Scientific Computing (Phys 2109/Ast 3100H)
II. Numerical Tools for Physical Scientists

SciNet HPC Consortium University of Toronto

Lecture 12: Molecular Dynamics

Winter 2013

SciNet compute • calcul CANADA

Lessons from HW from lecture 1

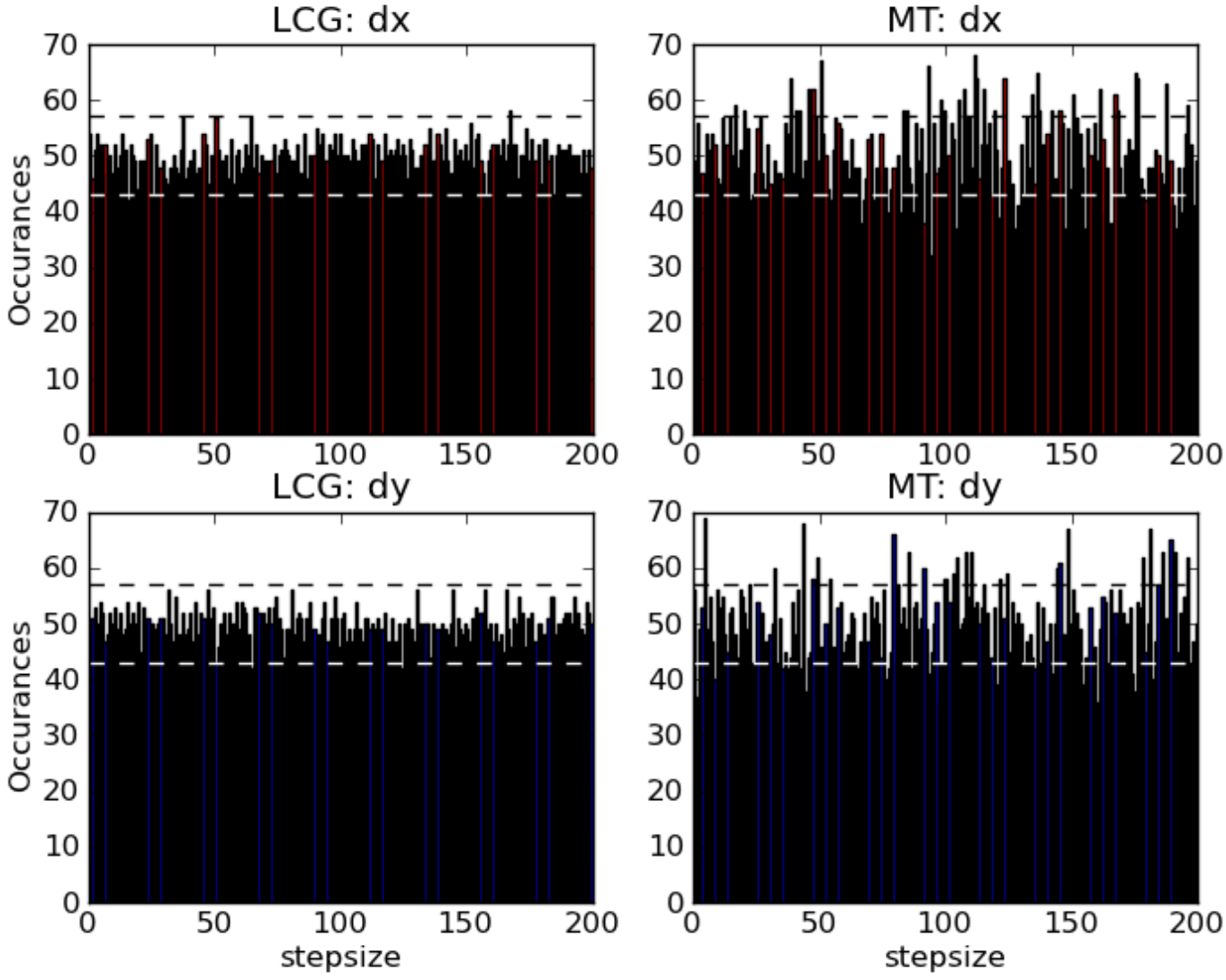
Floating point sums

Forward sum	=	1
Backward sum	=	1.25
Pairwise sum	=	2
Pairwise sum (sorted)	=	2
Forward sum error	=	1
Backward sum error	=	0.75
Pairwise sum (unsorted) error	=	0
Pairwise sum (sorted) error	=	0

