Enabling Simulation of Complex Systems: Hybrid Monte Carlo Methods

Elena Akhmatskaya
Ikerbasque / BCAM / FLE
Molecular Simulation: Basic Concepts
Hierarchy of scales

- **Applications**
  - Meso-scale modeling (segments)
  - Molecular mechanics (atoms)
  - Finite element analysis
  - Coarse-Grained
  - Meso-scale modeling (segments)

- **Physics**
  - Quantum mechanics (electrons)

- **Chemistry**
  - Molecular mechanics (atoms)

- **Engineering**
  - Coarse-Grained
  - Meso-scale modeling (segments)

- **Materials**
  - Molecular mechanics (atoms)

- **Time**
  - femtoseconds (fs)
  - picoseconds (ps)
  - nanoseconds (ns)
  - microseconds (μs)
  - milliseconds (ms)
  - seconds (s)
  - minutes (min)
  - hours

- **Length**
  - nanometers (nm)
  - micrometers (μm)
  - millimeters (mm)
  - meters

**We Are Here**

- Mechanical properties
- Electrical properties
- Transport properties
- Morphology
- Phase separation
- Compositional structure
- Component properties
- Interactions
- Mesoscale parameters
- Molecular structure
- Reactions
- Spectroscopic properties
- Force field parameters

**Applications**

- **Chemistry**
  - Molecular properties
  - Reactions
  - Spectroscopic properties
- **Physics**
  - Quantum mechanics
  - Molecular mechanics
- **Engineering**
  - Coarse-Grained
  - Meso-scale modeling
- **Materials**
  - Component properties
  - Interactions
  - Mesoscale parameters

**Physics**

- Quantum mechanics (electrons)
- Molecular mechanics (atoms)
- Coarse-Grained
- Meso-scale modeling (segments)

**Chemistry**

- Molecular mechanics (atoms)

**Engineering**

- Coarse-Grained
- Meso-scale modeling (segments)

**Materials**

- Component properties
  - Interactions
  - Mesoscale parameters

**Applications**

- Meso-scale modeling (segments)
- Molecular mechanics (atoms)
- Finite element analysis
- Coarse-Grained
- Meso-scale modeling (segments)

**Time**

- femtoseconds (fs)
- picoseconds (ps)
- nanoseconds (ns)
- microseconds (μs)
- milliseconds (ms)
- seconds (s)
- minutes (min)
- hours

**Length**

- nanometers (nm)
- micrometers (μm)
- millimeters (mm)
- meters
Molecular simulation

Why:
Supercomputing era: Molecular simulation is becoming the technique of choice to describe systems of ever-increasing complexity, to discover new phenomena

Where: Applications (examples)

Life Sciences

Nanoscience

Materials Science

How?
Modeling molecules

• In classical molecular simulation a molecule is described as a series of charged points (atoms) linked by springs (bonds)

• For each atom in every molecule, we need:
  ✓ Position (r)
  ✓ Momentum (mv)
  ✓ Charge (q)
  ✓ Bond information (which atoms, bond angles, etc.)
Potential energy function in molecular simulation

• Helps to *predict behavior* and structure of systems

• *Describes the interaction* energies of all atoms and molecules in the system

• Always an *approximation*
  ✓ Closer to real physics --> more computation time (more interactions increase accuracy)
Potential energy equation

\[ U = U_{\text{non-bonded}} + U_{\text{bonded}} \]

\[ U_{\text{non-bonded}} = U_{\text{van der Waals}} + U_{\text{electrostatic}} \]

\[ U_{\text{bonded}} = U_{\text{bond-stretch}} + U_{\text{angle-bend}} + U_{\text{rotate-along-bond}} \]
Potential energy equation (example)

\[
U (r^N, \omega^N) = \sum_{\text{bonds}} \frac{k_l}{2} (l - l_o)^2 + \sum_{\text{angles}} \frac{k_\theta}{2} (\theta - \theta_o)^2 + \sum_{\text{dihedrals}} \frac{k_\phi}{2} (1 + \cos (n\phi - \delta)) \\
+ \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[ 4 \varepsilon_{ij} \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \frac{q_i q_j}{4 \pi \varepsilon_o r_{ij}} \text{ (Coulombic)}
\]

