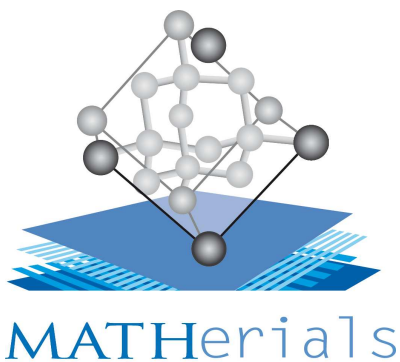
Tensorized Adaptive Biasing Force method for molecular dynamics

Virginie Ehrlacher[1], Tony Lelièvre[1] & Pierre Monmarché[2]

[1] CERMICS, Ecole des Ponts ParisTech & MATHERIALS project, INRIA

[2] Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie



IPAM, November 2017

Outline of the talk

Outline of the talk

Introduction

The aim of molecular dynamics simulations is to understand the relationships between the macroscopic properties of a molecular system and its atomistic features. In particular, one would like to evaluate numerically macroscopic quantities from models at the microscopic scale.

Many applications in various fields: biology, physics, chemistry, materials science.

Example: Compute the pressure of a liquid at a given density and temperature.

Various models: discrete state space (kinetic Monte Carlo, Markov State Model) or continuous state space (Langevin).

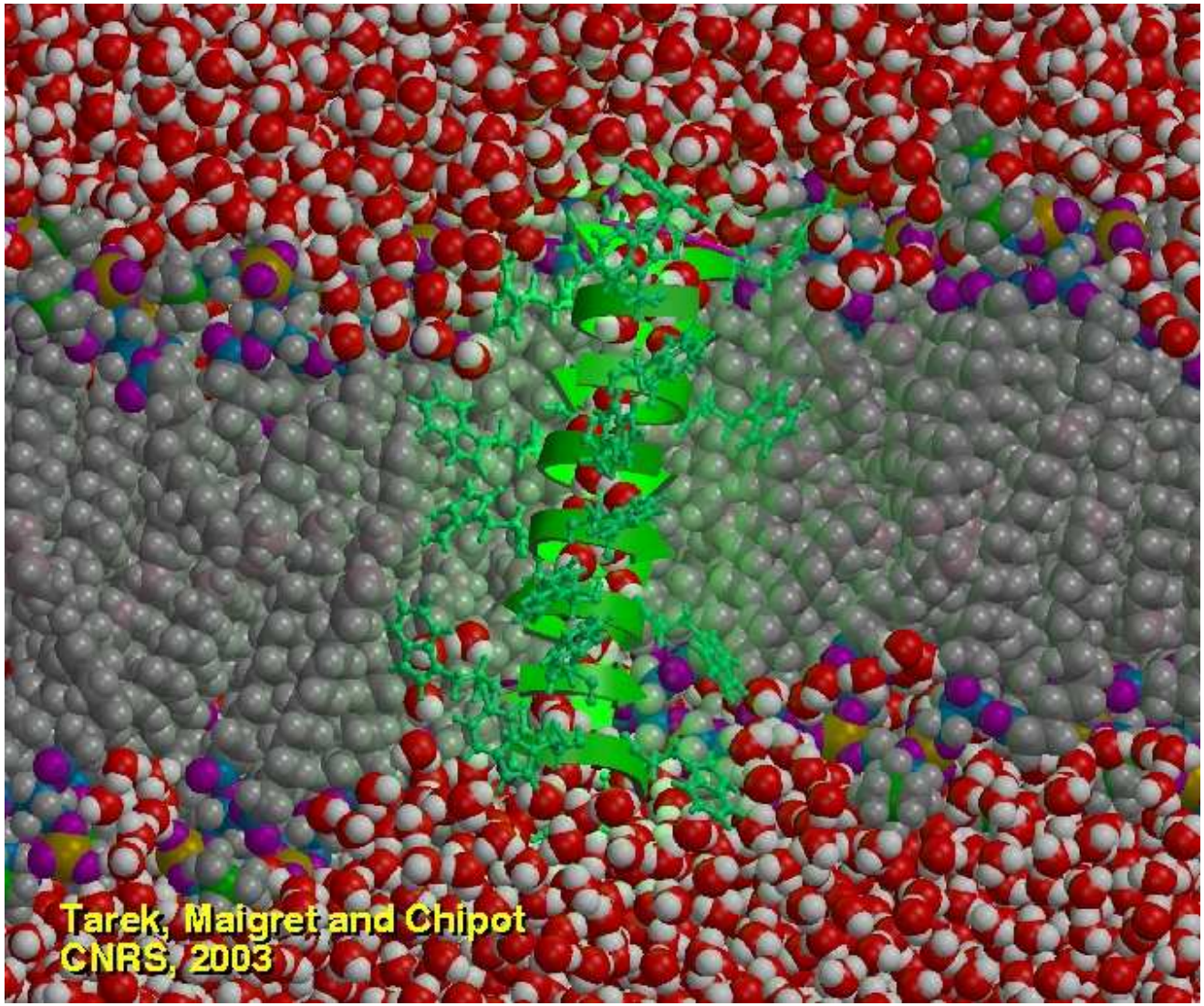
Main ingredient

Consider a system composed of N atoms, whose positions are denoted by a vector

$$(\boldsymbol{x}_1, ..., \boldsymbol{x}_N) = \boldsymbol{x} \in \mathbb{R}^{3N}.$$

The basic ingredient to describe the atomic system:
a potential V which associates to a configuration $(\boldsymbol{x}_1, ..., \boldsymbol{x}_N) = \boldsymbol{x} \in \mathbb{R}^{3N}$ an energy $V(\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$, and characterizes the interactions between the different atoms in the system.

