Institut des sciences
du calcul et des données
SORBONNE UNIVERSITÉ

MAESTRO

Learning the dynamics of systems with memory : Generalized Langevin equations

Hadrien Vroylandt

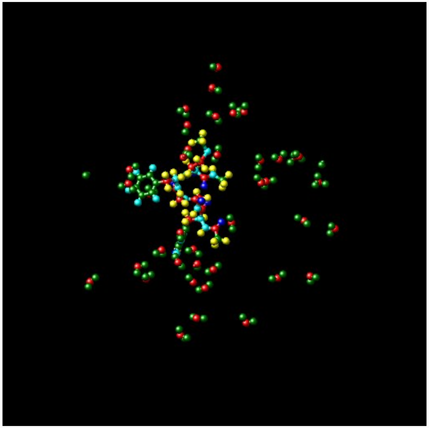
14 Juin 2022

Maestro (MAterials for Energy through STochastic sampling and high peRformance cOmputing), Institut des Sciences du Calcul et des Données

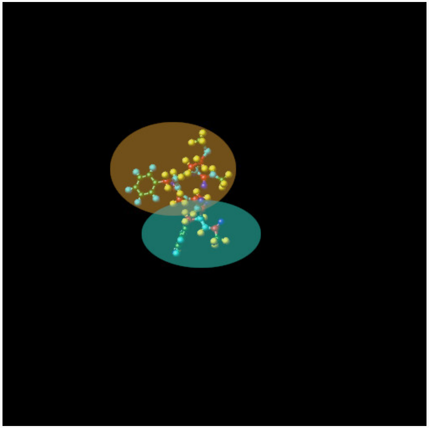
Dimensionality reduction

Tackle high dimensional configurational space with a low-dimensional projection.

- Choice of collective variables?
- *Effective dynamics?*



From all atoms simulations



To reduced dynamical models

- Speed up in computational times (multiscale dynamics)
- Better interpretability (less dimensions)

Maestro (MAterials for Energy through STochastic sampling and high peRformance cOmputing)

The MAESTRO project aims at a conceptual and operational breakthrough in computational materials science, particularly in the study of materials for energy applications and storage.

Collaboration between mathematicians, physicists and chemists @ ISCD, including Ludovic Goudenège, Pierre Monmarché, Fabio Pietrucci and Benjamin Rotenberg.

Prediciting thermodynamics (free energy) and kinetics (rates, timescales) properties of energy storage materials from numerical simulations.

Overview

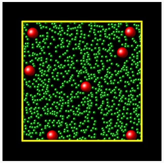
1) **The Generalized Langevin Equation (GLE): here comes the memory**

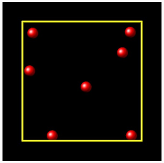
2) **Infering the dynamics?**

3) **Expectation maximization algorithm**

4) **Examples**

The Generalized Langevin Equation (GLE): here comes the memory



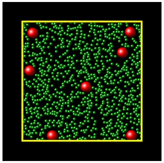


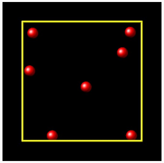
Dynamical (Hamilton) Equations:

$$\begin{cases} \dot{x}_{\text{red}} &= v_{\text{red}} \\ \dot{v}_{\text{red}} &= F_{r\to r}(x_{\text{red}}) + F_{g\to r}(x_{\text{red}}, x_{\text{green}}) \end{cases} \quad \begin{cases} \dot{x}_{\text{green}} &= v_{\text{green}} \\ \dot{v}_{\text{green}} &= F_{g\to g}(x_{\text{green}}) + F_{r\to g}(x_{\text{red}}, x_{\text{green}}) \end{cases}$$

Let's focus on red particles as collective variables.

Can we remove green particles variables? Yes, using Mori Zwanzig projection formalism.





Generalized Langevin equation[1]:

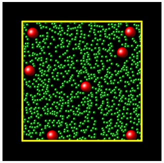
$$\dot{x}_{\mathrm{red}} = v_{\mathrm{red}}$$

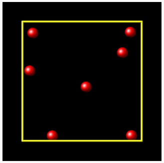
$$\dot{v}_{\mathrm{red}} = \widetilde{F}(x_{\mathrm{red}}) - \int_0^t K(\tau) v_{\mathrm{red}}(t-\tau)\mathrm{d}\tau + \xi(t)$$

The solvant is modeled via

- a non-Markovian memory (friction) kernel $K(\tau)$ (dissipation of energy)
- a diffusion term $\xi(t)$ (random noise coming from collision). This is a function of initial conditions

[1] Zwanzig, 2001





Generalized Langevin equation[1]:

$$\dot{x}_{\mathrm{red}} = v_{\mathrm{red}}$$

$$\dot{v}_{\mathrm{red}} = \widetilde{F}(x_{\mathrm{red}}) - \int_0^t K(\tau) v_{\mathrm{red}}(t-\tau)\mathrm{d}\tau + \xi(t)$$

The solvant is modeled via

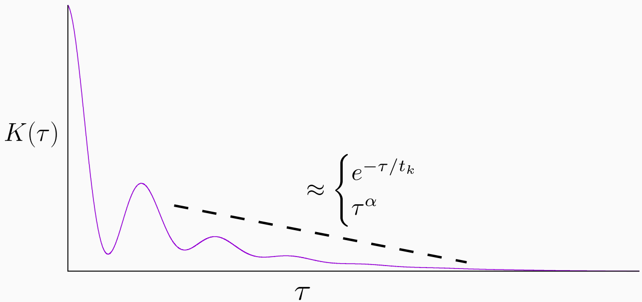
- a non-Markovian memory (friction) kernel $K(\tau)$ (dissipation of energy)
- a diffusion term $\xi(t)$ (random noise coming from collision). This is a function of initial conditions

Position-dependent form of the Generalized Langevin Equations for Hamiltonian systems[2]:

$$\dot{v}(t) = \widetilde{F}(x(t)) - \int_0^t K(\tau, x(t-\tau)) v(t-\tau)\mathrm{d}\tau + \xi(t)$$

[1] Zwanzig, 2001 [2] Vroylandt and Monmarché 2022, arXiv:2201.02457

What are we modeling with this memory? Dynamical timescales of the rest of the system



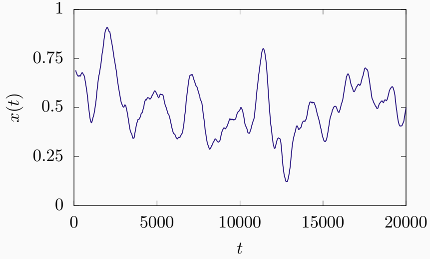
We can have "short" or "long" memory. We focus on "short" memory.
$K(\tau) = \delta(\tau) \Leftrightarrow$ Langevin description

Infering the dynamics?

