Biomolecular simulations

Patrice Koehl

Material Science

Chemistry

QUANTUM MECHANICS

Molecular Mechanics

Force Fields

Hierarchical Simulations

Meso-scale Modeling

Equilibrium & Rate Constants

Molecular Self-Assembly

Biochemistry

Chemistry

Material Science

Meso-scale Modeling

Multi-scale

Energy Minimization

Molecular dynamics

Monte Carlo methods
The two major assumptions in molecular simulations

1. Born-Oppenheimer approximation
   "the dynamics of electrons is so fast that they can be considered to react instantaneously to the motion of their nuclei"

2. Classical mechanics
   "The nuclei are treated as point particles that follow the classical laws of mechanics."

What is an atom?

- Classical mechanics: a point particle
- Defined by its position (x,y,z) and its mass
- May carry an electric charge (positive or negative), usually partial (less than an electron)
Atomic interactions

Torsion angles are 4-body.
Non-bonded pair.

Angles are 3-body.
Bonds are 2-body.

Strong valence energies

All chemical bonds:
\[ U = K (b - b_0)^2 \]

Angle between chemical bonds:
\[ U = K (\theta - \theta_0)^2 \]

Preferred conformations for torsion angles:
- \( \omega \) angle of the main chain
- \( \chi \) angles of the sidechains
  (aromatic, …)

vdW interactions

Lennard-Jones potential:
\[ U_{LJ}(r) = 4 \varepsilon \left( \left( \frac{R}{r} \right)^{12} - \left( \frac{R}{r} \right)^6 \right) \]

\[ R_h = \frac{R + R_j}{2}, \quad \varepsilon_h = \sqrt{\varepsilon_i \varepsilon_j} \]
**Electrostatics Interactions**

Coulomb potential

\[ U(r) = \frac{1}{4\pi \varepsilon_0} \frac{q_i q_j}{r} \]

**Computing Energy**

\[ E = \sum_{i,j} \frac{1}{2} K_i (b_i - b_j)^2 + \sum_{i,j,k} \frac{1}{2} K_i (\theta_i - \theta_j\theta_k) + \sum_{i,j,k} K_i \sin(\phi_i) \]

**Solvent**

Explicit or Implicit?
The SA model

Surface area potential

\[ W_{sp} = W_{cav} + W_{vdW} = \sum_{i=1}^{\infty} \sigma_i S_{A_i} \]

Hydrophobic potential: Surface Area, or Volume?


For proteins and other large bio-molecules, use surface.
Sphere Representations in Biology

DNA
Nucleosome
Viral DNA
Chromosome arrangements

Measuring a Union of Balls

Measuring a Union of Balls
Measuring a Union of Balls

Algorithm for computing Delaunay triangulation:

Input: $N$: number of points
       $C_i$: position of point $i$

1) Randomize points
2) For $i = 1:N$
   - Location: find tetrahedra $t$ that don’t contain $C_i$
     - Addition: divide $t$ into 4 tetrahedra
     - Correct: flip non-local tetrahedra

Output: list of tetrahedra

Measuring a Union of Balls

Compute Voronoi diagram from Delaunay complex: dual

Measuring a Union of Balls

Restrict Voronoi diagram to the Union of Balls:
Power diagram
**Measuring a Union of Balls**

Atom $i$:
Fraction in Voronoi cell: $\sigma_i$ and $\beta_i$

\[ A_i = 4\pi \sum_{j=1}^{N} r_i^2 \sigma_j \]

\[ V_i = \frac{4\pi}{3} \sum_{j=1}^{N} r_i^3 \beta_j \]

**Measuring a Union of Balls**

\[ A = S_i - S_{i,j} - S_{i,k} - S_{i,l} + S_{i,j} + S_{i,k} + S_{i,l} - S_{i,j,k,l} \]
Measuring Union of Balls

Applications to drug design

Binding pockets in 16S ribosomal RNA

PDB structure: 1HZN

HIV protease (3MKE)
Main cavity
Actual position of K54 (inhibitor)

Hygromycin B
**Computing energy**

Bonded interactions are local, and therefore their computation has a linear computational complexity (\(O(N)\)), where \(N\) is the number of atoms in the molecule considered.

The direct computation of the non-bonded interactions involve all pairs of atoms and has a quadratic complexity (\(O(N^2)\)). This can be prohibitive for large molecules.

\[
U_{NB} = \sum_{i,j,\text{nonbonded}} \left( \frac{R_i}{r_0} \right)^{12} - 2 \left( \frac{R_i}{r_0} \right)^6 + \sum_{i,j,\text{nonbonded}} \frac{q_i q_j}{4 \pi \varepsilon_0 r_{ij}}
\]

Cutoff schemes for faster energy computation

\[
U_{NB} = \sum_{i,j} \omega_{ij} S(r_{ij}) \left[ \left( \frac{R_i}{r_0} \right)^{12} - 2 \left( \frac{R_i}{r_0} \right)^6 \right] + \sum_{i,j} \omega_{ij} S(r_{ij}) \frac{q_i q_j}{4 \pi \varepsilon_0 r_{ij}}
\]

