Project work in Computational Physics

October 8, 2024

Below you will find some ideas for the project work in the course. You are welcome to choose some other problem of your own liking, after discussion with the teacher.

It is important that the problem is solved individually, but some collaboration with other students is accepted, and discussions are encouraged. You are welcome to consult the instructors of the course and the literature if help is needed.

The report should be written individually. The report should be typed on a computer and sectioned as a professional report, containing title, author, abstract, section headers, etc. Use spell checks. Computer produced plots should be included with figure numbering and captions. Give a brief background to the model and methods used but try to keep the report less than around 10-15 pages.

You are supposed to write and run your own computer program. You are free to choose programming language, a compiled one like C++, C, or FORTRAN is usually recommended for serious computational simulations, but Java or Python will most likely be sufficiently efficient for most tasks. Include a short listing of relevant parts of the computer code in the report.

Test your programs! Do your results look reasonable? Often there may be limiting cases where one knows the answer. Or one can test the program on small enough systems that an exact calculation is possible. Whenever feasible, include error bars of you results, computed, e.g., using binning or from independent simulations.

The basic objective to reach to pass the project exercise is to get your program working correctly and do some basic data analysis of the results. Some problems are easier, and some are harder. It is possible to get a high grade also from the easier ones, but you should aim to include either some more nontrivial simulation technique or advanced data analysis. You do not have to follow the instructions exactly, and it is allowed to skip or add steps if you want. For help and more information on the problem, please ask the instructors.

The most basic projects are marked with *, the slightly more nontrivial ones with **, and the more challenging ones with ***. See the following table for the grade criteria:

Grading criteria

Grade	Requirement	Criteria
A	** or ***	Working simulation including some nontrivial algorithm and/or data analysis. The report should clearly present the project objective, methods, and results in an excellent way and demonstrate your knowledge of the subject.
B	**	Working simulation including some more nontrivial algorithms and/or data analysis, presented very clearly in a very well written report that demonstrates your knowledge of the subject.
C	*	Working simulation and adequate data analysis, presented clearly in a well written report.
D	*	Basic working simulation and data analysis. Complete and well written report. Minor errors and omission tolerable.
E	*	Considerable progress towards a working simulation, and a complete report. Some errors, misunderstandings, or omissions tolerable.

1 Write a Monte Carlo program to simulate a Lennard-Jones fluid (e.g., argon) *

In the computer labs you have used molecular dynamics simulations to simulate argon. Write a program that implements Monte Carlo simulations instead. Use periodic boundary conditions and make the simulations at fixed N, V and T. Calculate energy and pressure. Depending on your ambition you can make use of cutoffs in the interaction and Verlet neighbor lists, or you can include interactions with all particles. Use the similar parameters, densities, and temperatures as you used in the MD lab on argon. Check that you get similar energy and pressure as in the MD-simulations. Don't forget the kinetic part of energy and pressure. Experiment with different ways of sampling obeying detailed balance (different maximum step lengths in the moves and moving one or several particles at the same time) and monitor how this affects the acceptance ratio. Start from a system equilibrated at wrong temperature and see how the choice of the moves affect convergence. Is there some correlation between acceptance ratio and rate of convergence? (See also the following project ideas for further things to study.)

2 Monte Carlo simulation of Lennard-Jones fluid in the NPT ensemble**

This project expands on the previous one. In addition, implement an NPT ensemble by including Monte Carlo moves that change the volume. See e.g., the book by Frenkel and Smit for how to do that.

3 Hybrid (Hamiltonian) Monte Carlo simulation for a Lennard-Jones fluid**

Implement HMC updates in a Lennard-Jones particle simulation and compare its efficiency with ordinary Metropolis Monte Carlo moves. HMC stands for Hybrid Monte Carlo or Hamiltonian Monte Carlo and uses an ordinary velocity Verlet integrator to propose MCMC moves, which are then accepted or rejected based on a Metropolis criterion. (See e.g., Frenkel-Smit). Make sure you can reproduce the results of the Argon lab and, in addition, compute the autocorrelation time to quantitatively compare with Metropolis MC.

