

Campinas Advanced School of Thermodynamics

Workshop on Molecular Simulations

Session 3 –MC with Cassandra

Edward Maginn and Eliseo Marin

Department of Chemical and Biomolecular Engineering

University of Notre Dame, USA

ed@nd.edu

All materials available on
MaginnGroup GitHub site

29 October, 2022



Monte Carlo

Monte Carlo methods encompass all methods that employ statistical simulation of some underlying system, whether or not the system represents a real physical process.

- nuclear reactor design
- radiation cancer therapy
- traffic flow
- stellar evolution
- econometrics / stock market forecasting
- oil well exploration
- VLSI design
- phase equilibria, material property calculations

Characteristics of Monte Carlo

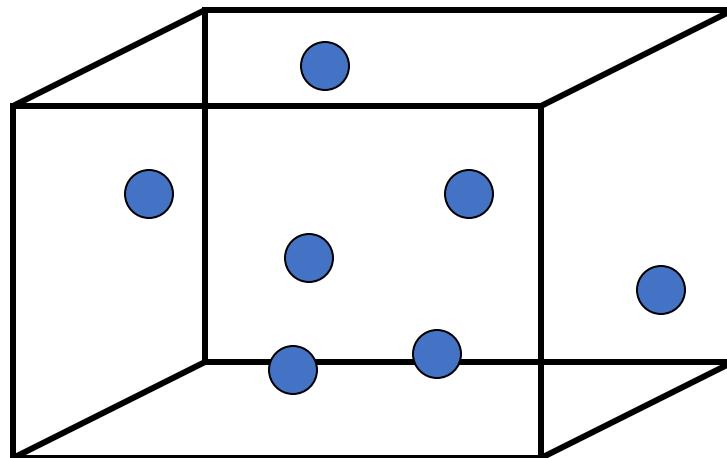
- Statistical simulation methods may be contrasted with conventional numerical discretization approaches:
- Conventional
 - describe system mathematically with differential eqns
 - discretize and solve equations numerically
- Monte Carlo
 - often simulate physical process directly; in this case, no need to write down differential equations.
 - requirement: physical or mathematical system must be characterized by **probability density functions**.
 - given a pdf, **randomly** sample from the pdf.

Monte Carlo Integration

- Example

$$Z = \int dr^N \exp[-\beta V(r^N)]$$

Randomly generate atomic configurations and evaluate the configurational integral



N molecules, T fixed, $V=L_x L_y L_z$

$$Z \approx \frac{V^N}{N_{trials}} \sum_{i=1}^{N_{trials}} \exp[-\beta V(r_1^i, \dots, r_N^i)]$$

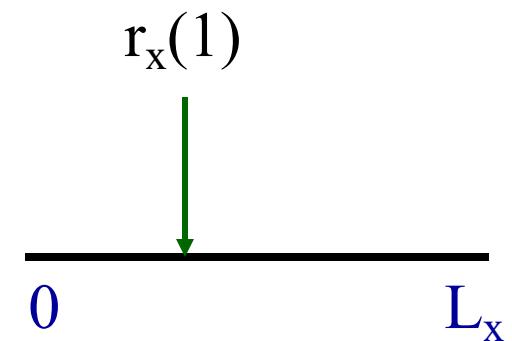
Monte Carlo Integration

- To evaluate the numerical integral, select random numbers from the $3N$ -dimensional *configuration space*. $3N$ function evaluations

$$r_x(1) = (\text{rand}) * L_x$$

rand is uniform random number on (0,1)

3N of these required



Limits of MC Integration

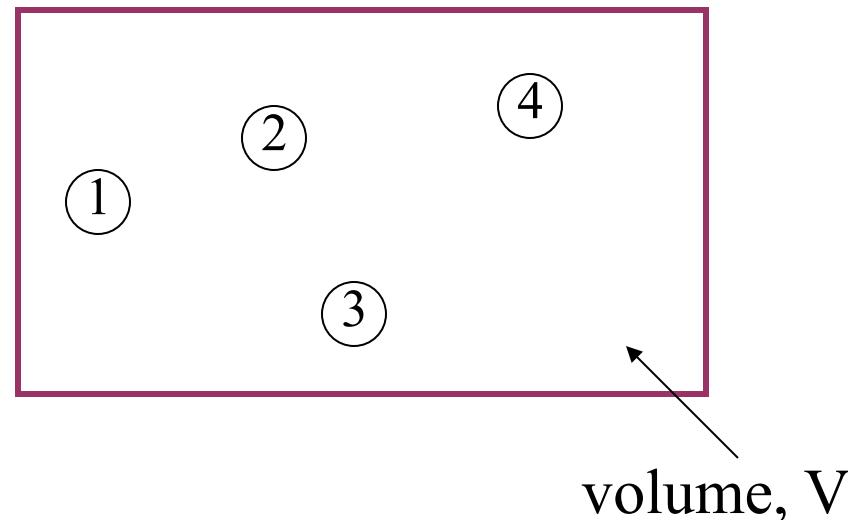
- Consider a system such as the following

- Potential is

$$V(r_1) + V(r_2) + V(r_3) + V(r_4) = V_{tot}$$

- Any possible problems?

As more atoms are added, the probability that two will overlap increases dramatically.



Limits of MC Integration

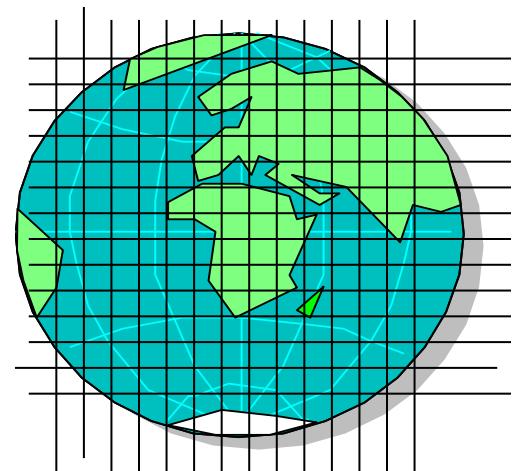
- If one pair overlaps, $V_{tot} \rightarrow \infty$
for all terms $\longrightarrow \sum \exp[-\beta V] \rightarrow 0, Z \approx 0$
- Convergence becomes difficult as N/V increases.
- Same problem if try to compute averages

$$\langle A \rangle = \frac{\int dr^N A \exp[-\beta V(r^N)]}{\int dr^N \exp[-\beta V(r^N)]}$$

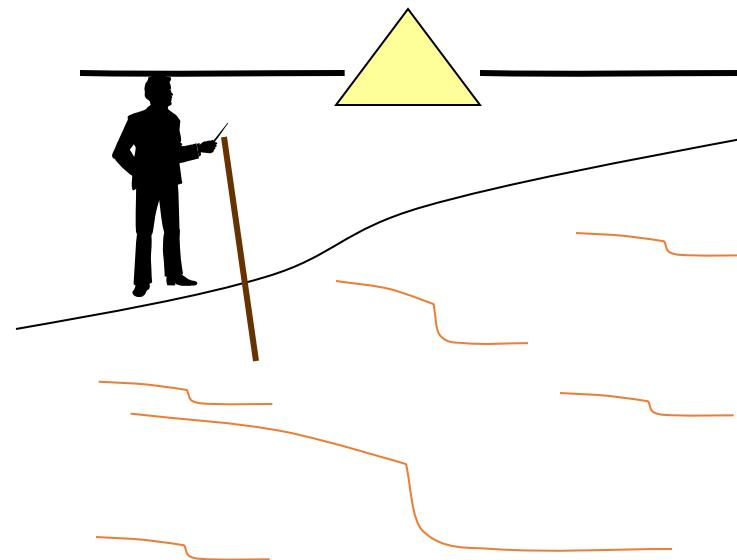
Requires non-random
sampling schemes!

Importance Sampling

- Example (after Frenkel and Smit): We would like to measure the depth of the Nile.



conventional quadrature



importance sampling

Markov Chains

- A stochastic process of “moves” or “trials” that satisfies two conditions:
- 1) outcome of each trial belongs to a finite set, called **state space**

$$\{\Gamma_1, \Gamma_2, \dots, \Gamma_m, \Gamma_n, \dots\}$$

- 2) outcome of each trial depends only on the outcome of the **immediately preceding trial**. No “memory”.

Markov Chains

- Definitions

π_{mn} = probability that a trial produces state n, given that previous state was m.

- π_{mn} depends on states m and n, but because there is no “memory”, it is independent of where it appears in the Markov chain.
- \prod = transition matrix;
collection of all π_{mn}

Markov Chains

- Note that Π contains all information on the Markov chain!
- Example (after Allen and Tildesley)
- Consider the two conditions
 - if a computer is up one day, there is a 60% chance it will be up tomorrow.

$$\pi_{uu} = 0.6$$

$$\pi_{ud} = 0.4$$

Example, cont.

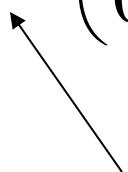
- If a computer is down one day, there is a 70% chance it will be down tomorrow

$$\pi_{dd} = 0.7$$

$$\pi_{du} = 0.3$$

- Transition matrix is

$$\Pi = \begin{pmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{pmatrix}$$



a stochastic matrix

$$\sum_m \pi_{mn} = 1$$

rows sum to 1; the system must be in some state at end of trial.

Limiting Distributions

- Limiting distribution

$$\rho = \lim_{t \rightarrow \infty} \rho^{(1)} \Pi^t$$

- Must satisfy eigenvalue equation

$$\rho \Pi = \rho$$

$\sum_m \rho_m \pi_{mn} = \rho_n$

eigenvector of transition matrix,
corresponding to eigenvalue of 1.

- ρ is completely determined by Π : independent of initial condition $\rho^{(1)}$!!!

Limiting Distributions

- Markov chains in which we arrive at the stationary solution regardless of initial conditions are **ergodic** and **irreducible**.
 - ergodic chains always have limiting distribution.
- In statistical mechanics problems, we do not know Π , but we do know ρ !

the pdf of our
ensemble!

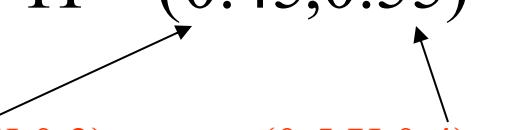
Back to Example

- Day 1: 50% chance computer is up.

$$\rho^{(1)} = (\rho_u, \rho_d) = (0.5, 0.5)$$

- What is the probability computer will be up/down on day 2?

$$\rho^{(2)} = \rho^{(1)} \Pi = (0.45, 0.55)$$



$$(0.5 \times 0.6) + (0.5 \times 0.3) \qquad \qquad (0.5 \times 0.4) + (0.5 \times 0.7)$$

Example, cont.

- Day 3 probabilities

$$\rho^{(3)} = \rho^{(2)\Pi} = \rho^{(1)\Pi\Pi} = (0.435, 0.565)$$

- Can show with a little effort...

$$\rho^{(5)} = (0.4287, 0.5713)$$

$$\rho^{(6)} = (0.4286, 0.5714)$$

$$\rho^{(20)} = (0.4286, 0.5714)$$

limiting
distribution
reached.

Back to Importance Sampling

- Each state point in a system that obeys the canonical ensemble must obey

$$\rho_m = \rho_{NVT}(\Gamma_m) d\Gamma$$

point in phase space for continuous distributions

- We wish to “sample” Γ -space according to probability distribution $\{\rho_m\}$.

Importance Sampling

- That is, pick a set of states $m_1, m_2, \dots m_{N_{\text{trials}}}$ such that the probability of each state is $m_t = \rho_{m_t}$
- If such a Markov chain can be constructed, then

$$\langle f \rangle_\rho = \frac{1}{N_{\text{trials}}} \sum_{i=1}^{N_{\text{trials}}} f(m_i)$$

ensemble average is an arithmetic average, because all states generated according to pdf.

- How? Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, *J. Chem. Phys.*, **21**, 1087, (1953).

Microscopic Reversibility

- One way to ensure the proper form of Π is to replace $\rho\Pi = \rho$ with a stronger condition called **microscopic reversibility**.

$$\rho_m \alpha_{mn} \pi_{mn} = \rho_n \alpha_{nm} \pi_{nm}$$

prob. of being in state m

prob. of attempting move from m to n

prob. of accepting move from m to n

prob. of being in state n

prob. of attempting move from n to m

prob. of accepting move from n to m

Microscopic Reversibility

- This condition satisfies

$$\rho\Pi = \rho$$

$$\sum_m \rho_m \pi_{mn} = \rho_n$$

- A scheme which satisfies microscopic reversibility was given by Metropolis et al.

