# Computational physics

Heiko Rieger and Adam Wysocki

Theoretical Physics
Saarland University

SS2017

# Contents

- Monte-Carlo methods

# Introduction to Monte-Carlo method

- Statistical physics deals with systems with many $\left(N \sim 10^{23}\right)$ degrees of freedom. For example:
  - System of $N$ classical particles in 3D $(\mathbf{r}_1, \ldots, \mathbf{r}_N, \mathbf{p}_1, \ldots, \mathbf{p}_N) = \left(\mathbf{r}^N, \mathbf{p}^N\right)$, with positions $\mathbf{r}_i = (x_i, y_i, z_i)$ and momenta $\mathbf{p}_i = (p_{x_i}, p_{y_i}, p_{z_i})$.
  - System of $N$ Ising spins $\mathbf{S} = (S_1, \ldots, S_N)$ with $S_i = \pm 1$ (model for magnetic dipole moments of atomic spins)
- System in thermodynamic equilibrium in the canonical or $NVT$ ensemble (temperature $T$, volume $V$ and number of particles $N$ fixed) is characterized by its energy $\mathcal{H}(\mathbf{r}, \mathbf{p})$ (Hamiltonian) or $\mathcal{H}(\mathbf{S})$ and the temperature of the heat bath $T \geq 0$.
- For example:

$$\mathcal{H}\left(\mathbf{r}^N, \mathbf{p}^N\right) = \underbrace{\sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m}}_{\text{kinetic energy}} + \underbrace{\sum_{i,j} V(\mathbf{r}_i - \mathbf{r}_j)}_{\text{potential energy}} \qquad (1)$$

or

$$\mathcal{H}(\mathbf{S}) = - \underbrace{\sum_{\langle ij \rangle} J_{ij} S_i S_j}_{\text{interaction energy}} - \underbrace{h \sum_{i=1}^{N} S_i}_{\text{external field}} \qquad (2)$$

# Introduction to Monte-Carlo method

- The canonical ensemble assigns a probability to each distinct microstate $\left(\mathbf{r}^N, \mathbf{p}^N\right)$ or $\mathbf{S}$ according to the so called Boltzmann distribution:

$$P\left(\mathbf{r}^N, \mathbf{p}^N\right) = \frac{1}{\mathcal{Z}} e^{-\beta\mathcal{H}\left(\mathbf{r}^N, \mathbf{p}^N\right)} \quad \text{or} \quad P\left(\mathbf{S}\right) = \frac{1}{\mathcal{Z}'} e^{-\beta\mathcal{H}(\mathbf{S})} \qquad (3)$$

  with inverse temperature $\beta = 1/T$ and canonical partition function (sum runs over all microstates of the system)

$$\mathcal{Z} = \int d\mathbf{p}^N d\mathbf{r}^N e^{-\beta\mathcal{H}\left(\mathbf{r}^N, \mathbf{p}^N\right)} \quad \text{or} \quad \mathcal{Z}' = \sum_{\mathbf{S}} e^{-\beta\mathcal{H}(\mathbf{S})} \qquad (4)$$

- Macroscopic observables of physical quantities $\mathcal{O}\left(\mathbf{r}^N, \mathbf{p}^N\right)$ or $\mathcal{O}\left(\mathbf{S}\right)$ are given by Boltzmann-weighted (or ensemble) averages

$$\langle\mathcal{O}\rangle = \int d\mathbf{p}^N d\mathbf{r}^N\, \mathcal{O}\left(\mathbf{r}^N, \mathbf{p}^N\right) P\left(\mathbf{r}^N, \mathbf{p}^N\right) \quad \text{or}$$

$$\langle\mathcal{O}\rangle = \sum_{\mathbf{S}} \mathcal{O}\left(\mathbf{S}\right) P\left(\mathbf{S}\right) \qquad (5)$$

# Introduction to Monte-Carlo method

- For example: kinetic energy $E_{kin} = \left\langle \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m} \right\rangle$ or mean magnetisation $M = \left\langle \sum_{i=1}^{N} S_i \right\rangle$.
- The weighted averages (thermodynamic expectation values) are multidimensional integrals or sums over all possible microstates of the systems.
- A straightforward approach to evaluate $\langle \mathcal{O} \rangle$ via a numerical quadrature cannot be performed with the today's computers even for systems with $N \approx 100$ particles.
- The idea of Monte Carlo methods is to evaluate the integrals or sums in a stochastic manner.

# Introduction to Monte-Carlo method

- Simple Monte Carlo sampling: generate randomly microstates $(\mathbf{r}, \mathbf{p})$ or $\mathbf{S}$ and calculate weighted averages. However, this method is useless, because most of the unbiased generated microstates have a negligible Boltzmann weight.
- **Importance sampling: generate microstates, which are already Boltzmann distributed.**
- How to generate a stochastic sequence of microstates $(\mathbf{r}, \mathbf{p})$ or $\mathbf{S}$, which asymptotically (in the thermodynamic limit) are distributed according to $P(\mathbf{r}, \mathbf{p})$ or $P(\mathbf{S})$?

# How to generate $P(\mathbf{S})$ for the Ising model

- Aim is to generate a sequence of microstates

$$\mathbf{S}_1 \to \mathbf{S}_2 \to \mathbf{S}_3 \to \dots \tag{6}$$

with transition probabilities $w(\mathbf{S}_n \to \mathbf{S}_{n+1})$ such that

$$\lim_{n\to\infty} P_n(\mathbf{S}) = P_{eq}(\mathbf{S}) = e^{-\beta\mathcal{H}(\mathbf{S})}/\mathcal{Z}', \tag{7}$$

where

$$P_n(\mathbf{S}) \equiv \text{probability of occurrence of } \mathbf{S} \text{ at } n\text{th position}$$
$$\text{in the sequence of microstates}$$

- How should transition probabilities $w(\mathbf{S}_n \to \mathbf{S}_{n+1})$ look like?
- Assume that the sequence is a realization of a Markov process, whose probability distribution $P_n(\mathbf{S})$ obeys a master equation:

$$P_{n+1}(\mathbf{S}) = P_n(\mathbf{S}) + \sum_{\mathbf{S}'} w\left(\mathbf{S}' \to \mathbf{S}\right) P_n\left(\mathbf{S}'\right)$$
$$- \sum_{\mathbf{S}'} w\left(\mathbf{S} \to \mathbf{S}'\right) P_n(\mathbf{S})$$

# Detailed balance

- Markov process is a **memorylessness** stochastic process, i.e., the probability of moving to the next state depends only on the present state and not on the previous states:

$$w\left(\mathbf{S}_n \to \mathbf{S}_{n+1}\right) \equiv P\left(\mathbf{S}_{n+1}|\mathbf{S}_n, \ldots, \mathbf{S}_1\right) = P\left(\mathbf{S}_{n+1}|\mathbf{S}_n\right) \quad (8)$$

- Let's define the probability current from state $\mathbf{S}$ to state $\mathbf{S}'$ as

$$\begin{aligned} J_n\left(\mathbf{S} \to \mathbf{S}'\right) &= -w\left(\mathbf{S}' \to \mathbf{S}\right) P_n\left(\mathbf{S}'\right) + w\left(\mathbf{S} \to \mathbf{S}'\right) P_n\left(\mathbf{S}\right) \\ &= -J_n\left(\mathbf{S}' \to \mathbf{S}\right) \end{aligned} \quad (9)$$

  Thus the master equation can be written as

$$P_{n+1}\left(\mathbf{S}\right) = P_n\left(\mathbf{S}\right) - \textstyle\sum_{\mathbf{S}'(\neq\mathbf{S})} J_n\left(\mathbf{S} \to \mathbf{S}'\right) \quad (10)$$

- In the steady-state $P_{n+1}\left(\mathbf{S}\right) = P_n\left(\mathbf{S}\right)$ and

$$\sum_{\mathbf{S}'(\neq\mathbf{S})} J_{\text{st}}\left(\mathbf{S} \to \mathbf{S}'\right) = 0, \ \forall\mathbf{S} \quad (11)$$

  i.e., the net flow from state **S** vanishes (note the analogy to Kirchhoff's junction rule).

