

# MATH+ Hackathon 2024: Isokann for MD simulations

**Luca Donati**

Zuse Institute Berlin  
Modeling and Simulation of Complex Processes  
Computational Molecular Design

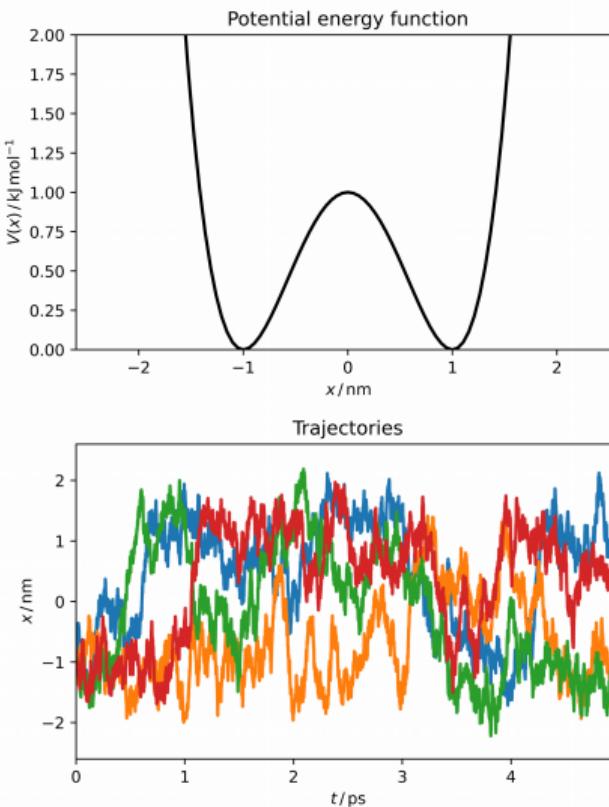
22-24 January 2024

- Theory
  - Stochastic Dynamics
  - The Propagator
  - The Koopman operator
  - Rates from membership functions
- ISOKANN method
- ISOKANN vs SqRA: 1D, 2D, 3D examples

We consider a dynamical system driven by the SDE

$$dx_t = -\nabla V(x_t)dt + \sqrt{2D}dW_t, \quad (1)$$

with  $V(x) : \Gamma \rightarrow \mathbb{R}$  potential energy function,  $D$  diffusion constant and  $W_t$  a standard Wiener process.



Associated to eq. 1, there exists the PDE

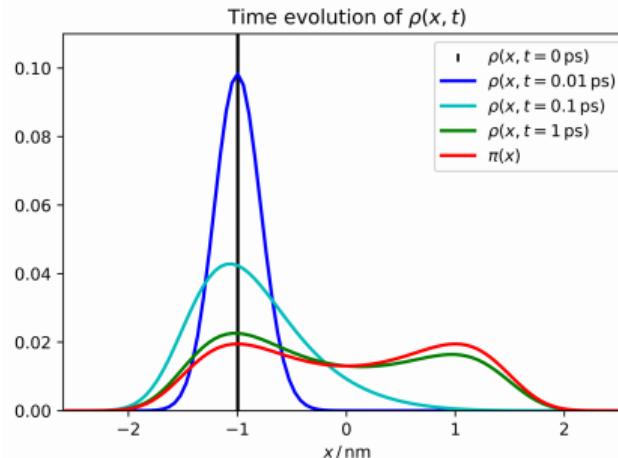
$$\frac{\partial \rho}{\partial t} = \mathcal{L}\rho(x, t), \quad (2)$$

where  $\mathcal{L}$  is the *Kolmogorov forward operator*.

The formal solution is

$$\begin{aligned} \rho(x, t + \tau) &= e^{\tau \mathcal{L}} \rho(x, t) \\ &= \mathcal{P}(\tau) \rho(x, t), \end{aligned} \quad (3)$$

where  $\mathcal{P}(\tau)$  is the *propagator*.



Stationary distribution:

$$\lim_{t \rightarrow +\infty} \rho(x, t) = \pi(x).$$

Similarly, we introduce the adjoint equation for arbitrary observable functions  $f(x)$

$$\frac{\partial f}{\partial t} = \mathcal{L}^* f(x, t), \quad (4)$$

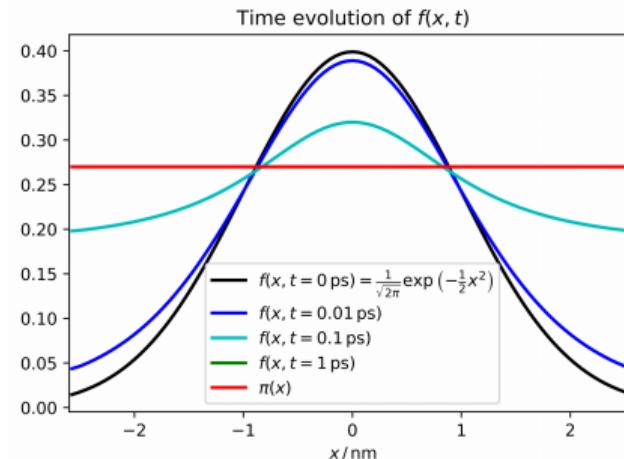
where  $\mathcal{L}^*$  is the *Kolmogorov backward operator*.

