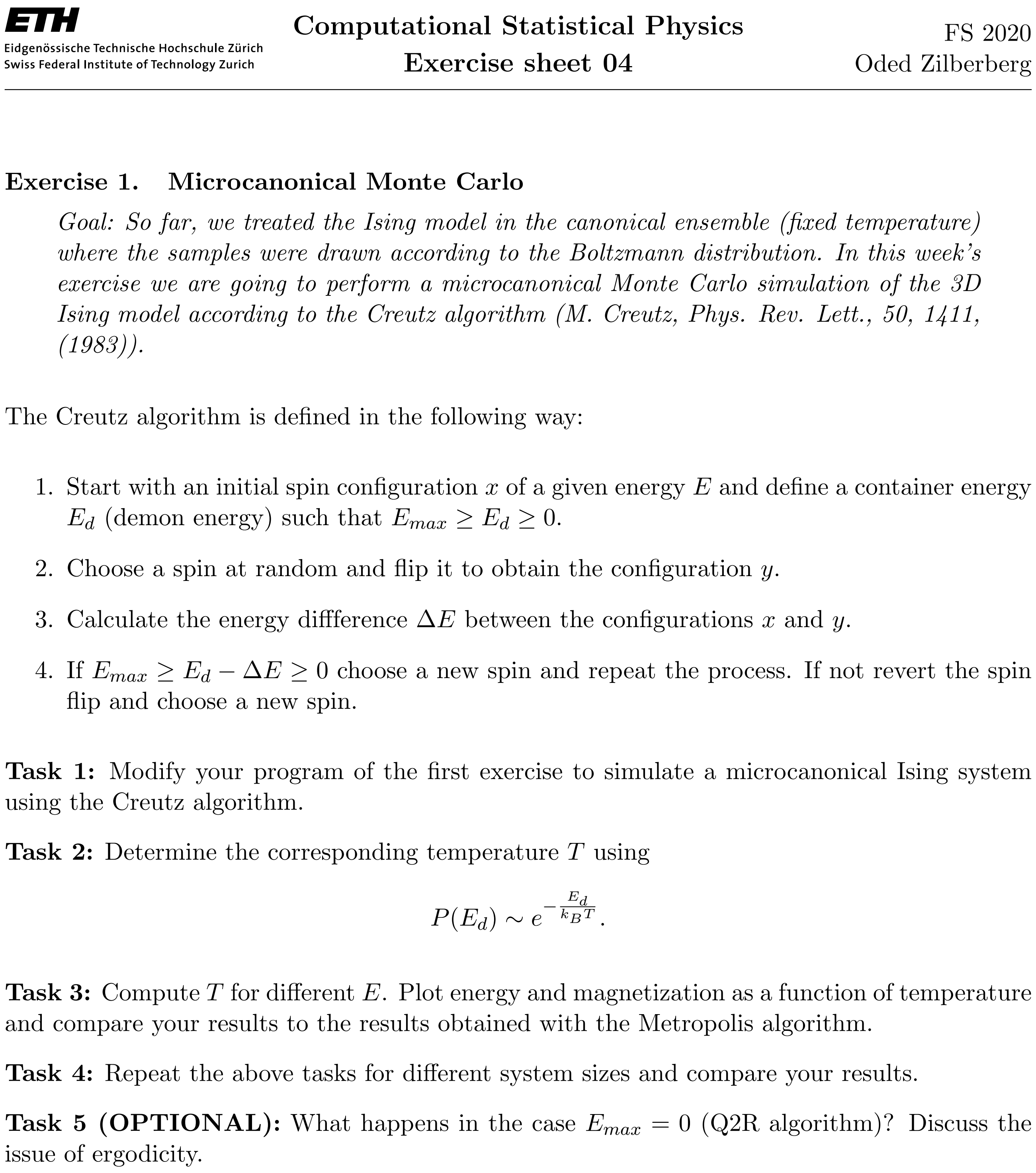
Exercise class 5

Maximilian Holst
Toni Heugel

Solution exercise sheet 4



ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computational Statistical Physics
Exercise sheet 04

FS 2020
Oded Zilberberg

Exercise 1. Microcanonical Monte Carlo

Goal: So far, we treated the Ising model in the canonical ensemble (fixed temperature) where the samples were drawn according to the Boltzmann distribution. In this week's exercise we are going to perform a microcanonical Monte Carlo simulation of the 3D Ising model according to the Creutz algorithm (M. Creutz, Phys. Rev. Lett., 50, 1411, (1983)).

The Creutz algorithm is defined in the following way:

1. Start with an initial spin configuration x of a given energy E and define a container energy E_d (demon energy) such that $E_{max} \geq E_d \geq 0$.
2. Choose a spin at random and flip it to obtain the configuration y.
3. Calculate the energy diffference ΔE between the configurations x and y.
4. If $E_{max} \geq E_d - \Delta E \geq 0$ choose a new spin and repeat the process. If not revert the spin flip and choose a new spin.

Task 1: Modify your program of the first exercise to simulate a microcanonical Ising system using the Creutz algorithm.

Task 2: Determine the corresponding temperature T using

$$P(E_d) \sim e^{-\frac{E_d}{k_B T}}.$$

Task 3: Compute T for different E. Plot energy and magnetization as a function of temperature and compare your results to the results obtained with the Metropolis algorithm.

Task 4: Repeat the above tasks for different system sizes and compare your results.

