

Computer Simulations of Dimer Models

Daniel Wray

University of Nottingham

January 16, 2019

Abstract

We study the statistics of dimer systems and problems with calculating expectation values. A Markov Chain Monte Carlo method that uses plaquette flipping to generate configurations is described. We observe and correct errors in our algorithm using a blocking technique, and numerical approximations of system properties are calculated with the results $\langle n_h \rangle = 0.497 \pm 0.005$ and $\text{var}(n_H) = 0.157 \pm 0.005$. Motivated by a need for efficiency, a more complex loop algorithm is developed. Using this algorithm equivalent simulations are executed with improved errors, yielding $\langle n_H \rangle = 0.5004 \pm 0.0007$ and $\text{var}(n_H) = 0.1592 \pm 0.0007$. The space of the system is discussed, and we find that boundaries of the system influence numerical properties calculated in a simulation. We analyse the two-monomer system and find they behave like charged particles due to the emergence of a coulombic potential in both two and three dimensions.

1 Introduction

The dimer model studies the behaviour of theoretical particles named dimers that bond to two neighbouring sites on a lattice. In this report lattices with square structure are considered; however, the theory can be generalised to lattices with an arbitrary number of neighbours at each site. Dimers are represented graphically as eccentric ellipses or rectangles that overlap sites on the lattice only once, for this reason the two-dimensional case is sometimes referred to as 'domino tiling' and an example configuration can be seen in figure (1). The dimer model has its roots in statistical physics, to illustrate the motivation for this project and why computer simulations are necessary it is useful to briefly investigate this physics. [1]

For any system we can define a partition function as

$$Z = \sum_C \exp\left(-\frac{E_C}{k_B T}\right), \quad (1)$$

where C is a configuration, E_C is the energy of a configuration and T is the temperature of the system. The physical significance of the partition function is that it represents the normalisation condition for the probability of finding the system in a specific configuration. If we then consider a quantity, Q , then the expectation of Q can be written

$$\langle Q \rangle = \frac{\sum_C Q_C \exp\left(-\frac{E_C}{k_B T}\right)}{\sum_C \exp\left(-\frac{E_C}{k_B T}\right)}, \quad (2)$$

where Q_C is the value of the quantity Q in configuration C . The systems this report considers have no preference over the configurations; however, one could imagine that configurations having different energies would make some more energetically favourable than others. This would require us to evaluate all the Boltzmann weights in our calculations. Since we do not wish to have this preference we set $E_C = 0$, or equivalently consider all our systems to be at infinite temperature. The consequence of this is

$$\langle Q \rangle = \frac{\sum_C Q_C}{\sum_C 1} = \frac{1}{\Omega} \sum_C Q_C, \quad (3)$$

here Ω is the total number of possible configurations. We can see that the expectation of the quantity Q re-

duces to the mean of the value over every configuration. Note that this sum takes place over Ω terms. For systems on a lattice of size 4×4 say, the total number of configurations is relatively small and could feasibly be calculated by brute force. Once lattice sizes exceed 4×4 , however, the total number of terms is huge and even a computer would struggle to calculate the total sum in a realistic amount of time.

The total number of configurations on an $n \times m$ lattice can be roughly approximated as [2]

$$\Omega \approx e^{0.3nm}. \quad (4)$$

The large number of configurations Ω sits at the core of the problem I will tackle in this report. That is, calculating the expectation value of a quantity without summing over all configurations.

The statistical method we will use to approximate the expectation values is the Monte Carlo method [3][4]. The central idea of the Monte Carlo method is to use randomness to sample the configuration space of a system and make an approximation based upon that sample. Provided the sample is taken from the set of every possible configuration and is completely random then the approximation will be accurate, moreover, as the sample size increases the approximation will become more accurate. Thus far we have only referred to an arbitrary quantity Q . When speaking generally this is acceptable; however, when simulations are to be executed, this must be done with reference to a real quantity. When running all simulations for this report the standard quantity measured was the number of dimers in the horizontal orientation, N_H . On a lattice of size $L = n = m$ (here after will be assumed) the analytical solution[1] in two-dimensions, given equation (3) is true, is

$$\langle n_H \rangle = \left\langle \frac{2N_H}{L^2} \right\rangle = \frac{1}{2}. \quad (5)$$

By picking random configurations of dimers on a lattice, calculating n_H in each configuration and then finding the mean, we can approximate the value of $\langle n_H \rangle$ and compare the results to equation (5).