4 Grand canonical Monte Carlo simulation of Lennard-Jones fluid**

Write a Monte Carlo simulation code for the grand canonical μVT-ensemble, where the particle number N may fluctuate. (See Frenkel & Smit, Chap. 5.6.) Find an interesting range of parameters to study, e.g., covering the melting transition or boiling transition or around the critical point and plot various quantities as function of temperature or chemical potential μ. The average number of particles is $\langle N \rangle$, and its variance is related to the compressibility κ.

5 Free energy alchemical calculation of chemical potential in a dense Lennard-Jones fluid***

When simulating a crowded system of particles, the probability of successfully inserting an additional particle becomes very low. In such cases it is better to gradually insert a particle by smoothly turning on its interactions with the other ones. The Hamiltonian for the whole system of particles is defined as

$$H = \sum_{i=0}^{N} \frac{\mathbf{p}_i^2}{2m} + \sum_{0<i<j\leq N} V(\mathbf{r}_i - \mathbf{r}_j) + \lambda \sum_{i=1}^{N} V(\mathbf{r_0} - \mathbf{r}_i),$$

where $\lambda \in [0,1]$ is a parameter controlling the strength of interaction between the particle at r_0 with the rest. Here $V(\mathbf{r})$ is the usual Lennard-Jones interaction. When $\lambda = 0$ the extra particle does not interact at all, while at $\lambda = 1$ it behaves identically to the other ones. The corresponding free energy difference $\Delta F = \mathrm{F}(\lambda = 1) - \mathrm{F}(\lambda = 0) = \mu - \mathrm{k_B T} \ln\left((\mathrm{N}+1)(\Lambda/L)^{\mathrm{d}}\right)$, where $\mu = \mathrm{F_{N+1}} - \mathrm{F_N}$ is the chemical potential, $\Lambda = h/\sqrt{2\pi m k_B T}$ is the thermal wavelength, and $d = 2\ or\ 3$ is the dimensionality. Use for example thermodynamic integration to estimate the free energy difference and thereby the chemical potential from simulations in a reasonably dense system. You can also use other techniques for estimating the free energy difference, such as free energy perturbation, Bennet's acceptance ratio (BAR), or WHAM.

When doing these calculations, it turns out to be better to replace the last term in the Hamiltonian $\lambda \sum_{i=1}^{N} V(\mathbf{r_0} - \mathbf{r}_i)$ by a softened version

$$\lambda \sum_{i=1}^{N} 4\epsilon(A^{-2} - A^{-1}), \qquad A = 0.5(1-\lambda) + (r/\sigma)^6 .$$

Note that this will change your expression for $dF/d\lambda$, but it will not change the end states.

6 MD / MC simulation of small Lennard-Jones clusters.***

Simulate a system of Lennard-Jones particles in an NVT-ensemble at relatively low density and temperature. You may choose to use MD or MC simulations. Consider, e.g., 10 particles in a volume $V = 10^3$. At low temperatures the particles will cluster together forming a small droplet. Study the melting of this cluster using parallel tempering or Wang-Landau simulations and compute (up to an additive constant) the free energy as function of temperature for a range covering the disintegration of the cluster.

7 Monte Carlo simulation of an Ising model: Comparison of different Metropolis updates*

Write your own implementation of a single-spin flip Monte Carlo simulation of the Ising model as in the Ising lab. Calculate the autocorrelation function of the magnetization and energy and determine the correlation times. Compare a single spin flip Metropolis Monte Carlo simulation using random updating, where each spin to flip is chosen randomly, sequential updating, where the spins are gone through in order, palindromic, where they are gone through in order and then in the reverse order.

8 Monte Carlo simulation and finite size scaling of the three-dimensional classical XY-model*

Repeat the same tasks as in the Ising lab for the three-dimensional classical XY-model. The energy is defined by

$$\mathrm{H} = -\mathrm{J}\sum_{\langle ij\rangle} cos\left(\theta_i - \theta_j\right)$$

where $\theta_i \in [0,2\pi)$ is phase angle at lattice point i, and the sum runs over all nearest neighbors on a lattice with $L^3 = N$ lattice points. This model has several different physical realizations: A planar magnet where each classical spin $\mathbf{S_i} = (cos\theta_i, sin\theta_i)$ is confined to the XY-plane, or a superconductor or superfluid which has a complex order parameter $\psi_i = e^{i\theta_i}$.

