partial\theta_j}\right]$. Log L is given by equation 2.

$$I((k\ s)^T) = \begin{pmatrix} \frac{\partial^2 \log L}{\partial k^2} & \frac{\partial^2 \log L}{\partial k \partial s} \\ \frac{\partial^2 \log L}{\partial s \partial k} & \frac{\partial^2 \log L}{\partial s^2} \end{pmatrix} \tag{21}$$

After computing all the derivatives, detailed in this appendix 6.2, this gives

$$I((k\ s)^T) = \begin{pmatrix} n\Psi^{(1)}(k) & \frac{n}{s} \\ \frac{n}{s} & \frac{nk}{s^2} \end{pmatrix}. \tag{22}$$

The theoretical Cramér-Rao lower bound can therefore be derived from equation as it is $I^{-1}((k\ s)^T)$.

$$I^{-1}((k\ s)^T) = \frac{s^2}{n^2\Psi^1(k)k - n^2} \quad \begin{pmatrix} n\Psi^{(1)}(k) & \frac{n}{s} \\ \frac{n}{s} & \frac{nk}{s^2} \end{pmatrix} \tag{23}$$

We can therefore compute the ratio between $\mathrm{Cov}(\hat{\theta}_N)$ and the inverse fisher matrix $I^{-1}(\theta)$, if this ratio tends towards the identity matrix as N increases it means we are reaching the Cramér-Rao lower bound. To visualise this, this ratio for n =50, 150, 3000 for both the MLE and moment methods are displayed thanks to the figure 4. On this figure, to be able to to visualise if the matrix tends toward a diagonal full of ones, the values of the diagonal have been added and divided by 2. As one could expect, the MLE method tends to that whereas the MM doesn't.

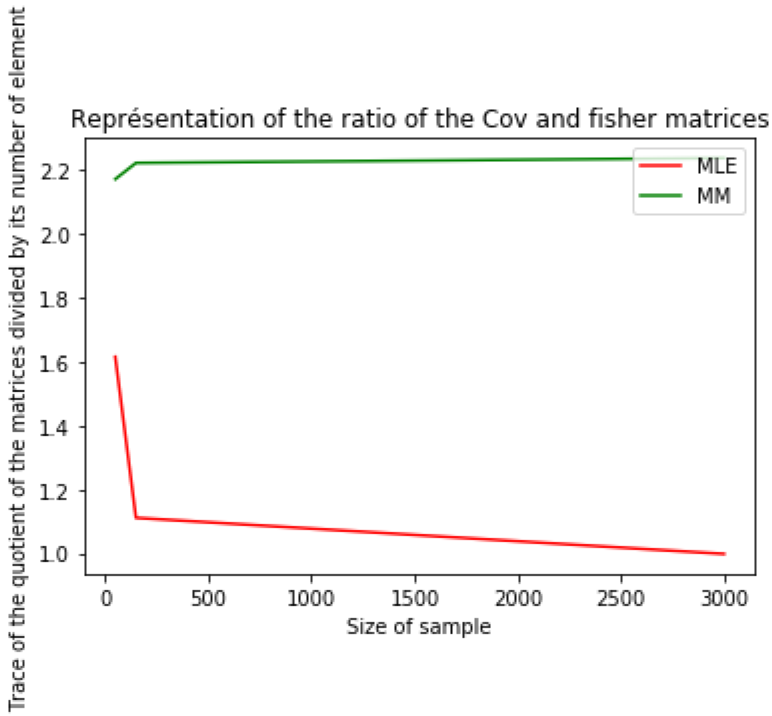

FIGURE 4 – Distribution of the trace of the ratio ,divided by 2 , for different N

3 First Part Conclusion

To conclude this first part of the report, noticeably the maximum likelihood method is superior to the moment method and gives overall better result for the mean and standard deviation as well as a better asymptotic properties. The only drawback is its computation time as it has already been stated it requires an iterative method to find a result for the paramater $\hat{K}$ whereas the other is easily implemented.

