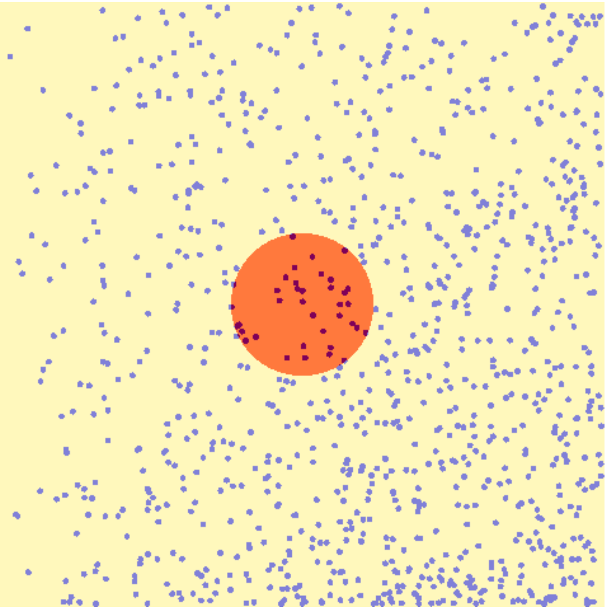
Metropolis–Hastings MC: Nonsymmetric matrix α

$\alpha_{i\to j}$ = matrix of the probability distribution of trial displacements

Metropolis: $\alpha_{i\to j} = \alpha_{j\to i}$

What if $\alpha_{i\to j} \neq \alpha_{j\to i}$?

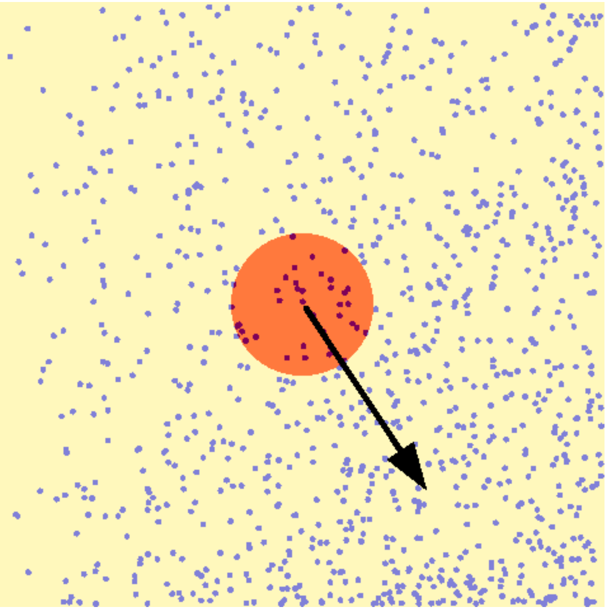


$$W_{i\to j} = \begin{cases} \alpha_{i\to j} & \text{if } \pi_j\alpha_{j\to i} \geq \pi_i\alpha_{i\to j} \\ \alpha_{i\to j}\dfrac{\pi_j\alpha_{j\to i}}{\pi_i\alpha_{i\to j}} & \text{if } \pi_j\alpha_{j\to i} < \pi_i\alpha_{i\to j} \\ 1 - \displaystyle\sum_{k,\,k\neq i} W_{i\to k} & \text{for } i = j \end{cases}$$

$$p_{\text{acc}} = \min\left\{1, \frac{\alpha_{j\to i}}{\alpha_{i\to j}}\exp(-\beta\Delta U)\right\}$$

This extension of the Metropolis algorithm is attributed to W. K. Hastings

Force bias



$$\alpha_{i\to j}^{\text{force bias}} = \alpha_{i\to j}\frac{\exp(\beta\lambda\vec{f}_k\cdot\Delta\vec{r})}{\int\exp(\beta\lambda\vec{f}_k\cdot\Delta\vec{r})\mathrm{d}\Delta\vec{r}}$$

Optimum $\lambda = 0.5$ (between Metropolis and heat-bath)

Similarly:
– torque-bias (rotations)
– virial-bias (volume change)