Example of a biological system



Courtesy of Chris Chipot, CNRS Nancy

Introduction

Newton equations of motion:

$$\begin{cases} d\boldsymbol{X}_t = M^{-1}\boldsymbol{P}_t\,dt, \\ d\boldsymbol{P}_t = -\nabla V(\boldsymbol{X}_t)\,dt, \end{cases}$$

where $\boldsymbol{X}_t = (\boldsymbol{X}_t^1, \cdots, \boldsymbol{X}_t^N)$ are the positions (respectively $\boldsymbol{P}_t = (\boldsymbol{P}_t^1, \cdots, \boldsymbol{P}_t^N)$ are the momenta) of the N atoms composing the molecule, and M a diagonal matrix containing the masses of the different atoms.

Introduction

Newton equations of motion + thermostat: Langevin dynamics:

$$\begin{cases} d\boldsymbol{X}_t = M^{-1}\boldsymbol{P}_t\,dt, \\ d\boldsymbol{P}_t = -\nabla V(\boldsymbol{X}_t)\,dt - \gamma M^{-1}\boldsymbol{P}_t\,dt + \sqrt{2\gamma\beta^{-1}}\,d\boldsymbol{W}_t, \end{cases}$$

where $\gamma > 0$, $\beta = (k_B T)^{-1}$ is proportional to the inverse of the temperature, and $\boldsymbol{W}_t$ is a $3N$-dimensional Brownian motion.

Introduction

Newton equations of motion + thermostat: Langevin dynamics:

$$\begin{cases} d\boldsymbol{X}_t = M^{-1}\boldsymbol{P}_t\,dt, \\ d\boldsymbol{P}_t = -\nabla V(\boldsymbol{X}_t)\,dt - \gamma M^{-1}\boldsymbol{P}_t\,dt + \sqrt{2\gamma\beta^{-1}}d\boldsymbol{W}_t, \end{cases}$$

where $\gamma > 0$, $\beta = (k_B T)^{-1}$ is proportional to the inverse of the temperature, and $\boldsymbol{W}_t$ is a $3N$-dimensional Brownian motion.

In the following, we focus on the *over-damped Langevin* (or gradient) dynamics:

$$d\boldsymbol{X}_t = -\nabla V(\boldsymbol{X}_t)\,dt + \sqrt{2\beta^{-1}}d\boldsymbol{W}_t,$$

Time discretization (Euler-Maruyama scheme): $\Delta t > 0$

$$\boldsymbol{X}_{n+1} = \boldsymbol{X}_n - \nabla V(\boldsymbol{X}_n)\,\Delta t + \sqrt{2\Delta t\beta^{-1}}\boldsymbol{G}_n$$

where $\boldsymbol{G}_0, \cdots, \boldsymbol{G}_n, \cdots$ are independent gaussian variables $\sim \mathcal{N}(0,1)$.

Ergodicity property

The over-damped Langevin dynamics is ergodic with respect to the Gibbs measure $\mu_{V,\beta}(d\boldsymbol{x})$ with

$$\mu_{V,\beta}(d\boldsymbol{x}) = Z_{V,\beta}^{-1} \exp(-\beta V(\boldsymbol{x}))\, d\boldsymbol{x},$$

where $Z_{V,\beta} = \int \exp(-\beta V(\boldsymbol{x}))\, d\boldsymbol{x}$.

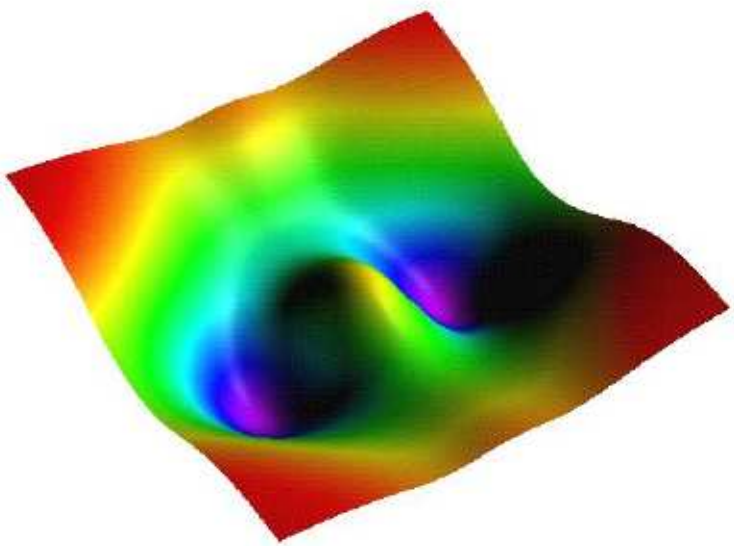
This property is extremely important to compute macroscopic quantities, in particular thermodynamic quantities (averages wrt $\mu_{V,\beta}$ of some observables): stress, heat capacity,...