# Detailed balance

- A stronger condition for stationarity is the so-called detailed balance,
$$J_{\mathrm{eq}}\left(\mathbf{S} \to \mathbf{S}'\right) = 0 \;\forall \mathbf{S}, \mathbf{S}', \tag{12}$$
i.e., there are no currents in the system at all.
- Detailed balance forbids **circular currents**, which are generally allowed in case of a weak stationarity condition. Circular currents correspond to **cyclic solutions** of the type $P_n\left(\mathbf{S}\right) = P_{n+m}\left(\mathbf{S}\right)$, where $m$ is the length of the cycle, and are excluded by detailed balance.
- The detailed balance condition for the time- or $n$-independent distribution reads as
$$w\left(\mathbf{S}' \to \mathbf{S}\right) P_{eq}\left(\mathbf{S}'\right) = w\left(\mathbf{S} \to \mathbf{S}'\right) P_{eq}\left(\mathbf{S}\right). \tag{13}$$

# How to choose $w\left(\mathbf{S} \to \mathbf{S}'\right)$

This means that, a set of transition probabilities $w\left(\mathbf{S} \to \mathbf{S}'\right)$, which satisfy:

$$\begin{aligned} \frac{w\left(\mathbf{S} \to \mathbf{S}'\right)}{w\left(\mathbf{S}' \to \mathbf{S}\right)} = \frac{P_{eq}\left(\mathbf{S}'\right)}{P_{eq}\left(\mathbf{S}\right)} &= e^{-\beta[\mathcal{H}(\mathbf{S}')-\mathcal{H}(\mathbf{S})]} \\ &= e^{-\beta\Delta\mathcal{H}(\mathbf{S},\mathbf{S}')} \quad \forall\, \mathbf{S}', \mathbf{S} \end{aligned} \tag{14}$$

guarantee that the stationary solution of the master equation, which also is the stationary distribution of the generated Markov process, is the Boltzmann distribution

$$P_{eq}\left(\mathbf{S}\right) = \frac{1}{\mathcal{Z}'} e^{-\beta\mathcal{H}(\mathbf{S})}. \tag{15}$$

# Metropolis–Hastings algorithm

$$w\left(\mathbf{S}\to\mathbf{S}'\right)=\begin{cases}e^{-\beta\Delta\mathcal{H}(\mathbf{S},\mathbf{S}')} & \text{if } \Delta\mathcal{H}\left(\mathbf{S},\mathbf{S}'\right)>0\\ 1 & \text{otherwise}\end{cases} \tag{16}$$

where $\Delta\mathcal{H}\left(\mathbf{S},\mathbf{S}'\right)=\mathcal{H}\left(\mathbf{S}'\right)-\mathcal{H}\left(\mathbf{S}\right)$ is the energy difference between the old $\mathbf{S}$ and the new state $\mathbf{S}'$.

- If $\Delta\mathcal{H}\leq 0$, i.e., new state has a lower energy than the old, accept always the new state.
- If $\Delta\mathcal{H}>0$, i.e., new state has a higher energy than the old, accept the new state with a probability $0<e^{-\beta\Delta\mathcal{H}}<1$.

# Practice issues of Monte-Carlo simulations

- Metropolis–Hastings algorithm:
  1. generate new state $\mathbf{S}'$
  2. calculate $\Delta\mathcal{H}(\mathbf{S}, \mathbf{S}')$
  3. generate a uniformly distributed random number $r \in [0, 1]$
  4. accept $\mathbf{S}'$ if $r < e^{-\beta\Delta\mathcal{H}}$, else go to (1)
- Assume that $\mathbf{S}_1, \ldots, \mathbf{S}_{\tau_{run}}$ is a sequence of $\tau_{run}$ microstates generated by the Metropolis algorithm and $P(\mathbf{S}_l) = P_{eq}(\mathbf{S}_l)$, then the ensemble averages of observables $\mathcal{O}(\mathbf{S})$ is given by

$$\langle\mathcal{O}\rangle = \lim_{\tau_{run}\to\infty}\left(\frac{1}{\tau_{run}}\sum_{l=1}^{\tau_{run}}\mathcal{O}(\mathbf{S}_l)\right), \tag{17}$$

  since the Boltzmann weight is already included in the frequency of occurrence of state $\mathbf{S}_l$.
- Generate new states $\mathbf{S}'$ such that the change in energy, $\Delta\mathcal{H}(\mathbf{S}, \mathbf{S}')$, is not to large and that half of the trials are accepted.

# Single-spin-flip dynamics

- A common update method of Ising spins $S_i = \pm 1$ is the single-spin-flip dynamics $S_i \to -S_i$, i.e., a trial state is

$$\mathbf{S}' = (S_1, \ldots, S_{i-1}, -S_i, S_{i+1}, \ldots, S_N) \tag{18}$$

- The energy change $\Delta \mathcal{H}\left(\mathbf{S}', \mathbf{S}\right)$ is

$$\Delta \mathcal{H}\left(S_i \to -S_i\right) = 2 \sum_{j \neq i}^{N} J_{ij} S_j S_i + 2hS_i = 2S_i b_i \tag{19}$$

with the local field $b_i = h + \sum_{j \neq i}^{N} J_{ij} S_j$.

- Acceptance ratio:

$$w\left(S_i \to -S_i\right) = \begin{cases} e^{-2\beta S_i b_i} & \text{if } S_i b_i > 0 \\ 1 & \text{otherwise} \end{cases} \tag{20}$$

Spin $S_i$ is parallel to the local field $b_i$ if $S_i b_i > 0$.

# Simple Monte-Carlo computer program

- Equilibrate the system before starting measurements.

```
set initial configuration; // e.g. high temperature
m_av=0; // calculate e.g. magnetization per spin
for (step=1;step<=MC_steps;step++)
 {for (i=1;i<=N;i++)
   {if (S[i]*b[i]<0)
     {S[i]=-S[i];}
    else if (rand()<exp(-2*beta*S[i]*b[i]))
     {S[i]=-S[i];}
    if (step>MC_equi_steps)
     {m_av+=S[i];}
   }
 }
m_av/=N*(MC_steps-MC_equi_steps);
```

# Ising model

- Ising Hamiltonian: $\mathcal{H}\left(\mathbf{S}\right) = -\sum_{\langle ij \rangle} J_{ij} S_i S_j - h \sum_{i=1}^{N} S_i$
  - Ferromagnetic interaction $J_{ij} > 0$ favors parallel spins.
  - Antiferromagnetic interaction $J_{ij} < 0$ favors antiparallel spins.
- In realistic model of a 3D magnet, spins are arranged on a cubic lattice and interact with its neighbors with the same interaction strength $J_{ij} = J$.
- Important observables:
  - Mean magnetization per spin:

$$\langle m \rangle = \left\langle \frac{1}{N} \sum_{i=1}^{N} S_i \right\rangle \in [-1, 1] \tag{21}$$

  - Susceptibility per spin:

$$\chi = \beta N \left( \left\langle m^2 \right\rangle - \left\langle m \right\rangle^2 \right) \in [-\beta N, \beta N] \tag{22}$$

  - Mean energy per spin:

$$e = \frac{1}{N} \left\langle \mathcal{H}\left(\mathbf{S}\right) \right\rangle \tag{23}$$

  - Specific heat per spin:

$$c = N \left( \left\langle e^2 \right\rangle - \left\langle e \right\rangle^2 \right) \tag{24}$$

# A bit of statistical mechanics

- Reminder: The canonical partition function and the Helmholtz free energy is

$$\mathcal{Z} = \textstyle\sum_{\mathbf{S}} e^{-\beta\mathcal{H}(\mathbf{S})} \quad \text{and} \quad \beta\mathcal{F} = -\ln\left(\mathcal{Z}\right), \tag{25}$$

respectively. Expectation values in the canonical ensemble are

$$\langle(\ldots)\rangle = \frac{\sum_{\mathbf{S}}(\ldots)\, e^{-\beta\mathcal{H}}}{\mathcal{Z}} \tag{26}$$

- Magnetization:

$$\begin{aligned}
\langle M\rangle &= \left\langle \sum_{i=1}^{N} S_i \right\rangle = -\frac{\partial\mathcal{F}}{\partial h} \\
&= \frac{\partial}{\partial h}\left\{\frac{1}{\beta}\ln\left[\sum_{\mathbf{S}}\exp\left(\beta\sum_{\langle ij\rangle}J_{ij}S_iS_j + \beta h\sum_i S_i\right)\right]\right\} \\
&= \frac{1}{\beta}\frac{\sum_{\mathbf{S}}\beta\left(\sum_i S_i\right)e^{-\beta\mathcal{H}}}{\sum_{\mathbf{S}}e^{-\beta\mathcal{H}}} = \sum_{\mathbf{S}}\frac{Me^{-\beta\mathcal{H}}}{\mathcal{Z}} = \langle M\rangle
\end{aligned} \tag{27}$$

# A bit of statistical mechanics

- Susceptibility (fluctuation of the magnetization):

$$\begin{aligned}\chi N &= \frac{\partial M}{\partial h} = -\frac{\partial^2 \mathcal{F}}{\partial h^2} = \frac{\partial}{\partial h}\frac{\sum_{\mathbf{S}}\left(\sum_i S_i\right)e^{-\beta\mathcal{H}}}{\mathcal{Z}} \\ &= \frac{\sum_{\mathbf{S}}\beta\left(\sum_i S_i\right)^2 e^{-\beta\mathcal{H}}}{\mathcal{Z}} - \frac{\beta\left(\sum_{\mathbf{S}}\left(\sum_i S_i\right)e^{-\beta\mathcal{H}}\right)^2}{\mathcal{Z}^2} \\ &= \beta\Big\{\Big\langle\Big(\sum_i S_i\Big)^2\Big\rangle - \Big\langle\sum_i S_i\Big\rangle^2\Big\} \qquad (28)\end{aligned}$$