$\lambda = 1$ & linearization ⇒ heat-bath

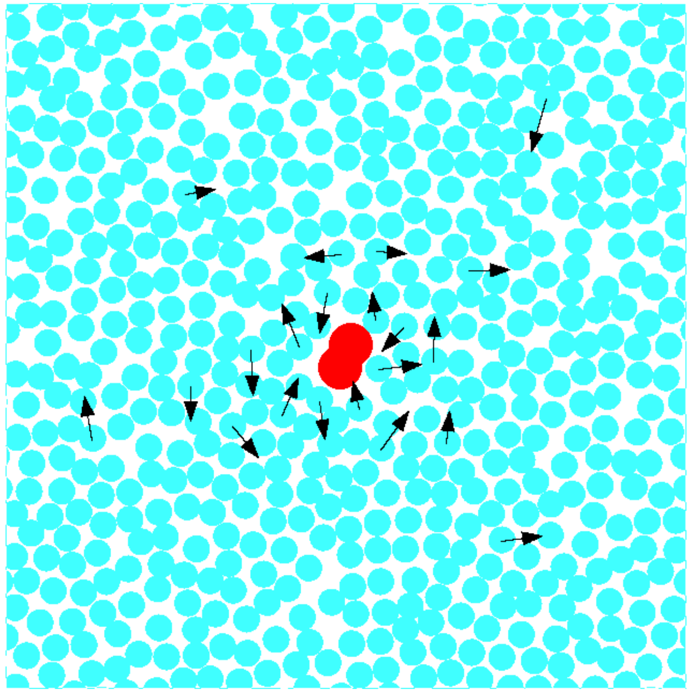
More tricks

- Global density change (close to the critical point)
- Tesselation to clusters and cluster moves
- *NPT* of hard bodies: molecules → clusters, swell//shrink wrt cluster centers
- Swapping particles/molecules/groups/clusters

Preferential sampling

Particles in the more interesting areas are moved more often.

Example: solvent molecules around a solute: $p_{\mathrm{pref}}(r) \overset{\text{e.g.}}{=} 1/(1+r^2/\sigma^2)$



Version 1:

- Choose a solvent molecule, i, at random
- If $u_{[0,1)} < p_{\mathrm{pref}}(|\vec r_i - \vec r_0|)$ ($\vec r_0$ = solute)
  - Generate a trial configuration $\vec r_i^{\,\mathrm{tr}}$
  - Accept it with $p_{\mathrm{acc}} = \min\left\{1, \frac{p_{\mathrm{pref}}(|\vec r_i^{\,\mathrm{tr}} - \vec r_0|)}{p_{\mathrm{pref}}(|\vec r_i - \vec r_0|)} \exp(-\beta \Delta U)\right\}$
    (otherwise continue with the old configuration)

$$\alpha_{\mathrm{pref}}(A \to A^{\mathrm{tr}}) = \alpha(A \to A^{\mathrm{tr}}) \frac{p_{\mathrm{pref}}(|\vec r_i - \vec r_0|)}{N}$$

Preferential sampling

Version 2:

- Choose a solvent molecule, i, from the distribution $p_{\text{pref}}(|\vec{r}_i - \vec{r}_0|)$, $i = 1, \ldots, N$

  An optimum algorithm with binary search has cost $\propto \log N$. Easy one:
  - select solvent i randomly
  - if not $u_{[0,1)} < p_{\text{pref}}(|\vec{r}_i - \vec{r}_0|)$, new selection
- Perform one MC step using the molecule:
  - generate a trial configuration $\vec{r}_i^{\,\text{tr}}$
  - accept it with probability

$$p_{\text{acc}} = \min\left\{1, \frac{p_{\text{pref}}(|\vec{r}_i^{\,\text{tr}} - \vec{r}_0|)/S^{\text{tr}}}{p_{\text{pref}}(|\vec{r}_i - \vec{r}_0|)/S} \exp(-\beta\Delta U)\right\}, \quad S = \sum_{i=1}^{N} p_{\text{pref}}(|\vec{r}_i - \vec{r}_0|)$$

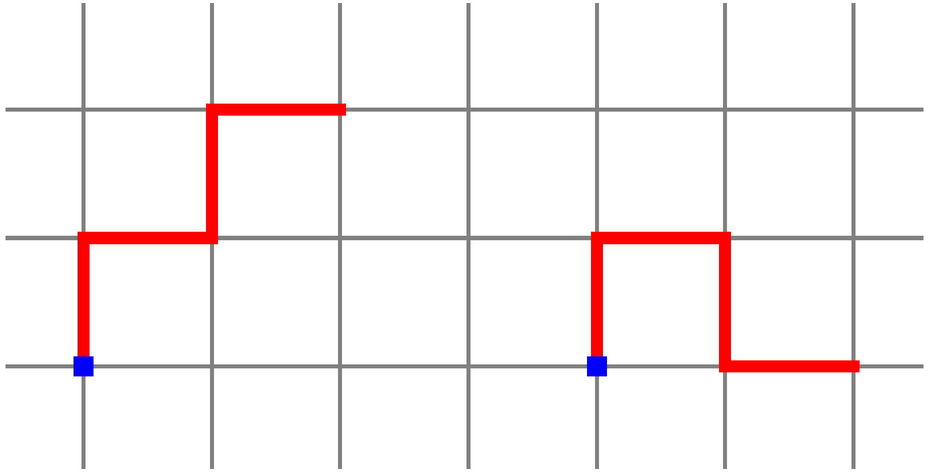
$$\alpha_{\text{pref}}(A \to A^{\text{tr}}) = \alpha(A \to A^{\text{tr}}) \frac{p_{\text{pref}}(|\vec{r}_i - \vec{r}_0|)}{S}$$

