

CHEM 101.6
Dartmouth College
Paul Robustelli

Lecture 1
1/11/22

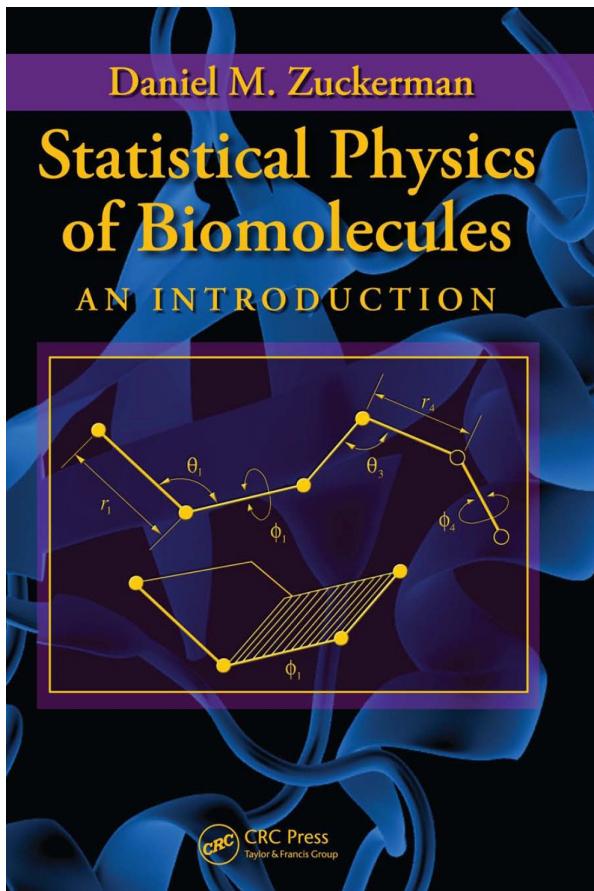
CHEM 101.6 Course Objectives

- Understand the basic principles of molecular modeling – with a focus on molecular simulations.
 - Force Fields, Molecular Dynamics, Monte Carlo Simulations, Simulation Statistics & Convergence, Enhanced Sampling Methods, Quantum Calculations of Small Molecule Energetics
- Learn to run and analyze the results of molecular simulations in a modern python environment
 - Python, jupyter notebooks, github, matplotlib, openMM, MDtraj, MDanalysis, and other python packages
- Run and analyze simulations of small molecules, proteins, and polymers
- Propose an independent molecular simulation or computational chemistry research question, use the basics learned in this course to investigate it, and present the background, theory, and results to the class
- Discover and obtain meaningful results from a software package I know nothing about.

CHEM 101.6 Textbooks

Primary Textbook

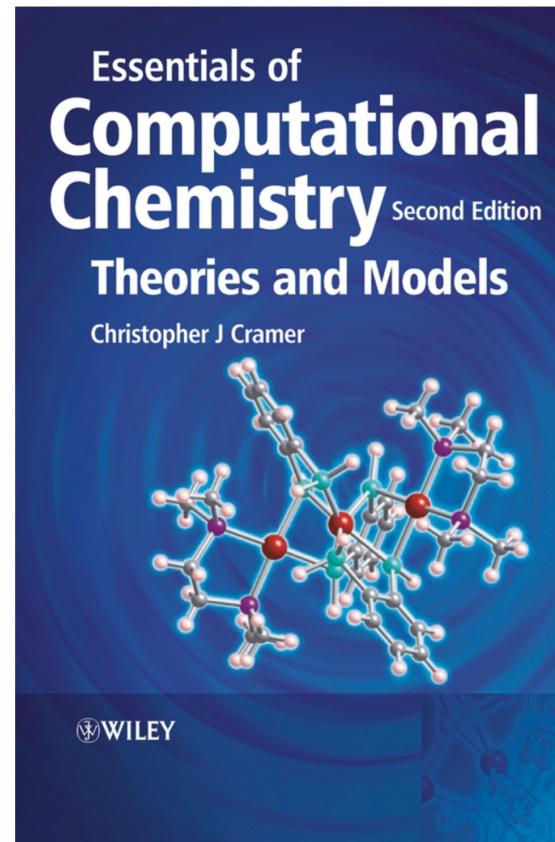
- Basic Principles of Statistical Physics and Probability Needed to Conceptually Understand Molecular Simulations (No Quantum Mechanics or Orbitals)



https://www.amazon.com/Statistical-Physics-Biomolecules-Daniel-Zuckerman-ebook-dp-B005H6YEBI/dp/B005H6YEBI/ref=mt_other?_encoding=UTF8&me=&qid=

Supplementary Textbook

- Introductory Computational Chemistry (Lots of Quantum Mechanics & Orbitals)



https://www.amazon.com/Essentials-Computational-Chemistry-Theories-Models-ebook-dp-B000W3YPRY/dp/B000W3YPRY/ref=mt_other?_encoding=UTF8&me=&qid=

Cramer Lectures:

https://www.youtube.com/playlist?list=PLkNVwyLvX_TFBLHCvApmvafqqQUHb6JwF

CHEM 101.6 (Tentative) Schedule

Week	Reading	Lecture	Project
1			<ul style="list-style-type: none">Set Up Course Infrastructure, run alkane simulations
2	Ch 1-2	Molecular Mechanics, Molecular Dynamics, Basic Probability Theory	<ul style="list-style-type: none">Analyze a butane md trajectoryModify a butane force field and compare simulation results
3	Ch 3-4	Protein Structure & Dynamics, Equilibrium, Dynamics, Time Correlations	<ul style="list-style-type: none">Run, visualize and analyze a protein MD simulationCalculate Experimental Data from a SimulationBasics of VMD (trajectory visualization)
4	Ch 5-6	Protein Folding & Disordered Proteins, Partition Functions, PMFs, Free Energy Surfaces	<ul style="list-style-type: none">Analyze protein folding and disordered protein simulations
5	Ch 7	Basic Thermodynamics, Reaction Coordinates, Enhanced Sampling Methods	<ul style="list-style-type: none">Run & Analyze replica exchange and metadynamics simulations
6	Ch 8	Properties of Water, Water Simulations, Polymer Simulations	<ul style="list-style-type: none">Running water simulations with different force fieldsAnalyze water behavior in explicit solvent protein/polymer simulations
7	CH 9	Ligand Binding, Alchemical Free Energy Calculations, Drug Design	<ul style="list-style-type: none">Analyze Ligand Binding Simulations, Free Energy Calculations
8		Quantum Chemistry Calculations, Neural Network Force Fields, Machine Learning Molecular Properties	<ul style="list-style-type: none">Small Molecule Geometry Optimizations, Fitting a Small Molecule Force Field
9/10		Final Project Student Presentations	<ul style="list-style-type: none">Final Project Student Presentations

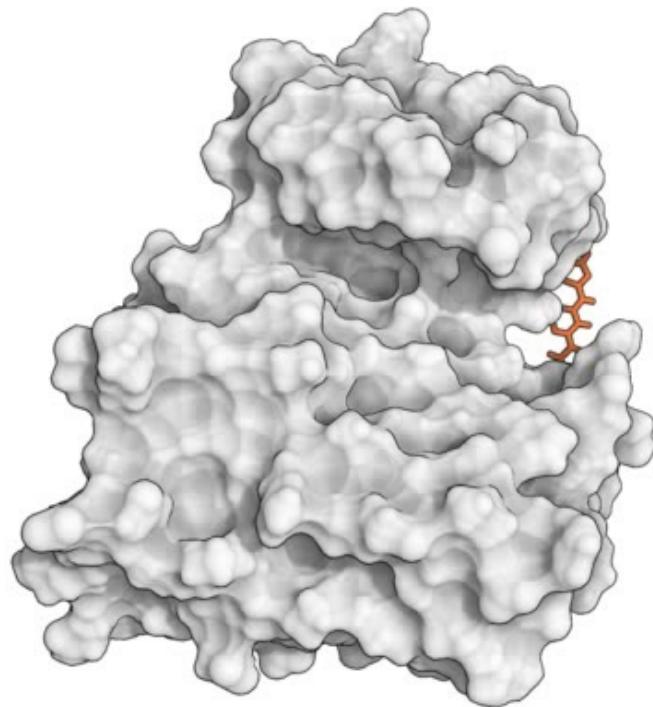
CHEM 101.6 Evaluations

- Weekly Projects: 50%
 - Upload Jupyter or collab notebooks to your CHEM101.6 github page by 5PM on Monday. Upload what you have by class on Tuesday, and you can update it later if you need to keep working on it
- Course Participation / Effort: 20%
 - Do you engage with the reading, discussions during lectures & labs, are you active on the slack channel? Are you demonstrating effort to make independent progress on your projects. Do you explore ideas independently?
- Final Project + Presentation : 30 %
 - You will conceive of a tractable research question and attempt to answer it with simulations and/or calculations. Some component of this should involve the use of a software package that we have not previously used in class. You will prepare a ~20-30 minute presentation on your research question, methods, and results to the class in weeks 9+10.

What's the Point of Molecular Modeling?

- Why do you want to study molecular modeling?
- What do scientists in your field(s) of interest use molecular modeling for?
- What would do you hope to accomplish with molecular modeling in your studies/career?

Weekly Conceptual Check-In: Unbiased Dasatinib:Src Kinase Binding MD Simulation



Each week, along with posting your weekly projects in your CHEM101.6 github repo, I'll ask everyone to write a paragraph discussing this simulation from the point of few of the concepts from the previous week's reading, lecture, and project.

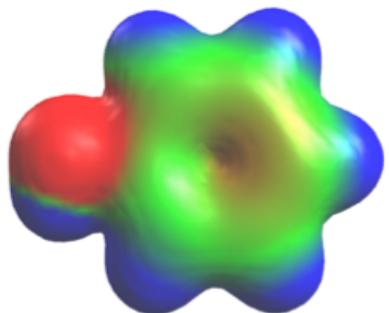
Each week you will:

- Discuss how concepts covered in class in the previous week relate to this simulation
- Propose new analyses you could perform on this trajectory inspired by those concepts
- Propose additional simulations that one could perform on this system to explore those concepts

Molecular Mechanics

The Potential Energy of Molecules Can be Approximated by Classical Mechanics

Quantum Mechanics

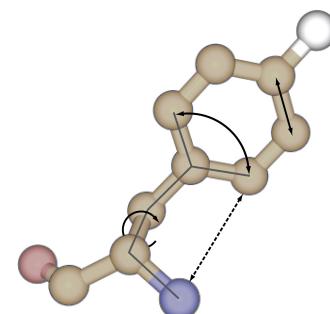


$$H|\Psi\rangle = E|\Psi\rangle$$

$$\left[-\frac{1}{2} \sum_i \nabla_i^2 - \sum_{i,A} \frac{Z_A}{r_{Ai}} + \sum_{i,j < i} \frac{1}{r_{ij}} + \sum_{A,B < A} \frac{Z_A Z_B}{r_{AB}} \right] |\Psi\rangle = E|\Psi\rangle$$

- $O(n^7)$ (n =electrons)
- Nuclei, Electrons, Electronic States
- Polarization, Electron Transfer, Electron Correlation
- Bond Breaking/Formations
- $\sim 10\text{-}100$ atoms
- $\sim \text{ps}$ dynamics

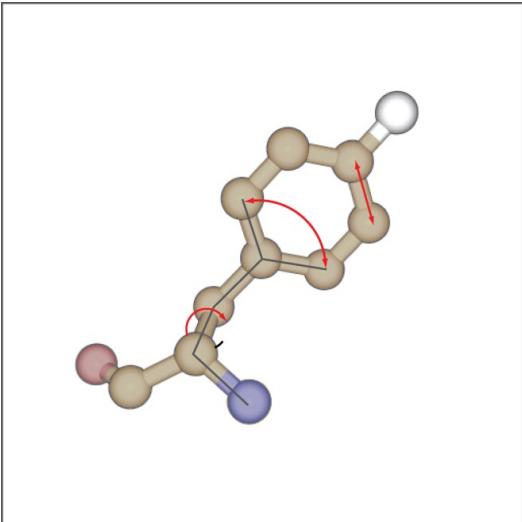
Molecular Mechanics



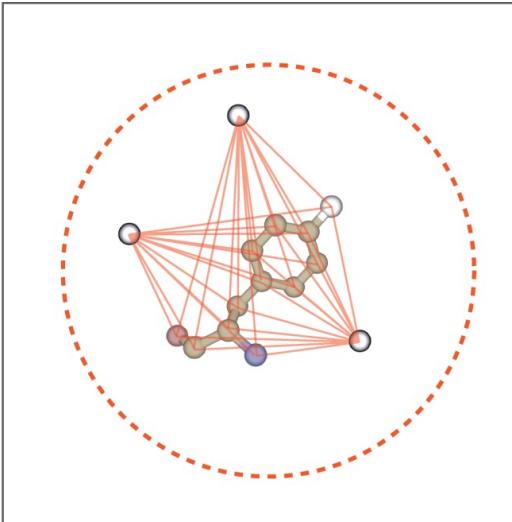
$$\begin{aligned} U(\mathbf{r}) = & \sum_{\text{bonds}} k_b (r - r_0)^2 + \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2 \\ & + \sum_{\text{torsions}} A [1 - \cos(n\tau - \phi)] + \sum_{\text{atoms}} \frac{B_{ij}}{r_{ij}^{12}} - \frac{C_{ij}}{r_{ij}^6} + \sum_{i,j < i} \frac{q_i q_j}{r_{ij}} \end{aligned}$$

- $O(n^2 \log n)$ (n =atoms)
- Nuclei only, Net atomic charges, Electronic Ground State
- Born-Oppenheimer Assumption, Two-body interactions
- $\sim 10,000$ to $1,000,000$ atoms
- $\sim \text{us-ms}$ dynamics

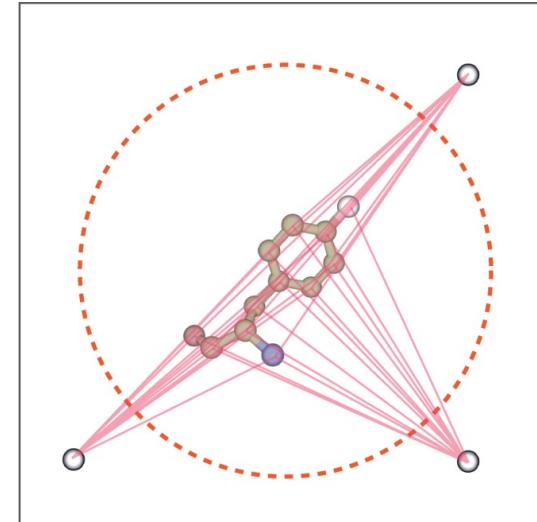
Molecular Mechanics Force Fields



Bonded terms



Near / Local Non-Bonded:
VdW & Electrostatic



Far / Global Non-Bonded:
Electrostatic

$$\begin{aligned} \mathbf{U}(\mathbf{r}) = & \sum_{\text{bonds}} k_b (r - r_0)^2 \\ & + \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2 \\ & + \sum_{\text{torsions}} A [1 - \cos(n\tau - \phi)] \end{aligned}$$

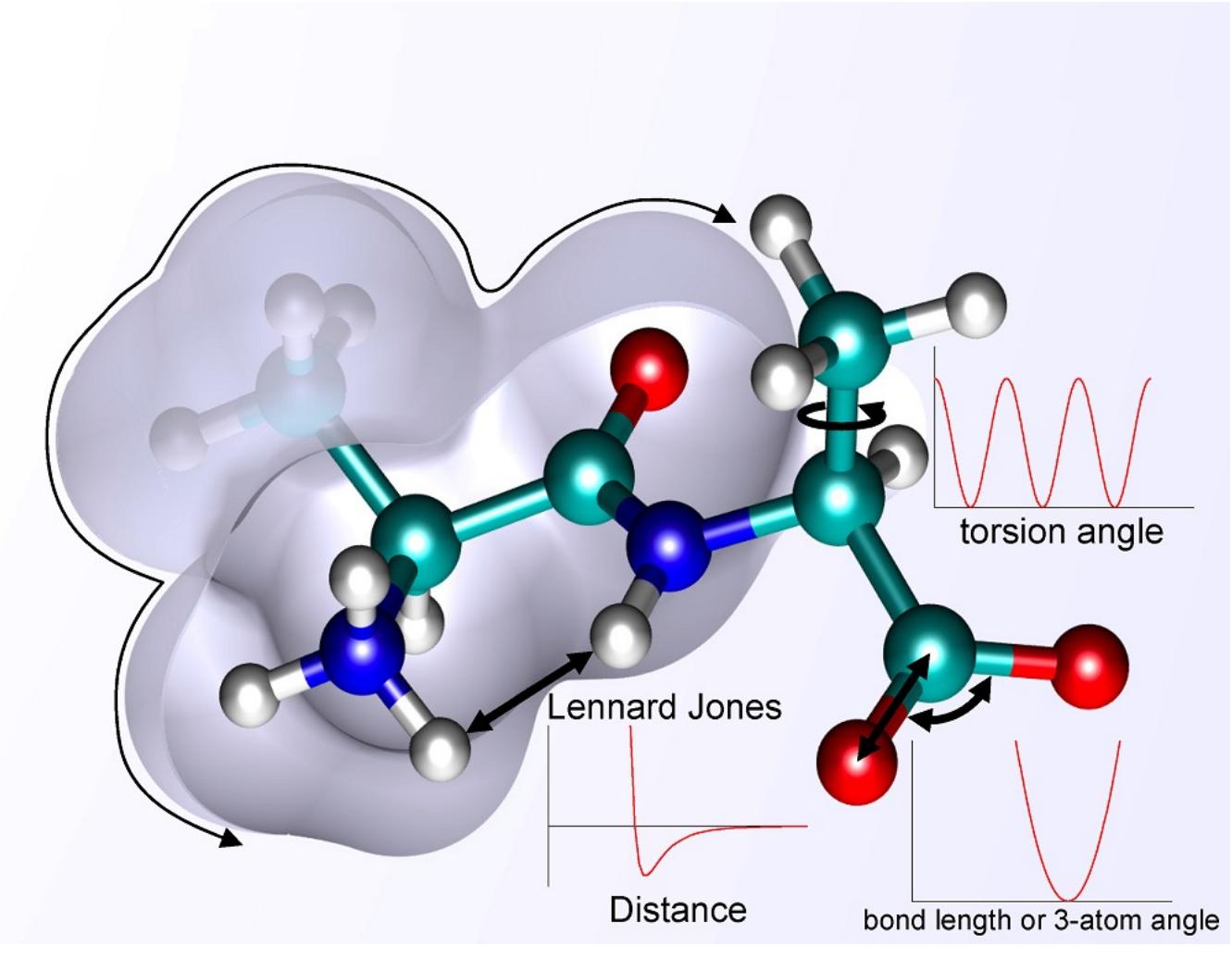
$$\begin{aligned} & + \sum_{i,j \in i} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \\ & + \sum_{i,j \in i} \frac{q_i q_j}{r_{ij}} \end{aligned}$$

$$+ \sum_{i,j \in i} \frac{q_i q_j}{r_{ij}}$$

Molecular Mechanics Force Fields

$$E_{Total} = E_{Bond\ Stretch} + E_{Bond\ Angles} + E_{Dihedrals} \\ + E_{Electrostatics} + E_{Van\ Der\ Waals}$$

$$E(r^N) = \\ \sum_{Bonds} \frac{1}{2} k_b (l - l_0)^2 \\ + \sum_{Angles} \frac{1}{2} k_a (\theta - \theta_0)^2 \\ + \sum_{Dihedrals} \frac{1}{2} V_n [1 + \cos(n\omega - \gamma)] \\ + \sum_{j=1}^{N-1} \sum_{i=j+1}^N \left\{ \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} + \epsilon_{i,j} \left[\left(\frac{r_{0ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{0ij}}{r_{ij}} \right)^6 \right] \right\}$$