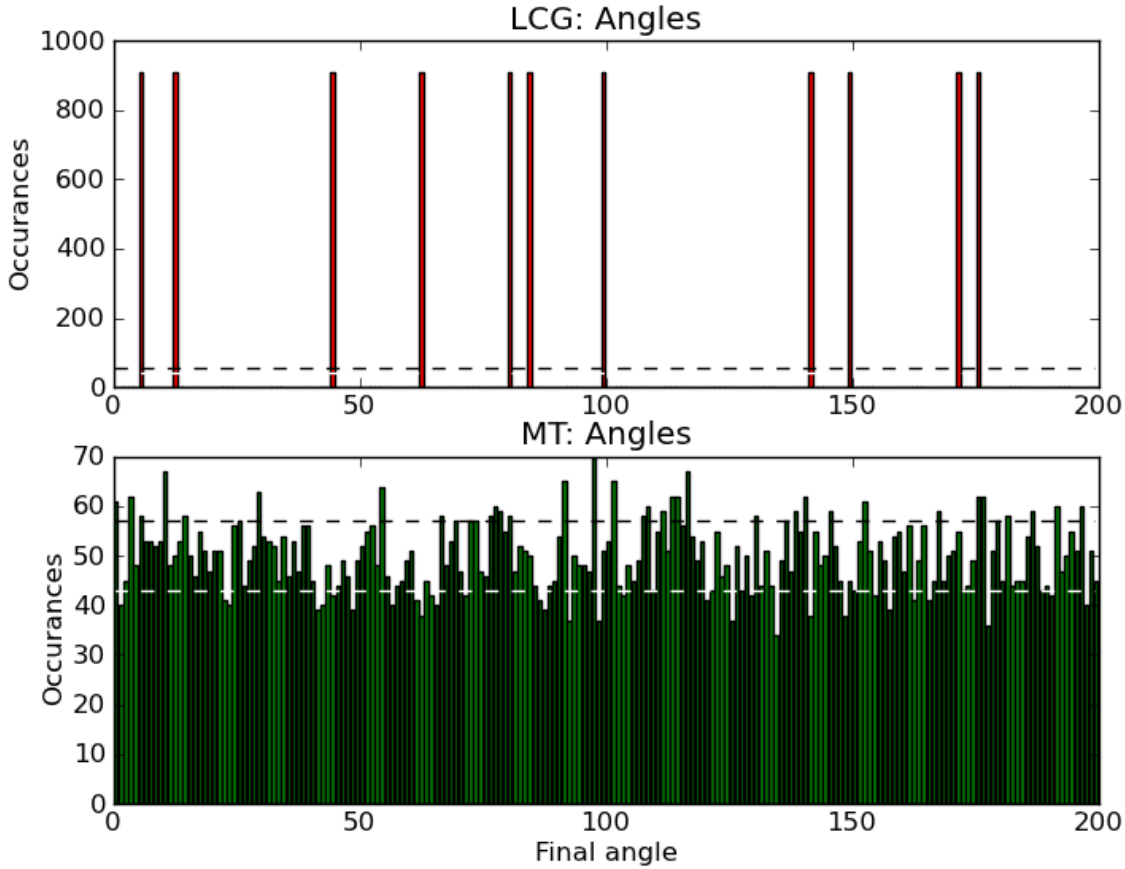
- Doing the summation forward just results in 1; at the first step, 1 + 1.e-8 = 1 + (something less than machine epsilon) = 1, and all 1e8 following steps then play out the same way.
- Even doing the sum backwards doesn't help; as soon as you add up enough 1e-8's to sum up to (1.e-8/machine epsilon), which is about 1/4, the following 1.e-8s don't contribute to sum, then you get the final 1.