Here, our focus switches from the statistics of approximating the expectation values, to effective ways of generating a random sample of dimer configurations from the large population.

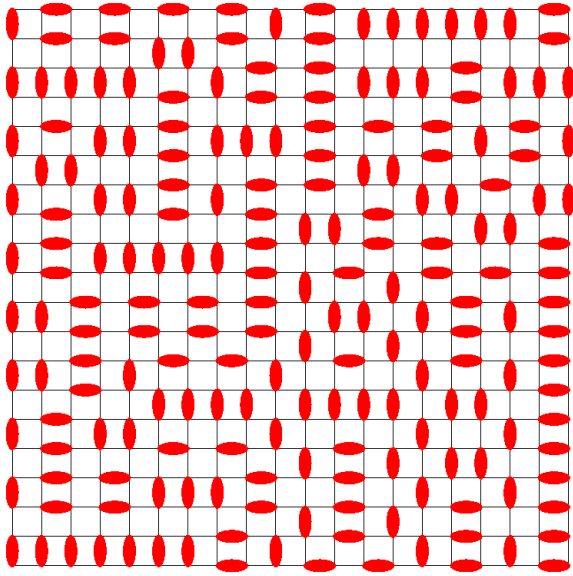


Figure 1: An example dimer configuration on a two-dimensional lattice with $L = 20$. This is also an example of a 'dimer covering' where no sites are left uncovered.

2 Plaquette Flip Algorithm

This algorithm is the simplest of the randomisation techniques and provides a starting point from which problems can be ironed out before the complexity of the algorithm is increased. The algorithm uses the idea of plaquettes, which are grid squares on the lattice, this means a lattice with L sites in each dimension has $L - 1$ plaquettes in each dimension. The algorithm then chooses a random plaquette on the lattice and attempts to flip the orientation of the dimers that sit on the four sites at the corners of the plaquette. This is not always possible; a successful flip can only occur when the four sites are occupied by two parallel dimers - either both horizontal or both vertical¹. This restriction is required to forbid the creation of uncovered sites, known as monomers[5]. A successful flip will alter the number of horizontal and vertical dimers by two and move the system from one configuration into another. The same algorithm can then be reapplied to the resulting configuration to

¹In three-dimensions the plaquettes can be flipped in the lateral direction. The number of directions the plaquettes can be flipped in is proportional to the number of dimensions the system is in.

move into yet another new configuration. This type of algorithm is named a Markov chain Monte Carlo method[3], named as such due to the chain like relationship between multiple configurations. While this algorithm allows us to traverse the space of possible configurations, it does so in such a way that the difference between two successive configurations is tiny. This causes problems when trying to find the expectation value of N_H . Conventional Monte Carlo methods require a purely random sample to represent the population; however, the plaquette flip algorithm returns highly correlated configurations from a very localised region in configuration space, the effect of which is a bias and a decrease in the reliability of the approximation[4].

A common solution is to section large numbers of configurations produced by the plaquette flip algorithm into 'blocks', then the expectation value can be approximated by taking the mean of the result from each block[6]. This solves the problem because, although the individual configurations between successive flips may be correlated, the large number of flips in each block mean that the blocks themselves are more representative of a random sample. The drawback being that the speed of the simulation is reduced, as far more flips must be taken to produce a single data-point.

Thus far the plaquette flip method takes a configuration and then returns a new configuration, but a choice must be made for the starting configuration that is initially given to the algorithm. We shall chose the initial configuration to have all the dimers in the horizontal orientation. With this choice the system will start at $n_H = 1$. Figure (2) shows the evolution of the system after 10 blocks of 500 plaquette flips are attempted. Notice that the choice of initial configuration causes a large deviation from the equilibrium position, until several plaquette flips are applied to randomise the dimers. A completely horizontal dimer configuration is highly ordered and does a poor job at representing the rest of the possible configurations[6]. These initial high order configurations are far from equilibrium and should therefore be removed, as they cause a significant error in the result. The blocking technique used to remove biases previously makes the removal of this second error trivial, as it is simply a case of throwing away the data from the first block. Provided there are enough pla-

quette flips per block, the system should be in equilibrium prior to the second block beginning. The removal of the first block insures all the data considered is taken from a system in equilibrium.

