Multi-scale Modeling & Simulation in Mechanics II

Homework 3: Multi-scale Modeling and Simulation of Macromolecules (relevant lectures: 2, 3, 4, 5, 6, 9)

Due ###

Introduction

Molecular simulations are commonly employed in the study of macromolecular systems, as they provide unique insights of the complex structure and dynamics of polymers at the molecular level. However, many of the interesting dynamic features of polymers occur at long time scales which are prohibitively expensive to study via all-atomistic (AA) simulations. This has necessitated multi-scale modeling strategies that can access larger spatiotemporal scales, with one such example being the coarse-grained (CG) modeling of polymer systems (Figure 1A).

CG modeling can greatly accelerate computation times, but also yield inaccurate dynamics relative to the atomistic system due to its reduced degrees of freedom (DOF). The inaccurate dynamics are caused by a divergence in the activation energy of glass formation between AA and CG systems (Figure 1B), which is a consequence of the reduction in configurational entropy s_c of the polymer from the reduced DOF. An elegant solution to this problem is the thermomechanically-consistent coarse graining (TCCG) strategy (see lecture 9). In TCCG, we renormalize the cohesive energy term ε in the standard 12-6 Lennard-Jones (LJ) potential of the CG model (Figure 2) so that it retains the relaxation dynamics of the AA model, and thus its activation energy of glass formation. **The objective of this homework is to develop hands-on experience in multi-scale modelling of polymers in the Arrhenius melt regime via the TCCG method.**

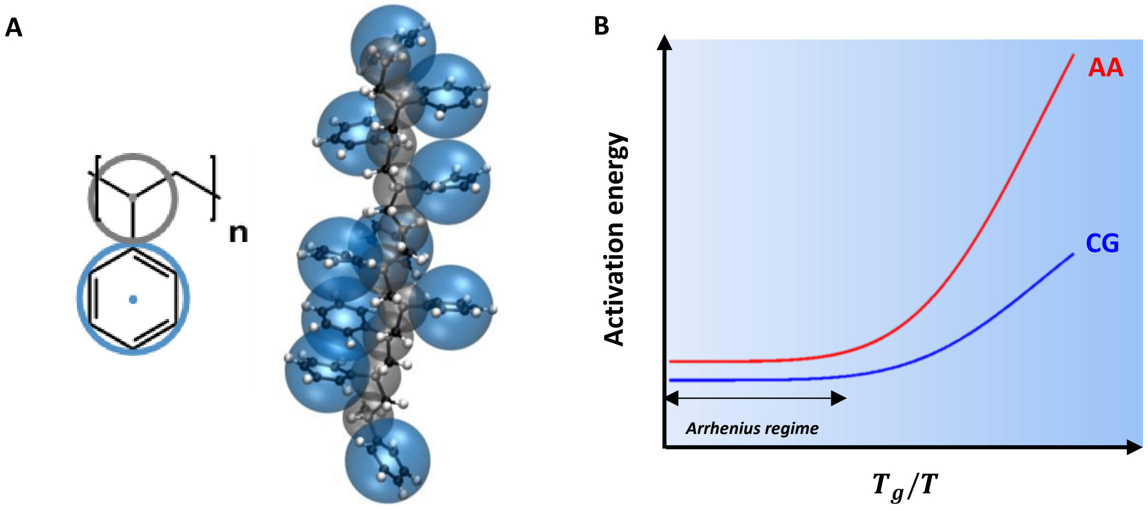

Figure 1. (A) Coarse graining procedure demonstrated on atactic polystyrene (PS). Each monomer is modeled into two super-atom 'beads', where the side-group bead is located at the center of the phenyl ring (blue – bead B) and the main-group bead is designated to be the first carbon attached to the phenyl ring (grey – bead A). **(B)** Schematic Angell plot of the activation energies ΔH_a of

the AA and CG models that diverge upon cooling from the *Arrhenius relaxation regime*. In the Arrhenius regime, the polymer is in a complete melt state and exhibits a constant activation energy. The ΔH_a scale according to the relation: $\Delta H_a(T) = z(T)\Delta H_a^*$ where ΔH_a^* is enthalpy of activation in the Arrhenius regime and $z(T)$ represents the size of cooperatively rearranging region (CRR) in the polymer, which grows upon cooling.[1]

1. Coarse grained modeling using VMD, and identifying ε and σ in the CG model via the radial distribution function (RDF) (40 points)

Typically, the CG modeling procedure involves the use of iterative Boltzmann inversions (IBI) to capture intra-molecular potentials such as bonds, angles and dihedral potentials (lecture 9).[2] Boltzmann inversion strategies can also be employed to the inter-molecular LJ potential parameters ε and σ, through the inversion of the RDF function $g(r)$.

