Molecular Dynamics Simulations Demystified:
(Part I) Principles and Algorithms

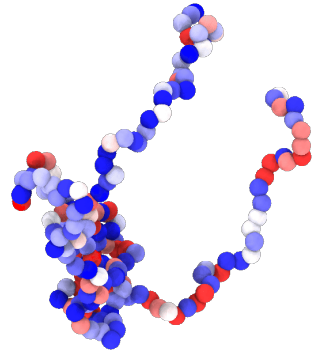
Instructor: **Jerelle Joseph**
email: jerellejoseph@princeton.edu

Contents

- Structure an MD simulation
- Representation of molecular systems
  - Potential energy functions and force fields
- Newtonian equations of motion
- MD integrators
  - Velocity Verlet algorithm

Overview of molecular dynamics

- **Microscopic description** of our system (e.g., atomistic model of a protein)
- Specify the **conditions** under which we want to study such system (i.e., statistical mechanical ensemble, T, p)
- **Sample** the system in the specified ensemble
  - Propagate using equations of motion
  - i.e., Generate configurations



- **Compute properties** of interest
  - Thermodynamic average from stat. mech.: $\langle \mathcal{X} \rangle = \int \mathcal{X}\left(\mathbf{r}^N\right) \mathcal{P}\left(\mathbf{r}^N\right) d\mathbf{r}^N$
  - Molecular dynamics estimation: $\langle \mathcal{X} \rangle = \lim_{t\to\infty} \frac{1}{t} \int_0^t \mathcal{X}\left(\mathbf{r}^N(t)\right) d\tau$

Basic structure of an MD simulation program

Overall, molecular dynamics programs are quite simple:

```
Initialize system (positions, velocities)
Perform energy minimization/massaging

for each time increment:
      compute forces acting on particles
      compute new positions and velocities
      (apply any extra algorithms)
      (compute properties, write data)

compute properties, write data
```

Describing molecular systems:
Potential energy functions

- Let $\mathbf{r}^{\mathrm{N}}$ represent the set of atomic coordinates of a system
- $U\left(\mathbf{r}^{\mathrm{N}}\right) =$ "potential energy function"
- $U\left(\mathbf{r}^{\mathrm{N}}\right)$ describes how the potential energy of a system changes as a function of atomic coordinates
- $U\left(\mathbf{r}^{\mathrm{N}}\right)$ is often factorized into contributions from bonded interactions (intramolecular) and non-bonded interactions (intermolecular)

Potential energy functions and force fields

$$U(\mathbf{r}^{\mathrm{N}}) = U_{\mathrm{bonded}}(\mathbf{r}^{\mathrm{N}}) + U_{\text{non-bonded}}(\mathbf{r}^{\mathrm{N}})$$

$$U_{\mathrm{bonded}}(\mathbf{r}^{\mathrm{N}}) = \sum_{ij,\mathrm{bonds}} U_{\mathrm{stretch}}(r_{ij}) + \sum_{ijk,\mathrm{angles}} U_{\mathrm{bend}}(\theta_{ijk}) + \sum_{ijkl,\mathrm{dihedrals}} U_{\mathrm{torsion}}(\phi_{ijkl})$$

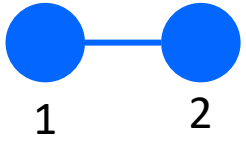
$$U_{\text{non-bonded}}(\mathbf{r}^{\mathrm{N}}) = \sum_{ij,\text{van der Waals}} U_{\mathrm{vdw}}(r_{ij}) + \sum_{ij,\mathrm{electrostatics}} U_{\mathrm{el}}(r_{ij})$$

Force field = functional forms of potential energy functions + the parameters (equilibrium bond lengths, angles, partial charges, bond force constants)

