ODE parameter estimation

Matthieu Poyer

10 juillet 2019

Content

Introduction

We have a model which is governed by this equation :

$$\boxed{dX_t = f(t, X_t, \theta)dt},$$

and the question is **how to find the parameter θ using the observations ?**

Introduction

From this model we have the function f, the time t and some observations Y which are modelised with some noise (that we may don't know). That's why I transform the equation in :

$$\boxed{dX_t = f(t, X_t, \theta)dt + \sigma dB_t}$$

The unknown is θ and maybe σ too.

An example : the Lotka Volterra model

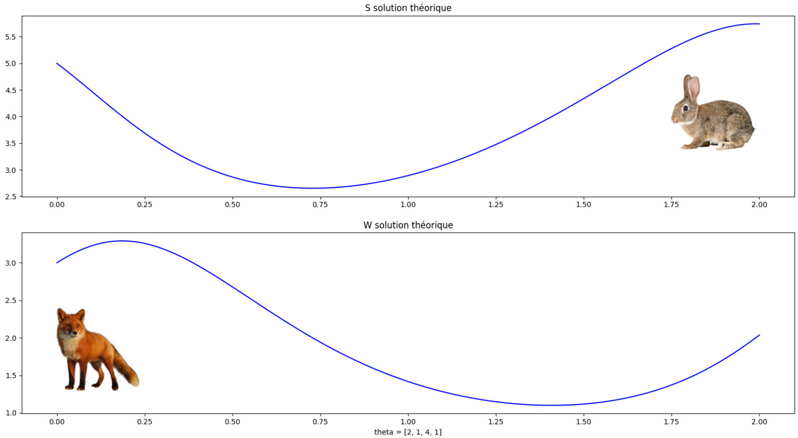
$$\boxed{dX_t = f(t, X_t, \theta)dt}$$

The problem that I used to test the theory is the Lotka-Volterra model :

$$\boxed{\begin{cases} \frac{dS}{dt} = S(\alpha - \beta W) \\ \frac{dW}{dt} = -W(\delta - \gamma S). \end{cases}}$$

So here the unknow are $\theta = (\alpha, \beta, \gamma, \delta)$, $X = (S, W)$ and f does not depend on t.

An example : the Lotka Volterra model



Likelihood

So we are working with :

$$\boxed{dX_t = f(t, X_t, \theta)dt + \sigma dB_t}$$

$$L(\theta) = p(x_{t_0}, t_0; \theta) \prod_{i=1}^{n} p(x_{t_i}, t_i \mid x_{t_{i-1}}, t_{i-1}; \theta)$$

where p is the probability to obtain x_{t_i} at t_i knowing $x_{t_{i-1}}$ when f is paramtrisezd by θ. So its log-likelihood becomes :

$$\boxed{l(\theta) = \log(p(x_{t_0}, t_0; \theta)) + \sum_{i=1}^{n} \log(p(x_{t_i}, t_i \mid x_{t_{i-1}}, t_{i-1}; \theta))}$$

Likelihood

We will use this formula :

$$\boxed{l(\theta) = \log(p(x_{t_0}, t_0; \theta)) + \sum_{i=1}^{n} \log(p(x_{t_i}, t_i \mid x_{t_{i-1}}, t_{i-1}; \theta))}$$

and we are looking for the maximum likelihood estimator

$$\widehat{\theta}_n = \underset{\theta \in \Theta}{\arg\max}\ \ l(\theta).$$

Likelihood

To find $\widehat{\theta}_n$ we need to simplify a little bit the log-likelihood. To do so we need to do quite simple approximations : Euler approximation :

$$X_{t_{i+1}} = X_{t_i} + f(t_i, X_{t_i}, \theta)(t_{i+1} - t_1) + \sigma(B_{t_{i+1}} - B_{t_i}).$$