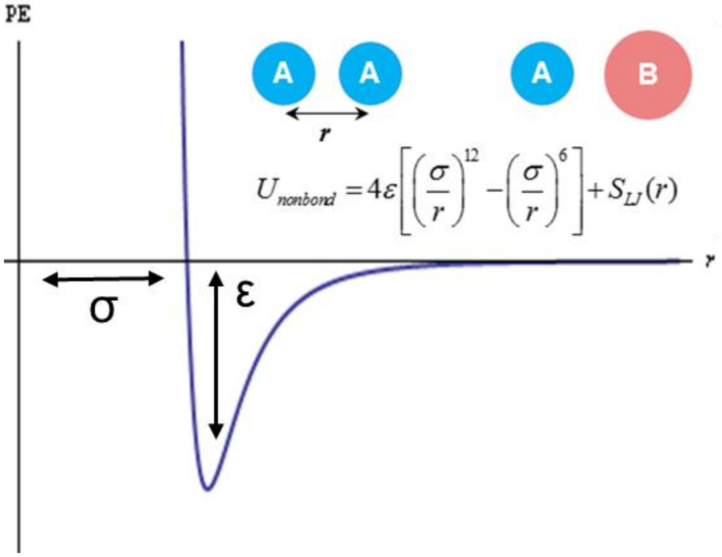

Figure 2. Schematic of a standard LJ 12-6 potential used for the CG non-bonded interactions. The equation in the figure denotes the functional form of the potential, with a GROMACS cutoff ($S_{LJ}(r)$).

In this exercise, we will investigate the $g(r)$ of the AA-PS model at CG force centers and perform a Boltzmann inversion to determine first approximations of ε and σ for the 2-bead CG model shown in Figure 1A. The AA-PS model has previously been equilibrated for 2 nanoseconds at 300 K, and the resulting data file is included in the zip file “Question1-1.zip” on Canvas. Upon unzipping the file, you will have the following files:

- AA_PS_300K.data – this is the equilibrated LAMMPS data file containing information the coordinates of all the atoms in the system. There are 50 chains in the system.
- CG_PS_beadcoordinates.tcl – This is the TCL script which you will be using on VMD to identify bead centers and analyze chain statistics (for instance, bonds, angles, dihedrals and pair correlations).

This exercise will also make extensive use of the VMD software. Some tips:

- The LAMMPS data file can be loaded in VMD by using the TK console (Extensions -> Tk console) with the command ***topo readlammpsdata {dataname}***
- The Label function will be useful in identifying atom indices (Mouse -> Label -> Atoms to select atoms, Graphics -> Labels to identify their indices and coordinates)
- The TCL script can be executed in the Tk console with the command ***source {scriptname}***.
- When visualizing individual polymer chains (Graphics –> Representation -> Selected atoms), use the command **fragment {#} for chain number, and index {#} for atom number.** Note that VMD counts atoms starting from 0 (so for instance, the first chain will be fragment 0). After isolating chains, reset view to focus on the isolated chains (Display -> Reset view).
- For rendering screenshots of VMD, a convenient command to use is **render TachyonInternal {filename}.jpg**

Please load the data file in VMD and answer the following questions (20 points):

1) What is the chain length of the polymer, and how many atoms are there per chain? (Don't forget the hydrogen atoms at the end of the chains)
2) Please provide the VMD indices (i.e. start from 0) for bead A in the first chain of PS.
3) Please provide the VMD indices for bead B (all 6 phenyl ring atoms) in the first chain of PS.
4) Please use the information in 1) to 3) to fill out missing information in the code (FILL) to accurately count the bead centers in the entire system. Note that for bead B, only 2 of the opposing phenyl atoms need to be chosen for the centre of mass. Execute the TCL script for nchains = 1 and make sure the answers are consistent with 2) and 3). Please attach screenshot of Tk console with the index outputs.
5) Now, insert the command **draw sphere ${com1 or com2} radius {choose radius} resolution 5000** into the code to draw spherical objects on the bead centers for each COM calculation. You can precede this command with **draw color {color}** to define different colors for different beads. Choose suitable colors and the values for the radius (see Figure 1A). Execute the TCL script for nchains = 50. Please attach screenshot of the AA polymer with spheres drawn for a. the first polymer chain, and b. a zoomed-out screenshot showing several chains. The locations of the spheres should correspond to the force centers identified in Figure 1A.

