Monte Carlo Simulation of Electron Stripe/Bubble Formation in Graphene

Mack Marshall, Calvin Pozderac, Carlos Lima, Brian Skinner

Center for Emergent Materials, The Ohio State University

Abstract

Twisted bilayer graphene is a fascinating material with much interest in many fields, though it especially gained notoriety after reported superconducting behavior. This led to the observance of an interaction potential that appears to have a small ”shoulder” in its shape. In this report, I explore the possibility of the formation of stripe and bubble electron phases based on both a simplified toy model of the shoulder potential, and the aggregation of the real potential and an external potential field. This was accomplished through the use of a Monte Carlo simulation written in Python, facilitated by Jupyter notebooks. All provided analysis is on a primarily qualitative basis.

I Introduction

Twisted Bi-layer Graphene

Graphene has been of some interest to physicist and materials scientists for some time. More recently, twisted bi-layer graphene - a configuration where the upper of two layers of graphene is twisted slightly - has gained special interest due to its flat band structure and demonstration of superconductivity [1]. This excitement led to further research, and the derivation of the Law of Interaction for gapped graphene by Kotov et al. in 2008. This interaction potential is plotted in Figure 5.

Electron Stripes and Bubbles

Typically, when modelling the Law of Interaction for electrons the plot looks a continuous downward-sloping line, similar to Figure 1a. However, in the presence of a magnetic field, this interaction plot develops a slight ”shoulder” to it. That is to say, for some range of distances, r, there is no increase or decrease in the interaction energy. The presence of this plateau enables the formation of stripes and bubbles, described by Fogler et al. in 1996 [2]. This shoulder is purely an artifact of the magnetic field, as it turns the wave functions into rings, which have a maximum partial overlap until they are completely overlapped. A simplified example of this interaction can be seen in Figure 3

Motivation and goals

The appearance of a small shoulder-like anomaly in the plot of the interaction potential for gapped graphene, first derived by Kotov et al. in 2008, gave rise to the suspicion that gapped graphene could similarly produce these stripe and bubble formations [3]. This project was undertaken in an attempt to explore whether or not this would in fact be possible with just the gapped graphene potential, and also in the presence of an external potential caused by the substrate. This exploration was undertaken by building a Monte Carlo simulation of electrons on a 2D plane in Jupyter Notebooks, and exploring both the simplified interaction models and the gapped graphene Law of Interaction

II Method

In order to explore the possibility of differing electron morphologies, Jupyter notebooks was used to construct a python program which could simulate a random assortment of points - representing electrons - and their motion around a 2D plane. In the following code described here, the only imported Python packages neccesary are numpy and matplotlib (for the generation of random numbers/arrays, and the output of usable plots respectively).

Monte Carlo Simulation

The basis of this simulation work is the Monte Carlo simulation. At its simplest, the Monte Carlo simulation is about producing random points within a specified boundary and analyzing something about those points - the most popular example being calculating the value of pi based on the ratio of points inside and outside of a circle within a square boundary.

Toy Model

In order to begin to explore the real proposed problem, we began by constructing a simulation with a simpler, more clearly defined structure, deemed the "toy model". To begin with, a series of functions were defined which could be called for different simulation runs. First, a function to calculate the interaction potential, V(r), was created. For the toy model, this was either a continuous function $\frac{1}{r^3}$, or a function of $\frac{1}{r^3}$ on either side of a flat line between two values of r, r_1 and r_2.

The purpose of these two basic V(r) functions was to test that the simulation was behaving according to expectations based on the math and different input parameters.

Next, a function to calculate and return the energy value of the initial, randomized state was constructed. This function accepts the series of points as an array. Then, a nested for-loop was written to accept each point in the array, calculate its distance from each other point, and pass that distance into the V(r) function. Each returned energy value was then added to the previous one to create a total energy value.

A similar function was constructed in order to calculate the change in energy associated with changing the position of a specific point. For this to be accomplished, the index of the changed point is passed in and its initial energy contribution is calculated. Then, the energy contribution of the new point position is calculated. The change between these values is added to the original energy value, and then passed back for evaluation.

Once these functions were constructed, a main block of code had to be established to use the functions to run the simulations. For this, several variables were defined and initialized, which can be seen in Table 1.

After initiating a set of points, they are fed into the function which calculates the initial energy value of the entire system, which can be used in the following evaluation to determine whether or not point moves are accepted later on.