Metropolis Method

- We let transitions from m to n proceed according to

$$\pi_{mn} = \begin{cases} \alpha_{mn}, & \rho_n \geq \rho_m, m \neq n \\ \alpha_{mn} \frac{\rho_n}{\rho_m}, & \rho_n < \rho_m, m \neq n \end{cases}$$

- Can remain in same state

$$\pi_{mm} = 1 - \sum_{n \neq m} \pi_{mn}$$

Metropolis Method

- In the standard Metropolis method, α is **symmetric**

and **stochastic**

$$\alpha_{mn} = \alpha_{nm}$$

note: nothing **requires** this! In fact, biased methods rely upon the use of asymmetric attempt probabilities to sample state space more efficiently.

$$\sum_n \alpha_{mn} = 1$$

Metropolis Method

- In the Metropolis formulation, the probability of accepting a move from m to n is given by

$$\min\left(1, \frac{\rho_n}{\rho_m}\right)$$

min function selects
the minimum value

Metropolis Acceptance Rules

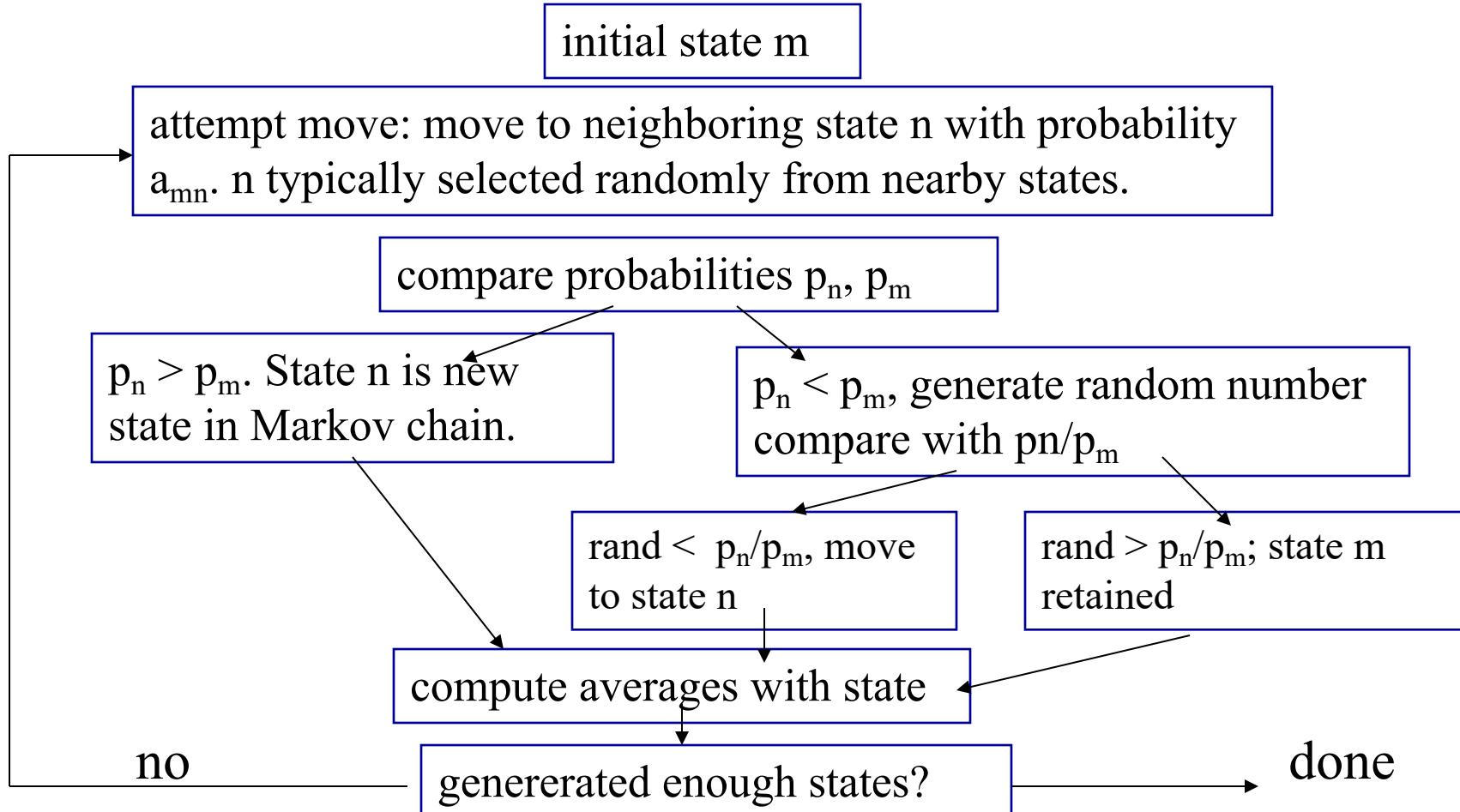
$$\min\left(1, \frac{\rho_n}{\rho_m}\right)$$

- If state n is more probable than state m, accept new state in our Markov chain.
- If state n is less probable than old state m, accept it with probability

$$\frac{\rho_n}{\rho_m} < 1$$

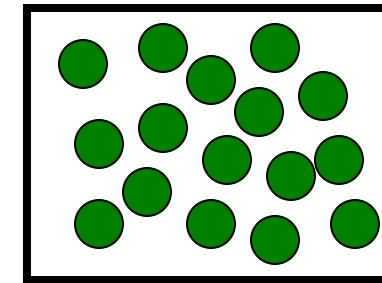
- What this means:
 - if new state is accepted, use it in collecting averages.
 - if new state is not accepted, retain old state and use it in computing averages.

Metropolis Algorithm



Example

- Consider a system of spherical molecules in the NVT ensemble.
Assume it is a “liquid”
and molecules interact via
a pair potential.
- Goal: generate “states” or
“conformations” that
asymptotically sample the NVT pdf.



Example

- The multi-dimensional “space” is

$$(r_1, r_2, \dots, r_N)$$

- PDF is given by

$$p_m = \rho_{NVT} (r_1^m, r_2^m, \dots, r_N^m) dr^N$$

volume of
configuration space

Example

- Consider a “move” to new state n . Note: this just involves a new configuration of molecules. Occurs with probability

- Ratio of probabilities given by $\frac{\exp[-\beta V(r_1^n, \dots, r_N^n)]}{Z}$

configurational
integrals cancel!

$$\frac{\rho_n}{\rho_m} = \frac{\frac{\exp[-\beta V(r_1^n, \dots, r_N^n)]}{Z}}{\frac{\exp[-\beta V(r_1^m, \dots, r_N^m)]}{Z}}$$

Example

- This ratio reduces to

$$\frac{\rho_n}{\rho_m} = \frac{\exp[-\beta V_n]}{\exp[-\beta V_m]} = \exp[-\beta \Delta V(m \rightarrow n)]$$

Z is not required! All that is needed is the **energy change** for the move.

Metropolis NVT Acceptance

- Acceptance probability for canonical ensemble reduces to

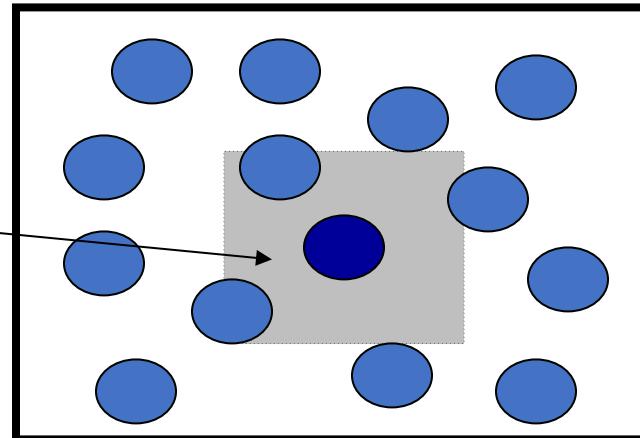
$$\min\left(1, \frac{\rho_n}{\rho_m}\right) = \begin{cases} 1, & V_n \leq V_m \\ \exp[-\beta\Delta V_{m \rightarrow n}], & V_n > V_m \end{cases}$$

- Metropolis selection rule only involves **differences in potential energy**. Configurational integrals are **not required** to compute properties of the system!!!

Implementation Details

- What is the form of α the attempt matrix?
- typically, $\alpha_{mn} = \alpha_{nm}$ equal attempt probabilities between states
- attempts only made over “small distances” of state space.

only attempt to move an atom over a short distance, where a “cavity” is likely to be.



Implementation Details

- Method
- pick atom i at random ($r_i(m)$)
- define “local environment” - cube of edge length R
- Only attempt moves within R

$$\alpha_{mn} = \frac{1}{N_R}; r_i^n \in R$$

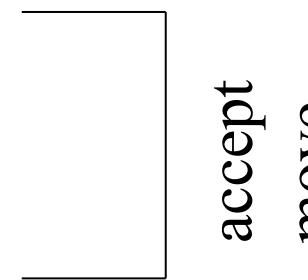
$$\alpha_{mn} = 0; r_i^n \notin R$$

Implementation Details

- Easy to implement with spherical molecules.
- “Moves” correspond to random Cartesian displacement.
 - adjustable parameter dr_{max} controls “boldness” of moves.
 - “bold” moves give large configurational changes (good) but increase likelihood that moves get rejected (bad).
- dr_{max} set “on the fly” to accept ca. 50%
- pseudocode

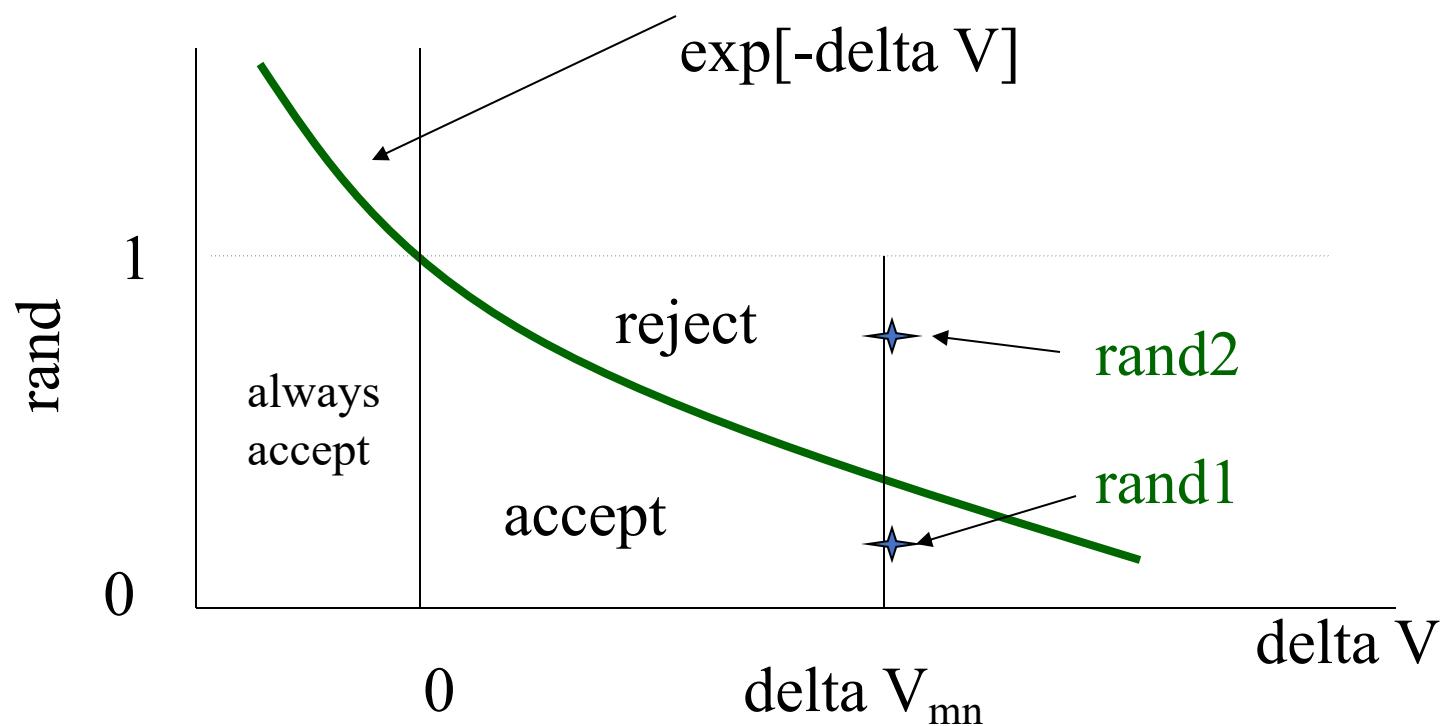
```
rxnew = rx + (2.0*(rand) - 1.0) * drmax
```

```
if (deltaV .le. 0)  
or if (exp[-beta*deltaV] .le. rand)  
then rx = rxnew
```