\(r^N=r_1, r_2 \ldots r_N\) - centers of mass of the molecules
\(\omega^N=\omega_1, \omega_2 \ldots \omega_N\) represent the orientational and other molecular coordinates
\(l, \theta, \phi\) are bond length, bond angles and dihedral angles, respectively,
\(q_i\) are point charges.
From microscopic to macroscopic

- Molecular simulations generate information at the *microscopic* level.
- The conversion of *microscopic* information to *macroscopic* observables requires *statistical mechanics*.
- Average values are defined as *ensemble averages*.
- An *ensemble* is a collection of all possible systems which have different microscopic states but have an identical macroscopic state.
- The *Ergodic hypothesis* states

\[
\text{Ensemble average} = \text{Time average}
\]
Ensembles

- Defined by a choice of variables to be fixed during simulation
- Come from
  - Physical situations
  - Necessity: Some ensembles aren’t well suited to certain phenomena
- Commonly used in biomolecular simulation: NVT, NPT, NVE
Traditional Molecular Simulation Methods
Simulation methods

- **Two basic simulation approaches:**
  - Deterministic (regular): **Molecular Dynamics (MD)**
    - Smooth trajectory - time has a clear interpretation
    - Provides thermodynamic and kinetic properties
  - Stochastic (random): **Monte Carlo (MC)**
    - Discontinuous trajectory – time has no clear meaning
    - Does not offer kinetic information

<table>
<thead>
<tr>
<th>Kinetic properties</th>
<th>Thermodynamic properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate constants</td>
<td>Free energies</td>
</tr>
<tr>
<td>Viscosity, diffusion</td>
<td>Pressure</td>
</tr>
<tr>
<td>Relaxation</td>
<td>Concentration</td>
</tr>
<tr>
<td>Time-correlations</td>
<td>Heat capacities</td>
</tr>
<tr>
<td>Collision frequencies</td>
<td>Fluctuations</td>
</tr>
</tbody>
</table>

A bit of history

**Monte Carlo** (formerly, “statistical sampling method”):

- **1901 Lord Kelvin**: for initializing trajectories of molecules undergoing elastic collisions with the walls of various vessels (manually by his assistant William Anderson).
- **1934 Enrico Fermi**: to study neutron moderation (with the aid of mechanical calculator).
- **1940s Nicholas Metropolis** and coworkers developed the method more formally while working at the Los Alamos National Laboratory with Fermi, Stanislaw Ulam and John von Neumann.
- **1947 Nicholas Metropolis** proposed a name “Monte Carlo”, following a conversation with Stan Ulam, in part because Ulam had an uncle who frequented the Monte Carlo gaming halls.
- **1953 Nicholas Metropolis** and co-workers: first clear published account.
- **Since early 1950s** (after the building of the first electronic computer): extensively used.

**Molecular Dynamics**: developed by Berni Alder and Tom Wainwright in 1956 at the Lawrence Radiation Laboratory at Livermore (now Lawrence Livermore National Laboratory) for study a solid-fluid phase transition for hard spheres.
Molecular Dynamics

A computer simulation technique where the time evolution of a set of interacting atoms is studied simply by integrating their equations of motion.

To sample:

- **Initialize system**
  - get a reasonable initial conformation
  - randomly assign velocities (sampled from a Maxwell-Boltzmann distribution)

- **Describe the forces on all atoms** using a chosen forcefield: $F = -\nabla U$

- **Integrate Newton’s equations** $M \frac{d^2 r}{dt^2} = F$, to predict positions and velocities at a new time. Recalculate forces (a few million times...). $M$ is a diagonal mass matrix of atomic masses, $\mathbf{r}$ is the collective atomic position vector