So we can find p (B is a brownian motion) :

$$p(x_{t_i}, t_i \mid x_{t_{i-1}}, t_{i-1}; \theta) \sim \mathcal{N}(x_{t_{i-1}} + f(t_{i-1}, X_{t_{i-1}}, \theta)(t_i - t_{i-1}),\ (\sigma\, {}^t\sigma)(t_i - t_{i-1}))$$

and the log-likelihood becomes

$$l(\theta) = -\frac{1}{2}\sum_{i=1}^{n}\left(\frac{(x_{t_i} - x_{t_{i-1}} - f(t_{i-1}, x_{t_{i-1}}, \theta)(t_i - t_{i-1}))^2}{\sigma^2(t_i - t_{i-1})} + \log(2\pi\sigma^2(t_i - t_{i-1}))\right).$$

Likelihood

Now I did a hypothesis, I supposed that $\exists \Delta t \; \forall i, \; t_i - t_{i-1} = \Delta t$ so that the log-likelihood becomes :

$$l(\theta) = -\frac{1}{2}\sum_{i=1}^{n}\left(\frac{\|\sigma^{-1}(x_{t_i} - x_{t_{i-1}} - f(t_{i-1}, x_{t_{i-1}}, \theta)(t_i - t_{i-1}))\|_2^2}{(t_i - t_{i-1})}\right.$$
$$\left. + \log((2\pi)^n \det(\sigma)^2 (t_i - t_{i-1}))\right).$$

So to maximize the log-likelihood is to maximises the previous equation.

Likelihood

So the new problem is how to minimise :

$$\boxed{S(\theta) = \sum_{i=1}^{n} \|\sigma^{-1}(x_{t_i} - x_{t_{i-1}} - f(t_{i-1}, x_{t_{i-1}}, \theta)(\Delta t))\|_2^2.}$$

and this is much more easy.

Remark : So, in our case, the max-likelihood estimator is the least squared estimator.

The Gauß-Newton method

The main idea of this method is to apply a Newton method to the derivative of S.

$$\boxed{\theta^{(\text{new})} = \theta - \text{Hess}_\theta(S)^{-1}\nabla_\theta(S)}$$

This isn't the relation that we can found on Wikipedia when we search "Gauß-Newton method", so we rewrite S as
$S(\theta) = \sum_{k=1}^{n} r_k^2(\theta)$.
So $r_k = \|\sigma^{-1}(x_{t_k} - x_{t_{k-1}} - f(t_{k-1}, x_{t_{k-1}}, \theta)(\Delta t))\|_2$ and we found :

$$\text{Hess}_\theta(S)_{i,j} = \sum_k 2r_k \frac{\partial^2 r_k}{\partial x_i \partial x_j} + 2\frac{\partial r_k}{\partial x_i}\frac{\partial r_k}{\partial x_j}.$$

The Gauß-Newton method

The formula that we found need the approximation (I have no main argument to justify it)

$$\left| r_k \frac{\partial^2 r_k}{\partial x_i \partial x_j} \right| \lll \left| \frac{\partial r_k}{\partial x_i} \frac{\partial r_k}{\partial x_j} \right| .$$

So the hessian can be rewritten as :

$$\mathrm{Hess}_\theta(S)_{i,j} = 2 \sum_k \frac{\partial r_k}{\partial x_i} \frac{\partial r_k}{\partial x_j} .$$

The Gauß-Newton method

$$\theta^{(\text{new})} = \theta - \underbrace{\text{Hess}_\theta(S)^{-1}}_{(2\text{Jac}_\theta(\mathbf{r})^\top \text{Jac}_\theta(\mathbf{r}))^{-1}} \underbrace{\nabla_\theta(S)}_{2\text{Jac}_\theta(\mathbf{r})^\top \mathbf{r}} .$$