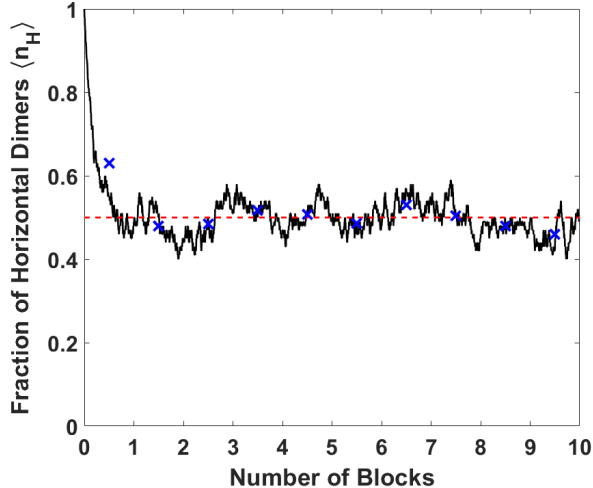


Figure 2: A system with $L = 20$ evolving using the plaquette flip algorithm. The initial configuration has all dimers in a horizontal orientation and each block contains 500 attempted flips. Blue crosses represent the mean of the corresponding block. Notice the deviation of the first cross due to the initial states not being in equilibrium.

With this full description of the plaquette flip algorithm, simulations can be executed to numerically determine the value of $\langle n_H \rangle$, testing its physical accuracy. Figure (3) shows the results of a simulation that used 11 blocks (the first of which was used to equilibrate the system and then discarded) with 1000 plaquette flips per block. Using the blocking technique and taking the mean of the values from each block yields a result of $\langle n_H \rangle = 0.497 \pm 0.005$. The simulation achieved this result in a time of 10s. Errors associated with the result can be reduced by either increasing the number of blocks or increasing the number of plaquette flips per block, the only cost being a larger computation time. The calculation of errors also happens to be benefited by using blocks, as the deviation of the blocks gives a much more realistic error than the deviation of all the configurations[3].

A second simulation using this algorithm was executed to calculate the value of $\text{var}(n_H)$, these results can be seen in Figure (3). The analytical solution[7]

of this is known in two-dimensions to be

$$\text{var}(n_H) = \langle n_H^2 \rangle - \langle n_H \rangle^2 = \frac{1}{2\pi} \approx 0.159. \quad (6)$$

After running this simulation the returned value was $\text{var}(n_H) = 0.157 \pm 0.005$, this again shows that the method was physically accurate. It is from this simulation we can see, although the plaquette flip algorithm is accurate, it is not efficient in the amount of computation required to converge on the result. This was one motivation for the development of alternative randomisation techniques.

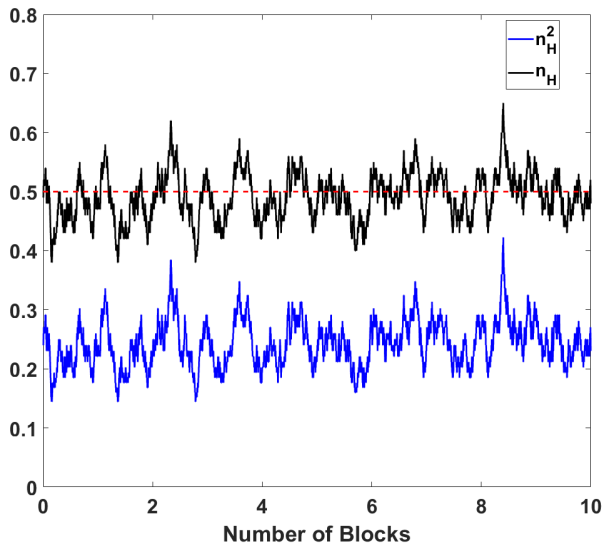


Figure 3: A final simulation using the plaquette flip algorithm with $L = 20$. The numerical results found were $\langle n_H \rangle = 0.497 \pm 0.005$ and $\text{var}(n_H) = 0.157 \pm 0.005$

3 Loop Algorithm

Flipping plaquettes is a relatively slow and simplistic method for generating random samples of dimer configurations, but this simplicity allowed more general problems to be isolated that would have arisen regardless of the randomisation technique. Now biases in the samples caused by initialisation and correlation have been removed, the complexity and efficiency of the algorithm can be increased.