The inference question

You are given:

Collective variable trajectories:



fit

Model: Generalized Langevin equation:

$$\dot{x} = v$$

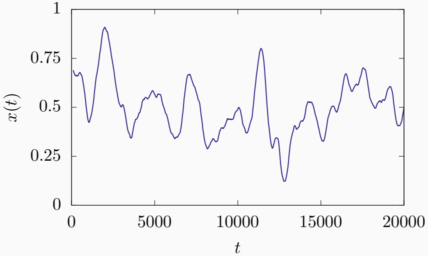
$$\dot{v} = F(x) - \int_0^t K(\tau) v(t-\tau)\mathrm{d}\tau + \xi(t)$$

How do you infer the parameters in this equation from simulation data?

The inference question

You are given:

Collective variable trajectories:



fit

Model: Generalized Langevin equation:

$$\dot{x} = v$$

$$\dot{v} = F(x) - \int_0^t K(\tau)v(t-\tau)\mathrm{d}\tau + \xi(t)$$

How do you infer the parameters in this equation from simulation data?

How do you generate new trajectories from our inferred dynamics

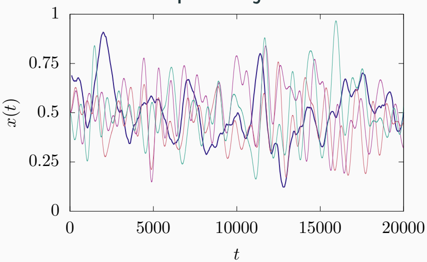
Model: Generalized Langevin equation:

$$\dot{x} = v$$

$$\dot{v} = F(x) - \int_0^t K(\tau)v(t-\tau)\mathrm{d}\tau + \xi(t)$$

sample

New sample trajectories:



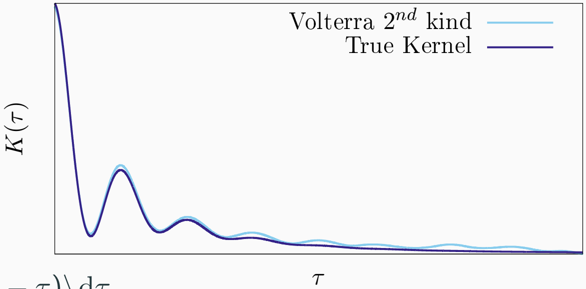
State of the art for memory fitting

Most of current approaches derive from GLE

$$\dot{v} = \widetilde{F}(x) - \int_0^t K(\tau)v(t-\tau)\mathrm{d}\tau + \xi(t)$$

a Volterra integral equation for the memory kernel

$$\left\langle v(0)\left(\dot{v}(t) - \widetilde{F}(x)\right)\right\rangle = -\int_0^t K(\tau)\left\langle v(0)v(t-\tau)\right\rangle \mathrm{d}\tau$$



That can be solved using

- Laplace transform[1]
- Integral discretization[2]
- ...

[a]Lei, Baker & Li, PNAS 2016, 113

[b]Daldrop, Kappler, Brünig, & Netz PNAS, 2018, 115

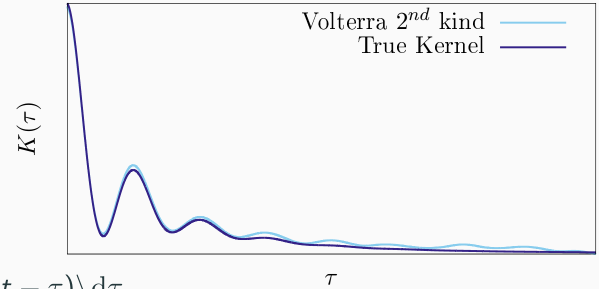
State of the art for memory fitting

Most of current approaches derive from GLE

$$\dot{v} = \widetilde{F}(x) - \int_0^t K(\tau)v(t-\tau)\mathrm{d}\tau + \xi(t)$$

a Volterra integral equation for the memory kernel

$$\left\langle v(0)\left(\dot{v}(t) - \widetilde{F}(x)\right)\right\rangle = -\int_0^t K(\tau)\left\langle v(0)v(t-\tau)\right\rangle \mathrm{d}\tau$$

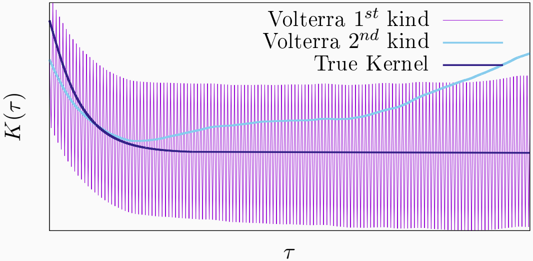


That can be solved using

- Laplace transform[1]
- Integral discretization[2]
- ...

But

- This does not give a generative model
- Convergence issue for some cases



[a] Lei, Baker & Li, PNAS 2016, 113

[b] Daldrop, Kappler, Brünig, & Netz PNAS, 2018, 115

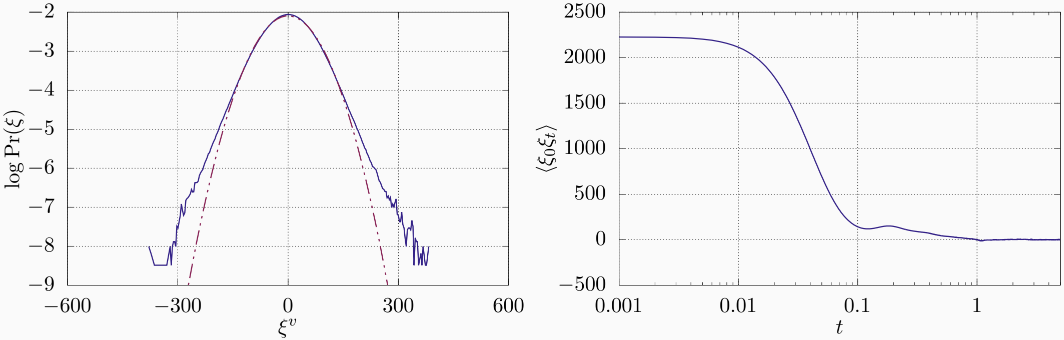
And the noise?