Task 5 (OPTIONAL): What happens in the case $E_{max} = 0$ (Q2R algorithm)? Discuss the issue of ergodicity.

Implementation

We want:

- Simulate and sample system at energy E
- Energy fluctuation $0 \leq E_D \leq E_{max}$

Structure:

1. Initialize a configuration with all spins in the state +1
2. Increase the energy until E is reached
    - Randomly flip spins that increase the energy
3. Sampling using the Creutz algorithm
    - Start with $E_D = 0$
    - Accept spin flips with $0 \leq E_D - \Delta E \leq E_{max}$
    - Update $E_D + = E_D$

Implementation

Structure:

1. Initialize a configuration with all spins in the state +1
2. Increase the energy until E is reached
   - Randomly flip spins that increase the energy
3. Sampling using the Creutz algorithm

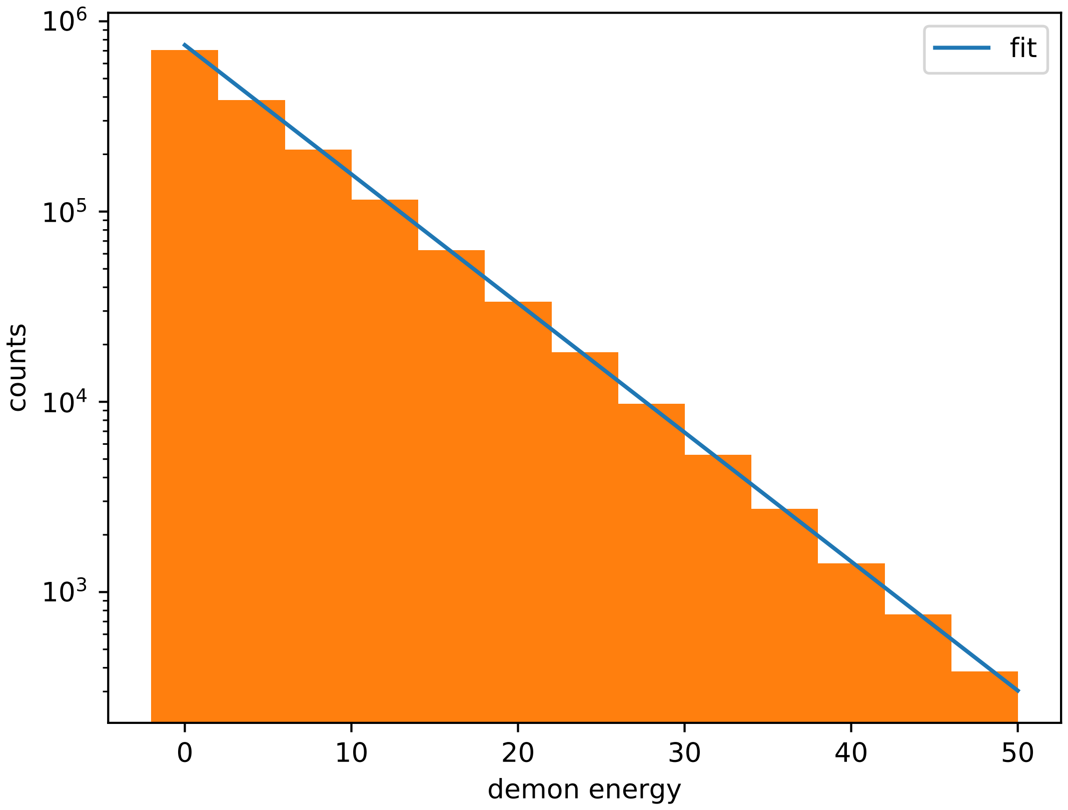
→ Repeat at different energy E

Analysis:

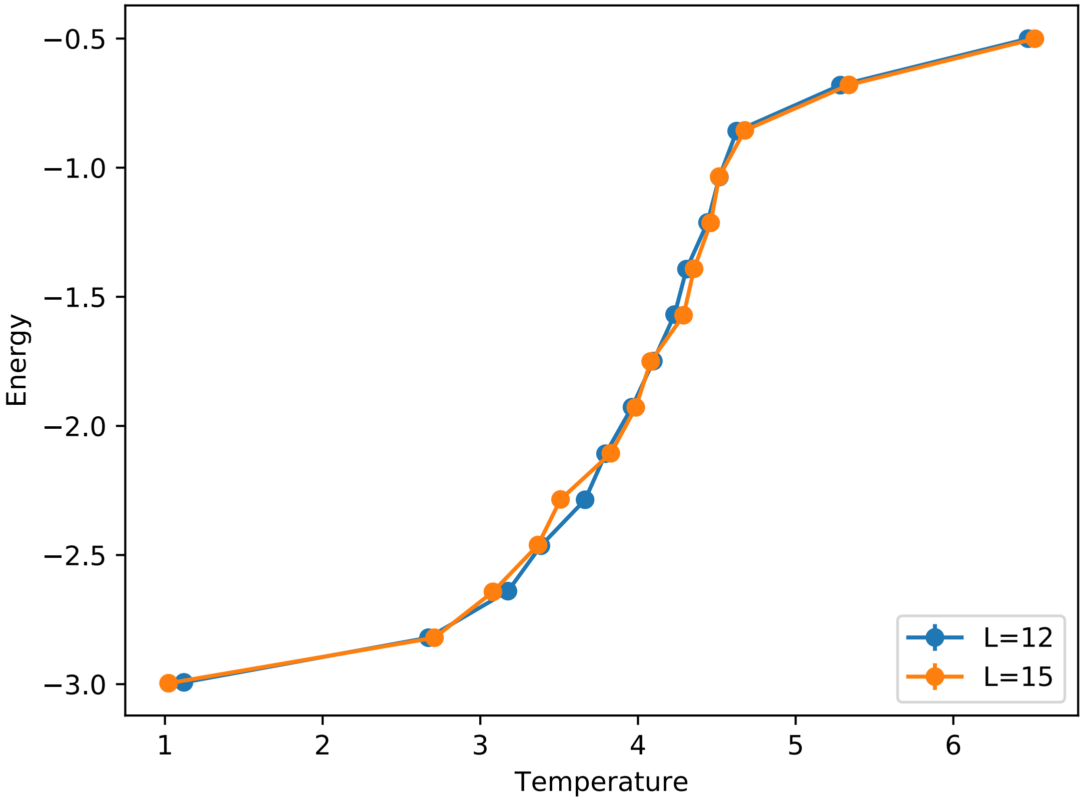
- Find temperature corresponding to energy E
  - $P(E_D) \propto \exp(-E_D/k_B T)$
- Plot $E(T)$ and $M(T)$