The Loop method[5][8], sometimes called the worm method, uses a random path to shuffle dimers around the lattice. An example loop that has been

applied to a completely horizontal arrangement of dimers is depicted in figure (4). Loops created by this algorithm follow the alternating pattern of travelling in the direction of a dimer and then jumping to the tip of a neighbouring dimer. Open loops can be formed between any two uncovered sites (monomers); however, for systems with a complete dimer covering there are no monomers and only closed loops can be formed. Closed loops can be induced by rotating a dimer about one of the sites it occupies. Moving a dimer in this manner creates a monomer at one of the sites it used to occupy, and the new site is now covered by two dimers. This configuration is clearly forbidden as a site cannot be covered by two dimers at once. The fastest way to return to a valid configuration would be to rotate the dimer back into its original spot, but this would never generate a new configuration. Instead if the other dimer that occupies the same site is rotated about its other site into a random orientation, then the double overlap has been removed from its original site and recreated at another. If this is repeated, randomly rotating dimers to remove the double overlap, then eventually a dimer will rotate to fill the monomer that was created initially. When this occurs, a new valid configuration has been found.

By changing the orientation of a dimer to create a monomer, the loop algorithm behaves like a chain reaction of dimers all being forced to change orientation one after another until the system is in a valid configuration. A useful restriction made by the algorithm when changing the orientation of a dimer is to disallow the starting orientation to be randomly chosen. This prevents the loop for going back on itself and making no progress. It also prevents the aforementioned problem of instantly returning back to the same configuration, meaning the probability of the random sample containing two of the same configuration is very low.

An implementation of the loop algorithm is also Markov chain Monte Carlo method[8], and like the plaquette flip method cannot generate configurations, it can only take one configuration and return another. Due to this, the algorithm must again be initialised. This is done in the same way as the plaquette flip, placing all dimers in the horizontal orientation, and thus data should only be taken from the system once it is in equilibrium. Bringing the system into equilibrium happens after far fewer applications of the loop

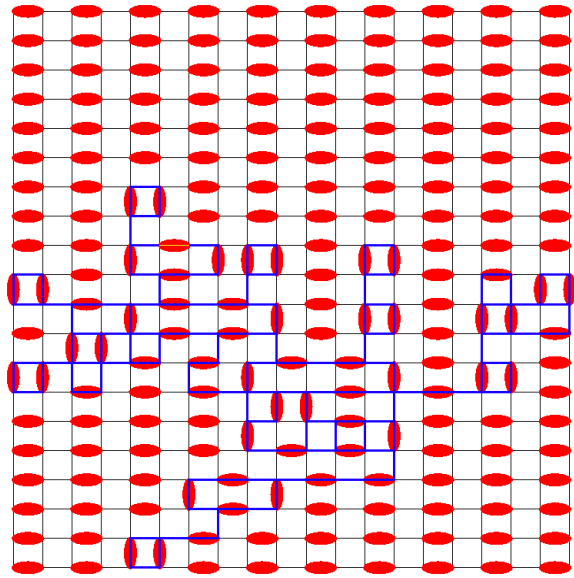


Figure 4: An example loop that has been applied to a horizontal dimer covering on a lattice of size $L = 20$.

algorithm compared to plaquette flips, as loops are not restricted to changing only two dimers per cycle. Figure (5) is a comparison of the performance of both algorithms. It is important to compare raw performance and not the number of flips against number loops because, although a loop may change the orientation of more dimers than a plaquette flip, loops are more computationally expensive.

Since large numbers of dimers can be altered by a single loop the correlation between successive configurations is small, causing the algorithm to converge onto a result significantly faster than the plaquette flip algorithm. Although the problems solved using the blocking technique still exist in the loop algorithm and the technique must still be used, the errors associated with them are smaller than those in the plaquette flip method[5]. To understand why it is important to still use the blocking technique, let us consider the probability of a loop of minimum length occurring.