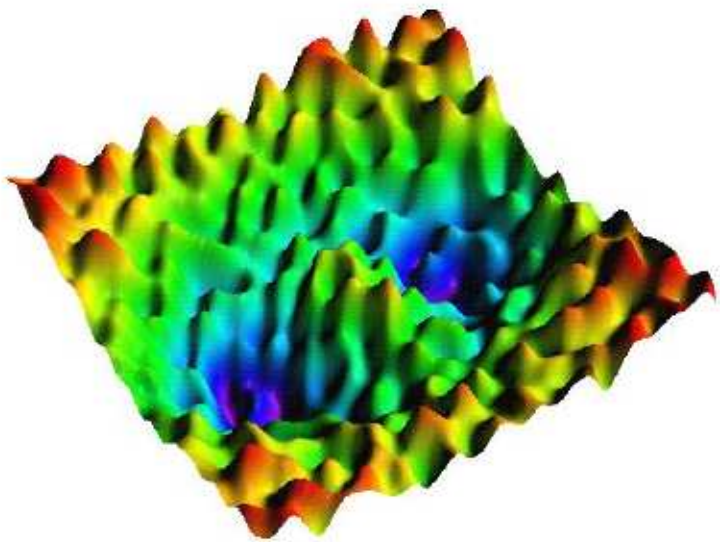
Indeed,

$$\mathbb{E}_{\mu_{V,\beta}}(\varphi(\boldsymbol{X})) = \int_{\mathbb{R}^{3N}} \varphi(\boldsymbol{x})\, \mu_{V,\beta}(d\boldsymbol{x}) = \lim_{T\to+\infty} \frac{1}{T} \int_0^T \varphi(\boldsymbol{X}_t)\, dt.$$

Difficulty: V has several local minima. As a consequence, $\boldsymbol{X}_t$ is a metastable process and $\mu_{V,\beta}$ is a multimodal measure.

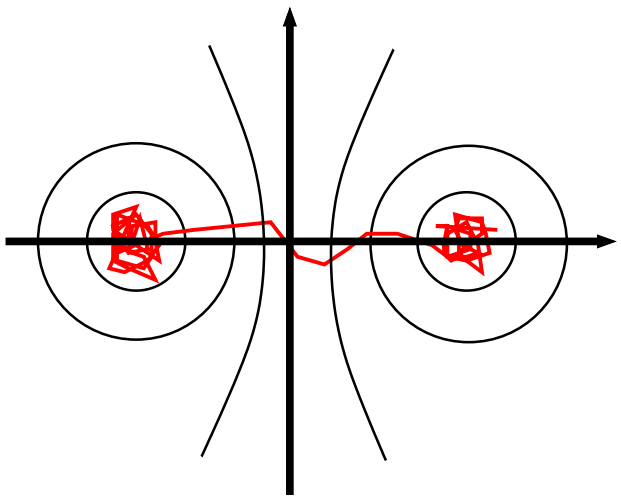
Potential energy landscapes

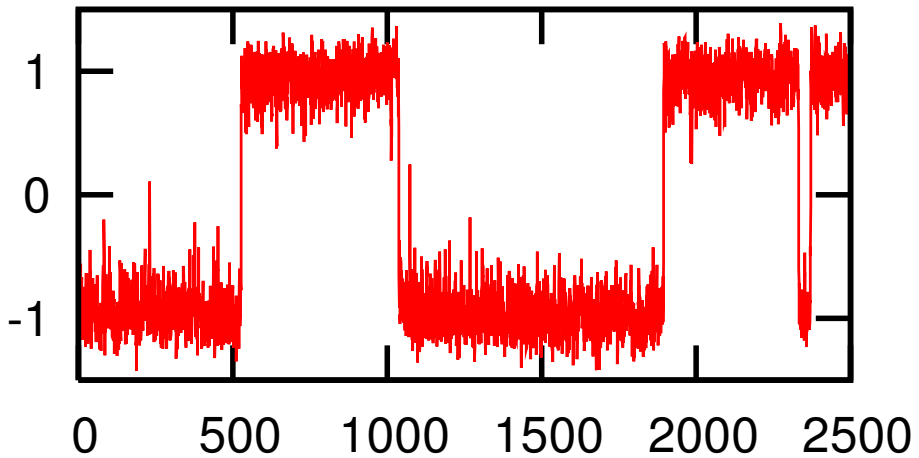




Metastability

- In practice, the dynamics is metastable: the system stays a long time in a well of V before jumping to another well:

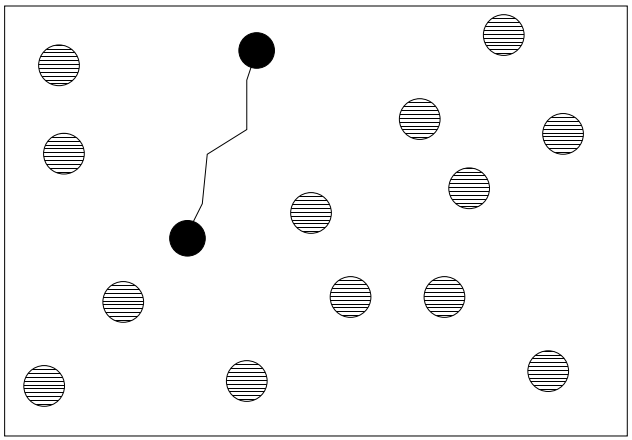


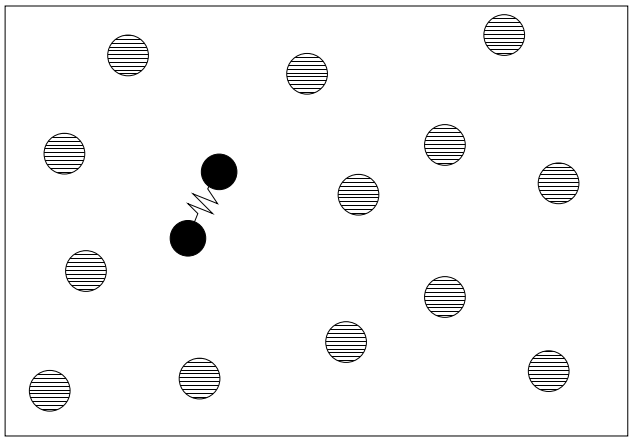


- Timescale of the microscopic process: 10^{-15} sec.
- Timescale of the macroscopic process: from 10^{-6} to 10^{2} sec.
- Slow convergence of trajectorial averages
- Transitions between metastable states are rare events