- Mean energy ($E/N = e$): $E = \langle\mathcal{H}\rangle = -\frac{\partial \ln(\mathcal{Z})}{\partial\beta} = \mathcal{F} - T\frac{\partial\mathcal{F}}{\partial T}$
- Specific heat ($C/N = c$ fluctuation of the energy):

$$C = \frac{\partial E}{\partial T} = -T\frac{\partial^2\mathcal{F}}{\partial T^2} = \frac{\langle H^2\rangle - \langle H\rangle^2}{T^2} \qquad (29)$$

# Phase behaviour of the Ising model

- Ising's exact solution in 1D for $N \to \infty$ (thermodynamic limit)

$$\begin{aligned} f &= \lim_{N\to\infty} \frac{\mathcal{F}}{N} = \lim_{N\to\infty} \frac{1}{\beta N} \ln\left(\mathcal{Z}_N\right) \\ &= \beta^{-1} \ln\left( e^{\beta J} \cosh\left(\beta h\right) + \sqrt{e^{2\beta J} \sinh^2\left(\beta h\right) + e^{-2\beta J}} \right) \end{aligned}$$

$$\Rightarrow m = \frac{\partial f}{\partial h} = \frac{e^{\beta J} \sinh\left(\beta h\right)}{\sqrt{e^{2\beta J} \sinh^2\left(\beta h\right) + e^{-2\beta J}}} \qquad (30)$$

  and $m \to 0$ for $h \to 0$ since $\sinh\left(\beta h\right) \to 0$ for $h \to 0$
  $\Rightarrow$ No phase transition of the Ising model with short-range interaction in 1D

- In higher dimensions ($d > 1$) and at zero external field ($h = 0$) Ising model exhibits an continuous phase transition from a paramagnetic (disordered) phase ($m = 0$) for $T > T_C$ to a ferromagnetic (ordered) phase ($m \neq 0$) for $T < T_C$.

# Phase behavior of the Ising model in $d > 1$

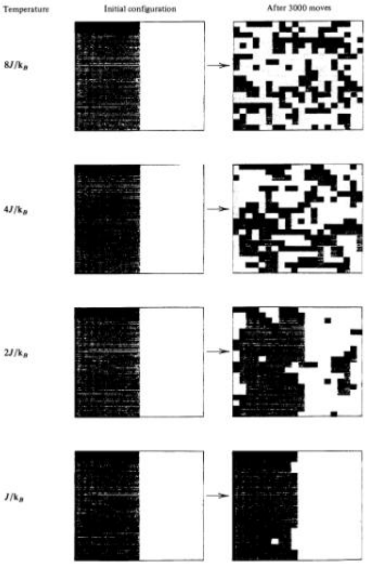


**Fig. 6.3.** Configurations in a Monte Carlo trajectory of a 20 × 20 Ising model.

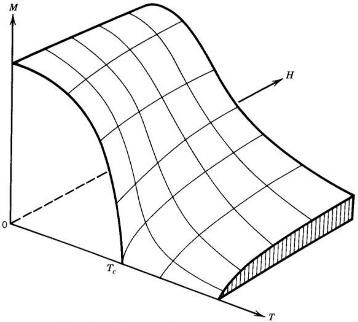


**Fig. 16.1** Equation of state surface for a magnetic system with a second-order phase transition at zero field ($H = 0$). The surface is symmetric under $M \to -M$. The lower half is not shown.

# Near the critical point: Exponents

- Magnetization is the order parameter, which describes the spontaneous symmetry breaking:

$$m \propto \begin{cases} 0 & \text{for } T \geq T_C \\ (T_C - T)^{\beta} & \text{for } T < T_C \end{cases}$$

- Continuous phase transition are characterized by divergent susceptibilities, an infinite correlation length, and a power-law decay of correlations near criticality.
- Susceptibility: $\chi \propto |T - T_C|^{-\gamma}$
- Specific heat: $c \propto |T - T_C|^{-\alpha}$
- The spatial spin-spin correlation function:

$$C(r) = \lim_{N \to \infty} \frac{1}{N} \sum_{\substack{i,j \text{ with} \\ r = |\mathbf{r}_i - \mathbf{r}_j|}} \{\langle S_i S_j \rangle - \langle S_i \rangle \langle S_j \rangle\} \propto r^{-(d-2+\eta)} e^{-r/\xi}$$

with correlation length $\xi \propto |T - T_C|^{-\nu}$.
$\Rightarrow$ At the critical point $T = T_C$: $C(r) \propto r^{-(d-2+\eta)}$

# Phase behavior of the Ising model in $d > 1$

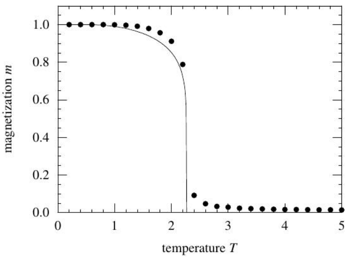


Figure 1: Magnetization $m$ versus temperature $T$ of a $100 \times 100$ Ising model in 2D.

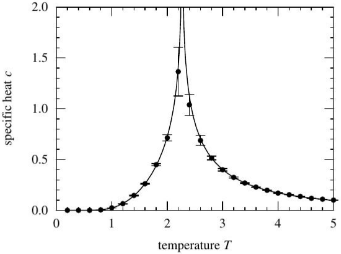


Figure 2: Specific heat $c$ versus temperature $T$ of a $100 \times 100$ Ising model in 2D.

# Near the critical point: Scaling relations

- Critical exponents obey scaling relations.
- For $T \neq T_C$:

$$\begin{aligned} \sum_{\mathbf{r}} C(r) &\propto \int dr\; r^{d-1} C(r) \approx \int_1^{\xi} dr\; r^{d-1} r^{-d+2-\eta} \\ &= \int_1^{\xi} dr\; r^{1-\eta} \propto \xi^{2-\eta} \propto \left|T - T_C\right|^{-(2-\eta)\nu} \end{aligned} \tag{31}$$

  and

$$\begin{aligned} \sum_{\mathbf{r}} C(r) &= \sum_{i,j} \left\{ \langle S_i S_j \rangle - \langle S_i \rangle \langle S_j \rangle \right\} \\ &= \left\langle M^2 \right\rangle - \left\langle M \right\rangle^2 \propto \chi \propto \left|T - T_C\right|^{-\gamma} \end{aligned} \tag{32}$$

  $\Rightarrow$ Fisher equality: $2 - \eta = \gamma/\nu$
- Rushbrooke inequality: $\alpha + 2\beta + \gamma \geq 2$ (Follows from the formula for the difference between heat capacities at constant $h$ and at constant $m$)

# Near the critical point: Scaling relations

- The singular part of the free energy $\mathcal{F}$ should vanish with increasing volume of the correlated domain:

$$\mathcal{F}_{sing} \propto \xi^{-d} \propto |T - T_C|^{\nu d} \tag{33}$$

From specific heat: $c \propto \frac{\partial^2 \mathcal{F}}{\partial T^2} \propto |T - T_C|^{-\alpha}$.

$$\Rightarrow \mathcal{F}_{sing} \propto |T - T_C|^{2-\alpha} \tag{34}$$

$\Rightarrow$ Hyperscaling relation: $2 - \alpha = \nu d$

# Finite size scaling

- Singularities are smoothed out in finite systems.
- For example, $\chi \to \infty$ as $T \to T_c$ in an infinite system. On the other hand, $\chi \propto L$ in a finite system of dimension $L = N^{1/d}$.
- Scaling ansatz:

$$\chi\left(L, T\right) \propto \left|t\right|^{-\gamma} \chi_0\left(L/\xi\right) \propto \xi^{\gamma/\nu} \chi_0\left(L/\xi\right) \tag{35}$$

with $t = \frac{T - T_c}{T_c}$ and

$$\chi_0\left(x\right) \propto \begin{cases} \text{const} & \text{, for } x \gg 1 \\ x^{\gamma/\nu} & \text{, for } x \ll 1 \end{cases} \tag{36}$$

$\Rightarrow$ Such a choice of $\chi_0$ ensures

- correct power-law scaling of $\chi$ as $L \to \infty$
- $T$-independent $\chi$ for $\xi \gg L$