Harmonic Bond Stretching

$$U(\mathbf{r}) = \sum_{\text{bonds}} k_b (r - r_0)^2$$

$$U(r) = U(r_{\text{eq}}) + \frac{dU}{dr} \Big|_{r=r_{\text{eq}}} (r - r_{\text{eq}}) + \frac{1}{2!} \frac{d^2U}{dr^2} \Big|_{r=r_{\text{eq}}} (r - r_{\text{eq}})^2$$

$$+ \frac{1}{3!} \frac{d^3U}{dr^3} \Big|_{r=r_{\text{eq}}} (r - r_{\text{eq}})^3 + \dots$$

$$U(r_{AB}) = \frac{1}{2} k_{AB} (r_{AB} - r_{AB,\text{eq}})^2$$

$$U(r_{AB}) = \frac{1}{2} [k_{AB} + k_{AB}^{(3)}(r_{AB} - r_{AB,\text{eq}}) + k_{AB}^{(4)}(r_{AB} - r_{AB,\text{eq}})^2] (r_{AB} - r_{AB,\text{eq}})^2$$

$$U(r_{AB}) = D_{AB} [1 - e^{-\alpha_{AB}(r_{AB} - r_{AB,\text{eq}})}]^2$$

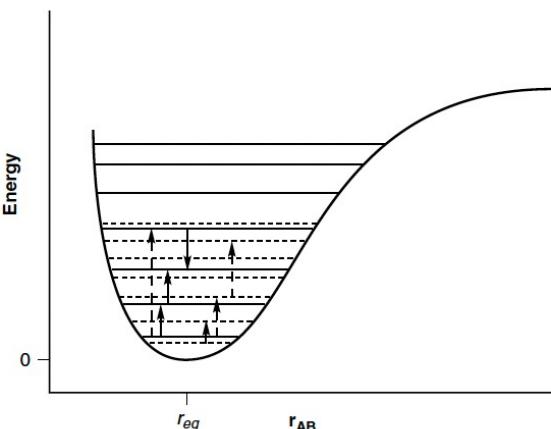
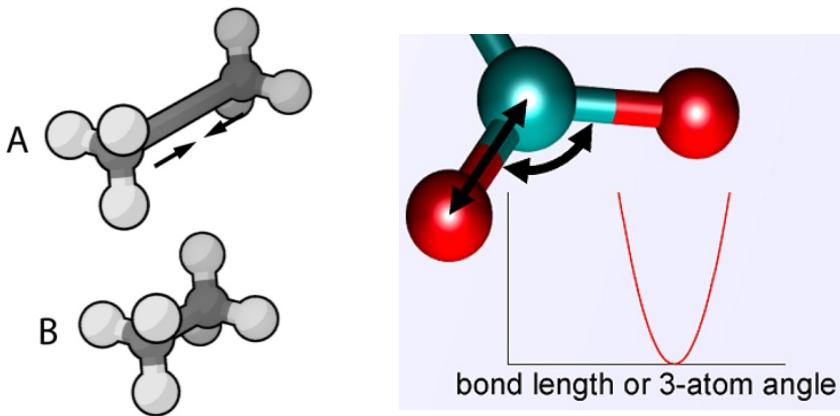
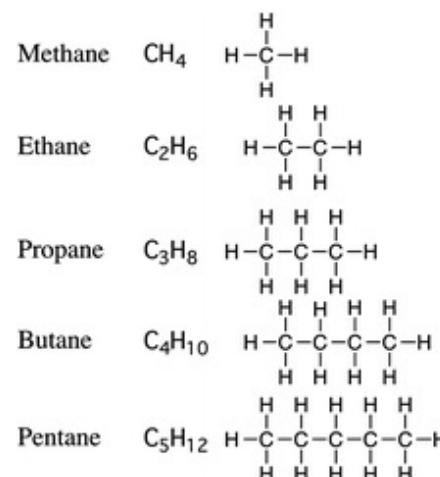


Figure 2.1 The first seven vibrational energy levels for a lighter (solid horizontal lines) and heavier (horizontal dashed lines) isotopomer of diatomic AB. Allowed vibrational transitions are indicated by solid vertical arrows, forbidden transitions are indicated by dashed vertical arrows



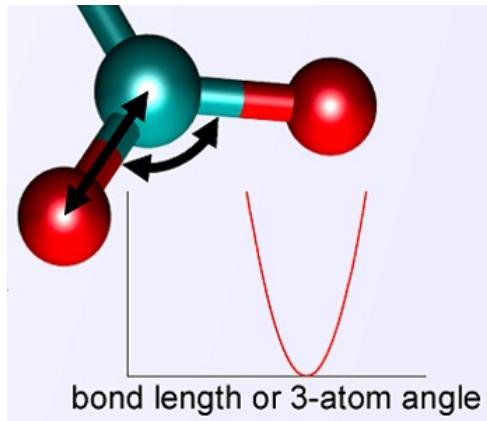
Parameter Transferability

Bond lengths in organic compounds^[9]

C-H	Length (pm)	C-C	Length (pm)	Multiple-bonds	Length (pm)
$\text{sp}^3\text{-H}$	110	$\text{sp}^3\text{-sp}^3$	154	Benzene	140
$\text{sp}^2\text{-H}$	109	$\text{sp}^3\text{-sp}^2$	150	Alkene	134
$\text{sp}\text{-H}$	108	$\text{sp}^2\text{-sp}^2$	147	Alkyne	120
		$\text{sp}^3\text{-sp}$	146	Allene	130
		$\text{sp}^2\text{-sp}$	143		
		sp-sp	137		

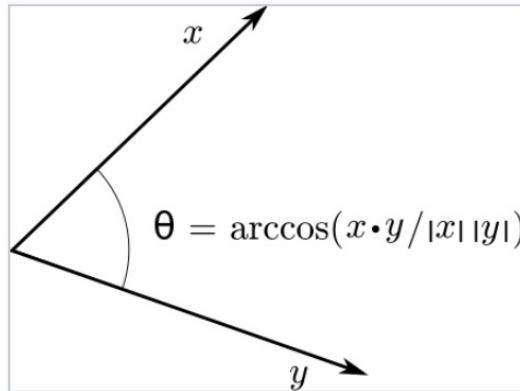
Harmonic Angle Bending

$$U(\mathbf{r}) = \sum_{\text{Angles}} \frac{1}{2} k_a (\theta - \theta_0)^2$$

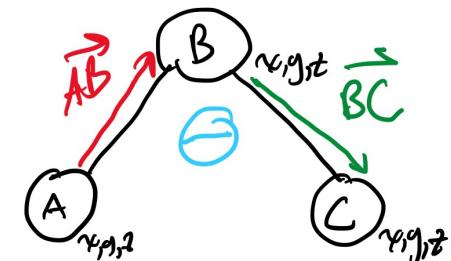


$$U(\theta_{ABC}) = \frac{1}{2} [k_{ABC} + k_{ABC}^{(3)}(\theta_{ABC} - \theta_{ABC,\text{eq}}) + k_{ABC}^{(4)}(\theta_{ABC} - \theta_{ABC,\text{eq}})^2 + \dots]$$

$$(\theta_{ABC} - \theta_{ABC,\text{eq}})^2$$



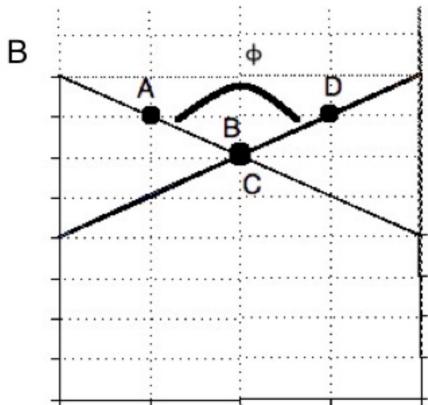
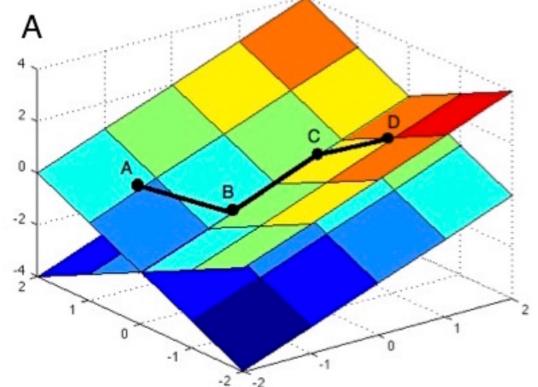
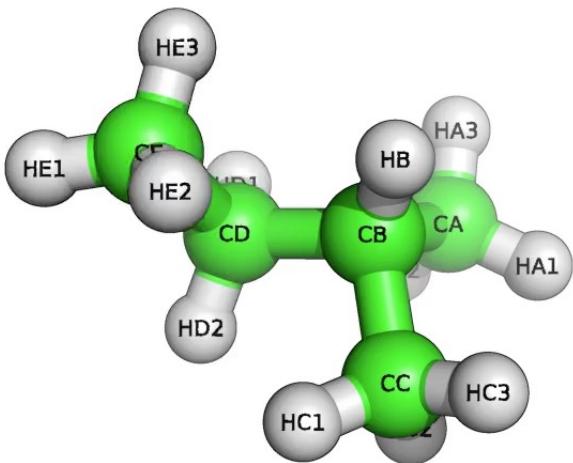
```
def ang(A, B, C):
    xba = atoms[A-1].x - atoms[B-1].x
    yba = atoms[A-1].y - atoms[B-1].y
    zba = atoms[A-1].z - atoms[B-1].z
    xbc = atoms[C-1].x - atoms[B-1].x
    ybc = atoms[C-1].y - atoms[B-1].y
    zbc = atoms[C-1].z - atoms[B-1].z
    dot = (xba*xbc) + (yba*ybc) + (zba*zbc)
    d_ab = dist(A, B)
    d_bc = dist(B, C)
    return (math.acos(dot/(d_ab*d_bc))) * 180 / math.pi
```



$$\theta = \arccos \left(\frac{\vec{AB} \cdot \vec{BC}}{|\vec{AB}| |\vec{BC}|} \right)$$

$$\begin{aligned}
\vec{AB} &= [A_x - B_x, A_y - B_y, A_z - B_z] \\
\vec{AB} &= [A_x - \beta_x, A_y - \beta_y, A_z - \beta_z] \\
\vec{BC} &= [C_x - B_x, C_y - B_y, C_z - B_z] \\
\vec{AB} \cdot \vec{BC} &= A_x B_x + A_y B_y + A_z B_z \\
&\quad + A_x C_x + A_y C_y + A_z C_z \\
&\quad + \beta_x C_x + \beta_y C_y + \beta_z C_z
\end{aligned}$$

Torsion (Dihedral) Angles



```
def dihe(A, B, C, D):
    xba = atoms[A-1].x - atoms[B-1].x
    yba = atoms[A-1].y - atoms[B-1].y
    zba = atoms[A-1].z - atoms[B-1].z
    xbc = atoms[C-1].x - atoms[B-1].x
    ybc = atoms[C-1].y - atoms[B-1].y
    zbc = atoms[C-1].z - atoms[B-1].z
    xcb = atoms[B-1].x - atoms[C-1].x
    ycb = atoms[B-1].y - atoms[C-1].y
    zcb = atoms[B-1].z - atoms[C-1].z
    xcd = atoms[D-1].x - atoms[C-1].x
    ycd = atoms[D-1].y - atoms[C-1].y
    zcd = atoms[D-1].z - atoms[C-1].z

    # Cross vectors
    Cbabc_x = (yba * zbc) - (zba * ybc)
    Cbabc_y = (zba * xbc) - (xba * zbc)
    Cbabc_z = (xba * ybc) - (yba * xbc)

    Ccbcd_x = (ycb * zcd) - (zcb * ycd)
    Ccbcd_y = (zcb * xcd) - (xcb * zcd)
    Ccbcd_z = (xcb * ycd) - (ycb * xcd)

    dot = (Cbabc_x * Ccbcd_x) + (Cbabc_y * Ccbcd_y) + (Cbabc_z * Ccbcd_z)
    Den = math.sqrt(Cbabc_x**2 + Cbabc_y**2 + Cbabc_z**2) * math.sqrt(Ccbcd_x**2 + Ccbcd_y**2 + Ccbcd_z**2)

    i = yba * zcd - zba * ycd
    j = zba * xcd - xba * zcd
    k = xba * ycd - yba * xcd
    f = i * xbc + j * ybc + k * zbc

    if f >= 0:
        return (math.acos(dot / Den)) * 180 / math.pi
    else:
        return -1 * ((math.acos(dot / Den)) * 180 / math.pi)
```

Torsion (Dihedral) Angles

$$\sum_{\text{Dihedrals}} \frac{1}{2} V_n [1 + \cos(n\omega - \gamma)]$$

$$\sum_{\text{torsions}} A [1 - \cos(n\tau - \phi)]$$

$$U(\omega_{ABCD}) = \frac{1}{2} \sum_{\{j\}_{ABCD}} V_{j,ABCD} [1 + (-1)^{j+1} \cos(j\omega_{ABCD} + \psi_{j,ABCD})]$$

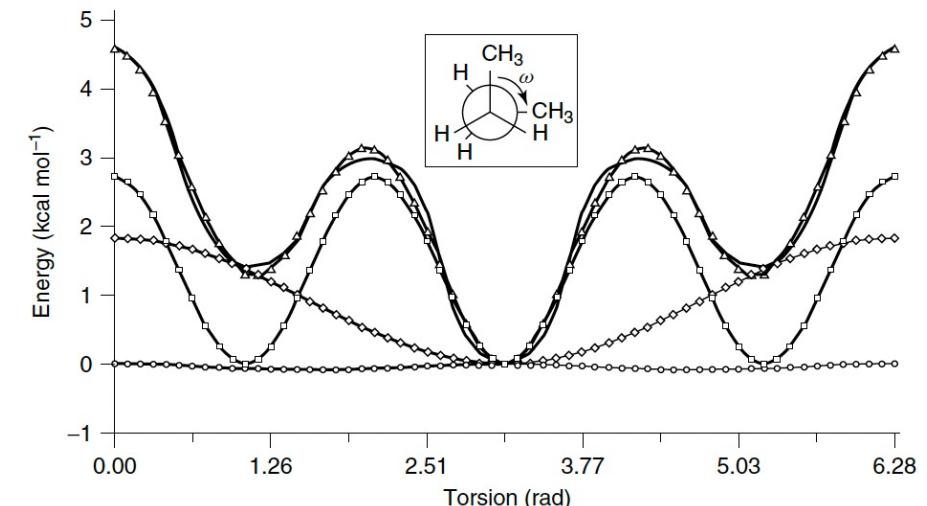
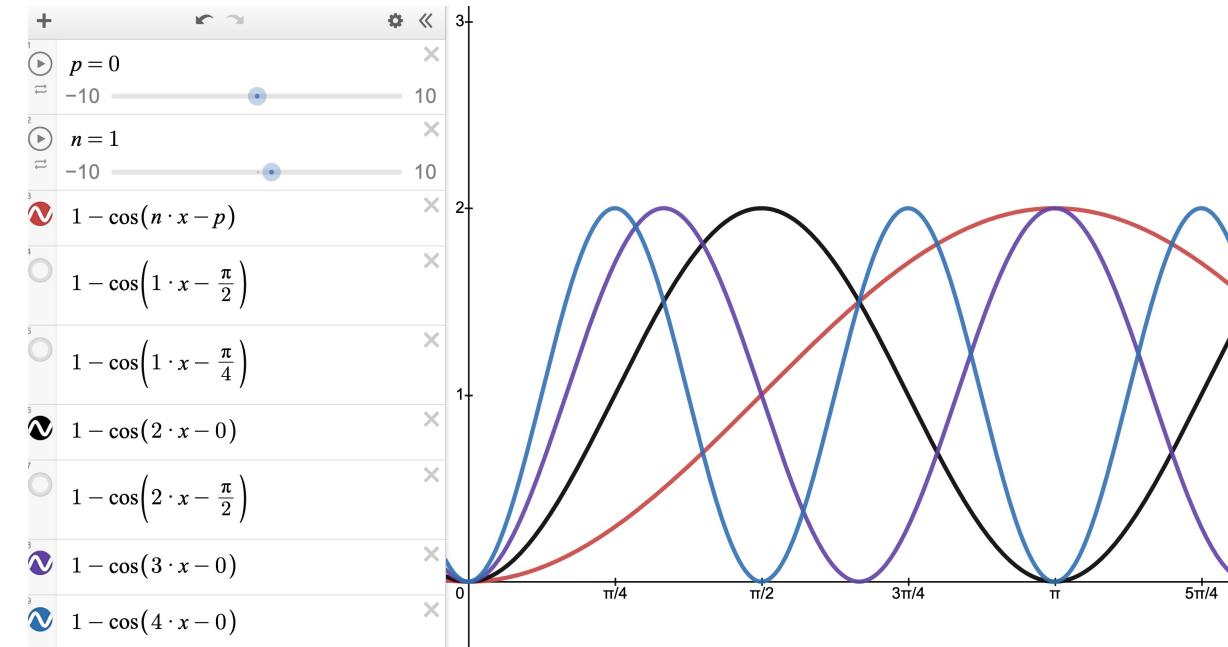


