This document was created by **Christian Buth**.

It is made available on my homepage

http://www.Christian.Buth.mysite.de

for *free*.

If you encounter problems in displaying this file or

you find mistakes feel free to contact me

cbuth@ix.urz.uni-heidelberg.de .

Department of Physics and Astronomy

High Performance Computing in Physics Project Physics 4

Two-Dimensional XY Model

Christian Buth

June 1999

Abstract

This project examines a two-dimensional system of magnets similar to the spin Ising model. The magnets can point in any direction. An implementation of a Monte Carlo algorithm is used to generate a configuration of the magnets with canonical probability distribution to simulate the effect of heat. The equilibrium configuration is investigated with rising temperature. A phase transition is observed and its impact on the system's energy per site and specific heat is examined. The employed algorithm can be parallelised. This is done and the improvement in terms of speed is measured.

Declaration

I declare that this project and report is my own work.

Signature: Date:

Supervisor: Dr. David Henty 3 Weeks

Contents

1 Introduction

This project examines a two-dimensional system of magnets. It is similar to the spin Ising model, but differs from it in an important way. The magnets can point in any direction in the XY-plane. This inhibits the occurence of magnetisation. The system is a square array of sites. Cyclic boundary conditions are used to make the finit size effects less dominant.

The opposite influences on the magnets being attracted by each other which increases the ordering and the random thermal motion disturbing their ordering are the two physical precepts employed in the program.

To carry out the simulation the Metropolis algorithm is implemented to generate a canonical probability distribution among the magnets. The number of sweeps till an equilibrium configuration is reached is determined The average energy per site is an observable and its expectation values is measured.

The main focus of the project lies on the effects on the system near a phase transition. The temperature where this transition occurs is measured in two different ways using the expectation value of the average energy per site and the specific heat of the system.

For a range of temperatures the configurations of the magnets are visualised and especially its change after the phase transition is examined.

The Metropolis algorithm can be parallelised when the magnets are updated in a red-black checkerboard scheme. This is done using a serial and a parallel random number generator. The parallelisation is carried out by employing the *OpenMP*[1] standard. The scaleability of the system is measured in terms of efficiency and speed-up for both parallel algorithms.

[1] *OpenMP* is described in [2].

2 Theoretical Background

The underlying physics in the project can be summarised in a few elementary equations.

In the system each site interacts with its neighbours. The sum of the energys of all sites gives the energy of the system

$$E = -\frac{1}{2}\sum_{i,j=1}^{L} \Big(\cos(\theta_{i,j} - \theta_{i+1,j}) + \cos(\theta_{i,j} - \theta_{i-1,j}) + \cos(\theta_{i,j} - \theta_{i,j+1}) + \cos(\theta_{i,j} - \theta_{i,j-1})\Big). \tag{1}$$

The average energy per site is

$$\varepsilon = \frac{E}{L^2}. \tag{2}$$

The *specific heat* of the system is defined as

$$C_V = \frac{\partial}{\partial T} <\varepsilon>. \tag{3}$$

The expectation value of the observable ε is

$$<\varepsilon> = \sum_{\mathbf{x}} \varepsilon(\mathbf{x}) \cdot \frac{e^{-E(\mathbf{x})/T}}{\sum\limits_{\mathbf{x}'} e^{-E(\mathbf{x}')/T}}, \tag{4}$$

where $\mathbf{x}$ is a configuration. Using equation (3) with (4) yields

$$C_V = \sum_{\mathbf{x}} \left(\frac{\varepsilon(\mathbf{x})E(\mathbf{x})}{T^2} \cdot \frac{e^{-E(\mathbf{x})/T}}{\sum\limits_{\mathbf{x}'} e^{-E(\mathbf{x}')/T}} - \varepsilon(\mathbf{x}) \cdot \frac{e^{-E(\mathbf{x})/T} \sum\limits_{\mathbf{x}'} \frac{E(\mathbf{x}')}{T^2} e^{-E(\mathbf{x}')/T}}{\left(\sum\limits_{\mathbf{x}'} e^{-E(\mathbf{x}')/T}\right)^2} \right).$$

Simplifying gives

$$C_V = \frac{L^2}{T^2}(<\varepsilon^2> - <\varepsilon>^2). \tag{5}$$

To do the simulation the METROPOLIS algorithm is used[2]. This algorithm is used to generate a canonical probability distribution of the magnets, i.e. a probability $\propto e^{-E/T}$, where E denotes the energy of the system for a finite temperature T. This type of distribution arises from the effects of heat which is random motion.

