



*Электронное научное издание
«Ученые заметки ТОГУ»
2018, Том 9, № 1, С. 286 – 292*

*Свидетельство
Эл № ФС 77-39676 от 05.05.2010
[http://pnu.edu.ru/ru/ejournal/about/
ejournal@pnu.edu.ru](http://pnu.edu.ru/ru/ejournal/about/ejournal@pnu.edu.ru)*

УДК 004.42

© 2018 г. С. А. Зайцев,
А. И. Андреев

(Тихоокеанский государственный университет, Хабаровск)

МОДЕЛИРОВАНИЕ МАГНИТНЫХ ФАЗОВЫХ ПЕРЕХОДОВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ CUDA

В статье исследуются магнитные свойства вещества на основе решеточной модели Изинга. Для изучения фазового перехода используется численный метод Монте-Карло. Предложен параллельный вариант алгоритма Метрополиса, который реализован с применением технологии CUDA.

Ключевые слова: фазовый переход второго рода, ферромагнетики, модель Изинга, спиновая решетка, метод Монте-Карло, Алгоритм Метрополиса, CUDA.

S. A. Zaytsev, A. I. Andreev

SIMULATION OF MAGNETIC PHASE TRANSITIONS USING CUDA TECHNOLOGY

The article studies the magnetic properties of matter on the basis of the lattice Ising model. To study the phase transition we use a numerical method Monte-Carlo. A parallel version of the Metro-policy algorithm, which is implemented using the CUDA technology, is proposed.

Keywords: phase transition of the second kind, ferromagnets, Ising model, spin lattice, Coin-Carlo method, metropolis Algorithm, CUDA

Введение

В настоящее время набирает популярность такое направление, как GPGPU (General-Purpose computing on Graphics Processing Units) – использование графических процессоров для решения неграфических задач. За счет использования высокой степени параллелизма удастся в значительной степени ускорить вычисления по сравнению с центральным процессором CPU. Для неграфических вычислений компанией nVidia была разработана технология CUDA (Compute Unified Device Architecture), заметно облегчает написание GPGPU-приложений. Данная технология предназначена для разработки приложений для массивно-параллельных вычислительных устройств. В данной работе мы используем GPU для моделирования магнитных свойств среды в области критических температур.

Теоретические основы

Модель Изинга используется для моделирования канонических ансамблей и представляет собой решетку, в узлах которой расположены спины. Для простоты принимается, что спины могут быть ориентированы только в двух направлениях – вверх и вниз. С помощью данной модели можно исследовать ферромагнетизм. В частности, представляет интерес поведение системы вблизи критической температуры, называемой температурой Кюри для ферромагнетиков и температурой Нееля для антиферромагнетиков. Эта критическая температура означает фазовый переход второго рода, в результате которого вещество теряет свои ферромагнитные свойства и становится парамагнетиком. При приближении к критической температуре теплоемкость и магнитная восприимчивость стремятся к бесконечности (испытывают скачок).

В одномерном случае фазовый переход не происходит, что было доказано автором модели Изингом. Однако в двух- и трехмерном случае обнаруживается фазовый переход. Для определения критической температуры используется метод, основанный на связи теплоемкости и магнитной восприимчивости со статистическими флуктуациями полной энергии в ансамбле [1]:

$$C = \frac{1}{k_B T^2} (\langle E^2 \rangle - \langle E \rangle^2)$$

$$\chi = \frac{1}{k_B T} (\langle M^2 \rangle - \langle M \rangle^2)$$

Двумерная модель имеет точное аналитическое решение, однако оно не применимо на практике. Основная проблема заключается в том, что необходимо искать статистическую сумму по всем состояниям системы:

$$Z = 2e^{8\beta J} + 12 + 2e^{-8\beta J}$$

где параметр $\beta = \frac{1}{T}$. Число всевозможных конфигураций равно 2^n . Поэтому даже для простейшей решетки из 4 узлов имеем $2^4 = 16$ различных конфигураций. С целью минимизировать краевые эффекты требуется исследовать решетки с намного большим числом узлов. Таким образом применение аналитического метода потребует слишком больших вычислительных ресурсов ЭВМ и не оправдано.

Для нахождения параметров системы мы прибегаем к методу Монте-Карло. Он представляет собой численный метод, суть которого заключается в нахождении связи между вероятностными характеристиками случайных процессов (математическими ожиданиями случайных величин) и величинами, являющимися решениями задачи [4].

