

A model project for reproducible papers: critical temperature for the Ising model on a square lattice

M. Dolfi, J. Gukelberger, A. Hehn, J. Imriška, K. Pakrouski, T. F. Rønnow, M. Troyer, I. Zintchenko*
Theoretische Physik, ETH Zurich, 8093 Zurich, Switzerland and
Theoretische Physik, ETH Zurich, 8093 Zurich, Switzerland

In this paper we present a simple, yet typical simulation in statistical physics, consisting of large scale Monte Carlo simulations followed by an involved statistical analysis of the results. The purpose is to provide an example publication to explore tools for writing reproducible papers. The simulation estimates the critical temperature where the Ising model on the square lattice becomes magnetic to be $T_c/J = 2.26934(6)$ using a finite size scaling analysis of the crossing points of Binder cumulants. In an Appendix and as supplementary material we provide the output of the simulations, the source and binaries of the codes, and Python scripts used to generate all figures and results.

I. INTRODUCTION

The principle that scientific publications have to be reproducible is a cornerstone of modern science. A theoretical paper typically contains all the steps required to follow the arguments and arrive at the final result. Experimental papers usually go to great lengths in describing the most important details and keep track of workflow in sacrosanct lab notebooks. In computational science the goal of reproducibility is harder to achieve. Reasons are the complexity of computer systems, codes, and analysis procedures and the absence of well established community guidelines and wide-spread tools for reproducibility.

To focus the discussion on comparison of procedures and tools, we picked one simple, yet representative example of a computer simulation in statistical physics. Calculating the critical temperature of the classical Ising model. The current manuscript already contains sufficient details, codes, and scripts to reproduce all the presented numerical results and figures. However, reaching this level of reproducibility required efforts that went far beyond simply obtaining the results. Our goal for future work is to use this paper as an example for a discussion on how reproducibility can best be achieved.

The total computation time for a small system size is in the order of several hours on a single core producing several MB of raw output data. Including large system sizes increases the accuracy of the results, but also the runtime and the amount of data produced and one might need to use large computer clusters. This allows to explore the scalability of tools for reproducibility in computational science.

II. MODEL

The Ising model dates back to 1920 when it was proposed by Wilhelm Lenz as a mathematical model for fer-

romagnetism and first analytically solved by his student Ernst Ising in one dimension [1]. We will consider the two-dimensional Ising model on a square lattice of size $L \times L$ with periodic boundary conditions. It is described by the Hamiltonian

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j. \quad (1)$$

where the sum is over nearest neighbours, $\sigma_i \in \{-1, +1\}$ is the spin on site i and J is the coupling strength. In the following we will focus on the ferromagnetic model $J > 0$.

Below a critical temperature T_c the model shows magnetic ordering, whereas above T_c it remains unordered. For the square lattice without an external field T_c is known analytically [2]

$$T_c = \frac{2J}{\ln(1 + \sqrt{2})} \approx 2.269185. \quad (2)$$

and provides a benchmark for our results.

III. METHODS

A Monte Carlo simulation using Wolff cluster updates [3] is used to construct new system configurations, employing the MT19937 Mersenne Twister pseudo random number generator [4]. For each parameter set more than 1280000 measurements are performed after discarding 10% additional Wolff updates for thermalization; error estimates are done with binning analysis.

The critical temperature can be roughly estimated from the connected susceptibility

$$\langle \chi \rangle_\beta = \beta L^2 \left(\langle m^2 \rangle - \langle |m| \rangle^2 \right), \quad (3)$$

where $\beta = \frac{1}{k_B T}$ and k_B is the Boltzmann constant. The average is taken over different configurations. $\langle \chi \rangle_\beta$ has a peak around T_c [5], which gives a first rough estimate.

The Binder cumulant

$$U_2 = \langle m^2 \rangle / \langle |m| \rangle^2 \quad (4)$$

* All authors contributed equally and are listed in alphabetical order.

