Computational Analysis of Boltzmann Distributions in an Einstein Solid

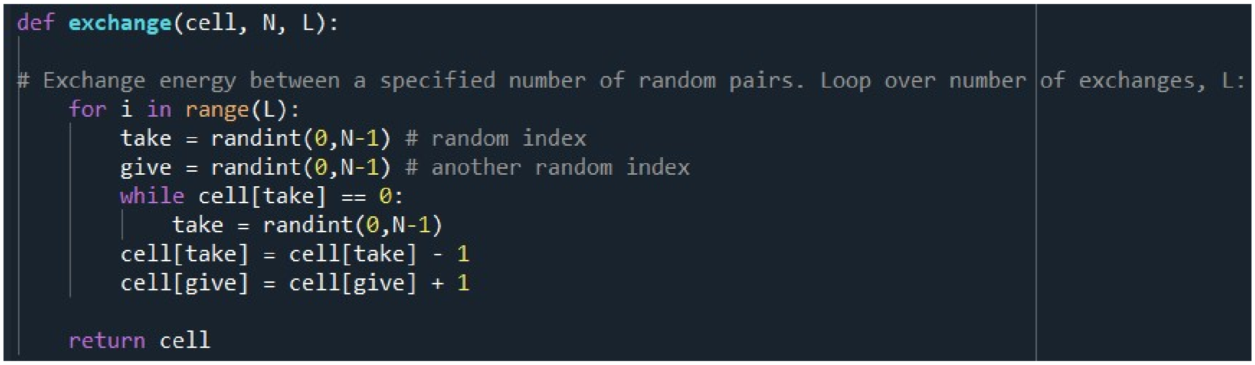
Chris McJunkin

University of Washington - Bothell

Introduction

In thermodynamics, an Einstein Solid considers an abstract object broken into individual cells containing some unitless energy. Each cell oscillates at the same frequency. If we consider an Einstein Solid with *N* cells each with an initial (average) energy of *q*, we can model how the solid evolves overtime, and see how the energy is distributed. Energy can be exchanged at random between each cell by taking one unit of energy and giving it randomly to another cell resulting in one exchange. Over several exchanges *L*, we can then find the probability that a cell will end up with an amount of energy. This probability tells us what the possible states the solid could be as a function of energy and follows a Boltzmann distribution (as we'll see later, the probability evolves from a Gaussian to a Boltzmann depending on the number of exchanges).

Our simulation works by creating a 1-dimensional array with the size of a perfect square (N must be a perfect square) to emulate behavior in a square solid in two dimensions. Since N is randomly selected, the dimensionality will not be much of a factor in deciding where the energy goes. However, some cleverness will be necessary if we wanted to later investigate the distribution if exchanges happened between neighbors.



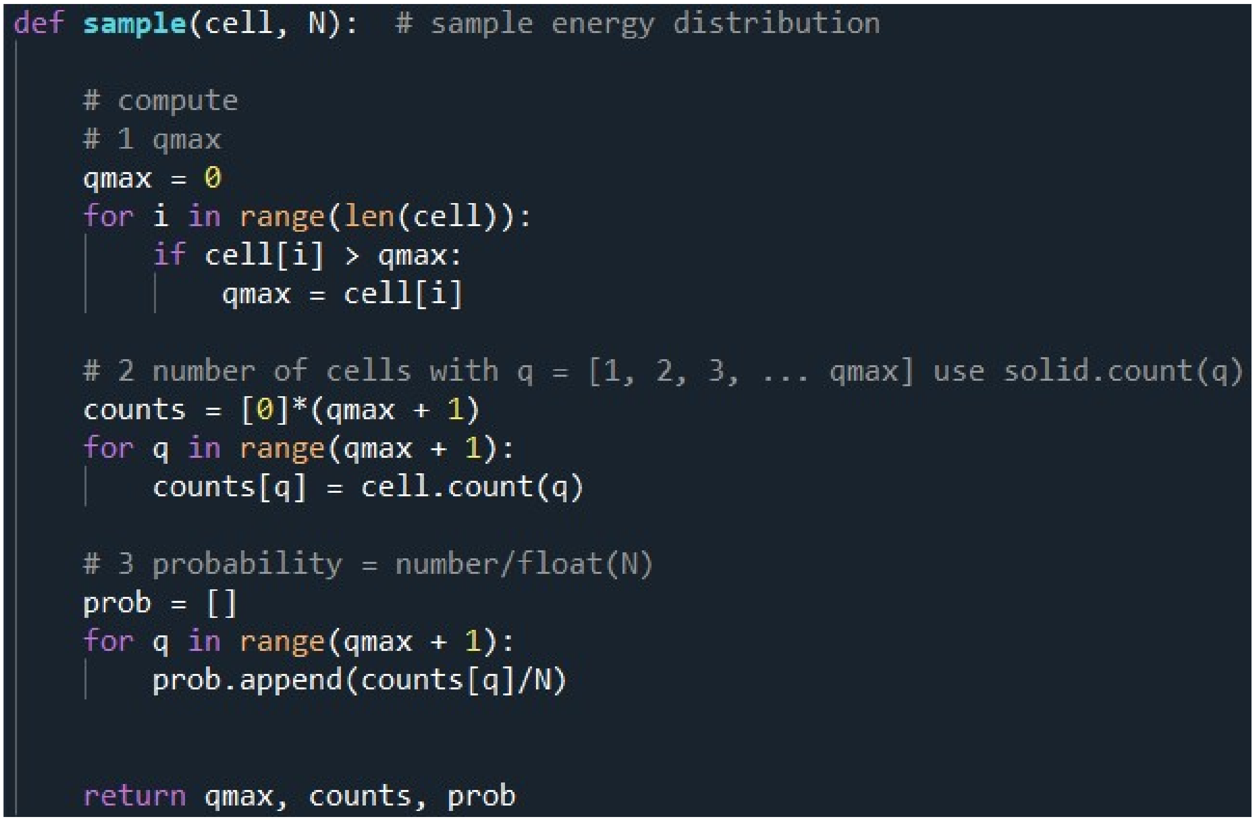
```
def exchange(cell, N, L):

# Exchange energy between a specified number of random pairs. Loop over number of exchanges, L:
    for i in range(L):
        take = randint(0,N-1) # random index
        give = randint(0,N-1) # another random index
        while cell[take] == 0:
            take = randint(0,N-1)
        cell[take] = cell[take] - 1
        cell[give] = cell[give] + 1

    return cell
```