The (absolute) magnetization is given by $m = \left|\frac{1}{N}\sum_j \boldsymbol{S}_\mathbf{j}\right| = \left|\frac{1}{N}\sum_j e^{\mathrm{i}\theta_\mathrm{j}}\right|$, and the Binder ratio is as usual

$$B = 1 - \frac{\langle m^4\rangle}{3\langle m^2\rangle^2}$$

Study the Binder ratio and the other thermodynamic quantities. Comment on the limits of the Binder ratio for low and high T. Determine the critical temperature and critical exponents. Study the equilibrium time near T_c.

(Hint: Do not bother to define a finite step length, $\Delta\vartheta$, for the maximum update of the phase angle. Instead, simply draw the updated phase angle uniformly in $[0,2\pi)$. There is no gain in the amount of uncorrelated data per CPU time in using a finite step length, even if the acceptance rate can be made larger. On the other hand, restricting ϑ to taking only, say, 1000 discrete values in $[0,2\pi]$, and using look-up tables for everything, may speed up the code.)

See also the following problems for variations.

9 Monte Carlo simulation and finite size scaling analysis of the 3D XY model using the Wolff algorithm**

Like project 8 but implement a Wolff algorithm instead of the Metropolis single spin flip.

10 Monte Carlo simulation of the 3D XY model using the worm algorithm**

Like project 8 but using the Worm algorithm instead. This begins with a reformulation of the partition function for the XY model as a system of directed closed loops,

$$\mathrm{Z} = \sum_{\{n_{ij} \in Z\}} exp\left(-\frac{K}{2}\sum_{\langle ij\rangle} n_{ij}^2\right)\prod_i \delta_{\sum_{k\in N_i} n_{ik},0}$$

The degrees of freedom here are the link variables n_{ij} which are integers and can be thought of as currents. The Kronecker-δ enforces a constraint of current conservation at each vertex i, since $\sum_{k\in N_i} n_{ik}$ denotes the sum of currents going out from i.

The Hamiltonian $H = \frac{K}{2}\sum_{\langle ij\rangle} n_{ij}^2$ is then basically the kinetic energy of these currents.

(The mapping from the XY model to this link-current model in this case contains an approximation, but the universality class of the transition will still be preserved.)

The "magnetization" is not directly accessible in this formulation, so instead study the so-called winding numbers, W_x, W_y, W_z

$$W_x = \sum_{\substack{\text{All links ij in}\\ x-\text{direction}}} n_{ij} \ , etc.$$

The fluctuations of this quantity $\langle W^2 \rangle$ can be related to the superfluid density and will become nonzero in the superfluid/superconducting phase, as the coupling constant K changes. Carry out a finite size scaling analysis of

$$Y(K,L) = \frac{1}{L^2}\langle W_x^2 \rangle = y([K - K_c]L^{1\nu}),$$

where y is a scaling function.

11 Monte Carlo simulation and finite size scaling analysis of the three-dimensional classical Heisenberg model*

In this project you should carry out similar tasks as in project 8, but for the 3D Heisenberg model, i.e., with the XY-spins replaced by 3-component classical Heisenberg spins. The Hamiltonian is then given by

$$H = -J \sum_{\langle ij \rangle} \mathbf{S}_i \cdot \mathbf{S}_j \,, \qquad \mathbf{S_i} = \left(S_i^x, S_i^y, S_i^z\right), \qquad \mathbf{S}_i^2 = 1 \,,$$

where as usual $\langle ij \rangle$ denote nearest neighbour pairs on a simple cubic lattice. Note that if you use spherical polar angles (ϑ,φ) to specify the direction the integration volume element is sinθ dθ dφ, as usual in spherical coordinates. The simulation can use either Metropolis single spin updates or Wolff cluster updates or both.

12 Wang-Landau simulation of an Ising model**

Implement the Wang-Landau method to estimate the density of states $\Omega(E)$ of the Ising model. Using the density of states, you should be able to compute canonical averages over a large range of temperatures.