4 Particle Filtering

The second part of this report deals with fish tracking. To do so, our animals and environment are modelized as a 2D system, the aquarium being a 2D rectangle and the fish points in this rectangle. For this system, a discrete state model of our fish motions is created. This model is

$$\begin{cases} \mathbf{x}_p(i+1) = \mathbf{x}_p(i) + v_i^{(p)} \mathbf{o}_p(i) t_s \\ \\ \mathbf{o}_p(i+1) = \begin{bmatrix} \cos \alpha_p(i+1) & -\sin_p \alpha(i+1) \\ \sin \alpha_p(i+1) & \cos_p \alpha(i+1) \end{bmatrix} \mathbf{o}_p(i). \end{cases} \tag{24}$$

Where $\mathbf{o}_p(i)$ is the orientation of fish p (unitary norm) at time index i (with i = 1, . . . ,N), $\mathbf{x}_p(i)$ is its position, $\mathbf{V}_i^{(p)} i \sim \Gamma(k, s)$ as described in the first part of the report, and ts is the sampling period

From a sample of noisy observation we should complete this model. For this, the speed v_i of the fish at time index i (with i = 1, . . . ,$N-1$) need to be estimated. This is made by basic 2D geometry using fishes position to compute the speed

$$v_i = \frac{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}{t_s} \tag{25}$$

From this, some discussion can be made about the noise of those speed estimation. Firstly, the noise is Gaussian and so are the observations. And the $x_{i+1} - x_i$, a difference of two Gaussian distributed r.v. is a also Gaussian with distribution $\mathcal{N}(0, \sigma_{i+1}^2 + \sigma_i^2)$; a linear combination of Gaussian r.v. gives a Gaussian random variable. [6] Then, this new r.v. is raised to the square giving a χ^2 random variable with mean $\sigma_{i+1}^2 + \sigma_i^2$ and variance $2(\sigma_{i+1}^2 + \sigma_i^2)^2$. By summing the χ^2 for the x direction and the y we get a new χ^2.

Using this speed estimation the parameters k and s can be calculated similarly as done in the first part of the report, using the MLE function given by equation 1 . This is done with the **EstimateGamma** function of our code. This implementation gives the numerical values $k = 3.9541$ and $s = 0.3028$. Those two parameters will later on be used as an input for the function **GenerateObservation** and also **StateUpdate** to simulate the behaviour of the fish.

Form here, the tracking of the fish and predator is still not performed. What is needed is a way to recover the true positions of the fish from the noisy observation. The observation model being :

$$\mathbf{y}_p(i) = \mathbf{x}_p(i) + \mathbf{n}_p. \tag{26}$$

$\mathbf{y}_p(i)$ is a vector containing the noisy position of fish p at time i and $\mathbf{x}_p(i)$ are the real observation to be found. These observations are given by the function **GenerateObservations** which performs iterations using **StateUpdate** and adds $\mathbf{n}_p \sim \mathcal{N}(0, \sigma_{obs}^2)$. **GenerateObservations** returns the true (called x) and noisy position (called y).

A method to recover the true position is to use a particule filter called a Monte Carlo filter. This filter is such that we estimate iteratively the true values of x (in our case x is a vector of Cartesian coordinates X,Y for each fish or predator, but it also can be their orientation) based on the observation.

The Monte Carlo algorithm is a prediction-update algorithm. [10] And is performed using this step by step logic :

1. An initial set of values is randomly generated using in this case the same function as the one provided to generate the initial fish position and orientation. Those values are called particles. This initial sample is called x_0. It is also acceptable to start by placing these particles at the initial known position of the fish.
2. A **prediction** is made at each step for the next position $\tilde{x}_{t+1}$ based on the position at the current step x_t. This **prediction** is output by the function *StateUpdate*. In other words, each particule is said to behave like an actual fish.

   After, the three following steps **update** the estimated position by correcting them.
3. The weights *weights* associated with the particles of each fish are computed by the pdf of the zero mean Gaussian noise. Knowing that $n_p = x_p(i) - y_p(i)$, realisation of the noise for the next step are given by $y_{t+1} - \tilde{x}_{t+1}$. By evaluating the bivarivate Gaussian pdf provided in the code with those noise values we therefore obtain the weights. Analogously, the bivariate Gaussian can be obtain by multiplying the univariate Gaussian distribution values.
4. By using those weights and the *randsample* MATLAB function only more accurate indices are kept. While resampling with those indices a new set of more accurate position is obtained.
5. Return to step 2 updating the position vector with the freshly sampled position vector

Finally, for each fish and the shark, the position vectors at the different time instants are computed by the mean of the most probable positions of all the particles at each instant. This gives our approximation of the position called x_{est} for the fish and xe_{est} for the shark.

To verify our result, the mean square error for both the noisy observation and the predicted positions have been computed in order to find out in what case the use of the particle filter is beneficial and gives more accurate results than just the observations. The MSE consist of doing the average of the square of the error committed with the estimated positions with respect to the the true positions. It therefore translate how far off from the true coordinates we have predicted/measured. To further increase reliability, the values showcased on the following graphs are made from the average over 8 experiments.