Lessons from HW from lecture 1

- dx/dy histograms seems reasonable, although the variance in the LCG case seems somewhat less than expected.



Distribution of angles



- Consecutive numbers out of the LCG are very strongly correlated, leading to just a handful of final angles picked out
- Moral of story – don't make up your own RNG. Even if simple statistics look reasonable, could get bitten.

SciNet
compute • calcul
CANADA

Lecture 12

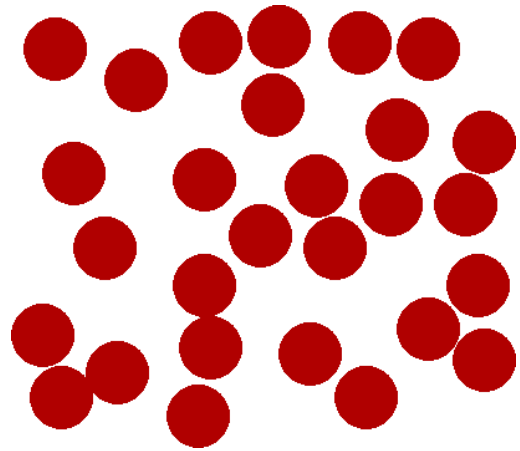
SciNet compute • calcul CANADA

Molecular Dynamics Simulations

N interacting particles

$$\mathbf{m_i \ddot{r}_i} = \mathbf{f_i}(\{\mathbf{r}\}, \{\dot{\mathbf{r}}_j\}, \mathbf{t})$$

+ initial conditions



What makes this different from other ODEs?

- Hamiltonian dynamics
- Very expensive evaluation of **f** if **N** is large

For both, we will only touch upon some aspects.

Note that **N**-body simulation fall within this class as well; the numerics does not case whether the particles are molecules or stars.

Hamiltonian dynamics

- Molecular Dynamics aims to compute *equilibrium*, *dynamical* and *transport* properties of *classical many body systems*.
- Many classical systems have Newtonian equations of motion:

$$\dot{\mathbf{r}} = \frac{\mathbf{1}}{\mathbf{m}}\mathbf{p} \qquad\qquad \dot{\mathbf{p}} = \mathbf{F} = -\frac{\mathbf{dU}}{\mathbf{dr}},$$

  or $\dot{\mathbf{x}} = \mathbf{Lx}$, with $\mathbf{LA} = \{\mathbf{A}, \mathbf{H}\}$, where $\mathbf{x} = (\mathbf{r}, \mathbf{p})$.
- Energy $\mathbf{H} = \frac{|\mathbf{p}|^2}{\mathbf{2m}} + \mathbf{U(r)}$ is conserved under the dynamics.
- Potential energy is typically a sum of pair potentials:

$$\mathbf{U(r)} = \sum_{\mathbf{(i,j)}} \varphi(\mathbf{r_{ij}}) = \sum_{\mathbf{i=1}}^{\mathbf{N}} \sum_{\mathbf{j=1}}^{\mathbf{i-1}} \varphi(\mathbf{r_{ij}}),$$

  which entails the following expression for the forces $\mathbf{F}$:

$$\mathbf{F_i} = -\sum_{\mathbf{j \neq i}} \frac{\mathbf{d}}{\mathbf{dr_i}} \varphi(\mathbf{r_{ij}}) = \sum_{\mathbf{j \neq i}} \underbrace{\varphi'(\mathbf{r_{ij}}) \frac{\mathbf{r_j - r_i}}{\mathbf{r_{ij}}}}_{\mathbf{F_{ij}}}$$

Hamiltonian dynamics as disguished importance sampling

- If the system is ergodic then time average equals the microcanonical average:

$$\lim_{t_{final} \to \infty} \frac{1}{t_{final}} \int_0^{t_{final}} dt\, A(x(t)) = \frac{\int dx\, A(x)\, \delta(E - H(x))}{\int dx\, \delta(E - H(x))}.$$