Outline of the talk

Another example (Dellago & Geissler): dimer in solution





- solvent-solvent, solvent-monomer: truncated LJ on $r = \|x_i - x_j\|$:
  $V_{WCA}(r) = 4\epsilon \left(\frac{\sigma^{12}}{r^{12}} - 2\,\frac{\sigma^6}{r^6} \right)$ if $r \leq \sigma$, 0 otherwise (repulsive potential)
- monomer-monomer: double well on $r = \|x_1 - x_2\|$

The distance between the two monomers evolves more slowly than the rest of the system.

In practice, quantities of interest depend on a few variables only.

Reaction coordinates

Let us assume that we are given an appropriate collection of slow variables: $\xi(\boldsymbol{X})$, where $\xi : \mathbb{R}^{3N} \to \mathbb{R}^d$, so that $d \ll N$ and $\xi(\boldsymbol{X})$ enables to properly characterize coarse-grained degrees of freedom of the system.

The vector ξ is called a vector of reaction coordinates or collective variables, and will be used to bias the dynamics (adaptive importance sampling technique).

These reaction coordinates can either be intuited by chemists or physicists, or be identified by machine learning techniques (diffusion maps, nonlinear manifold learning...)

Kevrekidis, Goldsmith, Arroyo, Clementi, ...

Free energy

(based on works by T. Lelièvre, F. Legoll, M. Rousset and G. Stoltz)

In the general case, we have:

- The image of the measure $\mu_{V,\beta}$ by ξ is a measure on $\mathbb{R}^d$:

$$\xi_* \mu_{V,\beta}(dz) = \exp(-\beta A(z))\, dz$$

where the free energy A is defined as follows:

$$\forall z \in \mathbb{R}^d, \quad A(z) = -\beta^{-1} \ln \left(\int_{S(z)} e^{-\beta V(\boldsymbol{x})} \delta_{\xi(\boldsymbol{x})-z}(d\boldsymbol{x}) \right),$$

with $S(z) = \{\boldsymbol{x},\, \xi(\boldsymbol{x}) = z\} \subset \mathbb{R}^{3N}$.

Adaptive biasing techniques

The bottom line of adaptive methods is the following: for "well chosen" ξ, the potential $V - A \circ \xi$ is "less rugged" than V.

Problem: A is unknown ! Idea: use a time dependent potential of the form

$$\mathcal{V}_t(\boldsymbol{x}) = V(\boldsymbol{x}) - A_t(\xi(\boldsymbol{x}))$$

where A_t is an approximation at time t of A, given the configurations visited so far.

Hopes:

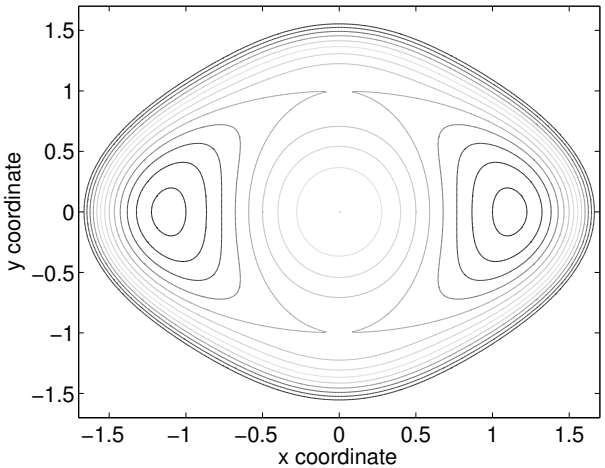
- build a dynamics which goes quickly to equilibrium,
- compute free energy profiles.

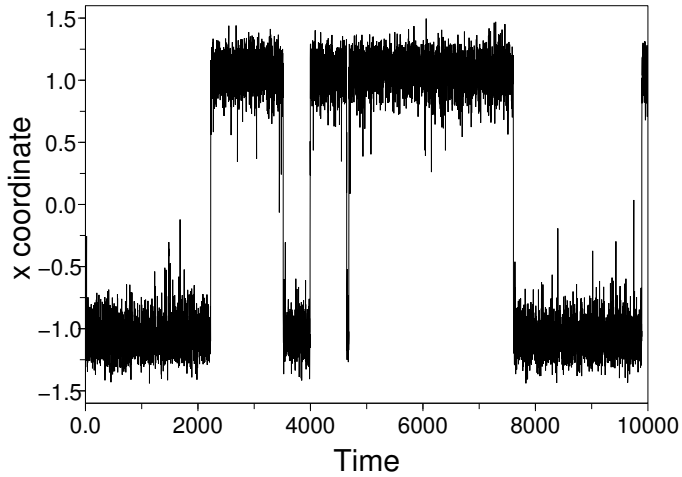
Wang-Landau, ABF, metadynamics: *Darve, Pohorille, Hénin, Chipot, Laio, Parrinello, Wang, Landau,...*