# Finite size scaling

With scaling function $\tilde{\chi}(x) = x^{-\gamma}\chi_0(x^\nu)$ (and analog for $\tilde{m}$ and $\tilde{c}$) we get rid of the unknown $\xi$:

$$m\left(L,T\right) \propto L^{-\beta/\nu}\tilde{m}\left(L^{1/\nu}t\right),\ \tilde{m}\left(x\right) \propto \begin{cases} \text{const} & \text{, for } x \ll 1 \\ x^{\beta} & \text{, for } x \gg 1 \end{cases} \tag{37}$$

$$\chi\left(L,T\right) \propto L^{\gamma/\nu}\tilde{\chi}\left(L^{1/\nu}t\right),\ \tilde{\chi}\left(x\right) \propto \begin{cases} \text{const} & \text{, for } x \ll 1 \\ x^{-\gamma} & \text{, for } x \gg 1 \end{cases} \tag{38}$$

$$c\left(L,T\right) \propto L^{\alpha/\nu}\tilde{c}\left(L^{1/\nu}t\right),\ \tilde{c}\left(x\right) \propto \begin{cases} \text{const} & \text{, for } x \ll 1 \\ x^{-\alpha} & \text{, for } x \gg 1 \end{cases} \tag{39}$$

$\Rightarrow$ For example, simulation data of $\chi_L(t)$ for different $L$ collapse on each other if we plot $L^{-\gamma/\nu}\chi_L(t)$ versus $L^{1/\nu}t$ with the right choice of $\gamma$, $\nu$ and $T_c$.

# Finite size scaling

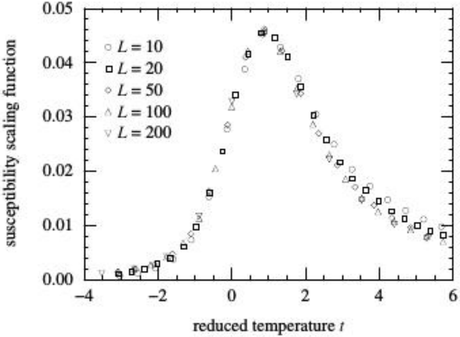


Figure 3: Data collapse of $\chi_L(t)$ for 2D Ising model. Plot of $L^{-\gamma/\nu}\chi_L(t)$ versus $L^{1/\nu}t$. Data collapse achieved for $\gamma = 1.76$, $\nu = 1$ and $T_c = 2.27J$.

# First choice method to estimate $T_c$

- Estimate of $T_c$ via Binder cumulant: $g = \frac{1}{2}\left(3 - \frac{\langle M^4\rangle}{\langle M^2\rangle^2}\right)$
- $g$ characterizes the deviation of $p(M)$, distribution of magnetization $M$, from a Gaussian distribution.

$$g = \begin{cases} 0 & \text{if } p(M) \text{ is a Gaussian distribution} \\ 1 & \text{if } p(M) = \delta(M \pm m) \end{cases} \tag{40}$$

  $\Rightarrow$ $g(T)$ is a step function in the thermodynamic limit.
- $g$ does not depend on $L$ at $T = T_c$, because

$$\langle |M|^n \rangle \propto L^{-n\beta/\nu} \tilde{M}_n\left(L^{1/\nu} t\right) \tag{41}$$

  with scaling functions $\tilde{M}_n$ and thus

$$g = \frac{1}{2}\left(3 - \frac{\tilde{M}_4\left(L^{1/\nu} t\right)}{\tilde{M}_2\left(L^{1/\nu} t\right)^2}\right) \propto \tilde{g}\left(L^{1/\nu} t\right) \tag{42}$$

  $\Rightarrow$ Curves of $g_L(t)$ for different $L$ intersect at the critical point $T = T_c$.

# Further scaling relation

- From Eq.(41) and Eq.(38) we get a further scaling relation:

$$\chi \sim N \left\langle |M|^2 \right\rangle \Rightarrow L^{\gamma/\nu} \sim L^d L^{-2\beta/\nu} \Rightarrow \gamma = d\nu - 2\beta \qquad (43)$$

  With hyperscaling relation ($2 - \alpha = \nu d$) follows the Rushbrooke equality

$$\alpha + 2\beta + \gamma = 2 - \nu d + 2\beta + \nu d - 2\beta = 2 \qquad (44)$$

# Known results of the $d$-dimensional Ising model

| | | | | | $(MFT)$ |
|---|---|---|---|---|---|
| | $d$ | 1 | 2 | 3 | $\geq 4$ |
| | $k_B T_C / J$ | 0 | $2.26917\dots$ | $4.511\dots$ | $2d$ |
| $\xi \propto t^{-\nu}$ | $\nu$ | 1 | 1 | $0.629\dots$ | $1/2$ |
| $c \propto t^{-\alpha}$ | $\alpha$ | 1 | 0 | $0.105\dots$ | 0 |
| $m \propto t^{\beta}$ | $\beta$ | 0 | $1/8$ | $0.326\dots$ | $1/2$ |
| $\chi \propto t^{-\gamma}$ | $\gamma$ | 1 | $7/4$ | $1.239\dots$ | 1 |
| $C \propto r^{-d+2-\eta}$ | $\eta$ | 1 | $1/4$ | $0.036\dots$ | 0 |
| $m \propto h^{1/\delta}$ | $\delta$ | $\infty$ | 15 | $4.789\dots$ | 3 |

# $q$-state Potts models

- $S_i \in \{1, 2, ..., q\}$ and $J > 0$ for ferromagnetic interaction
- The Hamiltonian is

$$\mathcal{H} = -J \sum_{\langle ij \rangle} \delta_{S_i S_j} \tag{45}$$

  with Kronecker delta $\delta_{S_i S_j}$, which equals one whenever $S_i = S_j$ and zero otherwise.
- The $q = 2$ Potts model is equivalent to the Ising model with $J_{Ising} = J_{Potts}/2$.
- Phase transition in 2D for $q \leq 4$ is continuous ($q > 2$ not Ising universality class) and for $q > 4$ first order.
- Order parameter is

$$m_\sigma = \left\langle \frac{1}{N(q-1)} \sum_{i=1}^{N} (q\delta_{S_i,\sigma} - 1) \right\rangle \tag{46}$$

  for $\sigma \in \{1, 2, ..., q\}$, such that $m_\sigma = 0$ for $T > T_c$ and $m_\sigma > 0$ for $T < T_c$.

# $q$-state Potts models

- There is a duality transformation for the $q$-state Potts model on a square lattice (like the Kramers–Wannier duality for Ising model), which maps the partition function $\mathcal{Z}$ at low $T$ onto $\mathcal{Z}$ at high $T$.
- The Potts model is self-dual if $q = (e^{\beta_c J} - 1)^2$, which defines the critical temperature

$$T_c = \frac{J}{\ln(1+\sqrt{q})}. \tag{47}$$

- For 2-state Potts model (Ising model):

$$T_c = \frac{J_{Potts}}{\ln(1+\sqrt{2})} = \frac{2J_{Ising}}{\ln(1+\sqrt{2})} \approx 2.27 J_{Ising}, \tag{48}$$

which agrees with Onsager's exact solutions.

# $q$-state Potts models

- Single-spin-flip dynamics:
  - For small $q$: choose a spin $S_i$ and generate a random number out of $\{1, \ldots, q\} \setminus S_i$.
  - For large $q$ use a heat-bath algorithm: choose a spin $S_k$ and generate a new value of spin $S_k = n \in \{1, 2, ..., q\}$ with probability

    $$p_n = e^{-\beta\mathcal{H}_n} / \sum_{m=1}^{q} e^{-\beta\mathcal{H}_m}, \tag{49}$$

    where $\mathcal{H}_n$ is the energy of the system when $S_k = n$. This works because

    $$\mathcal{H} = -J \sum_{\substack{\langle ij \rangle \\ (i,j \neq k)}} \delta_{S_i S_j} - J \sum_{\text{N.N. of } k} \delta_{S_i S_k} \tag{50}$$

    and the first sum cancels out from Eq.(49).

