Numerical methods in molecular dynamics

Tony Lelièvre

École des Ponts ParisTech and INRIA

Maison de la simulation, March 2018
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.

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

The basic ingredient: a potential $V$ which associates to a configuration $(x_1, \ldots, x_N) = x \in \mathbb{R}^{3N}$ an energy $V(x_1, \ldots, x_N)$. 

Introduction

Typically, $V$ is a sum of potentials modelling interaction between two particles, three particles and four particles:

$$V = \sum_{i<j} V_1(x_i, x_j) + \sum_{i<j<k} V_2(x_i, x_j, x_k) + \sum_{i<j<k<l} V_3(x_i, x_j, x_k, x_l).$$

For example,

$$V_1(x_i, x_j) = V_{LJ}(|x_i - x_j|)$$

where

$$V_{LJ}(r) = 4\epsilon \left( \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right)$$

is the Lennard-Jones potential.
Introduction

Newton equations of motion:

\[
\begin{align*}
    dX_t &= M^{-1}P_t \, dt, \\
    dP_t &= -\nabla V(X_t) \, dt,
\end{align*}
\]
Newton equations of motion + thermostat: Langevin dynamics:

\[
\begin{cases}
    dX_t = M^{-1} P_t \, dt, \\
    dP_t = -\nabla V(X_t) \, dt - \gamma M^{-1} P_t \, dt + \sqrt{2\gamma \beta^{-1}} dW_t,
\end{cases}
\]

where $\gamma > 0$. Langevin dynamics is ergodic wrt

$$d\mu(d\mathbf{x}) \otimes Z_p^{-1} \exp \left(-\beta \frac{p^t M^{-1} p}{2}\right) \, dp$$

with

$$d\mu = Z^{-1} \exp(-\beta V(\mathbf{x})) \, d\mathbf{x},$$

where $Z = \int \exp(-\beta V(\mathbf{x})) \, d\mathbf{x}$ is the partition function and $\beta = (k_B T)^{-1}$ is proportional to the inverse of the temperature.
Introduction

Newton equations of motion + thermostat: Langevin dynamics:

\[
\begin{aligned}
    dX_t &= M^{-1}P_t dt, \\
    dP_t &= -\nabla V(X_t) dt - \gamma M^{-1}P_t dt + \sqrt{2\gamma\beta^{-1}} dW_t,
\end{aligned}
\]

where \( \gamma > 0 \). Langevin dynamics is ergodic wrt \( \mu(dx) \otimes Z_p^{-1} \exp \left( -\beta \frac{p^t M^{-1}p}{2} \right) dp \) with

\[ d\mu = Z^{-1} \exp( -\beta V(x)) dx, \]

where \( Z = \int \exp( -\beta V(x)) dx \) is the partition function and \( \beta = (k_B T)^{-1} \) is proportional to the inverse of the temperature.

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

\[ dX_t = -\nabla V(X_t) dt + \sqrt{2\beta^{-1}} dW_t, \]

which is also ergodic wrt \( \mu \).
Introduction

These dynamics are used to compute macroscopic quantities:

(i) **Thermodynamic quantities** (averages wrt $\mu$ of some observables): stress, heat capacity, free energy, ...

$$
\mathbb{E}_\mu(\varphi(\mathbf{X})) = \int_{\mathbb{R}^d} \varphi(\mathbf{x}) \mu(d\mathbf{x}) \simeq \frac{1}{T} \int_0^T \varphi(\mathbf{X}_t) \, dt.
$$

(ii) **Dynamical quantities** (averages over trajectories): ensemble of reactive paths, transition states, transition rates, ...

$$
\mathbb{E}(\mathcal{F}((\mathbf{X}_t)_{t \geq 0})) \simeq \frac{1}{M} \sum_{m=1}^M \mathcal{F}((\mathbf{X}^m_t)_{t \geq 0}).
$$

**Difficulties:** (i) high-dimensional problem ($N \gg 1$); (ii) $\mathbf{X}_t$ is a metastable process and $\mu$ is a multimodal measure.
Metastability: energetic and entropic barriers

A two-dimensional schematic picture

Transitions between metastable states are rare events
A toy example in material sciences
The 7 atoms Lennard Jones cluster in 2D.