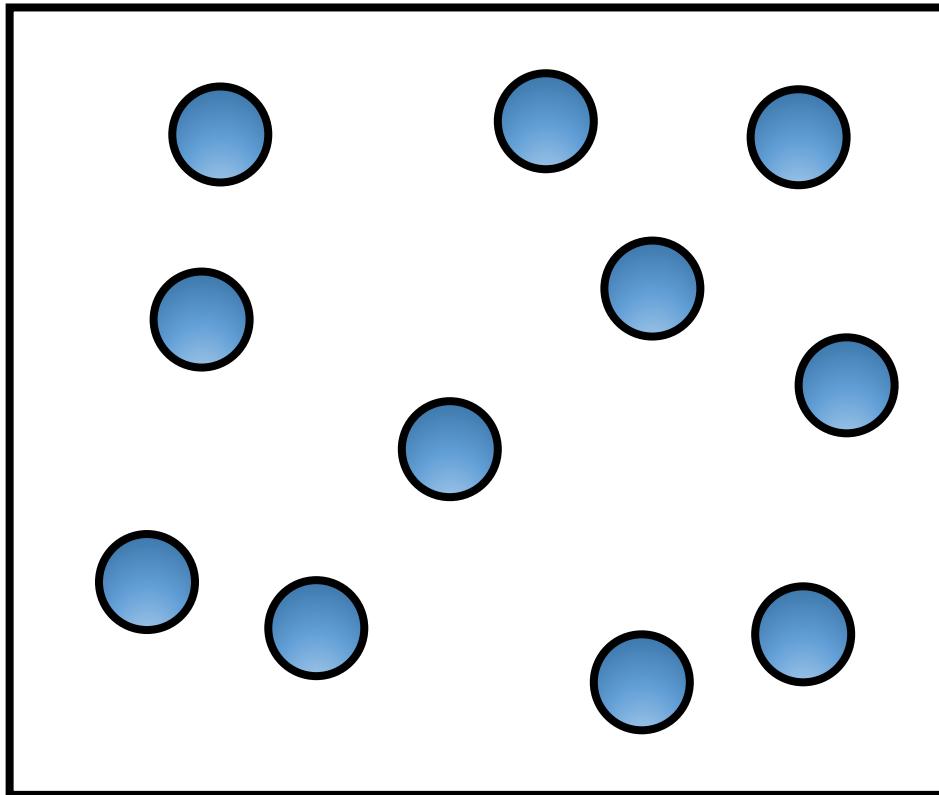


Implementation Details

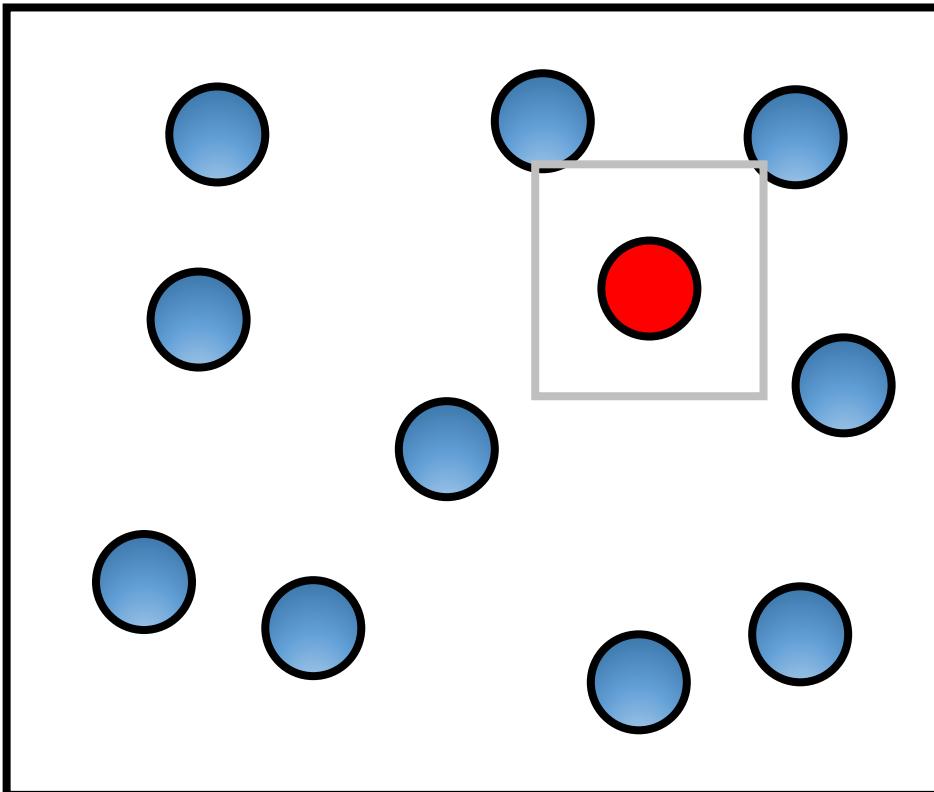
- Graphically, acceptance rule



Atomic systems

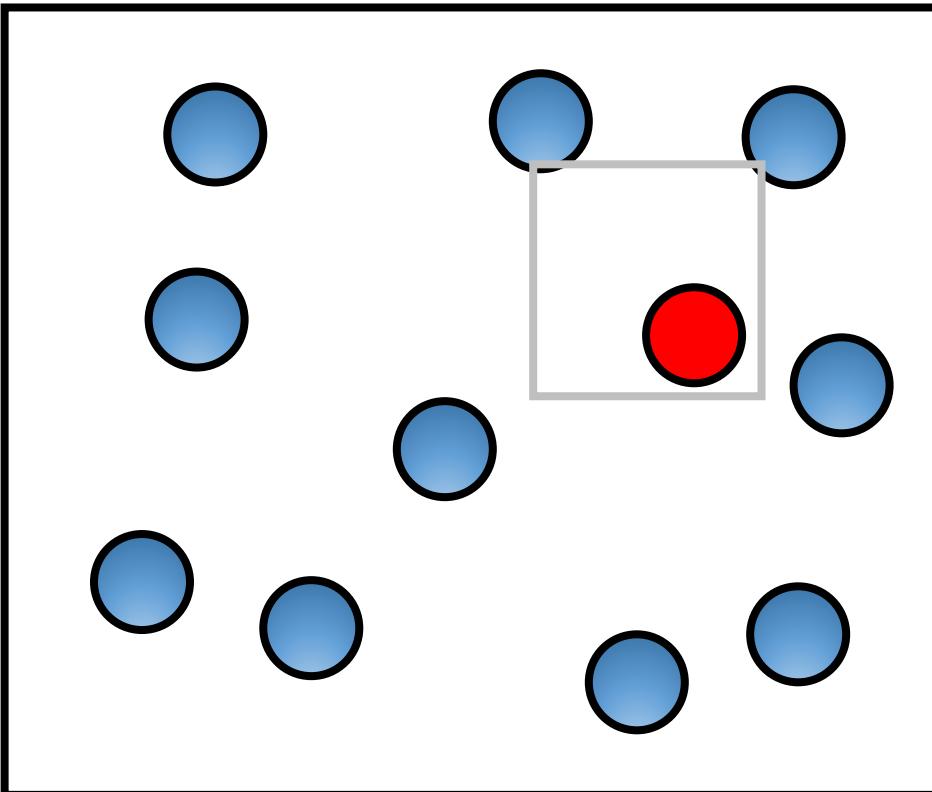


Atomic systems



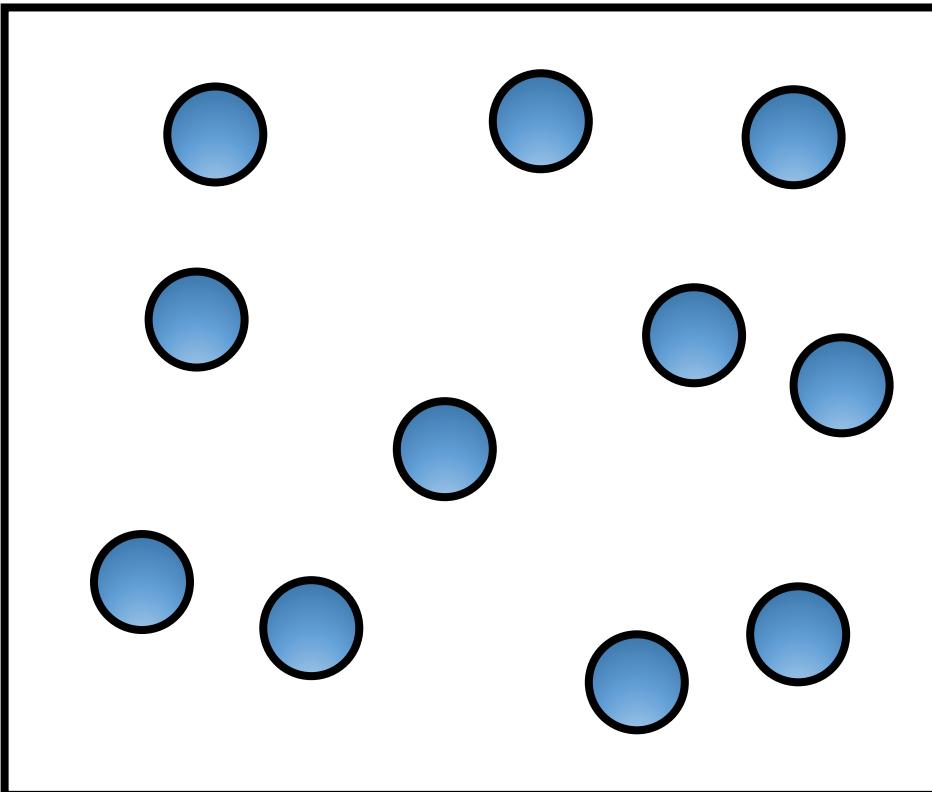
Choose random molecule and define region where move can be attempted

