How To Simulate Ising Model

David Forte[1] and Sancho Simões[2]

Universidade de Coimbra, PT
[1]ressurreicao@student.dei.uc.pt, student number: 2019219749
[2]sanchosimoes@student.dei.uc.pt, student number: 2019217590

1 Introduction

To explore the Ising Model, we will focus on its original version, which considers a two-dimensional cellular world. In this setup, each cell in the grid can take on either a -1 or 1 state. The change of state for each cell is influenced probabilistically by the states of its neighboring cells. Alternatively, we can employ mean field theory, which considers the local state of the entire grid.

The primary goal of this project is to implement the Ising Model and study its behavior within the context of the two-dimensional cellular world. By simulating the interactions between spins and examining the system's response to changes in temperature, we aim to gain insights into phase transitions and the emergence of collective behavior.

The code for this project is organized into five Python files, each serving a specific purpose:

- `game_of_ice.py`: This file contains the main implementation of the Ising Model. It includes functions for initializing the grid, updating the state of each cell based on its neighbors, and simulating the system over a specified number of iterations.
- `metrics.py`: This file includes functions for calculating statistical properties of the Ising Model, such as the magnetization, energy, and correlation functions. These functions help analyze the behavior of the system and its phase transitions.
- `perturbations.py`: This file has the functions to create perturbations in the simulation. These functions help to see what happens to the model when introduced to perturbations.
- `distribution_functions.py`: This file calculates the values to be assigned to the neighbors using a distributed approach.
- `simulation_grid_search.py and simulation.py`: This file serves as the entry point of the program. It imports the necessary functions from the other files and orchestrates the simulation, allowing you to run and analyze the Ising Model by executing this single script. It utilizes libraries such as `matplotlib` or `simcx` to create visualizations of the Ising Model simulation. It also has all parameters initialization.

By leveraging these five Python files, you will have a well-organized codebase that provides modularity and ease of maintenance. Each file contributes to a specific aspect of the Ising Model implementation, facilitating a clear understanding of the code's structure and purpose.

In the following sections, we will delve into the implementation details, discuss the simulation methodology, and present

2 Installation

To run the Ising Model project, you need to ensure that the following dependencies are installed on your system:

- **Python**: Make sure you have Python installed on your machine. You can download the latest version of Python from the official website (https://www.python.org) and follow the installation instructions for your operating system.

- **simcx**: This project relies on the 'simcx' library for simulating the Ising Model. You can install 'simcx' using pip, a package installer for Python. Open a terminal or command prompt and run the following command:

  ```
  pip install simcx
  ```

- **NumPy**: NumPy is a fundamental library for scientific computing in Python. It provides support for large, multi-dimensional arrays and a collection of mathematical functions to operate on these arrays. You can install NumPy using pip:

  ```
  pip install numpy
  ```

- **matplotlib**: Matplotlib is a plotting library that allows you to create various types of visualizations. It is used in this project for visualizing the Ising Model simulation results. Install it using pip:

  ```
  pip install matplotlib
  ```

- **SciPy**: SciPy is a library that extends the functionality of NumPy and provides additional scientific computing tools. It is used in this project for calculating statistical properties of the Ising Model. Install it using pip:

  ```
  pip install scipy
  ```

Once you have installed Python and the necessary dependencies, you're ready to run the Ising Model simulation.

3 Usage

To run the Ising Model simulation, navigate to the directory containing the project files in your terminal or command prompt. Then, simply run the ‘simulation_grid_search.py‘ file using Python:

```
python simulation_grid_search.py
```

You can modify the parameters by editing the variables at the top of the main function from ‘simulation_grid_search.py‘ file to suit your needs. For example, to change the grid size to 100x100.

Once you've made your desired changes, save the file and re-run it using the ‘python‘ command. The simulation will output a series of plots and data files in the ‘results‘ directory, which you can analyze further using your preferred tools or methods.

After running and once the simulation opens you can press space to resume or pause the simulation, click in the top screen to add perturbations (both mouse buttons work), you can use your scroll to make the perturbations bigger or smaller and you can also take print screen of the simulation by clicking P.

Note: Depending on your system and the parameters you choose, the simulation may take some time to run. You may also need to adjust the visualization settings or algorithms to suit your preferences or requirements.

4 Output

The Ising Model simulation produces the following output:

- **Simulation Plot**: The simulation generates a plot that visualizes the evolution of the Ising Model over time. This plot shows the state of the grid at different iterations, highlighting the emergence of patterns and phase transitions. The plot provides a visual representation of the system's behavior under the chosen parameters.
- **Parameter Log File**: A log file is created, which contains the chosen parameter values for the simulation. This file serves as a record of the parameters used, making it easy to refer back to the specific settings for each run. The log file helps in reproducing and documenting the results obtained.
- **Printed Image**: If configured, the simulation can also generate an image file of the plot. This image file allows for easy sharing, embedding in reports, or further analysis. By default, the image file is saved in a common format such as PNG or JPEG.

The output files will be saved in the project directory or a specified output directory, depending on the configuration in the code. Make sure to check the directory where the simulation script is located to find the generated files.

You can analyze the simulation plot to observe the dynamics of the Ising Model, including the formation of domains, phase transitions, and the impact

of different parameters on the system's behavior. The parameter log file provides a record of the specific settings used for each simulation run, facilitating reproducibility and enabling comparison between different experiments.

5 Conclusion

You can use multiple parameters at the same type but only one simulation will appear, to run the next parameter you have to close the current simulation window.

In conclusion, by following the steps outlined in this guide, you now have the knowledge and tools to successfully implement and study the Ising Model in its original form. Through simulations and analysis, you can gain valuable insights into phase transitions, collective behavior, and the influence of temperature and interactions. By harnessing the power of the Ising Model, you can deepen your understanding of complex systems and apply this knowledge to diverse fields such as statistical mechanics, finance, social systems, and more.