Figure 1: Though we are passing our solid full of cells into our exchange function, our solid is referred to as cell. To avoid confusion, consider cell to mean all cells in our solid object.

Here, the program takes in our solid, and follows some loop rules to randomly distribute the energy. Note that the cell can't give up its energy if it doesn't have any energy to give, thus a new cell is randomly chosen. As L increases in magnitude, more processing time is required. Therefore, majority of simulations will be limited to 10^5 unless specified otherwise.



```
def sample(cell, N):  # sample energy distribution

    # compute
    # 1 qmax
    qmax = 0
    for i in range(len(cell)):
        if cell[i] > qmax:
            qmax = cell[i]

    # 2 number of cells with q = [1, 2, 3, ... qmax] use solid.count(q)
    counts = [0]*(qmax + 1)
    for q in range(qmax + 1):
        counts[q] = cell.count(q)

    # 3 probability = number/float(N)
    prob = []
    for q in range(qmax + 1):
        prob.append(counts[q]/N)


    return qmax, counts, prob
```

Figure 2: Sample function allows us to determine the number of times an energy level shows up, the maximum energy level, and the probability of that energy level to appear.

From here, we now have a new array full of the counts of energy, and an array of probabilities. A curve fit model can be implemented in a logarithmic scale due to the exponential appearance of the distribution. For our theory curve fit, we have the following Boltzmann Probability function P:

$$P = C\ exp[\frac{-q}{kT}]$$

Where C is a normalization constant, and q is iterated over the spectrum of no energy to maximum energy. We can use an estimate of C, by using the first probability entry in the probability array.

Since we do not know the temperature of the Einstein Solid, we will need to find the relation between the average energy (our starting value at equilibrium) $\bar{q}$ and kT*:*

$$\bar{q} = \frac{1}{e^{1/kT} - 1}$$

$$kT = \frac{1}{\ln\left(\frac{1}{\bar{q}} + 1\right)}$$

Not only is this useful for calculating the theoretical curve, but we now have a slope for the logarithm of the theoretical curve, -1/kT. Comparing this logarithmic slope to the slope obtained from a linear fit curve, an error between the data and theory can be found.

Varying Amount of Oscillators

The first run of simulations will compare different sizes of solids. If we are given a small object, with a small number N, we expect to see the distribution of energy to mimic a qualitative “random”, in that there are not many cells to transfer energy to. In contrast, a larger N should correlate to a “truer” representation of a Boltzmann distribution because there are many options for the cells to distribute energy. This means we expect to see high probabilities for lower amounts of energy with large N, and roughly equal amounts of probabilities for smaller N. We should also anticipate a steady trend as N increases for the Boltzmann Probability Curve to develop.