A loop of minimum length would change the orientation of two dimers, making it equivalent to a single plaquette flip. When a random dimer has been chosen, on a lattice with square structure, there are 3 possible orientations that could randomly be selected. A plaquette flip changes the orientation of

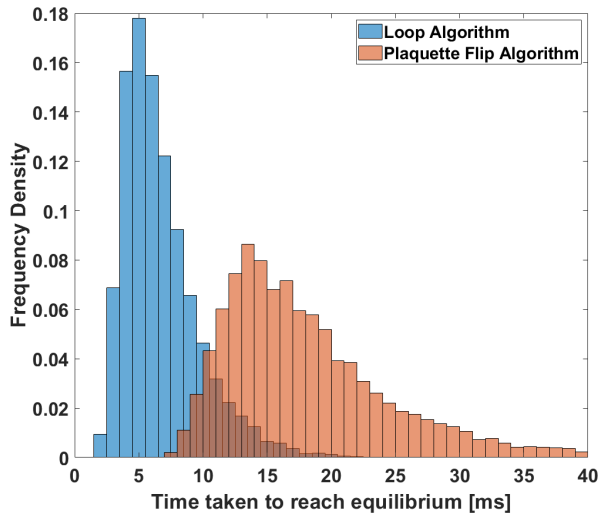


Figure 5: Starting from an initial horizontal dimer covering with $L = 20$ and applying each algorithm until the system is in equilibrium (defined as the point at which n_H first crosses 0.5). The performance of both algorithms can be compared by recording the time taken and plotting a distribution. 10^4 samples per algorithm were taken to create the distribution.

both dimers into the opposite of their current orientation, meaning that 2 of those 3 possible orientations can cause a loop of minimum length. Finally, the second dimer must also change its orientation, again having 3 possibilities. This time only one of those possibilities will create a loop equivalent to a plaquette flip, as to do so the dimer must rotate to fill the neighbouring monomer. The probability of these two events happening successively is $\frac{2}{9} \approx 0.22$. This crude calculation assumes all dimers to be in a horizontal configuration and is very idealised. For this reason we would expect the observed value to be significantly lower, but the loop of minimum loop length to still be the most frequent loop. Storing the loop length data of a simulation allows a probability density distribution to be approximated. Figure (6) confirms this calculation was an overestimate but that a minimum length loop is not only probable, it is the most probable length of loop. Meaning that, although the loop algorithm does a better job of selecting random configurations, a relatively large fraction of the time the loops are equivalent to plaquette flips; thus even with the loop algorithm it is still necessary to use the blocking technique.

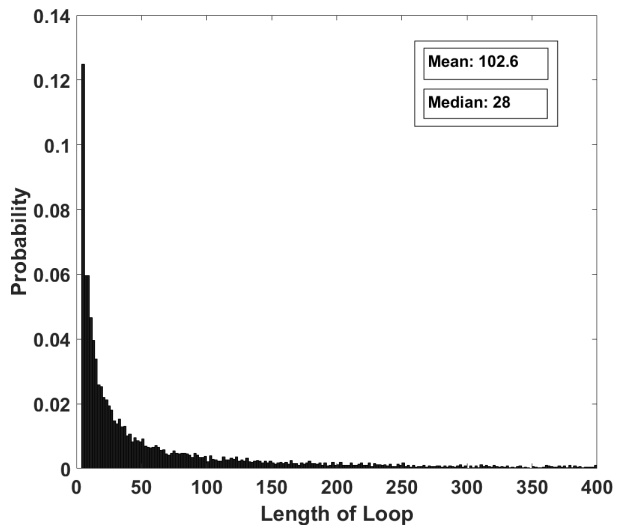


Figure 6: Using a sample of 10^4 loops drawn on a lattice with $L = 20$, a histogram approximation of a probability distribution can be formed.

Simulations equivalent to those carried out for the plaquette flip algorithm were also executed using the loop algorithm, to confirm the physical accuracy of the method. From our analysis of the algorithm, we would expect the loop method to converge onto a result much faster, consequently reducing the size of the errors. Figure (7) shows the result of these simulations. It is when comparing figure (7) to figure (3) that the differences between the algorithms can be seen clearly. The loop algorithms ability to change the orientation of large numbers of dimers is very apparent when making this comparison, due to the higher number of times the system crosses the equilibrium point. The results of the simulation were $\langle n_H \rangle = 0.5004 \pm 0.0007$ and $\text{var}(n_H) = 0.1592 \pm 0.0007$. The utility of an algorithm that changes the orientation of many dimers and produces results with small errors is the ability for the algorithm to be efficient on larger lattice sizes. As the lattice size increases the effect of a plaquette flip becomes more negligible and the accuracy drops. As seen in figure (6), in principle, a loop could be arbitrarily large and the algorithm does not suffer a decrease quite as dramatic.