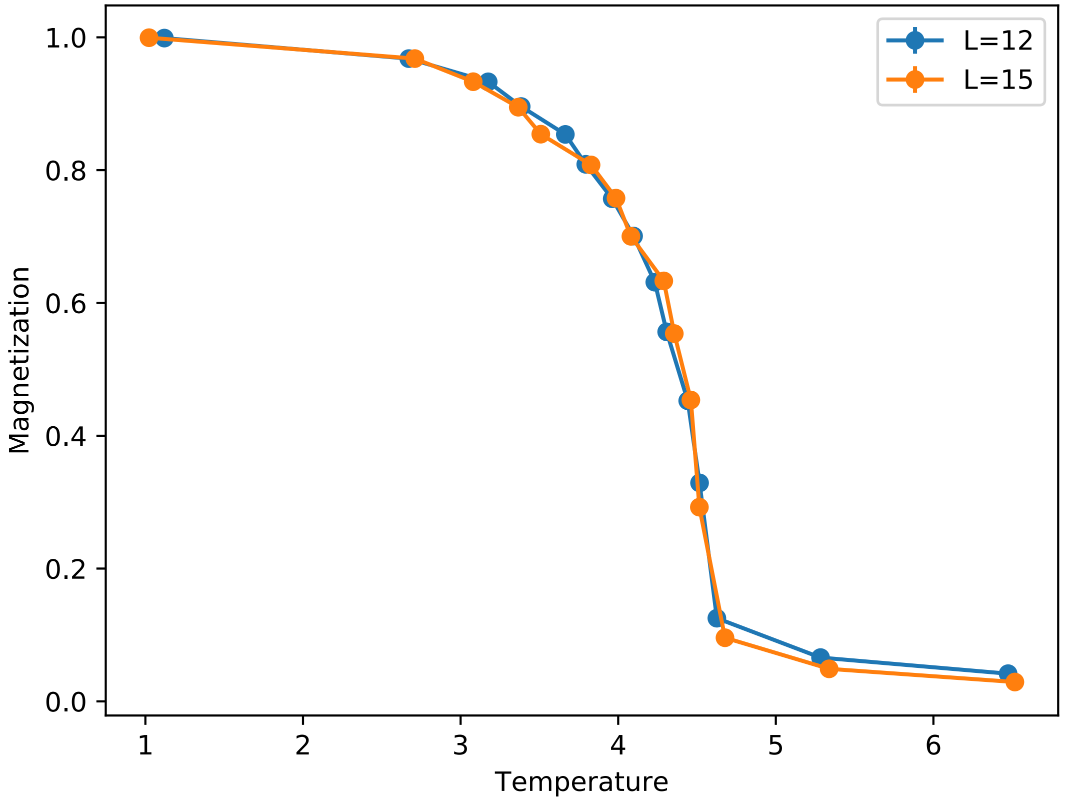
Code

Results



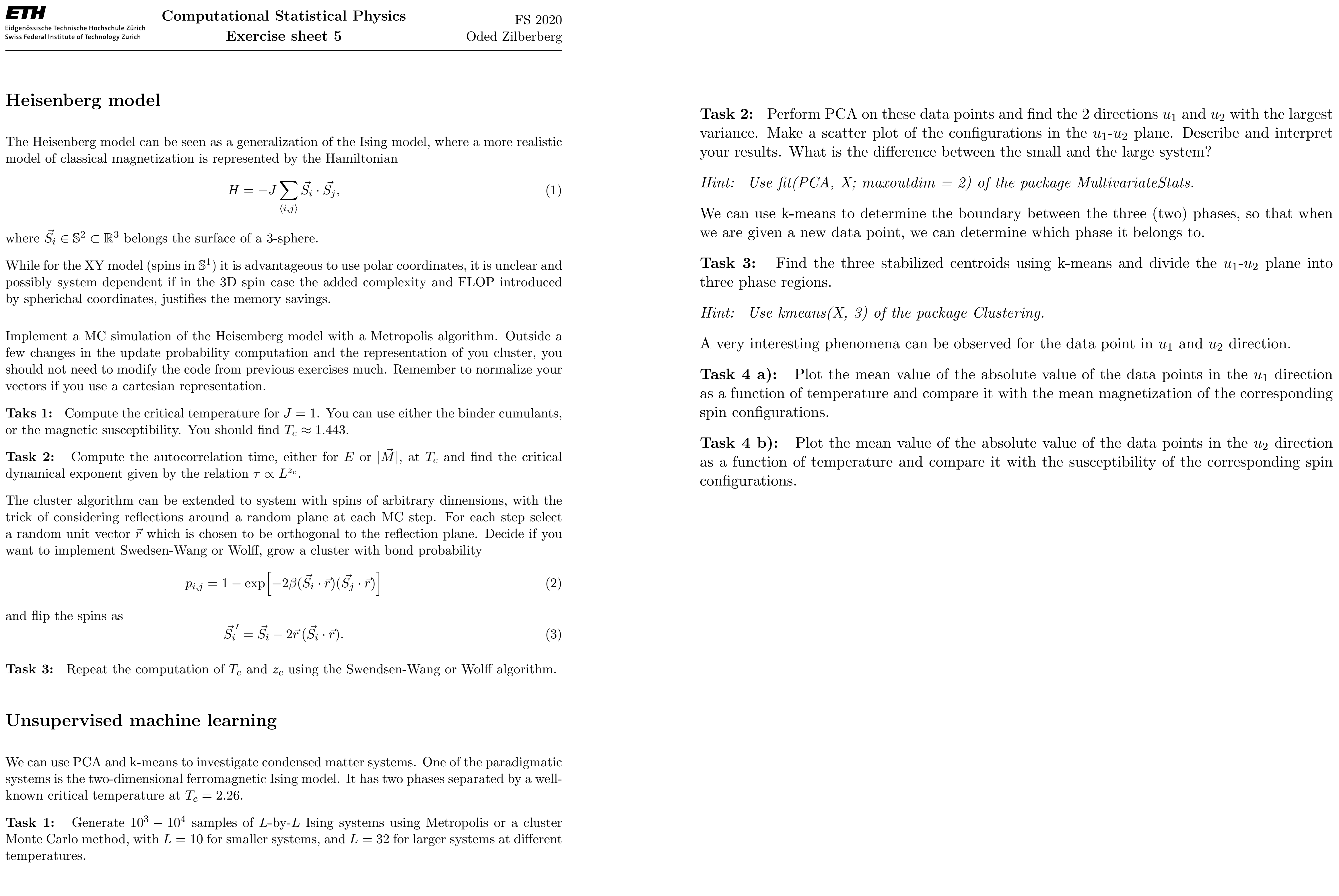
Results


−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
Energy
1
2
3
4
5
6
Temperature
L=12
L=15


1.0
0.8
0.6
0.4
0.2
0.0
Magnetization
1
2
3
4
5
6
Temperature
L=12
L=15

Exercise sheet 5

Exercise sheet 5



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computational Statistical Physics
Exercise sheet 5

FS 2020
Oded Zilberberg

Heisenberg model

The Heisenberg model can be seen as a generalization of the Ising model, where a more realistic model of classical magnetization is represented by the Hamiltonian

$$H = -J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j, \tag{1}$$

where $\vec{S}_i \in \mathbb{S}^2 \subset \mathbb{R}^3$ belongs the surface of a 3-sphere.

While for the XY model (spins in $\mathbb{S}^1$) it is advantageous to use polar coordinates, it is unclear and possibly system dependent if in the 3D spin case the added complexity and FLOP introduced by spherichal coordinates, justifies the memory savings.

Implement a MC simulation of the Heisemberg model with a Metropolis algorithm. Outside a few changes in the update probability computation and the representation of you cluster, you should not need to modify the code from previous exercises much. Remember to normalize your vectors if you use a cartesian representation.