- **Result**: positions, velocities of all atoms during a few nanoseconds

- **Take averages** to calculate thermodynamics and kinetic properties

**The most expensive part of a MD simulation is the calculation of interatomic forces**

- Long-range electrostatic interactions $O(N^2)$
Integration algorithms

- What is the best integration algorithm?
  Some checks:
  ✓ Stability at large time steps (LJ potential has powers of 12)
  ✓ computational efficiency (force calculation is expensive)
  ✓ conserved quantity is stable over many integration steps (long-time drift)
- Integrators
  Verlet: velocities are estimated using a finite difference which reduces accuracy to a second order
  Predictor-corrector: better local accuracy than Verlet but worse long-time drift
  Leapfrog: variation of Verlet which improves velocity estimations

\[
\begin{align*}
    v_i(t + \frac{\Delta t}{2}) &= v_i(t - \frac{\Delta t}{2}) + \frac{F_i(t)}{m_i} \Delta t \\
    r_i(t + \Delta t) &= r_i(t) + v_i(t + \frac{\Delta t}{2}) \Delta t
\end{align*}
\]
Monte Carlo

Neglects velocities and looks for minima on the potential surface by randomly probing configuration space

Usually:  *Metropolis Monte Carlo*

To sample:

- Generate a random move
- Evaluate the energy $U$
- Accept/reject with Boltzmann probability: $e^{-\frac{\Delta U}{k_BT}}$
- Accumulate integral
- Take averages in NVT ensemble

The temperature is prescribed and *fixed* via the Boltzmann factor $\beta = \frac{1}{k_BT}$
Properties of MC simulation

• System behaves as a Markov process
  ✓ Current state depends only on previous

• System is assumed to be ergodic
  ✓ Any state can be reached from any other state

• Accepted more by physicists than by chemists
  ✓ Not deterministic and does not offer time evolution of the system
MC vs. MD

Monte Carlo
- Only energy is needed
- Straightforward to perform NVT and NPT
- Easy to constrain some degrees of freedoms (not include them in trials)
- For some systems, large motions can be made (LJ particles) between consecutive configurations

- Hard to choose a trial move for complex systems, such as proteins, since proteins move collectively
- Step size decreases with system size
- Can’t easily get kinetic information

Molecular Dynamics
- Both energy and force are needed
- Requires temperature and pressure control for NVT and NPT
- Needs special techniques to constrain some degrees of freedoms
- The consecutive configurations are very similar

- MD can be applied to both simple and complex systems
- Same time step can be used for small or large systems
- Can generate kinetic data as well as thermodynamic data
Hybrid Monte Carlo Methods
Combining MD with MC: HMC

(HMC) Hybrid Monte Carlo (Duane et al., 1987) : “bad MD but good Monte Carlo”

- HMC combines short MD trajectories run in NVE ensemble with an MC rejection step. Each new trajectory is accepted with Metropolis probability

\[
\min(1, \exp(-\beta \Delta H)), \text{ where } \Delta H := H_{\text{new}} - H_{\text{old}} \text{ and } H = \frac{1}{2} \sum_{i=1}^{N} \frac{\|p_i\|^2}{m_i} + U(r_1, r_2, \ldots, r_N)
\]

after which velocities are randomly reset. Note that \(H\) is the total energy (Hamiltonian) function and \(\Delta H\) is not zero due to a numerical error.

- Can be viewed either as an efficient MC with a smart collective move or an MD with corrupted dynamics which rigorously samples from the target temperature.

- Exponential performance degradation with system size and time step.

- Samples in NVT ensemble but can be extended to NPT.
Combining MD with MC: GHMC

(GHMC) Generalized hybrid Monte Carlo (Kennedy, Pendleton, 2001; Horowitz, 1991)

- Instead of resetting momentum update on each step, GHMC only partially modifies it in the rigorous manner:

\[
\begin{pmatrix}
  u_i \\
  p_i
\end{pmatrix}
=\begin{pmatrix}
  \cos(\phi) & -\sin(\phi) \\
  \sin(\phi) & \cos(\phi)
\end{pmatrix}
\begin{pmatrix}
  u_i^n \\
  p_i^n
\end{pmatrix}
\]

where the noise vector \( u_i \in \mathbb{R}^3 \) is normally i.i.d. distributed and \( 0 < \phi \leq \pi/2 \)

- No acceptance (Metropolis) test for momentum update is required

\( \sqrt{(u_i^n, p_i^n)} \) and \( (u_i, p_i) \) are distributed according to the same normal distribution (Gaussian random variables under an orthogonal transformation)