The noise is obtained from the inversion of

$$\dot{x} = v$$

$$\dot{v} = F(x) - \int_0^t K(\tau)v(t-\tau)\mathrm{d}\tau + \xi(t)$$

It exhibits non Gaussian tails and correlation in time



Auxiliary/hidden variables approach

Generalized Langevin equation:

$$\begin{cases} \dot{x} &= v \\ \dot{v} &= F(x) - \int_0^t K(\tau) v(t-\tau) \mathrm{d}\tau + \xi(t) \end{cases}$$

$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{A}_{vh}\, \boldsymbol{h} - \boldsymbol{A}_{vv} \boldsymbol{v} + \boldsymbol{\sigma}_v\, \boldsymbol{W}_v(t) \\ \dot{\boldsymbol{h}} &= -\boldsymbol{A}_{hh}\, \boldsymbol{h} - \boldsymbol{A}_{hv} \boldsymbol{v} + \boldsymbol{\sigma}_h\, \boldsymbol{W}_h(t) \end{cases}$$

Aiming to capture most important missed degrees of freedom.

Auxiliary/hidden variables approach

Generalized Langevin equation:

Hidden variable

$$\begin{cases} \dot{x} &= v \\ \dot{v} &= F(x) - \int_0^t K(\tau) v(t-\tau) \mathrm{d}\tau + \xi(t) \end{cases}$$

$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{A}_{vh} \boldsymbol{h} - \boldsymbol{A}_{vv} \boldsymbol{v} + \boldsymbol{\sigma}_v \boldsymbol{W}_v(t) \\ \dot{\boldsymbol{h}} &= -\boldsymbol{A}_{hh} \boldsymbol{h} - \boldsymbol{A}_{hv} \boldsymbol{v} + \boldsymbol{\sigma}_h \boldsymbol{W}_h(t) \end{cases}$$

Aiming to capture most important missed degrees of freedom.

Gaussian white noise

Auxiliary/hidden variables approach

Generalized Langevin equation:

$$\begin{cases} \dot{x} &= v \\ \dot{v} &= F(x) - \int_0^t K(\tau) v(t-\tau)\mathrm{d}\tau + \xi(t) \end{cases}$$

$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{A}_{vh}\,\boldsymbol{h} \;-\boldsymbol{A}_{vv}\boldsymbol{v} + \boldsymbol{\sigma}_v\,\boldsymbol{W}_v(t) \\ \dot{\boldsymbol{h}} &= -\boldsymbol{A}_{hh}\,\boldsymbol{h} \;-\boldsymbol{A}_{hv}\boldsymbol{v} + \boldsymbol{\sigma}_h\,\boldsymbol{W}_h(t) \end{cases}$$

Aiming to capture most important missed degrees of freedom.

Equivalent equations?
Formal solution to last equation:

$$\boldsymbol{h}(t) = \int_0^t e^{-\boldsymbol{A}_{hh}\tau}\boldsymbol{A}_{hv}\boldsymbol{v}(t-\tau)\mathrm{d}\tau + \int_0^t e^{-\boldsymbol{A}_{hh}\tau}\boldsymbol{\sigma}_h\mathrm{d}\boldsymbol{W}_h(\tau)$$

Memory kernel is the sum of a Dirac and a Prony series:

$$K(\tau) = A_{vv}\delta(\tau) + \sum_i^{\dim h} c_i e^{-\lambda_i\tau}$$

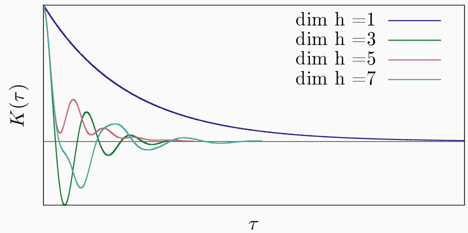
Auxiliary/hidden variables approach

Generalized Langevin equation:

$$\begin{cases} \dot{x} &= v \\ \dot{v} &= F(x) - \int_0^t K(\tau) v(t-\tau)\mathrm{d}\tau + \xi(t) \end{cases}$$

$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{A}_{vh}\boldsymbol{h} - \boldsymbol{A}_{vv}\boldsymbol{v} + \boldsymbol{\sigma}_v \boldsymbol{W}_v(t) \\ \dot{\boldsymbol{h}} &= -\boldsymbol{A}_{hh}\boldsymbol{h} - \boldsymbol{A}_{hv}\boldsymbol{v} + \boldsymbol{\sigma}_h \boldsymbol{W}_h(t) \end{cases}$$

Aiming to capture most important missed degrees of freedom.

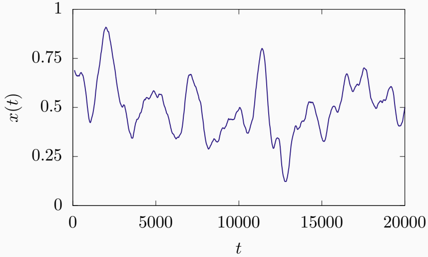


- GLE with memory kernel ⇒ Non-markovian but explicit
- Langevin equations with auxiliary variables ⇒ Markovian but hidden parts

The inference question (bis)

You are given:

Collective variable trajectories:



Model: Generalized Langevin equation with hidden variables

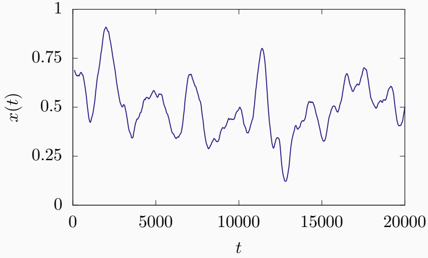
$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{A}_{vh}\boldsymbol{h} - \boldsymbol{A}_{vv}\boldsymbol{v} + \boldsymbol{\sigma}_v \boldsymbol{W}_v(t) \\ \dot{\boldsymbol{h}} &= -\boldsymbol{A}_{hh}\boldsymbol{h} - \boldsymbol{A}_{hv}\boldsymbol{v} + \boldsymbol{\sigma}_h \boldsymbol{W}_h(t) \end{cases}$$

How do you infer the parameters in this equation from simulation data?