Both versions are of the same efficiency if well optimized

Polymers and Rosenbluth sampling

A polymer in an athermal (very good) solvent is modeled by a self-avoiding random walk:

$$p = \frac{1}{4 \times 3 \times 3 \times 3}$$



$$p = \frac{1}{4 \times 3 \times 3 \times 2}$$

Both polymer should have the same weight!

Remedy: Rosenbluth weight (factor) of a step = number of possible continuations = R_i, the weight of a configuration obtained by the walk is:

$$R = \frac{1}{p} = \prod_{i=1}^{N} R_i$$

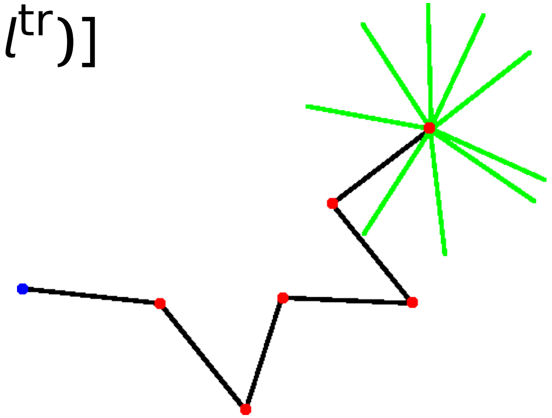
Generalization (the continuation is selected ∝ Boltzmann factor):

$$R_i = \sum_{l=1}^{k} \exp[-\beta U(i \to l)]$$

Generalization for continuous models

- Instead of k possible direction on a lattice, k random trial configurations are generated
- The new configuration is drawn with the probability given by

$$p(i \to l^{\mathrm{tr}}) = \frac{\exp[-\beta U(i \to l^{\mathrm{tr}})]}{R_i}, \qquad R_i = \sum_{l=1}^{k} \exp[-\beta U(i \to l^{\mathrm{tr}})]$$



The configuration weight is: $R = \prod_{i=1}^{N} R_i$

Limiting cases:

- For $k = 1$, the configuration weight is $R = \exp[-\beta \sum U(i \to l^{\mathrm{tr}})] = \exp(-\beta U^{\mathrm{tr}}_{\mathrm{total}})$ = Boltzmann probability of a polymer. This is the naive MC integration.
- For $k \to \infty$, the method is close to the previous lattice-based method; i.e., drawing the new configuration with the Boltzmann probability at every step. (NB: this is not the importance sampling because the Rosenbluth weights differ.)

Configurational bias Monte Carlo

Instead of one trial configuration in the Metropolis method, let us consider k of them (continuous models: random sample | lattice models: all possible). A trial configurations is drawn from the probability distribution:

$$p(\vec{r}_l^{\,\mathrm{tr}}) = \frac{\exp[-\beta U(\vec{r}_l^{\,\mathrm{tr}})]}{R_{\mathrm{tr}}}, \qquad R_{\mathrm{tr}} = \sum_{l=1}^{k} \exp[-\beta U(\vec{r}_l^{\,\mathrm{tr}})]$$

It is accepted with probability $\min\left\{1, \frac{R_{\mathrm{tr}}}{R_{\mathrm{old}}}\right\}$, where

$$R_{\mathrm{old}} = \exp[-\beta U(\vec{r}^{\,\mathrm{old}})] + \sum_{l=2}^{k} \exp[-\beta U(\vec{r}_l^{\,\mathrm{old,tr}})] \qquad R_{\mathrm{old}} = \sum_{l=1}^{k} \exp[-\beta U(\vec{r}_l^{\,\mathrm{old}})]$$

($\vec{r}_l^{\,\mathrm{old,tr}}$ = random configuration generated in the same way)
NB: for lattices, it may happen $R_{\mathrm{old}} = R_{\mathrm{tr}}$, which is the heat bath method

Stochastic matrix

$$W_{i\to j} = \begin{cases} \alpha_{i\to j} \dfrac{\exp[-\beta U(\vec{r}_j)]}{R_j} & \text{for } R_j > R_i \\[2ex] \alpha_{i\to j} \dfrac{\exp[-\beta U(\vec{r}_j)]}{R_j} \dfrac{R_j}{R_i} & \text{for } R_j < R_i \end{cases}$$

