Developing a Numerical Solution to Ferromagnetic Materials

Elizabeth Drueke

May 4, 2016

Abstract

Modeling lattice-type structures under the effects of magnetization has been achieved with reasonable accuracy using the Ising model. In order to accurately describe the larger of these systems, it is important to develop computer algorithms which can do large-scale computations in small amounts of time. To this end, we develop a Monte Carlo process using the Metropolis Algorithm to study the effects of magnetization on two-dimensional lattice systems of various sizes and at various temperatures, and use the results of this analysis to determine the accuracy of our algorithm. In particular, we focus on the effect of various random number generators on the results, to ensure that we are not encountering a strong systematic bias from this source.

Theory

Background Information

- Magnetization is Important:
  - Navigation
  - Physics Applications
- 3 Types:
  - Paramagnetism: spins align parallel to magnetic field.
  - Diamagnetism: spins align opposite to magnetic field.
  - Ferromagnetism: magnetization is retained even in absence of magnetic field.
- This presentation will discuss modeling a two-dimensional ferromagnetic material with the Ising model.

The Ising Model

- The Ising model describes the energy of a ferromagnetic material as [5]

$$E = -J \sum_{\langle kl \rangle}^{N} s_k s_l - B \sum_{k}^{N} s_k, \tag{1}$$

where $s_k = \pm 1$ is the spin of the k^{th} lattice point, J is a coupling constant expressing the strength of the interaction between the neighboring spins, B is the external magnetic field the material is in, and the first sum is taken over nearest neighbors (ie. lattice points directly above, below, to the left, or to the right of each other, assuming periodic boundary conditions).

- In the absence of a magnetic field, (1) becomes

$$E = -J \sum_{\langle kl \rangle}^{N} s_k s_l. \tag{2}$$

Statistical Mechanics

- Statistical mechanics is a science concerned primarily with the expected behavior exhibited by large groups of interacting objects (spins).
- It relies heavily on probability because it is impossible to determine exactly how each individual spin is behaving.

Statistical Mechanics

- We are often interested in the expected values of some physical quantity X, given by

$$\langle X \rangle = \sum_{i=1}^{N} X_i P_i, \quad (3)$$

where the sum is over all possible states, X_i is the value of X in state i, and P_i is a probability distribution function.

- The probability distribution function typically used in statistical mechanics is the Boltzmann distribution,

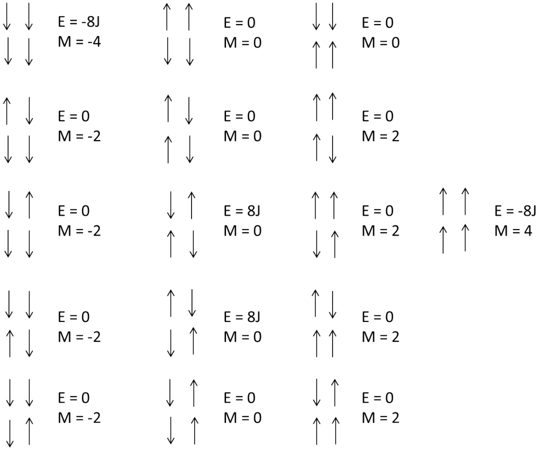
$$P_i = \frac{e^{-\frac{E_i}{k_B T}}}{Z}, \quad (4)$$

where E_i is the energy of state i, k_B is the Boltzmann constant, T is the temperature, and Z is the partition function, given by

$$Z = \sum_{i=1}^{N} e^{-\frac{E_i}{k_B T}}. \quad (5)$$

A 2×2 Lattice

- There will be $2^4 = 16$ possible states of the system.



