Review

- Fundamentals
  - units, properties, statistical mechanics

- Monte Carlo and molecular dynamics as applied to atomic systems
  - simulating in various ensembles
  - biasing methods for MC

- Molecular models for realistic (multiatomic) systems
  - inter- and intra- atomic potentials
  - electrostatics

- Now examine differences between simulations of monatomic and multiatomic molecules
Truncating the Potential

- Many molecular models employ point charges for electrostatic interactions.
- Potential-truncation schemes must be careful not to split the charges.
- For a 9Å truncation distance, using water-like charges, the interaction energy for a molecule with bare charge is (huge).
- Always use cutoff based on molecule separation, not atom.
  - *for large molecules, OK to split molecule but do not split subgroups*
Volume-Scaling Moves

Scaling atom displacements leads to large strain on intramolecular bonds

Instead perform volume scaling moves using molecule centers-of-mass (or something similar)

- Let $R_i$ be COM of molecule $i$
  
  $$R_i = \sum_{\text{atoms on } i} m_i r_j^{(i)}$$

- $q_j^{(i)}$ be position of atom $j$ w.r.t. $R_i$
  
  $$q_j^{(i)} = r_j^{(i)} - R_i$$

- For size scaling $s$, $L^{\text{new}} = sL^{\text{old}}$

  $$r_j^{(i)}(\text{new}) = sR_i + q_j^{(i)}$$

- Acceptance based on change in $-\beta(U + PV) + N_m \ln V$ (not atoms)
Rigid vs. Nonrigid Molecules

- MC and MD can be performed on molecules as already described
  - *MD moves advance atom positions based on current forces*
  - *MC moves translate atoms and accepts based on energy change*
  - *both are done considering inter- and intra-molecular forces*
  - *limiting distribution has same form*
    \[
    \pi = \frac{1}{h^{3N}} dp^N dr^N e^{-\beta \sum p_i^2 / 2m_i} e^{-\beta U(r^N)} \quad N = \text{Number of atoms}
    \]
  - *if this is all that is done, there is nothing more to say*

- Often it is much more efficient to use a rigid-bond model
  - *MD integration then doesn’t have to deal with fast intramolecular dynamics, so a larger time step can be used*
  - *MC can sample configurations more efficiently using rigid-body moves (even if model does not have rigid bonds)*
    but much care is needed to do this properly
Molecule Coordinate Frame

- Molecule-frame coordinates are defined w.r.t. molecule COM with molecule in a reference orientation.

- Simulation-frame coordinate is determined by molecule COM $R$ and orientation $\omega$.

- For rigid molecules, the molecule-frame coordinates never change.
Orientation 1.

- Orientation described in terms of rotation of molecule frame
- Direction cosines can be used to describe rotation
  \[ \alpha_1 = e'_x \cdot e_x, \quad \alpha_2 = e'_z \cdot e_y \]
  \[ \beta_1 = e'_y \cdot e_x, \quad \beta_2 = e'_y \cdot e_y \]
- Relation between same point in two frames
  \[ \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \]
- Rotation matrix
  \[ A = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{pmatrix} \]
  \[ r' = Ar \]
- Kinematics of rigid-molecule rotation described in terms of rotation of the molecule coordinate frame (i.e. the direction cosines)
  - \( r' \) never changes in a rigid molecule
Orientation 2.

- We also need to invert the relation
  - get the simulation-frame coordinate from the molecule frame value
    \[ r = A^{-1} r' = A^T r' \]

- Direction cosines are not independent
  - in 2D, all can be described by just one parameter
  - use rotation angle \( \theta \)
    \[
    A = \begin{pmatrix}
    \cos \theta & \sin \theta \\
    -\sin \theta & \cos \theta
    \end{pmatrix}, \quad A^{-1} = \begin{pmatrix}
    \cos \theta & -\sin \theta \\
    \sin \theta & \cos \theta
    \end{pmatrix}
    \]
    - inverse can be viewed as replacing \( \theta \) with \(-\theta\)
Euler Angles

- The picture in 3D is similar: \((x, y, z) \rightarrow (x', y', z')\)
- Nine direction cosines
- Three independent coordinates specify orientation
- Euler angles are the conventional choice \(\omega = \phi \theta \psi\)
  - three rotations give the simulation-frame orientation
3D Rotation Matrix