# *n*-vector or continuous spin model

- In the *n*-vector model, a *n*-component spins $\mathbf{S}_i \in \mathbb{R}^n$ with unit length ($|\mathbf{S}_i| = 1$) are placed on the vertices of a lattice.
- Hamiltonian: $\mathcal{H} = -J \sum_{\langle ij \rangle} \mathbf{S}_i \mathbf{S}_j$
- $n = 2$: XY (or rotor) model
  - Hamiltonian: $\mathcal{H} = -J \sum_{\langle ij \rangle} \cos(\theta_i - \theta_j)$
  - No spontaneous magnetization in 2D, however, there is a Kosterlitz–Thouless transition. It is a transition from bound vortex-antivortex pairs at low $T$ to unpaired vortices and anti-vortices at $T_c$. Furthermore, at high $T$ correlations decay exponentially fast, while at low $T$ they decay with power law.
  - Model for superfluid helium or hexatic liquid crystals.
  - Single-spin-update dynamics: $\theta_i \to \theta_i + \Delta\theta$ with random $\Delta\theta \in [0, 2\pi]$
- $n = 3$: Heisenberg model

# Kosterlitz–Thouless transition of the XY model

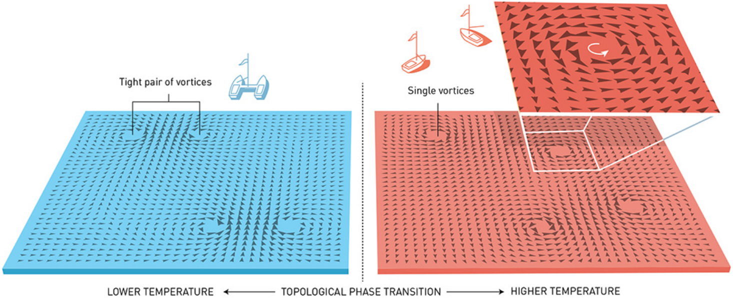


Illustration: ©Johan Jarnestad/The Royal Swedish Academy of Sciences

# Autocorrelation function

- Ideally, the simulation should explore the whole phase space and averages should be calculated from independent samples of the Boltzmann distribution. Thus it is crucial to know when two successive measurements are uncorrelated. Such an information could provide the correlation time $\tau$.
- Consider time autocorrelation function of the magnetization

$$\begin{aligned}\chi(t) &= \int dt'[m(t') - \langle m\rangle][m(t'+t) - \langle m\rangle] \\ &= \int dt'[m(t')m(t'+t) - \langle m\rangle^2] \qquad (51)\end{aligned}$$

- $\chi(t)$ can be also calculated via fast Fourier transform and convolution.
- Often $\chi(t) \propto e^{-t/\tau}$ and $\tau$ is an estimate of the correlation time.
- Alternative, calculate integrated correlation time via

$$\int_0^\infty dt \frac{\chi(t)}{\chi(0)} = \int_0^\infty dt\, e^{-\frac{t}{\tau}} = \tau \qquad (52)$$

- Thus, if run lasts a time $t_{max}$, then the number of independent measurements is of the order $n \approx \frac{t_{max}}{2\tau}$.

# Autocorrelation function

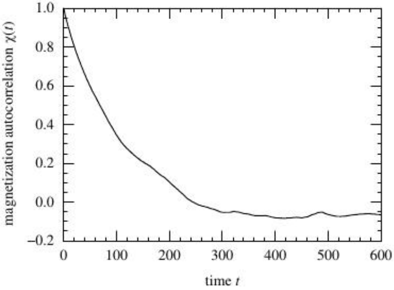
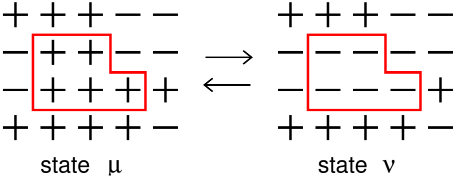


Figure 4: Magnetization autocorrelation function $\chi(t)/\chi(0)$ for $100 \times 100$ Ising model in 2D at $T = 2.4$ with $J = 1$. Time measured in MC steps per spin.

- Near critical point $T = T_c$ the correlation length $\xi \propto t^{-\nu}$ and time $\tau \propto t^{-z\nu}$ diverges (critical slowing down).
- Combining both scaling relations we get $\tau \propto \xi^z$ and in a finte system ($\xi > L$) near $T_c$ it is $\tau \propto L^z$.
- The value of $z$ depends on the algorithm and for the 2D Ising model simulated using Metropolis algorithm $z \approx 2.17$.

# The Wolff algorithm

- Near $T_c$ the CPU time needed to simulate one correlation time scales as $\tau_{CPU} \propto L^{d+z}$.
- For the Metropolis algorithm with single spin flip dynamics in 2D $\tau_{CPU} \propto L^{2+2.17}$.
- The probability near $T_c \approx 2.269J$ for a spin flip of a spin surrounded by 4 spins with same direction in 2D is $e^{-8J/T_c} \approx 0.0294\ldots$.
- Solution is a cluster-flipping algorithm which allows for non-local moves.
- Consider an example of cluster flipping from state $\mu$ to state $\nu$:



$$\begin{array}{ccccc} + & + & + & - & - \\ - & + & + & - & - \\ - & + & + & + & + \\ + & + & + & + & - \end{array} \quad \begin{array}{c} \longrightarrow \\ \longleftarrow \end{array} \quad \begin{array}{ccccc} + & + & + & - & - \\ - & - & - & - & - \\ - & - & - & - & + \\ + & + & + & + & - \end{array}$$

# The Wolff algorithm

- Number of broken or released bonds in going from $\mu$ to $\nu$:

$$\begin{aligned} m &= B(\mu \to \nu) = \{\text{number of broken } {+}{+} \text{ bonds}\} = 6 \\ n &= \tilde{B}(\mu \to \nu) = \{\text{number of released } {+}{-} \text{ bonds}\} = 4 \end{aligned}$$

- Number of broken or released bonds in going from $\nu$ to $\mu$:

$$\begin{aligned} B(\nu \to \mu) &= \{\text{number of broken } {-}{-} \text{ bonds}\} = \tilde{B}(\mu \to \nu) \\ \tilde{B}(\nu \to \mu) &= \{\text{number released } {+}{-} \text{ bonds}\} = B(\mu \to \nu) \end{aligned}$$

- The energy difference between the old and the new state is

$$\Delta E = E_\nu - E_\mu = 2JB(\mu \to \nu) - 2J\tilde{B}(\mu \to \nu) = 2J(m - n)$$

- The transition probabilities must satisfy detailed balance:

$$\frac{w(\mu \to \nu)}{w(\nu \to \mu)} = e^{-2\beta J(m-n)} \tag{53}$$

# The Wolff algorithm

- $w\left(\mu \rightarrow \nu\right)$ is a product of selection probability $g\left(\mu \rightarrow \nu\right)$ and acceptance probability $A\left(\mu \rightarrow \nu\right)$:

$$\frac{w(\mu \rightarrow \nu)}{w(\nu \rightarrow \mu)} = \frac{g(\mu \rightarrow \nu)\, A(\mu \rightarrow \nu)}{g(\nu \rightarrow \mu)\, A(\nu \rightarrow \mu)} \tag{54}$$

- Choose a seed spin at random and add successively neighbors with the same orientation to the cluster with probability $P_{add}$.
- There are $b$ bonds added to the cluster with probability $(P_{add})^b$ in the forward as well as in the reverse move.
- There are $m$ bonds broken (or not added to the cluster) in the forward move with probability $(1 - P_{add})^m$ and $n$ broken bonds in the reverse move with probability $(1 - P_{add})^n$. Thus the selection probabilities obey

$$\frac{g(\mu \rightarrow \nu)}{g(\nu \rightarrow \mu)} = \frac{(1 - P_{add})^m}{(1 - P_{add})^n} \cdot \frac{(P_{add})^b}{(P_{add})^b} \tag{55}$$

# The Wolff algorithm

- The condition for the acceptance probabilities is

$$\frac{w(\mu \to \nu)}{w(\nu \to \mu)} = \frac{A(\mu \to \nu)}{A(\nu \to \mu)} \cdot (1 - P_{add})^{m-n} = e^{-2\beta J(m-n)} \quad (56)$$

$$\Rightarrow \frac{A(\mu \to \nu)}{A(\nu \to \mu)} = \left[(1 - P_{add}) e^{2\beta J}\right]^{n-m} \quad (57)$$

- The optimal choice:

$$P_{add} = 1 - e^{-2\beta J} \quad \Rightarrow \quad \frac{A(\mu \to \nu)}{A(\nu \to \mu)} = 1 \quad (58)$$

  The optimal choice $A(\mu \to \nu) = A(\nu \to \mu) = 1$ satisfies detailed balance and guarantees that every move is accepted.

- The dynamic exponent of the 2D Ising model for the Wolff algorithm is $z \approx 0.25$ (Metropolis algorithm: $z \approx 2.17$).
- Wolff outperforms Metropolis near the critical point.

# Remark on Metropolis algorithm

- In the Metropolis algorithm the selection probabilities $g(\mu \to \nu)$ for all possible states $\nu$ are equal.
- With single-spin-flip dynamics there are then $N$ different spins in a system of $N$ spins that we could flip.
- Hence there are $N$ possible states $\nu$ and thus $g(\mu \to \nu) = g(\nu \to \mu) = \frac{1}{N}$.