Simulations of molecular systems

- Small molecules may be rigid (nitrogen, water, methane)
- Large molecules must be flexible . . . except (some) bonds

Vibrating (classical) bonds:

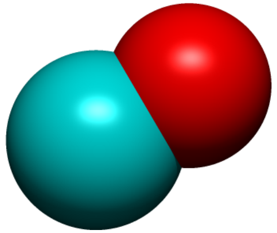
- ⊕ simplicity and consistency of the model
- ⊕ code simplicity
- ⊕ more realistic description of flexibility

- ⊖ technical problems with too stiff springs (short timestep in v MD, short trial moves in MC)
- ⊖ vibrational frequencies (esp. for hydrogens) are so high that cannot be treated by classical mechanics anyway
- ⊖ transfer of energy between the fast vibrations and slow degrees of freedom is slow (they are decoupled), unless a stochastic thermostat is used
- ⊖ flexible models are more complicated theoretically

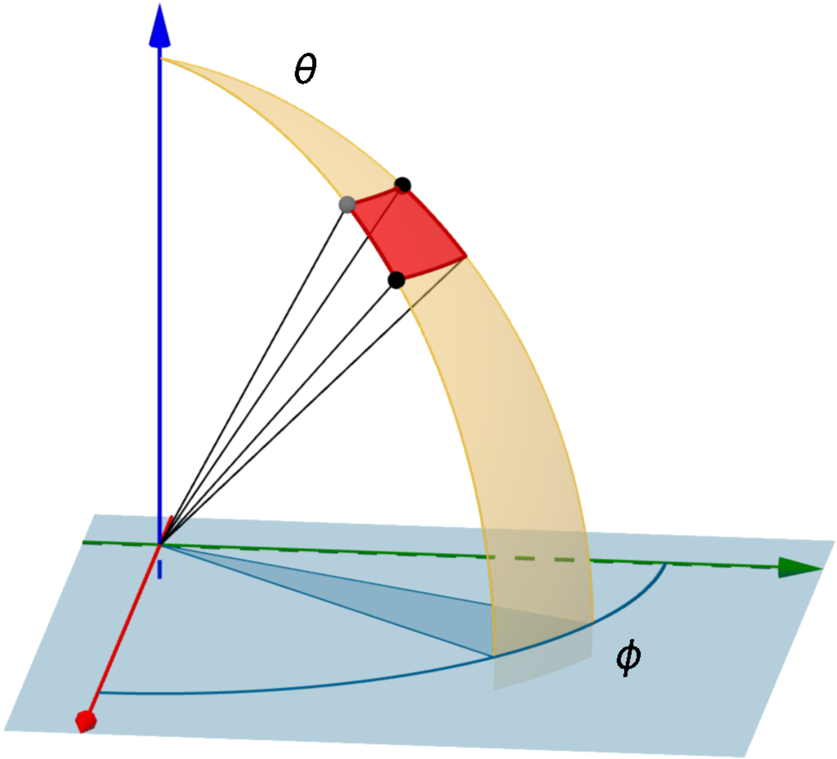
Fixed (constrained) bonds: just opposite

MC: molecules

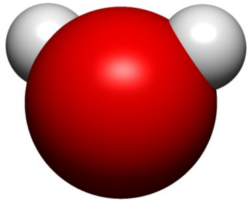
Example of a wrong algorithm for a linear molecule with axis (θ, ϕ):



$$\begin{aligned} \theta^{\mathrm{tr}} &= \theta + \Delta\theta u_{[-1,1]} \\ \phi^{\mathrm{tr}} &= \phi + \Delta\phi u_{[-1,1]} \end{aligned}$$



Example of a correct algorithm for a general body:



- choose an axis randomly:
  - – any of $\hat{x}, \hat{y}, \hat{z}$ in the body frame
  - – any of $\hat{x}, \hat{y}, \hat{z}$ Cartesian coordinates in 3D space
  - – any random vector
- rotate by angle $\Delta\alpha u_{[-1,1]}$, where $u_{[-1,1]}$ is a random number uniformly distributed in $[-1, 1]$

Rotation by $\Delta\alpha$ around $\hat{z}$-axis:

$$\begin{pmatrix} \cos\Delta\alpha & -\sin\Delta\alpha & 0 \\ \sin\Delta\alpha & \cos\Delta\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Quaternions

Quaternion:

$$q = w + xi + yj + zk$$

$$i^2 = j^2 = k^2 = -1,\ ij = -ji = k,\ ijk = -1 \text{ (a cykl.)}$$

$$|q|^2 = w^2 + x^2 + y^2 + z^2$$

Set $\{|q| = 1, q \simeq -q\}$ is isomorphic with $SO(3)$ (group of rotations), which can be used in MC, kinematics, and dynamics