Atomic systems



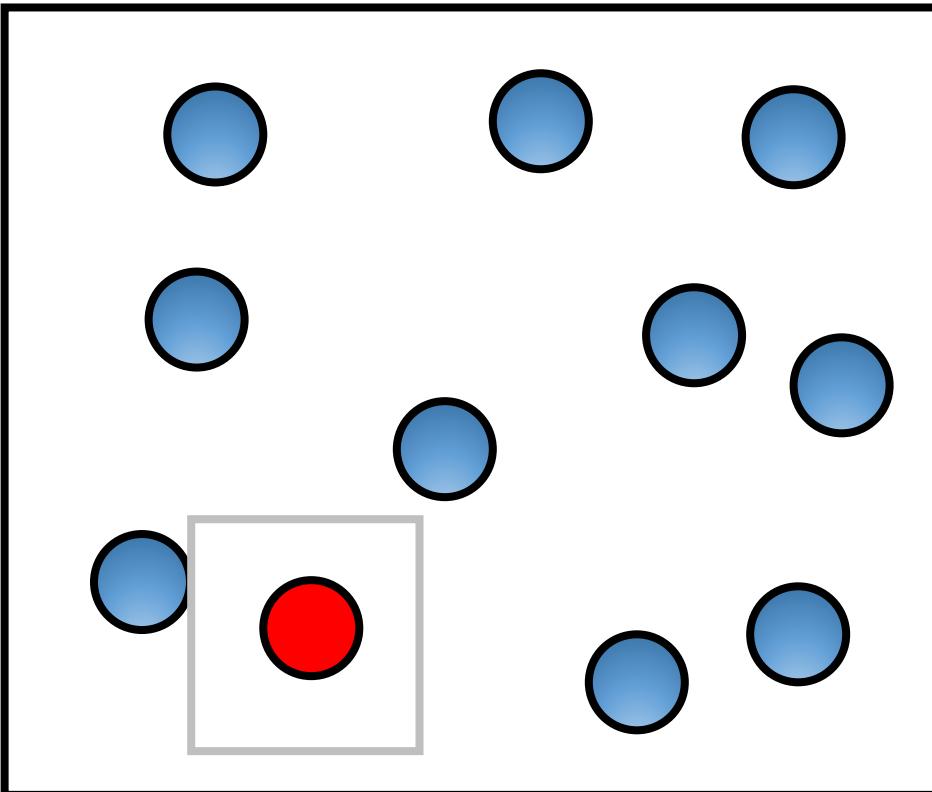
Make random move

Atomic systems



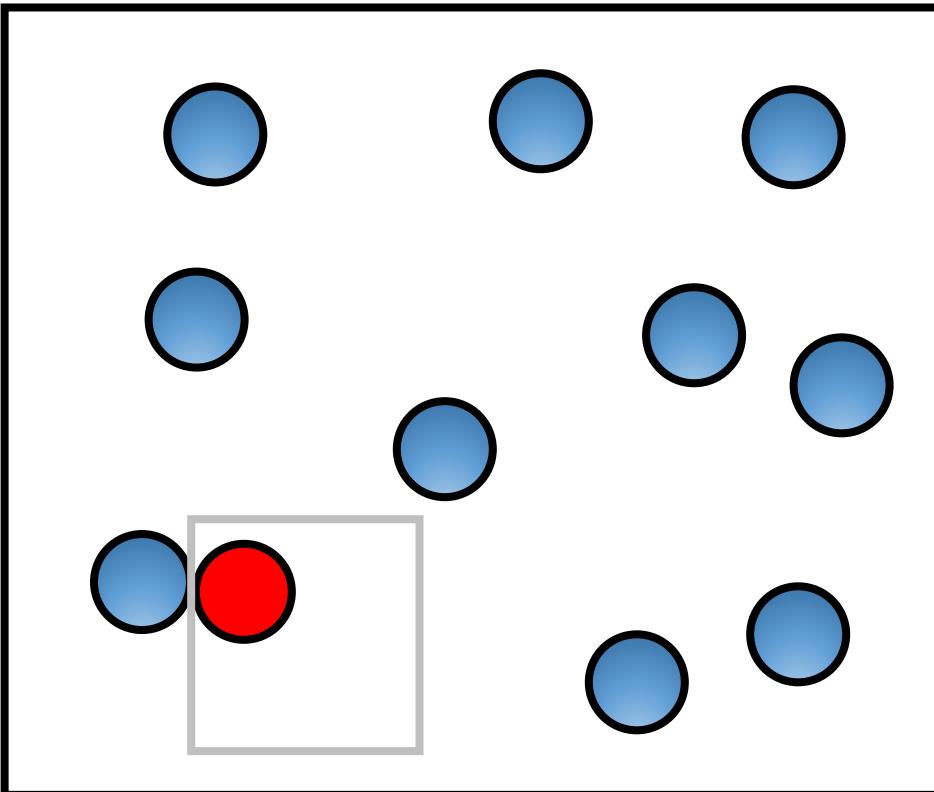
This move is accepted

Atomic systems



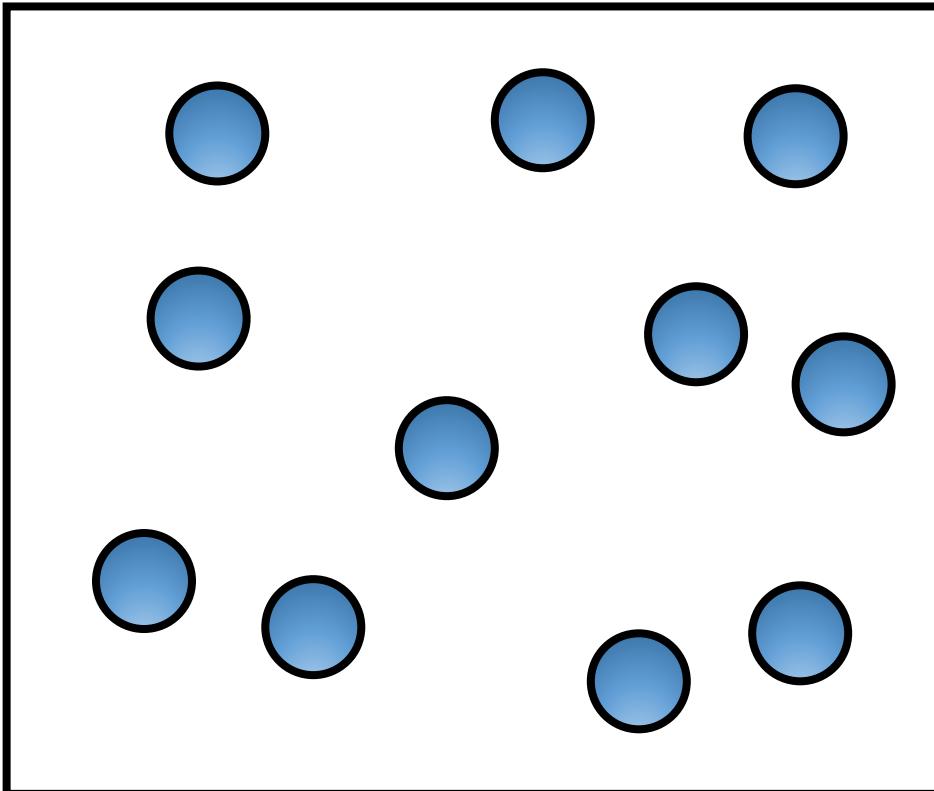
Choose random molecule and define region where move can be attempted

Atomic systems



Make random move

Atomic systems

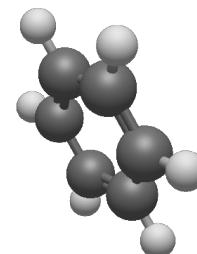
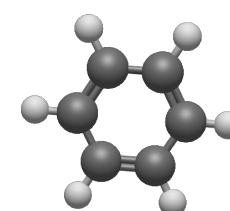
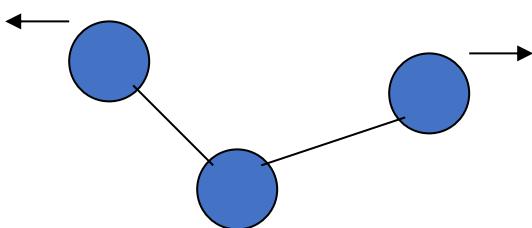


This move is rejected. Go back to old state.

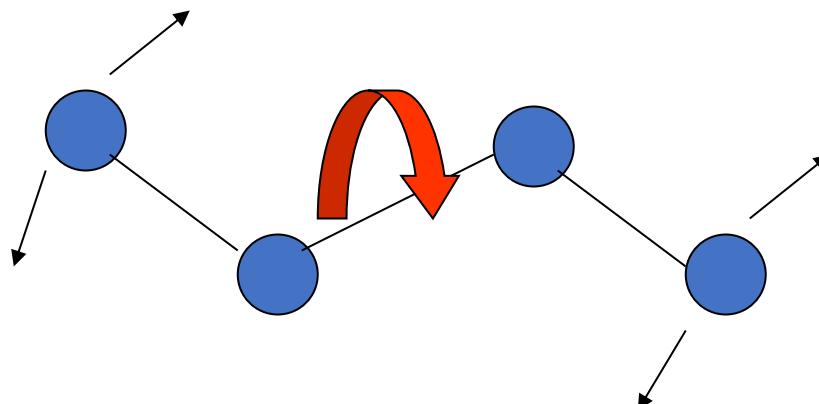
Degrees of freedom: Molecules

- For molecular systems, we can have other “moves”

- Rigid body rotations



- bond distortions



- bond rotations

Moves for molecules

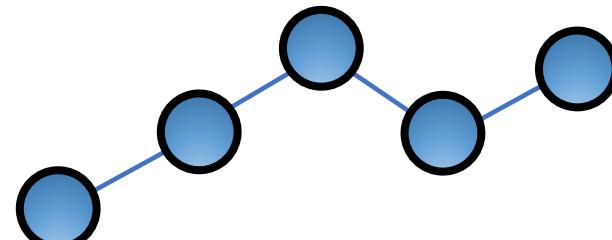
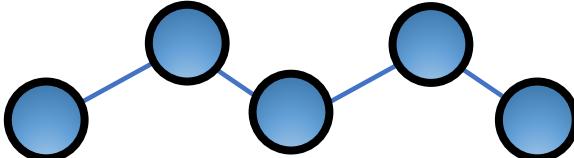
- Typically work in “internal” coordinates
 - Eulerian angles (molecule rotation)
 - Bond lengths, angles; dihedral angles
- Changing internal coordinates perturbs more than one atom position



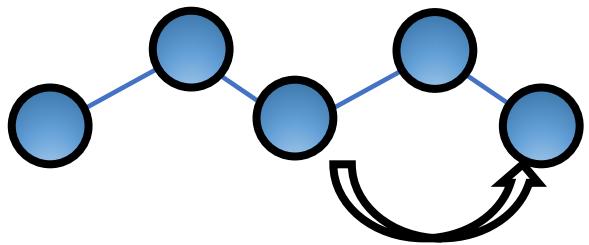
Dihedral angle
change

Moves for molecules

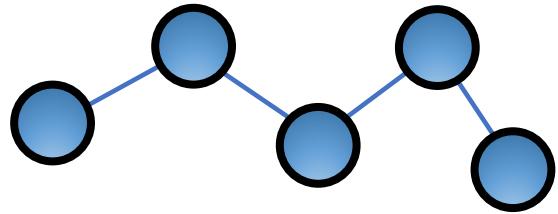
- Typically work in “internal” coordinates
 - Eulerian angles (molecule rotation)
 - Bond lengths, angles; dihedral angles
- Changing internal coordinates perturbs more than one atom position



Angle change

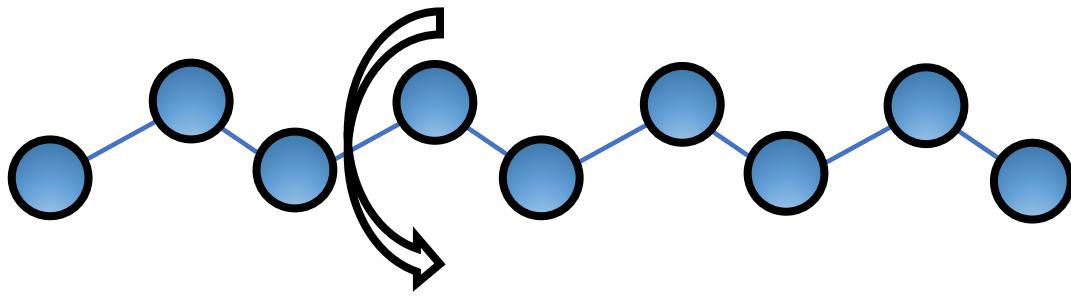


Angle change

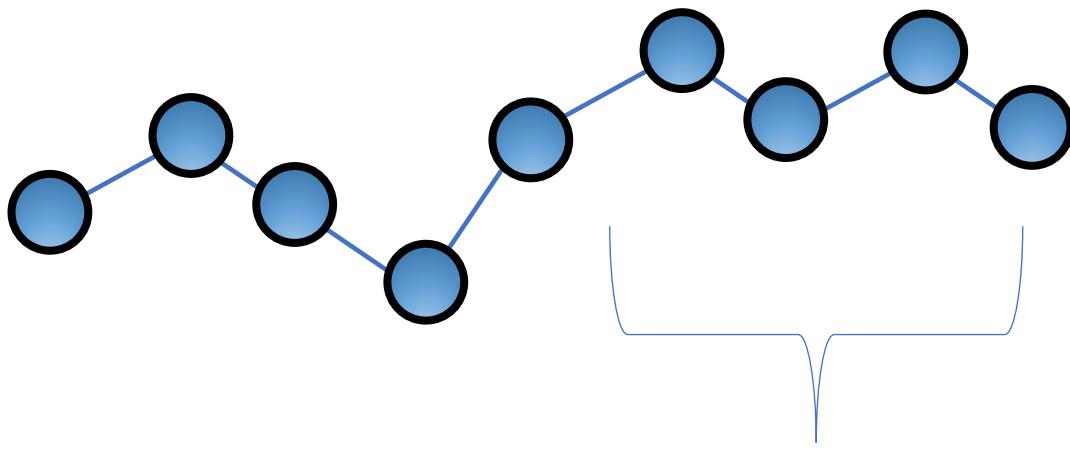


“stiff” degrees of freedom
like bonds and angles
require preferential
sampling – don’t sample on
full 180° range!

Large molecules – magnify changes



Large molecules – magnify changes



High probability of
overlap of many atoms
from perturbing a single
dihedral angle

Solution: Biased Monte Carlo

- Many variations and implementations
- Bias introduced in attempt probabilities $\alpha(i \rightarrow j)$

- Removed via acceptance rule

$$P_{accept}(i \rightarrow j) = \min \left[1, \frac{\alpha(j \rightarrow i)\rho(j)}{\alpha(i \rightarrow j)\rho(i)} \right]$$

- Logical choice $\alpha(i \rightarrow j) \approx \rho(j)$

$$\alpha(j \rightarrow i) \approx \rho(i)$$

- Thus $P_{accept}(i \rightarrow j) \approx 1$

Siepmann, J. I., and Frenkel, D., 1992, *Molec. Phys.*, 75, 59.

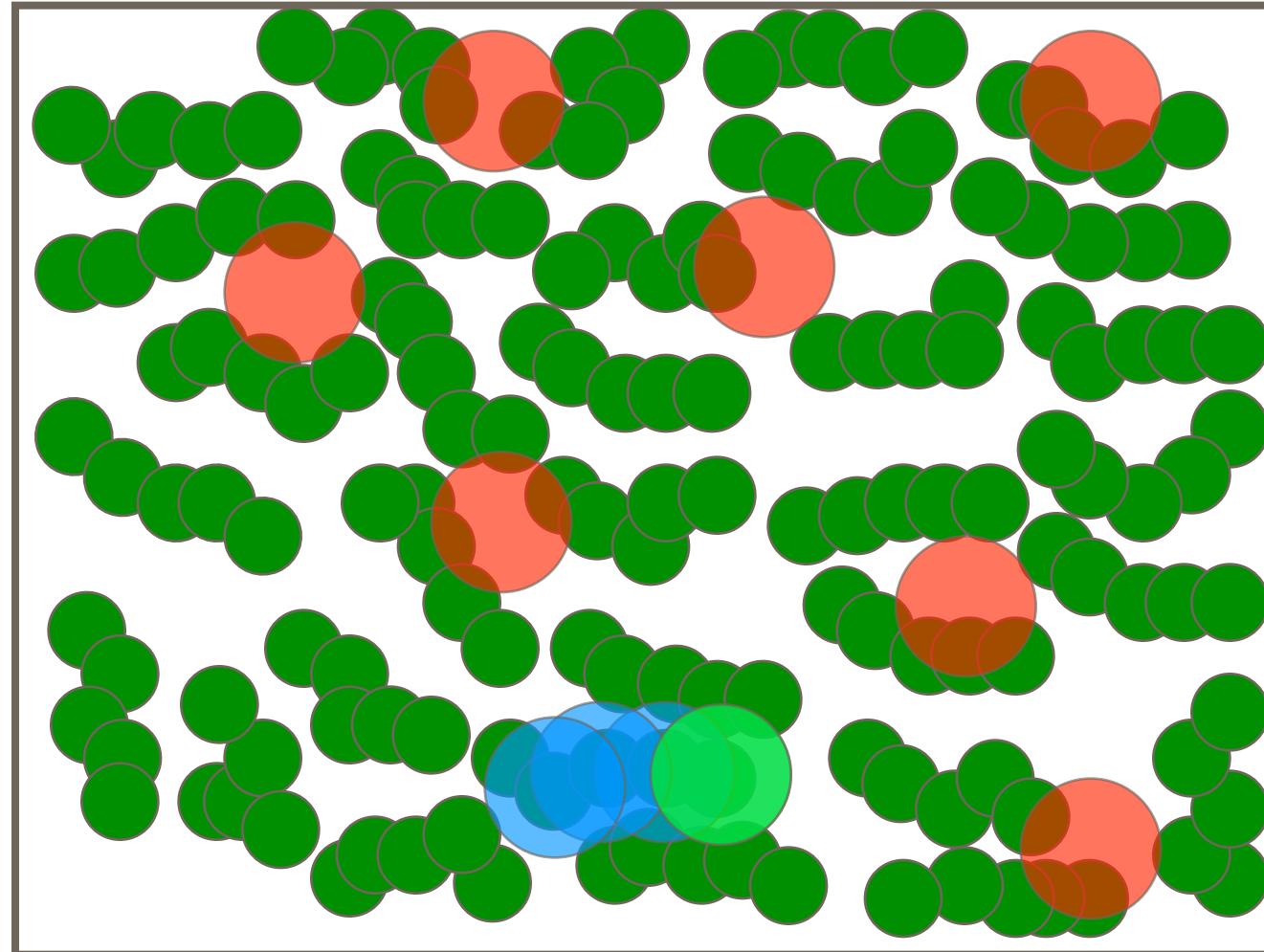
Frenkel, D., Mooij , G. C. A. M., and Smit, B., 1992, *J. Phys. Condens. Matter*, 4, 3053.

de Pablo, J. J., Laso, M., and Suter, U. W., 1992, *J. Chem. Phys.*, 96, 2395.