Figure: Low energy conformations of the Lennard-Jones cluster.
Simulations of biological systems

Unbinding of a ligand from a protein

(Diaminopyridine-HSP90, Courtesy of SANOFI)

Elementary time-step for the molecular dynamics $= 10^{-15}\text{s}$
Dissociation time $= 0.5\text{s}$
Introduction

For computing thermodynamics quantities, there is a clear classification of available methods (biasing, constraining, tempering), and the difficulties are now well understood (in particular for free energy computations, see for example [TL, Rousset, Stoltz, 2010]). On the opposite, computing efficiently dynamical quantities remains a challenge.

In this talk, we would like to discuss two methods to sample metastable trajectories: the Adaptive Multilevel Splitting method, and the Parallel Replica algorithm.

There are many other techniques: accelerated dynamics [Voter, Fichthorn], the string method [E, Ren, Vanden-Eijnden], transition path sampling methods [Chandler, Bolhuis, Dellago], milestoning techniques [Elber, Schuette, Vanden-Eijnden], importance sampling approaches [Dupuis, Vanden-Einjden, Weare, Schuette, Hartmann] etc...
Splitting strategies
Multilevel splitting

We would like to sample trajectories between two given metastable states $A$ and $B$. The main assumption is that we are given a one dimensional function $\xi : \mathbb{R}^d \to \mathbb{R}$ (s.t. $|\nabla \xi| \neq 0$) which "indexes" the transition from $A$ to $B$ in the following sense:

\[
A \subset \{x \in \mathbb{R}^d, \xi(x) < z_{\text{min}}\} \quad \text{and} \quad B \subset \{x \in \mathbb{R}^d, \xi(x) > z_{\text{max}}\},
\]

where $z_{\text{min}} < z_{\text{max}}$, and $\Sigma z_{\text{min}}$ (resp. $\Sigma z_{\text{max}}$) is “close” to $\partial A$ (resp. $\partial B$).

Example: $\xi(x) = \|x - x_A\|$ where $x_A \in A$ is a reference configuration in $A$. We are interested in the event $\{\tau_A < \tau_B\}$, starting from an initial condition on $\Sigma z_{\text{min}}$, where

\[
\tau_A = \inf\{t > 0, \ X_t \in A\}, \quad \tau_B = \inf\{t > 0, \ X_t \in B\}
\]

and

\[
\tau_z = \inf\{t > 0, \ \xi(X_t) > z\}.
\]
**Multilevel splitting**

**Question:** How to compute dynamical quantities using $\xi$? More precisely, we consider: (a) Reactive trajectories and (b) Transition times between the two metastable states $A$ and $B$.

We propose a **multilevel splitting approach** [Kahn, Harris, 1951] [Rosenbluth, 1955] to discard failed trajectories and branch trajectories approaching the rare set. We focus on an adaptive variant [Cérou, Guyader, 2007] [Cérou, Guyader, TL, Pommier, 2011]: the **Adaptive Multilevel Splitting** (AMS) algorithm.

Remark: The algorithm can be seen as a kind of adaptive Forward Flux Sampling [Allen, Valeriani, Ten Wolde, 2009]. It is also related to the Interface Sampling Method [Bolhuis, van Erp, Moroni 2003] and the Milestoning method [Elber, Faradjian 2004]. See the review paper [Bolhuis, Dellago, 2009]. Another splitting technique in MD: weighted ensemble method [Zuckerman, 2010].
A reactive trajectory between two metastable sets $A$ and $B$ is a piece of equilibrium trajectory that leaves $A$ and goes to $B$ without going back to $A$ in the meantime [Hummer, 2004] [Metzner, Schütte, Vanden-Eijnden, 2006].