(a)

A 2×2 Lattice

- Partition function:

$$Z = 2e^{-\frac{8J}{k_BT}} + 2e^{\frac{8J}{k_BT}} + 12 = 4\cosh\frac{8J}{k_BT} + 12. \tag{6}$$

- Expected value of energy:

$$\langle E\rangle = \frac{1}{Z}\left(16Je^{-\frac{8J}{k_BT}} - 16Je^{\frac{8J}{k_BT}}\right) = \frac{-32J\sinh\frac{8J}{k_BT}}{Z}. \tag{7}$$

- Magnetization and expected magnetization:

$$M_i = \sum_{i=1}^{N} s_i \Rightarrow \langle M\rangle = 0. \tag{8}$$

- Expected squared magnetization:

$$\langle|M|\rangle = \frac{1}{Z}\left(8e^{\frac{8J}{k_BT}} + 4\right). \tag{9}$$

A 2×2 Lattice

- Specific heat (variance of the energy of the system multiplied by some constant):

$$\begin{aligned}
C_V &= \frac{1}{k_B T^2}\left(\left\langle E^2\right\rangle - \langle E\rangle^2\right) \\
&= \frac{1}{k_B T^2}\left(\frac{256J^2\cosh\frac{8J}{k_BT}}{Z} - \left(\frac{-32J\sinh\frac{8J}{k_BT}}{Z}\right)^2\right) \\
&= \frac{256J^2}{k_B T^2 Z}\left(\cosh\frac{8J}{k_BT} - 4\sinh^2\frac{8J}{k_BT}\right)
\end{aligned} \tag{10}$$

A 2×2 Lattice

- Magnetic susceptibility (variance of the magnetization of the system multiplied by some constant):

$$\chi = \frac{1}{k_B T}\left(\langle M^2 \rangle - \langle M \rangle^2\right) = \frac{32 e^{\frac{8J}{k_B T}} + 8}{Z k_B T} \tag{11}$$

The Algorithm

lattice Class

- Each lattice object has the following objects:
  - size, an int which gives the size of the square lattice (ie. n for an $n \times n$ lattice),
  - spins, a dynamically allocated matrix of the spins in the matrix,
  - temp, a double which gives the temperature of the matrix,
  - averages, a dynamically allocated vector of the averages of E, E^2, M, M^2, and $|M|$,
  - MCcycles, an int which gives the number of Monte Carlo (MC) cycles to be used in the calculation,
  - CV, a double of the specific heat,
  - chi, a double of the susceptibility,
  - chi_absm, a double of the susceptibility calculated from $<|M|>$ rather than $<M>$ (that is, $\chi_{|M|} = <M^2> - <|M|>^2$),
  - accepted, an int of the number of Monte Carlo events accepted in the calculation, and
  - e_probs, a map of an energy value to the number of times that energy appears in the calculations.

`lattice` Class

- In addition to a default constructor, copy constructor, and destructor, `lattice` has a:
  - constructor from a size, temperature, number of Monte Carlo cycles, and the option to start at a lowest-energy state,
  - several "get" functions (eg. `get_E()`), which returns some statistical mechanical quantities,
  - several "set" function (eg. `set_temp(double t)`), which resets and recalculates some variable,
  - `plot_eprobs(string name)`, which plots the probability of various energy values E appearing in the calculations, and
  - `calc_stat_quants()`, which is a private function which performs all of the calculations and sets all of the variables as the `lattice` object is constructed.

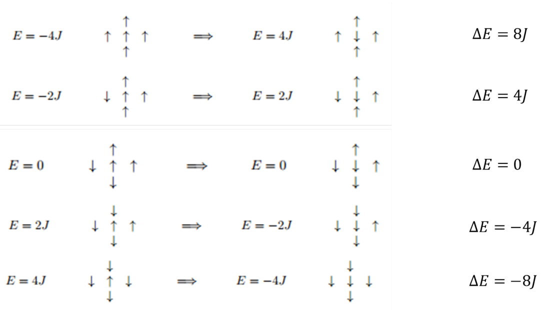
`calc_stat_quants()` and Monte Carlo

- Monte Carlo is required in any computer simulation of the Ising model.
- This is implemented in the `calc_stat_quants()` function, which is called by all constructors after a random (or not) initial state is set for the lattice:
    1. Calculating the initial energy with periodic boundary conditions
    2. Monte Carlo simulation (Metropolis Algorithm)
    3. Update statistical mechanical quantities

Metropolis Algorithm

1. Iterate through integers `cycle` which are less than the `MCcycles` parameter
2. Randomly change one of the spins in the lattice
3. Calculate the change in energy resulting from changing this spin by calling the `nearest_neighbors()` function
4. If change in energy is negative, generate a random number `myrand` (in $[0, 1]$) and compare to $w = e^{-\frac{\Delta E}{k_B T}}$
   - $w \leq$ `myrand`: keep the change in spin
   - $w >$ `myrand`: keep the spin matrix as it was before.