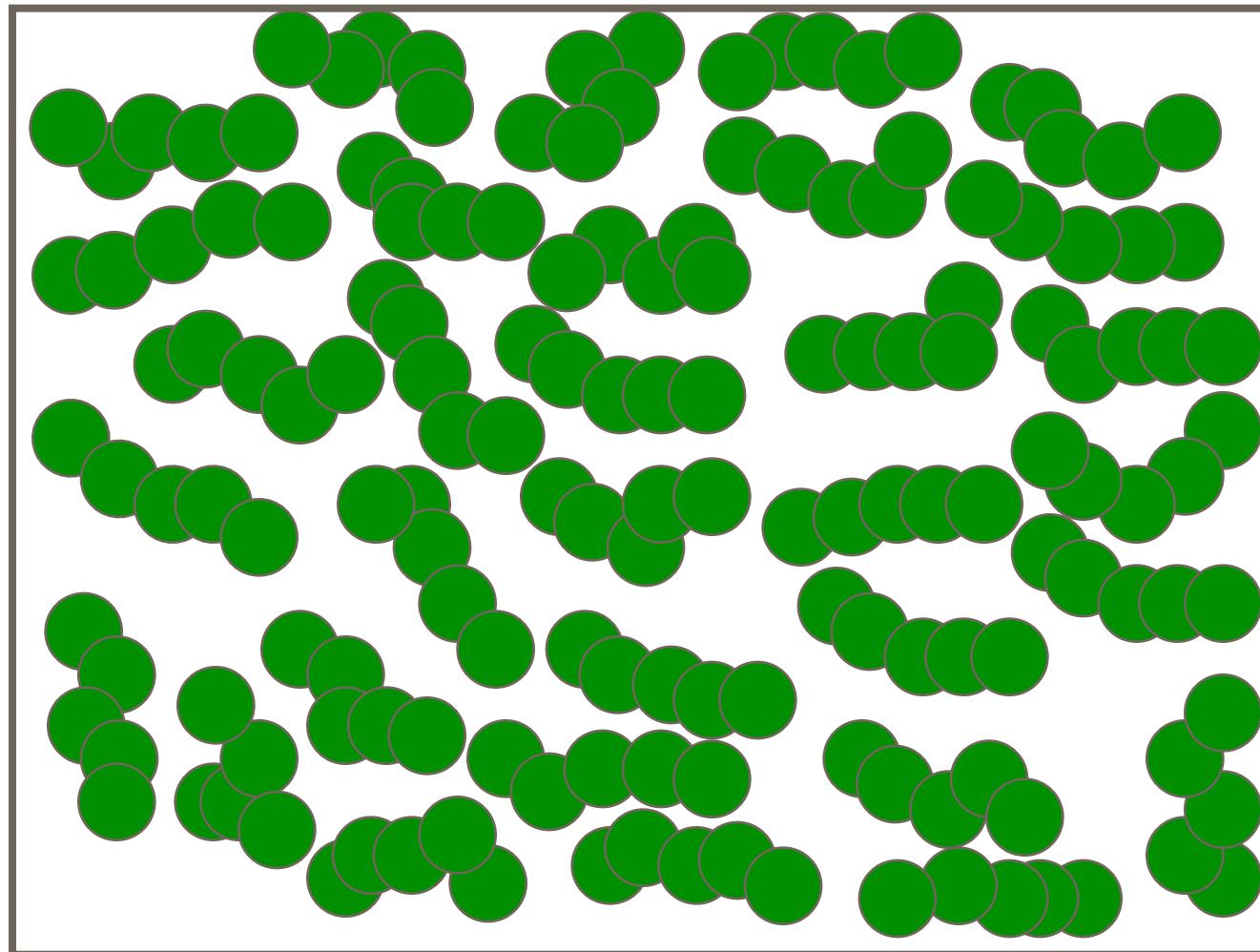
Macedonia, M. D. and Maginn, E. J., *Molec. Phys.*, 1999, 96, 1375

Configurational bias Monte Carlo

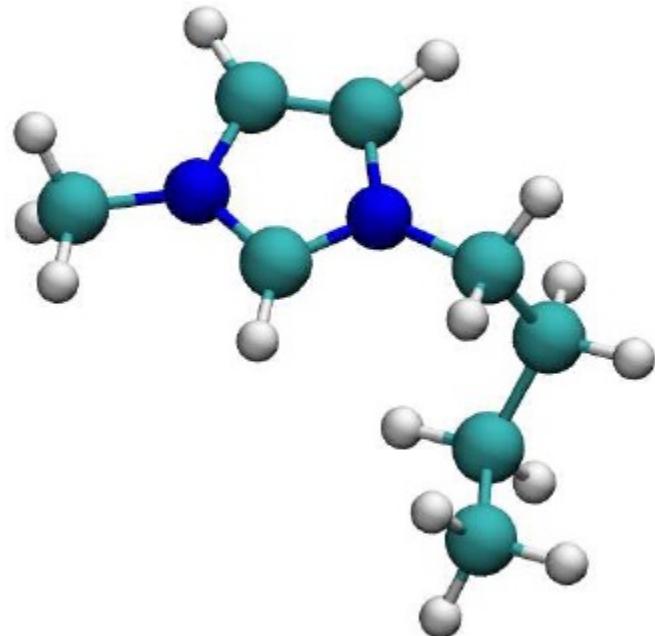
Consider insertion of an additional molecule



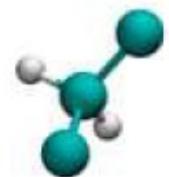
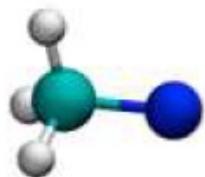
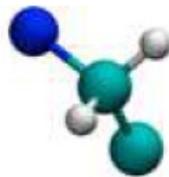
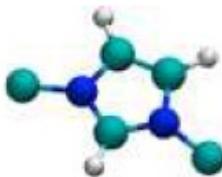
Configurational bias Monte Carlo



