KFU Graz

Dénes Sexty (denes.sexty@uni-graz.at)

**CP1: Monte Carlo Methods** WS 2023/24

# Monte Carlo projects

Working out one of the projects below is a requirement for completing the CP1 course.

You should prepare a short document (10-20 pages) that describes the model and the problem, describes the observables, shows the results using various plots and some discussion of the results. Key parts of the code (preferably with comments that make it understandable) should be included in an appendix. This document will be the basis of a short discussion in the exam, where some topics of the lecture will also be discussed.

1. **Potts model**

   Simulate the $q$-state Potts model in $d = 2$ dimensions on a quadratic lattice with the energy

   $$H[s] = -J \sum_{neighb.ij} \delta(s_i, s_j) - h \sum_i \delta(s_i, 1) \tag{1}$$

   and the spins are chosen from the set $s_i \in \{1, 2, \dots q\}$.

   Observables of interest are the magnetization $\sum_i \delta(s_i, 1)/L^2$, energy density or ("action density") $H[s]/L^2$ and their susceptibilities. Use several different $\beta$ values (from the Boltzman factor $e^{-\beta H}$), and find a phase transition for $q = 2$ and $q = 8$ at $h = 0$. Can you determine the order of the phase transitions?

2. **Percolation**

   Implement the Hoshen-Kopelman algorithm for counting clusters for a 2d percolation configuration on a square lattice, using site percolation (as discussed on the lecture).

   $n_s$ denotes the number of $s$ size clusters per lattice site, $P_\infty$ is the probability of belonging to the infinite cluster (or the largest cluster on a finite lattice. The average size of finite clusters is:

$$S = \frac{\sum_s s^2 n_s}{\sum_s s n_s} \tag{2}$$

Use finite size scaling to find $p_c$. Measure the two point function $G(r)$ which is the probability that two occupied points with distance $r$ belong to the same cluster. This depends on $r$ exponentially: $G(r) \sim \exp(-r/\xi)$ Extract the correlation length $\xi$ in a couple points around $p_c$.

Around the critical point one should have $S \sim |p - p_c|^{-\gamma}$. What is $\gamma$?

3. **Travelling Salesmen Problem (TSP):** Throw $N$ random dots (the cities) on a unit square and take their Euclidean distance as entries of the distance matrix $d_{ij}$. The goal is to find a route for a round trip represented by a permutation $p(i)$ of $1, \ldots, N$ such that the total distance travelled

$$D = \sum_{i=1}^{N} d_{p(i),p(i+1)} \tag{3}$$

is minimal (we define $p(N+1) = p(1)$ to make a round trip).

Use a Metropolis algorithm with simulated annealing to find a route with minimum annealing. Do $M$ simulated annealing sweeps, where you do $N_s$ sweeps before changeing the inverse temperature with a step $\Delta\beta$. What should be $\beta_{\text{min}}$ and $\beta_{\text{max}}$? Play around with the parameters. How many sweeps are need to find a "good" minimum? How does that depend on $N$? Calculate the average minimal path length for $N = 20$ and $N = 40$ with error estimation (the average is over the position of the cities).

4. Use a **genetic algorithm** to find a minimum for the Travelling salesman Problem described above.

Start with $N_{pool}$ random permutations and delete a certain percentage of them, which are the least fit (their route is the longest). Then fill up the gene-pool by mutations (swapping two or more elements in the permutation, reverse some part of the route, etc.) and crossovers, where you take two "parents" to give an offspring.

You can take e.g. the **Ordered crossover** where you take some consequetive part of parent 1 and let that be a part of the child permutation.

The rest you fill up in the order you take from parent 2. For example: we take $p(3-6)$ from the first parent and fill in the rest from parent 2:

$$\begin{aligned} \text{parent}_1 &: \mathit{52873416} \\ \text{parent}_2 &: 17642583 \\ \text{child} &: 16\mathit{8734}25 \end{aligned}$$

Another crossover: **Edge recombination.** Take two parents and build neighbor lists using edges which are in the two parents.

$$\begin{aligned} \text{parent}_1 &: \mathit{523416} \\ \text{parent}_2 &: 164253 \end{aligned}$$

Neighbour lists:

| | |
|---|---|
| 1 : | 3, 4, 6 |
| 2 : | 3, 4, 5 |
| 3 : | 1, 2, 4, 5 |
| 4 : | 1, 2, 3, 6 |
| 5 : | 2, 3, 6 |
| 6 : | 1, 4, 5 |

Now use the following algorithm:

1 take $X$: first number from a random parent

2 append $X$ to the child. If the child is full, we are done.