Possible Values of ΔE for a 2×2 Lattice



(b)

Figure: The possible values of ΔE for the 2×2 lattice after changing one spin [5].

Benchmarks

Expected Results for 2×2 Lattice

Statistical Quantity	2×2 Accepted Value
Z	5973.917
$\langle E \rangle$	-7.984
$\langle \lvert M \rvert \rangle$	3.993
$\langle E^2 \rangle$	63.871
$\langle M^2 \rangle$	15.969
C_V	0.127
χ	16.001

Table: The accepted calculated values for the 2×2 lattice under the Ising model for a temperature $T = 1/k_B$.

Computed Values for 2×2 Lattice

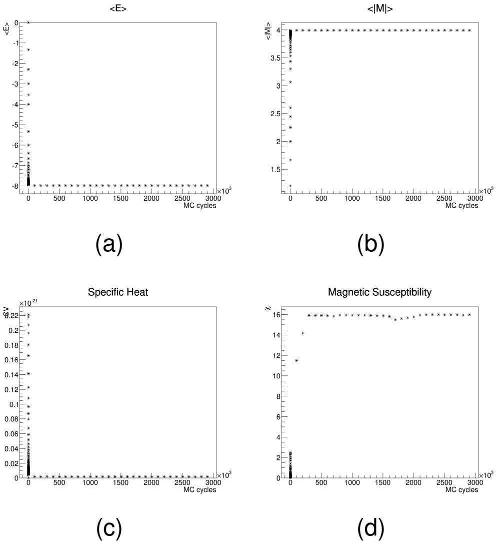
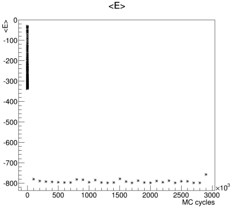


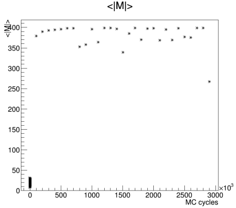
Figure: The (a) $< E >$, (b) $< |M| >$, (c) C_V, and (d) χ for a 2×2 lattice at temperature $T = 1/k_B$.

Results

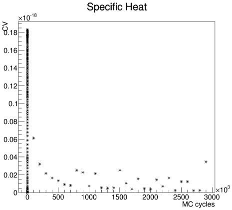
Number of Required Monte Carlo



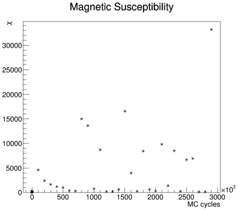
(a)



(b)



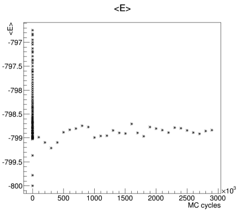
(c)



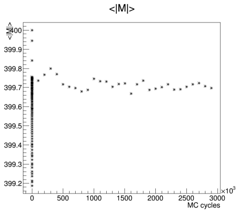
(d)

Figure: Statistical quantities for a 20×20 lattice at a temperature $T = 1/k_B$ with an initial state of a random selection of spins.

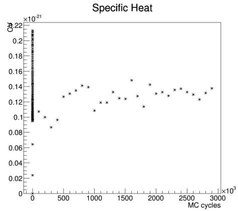
Number of Required Monte Carlo



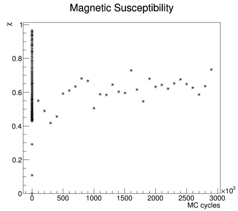
(a)



(b)



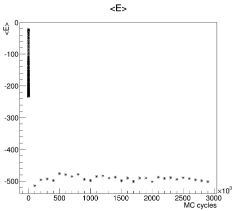
(c)



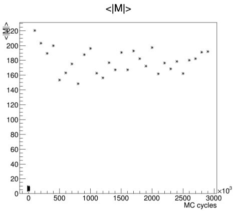
(d)

Figure: Statistical quantities for a 20×20 lattice at a temperature $T = 1/k_B$ with a steady-state initial state.