Matrix of rotation:

$$\Omega = \begin{pmatrix} w^2+x^2-y^2-z^2 & 2xy-2zw & 2zx+2yw \\ 2xy+2zw & w^2-x^2+y^2-z^2 & 2yz-2xw \\ 2zx-2yw & 2yz+2xw & w^2-x^2-y^2-z^2 \end{pmatrix}$$

There are 4 algebras (above $\mathbb{R}$ or a field) with division, which are a vector space with a norm so that $|xy| = |x||y|$: real numbers, complex numbers, quaternions (multiplication is not commutative) and octonions (Cayley algebra, multiplication is not commutative nor associative).
Field is something with operations $+ - * /$ with the same structure as $\mathbb{R}$.
Algebra is a vector space over a field (of “scalars”) with “$\cdot$”, where $(x+y)\cdot z = x\cdot z + y\cdot z$, $z\cdot(x+y) = z\cdot x + z\cdot y$, $(ax)\cdot(by) = (ab)(x\cdot y)$ for vectors x, y, z and scalars a, b.

MC: Molecules with internal degrees of freedom

- Cartesian coordinates → generalized coordinates (Jacobian is needed – difficult)
- Cartesian coordinates → orthogonal coordinates (Jacobian does not change while moving along one coordinate)
- Frozen degrees of freedom (bonds)

Polymers:

- standard moves
- crankshaft move
- reptation for linear homopolymers: a tail bead is cut off and moved at the head
  – one random new position = standard Metropolis
  – more trial moves = configurational bias MC
- various enhance sampling methods:
  umbrella sampling, parallel tempering, Wang–Landau

http://www.youtube.com/watch?v=rCTSG-SrShk&feature=related

MD: bonds

Integration of the equations of motion for systems with fixed bond lengths or angles is not the same as the infinitely-large force constant limit of the corresponding flexible system.

What to fix:

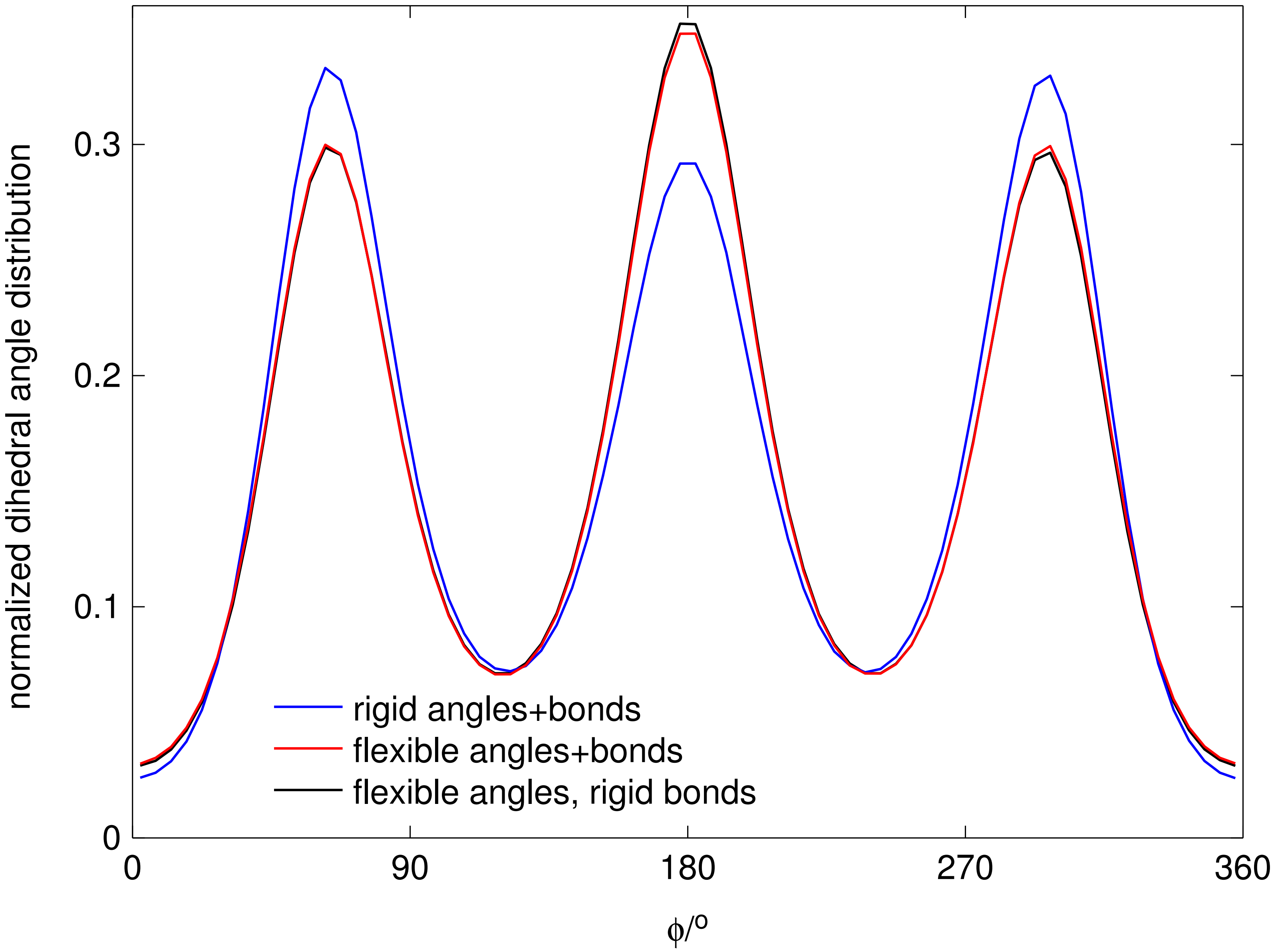
- nothing – short timestep, bad equipartition (convergence of different degrees of freedom)
- bonds with hydrogen only – longer timestep allowed, better (but not the best) equipartition
  GROMACS: special algorithm for H, $h = 4$ ps
- all bond lengths – larger systematic errors, not good for rigid geometries (fullerene) and heavy atoms
- all bond lengths + angles with hydrogens – cheap and less precise, but longer timestep allowed + good equipartition
- all bond lengths + all angles – WRONG except small molecules

