Molecular Dynamics

Chapter 1.1 Molecular Dynamics: Basics

Molecular dynamics is an important tool to investigate the microscopic behaviors by integrating the motions of particles or particle clusters, based on Newtonian dynamics.

Notes: it is not only for molecules; it is applicable for any particle.

Useful information to be obtained:
1. Time-dependent properties (unique compared with MC)
2. Time-independent properties (ensemble properties)
### A Brief History

#### Historical Perspective on Molecular Dynamics Simulations

<table>
<thead>
<tr>
<th>Year</th>
<th>Chemistry</th>
<th>Statistical Mechanics</th>
<th>Materials Science</th>
<th>Biochemistry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1930</td>
<td>Eyring 1936</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1940</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1950</td>
<td>Teller, 1953</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td>Nose, 1984</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1990</td>
<td>Berne, 1991</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Molecular Dynamics

A Brief History

<table>
<thead>
<tr>
<th>Field</th>
<th>Traditional Interest</th>
<th>Influence on Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chemistry</td>
<td>– accurate reaction rates; – molecular energy transfer</td>
<td>– few atoms</td>
</tr>
<tr>
<td></td>
<td>(e.g. vibration, translation)</td>
<td>– short times (vibrations)</td>
</tr>
<tr>
<td></td>
<td>– very accurate forces</td>
<td>– accurate integrators</td>
</tr>
<tr>
<td>Statistical</td>
<td>– correlated many-body motion (esp. liquids)</td>
<td>– many atoms</td>
</tr>
<tr>
<td>Mechanics</td>
<td></td>
<td>– long times (diffusion+)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– simple potentials</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– stable integrators</td>
</tr>
<tr>
<td>Materials</td>
<td>– solid-state structures (surfaces, defects)</td>
<td>– many atoms (million+)</td>
</tr>
<tr>
<td>Science</td>
<td>– many-atom dynamics (e.g. crack motion)</td>
<td>– relatively accurate forces</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(quantum-based analytic forces)</td>
</tr>
<tr>
<td>Biochemistry</td>
<td>– structure</td>
<td>– simple/accurate forces</td>
</tr>
<tr>
<td></td>
<td>– correlated motion (e.g. protein folding)</td>
<td>– quest for long times.</td>
</tr>
</tbody>
</table>

Molecular dynamics simulations, as developed by each of the historical branches, have many of the ‘nuts and bolts’ in common, but with some fundamental differences:
The world is deterministic; however complex the composites are, given the initial conditions (initial position and initial velocity), the fates thereafter are determined, following the Newtonian dynamics and the histories can also be rigorously traced.

\[ F = - \frac{dE}{dR} \]

\[ F = ma = m \frac{d^2 R}{dt^2} \]

**NEWTON'S EQUATION OF MOTION**

**INITIAL CONDITIONS**
Motion Decoupling Approximation

The Mass “m” is defined for the unit particle or particle cluster by assuming that the component particles have much smaller masses and their motions are much faster than the unit particle. This is mathematically dealt with by the mean field treatments.

\[ F = -\frac{dE}{dR} = -\frac{d\langle E \rangle_R}{dR} \]

**Note:** This practically needs to be done in the potential generation steps, for instance, in coarse grained or multi-scale simulations.

THE BORN-OPPENHEIMER APPROXIMATION

\[ H\Psi(R, r) = E\Psi(R, r) \]

\[ \Psi(R, r) = \Psi^{el}(R, r)\Psi^{n}(R) \]

**SCHRODINGER EQUATION**

**APPROXIMATE WAVEFUNCTION**
Dimension Reduction

Dimension reduction can be intentionally designed to speed up otherwise extremely expensive force calculations in the following ways.

1. Mean field treatment

\[ F = - \frac{dE}{dR} = - \frac{d\langle E \rangle_R}{dR} \]

2. Boundary condition setup
3. Interaction reduction by approximate long range interactions

Notes: Errors need to be analyzed for each approximation treatment.