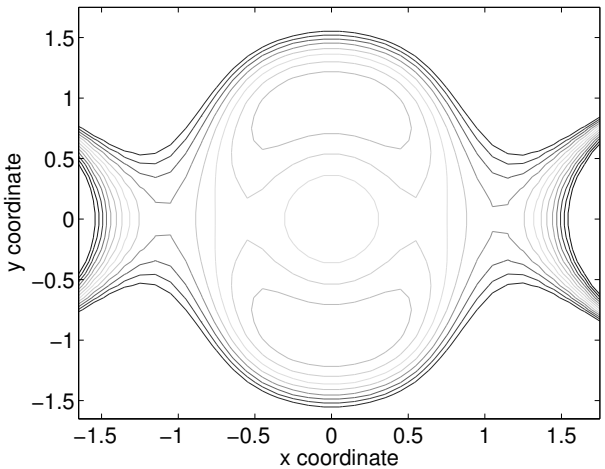
Free energy biased potential

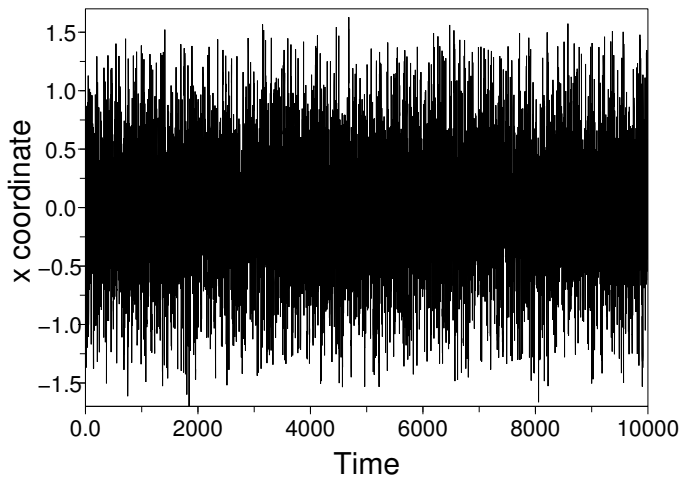
A 2D example of a free energy biased trajectory: energetic barrier.

$\xi(x, y) = x$









The standard ABF method

For the sake of simplicity, let us assume here that $d = 1$.

For the Adaptive Biasing Force (ABF) method, the idea is to use the formula

$$\nabla_z A(z) = \int_{\Sigma(z)} f(\boldsymbol{x})\, d\mu_{\Sigma(z)} = \mathbb{E}_\mu \left[f(\boldsymbol{X}) | \xi(\boldsymbol{X}) = z \right],$$

where f is an explicit function depending on V, β, ξ.

The mean force $\nabla_z A(z)$ is the average of f with respect to $\mu_{S(z)}$, where $\mu_{S(z)}$ is the probability measure $\mu_{V,\beta}$ conditioned to $\xi(\boldsymbol{x}) = z$.

Notice that actually, whatever A_t is,

$$\nabla_z A(z) = \frac{\displaystyle\int_{\Sigma(z)} f(\boldsymbol{x})\, e^{-\beta(V - A_t \circ \xi)}\, \delta_{\xi(\boldsymbol{x}) - z}}{\displaystyle\int_{\Sigma(z)} e^{-\beta(V - A_t \circ \xi)}\, \delta_{\xi(\boldsymbol{x}) - z}}.$$

The standard ABF method

Thus, we would like to simulate:

$$\begin{cases} d\boldsymbol{X}_t = -\nabla(V - A \circ \xi)(\boldsymbol{X}_t)\, dt + \sqrt{2\beta^{-1}} d\boldsymbol{W}_t, \\ \nabla_z A(z) = \mathbb{E}_{\mu_{V,\beta}}\left(f(\boldsymbol{X}) | \xi(\boldsymbol{X}) = z\right) \end{cases}$$

but A is unknown...

The standard ABF method

Thus, we would like to simulate:

$$\begin{cases} d\boldsymbol{X}_t = -\nabla(V - A \circ \xi)(\boldsymbol{X}_t)\, dt + \sqrt{2\beta^{-1}} d\boldsymbol{W}_t, \\ \nabla_z A(z) = \mathbb{E}_{\mu_{V,\beta}} \left(f(\boldsymbol{X}) | \xi(\boldsymbol{X}) = z \right) \end{cases}$$

but A is unknown...

The ABF dynamics reads:

$$\begin{cases} d\boldsymbol{X}_t = -\nabla(V - A_t \circ \xi)(\boldsymbol{X}_t)\, dt + \sqrt{2\beta^{-1}} d\boldsymbol{W}_t, \\ \nabla_z A_t(z) = \mathbb{E} \left(f(\boldsymbol{X}_t) | \xi(\boldsymbol{X}_t) = z \right). \end{cases}$$

The standard ABF method

Thus, we would like to simulate:

$$\begin{cases} d\boldsymbol{X}_t = -\nabla(V - A \circ \xi)(\boldsymbol{X}_t)\, dt + \sqrt{2\beta^{-1}} d\boldsymbol{W}_t, \\ \nabla_z A(z) = \mathbb{E}_{\mu_{V,\beta}} \left(f(\boldsymbol{X}) | \xi(\boldsymbol{X}) = z \right) \end{cases}$$

but A is unknown...

The ABF dynamics reads:

$$\begin{cases} d\boldsymbol{X}_t = -\nabla(V - A_t \circ \xi)(\boldsymbol{X}_t)\, dt + \sqrt{2\beta^{-1}} d\boldsymbol{W}_t, \\ \nabla_z A_t(z) = \mathbb{E}\left(f(\boldsymbol{X}_t) | \xi(\boldsymbol{X}_t) = z \right). \end{cases}$$

[Lelièvre, Otto, Rousset, Stoltz, 2007]...