Figure 2.4 Fourier decomposition of the torsional energy for rotation about the C–C bond of *n*-butane (bold black curve, energetics approximate). The Fourier sum (Δ) has a close overlap, and is composed of the onefold (\diamond), twofold (\circ), and threefold (\square) periodic terms, respectively

Fourier dihedrals

$$V_F(\phi_{ijkl}) = \frac{1}{2} [C_1(1 + \cos(\phi)) + C_2(1 - \cos(2\phi)) + C_3(1 + \cos(3\phi)) + C_4(1 - \cos(4\phi))]$$

Ryckaert-Belleman dihedral potential

$$V_{rb}(\phi_{ijkl}) = \sum_{n=0}^5 C_n (\cos(\psi))^n,$$

“Improper” Torsion Angles

Out-of-plane Bending

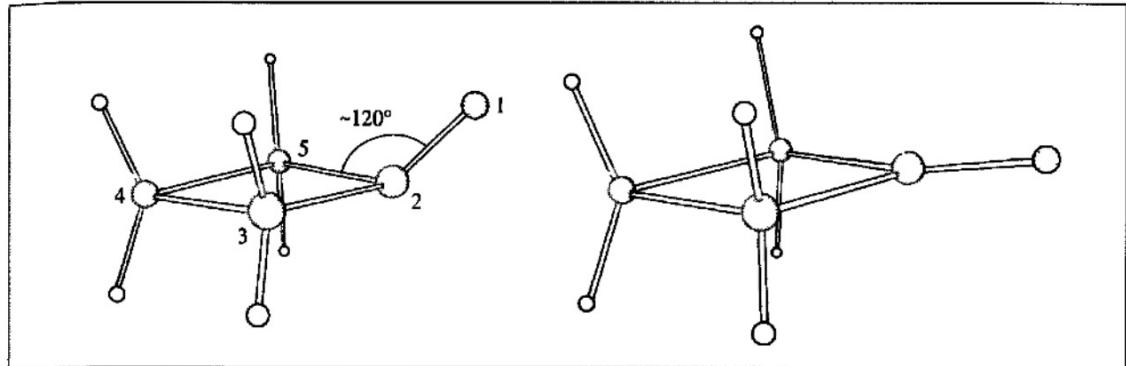
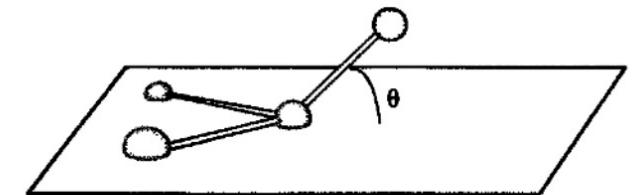
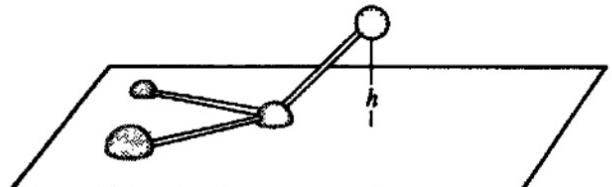


Fig 4.9: Without an out-of-plane term, the oxygen atom in cyclobutane is predicted to lie out of the plane of the ring (left) rather than in the plane.

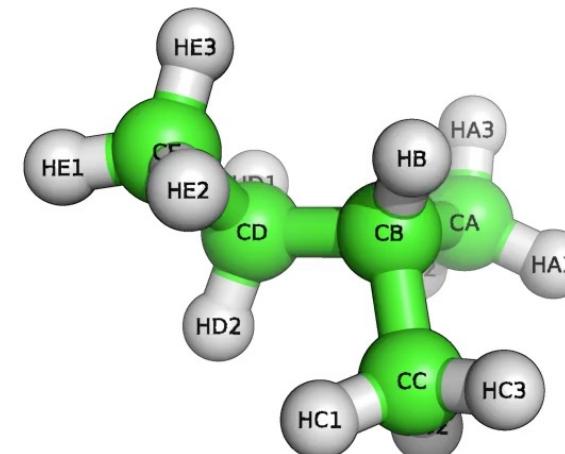
$$\nu(\omega) = k(1 - \cos 2\omega)$$



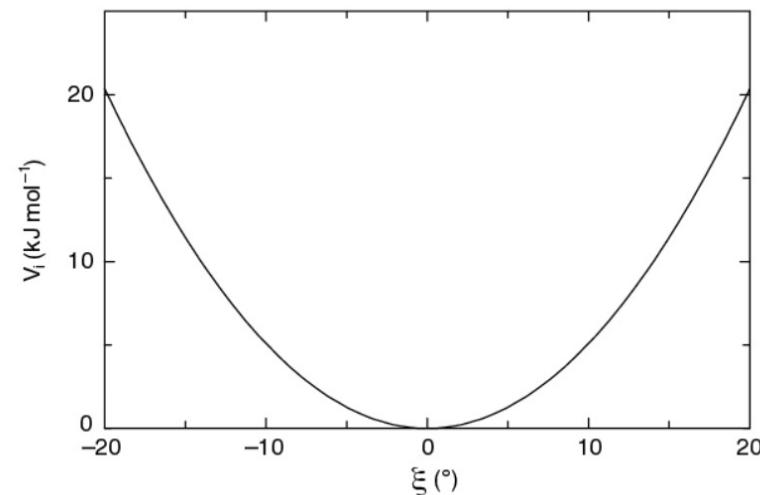
$$\nu(\theta) = \frac{k}{2} \theta^2;$$



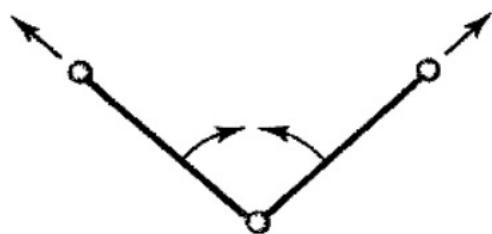
$$\nu(h) = \frac{k}{2} h^2$$



$$V_{id}(\xi_{ijkl}) = \frac{1}{2} k_\xi (\xi_{ijkl} - \xi_0)^2$$

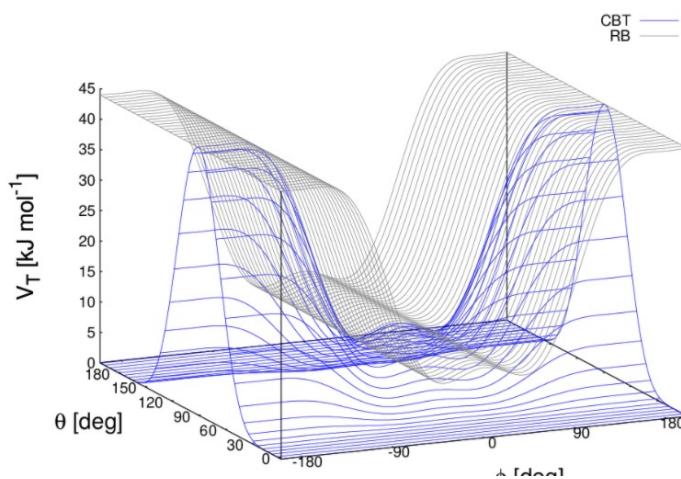


Bonded Cross Terms

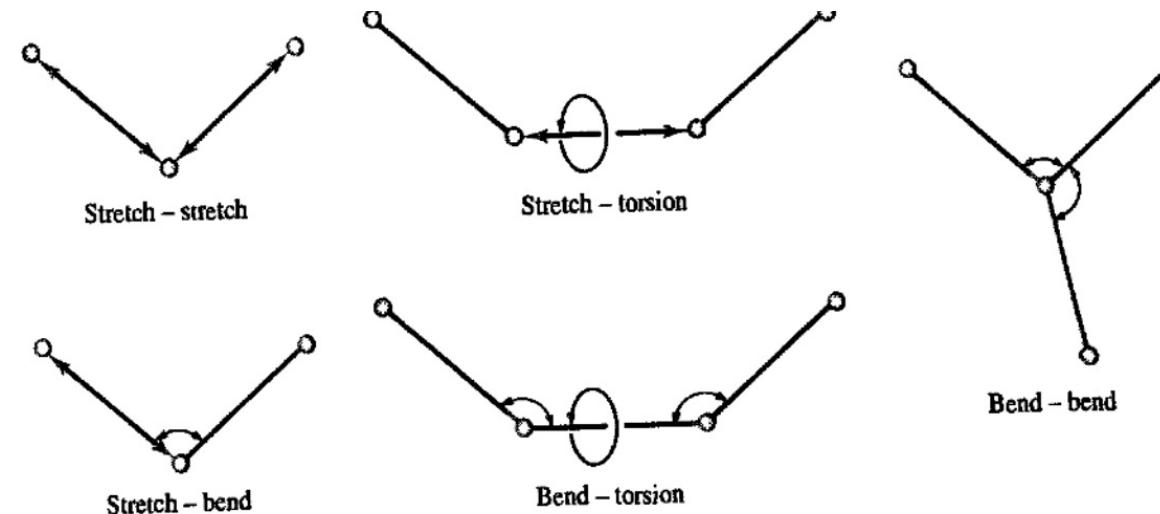


$$U(r_{AB}, \theta_{ABC}) = \frac{1}{2} k_{AB,ABC} (r_{AB} - r_{AB,eq})(\theta_{ABC} - \theta_{ABC,eq})$$

$$U(r_{BC}, \omega_{ABCD}) = \frac{1}{2} k_{BC,ABCD} (r_{BC} - r_{BC,eq})[1 + \cos(j\omega + \psi)]$$



$$V_{CBT}(\theta_{i-1}, \theta_i, \phi_i) = k_\phi \sin^3 \theta_{i-1} \sin^3 \theta_i \sum_{n=0}^4 a_n \cos^n \phi_i.$$



$$\nu(l_1, l_2) = \frac{k_{l_1, l_2}}{2} [(l_1 - l_{1,0})(l_2 - l_{2,0})]$$

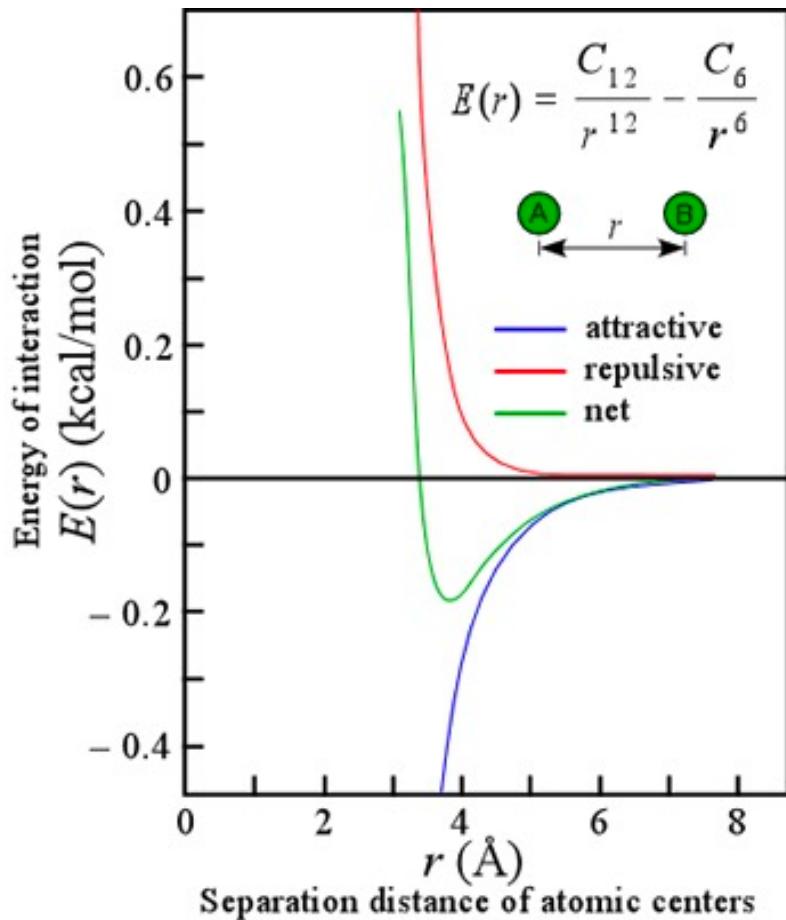
$$\nu(l_1, l_2, \theta) = \frac{k_{l_1, l_2, \theta}}{2} [(l_1 - l_{1,0}) + (l_2 - l_{2,0})](\theta - \theta_0)$$

$$\nu(l, \omega) = k(l - l_0) \cos n\omega$$

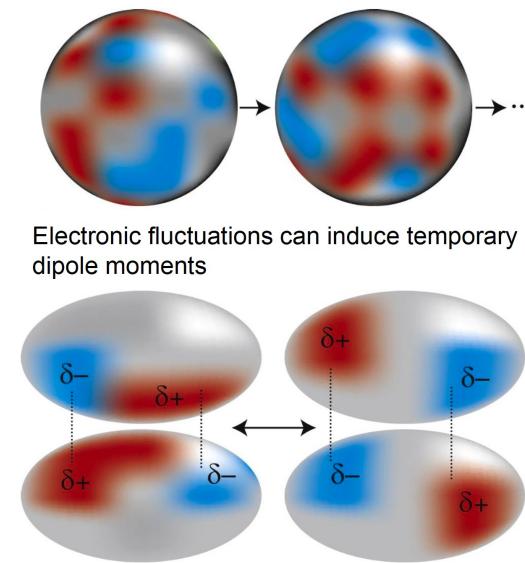
$$\nu(l, \omega) = k(l - l_0)[1 + \cos n\omega]$$

Van der Waals / “London Dispersion”

$$+ \sum_{j=1}^{N-1} \sum_{i=j+1}^N \epsilon_{i,j} \left[\left(\frac{r_{0ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{0ij}}{r_{ij}} \right)^6 \right]$$



Attractive $-(1/r^6)$ term arises from vdW (or “dispersion”): transient fluctuating dipoles in atoms can polarize each other generating an attractive force



Repulsive $+(1/r^{12})$ is due to strong repulsion between the core (non-valence) electrons between atoms – results from Pauli Exchange principle – also known as “exchange-repulsion” or “exchange”

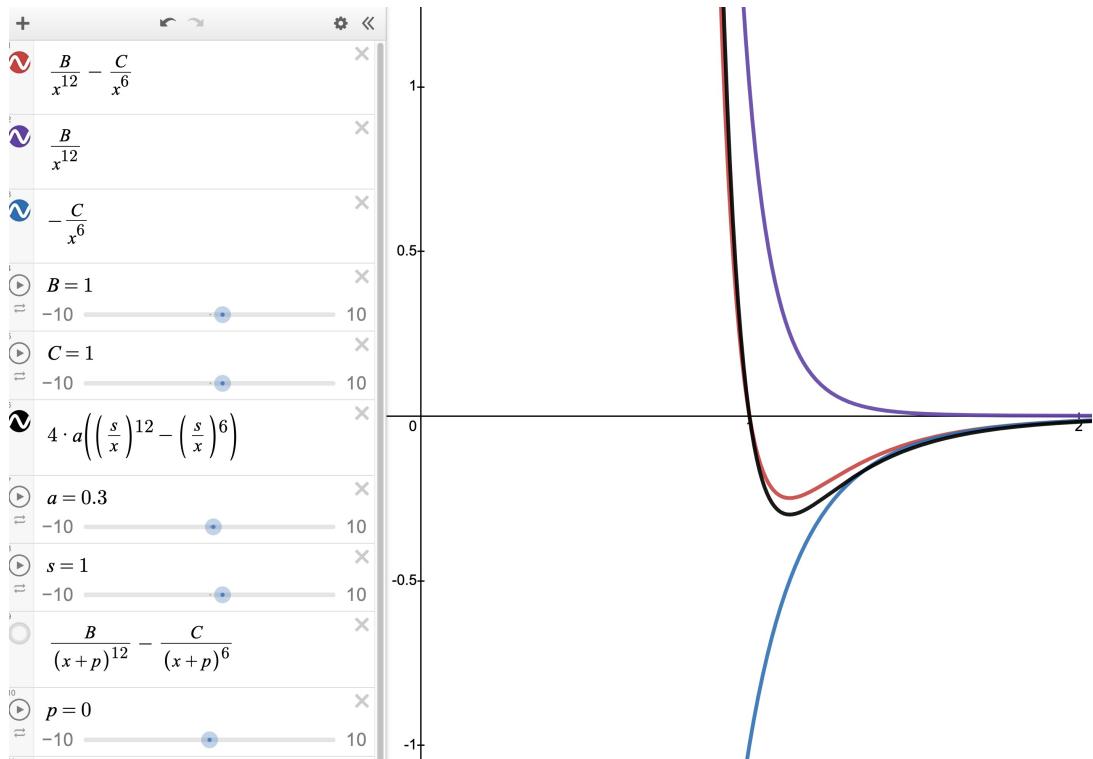
Van der Waals / Lennard-Jones

$$V_{\text{LJ}}(r) = \frac{A}{r^{12}} - \frac{B}{r^6},$$

where, $A = 4\varepsilon\sigma^{12}$ and $B = 4\varepsilon\sigma^6$. Conversely, $\sigma = \sqrt[6]{\frac{A}{B}}$ and $\varepsilon = \frac{B^2}{4A}$.