The filter is expected to work more efficiently the bigger the initial sample set. This is true up to a certain point (this might be because of calculation reasons) (see fig 5). Also, the algorithm should improve here again up to a limit as the time step decreases (as showcased on fig 5). Instinctively ; it is easier to predict smaller leap than bigger ones (see fig 16), especially considering the fact that our orientation are not corrected by the algorithm. Moreover, it can be deduced that as the variance of the noise increases (σ^2_{obs}), both the MLE for the noisy observations and the predicted ones will both worsen, however, the impact of the use of the particle filter will be much more important and it allows it to better correct noise induced errors, it might reduce its MLE up to roughly 20% less of the one for noisy measurement . Therefore the use of a particle filter becomes interesting when the noise level is high enough, as seen on figure 6.

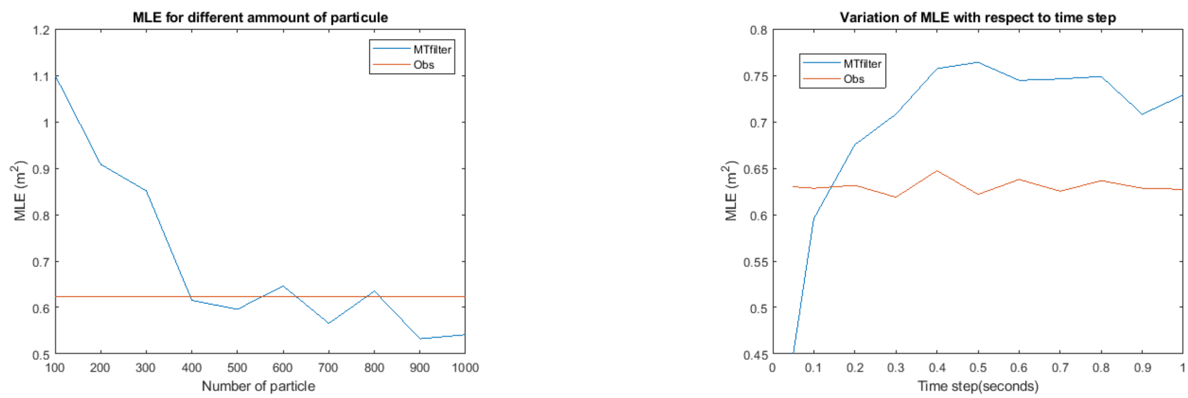

FIGURE 5 – On the left figure , the evolution of the MLE when the number of particle increases for a determined σ^2_{obs}=0.5 and a time step=0.05s. On the right one, evolution of the MLE when the time step increases with the same σ^2_{obs}=0.5 and for 500 particles.

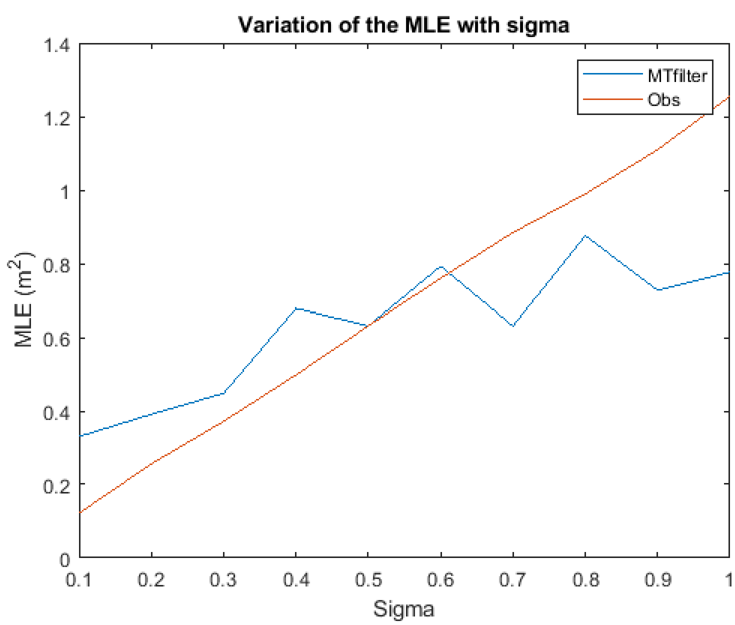

FIGURE 6 – Evolution of the MLE with variation of σ^2_{obs} , for 500 particles and a 0.05s timestep. as showcased on the graph, the ability of the filter to correct the noisy measurements increases with the noise level.