- Momentum reversal is needed after rejected MD step to keep a detailed balance condition

- GSHMC consists of two steps: (1) numerical propagation of the system under the conservative dynamics (MD) followed by Metropolis correction for numerically induced errors in the conservation of total energy and (2) a partial momentum update

- Samples in NVT ensemble
GHMC: Pros & Cons

**Pros**

- Keeps more dynamical information than HMC
- Special cases:
  - microcanonical MD ($\phi = 0$)
  - HMC ($\phi = \pi/2$)
  - Langevin Monte Carlo (a single MD step)
  - Langevin dynamics $\phi = \sqrt{2\gamma h} \ll 1$
  
  \[ h = \Delta t, \quad \gamma > 0 \]

**Cons**

- Exponential performance degradation with system size and time step
- Momentum reversal (flip) after rejected MD step
- A low acceptance rate leads to an undesirable Zitterbewegung due to an accumulation of momentum flips
Combining MD with MC: GSHMC

(GSHMC) Generalized shadow hybrid Monte Carlo (Akhmatskaya, Reich, 2008)
GHMC which samples with respect to modified Hamiltonians

Objective: to improve the acceptance rate in GHMC for large system sizes by reducing a discretization error in numerical solution:
- more efficient sampling
- less momentum reversals => dynamical information is retained

• Acceptance probability for GHMC: \( \min(1, \exp(-\beta \Delta H)) \)
• \( \langle \Delta H \rangle = O(N \Delta t^p) \) \( p \) is an order of a time-stepping method \( (p = 2 \text{ for Verlet}) \)
• Verlet is symplectic \( \Rightarrow \) modified Hamiltonians \( H_{\Delta t} \) (an asymptotic expansion in powers of the discretization parameter) exists (J. Sanz-Serna, M. Calvo, 1994)
• \( H_{\Delta t} \) is conserved by symplectic integrators to even higher accuracy in step size than true Hamiltonians \( \langle \Delta H_{\Delta t} \rangle = O(N \Delta t^{2m}), m \geq 4 \), \( m \) is an order of \( H_{\Delta t} \) (Benettin, Giorgilli, 1994)
• Relatively inexpensive and arbitrarily accurate approximation to true Hamiltonian (Skeel, Hardy, 2001)
Ideas behind GSHMC

- Replaces Hamiltonians in the Metropolis criterion with modified Hamiltonians:
  \[ \min (1, \exp(-\beta \Delta H_M)) \]

- Uses the momentum update introduced in GHMC combined with modified Metropolis acceptance criterion to recover the Boltzmann distribution:

\[
\min \left\{ \frac{\exp(-\beta[H_{\Delta t}(q, p) + \frac{1}{2}u^T M^{-1} u])}{\exp(-\beta[H_{\Delta t}(q, p^n) + \frac{1}{2}u^n T M^{-1} u^n])} \right\}
\]

- Re-weights averages due to use of a modified ensemble

Given an observable \( \Omega(q, p) \) and its values \( \Omega_i, i=1, 2, \ldots K \) along a sequence of states \( (q_i, p_i), i = 1, 2, \ldots K \) re-weight \( \Omega_i \) to compute averages:

\[
\langle \Omega \rangle = \frac{\sum_{i=1}^{I} w_i \Omega_i}{\sum_{i=1}^{I} w_i} \quad \quad w_i = \exp(-\beta(H(q_i, p_i) - H_{\Delta t}(q_i, p_i)))
\]
Advantages of GSHMC

- All advantages of HMC
- Speedup over HMC is of order \((\text{Number of Atoms})^{1/4}\)
- Retains the dynamical information
- Is very easy to be combined with other enhanced sampling ideas due to its flexibility

GSHMC outperforms HMC by a factor of order of \((\text{Number of Atoms})^{1/4}\) for small time steps
Evolution of hybrid Monte Carlo methods

**MD**

+ **HMC** (D1987) → **SHMC/S2HMC** (IH2004; S2008)

**MC**

Partial momentum update (dynamics)

**GHMC** (H1991; KP2001) → **GSHMC/GS2HMC** (AR2008)

**meso-GHMC** (AR2008)

Mesoscale (applications range)

