Optimization of CHARMM36-UA Force Field Parameters for Phospholipid Bilayer Simulation

CMSC-663/664
10/5/2017

Student:
Wei Shan Chin
wschin@terpmail.umd.edu

Advisor:
Dr. Jeff Klauda
Associate Professor of Chemical Engineering
UMD College Park
jbklauda@umd.edu
Background Information – Why Molecular Dynamics?

• Quantum Mechanics – infeasible for large biomolecules (>~10^4 atoms, e.g. phospholipid bilayer), biologically relevant time scale (10^-6-10^-3s)

• Molecular dynamics (partial solution?) – assume atoms follow Newton’s law
  o Discretize time into intervals (~10^-15 s)
  o At every time step:
    ▪ Compute forces acting on each atom using force field
    ▪ Update position and velocity of each atom
      ▪ Based on Newton’s Law of Motion \( \frac{\partial^2 r_i}{\partial t^2} = \frac{F}{m_i} \)
      ▪ Note that \( F(r) = -\nabla V(r) \)

• Force field: \( E = f(atomic \ positions) \)
  • Semi empirical
  • Fitted to quantum mechanics/experimental data
Background Information – More on Molecular Dynamics

• Molecular dynamics in different statistical ensembles
  o Canonical (NVT)
    • Number of particles, system volume and temperature = constant
    • Requires the particles to interact with a thermostat
  o Isobaric-isothermal (NPT)
    • Number of particles, pressure and temperature = constant
    • Requires particles to interact with a thermostat and barostat

• Thermostat/barostat: uses heat bath/piston which allow heat/volume exchange
Background Information – Force Field Functional Form

\[ V = V_{nb} + V_b \]

\[ V_b = \sum_{\text{bonds}} k_b(b - b_o)^2 + \sum_{\text{angles}} k_\theta(\theta - \theta_o)^2 + \sum_{\text{dihedrals}} k_\Phi[1 + \cos(n\Phi - \delta)] \]

\[ V_{nb} = \sum_{\text{nonbonded}} \epsilon_{ij} \left[ \left( \frac{R_{\text{min},ij}}{r_{ij}} \right)^{12} - 2 \left( \frac{R_{\text{min},ij}}{r_{ij}} \right)^6 \right] \]

where \( V_b \) : intramolecular (bonded terms), 
\( V_{nb} \) : intramolecular (non-bonded terms)

Note: Higher order bonded terms and long range electrostatic non-bonded terms may be included depending on the application

Background Information – Atomic Representations

• Phospholipid head:
  • All-atom
  • CHARMM36 Lipid Force Field

• Phospholipid tail:
  • United-atom (except the first methylene group): no explicit hydrogen
  • CHARMM36-UA

• Model molecules:
  • n-pentadecane
  • methyl acetate
  • cis-5-decene
  • methyl hexanoate

1. CHARMM. (Chemistry at HARvard Macromolecular Mechanics).

1. CHARMM. (Chemistry at HARvard Macromolecular Mechanics).
Goal

• Existing CHARMM36-UA parameters overestimates density of lipid tails (optimized via manual adjustment)

• This project will hopefully go some way towards addressing that by developing an automated tool for force field parameter optimization

• In addition, this work will involve testing the model against various properties that were not explicitly considered by the previously developed model
Approach - Overview

- **Molecular Dynamics Simulation**
- **MD Prediction:**
  - Energies, densities, isothermal compressibility, heat of vaporization.
- **Reference Data:**
  - Quantum mechanical calculations and experiments.
- **Objective Function:**
  \[ \left( y_{i, \text{exp}} - y_{i, \text{mod}} \right)^2 + \omega_i^2 \]
  \[ \nabla \omega, H_\omega \]
- **Force Field:**
  - CHARMM36-UA (Dihedral and LJ parameters)
- **Optimization Algorithm:**
  - Levenberg Marquardt (LM) Algorithm
- **Update parameters**
  - (initialize using parameters from a similar model)
- **Optimized parameters**
  - (if converged)
Approach – Basic Workflow for Running Molecular Dynamics Simulation