Bonded potentials

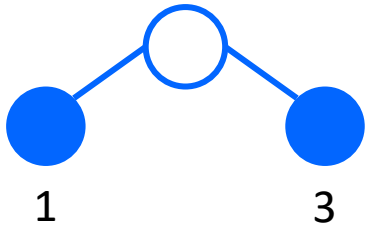
Bonded potentials capture covalent intramolecular interactions. They are usually (at least in molecules) partitioned onto n-body terms up to $n = 4$:

$$U_{\text{bonded}}(\mathbf{r}^{\text{N}}) = \sum_{ij,\text{bonds}} U_{\text{stretch}}(r_{ij}) + \sum_{ijk,\text{angles}} U_{\text{bend}}(\theta_{ijk}) + \sum_{ijkl,\text{dihedrals}} U_{\text{torsion}}(\phi_{ijkl})$$



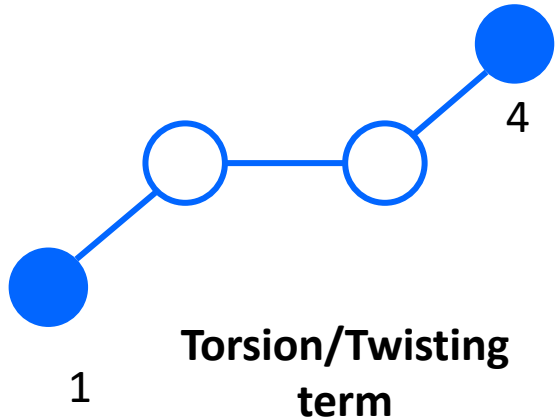
Stretching term

1-2 interactions



Bending term

1-3 interactions



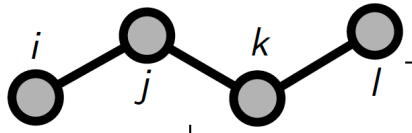
Torsion/Twisting term

1-4 interactions

Bonded potentials (Degrees of freedom)

Bonded potentials capture covalent intramolecular interactions. They are usually (at least in molecules) partitioned onto n-body terms up to $n = 4$:

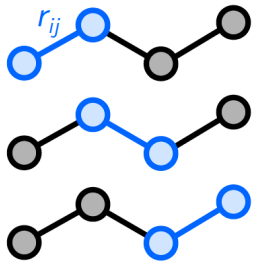
$$U_{\text{bonded}}(\mathbf{r}^{\text{N}}) = \sum_{ij,\text{bonds}} U_{\text{stretch}}(r_{ij}) + \sum_{ijk,\text{angles}} U_{\text{bend}}(\theta_{ijk}) + \sum_{ijkl,\text{dihedrals}} U_{\text{torsion}}(\phi_{ijkl})$$



Bonds

DoF: bond distance

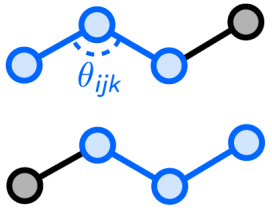
$$r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$$



Angles

DoF: angle between bond vectors*

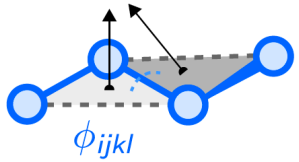
$$\cos\theta_{ijk} = \frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{kj}}{|\mathbf{r}_{ij}||\mathbf{r}_{jk}|}$$



Dihedrals

DoF: angle between plane normals*

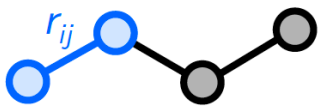
$$\cos\phi_{ijkl} = \frac{\mathbf{n}_{ijk} \cdot \mathbf{n}_{jkl}}{|\mathbf{n}_{ijk}||\mathbf{n}_{jkl}|}$$



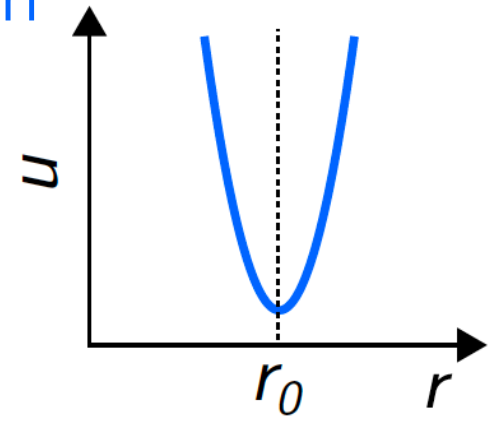
* Conventions may differ ($\mathbf{r}_{ij} \cdot \mathbf{r}_{kj} \neq \mathbf{r}_{ij} \cdot \mathbf{r}_{jk}$ etc.)