3 Program

3.1 Usage

Initially, the program displays a menu from which the user can select a method how one Monte Carlo sweep shall be done. Then the program asks whether it shall display the

[2] A detailed explanation of the METROPOLIS algorithm can be found in chapter 5 of [1].

average energy per site, the acceptance rate and the balance after each sweep. This is useful for testing the code.

Afterwards the system size is asked for. The system size is the number of cells of one edge of the square system. Now the user of the program has to specify how many sweeps he wants and what temperature is used. Then the delta is requested. This parameter influences the acceptance rate, because it determines how big the difference in energy may become for old and proposed configurations.

The next question asks if the user whishes a hot start with the system initialised randomly or a cold start with all angles initialised to zero. When the first question whether the program shall display its configuration after each sweep was answer with no then the user must specify the sweep number after which measurement of observables shall begin.

Afterwards the program commences computing.

3.2 Structure

The program is written in FORTRAN77 and comprises of three parts. The first part is the main program which interacts with the user and contains the main loop to do the Monte Carlo simulation.

In the second part you find supporting procedures. These measure the average energy per site, set up the initial configuration, provide cyclic boundary conditions and visualise the magnets.

The third part are the routines for doing one Monte Carlo sweep.

Some parts of the program are implemented parallely. The parallelisation is done using *OpenMP*. Some of the supporting procedures are implemented serially and parallelly. Some of the routines for doing one Monte Carlo sweep are also parallelised.

4 Results and Discussion

4.1 Simulation

The average energy per site ε is a good observable to estimate when the system is in equilibrium. When ε of a cold-started system agrees with the ε of the warm-started system one can take the number of sweeps to achieve this for the number of sweeps to reach equilibrium. Figures 1, 2 and 3 illustrate this point.

After equilibration there is no obvious correlation between the ε of successive sweeps. This shows that the random number generator `uni()` produces values which are an excellent approximation to fully random numbers. There are no obvious correlations between the plots of hot and cold starts, either, although they use the same sequence of random numbers. This is the fact because the same random number is interpreted differently in both cases.

In the beginning the differences between the two configurations are big thus a proposed change in the hot-started case leads to a different energy difference than the same change in the cold-started case. This leads to a desycronisation of the two random number sequences. Therefore the two graphs seem to be independent in equilibrium. In the case that the

sweeping is done in parallel another pseudo-random component is introduced by *OpenMP* due to its partitioning of the system.

Figure 4 shows the expectation value for the average energy per site and was computed with `h4epsilon`. It increases slowly in the beginnig. Then it rises rapidly. In the end it increases slowly again. This behaviour shows that a phase transition occurs because the slope of the graph varies appreciably. Due to equation (3) this means that the specific heat is temporary high. This indicates a phase transition. Figure 4 shows that the *critical temperature* is $T_C = 1.10 \pm 0.05$.

Figures 5, 7, 9 and 11 show the direction of the magnets, figures 6, 8, 10 and 12 visualise the sytems with streamlines. The visualisation using streamlines turns out to be much more helpful because global trends can be seen easier. The hedgehog figures tend to be too microscopic to be of any great value as Monte Carlo simulations are used for statistical processes one has to consider bigger structures especially when visulising the calculated data.

The increasing temperature shows a remarkable change in the system. At a low temperatur, figure 5, the magnets form large structures of aligned chains of neighbouring sites. This looks very ordered. Rising the temperatures, figure 7, shortens the chains and removes the interconnections among them. This increases the amount of disorder in the system. This is exactly what is predicted by the underlying physics. Rising the temperature of a system leads to a corresponding increase in entropy!

Figure 9 shows that a phase transition has happend because the former regular distribution of the streamlines is replaced by a smaller scale clustring and formation of vortices. These little clusters interact only weakly with each other but are not totally independent. This corresponds to the change in structure when heating a solid till it becomes a liquid. Of course this is just an analogy. In figure 11 the degree of association between the clusters is appreciably smaller than in figure 9. This trend enforces further with rising temperature till the energy due to heat is bigger than the energy between neighbouring sites. Then a second phase transition will happen and the system will be totally disordered. Using the analogy of a substance this means that the system is gaseous. This case is not very interesting because the effects of ordering give the system its characteristic behaviour and importance.

