A Monte Carlo Study of the Classical, Isotropic, 3D Heisenberg Model

Numerical Studies of Stochastic Spin Systems

Michael Conroy

PHY 471 Capstone Project

Spring 2014

Professor: Dr. Matthew Enjalran

May 2, 2014

Southern Connecticut State University
SCSU

Table of Contents

Southern Connecticut State University
SCSU

Goal and Purpose

- Simulate the classical, isotropic, 3D Heisenberg Model on the simple cubic lattice
- Utilize the Monte Carlo method with the Metropolis Algorithm
- Compare simulation data to literature data
- Explore numerical analysis approach to the simulation

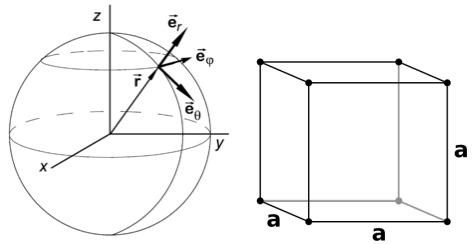
Heisenberg Model

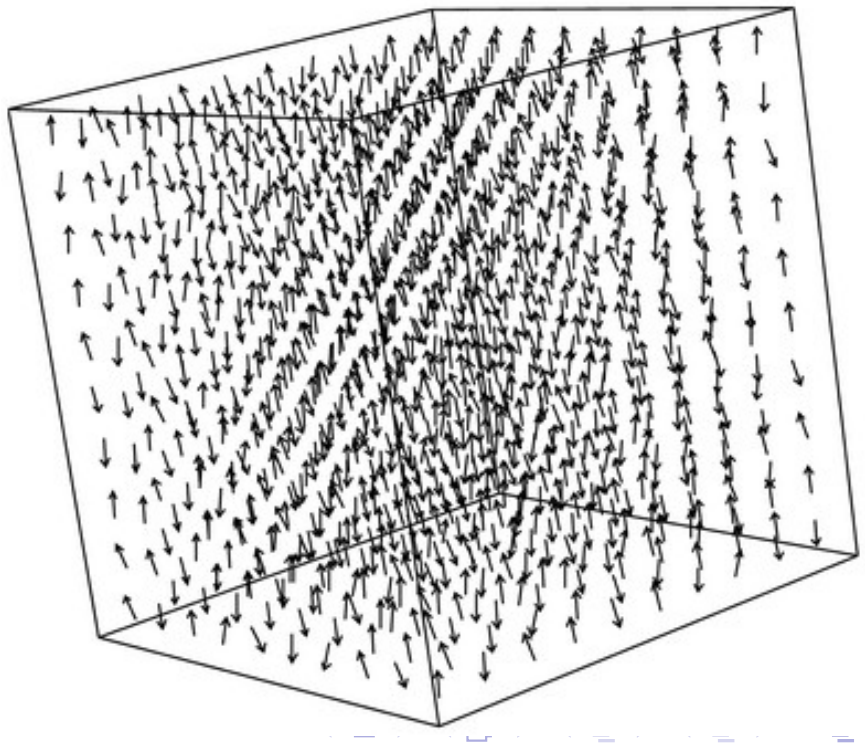
- Continuous spin model
- Hamiltonian:

$$H = -J \sum_{<ij>}^{N} \vec{S}_i \cdot \vec{S}_j,$$

where $|\vec{S}_i| = 1$.