A yet undiscussed detail of dimer systems are the boundary conditions. If a dimer is situated at the boundary of the lattice, the behaviour of the dimer is affected by the geometry of the space[1]. Thus far

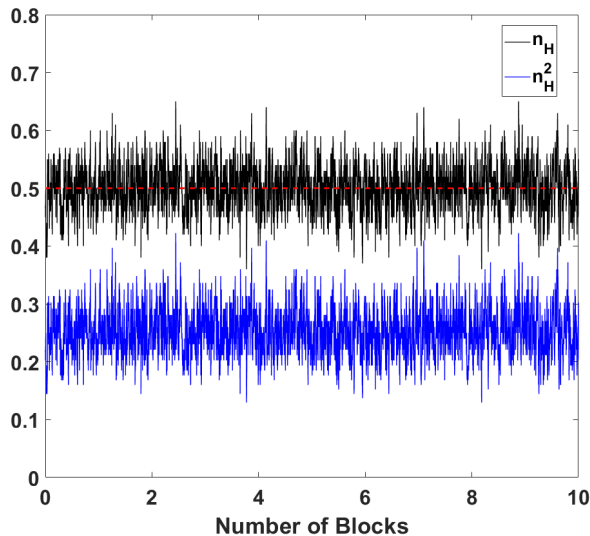


Figure 7: The results of a simulation using the loop algorithm on a lattice with closed boundaries and $L = 20$. The numerical results were $\langle n_H \rangle = 0.5004 \pm 0.0007$ and $\text{var}(n_H) = 0.1592 \pm 0.0007$. Notice how the variation between successive configurations is much higher when compared to the plaquette flip algorithm seen in figure (3).

we have only considered systems with closed boundaries, these have the effect of a hard border that the dimers cannot cross. When the orientation of the dimers are randomly changed, the possible choices are reduced by this boarder. For dimers in the corner of the lattice this is amplified and there are situations in which there is only one possible orientation that could be chosen. Because the boundaries influence the behaviour of the dimers, as expected, the results of certain numerical properties are also influenced. Toroidal geometry causes the boundaries to be periodic. When a dimer is at an edge of the lattice it is no longer influenced by the existence of an edge, as a hard border no longer exists. The borders are effectively sewn together and free movement for dimers to span from one side to the other is permitted. Implementing this into the loop algorithm is very simple and is only a case of changing the definition of a lattice site neighbours[5]. Periodic boundaries were never implemented into the plaquette flip algorithm in this report, mainly because the loop algorithm is more efficient, but also because the ability to cross the edge creates a flaw in the method. If we con-

sider a configuration that contains no parallel plaquette pairs, this is completely inaccessible to the plaquette flip algorithm. As the algorithm can no longer access the entire search space, it will produce subtly biased search results.

Figure (8) depicts many simulations executed with varying lattice size and boundary conditions. These simulations produce an interesting graph with some unexpected behaviour. First let us analyse the graph for results we expected. Periodic boundaries have slightly different numerical results when compared to closed boundaries, we expect this as the boundaries influence the dimers. We also see the closed boundaries results are in alignment with equation (6) and the variance is approximated rather well. We did not expect; however, the curvature in the graph for small L as equation (6) predicts a constant. This could be attributed to the fluctuations in the number of horizontal dimers being highly volatile. For example, changing the orientation of two dimers on a lattice of size $L = 100$ would alter the value of n_H by 2×10^{-4} , whereas on a lattice with $L = 4$ this would change the value by 0.125, a much larger amount. This difference could cause the observed inaccuracy at very small L .