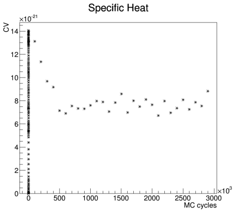
Temperature Dependence of MC



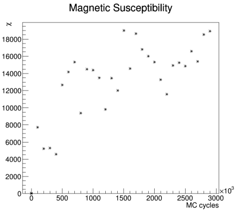
(a)



(b)



(c)



(d)

Figure: Statistical quantities for a 20×20 lattice at a temperature $T = 2.4/k_B$ with an initial state of a random selection of spins.

Temperature Dependence of MC

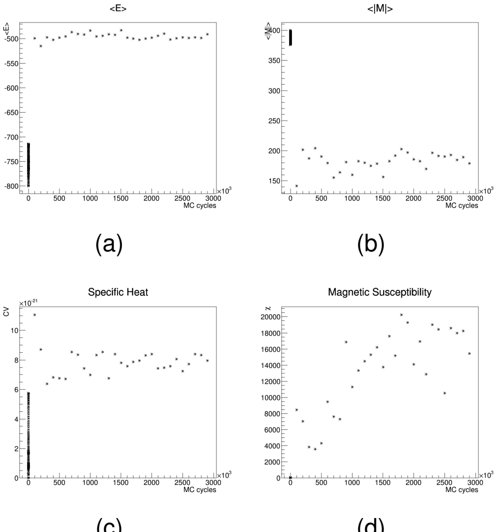
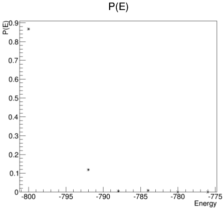


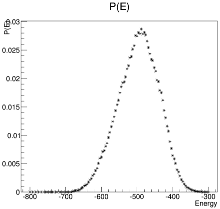
Figure: Statistical quantities for a 20×20 lattice at a temperature $T = 2.4/k_B$ with a steady-state initial state.

$P(E)$

- $P(E)$: number of times a given energy E occurs in the MC process divided by the number of cycles used.



(a)



(b)

Figure: $P(E)$ for a 20×20 lattice at a temperature of (a) $T = 1/k_B$ and (b) $T = 2.4/k_B$.

Critical Temperature

- Phase transition: when the matter undergoes a change of phase (eg. liquid to gas or solid to liquid, etc.)
  - Occurs at the critical temperature of the system, denoted T_C
- Near T_C, many physical quantities can be characterized by a power law exponent.
  - Mean magnetization:

$$< M(T) > \sim (T - T_C)^{1/8}, \tag{12}$$

  - Heat capacity:

$$C_V(T) \sim |T_C - T|^0, \tag{13}$$

  - Susceptibility:

$$\chi(T) \sim |T - T_C|^{\frac{7}{4}}. \tag{14}$$

  - Correlation length ξ (measure of how correlated the spins within the lattice are to one another) [5]:

$$\xi(T) \sim |T_C - T|^{-\nu} \tag{15}$$

Critical Temperature of an Infinite Lattice

- ξ scales with the size of the lattice, so we can use a finite lattice to approximate an infinite system

$$T_C(L) - T_C(L = \infty) = aL^{\frac{1}{\nu}} \tag{16}$$

for some constant a for a lattice of size L.

- By setting $T = T_C$, we can see [5]

$$< M(T) > \sim (T - T_C)^{\frac{1}{8}} \rightarrow L^{\frac{1}{8\nu}} \tag{17}$$

$$C_V(T) \sim |T_C - T|^{-\frac{7}{4}} \rightarrow L^0 \tag{18}$$

$$\chi(T) \sim |T_C - T|^0 \rightarrow L^{\frac{7}{4\nu}}. \tag{19}$$

- In order to study T_C, we look for indications of a phase transition in the graphs of $< E >$, $< |M| >$, C_V, and χ[1] for various lattice sizes.

[1] Here, χ is calculated with $< |M| >$.