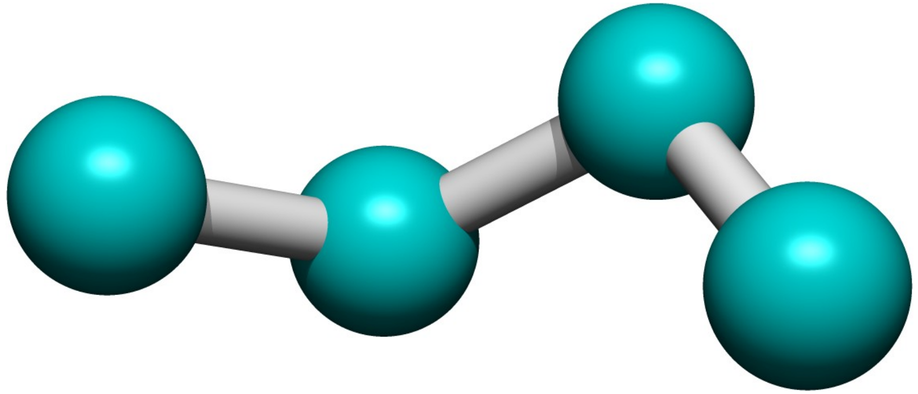
Methods:

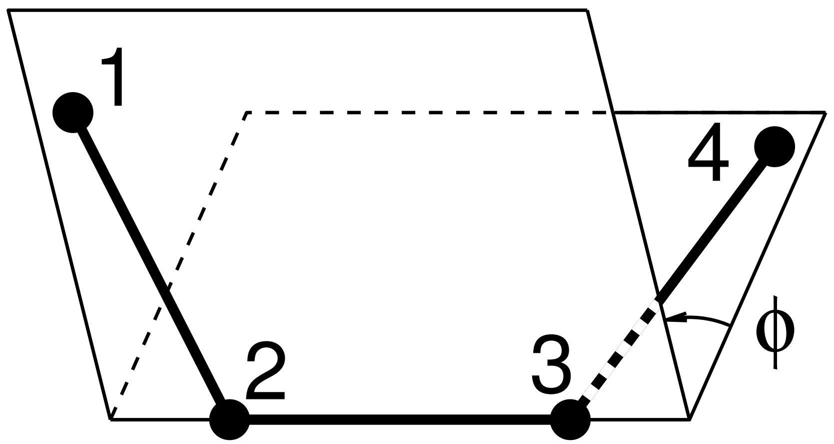
- SHAKE (+Verlet)
- Lagrangian constraint dynamics