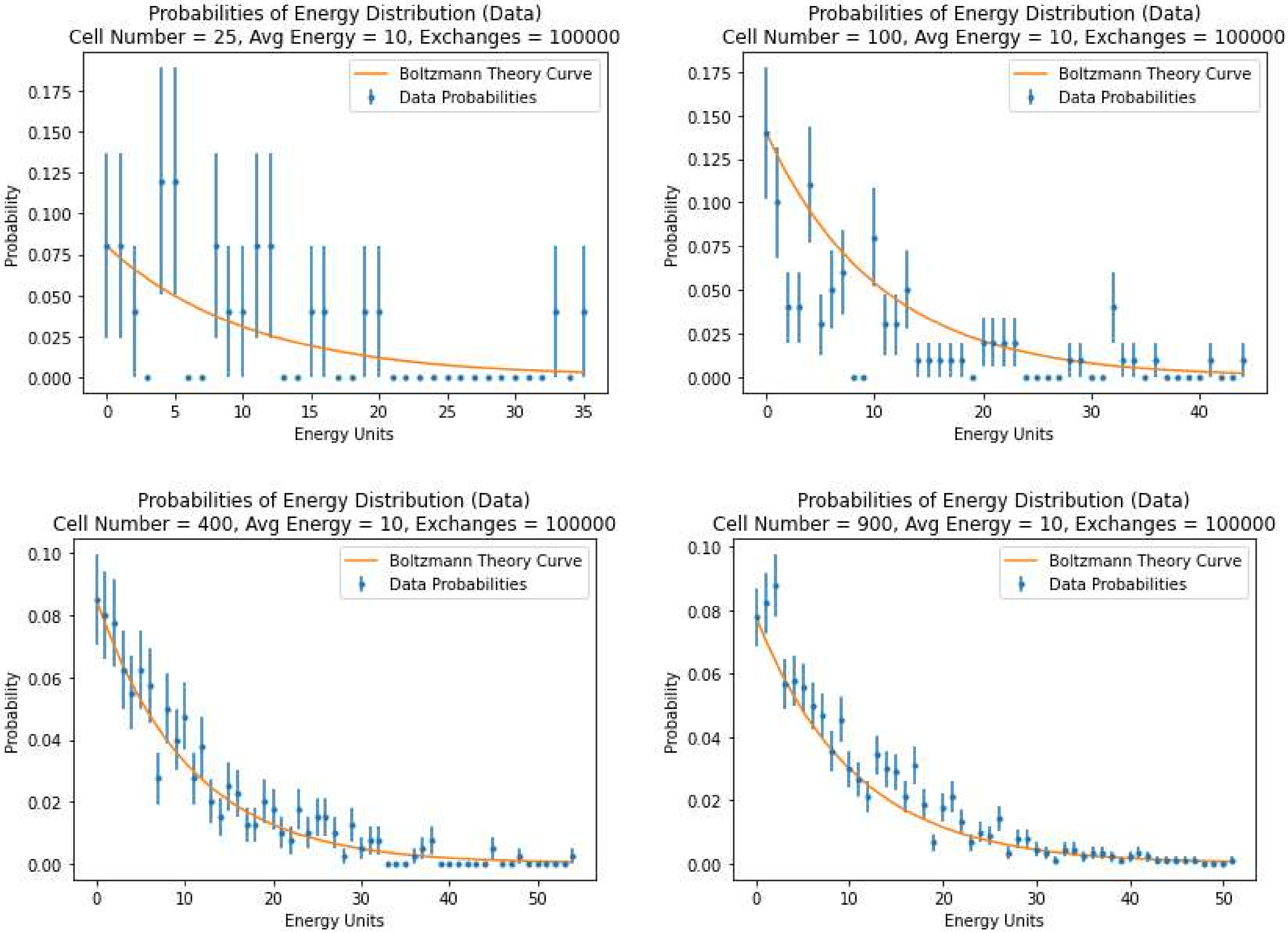


Figure 3: Probabilities of N = 25, 100, 400, 900 starting top left, plotted against the obtained Boltzmann curves.

Plotting the theory against our data for different number of N, we do indeed see a trend towards a Boltzmann distribution. But caution should be heeded as N increases, since we'll need to increase the number of exchanges happening to allow the oscillators enough time to settle.