Philosophy: Avoiding unnecessary errors;
Reducing unavoidable errors;
Balancing efficiency.
Motion Following

\[ F = ma = m \frac{d^2 R}{dt^2} \]

\[ F = -\frac{dE}{dR} = R \]

**Analytic solution** can be easily achieved for a low dimensional system with a simple interaction potential function \( E(R) \).

**Explicit solution** requires the propagation of position, velocity, and force employing a finite time step.

**Taylor Expansion** of the coordinate of a particle, around time \( t \),

\[ r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m} \Delta t^2 + \frac{\Delta t^3}{3!} r + o(\Delta t^4) \]
Source of Finite-Time Propagation Error

\[ r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m} \Delta t^2 + \frac{\Delta t^3}{3!} + \mathcal{O}(\Delta t^4) \]

1. Sampling Error

In order to reconstruct the continuous behavior based upon the discrete number of samples, sampling frequency (dependent on the time step) should not be much smaller than the intrinsic frequency in the sample.
Source of Finite-Time Propagation Error

\[ r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m} \Delta t^2 + \frac{\Delta t^3}{3!} r + \mathcal{O}(\Delta t^4) \]

2. Propagation Error

It can be reduced by applying advanced integration scheme, For instance, Verlet algorithm:

So:

\[ r(t - \Delta t) = r(t) - v(t)\Delta t + \frac{f(t)}{2m} \Delta t^2 - \frac{\Delta t^3}{3!} r + \mathcal{O}(\Delta t^4) \]

\[ r(t + \Delta t) = 2r(t) - r(t - \Delta t) + \frac{f(t)}{m} \Delta t^2 + \mathcal{O}(\Delta t^4) \]

Note: No velocity is used to compute the new position.
Verlet Algorithm Flow Diagram

\[
\begin{array}{ccc}
    \text{t-} \Delta t & \text{t} & \text{t+} \Delta t \\
    \mathbf{r} & \text{Given current position and position at end of previous time step} & \\
    \mathbf{v} & & \\
    \mathbf{F} & & \\
\end{array}
\]

\[r_{n+1} = 2r_n - r_{n-1} + \left(\frac{F_n}{m}\right) \Delta t^2 + O(\Delta t^4)\ldots\]
Compute the force at the current position

\[ r_{n+1} = 2r_n - r_{n-1} + \left( \frac{F_n}{m} \right) \Delta t^2 + O(\Delta t^4) \ldots \]
Verlet Algorithm Flow Diagram

Compute new position from present and previous positions, and present force

\[ r_{n+1} = 2r_n - r_{n-1} + \left( \frac{F_n}{m} \right) \Delta t^2 + O(\Delta t^4)... \]
Verlet Algorithm Flow Diagram

\[ r_{n+1} = 2r_n - r_{n-1} + \left( \frac{F_n}{m} \right) \Delta t^2 + O(\Delta t^4) \ldots \]
Molecular Dynamics

Propagation Error \( O(\Delta t^4) \)

Dependent on various factors: structure, dynamics, temperature …

Structural Feature is always dominant practically due to the existence of hard boundaries between particles.

Note: For molecular solution system, 1 fs is an appropriate time step at room temperature. It should be varied with the change of the temperature.
Molecular Dynamics

Chapter 1.1

Scale is the Intrinsic Problem in MD

Dimension Scale

\[ F = - \frac{dE}{dR} \]

Time Scale

\[ F = ma = m \frac{d^2R}{dt^2} \]

Note: Error analysis is an essential component; algorithm design based on various approximations is the other essential component.
**Macroscopic Properties: Ensemble**

MD is a sampling tool to recover macroscopic properties.