The simulation uses cyclic boundary conditions. The impact of this can be seen clearly in figure 12, where vortices are split in between and continued on the oppsite side.

4.2 Specific Heat

`h4heat` is used to compute figure 13. It shows the specific heat C_V for varying temperatures. Around the critical temperature there is a rise in C_V. Using the fact that the peak is approximately symmetric $T_C = 1.10 \pm 0.03$ which is in excellent agreement with the formerly obtained value. Figure 13 shows that the error of the specific heat is fairly high. It should be a smooth peak.

Figures 14 and 15 show the specific heat with respect of the number of sweeps for measuring it. The graphs are quite similar because they exhibit the same features. A jagged quick rise in the beginning. Afterwards a rise to a local maximum at ≈ 4000, followed by a smoth behaviour towards higher numbers of sweeps.

As there is a correlation between $<\varepsilon>$ and $<\varepsilon^2>$ so there is a correlation between the

two figures according to equation 5 becaues ε changes similarily in both simulations.

4.3 Parallelisation

To parallelise the Monte Carlo simulation one has to loop over the cells in red-black ordering. This makes red sites independent of the black and vice versa due to equation (1) which depends only on the nearest neighbours and is used for the updating of the sites. With this scheme one can update the red sites in random order keeping the black cells fixed. Afterwards the black sites are updated and the red cells are kept fixed.

It is important to generate the random numbers for the *parallel* subroutine that does a Monte Carlo sweep using the *serial* random number generator `uni()` in a `SINGLE` section of the `PARALLEL` section that will compute the whole sweep, because *OpenMP* is allowed to adjust the number of threads used in `PARALLEL` sections. This adjustment may happen before such a section is entered and stays fixed till the section is left.

The *speed-up* and the *efficiency*[3] were measured using the real ellapsed time for the program execution, confer `h4perf`. The reference time for one processor was measured using the serial lexicographic updating scheme because it is the most efficient serial algorithm. The data points for one processor in the graphs 16, 17, 18 and 19 are the speed-up and the efficiency of the parallel algorithm executing on one processor against the serial lexicographic algorithms. The figures show that the parallel algorithm on one processor is slightly less fast than its serial counterpart.

The fact that the real ellapsed time was used for all measurements causes a considerable amount of error. Nevertheless some trends can be seen. Figures 16 and 18 show that the Monte Carlo sweep using `uni()` and the Monte Carlo sweep using `puni()` scale well. This is confirmed by figures 17 and 19 which show the efficiency of the two algorithms. The far better speed-up and efficiency for the sytem size 60 compared to the system size 20 comes from the problem itself. The amount of work that needs to be done scales like L^2. That the speed-up for a system of size 60 does not differ appreciably from that of a system size of 100 is probably due to errors in time measurement.

Figures 18 and 19 show that the usage of the *parallel* random number generator does not speed up the program appreciably. This indicates that it is very fast compared to the rest of the code in the `PARALLEL` section. An analysis of the code of `puni()` shows that it is similar to that of `uni()` except the fact that it uses an array of seeds, one for each thread, instead of a single variable. This justifies the former conclusion.

5 Conclusion

Although the project deals with magnets the basic ideas of the employed techniques are used in many areas of science. Thus the exercise gives an idea how to tackle statistical problems.

The project shows that even very small physical problems may need powerful computers to solve them numerically.

[3] For a definition of these terms see page 27 of [1].

The examination of the system reveiled following features. A phase transition of the system causes the energy contained in the system and the specific heat to increase rapidly. The specific heat drops afterwards to a lower value, the energy remains high. The system's ordered structures become smaller and change in their appearance, vortices form.

6 References

[1] Lecture notes to *High Performance Computing in Physics*.
[2] Technology Watch Report of the Edinburgh Parallel Computing Centre (EPCC) on OpenMP: http://www.epcc.ed.ac.uk/epcc-tec/documents/techwatch-openmp/form2.html .