Bayesian Inference and Data Assimilation

Sebastian Reich

Universität Potsdam

12 April 2021

Consider a particle with position $q(t)$ and mass m attached to an elastic spring with spring constant κ. We assume that there is no friction.

The second-order **Newtonian equations of motion** are

$$m\ddot{q}(t) = -\kappa q(t)\,.$$

Or, after introducing the momentum $p(t) = m\dot{q}(t)$, the equivalent pair of first-order **differential equations**

$$\dot{q}(t) = m^{-1}p(t),$$
$$\dot{p}(t) = -\kappa q(t)$$

is obtained.

Let us set $m = 1$ and $\kappa = 1$ for simplicity. It is easy to verify that the **trigonometric functions**

$$q(t) = \sin t, \qquad q(t) = \cos t$$

both satisfy the associated Newtonian equation of motion

$$\ddot{q}(t) = -q(t)\,.$$

It is also true that any **linear superposition** will also satisfy the equations of motion, that is,

$$q(t) = A\cos t + B\sin t$$

with $A, B \in \mathbb{R}$ arbitrary.

This follows from

$$\dot{q}(t) = -A\sin t + B\cos t, \qquad \ddot{q}(t) = -A\cos t - B\sin t = -q(t)\,.$$

To fix a **solution** $q(t)$, we need to **infer** the two parameters A and B.

Assume, for example, that you **observe** the oscillator at positions $q(1) = 5$ and $q(2) = 4$ at times $t = 1$ and $t = 2$, respectively.

It follows that

$$5 = A\cos 1 + B\sin 1, \qquad 4 = A\cos 2 + B\sin 2$$

or, in matrix notation,

$$\begin{pmatrix} 5 \\ 4 \end{pmatrix} = \begin{pmatrix} \cos 1 & \sin 1 \\ \cos 2 & \sin 2 \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix}.$$

Thus $A \approx 1.4030$ and $B \approx 5.0411$. Now you can **predict** the behaviour of the oscillator for all times:

$$q(t) \approx 1.4030\cos t + 5.0411\sin t\,.$$

This is a first (trivial) example of **data assimilation**.

Let us make to problem a bit more challenging. Assume that you have observed several (N) positions $q_{\mathrm{obs}}(t_n)$ at times t_n, which are subject to observation errors.

It makes sense to find the two parameters A and B that minimise the loss function

$$l(A,B) = \frac{1}{2}\sum_{n=1}^{N}(q_{\mathrm{obs}}(t_n) - q(t_n))^2$$

subject to

$$q(t) = A\cos t + B\sin t\,.$$

The two parameters are found by minimising the loss function l. This corresponds to the **maximum likelihood estimator** also called the **method of least-squares**.

Exercise: Find the (linear) **normal equations**

$$0 = \partial_A l(A,B)\,,$$
$$0 = \partial_B l(A,B)\,.$$

Our approach here has been **model-driven** (harmonic oscillator). Alternatively, we could have just taken the data $q_{\mathrm{obs}}(t_n)$ and fit a polynomial (or a neural network or something else). This would constitute a **data-driven** approach of classical **machine learning**/statistics.

The **model-driven** approach **generalises** (predicts) extremely well provided the model (harmonic oscillator) is correct! This is in contrast to purely data-driven approaches.

What if we, for example, do not know the **mass**, m, of the particle?

Then we need to deal with the more general harmonic oscillator model

$$m\ddot{q}(t) = -q(t)\,.$$

But we can still write down its **general solution**:

$$q(t) = A\cos(m^{-1/2}t) + B\sin(m^{-1/2}t)\,,$$

where now both A and B as well as the mass, $m > 0$, are unknown!

Note that

$$\ddot{q}(t) = -m^{-1}A\cos(m^{-1/2}t) - m^{-1}B\sin(m^{-1/2}t) = -m^{-1}q(t)\,.$$

We generalise the method of least squares to include the unknown parameter m:

$$l(A,B,m) = \frac{1}{2}\sum_{n=1}^{N}(q_{\rm obs}(t_n) - q(t_n))^2 \,.$$

This problem is no longer quadratic and methods such as **gradient descent** or **Gauss-Newton** are required; all of which require the **gradient**

$$\nabla l(A,B,m) := \left(\begin{array}{c} \partial_A l(A,B,m) \\ \partial_B l(A,B,m) \\ \partial_m l(A,B,m) \end{array} \right) \in \mathbb{R}^3 \,.$$

Hint: Replace m by $\theta = m^{-1/2}$ and minimise with respect to θ instead!

