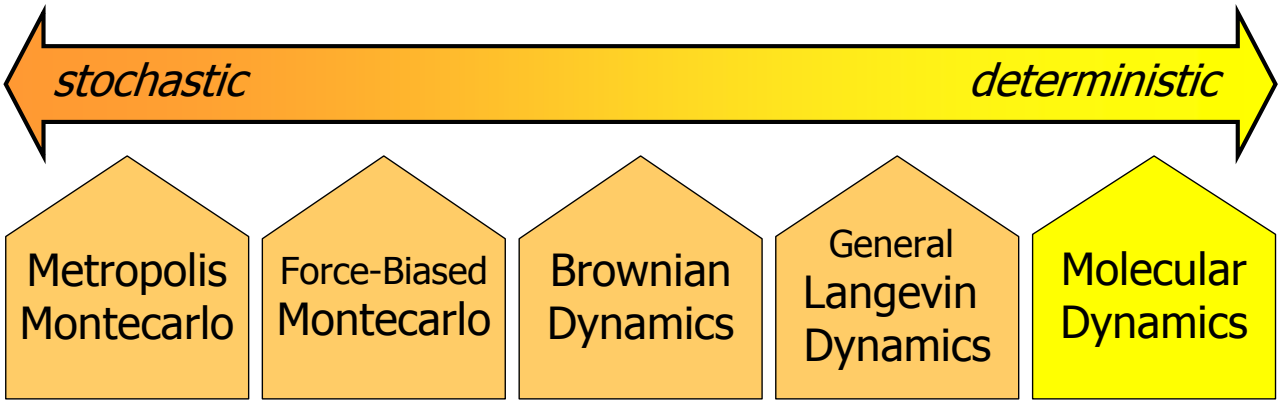
Molecular Dynamics

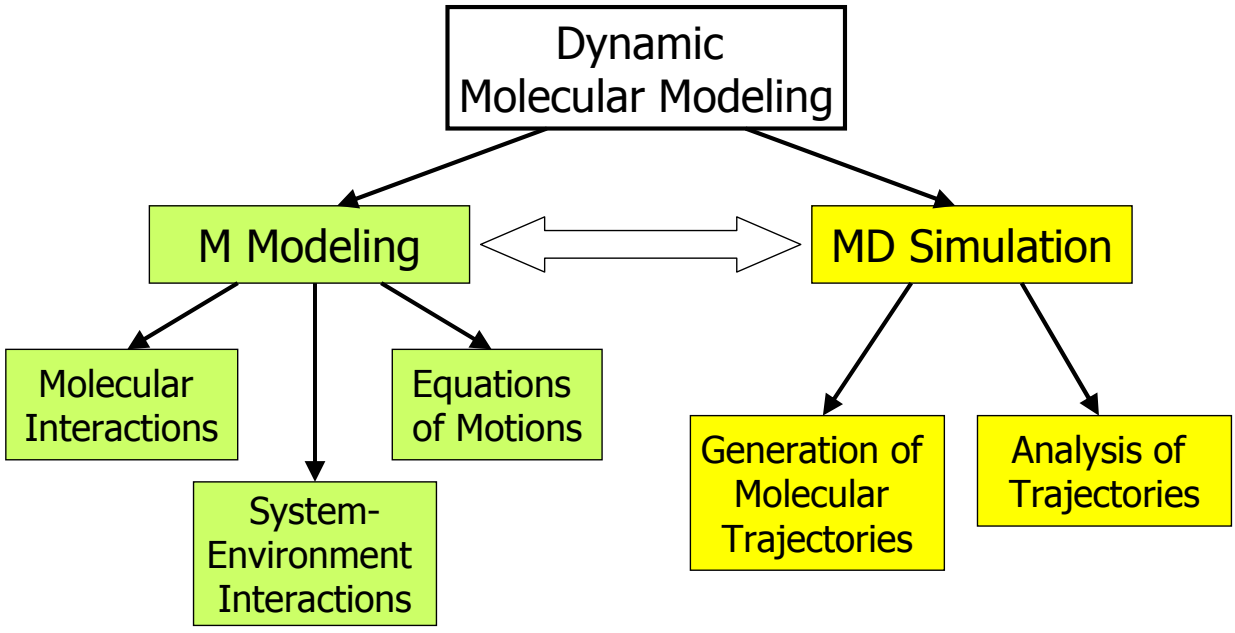
Molecular Simulation Methods



What's MD?

