Abstract:

This thesis presents a Python script that performs Stepwise Monte Carlo simulations according to the methods described by Frenkel & Smit. Specifically, this code is used to determine adsorption isotherms of monatomic gases onto a carbon scaffold with the grand canonical ensemble. In defense of these results, various experiments were run to study the behavior of the simulation and verify several assumptions regarding reproduction and equilibrium convergence. We describe in detail the algorithms used in the script as well as how to appropriately use and edit the script for other experiments.

* Fundamentally an interdisciplinary topic, so special care is made to properly cover basics of the thermodynamics of adsorption and Monte Carlo algorithms
* Present a Python monte Carlo simulation script
* Specifically tuned to determine phase behaviors of Lennard-Jones fluids with the grand canonical (VT) ensemble.
* Using the method determined from Metropolis et al.
* Using Monte carlo trial moves as described in Frenkel & Smit
* Presenting a bunch of Isotherms for carbon scaffolds
* Extendible to other scaffold types
* Proof of comparison with other EOS without scaffold
* Detailed Description of the Simulation Software developed
* Studied repeatability to determine local minima
* Verifying assumptions related to the simplification of system.
* Code is straightforward mathematical approach to simulation for study and adaptation
* Thorough explanation of code and algorithms
* As the “experiments” are fundamentally different from laboratory work, the process of applying software will be explained more carefully.
* Audiences of this paper
  + Chemical Engineers versed in thermodynamics but little to no experience in programming
  + Programmers who are knowledgeable in numerical algorithms but not necessarily in adsorption
  + Math folks with knowledge of stochastic methodologies

INTRODUCTION

* Explain that each run is an experiment and the ramifications of that
* Useful for experimenting with systems that involve significant monetary or time investment
* Extremely Useful as a first look of a system to determine viability
* Explain the difference between Monte Carlo and Molecular Dynamics
* Heavy use of stochastic elements to eventually converge on an average behavior
* We’re using Python because its easy to read and performant “enough” for our needs
* Repeated random sampling.
* Nondeterministic processes, problems to complex to sove analytically, high demenisonality
* Importance sampling
* Challenges related to monte carlo simulations
* Broad description of trial moves.
* Why many cycles are required and the core bottleneck of performance
* Ergodic?
* No Time element involved and the ramifications of that
* Common misconceptions
* Avoid Local minima of energy.
* Force tensors not trivial to compute for molecular dynamics
* Brief history of Molecular Monte Carlo as well as standout materials.
* Brief Description of the organization of the paper and the overall goals accomplished.
* INTRODUCTION
  + WHY USE SIMULATIONS
    - Before developing adsorption zeolites, its helpful to estimate their performance.
    - Its so much cheaper to develop software to determine the adsorption properties
    - Monte carlo lets our simulation run based on probability and avoiding dynamics. Cheaper in terms of computation.
  + WHAT IS THE GOAL
    - We want to approximate the physical properties of a fluid using molecular monte carlo simulation.
    - Once we prove the simulation works for bulk fluids as expected, apply scaffold of carbons.
  + WHAT IS THE RESULT?
    - We successfully replicated thermal properties of bulk fluid and have estimated isotherms for various carbon scaffolds
  + WHAT IS THE TL;DR METHOD?
    - Grand canonical
    - Lennard Jones
    - Monte Carlo

Define Simulation Properties:

**Description of our location grid.**

How the locations of particles are tracked:

Changing the dimensions of arrays is computationally intensive and should be avoided at all possible

To combat this, we set a for our core particles, in this case particles.

We have a separate vector describing particle locations, X, Y, Z, such that each index refers to a specific particle.

We also set describing the number of particles in our unit cell.

Reminder that indexing starts at 0, so the last particle is located at index

Define System Properties:

Start Equilibration Run:

Start Production Run:

Publish Results:

How Particles are handled

Constantly changing the size of arrays is bad, so we shouldn’t do that

We don’t need to do complex operators with the matrix as a whole, we only really need one particle at a time.

In fact, we don’t need an arbitrary number of particles, we can have a maximum amount and ignore the rest of the data structure.

So the structure works like so:

We keep track of the x, y, and z coordinates of each particle in the unit cell and split that information into an array of x positions, y positions, and z positions. By knowing the index of where we should stop looking, namely n particles, we only access what we need.

To give you a better idea of what we’re doing, heres the basic operations:

For each particle:

For each element up to the Nth

Move a particle:

Change the ith element of X, Y, and Z to the appropriate values

Add a particle:

Set the ’th element of X, Y, and Z to the appropriate value, then set

Remove a particle:

Set the ith element of X, Y, and Z to equal the N’th element of X, Y, and Z, then set N\_h = N\_h -1.

Brief Description of each algorithm

move

add

remove

dist\_hi( x, y, z, i ): # Calculate the squared distance between a proposed particle at location [ x, y, z ] and the particle applying unit square periodicity

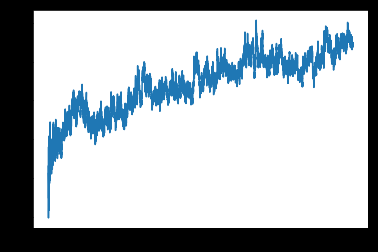
:

dist\_ci( x, y, z, i ): # Calculate the squared distance between a proposed particle at location [ x, y, z ] and the particle applying unit square periodicity

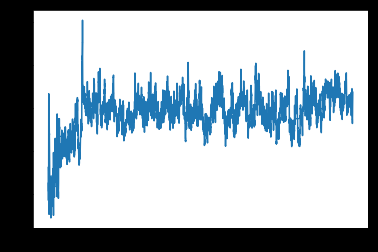
:

dist\_h # Calculate each squared distance of a proposed particle at location and each particle in the system.