**Difficulty**: A trajectory leaving $A$ is more likely to go back to $A$ than to reach $B$. 
Splitting algorithm: basic idea

The idea of splitting algorithms (FFS, TIS, RESTART, ...) is to write the rare event

$$\{ \tau_B < \tau_A \}$$

as a sequence of nested events: for $z_{\text{min}} < z_1 < \ldots < z_{\text{max}}$,

$$\{ \tau_{z_1} < \tau_A \} \supset \{ \tau_{z_2} < \tau_A \} \supset \ldots \supset \{ \tau_{z_{\text{max}}} < \tau_A \} \supset \{ \tau_B < \tau_A \}$$

and to simulate the successive conditional events: for $k = 1, 2, \ldots$,

$$\{ \tau_{z_q} < \tau_A \} \text{ knowing that } \{ \tau_{z_{q-1}} < \tau_A \}.$$

It is then easy to build an unbiased estimator of

$$\mathbb{P}(\tau_B < \tau_A) = \mathbb{P}(\tau_{z_1} < \tau_A)\mathbb{P}(\tau_{z_2} < \tau_A|\tau_{z_1} < \tau_A) \ldots \mathbb{P}(\tau_B < \tau_A|\tau_{z_{\text{max}}} < \tau_A)$$
Splitting algorithm: adaptive level computation

**Problem:** How to choose the intermediate levels \((z_q)_{q \geq 1}\) ?

It is easy to check, for a given number of intermediate levels, the optimum in terms of variance is attained if

\[
\mathbb{P}(\tau_{z_q} < \tau_A | \tau_{z_{q-1}} < \tau_A) \text{ is constant .}
\]

This naturally leads to adaptive versions (AMS, nested sampling) where the levels are determined by using *empirical quantiles*: choose \(k < n\), and given \(n\) trajectories \((X^m_{t \wedge \tau_A})_{t > 0, m=1,\ldots,n}\) in the event \(\{\tau_{z_{q-1}} < \tau_A\}\), choose \(z_q\) so that

\[
\mathbb{P}(\tau_{z_q} < \tau_A | \tau_{z_{q-1}} < \tau_A) \approx \left(1 - \frac{k}{n}\right).
\]

The level \(z_q\) is the \(k\)-th order statistics of \(\sup_{t \geq 0} \xi(X^{m}_{t \wedge \tau_A})\):

\[
\sup_{t \geq 0} \xi(X^{(1)}_{t \wedge \tau_A}) < \ldots < \sup_{t \geq 0} \xi(X^{(k)}_{t \wedge \tau_A}) =: z_q < \ldots < \sup_{t \geq 0} \xi(X^{(n)}_{t \wedge \tau_A}).
\]
AMS: estimator of the rare event probability (1/2)

Let \( Q_{\text{iter}} \) be the number of iterations to reach the level \( z_{\text{max}} \):

\[
Q_{\text{iter}} = \min\{q \geq 0, z_q > z_{\text{max}}\}
\]

(where \( z_0 \) is the \( k \)-th order statistics of the \( n \) initial trajectories). Then, one obtains the estimator:

\[
\left(1 - \frac{k}{n}\right)^{Q_{\text{iter}}} \sim \mathbb{P}(\tau_{z_{\text{max}}} < \tau_A).
\]
AMS: estimator of the rare event probability (2/2)

At iteration $Q_{\text{iter}}$, one has an ensemble of $n$ trajectories starting from $\Sigma_{z_{\text{min}}}$ and such that $\tau_{z_{\text{max}}} < \tau_A$. Thus

$$\hat{p}_{\text{corr}} := \frac{1}{n} \sum_{\ell=1}^{n} 1\{T_B(X^\ell, Q_{\text{iter}}) < T_A(X^\ell, Q_{\text{iter}})\} \simeq \mathbb{P}(\tau_B < \tau_A | \tau_{z_{\text{max}}} < \tau_A).$$

$\hat{p}_{\text{corr}}$ is the proportion of trajectories reaching $B$ before $A$ at the last iteration $Q_{\text{iter}}$.

Therefore, an estimator of $\mathbb{P}(\tau_B < \tau_A)$ is

$$\left(1 - \frac{k}{n}\right)^{Q_{\text{iter}}} \hat{p}_{\text{corr}}.$$
AMS Algorithm
AMS Algorithm
AMS Algorithm
AMS Algorithm
AMS Algorithm
Introduction

Splitting strategies and the AMS algorithm

The parallel replica algorithm

AMS Algorithm
AMS Algorithm: the case of Markov chains

In practice, the dynamics are *discrete in time* and thus, it may happen that more than $k$ trajectories are such that

$$
\sup_{t \geq 0} \xi(X^{\ell}_{t \wedge \tau_A}) \leq \sup_{t \geq 0} \xi(X^{(k)}_{t \wedge \tau_A}) =: z_q
$$

In this case, all the trajectories with maximum level smaller or equal than $z_q$ should be discarded.