- Molecular Dynamics: methods to compute **trajectories** of timely-linked states in the phase space of a system of n particles.
- The position for the i-th particle is $\boldsymbol{x}_i$ and the linear momentum $\boldsymbol{p}_i$ ($\boldsymbol{p}_i = m\boldsymbol{v}_i$)
- The phase state for the whole system is given by both the position vector $\mathbf{X}$ and the linear momentum vector $\mathbf{P}$:
  $\mathbf{X}, \mathbf{P} \in \Re^{3N}$
- Collective velocity: $\boldsymbol{V} \in \Re^{3N}$
  Diagonal Matrix mass: $\boldsymbol{M} \in \Re^{3N \times 3N}$ for each part., rep. 3 times
- MD ***simulations*** are used to estimate systems behaviors that cannot be obtained analytically.
- MD represents the computer approach to statistical mechanics.

MD in Dynamic Molecular Modeling



Equations of Motion

- The MD approach is very simple in principle:
  to simulate the system evolution under the influence of a FF,
  according to Newton's equations of motion.
- For each particle,
$$m_i \dot{\mathbf{v}}_i(t) = F(\mathbf{x}_i) = -\nabla U(\mathbf{x}_i(t)) + \ldots$$
$$\dot{\mathbf{x}}_i(t) = \mathbf{v}_i(t)$$
- Collectively,
$$\mathbf{M}\dot{V}(t) = F(X) = -\nabla U(X(t)) + \ldots$$
$$\dot{X}(t) = V(t)$$
- These eq. must be integrated **numerically**, obtaining
  e.g. a sequence of pairs $\{X^n, V^n\}$ for integers n that represent
  discrete times $t=n\Delta t$ at intervals (**timesteps**) Δt ($\approx 10^{-15}$ sec)

Hamilton's Eq.s of Motion

- An alternative way to tackle the dynamics of a system is by the Hamilton's Equations of Motions

$$\dot{P} = -\frac{\partial H}{\partial X} \qquad \dot{X} = \frac{\partial H}{\partial P}$$

with

$$H(X,P) = \tfrac{1}{2} P^T M^{-1} P + U(X)$$

Algorithmic Stability

- An important characteristic of algorithms for finite-difference methods is **stability**, i.e. the capacity to avoid the amplification of errors from one step to the next.
- An algorithm is ***conditionally stable*** if it is stable for small Δt, but it becomes unstable when Δt goes beyond a critical threshold.
- Stability depends both on the used algorithm and the kind of differential equations.
- With nonlinear differential equations, no general analytical stability analysis is possible. After linearization,-> Lyapunov etc.
- *Stable solutions are not necessarily accurate ones!*

Finite-Difference Methods

- Among all FD methods, Runge-Kutta (RK) are the most popular.
- They are based on the Euler's method:

$$x(t+\Delta t) = x(t) + \dot{x}_{(est)}\,\Delta t$$

- Each RK method has its own way to estimate $\dot{x}_{(est)}$
- RKs generally show good stability, but are not suitable to MD because require too many force evaluations (usually 4) per atom per step!