Inputs:
- initial positions, velocities
- topology file ("molecule-force field dictionary")
- force field parameter file

Minimization (Gradient descent/CG) and Equilibration (equilibrate to desired temperature and pressure)

Dynamics/Production (determine ensemble average of macroscopic properties)

Outputs: Positions, velocities, energies
Approach – How to Estimate Experimental Property

• Example: Density

• Compute the ensemble-averaged volume \( \langle V \rangle \) from equilibrated trajectory (NPT ensemble)

• Divide total mass of molecules in the box by \( \langle V \rangle \) to obtain density
Approach – How to Estimate Experimental Property

• Isothermal Compressibility
  \[ \kappa_T = -\frac{1}{V} \left( \frac{\partial V}{\partial P} \right)_T = \frac{1}{k_B T \langle V \rangle} [\langle V^2 \rangle - \langle V \rangle^2] \]

• Enthalpy of vaporization
  \[ \Delta H_{vap} = \langle H_g \rangle - \frac{\langle H_l \rangle}{N_{mol}} \]
  \[ \langle H_g \rangle = \langle E_g \rangle + k_B T ; \langle H_l \rangle = \langle E_l \rangle + P\langle V_l \rangle \]
  \[ \text{NVT, NPT simulation for gas and liquid phase respectively} \]

where \( k_B = \) Boltzmann’s constant, \( H = \) enthalpy
Approach – How to Estimate Parametric Derivative of Ensemble-averaged Properties

• Direct finite difference approximation susceptible to numerical noise – need an analytical estimate

• For an NPT ensemble simulation, the analytical form of ensemble averaged observable $A$ is as follows:

\[
\langle A \rangle_\omega = \frac{\int A(r, V; \omega) e^{-\frac{1}{k_B T} (E(r, V; \omega) + PV)} dr dV}{\int e^{-\frac{1}{k_B T} (E(r, V; \omega) + PV)} dr dV}
\]

• If $A$ does not directly depend on $\omega$ (density and isothermal compressibility),

\[
\frac{d\langle A \rangle_\omega}{d\omega} = -\frac{1}{k_B T} \left[ \langle A \frac{dE}{d\omega} \rangle_\omega - \langle A \rangle_\omega \langle \frac{dE}{d\omega} \rangle_\omega \right]
\]

Approach – How to Estimate Parametric Derivative of Ensemble-averaged Properties

• Exception (Heat of vaporization):

\[
\frac{d}{d\omega} \langle \Delta H_{\text{vap}} \rangle_{\omega} = \langle \frac{dE_g}{d\omega} \rangle - \frac{1}{k_B T} \left[ \langle H_g \frac{dE_g}{d\omega} \rangle - \langle H_g \rangle \langle \frac{dE_g}{d\omega} \rangle \right] - \\
\frac{1}{N_{\text{mol}}} \left[ \langle \frac{dE_l}{d\omega} \rangle - \frac{1}{k_B T} \left[ \langle H_l \frac{dE_l}{d\omega} \rangle - \langle H_l \rangle \langle \frac{dE_l}{d\omega} \rangle \right] \right]
\]

\[\langle H_g \rangle = \langle E_g \rangle + k_B T; \quad \langle H_l \rangle = \langle E_l \rangle + P\langle V_l \rangle\]
Approach - Optimization

• Parametrization strategy:
  o Density, isothermal compressibility & heat of vaporization: LJ parameters
  o QM dihedral PES: dihedral parameters
  o Decoupled from each other – can be done iteratively

• Least square objective function

\[ F = \sum_{y_i} k_{y_i} [y_{i,\text{exp}} - y_{i,\text{mod}}]^2 \]

where \( y = \) fitted property (density etc.), \( \omega = \) force field parameters
  o Might introduce regularization terms if necessary
Approach - Optimization

• Levenberg–Marquardt (LM) Algorithm
  - \([H + \lambda I] \Delta \omega = J\)
  - Small \(\lambda\) = Gauss Newton (when close to local opt.) and otherwise = gradient descent
  - Estimate the LSD component of \(H\) using Gauss Newton approximation \(J^T J\)
  - Convergence: gradient norm < specified threshold

\[
J_{ij} = \frac{\partial [y_{i,exp} - y_{i,mod}]}{\partial \omega_j}, \quad H = \text{Hessian matrix}
\]
Computational Considerations

