GSoC 2022 Project Proposal

Implementing REMD methods in Molly.jl

Submitted by: Jaydev Singh Rao

Email: jaydev19@iiserb.ac.in or jaydevsrao@gmail.com
Website: https://jaydevsr.github.io
Github: JaydevSR, Slack: Same name.

Introduction

Molecular dynamics (MD) is a class of simulation methods that are used to study motion of interacting particles (atom or molecules) with time by solving their equations of motion. Molly.jl is a package in native julia that implements many such molecular dynamics algorithms for study of interacting molecular systems like gases, biomolecules etc. This proposal is about the implementation of Replica Exchange Molecular Dynamics (REMD) method, which is one very important technique of MD simulation, in the Molly.jl package as it lacks this feature at present.

What is REMD?

REMD method is a MD simulation method which involves running several MD simulations of a system at different values of some parameter (eg. temperature, energy etc.) in parallel and exchanging the states from these parallel simulations at periodic intervals (see fig. 1). This leads to the sampling of states which would not be accessible by a single run for a single parameter value. Due to this, REMD is very attractive and in significant use in research due to its ability to escape local minima and explore the configuration space of complex biomolecules to a greater extent compared to classical molecular dynamics simulations.

The implementation of these methods into Molly.jl will provide a very good addition to its overall capabilities. By the end on the proposed project it is expected that temperature based REMD method and some variants of this method will be implemented along with suitable tests and documentation.

The Proposal

The concrete plans that are proposed for this are as follows:

1. Implementation of temperature-REMD method which will run multiple replicas of a MD simulation at **different temperature** values on multiple threads in parallel and exchange

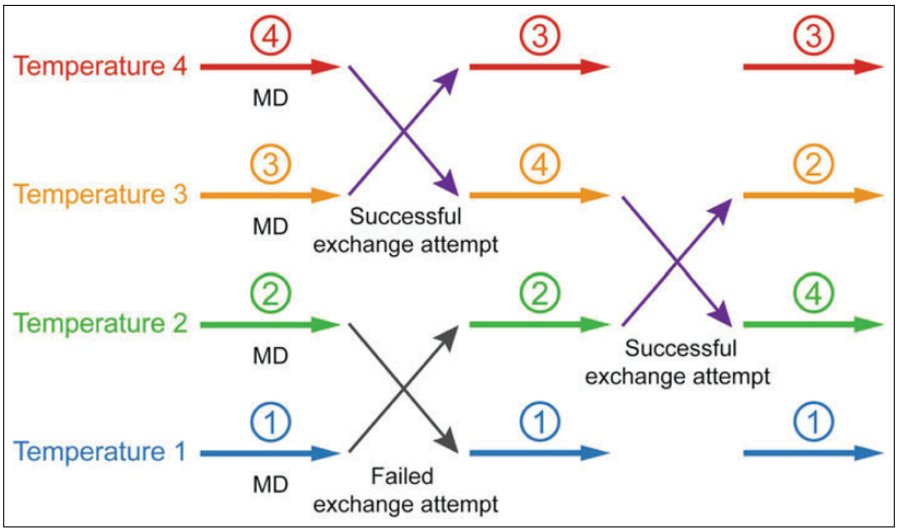

Figure 1: A visual desctription of temperature-REMD simulation.

these at periodic intervals based on Metropolis-Hastings ratios which will give the probability by which such exchange should happen.

Metropolis-Hastings Ratios. *The probability of accepting an exchange between any two states is given by:*

$$P(accept) = \begin{cases} 1 & \Delta < 0 \\ \exp(-\Delta) & \Delta \geq 0 \end{cases}$$

Here, Δ is a quantity that is proportional to energy difference of two states and the difference of inverse temperature.

2. After the implementation is complete, the next step will be to test the implementation on some model systems for which we can get good results from classical MD itself. For example, simulation of Argon using Leonard-Jones potential or simulation of folding of some small protein.

3. One of the integral tasks other that the algorithm implementation itself would be writing sufficient unit tests that cover the complete use cases of temperature-REMD and documenting all the changes and additions made during the complete process.

4. After the successfully completion of the above tasks, a natural next step would be extending the implementation to other REMD methods. Possible variants that can be implemented:

   - Hamiltonian-REMD is the method which is most similar to temperature-REMD and will be easiest to tackle next. It involves running MD simulations of same system at **different energies** instead of temperatures.
   - Replica Exchange Umbrella Sampling (REUS) is an another method which involves adding a harmonic potential term (called umbrella potential) to the original potential energy of the system centered at some different location for each replica. The umbrella potential term prioritizes location where it is centered at, during the MD simulation.

All these methods provide different advantages and can be used in different situations depending upon the biomolecules and the type of process which is being simulated.

Possible hurdles

- The MD simulations in Molly.jl can already utilize multiple threads for better performance. One possible hurdle would be effective use of available threads for MD simulations and simulating multiple replicas which does not sabotage performance as well as accuracy of results.
- Another difficulty can arrive due to the stochastic nature of MD simulations. Due to this the tests should be written in such a way that these are independent of this randomness in results and are not misleading.

Deliverables

Given below is an initial version of the timeline that is to be followed:

- **Phase 1: Before Mid-term Evaluation**
  1. (Before June 13) Understand Molly.jl Codebase.
  2. Implement temperature-REMD.
  3. Test the implementation on model systems (eg. small protein folding, Argon gas).
  4. Add documentation and provide some unit test coverage.
- **Phase 2: After Mid-term Evaluation**
  1. Changes based on feedback from evaluation.
  2. Implementing other variations of REMD.
  3. Test any new implementations on model systems.
  4. Documentation of new changes and maximizing unit test coverage.

What after GSoC?

The experience that I will gain from working on Molly.jl will pave way for future contributions to this package which I am very much interested in and will be grateful for, if I get a chance. I would also like to continue working on anything that is proposed but due to any reason is left uncompleted during the course of the contribution period.

Code Portfolio

There are two projects that I have worked on in the past and am particularly proud of. One is implementing Monte Carlo algorithms i.e. Metropolis-Hastings and Wolff algorithms for studying XY and Ising model (link to project). The other is implementing Numerov-Cooley method, which is a method for solving the Schrödinger's Equation by integrating it and estimating energy by a variational principle (link to project). This is because both of these projects involved learning methods from literature and implementing those from scratch in Julia. This was a really fun and enriching experience for me which is also in line with the requirements of this proposal.

Other than that although I don't have experience contributing to any open source project, as a way of contribution, I have raised issues in the past whenever I felt that those will be constructive to the particular project. Such examples include this and this, both of which have led to PRs in Julia and PlutoUI.jl.

About Me

I am a third year undergraduate student from India. I study Electrical Engineering and Computer Science at IISER Bhopal. I am highly passionate about the field of computational physics in general as it satisfies my intellectual needs from love of both physics and computer science. I have been using Julia since the release of v1.0 and learning it bit by bit it has become **the** programming languge for me. Other than posts on discourse/slack and some forum comments, I have mostly been an external observer in the Julia community. I am motivated towards working on this project because it will give me a chance to take active part in the community that is built around the programming language we all love.

Logistics

During the months of May to July which is summer vacation for me, I have no other commitments except one research project that I have been working on in the past. After that I will have have my semester courses in the remaining period of GSoC (August and September) which I can manage along with this project.