Making the Atoms Dance!

Where do molecular dynamics simulation algorithms come from?

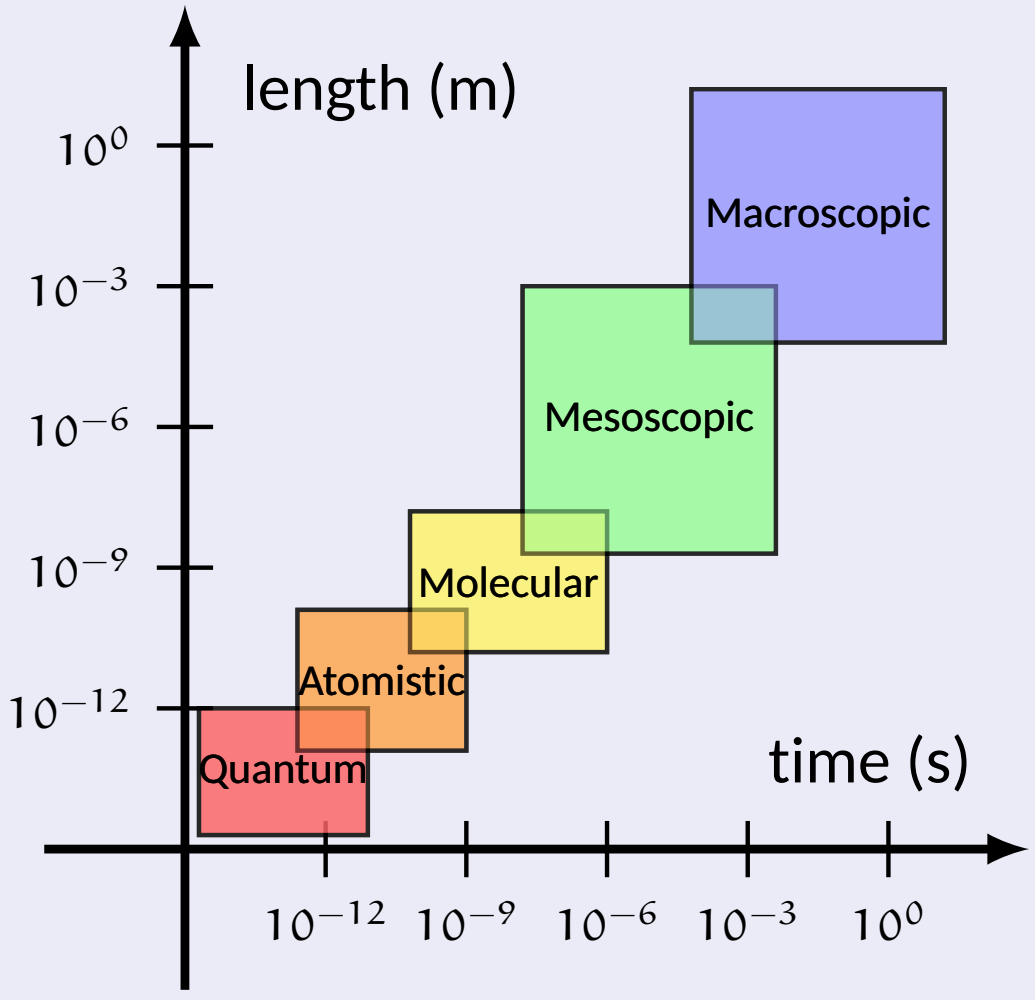
M P Allen

Department of Physics
University of Warwick
University of Bristol

Molecular Simulation

What Is It?

Time and Length Scales



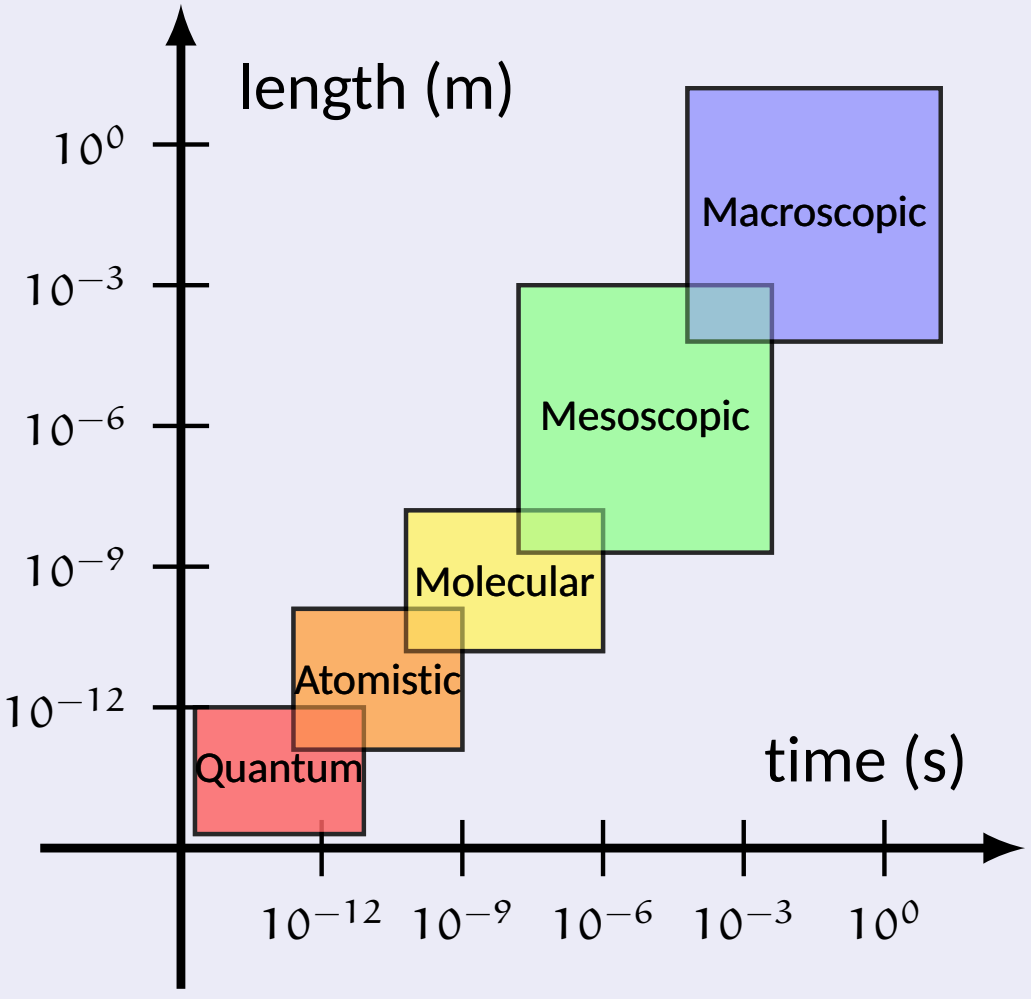
Methods

FE	Finite Elements
CFD	Computational Fluid Dynamics
LB	Lattice Boltzmann
MPC	MultiParticle Collision Dynamics
DPD	Dissipative Particle Dynamics
BD	Brownian Dynamics
MC	Monte Carlo
MD	Molecular Dynamics
CP	Electronic Structure

Molecular Simulation

What Is It?

Time and Length Scales



Methods

FE	Finite Elements
CFD	Computational Fluid Dynamics
LB	Lattice Boltzmann
MPC	MultiParticle Collision Dynamics
DPD	Dissipative Particle Dynamics
BD	Brownian Dynamics
MC	Monte Carlo
MD	Molecular Dynamics
CP	Electronic Structure

Microscopic to Macroscopic

Microscopic Evolution

$$\mathbf{q} = \{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_N\} \qquad \text{atomic coordinates}$$

$$\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N\} \qquad \text{conjugate momenta}$$

$$\mathbf{q}_t = \mathbf{q}(t)\ ,\ \mathbf{p}_t = \mathbf{p}(t) \qquad \text{time dependence}$$

$$\langle A \rangle = \frac{1}{t_{max}} \int_0^{t_{max}} dt\, A(\mathbf{q}_t, \mathbf{p}_t) \qquad \text{time average}$$

We compare results $\langle A \rangle$ with theories or experiments

- thermodynamics, phase transitions
- structure
- dynamical properties

Hamiltonian Mechanics

Hamiltonian = Kinetic energy + Potential energy

Equations of motion

$$H(\mathbf{q},\mathbf{p}) = K(\mathbf{p}) + U(\mathbf{q}) = \tfrac{1}{2}\left|\mathbf{p}\right|^2/m + U(\mathbf{q})$$

$$\dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt} = \mathbf{p}/m = \frac{\partial K}{\partial \mathbf{p}}\,, \quad \dot{\mathbf{p}} = \frac{d\mathbf{p}}{dt} = \mathbf{f} = -\frac{\partial U}{\partial \mathbf{q}}$$

- A system of coupled ordinary differential equations.
- Can be numerically integrated step-by-step.

The MD Algorithm

The Key Problem

$$\begin{pmatrix} \mathbf{q}(t) \\ \mathbf{p}(t) \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{q}(t+\Delta t) \\ \mathbf{p}(t+\Delta t) \end{pmatrix} , \quad t_{max} = n_{step}\Delta t$$

- Wish to make the timestep Δt as large as possible.
- Cannot follow true trajectory for long times t_{max}.
- Trajectories diverge from each other exponentially.
- However long-term energy conservation is possible.

The MD Algorithm

Velocity Verlet Equations

$$
\begin{aligned}
\mathbf{p}(t + \tfrac{1}{2}\Delta t) &= \mathbf{p}(t) + \tfrac{1}{2}\Delta t\, \mathbf{f}(t) \\
\mathbf{q}(t + \Delta t) &= \mathbf{q}(t) + \Delta t\, \mathbf{p}(t + \tfrac{1}{2}\Delta t)/m \\
\mathbf{f}(t + \Delta t) &= \mathbf{f}\big(\mathbf{q}(t + \Delta t)\big) \\
\mathbf{p}(t + \Delta t) &= \mathbf{p}(t + \tfrac{1}{2}\Delta t) + \tfrac{1}{2}\Delta t\, \mathbf{f}(t + \Delta t)
\end{aligned}
$$

Propagators

Formal Mechanics

Liouville Equations for $A(\mathbf{q}, \mathbf{p})$

$$\frac{dA}{dt} = \frac{\partial A}{\partial \mathbf{q}} \cdot \dot{\mathbf{q}} + \frac{\partial A}{\partial \mathbf{p}} \cdot \dot{\mathbf{p}} \qquad \text{chain rule}$$

$$= \left(\frac{\partial A}{\partial \mathbf{q}} \cdot \frac{\partial H}{\partial \mathbf{p}} - \frac{\partial A}{\partial \mathbf{p}} \cdot \frac{\partial H}{\partial \mathbf{q}} \right) \qquad \text{Hamilton's equations}$$

$$= (A, H) \qquad \text{Poisson bracket}$$

$$\equiv i\mathcal{L}A \qquad \text{Liouville operator}$$

$$A(\mathbf{q}_t, \mathbf{p}_t) = e^{i\mathcal{L}t} A(\mathbf{q}_0, \mathbf{p}_0) \qquad \text{Propagator}$$

Propagators

Molecular Dynamics

Trotter Decomposition

$$e^{i\mathcal{L}t_{max}} = \left(e^{i\mathcal{L}\Delta t}\right)^{n_{step}}_{approx} + \mathcal{O}\left(n_{step}\Delta t^3\right), \quad t_{max} = n_{step}\Delta t$$

Approximate propagator correct at short timesteps $\Delta t \to 0$.
Useful approximations arise from splitting $i\mathcal{L} = i\mathcal{P} + i\mathcal{Q}$.

Non-symmetric Splittings

$$e^{i\mathcal{L}\Delta t} = e^{(i\mathcal{P}+i\mathcal{Q})\Delta t} \approx e^{i\mathcal{P}\Delta t}\, e^{i\mathcal{Q}\Delta t}$$

$$\text{or} \quad \approx e^{i\mathcal{Q}\Delta t}\, e^{i\mathcal{P}\Delta t}$$

Propagators

Splitting the Liouville Operator

Drift

$$i\mathcal{Q} = \dot{\mathbf{q}} \cdot \frac{\partial}{\partial \mathbf{q}} = \frac{\mathbf{p}}{m} \cdot \frac{\partial}{\partial \mathbf{q}}$$

$$e^{i\mathcal{Q}\Delta t} A(\mathbf{q}, \mathbf{p}) = A\left(\mathbf{q} + \frac{\mathbf{p}}{m}\Delta t, \mathbf{p}\right)$$

Kick

$$i\mathcal{P} = \dot{\mathbf{p}} \cdot \frac{\partial}{\partial \mathbf{p}} = \mathbf{f} \cdot \frac{\partial}{\partial \mathbf{p}}$$

$$e^{i\mathcal{P}\Delta t} A(\mathbf{q}, \mathbf{p}) = A(\mathbf{q}, \mathbf{p} + \mathbf{f}\Delta t)$$

Propagators

Velocity Verlet Algorithm

Symmetric Splitting

$$\mathrm{e}^{\mathrm{i}\mathcal{L}\Delta t} = \mathrm{e}^{(\mathrm{i}\mathcal{P}+\mathrm{i}\mathcal{Q})\Delta t} \approx \underbrace{\mathrm{e}^{\mathrm{i}\mathcal{P}\Delta t/2}}_{\text{kick}} \underbrace{\mathrm{e}^{\mathrm{i}\mathcal{Q}\Delta t}}_{\text{drift}} \underbrace{\mathrm{e}^{\mathrm{i}\mathcal{P}\Delta t/2}}_{\text{kick}}$$

- Exactly time reversible and symplectic.
- Conserving phase space volume $\mathrm{d}\mathbf{q}_t\,\mathrm{d}\mathbf{p}_t = \mathrm{d}\mathbf{q}_0\,\mathrm{d}\mathbf{p}_0$.
- This gives the velocity Verlet algorithm.

Velocity Verlet Algorithm

MD Algorithm

Direct translation of symmetrically split propagator.

```
do step = 1, nstep
   p = p + ( dt / 2 ) * f      ! half-step kick
   q = q + dt * p / m          ! full-step drift
   f = force ( q )             ! force evaluation
   p = p + ( dt / 2 ) * f      ! half-step kick
end do
```

Propagators

Energy Stability

- Symplectic dynamics exactly conserves $\mathrm{H}^{\ddagger}$
- $\mathrm{H}^{\ddagger} = \mathrm{H} + \mathcal{O}(\Delta t^2)$
- H is true hamiltonian
- $\mathrm{H}^{\ddagger}$ is pseudo-hamiltonian or shadow hamiltonian

Such a stability property is extremely useful in MD, since we wish to sample constant-energy states.

Example

Consider a simple one-dimensional harmonic oscillator,

- mass m, spring constant k
- natural frequency $\omega = \sqrt{k/m}$

Harmonic Oscillator Equations

$$\dot{q} = p/m\,, \quad \dot{p} = -kq\,, \quad H(q,p) = \tfrac{1}{2}p^2/m + \tfrac{1}{2}kq^2$$