Under suitable assumptions, it holds that $\nabla_z A_t$ converges (in some sense) to $\nabla_z A$ as t goes to infinity.

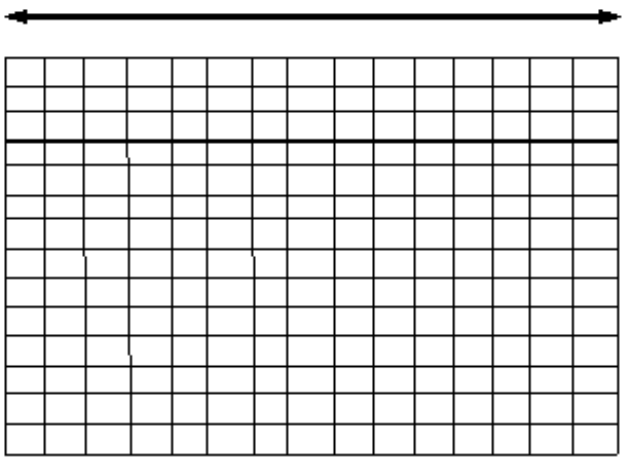
Curse of dimensionality

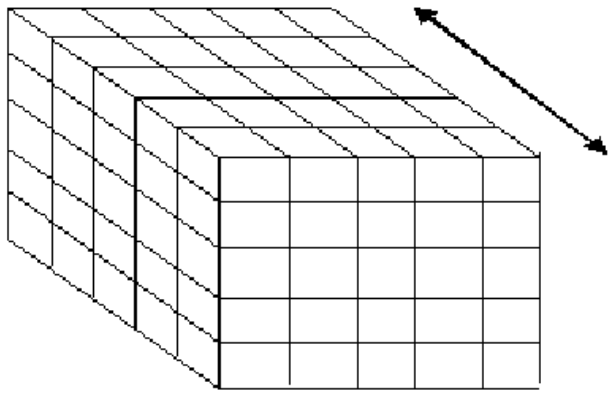
Problem: When the number of relevant reaction coordinates d becomes large (say a few dozens), standard ABF methods suffer from the curse of dimensionality.

$A(z)$ (or $\nabla_z A(z)$) has to be discretized on a grid whose size scales exponentially with respect to d:

$$DIM = P^2$$

$$DIM = P^3$$





$$DIM = P^d$$

Tensor approximation

One method to circumvent this curse of dimensionality is to use tensor methods.

Rough idea: approximate $A(z)$ under the following form

$$A(z) \approx \sum_{k=1}^{n} r_k^1(z_1) \cdots r_k^d(z_d)$$

with n small (hopefully).

$$\boxed{DIM = nPd}$$

Goal of the talk: Propose a modified ABF algorithm, well-suited for the use of tensor approximations, in order to circumvent the curse of dimensionality.

Outline of the talk

Extended ABF method

[Zhao, Fu, Lelièvre, Shao, Chipot, Cai, 2017]...

From now on, for the sake of simplicity, we will assume that the molecular system is confined in a finite domain $\mathcal{D}_{\boldsymbol{x}} = [-L, L]^N$ with periodic boundary conditions.

Besides, let us assume that $\xi : \mathcal{D}_{\boldsymbol{x}} \to \mathcal{D}_z := \Pi_{i=1}^d \mathcal{D}_{z_i}$, where $\mathcal{D}_{z_i}$ is a closed bounded interval of $\mathbb{R}$.

The Extended ABF (EABF) method is a variant of the standard ABF method, which will be adapted in our context.

It consists in introducing an auxiliary variable $z \in \mathcal{D}_z$ and an auxiliary extended potential $W_\delta : \mathcal{D}_{\boldsymbol{x}} \times \mathcal{D}_z \to \mathbb{R}$ defined by

$$W_\delta(\boldsymbol{x}, z) := V(\boldsymbol{x}) + \frac{1}{2\delta} \sum_{i=1}^{d} |\xi_i(\boldsymbol{x}) - z_i|^2$$

for a given *small* parameter $\delta > 0$.

Extended ABF method

The idea of the EABF method is to apply a standard ABF method to the extended system $(\boldsymbol{x}, z) \in \mathcal{D}_{\boldsymbol{x}} \times \mathcal{D}_z$ characterized by the interaction potential W_δ with reaction coordinates z.

The free energy of this system is then

$$A_\delta(z) = -\frac{1}{\beta} \ln \left(\int_{\mathcal{D}_{\boldsymbol{x}}} e^{-\beta W_\delta(\boldsymbol{x}, z)} \, d\boldsymbol{x} \right)$$

Denoting by

$$K_\delta(y, z) := \Pi_{i=1}^{d} e^{-(y_i - z_i)^2/\delta},$$

this implies that

$$\exp\left(-\beta A_\delta(z)\right) = \int_{\mathcal{D}_{\boldsymbol{x}}} e^{-\beta W_\delta(\boldsymbol{x}, z)} \, d\boldsymbol{x} = \int_{\mathcal{D}_z} K_\delta(y, z) \exp\left(-\beta A(y)\right) \, dy$$

$\exp\left(-\beta A_\delta(z)\right)$ is the convolution of $\exp\left(-\beta A(z)\right)$ with a Gaussian kernel of width δ. A_δ is thus an approximation of A for $\delta > 0$ small.