The actual estimator of $\mathbb{P}(\tau_B < \tau_A)$ thus reads:

$$
\hat{p} = \left(1 - \frac{K_1}{n}\right) \ldots \left(1 - \frac{K_{Q_{iter}}}{n}\right) \hat{p}_{corr}
$$

instead of $(1 - \frac{k}{n})^{Q_{iter}} \hat{p}_{corr}$, where $K_q \geq k$ is the effective number of discarded trajectories at iteration $q$. 
AMS Algorithm: unbiasedness

Theorem [C.-E. Bréhier, M. Gazeau, L. Goudenège, TL, M. Rousset, 2016]: For any choice of $\xi$, $n$ and $k$,

$$\mathbb{E}(\hat{p}) = \mathbb{P}(\tau_B < \tau_A).$$

The proof is based on Doob’s stopping theorem on a martingale built using filtrations indexed by the level sets of $\xi$. Actually, this result is proved for general path observables and in a much more general setting.

Practical counterparts:

- The algorithm is easy to parallelize.
- One can compare the results obtained with different reaction coordinates $\xi$ to gain confidence in the results.
Computing transition times

To use the algorithm to compute transition times, we split a transition path from $A$ to $B$ into: excursions from $\partial A$ to $\Sigma_{z_{\text{min}}}$ and then back to $\partial A$, and finally an excursion from $\partial A$ to $\Sigma_{z_{\text{min}}}$ and then to $B$. Assuming that $A$ is metastable ($p \ll 1$), one obtains that the mean transition time is:

$$
\mathbb{E}(T) = \left( \frac{1}{p} - 1 \right) \mathbb{E}(T_1 + T_2) + \mathbb{E}(T_1 + T_3)
$$

where:

- $p$ is the probability, once $\Sigma_{z_{\text{min}}}$ has been reached, to go to $B$ rather than $A$ (approximated by $\hat{p}_N$);
- $\mathbb{E}(T_1 + T_2)$ is the mean time for an excursion from $\partial A$ to $\Sigma_{z_{\text{min}}}$ and then back to $\partial A$ (approximated by DNS);
- $\mathbb{E}(T_1 + T_3)$ is the mean time for an excursion from $\partial A$ to $\Sigma_{z_{\text{min}}}$ and then to $B$ (approximated by the AMS algorithm).
Recent results using NAMD

We are currently implementing AMS in the NAMD software (collaboration with SANOFI, C. Mayne and I. Teo, PhD of L. Silva Lopes with J. Hénin).

Three test cases:

- Alanine di-peptide (test case)
- benzamidine-trypsin dissociation rate
- $\beta$-cyclodextrin (in progress)
Two reaction coordinates:

- $\xi_1$ is a continuous piecewise affine function of $\varphi$
- $\xi_2(\varphi, \psi) = \min(d_A(\varphi, \psi), 6.4) - \min(d_B(\varphi, \psi), 3.8)$

Computational setting: no solvent, force field: CHARMM27. AMS with $n = 500$ to 1000 replicas and $k = 1$. 
Free energy landscape and zones A (yellow) and B (black).
Probability estimations using different initial conditions: $D=$DNS, $1=\xi_1$, $2=\xi_2$. 

Alanine di-peptide (3/6)
Flux of reactive trajectories, starting from two different initial conditions.
Transition time obtained for two values of $z_{min}$: D=DNS, 1=$\xi_1$, 2=$\xi_2$. Reference value obtained by DNS over a 97 DNS simulations of 2$\mu$s.
Estimate of the committor function using AMS.
Benzamidine-trypsin (1/2)

We recently used AMS to estimate the off rate of benzamidine from trypsin [I. Teo, C. Mayne, K. Schulten and TL, 2016].
Benzamidine-trypsin (2/2)

We obtain a dissociation rate $k_{\text{off}} = (260 \pm 240)s^{-1}$ within the same order of magnitude as the experimentally measured rate $(600 \pm 300)s^{-1}$.

The overall simulation time taken, summed over all 1000 replicas, was $2.1\mu s$ ($2.3\mu s$ after including direct MD and steered MD simulations), which is four orders of magnitude shorter than the estimated dissociation time of one event.

The main practical difficulty seems to be the determination of a 'good' domain $A$.