5 Second part conclusion

The Monte Carlo particle filter is an efficient but calculation heavy process to correct noisy measurement given a behaviour function. Its ability to correct increases overall as the noise level increases, the number of particle increases (but calculation time also), and the time step decreases. In our case we found acceptable values such as 500 particles, a time step of 0.05 seconds and a variance of σ_{obs}^2 of 0.5.

6 Appendix

6.1 Regularity condition

Proof of the regularity condition for the pdf given in equation 1.

$$\frac{\partial \log L}{\partial s} = \frac{\partial((k-1)\sum_i \log v_i - \frac{1}{s}(v_1 + v_2 + v_3 + ... + v_n) - nk\log(s) - n\log(\Gamma(k))}{\partial s} \tag{27}$$

The terms not depending on s are cancelled and this gives

$$\frac{\partial \log L}{\partial s} = \frac{\sum_{i=1}^{N} v_i}{s^2} - \frac{nk}{s}. \tag{28}$$

And

$$E\left[\frac{\partial \log L}{\partial s}\right] = E\left[\frac{\sum_{i=1}^{N} v_i}{s^2} - \frac{nk}{s}\right]. \tag{29}$$

The mean just apply on the $\sum_{i=1}^{N} v_i$ such that

$$E\left[\frac{\partial \log L}{\partial s}\right] = \frac{1}{s^2} E\left[\sum_{i=1}^{N} v_i\right] - \frac{nk}{s}. \tag{30}$$

Finally, we fulfill the regularity condition because

$$E\left[\frac{\partial \log L}{\partial s}\right] = \frac{nks}{s^2} - \frac{nk}{s} = 0. \tag{31}$$

6.2 Derivation of the first and second order derivative of Log L

The first order partial derivatives of Log L are :

$$\frac{\partial \log L}{\partial s} = \frac{\partial((k-1)\sum_i \log v_i - \frac{1}{s}(v_1 + v_2 + v_3 + ... + v_n) - nk\log(s) - n\log(\Gamma(k))}{\partial s} \tag{32}$$

Cancelling the terms not depending on s and by derivating the functions of s we have

$$\frac{\partial \log L}{\partial s} = \frac{\sum_{i=1}^{N} v_i}{s^2} - \frac{nk}{s}. \tag{33}$$

$$\frac{\partial \log L}{\partial k} = \frac{\partial((k-1)\sum_i \log v_i - \frac{1}{s}(v_1 + v_2 + v_3 + ... + v_n) - nk\log(s) - n\log(\Gamma(k))}{\partial k} \tag{34}$$

Cancelling the terms not depending on k and by derivating the functions of k we have

$$\frac{\partial \log L}{\partial k} = \sum_i \log v_i - n\log(s) - n\frac{\Gamma^{'}(k)}{\Gamma(k)} \tag{35}$$

Where $\frac{\Gamma^{'}(k)}{\Gamma(k)} = \psi(k)$ is the digamma function.

The second order partial derivatives of Log L are : As in equation 33 the only term depending on k is $-n\psi(k)$ we obtain

$$\frac{\partial^2 \log L}{\partial k^2} = -n\Psi^{(1)}(k). \tag{36}$$

As in equation 33 the only term depending on s is $-n\log(s)$ we obtain

$$\frac{\partial^2 \log L}{\partial s \partial k} = -\frac{n}{s}. \tag{37}$$

As in equation 35 both terms depend on k we obtain

$$\frac{\partial^2 \log L}{\partial s^2} = \frac{-2\sum_{i=1}^{N} v_i}{s^3} + \frac{nk}{s^2} \tag{38}$$

As in equation 35 the only term depending on k is the right term $-\frac{n}{s}$ we obtain