The TCL script should have outputted the coordinates of the force centers in com_AA_unwrapped and com_BB_unwrapped files. Next, we will use the coordinates to calculate the RDF of the effective force centers using LAMMPS. Please download "Question1-2.zip" on Canvas. Upon unzipping the file, you will have the following files:

- lammpsdatafile_RDF.m – This MATLAB script takes in the coordinate files and writes a new LAMMPS data file according to the format http://lammps.sandia.gov/doc/2001/data_format.html. Note that this procedure also defines bonds, which is important as the RDF is a pairwise calculation (i.e. not involving directly bonded pairs).
- lammpstrajectory_RDF.m – This MATLAB script creates a LAMMPS trajectory file which can be used to back-calculate the RDF of CG force centers.
- initial.in – LAMMPS simulation input script for calculating the RDF
- submit.sh – Quest submission script
- lmp_openmpi_12Apr13_NOGPU – LAMMPS package (please load this into your **bin** folder, and type **chmod 777 lmp_openmpi_12Apr13_NOGPU** when in the bin folder to obtain full access to the file prior to simulations).

Please run the MATLAB scripts, and place the PSAA_RDF.data, PSAA_traj.lammps, initial.in and submit.sh files in a single folder on Quest and run the simulations (they should finish almost immediately, since we are only calculating the RDF at the last timestep of the previous AA relaxation run). The simulations should output the RDF data for bead A (tmpA.rdf) and bead B (tmpB.rdf) – please see http://lammps.sandia.gov/doc/compute_rdf.html for output format, and http://lammps.sandia.gov/doc/units.html for units.

Once the simulations are complete, please answer the following questions and perform necessary calculations (20 points):

1) Plot (using overlay) the RDFs for bead A and bead B. Please plot the displacement units in Angstroms. Mark the locations of the 1st nearest neighbor distances.
2) Invert the RDFs into a potential using the equation: $V = k_B T ln[g(r)]$. Please use kcal/(mol K) for the units for k_B. Plot V_A and V_B together on a single plot.
3) Identify the values of ε and σ from the potential plots for bead A and bead B, and mark them on the plots. We now have a first estimate of ε and σ from the chain RDF statistics.
4) You will notice that bead B has a much smaller 1st nearest neighbor distance and σ than bead A. Why might this occur from our pairwise RDF calculation of PS? Think about PS's molecular structure and its influence on packing.

The exercise you have just completed can also be done for calculating the probability densities of the bonds, angles, and dihedrals of the force centers of the AA system, which is the foundation of IBI-based multi-scale modeling of polymers.

2. Tuning the CG model to capture Arrhenius dynamics via molecular simulations (60 points)

We will now use the ε and σ values you obtained for bead A and bead B to calculate the self-diffusion dynamics of polystyrene, which is a measure of its dynamics in the Arrhenius melt regime. For this exercise, please download "Question 2-1.zip" on Canvas. Upon unzipping the file, you will have the following files:

- AAdiff.xlsx – diffusion results which have been previously calculated for AA-PS system with N = 10 for T = 400 K to 640 K. These will be the target for calibrating non-bonded parameters.
- in_restart.initial – LAMMPS submission script for self-diffusion simulations. Note that 'kelvin' and 'kelvtwo' are dummy variables which are replaced with temperature values using the BASH script.
- restart.equil – equilibrated LAMMPS restart file (similar to a data file) containing the coordinates and velocity of the CG model relaxed at 750 K for 2 nanoseconds.
- AAB_avg_potential_12.table and AAA_avg_potential_13.table – LAMMPS angular potential files for CG-PS, obtained using IBI.
- submit_bash.sh – BASH script which automates multiple submissions of jobs in Quest.
- submit.sh – Quest submission script.