**MTS-GHMC** (AR2009)

Multiscale (speed-up)

**MTS-GSHMC** (AR2009)

Multiscale (speed-up)

**meso-GSHMC** (AR2008)

Mesoscale (applications range)

---

D1987: *Duane et. al 1987*
IH2004: *Izaguirre & Hampton, 2004*
S2008: *Sweet wt. al, 2008*
KP2001: *Kennedy & Pendleton, 2001*
H1991: *Horowitz, 1991*
Multiscale Simulation

- Biological systems are multi-scale in nature
- Dynamics of proteins contain motions over different time scales
- An extreme case of multi-scale simulation
  - fast bond vibrations and slow folding processes in a single simulation
  - differ by >10 orders of magnitude!

- a time step used in traditional MD simulation should be at the scale of the fastest process, i.e. of femtoseconds. This leads to extremely expensive simulation of the slow processes (~10^{15} time steps)
Multiple time stepping (MTS)

Makes use of multi-scale nature of the macromolecular system:
- Fast/slow force splitting: \( \frac{d^2}{dt^2} r = F^\text{fast}(r) + F^\text{slow}(r) \),
  where \( F^\text{fast} = -\nabla U^\text{fast} \) and \( F^\text{slow} = -\nabla U^\text{slow} \) and subject to \( U = U^\text{fast} + U^\text{slow} \).
- Bonded: “fast”
- Long range non-bonded: “slow”
- Evaluates slow forces less frequently
  - Fast forces cheap and easy to compute in parallel
  - Slow force evaluation expensive (scales quadratically in the number of atoms in the system) and more difficult to parallelize efficiently
- Multiple time stepping integrators are required to solve modified ODEs equations

Fast impulses, \( \delta t \) (inner step size)

Time, \( t \)

Slow impulses, \( \Delta t \) (outer step size)
Drawbacks of the existing MTS methods

- Do not exactly sample from the target temperature
- Suffer from severe resonance instabilities
  - Mollified Impulse (MOLLY) to overcome linear instabilities (Izaguirre, et al., 1999)
MTS-GSHMC: Ideas behind the method

**MTS-GSHMC)** Multiple time stepping generalized shadow hybrid Monte Carlo (Akhmatskaya, Reich, 2009):

- Uses GSHMC to provide the desired weak stochastic stabilization for the MD multi-scale integrator to enhance computational performance
- Develops a novel technique for computing $k^{\text{th}}$ order shadow Hamiltonians for multiple-time-stepping (MTS) methods
- Implements a mollified impulse MTS method in the Molecular Dynamics Monte Carlo (MDMC) step of the GSHMC to eliminate resonance induced instabilities
## Comparison with prior art

**Test system:** one-dimensional chain of diatomic molecules interacting through LJ potential

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Achievable effective outer step-size (performance)</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>19.2</td>
</tr>
<tr>
<td>Rigorous temperature control (accuracy)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
MTS-GSHMC: Benefits

• Enables efficient detailed atomistic simulations of large macromolecular systems
• **Stability**: MTS-GSHMC is always stable while other MTS methods suffer from resonance or non-resonance induced instabilities
• **Accuracy**: MTS-GSHMC rigorously samples from the constant temperature ensemble in contrast to existing MTS methods
• **Efficiency**: MTS-GSHMC is able to reduce the frequency of calculation of expensive slow forces by a factor of ~20 compared with traditional MD methods and by a factor of ~2 over the best performing MTS approaches
• Provides efficient sampling in simulation of complex systems
• Is suitable to massively parallel computing
Meso-scale modeling

- **Meso-scales**: transitional regions between macroscopic and microscopic regimes
  - atomistic methods (microscopic) e.g. MD, too expensive
  - continuum solvers (FD, FE,...) neglect microstructure
- Several **phenomena** occurring at meso-scales:
  - *fluid mixtures*: emulsions, surfactants, phase separation
  - *colloid suspensions*: aggregation, clustering, dispersion
  - *polymeric solution*: dense solution, melting
- Dissipative particle dynamics (DPD)
  - particle based
  - mesoscopic
  - Newton’s Laws
  - three types of forces between particles: conservative, dissipative, fluctuation
  - correct hydrodynamic behaviour
  - known bottleneck: none of existing implementations can reproduce correctly thermodynamic quantities independently of time-step in MD under full DPD dynamics
Combining MD with MC: meso-GSHMC