Aim of the modified EABF method

Our aim is to define for all time $t \geq 0$ a bias function A_t which

- will be easy to compute with tensor approximations;
- will converge in the long time limit to an approximation of A_δ

and to sample the process

$$\begin{cases} d\boldsymbol{X}_t = -\nabla_{\boldsymbol{x}} W_\delta(\boldsymbol{X}_t, Z_t)\, dt + \sqrt{2\beta^{-1}}\, dW_t^1, \\ dZ_t = -\nabla_z W_\delta(\boldsymbol{X}_t, Z_t)\, dt + \nabla_z A_t(Z_t) dt + \sqrt{2\beta^{-1}}\, dW_t^2, \end{cases}$$

where W_t^1 and W_t^2 are two independent Brownian motions.

Equivalent reformulation of A_δ

Note that A_δ (up to an additive constant) is the unique minimizer of

$$\mathcal{J}_\delta(f) := \int_{\mathcal{D}_{\boldsymbol{x}} \times \mathcal{D}_z} |\nabla_z W_\delta(\boldsymbol{x}, z) - \nabla_z f(z)|^2 e^{-\beta W_\delta(\boldsymbol{x}, z)} \, d\boldsymbol{x} \, dz$$

over the set

$$\left\{ f \in H^1(\mathcal{D}_z), \int_{\mathcal{D}_z} f = 0 \right\} =: H^{1,0}(\mathcal{D}_z).$$

General procedure of the (ideal) algorithm

Let $T > 0$ and $T_n := nT$ for all $n \in \mathbb{N}$.

- Set $n = 0$, $t_0 = 0$ and $A_0 = 0$;
- Suppose that A_{T_n} has already been defined. Then, sample the process with $A_t = A_{T_n}$ for time $T_n \leq t \leq T_{n+1}$.
- At time T_{n+1}, define $A_{T_{n+1}} := g_{n+1}$ where g_{n+1} is the unique minimizer of

$$\operatorname*{argmin}_{f \in H^{1,0}(\mathcal{D}_z)} \mathcal{J}_{T_{n+1}}(f) \tag{1}$$

  where $\mathcal{J}_{T_{n+1}}$ will be defined later.
- Set $n := n + 1$.

Unbiased empirical measure of the process

At time $t \geq 0$, a sample $(\boldsymbol{X}_s, Z_s)_{0 \leq s \leq t}$ of the stochastic process defined above is available. The unbiased empirical measure ν_t is defined by

$$\int_{\mathcal{D}_{\boldsymbol{x}} \times \mathcal{D}_z} f(\boldsymbol{x}, z)\, d\nu_t(\boldsymbol{x}, z) = \frac{\int_0^t f(\boldsymbol{X}_s, Z_s) w_s\, ds}{\int_0^t w_s\, ds}$$

where $w_s := \exp(-\beta A_s(Z_s))$.

This empirical measure is expected to converge to $\mu_{W_\delta, \beta} \sim e^{-\beta W_\delta(\boldsymbol{x}, z)}$ as t goes to infinity.

As a consequence, we would like to define

$$\mathcal{J}_t(f) := \int_{\mathcal{D}_{\boldsymbol{x}} \times \mathcal{D}_z} |\nabla_z W_\delta(\boldsymbol{x}, z) - \nabla_z f(z)|^2 \nu_t(\boldsymbol{x}, z)\, d\boldsymbol{x}\, dz$$

Problem: The measure ν_t is singular so that the minimization problem (1) is not well-posed if we choose $\mathcal{J}_t$ as above. We have to introduce another regularization!

Cost functional

Let $\epsilon > 0$ and let

$$K_\epsilon(y, z) := \Pi_{i=1}^{d} e^{-|y_i - z_i|^2/\epsilon}.$$

The actual cost functional will be

$$\mathcal{J}_t(f) := \int_{\mathcal{D}_{\boldsymbol{x}} \times \mathcal{D}_z \times \mathcal{D}_z} |\nabla_z W_\delta(\boldsymbol{x}, y) - \nabla_z f(z)|^2 \nu_t(\boldsymbol{x}, y) K_\epsilon(y, z)\, d\boldsymbol{x}\, dz\, dy,$$

so that

$$\mathcal{J}_t(f) := \int_{\mathcal{D}_z} |F_t(z) - \nabla_z f(z)|^2 \theta_t(z)\, dz,$$

where

$$\theta_t(z) := \left(\int_0^t w_s\, ds \right)^{-1} \int_0^t K_\epsilon(Z_s, z) w_s\, ds$$

and

$$F_t(z) := \left(\int_0^t K_\epsilon(Z_s, z) w_s\, ds \right)^{-1} \int_0^t \nabla_z W_\delta(\boldsymbol{X}_s, Z_s) K_\epsilon(Z_s, z)\, ds$$

Long-time convergence of the ideal algorithm

Theorem (VE, Lelièvre, Monmarché, 2017)

For all $n \in \mathbb{N}$, minimization problem (1) is well-defined in the sense that there exists a unique minimizer. Besides, the following convergence holds almost surely

$$\|\nabla A_t(z) - \nabla A_{\delta,\epsilon}(z)\|^2_{L^2(\mathcal{D}_z)} \underset{t\to+\infty}{\longrightarrow} 0,$$

where $A_{\delta,\epsilon}(z)$ is the unique minimizer in $H^{1,0}(\mathcal{D}_z)$ of the functional

$$\mathcal{J}_{\delta,\epsilon}(f) := \int_{\mathcal{D}_{\boldsymbol{x}}\times\mathcal{D}_z\times\mathcal{D}_z} |\nabla_z W_\delta(\boldsymbol{x}, y) - \nabla_z f(z)|^2 e^{-\beta W_\delta(\boldsymbol{x},y)} K_\epsilon(y,z)\, d\boldsymbol{x}\, dz\, dy.$$

The proof relies on an adaptation of the so-called ODE method of Benaïm and Bréhier.