Метод Монте-Карло обладает высокой степенью параллелизма, что позволяет равномерно нагружать вычислительные узлы компьютерных систем. Точность метода растет с увеличением числа выборки.

Алгоритм Метрополиса для GPU

Для реализации численного метода Монте-Карло в применении к модели Изинга существует т.н. алгоритм Метрополиса. В данной работе используется его разновидность – алгоритм переворота («опрокидывания») спина. Алгоритм с динамикой переворота спина является вполне разумным приближением к реальной динамике анизотропного магнита, спины которого связаны с колебаниями решетки. Можно ожидать, что один шаг Монте-Карло на спин пропорционален среднему времени между опрокидываниями спинов, наблюдаемому в лабораторном эксперименте. Следовательно, мы можем рассматривать динамику опрокидывания спинов как настоящий нестационарный процесс и наблюдать релаксацию к равновесию по прошествии достаточно длительного времени.

Алгоритм Монте-Карло в общем виде представляется следующим образом [6]:

1. Задание начальной конфигурации.
2. Генерация нового состояния \mathbf{X} .
3. Вычисляется изменение энергии в результате перехода ΔE .
4. При $\Delta E \leq 0$ новую конфигурацию принять и вернуться к шагу 2.
5. Вычислить $W = \exp\left(-\frac{\Delta E}{k_B T}\right)$.
6. Сгенерировать равномерно распределенное случайное число $R \in [0,1]$.
7. Если $R < W$, принять новую конфигурацию и перейти к шагу 2.
8. Иначе оставить старую конфигурацию и перейти к шагу 2.

Система движется к состоянию с минимальной энергией. Шаг 4 означает, что мы всегда принимаем новую конфигурацию, имеющую меньшую энергию, чем предыдущая. Конфигурации, уменьшающие энергию системы, принимаются только с больцмановской вероятностью.

На втором шаге алгоритма происходит пробный переворот случайного спина. Далее вычисляется изменение энергии в результате этого переворота, которое определяется суммой ближайших соседних спинов (в нашей работе мы используем радиус корреляции равный единице, (рис. 1).

Проблемы, связанные с наличием у системы границ, устраняются обычно граничными условиями. Мы учитываем простые граничные условия, когда последний спин решетки будет взаимодействовать с первым спином (тороидальные граничные условия).

Описанный выше алгоритм Метрополиса является последовательным и соответствует одному временному шагу системы. На одном временном шаге происходит последовательное выполнение пунктов 2-8 алгоритма – попытка переворота случайного спина решетки. Число таких попыток равно числу узлов, и в среднем происходит пробное «опрокидывание» каждого спина. При этом если в результате переворота одного спина была принята новая конфигурация, полученный результат может оказать влияние на последующие итерации, так как при работе с конкретным спином находится сумма его четырех ближайших соседей. Таким образом алгоритм Метрополиса изначально является последовательным и не пригоден для многопоточной реализации, особенно на многоядерных ЭВМ.

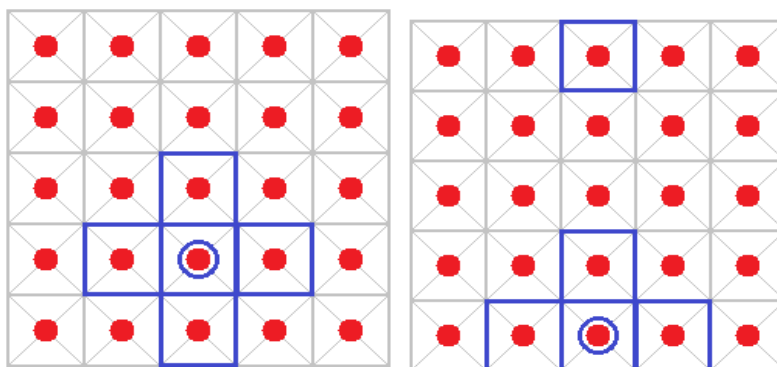


Рис. 1. Сумма ближайших соседних спинов в узле с учетом тороидальных граничных условий

Для решения проблемы последовательности алгоритм Метрополиса был существенно переработан с учетом специфики расчетов на GPU. Так, на одном временном шаге вычислительный процесс разбивается на два этапа. Узлы решетки разбиваются по четным и нечетным диагоналям (рис. 2). На первом этапе действия проводятся над узлами, расположенными на четных диагоналях. На втором этапе процесс повторяется, но с узлами, расположенными на нечетных диагоналях. Таким образом, за два условных такта GPU происходит вычисление одного шага по времени, тогда как на CPU этот же самый процесс занял бы N условных тактов процессора, где N – число узлов решетки. Это обусловлено тем, что одно ядро GPU вычисляет всего два спина, в то время как одно ядро CPU должно обработать все спины решетки.