- Applications

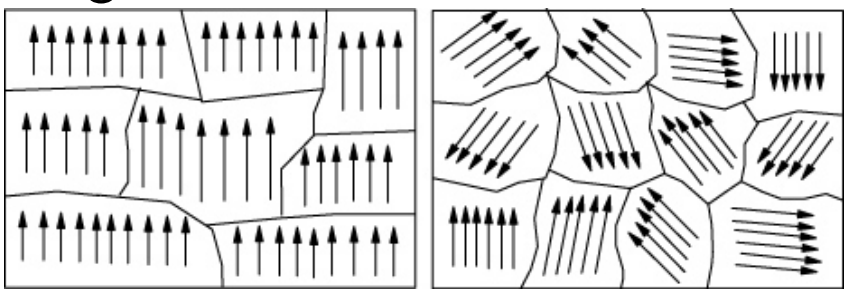




n Connecticut
niversity

Magnetism and Magnetism in Statistical Mechanics

- Magnetic Domains



- Paramagnetism, ferromagnetism, and antiferromagnetism

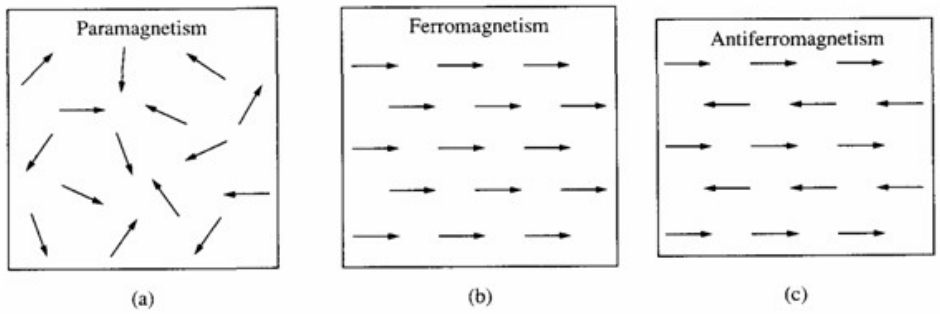
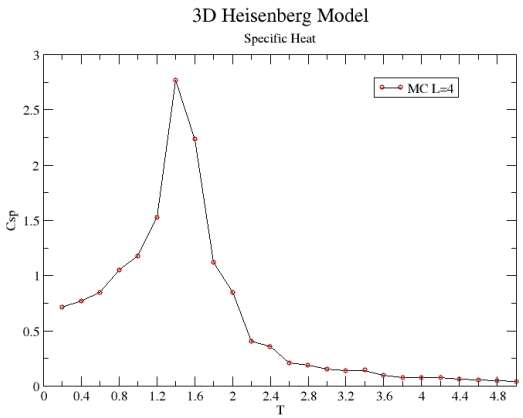


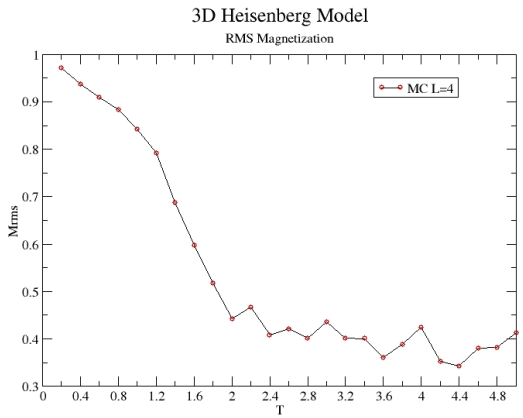
Fig. 11.58 Schematic representations of magnetic dipole arrangements in (a) paramagnetic, (b) ferromagnetic, and (antiferromagnetic materials.

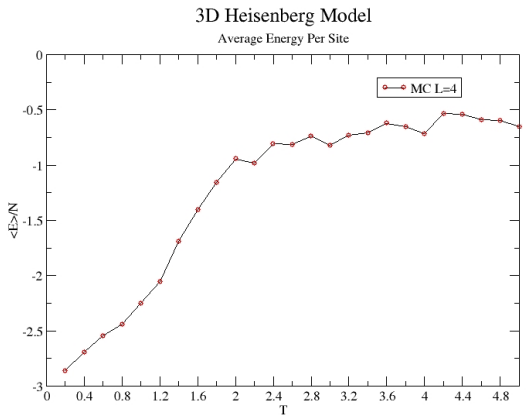
Southern Connecticut State University
SCSU

Phase Transitions

- Critical Temperature: $\beta_c \approx 0.69$ or $T_c \approx 1.45K$
- Order Parameter