Taks 1: Compute the critical temperature for $J = 1$. You can use either the binder cumulants, or the magnetic susceptibility. You should find $T_c \approx 1.443$.

Task 2: Compute the autocorrelation time, either for E or $|\vec{M}|$, at T_c and find the critical dynamical exponent given by the relation $\tau \propto L^{z_c}$.

The cluster algorithm can be extended to system with spins of arbitrary dimensions, with the trick of considering reflections around a random plane at each MC step. For each step select a random unit vector $\vec{r}$ which is chosen to be orthogonal to the reflection plane. Decide if you want to implement Swedsen-Wang or Wolff, grow a cluster with bond probability

$$p_{i,j} = 1 - \exp\Big[-2\beta(\vec{S}_i \cdot \vec{r})(\vec{S}_j \cdot \vec{r})\Big] \tag{2}$$

and flip the spins as

$$\vec{S}_i{}' = \vec{S}_i - 2\vec{r}\,(\vec{S}_i \cdot \vec{r}). \tag{3}$$

Task 3: Repeat the computation of T_c and z_c using the Swendsen-Wang or Wolff algorithm.

Unsupervised machine learning

We can use PCA and k-means to investigate condensed matter systems. One of the paradigmatic systems is the two-dimensional ferromagnetic Ising model. It has two phases separated by a well-known critical temperature at $T_c = 2.26$.

Task 1: Generate $10^3 - 10^4$ samples of L-by-L Ising systems using Metropolis or a cluster Monte Carlo method, with $L = 10$ for smaller systems, and $L = 32$ for larger systems at different temperatures.

Task 2: Perform PCA on these data points and find the 2 directions u_1 and u_2 with the largest variance. Make a scatter plot of the configurations in the u_1-u_2 plane. Describe and interpret your results. What is the difference between the small and the large system?

Hint: Use fit(PCA, X; maxoutdim = 2) of the package MultivariateStats.

We can use k-means to determine the boundary between the three (two) phases, so that when we are given a new data point, we can determine which phase it belongs to.

Task 3: Find the three stabilized centroids using k-means and divide the u_1-u_2 plane into three phase regions.

Hint: Use kmeans(X, 3) of the package Clustering.

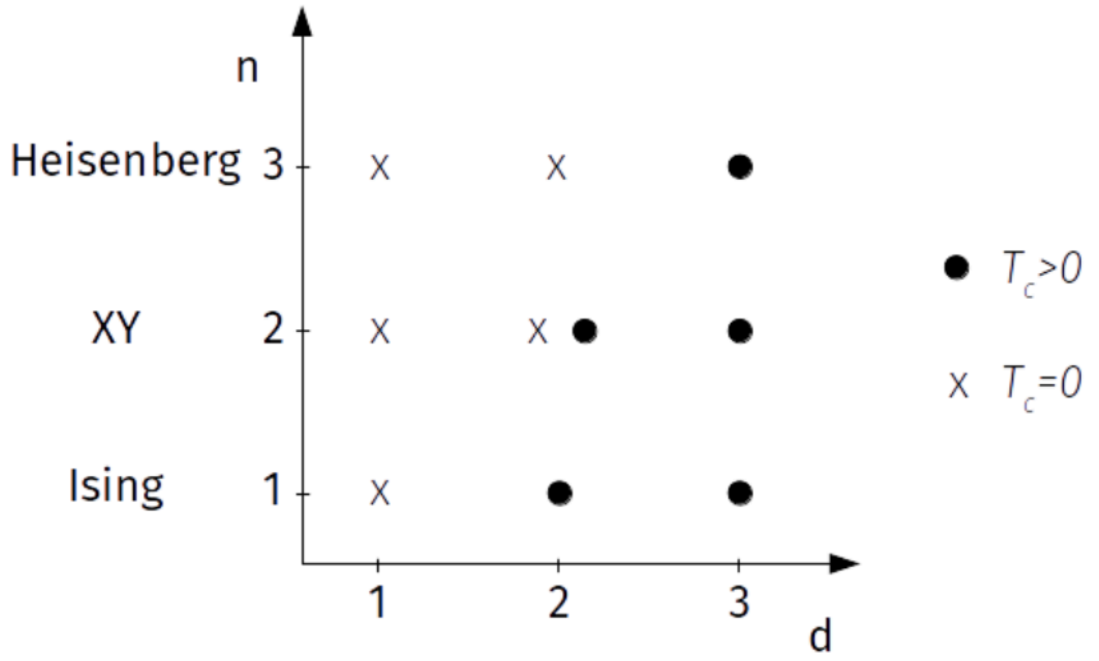
A very interesting phenomena can be observed for the data point in u_1 and u_2 direction.

Task 4 a): Plot the mean value of the absolute value of the data points in the u_1 direction as a function of temperature and compare it with the mean magnetization of the corresponding spin configurations.

Task 4 b): Plot the mean value of the absolute value of the data points in the u_2 direction as a function of temperature and compare it with the susceptibility of the corresponding spin configurations.

Heisenberg model

$$H = -J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j \qquad \vec{S}_i \in \mathbb{S}^2 \subset \mathbb{R}^3$$



Instructions: Heisenberg model

Metropolis algorithm:

- Change cluster representation
- Spin update:
    - $\overrightarrow{S}_{old} \rightarrow \overrightarrow{S}_{new}$, $\overrightarrow{S}_{new}$ is new random spin on S^2
    - Find new ΔE
    - Change update probability
- Keep structure
    - Thermalization, subsweeps …
- Magnetization $\overrightarrow{M} = \sum_i \overrightarrow{m}_i$, $|\overrightarrow{M}|$

Instructions: Heisenberg model

- Find critical temperature
- Compute the autocorrelation of $|\overrightarrow{M}|$ time at T_c
- Find the critical dynamical exponent z_c
  - $\tau \propto L^{z_c}$

Instructions: Heisenberg model

Wolff

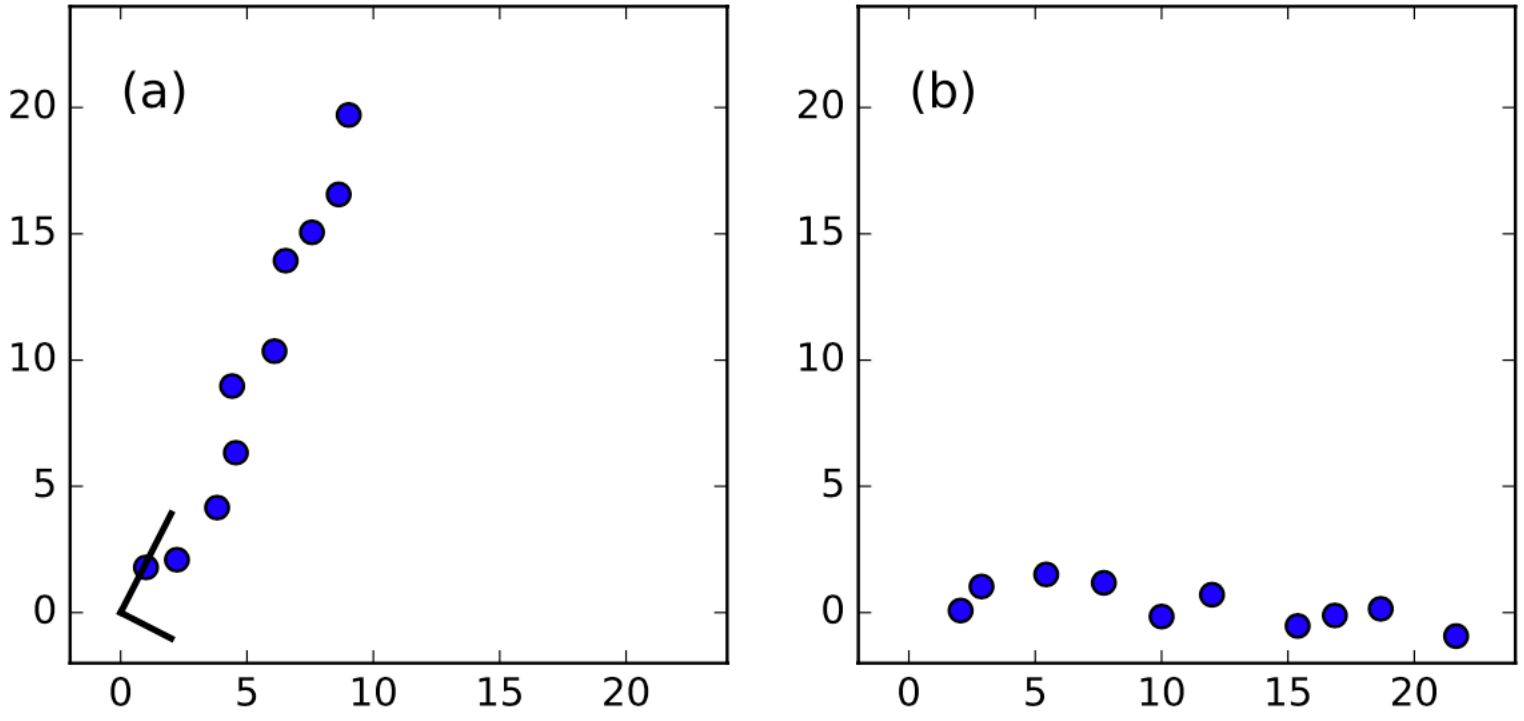
- Change cluster representation
- Spin update: reflect spin
- Build cluster:
  - Pick random spin (i,j,k)
  - Pick random vector $\vec{r}$
    - Bond probability: $p_{i,j} = 1 - \exp\left[-2\beta(\vec{S}_i \cdot \vec{r})(\vec{S}_j \cdot \vec{r})\right]$
    - $\overrightarrow{S}_{old} \to \overrightarrow{S}_{new} = \overrightarrow{S}_{old} - 2\vec{r}(\overrightarrow{S}_{old}\vec{r})$
    - Repeat for neighbors
- Keep structure
  - Thermalization …

Instructions: Heisenberg model

- Find critical temperature
- Compute the autocorrelation of $|\overrightarrow{M}|$ time at T_c
- Find the critical dynamical exponent z_c
    - $\tau \propto L^{z_c}$

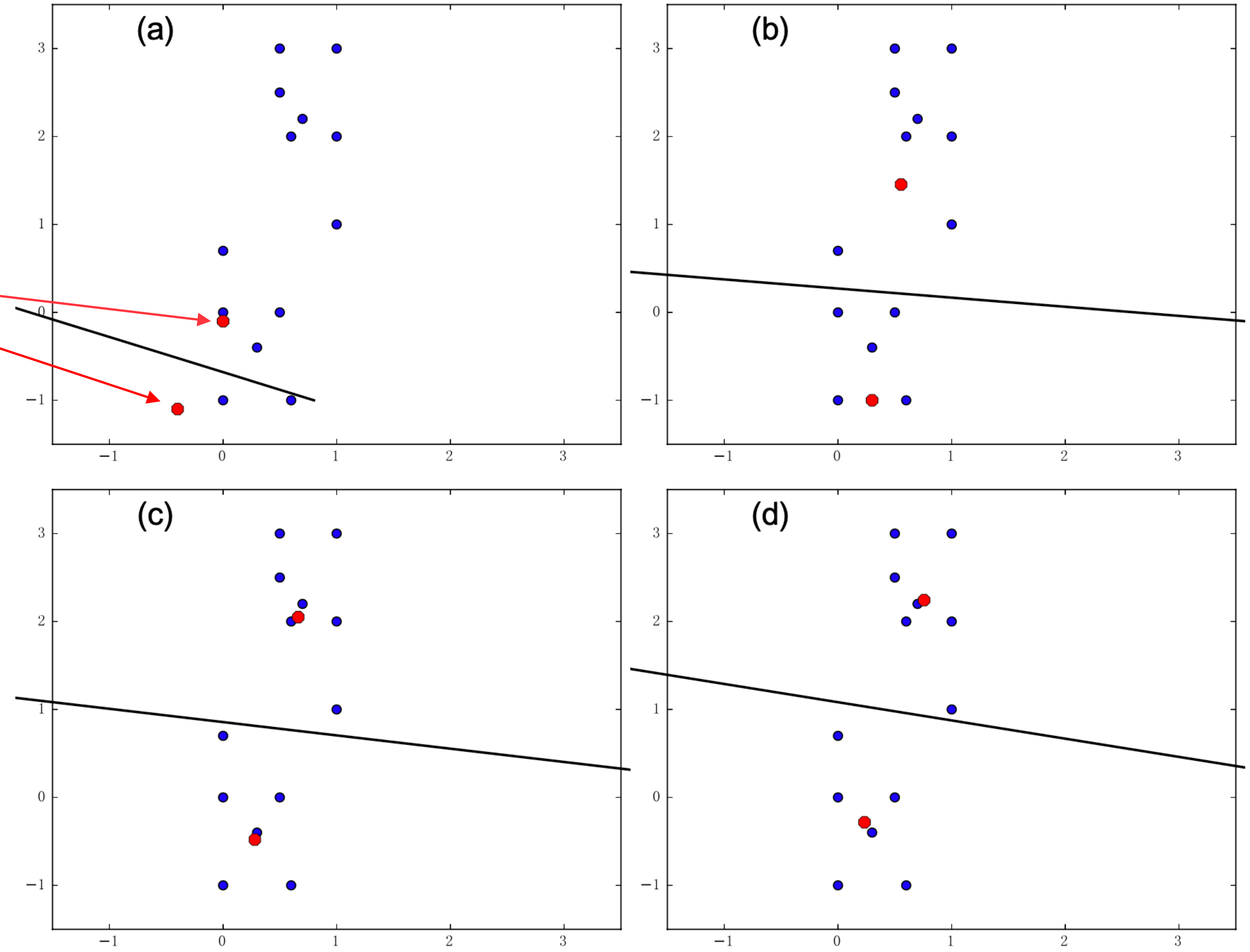
Unsupervised machine learning
PCA and k-means