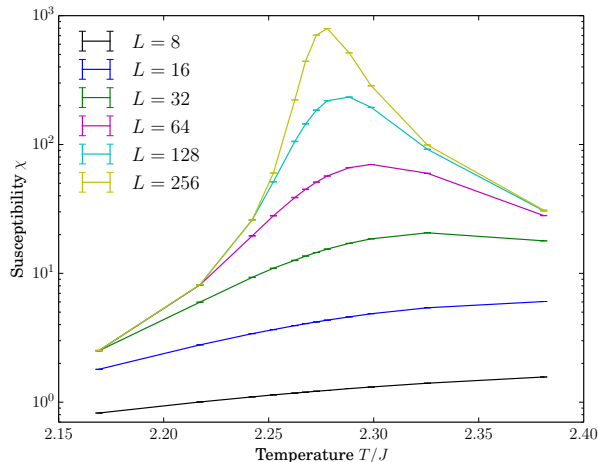


FIG. 1. Temperature dependence of the susceptibility for different system sizes L .

provides a more precise technique to extract the critical temperature. For different system sizes the temperature dependence of U_2 is expected to cross at different points. The crossing points can be shown to follow

$$T_c^*(L) = T_c + AL^{-1/\nu} \quad (5)$$

where the critical exponent $\nu = 1$ in two dimensions [6]. To extract T_c^* for an infinite system we now fit the positions of the crossing points between system sizes L and $L/2$, respectively, using a least-squares fit weighted with the size of the error bars at each system size, minimizing $\chi^2 = \sum_L (T_L - T_L^i)^2 / \zeta_L^2$, where T_L is the crossing point temperature for the systems with sizes L and $L/2$, ζ_L is its standard error, and T_L^i is the value of the linear fitting function. The fit is then extrapolated to the limit $1/L \rightarrow 0$. A Jackknife analysis with at least 78 bins is used to estimate the errors of the Binder cumulants and their crossings.

IV. RESULTS AND DISCUSSION

Figure 1 shows the connected susceptibility defined in Eq. 3 as a function of temperature for system sizes $L = 8, 16, \dots, 256$. It peaks around $T \approx 2.275J$, which gives a first estimate for the critical temperature.

Fig. 2 shows the intersections of Binder cumulants U_2 defined in Eq. 4 with cubic interpolation between all the temperature points within each system size. For each pair of consecutive system sizes we identify the temperature at the crossing point. This temperature is plotted in Fig. 3 as a function of the larger system size in each pair.

The scaling law of the Binder cumulant

$$U_2 = \mathcal{F}(L^{1/\nu}(T - T_c)/T_c), \quad (6)$$

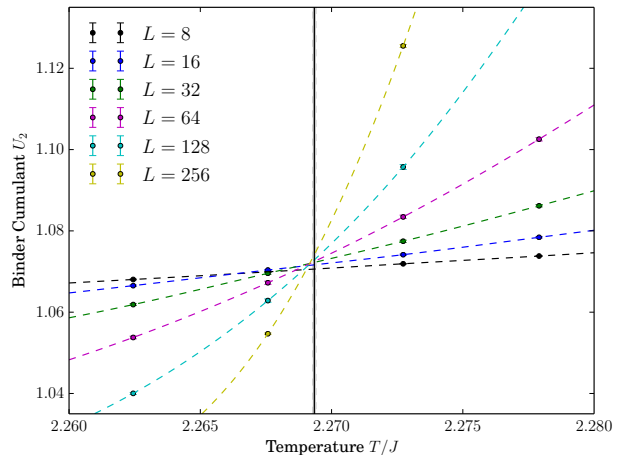


FIG. 2. Temperature dependence of the Binder cumulants for different system sizes. Vertical line and the grey area around it indicate our estimate for critical temperature and for the error respectively. Dashed lines are fits to a cubic polynomial.