Statistical Quantities Used to Estimate T_C

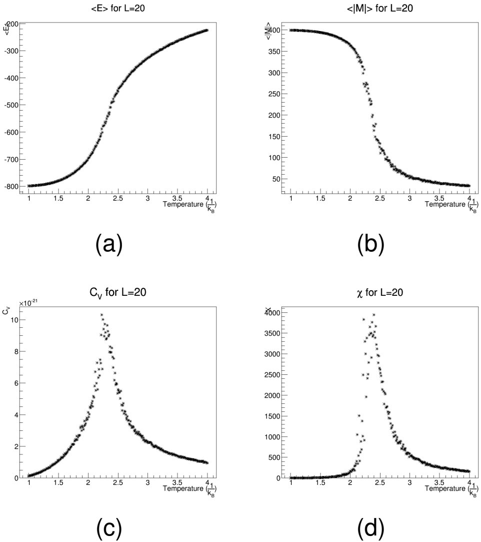
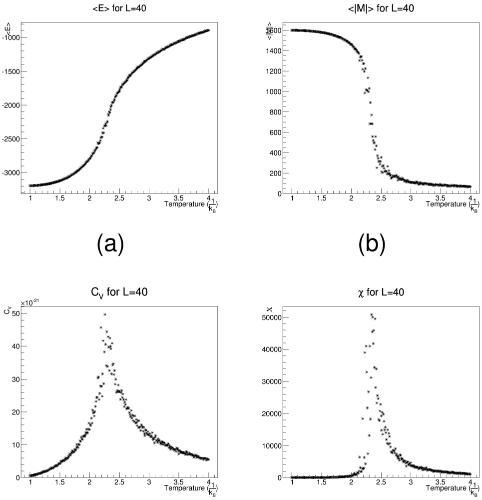


Figure: Statistical quantities for a 20×20 lattice with a steady initial state plotted against the temperature.

Statistical Quantities Used to Estimate T_C



(a) (b)

(c) (d)

Figure: Statistical quantities for a 40×40 lattice with a steady initial state plotted against the temperature.

Statistical Quantities Used to Estimate T_C

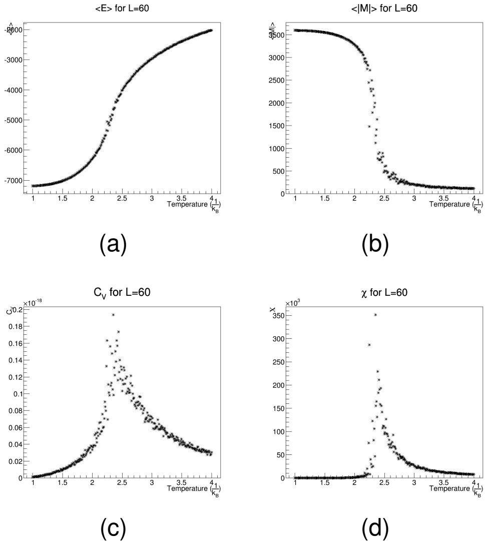


Figure: Statistical quantities for a 60×60 lattice with a steady initial state plotted against the temperature.

Statistical Quantities Used to Estimate T_C

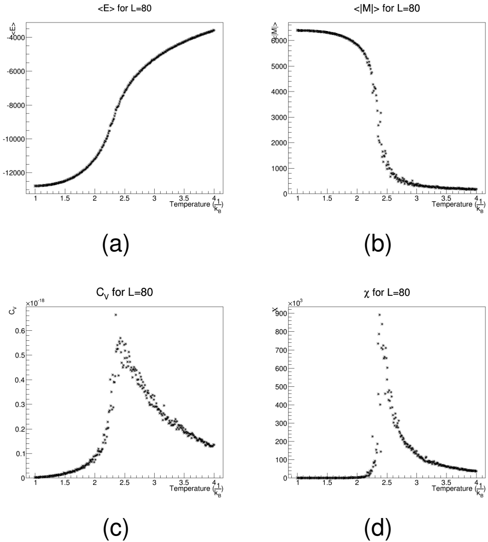


Figure: Statistical quantities for a 80×80 lattice with a steady initial state plotted against the temperature.

Estimating $T_C\,(L = \infty)$

- We wish to use these results to determine T_C in the limit $L \to \infty$, using $\nu = 1$
- Accepted value: $T_C = 2.269/k_B$ [5]
- To estimate, we need the information of $T_C(L)$.