Example functional form for stretch term

Harmonic bond



$$U_{\text{stretch}}\left(r_{ij}\right) = \frac{1}{2} k_r \left(r_{ij} - r_0\right)^2$$



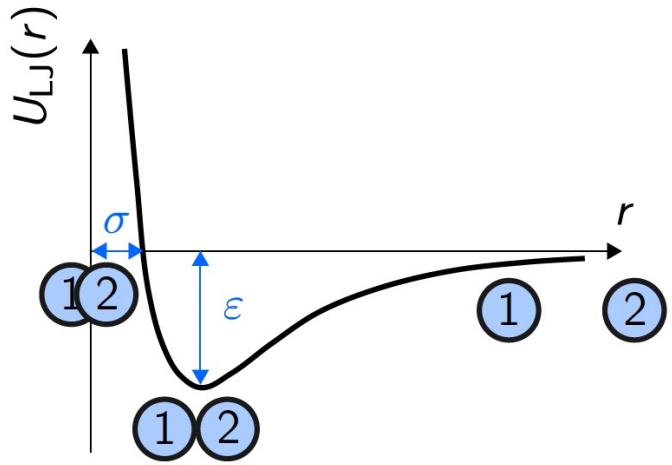
- Bond stretching represents a 'stiff degree of freedom', responsible for fastest vibrations at a femtosecond scale
- They are often modeled using Harmonic functions
- k_r (bond force constant) and r_0 (equilibrium bond length) are constants
- Many force fields use the similar potential energy functional forms, but differ in their constants

Example functional form for van der Waals interactions

$$U_{\text{non-bonded}}(\mathbf{r}^{\text{N}}) = \sum_{ij,\text{van der Waals}} U_{\text{vdw}}(r_{ij}) + \sum_{ij,\text{electrostatics}} U_{\text{el}}(r_{ij})$$

- The van der Waals term attempts to capture non-charged non-bonded interactions
- Lennard-Jones like potentials are often used for this purpose

$$U_{\text{vdw}}(r_{ij}) = U_{\text{LJ}}(r_{ij}) = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