How do you generate new trajectories from our inferred dynamics

You are given:

Collective variable trajectories:



Model: Generalized Langevin equation with hidden variables

$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{A}_{vh}\boldsymbol{h} - \boldsymbol{A}_{vv}\boldsymbol{v} + \boldsymbol{\sigma}_v \boldsymbol{W}_v(t) \\ \dot{\boldsymbol{h}} &= -\boldsymbol{A}_{hh}\boldsymbol{h} - \boldsymbol{A}_{hv}\boldsymbol{v} + \boldsymbol{\sigma}_h \boldsymbol{W}_h(t) \end{cases}$$

How do you infer the parameters in this equation from simulation data?

Expectation maximization algorithm.

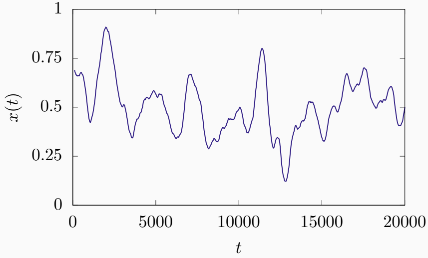
- Iterative algorithm to deal with hidden variables
- E-step: expectation (recovering hidden variables history)
- M-step: maximization of likelihood (recovering parameters)

How do you generate new trajectories from our inferred dynamics

The inference question (bis)

You are given:

Collective variable trajectories:



Model: Generalized Langevin equation with hidden variables

$$
\begin{cases}
\dot{\boldsymbol{x}} &= \boldsymbol{v} \\
\dot{\boldsymbol{v}} &= \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{A}_{vh}\boldsymbol{h} - \boldsymbol{A}_{vv}\boldsymbol{v} + \boldsymbol{\sigma}_v \boldsymbol{W}_v(t) \\
\dot{\boldsymbol{h}} &= -\boldsymbol{A}_{hh}\boldsymbol{h} - \boldsymbol{A}_{hv}\boldsymbol{v} + \boldsymbol{\sigma}_h \boldsymbol{W}_h(t)
\end{cases}
$$

How do you infer the parameters in this equation from simulation data?

Expectation maximization algorithm.

- Iterative algorithm to deal with hidden variables
- E-step: expectation (recovering hidden variables history)
- M-step: maximization of likelihood (recovering parameters)

How do you generate new trajectories from our inferred dynamics

Once the parameters are fitted, this is a generative model.

Expectation maximization algorithm

$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{A}_{vh}\boldsymbol{h} - \boldsymbol{A}_{vv}\boldsymbol{v} + \boldsymbol{\sigma}_v \boldsymbol{W}_v(t) \\ \dot{\boldsymbol{h}} &= -\boldsymbol{A}_{hh}\boldsymbol{h} - \boldsymbol{A}_{hv}\boldsymbol{v} + \boldsymbol{\sigma}_h \boldsymbol{W}_h(t) \end{cases}$$

Discretizing the dynamics
We use a numerical scheme that discretizes the Langevin equation and a parametrization $\boldsymbol{\Theta} = \{F, \boldsymbol{A}, \boldsymbol{\sigma}\}$ of the (force, friction, diffusion) coefficients.

$$(\boldsymbol{X}_{t+\Delta t}, \boldsymbol{h}_{t+\Delta t}) = (\boldsymbol{X}_t, \boldsymbol{h}_t) + G_\Theta(\boldsymbol{X}_t, \boldsymbol{h}_t) + \boldsymbol{S}_{\boldsymbol{\Theta}} \boldsymbol{W}(t)$$

We separate between visible $\boldsymbol{X}_t$ and hidden $\boldsymbol{h}_t$ variables

Transition probability
Hence we have a *gaussian* transition probability

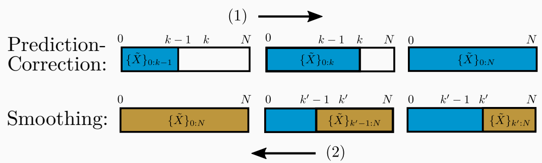
$$\mathbb{P}\left(\boldsymbol{X}_{t+\Delta t}, \boldsymbol{h}_{t+\Delta t} | \boldsymbol{X}_t, \boldsymbol{h}_t\right)_{\boldsymbol{\Theta}} = \mathcal{N}\left((\boldsymbol{X}_t, \boldsymbol{h}_t) + G_\Theta(\boldsymbol{X}_t, \boldsymbol{h}_t), \boldsymbol{S}_{\boldsymbol{\Theta}} \boldsymbol{S}_{\boldsymbol{\Theta}}^{\mathrm{T}}\right)$$

E-step: Reconstructing hidden variables

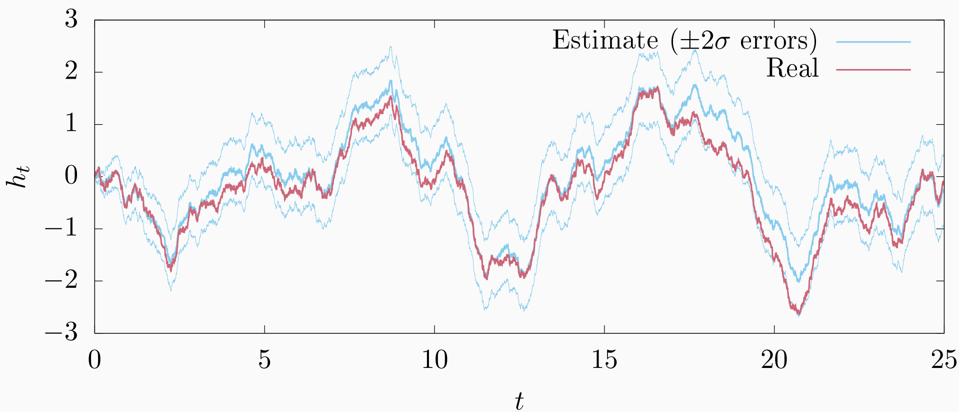
If you know the parameters: Markovianity of the system $\Rightarrow$ We use a predictor-corrector-smoother (or Kalman filer and Rauch smoother)