Computational setting: 68 789 atoms, with 21 800 water molecules, 62 sodium ions, and 68 chloride ions. Water: TIP3P model. CHARMM36 force field, with parameters for benzamidine obtained from the CGenFF force field. NPT conditions, at 298 K and 1 atm Langevin thermostat and barostat settings, using 2 fs time steps. AMS with $n = 1000$ replicas and $k = 1$. 
Another example: Radiation protection (1/2)

Monte Carlo particle transport

Concrete tunnel with a neutron source

How to compute the neutron flux at the detector?

**Challenge**: the flux is very small
Another example: Radiation protection (2/2)

Example 2: In collaboration with CEA (Eric Dumonteil, Cheikh Diop and Henri Louvin), AMS is now implemented in the Tripoli code.
Concluding remarks on AMS (1/2)

Practical recommendations:

- A careful implementation of the splitting step leads to unbiased estimators for non-normalized quantities.
- Perform many independent realizations of AMS.
- Use $\xi$ as a numerical parameter.

The algorithm is very versatile:

- Non-intrusivity: the MD integrator is a black box.
- Can be adapted to generate trajectories of any stopped process.
- Can be applied both to entropic and energetic barriers, to non-equilibrium systems, non-homogeneous Markov process, random fields, ...
- Algorithmic variants: other resampling procedure, additional selection, ...
Works in progress:

- Implementation in the NAMD software (collaboration with SANOFI, C. Mayne and I. Teo), and in TRIPOLI (collaboration with CEA).
- Adaptive computation of better and better $\xi$.
- Analysis of the efficiency as a function of $\xi$. For optimal choice of $\xi$, the cost of AMS is (for $n$ large)

$$\left((\log p)^2 - \log p\right)$$

much better than the cost of naive Monte Carlo: $\frac{1-p}{p}$. How does this degrade when $\xi$ departs from the optimal case?
Accelerated dynamics

The bottom line of the accelerated dynamics proposed by A. Voter in the late 90’s is to get efficiently the state-to-state dynamics. Three algorithms: Parallel replica, Hyperdynamics, Temperature Accelerated Dynamics.

Let us consider the overdamped Langevin dynamics:

\[ dX_t = -\nabla V(X_t) \, dt + \sqrt{2\beta^{-1}} \, dW_t \]

and a state \( W \subset \mathbb{R}^d \).

Objective: generate very efficiently the exit event \((T_W, X_{T_W})\) from a state \( W \), where

\[ T_W = \inf \{ t \geq 0, X_t \notin W \}. \]
The Quasi-Stationary Distribution (1/3)

How to take advantage of metastability to build efficient sampling techniques?

**Lemma:** Let $X_t$ start in the well $W$. Then there exists a probability distribution $\nu$ with support $W$ such that

$$\lim_{t \to \infty} \mathbb{L}(X_t | T_W > t) = \nu.$$

**Remark:** Rigorous definition of a metastable state:

exit time $\gg$ local equilibration time
The Quasi-Stationary Distribution (2/3)

**Property 1:** \( \forall t > 0, \forall A \subset W, \)
\[
\nu(A) = \frac{\int_W \mathbb{P}(X_t^x \in A, t < T^x_W) \nu(dx)}{\int_W \mathbb{P}(t < T^x_W) \nu(dx)}.
\]

If \( X_0 \sim \nu \) and if \( (X_s)_{0 \leq s \leq t} \) has not left the well, then \( X_t \sim \nu \).

**Property 2:** Let \( L = -\nabla V \cdot \nabla + \beta^{-1} \Delta \) be the infinitesimal generator of \( (X_t) \). Then the density \( u_1 \) of \( \nu \) \( (d\nu = u_1(x)dx) \) is the first eigenfunction of \( L^* = \text{div} (\nabla V + \beta^{-1} \nabla) \) with absorbing boundary conditions:
\[
\begin{cases}
L^* u_1 = -\lambda_1 u_1 \text{ on } W, \\
u_1 = 0 \text{ on } \partial W.
\end{cases}
\]
The Quasi-Stationary Distribution (3/3)

Property 3: If $X_0 \sim \nu$ then,

- the first exit time $T_W$ from $W$ is exponentially distributed with parameter $\lambda_1$;
- $T_W$ is independent of the first hitting point $X_{T_W}$ on $\partial W$;
- the exit point distribution is proportional to $-\partial_n u_1$: for all smooth test functions $\varphi : \partial W \to \mathbb{R}$,

$$
\mathbb{E}^\nu(\varphi(X_{T_W})) = -\frac{\int_{\partial W} \varphi \partial_n u_1 d\sigma}{\beta \lambda \int_W u_1(x) dx}.
$$

Remark: This is reminiscent of what is assumed in Transition State Theory (first order kinetics) and kinetic Monte Carlo models.
Escaping from a metastable state

How to use these properties to build efficient algorithms?