Before running simulations, it is important to understand some of the key simulation commands that are used. By examining the in_restart.initial file, please answer the following questions (15 points):

1) Briefly describe the statistical ensembles used in this simulation and what they do.
2) Identify the minimization algorithm used in this script and discuss why minimization is necessary for properly equilibrating the system.
3) For how long is the center of mass diffusion calculated for? Please provide the answer in nanoseconds (Hint: consider the timestep).

Please place the in_restart.initial, restart.equil, AAB_avg_potential_12.table and AAA_avg_potential_13, submit_bash.sh, and submit.sh files in a single folder on Quest. In the submit_bash.sh file, please enter in your first estimates for $\varepsilon_{AA},\ \varepsilon_{BB}, \sigma_{AA}\ and\ \sigma_{BB}$. The bash script will pick up these variables and run diffusion calculations for five temperatures: 400, 440, 500, 560, and 620 K. To submit simulations, please use the command **sh submit_bash.sh**. This should output the MSD of the center-of-mass of the chains in the file msd_molecule.data. Please run the simulations (should only take a couple of minutes), perform necessary calculations (see lecture 5), and answer the following questions (30 points):

1) Plot the AA diffusion data and fit it using an Arrhenius equation. Please report the D^0 (m^2/s) and Q (kJ/mol) values.
2) Run the five-temperature simulations for the CG system. The simulation time for an equivalent AA system to calculate the center of mass diffusion was approximately 50,000 CPU seconds. How much faster were the CG simulation? (Hint: see 'loop time' under out.txt).
3) By fitting the MSD results of the five temperature runs of the CG system in the diffusive regime using the Einstein relation for liquids, please calculate the D values for each temperature. Please plot in log10 [MSD(Angstroms^2)] vs log10 [time(fs)] and show an Einstein fit for the diffusive regime for the 400 K run. Please identify the sub-diffusive

and diffusive regimes. Briefly comment on their meaning with respect to Brownian motion and caging of the polymers.

4) Fit the CG data using an Arrhenius equation. Please report the D^0 (m^2/s) and Q (KJ/mol) values. Please overlay the CG data with the AA data into a single plot. You will notice that the CG values have much faster diffusive dynamics than the AA. Why might this be the case? (Hint: consider Figure 1B).

As you have observed, the RDF based non-bonded parameters will yield faster dynamics in the CG model relative to the AA. As such, the cohesive energy parameter ε must be *renormalized*. For this task, we will keep the ratios of ε_{AA} and ε_{BB} intact, and define a scaling parameter α where:

$$\frac{\varepsilon_{AA}}{\varepsilon_{BB}} = \alpha \frac{\varepsilon_{AA}^0}{\varepsilon_{BB}^0}$$

Where ε_{AA}^0 and ε_{BB}^0 will be the original parameters you have used for the previous question. Please conduct the following simulations and answer the following questions (15 points):

1) Please repeat the five temperature simulations with increasing α. Do a few iterations up to $\alpha = 2.5$. Plot the D results, and fit the α-varied simulation results using an Arrhenius equation. Report the D^0 (m^2/s) and Q (KJ/mol) values.
2) Plot Q vs α. Overlay the AA Q results on the plot. Briefly describe the trend observed. What is the *rough* final value of α to match the AA self-diffusion results in the Arrhenius regime? This value should yield *approximately* the same D and Q values. Please also plot the AA and α-tuned CG data in one Arrhenius plot.
3) Please read Ref. 3 and briefly explain why scaling ε affect the Q in the Arrhenius regime in the manner you observed in 2).

Note that α should be independent of temperature in the Arrhenius regime (Figure 1B), but will scale differently with temperature upon the onset of glass formation. In that case, α must be iteratively scaled with temperature to match the segmental dynamics in a procedure called *temperature-dependent energy renormalization.* α also takes a fixed value in the glassy regime, and can be tuned to match glassy dynamics such as the Debye-Waller factor and elasticity.

References

1. Adam, G., & Gibbs, J. H. (1965). On the temperature dependence of cooperative relaxation properties in glass-forming liquids. *The journal of chemical physics*, *43*(1), 139-146.

2. Reith, D., Pütz, M., & Müller-Plathe, F. (2003). Deriving effective mesoscale potentials from atomistic simulations. *Journal of computational chemistry*, *24*(13), 1624-1636.

3. Xu, W. S., Douglas, J. F., & Freed, K. F. (2016). Influence of cohesive energy on relaxation in a model glass-forming polymer melt. *Macromolecules*, *49*(21), 8355-8370.