Using evolution equation

$$\begin{pmatrix} \boldsymbol{X}_{t+\Delta t} \\ \boldsymbol{h}_{t+\Delta t} \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_t \\ \boldsymbol{h}_t \end{pmatrix} + \boldsymbol{G}_{\Theta}(\boldsymbol{X}_t, \boldsymbol{h}_t) + \boldsymbol{S}_{\Theta} \boldsymbol{W}(t)$$



You get probabilistic reconstruction of the hidden variables $\mathbb{P}\left(\{\boldsymbol{h}_t\}_0^T | \{\boldsymbol{X}_t\}_0^T, \boldsymbol{\Theta}\right)$



Expectation-Maximization algorithm

The probability of a trajectory is

$$\mathbb{P}\left(\{\boldsymbol{X}_t, \boldsymbol{h}_t)\}_0^T | \boldsymbol{\Theta}\right) = \mathbb{P}\left((\boldsymbol{X}_0, \boldsymbol{h}_0)\right) \prod_{t=0}^{T} \mathbb{P}\left((\boldsymbol{X}_{t+\Delta t}, \boldsymbol{h}_{t+\Delta t} | (\boldsymbol{X}_t, \boldsymbol{h}_t)\right)_{\boldsymbol{\Theta}}$$

When dealing with hidden variables, we average over possible hidden variables values

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \int \mathrm{d}\{\boldsymbol{h}_t\}_0^T \mathbb{P}\left(\{\boldsymbol{h}_t\}_0^T | \{\boldsymbol{X}_t\}_0^T, \boldsymbol{\Theta}\right) \log \mathcal{L}(\{\boldsymbol{X}_t, \boldsymbol{h}_t\}_0^T; \boldsymbol{\Theta})$$

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} F(\mathbb{P}\left(\{\boldsymbol{h}_t\}; \boldsymbol{\Theta}\right); \boldsymbol{\Theta})$$

Expectation-Maximization algorithm

The probability of a trajectory is

$$\mathbb{P}\left(\{\boldsymbol{X}_t, \boldsymbol{h}_t\}_0^T | \boldsymbol{\Theta}\right) = \mathbb{P}\left((\boldsymbol{X}_0, \boldsymbol{h}_0)\right) \prod_{t=0}^{T} \mathbb{P}\left((\boldsymbol{X}_{t+\Delta t}, \boldsymbol{h}_{t+\Delta t} | (\boldsymbol{X}_t, \boldsymbol{h}_t)\right)_{\boldsymbol{\Theta}}$$

When dealing with hidden variables, we average over possible hidden variables values

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \int \mathrm{d}\{\boldsymbol{h}_t\}_0^T \mathbb{P}\left(\{\boldsymbol{h}_t\}_0^T | \{\boldsymbol{X}_t\}_0^T, \boldsymbol{\Theta}\right) \log \mathcal{L}(\{\boldsymbol{X}_t, \boldsymbol{h}_t\}_0^T; \boldsymbol{\Theta})$$

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} F(\mathbb{P}\left(\{\boldsymbol{h}_t\}; \boldsymbol{\Theta}\right); \boldsymbol{\Theta})$$

Algorithm - Iterative strategy

- Initialize parameters at some value $\boldsymbol{\Theta}_0$

The probability of a trajectory is

$$\mathbb{P}\left(\{\boldsymbol{X}_t, \boldsymbol{h}_t)\}_0^T|\boldsymbol{\Theta}\right) = \mathbb{P}\left((\boldsymbol{X}_0, \boldsymbol{h}_0)\right)\prod_{t=0}^{T}\mathbb{P}\left((\boldsymbol{X}_{t+\Delta t}, \boldsymbol{h}_{t+\Delta t}|(\boldsymbol{X}_t, \boldsymbol{h}_t)\right)_{\boldsymbol{\Theta}}$$

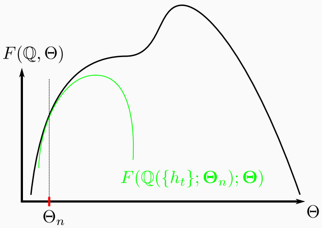
When dealing with hidden variables, we average over possible hidden variables values

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \int \mathrm{d}\{\boldsymbol{h}_t\}_0^T \mathbb{P}\left(\{\boldsymbol{h}_t\}_0^T|\{\boldsymbol{X}_t\}_0^T, \boldsymbol{\Theta}\right) \log \mathcal{L}(\{\boldsymbol{X}_t, \boldsymbol{h}_t\}_0^T; \boldsymbol{\Theta})$$

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} F(\mathbb{P}\left(\{\boldsymbol{h}_t\}; \boldsymbol{\Theta}\right); \boldsymbol{\Theta})$$

Algorithm - Iterative strategy

- Initialize parameters at some value $\boldsymbol{\Theta}_0$
- E-step: Compute expectation of the log-likelihood over hidden variables $F(\mathbb{Q}\left(\{\boldsymbol{h}_t\}; \boldsymbol{\Theta}_n\right); \boldsymbol{\Theta})$



Expectation-Maximization algorithm

The probability of a trajectory is

$$\mathbb{P}\left(\{\boldsymbol{X}_t, \boldsymbol{h}_t)\}_0^T|\boldsymbol{\Theta}\right) = \mathbb{P}\left((\boldsymbol{X}_0, \boldsymbol{h}_0)\right)\prod_{t=0}^{T}\mathbb{P}\left((\boldsymbol{X}_{t+\Delta t}, \boldsymbol{h}_{t+\Delta t}|(\boldsymbol{X}_t, \boldsymbol{h}_t)\right)_{\boldsymbol{\Theta}}$$