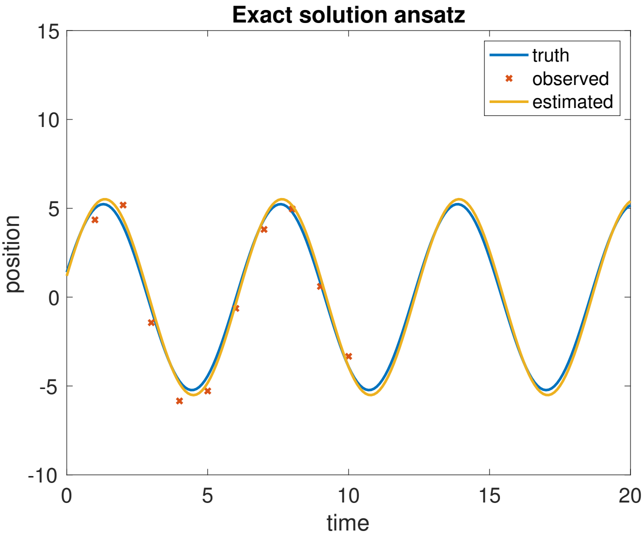
Exercise: Compute the gradient $\nabla l(A,B,\theta)$ for the loss function $l(A,B,\theta)$.

Gradient descent produces a sequence of approximations $x_i = (A_i, B_i, \theta_i)^{\rm T} \in \mathbb{R}^3$ via

$$x_{i+1} = x_i - \alpha \nabla l(x_i)$$

with step-sizes $\alpha > 0$.

We use $m = 1$, $\kappa = 1$ and $t_n = n$, $n = 1, \ldots, N$, with $N = 10$.



1000 steps of gradient descent with $\alpha = 0.0001$ and starting from $m = 1.5$ yield $m \approx 1.015$.

So far we have assumed that the positions $q(t)$ can be explicitly characterised in terms of appropriate parameters A and B. In general, this is not possible and the governing differential equations need to be solved **numerically**.

The **forward Euler method**

$$\begin{aligned} q_{k+1} &= q_k + \frac{\Delta t}{m} p_k, \\ p_{k+1} &= p_k - \Delta t \kappa q_k, \\ t_{k+1} &= t_k + \Delta t, \end{aligned}$$

$k = 0, \ldots, K-1$, with **step-size** $\Delta t > 0$ is the simplest of all approximations.

We rewrite in matrix notation

$$z_{k+1} = C\, z_k, \qquad t_{k+1} = t_k + \Delta t,$$

with $z_k = (q_k, p_k)^{\mathrm{T}} \in \mathbb{R}^2$ and

$$C := \begin{pmatrix} 1 & \frac{\Delta t}{m} \\ -\Delta t \kappa & 1 \end{pmatrix}$$

The **unknown parameters** are the **initial values** of $z_0 = (q_0, p_0)^{\mathrm{T}}$ at time $t = 0$.

How to **infer** the initial z_0 from observed positions $q_{\rm obs}(t_n)$ at times $t_n \geq 0$?

Let us again set $N = 2$ and $t_1 = 1$ and $t_2 = 2$ for simplicity with $q_{\rm obs}(1) = 5$ and $q_{\rm obs}(2) = 4$. Let $e_1 = (1, 0)$ and $\Delta t = 1/L$. For example, $L = 10$ and $\Delta t = 0.1$.

Then

$$q_{\rm obs}(1) = 5 = e_1\, C^L\, z_0, \qquad q_{\rm obs}(2) = 4 = e_1\, C^{2L}\, z_0\,.$$

These are two equations in the two unknowns $z_0 = (q_0, p_0)^{\rm T}$.

Consider now again the case with **observation errors** and observations taken at **integer times** $t_n = n$, $n = 1, \ldots, N$. Then the loss function becomes

$$l(z_0) = \frac{1}{2} \sum_{n=1}^{N} (q_{\rm obs}(t_n) - e_1\, C^{nL}\, z_0)^2\,.$$

The appropriate initial value z_0 is now found as the minimiser of this functional.

This leads to the **method of least squares** yet again. Try to compute the associated **normal equation**.

How to adjust this procedure when the **mass**, m, is again unknown?