$$\frac{\partial^2 \log L}{\partial k \partial s} = -\frac{n}{s}. \tag{39}$$

6.3 Graphs of convergence

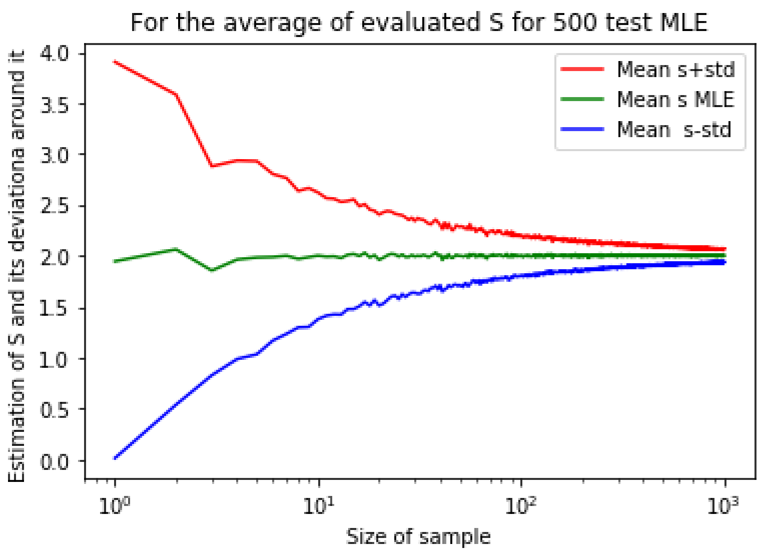

FIGURE 7 – Convergence of $\hat{S}$ with MLE

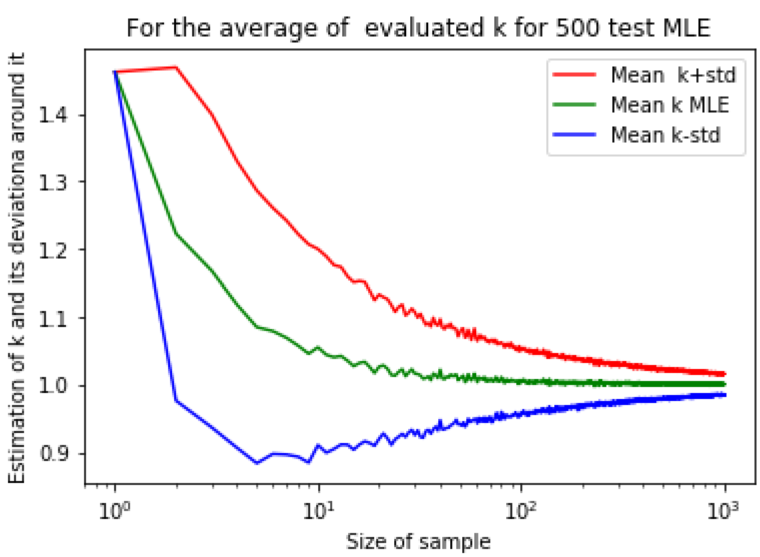

FIGURE 8 – Convergence of $\hat{K}$ with MLE

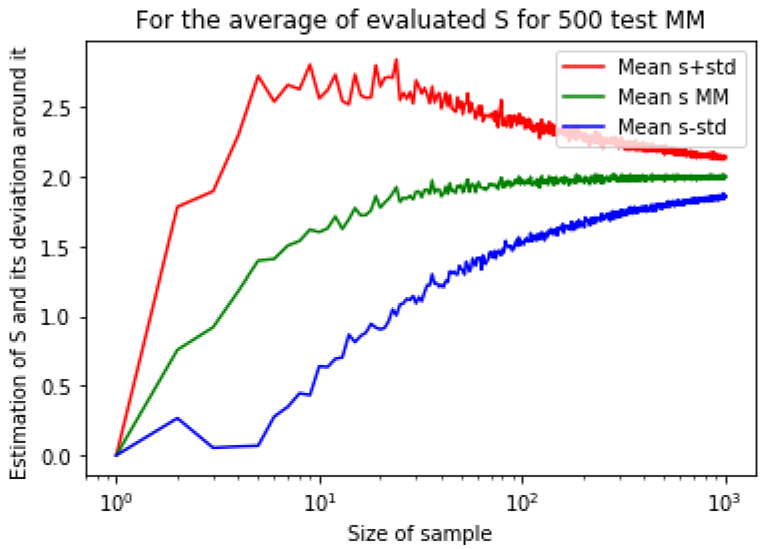

FIGURE 9 – Convergence of $\hat{S}$ with moment method

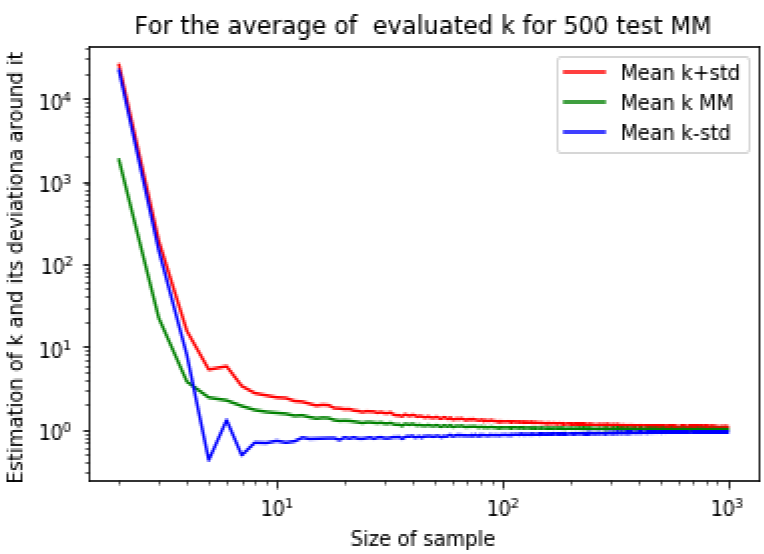

FIGURE 10 – Convergence of $\hat{K}$ with moment method, watch out for the logarithmic scale, we