The conversion of this microscopic information to macroscopic observables such as pressure, energy, heat capacities, etc., requires statistical mechanics.
Macroscopic Properties: Ensemble

Ensemble Average is an average over all the points in the phase space.
AVERAGES

Thermodynamic (Ensemble) average: average over all points in phase space at a Single time

Dynamic average: average over a single point in phase space at All times

Ergodic hypothesis: for infinitely long trajectory, thermo average = Dynamic average

\[ \langle A \rangle_{time} = \langle A \rangle_{ensemble} \]

A large number of observations made on a single system at N arbitrary instants in time have the same statistical properties as observing N arbitrarily chosen systems at the same time.
To get a thermo average, need to know the probability of finding the system at each point (state) in phase space.

\[ \rho(r^N, p^N) = \frac{1}{Q} \exp \left[ -\frac{U(r^N, p^N)}{k_b T} \right] \]

Probability density

\[ Q = \iiint dp^N dr^N \exp \left[ -\frac{U(r^N, p^N)}{k_b T} \right] \]

Partition function

Ensemble (thermodynamic) average of observable \( A(r^N, p^N) \):

\[ \langle A \rangle_{ens} = \iiint dp^N dr^N A(r^N, p^N) \rho(r^N, p^N) \]
In an MD simulation, we determine a time average of A, which is expressed as

$$\langle A \rangle_{\text{time}} = \lim_{\tau \to \infty} \frac{1}{\tau} \int_{t=0}^{\tau} A(p^N(t),r^N(t)) dt \approx \frac{1}{M} \sum_{t=1}^{M} A(p^N,t^N)$$

where $\tau$ is the simulation time, $M$ is the number of time steps in the simulation and $A(p^N,r^N)$ is the instantaneous value of A.
Thermodynamic (equilibrium) averages can be calculated via time averaging:

\[ \langle A \rangle \approx \frac{1}{N} \sum_{i=1}^{N} A(t_i) \]

**Examples:**

\[ \langle U \rangle \approx \frac{1}{N} \sum_{i=1}^{N} U(t_i) \]

Average potential energy

\[ \langle K \rangle \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{m_j}{2} v_j(t_i) v_j(t_i) \]

Average kinetic energy
Other quantities obtained from MD: Heat Capacity

Expressed in terms of fluctuations in the energy

\[ C_v = \left( \frac{\partial E}{\partial T} \right)_{V,N} = -k\beta^2 \left( \frac{\partial^2 (\beta A)}{\partial \beta^2} \right)_{V,N} \]

\[ = -k\beta^2 \frac{\partial}{\partial \beta} \frac{1}{Q(\beta)} \int dr^N dp^N E e^{-\beta E} \]

\[ C_v = k\beta^2 \left[ \langle E^2 \rangle - \langle E \rangle^2 \right] \]
**Molecular Dynamics**

Chapter 1.1

**Pseudo-Ergodicity and Time Scale**

**Challenge:** Small time step required to ensure small motion following error conflicts with long time propagation needed for ergodicity.

**Notes:** correct ensemble averaging brings in another challenging topic in the MD algorithm developments.
Good Integrator

1. Small propagation error (allowing large time step);
2. Small velocity propagation error;
3. Time reversibility;
4. Energy conservation:

\[ E = U_{\text{pot}} + E_{\text{kinetic}} \]
Integration: Verlet Algorithm

\[ r_{n+1} = r_n + v_n \Delta t + \frac{1}{2} \left( \frac{F_n}{m} \right) \Delta t^2 + O(\Delta t^3) \]  
\[ r_{n-1} = r_n - v_n \Delta t + \frac{1}{2} \left( \frac{F_n}{m} \right) \Delta t^2 - O(\Delta t^3) \]

Sum the forward and backward expansions:

\[ r_{n+1} = 2r_n - r_{n-1} + \left( \frac{F_n}{m} \right) \Delta t^2 + O(\Delta t^4) \]

1. Use \( r_n \) to calculate \( F_n \)
2. Use \( r_n, r_{n-1} \) and \( F_n \) (step 1) to calculate \( r_{n+1} \)

Subtract the forward and backward expansions:

\[ v_n = \frac{r_{n+1} - r_{n-1}}{2\Delta t} + O(\Delta t^2) \]

Propagate velocities
Integration: Verlet Algorithm

Advantages:

1. Integration does not require the velocities, only position information is taken into account.
2. Only a single force evaluation per integration cycle. (Force evaluation is the most computationally expensive part in the simulation).
3. This formulation, which is based on forward and backward expansions, is naturally reversible in time (a property of the equation of motion).