Next, a for-loop was constructed to run for the length of *num nudge*. In each iteration, a random number from 0-($num - 1$) was generated, which corresponded to the index of a point within the array, points. Then, to simulate random motion, two more random values are generated, corresponding to a radius and an angle. This creates a vector, which can be broken into two x and y values, which can be added to the point at the selected index. Once this new position has been calculated, the function written to update the energy value can be called to produce a "U new" variable, which can be compared to the current energy value. If the new energy is lower than the current energy, the change is automatically accepted. In order to incorporate the effects of temperature, a Boltzmann distribution is used to to determine if a higher energy state will be accepted. The probability is:

$$\exp(\frac{-1 * \Delta U}{T * E_{typ}}) > r \tag{1}$$

where ΔU is the change in energy, T is the arbitrarily assigned 'temperature' value, E_{typ} is the interaction energy between two points at r_{typ}, and r is a randomly selected value between 0 and 1. At 0 T, this simplifies to only accepting moves which lower the energy of the system. Being able to temporarily raise the energy of the system is desirable in cases where metastable states can arise, which require an increase in energy to move into the true equilibrium state.

Table 1: Table containing useful variables and their descriptions.

Variable	Purpose/Meaning
L	Side length of the 2D square plane, boundary for points
num	Number of points to be randomly assigned to positions within the plane
num nudge	Number of random moves to be performed, can think of as analogous to 'how long' the system is allowed to settle; usually fixed as a scalar $\times$ num
points	A num $\times$ 2 array of random points, basically a list of x and y points
r typ	The 'average' distance between nearest-neighbour points on the plane, assuming even distribution
T	The 'temperature' of the system, which is in arbitrary units but used to simulate the effect of raising and lowering the system energy, as one would do by changing temperature
V_0	Used to assign the strength of an external potential
k_0	Used to determine the frequency of an external potential
alpha	Used to manipulate the appearance of the shoulder potential experienced by the interaction of the two planes

Extension to Real-World Potentials

Once the toy model was established and tested to determine that it worked properly, the V(r) formula was edited to reflect the relationship from the twisted bi-layer graphene which is graphically represented in Figure 5. This formula contains a variable, α which is used to manipulate the position of the right-hand of the shoulder, and the prominence of the shoulder itself. In short, larger values of α lead to a wider, more prominent shoulder in the potential function.

Addition of a Potential Field

Finally, another function was written to add the energy of the external potential to each point during the energy calculations. This is where the V_0 and k_0 are implemented in order to manipulate the electron formations. The equation used to factor this in was:

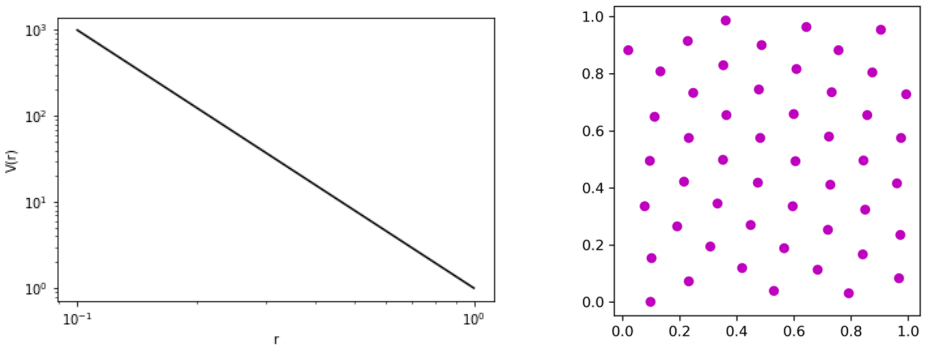
$$u = V_0 \cdot sin(kx) \cdot sin(ky) \tag{2}$$

Where k = n·π/L, and V_0 is an arbitrarily assigned field strength.

III Results and Discussion

Toy Model Plots

As expected, when the potential is established to be consistently repulsive without a shoulder, the result is a Wigner crystal structure at 0 K. One can imagine overlaying the pictured structure with three different lattice lines, whose overlapping would produce equilateral triangles.