L	T_C $(1/k_B)$ (from C_V)	T_C $(1/k_B)$ (from χ)	Average T_C $(1/k_B)$
20	2.25	2.35	2.30
40	2.30	2.35	2.33
60	2.35	2.40	2.38
80	2.40	2.45	2.43

Table: Approximate values for $T_C(L)$ for $L = 20, 40, 60,$ and 80 used with (16) to determine $T_C(L = \infty)$.

Estimating $T_C\,(L = \infty)$

- We know
$$T_C(L=\infty) = T_C(L) - aL^{-1},$$
for some $a \in \mathbb{R}$. So
$$T_C(L=\infty) = \frac{2.30}{k_B} - \frac{a}{20}$$
$$T_C(L=\infty) = \frac{2.33}{k_B} - \frac{a}{40}$$
$$T_C(L=\infty) = \frac{2.38}{k_B} - \frac{a}{60}$$
$$T_C(L=\infty) = \frac{2.43}{k_B} - \frac{a}{80}.$$
- Because (16) gives us 2 equations in 2 unknowns, and we are looking at four different lattice sizes, we can get any number of 8 possible solutions for T_C and a.

Estimating $T_C\,(L = \infty)$

- We find $\frac{2.35}{k_B} \leq T_C \leq \frac{2.575}{k_B}$ with error $3.57\% \leq \sigma \leq 13.49\%$ compared with the accepted result of $2.269/k_B$ [5]

Estimating $T_C\,(L = \infty)$

```
In[16]:= Clear[a, TC]

In[17]:= Solve[TC == (2.3 / kB) - (a / 20) && TC == (2.325 / kB) - (a / 40), {TC, a}]

Out[17]= {{TC -> 2.35/kB, a -> -1./kB}}

In[18]:= Solve[TC == (2.375 / kB) - (a / 60) && TC == (2.425 / kB) - (a / 80), {TC, a}]

Out[18]= {{TC -> 2.575/kB, a -> -12./kB}}

In[19]:= Solve[TC == (2.3 / kB) - (a / 20) && TC == (2.375 / kB) - (a / 60), {TC, a}]

Out[19]= {{TC -> 2.4125/kB, a -> -2.25/kB}}

In[20]:= Solve[TC == (2.3 / kB) - (a / 20) && TC == (2.425 / kB) - (a / 80), {TC, a}]

Out[20]= {{TC -> 2.46667/kB, a -> -3.33333/kB}}

In[21]:= Solve[TC == (2.325 / kB) - (a / 40) && TC == (2.375 / kB) - (a / 60), {TC, a}]

Out[21]= {{TC -> 2.475/kB, a -> -6./kB}}

In[22]:= Solve[TC == (2.325 / kB) - (a / 40) && TC == (2.425 / kB) - (a / 80), {TC, a}]

Out[22]= {{TC -> 2.525/kB, a -> -8./kB}}
```

Figure: Using Mathematica [6] to solve for the possible values of $T_C(L = \infty)$ given the data from Table 2 and (16).

Testing Random Numer Generators

The Tests

- Computer-algorithm random number generators are inherently deterministic, while random number sequences must be uncorrelated.
    - Pseudorandom number generators
- Tests were proposed by Kendall and Babington Smith [8] to check whether sequences of numbers were actually random: the frequency, serial, poker, and gap tests.

Frequency Test

- Goal: ensure every digit occurs approximately an equal number of times in the sequence [9, 4, 3].
  - Distribution is uniform
  - Every digit occurs in the sequence approximately 10% of the time

Serial Test

- Goal: check no two digits tend to occur in a particular order [9, 4, 3]
  - Like the frequency test, but ensuring all groupings of two digits occur approximately 1% of the time within the sequence

Poker Test

- Goal: make sure that certain possible combinations of numbers appear with approximately the correct probability.
- Procedure:
  1. Break digits into blocks of some fixed size (eg. 5 each)
  2. Compare with might be expected in a theoretical poker hand
- The expected results are shown in Table 3 [3]

Hand Type	Frequency (%)
5 of a Kind	0.01
4 of a Kind	0.45
3 of a Kind	7.2
2 of a Kind	50.4
Full House	0.9
Two Pair	10.8
Other	30.24

Table: Accepted frequencies for various poker hands [8].