Disadvantages:

1. The velocities, which are required for energy evaluation are calculated in an approximate manner only through the equation: \( \mathbf{v}_n = \frac{(\mathbf{r}_{n+1} - \mathbf{r}_{n-1})}{2\Delta t} \). (large errors).
2. Need to know \( \mathbf{r}_{n+1} \) to calculate \( \mathbf{v}_n \).
Integration: Leap Frog Algorithm

\[ v_{n-1/2} \equiv v\left( t - \frac{\Delta t}{2} \right) \equiv \frac{r(t) - r(t - \Delta t)}{\Delta t} = \frac{r_n - r_{n-1}}{\Delta t} \quad v_{n+1/2} \equiv v\left( t + \frac{\Delta t}{2} \right) \equiv \frac{r(t + \Delta t) - r(t)}{\Delta t} = \frac{r_{n+1} - r_n}{\Delta t} \]

Evaluate velocities at the midpoint of the position evaluations and Vice versa.

Where \( v_{n+1/2} \) is the velocity at \( t+(1/2)\Delta t \)

\[ r_{n+1} = r_n + v_{n+1/2} \Delta t \]
\[ v_{n+1/2} = v_{n-1/2} + \left( \frac{F_n}{m} \right) \Delta t \]

1. Use \( r_n \) to calculate \( F_n \)
2. Use \( F_n \) and \( v_{n-1/2} \) to calculate \( v_{n+1/2} \)
3. Use \( r_n \) and \( v_{n+1/2} \) to calculate \( r_{n+1} \)

\[ v_n = \left( \frac{v_{n+1/2} + v_{n-1/2}}{2} \right) \quad \text{Instantaneous velocity at time } t \]
Integration: Leap Frog Algorithm

**Molecular Dynamics**

Given current position, and velocity at last half-step

\[ r_{n+1} = r_n + v_{n+\frac{1}{2}} \Delta t \]
\[ v_{n+\frac{1}{2}} = v_{n-\frac{1}{2}} + \left( \frac{F_n}{m} \right) \Delta t \]
Molecular Dynamics

Integration: Leap Frog Algorithm

\[ r_{n+1} = r_n + v_{n+\frac{1}{2}} \Delta t \]

\[ v_{n+\frac{1}{2}} = v_{n-\frac{1}{2}} + \left( \frac{F_n}{m} \right) \Delta t \]
Molecular Dynamics

Integration: Leap Frog Algorithm

\[ r_{n+1} = r_n + v_{n+1/2} \Delta t \]
\[ v_{n+1/2} = v_{n-1/2} + \left( \frac{F_n}{m} \right) \Delta t \]
Molecular Dynamics

Integration: Leap Frog Algorithm

\[ r_{n+1} = r_n + v_{n+1/2} \Delta t \]

\[ v_{n+1/2} = v_{n-1/2} + \left( \frac{F_n}{m} \right) \Delta t \]
Integration: Leap Frog Algorithm

\[
\begin{align*}
    r_{n+1} & = r_n + v_{n+\frac{1}{2}}\Delta t \\
    v_{n+\frac{1}{2}} & = v_{n-\frac{1}{2}} + \left(\frac{F_n}{m}\right)\Delta t
\end{align*}
\]
Advantages:
1. Improved evaluation of velocities.
2. Direct evaluation of velocities gives a useful handle for controlling the temperature in the simulation.
3. Reduces the numerical error problem of the Verlet algorithm. Here $O(\Delta t^4)$ terms are added to $O(\Delta t^2)$ terms.