Verlet Algorithm

- Shows exceptional stability over long times.
- Update for trajectory positions:

$$\mathbf{x}_i(t_0+\Delta t)=\mathbf{x}_i(t_0)+\dot{\mathbf{x}}_i(t)\big|_{t=t_0}\Delta t+\frac{1}{2!}\ddot{\mathbf{x}}_i(t)\big|_{t=t_0}(\Delta t)^2+\ldots$$

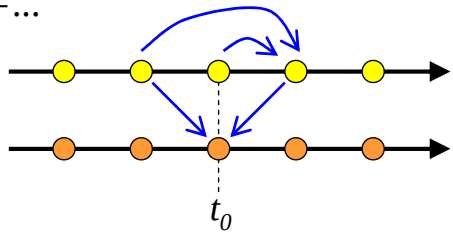
$$\mathbf{x}_i(t_0-\Delta t)=\mathbf{x}_i(t_0)-\dot{\mathbf{x}}_i(t)\big|_{t=t_0}\Delta t+\frac{1}{2!}\ddot{\mathbf{x}}_i(t)\big|_{t=t_0}(\Delta t)^2+\ldots$$

summing up,

$$\mathbf{x}_i(t_0+\Delta t)=2\mathbf{x}_i(t_0)-\mathbf{x}_i(t_0-\Delta t)+\ddot{\mathbf{x}}_i(t)\big|_{t=t_0}(\Delta t)^2-\ldots$$

velocity can be obtained as

$$\mathbf{v}_i(t_0)=\frac{\mathbf{x}_i(t_0+\Delta t)-\mathbf{x}_i(t_0-\Delta t)}{2\Delta t}$$



Verlet Algorithm (collective)

- Let's call the "acceleration"

$$\tilde{F}(X(t))=\mathbf{M}^{-1}F(X(t))=-\mathbf{M}^{-1}\nabla U(X(t))$$

- Update for trajectory positions:

$$X(t_0+\Delta t)=X(t_0)+\dot{X}(t)\big|_{t=t_0}\Delta t+\frac{1}{2!}\tilde{F}(t)\big|_{t=t_0}(\Delta t)^2+\ldots$$

$$X(t_0-\Delta t)=X(t_0)-\dot{X}(t)\big|_{t=t_0}\Delta t+\frac{1}{2!}\tilde{F}(t)\big|_{t=t_0}(\Delta t)^2-\ldots$$

summing up,

$$X(t_0+\Delta t)=2X(t_0)-X(t_0-\Delta t)+\tilde{F}(t)\big|_{t=t_0}(\Delta t)^2+\ldots$$

velocity can be obtained as

$$V(t_0)=\frac{X(t_0+\Delta t)-X(t_0-\Delta t)}{2\Delta t}$$

Leapfrog

- Velocity is defined at "half-steps".

$$V(t_0+\frac{\Delta t}{2}) = V(t_0-\frac{\Delta t}{2}) + \Delta t \tilde{F}(t)\Big|_{t=t_0}$$

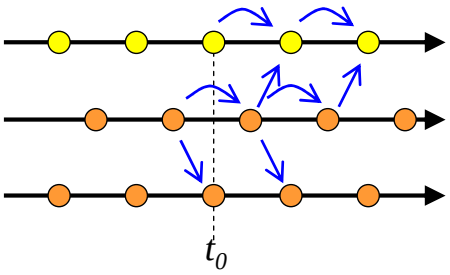
then the next position :

$$X(t_0+\Delta t) = X(t_0) + \Delta t V(t_0+\frac{\Delta t}{2})$$

This can be derived taking into account that

$$V(t_0) = V(t_0-\frac{\Delta t}{2}) + \frac{\Delta t}{2}\tilde{F}(t)\Big|_{t=t_0}$$

follows that $V(t_0+\frac{\Delta t}{2}) = V(t_0-\frac{\Delta t}{2}) + \Delta t \tilde{F}(t)\Big|_{t=t_0}$



Predictor-Corrector Algorithms

- Steps:
  1. Predict X(t+Δt) and V(t+Δt) at the end of the next step
  2. With such results, evaluate forces at t+Δt
  3. Use the values computed so far to correct the predictions
- Steps 2-3 could be iterated
- Used in MD:
  Gear - higher order, requires up to $X^{(v)}$