The formal solution is

$$\begin{aligned} f(x, t + \tau) &= e^{\tau \mathcal{L}^*} f(x, t) \\ &= \mathcal{K}(\tau) f(x, t), \end{aligned} \quad (5)$$

where  $\mathcal{K}(\tau)$  is the *Koopman operator*:

$$\mathcal{K}(\tau) = (\mathcal{P}(\tau))^*. \quad (6)$$



$$\lim_{t \rightarrow +\infty} f(x, t) = \text{const.}$$

- *Membership functions*

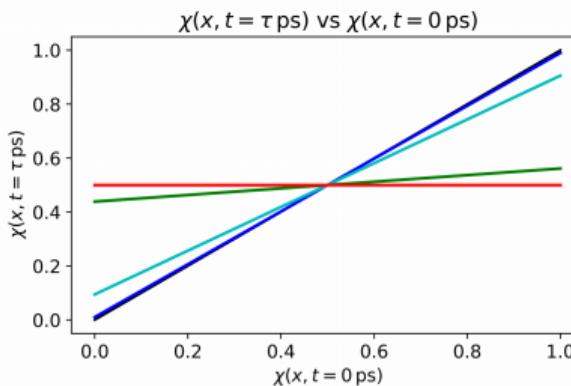
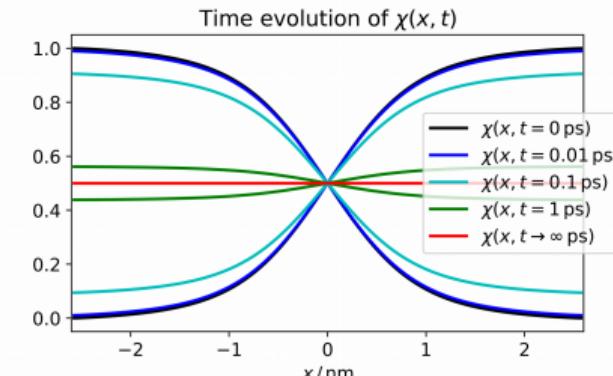
$$f(x) = \chi(x) : \Gamma \rightarrow [0, 1].$$

- The Koopman operator acts on  $\chi(x)$  as

$$\begin{aligned}\chi(x, t + \tau) &= \mathcal{K}(\tau)\chi(x, t) \\ &= \gamma_1\chi(x, t) + \gamma_2.\end{aligned}$$

- From linear regression we can estimate  $\gamma_1$  and  $\gamma_2$ , and exit rates [1]:

$$\kappa = -\frac{1}{\tau} \log(\gamma_1) \left( 1 + \frac{\gamma_2}{\gamma_1 - 1} \right)$$



[1] Marcus Weber, and Natalia Ernst. *arXiv:1708.00679*.

Given an initial function  $f_0(x)$ :

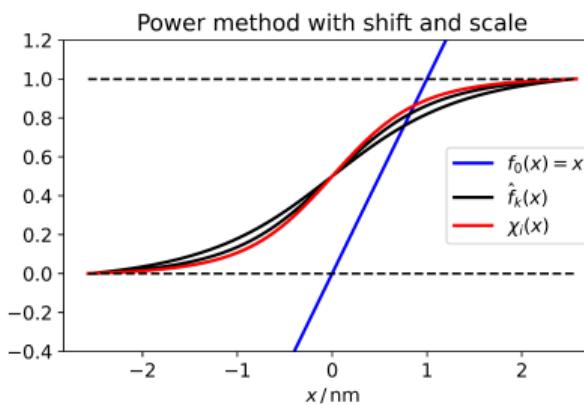
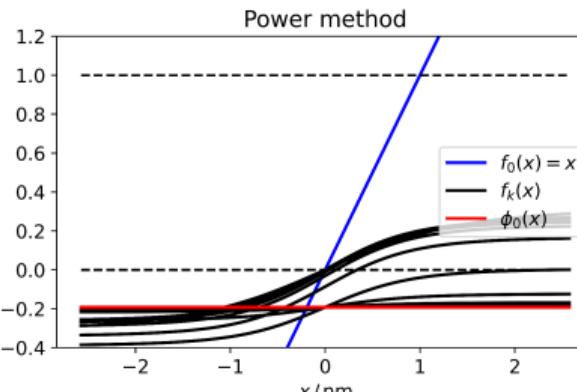
$$\begin{aligned} f_{k+1} &= \frac{\mathcal{K}(\tau) f_k}{\|\mathcal{K}(\tau) f_k\|} \\ \lim_{k \rightarrow \infty} f_{k+1} &= \phi_0(x) \end{aligned}$$