No problem at all formally: Just note that the matrix C depends on m, that is,

$$C(m) = \left(\begin{array}{cc} 1 & \frac{\Delta t}{m} \\ -\Delta t \kappa & 1 \end{array} \right).$$

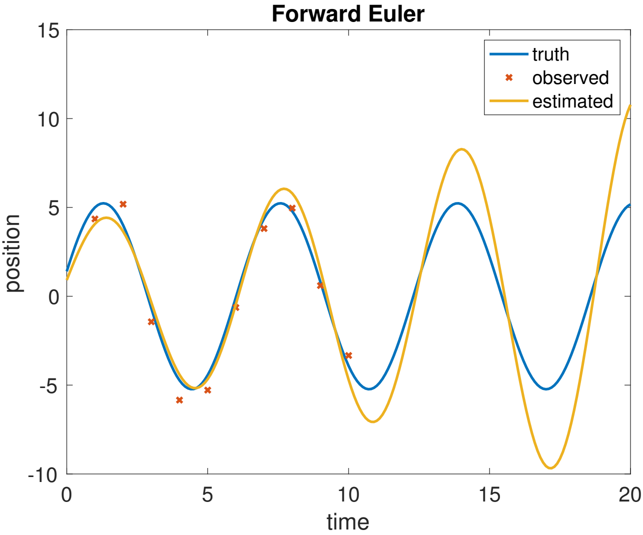
This leads to the extended **loss function**

$$l(z_0, m) = \frac{1}{2} \sum_{n=1}^{N} (q_{\rm obs}(t_n) - e_1\, C(m)^{nL}\, z_0)^2 \,.$$

Question: Can you compute the **gradient** of this loss function? How would you go about finding a **minimiser** of l?

Question: Could one infer both m and κ from the observed particle positions $q_{\rm obs}(t_n)$?

We again set $m = 1$, $\kappa = 1$, $L = 10$, $\Delta t = 0.1$, $t_n = n$, and $N = 10$.



The numerical example has demonstrated that the forward Euler approximation leads to large **prediction errors**. This can be understood as a **model error** which in this case is caused by the **numerical approximation** (forward Euler method).

This numerical error can be largely reduced by the following simple modification:

$$
\begin{aligned}
q_{k+1} &= q_k + \frac{\Delta t}{m} p_k, \\
p_{k+1} &= p_k - \Delta t \kappa \, q_{k+1}, \\
t_{k+1} &= t_k + \Delta t,
\end{aligned}
$$

called a **symplectic Euler method**, or, more compactly,

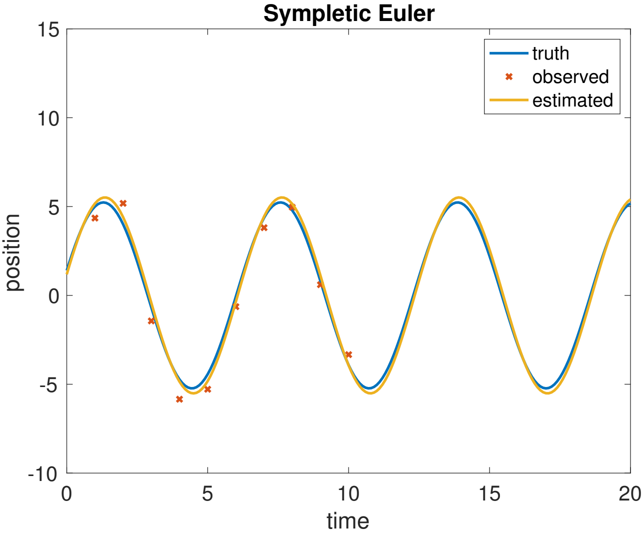
$$
z_{k+1} = C\, z_k, \quad C := \begin{pmatrix} 1 & \frac{\Delta t}{m} \\ -\Delta t \kappa & 1 - \frac{\Delta t^2 \kappa}{m} \end{pmatrix}.
$$

Model errors can arise for many other reasons. For example, our harmonic oscillator equations ignore that there is **friction**; or **Hooks's law** might not be applicable since the restoring spring force is actually nonlinear, e.g.,

$$
F(q) = -\kappa q - \eta q^3
$$

with $\eta > 0$, etc.

We repeat the previous experiment with forward Euler being replaced by its symplectic counterpart.



We have looked at very simple oscillatory dynamics. Much more complex dynamical phenomena such as **chaos** can be encountered. An example is provided by the **Lorenz-63 model** which is discussed in the Prolog. The **dynamics** is either described by differential equations

$$\dot{z} = f(z, t)$$

and/or discrete-time iterations

$$z^{n+1} = \Psi(z^n, t_n)\,.$$

The **aim** of **data assimilation** (DA) is to adjust such **(mechanistic) models** to **data** in order to make **predictions**. In ML one would say that the data-fitted models **generalise** well.

DA focuses on **time-dependent phenomena** and tries to **predict future events**.

In the Prolog from the book, you will also find a more detailed discussion on **model- versus data-driven** approaches to prediction.

We will primarily follow a **Bayesian approach** to DA. I.e., we will attempt to **quantify uncertainties** in addition to providing a **best estimate**.