13 Wang-Landau simulation of a Q-state Potts model**

As in project 12, but for a Q-state Potts model, where the energy is

$$H = -J \sum_{\langle ij \rangle} \delta_{q_i,q_j}, \qquad q_i = 1,2,\dots,Q.$$

14 Parallel tempering simulation of an Ising model***

Implement a Monte Carlo program that uses parallel tempering to simulate a range of temperatures including the critical one in two or three dimensions. You may also combine this project with free energy calculations, or use histogram reweighting.

15 Calculation of free energy in an Ising, XY, or Potts, model**

Compute the free energy as function of temperature (up to an additive constant) for statistical mechanics model, e.g., an Ising model, a XY model, or a Potts model.
You may use e.g., thermodynamic integration, free energy perturbation, Bennet's acceptance ratio, the weighted histogram method (WHAM), or from nonequilibrium simulations.

16 Swendsen-Wang simulation of Ising, XY, or Potts model***

Another cluster method is the so-called Swendsen-Wang method. Here one first divides the lattice into clusters by assigning a bond $n_{ij} = 0$ or 1 to each link of the lattice. Then one identifies all connected clusters of the system, where the clusters are defined by those lattice sites connected by bond $n_{ij} = 1$. This can be done using the Hoshen-Kopelman algorithm. Then all degrees of freedom of each cluster is assigned a new random value independently.

17 Event driven simulation using the Gillespie algorithm**

When simulating a Markov chain or process on discrete states there is a possibility of accelerating the process in case the rejection rate is high: One simply samples the number of time steps to the next event that leaves the current state and then increments time with this amount in one go. To do this one need to enumerate all possible MC moves from the current state, which is a rather costly operation, so it only pays off if the rejection rate is high. There are a few different variants of this approach, the Gillespie algorithm being one of the most sophisticated. Implement this algorithm for some model, e.g., an Ising model, or for a system of chemical reactions.

18 Molecular dynamics of a hard sphere fluid**

Write the program. This must be written differently than a program using continuous forces and requires some fairly elaborate programming. In the hard sphere fluid, the force is zero all the time except when two spheres collide. Then the force is infinite. Therefore, you should start by making a table of the time for the next collision of each particle and

the number of the particle it will collide with. Then you just assume that all particles will move along straight lines (taking the periodic boundaries into account) with their given (random) initial velocities. Then you pick the shortest time in this table. Now, you subtract this time from all other times in the table. If it was particle *i* and *j* that collided, you have to calculate their new velocities from the rules of elastic collisions. It might be useful to consult a textbook in classical mechanics for this problem. Further you have to calculate the next time they will collide with all other particles and update the table accordingly. Finally, you may iterate this procedure.

Measure the density as the degree of close packing $\eta = 4\pi NR^3/3V$. You may use dimensionless units ($R = 1$). The velocities should be random with an average zero. You can, however, use dimensionless velocities. Chose them for instance in such a way that you will get the mean square velocity equal to 1. Use the program to calculate the pair correlation function of the hard sphere fluid at a couple of densities. You should get a pair correlation function that is zero for r less than 2 and has a couple of oscillations with a period in r that is of the order 2. Reasonable degrees of close packing should be in the interval 0.1 up to maybe 0.5. (References: Allen and Tildesley, p. 102-108 or B.J. Berne, Statistical Mechanics part B p 1-40)

19 Temperature control in MD**

Program the Nosé-Hoover thermostat and compare it to the simple Berendsen velocity scaling with a time constant used in the argon computer lab. The equations of motion can be written on the following form for both thermostats:

$$\frac{d}{dt}\bar{p}_i = -\nabla_i U - \xi \bar{p}_i \quad i = 1, ...N$$

The difference is in the equation of motion for the heat bath degree of freedom (ξ) which in the Berendsen scaling method can be written:

$$\xi = \frac{1}{2\tau}[\frac{1}{3Nmk_BT}\sum_i \bar{p}_i^2 - 1]$$

while for Nosé-Hoover it is:

$$\frac{d}{dt}\xi = \omega^2[\frac{1}{3Nmk_BT}\sum_i \bar{p}_i^2 - 1]$$

Both methods contain a parameter τ or ω. Check for both methods, how the values of these parameters affect temperature stability, fluctuations, and the responses of the system to small sudden changes in the temperature of the heat bath. Check also how well the velocities obey a Maxwellian distribution in the two cases. Calculate the heat capacity from the canonical ensemble equation based on energy fluctuations for both types of

simulation. Compare the results with each other, with the heat capacity obtained from temperature fluctuations in a micro canonical (NVE) simulation and with experimental heat capacities. You may use the MD-program from, the argon computer lab as a template and make the changes there. (Reference, Frenkel & Smit p. 125-141.) You may also implement and compare with the stochastic velocity rescaling of Bussi et al., 2007, (Canonical sampling through velocity rescaling. *The Journal of Chemical Physics*, *126*(1), 014101. https://doi.org/10.1063/1.2408420).