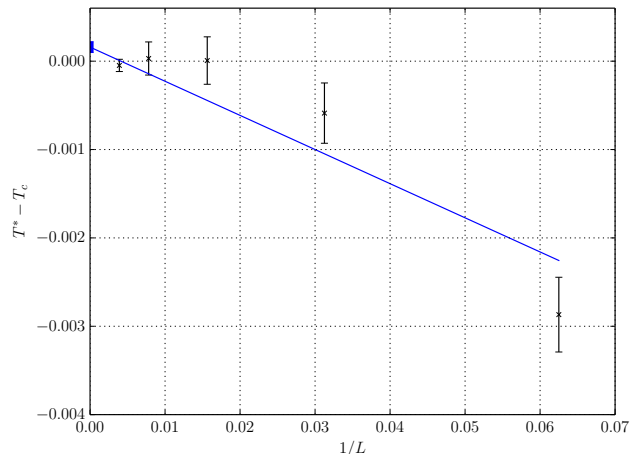


FIG. 3. Finite size scaling of the Binder cumulant crossing points. The dependence can be shown to satisfy $T_c^*(L) = T_c + AL^{-1/\nu}$ and $T_c^*/J = 2.26934(6)$ is extracted from a weighted least-squares fit as described in the text.

where \mathcal{F} is a universal function, provides a check for the estimate of the critical temperature T_c^* . In Fig. 4 we show a data collapse, plotting U_2 for different system sizes versus $L^{1/\nu}(T - T_c)/T_c$. According to finite size scaling the curves for different sizes L should be on top of one another in the vicinity of the critical point, as we can indeed see when using our estimate T_c^* . The reader is invited to use our scripts to test different values of T_c^* and ν in the data collapse. Further instructions are provided in the Appendix.

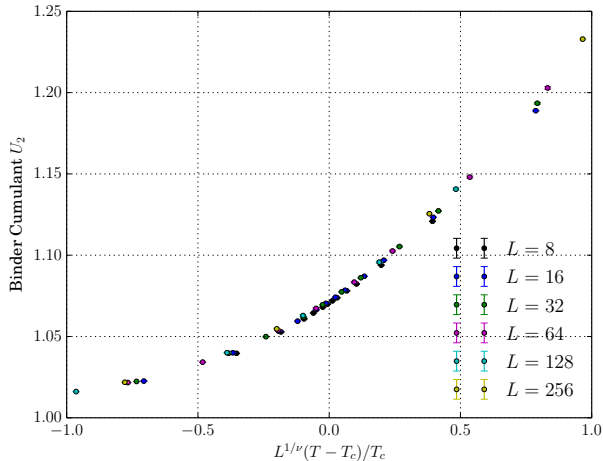


FIG. 4. Data collapse of Binder cumulant. T_c^* is obtained from Fig. 3.

V. CONCLUSION

Our final result of $T_c^*/J = 2.26934(6)$ is consistent with the analytically known value, which is a good check for the correctness of the analysis in this demonstration paper. Instructions to reproduce all figures and results are provided in Appendix A.

As one might expect, the effort involved in making this paper reproducible was much larger than the effort for creating the first results. An important lesson learned from this project is that it will be important to develop best practices and better tools to more easily make simulations reproducible. This prototypical simulation can be a good non-trivial but not too complex example as a test project to explore the capabilities of various tools.

Appendix A: Instructions for reproducing the results

This appendix contains detailed instructions for reproducing the results, both the raw simulation output in section A 1 and the analysis of the data in section A 2. Finally, section A 3 suggests possibilities for further simulations and analysis of the data.

All the analysis scripts, paper and code sources are available as supplementary material to this paper and can also be downloaded from **to be inserted**.

1. Obtaining the raw data

The output of the simulations can either be downloaded from our archive, as discussed in section A 1 a, or it can be reproduced by running the simulations. For a big project readers would typically download the data

and only redo the evaluation part. To reproduce the data one needs to obtain the simulation codes, either as pre-compiled binary or by building them from source (see section A 1 b) and then perform the simulations (see section A 1 c).

a. Downloading the raw data

The raw simulation output is available from our archive at <http://archive.comp-phys.org/phys.ethz.ch/provenance/data/20131014.tar>. It should be extracted into the `data/` directory of the source tree, to be accessible to the evaluation scripts. A convenience script `getdata.sh` is provided to download and extract the data on Linux, Unix, and MacOS X. On Windows platforms we recommend to manually download and extract the data.

b. Obtaining the code

There are three versions of the code that can be used interchangeably to create the output data:

- `ising_single`: a single-threaded simulation code
- `ising_threaded`: a multi-threaded simulation code. It spawns one thread per set of input parameters, up to the maximum number of cores of the machine.
- `ising_mpi`: an MPI version of the code, using distributed memory parallelism.