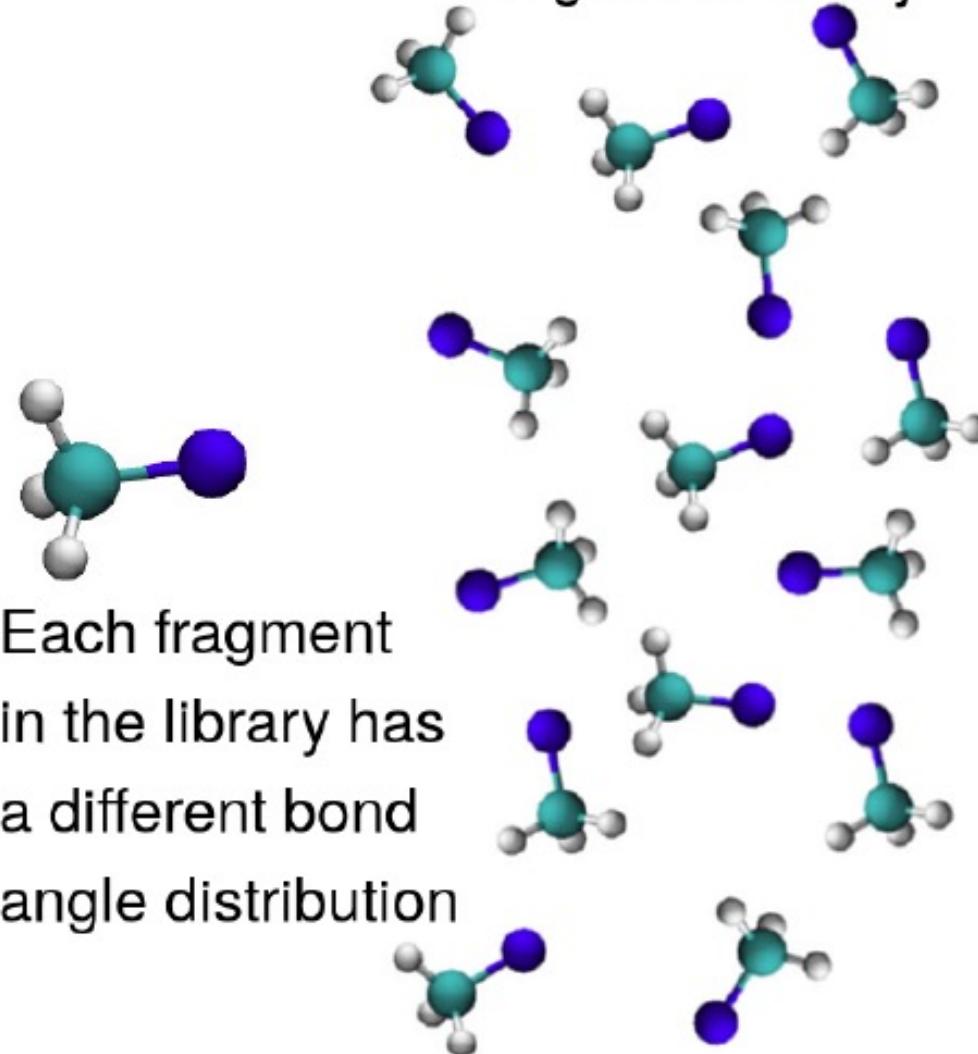
Cassandra uses fragment sampling



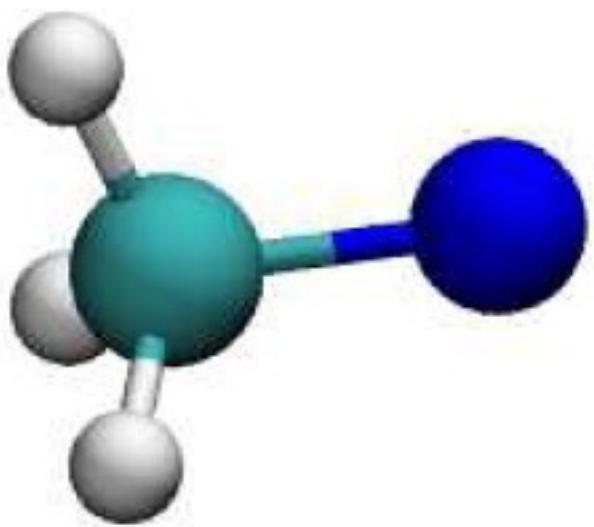
Fragment sampling

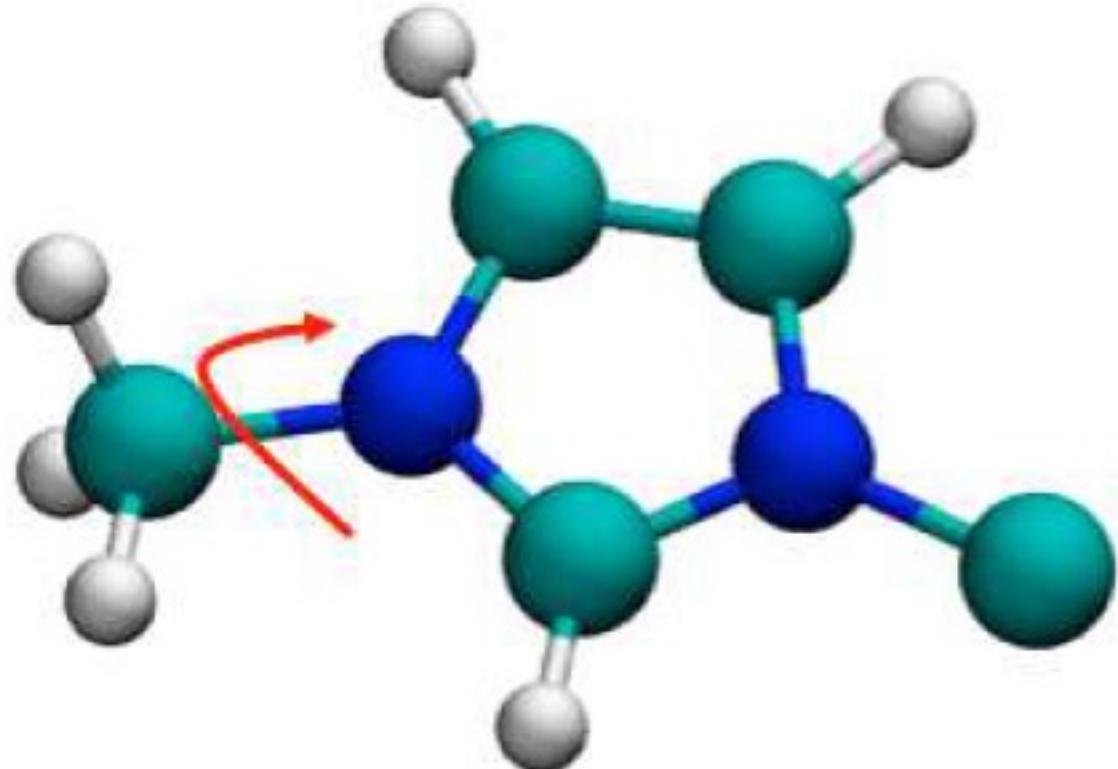


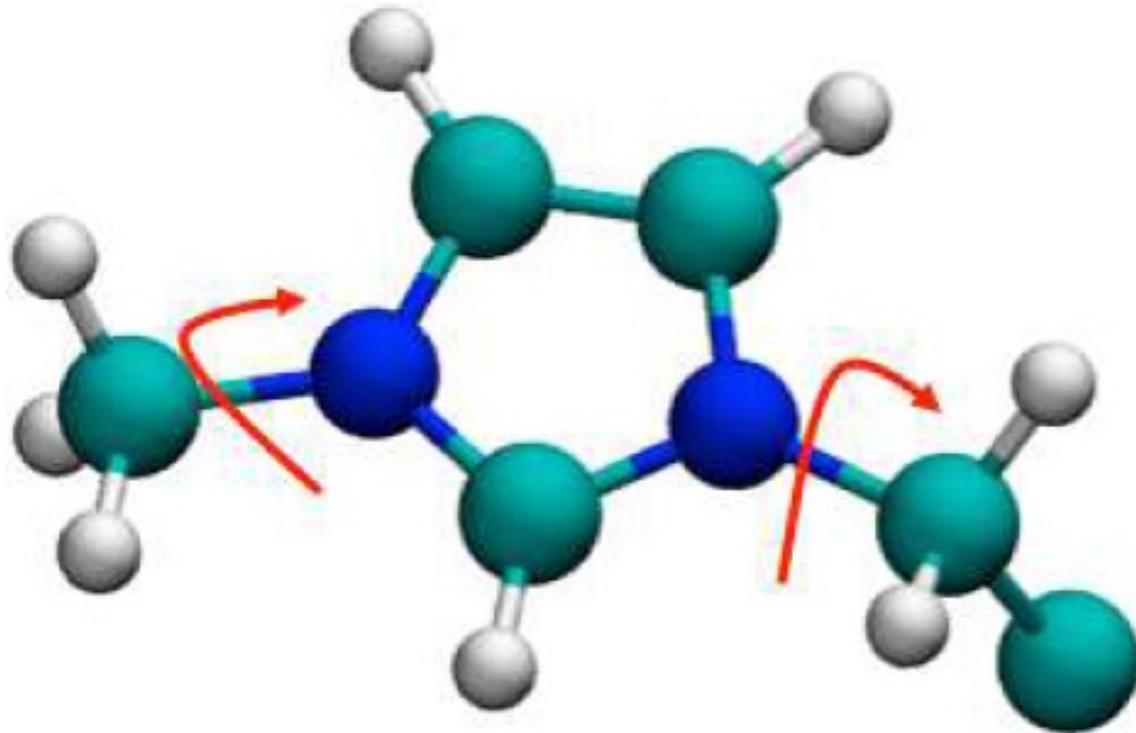
Fragment Library

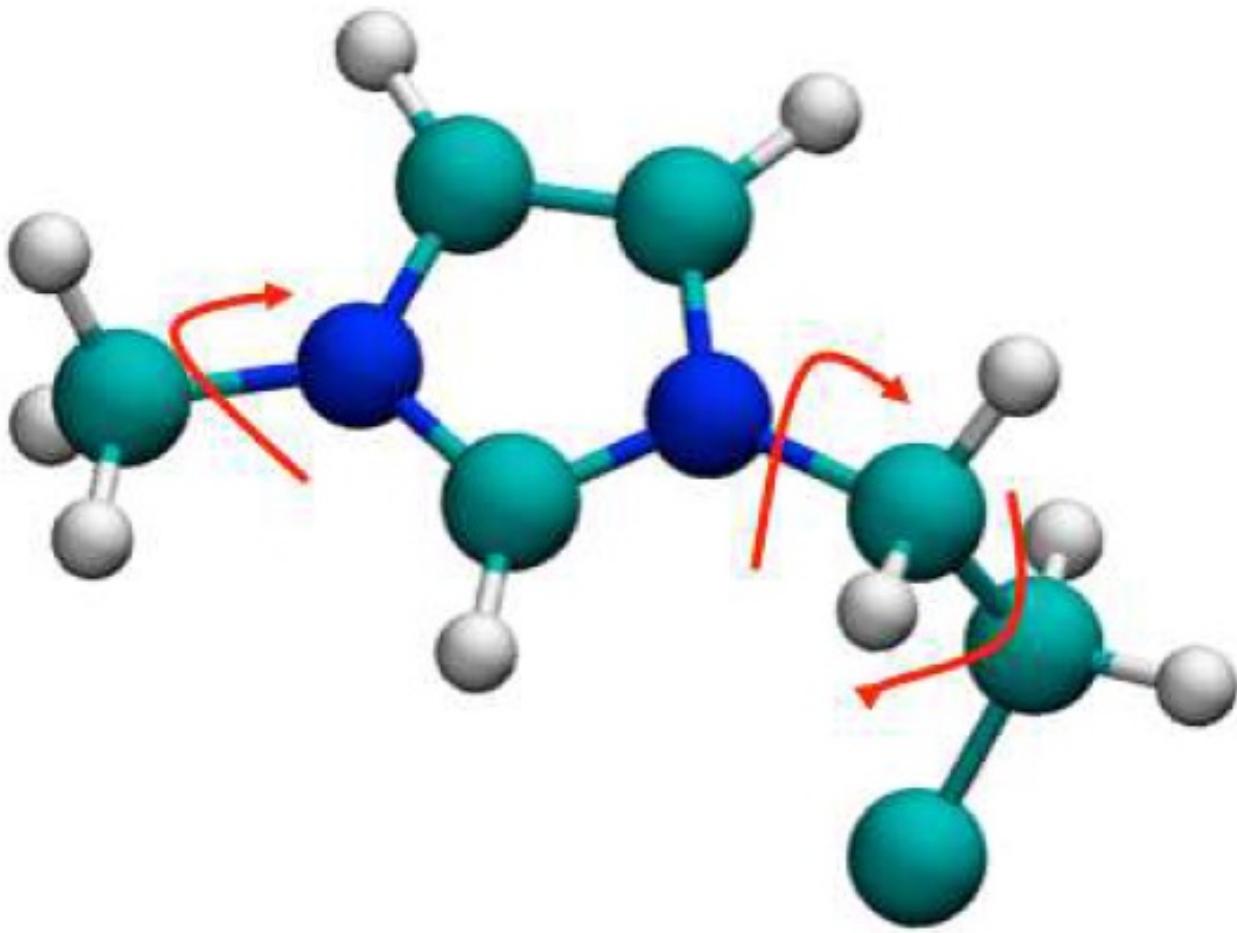


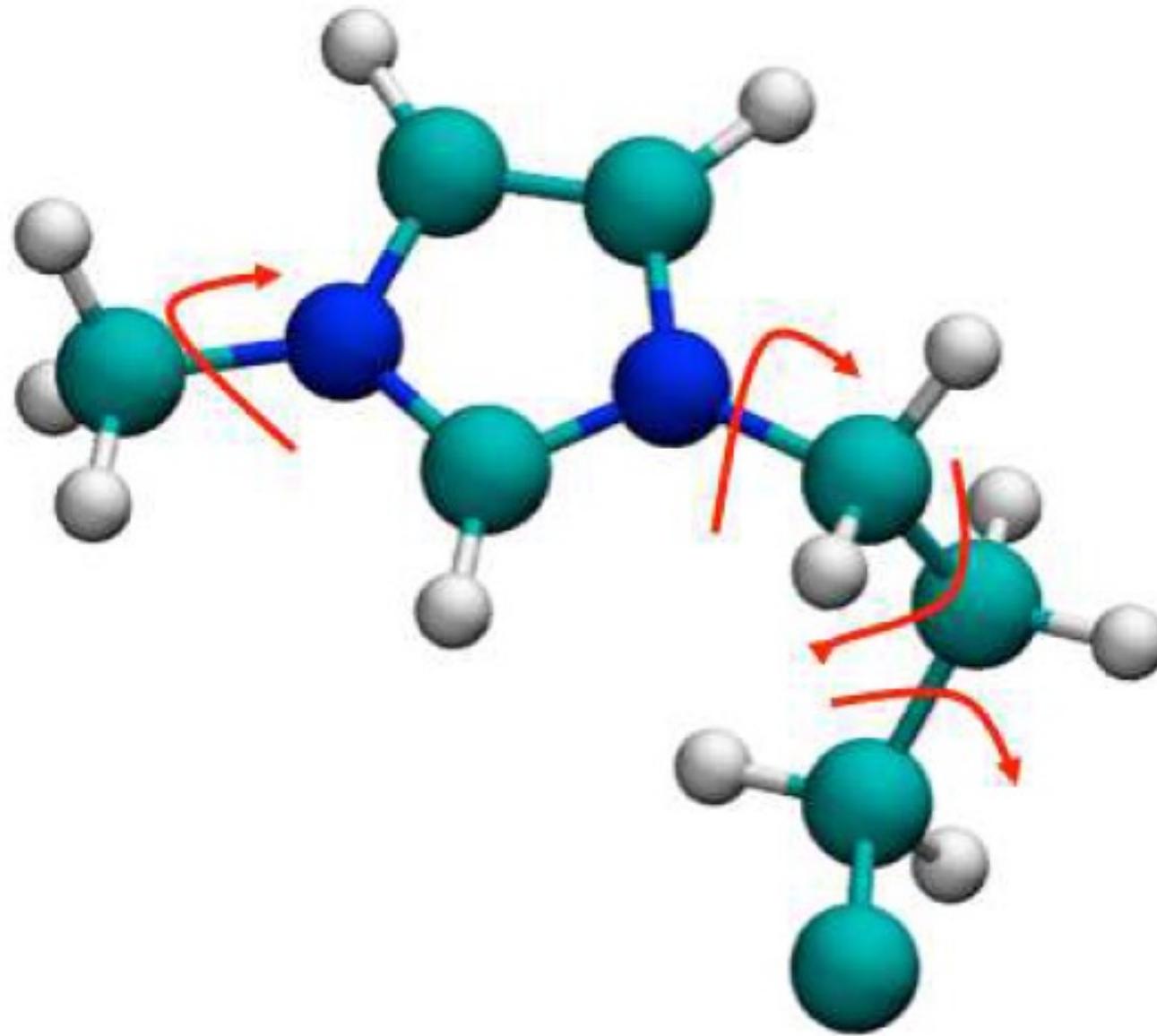
Each fragment
in the library has
a different bond
angle distribution