20 Effects of frustration in the two-dimensional Ising model on a triangular lattice**

The two-dimensional Ising model can also be defined on a triangular lattice, and then each spin has six nearest neighbors. Study it in zero applied field. This may easily be done by simply adding a coupling along one of the diagonals of each elementary square in a Monte Carlo code for square lattices.

Compute the Binder ratio and determine T_c from it. Explain why it is bigger or smaller than the value for the square lattice. Do a finite size scaling analysis of g to determine ν. Is it smaller or bigger than for a square lattice? Try to explain!

Finally, change the sign of the coupling constant J to $J = -1$, which corresponds to an antiferromagnet. On a square lattice this can be transformed back to the ferromagnetic case $J = +1$ by flipping every other spin in a checker-board pattern. But for $J = -1$ a new phenomenon enters called frustration: no matter how the spins around an elementary triangular plaquette orient themselves, one of the three interactions will be in the excited state. Perform the same simulations as for the ferromagnetic case, and a finite-size scaling analysis, if possible, if not explain why.

21 Wang-Landau simulation of two-dimensional Ising model on a triangular lattice***

Extend the previous project by making a Wang-Landau simulation to calculate the density of states and from that the entropy at low temperatures. To get the entropy you may use that $S(T \to \infty)$ can be easily computed.

22 Self-avoiding random walks*

An interesting and extensively studied problem in polymer physics is to model a polymer chain in a good solvent as a self-avoiding random walk, where each step taken by the walker represents one segment of the polymer (one monomer). Crudely speaking, the

distinction between "good" and "poor" solvent is that in a good solvent the interactions between the polymer and the solvent molecules is more important than the long-range interactions between polymer molecules, and the latter can be ignored. You will study the average end-to-end distance in various spatial dimensions by Monte Carlo simulation of random walks on simple cubic lattices.

(a) Free random walk. The ordinary "free" random walk can be simulated by simple sampling of random steps on a simple "cubic" lattice in 1,2,3 and 4 dimensions). Start at the origin. Take one step to any nearest neighbor lattice site at random, and from there take another step, etc. Calculate the exponent $p = 2\nu$ in $\langle R_N^2 \rangle \sim N^p$ in the limit of a large number of steps N, where R_N is the end-to-end distance of the chain after N steps. It is useful to average the results over many walks, so the program should do at least 10^4 independent walks in one run. Derive the exact result $p = 1$ and compare with the simulation results.

(b) Simple sampling of SAW. Modify your program to simulate a model of a polymer as a self-avoiding random walk on the lattice. Self-avoiding means that the walk cannot cross its previous path. This can be sampled as follows: Do an ordinary walk, as in (a). Do not use moves that bring the walker back to position occupied at the previous step, so in d dimensions use an equal probability to move in $2d - 1$ directions. If the walker reaches a site where it has previously been, the walk terminates. It is thus necessary to keep track of all the previous locations of the walker. This can be done by storing a vector with visited coordinates (storing the whole lattice and tagging visited sites requires unnecessary memory usage). Calculate the exponent $p = 2\nu$ in 2, 3, and 4 dimensions. How long chains can be reached with reasonable convergence? Discuss the values of p and compare with the results from (a) and discuss why p goes up. Compare your result with the Flory mean field theory result $\nu = 3/(d + 2)$, which should be remarkably accurate in $d = 2,3$ dimensions, and exact in $d = 4$ ($d = 4$ is the so called upper critical dimension for a self-avoiding random walk, i.e. the dimension above which the self-avoiding interaction becomes irrelevant). What do you think would happen with p for self-attracting walks? Compute the absolute dimensionless entropy per monomer from the equation: $S/k_B(N - 2) = \ln\Omega(N)/(N - 2)$. For the random walk $\Omega(N) = (2d)^{N-2}$ which gives a constant entropy per monomer. For the SAW $\Omega(N) = W(N)(2d - 1)^{N-2}$ were $W(N)$ is the fraction of successful attempts. This gives an entropy per monomer that decreases with N. Determine the functional form of this decrease for large N.