3 delete $X$ from the neighbor lists

4 if $X$'s neighbor list is empty, set $Z$ to a random new number not already used
else: set $Z$ to a neighbor of $X$ which has the least number of neighbors itself. (choose randomly if there is a tie)

5 $X = Z$, goto step 2.

e.g. we could get here:
randomly chose 2nd parent: child = 1

now after deleting 1 from the lists, we look at 3,4,6 (neighbors of 1) and 6 has only 2 neighbors: child = 16
after deleting 6, 4 and 5 have both 2 neighbors, randomly choose : child = 164
after deleting 4, 2 and 3 have both 2 neighbors, rnd. : child = 1643
after deleting 3, 2 and 5 have both 1 neighbor, rnd. : child = 16432
and finally: child = 164325.

Try $N_{gen}$ successive generations to look for a minimum, try to find the optimal parameters for a fast search (probabilities and type for mutation, crossover, pool size, etc.) Calculate the average minimal path length for $N = 20$ and $N = 40$ with error estimation (the average is over the position of the cities).

5. **Random walks**

   Implement the reptation algorithm or the pivot algorithm or the biased sampling method for self avoiding random walks. Measure the average distance of the beginning and the end as a function of the number of steps. Verify your algorithm by comparing results with the naive algorithm (generating no-backtrack random walks and throwing them away if there is an intersection).

   The distance of the endpoint to the beginning depends on the number of steps like $R \sim N^\nu$. Find $\nu$ in $d = 2$ or $d = 3$.

   How does $R_{max}$ (the maximal distance during the walk) depend on $N$?

6. **Anomalous diffusion**

   Fill a quadratic lattice with a percolation configuration, i.e. randomly fill each site on the lattice with 0 or 1 such that each site has a 1 with probability $p$, independently of others.

   Now study a random walk on the lattice such that the walker is only allowed to step on the clusters, which are the sites which have 1.

   Measure the average displacement as a function of time as a function of $p$:

   $$\langle R^2 \rangle_p \sim t^x \tag{4}$$

   Extract the exponent $x$ as a function of $p$.

7. **XY model in d=3**

Simulate the XY model in d=3. XY model has variables which are continous: they are angle variables: $\phi_x \in [0, 2\pi]$ The action is given by:

$$S = -\beta \sum_x \sum_{\nu=0}^{2} \cos(\phi_x - \phi_{x+\hat{\nu}}) - h \sum_x \cos(\phi_x) \tag{5}$$

Write an importance sampling simulation using local Metropolis updates. Experiment with the proposal step a bit to achieve an acceptance rate of 0.5 or so. As a function of $\beta$ there is a phase transition, detected by e.g. the action density $\langle S \rangle / \Omega$ or the magnetization $M = (1/\Omega) \sum_x \cos(\phi_x)$ or their susceptibilities with $\Omega = L^3$ the volume of the lattice.

Measure the critical $\beta$ at $h = 0,\ 0.1$, and $0.2$. (Use finite size scaling)

8. **XY model in d=3** Same as above, but use the Langevin equation to simulate. Measure the average action at $\beta = 0.3, h = 0$ at different Langevin steps to find out how small timestep you need to be close to the $\Delta\tau = 0$ limit. Use this timestep for other parameters too.

9. Simulate the $\lambda\phi^4$ Theory in 2 spatial dimensions, defined by the action

$$S = \beta \left( \sum_x \frac{1}{2} \sum_\nu (\Phi_x - \Phi_{x+\nu})^2 + h\Phi_x + \frac{m^2}{2}\Phi_x^2 + \frac{\lambda}{24}\Phi_x^4 \right) \tag{6}$$

with real variables $\Phi_x \in \mathbb{R}$. Use $m^2 = -1$ and $\lambda = 6$. Observables of interest are for example the action density, the magnetization $M = (1/L^2) \sum_x \Phi_x$ and their susceptiblities.

Write an importance sampling simulation using local Metropolis updates. Experiment with the proposal step a bit to achieve an acceptance rate of 0.5 or so.

As a function of $\beta$ there is a phase transition, detected by e.g. the action density $\langle S \rangle / \Omega$ or the magnetization $M = (1/\Omega) \sum_x (\phi_x)$ (or their susceptibilities) with $\Omega = L^3$ the volume of the lattice.

Measure the critical $\beta$ at $h = 0,\ 0.1$, and $0.2$. (Use finite size scaling)

10. Simulate the $\lambda\phi^4$ Theory.

Same as above, but use the Langevin equation. Measure the average action at $\beta = 1, h = 0$ at different Langevin steps to find out how small timestep you need to be close to the $\Delta\tau = 0$ limit. Use this timestep for other parameters too.

11. **Ising model** with Demon algorithm or worm algorithm. (more details if we manage to cover these algorithms on the lecture)