can clearly see that the values for a small sample size are completely wrong

6.4 Graphs using the approximation of equation 14

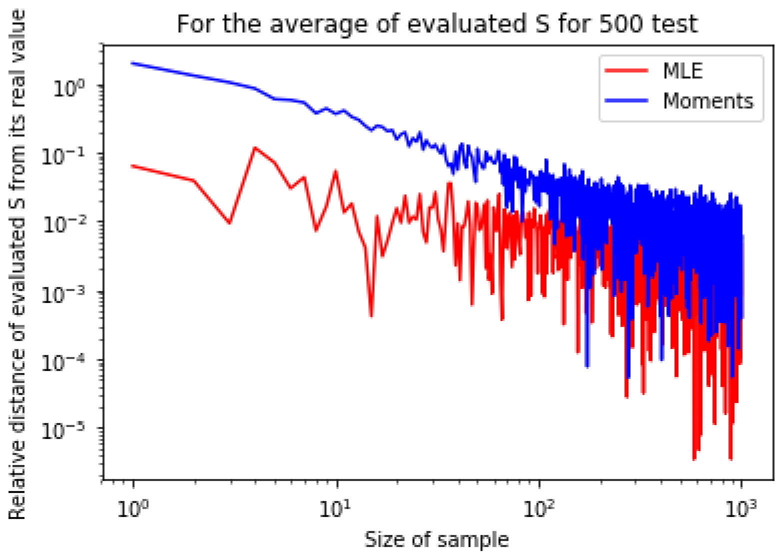

FIGURE 11 – Comparison of the average on 500 test evaluation of $\hat{S}$ with its real value. In red the MLE evaluation and in blue the Moment method

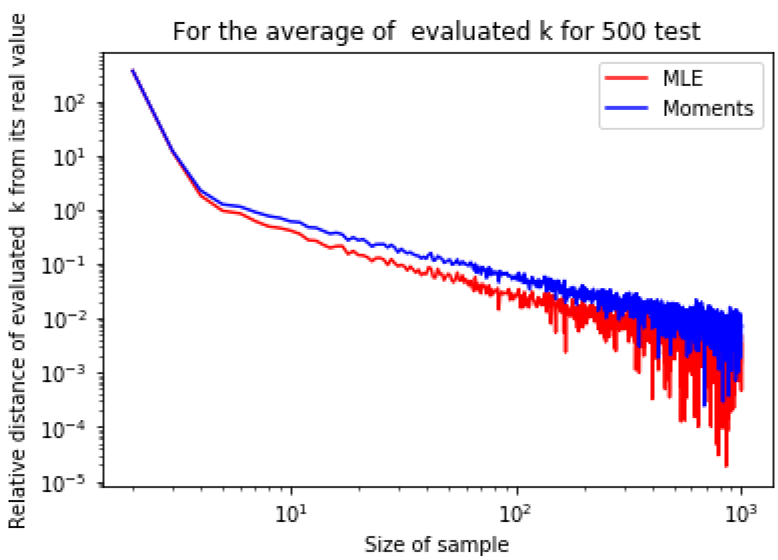

FIGURE 12 – Comparison of the average on 500 test evaluation of $\hat{K}$ with its real value. In red the MLE evaluation and in blue the Moment method