$$U(r_{AB}) = 4\varepsilon_{AB} \left[\left(\frac{\sigma_{AB}}{r_{AB}}\right)^{12} - \left(\frac{\sigma_{AB}}{r_{AB}}\right)^6 \right]$$

$$r_{AB}^* = 2^{1/6}\sigma_{AB} \quad U = -\varepsilon_{AB}$$



Lennard Jones

<https://www.desmos.com/calculator/nqtmgtibdp>

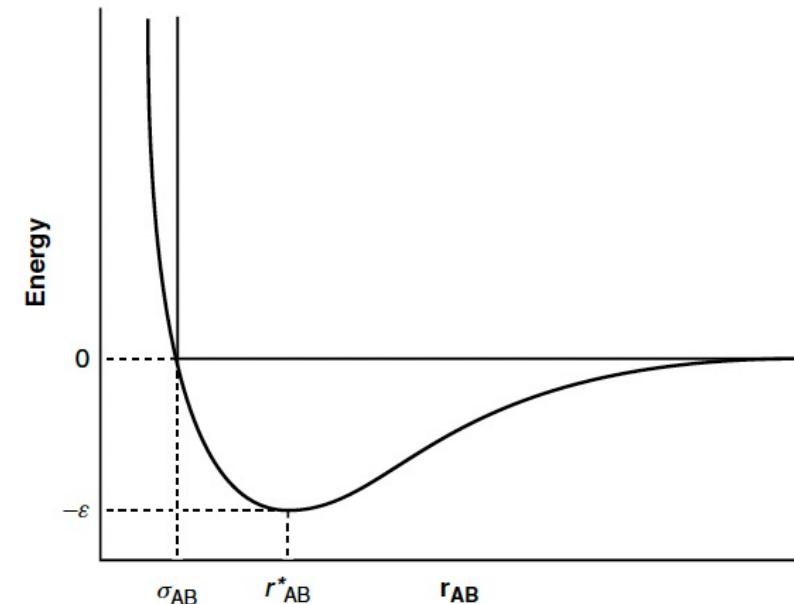


Figure 2.6 Non-attractive hard-sphere potential (straight lines) and Lennard–Jones potential (curve). Key points on the energy and bond length axes are labeled

Potential Energy Surfaces

$$\mathbf{X} \equiv (x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_N, y_N, z_N)$$

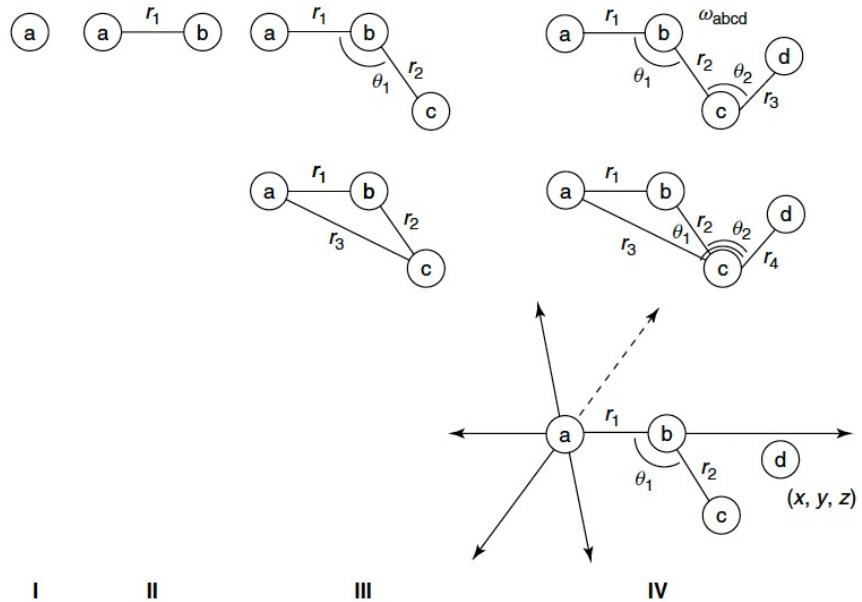


Figure 1.2 Different means for specifying molecular geometries. In frame **I**, there are no degrees of freedom as only the nature of atom ‘a’ has been specified. In frame **II**, there is a single degree of freedom, namely the bond length. In frame **III**, location of atom ‘c’ requires two additional degrees of freedom, either two bond lengths or a bond length and a valence angle. Frame **IV** illustrates various ways to specify the location of atom ‘d’; note that in every case, three new degrees of freedom must be specified, either in internal or Cartesian coordinates

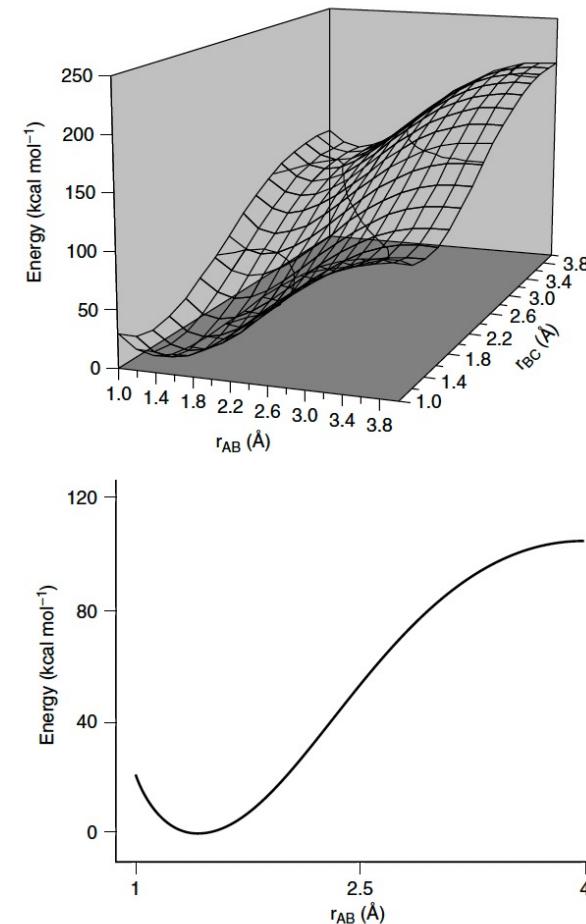


Figure 1.3 The full PES for the hypothetical molecule ABC requires four dimensions to display ($3N - 6 = 3$ coordinate degrees of freedom plus one dimension for energy). The three-dimensional plot (top) represents a hyperslice through the full PES showing the energy as a function of two coordinate dimensions, the AB and BC bond lengths, while taking a fixed value for the angle ABC (a typical choice might be the value characterizing the global minimum on the full PES). A further slice of this surface (bottom) now gives the energy as a function of a single dimension, the AB bond length, where the BC bond length is now also treated as frozen (again at the equilibrium value for the global minimum)

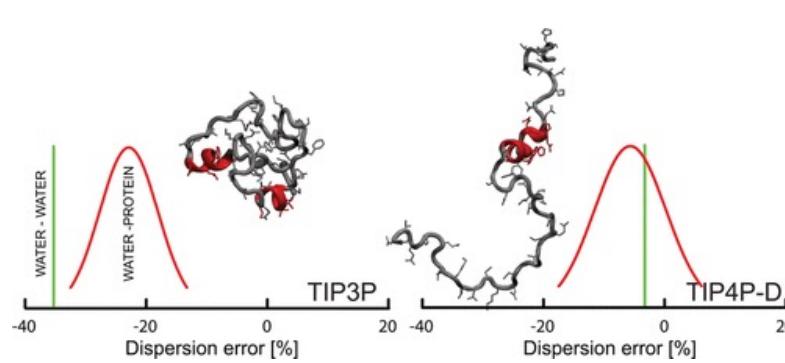
Why a Lennard-Jones (“12-6”) Potential?

$$U(r_{AB}) = 4\epsilon_{AB} \left[\left(\frac{\sigma_{AB}}{r_{AB}} \right)^{12} - \left(\frac{\sigma_{AB}}{r_{AB}} \right)^6 \right]$$

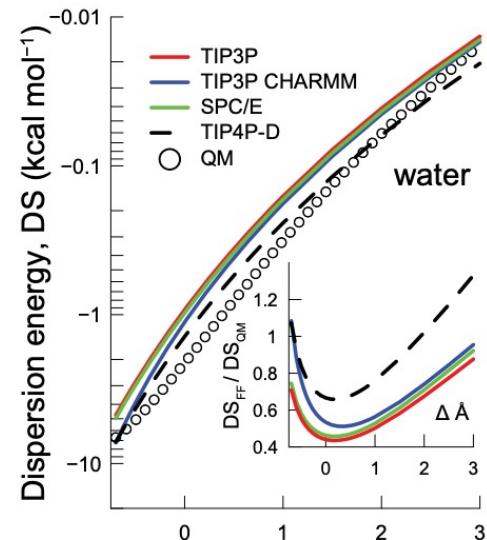
The are more physically motivated attractive dispersion terms than $-1/r^6$ and repulsive “exchange” terms than $1/r^{12}$

Quantum Mechanical Models of the Condensed Phase Dispersion have more terms than $-1/r^6$

$$E_{Dispersion} = - \sum_{n=6,8,10,\dots} \frac{c_n}{r^n}$$



<https://pubs.acs.org/doi/abs/10.1021/jp508971m>



Morse Potential

$$U(r_{AB}) = D_{AB} [1 - e^{-\alpha_{AB}(r_{AB} - r_{AB,eq})}]^2$$

Buckingham Potential

$$\Phi_{12}(r) = A \exp(-Br) - \frac{C}{r^6}$$

Hill Potential

$$U(r_{AB}) = \epsilon_{AB} \left[\frac{6}{\beta_{AB} - 6} \exp\left(\beta_{AB} \frac{1 - r_{AB}}{r_{AB}^*}\right) - \frac{\beta_{AB}}{\beta_{AB} - 6} \left(\frac{r_{AB}^*}{r_{AB}}\right)^6 \right]$$

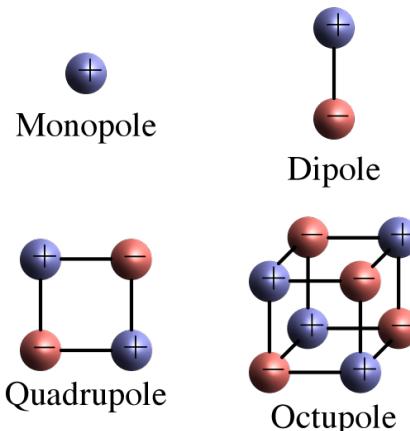
AMOEBA damped 14-7

$$U_{vdW} = U_{repulsion} + U_{dispersion} = \epsilon_{ij} \left(\frac{1 + \delta}{\rho_{ij} + \delta} \right)^7 \left(\frac{1 + \gamma}{\rho_{ij}^7 + \gamma} - 2 \right)$$

Electrostatics & Multipoles

A coulomb potential uses the simplifying assumption of “point” charges centered at nuclei

$$+ \sum_{j=1}^{N-1} \sum_{i=j+1}^N \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$$



Dipole:Dipole Interactions

$$U_{AB/CD} = \frac{\mu_{AB}\mu_{CD}}{\epsilon_{AB/CD}r_{AB/CD}^3} (\cos \chi_{AB/CD} - 3 \cos \alpha_{AB} \cos \alpha_{CD})$$

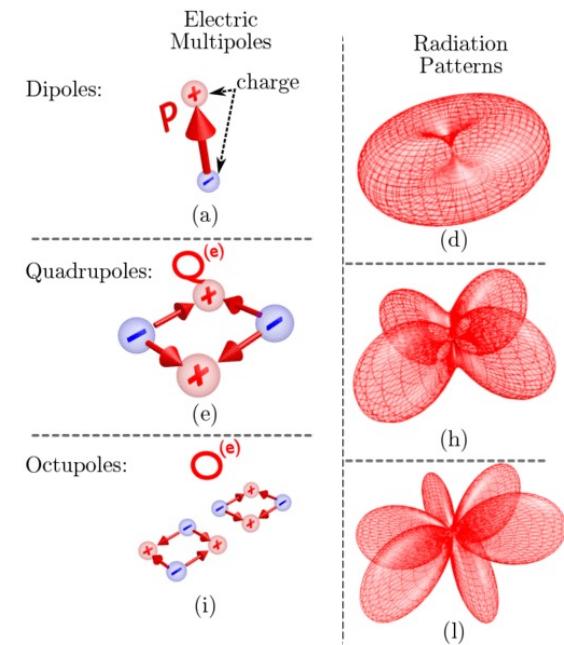
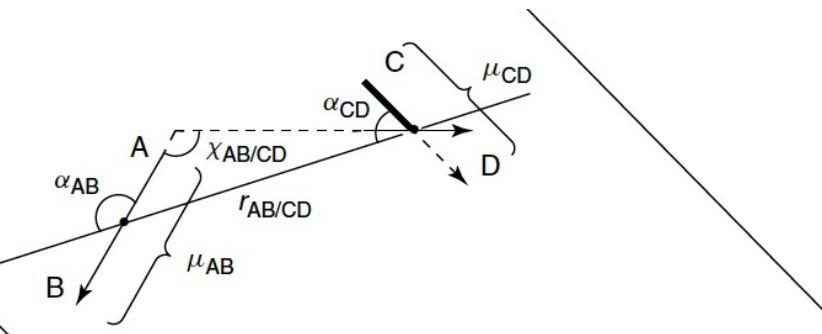


TABLE III.
Electrostatic Models.

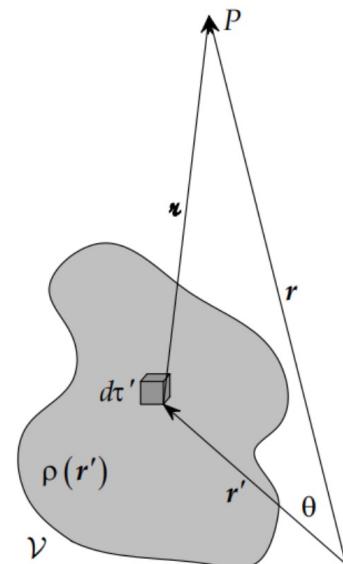
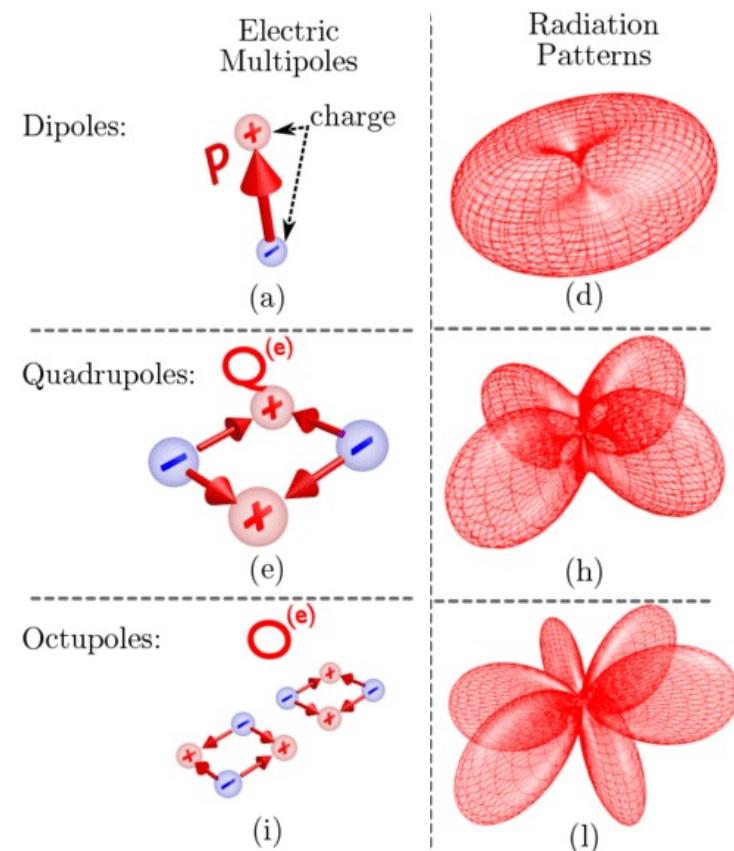
Model	Type	Description
[mul]	Partial charge	Mulliken partial charges at nuclei
[pd]		Potential derived partial charges at nuclei
[md]	Multipole ^a	A monopole and dipole are included at each nucleus
[mdq]		A monopole, dipole, and quadrupole are included at each nucleus
[mdqo]		A monopole, dipole, quadrupole, and octupole are included at each nucleus
[cpk]	Nonspherical CPK ^a	Generalized atomic charge densities represented without approximation

^aIncludes electrostatic interaction energies of types 1 — 2 and 1 — 3.