(meso-GSHMC) Meso-generalized shadow hybrid Monte Carlo (Akhmatskaya, Reich, 2009):

• Puts DPD within the framework of Markov chain Monte Carlo methods which implies rigorous sampling from the canonical distribution regardless of the chosen time step
• Suggests a novel (local) momentum refreshment Monte Carlo step, which conserves the Boltzmann velocity distribution as well as total linear and angular momentum.
• Will be applicable to a wide range of particle-based meso-scale models
Comparison of the new meso-scale methods with prior art. Numerical tests

**Numerically observed temperature** $<k_B T>$ **vs step-size** $\Delta t$

**Model A:**
4000 particles; conservative forces are set to zero

**Model C:**
2867 particles; Lennard-Jones fluid (non-zero conservative forces)

Meso-GSHMC/GHMC exactly reproduce the target temperature for all values of step-size, while the DPD method leads to a nearly linear increase in the numerically observed temperature with respect to the step-size.
HPC Implementation of Molecular Simulation
Scenario I: One, long simulation

- **Divide the calculation between processors**
  - requires smart parallelization strategies
  - requires fast communication

- **Efficient implementation**

  **Highly parallel software:**

  **Building special-purpose machines:**

  Expected: ~80x faster than Desmond on commodity hardware and ~100x faster than Blue Matter on Blue Gene

- **Drawbacks:**
  - scalability is limited (currently ~tens of thousands processors)
  - computationally expensive
Scenario II: Many short simulations

- Run many short simulations with identical initial structures but different velocities
  - Requires new algorithms!
  - Communication is cheaper than in Plan A
- Implementation
  - Folding@Home in Stanford University: Worldwide grid computing to understand protein folding mechanism
  - Blue Gene Project at IBM: Protein science on a supercomputer

- Drawbacks:
  - Still very computationally demanding
  - Needs ~10K-100K processors to obtain statistically meaningful result
  - Lack of dynamical information
  - Excessive data storage requirements
Our way

HMC methods can be viewed as an improved scenario B approach

- Rigorous
- Less computationally demanding due to faster convergence to the target
- Reduced data storage as a result of filtering data on the fly
  but
- Communication is more expensive than in plan B
  - smart implementation is required!

- High degree of parallelism
- Minimal communication and synchronization
- Well suited to peta-scale computing
HMC Methods: Applications
Simulations of a spider venom toxin in a membrane environment

Collaborative project with the group of Prof. Mark Sansom at Oxford University: *Improved sampling for simulations of interfacial membrane proteins*

- Toxin interacts with ion channel by binding to the voltage-sensing domain of the channel and changes the energetics of voltage-dependent gating
- Defects in ion channels have been implicated in a wide range of diseases including type 2 diabetes, cardiac arrhythmia, schizophrenia
Application of GSHMC to peptide toxin/bilayer system

Simulation model:

- The toxin is initially placed at the wrong location
- The rate of drift of the toxin to the preferred interfacial location is measured during the simulation
- Simulation techniques: CG-MD, GSHMC (FLE)

Results:

- FLE’s approach (GSHMC) is a factor of 10 more efficient than traditional MD
Conclusions

We introduced a class of hybrid methods developed in Fujitsu Laboratory of Europe in collaboration with Prof Sebastian Reich (Potsdam University) which were designed to make feasible efficient and detailed simulation of large and complex systems:

- Based on Molecular Dynamics and Monte Carlo
- Accurate
  - exact temperature control
  - extract dynamical information in a rigorous manner
- Efficient
  - capable sampling for large systems
  - longer time steps
  - outperform the traditional methods
- Remove drawbacks of existing methods
  - overcome exponential performance degradation with system size/time step
  - Meso: data are independent on the integration step
  - MTS: do not suffer from resonance/non-resonance induced instabilities
- Flexible
  - Easy to combine with other enhanced sampling methods
- Suitable to massively parallel computers
  - Allow for a range of parallel strategies