Assume that the stochastic process remained trapped for a very long time in a metastable state $W$. How to accelerate the escape event from $W$, in a statistically consistent way?

Remark: In practice, one needs to:

- Choose the partition of the domain into (metastable) states;
- Associate to each state an equilibration time (a.k.a. decorrelation time).

These are not easy tasks... we will come back to that.

Remark: All the algorithms below equally apply to the Langevin dynamics but the extensions of the mathematical results to the Langevin dynamics are not straightforward...
The Parallel Replica Algorithm

Idea: perform many independent exit events in parallel.

Two steps:

- Distribute $N$ independent initial conditions in $W$ according to the QSD $\nu$;
- Consider the first exit event, and multiply it by the number of replicas.
Why is it consistent?

- Exit time is independent of exit point so that

\[
X_{T_W I_0} \overset{\mathcal{L}}{=} X_{T_W 1}^{I_0},
\]

where \( I_0 = \arg \min_i (T_W^i) \);

- Exit times are i.i.d. exponentially distributed so that, for all \( N \),

\[
N \min(T_W^1, \ldots, T_W^N) \overset{\mathcal{L}}{=} T_W^1.
\]
The Parallel Replica Algorithm

The full algorithm is in three steps:

- Decorrelation step
- Dephasing step
- Parallel step
The Parallel Replica Algorithm

Decorrelation step: run the dynamics on a reference walker...
The Parallel Replica Algorithm

Decorrelation step: ... until it remains trapped for a time $\tau_{corr}$. 
The Parallel Replica Algorithm

Dephasing step: generate new initial conditions in the state.
The Parallel Replica Algorithm

Dephasing step: generate new initial conditions in the state.
The Parallel Replica Algorithm

Dephasing step: generate new initial conditions in the state.
The Parallel Replica Algorithm

Dephasing step: generate new initial conditions in the state.
The Parallel Replica Algorithm

Dephasing step: generate new initial conditions in the state.
The Parallel Replica Algorithm

Dephasing step: generate new initial conditions in the state.
The Parallel Replica Algorithm

Dephasing step: generate new initial conditions in the state.
The Parallel Replica Algorithm

Parallel step: run independent trajectories in parallel...
The Parallel Replica Algorithm

Parallel step: ... and detect the first transition event.
The Parallel Replica Algorithm

Parallel step: update the time clock: $T_{simu} = T_{simu} + NT$. 
The Parallel Replica Algorithm

A new decorrelation step starts...
The Parallel Replica Algorithm

New decorrelation step
The Parallel Replica Algorithm

The three steps of ParRep:

- **Decorrelation step**: does the reference walker remain trapped in a set?
- **Dephasing step**: prepare many initial conditions in this trapping set.
- **Parallel step**: detect the first escaping event.
The generalized Parallel Replica Algorithm

This algorithm is very versatile: it works for entropic barriers, and for any partition of the state space into states. But it requires some a priori knowledge on the system: the equilibration time $\tau_{corr}$ attached to each state.

Two questions: How to choose $\tau_{corr}$? How to sample the QSD?