(c) SAW by biased sampling. There is an easy way to improve the poor acceptance rate at large N of the simple sampling method, by using biased sampling. For the trial move at "time step" N, keep track of the number of available, empty neighbour

sites. Let $a(N)$ be the number of available moves at time step N. Do one of the available moves at random, and compute $R^2(N)\Pi_{j=1}^{N} a(j)$. Repeat for many walks i = 1,...,N_w. The desired average is an average over walks: $\langle R^2(N)\rangle = \sum_i R_i^2(N)\Pi_{j=1}^{N} a_i(j) / \sum_i \Pi_{j=1}^{N} a_i(j)$. Calculate ν and study the statistics of the method and compare with (b).

23 Optimization of the traveling salesman problem using simulated annealing**

The traveling salesman problem is a classic hard optimization problem. The salesman wants to minimize the distance traveled while visiting each of N cities. Assume that the N cities are positioned randomly in a square of unit side-length, and measure distances by the usual Euclidian distance. The function to be minimized is the total length,

$$H = \sum_{n=1}^{N} \|\mathbf{r}_i - \mathbf{r}_{i+1}\|,$$

where $\mathbf{r_i}$ are the city locations, where city $i+1$ is visited after city i, and $\mathbf{r}_{N+1} = \mathbf{r}_1$. Introduce Monte Carlo moves which propose a deformation of the path, e.g., by switching the order in which two randomly chosen cities are visited and accept or reject by a usual Metropolis algorithm at a "temperature" T. Start the simulation at high T and slowly lower it towards zero. If done sufficiently slowly the final state will be the ground state, i.e., the optimal solution with high probability.

Use this to find (nearly) optimal solutions to the traveling salesman problem. Repeat the annealing process several times for a given layout of the cities and record the minimal values of H found. For small N you can compare with exact solutions by simply comparing the $N!$ different paths, but when N grows this is no longer feasible.

For larger N, study how the width of the distribution of final lengths varies with cooling rate and number of cities N. Investigate when the method works to find optimal or nearly optimal solutions. Try this with few different realizations of the random city locations. You can also experiment with different trial moves in the algorithm.

24 Traveling salesman problem using parallel tempering***

Do the same problem as the previous using parallel tempering instead of simulated annealing.

25 Lee-Kosterlitz method to study discontinuous phase transitions***

The q-state Potts model undergoes a discontinuous phase transition at some critical temperature T_c, if $q > 4$, where q is the number of states per site.
The Hamiltonian of the Potts model is defined by

$$H = -J \sum_{\langle ij \rangle} \delta_{s_i, s_j}$$

where $\delta_{a,b}$ is the Kronecker delta, and $\{s_i \in 1, \dots, q\}$ are describe the discrete states of the system. The sum runs over nearest neighbours in a 2D lattice.
Choose a value $q = 8$ for which the transition should be strongly discontinuous.
Implement a Wang-Landau simulation for the model and compute the density of states.
From the density of states, it is possible to compute the canonical (Boltzmann) probability as function of energy E, $P(E) = \Omega(E) \exp(-\beta E)/Z$. When the temperature is close to T_c this probability distribution $P(E)$ will have two peaks corresponding to the different phases, ordered and disordered, of the model.
Use the method introduced by Lee and Kosterlitz to study the transition and verify its discontinuous character.
See *New numerical method to study phase transitions,* J. Lee and J. M. Kosterlitz, Phys. Rev. Lett. **65**, 137 (1990) https://doi.org/10.1103/PhysRevLett.65.137
and
Kosterlitz, J.M., Lee, J., Granato, E. (1993). *A New Numerical Method to Study Phase Transitions.* In: Landau, D.P., Mon, K.K., Schüttler, HB. (eds) Computer Simulation Studies in Condensed-Matter Physics IV. Springer Proceedings in Physics, vol 72. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-84878-0_4