- \(\omega_{ij}\): weights (0 < \(\omega_{ij}\) < 1). Can be used to exclude bonded terms, or to scale some interactions (usually 1-4)
- \(S(r)\): cutoff function

Three types:

1) Truncation: \(S(r) = \begin{cases} 1 & r < b \\ 0 & r \geq b \end{cases}\)

2) Smooth cutoff

3) Other methods
2. Switching

\[ S(r) = \begin{cases} \frac{1}{1 + y(r)^2} & r < a \\ a & a \leq r < b \\ 0 & r \geq b \end{cases} \]

with \[ y(r) = \frac{r - a}{b - a} \]

3. Shifting

\[ S'_c(r) = \begin{cases} \left[ 1 - \left( \frac{r - a}{b - a} \right)^2 \right]^\gamma & r < c \\ 0 & c \leq r < b \end{cases} \]

or

\[ S'_c(r) = \begin{cases} \left[ 1 - \frac{r - a}{b - a} \right]^\gamma & r < c \\ 0 & c \leq r < b \end{cases} \]

---

**Units in Molecular Simulations**

Most force fields use the AKMA (Angstrom – Kcal – Mol – Atomic Mass Unit) unit system:

<table>
<thead>
<tr>
<th>Quantity</th>
<th>AKMA unit</th>
<th>Equivalent SI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>1 Kcal/Mol</td>
<td>4184 Joules</td>
</tr>
<tr>
<td>Length</td>
<td>1 Angstrom</td>
<td>(10^{-10}) meter</td>
</tr>
<tr>
<td>Mass</td>
<td>1 amu (H=1amu)</td>
<td>1.6605655 (10^{-27}) Kg</td>
</tr>
<tr>
<td>Charge</td>
<td>1 e</td>
<td>1.6021892 (10^{-19}) C</td>
</tr>
<tr>
<td>Time</td>
<td>1 unit</td>
<td>(4.88882 \times 10^{-14}) second</td>
</tr>
<tr>
<td>Frequency</td>
<td>1 cm(^{-1})</td>
<td>(18.836 \times 10^{16}) rd/s</td>
</tr>
</tbody>
</table>

---

**Some Common force fields in Computational Biology**

- **ENCAD** (Michael Levitt, Stanford)
- **AMBER** (Peter Kollman, UCSF; David Case, Scripps)
- **CHARMM** (Martin Karplus, Harvard)
- **OPLS** (Bill Jorgensen, Yale)
- **MM2/MM3/MM4** (Norman Allinger, U. Georgia)
- **ECEPP** (Harold Scheraga, Cornell)
- **GROMOS** (van Gunsteren, ETH, Zurich)

Biomolecular Simulations

- Molecular Mechanics force fields
- Energy Minimization
- Molecular dynamics
- Monte Carlo methods

Computing energy

\[ U = \sum_{i} \frac{1}{2} K \{ b - b_i \}^2 + \sum_{i} \frac{1}{2} K \{ \theta - \theta_i \}^2 + \sum K_i \{ 1 - \cos(\phi) \} \]
\[ + \sum \epsilon \left( \frac{R_i}{\rho_i} \right)^2 - 2 \frac{R_i}{\rho_i} \]
\[ + \sum \frac{0.4}{4\pi \epsilon_0} \frac{q_i q_j}{\beta} \]

\( U \) is a function of the conformation \( C \) of the protein. The problem of minimizing \( U \) can be stated as finding \( C \) such that \( U(C) \) is minimum.

The minimizers

Minimization of a multi-variable function is usually an iterative process, in which updates of the state variable \( x \) are computed using the gradient, and in some (favorable) cases the Hessian.

Iterations are stopped either when the maximum number of steps (user’s input) is reached, or when the gradient norm is below a given threshold.

Steepest descent (SD):

The simplest iteration scheme consists of following the “steepest descent” direction:

\[ x_{k+1} = x_k - \alpha \nabla f(x_k) \]

(\( \alpha \) sets the minimum along the line defined by the gradient)

Usually, SD methods lead to improvement quickly, but then exhibit slow progress toward a solution.

They are commonly recommended for initial minimization iterations, when the starting function and gradient-norm values are very large.
Conjugate gradients (CG):
In each step of conjugate gradient methods, a search vector $p_k$ is defined by a recursive formula:

$$p_{k+1} = -\nabla f(x_k) + \beta_k p_k$$

The corresponding new position is found by line minimization along $p_k$:

$$x_{k+1} = x_k + \lambda_k p_k$$

The CG methods differ in their definition of $\beta_k$.

Newton’s methods:
Newton’s method is a popular iterative method for finding the 0 of a one-dimensional function:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

It can be adapted to the minimization of a one-dimensional function, in which case the iteration formula is:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Several implementations of Newton’s method exist: quasi-Newton, truncated Newton, “adopted-basis Newton-Raphson” (ABNR),…

Biomolecular Simulations

- Molecular Mechanics force fields
- Energy Minimization
- Molecular dynamics
- Monte Carlo methods
What is a molecular dynamics simulation?

- Simulation that shows how the atoms in the system move with time
- Typically on the nanosecond timescale
- Atoms are treated like hard balls, and their motions are described by Newton’s laws.