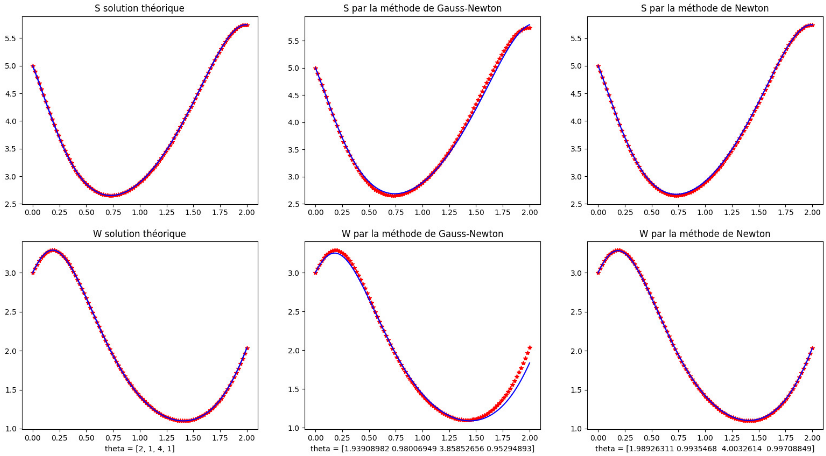
So the formula, which describes the Gauß-Newton method is :

$$\boxed{\theta^{(\text{new})} = \theta - (\text{Jac}_\theta(\mathbf{r})^\top \text{Jac}_\theta(\mathbf{r}))^{-1} \text{Jac}_\theta(\mathbf{r})^\top \mathbf{r}},$$