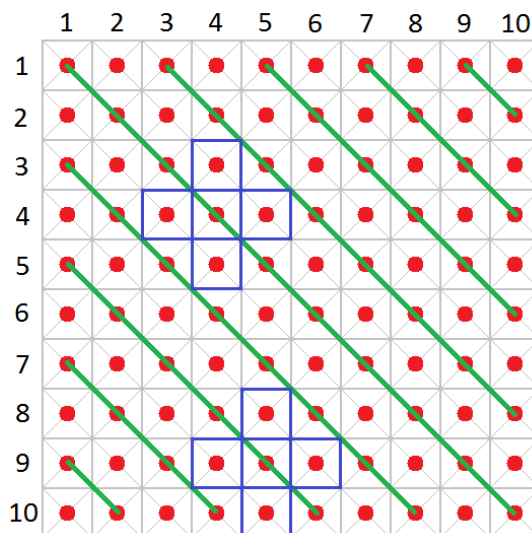


Рис. 2. Разбиение решетки на четные и нечетные диагонали

Тем не менее, на практике различие в скорости расчетов зависит от числа узлов решетки. Для небольших решеток (15x15 и менее) применение видеокарты для расчетов не дает никакого выигрыша. Это связано с тем, что перед запуском расчетов на видеокарте ее следует подготовить: выделить память по данным, загрузить исходные значения и запустить функцию-ядро на выполнение. Пересылка данных между памятью видеокарты и ОЗУ является дорогостоящей операцией, поэтому вести расчеты для небольших решеток выгоднее на CPU. Однако с ростом числа узлов разница в скорости вычислений между CPU и GPU может составлять более чем в 15 раз в пользу видеокарты (для решеток 64x64 и более крупных).

Конечной целью программы является нахождение критической температуры. Как было сказано, в точке Кюри теплоемкость и магнитная восприимчивость магнитной системы имеют асимптоту. Получив набор точек в диапазоне температур, мы можем построить графики зависимости теплоемкости и восприимчивости. Чтобы сгладить флуктуации этих величин, производится аппроксимация полиномом n -ой степени. Путем поиска корней полинома находятся точки экстремума. Отбирая эти точки по условию перемены знака, находим искомую критическую температуру.

Результаты

В таблице 1 приводятся сравнительные результаты вычислений средней энергии и намагнитченности с использованием последовательного алгоритма на CPU и параллельного на GPU. Все результаты нормированы на один спин. Начальная конфигурация принималась такой, чтобы все спины были направлены вверх. Параметр $nmcs$ – число статистических испытаний. T – температура окружающей среды, выраженная в единицах энергии: $E = kT$, где k – постоянная Больцмана, принятая равной единице. Для сравнения скорости алгоритмов приводится время их выполнения.

Таблица 1

Сравнение алгоритмов Метрополиса для CPU и GPU

Параметры решетки		$\langle E \rangle$	$\langle E^2 \rangle$	$\langle M \rangle$	$\langle M^2 \rangle$	Время, мс
2x2, $T=2$, $nmcs=1000$	CPU	-1,65	3,36	0,83	0,86	< 0
	GPU	-1,74	3,47	0,87	0,90	45
300x300, $T=2$, $nmcs=1000$	CPU	-1,75	3,06	0,92	0,85	18897
	GPU	-1,76	3,09	0,93	0,86	1264
300x300, $T=1$, $nmcs=1000$	CPU	-1,99	3,97	1,00	1,00	19159
	GPU	-2,00	3,99	1,00	1,00	1035
100x100, $T=1$, $nmcs=10000$	CPU	-1,99	3,97	1,00	1,00	20711
	GPU	-2,00	3,99	1,00	1,00	1535
300x300, $T=1$, $nmcs=10000$	CPU	-1.99	3,97	1,00	1,00	194591
100x100, $T=1$, $nmcs=10000$	GPU	-2,00	3,99	1,00	1,00	10343

Также из результатов тестирования видно, что точность результатов зависит от размеров решетки, а разброс результатов достаточно велик для маленьких решеток, на что оказывают влияние краевые эффекты и малое число выборки.