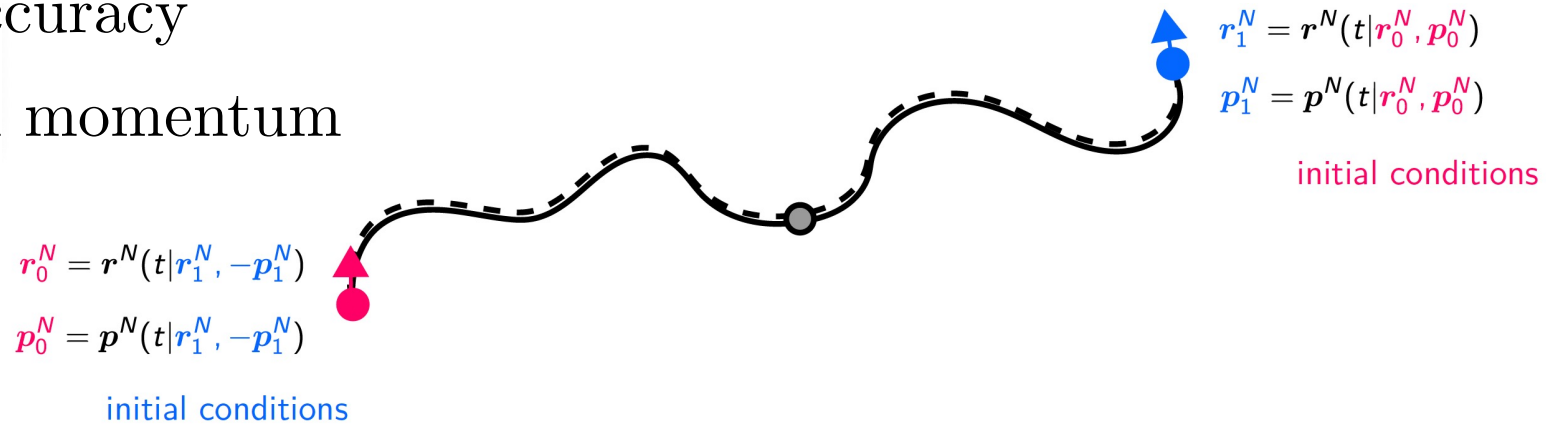


Equations of motion: updating positions and velocities

- At the start of the simulation, we need to assign the **initial positions** (e.g., from PDB structure) and **velocities** (e.g., from Maxwell–Boltzmann distribution of speeds at target temperature)
- Then, given the **current atomic positions, velocities and forces**...
- We need to integrate equations of moton to find the **next positions and velocities** over some finite time (δt aka **timestep**)
- Newtonian equations of motion are most commonly employed
- They are particularly convenient when working in Cartesian coordinates
- Newtonian: $\mathbf{F} = \frac{d\mathbf{p}}{dt} = m\frac{d^2\mathbf{r}}{dt^2} = m\mathbf{a}$
- The force is taken as the negative gradient of the potential energy
- Force: $\mathbf{F} = -\nabla U(\mathbf{r})$ (i.e., conservative force)

MD Integrators: integrating equations of motion

- In its simplest, common manifestation, MD corresponds to a numerical integration of Newton's equations of motion
- Absent of other treatments, the configurations belong to the NVE (microcanonical) ensemble
- MD integration schemes (aka integrators) should have certain desirable properties
  - minimal need to compute forces (a possibly expensive calculation)
  - good stability and accuracy
  - conserves energy and momentum
  - time-reversible



MD Integrators: integrating equations of motion

- Most integration schemes are built by considering a Taylor expansion in time from a given position
- For example, given the current position, the next atomic position can be found via a Taylor expansion

$$\begin{aligned}\mathbf{r}(t+\delta t) &= \mathbf{r}(t) + \frac{d\mathbf{r}}{dt}\delta t + \frac{d^2\mathbf{r}}{dt^2}\frac{\delta t^2}{2} + \mathcal{O}\left(\delta t^3\right) \\ &= \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{\mathbf{f}(t)}{2m}\delta t^2 + \mathcal{O}\left(\delta t^3\right)\end{aligned}$$

- There are several reasonable schemes for integrating these equations of motion (**Verlet, Leapfrog, Velocity Verlet**)

The Integrator: Velocity Verlet algorithm

(1) Advance positions $\mathbf{r}_i(t)$ by Taylor expansion from t to $t+\delta t$:

$$\mathbf{r}_i(t+\delta t) = \mathbf{r}_i(t) + \delta t \mathbf{v}_i(t) + \frac{\delta t^2}{2m_i}\mathbf{f}_i(t) + O\left(\delta t^3\right)$$

(2) Derive the backward Taylor expansion from $t+\delta t$ to t :

$$\mathbf{r}_i(t) = \mathbf{r}_i(t+\delta t) - \delta t \mathbf{v}_i(t+\delta t) + \frac{\delta t^2}{2m_i}\mathbf{f}_i(t+\delta t) - O\left(\delta t^3\right)$$

(3) Add the backward expansion to the forward expansion:

$$\mathbf{r}_i(t+\delta t) + \mathbf{r}_i(t) = \mathbf{r}_i(t) + \mathbf{r}_i(t+\delta t) +$$
$$+ \delta t\left[\mathbf{v}_i(t) - \mathbf{v}_i(t+\delta t)\right] + \frac{\delta t^2}{2m_i}\left[\mathbf{f}_i(t) + \mathbf{f}_i(t+\delta t)\right]$$

Simplify & rearrange to obtain expression for advancing velocities:

$\mathbf{v}_i(t+\delta t) = \mathbf{v}_i(t) + \frac{\delta t}{2m_i}\left[\mathbf{f}_i(t) + \mathbf{f}_i(t+\delta t)\right]$ **Velocity Verlet algorithm**

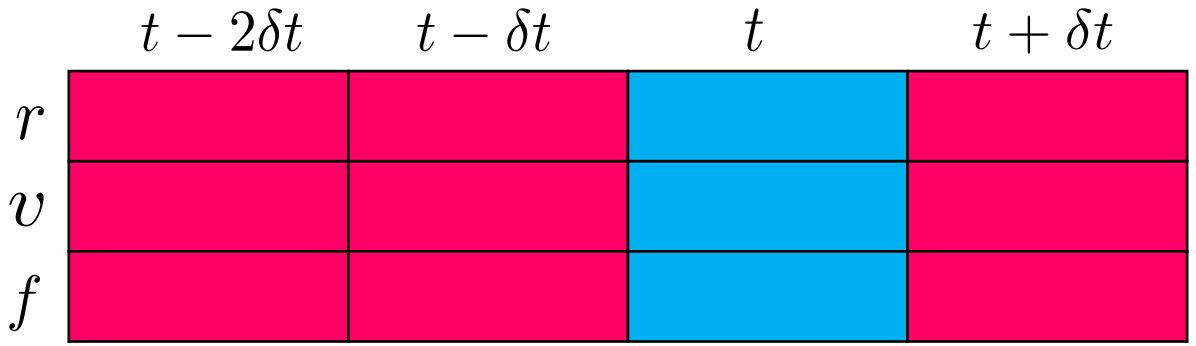
The Integrator: Velocity Verlet algorithm

Consider expansions in position and velocity:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \frac{d\mathbf{r}}{dt}\delta t + \frac{d^2\mathbf{r}}{dt^2}\frac{\delta t^2}{2} + \mathcal{O}\left(\delta t^3\right)$$

$$= \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{\mathbf{f}(t)}{2m}\delta t^2 + \mathcal{O}\left(\delta t^3\right)$$

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t) + \frac{\mathbf{f}(t+\delta t) + \mathbf{f}(t)}{2m}\delta t$$



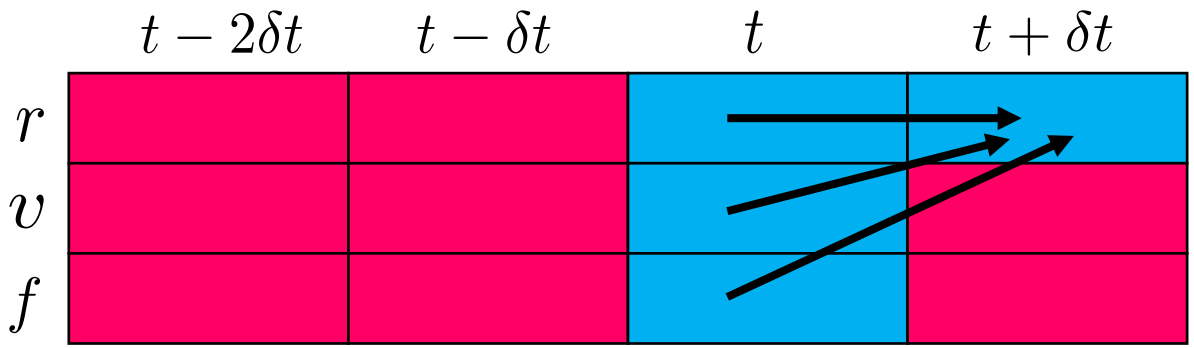
The Integrator: Velocity Verlet algorithm

Consider expansions in position and velocity:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \frac{d\mathbf{r}}{dt}\delta t + \frac{d^2\mathbf{r}}{dt^2}\frac{\delta t^2}{2} + \mathcal{O}\left(\delta t^3\right)$$

$$= \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{\mathbf{f}(t)}{2m}\delta t^2 + \mathcal{O}\left(\delta t^3\right)$$

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t) + \frac{\mathbf{f}(t+\delta t) + \mathbf{f}(t)}{2m}\delta t$$