Dihedral angle distribution of butane

United-atom model (CHARMM19) of butane

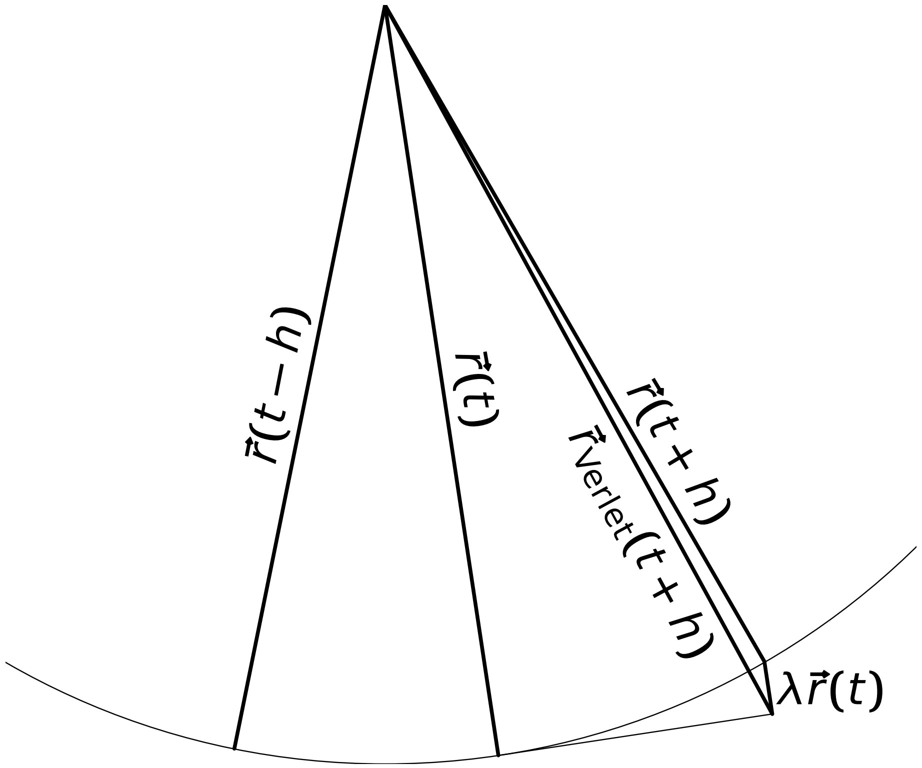






Fixed constraints in MD: SHAKE

Matematical pendulum:



$$\vec{r}(t+h) = \vec{r}_{\text{Verlet}}(t+h) - \frac{h^2}{m}\vec{f}_c(t)$$

$$= 2\vec{r}(t) - \vec{r}(t-h) + h^2 \frac{\vec{f}(t) - \vec{f}_c(t)}{m}$$

$$\frac{h^2\vec{f}_c(t)}{m} = \lambda\vec{r}(t)$$

$$|\vec{r}(t+h)|^2 = |\vec{r}(t)|^2 = l^2$$

$$[\vec{r}_{\text{Verlet}}(t+h) - \lambda\vec{r}(t)]^2 = \vec{r}(t)^2$$

$$\vec{r}_{\text{Verlet}}(t+h)^2 - 2\lambda\vec{r}_{\text{Verlet}}(t+h)\cdot\vec{r}(t) + \lambda^2\vec{r}(t)^2 = \vec{r}(t)^2$$

$$\lambda \approx \frac{|\vec{r}_{\text{Verlet}}(t+h)|^2 - |\vec{r}(t)|^2}{2\vec{r}_{\text{Verlet}}(t+h)\cdot\vec{r}(t)} = \frac{|\vec{r}_{\text{Verlet}}(t+h)|^2 - l^2}{2\vec{r}_{\text{Verlet}}(t+h)\cdot\vec{r}(t)}$$

SHAKE

General atom–atom bond:

$$\vec r_i(t+h) = \vec r_{\mathrm{Verlet},i}(t+h) + \lambda \frac{1/m_i}{1/m_i + 1/m_j} \vec r_{ij}$$

$$\vec r_j(t+h) = \vec r_{\mathrm{Verlet},j}(t+h) - \lambda \frac{1/m_j}{1/m_i + 1/m_j} \vec r_{ij}$$

where

$$\lambda = \frac{|\vec r_{\mathrm{Verlet},ij}(t+h)|^2 - \overset{l^2}{|\vec r_{ij}(t)|^2}}{2\vec r_{\mathrm{Verlet},ij}(t+h)\cdot \vec r_{ij}(t)}$$