Algorithm Quality Evaluation

- A finite-difference algorithm in MD can be evaluated by its ability in conserving the system energy E
- A possible index is the *rms global error per step*, computed over M steps:

$$\langle g_e \rangle = \sqrt{\frac{\sum_{j=1}^{M} \left(E(j\Delta t) - E(0) \right)^2}{M}}$$

Hamiltonian: System Energy

- The overall system energy is usually computed by means of the Hamiltonian
  - In terms of collective velocity:

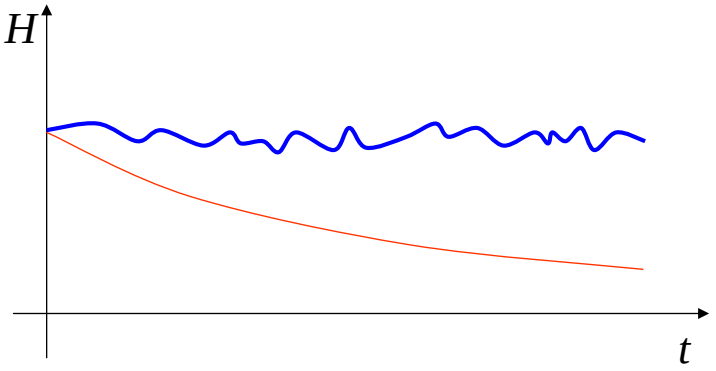
$$H(t) = \tfrac{1}{2} V^T(t)\, \mathbf{M}\, V(t) + U(X(t))$$

  - In terms of collective linear momentum:

$$H(t) = \tfrac{1}{2} P^T(t)\, \mathbf{M}^{-1}\, P(t) + U(X(t))$$

Symplectic Integrators

- Symplectic Integrators are numerical integration schemes, designed for the solution of Hamilton's equations of motions.
- SI hold the property of conserving the Hamiltonian of the system (apart some perturbations).
- Simple Euler and RK schemes are not SI; Verlet is a SI.

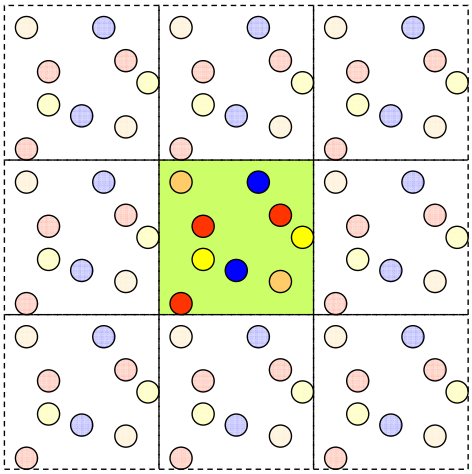


Reduced Units

- To reduce the risk of encountering values outside the representation range, often *reduced units* are used

Periodic Boundary Conditions

- The simulation considers the system in a "container" of some sort
- A bounded system with no "physical wall" is dealt with by *periodic boundary conditions (pbc)*
- Wraparound effect



Steps in MD Simulations

- A MD simulation goes through 3 phases:

1) Initialization – to specify initial coordinates and velocities (according to Maxwell-Boltzmann distribution, given T)
2) Equilibration – initial round to accommodate exchanges between kinetic and potential energies up to a mean value for them
3) Actual run (production)

- (lecture 3a on desktop)

Dealing with Constraints

- Possible approach to increase timesteps:
  introduction of constraints on bonds
- *Basic idea:*
  by adding such constraints,
  the highest-frequencies vibrations do not show up
- *Problem:*
  in macromolecules, different kinds of vibrations
  are "tangled" in complex ways
- …anyway, bonds involving H
  can be usually taken as rigid
- …conversely, rigid bond angles
  do not usually deliver realistic/reliable models

Lagrange Multipliers (I)

- Some significant ("holonomic", i.e. "rigid") constraints
  can be (collectively) expressed as

$$g(X(t)) = C \quad \text{or, with no loss in generality,} \quad g(X(t)) = 0$$

- E.g. a rigid bond between atom i and j as the k-th constraint:

$$g_k = r_{ij}^2(t) - \bar{r}_{ij}^2 = 0 \quad \text{with} \quad r_{ij} = \mathbf{x}_i - \mathbf{x}_j$$