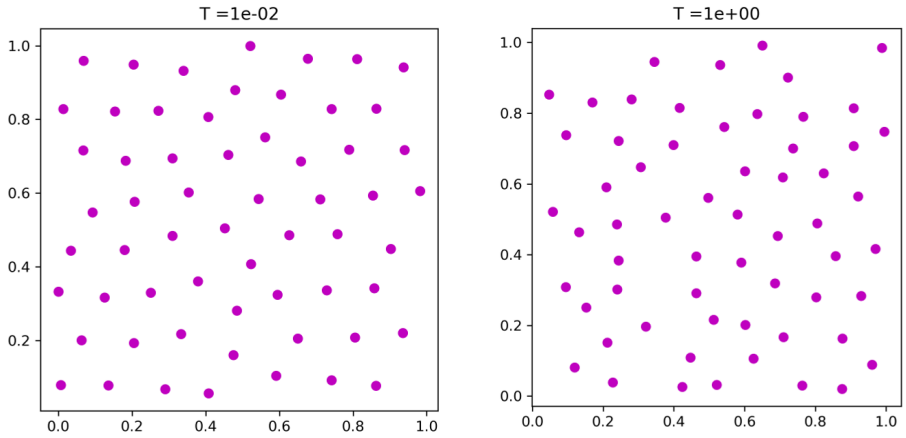
(a) Plot of V(r), on log-log scale, when the potential is simplified to $\frac{1}{r^3}$ for the toy model.

(b) Electron arrangement following the $\frac{1}{r^3}$ potential. At 0 K, this forms a recognizable triangular lattice.

Figure 1: Plot of the simplified potential and resultant e^- lattice at 0 K

In addition, the incorporation of an arbitrary temperature unit produces an expected effect. Here it can be seen that raising said temperature increases the disorder of the system, which is analogous to melting a crystal structure. A melting temperature of the system seems to appear at T=1E-2 K, so this temperature value was subsequently used to encourage the formation of non-Wigner morphologies (which might otherwise be prevented by the metastable nature of the Wigner crystal structure).

When the toy model was adjusted to create an artificial shoulder potential, it was able to produce stripes and bubbles. In this small system size, the density of the points determined whether stripes, bubbles, or Wigner structure would emerge. At extremely high densities, there simply was not enough space for the electrons to sort themselves into clusters. At low densities, there was no energetic payoff for this clustering either. The effect of temperature on these formations was also briefly explored. The established melting temperature was used in many cases to reduce the possibility of the system being caught in a metastable state.



(a) Simulated structure of the e^- around the assigned T_{melt} of the system.

(b) Simulated structure of the e^- at a temperature much greater than T_{melt} of the system.

Figure 2: Raising the value of the arbitrary temperature input created disorder in the resulting e^- structure.

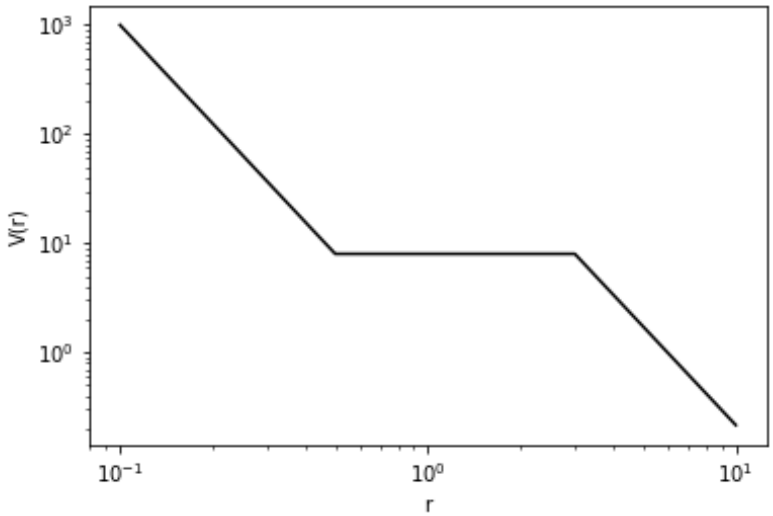
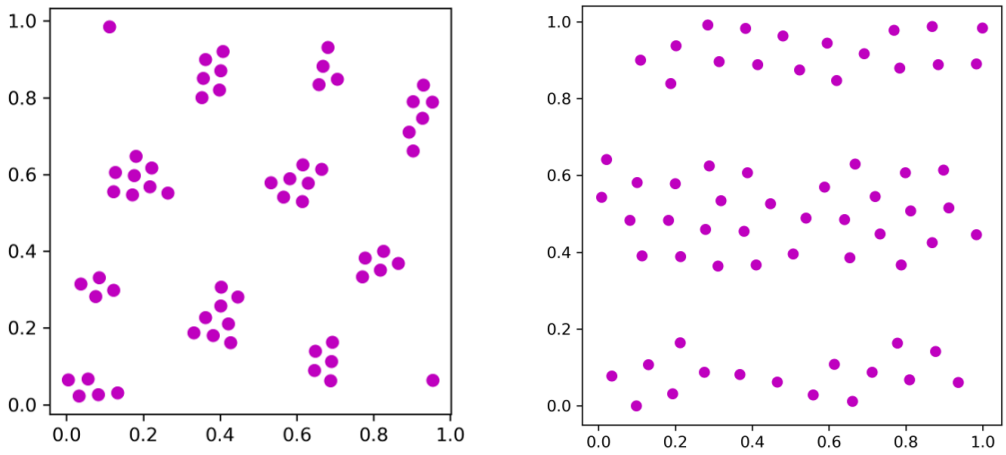