Motivation PCA



Unsupervised machine learning
k-means

Algorithm k-means



Centroids

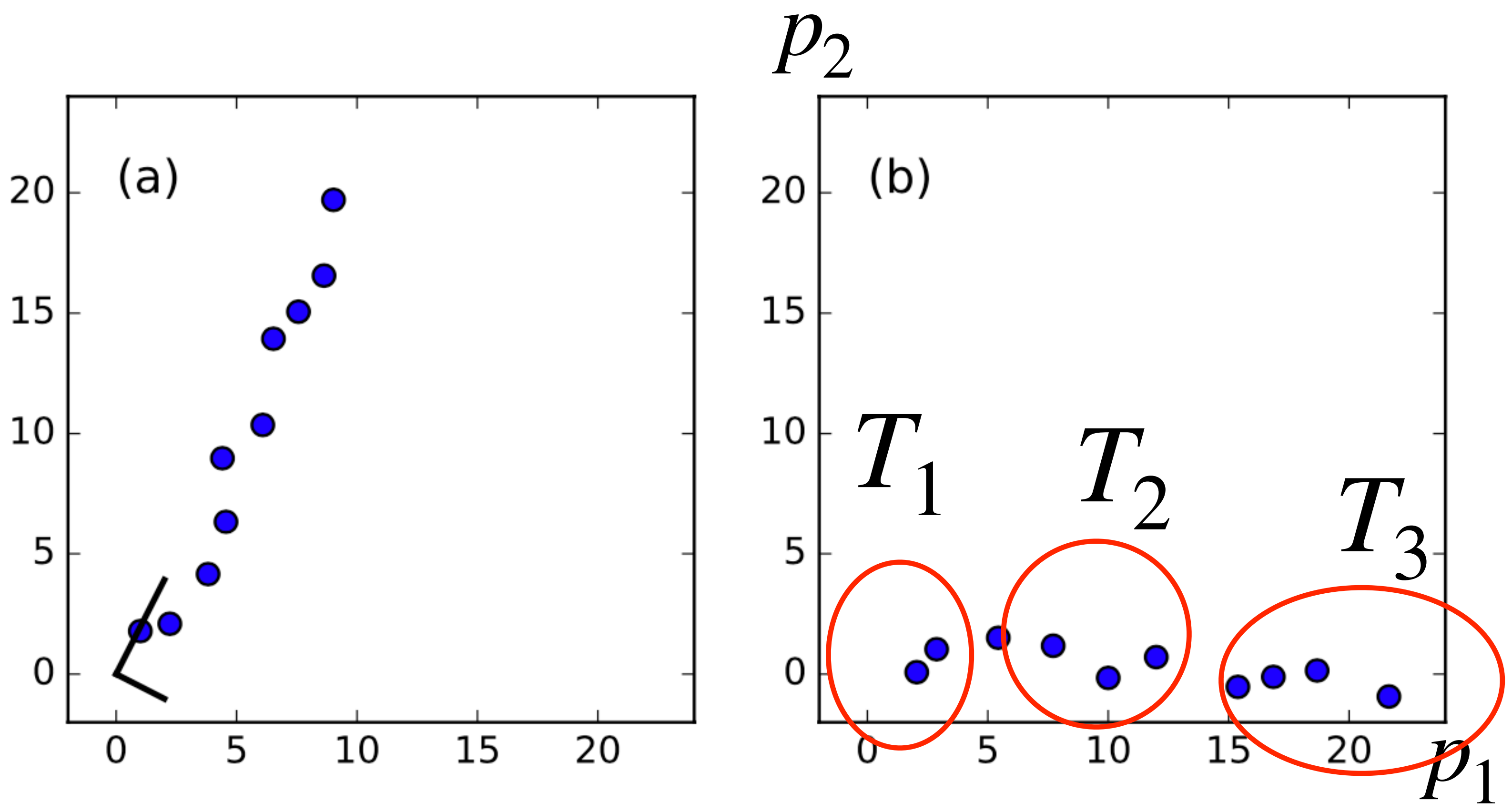
Instructions: Unsupervised machine learning