Disadvantages:
1. The velocities at time $t$ are still approximate.
2. Computationally a little more expensive than Verlet.
Integration: Velocity Verlet Algorithm

\[ r_{n+1} = r_n + v_n \Delta t + \frac{1}{2} \left( \frac{F_n}{m} \right) \Delta t^2 \]

\[ v_{n+1} = v_n + \frac{1}{2} \left( \frac{F_n}{m} + \frac{F_{n+1}}{m} \right) \Delta t^2 \]
Integration: Velocity Verlet Algorithm

Given current position, velocity, and force
Molecular Dynamics

Integration: Velocity Verlet Algorithm

Compute new position
Integration: Velocity Verlet Algorithm

Compute velocity at half step
Integration: Velocity Verlet Algorithm

Molecular Dynamics

Chapter 1.1

Compute force at new position
Molecular Dynamics

Chapter 1.1

Integration: Velocity Verlet Algorithm

\[ \begin{align*}
  & t - \Delta t & t & t + \Delta t \\
  r & & & \\
  v & & & \\
  F & & & \\
\end{align*} \]

Compute velocity at full step
Molecular Dynamics

Integration: Velocity Verlet Algorithm

- Advance to next time step, repeat.
Integration: Velocity-Corrected Verlet

\[ r(t + 2\Delta t) = r(t) + 2v(t)\Delta t + \dot{v}(t)(2\Delta t)^2 / 2! + \ddot{v}(t)(2\Delta t)^3 / 3! + \ldots \]

\[ r(t + \Delta t) = r(t) + v(t)\Delta t + \dot{v}(t)(\Delta t)^2 / 2! + \ddot{v}(t)(\Delta t)^3 / 3! + \ldots \]

\[ r(t - \Delta t) = r(t) - v(t)\Delta t + \dot{v}(t)(\Delta t)^2 / 2! - \ddot{v}(t)(\Delta t)^3 / 3! + \ldots \]

\[ r(t - 2\Delta t) = r(t) - 2v(t)\Delta t + \dot{v}(t)(2\Delta t)^2 / 2! - \ddot{v}(t)(2\Delta t)^3 / 3! + \ldots \]

\[ 12v(t)\Delta t = 8[r(t + \Delta t) - r(t - \Delta t)] - [r(t + 2\Delta t) - r(t - 2\Delta t)] \]

\[ v(t) = \frac{v(t + \Delta t / 2) - v(t - \Delta t / 2)}{2} + \frac{\Delta t}{12} [\ddot{v}(t - \Delta t) + \ddot{v}(t + \Delta t)] \]
Molecular Dynamics

Integration: High Order Velocity Treatment in Verlet-Like Algorithms

Advantages:
1. Improved evaluation of velocities.
2. Secure the energy conservation (kinetic energy portion), especially for canonical ensemble.

Disadvantages:
1. Cost in time and memory, but not problem any more.
2. Verlet in general has a larger error in spatial propagation.
Integration: High Order Spatial Scheme

-------- Gear Predictor-Corrector Algorithm

\[
    r^p(t + \Delta t) = r(t) + v(t)\Delta t + a(t)(\Delta t)^2 / 2 + b(t)(\Delta t)^3 / 6 + ...
\]

\[
    v^p(t + \Delta t) = v(t)\Delta t + a(t)(\Delta t) + b(t)(\Delta t)^2 / 2 + ...
\]

\[
    a^p(t + \Delta t) = a(t) + b(t)(\Delta t) + ...
\]

\[
    b^p(t + \Delta t) = b(t) + ...
\]
Integration: High Order Spatial Scheme

------- Gear Predictor-Corrector Algorithm

\[
\begin{align*}
r^c(t + \Delta t) &= r^p(t + \Delta t) + c0\Delta a(t + \Delta t) \\
v^c(t + \Delta t) &= v^p(t + \Delta t) + c1\Delta a(t + \Delta t) \\
a^c(t + \Delta t) &= a^p(t + \Delta t) + c2\Delta a(t + \Delta t) \\
b^c(t + \Delta t) &= b^c(t + \Delta t) + c3\Delta a(t + \Delta t) \\
\Delta a(t + \Delta t) &= a^{\text{comp}}(t + \Delta t) - a^p(t + \Delta t)
\end{align*}
\]
Lyapunov Instability

\[ r(t) = f[r(0), p(0); t] \]

\[ r'(t) = f[r(0), p(0) + \epsilon; t] \]

\[ |\Delta r(t)| \sim \epsilon \exp(\lambda t) \]