Figure 3: Plot of simplified V(r) plot with shoulder potential present. Diagonal portions trend down at the same $\frac{1}{r^3}$ rate as the previous potential plot.

The most important facet of this exploration was the connection to the shape of the potential plot. As may be expected, the bubbles and stripes had widths and inter-formation spacing of roughly the same width as the shoulder in the potential.

This information was valuable in addressing the next step of the simulation, wherein the gapped graphene potential was added for analysis.



(a) Bubble formation of electrons caused by shoulder potential.

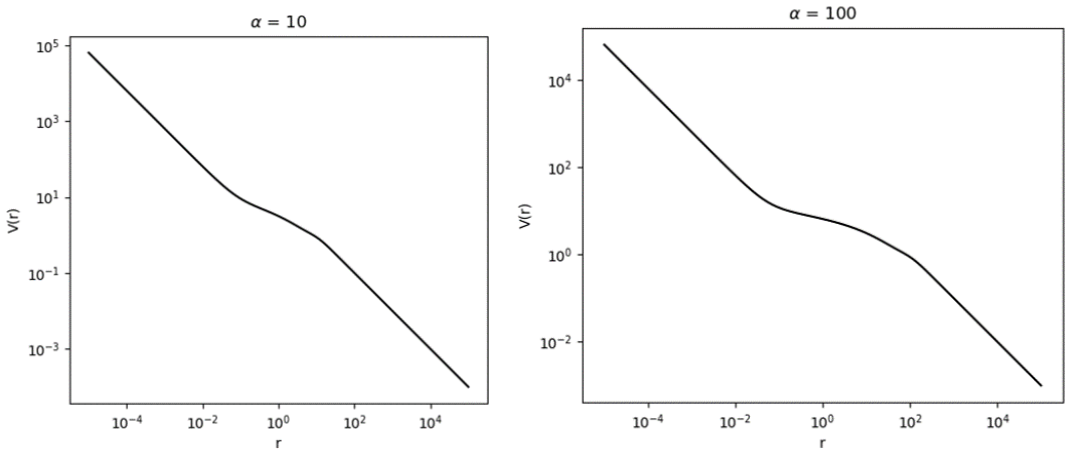
(b) Stripe formation of electrons caused by shoulder potential

Figure 4: Bubble and stripe formations caused by the toy model shoulder potential.

Slight Shoulder potential and External Potential

When working with the gapped graphene potential, the first step was to model the effects of the gapped graphene potential alone. As expected, based on visual analysis of the plotted potential, this did not result in stripe and bubble morphologies. It is worth noting here as well that this also presented one of the limitations of the set boundary conditions. The slope of this potential is closer to 1/r than the $1/r^3$ potential used in the toy model, which appears to have worked with the square boundaries to produce a square lattice. This suggests it may be worthwhile to attempt modelling this interaction in a parallelogram-shaped boundary to allow it to form the Wigner crystal structure again.