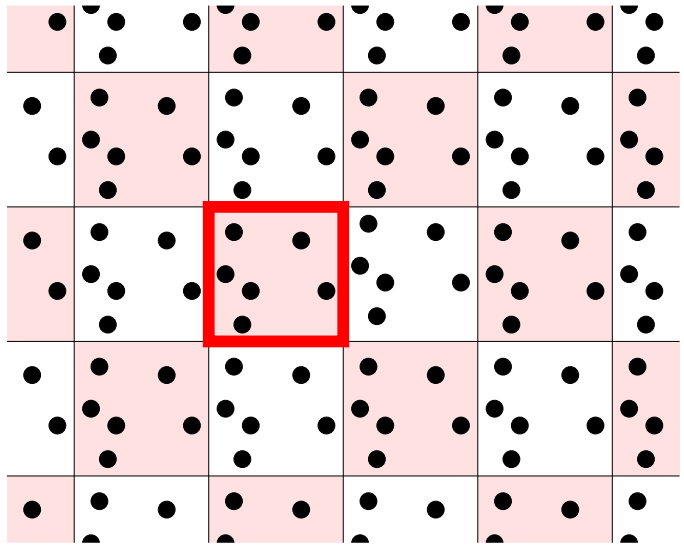
- For large **N**, microcanonical and canonical averages are equal for many quantities **A**.

- Need long times t_{final}!

SciNet compute • calcul CANADA

Boundary conditions

- When simulating finite systems, a wall potential would give finite size effects and destroy translation invariance.
- More benign: *Periodic Boundary Conditions*
- All particles in box have coordinates between $-\mathbf{L}/\mathbf{2}$ and $\mathbf{L}/\mathbf{2}$.
- A particle exiting simulation box is put back at the other end.



- The box with thick red boundaries is our simulation box.
- Other boxes are copies, or "periodic images"
- The other squares contain particles with shifted positions
- "Flat torus"

Force calculations

- A common pair potential is the Lennard-Jones potential

$$\varphi(\mathbf{r}) = 4\varepsilon \left[\left(\frac{\sigma}{\mathbf{r}} \right)^{12} - \left(\frac{\sigma}{\mathbf{r}} \right)^{6} \right],$$

  - σ is a measure of the range of the potential.
  - ε is its strength.
  - The potential is positive for small $\mathbf{r}$: repulsion.
  - The potential is negative for large $\mathbf{r}$: attraction.
  - The potential goes to zero for large $\mathbf{r}$: short-range.
  - The potential has a minimum of $-\varepsilon$ at $\mathbf{2^{1/6}\sigma}$.
- Computing all forces in an N-body system requires the computation of $\mathbf{N(N-1)/2}$ forces $\mathbf{F_{ij}}$
- Force Computation often the most demanding part of MD.
- Avoid infinite sums: modify the potential such that it becomes zero beyond a certain *cut-off* distance $\mathbf{r_c}$:

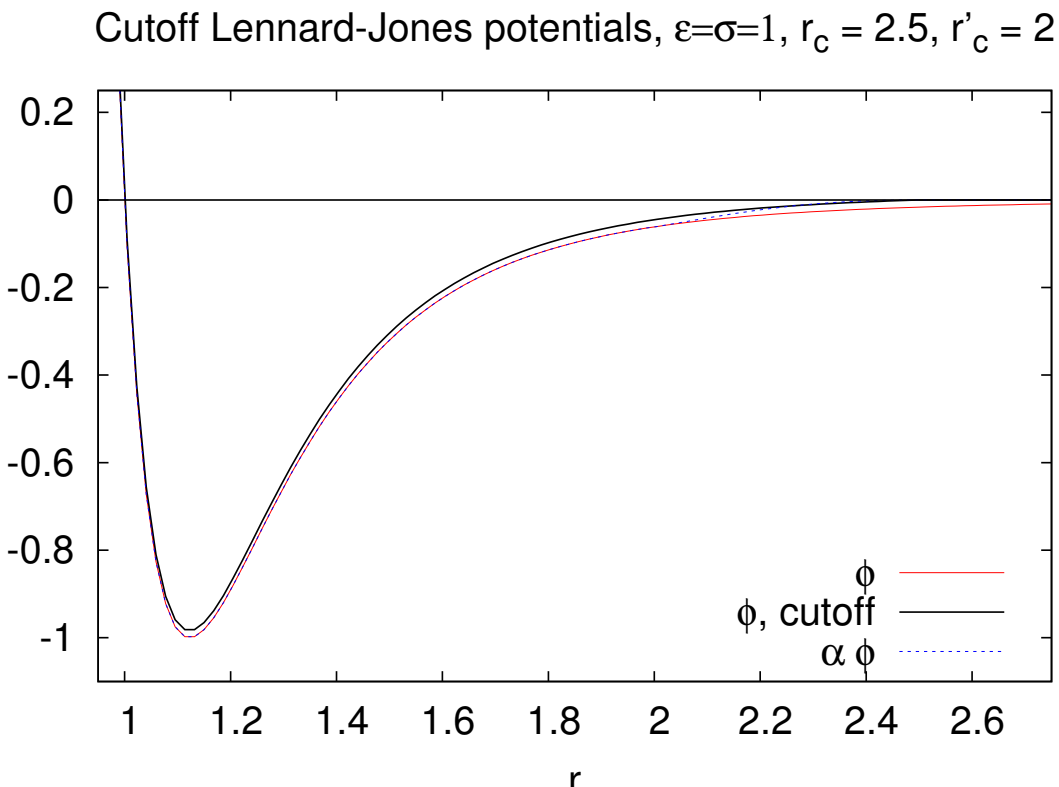
$$\varphi'(\mathbf{r}) = \begin{cases} \varphi(\mathbf{r}) - \varphi(\mathbf{r_c}) & \text{if } \mathbf{r} < \mathbf{r_c} \\ \mathbf{0} & \text{if } \mathbf{r} \geq \mathbf{r_c} \end{cases}$$

- To also avoid discontinuities in derivatives, one can use a schemes such as

$$\varphi''(r) = \alpha(r)\varphi(r) \tag{1}$$

where

$$\alpha(r) = \begin{cases} 1 & r < r_c' \\ \frac{(r_c - r)^2(r_c - 3r_c' + 2r)}{(r_c - r_c')^3} & r_c' \le r \le r_c \\ 0 & r > r_c \end{cases} . \tag{2}$$



Streamlining the force evaluation

Cell divisions

- Divide the simulation box into cells larger than the cutoff $\mathbf{r_c}$.
- Make a list of all particles in each cell.
- In the sum over pairs in the force computation, only sum pairs of particles in the same cell or in adjacent cells.

Streamlining the force evaluation

Neighbour lists (also called Verlet lists)

- Make a list of pairs of particles that are closer than $\mathbf{r_c} + \boldsymbol{\delta}\mathbf{r}$.
- Sum over the list of pairs to compute the forces.
- The neighbour list are to be used in subsequent force calculations as long as the list is still valid.
- Invalidation criterion: a particle has moved more than $\boldsymbol{\delta}\mathbf{r}/\mathbf{2}$.
- Therefore, before a new force computation, check if any particle has moved more than $\boldsymbol{\delta}\mathbf{r}/\mathbf{2}$ since the last list-building. If so, rebuild the Verlet list, otherwise use the old one.

For large systems, turns $\mathbf{N^2}$ into $\mathbf{N}$.

Force calculations

Long range interaction

Electrostatics, gravity, are example of long range interactions that cannot be cut off without seriously altering the physics.
Requires special techniques such as

- Barnes-Hut
- Particle Mesh Ewald

Desirable qualities for a molecular dynamics integrator

- *Accuracy*
- *Efficiency*
- *Stability*
- *Respect physical laws:*
  - Time reversal symmetry
  - Conservation of energy
  - Conservation of linear momentum
  - Conservation of angular momentum
  - Conservation of phase space volume

The most efficient algorithm is then the one that allows the largest possible time step for a given level of accuracy, *while maintaining stability and preserving conservation laws*.