# Pseudocode of the Wolff algorithm

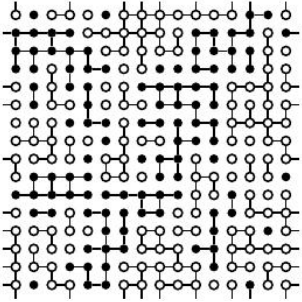
```
i=1+(int) (N*ran3(&seed)); //random seed spin
stack[0]=i;
pointer=1;
old_s=spin[i];
new_s=-spin[i];
spin[i]=new_s; //flip seed spin
while(pointer!=0)
 {i=stack[--pointer]; //go to next unchecked spin
  for (m=1;m<=4;m++) //go over 4 neighbors of spin i
   {j=neighb[i][m];
    if (spin[j]==old_s) //spin has the sign like seed
     {if (ran3(&seed)<prob) //prob=1.0-exp(-2.0/T);
       {stack[pointer++]=j; //store index of new spin
        spin[j]=new_s; //flip neighbor
       }
     }
   }
 }
```

## Improved estimator for the susceptibility

- We show that the susceptibility is related to the mean size $\langle n \rangle$ of the clusters flipped by the Wolff algorithm

$$\chi = \beta \langle n \rangle \tag{59}$$

- For the proof consider a modified Wolff algorithm. Connect every pair of neighbouring spins, which are pointing in the same direction, with a bond with probability $P_{add}$. The whole lattice is divided into correct Wolff clusters.



# Improved estimator for the susceptibility

- Choose a seed spin from the lattice at random, and flip the cluster to which it belongs.
- The total magnetization $M$ of the lattice is $M = \sum_i S_i n_i$, where $n_i$ is the size and $S_i$ the spin value of the $i$th cluster.
- Mean square magnetization is

$$\begin{aligned} \left\langle M^2 \right\rangle &= \left\langle \sum_i S_i n_i \sum_j S_j n_j \right\rangle \\ &= \left\langle \sum_{i \neq j} S_i S_j n_i n_j \right\rangle + \left\langle \sum_i S_i^2 n_i^2 \right\rangle = \left\langle \sum_i n_i^2 \right\rangle , \end{aligned} \tag{60}$$

  since $S_i S_j = \pm 1$ are equally likely and $S_i^2 = 1$.
- The magnetization per spin $\left\langle m^2 \right\rangle = \frac{1}{N^2} \left\langle \sum_i n_i^2 \right\rangle$

# Improved estimator for the susceptibility

- The probability to flip a particular cluster $i$ is

$$p_i = \frac{n_i}{N}, \tag{61}$$

  since the probability to choose a seed from cluster $i$ is proportional to its size $n_i$.

- The average size of a flipped cluster is

$$\langle n \rangle = \left\langle \sum_i p_i n_i \right\rangle = \frac{1}{N} \left\langle \sum_i n_i^2 \right\rangle = N \left\langle m^2 \right\rangle = \frac{\chi}{\beta}, \tag{62}$$

  the last equality is valid only for $T \geq T_c$ where $\langle m \rangle = 0$.

- The statistical error in the cluster definition of the $\chi$ is smaller than that obtained using the fluctuations of $m$.

# The Swendsen-Wang algorithm

- Is similar to Wolff algorithm. However, unlike Wolff algorithm, Swendsen-Wang algorithm is a multi-cluster-flipping algorithm.
- Pseudo code:
    - Connect every pair of neighboring spins, which are pointing in the same direction, with a bond with probability $P_{add} = 1 - e^{-2\beta J}$.
    - Identify all connected clusters using the Hoshen-Kopelman algorithm.
    - For each cluster in turn, decide independently with probability $\frac{1}{2}$ whether to flip it or not.
- Swendsen-Wang algorithm has a similar performance as Wolff algorithm ($z \approx 0.25$ in 2D).
- Wolff algorithm is always two times faster, since only half of the clusters generated are flipped in the Swendsen-Wang algorithm.

# The Swendsen-Wang algorithm

- Wolff algorithm preferentially flips larger clusters because the probability to select a seed from a cluster of size $n_i$ is $p_i = \frac{n_i}{N}$. The Swendsen-Wang algorithm treats all clusters equally, regardless of their size, and therefore wastes much effort on small clusters, which do not contribute to the critical slowing down.
- There exists also an improved estimator for the susceptibility.

# Cluster algorithm for $q$ state Potts model

- Wolff algorithm for Potts model is similar to that for Ising model
  - Choose a seed spin at random from the lattice.
  - Add neighbor spins with probability $P_{add} = 1 - e^{-\beta J}$ instead of $P_{add} = 1 - e^{-2\beta J}$ as in the case of the Ising model (a broken bond costs now $J$ instead of $2J$).
  - Choose at random a new value for the spins in the cluster, different from the present value, and set all the spins to that new value.
- The Swendsen-Wang algorithm is analog to the case of the Ising model, however, a new value of all spins in the cluster is chosen randomly from $\{1, ..., q\}$.

# Cluster algorithm for continuous spin model

- Similar to Wolff algorithm for the Ising model.
- Choose a random direction $\mathbf{n}$ with $|\mathbf{n}| = 1$.
- Choose a seed spin $i$ with $\mathbf{S}_i \cdot \mathbf{n} > 0$ and add bonds to neighbor spin $j$ if $\mathbf{S}_j \cdot \mathbf{n} > 0$ with probability

$$P_{add}(S_i, S_j) = 1 - \exp\left[-2\beta(\mathbf{S}_i \cdot \mathbf{n})(\mathbf{S}_j \cdot \mathbf{n})\right] \tag{63}$$

- Flip cluster by reflecting all the spins in the plane perpendicular to $\mathbf{n}$.

# The single histogram method

- Method allows to extrapolate simulation results obtained at temperature $T_0$ to nearby $T \neq T_0$.
- Canonical distribution at $\beta = 1/T$ can be easily related to the distribution at other $\beta_0 = 1/T_0$ via

$$\begin{aligned} P_\beta(S) &\propto e^{-\beta H(S)} = e^{-(\beta-\beta_0)H(S)} e^{-\beta_0 H(S)} \\ &= C e^{-(\beta-\beta_0)H(S)} P_{\beta_0}(S), \end{aligned} \tag{64}$$

  with an so far undetermined constant $C$.
- The expectation value of $\mathcal{O}$ at temperature $\beta = 1/T$ is

$$\begin{aligned} \langle \mathcal{O} \rangle_\beta &= \frac{\sum_{\{S\}} \mathcal{O}(S) P_\beta(S)}{\mathcal{Z}_\beta} \\ &= C \frac{\mathcal{Z}_{\beta_0}}{\mathcal{Z}_\beta} \frac{\sum_{\{S\}} \mathcal{O}(S) e^{-(\beta-\beta_0)H(S)} P_{\beta_0}(S)}{\mathcal{Z}_{\beta_0}} \\ &= C \frac{\mathcal{Z}_{\beta_0}}{\mathcal{Z}_\beta} \left\langle \mathcal{O} e^{-(\beta-\beta_0)H} \right\rangle_{\beta_0} \end{aligned} \tag{65}$$

# The single histogram method

- Setting $\mathcal{O} = 1$ we determine the constant

$$C = \frac{\mathcal{Z}_\beta}{\mathcal{Z}_{\beta_0}} \frac{1}{\left\langle e^{-(\beta-\beta_0)H} \right\rangle_{\beta_0}} \tag{66}$$

  and obtain finally the fundamental equation of the histogram method as

$$\langle \mathcal{O} \rangle_\beta = \frac{\left\langle \mathcal{O} e^{-(\beta-\beta_0)H} \right\rangle_{\beta_0}}{\left\langle e^{-(\beta-\beta_0)H} \right\rangle_{\beta_0}} \tag{67}$$

- During a MC simulation at $\beta_0 = 1/T_0$ we perform $M$ measurements of $\mathcal{O}(S_t)$ at configurations $S_t$ and the estimator of $\mathcal{O}$ at $\beta_0$ is

$$\langle \mathcal{O} \rangle_{T_0} = \lim_{M\to\infty} \tfrac{1}{M} \textstyle\sum_{t=1}^{M} \mathcal{O}(S_t). \tag{68}$$

- Therefore, the estimator of $\mathcal{O}$ at $\beta = 1/T \neq \beta_0$ using MC configurations at $\beta_0$ is

$$\langle \mathcal{O} \rangle_T = \lim_{M\to\infty} \frac{\sum_{t=1}^{M} \mathcal{O}(S_t) e^{-(\beta-\beta_0)H(S_t)}}{\sum_{t=1}^{M} e^{-(\beta-\beta_0)H(S_t)}} \tag{69}$$

# The single histogram method

- To get the mean energy $\langle E \rangle_T$ at $T$ calculate

$$\langle E \rangle_T = \frac{\sum_E E \cdot N(E) e^{-(\beta-\beta_0)E}}{\sum_E N(E) e^{-(\beta-\beta_0)E}}, \tag{70}$$

  where $N(E)$ is the histogram of energies obtained from MC simulation at $T_0$.
- Method works as long as $|\langle E \rangle_T - \langle E \rangle_{T_0}| \leq \sigma_E$, where $\sigma_E$ is the standard deviation of $N(E)$ and $\sigma_E^2 = C(T_0)/\beta_0^2$ with specific heat $C(T_0)$.