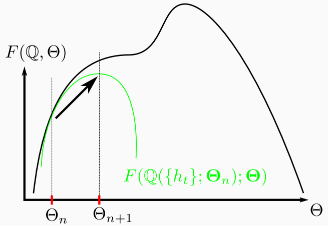
When dealing with hidden variables, we average over possible hidden variables values

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \int \mathrm{d}\{\boldsymbol{h}_t\}_0^T \mathbb{P}\left(\{\boldsymbol{h}_t\}_0^T|\{\boldsymbol{X}_t\}_0^T, \boldsymbol{\Theta}\right) \log\mathcal{L}(\{\boldsymbol{X}_t, \boldsymbol{h}_t\}_0^T; \boldsymbol{\Theta})$$

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} F(\mathbb{P}\left(\{\boldsymbol{h}_t\}; \boldsymbol{\Theta}\right); \boldsymbol{\Theta})$$

Algorithm - Iterative strategy

- Initialize parameters at some value $\boldsymbol{\Theta}_0$
- E-step: Compute expectation of the log-likelihood over hidden variables $F(\mathbb{Q}\left(\{\boldsymbol{h}_t\}; \boldsymbol{\Theta}_n\right); \boldsymbol{\Theta})$
- M-step: Maximize the expectation of the log-likelihood



Expectation-Maximization algorithm

The probability of a trajectory is

$$\mathbb{P}\left(\{\boldsymbol{X}_t, \boldsymbol{h}_t)\}_0^T | \boldsymbol{\Theta}\right) = \mathbb{P}\left((\boldsymbol{X}_0, \boldsymbol{h}_0)\right) \prod_{t=0}^{T} \mathbb{P}\left((\boldsymbol{X}_{t+\Delta t}, \boldsymbol{h}_{t+\Delta t} | (\boldsymbol{X}_t, \boldsymbol{h}_t)\right)_{\boldsymbol{\Theta}}$$

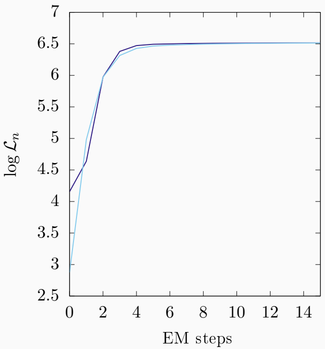
When dealing with hidden variables, we average over possible hidden variables values

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \int \mathrm{d}\{\boldsymbol{h}_t\}_0^T \mathbb{P}\left(\{\boldsymbol{h}_t\}_0^T | \{\boldsymbol{X}_t\}_0^T, \boldsymbol{\Theta}\right) \log \mathcal{L}(\{\boldsymbol{X}_t, \boldsymbol{h}_t\}_0^T; \boldsymbol{\Theta})$$

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} F(\mathbb{P}(\{\boldsymbol{h}_t\}; \boldsymbol{\Theta}); \boldsymbol{\Theta})$$

Algorithm - Iterative strategy

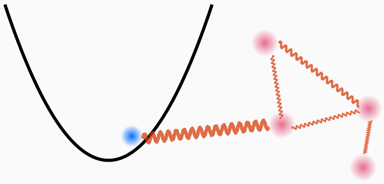
- Initialize parameters at some value $\boldsymbol{\Theta}_0$
- E-step: Compute expectation of the log-likelihood over hidden variables $F(\mathbb{Q}(\{\boldsymbol{h}_t\}; \boldsymbol{\Theta}_n); \boldsymbol{\Theta})$
- M-step: Maximize the expectation of the log-likelihood
- Iterate until convergence



Examples

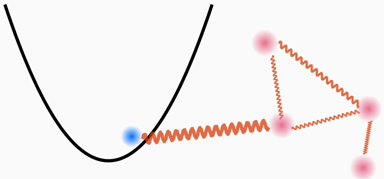
Validation of the algorithm

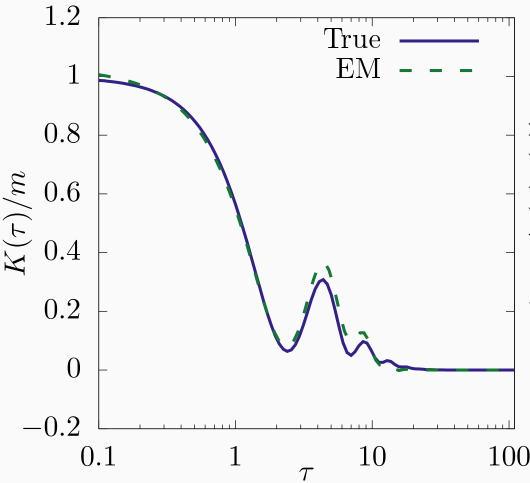
- Quadratic potential
- 5 hidden dimensions
- Random value for the friction matrix A with an hierarchical structure.
- 20 trajectories of 25.10^3 timesteps.

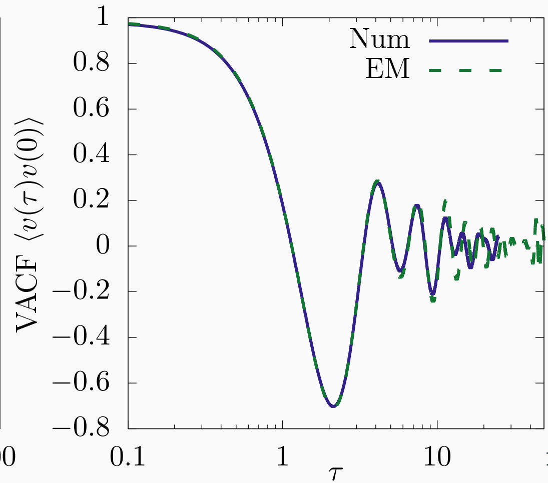


Validation of the algorithm

- Quadratic potential
- 5 hidden dimensions
- Random value for the friction matrix A with an hierarchical structure.
- 20 trajectories of 25.10^3 timesteps.