We propose a generalized Parallel Replica algorithm [Binder, TL, Simpson, 2014] to solve these issues. It is based on two ingredients:

- the Fleming-Viot particle process
- the Gelman-Rubin statistical test
The Fleming-Viot particle process

Start $N$ processes i.i.d. from $\mu_0$, and iterate the following steps:

1. Integrate (in parallel) $N$ realizations ($k = 1, \ldots, N$)

$$dX_t^k = -\nabla V(X_t^k) \, dt + \sqrt{2\beta^{-1}} \, dW_t^k$$

until one of them, say $X_t^1$, exits;

2. Kill the process that exits;

3. With uniform probability $1/(N-1)$, randomly choose one of the survivors, $X_t^2, \ldots, X_t^N$, say $X_t^2$;

4. Branch $X_t^2$, with one copy persisting as $X_t^2$, and the other becoming the new $X_t^1$.

It is known that the empirical distribution

$$\mu_{t,N} \equiv \frac{1}{N} \sum_{k=1}^{N} \delta_{X_t^k}$$

satisfies:

$$\lim_{N \to \infty} \mu_{t,N} = \mathcal{L}(X_t|t < T_W).$$
The generalized Parallel Replica algorithm

1. Run a reference walker, using standard MD.
2. Each time the reference walker enters a state, start a Fleming-Viot particle process (with $N$ replicas simulated in parallel) with initial condition the entering point.
3. If the reference walker exits before the Fleming-Viot particle process reaches stationarity go back to 1. Else go to the parallel step.
4. Parallel step: Starting from the end points of the Fleming-Viot particle process (approximately i.i.d. with law the QSD), run independent MD and consider the first exit event. Multiply the first exit time by $N$ and go back to 1, using the first exit point as initial condition.

The time at which the Fleming-Viot particle process becomes stationary is determined using the Gelman-Rubin statistical test.
Numerical results

We recently tested the parallel replica algorithm applied to biological systems (postdoc Florent Hédin):

- Conformational equilibrium of the alanine dipeptide
- Dissociation of the FKBP-DMSO protein-ligand system

Main differences with materials science: definition of the states using collective variables, the states do not define a partition, much more rugged landscapes.

Current implementation within OpenMM, see https://gitlab.inria.fr/parallel-replica
Definition of ParRep domains based on a free energy surface: we study the transition time from $C_{7eq}$ (outside the red rectangle) to $C_{7ax}$ (inside the red rectangle).
Alanine dipeptide (2/5)

Cumulative distribution function of the transition time.
Alanine dipeptide (3/5)

Convergence of the mean transition time.
Alanine dipeptide (4/5)

Distribution of the correlation times computed by FV.
### Alanine dipeptide (5/5)

<table>
<thead>
<tr>
<th>tol</th>
<th>WT(s)</th>
<th>tsim(ns)</th>
<th>Speed(ns/day)</th>
<th>Eff. speedup</th>
<th>(Eff./Exp.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>6015</td>
<td>10008</td>
<td>143752</td>
<td>156</td>
<td>70%</td>
</tr>
<tr>
<td>0.025</td>
<td>5239</td>
<td>10103</td>
<td>166609</td>
<td>181</td>
<td>80%</td>
</tr>
<tr>
<td>0.05</td>
<td>4973</td>
<td>10032</td>
<td>174296</td>
<td>189</td>
<td>84%</td>
</tr>
</tbody>
</table>

Effective speed-up as a function of the tolerance, for 224 cores run in parallel.
FKBP-DMSO (1/3)

FKBP-DMSO complex, corresponding to the RCSB-PDB entry “1D7H”
DMSO in its binding cavity; distances used to define the cavity.
FKBP-DMSO (3/3)

Cumulative distribution function of the dissociation times.
Concluding remarks on GenParRep

• The algorithm equally applies if the states are disjoint but do not constitute a partition of the state space.
• The algorithm is very versatile: energetic and entropic barriers, non equilibrium systems, Continuous Time Markov Chain (Aristoff, Plechac, Wang), ...
• There are two other accelerated dynamics: Hyperdynamics and Temperature Accelerated Dynamics (TAD) which can also be analyzed using the notion of QSD.
• The QSD is a good intermediate between continuous state dynamics and kMC models (Markov state models). Transition rates could be defined starting from the QSD.
• It can be used to analyze the validity of the transition state theory and kMC models, in the small temperature regime.
The Parallel Trajectory Splicing algorithm

Idea [Perez, Cubuk, Waterland, Kaxiras, Voter, 2015] :

- Simulate in parallel short trajectories which start from the QSD in a state, and end at the QSD in a state.
- Glue together these short trajectories to build the full dynamics.

We are currently working in collaboration with M. Haefele and J. Derouillat on related ideas (precompute exit events), with applications to biological systems (post doc Florent Hédin).
References

Some papers I mentioned:


