Molecular Dynamics

A. Donev, Courant, 2023

Consider a material made of an inert element like argon, so the atoms don't form molecules.

[MD applies to much more complicated materials but not for this lecture]

The material (at a given "temperature" and density) could be a

gas, liquid, or solid

[MD can be done at constant pressure and temperature but not here]

The state of the material, in classical mechanics, is described by the state $\vec{z} = \{\vec{r}, \vec{p}\}$

$$\vec{r}(t) = \{\vec{r}_1(t) \in \mathbb{R}^3, \dots, \vec{r}_N(t) \in \mathbb{R}^3\} \in \mathbb{R}^{3N}$$

$$\vec{p}(t) = \{m_1 \vec{v}_1(t) \in \mathbb{R}^3, \dots, m_N \vec{v}_N(t) \in \mathbb{R}^3\} \in \mathbb{R}^{3N}$$

$\vec{r}_i$ = position of (nucleus of) atom i

$\vec{v}_i = \frac{d\vec{r}_i}{dt} = \dot{r}_i$ = velocity of atom i

(Drop vector notation from now on)

Let's assume the atoms are confined to stay in a box, e.g.:

$$\vec{r}_i \in [0, L]^3$$

with some boundary conditions, usually periodic (models "bulk" materials).

ODEs to solve:

$$\frac{d^2 \vec{r}_i(t)}{dt^2} = \frac{F_i(\vec{r}(t), t)}{m_i}$$

(Assume no t dependence)

$\uparrow$ acceleration = force / mass (Newton's 2nd law)

force

$$\vec{F} = -\frac{\partial U(\vec{r}, t)}{\partial \vec{r}}$$

where $U \in \mathbb{R}$ is the interaction energy (modeling)

Now all we need to do is to solve a system of ODEs (simulation), easy, right? NO

(1) $N = O(10^5 - 10^7)$, e.g., protein in water

(2) We need to be stable and "accurate" (in some sense) for long-time integration (e.g., protein folding)

(3) We want simulations to last 1 ms but ODEs are super stiff, with intrinsic molecular timescales $O(fs)$

Hopeless, right? Maybe, but, let's try to make the goal achievable by carefully defining what we mean by accurate.

MD equations are chaotic - cannot hope for accuracy at the trajectory level.

It means at best we can hope for convergence / accuracy of statistical ensembles, i.e., distributions over path space. (this is like "weak" convergence)

Let's imagine running MD for a looong time.

Conjecture (holy grail in dynamical systems)

① MD dynamics is mixing

precise definition requires measure theory

Weaker assumption (required),
(ergodic theory)
(also a holy grail in dynamical systems)

② MD dynamics is ergodic.

Phase space is the set $\vec{z} = \{\vec{r}, \vec{p}\}$ $\vec{z}$ belongs to.

Ergodic $\simeq$ $\vec{z}$ eventually (finite time) all parts of phase space. Average behavior of process can be deduced from the trajectory of a "typical" point (non-typical points measure zero). Equivalently (per wiki), and importantly for MD, a large collection of finite trajectories can "represent" (statistically) the average statistical properties of all trajectories of the system.

I.E. we can run many MD trajectories in parallel and average statistics. Same as Monte Carlo (trivial parallelization at the ensemble level).

③ MD dynamics is ergodically mixing

Given two subsets of phase space, if I start in one subset and run MD for a long (but finite) time, I will visit the other subset.

(All taken from Wikipedia 🙂)

Let's assume these assumptions are true (not proven yet mathematically, only in some special systems of billiards)

Unknown: Is MD "reversible" or "irreversible" (arrow of time)

Since we want to know statistical properties, look instead at probability density $\rho(z,t)$ over phase space.

- Start with initial condition sampled from $\rho(z,t=0)$ and follow in time.
  (run many trajectories)

On what set is the measure supported?

Let's look at the total energy or Hamiltonian $H(z) \in \mathbb{R}$,

$$H(\vec{r},\vec{p}) = \underbrace{\sum_i \frac{1}{2} m_i v_i^2}_{\text{kinetic}} + \underbrace{U(\vec{r})}_{\text{potential energy}}$$

$$= \frac{1}{2}\sum_i \frac{p_i^2}{m_i} + U(\vec{r})$$

The equations of MD (Newton's law) can be rewritten in the form of Hamiltonian dynamics (generalizes Newton's laws to other dynamics with some state variables ($\vec{r}$) & conjugate variables ($\vec{p}$), links to quantum mechanic & more. E.g. Euler eqs of fluid mechanics are a continuum Hamiltonian system)

$$\begin{cases} \dot{r}_i = \dot{q}_i = \dfrac{\partial H}{\partial p_i} \\ \dot{p}_i = -\dfrac{\partial H}{\partial q_i} \end{cases} \Rightarrow \dot{z} = L \frac{\partial H}{\partial z}$$

$L = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$ for MD, more generally a shew-symmetric matrix

$$\frac{dH}{dt} = \frac{\partial H}{\partial z} \cdot \dot{z} = \frac{\partial H}{\partial z} \cdot L \frac{\partial H}{\partial z} =$$

$$= \sum_{i,j} \frac{\partial H}{\partial z_i} L_{ij} \frac{\partial H}{\partial z_j} = \sum_{i>j} \frac{\partial H}{\partial z_i} L_{ij} \frac{\partial H}{\partial z_j}$$

$$+ \sum_{j>i} \frac{\partial H}{\partial z_i} L_{ij} \frac{\partial H}{\partial z_j} = \sum - \sum = 0$$
$$i>j \qquad i>j$$

$$\frac{dH}{dt} = 0 \Rightarrow \text{energy is conserved}$$

Energy is a dynamical invariant. The only other dynamical invariant with periodic BCs is total momentum:

$$\vec{P} = \sum_i m_i \vec{v}_i$$

$$\frac{d\vec{P}}{dt} = \vec{m} \cdot \frac{d\vec{v}}{dt} = \vec{m} \cdot \frac{\vec{F}}{\vec{m}} = \sum \vec{F}_i$$

If $U(r)$ is translationally invariant,

$$U(r_1, \ldots, r_N) = U(r_1 - r_N, r_2 - r_N, \ldots, r_{N-1} - r_1)$$

then $\sum \vec{F}_i = -\sum \frac{\partial U}{\partial \vec{r}_i} = 0 \Rightarrow$

$$\frac{d\vec{p}}{dt} = 0$$

Aside: Noether's theorem:
Great but overlooked female mathematician

If a system has a continuous symmetry property, then there are corresponding quantities that are conserved in time. $\Rightarrow$

If $U(\vec{r})$ and BCs are rotationally invariant,

$$U(\vec{r}) = U(\overleftrightarrow{R}\,\vec{r}), \quad \overleftrightarrow{R} \text{ unitary}$$

then angular momentum is also conserved (not in periodic BCs since unit cell breaks (continuous) rotational symmetry)

Let $I(z) \in \mathbb{R}^n$, $n \ll N$ typically), denote the set of all conserved quantities.

If dynamics is ergodic & mixing, there exists an equilibrium distribution $\rho_{eq}(z) = \lim_{t \to \infty} \rho(t)$ (invariant measure) and

$$\rho_{eq}(z) = \Psi(I(z))$$

↑ some function

Fundamental assumption of statistical mechanics (very far from proven!):

The probability of being in any "microstate" $\vec{z}$ consistent with the dynamical invariants is equal among all microstates (uniform mixing)

(Principle of equiprobability)

Discuss: What does $\delta(H(z)-E)$ mean?

$$\rho_{eq}^{micro} = \delta(H(z)-E)$$

(here energy $E = H_0(z)$ = conserved)

This is at constant N, energy, and volume V (fixed periodic box), called the microcanonical ensemble

At constant temperature, N, and V,

$$S_{eq}^{canon}(\vec{z}) = \frac{1}{Z} e^{-\frac{H(z)}{k_B T}}$$

Gibbs-Boltzmann distribution

normalization (partition function), called the canonical ensemble.

Less important for us but for the record

(1) const NVE = microcanonical

(2) const NVT = canonical (must add thermostat)

(3) const μVT = macrocanonical (must add & remove particles)

(4) const NPT = isobaric ensemble (must add thermostat & allow unit cell to expand and deform by shear)

Let $f(z)$ be a given observable (phase function)

Ergodicity implies:

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T f(z(t))\, dt$$

$$= \langle f \rangle_{\rho_{eq}} = \int f(z)\rho_{eq}(z)\, dz$$

suitable Haar measure over phase space

We can use long MD trajectories (or many short ones if we choose initial condition by sampling $\rho_{eq}(z)$ using Monte Carlo) to compute observables "at (thermodynamic) equilibrium"

Hamiltonian dynamics conserves phase space volume (is a measure conserving flow)

Let $\vec{u}(z) = \frac{dz}{dt} = L\frac{\partial H}{\partial z}$

$\vec{u}$ ↑ flow vector field

$$\Rightarrow \nabla_z \cdot u = \vec{1} \cdot \frac{\partial \dot{z}}{\partial z} =$$

$$= \vec{1} \cdot \frac{\partial \dot{z}}{\partial z} = \sum_i \left[\frac{\partial}{\partial p_i}(\dot{p}_i) + \frac{\partial}{\partial q_i}(\dot{q}_i)\right]$$

$$= \sum_i \left[\frac{\partial}{\partial p_i}\left(-\frac{\partial H}{\partial q_i}\right) + \frac{\partial}{\partial q_i}\left(\frac{\partial H}{\partial p_i}\right)\right] = 0$$

Flow in phase space is incompressible

It is crucial for numerical methods to satisfy the same property in order to conserve the invariant measure, not just energy

Such numerical methods are called symplectic integrators (+ additional conditions not shown)

$$\Rightarrow \frac{\partial \rho}{\partial t} = -\vec{\nabla}_z \cdot (\rho \vec{u}) = -\vec{u} \cdot \nabla_z \rho$$

$$\Rightarrow \boxed{\frac{\partial \rho}{\partial t} + \vec{u} \cdot \nabla \rho = 0}$$ Liouville equation

Statistical rewriting of Newton's laws (two are equivalent - just plug $\rho(z,0) = \delta(z-z_0)$ in Liouville's equation)

Recall $\rho_{eq} = \varphi(H(z))$

$$\Rightarrow \partial_t \rho_{eq} = \vec{u} \cdot \nabla_z \rho_{eq} =$$

$$\frac{\partial \varphi}{\partial H} \sum_i \left(\frac{\partial H}{\partial q_i} \cdot \dot{q}_i + \frac{\partial H}{\partial p_i} \cdot \dot{p}_i \right) = 0$$

Notation:

Liouville equation in "fancy" notation:

$$\partial_t \rho = -\mathcal{L}\rho \leftarrow \text{LINEAR}$$

$\mathcal{L}$: Liouville operator

where $\mathcal{L}$ is a linear differential operator in phase space:

$$\mathcal{L}\rho = \frac{\partial H}{\partial p}\cdot\frac{\partial \rho}{\partial q} - \frac{\partial H}{\partial q}\cdot\frac{\partial}{\partial p}$$

All nothing but rewriting Newton's laws in ever-prettier ways

$$\rho(t) = e^{-\mathcal{L}t}\,\rho(0)$$

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$$

$\mathcal{L}_1 = \frac{\partial H}{\partial p}\cdot\frac{\partial}{\partial q}$ → free streaming of positions at constant p

$\mathcal{L}_2 = -\frac{\partial H}{\partial q}\cdot\frac{\partial}{\partial p}$ ↑ impulse/momentum kick at constant q

Numerical methods for MD

$$\frac{d^2 r_i}{dt^2} = \frac{F_i(r)}{m_i}$$

$$\frac{r_i^{k-1} - 2r_i^{k} + r_i^{k+1}}{\Delta t^2} = F_i(r)$$

2nd order in time, linear multistep method, called the Stormer-Verlet algorithm

This is the first and simplest integrator for MD, and it works, and is still used! Why?

Sidenote: First MD was hard-particle (billiard) MD & event-driven (newde overloaded)

Berni Alder, Tom Wainwright & Mary Ann Mansigh Karlsen

Def: A one step integrator is symplectic if it conserves phase space volume, i.e., it maintains measures (e.g., the equilibrium distribution), i.e. $z \xrightarrow[\Delta t]{} z'$

$$|\mathcal{J}| = \left| \frac{\partial z'}{\partial z} \right| = 1$$

Claim: Verlet's method is symplectic (homework ☺)

Aside: Empirical evidence & theory suggests symplectic integrators conserve energy better over long times.
Modern explanation: Symplectic integrators exactly conserve a shadow Hamiltonian

$$\tilde{H} = H + O(\Delta t^p)$$

e.g. for Verlet $\tilde{H} = \sum_i \frac{p_i^2}{2m_i} (\Delta t) + \dots$

Let's rewrite Verlet's method as a one step method

Positional Verlet integrator

$$\begin{cases} r_i^{n+1/2,*} = r_i^n + v_i^n \dfrac{\Delta t}{2} & \text{(midpoint)} \\ v_i^{n+1} = v_i^n + \dfrac{\Delta t}{m_i} F_i(r^{n+1/2,*}) & \text{one force eval} \\ r^{n+1} = r^{n+1/2} + v^{n+1} \dfrac{\Delta t}{2} \end{cases}$$

If you think about it, you will see this is equivalent to Verlet (but more expensive), and is a Strang splitting method:

$$e^{\mathcal{L}\Delta t} \approx e^{\mathcal{X}_1 \Delta t/2} e^{\mathcal{Y}_2 \Delta t} e^{\mathcal{X}_1 \Delta t/2}$$

Composition of symplectic steps is symplectic

If instead we use the splitting

$$e^{\mathcal{X}_2 \Delta t/2}\, e^{\mathcal{Y}_1 \Delta t}\, e^{\mathcal{Y}_2 \Delta t/2}$$

we get the

Velocity Verlet integrator

$$\begin{cases} r_i^{n+1} = r_i^n + v_i^n \Delta t + \dfrac{\Delta t^2}{2m_i} F_i(r^n) \\ v_i^{n+1} = v_i^n + \dfrac{\Delta t}{2m}\left[F_i(r^n) + F_i(r^{n+1})\right] \end{cases}$$

store & reuse in next step

To improve efficiency, observe

$$e^{\mathcal{X}_2 \Delta t/2}\, e^{\mathcal{Y}_1 \Delta t}\, e^{\mathcal{Y}_2 \Delta t/2}\, e^{\mathcal{X}_2 \Delta t/2}\, e^{\mathcal{Y}_1 \Delta t}\, e^{\mathcal{Y}_2 \Delta t/2} \cdots =$$

$$e^{\mathcal{X}_2 \Delta t/2}\, e^{\mathcal{Y}_1 \Delta t}\, e^{\mathcal{Y}_2 \Delta t}\, e^{\mathcal{Y}_1 \Delta t}\, e^{\mathcal{Y}_2 \Delta t} \cdots$$

startup; Verlet; Verlet

which only does a half step at the beginning or end of run. (20½)

Aside due to Charlie Peskin:

One can rewrite Verlet's method in many different ways (more on that later), but the most natural/symmetric way is as a staggered scheme:

$$\begin{cases} m_i \left(\dfrac{v_i^{n+1/2} - v_i^{n-1/2}}{\Delta t} \right) = F_i(\vec{r}^{\,n}) \\ \dfrac{x^{n+1} - x^n}{\Delta t} = v_i^{n+1/2} \end{cases}$$

Aside due to Jonathan Goodman:

If one sees this as a map

$$(m_i v_i^{n-1/2}, r_i^n) \to (m_i v_i^{n+1/2}, r_i^{n+1})$$

This is a symplectic integrator because it preserves the "symplectic two-form"

Good things about MD:

- It includes atomistic detail & can account for lots of chemistry (but not all, e.g. hydrogen bonds in water bad)
- It can be parallelized, and there are lots of good & fast codes for it: LAMMPS (CPU + MPI based, most "open"), HOOMD (GPU accelerated, geometric) UAMD (Raul Perez, was at Courant in Doucet group)

Bad things about MD:

There is no good way (yet) to parallelize in time, only in space, which severely limits what we can do with MD

① The required time step size $\Delta t \simeq 1$ fs. Longest MD on non-specialized supercomputers up to 1 ns, but we want ms or even s (DNA in water). Specialized hardware (D.E. Shaw's Anton 3 supercomputer, $N < 10^5$ atoms, ~200 ms per day on 64 nodes) resonance problem

② Multiple time step methods can help some but still

MD is infeasible for simulations at the second timescale (biology): need coarse graining