Symplectic integrators

Momentum Verlet Scheme (first version)

$$\mathbf{r_{n+1}} = \mathbf{r_n} + \frac{\mathbf{p_n}}{\mathbf{m}}\mathbf{h} + \frac{\mathbf{F_n}}{\mathbf{2m}}\mathbf{h^2}$$

$$\mathbf{p_{n+1}} = \mathbf{p_n} + \frac{\mathbf{F_{n+1}} + \mathbf{F_n}}{\mathbf{2}}\mathbf{h}$$

The momentum rule appears to pose a problem since $\mathbf{F_{n+1}}$ is required. But to compute $\mathbf{F_{n+1}}$, we need only $\mathbf{r_{n+1}}$, which is computed in the integration step as well.
Equivalent to position Verlet scheme.

Symplectic integrators

Momentum Verlet Scheme (second version)

The extra storage step can be avoided by introducing the half step momenta as intermediates:

$$\mathbf{p_{n+1/2}} = \mathbf{p_n} + \frac{1}{2}\mathbf{F_n}\mathbf{h}$$

$$\mathbf{r_{n+1}} = \mathbf{r_n} + \frac{\mathbf{p_{n+1/2}}}{\mathbf{m}}\mathbf{h}$$

$$\mathbf{p_{n+1}} = \mathbf{p_{n+1/2}} + \frac{1}{2}\mathbf{F_{n+1}}\mathbf{h}$$

Also nice and symmetric:

1. Half momentum step
2. Full position step
3. Half momentum step

First step the same as the last (with updated F).

Symplectic integrators from Hamiltonian splitting methods

- For sampling, one wants a long trajectory (formally $\mathbf{t_f} \longrightarrow \infty$).
- It is therefore important that an integration algorithm be stable.
- The momentum Verlet scheme, on the other hand, is much more stable than, say, the Euler scheme.
- To see why, one should re-derive the momentum Verlet scheme from a completely different starting point, using a so-called *Hamiltonian splitting method* (also known as *Geometric integration*).

Symplectic integrators from Hamiltonian splitting methods

Very, very briefly:

- Any Hamiltonian $\mathbf{H}$ a flow on phase space: $\mathbf{U(t)}$
- Split up Hamiltonian in $\mathbf{k}$ parts, $\mathbf{H_1 \ldots H_k}$.
- Gives $\mathbf{k}$ flows: $\mathbf{U_1 \ldots U_K}$.
- Baker-Campbell-Hausdorff formula gives approximate factorization, e.g.
  $\mathbf{H = H_1 + H_2 \Rightarrow U(h) \approx U_1(h/2)U_2(h)U_1(h/2)}$
- Symmetric form reduced order and preserves time reversibility.
- This is momentum Verlet!
- Further using BCH, one can derive a shadow Hamiltonian.
- $\Rightarrow$ simulated system retains all hamiltonian properties.

Where are the md libraries?

There typically aren't any. MD packages are usually applications with a lot of parameters, that used other libraries. Examples:

- Gromacs
- NAMD
- LAMMPS

which all differ in intended usages, available force fields, serial speed (platform dependent), parallel scalability, etc.

Frameworks

Some MD packages come more as frameworks, which could be used as a library, within e.g. a C++ program. OpenMM out of Stanford is a prime example which is still actively maintained (and in fact used in the GPU implementation of Gromacs). https://simtk.org/home/openmm

Homework

Part 1

Compute numerically:

$$\int_0^3 f(x)\,dx$$

with

$$f(x) = \ln(x)\sin(x)e^{-x},$$

using three different methods:

1. Extended Simpsons' rule
2. Gauss-Legendre quadrature
3. Monte Carlo sampling

Compare the convergence of these methods by increasing number of function evaluations.
Hint: what is f(0)?

Part 2

Using an adaptive 4th order Runge-Kutta approach, with a relative accuracy of 1e-4, compute the solution for $\mathbf{t = [0, 100]}$ of the following set of coupled ODE(Lorenz oscillator)

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= (\rho - z)x - y \\ \frac{dz}{dt} &= xy - \beta z \end{aligned}$$

with $\sigma = 10, \beta = 8/3, \rho = 28$, and with initial conditions

$$\begin{aligned} x(0) &= 10 \\ y(0) &= 20 \\ z(0) &= 30 \end{aligned}$$

Plot the result.

Hint: study the GSL documentation.

SciNet compute • calcul CANADA