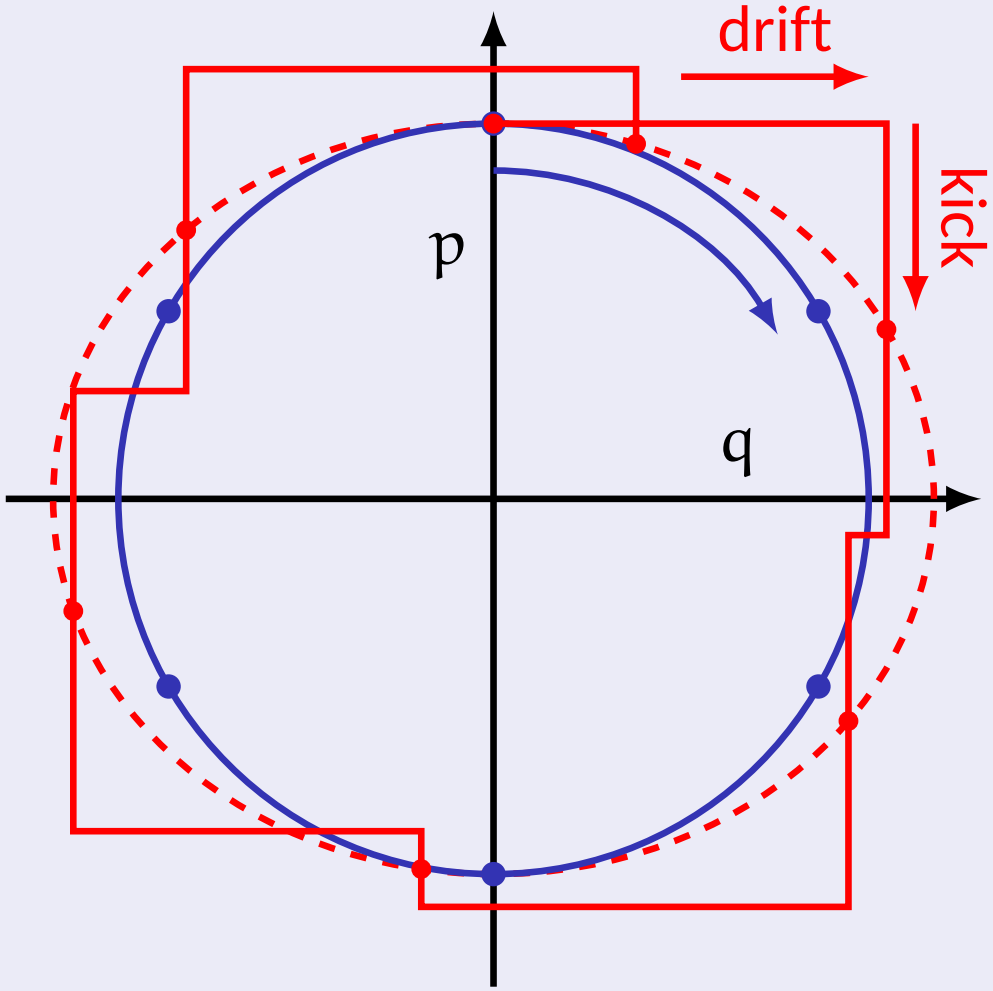
The Shadow Hamiltonian

$$H^{\ddagger}(r,p) = \tfrac{1}{2}p^2/m + \tfrac{1}{2}kq^2\left[1 - (\tfrac{1}{2}\omega\Delta t)^2\right]$$

Exactly conserved by velocity Verlet algorithm.

Example

Harmonic Oscillator



In a phase portrait, the simulated system remains on an ellipse

$$H^{\ddagger} = \text{constant}$$

which differs slightly from the ellipse,

$$H = \text{constant}$$

for small $\omega\Delta t$ (here $\omega\Delta t = \frac{1}{3}\pi$).

Multiple Timesteps

- Suppose there are slow forces $\mathbf{f}_{\text{slow}}$, and fast forces $\mathbf{f}_{\text{fast}}$.
- Momentum satisfies $\dot{\mathbf{p}} = \mathbf{f}_{\text{slow}} + \mathbf{f}_{\text{fast}}$.
- Break up Liouville operator $i\mathcal{L} = i\mathcal{P}_{\text{slow}} + i\mathcal{P}_{\text{fast}} + i\mathcal{Q}$:

Multiple Timestep Liouville Operator

$$i\mathcal{P}_{\text{slow}} = \mathbf{f}_{\text{slow}} \cdot \frac{\partial}{\partial \mathbf{p}}, \quad i\mathcal{P}_{\text{fast}} = \mathbf{f}_{\text{fast}} \cdot \frac{\partial}{\partial \mathbf{p}}, \quad i\mathcal{Q} = \frac{\mathbf{p}}{m} \cdot \frac{\partial}{\partial \mathbf{q}}$$

$$e^{i\mathcal{L}\Delta t} \approx e^{i\mathcal{P}_{\text{slow}}\Delta t/2}\, e^{(i\mathcal{P}_{\text{fast}}+i\mathcal{Q})\Delta t}\, e^{i\mathcal{P}_{\text{slow}}\Delta t/2}$$

$$e^{(i\mathcal{P}_{\text{fast}}+i\mathcal{Q})\Delta t} \approx \left(e^{i\mathcal{P}_{\text{fast}}\delta t/2}\, e^{i\mathcal{Q}\delta t}\, e^{i\mathcal{P}_{\text{fast}}\delta t/2}\right)^n, \quad \delta t = \Delta t/n$$

- $\mathbf{f}_{\text{fast}}$ computed once per δt.
- $\mathbf{f}_{\text{slow}}$ computed once per $\Delta t = n\delta t$.

Multiple Timesteps

Algorithm

Direct translation of split propagator.

```
do step = 1, nstep
   p = p + ( dt / 2 ) * f_slow
   do small_step = 1, n
      p      = p + ( dt_small / 2 ) * f_fast
      q      = q + dt_small * p / m
      f_fast = force_fast ( q )
      p      = p + ( dt_small / 2 ) * f_fast
   end do
   f_slow = force_slow ( q )
   p      = p + ( dt / 2 ) * f_slow
end do
```

Conclusions

- Molecular dynamics solves the classical equations of motion step by step on a computer.
- Modern algorithms are derived by splitting the Liouville operator into separately-integrable parts.
- The algorithms conserve a shadow hamiltonian.
- This is fun, as well as useful!
- Feel free to
  - . . . ask questions
  - . . . look at kittens