For convenience we provide precompiled binaries of `ising_single` and `ising_threaded` for Linux (**version to be inserted**), MacOS X version 10.6 or higher (64-bit), and Windows 8 (32-bit) at **to be inserted**. An MPI version `ising_mpi`, compiled for OpenMPI 1.6.5 is provided only for MacOS X and Linux. Further precompiled versions may be obtained from the authors. These binaries should be placed in the `bin/` directory.

Alternatively the binaries can be built from source. This requires the installation of the ALPS libraries [7] for parallel Monte Carlo simulations. After setting the environment variable `ALPS_ROOT` to the location of the ALPS installation, the simulation codes can be built using the CMake files provided in the `src/` directory. The root of the source tree should be specified as the `CMAKE_INSTALL_PREFIX/` to install the executables into the `bin/` directory.

A convenience script `buildall.sh` is provided for Unix-based operating systems. On Windows we recommend the precompiled binaries.

c. Running the code

The simulation is started by executing one of the versions (`ising_single`, `ising_threaded` or `ising_mpi`) with `as` arguments.

All numerical data can be recreated by running `python simulate_all.py` (this will take a few hundred CPU hours. The simulations up to cluster size $L = 64$ only require several CPU-hours using and may be performed by `python simulate_small.py`). By default these scripts will use the multi-threaded version. The single-threaded or MPI version can be selected by changing the variable `executable` in these scripts.

2. Data analysis

Analyzing the raw data calculates the estimate for T_c and produces the figures for the paper and text files containing the information in the figures (for easy access to the numerical values of the data shown).

The data analysis requires Python 2.7 and the following packages:

- PyTables [8]
- numpy
- scipy
- matplotlib

The Python scripts `susceptibility.py`, `binder_cumulant.py` and `binder_collapse.py` in the `figures` directory create all figures needed for the paper in separate subdirectories. The estimate of the critical temperature T_c^* is printed to the standard output by `binder_cumulant.py` and has to be manually copied into `binder_collapse.py` where it is used as the input; this is not needed if you downloaded the raw data from our archive – the corresponding value of T_c^* is already set in `binder_collapse.py`.

For each figure a corresponding directory is created, containing a PDF file and further information.