AMOEBA (atomic multipole optimized energetics for biomolecular applications)

<https://pubs.acs.org/doi/full/10.1021/jp910674d>

Electrostatics & Multipoles



$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \int_{V'} \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} dV'$$

$$\begin{aligned} |\mathbf{r} - \mathbf{r}'| &= \sqrt{|r^2 - 2\mathbf{r} \cdot \mathbf{r}' + r'^2|} \\ &= r \sqrt{1 - 2\frac{\hat{\mathbf{r}} \cdot \mathbf{r}'}{r} + \left(\frac{r'}{r}\right)^2} \end{aligned}$$

Binomial Expansion:

$$\frac{1}{\sqrt{1 - 2\frac{\hat{\mathbf{r}} \cdot \mathbf{r}'}{r} + \left(\frac{r'}{r}\right)^2}} = 1 - \frac{\hat{\mathbf{r}} \cdot \mathbf{r}'}{r} + \frac{1}{2r^2}(r'^2 - 3(\hat{\mathbf{r}} \cdot \mathbf{r}')^2) + O\left(\frac{r'}{r}\right)^3$$

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0 r} \int_{V'} \rho(\mathbf{r}') \left(1 - \frac{\hat{\mathbf{r}} \cdot \mathbf{r}'}{r} + \frac{1}{2r^2}(3(\hat{\mathbf{r}} \cdot \mathbf{r}')^2 - r'^2) + O\left(\frac{r'}{r}\right)^3 \right) dV'$$

$$V(\mathbf{r}) = V_{\text{mon}}(\mathbf{r}) + V_{\text{dip}}(\mathbf{r}) + V_{\text{quad}}(\mathbf{r}) + \dots$$

$$V_{\text{mon}}(\mathbf{r}) = \frac{1}{4\pi\epsilon_0 r} \int_{V'} \rho(\mathbf{r}') dV'$$

$$V_{\text{dip}}(\mathbf{r}) = -\frac{1}{4\pi\epsilon_0 r^2} \int_{V'} \rho(\mathbf{r}') (\hat{\mathbf{r}} \cdot \mathbf{r}') dV'$$

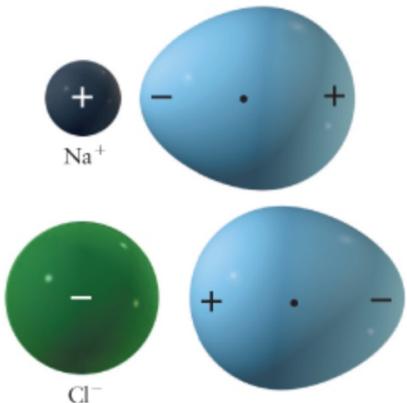
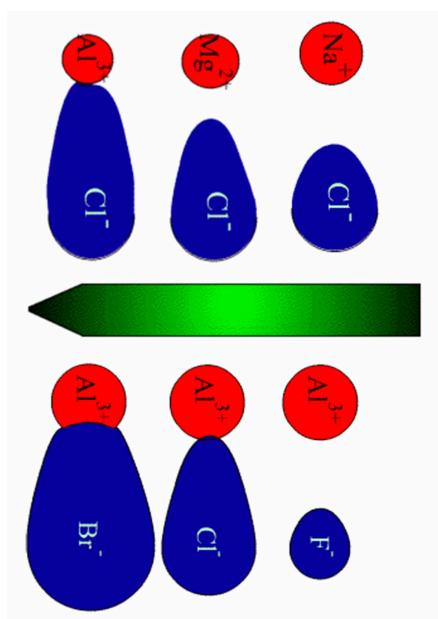
$$V_{\text{quad}}(\mathbf{r}) = \frac{1}{8\pi\epsilon_0 r^3} \int_{V'} \rho(\mathbf{r}') (3(\hat{\mathbf{r}} \cdot \mathbf{r}')^2 - r'^2) dV'$$

Polarizability + Many Body Effects

polarizability

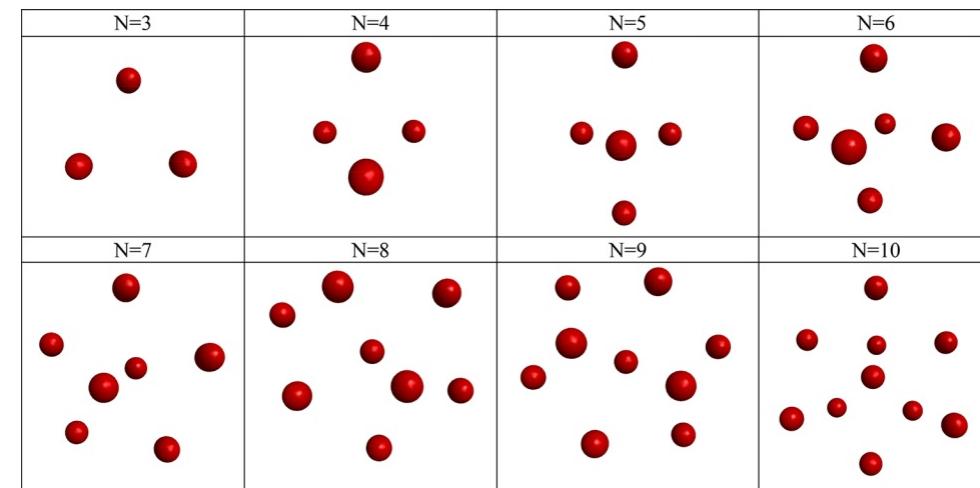
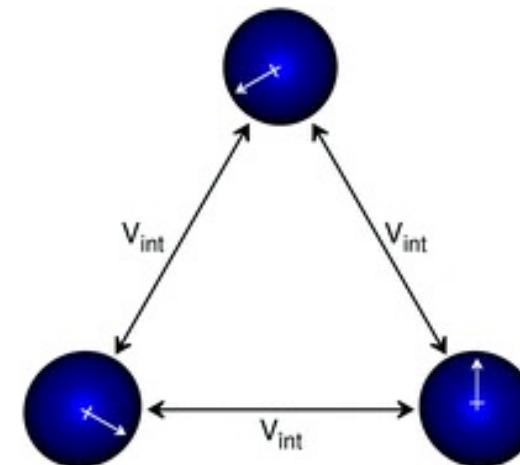
A measure of the extent to which electrons in atoms or molecules are displaced by electric fields, most often the electric fields due to a nearby ion or electromagnetic radiation, leading to induced dipole moments

High
↓
Low
Electrical Field Intensity and Direction

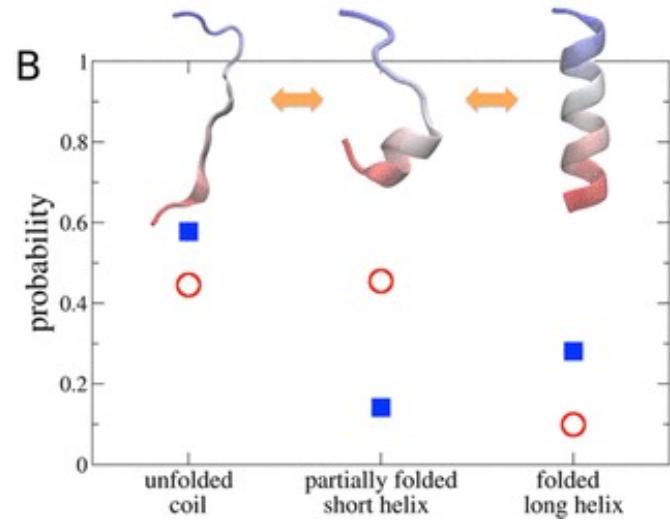
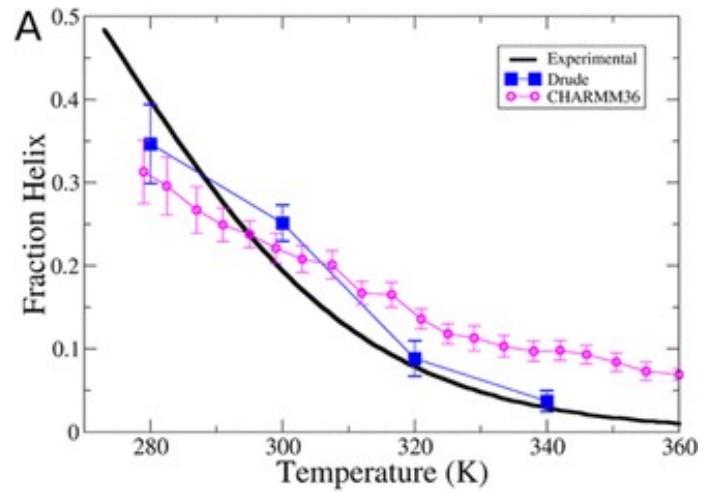
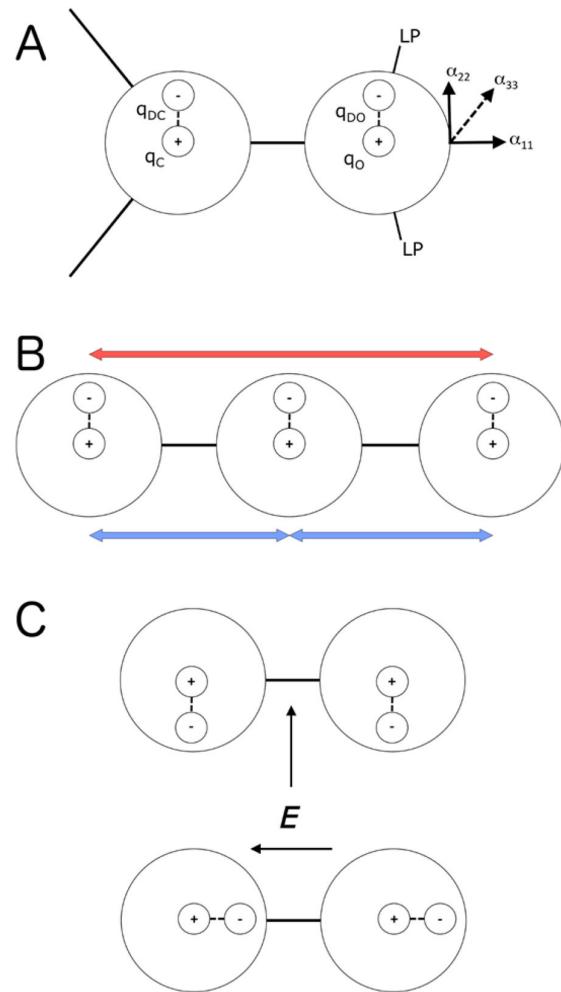
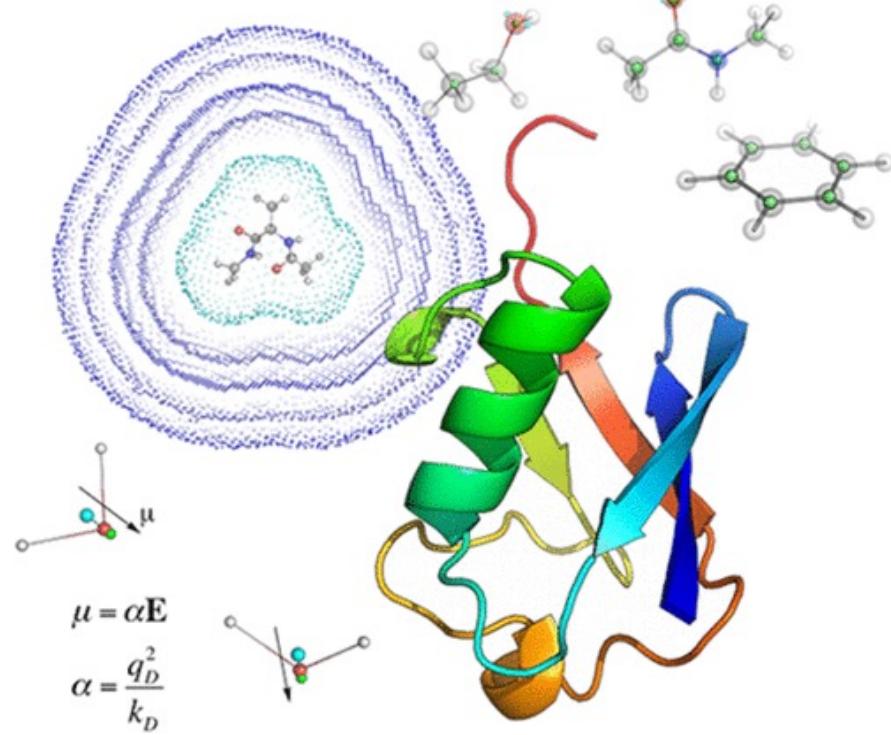


Many Body Dispersion & Electrostatics

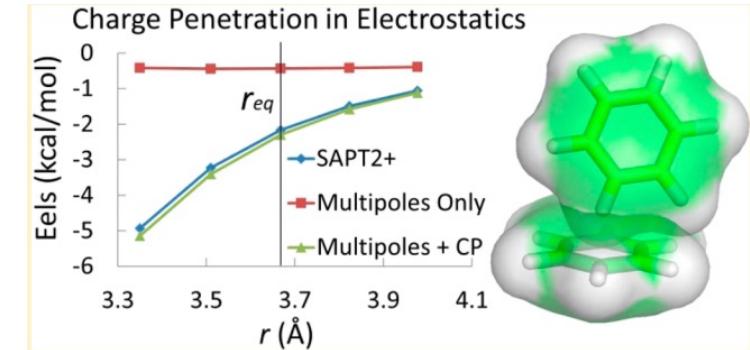
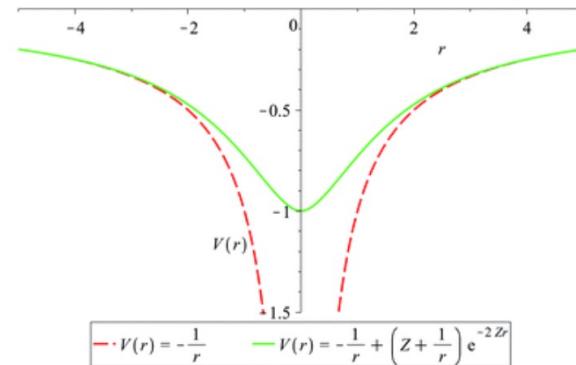
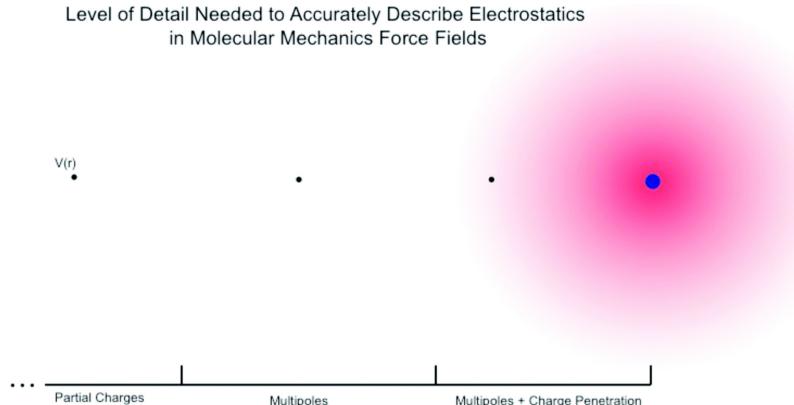
$$\nu(A, B, C) \neq \nu(A, B) + \nu(A, C) + \nu(B, C)$$



Polarizable / Drude Models of Electrostatics



Charge Penetration & Smeared Charges



$$V(r) = -\frac{1}{r} \left(1 - (1 + Zr)e^{-2Zr}\right) = -\frac{1}{r} \cdot f_{\text{damp}}(r)$$

<https://pubs.acs.org/doi/10.1021/acs.jctc.5b00267>

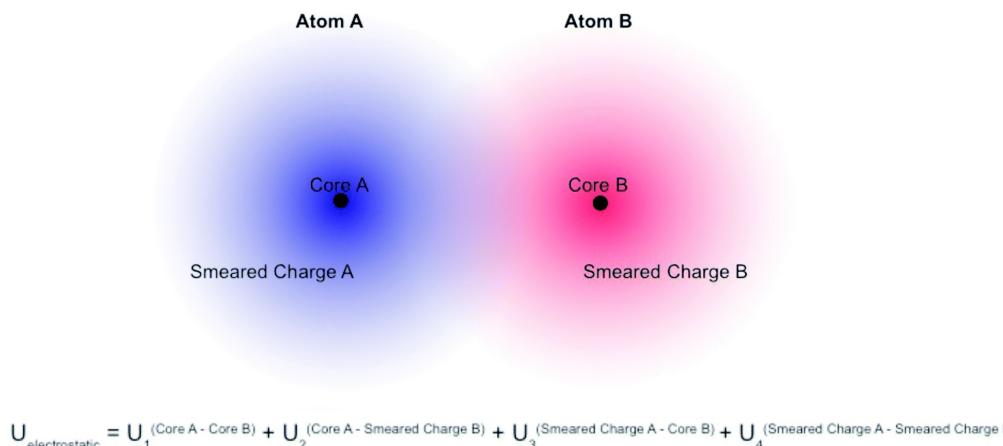


Fig. 3 Electrostatic energy of charge penetration-corrected, smeared-charge atomic interactions. The total electrostatic energy is split into four parts. The first term is the energy of the core–core, point–point interaction. The second and third terms are the energies of each core in the electrostatic potential of the opposing smeared charge. The fourth term is the energy of the overlap between smeared charge distributions.