At time t: $\mathbf{r}(t)$, $\mathbf{v}(t)$, $\mathbf{f}(t)$

1. Update pos. to t+δt with current pos., vels., & frcs.

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{\mathbf{f}(t)}{2m}\delta t^2$$

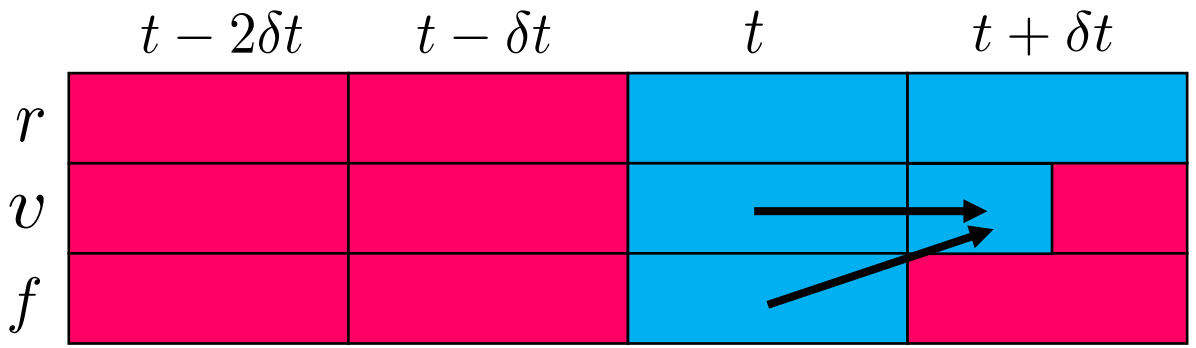
The Integrator: Velocity Verlet algorithm

Consider expansions in position and velocity:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \frac{d\mathbf{r}}{dt}\delta t + \frac{d^2\mathbf{r}}{dt^2}\frac{\delta t^2}{2} + \mathcal{O}\left(\delta t^3\right)$$

$$= \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{\mathbf{f}(t)}{2m}\delta t^2 + \mathcal{O}\left(\delta t^3\right)$$

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t) + \frac{\mathbf{f}(t+\delta t) + \mathbf{f}(t)}{2m}\delta t$$



At time t: $\mathbf{r}(t)$, $\mathbf{v}(t)$, $\mathbf{f}(t)$

1. Update pos. to t+δt with current pos., vels., & frcs.

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{\mathbf{f}(t)}{2m}\delta t^2$$

2. Perform partial update to vels. with current vels., frcs.

$$\mathbf{v}^*(t+\delta t) = \mathbf{v}(t) + \frac{\mathbf{f}(t)}{2m}\delta t$$

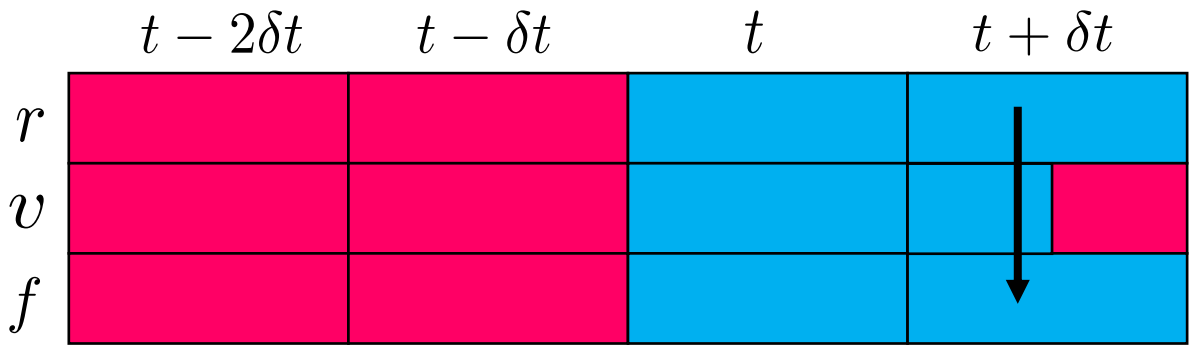
The Integrator: Velocity Verlet algorithm