Outline of the talk

Back to the minimization problem

Each iteration of the algorithm requires the resolution of minimization problems of the following form:

find $g \in H^{1,0}(\mathcal{D}_z)$ the unique minimizer of

$$\min_{f \in H^{1,0}(\mathcal{D}_z)} \mathcal{J}_t(f),$$

where

$$\mathcal{J}_t(f) := \int_{\mathcal{D}_z} |F_t(z) - \nabla_z f(z)|^2 \theta_t(z)\, dz.$$

Recall that

$$H^{1,0}(\mathcal{D}_z) = \left\{ f \in H^1(\mathcal{D}_z), \quad \int_{\mathcal{D}_z} f = 0 \right\}.$$

When d is large, this minimization problem cannot be solved by standard methods.

General (standard) greedy algorithm

Let R be a Hilbert space of functions depending on the d variables $z_1, \cdots, z_d$, and let

$$g = \text{argmin}_{f \in R} \mathcal{J}(f). \tag{2}$$

A standard tensor greedy algorithm for the resolution of problem (1) reads as follows:

- set $n = 0$ and $g_0 = 0$.
- Iteration n: compute $h_{n+1} \in \Sigma$ solution to

$$h_{n+1} \in \underset{h \in \Sigma}{\text{argmin}}\, \mathcal{J}(g_n + h).$$

Define $g_{n+1} := g_n + h_{n+1}$ and set $n := n + 1$;

where Σ is a dictionary of pure tensor-product functions:

$$\Sigma = \{r_1(z_1) \cdots r_d(z_d), \quad r_1 \in R_1, \cdots, r_d \in R_d\}$$

where R_i is some Hilbert space of functions depending only on the variable z_i.

Remark: Other more advanced tensor formats can be chosen as dictionary (Tucker, Tensor Train, Hierarchical Tensor Train...)

Convergence of the greedy algorithm

[Cancès, Lelièvre, VE, 2011], [Falco, Nouy, 2012]

Under the following assumptions:

(A1) $\mathcal{J}$ is strongly convex on R;

(A2) $\mathcal{J}$ is differentiable and Lipschitz on bounded subsets of R;

(A3) Σ is weakly closed in R;

(A4) Span Σ is dense in R;

it holds that $(g_n)_{n\in\mathbb{N}^*}$ strongly converges to g in R.

Back to the modified ABF context

Recall that we are interested in finding $g \in R$ the unique minimizer of

$$\min_{f \in R} \mathcal{J}_t(f),$$

where

$$\mathcal{J}_t(f) := \int_{\mathcal{D}_z} |F_t(z) - \nabla_z f(z)|^2 \theta_t(z)\, dz.$$

and

$$R = H^{1,0}(\mathcal{D}_z) = \left\{ f \in H^1(\mathcal{D}_z), \quad \int_{\mathcal{D}_z} f = 0 \right\}.$$

What should be a proper choice of spaces R_i to ensure the convergence of a greedy algorithm?

Choices which do **not** work:

- $R_i := \left\{ r_i \in H^1(\mathcal{D}_{z_i}), \cdots, \int_{\mathcal{D}_{z_i}} r_i = 0 \right\}$;
- $R_1 = \left\{ r_1 \in H^1(\mathcal{D}_{z_1}), \cdots, \int_{\mathcal{D}_{z_1}} r_1 = 0 \right\}$ and $R_i = H^1(\mathcal{D}_{z_i})$ for $i \neq 1$;

Modified greedy algorithm

Actually, we have to introduce d dictionaries:

$$\Sigma_i := \left\{ r_1(z_1) \cdots r_i(z_i) \cdots r_d(z_d), \quad r_i \in H^{1,0}(\mathcal{D}_{z_i}), \quad r_j \in H^1(\mathcal{D}_{z_j}),\ j \neq i \right\}.$$

A modified greedy algorithm is needed to have a converging sequence of approximations:

- set $n = 0$ and $g_0 = 0$.
- Iteration n: set $g_{n,0} = g_n$.

  For $i = 1, \cdots, d$, compute $h_{n+1,i} \in \Sigma_i$ solution to

$$h_{n+1,i} \in \underset{h \in \Sigma_i}{\operatorname{argmin}}\, \mathcal{J}_t(g_{n,i-1} + h). \tag{3}$$

  Set $g_{n,i} := g_{n,i-1} + h_{n+1,i}$.

  Set $g_{n+1} = g_{n,d}$ and $n := n + 1$.

Convergence of the modified PGD algorithm

Theorem (VE, Lelièvre, Monmarché, 2017)

Each iteration of the modified PGD algorithm is well-defined in the sense that there always exists at least one minimizer to (3). Besides, the sequence $(g_n)_{n\in\mathbb{N}^*}$ *strongly converges in* $H^1(\mathcal{D}_z)$ *to* g.

Conclusions and perspectives

An idealized modified version of the ABF method, well-suited for tensor approximations, is proved to converge in the long time limit to some approximation of the free energy of the molecular system.

A provably convergent greedy algorithm is proposed to approximate the solutions of the intermediate problems by tensors in the case when the number of reaction coordinates is significant.

- Numerical implementation of the algorithm on real molecular systems;
- Appropriate choice of δ, ϵ?
- Theoretical convergence results on an intertwinned ABF/greedy procedure?

Thank you for your attention!