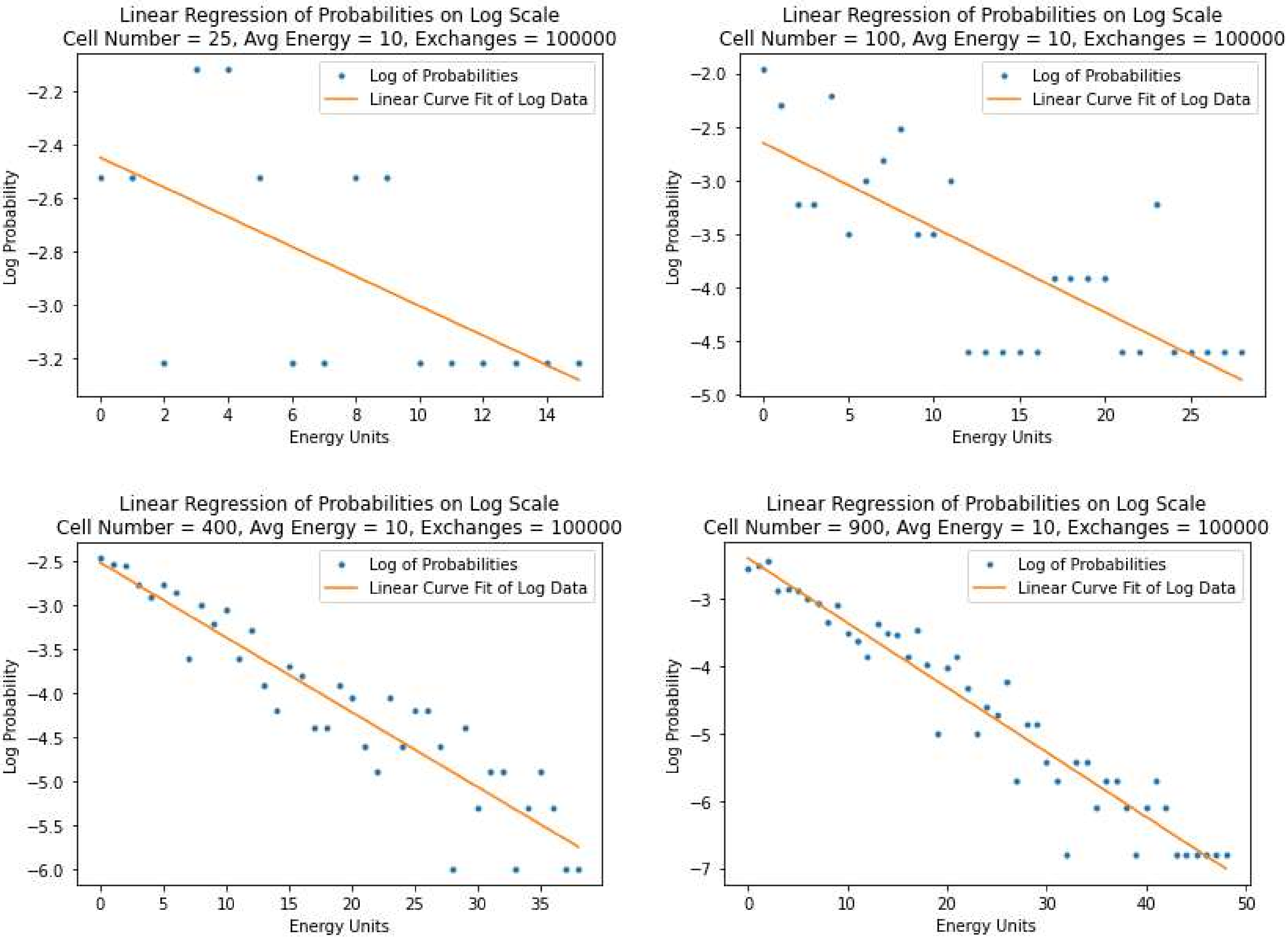
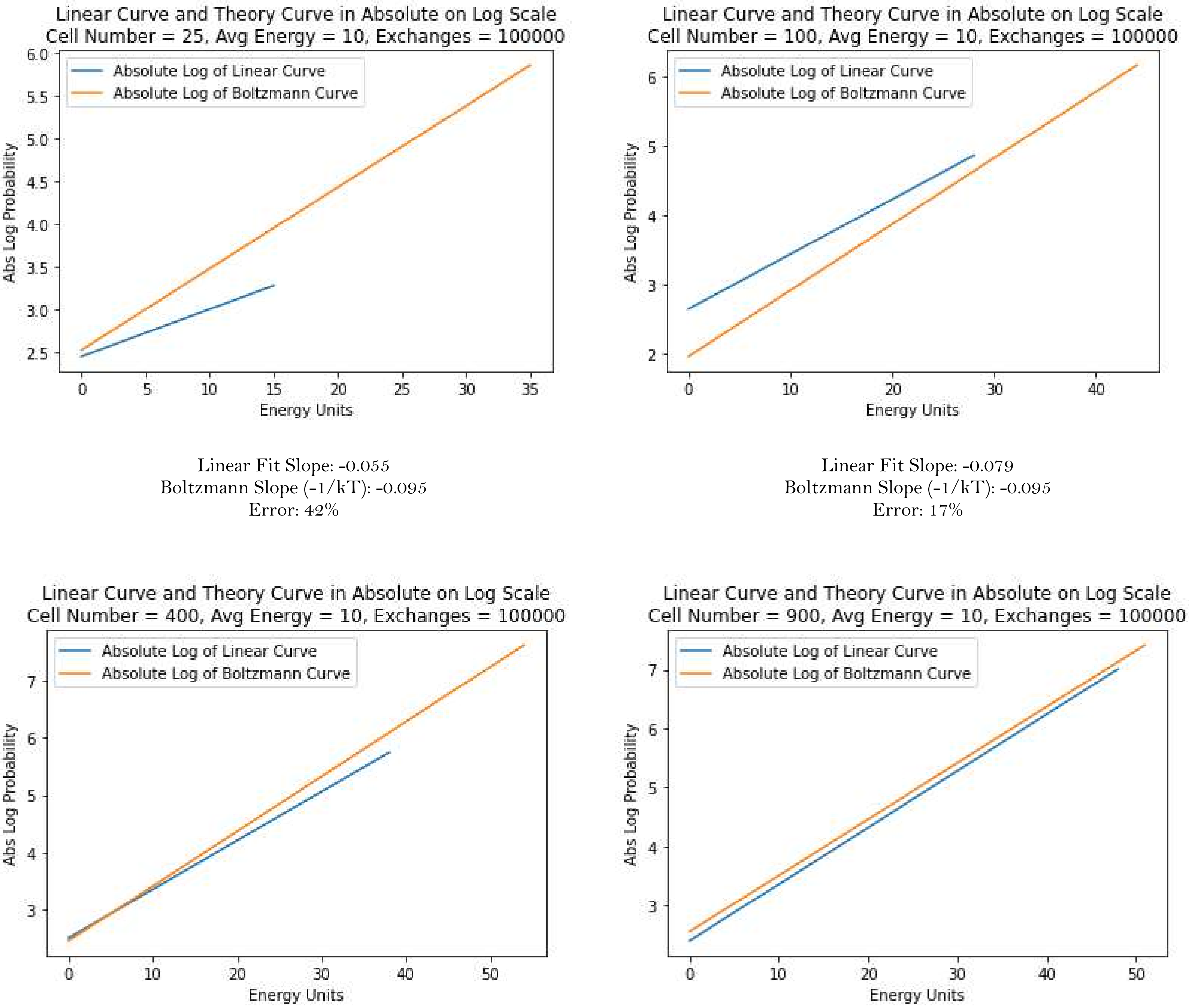


Figure 4: N = 25, 100, 400, 900 starting from top left, plotted against their respective linear regressions.

A linear regression is also taken of the logarithm of the acquired probabilities. The trend remains true here as with figure 3. Interesting to note that as N increases, the groupings of probabilities tend closer to the linear regression.



Linear Curve and Theory Curve in Absolute on Log Scale
Cell Number = 25, Avg Energy = 10, Exchanges = 100000

Linear Fit Slope: -0.055
Boltzmann Slope (-1/kT): -0.095
Error: 42%

Linear Curve and Theory Curve in Absolute on Log Scale
Cell Number = 100, Avg Energy = 10, Exchanges = 100000

Linear Fit Slope: -0.079
Boltzmann Slope (-1/kT): -0.095
Error: 17%

Linear Curve and Theory Curve in Absolute on Log Scale
Cell Number = 400, Avg Energy = 10, Exchanges = 100000

Linear Fit Slope: -0.085
Boltzmann Slope (-1/kT): -0.095
Error: 11%

Linear Curve and Theory Curve in Absolute on Log Scale
Cell Number = 900, Avg Energy = 10, Exchanges = 100000