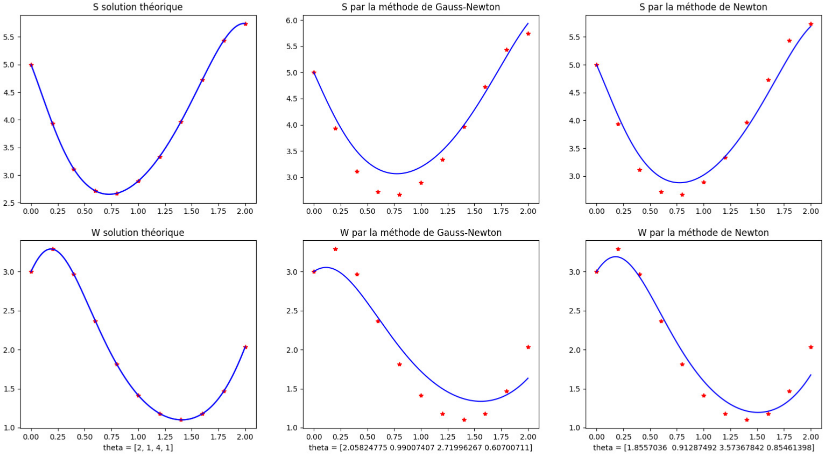
where

$$\mathbf{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix} .$$

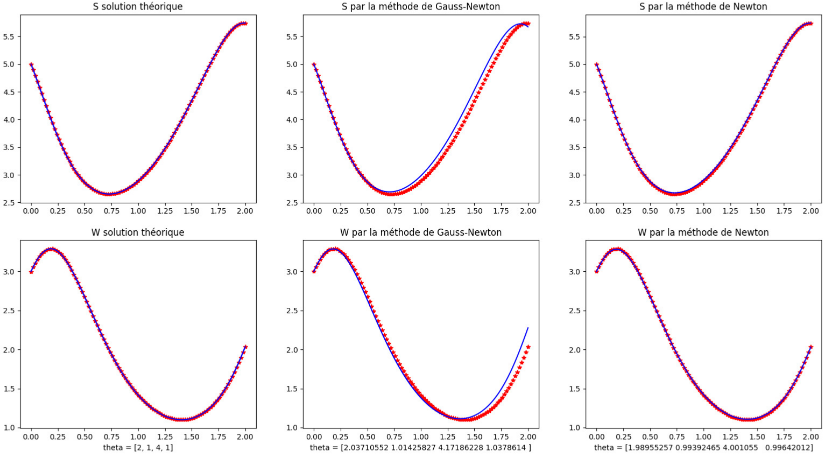
The Lotka-Volterra model without noise



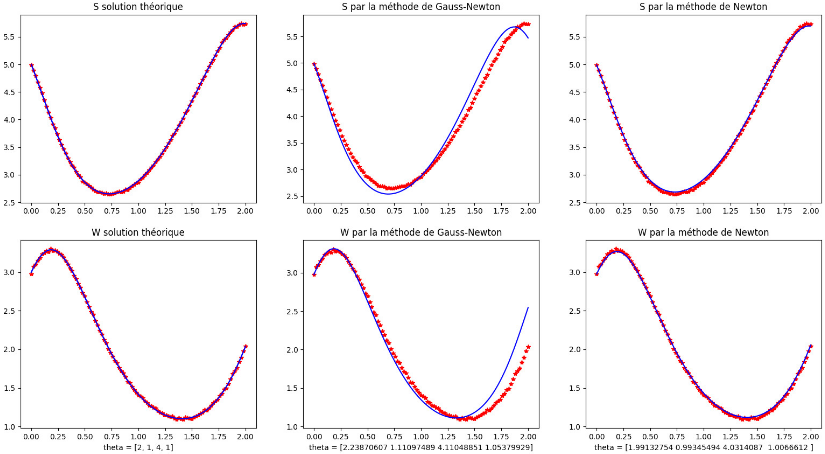
The Lotka-Volterra model without noise



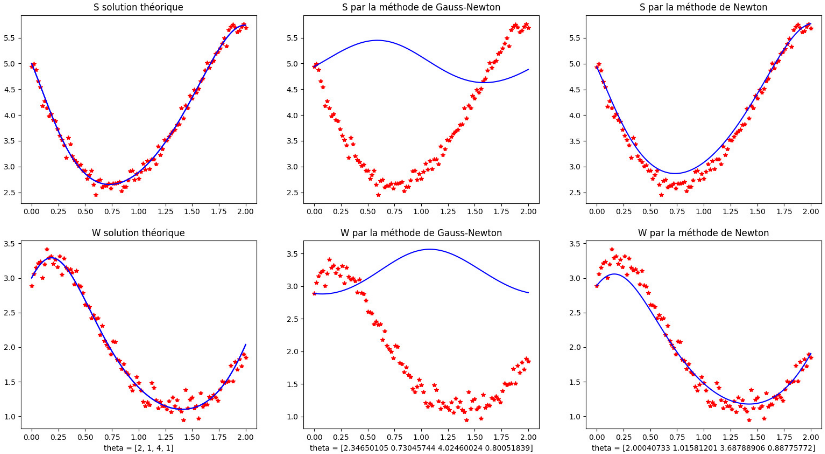
The Lotka-Volterra model with very small noise (0,001)



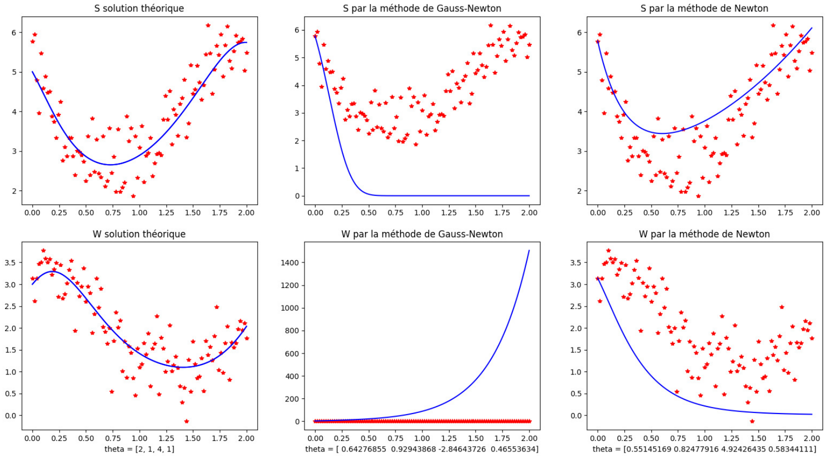
The Lotka-Volterra model with small noise (0,01)