- Rotation matrix expressed in terms of Euler angles \( r' = Ar \)

\[
A = \begin{pmatrix}
\cos \psi \cos \phi - \cos \theta \sin \phi \sin \psi & \cos \psi \sin \phi + \cos \theta \cos \phi \sin \psi & \sin \psi \sin \theta \\
-\sin \psi \cos \phi - \cos \theta \sin \phi \cos \psi & -\sin \psi \sin \phi + \cos \theta \cos \phi \cos \psi & \cos \psi \sin \theta \\
\sin \theta \sin \phi & -\sin \theta \cos \phi & \cos \theta
\end{pmatrix}
\]

- To get space-fixed coordinate, multiply molecule-fixed vector by \( A^{-1} \)
  - again, \( A^{-1} = A^T \) 
  - \( r = A^T r' \)
Consider a simple diatomic

- positions of two atoms described by
  \[ x_1, y_1, z_1, x_2, y_2, z_2 \]

- can instead describe by molecule COM and stretch/orientation coordinates
  \[ X, Y, Z, L, \theta, \phi \]

- in molecule frame, each atom position is given by
  \[ r'_1 = Le_z \]
  \[ r'_2 = -Le_z \]
To get space-fixed coordinates, use rotation matrix

\[ r_1 = R + A^T r'_1 \]
\[ r_2 = R + A^T r'_2 \]
\[ r'_1 = L e_z \]
\[ r'_2 = -L e_z \]

- **matrix**

\[
A^T = \begin{pmatrix}
\cos \phi & \sin \phi & \sin \theta \sin \phi \\
-\cos \theta \sin \phi & \cos \theta \cos \phi & -\sin \theta \cos \phi \\
0 & \sin \theta & \cos \theta \\
\end{pmatrix}
\]

- **the result is**

\[
x_1 = X + L \sin \theta \sin \phi \quad x_2 = X - L \sin \theta \sin \phi \\
y_1 = Y - L \sin \theta \cos \phi \quad y_2 = Y + L \sin \theta \cos \phi \\
z_1 = Z + L \cos \theta \quad z_2 = Z - L \cos \theta
\]
Transforming Coordinates 3.

- The ensemble distribution for the transformed coordinates is obtained via the Jacobian

\[
\pi = \frac{1}{Qh^3} dp^N dr^N e^{-\beta \sum p_i^2 / 2m_i} e^{-\beta U(r^N)}
\]

\[
= \frac{1}{Qh^3} dp^N dq^N |J| e^{-\beta \sum p_i^2 / 2m_i} e^{-\beta U(q^N)}
\]

- **the elements of \( J \) are the derivatives**

\[
J_{\alpha\beta} = \frac{\partial r_\alpha}{\partial q_\beta}
\]

\[
\begin{pmatrix}
X & Y & Z & L & \phi & \theta \\
x_1 & 1 & 0 & 0 & \sin \theta \sin \phi & L \sin \theta \cos \phi & L \cos \theta \sin \phi \\
y_1 & 0 & 1 & 0 & -\sin \theta \cos \phi & L \sin \theta \sin \phi & -L \cos \theta \cos \phi \\
z_1 & 0 & 0 & 1 & \cos \theta & 0 & -L \sin \theta \\
x_2 & 1 & 0 & 0 & -\sin \theta \sin \phi & -L \sin \theta \cos \phi & -L \cos \theta \sin \phi \\
y_2 & 0 & 1 & 0 & \sin \theta \cos \phi & -L \sin \theta \sin \phi & L \cos \theta \sin \phi \\
z_2 & 0 & 0 & 1 & -\cos \theta & 0 & L \sin \theta
\end{pmatrix}
\]

- **For this transformation**

\[
|J| = 8L^2 \sin \theta
\]

- But we also need to transform the momenta

\[
x_1 = X + L \sin \theta \sin \phi \\
y_1 = Y - L \sin \theta \cos \phi \\
z_1 = Z + L \cos \theta \\
x_2 = X - L \sin \theta \sin \phi \\
y_2 = Y + L \sin \theta \cos \phi \\
z_2 = Z - L \cos \theta
Transforming Coordinates 4.