<https://pubs.rsc.org/en/content/articlelanding/2017/cp/c6cp06017>

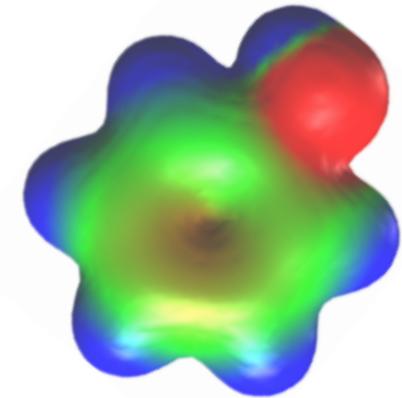
Table 1 Proposed methods for incorporating charge penetration into molecular mechanics electrostatic energy. For consistency, Z is the nuclear charge, ρ is the total charge density of the electrons, q is the total charge of the electron cloud, V is the number of valence electrons, c is the partial charge, n is the number of “screening electrons”, and r is the internuclear distance. In the first row, the charge density is either a promolecular charge density (Engels) or a density from hermite gaussians in the GEM model (Cisneros)

Model	Core A–core B	Core A-smeared charge B	Smeared charge A–core B	Smeared charge A-smeared charge B
Engels;	$\frac{Z_A Z_B}{r}$	$\int_{-\infty}^{\infty} \frac{Z_A \rho_B(r_2)}{ R_A - r_2 } dr_2$	$\int_{-\infty}^{\infty} \frac{Z_B \rho_A(r_1)}{ R_B - r_1 } dr_1$	$\int \int_{-\infty}^{\infty} \frac{\rho_A(r_1) \rho_B(r_2)}{ r_1 - r_2 } dr_1 dr_2$
Cisneros	$\frac{Z_A Z_B}{r}$	$\frac{Z_A q_B}{r} f_{\text{damp}}(r)$	$\frac{Z_B q_A}{r} f_{\text{damp}}(r)$	$\frac{q_A q_B}{r} f_{\text{overlap}}(r)$
Gordon	$\frac{Z_A Z_B}{r}$	$\frac{Z_A (c_B - V_B)}{r} f_{\text{damp}}(r)$	$\frac{Z_B (c_A - V_A)}{r} f_{\text{damp}}(r)$	$\frac{(c_A - V_A)(c_B - V_B)}{r} f_{\text{overlap}}(r)$
Piquemal	$\frac{V_A V_B}{r}$	$\frac{V_A (c_B - V_B)}{r} f_{\text{damp}}(r)$	$\frac{V_B (c_A - V_A)}{r} f_{\text{damp}}(r)$	$\frac{(c_A - V_A)(c_B - V_B)}{r} f_{\text{overlap}}(r)$
Truhlar	$\frac{(c_A + n_A)(c_B + n_B)}{r}$	$\frac{(c_A + n_A)n_B}{r} f_{\text{damp}}(r)$	$\frac{(c_B + n_B)n_A}{r} f_{\text{damp}}(r)$	$\frac{n_A n_B}{r} f_{\text{overlap}}(r)$

Parameterization Strategies

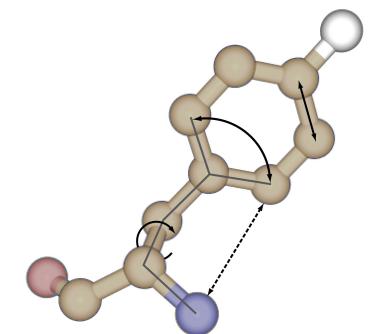
- Many Popular Force Fields: CHARMM, AMBER, OPLS

$$\begin{aligned} \mathbf{U}(\mathbf{r}) = & \sum_{\text{bonds}} k_b (r - r_0)^2 + \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2 + \sum_{\text{torsions}} A [1 - \cos(n\tau - \phi)] \\ & + \sum_{i,j \in} 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right] + \sum_{i,j \in} \frac{q_i q_j}{r_{ij}} \end{aligned}$$



Common Parameterization Strategies:

- Top line from X-ray structures, quantum calculations, vibrational spectroscopy
- partial charges from fits to electrostatic potential from HF/6-31G*
- van der Waals ϵ ; σ from neat liquids (vs. water/solute simulations as in CHARMM)
- torsional parameters from quantum calculations
- further empirical corrections

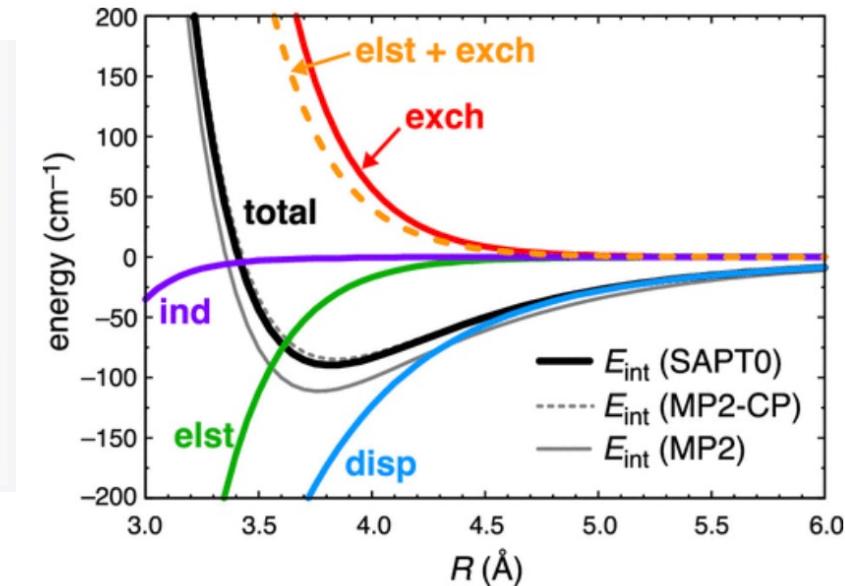
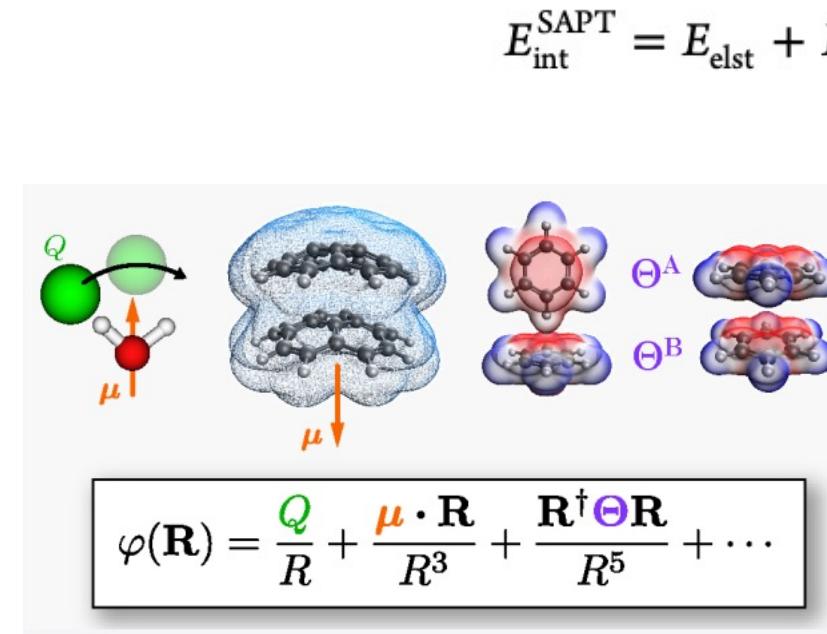


$$Z = \left[\sum_i^{\text{Observables}} \sum_j^{\text{Occurrences}} \frac{(\text{calc}_{i,j} - \text{expt}_{i,j})^2}{w_i^2} \right]^{1/2}$$

Torsion terms make some *ab initio* force field developers furious



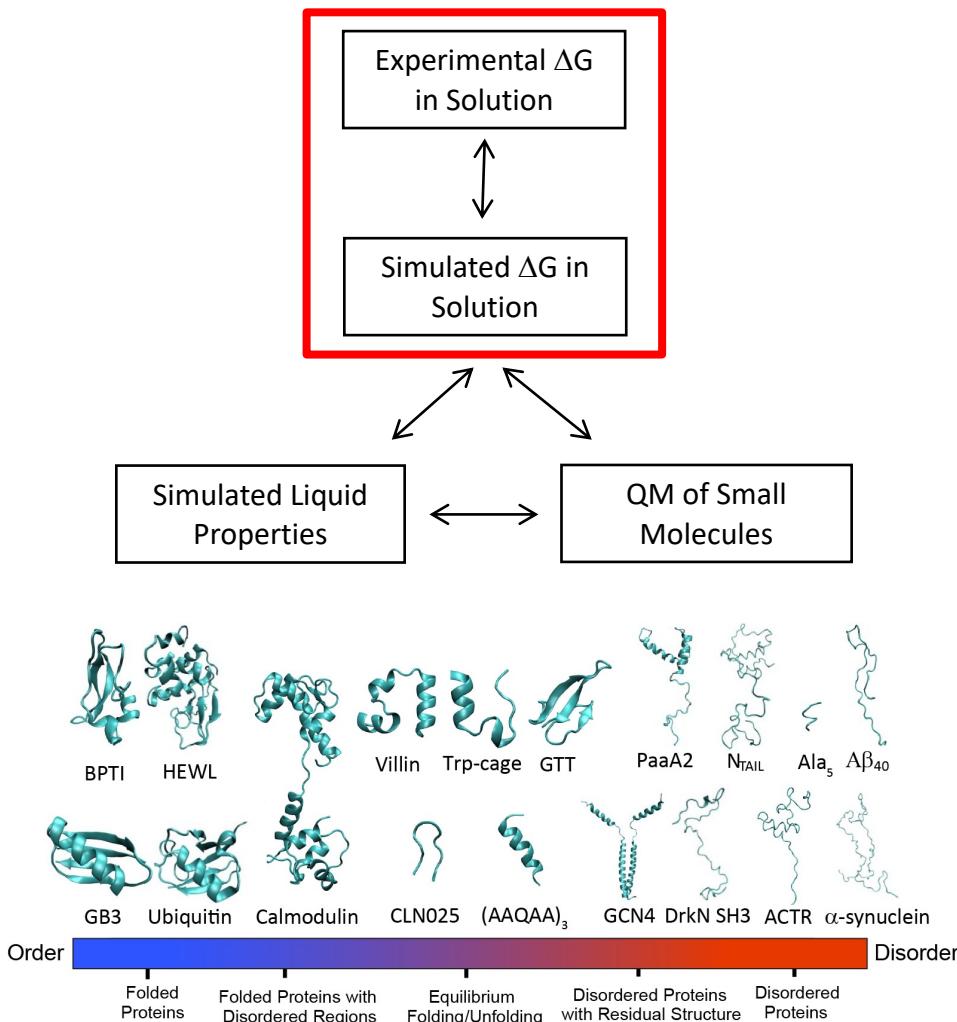
Symmetry Adapted Perturbation Theory (SAPT)



Are torsion terms a garbage can to collect effects that we, due to personal failings and ineptitude, have failed to capture with accurate, physics-based interatomic potentials?

<https://pubs.acs.org/doi/pdf/10.1021/acs.jpca.1c05962>

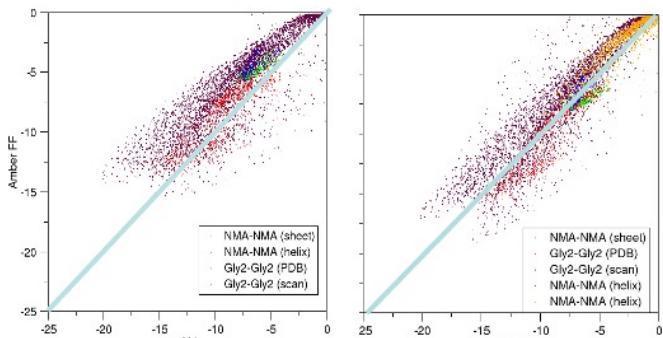
...but they (currently) work better than *ab initio* force fields



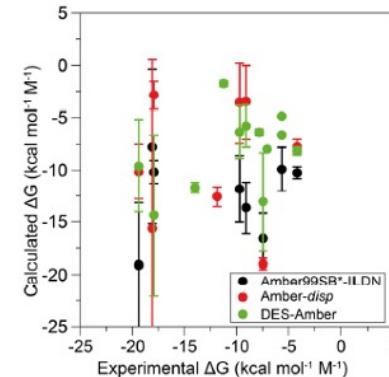
Protein	Experimental Measurements
Disordered Peptides	
Ala ₅	$^1J_{NC\alpha}$, $^2J_{NC\alpha}$, $^3J_{H\alpha C}$, $^3J_{HNC'}$, $^3J_{HNC\beta}$, $^3J_{HNH\alpha}$, $^3J_{HNC\alpha}$, and $^3J_{H\alpha C}$ backbone scalar couplings ¹
(AAQAA) ₃	Temperature-dependent helical fraction ²
CLN025	Temperature-dependent hairpin fraction ³
Fast-Folding Proteins	
Villin	Temperature-dependent native-state stability ⁴
Trp-cage	Temperature-dependent native-state stability ⁵
GTT	Temperature-dependent native-state stability ⁶
Folded Proteins	
Ubiquitin	$^3J_{HNH\alpha}$, $^3J_{HNC}$, $^3J_{HNC\beta}$, and $^3J_{H\alpha C}$ backbone scalar couplings, ^{7,8} hydrogen bond 3H scalar couplings ^{9,10} , $^3J_{H\alpha H\beta}$, $^3J_{CC'}$, and $^3J_{NC}$ side-chain scalar couplings, ^{8,11} backbone RDCs, ^{12,13} side-chain RDCs, ¹³ backbone amide S2 order parameters, ¹⁴ side-chain methyl S2 order parameters ¹⁵
GB3	$^3J_{HNH\alpha}$, $^3J_{HNC}$, $^3J_{HNC\beta}$, and backbone scalar couplings, ¹⁶ hydrogen bond 3H scalar couplings ¹⁷ , $^3J_{H\alpha H\beta}$, $^3J_{CC'}$, and $^3J_{NC}$ side-chain scalar couplings ¹¹ , backbone RDCs ¹⁸ , and backbone amide S2 order parameters ¹⁹
HEWL	backbone RDCs ²⁰ , backbone amide S2 order parameters ²¹ , side-chain methyl S2 ²² , $^3J_{H\alpha H\beta}$ and $^3J_{CC'}$ side-chain scalar couplings ²³
BPTI	$^3J_{HNH\alpha}$ backbone scalar couplings, ²⁴ $^3J_{H\alpha H\beta}$ side-chain scalar couplings ²⁵ , backbone RDCs ²⁶
Disordered Proteins	
drKN	Backbone chemical shifts ²⁷ , backbone RDCs ²⁸ , backbone $^3J_{HNH\alpha}$ scalar couplings ²⁷ , PREs ²⁹ , Rg ³⁰
ACTR	Backbone chemical shifts ³¹ , backbone RDCs ³¹ , PREs ³¹ , Rg ³²
NTAIL	Backbone chemical shifts ³³ , backbone RDCs ³⁴ , Rg ³⁴
α -synuclein	Backbone chemical shifts ³⁵ , backbone RDCs ³⁶ , backbone $^3J_{HNH\alpha}$ and $^3J_{CC'}$ scalar couplings ³⁷ , PREs ^{36,38-41} , Rg ^{42,43}
PaaA2	Backbone chemical shifts ⁴⁴ , backbone RDCs ⁴⁴ , Rg ⁴⁴
AB40	Backbone chemical shifts ⁴⁵ , backbone RDCs ⁴⁶ , backbone $^3J_{HNH\alpha}$ scalar couplings ⁴⁷ , Rg ⁴⁸
Partially Disordered Dimer	
GCN4	Backbone chemical shifts ⁴⁹ , backbone amide S2 order parameters ⁴⁹
Multi-Domain Protein	
CaM	Backbone chemical shifts ⁵⁰ , backbone RDCs ⁵¹ , Rg ⁵²

...but they (currently) work better than *ab initio* force fields

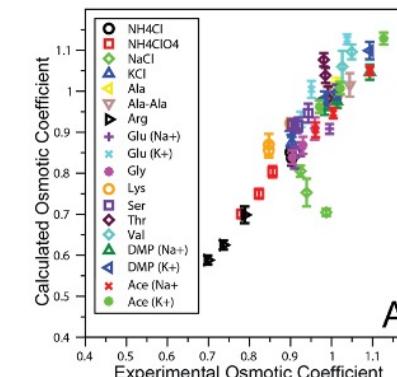
Quantum Calculations



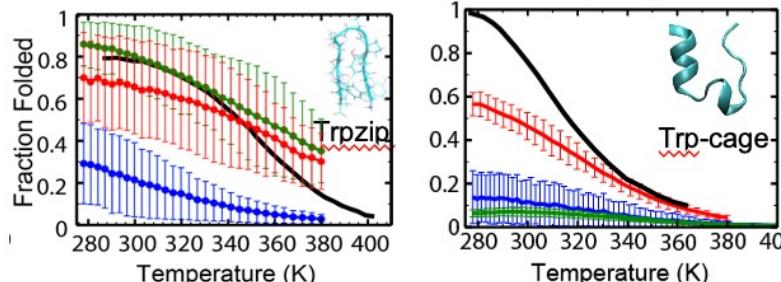
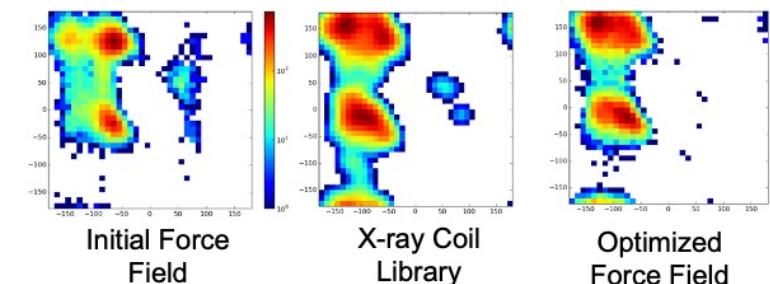
Protein Complex Stability



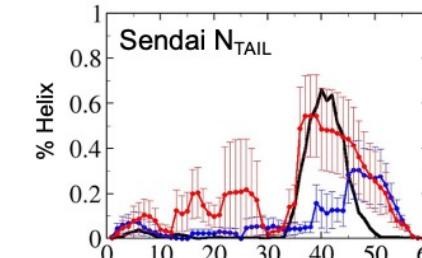
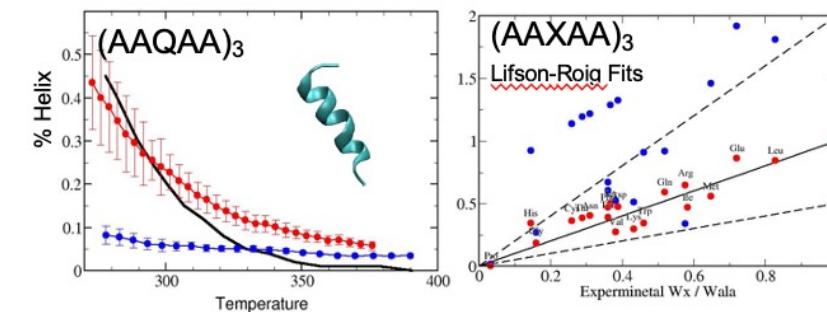
Solution Properties