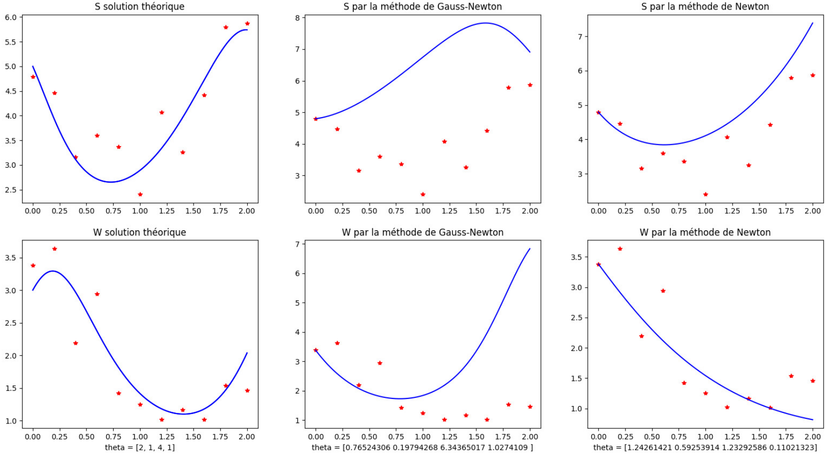
The Lotka-Volterra model with noise (0,1)



The Lotka-Volterra model with noise (0,5)



The Lotka-Volterra model with noise (0,5)



The Gradient descent method

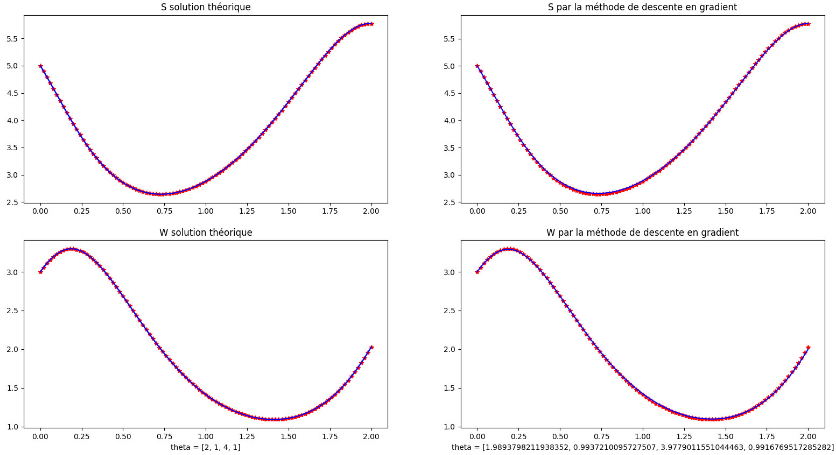
This algorithm uses this formula :

$$\boxed{\theta^{(\text{new})} = \theta - \alpha' \; \nabla_\theta S}$$

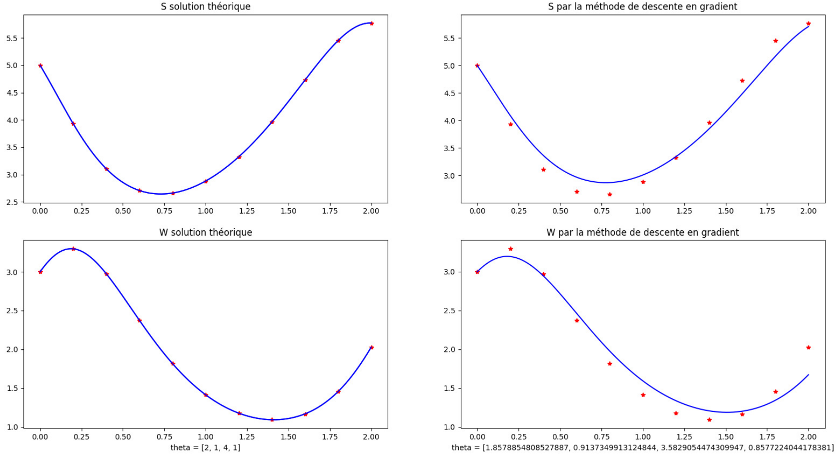
where $\nabla_\theta S$ is the gradient of S at θ and α' need to be determined.
We choose α' such that :

$$\alpha' = \underset{\alpha \in \mathbb{R}}{\arg\min}\{S(\theta - \alpha \nabla_\theta S)\}$$