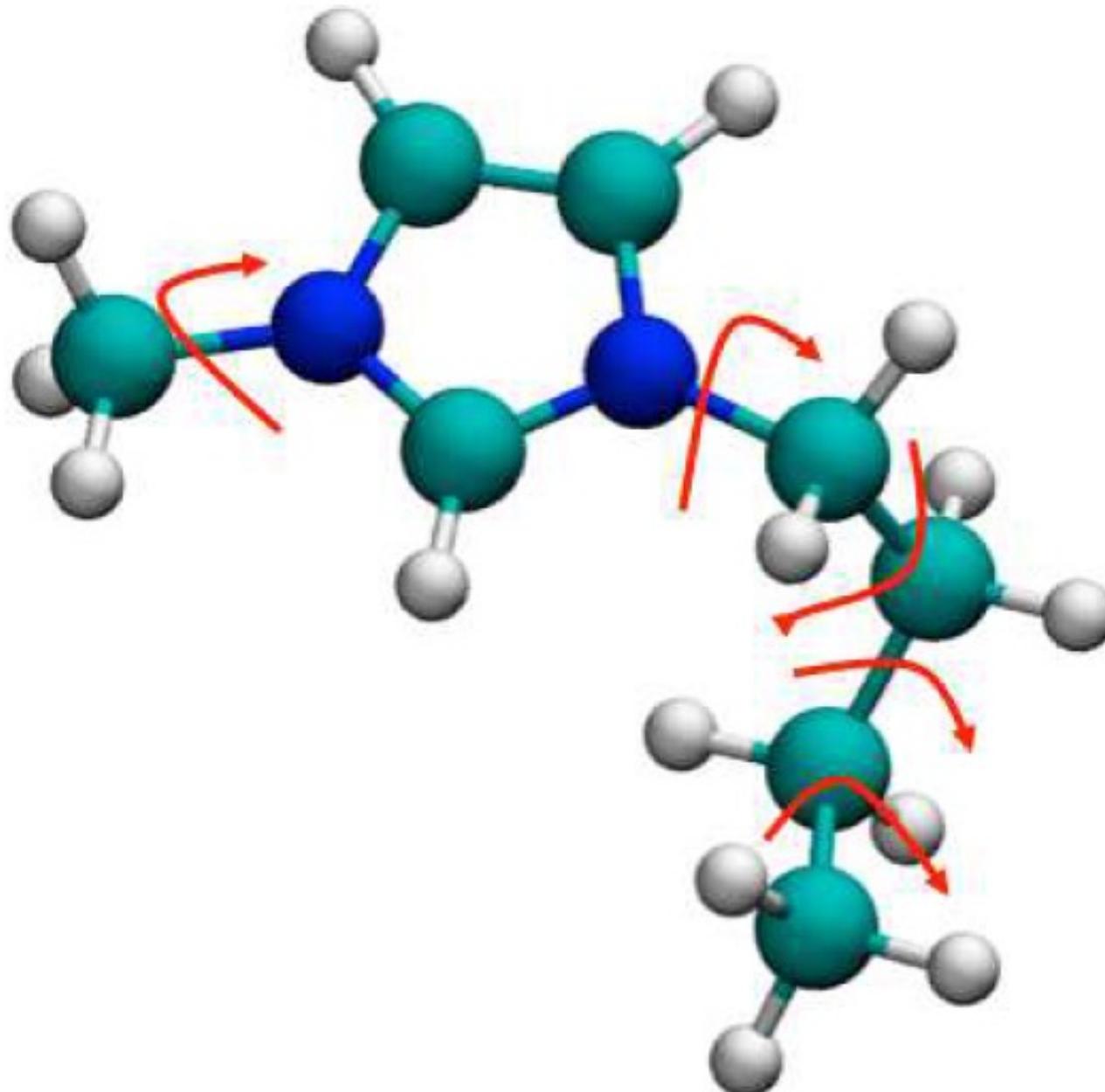












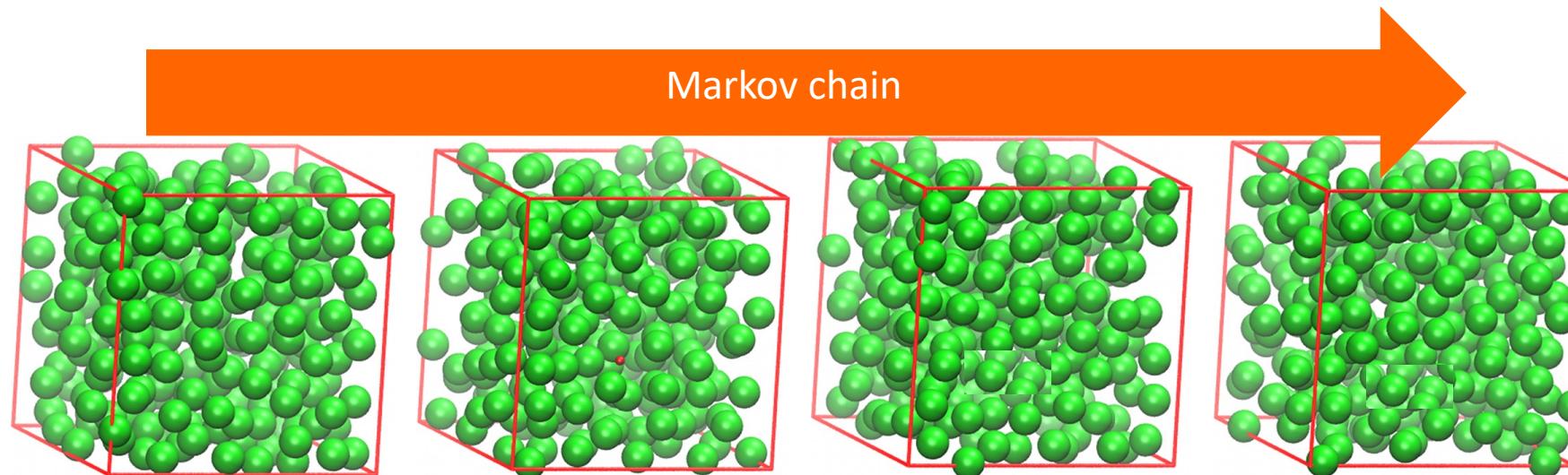
Atomistic Monte Carlo

- Sample configurations *stochastically* according to an ensemble probability density

$$\rho^{NVT} \propto \exp\left[-\frac{\mathcal{V}(r)}{k_B T}\right]$$

- Conformation “accepted” via Metropolis rule

$$acc(o \rightarrow n) = \min\left(1, \frac{\alpha(n \rightarrow o)\rho_n}{\alpha(o \rightarrow n)\rho_o}\right)$$



Cassandra

- Open source (GPL) atomistic Monte Carlo package developed at the University of Notre Dame
- Available at: cassandra.nd.edu or github.com/MaginnGroup/Cassandra
- Capabilities
 - NVT, NPT, GEMC (NVT and NPT), GCMC ensembles
 - Handles popular force fields (Amber, CHARMM): Lennard-Jones 12-6 or Mie-potential + fixed partial charges
 - Ewald and damped shifted force method for charges
 - Parallelized with OpenMP
 - Comes with several setup and analysis scripts

Cassandra code

The screenshot shows the GitHub repository page for **MaginnGroup / Cassandra**. The repository is public and has 794 commits across 2 branches and 10 tags. The main content area displays a list of recent commits by **emarinri**, including changes to GitHub templates, documentation, examples, scripts, and source code. The right sidebar provides an **About** summary, links to the project's website (**cassandra.nd.edu/**), and metrics like 27 stars, 7 watching, and 17 forks. It also lists 6 releases, with the latest being version 1.3.0 from July 16. A note indicates no packages have been published.

About

Cassandra is a Monte Carlo package to conduct atomistic simulations.

cassandra.nd.edu/

molecular-simulation
monte-carlo-simulation
atomistic-simulations

Readme
GPL-3.0 license
27 stars
7 watching
17 forks

Releases 6

1.3.0 **Latest**
on Jul 16

+ 5 releases

Packages

No packages published
Publish your first package

Code

master 2 branches 10 tags

Go to file Add file Code

emarinri Change error message when reaching volume moves in log space ad5bdd1 on Aug 2 794 commits

.github Update GitHub templates. 4 months ago

Documentation Bump version: 1.2.6 → 1.3.0 3 months ago

Examples Fix GEMC examples and prevent log volume moves 2 months ago

Scripts Fix GEMC examples and prevent log volume moves 2 months ago

Src Change error message when reaching volume moves in log space 2 months ago

.bumpversion.cfg Bump version: 1.2.6 → 1.3.0 3 months ago

.gitignore Add __pycache__ and failureLog dirs to gitignore 3 years ago

CONTRIBUTING.md Minor changes to contributing guidelines 3 years ago

LICENSE Rename license file 3 years ago

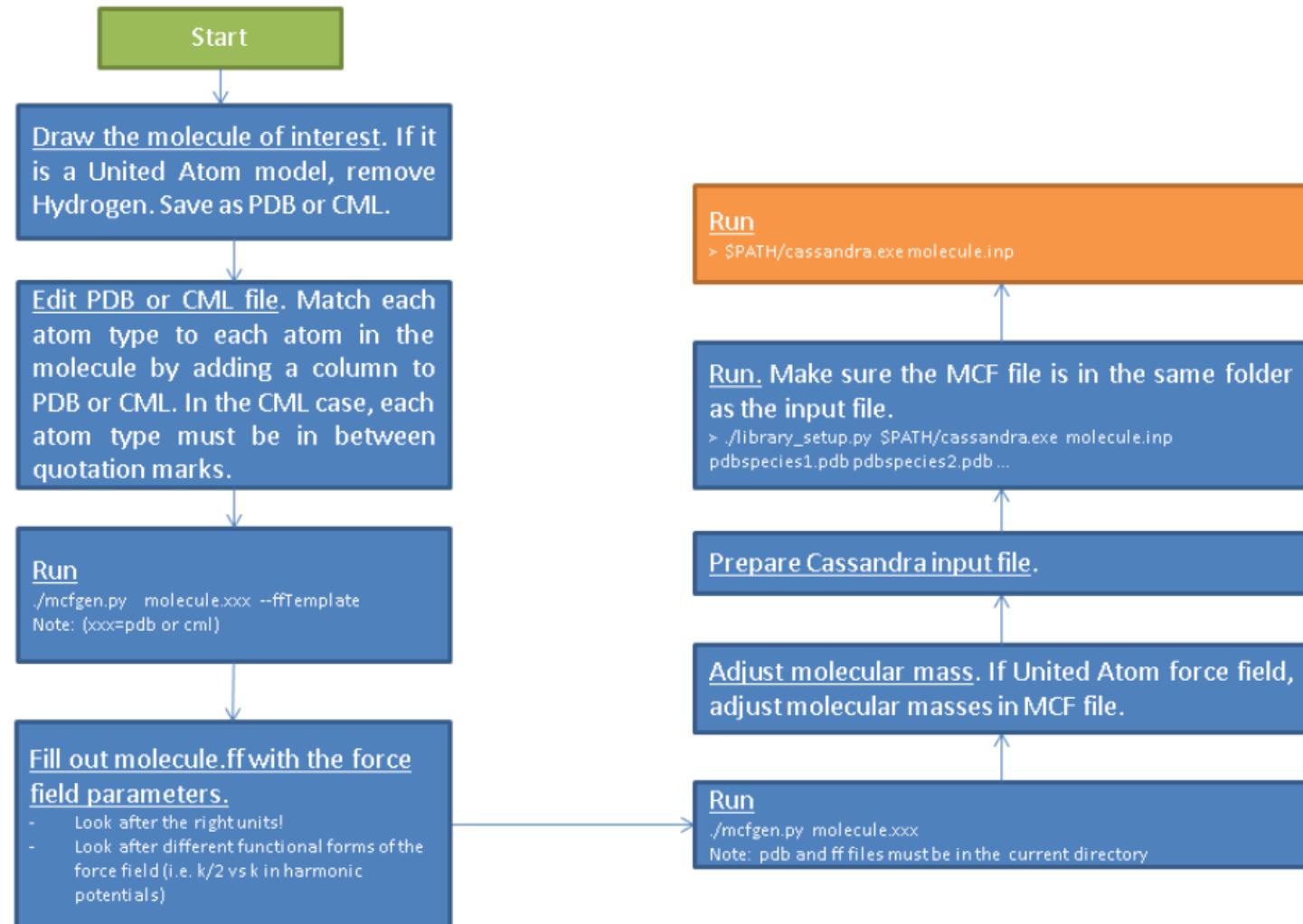
README.md Doc additions: restricted insertions, RTD link in README (#76) 2 years ago

azure-pipelines.yml Switch cron build to weekly/always exec 17 months ago

requirements-py.txt Add file to describe python requirements for scripts 3 years ago

README.md

Simulation workflow



Required files

- **Input file** – contains information on the ensemble, system to be simulated, move probabilities, simulation type and length, etc.
- Edit manually and follow documentation

```
File Edit Options Buffers Tools Help
! This is the input file for a short simulation of water
nvt.out
!-----

# Sim_Type
nvt_mc
!-----

# Nbr_Species
1
!-----

# VDW_Style
lj cut_tail 7.0
!-----

# Charge_Style
coul ewald 7.0 0.000001
!-----

# Seed_Info
21498 489625
!-----

# Rcutoff_Low
0.85
!-----

# Molecule_Files
spc.mcf 90
!-----one line per species

# Box_Info
1
cubic
14.02
!-----line 1: number of boxes
!-----line 2: box type
!-----line 3: box size

# Temperature_Info
300.0
!-----
```

Required files

- Molecular connectivity file (*.mcf) – contains atomic connectivity, force field information, atom labels
- More difficult; use scripts
- Check out Examples
- mosdef_cassandra automates this

```
*****
!Molecular connectivity file for spc.pdb
*****
!Atom Format
!index type element mass charge vdw_type parameters
!vdw_type="LJ", parms=epsilon sigma
!vdw_type="Mie", parms=epsilon sigma repulsion_exponent dispersion_exponent

# Atom_Info
3
1    O_s1    0    15.999   -0.82    LJ    78.208   3.166
2    H_s1    H    1.008    0.41    LJ    0.000    0.000
3    H_s1    H    1.008    0.41    LJ    0.000    0.000

!Bond Format
!index i j type parameters
!type="fixed", parms=bondLength

# Bond_Info
2
1    1    2    fixed    1.000
2    1    3    fixed    1.000

!Angle Format
!index i j k type parameters
!type="fixed", parms=equilibrium_angle
!type="harmonic", parms=force_constant equilibrium_angle

# Angle_Info
1
1    2    1    3    fixed    109.47

# Dihedral_Info
0

# Improper_Info
0

!Intra Scaling
!vdw_scaling    1-2 1-3 1-4 1-N
!charge_scaling 1-2 1-3 1-4 1-N

# Intra_Scaling
0. 0. 0.0000 1.
0. 0. 0.0000 1.

!Fragment Format
!index number_of_atoms_in_fragment branch_point other_atoms
```