Structural Database Optimizations

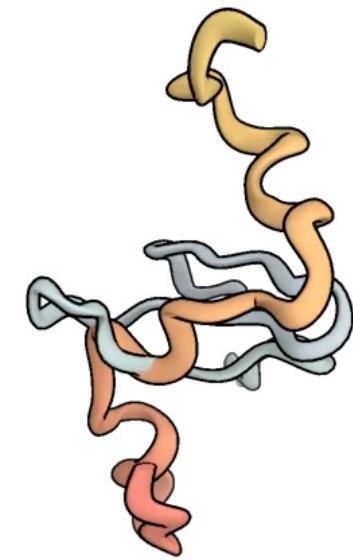
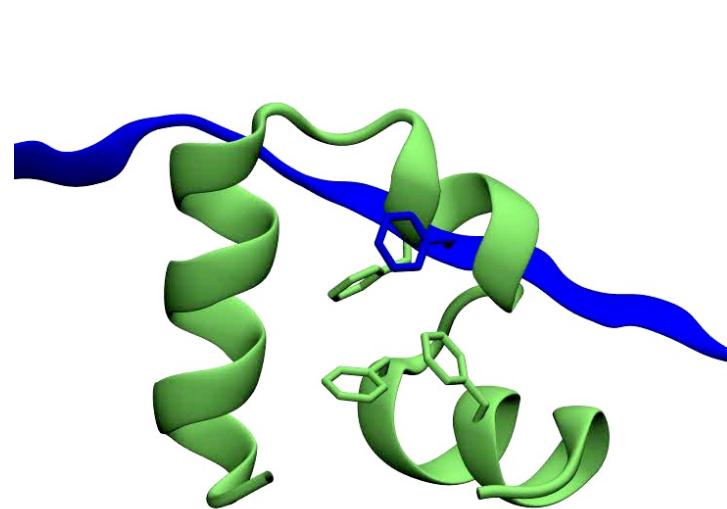
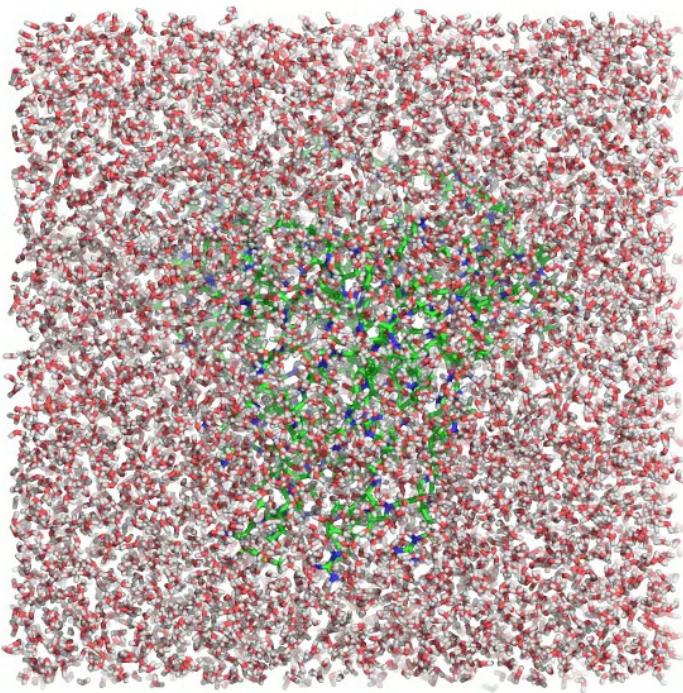


Solution Free Energies

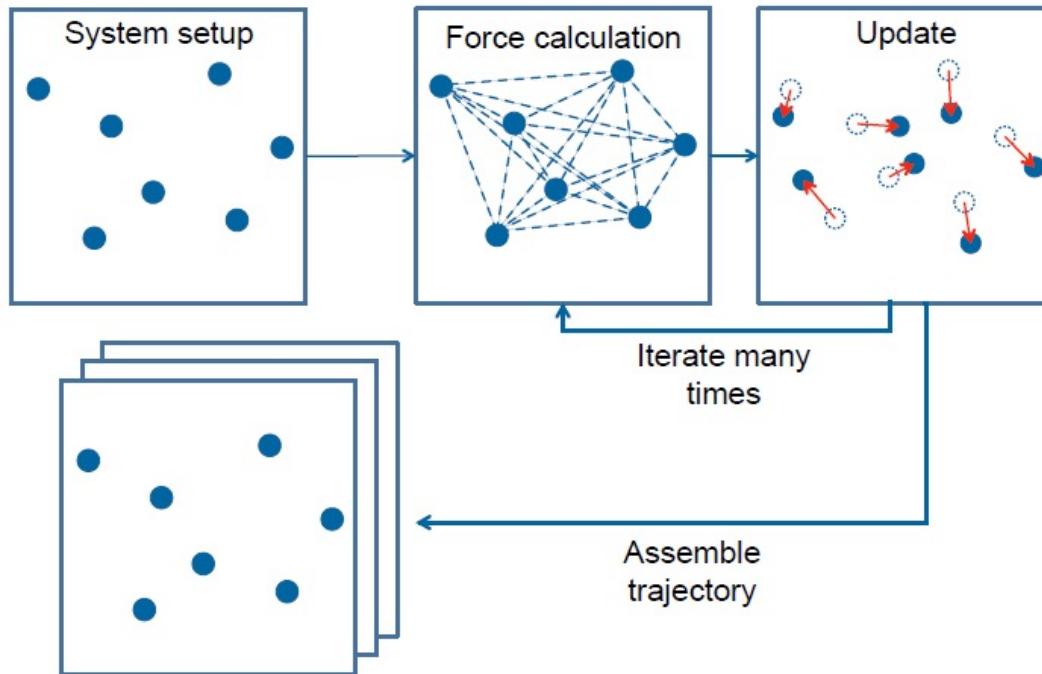


DES-amber Force Field: Piana*, Robustelli* et. al. JCTC, 2020

Running MD Simulations



Molecular Dynamics Simulations Solve Newton's Equations of Motion



1. Calculate forces on each particle.

$$\mathbf{f}_i = m_i \mathbf{a}_i = -\nabla_{\mathbf{r}_i} U$$

$$\mathbf{f}_i = \begin{bmatrix} f_{ix} \\ f_{iy} \\ f_{iz} \end{bmatrix} = \begin{bmatrix} -\partial U / \partial x_i \\ -\partial U / \partial y_i \\ -\partial U / \partial z_i \end{bmatrix}$$

2. Propagate the system by integrating Newton's equations of motion.

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{f}_i = -\nabla_{\mathbf{r}_i} U$$

Molecular Dynamics

Program MYMD

```
call init  
t = 0  
do while (t .lt. tmax)  
    call force (x, f, en)  
    call integrate (x, f, en)  
    t = t + delt  
    call sample  
enddo  
stop  
end
```

simple MD program

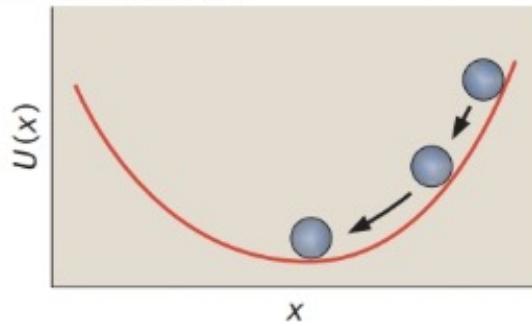
initialization

MD loop
calculate the force
integrate equation of motion

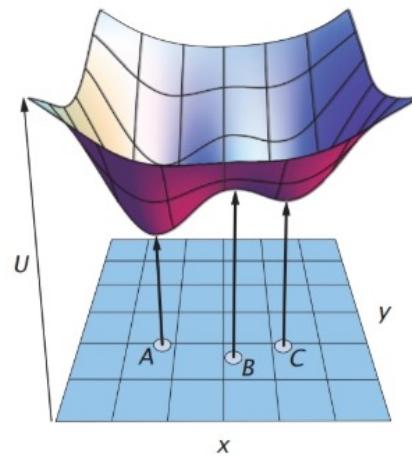
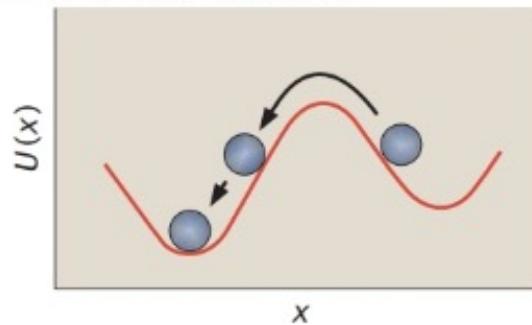
sample averages

Molecular Dynamics Simulations Solve Newton's Equations of Motion

(A) minimization



(B) molecular dynamics



1. Calculate forces on each particle.

$$\mathbf{f}_i = m_i \mathbf{a}_i = -\nabla_{\mathbf{r}_i} U$$

$$\mathbf{f}_i = \begin{bmatrix} f_{ix} \\ f_{iy} \\ f_{iz} \end{bmatrix} = \begin{bmatrix} -\partial U / \partial x_i \\ -\partial U / \partial y_i \\ -\partial U / \partial z_i \end{bmatrix}$$

2. Propagate the system by integrating Newton's equations of motion.

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{f}_i = -\nabla_{\mathbf{r}_i} U$$

Molecular Dynamics Simulations Solve Newton's Equations of Motion

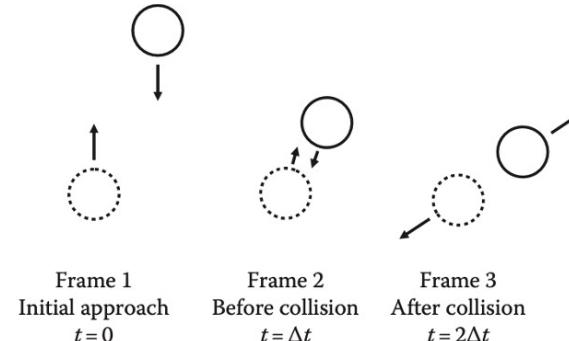
1. Calculate forces on each particle.

$$\mathbf{f}_i = m_i \mathbf{a}_i = -\nabla_{\mathbf{r}_i} U$$

$$\mathbf{f}_i = \begin{bmatrix} f_{ix} \\ f_{iy} \\ f_{iz} \end{bmatrix} = \begin{bmatrix} -\partial U / \partial x_i \\ -\partial U / \partial y_i \\ -\partial U / \partial z_i \end{bmatrix}$$

2. Propagate the system by integrating Newton's equations of motion.

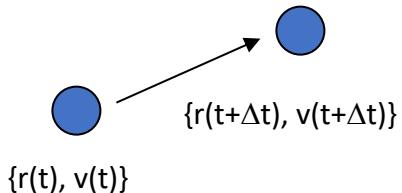
$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{f}_i = -\nabla_{\mathbf{r}_i} U$$



$$v(t + \Delta t) = v(t) + a(t)\Delta t = v(t) + \left[\frac{f(t)}{m} \right] \Delta t.$$

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \left[\frac{f(t)}{2m} \right] (\Delta t)^2,$$

Integrator: Verlet Algorithm



In MD simulations, we can create an initial configuration $\{q_i\}$ and assign random velocities (or thus momenta $\{p_i\}$) to particles. Instead of solving the coupled ODEs, we use numerical algorithms to integrate the Newton's equations of motion. A popular algorithm is the Verlet integrator, which can be derived by Taylor expanding the coordinate of a particle around time t :

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \frac{\ddot{r}}{6}\Delta t^3 + O(\Delta t^4)$$

$$r(t - \Delta t) = r(t) - v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 - \frac{\ddot{r}}{6}\Delta t^3 + O(\Delta t^4)$$

Combining the two equations above we have:

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

so that we can estimate the new position of a particle at time $t + \Delta t$ with an error of order Δt^4 :

$$r(t + \Delta t) \approx 2r(t) - r(t - \Delta t) + \frac{f(t)}{m}\Delta t^2$$

in which the force $f(t)$ can be obtained as:

$$f(t) = -\frac{\partial U(r)}{\partial r}$$

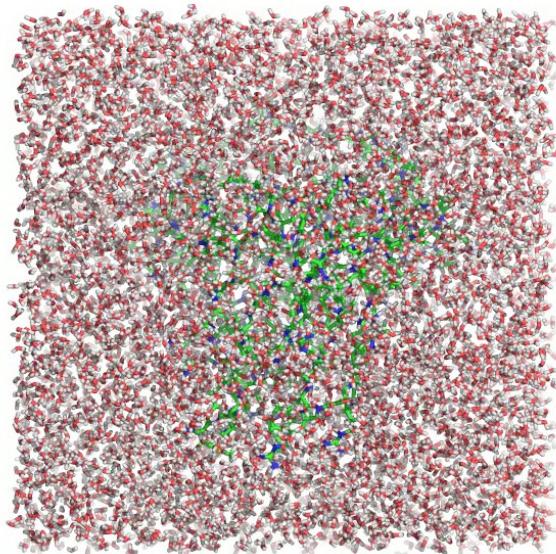
The above integration only updates particle locations. Nevertheless, we can also compute the velocity in order to measure the temperature of the system:

$$v(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t} + O(\Delta t^2)$$

Molecular Dynamics Thermostats & Barostats

The temperature is given by the total kinetic energy of the N -particle system:

$$E_{\text{kin}} = \frac{1}{2} \sum_{i=1}^N m_i v_i^2$$



From this the absolute temperature T can be computed using:

$$\frac{1}{2} N_{\text{df}} k T = E_{\text{kin}}$$

where k is Boltzmann's constant and N_{df} is the number of degrees of freedom

G. Bussi, D. Donadio, and M. Parrinello, "Canonical sampling through velocity rescaling," *J. Chem. Phys.*, **126** 014101 (2007).

<https://manual.gromacs.org/documentation/2021.2/reference-manual/algorithms/molecular-dynamics.html>

$$dK = (K_0 - K) \frac{dt}{\tau_T} + 2 \sqrt{\frac{KK_0}{N_f}} \frac{dW}{\sqrt{\tau_T}},$$

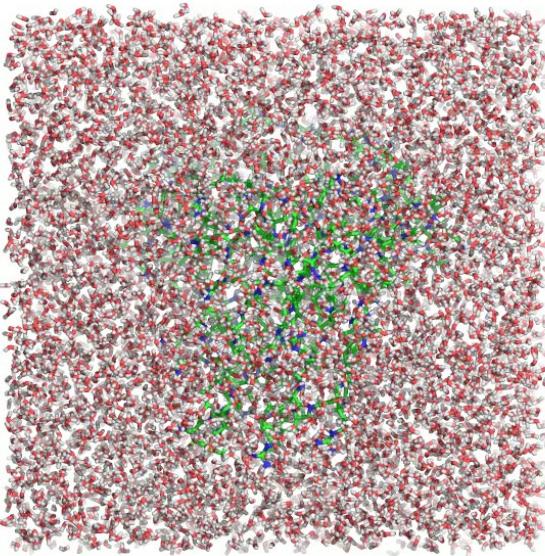
<https://aip.scitation.org/doi/10.1063/1.2408420>

where K is the kinetic energy, N_f the number of degrees of freedom and dW a Wiener process. There are no additional parameters, except for a random seed. This thermostat produces a correct canonical ensemble and still has the advantage of the Berendsen thermostat: first order decay of temperature deviations and no oscillations. Wiener process is a real valued [continuous-time stochastic process](#)

Molecular Dynamics Thermostats & Barostats

Virial and pressure

The scalar virial of the system due to the dispersion interaction between two particles i and j is given by:



$$\Xi = -\frac{1}{2} \mathbf{r}_{ij} \cdot \mathbf{F}_{ij} = 3 C_6 r_{ij}^{-6}$$

The pressure is given by:

$$P = \frac{2}{3V}(E_{kin} - \Xi)$$

The long-range correction to the virial is given by:

$$\Xi_{lr} = \frac{1}{2} N\rho \int_0^\infty 4\pi r^2 g(r)(\Xi - \Xi_c) dr$$

The pressure tensor \mathbf{P} is calculated from the difference between kinetic energy E_{kin} and the virial Ξ :

$$\mathbf{P} = \frac{2}{V}(E_{kin} - \Xi)$$

where V is the volume of the computational box. The scalar pressure P , which can be used for pressure coupling in the case of isotropic systems, is computed as:

$$P = \text{trace}(\mathbf{P})/3$$

The virial Ξ tensor is defined as:

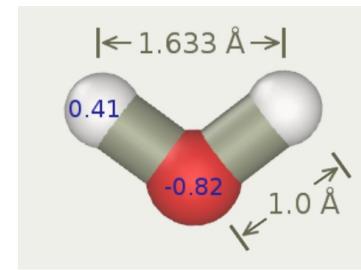
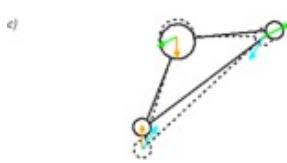
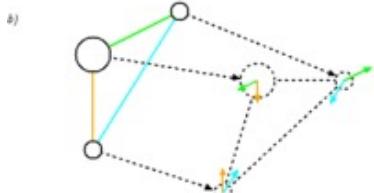
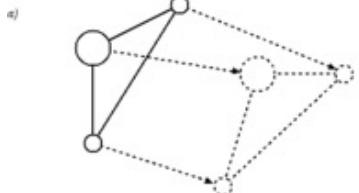
$$\Xi = -\frac{1}{2} \sum_{i < j} \mathbf{r}_{ij} \otimes \mathbf{F}_{ij}$$

The GROMACS implementation of the virial computation is described in sec. [Virial and pressure](#)

Molecular Dynamics Constraints: SHAKE, LINCS, SETTLE

$$v(t + \Delta t) = v(t) + a(t)\Delta t = v(t) + \left[\frac{f(t)}{m} \right] \Delta t.$$

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \left[\frac{f(t)}{2m} \right] (\Delta t)^2,$$



3 LINCS

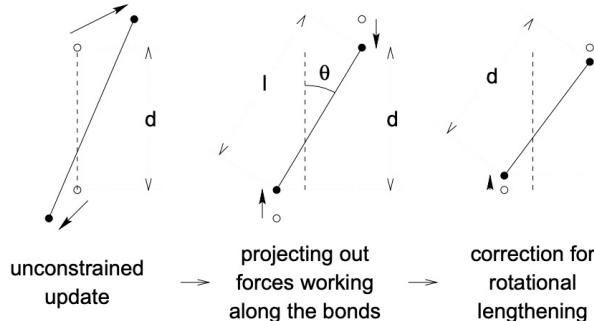


Figure 3.1: Schematic picture showing the three position updates needed for one time step. The dashed line is the old bond of length d , the solid lines are the new bonds. $l = d \cos \theta$ and $p = (2d^2 - l^2)^{\frac{1}{2}}$.