The observation of the relationship between shoulder and bubble size led to a formula establishing the boundary size to be roughly $n \cdot \alpha$. This was chosen through observation of the plots at different values of α. It can be seen that the left end of the shoulder remains relatively in the same position while the right end of the shoulder exists at approximately the value of α. Additionally, the left hand end of the shoulder is observed to be around 0.1, which is so small in comparison to the width of the shoulder that it could be estimated the shoulder width was approximately α arbitrary units of distance. This n parameter established how many maxima (and subsequently how many minima) appear accross the plane. This is important because it ensures that the minima will have roughly the same width as the shoulder, leading to bubble formation. If the size of the minima does not line up with the shoulder width, as demonstrated in Figure 8b, then the two forces will not work together to cause the formation of bubbles. These plots are directly comparable because while L doubles, the density of particles remains fixed, so that the only meaningful difference between these two plots is, in fact, that the external potential minima was adjusted to no longer match the physical spacing for which the particles were primed. None of the simulations run in this manner was able to produce stripes.



(a) V(r) plotted on a log-log scale with α=10.

(b) V(r) plotted on a log-log scale with α=100.

Figure 5: Interaction potentials for α = 10, 100. As α increases, the prominence and right curve of the shoulder increase as well.

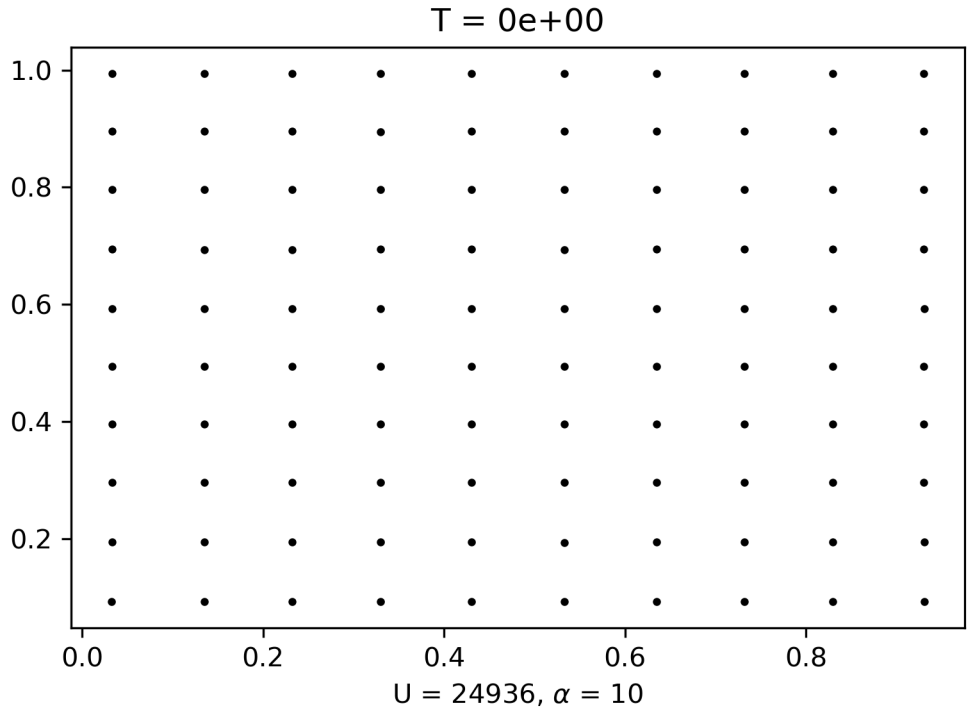


Figure 6: Simulation of electrons using gapped graphene potential. As seen here, stripes and bubbles did not form. Additionally, the electrons fell into a metastable square lattice formation, which is likely an artifact of the system boundaries.