Cassandra system setup workflow without MoSDeF

Received: 2 February 2021 | Revised: 6 April 2021 | Accepted: 11 April 2021

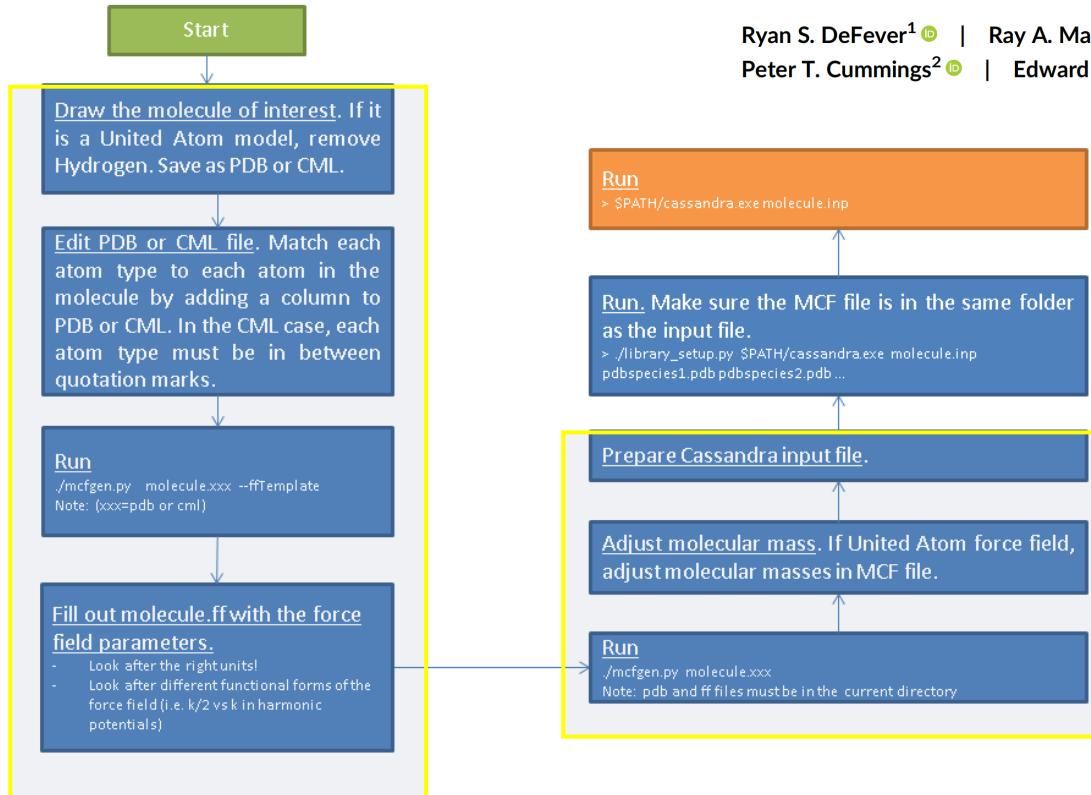
DOI: 10.1002/jcc.26544

SOFTWARE NOTE

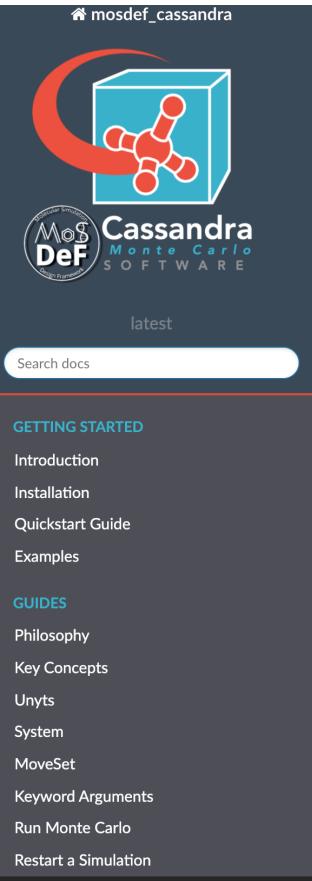


MoSDeF Cassandra: A complete Python interface for the Cassandra Monte Carlo software

Ryan S. DeFever¹ | Ray A. Matsumoto² | Alexander W. Dowling¹ | Peter T. Cummings² | Edward J. Maginn¹



= steps replaced by mbuild+foyer



MoSDeF Cassandra

[cite](#) [mosdef_cassandra](#) [codecov](#) 97% [Azure Pipelines](#) failed

Overview

MoSDeF Cassandra is a Python interface for the Cassandra Monte Carlo software. It offers complete integration with the MoSDeF tools and a user-friendly interface for Cassandra.

Warning

MoSDeF Cassandra is still in early development (0.x releases). The API may change unexpectedly.

Resources

- [Installation guide](#): Instructions for installing MoSDeF Cassandra
- [Key Concepts](#): How we think about MoSDeF Cassandra
- [GitHub repository](#): View the source code, contribute, and raise issues
- [Cassandra](#): Learn more about the Cassandra Monte Carlo software
- [Cassandra repository](#): View the source of the Cassandra Monte Carlo software
- [MoSDeF tools](#): A collection of tools for constructing systems and applying forcefield parameters for particle-based simulations

Citation

Please cite [MoSDeF Cassandra](#), [Cassandra](#), and the [MoSDeF](#) suite of tools if you use this tool in your research. See [here](#) for details.

Stable Python wrapper for Cassandra that enables seamless use of MoSDeF tools

1. Create the System
2. Create the MoveSet
3. Pass System and MoveSet to run function

mosdef_cassandra

We have generated 4 examples for you:

- NVT simulation of methane: NVT.ipynb
- NPT simulation of ethanol using damped shifted force: NPT.ipynb
- Gibbs ensemble simulation of methane: gemc.py
- Grand canonical ensemble simulation of methane adsorption in silicalite:
 - p_vs_mu.ipynb; Methane absorption.ipynb
 - Standard mosdef_cassandra Python script for you to use
 - Jupyter notebook for the workshop

Default values (can override)

NVT:

- prob_translate = 0.33
- prob_rotate = 0.33
- prob_regrow = 0.34
- max_translate: 2.0 Angstroms
- max_rotate : 30.0 degrees
- max_volume : 500 Angstroms³ for Box 1,
5000 Angstroms³ for Box 2
- max_dihedral : 0.0 degrees
- cbmc_n_insert : 10
- cbmc_n_dihed : 10
- cbmc_rcut : 6.0 Angstroms

Cassandra is smart enough to know which of these don't apply to a particular molecule (no cbmc moves for spheres, etc.

Customization

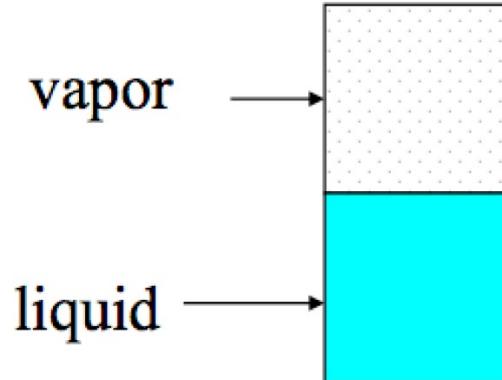
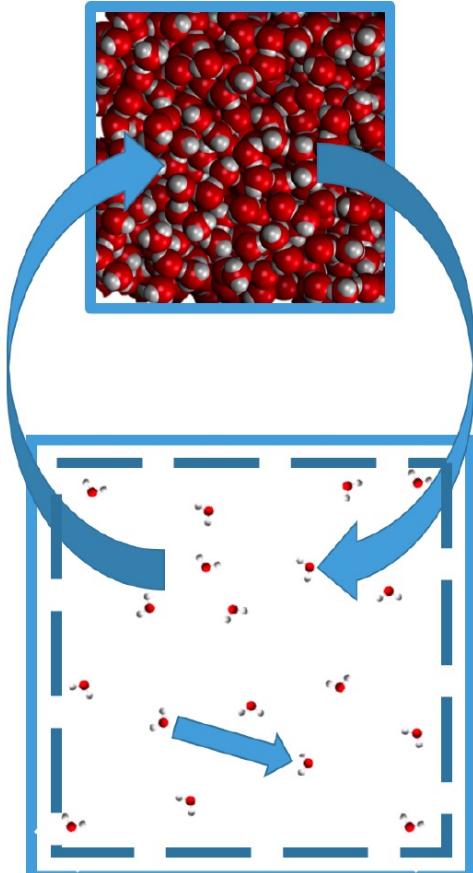
```
thermo_props = [  
    "energy_total",  
    "energy_intervdw",  
    "pressure",  
    "volume",  
    "nmols",  
    "mass_density",  
]
```

Tell code what properties
to write to *.prp file

```
custom_args = {  
    "run_name": "equil",  
    "charge_style": "none",  
    "rcut_min": 2.0 * u.angstrom,  
    "vdw_cutoff": 14.0 * u.angstrom,  
    "units": "sweeps",  
    "steps_per_sweep": 450,  
    "coord_freq": 50,  
    "prop_freq": 10,  
    "properties": thermo_props,  
}
```

Override defaults

Gibbs ensemble MC



$$\begin{aligned}\mu_i^v &= \mu_i^l \\ T^v &= T^l \\ P^v &= P^l\end{aligned}$$

The Gibbs Ensemble Monte Carlo (GEMC) method is a widely used method for computing vapor-liquid equilibria. A distinguishing feature is that two boxes are used to represent each bulk phase. There is no explicit interface.

What not direct coexistence?

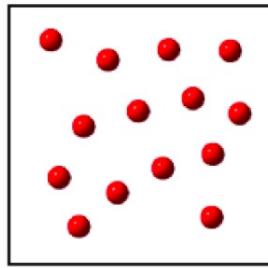
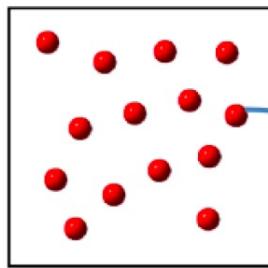
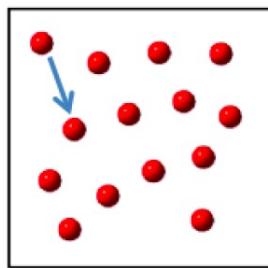
This method was designed in response to the challenge of simulating VLE using systems with explicit interfaces.

Number of particles	% of particles near interface
1,000	49%
64,000	14%
1,000,000	6%

Measuring VLE bulk properties can require large system sizes if we use an interface! We would need to arbitrarily define where each phase is.

GEMC moves

Thermal equilibration
 $T^I = T^{II}$



Chemical potential equilibration
 $\mu_i^I = \mu_i^{II}$

Pressure equilibration
 $P^I = P^{II}$

The acceptance rule for thermal equilibration moves is

$$\frac{acc(o \rightarrow n)}{acc(n \rightarrow o)} = \frac{\alpha_{nm}}{\alpha_{mn}} e^{-\beta \Delta U_A}$$

The acceptance rule for particle transfers is

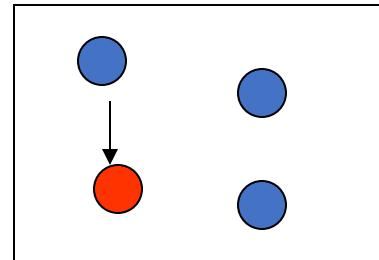
$$\frac{acc(o \rightarrow n)}{acc(n \rightarrow o)} = \frac{\alpha_{nm}}{\alpha_{mn}} e^{-\beta \Delta U_A - \beta \Delta U_B}$$

For volume exchange, we can obtain

$$\frac{acc(o \rightarrow n)}{acc(n \rightarrow o)} = \frac{(V_A + \Delta V)^{N_A} (V_B - \Delta V)^{N_B}}{V_A^{N_A} V_B^{N_B}} e^{-\beta \Delta U_A - \beta \Delta U_B}$$

GCMC Moves

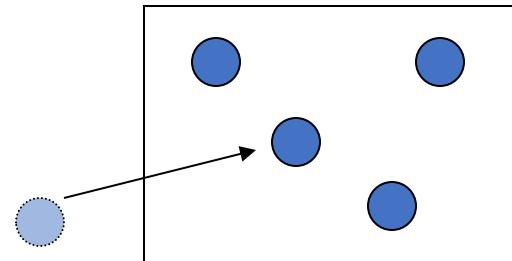
- 1) Thermal equilibration:
acceptance rule same
as in canonical



$$\min\left(1, \frac{\alpha_{nm}}{\alpha_{mn}} \exp[-\beta\Delta V]\right)$$

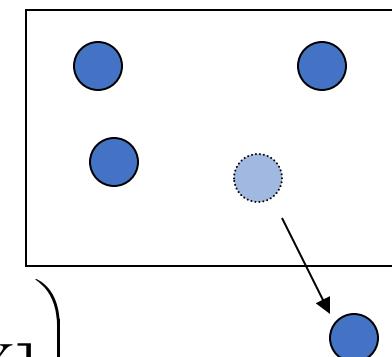
GCMC Moves

- 2) “chemical potential”
 - Insertions $N \rightarrow N+1$



$$P^{ins}_{mn} = \min\left(1, \left[\frac{\alpha_{nm}}{\alpha_{mn}} \right] \frac{\Omega}{Z^{ig}} \frac{f\beta V}{(N_m + 1)} \exp[-\beta\Delta V]\right)$$

- Deletions $N \rightarrow N-1$



$$P^{del}_{nm} = \min\left(1, \left[\frac{\alpha_{mn}}{\alpha_{nm}} \right] \frac{Z^{ig}}{\Omega} \frac{N_m}{f\beta V} \exp[-\beta\Delta V]\right)$$