4 Effective Interactions

With an efficient, physically accurate algorithm of randomising the dimers and techniques to approximate numerical properties of the system, we can now use this to simulate some more complex physics. Here we will investigate the effective interaction of two monomers on an otherwise dimer covered lattice. To do this we must reconsider the partition function from equation (1). A partition function takes a sum over every possible configuration. Previously we had been considering only the configurations with no monomers present. Now, we only care about the system with two dimers at positions r_0 and r_1 and must sum up all possible configurations. Initially counter-intuitive, removing a dimer (adding two monomers) means the total number of configurations increases. This is because, not only do we have to consider the configurations of the dimers, but also then the configurations of the dimers for every possible position of the monomers. We can write the partition function

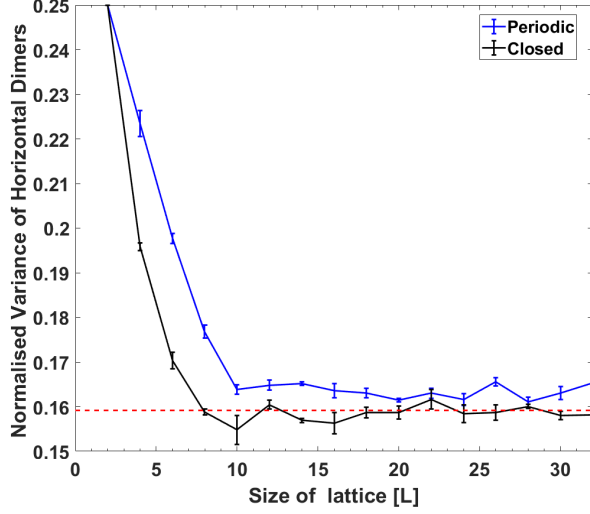


Figure 8: The variance of horizontal dimers $\text{var}(n_H)$ as a function of the lattice size. This was repeated for both periodic and closed boundaries. Notice that for periodic boundaries the variance converges onto a slightly larger value. There is also a region for small L where the variance does not converge onto the analytical value.

for the two-monomer system as

$$Z_m = \sum_{r_0, r_1} \sum_{C(r_0, r_1)} \exp\left(-\frac{E_{C(r_0, r_1)}}{k_B T}\right), \quad (7)$$

where we have used the previous logic to split the sum over all configurations into the sum of monomer positions and dimer orientations[9]. In the introduction we also set E_C to zero to avoid the preferential orientating of dimers. Here, since we are trying to find the interaction of monomers, the possible configurations C in this system are a function of the position of the monomers, and this restricts us from setting E_C to zero. To remove the dependence E_C has on the configurations of dimers with fixed monomer positions, we can define a new variable V_m as the effective potential of the two monomers, which satisfies

$$\exp\left(-\frac{V_m(R)}{T}\right) = \sum_{C(r_0, r_1)} \exp\left(-\frac{E_{C(r_0, r_1)}}{k_B T}\right), \quad (8)$$

where R is the separation of the monomers $r_1 - r_0$. Giving us a partition function for this two-monomer

system in the form

$$Z_m = \sum_{r_0, r_1} \exp\left(-\frac{V_m(R)}{T}\right). \quad (9)$$

With the partition function well defined, our interest can be turned to the effective potential between the monomers. Solving equation (8) for the effective potential is relatively simple and gives us

$$V_m(R) = -T \ln \left[\sum_{C(r_0, r_1)} \exp\left(-\frac{E_{C(r_0, r_1)}}{k_B T}\right) \right]. \quad (10)$$

The form of this potential does not sum over all possible values of r_0, r_1 and we can set E_C to zero (or equivalently $T \rightarrow \infty$) as in the introduction. Doing this will cause the exponential term to be unity for all terms in the sum. The sum can now be evaluated as

$$V_m(R) = -T \ln G(R) \quad (11)$$

Where we have used $G(R)$ to represent the value of the sum given the separation of the monomers. This variable G is analogous to the total number of dimers Ω [7]. The difference is that G is not a constant property of the system. A final step must be taken to make the potential solvable. Notice that equation (11) has a dependence on T . We have effectively set this value to infinity and therefore the equation should blow up. To get around this we define a new variable

$$U_m(R) = \frac{V_m(R)}{T} = -\ln G(R), \quad (12)$$

which encodes all the valuable information about the effective potential without blowing up to infinity.

With the mathematics understood we can proceed to use our simulation to numerically approximate the quantity G and therefore the potential between the two monomers. Due to the large number of possible positions of the monomers, a large quantity of data will be required. One would assume that to run this simulation, monomers will be placed onto the lattice by removing a dimer. Then the loop algorithm would be used to permute through the possible configurations and approximate G . However, there is a much more efficient method that exploits the loop algorithm to massively increase the data produced by a single application of the algorithm. By using a lattice

with a complete dimer covering and then measuring the distance at each stage in the loop algorithm from the start position, we can reproduce the behaviour of two monomers[10]. The key here is that instead of producing one data point per loop, depending on the size of a loop, the algorithm produces several data-points. A small caveat when using this method is to remember that the monomers can never occupy the same lattice site and so since the grid method will produce closed loops the data-points at $R = 0$ will all be omitted from the statistics. To view the shape and structure of the potential a two-dimensional distribution of G can be drawn. Figure (9) shows the resulting distribution of this simulation.