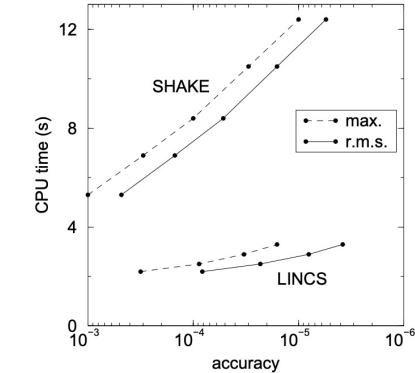


Figure 3.2: The CPU time for the constraint algorithms in lysozyme plotted as a function of the accuracy, i.e. the relative deviation of the constraints averaged over 500 steps. The dashed lines show the maximum deviation and the solid lines show the r.m.s. deviation for SHAKE and for LINCS.

<http://www.cs.rug.nl/~bekker/publications/lincs.pdf>

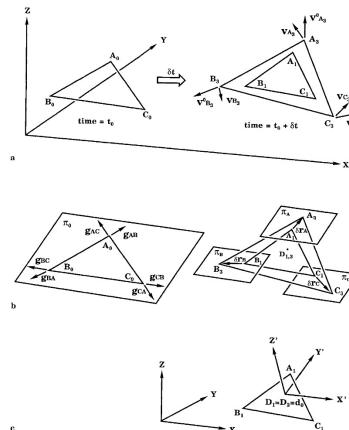


Figure 1. Schematic drawing of the geometric basis of the SETTLE method. (a) $\Delta A_0 B_0 C_0$ is a triangle at time t_0 , $\Delta A_1 B_1 C_1$ is the triangle that would have been reached at time $t_0 + \delta t$ in the absence of any constraints, and $\Delta A_3 B_3 C_3$ is the corresponding one after applying constraints. $v_{A_0}^0$, $v_{B_0}^0$, and $v_{C_0}^0$ and v_{A_3} , v_{B_3} , and v_{C_3} are velocities of vertices before and after adding correction velocities at time $t_0 + \delta t$, respectively. (b) The plane π_0 includes $\Delta A_0 B_0 C_0$. Three planes, π_A , π_B , and π_C , are parallel to π_0 . Each of them includes one of the apexes of A_1 , B_1 , and C_1 , respectively. The constraint forces, \mathbf{g}_{ij} , are directed along the bond at time t_0 . Each of the displacement vectors, $\delta\mathbf{r}_i$, is therefore on π_A , π_B , or π_C , respectively, with the resulting positions of the vertices A_3 , B_3 , and C_3 lying on each of those planes. D_1 and D_3 are centers of mass of $\Delta A_1 B_1 C_1$ and $\Delta A_3 B_3 C_3$, respectively. They coincide with each other because the sum of the constraint forces is zero by definition. (c) An alternative orthogonal coordinate system $X'Y'Z'$ with its origin at $D_1 = D_3 = d_0$. The $X'Y'$ plane is parallel to π_0 and the $Y'Z'$ plane includes A_1 . Since π_0 and A_1 are available after an unconstrained step, the matrix of orthogonal transformation is uniquely defined.

SETTLE

<https://onlinelibrary.wiley.com/doi/epdf/10.1002/jcc.540130805>

[https://en.wikipedia.org/wiki/Constraint_\(computational_chemistry\)](https://en.wikipedia.org/wiki/Constraint_(computational_chemistry))

<https://manual.gromacs.org/current/reference-manual/algorithms/constraint-algorithms.html?highlight=constraints>

Full MD Update Algorithm

Given: Positions \mathbf{r} of all atoms at time t Velocities \mathbf{v} of all atoms at time $t - \frac{1}{2}\Delta t$ Accelerations \mathbf{F}/m on all atoms at time t . (Forces are computed disregarding any constraints) Total kinetic energy and virial at $t - \Delta t \Downarrow$

1. Compute the scaling factors λ and μ according to (29) and (45) \Downarrow
2. Update and scale velocities: $\mathbf{v}' = \lambda(\mathbf{v} + \mathbf{a}\Delta t)$ \Downarrow
3. Compute new unconstrained coordinates: $\mathbf{r}' = \mathbf{r} + \mathbf{v}'\Delta t$ \Downarrow
4. Apply constraint algorithm to coordinates: constrain($\mathbf{r}' \rightarrow \mathbf{r}''; \mathbf{r}$) \Downarrow
5. Correct velocities for constraints: $\mathbf{v} = (\mathbf{r}'' - \mathbf{r})/\Delta t$ \Downarrow
6. Scale coordinates and box: $\mathbf{r} = \mu\mathbf{r}''; \mathbf{b} = \mu\mathbf{b}$

Periodic Boundary Conditions

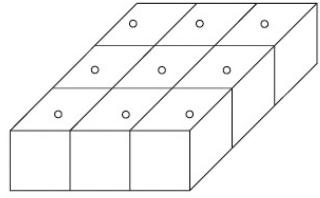
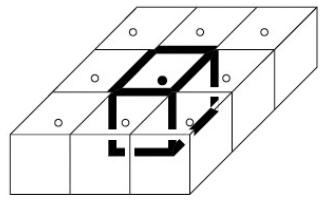
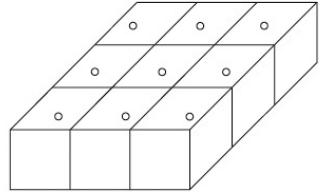


Figure 3.6 Exploded view of a cubic simulation cell surrounded by the 26 periodic images generated by PBCs. If the solid particle translates to a position that is outside the simulation cell, one of its periodic images, represented by open particles, will translate in

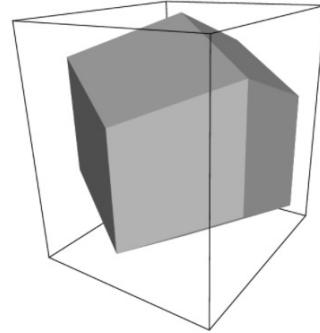


Fig. 2 A rhombic dodecahedron

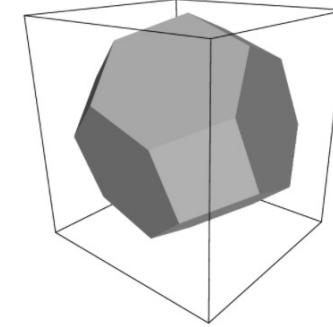
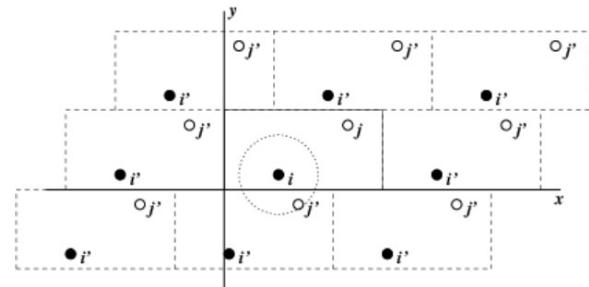
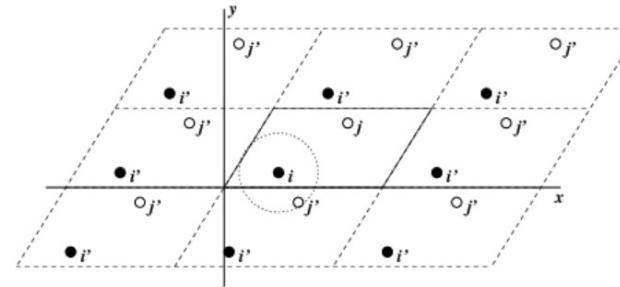


Fig. 3 A truncated octahedron



Cutoffs and Particle Mesh Ewald Summations

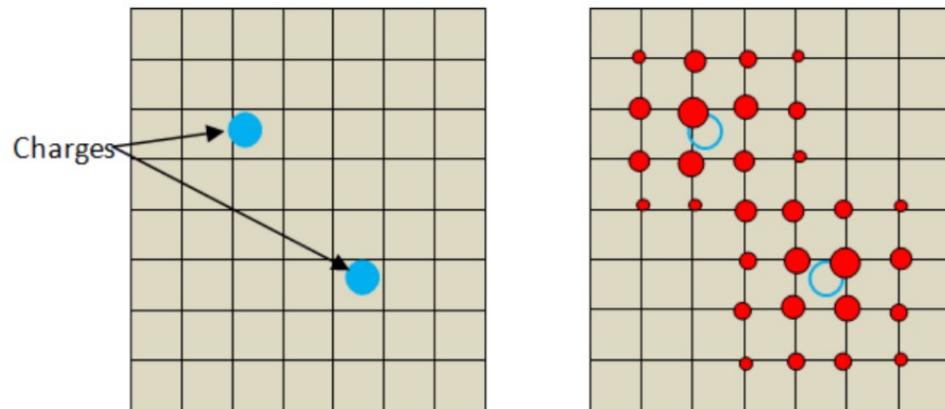
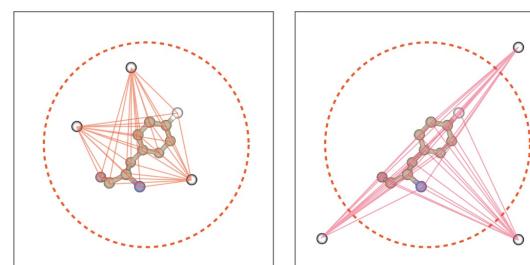
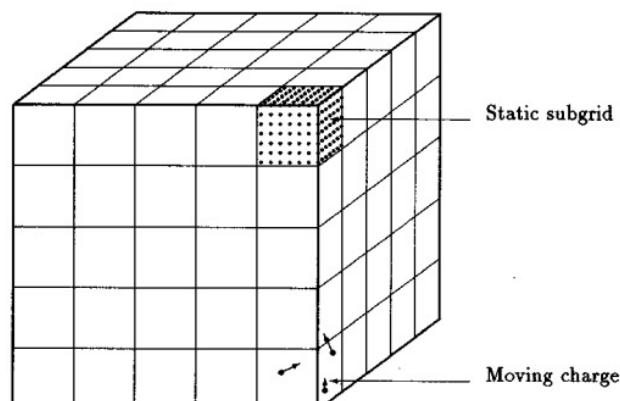


Figure 2.2: Example of charge spreading in 2D for interpolation order $l = 4$; circle sizes signify absolute values of charge contributions



$$\begin{aligned}
 & + \sum_{i,j \in \text{c}} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \\
 & + \sum_{i,j \in \text{c}} \frac{q_i q_j}{r_{ij}}
 \end{aligned}$$

2.5 Stages of the PME algorithm

The PME algorithm part for computing the reciprocal energy and forces can be summed up as follows:

1. Calculate the B-spline interpolation values for all particles (equation (2.5)).
2. Spread the particle charges on a discrete 3D grid (using the spline values), calculating the array Q (equation (2.6)).
3. Perform three-dimensional real-to-complex FFT of the grid.
4. Calculate the reciprocal energy contribution (equation (2.7)), transform the grid.
5. Perform three-dimensional complex-to-real FFT of the transformed grid.
6. Gather the particle forces from the grid (using the spline values) (equation (2.9)).

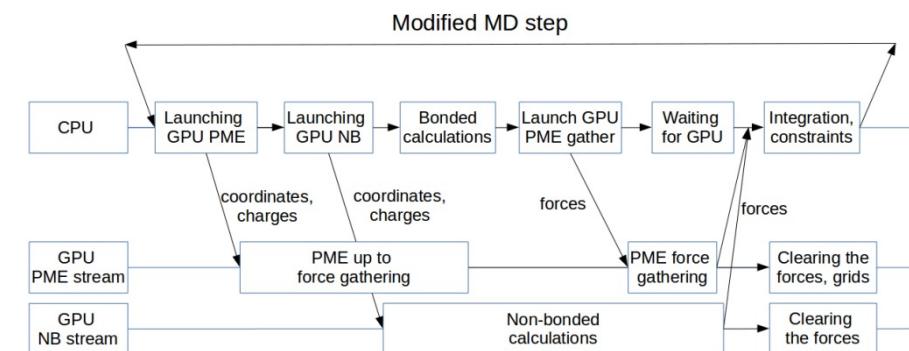


Figure 4.2: Timeline of a modified MD step with PME GPU

The Ergodic Hypothesis

$$\langle A \rangle_{ensemble} = \langle A \rangle_{time}$$

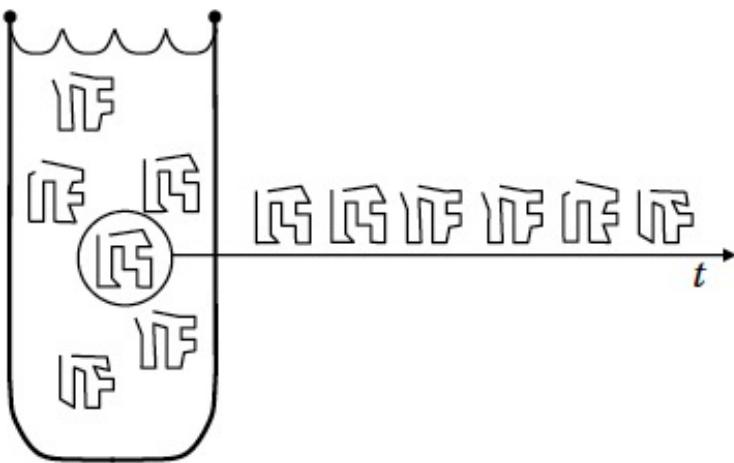


FIGURE 4.2 Why time and ensemble averages have to be the same. On the left is a schematic test tube containing an ensemble of molecules in different configurations, while the time evolution of one of the molecules is shown schematically on the right. Nature is only capable of producing forces and dynamics, so each individual molecule evolves according to the basic laws of physics. If we take a snapshot of all molecules in a test tube, because their dynamics are essentially uncorrelated, we will find the equilibrium ensemble. In other words, the equilibrium ensemble exactly represents the fraction of time a molecule spends in different configurations. Configurations appear more commonly in the equilibrium ensemble because they take up more time in a “movie” of any single molecule.

Does Equilibrium Exist?



Phase Space

$$\mathbf{X}' = (x_1, y_1, z_1, p_{x,1}, p_{y,1}, p_{z,1}, x_2, y_2, z_2, p_{x,2}, p_{y,2}, p_{z,2}, \dots) \quad (3.1)$$

For ease of notation, the position coordinates and momentum coordinates are defined as

$$\mathbf{q} = (x_1, y_1, z_1, x_2, y_2, z_2, \dots) \quad (3.2)$$

$$\mathbf{p} = (p_{x,1}, p_{y,1}, p_{z,1}, p_{x,2}, p_{y,2}, p_{z,2}, \dots) \quad (3.3)$$

allowing us to write a (reordered) phase space point as

$$\mathbf{X} = (\mathbf{q}, \mathbf{p}) \quad (3.4)$$

$$\langle A \rangle = \iint A(\mathbf{q}, \mathbf{p}) P(\mathbf{q}, \mathbf{p}) d\mathbf{q} d\mathbf{p} \quad P(\mathbf{q}, \mathbf{p}) = Q^{-1} e^{-E(\mathbf{q}, \mathbf{p})/k_B T} \quad Q = \iint e^{-E(\mathbf{q}, \mathbf{p})/k_B T} d\mathbf{q} d\mathbf{p}$$

For many properties of interest, we can factor phase space into independent contributions for position and momenta:

$$\langle A \rangle = \int A(\mathbf{q}) \left[\int P(\mathbf{p}, \mathbf{q}) d\mathbf{p} \right] d\mathbf{q} + \int A(\mathbf{p}) \left[\int P(\mathbf{p}, \mathbf{q}) d\mathbf{q} \right] d\mathbf{p} \quad \langle A \rangle = \int A(\mathbf{q}) P(\mathbf{q}) d\mathbf{q} + \int A(\mathbf{p}) P(\mathbf{p}) d\mathbf{p}$$

Metropolis Monte Carlo Simulations

$$\langle A \rangle = \int A(\mathbf{q}) \left[\int P(\mathbf{p}, \mathbf{q}) d\mathbf{p} \right] d\mathbf{q} + \int A(\mathbf{p}) \left[\int P(\mathbf{p}, \mathbf{q}) d\mathbf{q} \right] d\mathbf{p} \quad \langle A \rangle = \int A(\mathbf{q}) P(\mathbf{q}) d\mathbf{q} + \int A(\mathbf{p}) P(\mathbf{p}) d\mathbf{p}$$

In MD we sample properties based on a continuous time trajectory:

$$\langle A \rangle = \frac{1}{M} \sum_i^M A(t_i) \quad \langle A \rangle = \lim_{t \rightarrow \infty} \frac{1}{t} \int_{t_0}^{t_0+t} A(\tau) d\tau$$

Metropolis Breakthrough: “Instead of choosing configurations randomly, then weighting them with $\exp(-E/k_B T)$, we choose configurations with a probability $\exp(-E/k_B T)$ and weight them evenly”

$$\langle A \rangle = \frac{1}{X} \sum_{i=1}^X A(\mathbf{q}_i)$$

Where configuration \mathbf{q}_i is sampled with a probability based on the metropolis criteria:

$$p = \min \left[1, \frac{\exp(-E_2/k_B T)}{\exp(-E_1/k_B T)} \right]$$

Monte Carlo Simulations

Another way to generate a canonical ensemble

- (i). Randomly generate a new conformation. For proteins, usually in torsional space, such as rotate around a dihedral angle
- (ii). Accept or reject the new conformation based on the Metropolis criterion:

$$T(x \rightarrow x') = \begin{cases} 1 & \text{for } \Delta \leq 0 \\ \exp(-\Delta) & \text{for } \Delta > 0 \end{cases}$$

$$\Delta = \frac{1}{kT} (U(x') - U(x))$$

Monte Carlo vs. Molecular Dynamics

- Only energy is needed
- Straightforward to perform NVT and NPT
- Easy to constrain some degrees of freedoms (not include them in trials)
- For some systems, large motions can be made (LJ particles) between consecutive configurations
- Hard to make trials for complex systems, such as proteins, since proteins move collectively
- Step size decreases with system size
- Can't easily get kinetic information
- Both energy and force are needed
- Requires temperature and pressure control for NVT and NPT
- Needs special techniques to constrain some degrees of freedoms
- The consecutive configurations are very similar
- MD can move simple and complex systems the same way
- Same time step can be used for small or large systems
- Can generate kinetic data as well as thermodynamic data

Monte Carlo Simulations For Protein Structure Prediction