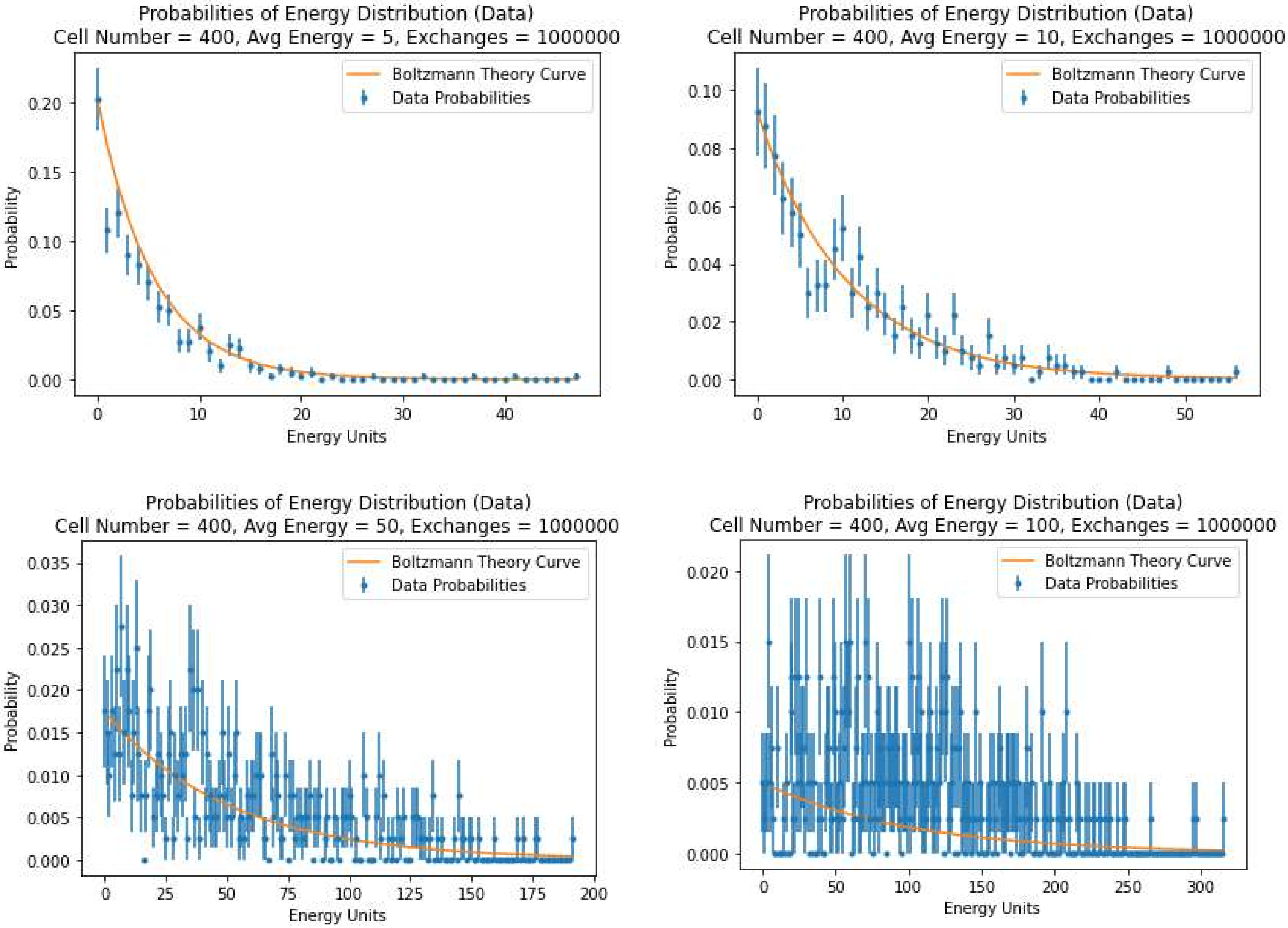
Linear Fit Slope: -0.096
Boltzmann Slope (-1/kT): -0.095
Error: 1%

Figure 5: Linear regressions plotted against the Boltzmann curve with their respective slopes.

We can see the error of small N=25 is 42%. With larger N=900, error is 1% heavily indicating the larger the system, the higher the probability of the oscillators in the system to have lower energy.

Varying the Average Energy

Next, we can adjust the initial equilibrium energy per cell in each system. As per the equation for P, making our substitution for kT, the probability of the energies should grow as the average energy grows, and tends away from a Boltzmann curve. For these series of simulations, a standard of N = 400, L = 10^6 will be used.



Probabilities of Energy Distribution (Data)
Cell Number = 400, Avg Energy = 5, Exchanges = 1000000

Probabilities of Energy Distribution (Data)
Cell Number = 400, Avg Energy = 10, Exchanges = 1000000

Probabilities of Energy Distribution (Data)
Cell Number = 400, Avg Energy = 50, Exchanges = 1000000

Probabilities of Energy Distribution (Data)
Cell Number = 400, Avg Energy = 100, Exchanges = 1000000

Figure 6

If we interpret the energy as temperature instead, its interesting to note that as the temperature increases, chaos, or *entropy* seems increases. Which is what we expect from thermodynamics. The Boltzmann curve no longer holds true, and the scale of probabilities decreases significantly.