Characteristic protein motions

<table>
<thead>
<tr>
<th>Type of motion</th>
<th>Timescale</th>
<th>Amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bond stretching</td>
<td>0.01 ps</td>
<td>&lt; 1 Å</td>
</tr>
<tr>
<td>angle bending</td>
<td>0.1 ps</td>
<td></td>
</tr>
<tr>
<td>methyl rotation</td>
<td>1 ps</td>
<td></td>
</tr>
<tr>
<td>Medium scale</td>
<td></td>
<td></td>
</tr>
<tr>
<td>loop motions</td>
<td>ns – µs</td>
<td>1-5 Å</td>
</tr>
<tr>
<td>SSE formation</td>
<td>ns – µs</td>
<td></td>
</tr>
<tr>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>protein tumbling</td>
<td>20 ns</td>
<td>&gt; 5 Å</td>
</tr>
<tr>
<td>(water tumbling)</td>
<td>(20 ps)</td>
<td></td>
</tr>
<tr>
<td>protein folding</td>
<td>ms – hrs</td>
<td></td>
</tr>
</tbody>
</table>

Why MD simulations?

- Link physics, chemistry and biology
- Model phenomena that cannot be observed experimentally
- Understand protein folding...
- Access to thermodynamics quantities (free energies, binding energies,...)
How do we run a MD simulation?

- Get the initial configuration
  From x-ray crystallography or NMR spectroscopy (PDB)

- Assign initial velocities
  At thermal equilibrium, the expected value of the kinetic energy of the system at temperature $T$ is:
  \[
  \langle E_{\text{kin}} \rangle = \frac{1}{2} \sum_{i=1}^{N} m_i v_i^2 = \frac{1}{2} (3N) k_B T
  \]

  This can be obtained by assigning the velocity components $v_i$ from a random Gaussian distribution with mean 0 and standard deviation $(k_B T/m_i)$:
  \[
  \langle v_i^2 \rangle = \frac{k_B T}{m_i}
  \]

- For each time step:
  - Compute the force on each atom:
    \[
    F(X) = -\nabla E(X) = -\frac{\partial E}{\partial X}
    \]
  - Solve Newton’s 2nd law of motion for each atom, to get new coordinates and velocities
    \[
    M \ddot{X} = F(X)
    \]
  - Store coordinates

Newton’s equation cannot be solved analytically:
Use stepwise numerical integration

MD as a tool for minimization

Molecular dynamics uses thermal energy to explore the energy surface

Energy minimization stops at local minima
The actual transition time from A to B is very quick (a few pico seconds). What takes time is waiting. The average waiting time for going from A to B can be expressed as:

\[ \tau_{A \rightarrow B} = C e^{\frac{\Delta G}{kT}} \]

### Biomolecular Simulations

- Molecular Mechanics force fields
- Energy Minimization
- Molecular dynamics
- Monte Carlo methods

### Monte Carlo: random sampling

A simple example:

Evaluate numerically the one-dimensional integral:

\[ I = \int_a^b f(x) \, dx \]

Instead of using classical quadrature, the integral can be rewritten as

\[ I = (b - a) \langle f(x) \rangle \]

\( \langle f(x) \rangle \) denotes the unweighted average of \( f(x) \) over \([a,b]\), and can be determined by evaluating \( f(x) \) at a large number of \( x \) values randomly distributed over \([a,b]\).
A famous example: Buffon’s needle problem

The probability that a needle of length $L$ overlaps with one of the lines, distant from each other by $D$, with $L \leq D$ is:

$$P = \frac{2L}{\pi D}$$

For $L = D$:

$$P = \frac{2}{\pi}$$

Method to estimate $\pi$ numerically:

“Throw” $N$ needles on the floor, find needles that cross one of the line (say $C$ of them). An estimate of $\pi$ is:

$$\pi = \frac{2N}{C}$$

Buffon, G. Editor's note concerning a lecture given by Mr. Le Clerc de Buffon to the Royal Academy of Sciences in Paris. Académie des Sciences, pp. 43-45, 1733.

Buffon, G. "Essai d'arithmétique morale." Histoire naturelle, générale et particulière, Supplément 4, 46-123, 1777

Monte Carlo Sampling for protein structure

The probability of finding a protein (biomolecule) with a total energy $E(X)$ is:

$$P(X) = \frac{\exp(-\frac{E(X)}{kT})}{\sum \exp(-\frac{E(Y)}{kT})}$$

Partition function

Estimates of any average quantity of the form:

$$\langle A \rangle = \int A(X)P(X)dX$$

using uniform sampling would therefore be extremely inefficient.

Monte Carlo for sampling conformations

The Metropolis Monte Carlo algorithm:

1. Select a conformation $X$ at random. Compute its energy $E(X)$
2. Generate a new trial conformation $Y$. Compute its energy $E(Y)$
3. Accept the move from $X$ to $Y$ with probability:

$$P = \min(1, \exp\left(-\frac{E(Y) - E(X)}{kT}\right))$$

Pick a random number $RN$, uniform in [0,1].

If $RN < P$: accept the move.
4. Repeat 2 and 3.