Southern Connecticut State University
SCSU

Numerical Analysis

- No analytic solutions
- Intractable problems
- Monte Carlo simulations
  - Importance sampling

Southern Connecticut State University
SCSU

Statistical Mechanics

- Canonical Ensemble
- Boltzmann Distribution: $p_\mu = \frac{1}{Z(\beta)} e^{-\beta E(\mu)}$
- Partition Function: $Z(\beta) = \sum_\mu e^{-\beta E(\mu)}$
- Most macroscopic thermodynamic variables of a system can be expressed by the partition function or its derivatives!
- For example, energy, specific heat, entropy, free energy...

Southern Connecticut State University
SCSU

Calculating the Physical Quantities

- How do we calculate the required physical quantitites of the Heisenberg Model?
- Energy and specific heat:

  $E = -\frac{J}{2}\sum_{<ij>}^{N} \vec{S_i} \cdot \vec{S_j}$ (factor of $\frac{1}{2}$ for double counting)

  $C = k\beta^2(\langle E^2 \rangle - \langle E \rangle^2)$
- Magnetization:

  $m_{rms} = \sqrt{M_x^2 + M_y^2 + M_z^2},$

  where $M_\alpha = \frac{1}{N}\sum_i \vec{S_{i\alpha}}$

Southern Connecticut State University
SCSU

Application of Metropolis Monte Carlo

- Numerical Analysis
  - No analytic solution or intractable
- Pseudo-random number generation
- Monte Carlo Simulation
  - Importance sampling: must satisfy Markov processes, ergodicity, and detailed balance
  - Metropolis importance sampling scheme/algorithm
    1. Choose initial state
    2. Choose a site
    3. Calculate ΔE if "flip" the spin
    4. If $\Delta E \leq 0$, accept "flip" and go to (7), otherwise (5)
    5. RNG: $0 < r < 1$
    6. If $r < \exp(-\beta \Delta E)$, accept "flip"
    7. Go to next site and go to (3)...
  - Acceptance Ratio

Southern Connecticut State University
SCSU

Implementation
Software

- KISS: "Keep It Simple Stupid"
- Functional Approach in C
- 3D and 4D Pseudo-arrays of Pointers
- GNU GCC Compiler, Code::Blocks IDE