:



With 10 equil cycles, 50000 prod cycles, 100 moves per cycle



* MOLECULAR SIMULATION WITH MONTE CARLO
  + WHAT ARE MONTE CARLO SIMULATIONS?
    - Simulations that involve randomness in some way.
    - By repeating a series of possible inputs, various probabilities of outcomes are possible.
    - Upon enough iterations, the outputs converge theoretically to the average outcome and with higher confidence.
  + METROPOLIS ALGORITHM
    - By a specific person and with a few assumptions, we bias our “reasonable inputs” with by potential energy.
    - Tenets
      * At “simulation equilibrium” a trial move and its reverse should be equally probable.
      * If a move reduces energy, should be accepted
      * If a move increases energy, should be accepted with some probability to avoid getting stuck in local minima
  + FORCE FIELD LENNARD JONES
    - WHAT IS IT
      * A method to approximate potential energy within our system
      * A combination of repulsive and attractive forces between two spherical particles
    - HOW ARE WE APPLYING IT?
      * Unit Cell
        + Periodicity
        + Estimation of density
      * Sum of all energy between all particles
    - SPECIAL OPTIONS
      * Truncated, correction, shift
      * Reduced Units
      * Delta Adjustment
  + GRAND CANONICAL ENSEMBLE
    - STATISTICAL MECHANICS
      * Definition of Grand Canonical constant mu V T
      * Definition of equilibrium (mu = mu, T=T,
    - TRIAL MOVE TYPES AND DERIVATION
      * Ideal gas of reservoir
      * Move
      * Exchange
    - OTHER PROPERTIES
      * Computing Total Energy
      * Pressure from Virial
  + PROGRAMMING AND USAGE OF SIMULATION
    - DESCRIPTION OF ALGORITHMS shorthand
      * Manipulating particles
    - RESULTS OF SYSTEM AS IS
    - RESULTS OF SYSTEM WITH ADDITIONAL CARBON ATOMS
* APPENDIX
  + DESCRIPTION OF ALGORITHMS
  + CHECKING ASSUMPTIONS
    - CHECKING WITH EOS FOR CARBON LESS
    - TRUNCATION vs NON-TRUNCATED
    - FLOATING POINT ERRORS
    - ENOUGH SIMULATION STEPS?

What is a monte carlo simulation?

Source to a place more equipped to talk about it than here

Monte carlos are used to solve complex integral problems, repeatedly evaluating a point in random space and averaging over the results. The simplest form (naïve) monte carlo experiment is treating every point in random space with equal probability.

It is a technique used to apply randomness to a proposed system and generate a representative range of outputs for studying. This stochastic nature makes it excellent for computational methods.

With respect to molecular modeling, this approach is not to be confused with “molecular dynamics” as that attempts to model motion and behavior of particles, rather than their location.

Molecules are instead modeled by probabilistic locations, relying on equilibrium statistical mechanics to appropriately model a system. It is tempting to look at some of the plots and results and assume there is a temporal aspect, there is not.

These typically require a lot of computer power, these are primarily computational approaches and thus many of the problems in this field are about computers rather than theory?

We are generating step-wise states according to Boltzmann probabilities. This is how Metroplis monte carlo simulations are run

Can devise special sampling methods

Provies thermodynamic and structural properties

Random walk algorithm,

Monte Carlo as applied to Molecular Modeling

Uses Markov chain procedure through a series of steps. A new state is accepted at random, aka a “trial move’.

What is a Metroplolis Monte Carlo Simulation.

Mention metropolis et al.

Metropolis is our method of biasing our sampling to “reasonable” states.

**Grand canonical ensemble Monte Carlo for a Lennard-Jones fluid**

[D.J. Adams](https://www.tandfonline.com/author/Adams%2C+DJ)

Pages 307-311 | Received 09 Jul 1974, Published online: 22 Aug 2006

**Chemical potential of hard-sphere fluids by Monte Carlo methods**

[D.J. Adams](https://www.tandfonline.com/author/Adams%2C+DJ)

Pages 1241-1252 | Received 07 Feb 1974, Published online: 22 Aug 2006

### Simulating prescribed particle densities in the grand canonical ensemble using iterative algorithms

J. Chem. Phys. **128**, 124102 (2008); <https://doi.org/10.1063/1.2839302>

[Attila Malasics](https://aip.scitation.org/author/Malasics%2C+Attila)1*,*[Dirk Gillespie](https://aip.scitation.org/author/Gillespie%2C+Dirk)2*, and*[Dezső Boda](https://aip.scitation.org/author/Boda%2C+Dezs%C5%91)1,2, a)

**Grand Canonical Monte Carlo Simulation Study of Water Adsorption in Silicalite at 300 K**

[**Joël Puibasset**](https://pubs.acs.org/author/Puibasset%2C+Jo%C3%ABl)[**\***](https://pubs.acs.org/doi/10.1021/jp7097153#cor1)[**†**](https://pubs.acs.org/doi/10.1021/jp7097153#afn1)**and**[**Roland J.-M. Pellenq**](https://pubs.acs.org/author/Pellenq%2C+Roland+J-M)[**‡**](https://pubs.acs.org/doi/10.1021/jp7097153#afn2)