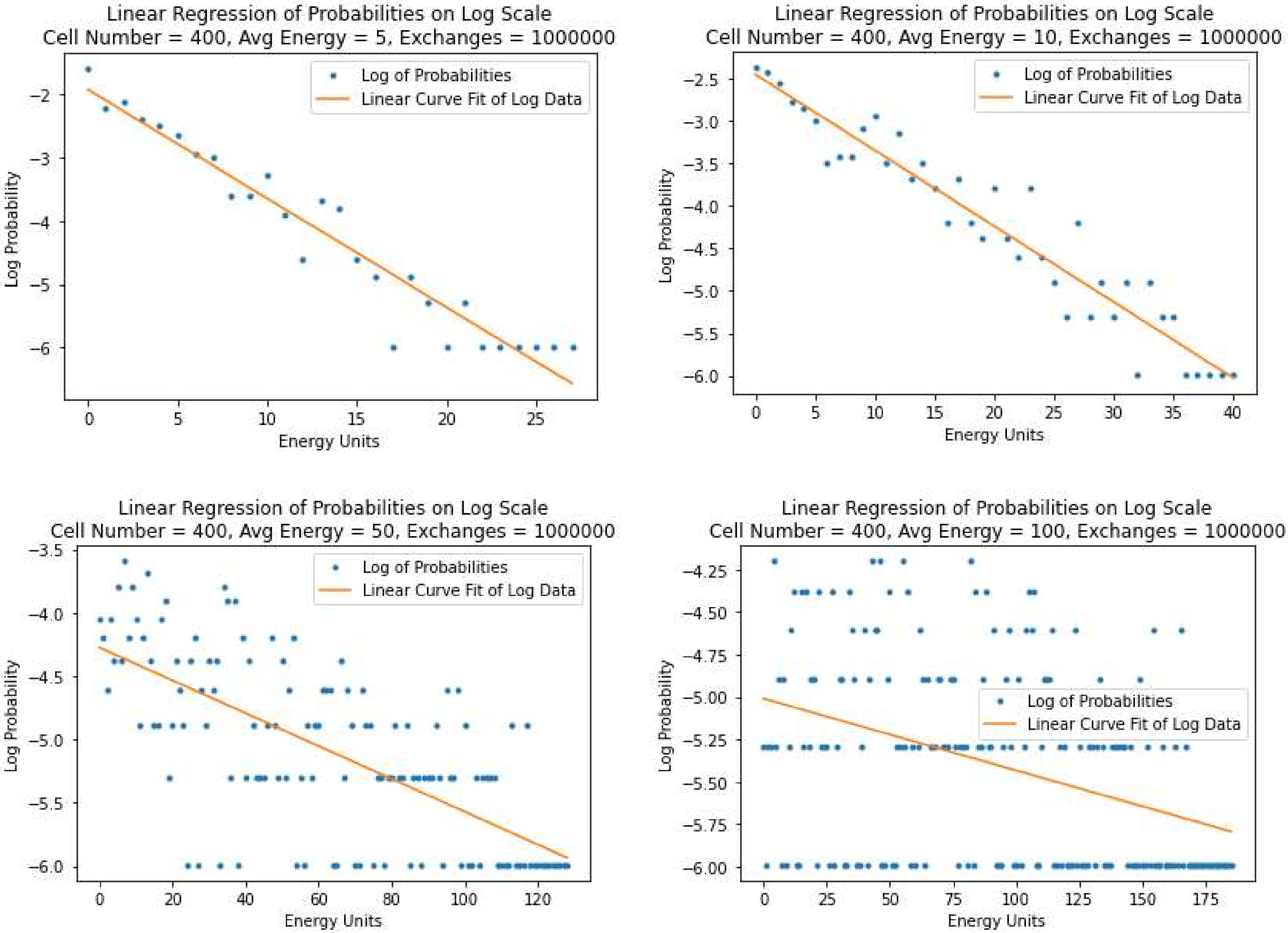
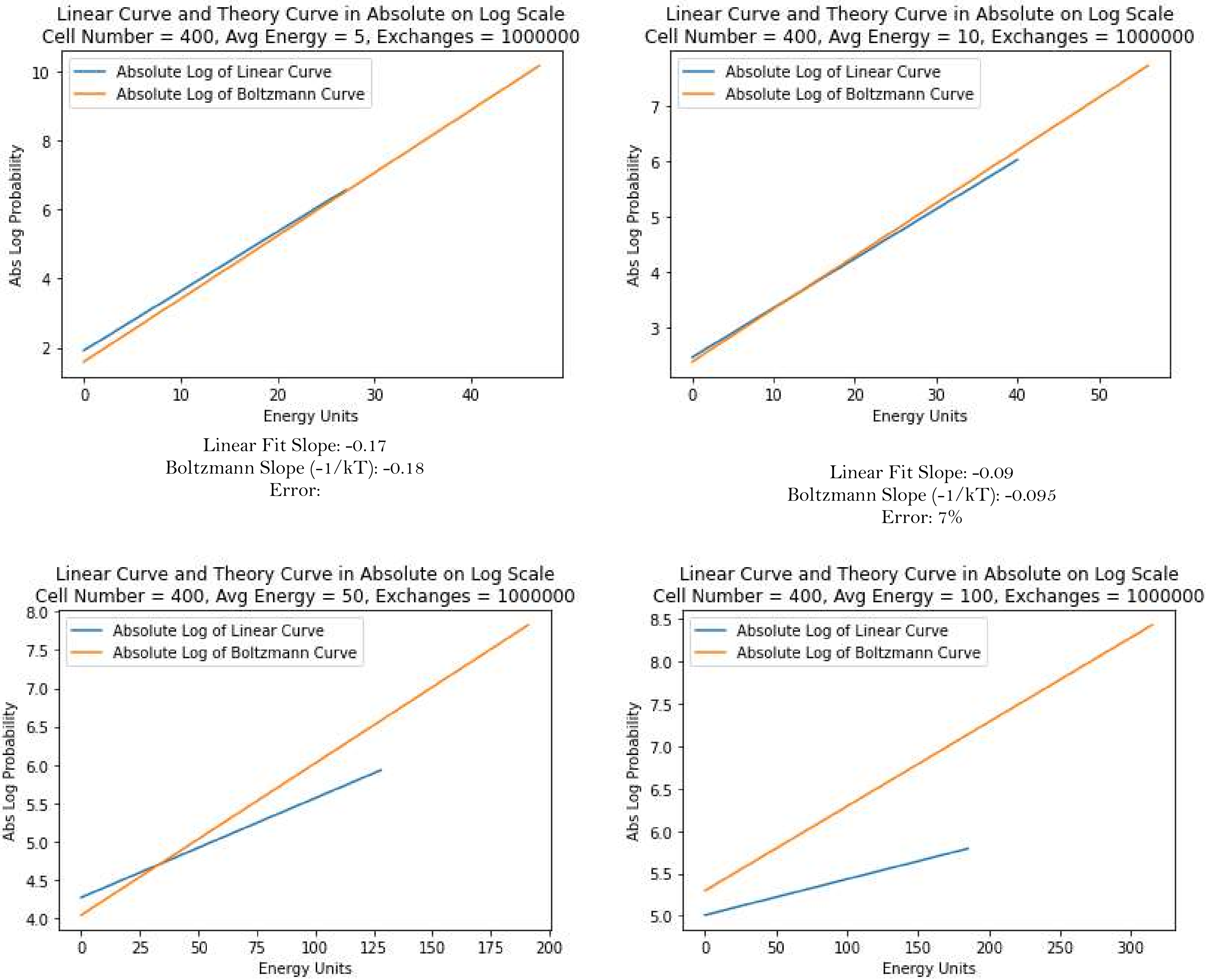