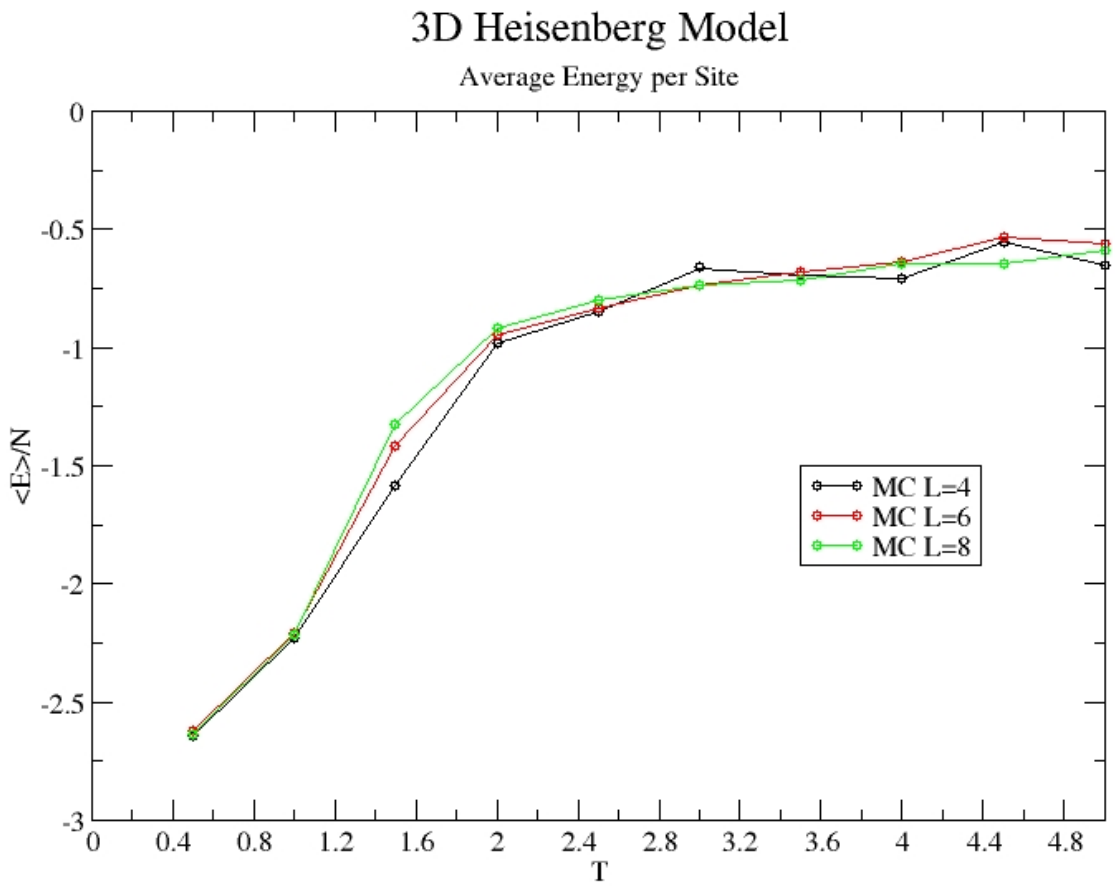
```
//Declare 4D arrays
printf("Declaring 4D arrays...\n\n");
//lattice
//double lattice[LENGTH][LENGTH][LENGTH][2]; //The Lattice[x loc][y loc][z loc][0=theta/1=phi
double ****lattice;
lattice = (double****)malloc(LENGTH * sizeof(double ***));
    if (lattice == NULL)
    {
        printf("7: Out of memory!\n");
        exit(7);
    }
for(i = 0; i < LENGTH; i++)
{
/*-------------------------------------------*/
    lattice[i] = (double***)malloc(LENGTH * sizeof(double**));
        if (lattice == NULL)
        {
            printf("8: Out of memory!\n");
            exit(8);
        }
    for(j = 0; j < LENGTH; j++)
    {
        lattice[i][j] = (double**)malloc(LENGTH * sizeof(double*));
            if (lattice == NULL)
            {
                printf("9: Out of memory!\n");
                exit(9);
            }
        for (k = 0; k < LENGTH; k++)
        {
            lattice[i][j][k] = (double*)malloc(Y_SIZE * sizeof(double));
                if (lattice == NULL)
                {
                    printf("10: Out of memory!\n");
                    exit(10);
                }
        }
    }
} //Declare 4D arrays
```

Implementation
Hardware

- Workstation
  Lenovo IdeaPad Y580
  Intel i7-3630QM, 8-thread, 3.4 GHz (max, single core)
  16 GB ram, 256 GB SSD
  GeForce GTX 660M (overclocked to 1 GHz)
  Fedora 20 Linux, Scientific Spin
- Simulation Machines
  Custom Built PCs
  AMD Opteron 6212, 16-thread, 3.2 GHz (max, $\leq$ 4 core)
  32GB ram, Fedora and Ubuntu Linux