На рис. 3-4 приведены графики зависимости теплоемкости от температуры. Здесь можно увидеть, что система испытывает фазовый переход при приближении к определенной температуре. Путем построения аппроксимирующего полинома и нахождения его корней определяется значение температуры в этой точке. Построение графиков проводилось с температурным шагом, равным 0,05, число шагов Монте-Карло $nmcs$ при каждом значении температуры принималось равным 1000.

Для решетки размером 8x8 была получена критическая температура $T_c = 2,619$, для решетки размером 128x128 $T_c = 2,720$.

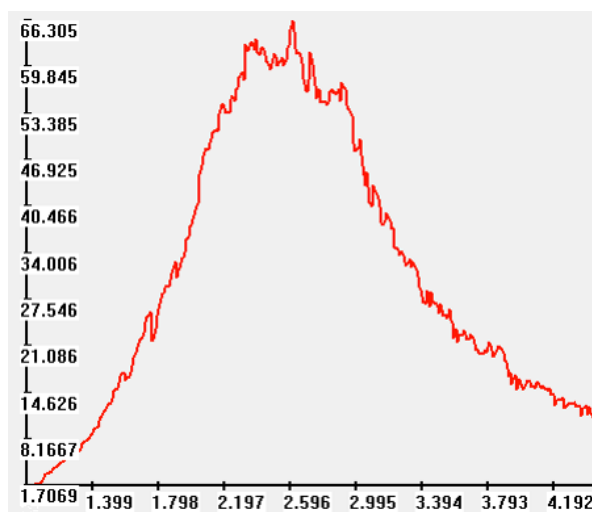


Рис. 3. Зависимость теплоемкости от температуры для решетки 8x8

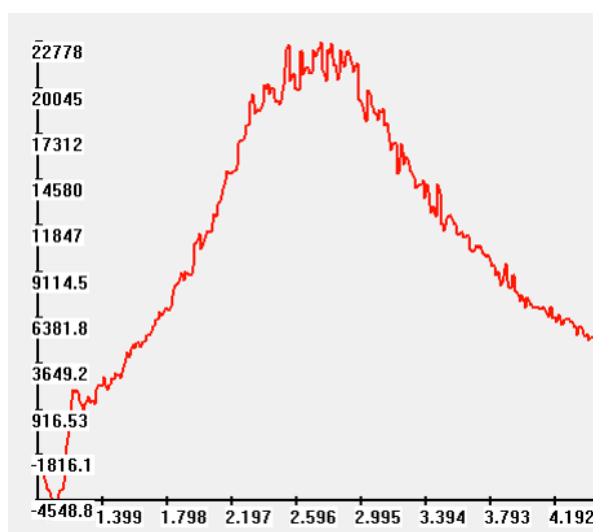


Рис. 4. Зависимость теплоемкости от температуры для решетки 128x128

Заключение

В данной работе предлагается параллельный алгоритм для моделирования двумерной квадратной решетки Изинга с радиусом корреляции равным единице. Эффективность алгоритма протестирована на решетках различных размеров с использованием технологии CUDA. Исследуется поведение теплоемкости и магнитной восприимчивости ферромагнетика вблизи точки Кюри. Получено значение критической температуры фазового перехода, близкое к теоретическому.

Список литературы

- [1] Х. Гулд, Я. Тобочник – Компьютерное моделирование в физике: часть 2. М.: Мир, 1990. — 400 с.
- [2] К. Хуанг – Статистическая механика. М.: Мир – 1966. - 346 с.
- [3] Г. А. Мартынов. Проблема фазовых переходов в статистической механике // УФН, Т.169, №6, 1999, С.595-624.
- [4] И. М. Соболев - Численные методы Монте-Карло. М.: Наука – 1973. – 312 с.

- [5] К. Биндер, Д. Хеерманов – Моделирование методом Монте-Карло в статистической физике. М.: Наука – физматлит. – 1995. – 141.
- [6] Д. В. Хеерман – Методы компьютерного эксперимента в теоретической физике. М.: Наука, 1990. — 175 с.
- [7] А. В. Боресков, А. А. Харламов – Основы работы с технологией CUDA. М.: ДМК Пресс, 2010. 232 с.
- [8] А. В. Боресков и др. – Параллельные вычисления на GPU. Архитектура и программная модель CUDA. М.: Издательство Московского университета, 2012. 336 с.
- [9] К. А. Некрасов, С. И. Поташников, А. С. Боярченков, А. Я. Купряжкин – Метод Монте-Карло на графических процессорах. Екатеринбург: изд. Урал ун-та, 2016. – 60с..