2D Ising model

- Generate spin configurations
  - Different system size and temperature
  - Store magnetization and susceptibility
  - Use code from Ex 1 or 3
- Perform PCA to find directions u_1 and u_2
  - Make scatter plot in u_1-u_2 plane
  - Interpret results
- K-means: find centroids
  - You can determine phase of new configurations

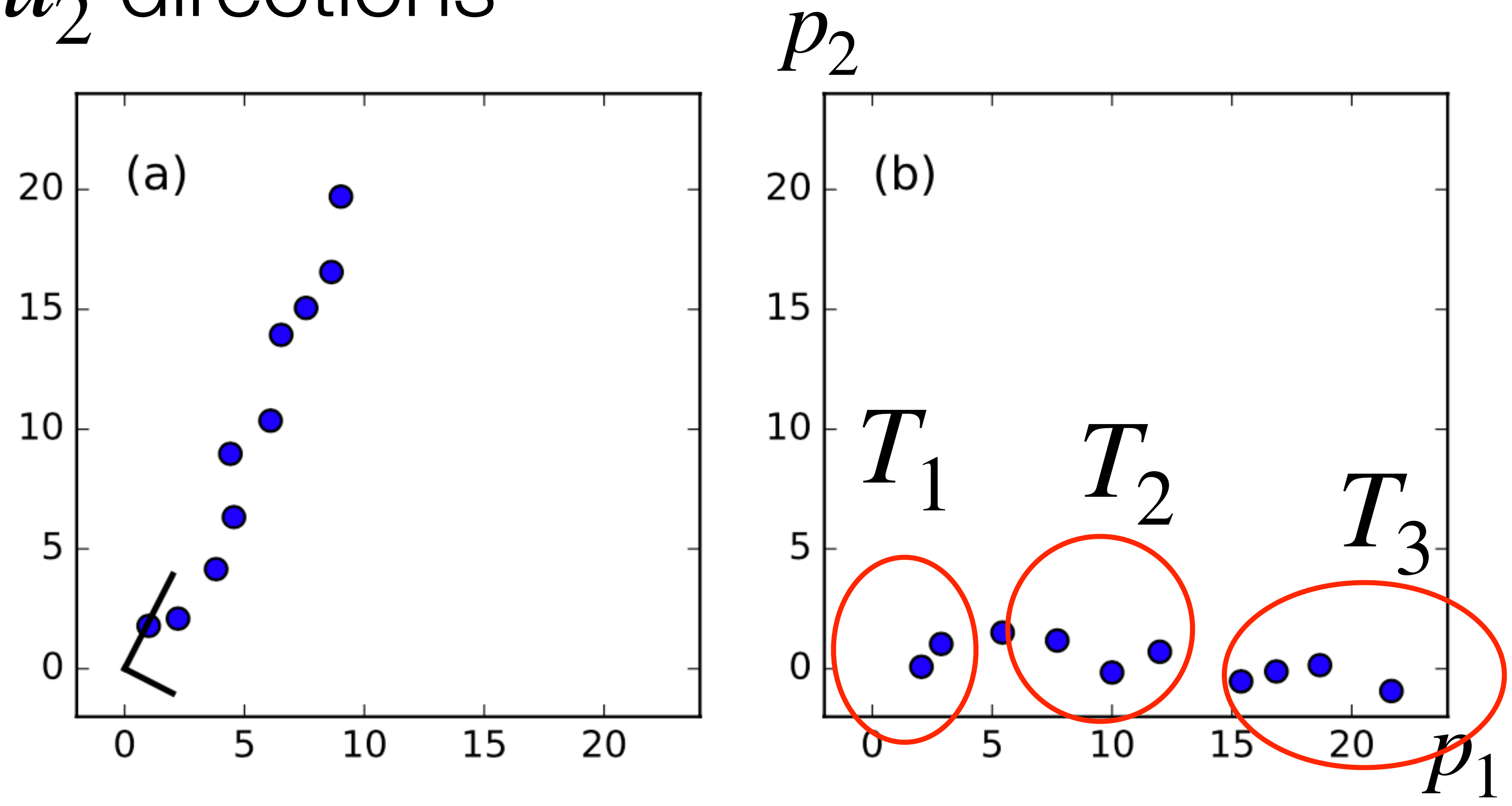
Instructions: Unsupervised machine learning

- Plot $\langle | p_1 | \rangle$ as a function of T:
  - p_1: data points in u_1 directions
  - Compare values with magnetization



Instructions: Unsupervised machine learning

- Plot $\langle |p_1| \rangle$ as a function of T:
  - p_1: data points in u_1 directions
  - Compare values with magnetization
- Plot $\langle |p_2| \rangle$ as a function of T:
  - p_2: data points in u_2 directions
  - Compare values with susceptibility


(a)
0
5
10
15
20
0
5
10
15
20
p_2
(b)
T_1
T_2
T_3
0
5
10
15
20
0
5
10
15
20
p_1