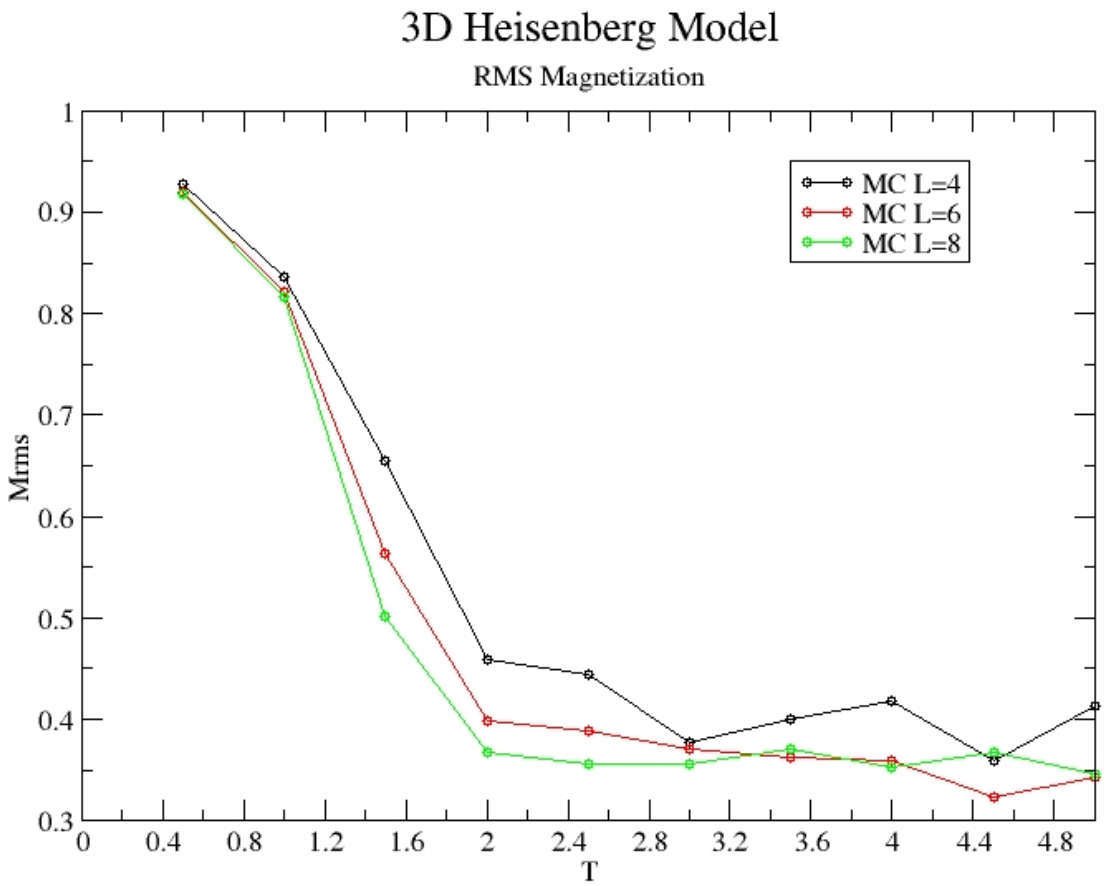
Southern Connecticut State University
SCSU