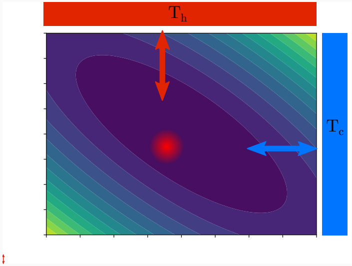






Non-equilibrium 2D setup[1]

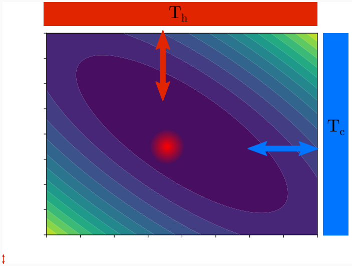
- Quadratic potential with two different temperatures in x and y directions.
- 2 hidden dimensions
- Random value for the friction matrix A.
- 20 trajectories of 30.10^3 timesteps.

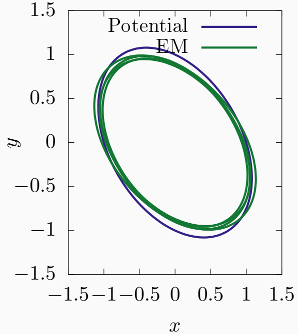


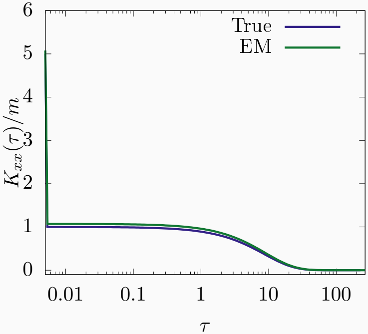
[1] Mancois, Marcos, Viot, and Wilkowski, 2018

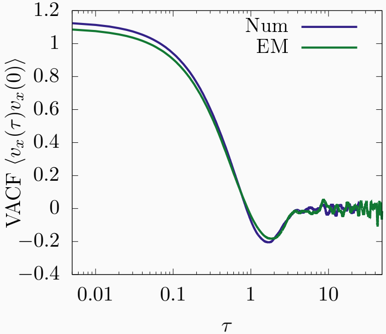
Non-equilibrium 2D setup[1]

- Quadratic potential with two different temperatures in x and y directions.
- 2 hidden dimensions
- Random value for the friction matrix A.
- 20 trajectories of 30.10^3 timesteps.





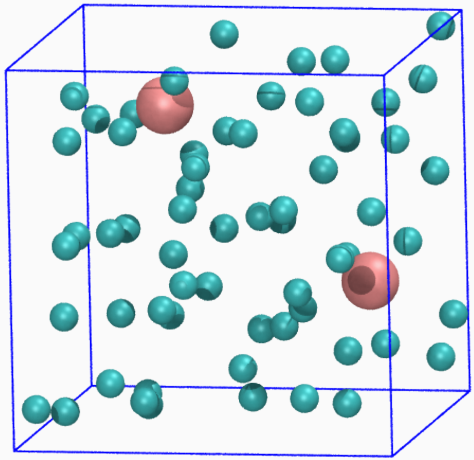




[1] Mancois, Marcos, Viot, and Wilkowski, 2018

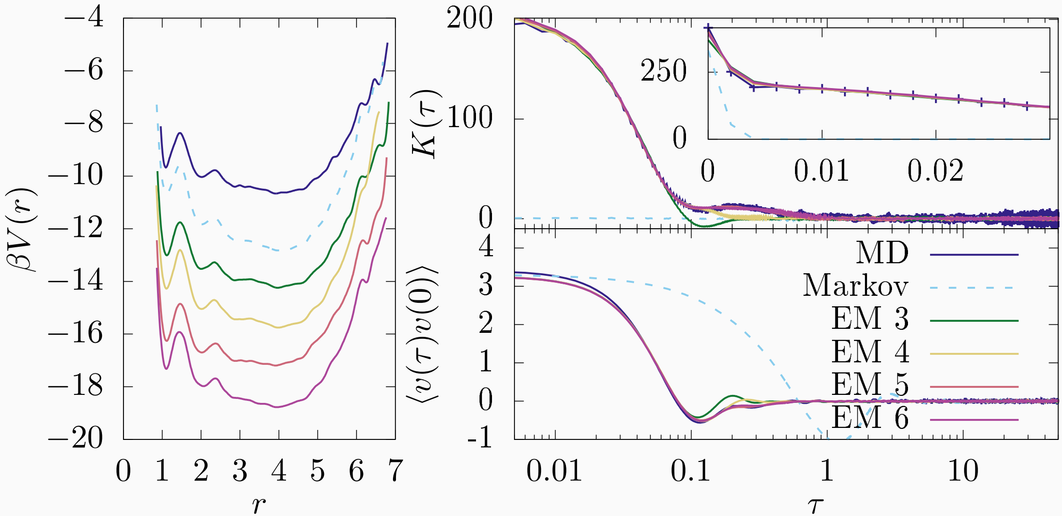
Lennard-Jones Dimer

A molecular dynamics example. We set up 512 LJ particles in a 3D box with periodic boundary conditions. 2 of the particles form a dimer.



Distance r between the two dimer particles as collective variable.

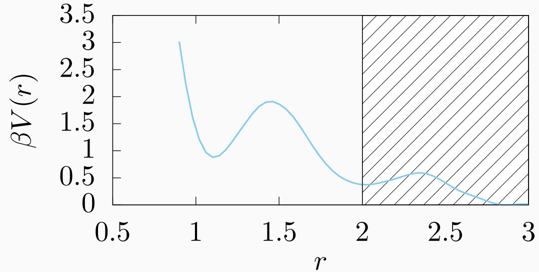
Lennard-Jones Dimer

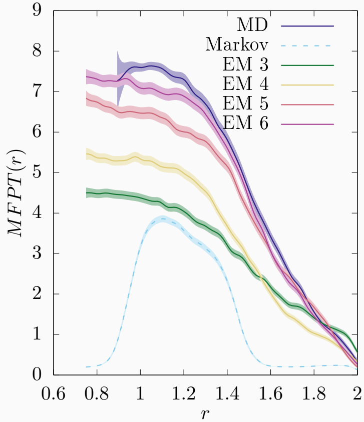


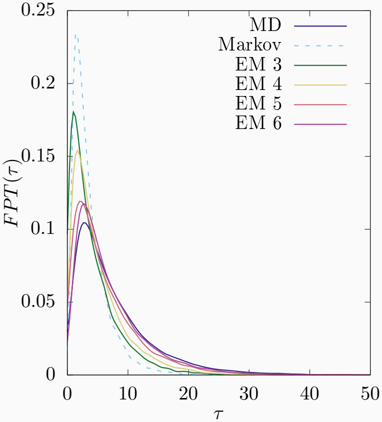
Generate 50 trajectories of 2×100000 timesteps from the fitted model.