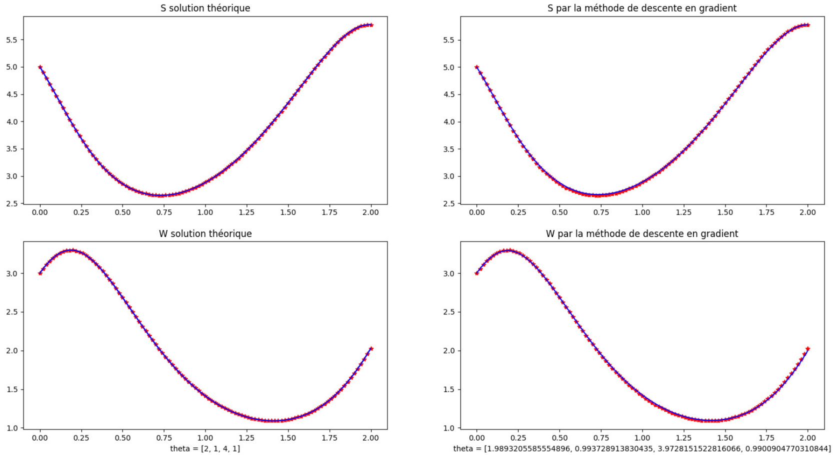
The Lotka-Volterra model without noise



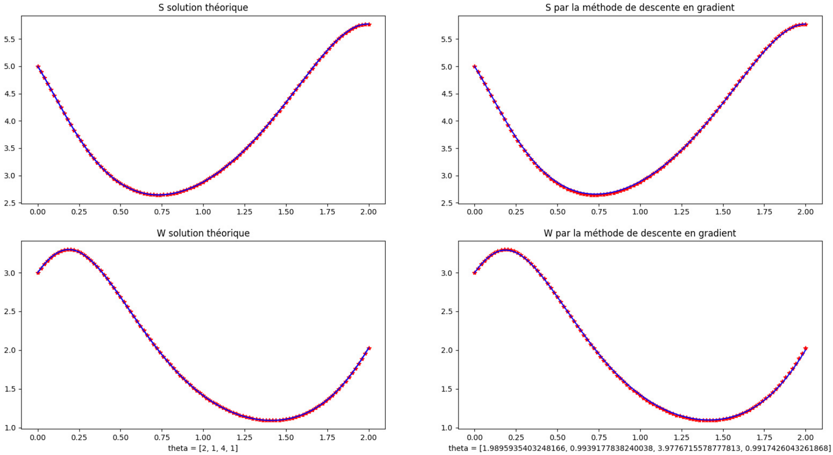
The Lotka-Volterra model without noise



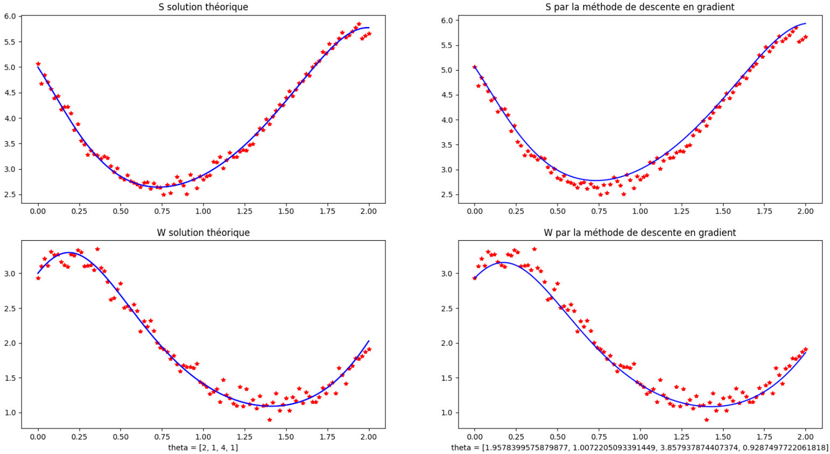
The Lotka-Volterra model with very small noise (0,001)