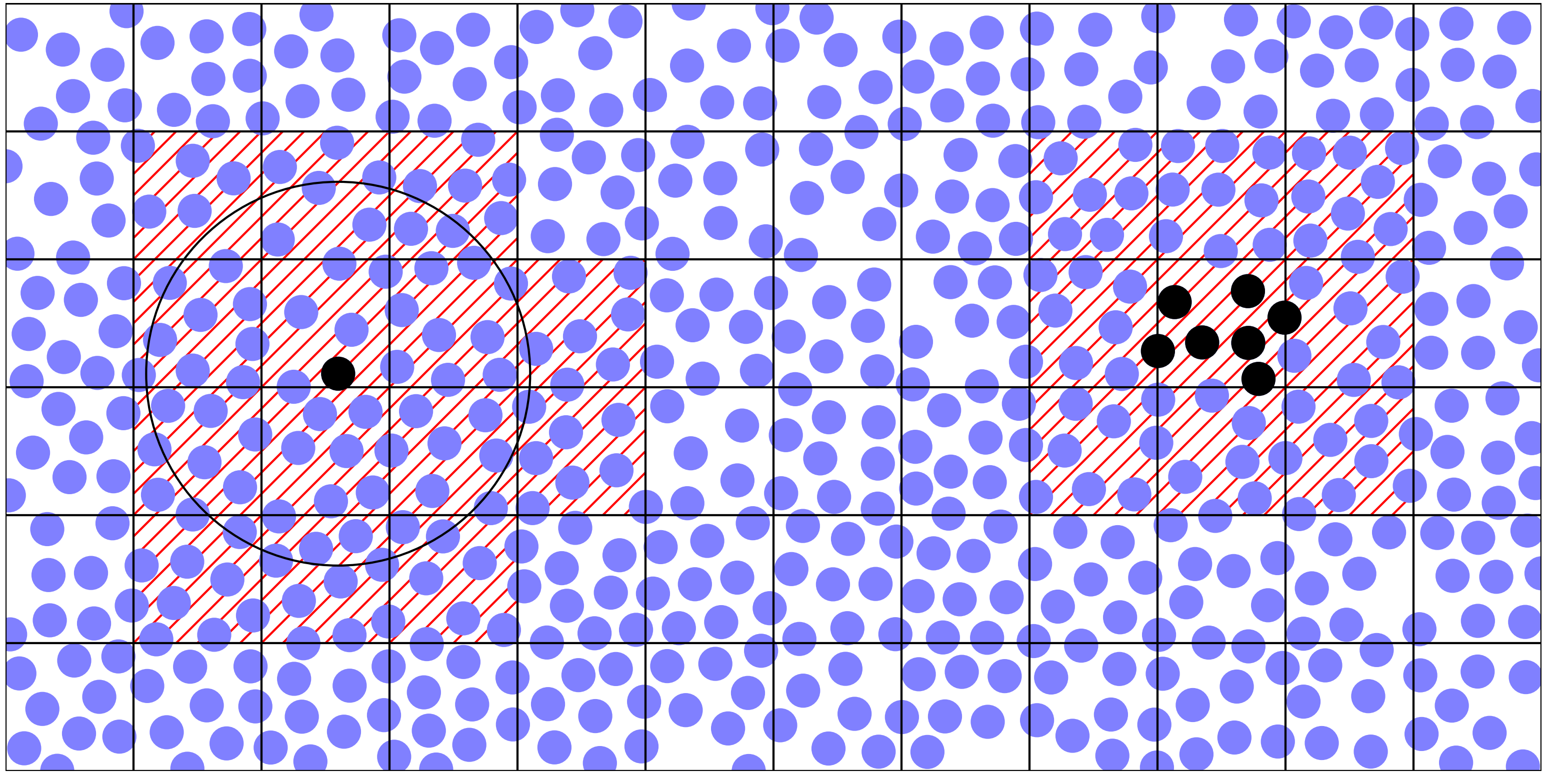
$\vec r_{ij} = \vec r_j - \vec r_i,\ r_{ij} = |\vec r_{ij}|$

- Based on the Verlet method
- Center-of-mass is conserved (integral of motion)!
- Complex molecules: repeat iteratively until converged
- Speed up: superrelaxation
- Velocity version: RATTLE, more variants
- For $m_i = \infty$, $\vec r_i = (0,0,0)$, $\vec r = \vec r_j = \vec r_{ij}$ ⇒ mathematical pendulum

Optimization I

For short-ranged pair potentials (also r-space Ewald):

- all pairs ($N < 300$)
- neighbor list ($N \approx 1000$)
- domain-decomposition algorithms: linked-cell list method



Optimization II

A couple of tricks:

- MD: multiple timestep MD
- MC: multimoves (near the critical point – critical slowing-down)
- MC: identity change, non-Boltzmann sampling, . . .
- hybrid MC/MD (not very good)

Programming tricks:

- cache
- nearest neighbors in periodic boundary conditions
- tables: pair potential calculated by splines

Parallel code:

- usually based on domain decomposition (linked-cell list)
- standard computers 4–32 cores
- Graphics Processing Units: thousands of processors, more difficult to program