○ Begin with the Lagrangian

• in the original coordinate system

\[ L = K - U \]

\[ = \frac{1}{2} (\dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2 + \dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2) + U(x_1, y_1, z_1, x_2, y_2, z_2) \]

\[ = \frac{1}{2} \sum_i \dot{r}_i^2 + U(r) \]

• transform to new coordinates

\[ \dot{x}_1 = \frac{\partial x_1}{\partial X} \dot{X} + \frac{\partial x_1}{\partial Y} \dot{Y} + \frac{\partial x_1}{\partial Z} \dot{Z} + \frac{\partial x_1}{\partial L} \dot{L} + \frac{\partial x_1}{\partial \theta} \dot{\theta} + \frac{\partial x_1}{\partial \phi} \dot{\phi} \]

\[ \dot{y}_1 = \text{etc.} \]

• in general

\[ \dot{r}_i = \sum_{\alpha} \frac{\partial r_i}{\partial q_\alpha} \dot{q}_\alpha \]

\[ \dot{r}_i^2 = \sum_\beta \sum_{\alpha} \frac{\partial r_i}{\partial q_\alpha} \frac{\partial r_i}{\partial q_\beta} \dot{q}_\alpha \dot{q}_\beta \]

\[ = \mathbf{q} \cdot \mathbf{g}_i \cdot \mathbf{q} \]

\[ \sum_i \dot{r}_i^2 = \dot{\mathbf{q}} \cdot \mathbf{G} \cdot \dot{\mathbf{q}} \]
Transforming Coordinates 5.

- **Derive momenta**
  
  \[ L = \frac{1}{2} \mathbf{q} \cdot G \cdot \dot{\mathbf{q}} - U(\mathbf{q}) \]
  
  \[ p_\alpha \equiv \frac{\partial L}{\partial \dot{q}_\alpha} = \sum_\beta G_{\alpha \beta} \dot{q}_\beta \]
  
  \[ \mathbf{p} = G\dot{\mathbf{q}} \]
  
  \[ \dot{\mathbf{q}} = G^{-1}\mathbf{p} \]
  
  \[ \mathbf{q} \cdot G \cdot \dot{\mathbf{q}} = \mathbf{p} \cdot G^{-1} \cdot \mathbf{p} \]
  
  Uses \( G \) and thus \( G^{-1} \) are symmetric

- **The Hamiltonian is**
  
  \[ H = K(\mathbf{p}, \mathbf{q}) + U(\mathbf{q}) \]
  
  \[ = \frac{1}{2} \mathbf{p} \cdot G^{-1} \cdot \mathbf{p} + U(\mathbf{q}) \]

- **The Jacobian for the momentum transformation is the reciprocal of the Jacobian for the coordinate transformation**

  \[ \pi = \frac{1}{Qh^3N} d\mathbf{p}_q^N dq^N \left| J \left| J^{-1} e^{-\frac{1}{2} \mathbf{p}_q \cdot G^{-1} \mathbf{p}_q} e^{-\beta U(q^N)} \right| \right. \]

  - we don’t have to worry about the Jacobian with the full transform
Integrating Over Momenta

☐ If we integrate out the momentum coordinates, the Jacobian again arises

\[ \pi(p, q) = \frac{1}{Qh^{3N}} dp^N dq^N e^{-\frac{1}{2} \beta \mathbf{p} \cdot \mathbf{G}^{-1} \cdot \mathbf{p}} e^{-\beta U(q^N)} \]

\[ \pi(q) = \frac{1}{Qh^{3N}} dq^N e^{-\beta U(q^N)} \int dp^N e^{-\frac{1}{2} \beta \mathbf{p} \cdot \mathbf{G}^{-1} \cdot \mathbf{p}} \]

\[ = \frac{1}{Qh^{3N}} dq^N e^{-\beta U(q^N)} |G|^{1/2} \]

☐ For the diatomic, this term is

\[ |G|^{1/2} = |J| = c L^2 \sin \theta \quad \text{c = constant} \]

☐ In MC simulation, the terms must be included in the construction of the transition-probability matrix
Averages with Constraints 1.

- Very stiff coordinates are sometimes treated as rigidly constrained
  - e.g., the bond length $L$ in the diatomic may be held at a constant value
- MC and MD have different ways to enforce this constraint
- Regardless of simulation technique, the constrained-system ensemble average may differ from the unconstrained value
  - even when compared to the limit of an infinitely stiff bond!
- Why the difference?
  - A rigid constraint implies no kinetic energy in vibration
  - Examine the Lagrangian

\[
L = \frac{1}{2} \sum_{\beta \neq L} \sum_{\alpha \neq L} \frac{\partial r_i}{\partial q_\alpha} \frac{\partial r_i}{\partial q_\beta} q_\alpha q_\beta - U(q^S; L)
\]

$s =$ “soft” coordinate

\[
= \frac{1}{2} \dot{q}^S \cdot G^S \cdot \dot{q}^S - U(q^S; L)
\]
Averages with Constraints 2.