χ^2 Test

- Goal: determine whether there is any correlation between two variables x and y.
- Procedure:
  1. Compute χ^2:
     $$\chi^2 = \sum \frac{(v - v_e)^2}{v}, \tag{20}$$
     where v is the value observed and v_e is the value expected [8]
  2. Determine the number of degrees of freedom of the problem, given by
     $$DF = (n_x - 1) * (n_y - 1), \tag{21}$$
     where n_x is the number of levels of variable x and n_y is the number of levels of variable y [1].
  3. Consult a χ^2 table (or, in our case, an online p-value calculator [2]) to determine the p-value given the number of degrees of freedom and the computation of χ^2. We look for p to be above a certain significance level, which we take to be 0.1 (although values of 0.01 and 0.05 are also common) [3].

Degrees of Freedom for χ^2 Test

Test	Degrees of Freedom
Frequency Test	9
Serial Test	90
Poker Test	4

Table: The degrees of freedom of the various tests used in this analysis [8].

ran0

- Minimal Standard Generator – which is a multiplicative congruential generator [10]
- Depends on the recurrence relation

$$I_{j+1} = aI_j \mod m \tag{22}$$

- Proposed by Park and Miller with the settings $a = 7^5 = 16807$ and $m = 2^{31} - 1 = 2147483647$.
- Period is expected to be $2^{31} - 1 = 2.1 \times 10^9$ [10].

ran0

Statistic	Value	Significance
μ	4.50159	0.04%
σ	2.87309	0.45%
χ^2 (Frequency Test)	5.68698	0.7708
χ^2 (Serial Test)	86.2657	0.5919
χ^2 (Poker Test)	10.0296	0.0399

Table: Results of the frequency, serial, and poker tests performed on `ran0`, and the significance of those results (the percent difference from the expected value in the case of the mean μ and the standard deviation σ, and the p-value for the χ^2 tests).

- ran1 uses ran0 for its random variable but also uses a shuffling algorithm developed by Bays and Durham to remove lower-order correlations:
  - A randomly generated element of the sequence, I_j, is chosen to be output as some other element of the sequence (typically I_{j+32})
- Period is expected to be $\sim 10^8$ [10].

Statsitsic	Value	Significance
μ	4.50043	0.01%
σ	2.87041	0.54%
χ^2 (Frequency Test)	7.19849	0.6165
χ^2 (Serial Test)	112.818	0.0522
χ^2 (Poker Test)	5.22794	0.2647

Table: Results of the frequency, serial, and poker tests performed on ran1, and the significance of those results (the percent difference from the expected value in the case of the mean μ and the standard deviation σ, and the p-value for the χ^2 tests).

ran2

- Claimed to be a ”perfect” random number generator (up to floating point errors) by [10], `ran2` adds another level of shuffling to further remove the correlations existing in `ran1`, as well as other randomness checks originally proposed by L’Ecuyer.
- Period is expected to be the least common multiple of the periods of the first two algorithms
  - Add the two sequences produced by `ran0` and `ran1` modulo the modulus m of either of them ($> 10^8$) [10]

ran2

Statsitsic	Value	Significance
μ	4.4999	$< 0.01\%$
σ	2.87382	0.42%
χ^2 (Frequency Test)	8.62035	0.4730
χ^2 (Serial Test)	93.7899	0.3714
χ^2 (Poker Test)	5.28678	0.2591

Table: Results of the frequency, serial, and poker tests performed on `ran2`, and the significance of those results (the percent difference from the expected value in the case of the mean μ and the standard deviation σ, and the p-value for the χ^2 tests).

ran3

- Originally presented by Knuth
- Based on subtractive methods – Fibonacci generator
  - The next term in the sequence comes from the subtraction of two previous terms.
    $$x_k = x_{k-a} - x_{k-b} \tag{23}$$
    for some fixed $a, b \in \mathbb{Z}$ known as lags.
  - Fibonacci generators are very efficient and pass most randomness tests thrown at them, although they do require more storage than linear congruential algorithms [7] (of which `ran0`, `ran1`, and `ran2` are special cases).