Data Plots
Energy



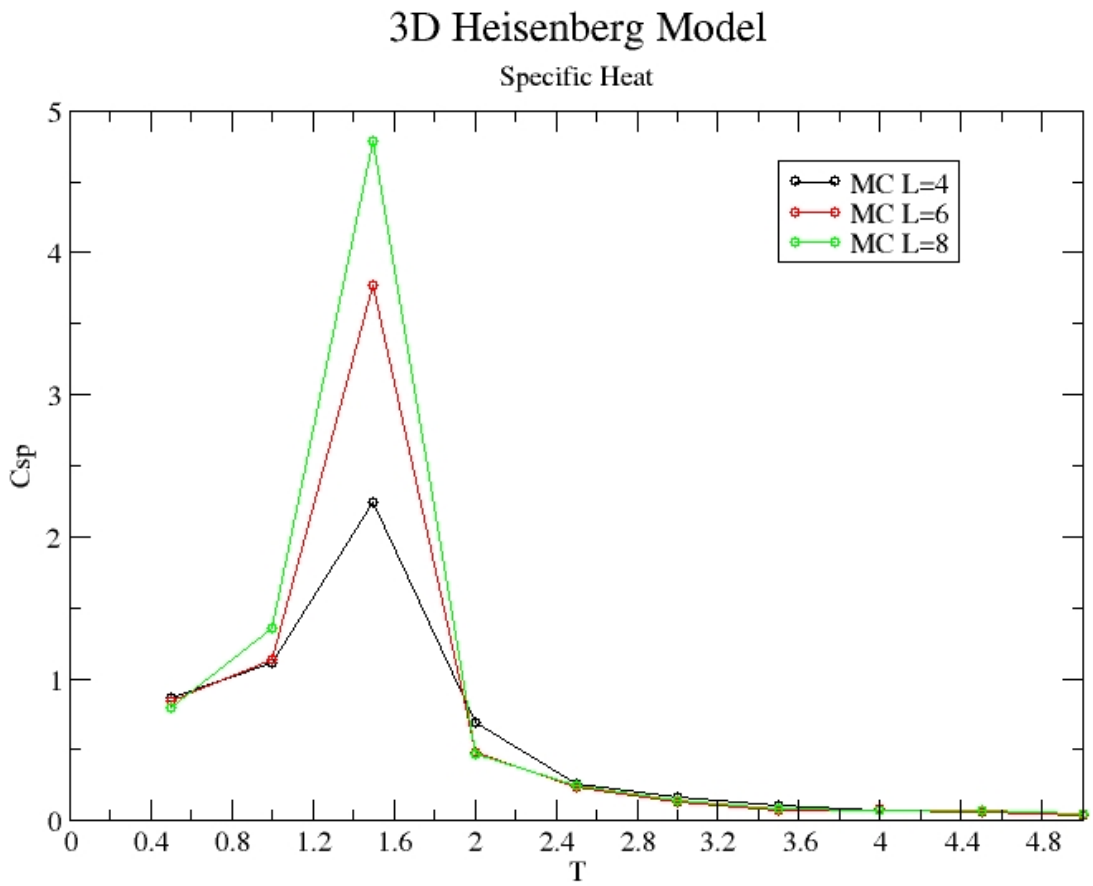
Southern Connecticut State University
SCSU

Data Plots
Magnetization



Southern Connecticut State University
SCSU

Data Plots
Specific Heat



Southern Connecticut State University
SCSU

Data Plots
Magnetic Susceptibility

- **Susceptibility calculation not straightforward due to rotational invariance in 3D Heisenberg Model!**

Southern Connecticut State University
SCSU

Conclusion

- Data matches predicted behavior
- Phase transition at critical temperature of $T_c \approx 1.45K$
- Susceptibility not as straightforward to calculate as in the Ising Model
- Acceptance ratio oddity
- Programmatic concerns
  - Code in Fortran for readability, debugging, and intrinsic functions (but be careful!)
  - Fix arrays and possibly implement structures
  - Improve simulation run time! Optimize code!

Southern Connecticut State University
SCSU

Next Steps and Future Work

- Code improvements
- Susceptibility
  - Correlation function calculation
- GPU CUDA implementation
- Spin dynamics
- Magnetic frustration and other lattices
- Possible project: LED Cube visualization of 3D Ising model phase change

Southern Connecticut State University
SCSU

Acknowledgments and References

- **Acknowledgments**

  Tom Sadowski for his invaluable help with programming in C.
- **References**

  - Landau, D. P., Binder, K. (2009). A guide to Monte Carlo simulations in statistical physics (3rd ed.). Cambridge: Cambridge University Press.
  - Newman, M. E., Barkema, G. T. (1999). Monte Carlo methods in statistical physics. Oxford: Clarendon Press ;.
  - Gould, H., Tobochnik, J. (2010). Statistical and thermal physics: with computer applications. Princeton, N.J.: Princeton University Press.
  - Tangirala, Sairam. Monte Carlo Simulation of 1D Heisenberg Model. http://www.scribd.com/doc/50664110/Monte-Carlo-Simulation-of-1D-Heisenberg-Model
  - arXiv:hep-lat/9301002
  - arXiv:hep-lat/0209017

Southern Connecticut State University
SCSU