  Satisfactory extrapolation should be obtained in

$$T \in [T_0 - \Delta T, T_0 + \Delta T] \quad \text{with} \quad \Delta T = T_0/\sqrt{C(T_0)}. \tag{71}$$

- For the magnetisation we need a two dimensional histogram $N(E, M)$ and

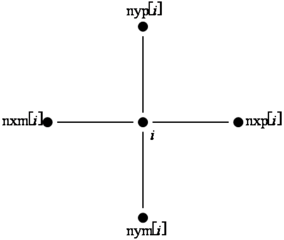
$$m = \frac{\langle M \rangle_T}{N} = \frac{\sum_{M,E} M \cdot N(E, M) e^{-(\beta-\beta_0)E}}{\sum_{M,E} N(E, M) e^{-(\beta-\beta_0)E}} \tag{72}$$

# Hints for the Metropolis simulation of the 2D Ising model

- Set $J = 1$ such that $T$ in units $Jk_B$. Set number of spins to $N = L^2$.

- Construct a neighbor list using periodic boundary conditions.
  4 array of size $N$: `nxm[N]`, `nxp[N]`, `nym[N]`, `nyp[N]`.
  For example:

```
for (i=1;i<=N;i++)
{nxp[i]=i+1
  if (i%L==0) {nxp[i]=i+1-L;}
}
```



- Make a table for the acceptance probabilities for $\Delta E > 0$.
  $\Delta E = 2 S_i h_i \in \{4, 8\}$ with
  $h_i = S_{\texttt{nxm[i]}} + S_{\texttt{nxp[i]}} + S_{\texttt{nym[i]}} + S_{\texttt{nyp[i]}}$
  $\Rightarrow e^{-\beta \Delta E} \in \left\{ e^{-4\beta}, e^{-8\beta} \right\}$

```
for (i=0;i<=4;i++) {prob[i]=exp(-2.0*((double) j)/T);}
```

# Hints for the Metropolis simulation of the 2D Ising model

- Use an array for the local fields:

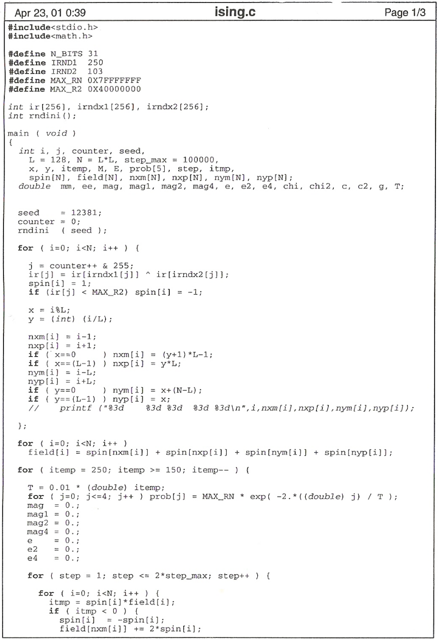
  ```
  for (i=0;i<N;i++)
  {field[i]=spin[nxm[i]]+spin[nxp[i]]
  +spin[nym[i]]+spin[nyp[i]];}
  ```

  and update `field[i]` if a spin flip took place:

  ```
  field[nxm[i]]+=2*spin[i];
  field[nxp[i]]+=2*spin[i];
  field[nym[i]]+=2*spin[i];
  field[nyp[i]]+=2*spin[i];
  ```

- Equilibrate the system first (check if energy reached a steady state) and measure afterwards the observables, like: $\langle m \rangle$, $\langle m^2 \rangle$, $\langle m^4 \rangle$, $\langle e \rangle$, $\langle e^2 \rangle$, $C(r)$.

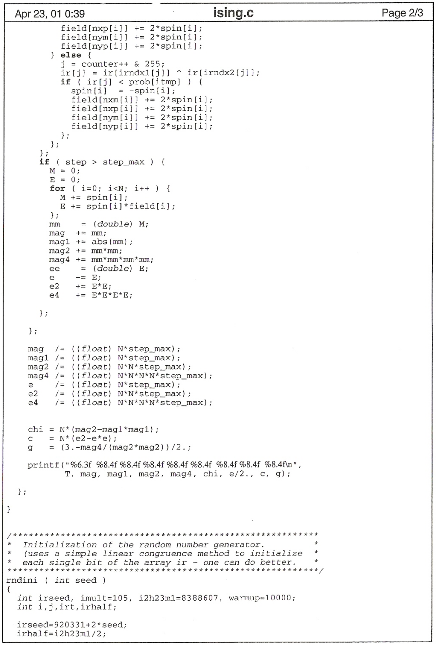
# Riegers implementation of the Metropolis algorithm



Apr 23, 01 0:39 **ising.c** Page 1/3

```
#include<stdio.h>
#include<math.h>

#define N_BITS 31
#define IRND1  250
#define IRND2  103
#define MAX_RN 0X7FFFFFFF
#define MAX_R2 0X40000000

int ir[256], irndx1[256], irndx2[256];
int rndini();

main ( void )
{
  int i, j, counter, seed,
    L = 128, N = L*L, step_max = 100000,
    x, y, itemp, M, E, prob[5], step, itmp,
    spin[N], field[N], nxm[N], nxp[N], nym[N], nyp[N];
  double  mm, ee, mag, mag1, mag2, mag4, e, e2, e4, chi, chi2, c, c2, g, T;


  seed    = 12381;
  counter = 0;
  rndini  ( seed );

  for ( i=0; i<N; i++ ) {

    j = counter++ & 255;
    ir[j] = ir[irndx1[j]] ^ ir[irndx2[j]];
    spin[i] = 1;
    if (ir[j] < MAX_R2) spin[i] = -1;

    x = i%L;
    y = (int) (i/L);

    nxm[i] = i-1;
    nxp[i] = i+1;
    if ( x==0     ) nxm[i] = (y+1)*L-1;
    if ( x==(L-1) ) nxp[i] = y*L;
    nym[i] = i-L;
    nyp[i] = i+L;
    if ( y==0     ) nym[i] = x+(N-L);
    if ( y==(L-1) ) nyp[i] = x;
    //    printf ("%3d    %3d %3d  %3d %3d\n",i,nxm[i],nxp[i],nym[i],nyp[i]);

  };

  for ( i=0; i<N; i++ )
    field[i] = spin[nxm[i]] + spin[nxp[i]] + spin[nym[i]] + spin[nyp[i]];

  for ( itemp = 250; itemp >= 150; itemp-- ) {

    T = 0.01 * (double) itemp;
    for ( j=0; j<=4; j++ ) prob[j] = MAX_RN * exp( -2.*((double) j) / T );
    mag  = 0.;
    mag1 = 0.;
    mag2 = 0.;
    mag4 = 0.;
    e    = 0.;
    e2   = 0.;
    e4   = 0.;

    for ( step = 1; step <= 2*step_max; step++ ) {

      for ( i=0; i<N; i++ ) {
        itmp = spin[i]*field[i];
        if ( itmp < 0 ) {
          spin[i]  = -spin[i];
          field[nxm[i]] += 2*spin[i];
```

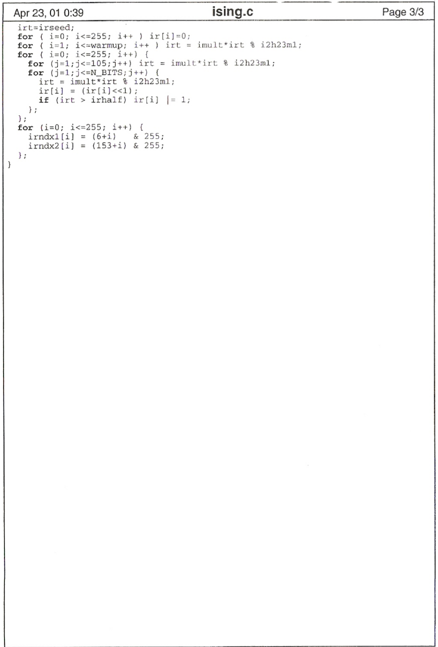
# Riegers implementation of the Metropolis algorithm