• Optimization << MD simulation (in terms of computing time)
  o MD simulation of n-pentadecane (64 n-pentadecane molecules for 20 ns: 5 hours on 32 cores on Deepthought, using NAMD)
  o Computational cost associated with the LM solver is comparatively trivial
    • Recall that $J_{ij} = \frac{\partial [y_{i,\exp} - y_{i,\text{mod}}]}{\partial \omega_j}$
    • $\# \omega_j, \# y_i = O(10)$
    • Major computational expense comes from computing $\lambda$, $J$, $J^T J$
  o Same applies to the optimization of dihedral parameters
Approach - Optimization

• Dihedral parameter fitting

\[ V_{\text{dihedral}} = \sum_{k} \sum_{n=1}^{6} k_{\Phi,n}[1 + \cos(n\Phi_k - \delta_{nk})] \]

  - n up to 6: symmetry, prevent overfitting and avoid high frequency oscillations
  - \( \delta_{nk} = 0, 180^\circ \) (for odd and even n respectively);
    \( k_{\Phi,n} > 0 \)
  - Linear Least Square (LLS) problem
    \[ y = X\beta + \epsilon \Rightarrow [X^TX]\hat{\beta} = X^T y \]

Implementation

• Molecular simulation – freely available NAMD (Nanoscale Molecular Dynamics) and CHARMM
• Optimizer – implement in Python from scratch (or C if necessary)
• Reading and manipulating data from trajectory – open source Python library MDAnalysis + NumPy + Python + bash
• Hardware – UMD’s Deepthought and Deepthought2 HPC clusters through allocations provided by Dr. Klauda
Validation

• Confirm if analytical estimate of parametric derivatives is in good agreement with finite difference

• Run the optimizer for a few test cases and compare results to those obtained using SciPy’s Levenberg-Marquardt solver (Python interface to MINPACK’s implementation)
Test Problem

• Test problem
  o Model molecule (methane, \text{CH}_4)  
    • Single site - helpful for debugging, computationally cheaper
  o Properties considered in regression: density, isothermal compressibility, heat of vaporization at 39 °C (and 0 to 100 °C)
  o Validate if fitted model accurately simulates kinetic properties (diffusion coefficient and viscosity) – may include in regression if necessary
  o Verify if simulations agree with observed phase behavior – simulation at T > melting point
  o Compare the fitted parameter values to similar models from the literature
Expected Results

• The optimizer should take <40 iterations to converge to a local optimum
• The analytical derivatives of properties should agree well with finite difference estimates
• Simulation results should be in good agreement with experimentally observed densities, isothermal compressibilities and heats of vaporization
• The model should correctly reproduce the phase and kinetic behavior of the model molecules (deviation in viscosity and diffusion coefficient should be < 5-10%)
Schedule/Milestones

• Phase 1: Software Preparation (by early Nov)
  o Get familiar with NAMD/CHARMM
  o Write scripts/routines for computing and analyzing macroscopic properties, parametric derivatives etc.
  o Develop a rough implementation of optimizer

• Phase 2: Refinement/Debugging/Preliminary Testing (by mid Dec – early Feb)
  o Debug and refine code as necessary (e.g. making parameter update more automated) + some preliminary validation
Schedule/Milestones

• Phase 3: Validation and Continual Refinement/Debugging (by mid March)
  o Validate if the model performs well for the test case
  o Validate if fitted model accurately simulates kinetic properties (diffusion coefficient and viscosity) – may include in regression if necessary
  o Verify if model correctly simulates phase behavior (simulations at T > melting point)

• Phase 4: Validation and Continual Refinement/Debugging (by mid May)
  o Include other macroscopic properties of interest in regression/validation
  o Test the model for bigger systems (phospholipid membrane)
Deliverables

• Final version of code and scripts (as well as some relevant documentation)
• Datasets and optimization results for test case
• Mid year report and presentation
• Final report and presentation
Bibliography