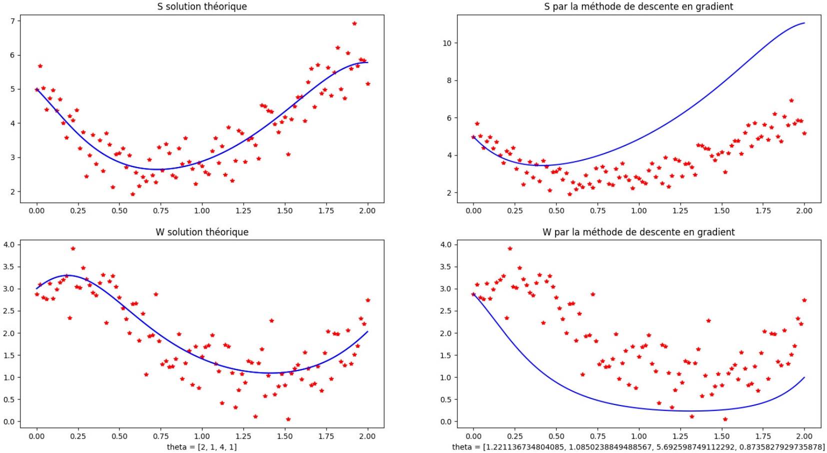
The Lotka-Volterra model with small noise (0,01)



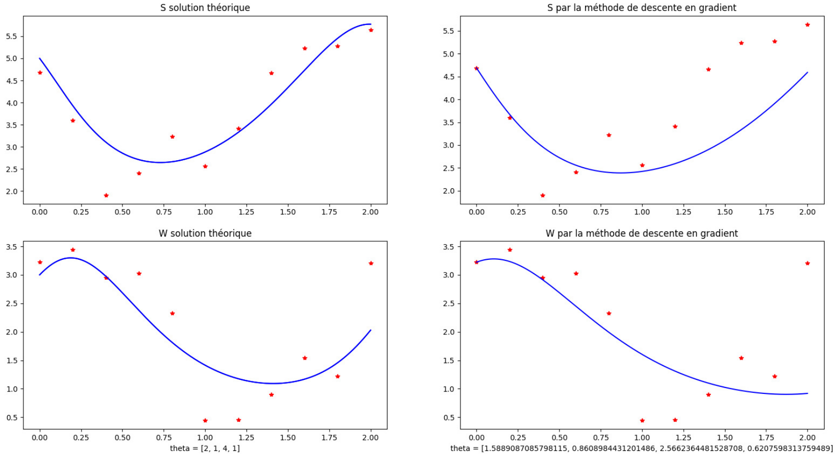
The Lotka-Volterra model with noise (0,1)



The Lotka-Volterra model with noise (0,5)



The Lotka-Volterra model with noise (0,5)



The Levenberg-Marquardt method

If we come back to the Gauß-Newton formula and if we note $F = X - f(t, X; \theta)$ we have :

$$(\mathrm{Jac}_\theta(F)^\top \mathrm{Jac}_\theta(F))(\theta^{(\text{new})} - \theta) = \mathrm{Jac}_\theta(F)(X - F).$$

To make the notations easier we will replace $\mathrm{Jac}_{\theta(k)}(F)$ by $\mathbf{J}$. The Levenberg-Marquardt method consists to add a term (and there are two possibles) :

$$(\mathbf{J}^\top \mathbf{J} - \lambda \mathrm{Id})(\theta^{(\text{new})} - \theta) = \mathbf{J}(X - F)$$
$$(\mathbf{J}^\top \mathbf{J} - \lambda \mathrm{diag}(\mathbf{J}^\top \mathbf{J}))(\theta^{(\text{new})} - \theta) = \mathbf{J}(X - F)$$