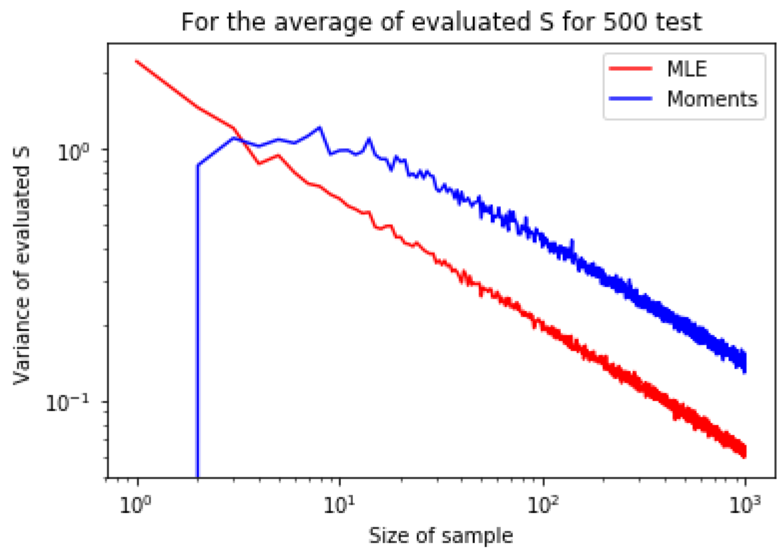

FIGURE 13 – Variance of determined $\hat{S}$. In red the MLE evaluation and in blue the Moment method

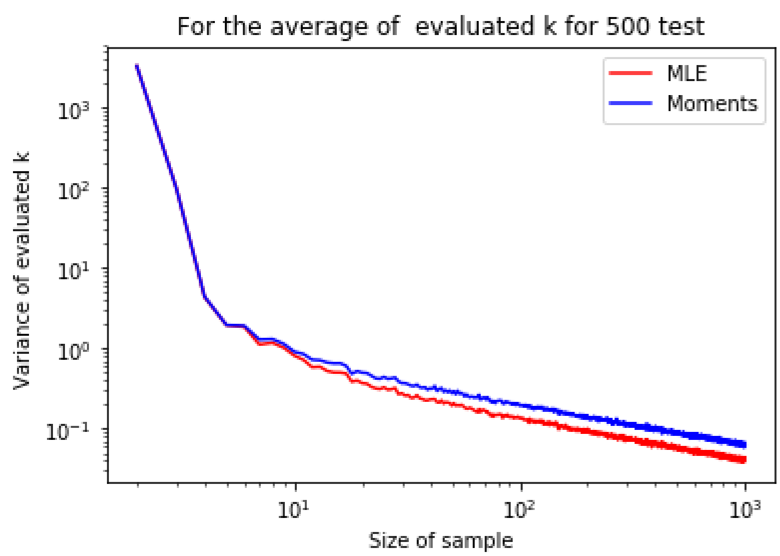

FIGURE 14 – Variance of determined $\hat{K}$. In red the MLE evaluation and in blue the Moment method

6.5 Code

```python

kshape=1

sscale=2 # bc of the definition of the gamma for python

#n=1000

#Y=np.random.gamma(kshape, sscale,n)

Y=np.random.gamma(kshape, sscale,1)
print(Y)

Number=np.arange(1,1001)
Experiment=np.arange(1,501)

def function(k,V,n,s):
    return  (np.sum(np.log(V)) -n*np.log(s) - n*(scipy.special.digamma(k)))







#Y=np.zeros((500,1000))
s=np.zeros((500,1000))
k=np.zeros((500,1000))
sm=np.zeros((500,1000))
km=np.zeros((500,1000))
for n in Number:
    for e in Experiment:

        Y=np.random.gamma(kshape, sscale,n)

        s[e-1,n-1]=  (np.sum(Y))/(n*kshape)  #MLE
        k[e-1,n-1]=scipy.optimize.fsolve(function,0.75,(Y,n,s[e-1,n-1]))
        #k[e-1,n-1]= n*np.sum(Y) /(n*np.sum(Y*np.log(Y)) - (np.sum(np.log(Y))*np.sum(Y))) #MLE
        km[e-1,n-1]= (np.mean(Y)**2)/(np.var(Y)) #methods of moment
        sm[e-1,n-1]= (np.mean(Y)/km[e-1,n-1] ) #methods of moment