- Collectively, also the all h constraints can be represented as a
  vector G, whose elements are always 0:

$$G(X(t)) = 0 \qquad G \in \Re^k$$

Lagrange Multipliers (II)

- To consider the constraints within the equations of motions, we can add another term (always 0) to the potential energy:

$$U(X(t)) \longrightarrow U(X(t)) + G(X(t))^T \Lambda \qquad G, \Lambda \in \Re^k$$

$$\left.\begin{aligned} \mathbf{M}\dot{V}(t) &= -\nabla U(X(t)) \\ \dot{X}(t) &= V(t) \end{aligned}\right\} \Rightarrow \begin{aligned} \mathbf{M}\dot{V}(t) &= -\nabla\left(U(X(t)) + G(X(t))^T \Lambda\right) \\ \dot{X}(t) &= V(t) \\ G(X(t)) &= 0 \end{aligned}$$

- I.e., each constraint can be sustained by a "fictitious" force
- A corresponding FD method will look something like

$$X(t_0 + \Delta t) = \hat{X}(t_0 + \Delta t) + \mathbf{M}^{-1}\nabla\left(G(X(t_0))^T \Lambda\right)(\Delta t)^2$$

$$G(X(t_0 + \Delta t)) = G\left(\hat{X}(t_0 + \Delta t) + \mathbf{M}^{-1}\nabla\left(G(X(t_0))^T \Lambda\right)(\Delta t)^2\right) = 0$$

Lagrange Multipliers: Issues

- Typical possible change in Δt : from Δt=1fs to Δt=2fs.
- At each step, additional work is required to deal with constraints (iterative solution of a nonlinear system in the λ_h unknowns)
- Fortunately, working with constraints usually means working with sparse matrices (in fact not all particles are involved in all constraints) and efficient numeric methods are available for them
- Any proposal for a constrained model is related to specific, proper algorithms for it
- Examples: SETTLE, SHAKE, RATTLE…

Reducing Δt: MTS (Multi-TimeStep) Methods

- *Basic idea:*
  forces due to long-distance interactions change at a lower pace than the others, → they could be evaluated less frequently
- E.g. we can split force components in F_{fast}, F_{med} and F_{slow}, with corresponding timesteps $\Delta t_{fast} \leq \Delta t_{med} \leq \Delta t_{slow}$ ($\Delta t_{med} = k_1 \Delta t_{fast}$, $\Delta t_{slow} = k_2 \Delta t_{med}$ with $k_1, k_2 \in N$)
- Between updates, the slower components can be inserted in the integration scheme either by *extrapolation* (kept constant) or by *impulses* (delta functions)
- *Problems:*
  simple extrapolation schemes are not symplectic;
  symplectic impulse MTS schemes have been found, but spurious behaviors occur as $\Delta t_{slow} \approx$ natural system frequency

Statistical Ensembles

- What said so far is valid for the so-called **Micro-Canonical ensemble** (constant NVE, i.e. number of particles, volume and total energy), with total energy (E) as a constant of motion.
- The micro-canonical ensemble is used for studying adiabatic conditions
- A **statistical ensemble** is an ideal collection of (infinite) copies of a system (of particles), each able to eventually assume every possible state/configuration *under given conditions*
- A statistical ensemble can be viewed as an (infinite) number of observations over the same physical systems under the same conditions

Other Ensembles

- It can be important studying molecular properties as functions of other quantities (not necessarily volume and energy).
- Up to this extent, other ensembles are needed
  - **Canonical ensemble** or NVT, i.e. constant volume and temperature
  - **Isothermal-isobaric ensemble** or NPT, i.e. constant pressure and temperature
  - **NPH**, i.e. constant pressure and enthalpy
- We shall skip the discussion on methods for such ensembles.