Figure 7

If we try to make a linear fit, the data starts to resemble random noise and no longer holds to a tight fit. With low amounts of energy, we expect the probabilities to hold to the linear curve. This means that there isn't enough energy to pass around.



Linear Fit Slope: -0.17
Boltzmann Slope (-1/kT): -0.18
Error:

Linear Fit Slope: -0.09
Boltzmann Slope (-1/kT): -0.095
Error: 7%

Linear Fit Slope: -0.013
Boltzmann Slope (-1/kT): -0.0198
Error: 35%

Linear Fit Slope: -0.004
Boltzmann Slope (-1/kT): -0.001
Error: 57%

Figure 8

The Boltzmann slope decreases because we are increasing the energy, in turn increasing kT. But the slope of the linear fit is decreasing at a faster rate, as seen in the bottom right in figure 7. So, we do see the more energy the system starts with, the more difficult it is to make a prediction where we might fight different energy amounts, trending towards a constant linear curve.

From Gaussian to Boltzmann

As q and N vary, the behavior of the distribution tends away from Boltzmann. In figure 3, we can kind of see a trend away when N = 900. If we have a large enough system not able to run enough cycles, then the distribution should take on another form. That form is Gaussian. Both Gaussian and Boltzmann distributions are exponential in nature and rely on an average. In the case of Boltzmann, our average is already established as kT.

We can investigate this in a few different ways, by either increasing N, or starting at a low L and increasing. In the case of increasing N, we can see the Gaussian take shape due to not enough oscillators exchanging energy. For low L, Gaussian takes shape.

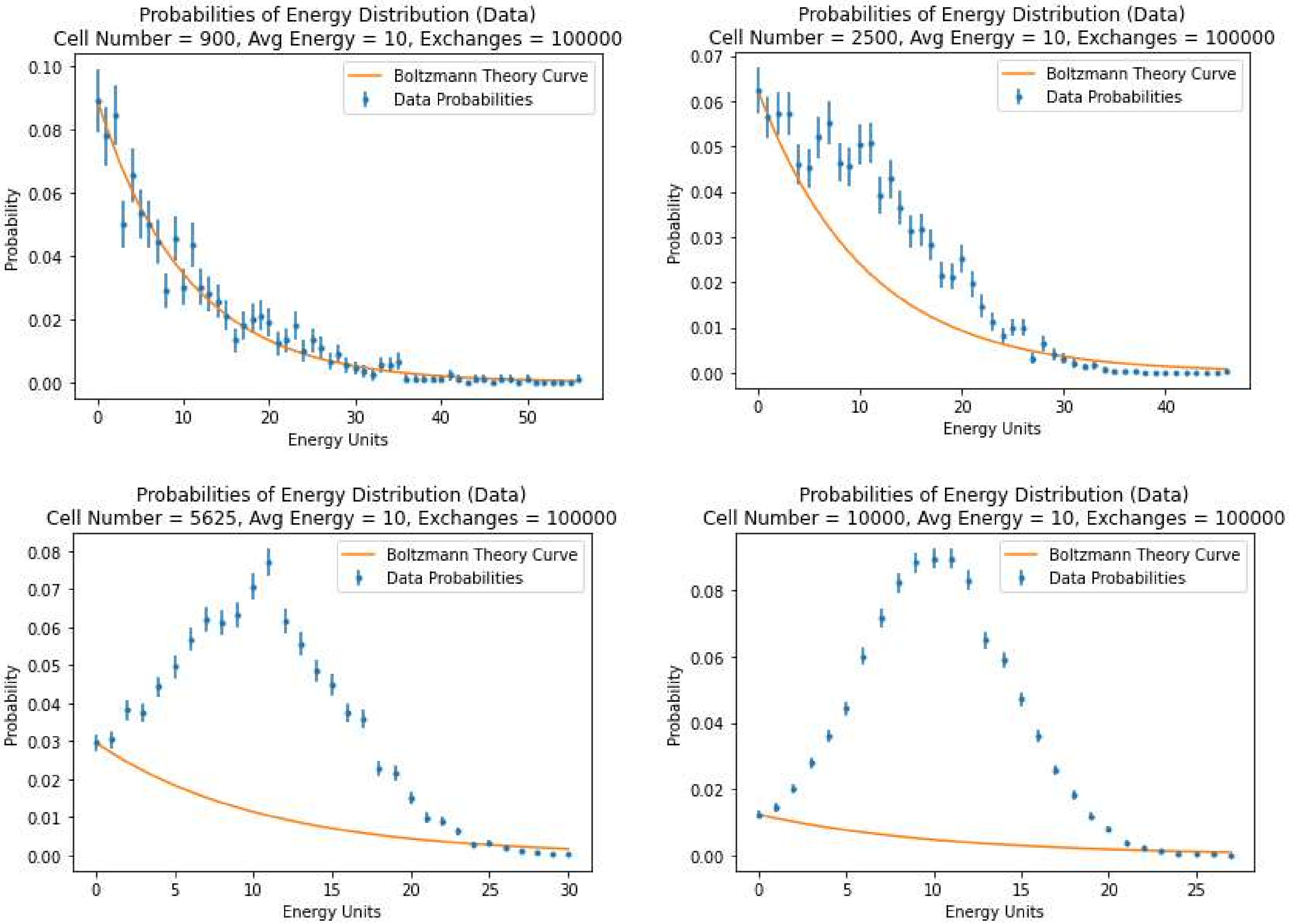


Figure 9. Starting from N = 900, Gaussian develops centered around average energy. The theory curve is what we expect given enough runs for the size of N.