We can change the timestep if wanted.

First passage time







Discussion

- This method fits the full statistical model for the dynamics (force + friction + noise).
- Generative model $\Rightarrow$ Possibility to use the inferred dynamics in future computations.
- Works for multidimensionnal and non equilibrium systems.
- Can be quite long to run...

Discussion

- This method fits the full statistical model for the dynamics (force + friction + noise).
- Generative model $\Rightarrow$ Possibility to use the inferred dynamics in future computations.
- Works for multidimensionnal and non equilibrium systems.
- Can be quite long to run...

Thank you for your attention and questions?

Vroylandt, Hadrien, Ludovic Goudenège, Pierre Monmarché, Fabio Pietrucci, and Benjamin Rotenberg.*Likelihood-based non-Markovian models from molecular dynamics*. PNAS, 2022, 119

Hadrien Vroylandt and Pierre Monmarché. *Position-dependent memory kernel in generalized Langevin equations: theory and numerical estimation.* arXiv:2201.02457 (accepted in Journal of Chemical Physics)

Harmonic case

Let's assume that our green particles are harmonic particles, harmonically coupled to the red ones.

$$\begin{cases} \dot{x}_{\text{red}} &= v_{\text{red}} \\ \dot{v}_{\text{red}} &= F_{r\to r}(x_{\text{red}}) + \sum_j \gamma_j (x^j_{\text{green}} - \frac{\gamma_j}{\omega_j^2} x_{\text{red}}) \end{cases} \quad \begin{cases} \dot{x}^j_{\text{green}} &= v^j_{\text{green}} \\ \dot{v}^j_{\text{green}} &= -\omega_j^2 x^j_{\text{green}} + \gamma_j x_{\text{red}} \end{cases}$$

The motion of the green particles is easily solved (Second order differential equation + integration by part):

$$x^j_{\text{green}}(t) - \frac{\gamma_j}{\omega_j^2} x_{\text{red}}(t) = x^j_{\text{green}}(0) \cos \omega_j t - \gamma_j \int_0^t v_{\text{red}}(t-s) \frac{\cos \omega_j(s)}{\omega_j} \mathrm{d}s$$

This lead to the equation of motion for the red particules

$$\begin{cases} \dot{x}_{\text{red}} &= v_{\text{red}} \\ \dot{v}_{\text{red}} &= F_{r\to r}(x_{\text{red}}) - \int_0^t v_{\text{red}}(t-s) \sum_j \frac{\cos \omega_j(s)}{\omega_j} \mathrm{d}s + \sum_j x^j_{\text{green}}(0) \cos \omega_j t \end{cases}$$

Details: Derivation of GLE

Projection operator
For a given basis function family $\{b\}$, we define the projector as

$$\mathcal{P}_{\{b\}} F = B \cdot \langle B, B \rangle^{-1} \cdot \langle B, F \rangle$$

with scalar product $\langle f, g \rangle = \int \mathrm{d}\boldsymbol{X} \rho_{eq}(\boldsymbol{X}) f^{\mathrm{T}}(\boldsymbol{X}) g(\boldsymbol{X})$

The evolution equation of our observables are given by

Evolution equation

$$\frac{\partial \mathcal{O}(\boldsymbol{X}, t)}{\partial t} = \mathcal{L}\mathcal{O}(\boldsymbol{X}, t) = e^{t\mathcal{L}}\mathcal{L}\mathcal{O}(\boldsymbol{X}, 0) = e^{t\mathcal{L}}\mathcal{P}_{\{b\}}\mathcal{L}\mathcal{O}(\boldsymbol{X}, 0) + e^{t\mathcal{L}}(1 - \mathcal{P}_{\{b\}})\mathcal{L}\mathcal{O}(\boldsymbol{X}, 0)$$

Duhamel-Dyson identity:$e^{t\mathcal{L}} = e^{t(1-\mathcal{P}_{\{b\}})\mathcal{L}} + \int \mathrm{d}s e^{(t-s)\mathcal{L}} \mathcal{P}_{\{b\}} \mathcal{L} e^{s(1-\mathcal{P}_{\{b\}})\mathcal{L}}$

$$\begin{aligned} \frac{\partial \mathcal{O}(\boldsymbol{X}, t)}{\partial t} =& e^{t\mathcal{L}}\mathcal{P}_{\{b\}}\mathcal{L}\mathcal{O}(\boldsymbol{X}, 0) + \int \mathrm{d}s e^{(t-s)\mathcal{L}} \mathcal{P}_{\{b\}} \mathcal{L} e^{s(1-\mathcal{P}_{\{b\}})\mathcal{L}} (1 - \mathcal{P}_{\{b\}})\mathcal{L}\mathcal{O}(\boldsymbol{X}, 0) \\ &+ e^{t(1-\mathcal{P}_{\{b\}})\mathcal{L}}(1 - \mathcal{P}_{\{b\}})\mathcal{L}\mathcal{O}(\boldsymbol{X}, 0) \\ \frac{\partial \mathcal{O}(\boldsymbol{X}, t)}{\partial t} =& e^{t\mathcal{L}}\mathcal{P}_{\{b\}}\mathcal{L}\mathcal{O}(\boldsymbol{X}, 0) + \int \mathrm{d}s e^{(t-s)\mathcal{L}} \mathcal{P}_{\{b\}} \mathcal{L} F(\boldsymbol{X}, s) + F(\boldsymbol{X}, t) \end{aligned}$$

with $F(\boldsymbol{X}, t) = e^{t(1-\mathcal{P}_{\{b\}})\mathcal{L}}(1 - \mathcal{P}_{\{b\}})\mathcal{L}\mathcal{O}(\boldsymbol{X}, 0)$