If we apply an appropriate linear transformation  $\mathcal{S}$ :

$$\begin{aligned} \hat{f}_{k+1} &= \frac{\mathcal{S}\mathcal{K}(\tau) f_k}{\|\mathcal{S}\mathcal{K}(\tau) f_k\|} \\ \lim_{k \rightarrow \infty} \hat{f}_{k+1} &= \chi_i(x) \quad i = 1, 2 \end{aligned}$$

For a two-state system,  $\mathcal{S}$  is given by

$$\mathcal{S}\mathcal{K}(\tau) f_k = \frac{\mathcal{K}(\tau) f_k - \min(\mathcal{K}(\tau) f_k)}{\max(\mathcal{K}(\tau) f_k) - \min(\mathcal{K}(\tau) f_k)}$$



- In the previous examples,  $\mathcal{K}(\tau)$  was estimated by SqRA. Alternatively, it can be estimated as

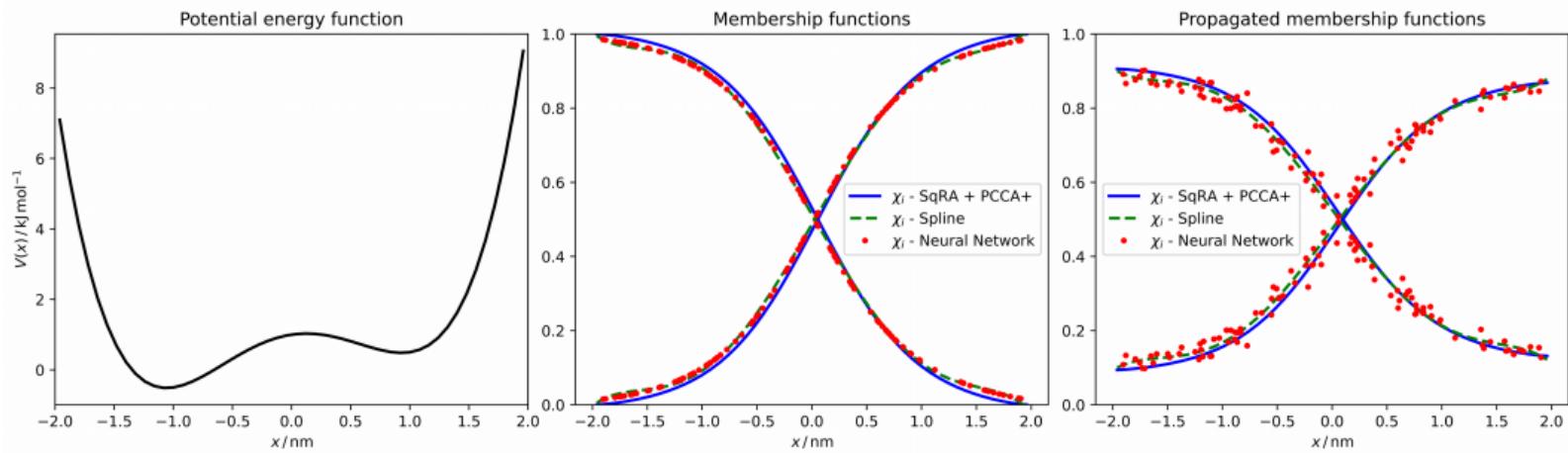
$$\begin{aligned}\mathcal{K}(\tau)f(x) &= \mathbb{E}[f(x_\tau)|x_0 = x] \\ &\approx \frac{1}{N} \sum_{n=1}^N f(x_{\tau,n}) = \bar{f}(x_\tau),\end{aligned}\tag{7}$$

where  $x_{\tau,n}$  is the end-point of the  $n$ th simulation of length  $\tau$  starting in  $x_0 = x$ .

- The shift-and-scale power method algorithm becomes

$$\begin{aligned}\hat{\bar{f}}_{k+1} &= \frac{\mathcal{S}\bar{f}_k}{\|\mathcal{S}\bar{f}_k\|} \\ \lim_{k \rightarrow \infty} \hat{\bar{f}}_{k+1} &= \bar{\chi}_i(x) \quad i = 1, 2.\end{aligned}$$

- The initial function is arbitrary, it is useful to consider an interpolating function such as a *spline* or *neural network*.



-	SqRA + PCCA+	Spline	Rel. Err.	Neural Network	Rel. Err.
$\kappa_{12} \text{ ps}^{-1}$	1.773	1.811	0.021	1.837	0.035
$\kappa_{21} \text{ ps}^{-1}$	2.483	2.391	0.037	2.414	0.027