- The Jacobian for the coordinate transform is the same as for the unconstrained average.
- But the momentum Jacobian no longer has the term for the constrained coordinate.
- Thus, in general, the distribution of unconstrained coordinates differs

\[
\pi(p_s, q_s; L) = \frac{1}{Qh^{3N-l}} dp_s^{N-l} dq_s^{N-l} e^{-\frac{1}{2}p_s \cdot G_s^{-1} \cdot p_s} e^{-\beta U(q_s^N)}
\]

\[
\pi(q_s; L) = \frac{1}{Qh^{3N-l}} dq_s^{N-l} e^{-\beta U(q_s^N)} \int dp_s^{N-l} e^{-\frac{1}{2}p_s \cdot G^{-1} \cdot p}
\]

\[
= \frac{1}{Qh^{3N-l}} dq_s^{N-l} e^{-\beta U(q_s^N)} \left| G_s \right|^{1/2}
\]

- The difference is

\[
\frac{\pi(q_s)}{\pi(q_s; L)} = \sqrt{\frac{|G_s|}{|G|}}
\]
Averages with Constraints 3.

- To get correct (unconstrained-system) averages from a simulation using constraints, averages should be multiplied by this factor

\[ \langle M \rangle_{\text{unconstrained}} = \left\langle M \sqrt{\frac{|G_s|}{|G|}} \right\rangle_{\text{constrained}} \]

- Evaluating this quantity could be tedious
  - but there is a simplification
  - the ratio of determinants (of N-by-N and (N-l)-by-(N-l) matrices) can be given in terms of the determinant of an l-by-l matrix

\[ \frac{|G_s|}{|G|} = |H| \quad H_{\alpha \beta} = \sum_i \frac{\partial \sigma_{a}}{\partial r_i} \frac{\partial \sigma_{\beta}}{\partial r_i} \]

- for the diatomic with L constrained, \( H = 1 \)
Rotational Dynamics

- For completely rigid molecules, only translation and rotation are performed.
- Translational dynamics uses methods described previously, but now applied to the COM.
- Rotational dynamics must consider angular velocities and accelerations.
- Can treat via rotation of the molecule-frame coordinates in the spaced-fixed frame.

\[ \dot{e}^s = \omega \times e^s \]

- Changes in angular velocity are given via torque on molecule.
Quaternions

- Rate of change of the Euler angles looks like this
  \[ \dot{\phi} = -\omega_x \frac{\sin \phi \cos \theta}{\sin \theta} + \omega_y \frac{\cos \phi \cos \theta}{\sin \theta} + \omega_z \]

- A problem arises when \( \theta \) is near 0
  - *no physical significance, but very inconvenient to integration of equations of motion*

- Quaternions can be used to circumvent the problem
  - *describe orientation with 4 (non-independent) variables*
  - *rotation matrix, equations of motion simply expressed in terms of these quantities*
  - *note: \( q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \)*

  \[
  q_0 = \cos \frac{\theta}{2} \cos \frac{\phi + \psi}{2} \\
  q_1 = \sin \frac{\theta}{2} \cos \frac{\phi - \psi}{2} \\
  q_2 = \sin \frac{\theta}{2} \sin \frac{\phi - \psi}{2} \\
  q_3 = \cos \frac{\theta}{2} \sin \frac{\phi + \psi}{2}
  \]
Monte Carlo Rotations

- MC simulations of molecules include rotation moves
  - *must do this to sample orientations of rigid molecules*
  - *not strictly necessary for non-rigid molecules, but very helpful*
  - *very easy to do this incorrectly*

- Trial rotation of a linear molecule
  - *Let present orientation be given by vector* $\mathbf{u}$
  - *Generate a unit vector* $\mathbf{v}$ *with random orientation*
  - *Let new trial orientation be given by*
    \[ \mathbf{u}^{\text{new}} = \mathbf{u}^{\text{old}} + \gamma \mathbf{v} \]
    *where* $\gamma$ *is a fixed scale factor that sets the size of the perturbation*

- Nonlinear molecule
  - *same procedure, but do perturbation on the 4-dimensional vector of quaternions*
Random Vector on a Sphere

- Acceptance-rejection method of von Neumann
- Iterate
  1. Generate 3 uniform random variates, $r_1$, $r_2$, $r_3$ on $(0,1)$
  2. Calculate $z_i = 1 - 2r_i$, $i = 1, 3$, so that the vector $z$ is distributed uniformly in a cube of side 2, centered on the origin
  3. Form the sum $z^2 = z_1^2 + z_2^2 + z_3^2$
  4. If $z^2 < 1$, take the random vector as $(z_1/z, z_2/z, z_3/z)$ and quit
  5. Otherwise, reject the vector and return to (a)

- Alternative algorithms are possible