Coarse-Graining Hierarchy

See Teaching → "Coarse Grained Modeling of Materials" on my webpage

① Quantum Density Functional Theory (DFT) ← approximations + Car-Parrinello MD = Ab-initio MD

covalent bond breaking and forming, electronic polarization effects delocalization of orbitals

$\hbar$ small ↓ CG "correspondence principle" Ehrenfest theorem

② Classical (Hamiltonian) MD

- fixed covalent bonds
- all atom details

Up to here things have a strong basis in physics & math

From now on we enter the world of non-equilibrium thermodynamics, which is not precise and its foundations/principles are empirical & not true "laws of nature"

Let's consider a

Protein / Polymer (DNA) in water
(RNA-ase) (suspension)
(my expertise)

(3) Smoothed Dissipative Particle Dynamics (SDPD), Pep Español

Particles represent groups of atoms

There is both Hamiltonian dynamics and dissipation via Langevin (stochastic) terms (pairwise momentum exchange — conserves momentum & thus fluid dynamics

(S)DPD is an explicit solvent method – the water is explicitly simulated as particles (Lagrangian). One looses chemical specificity (anyone claiming otherwise is wrong 😉) but easily excludes solvent from penetrating the solute. Includes Brownian Motion

④ Another explicit solvent but now in an Eulerian formulation suitable for grid-based discretization is fluctuating hydrodynamics (my specialty): fluid represented via grid based CFD-type solver with noise (stochastic stress white in space & time), but solute explicitly modeled as in (S)DPD.

One can in principle include energy transport and compressibility (imagine ultrasound manipulation of biological samples or colloidal suspensions), confinement (fluid boundaries).

But, no good way (yet) to keep fluid from penetrating the solvent — existing methods are mostly of immersed boundary type (fluid everywhere in domain).

How to account for electrostatics, solvent polarization (dielectric), ions in water (electrolyte), etc.

My group works on these questions in the context of zero Re (Stokes flow) overdamped limit of no inertia (fluid or particles)

MD Demo code

A (tangential) aside:

Fortran (Disclaimer: I was on the Fortran committe 5 years)

- Oldest / first standardized high-level language, specialized for scientific computing! Think "Hidden Figures" movie female (African American) programmers
- Still alive and kicking!
- Main competitor to C++ for scientific computing
- Widely used in essentially all serious codes to do the most compute-intensive stuff on CPUs
- Not yet a standardized GPU extension (but in progress) (!!?)

(or use OpenACC directives)

- Very easy to learn basics
(I have had ~4-5 undergrads learn it and use it in 1-2 weeks)

- Very debuggable by design
(many errors forbidden at compile time, no "multiple dispatch" or metatemplates etc.)

- One serious drawback is the lack of genericity (think templates in C++) - I tried by proposing it to the committee but it was deemed too much work / too complex)

Revisions of Fortran:

not including Fortran 2018 & 202X

- First non-standard IBM version in April 1957
- Fortran 66, Fortran 77
- Fortran 90 & 95

First major revision that made it a modern language (controversial!)

Array syntax (like Matlab) comes from F90. Fortan 95 cleaned up some mistakes / omissions.

Features: | Equivalent in C++

i) Array syntax & assumed-shape arrays → None (boost, eigen, & other extensions)

ii) Allocatable arrays → like malloc but better :)

iii) Derived data types → like struct

©

iv) modules → something in between an include .h file & namespaces in C++ (better than either ☺)

Heavily abused to act as a class (not advised)

– Fortran 2003 (I was involved – mostly implemented in gfortran)

Another major revision to catch up (and improve upon!) C++

Features	C++ equivalent
– Lift all restrictions on allocatable arrays	→ – none built in, only extensions
– IEEE interoperability	
– Interoperability with C (and thus python, matlab, etc.)	
– CLASSes Object-oriented features based on Ada	→ class in C++

- Parametrited types → compile-time templates

- Procedure pointers (with compile-time checking)

- Function pointers (no compile or run time checking)

- Fortran 2008 (I was involved)

Another major revision, so major that some features are still not implemented by many compilers. Mostly but not fully/efficiently implemented in gfortran

It's main new feature are

co-arrays - parallelism based on "Single Instruction, multiple data model", partitioned but global memory, related to Universal Parallel C. Efficient implementation requires direct memory access (Cray model)

E