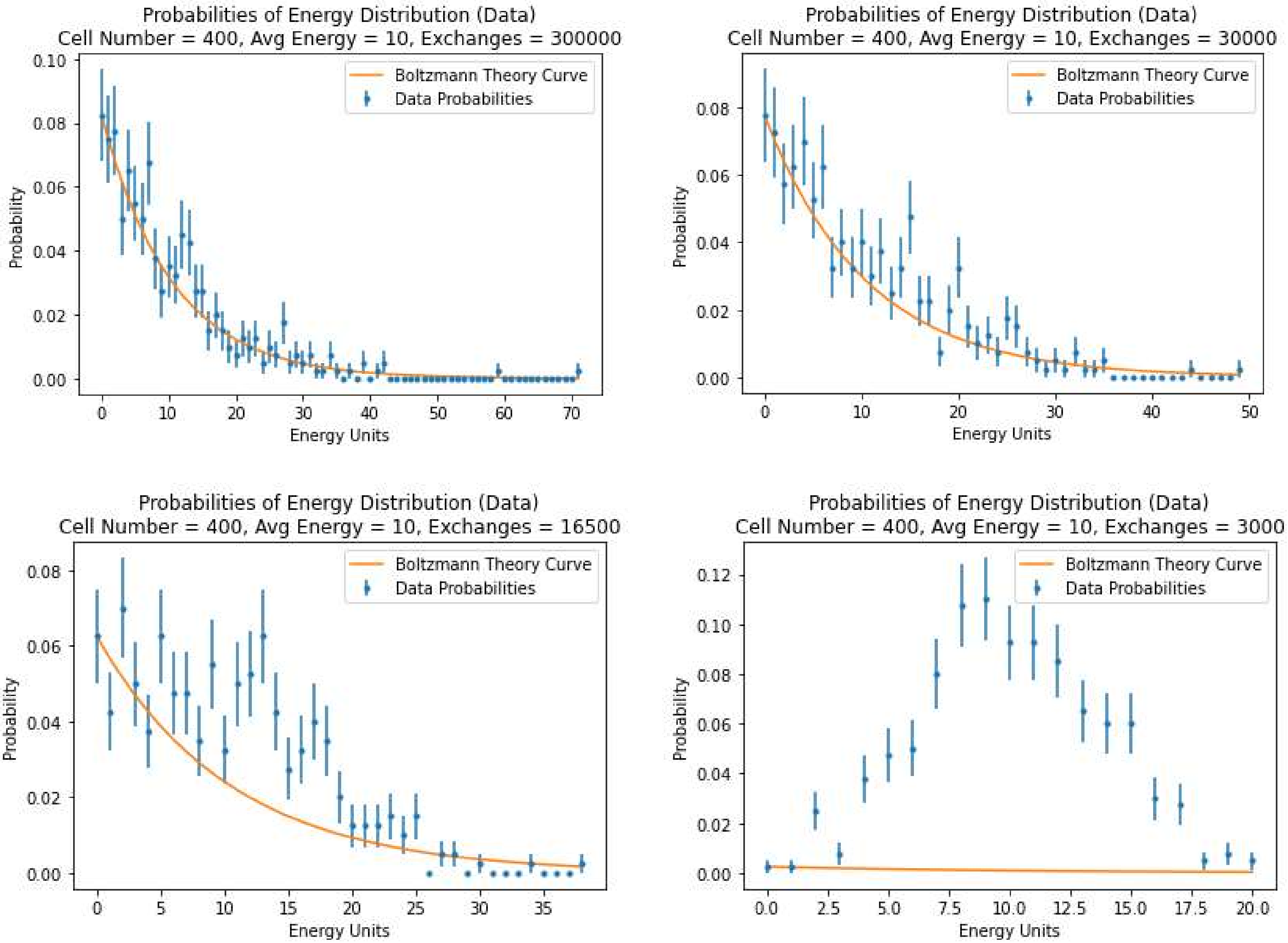


Figure 10. Exchanges from N = 400. Bottom left, a deviation can be observed to start forming, leading to the bottom right. Bottom left is running at a cycle between L = 30000 and L = 3000

Discussion

The results can be observed in figures 9 and 10. An increase in N approaches Gaussian faster than lower L. Future experiments could reveal the relationship between the number of oscillators or cells in a system to a correct exchange rate to reliably get a Boltzmann relationship. From our results, we can see that as a system grows larger, there are many more exchanges of energy (or temperature) to happen. For an extreme case, the universe is a system of an unfathomably large amount of particles, each with their own individual temperatures, and as a result, would require a very large number of exchanges to be able to predict where we might be able to find particles at a particular temperature. Thus, this implies a chaotic system. Looking at a smaller system, this is not the case.

Another way we could apply this is to consider which non-physical systems may hold true to the model of "oscillators with energy", ranging from economical, to biological. All that would need to happen is for our exchange function to change to portray these non-physical systems accurately. From there, we can determine the levels of chaos and stability in those non-physical systems.