Statsitsic	Value	Significance
μ	4.50161	0.04%
σ	2.87307	0.45%
χ^2 (Frequency Test)	5.92085	0.7478
χ^2 (Serial Test)	106.29	0.1157
χ^2 (Poker Test)	7.28835	0.1214

Table: Results of the frequency, serial, and poker tests performed on `ran3`, and the significance of those results (the percent difference from the expected value in the case of the mean μ and the standard deviation σ, and the p-value for the χ^2 tests).

`rand()`

- Standard in the C++ library
- Linear congruential algorithm

Statistic	Value	Significance
μ	4.50236	0.05%
σ	2.87138	0.51%
χ^2 (Frequency Test)	7.25112	0.6110
χ^2 (Serial Test)	79.4787	0.7784
χ^2 (Poker Test)	3.55947	0.4689

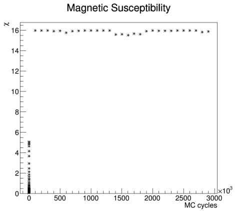
Table: Results of the frequency, serial, poker, and gap tests performed on `ran4`, and the significance of those results (the percent difference from the expected value in the case of the mean μ and the standard deviation σ, and the p-value for the χ^2 tests).

Effect on Physics

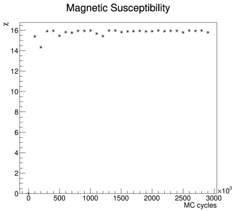
- Overall, we find that the choice of random number generator didn't have major contributions to the results determined. However, small differences were observed.[2]
  - Example, it was observed that, if we plotted the number of accepted MC against the total number of MC cycles used in the calculation, the slopes were slightly different. In particular, `ran1`, `ran3`, and `rand()` have more accepted MC than do `ran0` or `ran2`.

[2] Throughout this discussion, we are working with the generation of a 2×2 lattice with an initial random configuration at a temperature of $T = 1/k_B$.

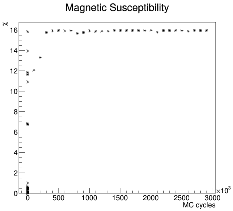
Effect on χ



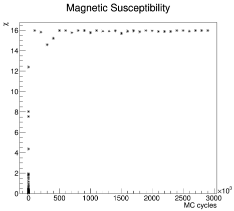
(a)



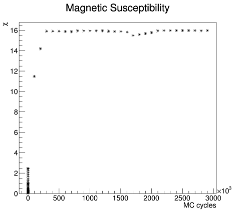
(b)



(c)



(d)



(e)

Figure: Magnetic susceptibility χ as a function of the number of MC cycles used in the calculation for (a) `ran0`, (b) `ran1`, (c) `ran2`, (d) `ran3`, and (e) `rand()`.

Conclusions

Conclusions

- Developed framework for implementation of Ising model in two dimensions
- Used framework to calculate and analyze various statistical quantities
- Used statistical quantities to estimate T_C for an infinite lattice to within 14% of the accepted value
- Tested various random number generators for randomness and effect on physics

Thank you!

Thank you to Morten Hjorth-Jensen for all of his help throughout this semester with this and the other projects.
Thank you to John Hall for allowing me to include the pseudorandom sequence in this analysis.
Thank you to Christie Campbell whose slide formatting I used for this presentation.

Questions

[1] Chi-square test for independence. web, 2016.

[2] P value calculator. web, 2016.

[3] E. Drueke. Pseudorandom sequences and random number generators. April 2016.

[4] F. Gruenberger. Notes. *Mathematics of Computation*, 4:244–245, 1950.

[5] M. Hjorth-Jensen. Computational physics lecture notes fall 2015, August 2015.

[6] W. R. Inc. *Mathematica 8.0*, 2010.

[7] D. Kahaner, C. Moler, and S. Nash. *Numerical Methods and Software*. Prentice Hall, 1989.

[8] M. G. Kendall and B. B. Smith. Randomness and random sampling numbers. *Journal of the Royal Statistical Society*, 101(1):147–166, 1938.

[9] T. G. Newman and P. L. Odell. *The Generation of Random Variates*, volume 29 of *Griffin's Statistical Monographs and Courses*. Hafner Publishing Company, 42 Drury Lane, London, WC2B 5RX, 1971.

[10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3 edition, September 2007.