NE451
Simulation Methods

Hands-on Lab: Monte Carlo Simulation of Polymer Chains

University of Waterloo
Canada

Overview and Learning Outcomes

Problem Overview:

- We simulate a system of particles interacting via two types of potentials:
  - **Lennard-Jones (LJ)**: Non-bonded interactions between neutral atoms/molecules.
  - **FENE**: Bonded interactions modeling particle chains.
- The challenge is to compute the total energy of the system and simulate the dynamics using the Monte Carlo method.

Learning Outcomes:

- Understand the form and role of the LJ and FENE potentials.
- Recognize numerical issues in FENE (logarithm domain problem).
- Learn how Monte Carlo methods simulate particle dynamics.
- Implement and analyze a basic MC algorithm for energy sampling.

Lennard-Jones Potential

The LJ potential is defined as:

$$V_{\mathrm{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right]$$

- ϵ: Depth of the potential well.
- σ: Distance at which potential = 0.
- r: Inter-particle distance.

FENE Potential

The FENE potential is defined as:

$$V_{\text{FENE}}(r) = -\frac{1}{2} K R^2 \ln\left[1 - \left(\frac{r - r_0}{R}\right)^2\right]$$

- K: Spring constant.
- R: Maximum bond length.
- r_0: Equilibrium bond length.
- Issue: Argument of ln can become negative $\rightarrow$ NaN values.

FENE Potential in the Kremer–Grest (KG) Model

The Kremer–Grest (KG) form of the FENE potential is:

$$U_{ij}^{ch} = \begin{cases} -\frac{1}{2} k R_0^2 \ln\left(1 - \left(\frac{r_{ij}}{R_0}\right)^2\right), & r_{ij} \leq R_0 \\ \infty, & r_{ij} > R_0 \end{cases}$$

- Used to model polymer chains as bead–spring systems.
- Ensures bonds do not extend beyond R (finite extensibility).
- No explicit equilibrium bond length r_{eq} is included.
- In practice: chains are initialized with a mean bond length (e.g. $b \approx 0.97\sigma$ in the KG model).

Physical Interpretation of KG FENE

- In the KG model, the equilibrium bond length ($\approx 0.97\sigma$) is not explicitly built into the FENE potential.
- Instead, it emerges from the competition between:
  - FENE attraction (keeps beads connected).
  - Lennard-Jones repulsion (prevents overlap).
- Bonds thus **fluctuate within a finite range**, but cannot break or stretch beyond R.
- This combination reproduces realistic polymer elasticity and prevents chain crossing/backfolding.

Alternative FENE Form with $r_{eq} \neq 0$

A more general form encountered in the literature is:

$$U^{FENE} = \begin{cases} -\frac{1}{2}k(r_{\max} - r_{eq})^2 \ln\left(1 - \left(\frac{r - r_{eq}}{r_{\max} - r_{eq}}\right)^2\right), & 2r_{eq} - r_{\max} < r \\ \infty, & \text{otherwise} \end{cases}$$

- Here, r_{eq} is the equilibrium bond length.
- If $r_{eq} = 0$, this reduces to the KG form.
- Interpretation:
  - With $r_{eq} \neq 0$: bonds oscillate around r_{eq}.
  - With KG ($r_{eq} = 0$): equilibrium bond length is not enforced by the functional form; instead it arises from the interplay of **FENE + LJ repulsion**.

Why Use FENE Potential in This Lab?

- Captures the **connectivity of polymer chains** (bonded interaction).
- Prevents unphysical bond stretching beyond R.
- Naturally produces an effective equilibrium bond length when combined with LJ interactions.
- Widely used in coarse-grained polymer simulations (standard in molecular dynamics of bead–spring chains).
- In our Monte Carlo simulation:
  - LJ handles non-bonded interactions.
  - FENE ensures bonded pairs remain physically connected.

Monte Carlo Simulation (Hands-on)

In this lab, you will implement the simulation in a Jupyter Notebook.

- Generate trial moves by perturbing particle positions.
- Compute energy change ΔE from LJ + FENE potentials.
- Accept moves if $\Delta E < 0$, or with probability:

$$P_{\text{acc}} = \exp(-\beta \Delta E), \quad \beta = \frac{1}{k_B T}$$

- Repeat for many steps to generate a trajectory.

You will complete the missing parts of the code in the notebook to run this simulation.

Monte Carlo Algorithm (Lab Exercise)

Algorithm outline — to be coded by you in the Jupyter Notebook:

1. Initialize particle configuration.
2. For each step:
   - Propose a trial move.
   - Compute total energy (LJ + FENE).
   - Accept/reject move using the Boltzmann criterion.
3. Repeat for many steps to explore configuration space.
4. Record the trajectory of particle positions for analysis.

Students will implement this algorithm in code during the lab.