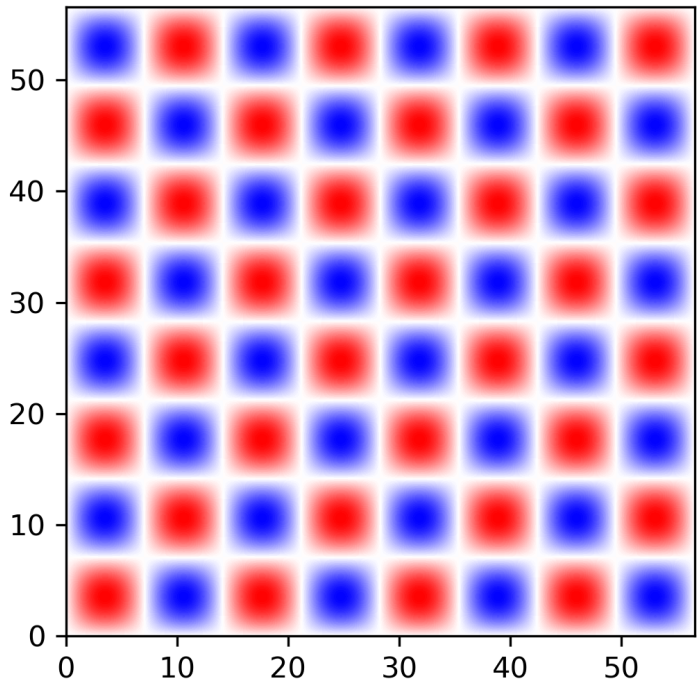
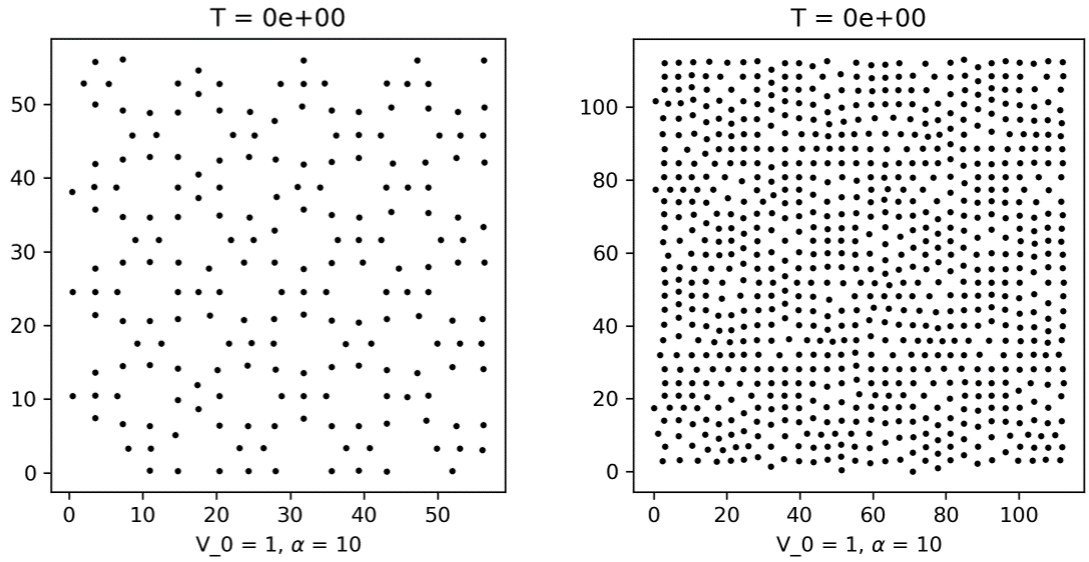


Figure 7: Representation of the external potential introduced by the substrate. Red squares represent energy maxima, while blue squares represent energy minima.



(a) Simulation for which n = 4 and L = n·α

(b) Simulation for which n = 4 and L = 2·n·α

Figure 8: Simulation of gapped graphene where a) the width of the external potential minima aligns with the shoulder width and b) the width of the external potential minima is roughly twice the shoulder width

IV Conclusions

To summarize, this exploration led to an understanding the physical space relationship with the Law of Interaction plot and stripe/bubble size. Additionally, the inter-electron interactions in gapped graphene, when compounded with an external interaction potential that aligned well with the shoulder in the former plot, was able to produce bubbles, but not stripes. Further work should be done to understand the system more fully, and more clearly establish a relationship between the two potentials at play, and the distribution of electrons in the system.

References

References

[1] Leon Balents et al. “Superconductivity and storng correlations in moiré flat bands”. In: *Nature Physics* (July 2020).

[2] M. M. Fogler, A.A. Koulakov, and B.I. Shklovskii. “The Ground State of a Two-Dimensional Electron Liquid in a Weak Magnetic Field”. In: *Condensed Matter Physics* (Jan. 23, 1996).

[3] Valeri N. Kotov, Vitor M. Pereira, and Bruno Uchoa. “Polarization charge distribution in gapped graphene: Perturbation theory and exact diagonalization analysis”. In: *Physical Review B* (Aug. 22, 2008).