Apr 23, 01 0:39 **ising.c** Page 2/3

```
          field[nxp[i]] += 2*spin[i];
          field[nym[i]] += 2*spin[i];
          field[nyp[i]] += 2*spin[i];
        } else {
          j = counter++ & 255;
          ir[j] = ir[irndx1[j]] ^ ir[irndx2[j]];
          if ( ir[j] < prob[itmp] ) {
            spin[i]  = -spin[i];
            field[nxm[i]] += 2*spin[i];
            field[nxp[i]] += 2*spin[i];
            field[nym[i]] += 2*spin[i];
            field[nyp[i]] += 2*spin[i];
          };
        };
      };
      if ( step > step_max ) {
        M = 0;
        E = 0;
        for ( i=0; i<N; i++ ) {
          M += spin[i];
          E += spin[i]*field[i];
        };
        mm    = (double) M;
        mag  += mm;
        mag1 += abs(mm);
        mag2 += mm*mm;
        mag4 += mm*mm*mm*mm;
        ee    = (double) E;
        e    -= E;
        e2   += E*E;
        e4   += E*E*E*E;

      };

    };

    mag  /= ((float) N*step_max);
    mag1 /= ((float) N*step_max);
    mag2 /= ((float) N*N*step_max);
    mag4 /= ((float) N*N*N*N*step_max);
    e    /= ((float) N*step_max);
    e2   /= ((float) N*N*step_max);
    e4   /= ((float) N*N*N*N*step_max);


    chi = N*(mag2-mag1*mag1);
    c   = N*(e2-e*e);
    g   = (3.-mag4/(mag2*mag2))/2.;

    printf("%6.3f %8.4f %8.4f %8.4f %8.4f %8.4f %8.4f %8.4f %8.4f\n",
           T, mag, mag1, mag2, mag4, chi, e/2., c, g);

  };

}


/************************************************************
*  Initialization of the random number generator.           *
*  (uses a simple linear congruence method to initialize    *
*  each single bit of the array ir - one can do better.     *
*************************************************************/
rndini ( int seed )
{
  int irseed, imult=105, i2h23m1=8388607, warmup=10000;
  int i,j,irt,irhalf;

  irseed=920331+2*seed;
  irhalf=i2h23m1/2;
```

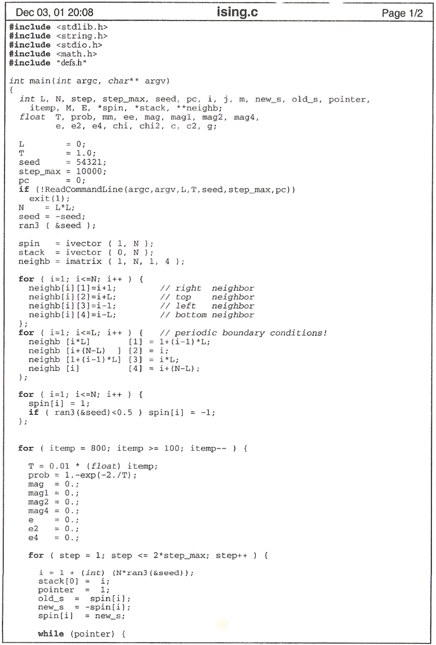
# Riegers implementation of the Metropolis algorithm



Apr 23, 01 0:39 **ising.c** Page 3/3

```
  irt=irseed;
  for ( i=0; i<=255; i++ ) ir[i]=0;
  for ( i=1; i<=warmup; i++ ) irt = imult*irt % i2h23m1;
  for ( i=0; i<=255; i++) {
    for (j=1;j<=105;j++) irt = imult*irt % i2h23m1;
    for (j=1;j<=N_BITS;j++) {
      irt = imult*irt % i2h23m1;
      ir[i] = (ir[i]<<1);
      if (irt > irhalf) ir[i] |= 1;
    };
  };
  for (i=0; i<=255; i++) {
    irndx1[i] = (6+i)   & 255;
    irndx2[i] = (153+i) & 255;
  };
}
```

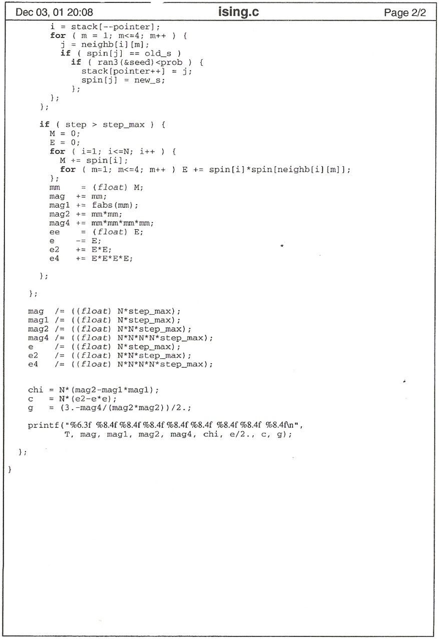
# Riegers implementation of the Wolff algorithm



Dec 03, 01 20:08 **ising.c** Page 1/2

```c
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <math.h>
#include "defs.h"

int main(int argc, char** argv)
{
  int L, N, step, step_max, seed, pc, i, j, m, new_s, old_s, pointer,
    itemp, M, E, *spin, *stack, **neighb;
  float  T, prob, mm, ee, mag, mag1, mag2, mag4,
         e, e2, e4, chi, chi2, c, c2, g;

  L        = 0;
  T        = 1.0;
  seed     = 54321;
  step_max = 10000;
  pc       = 0;
  if (!ReadCommandLine(argc,argv,L,T,seed,step_max,pc))
    exit(1);
  N    = L*L;
  seed = -seed;
  ran3 ( &seed );

  spin   = ivector ( 1, N );
  stack  = ivector ( 0, N );
  neighb = imatrix ( 1, N, 1, 4 );

  for ( i=1; i<=N; i++ ) {
    neighb[i][1]=i+1;          // right  neighbor
    neighb[i][2]=i+L;          // top    neighbor
    neighb[i][3]=i-1;          // left   neighbor
    neighb[i][4]=i-L;          // bottom neighbor
  };
  for ( i=1; i<=L; i++ ) {    // periodic boundary conditions!
    neighb [i*L]         [1] = 1+(i-1)*L;
    neighb [i+(N-L)    ] [2] = i;
    neighb [1+(i-1)*L]   [3] = i*L;
    neighb [i]           [4] = i+(N-L);
  };

  for ( i=1; i<=N; i++ ) {
    spin[i] = 1;
    if ( ran3(&seed)<0.5 ) spin[i] = -1;
  };


  for ( itemp = 800; itemp >= 100; itemp-- ) {

    T = 0.01 * (float) itemp;
    prob = 1.-exp(-2./T);
    mag  = 0.;
    mag1 = 0.;
    mag2 = 0.;
    mag4 = 0.;
    e    = 0.;
    e2   = 0.;
    e4   = 0.;

    for ( step = 1; step <= 2*step_max; step++ ) {

      i = 1 + (int) (N*ran3(&seed));
      stack[0] =  i;
      pointer  =  1;
      old_s   =  spin[i];
      new_s   = -spin[i];
      spin[i]  = new_s;

      while (pointer) {
```

# Riegers implementation of the Wolff algorithm



Dec 03, 01 20:08 **ising.c** Page 2/2

```
        i = stack[--pointer];
        for ( m = 1; m<=4; m++ ) {
          j = neighb[i][m];
          if ( spin[j] == old_s )
            if ( ran3(&seed)<prob ) {
              stack[pointer++] = j;
              spin[j] = new_s;
            };
        };
      };

      if ( step > step_max ) {
        M = 0;
        E = 0;
        for ( i=1; i<=N; i++ ) {
          M += spin[i];
          for ( m=1; m<=4; m++ ) E += spin[i]*spin[neighb[i][m]];
        };
        mm     = (float) M;
        mag   += mm;
        mag1  += fabs(mm);
        mag2  += mm*mm;
        mag4  += mm*mm*mm*mm;
        ee     = (float) E;
        e     -= E;
        e2    += E*E;
        e4    += E*E*E*E;

      };

    };

    mag  /= ((float) N*step_max);
    mag1 /= ((float) N*step_max);
    mag2 /= ((float) N*N*step_max);
    mag4 /= ((float) N*N*N*N*step_max);
    e    /= ((float) N*step_max);
    e2   /= ((float) N*N*step_max);
    e4   /= ((float) N*N*N*N*step_max);


    chi = N*(mag2-mag1*mag1);
    c   = N*(e2-e*e);
    g   = (3.-mag4/(mag2*mag2))/2.;

    printf("%6.3f  %8.4f %8.4f %8.4f %8.4f %8.4f  %8.4f %8.4f  %8.4f\n",
           T, mag, mag1, mag2, mag4, chi, e/2., c, g);

  };

}
```

# Literature

- M. E. J. Newman, G. T. Barkema: *Monte Carlo Methods in Statistical Physics*
- David P. Landau, Kurt Binder: *A Guide to Monte Carlo Simulations in Statistical Physics*
- M. P. Allen, D. J. Tildesley: *Computer Simulation of Liquids*
- Skript by J. Kierfeld:

  `http://t1.physik.tu-dortmund.de/files/kierfeld/teaching/LectureNotes/kierfeld_CompPhys.pdf`