```

FIGURE 15 – Code for generating the values

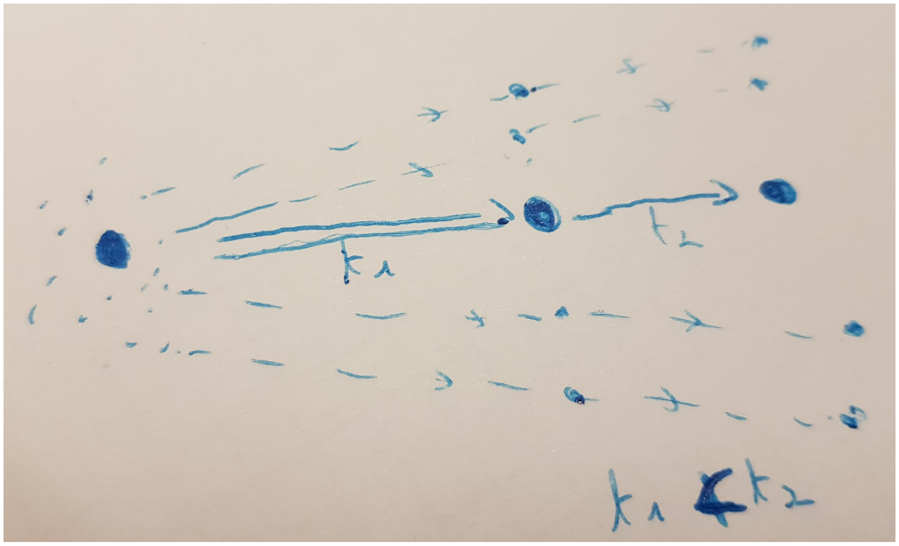

FIGURE 16 – Interpretation of better result for smaller time steps. The particles have more room to spread out for bigger time steps values

Références

[1] Macro Linda_V. Likelihood function of a gamma distributed sample (consulté le 22/04/2019). *https ://math.stackexchange.com/questions/310487/likelihood-function-of-a-gamma-distributed-sample.*

[2] William Feller. An introduction to probability theory and its applications, vol. 2, p.258-259 (consulté le 22/04/2019). *https ://www.researchgate.net/profile/William_Balthes/post/Stronger_limit_Theorems_then_the_strong_law_of_large_numbers/attachment/59d6256e6cda7b8083a21880/AS%3A445697653776384%401483274010393/download/William+Feller-An+Introduction+To+Probability+Theory+And+Its+Applications.+Vol+II.pdf*, 1991.

[3] Wikipédia. Gamma distribution. *https ://en.wikipedia.org/wiki/Gamma$_d$istribution.*

[4] Pierre-Antoine Absil. Slides chapter 3 - estimation theory (consulté le 22/04/2019). *https ://moodleucl.uclouvain.be/pluginfile.php/1197143/mod$_r$esource/content*/3/sec_3_1 − 5_2018 − 2019_01*PA.pdf*, 2019.

[5] Wikipédia. Central limit theorem, classical clt (consulté le 23/04/2019). *https ://en.wikipedia.org/wiki/Central_limit_theoremClassical_CLT.*

[6] André Preumont. Vibrations aléatoires et analyse spectrale (consulté le 22/04/2019). *https ://-books.google.be/books ?id=BelUyYc6j7YCpg=PA73lpg=PA73dq=op%C3%A9ration%20lin%C3%A9aire%20d%27un%20variab ensa = Xved = 2ahUKEwiD24H − 9ObhAhUQfFAKHdHmAtYQ6AEwB3oECAkQAQfbclid = IwAR07hsfCTfS1qxdd747Jvvd6OS1QlBX8fCvL4E5Oun − APMBEPK8aIIANHzYv = onepageq = op%C3%A9ration%20lin%C3%A9aire%20d′un%20variable%20gaussiennef = false*, 2018.

[7] Zhi-Sheng YE and Nan CHEN. Maximum likelihood-like estimators for the gamma distribution (consulté le 22/04/2019). *https ://www.isem.nus.edu.sg/staff/yez/papers/Estimating%20equations%20for%20the%20Gamma%20distribution.pdf ?fbclid=IwAR2CV$_7$nqGE7HKQ1CtXl6gQfDtN9mb0rv7$_k$Ob8r6c4PWTCPgnoN5 − F7SNM.*

[8] Wikipédia. Méthode des moments (statistiques) (consulté le 22/04/2019). *https ://fr.wikipedia.org/wiki/M%C3%A9thode_des_moments_(statistiques) ?fbclid=IwAR2o_9ESR7-ds3XO6Ykgri3ckQ0n6Di-4MpLLQwnpMbeGu5nQBFI8fwAVtc.*

[9] Norman C. Beaulieu Douglas P. Wiens, Julian Cheng. A class of method of moments estimators for the two-parameter gamma family (consulté le 22/04/2019). *http ://www.stat.ualberta.ca/ wiens/-home%20page/pubs/gamma.pdf ?fbclid=IwAR3iJXQ8FrSiE3PCasrINv2qL0bJH6Jj0HM$_T$ivzuLMQ5LfoJ0Tu5bgEeqk.*

[10] A. Srivastava. Computational methods in statistics. 2009. *https ://en.wikipedia.org/wiki/Gamma$_d$istribution.*