3. Suggestions for further analysis and modifications to the simulations

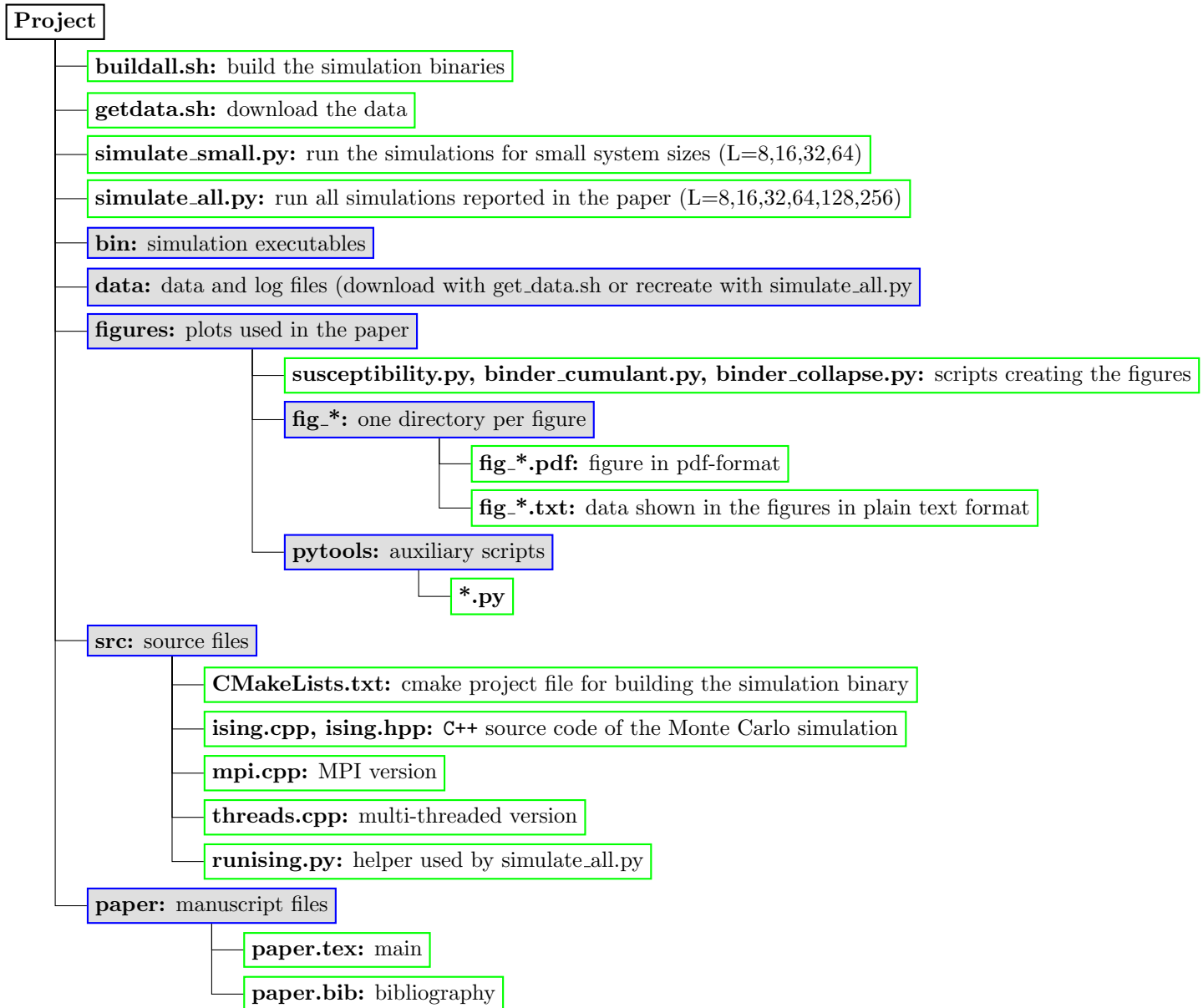
a. Modifying simulation parameters

The script `simulate_all.py` can easily be customised to calculate different system sizes, improve statistics by increasing the number of sweeps, or extending the temperature range.

b. Data collapse

It is instructive to explore the data collapse plot in figure 4. By changing the parameters T_c (and ν) one can explore the range of values which there is still good data collapse. That range gives an indication of the error on T_c . Values of the critical exponent ν and the critical temperature T_c^* can be changed in the script `binder_collapse.py` to explore the data collapse.

Appendix B: Overview of the source tree



-
- [1] E. Ising, *Zeitschrift für Physik A Hadrons and Nuclei* **31**, 253 (1925), 10.1007/BF02980577.
 [2] L. Onsager, *Phys. Rev.* **65**, 117 (1944).
 [3] U. Wolff, *Phys. Rev. Lett.* **62**, 361 (1989).
 [4] M. Matsumoto and T. Nishimura, *ACM Trans. Model. Comput. Simul.* **8**, 3 (1998).
 [5] K. Binder, *Zeitschrift für Physik B Condensed Matter* **43**, 119 (1981), 10.1007/BF01293604.
 [6] A. Pelissetto and E. Vicari, *Physics Reports* **368**, 549 (2002).
 [7] <http://alps.comp-phys.org>.
 [8] <http://www.pytables.org>.