Bayesian approach

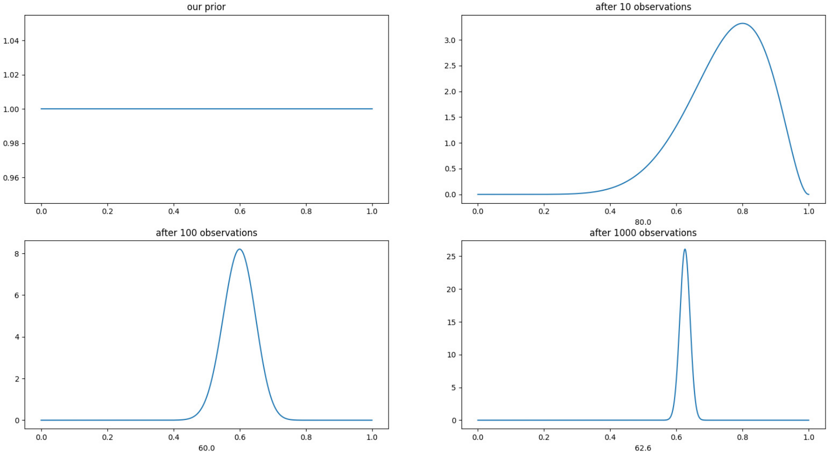
We also can use a Bayesian approach to answer the problem.

What is a bayesian approach ?

A bayesian approach consists on giving a law (prior) and adjust this law with the observation.

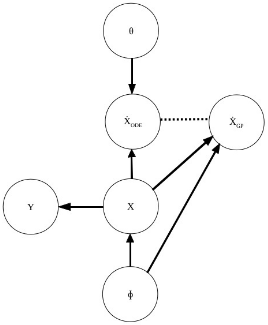
Bayesian approach

Example :



Gradient Matching

The main idea is to use a Gaussian Processes (GP) as a prior :



where θ are
the parameters that we are searching,
ϕ is our prior : the GP.

$$\mathrm{p}(\phi)\mathrm{p}(x \mid \phi)\mathrm{p}(\dot{x}_{GP} \mid \phi, x)\mathrm{p}(\theta)$$
$$\mathrm{p}(\dot{x}_{ODE} \mid x, \theta)\mathrm{p}(y \mid x)\delta(\dot{x}_{ODE} - \dot{x}_{GP})$$

Gradient Matching

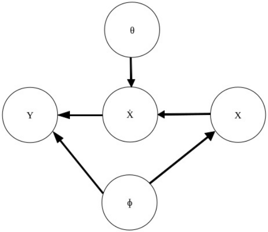
$$\mathrm{p}(\phi, x, \dot{x}, y, \theta) = \mathrm{p}(\phi)\mathrm{p}(x \mid \phi)\mathrm{p}(\dot{x}_{GP} \mid \phi, x)\mathrm{p}(\theta)\mathrm{p}(\dot{x}_{ODE} \mid x, \theta)\mathrm{p}(y \mid x)\delta(\dot{x}_{ODE} - \dot{x}_{GP})$$

Now we change our priors thanks :

$$\begin{aligned} \phi &\sim \mathrm{p}(\phi \mid y) \\ x &\sim \mathrm{p}(x \mid \phi, y) \\ \theta &\sim \mathrm{p}(\theta \mid x, \phi) \end{aligned}$$

Gradient Matching

An other approach is :



where θ are
the parameters that we are searching,
ϕ is our prior : the GP.

$$p(\phi)p(x \mid \phi)p(\dot{x} \mid \theta, x)p(\theta)p(y \mid \dot{x}, \phi)$$

$$\begin{aligned} \phi, \theta &\sim p(\phi, \theta \mid y, x) \\ x &\sim p(x \mid \phi, \theta, y) \end{aligned}$$

The Lotka-Volterra model without and with noise (0,5)