Consider expansions in position and velocity:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \frac{d\mathbf{r}}{dt}\delta t + \frac{d^2\mathbf{r}}{dt^2}\frac{\delta t^2}{2} + \mathcal{O}\left(\delta t^3\right)$$

$$= \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{\mathbf{f}(t)}{2m}\delta t^2 + \mathcal{O}\left(\delta t^3\right)$$

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t) + \frac{\mathbf{f}(t+\delta t) + \mathbf{f}(t)}{2m}\delta t$$



At time t: $\mathbf{r}(t)$, $\mathbf{v}(t)$, $\mathbf{f}(t)$

1. Update pos. to t+δt with current pos., vels., & frcs.

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{\mathbf{f}(t)}{2m}\delta t^2$$

2. Perform partial update to vels. with current vels., frcs.

$$\mathbf{v}^*(t+\delta t) = \mathbf{v}(t) + \frac{\mathbf{f}(t)}{2m}\delta t$$

3. Compute new frcs. with new pos.

$$\mathbf{f}(t+\delta t) = -\nabla U\left(\mathbf{r}^N(t+\delta t)\right)$$

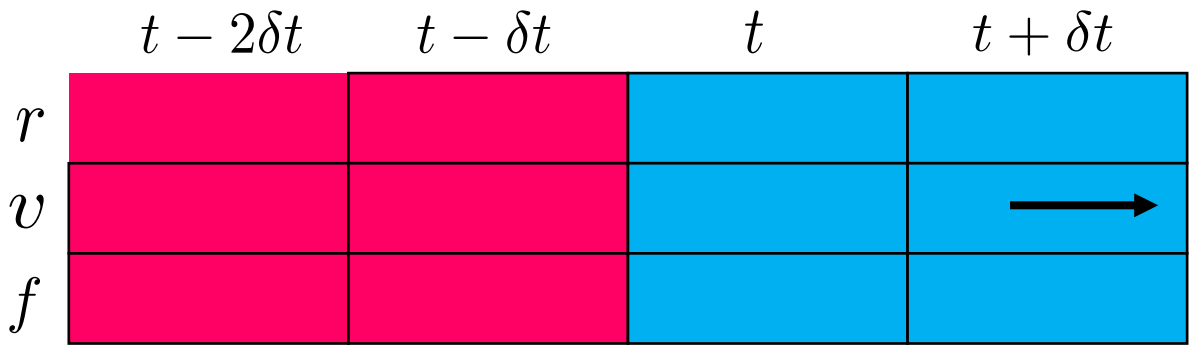
The Integrator: Velocity Verlet algorithm

Consider expansions in position and velocity:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \frac{d\mathbf{r}}{dt}\delta t + \frac{d^2\mathbf{r}}{dt^2}\frac{\delta t^2}{2} + \mathcal{O}\left(\delta t^3\right)$$

$$= \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{\mathbf{f}(t)}{2m}\delta t^2 + \mathcal{O}\left(\delta t^3\right)$$

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t) + \frac{\mathbf{f}(t+\delta t) + \mathbf{f}(t)}{2m}\delta t$$



At time t: $\mathbf{r}(t)$, $\mathbf{v}(t)$, $\mathbf{f}(t)$

1. Update pos. to t+δt with current pos., vels., & frcs.

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{\mathbf{f}(t)}{2m}\delta t^2$$

2. Perform partial update to vels. with current vels., frcs.

$$\mathbf{v}^*(t+\delta t) = \mathbf{v}(t) + \frac{\mathbf{f}(t)}{2m}\delta t$$

3. Compute new frcs. with new pos.

$$\mathbf{f}(t+\delta t) = -\nabla U\left(\mathbf{r}^N(t+\delta t)\right)$$

4. Complete velocity update

$$\mathbf{v}(t+\delta t) = \mathbf{v}^*(t+\delta t) + \frac{\mathbf{f}(t+\delta t)}{2m}\delta t$$