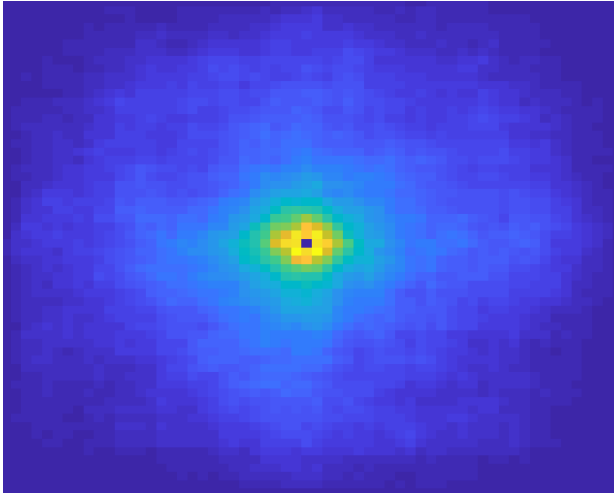


Figure 9: A heat-map representation of $G(R)$ in two-dimensions. The origin of the black dot at the center is due to the monomers being unable to overlap and therefore R is never zero.

While Figure (9) gives a visual understanding of the potential, it is not the most useful representation of the data. Using the same data to plot the graph of G against R allows the curve to be approximated using an analytical expression, which is far more useful. Figure (10) shows this graph and an approximate curve with the expression

$$G(R) = R^{(-k)}, \quad (13)$$

where k is a constant. This expression is useful because we can sub into equation (12) and see that the effective potential between two monomers in two dimensions is

$$U_m(R) = k \ln R. \quad (14)$$

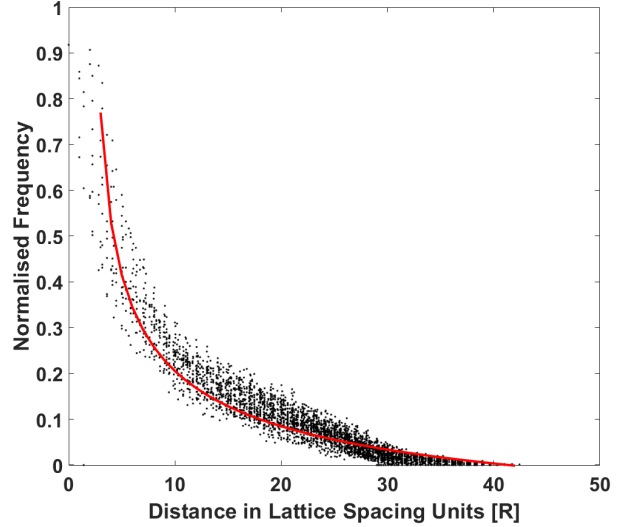


Figure 10: Using all the data gathered in Figure (9) a frequency against distance plot can be drawn. The distribution is thick due to there being more than one lattice site with the same distance, and they do not necessarily have the same frequency. The red line is a best fit given the expected form.

Perhaps the most remarkable feature to be noticed about this result is that the potential shares the exact same form as the potential for charged particles in two dimensions. To further investigate the link between the behaviour of monomers and charged particles, all elements in the simulation thus far were extended into three-dimensions. This required the loop algorithm to move in the lateral direction. The physical accuracy of this modification was again checked by counting the number of horizontal dimers and observing it to, in the three dimensional case, converge at $\langle n_H \rangle = \frac{1}{3}[1]$. Figure (11) shows a distribution similar to figure (9), only in this case the distribution of G is three-dimensional and the figure shows a slice through the xy plane. Again, we can fit a curve to the data. This is shown in figure (12). This time in three dimensions the resulting expression is

$$G(R) = \exp \frac{k}{R}, \quad (15)$$

with an effective potential of

$$U_m(R) = -\frac{k}{R}. \quad (16)$$

Even in the three-dimensional case, the effective interaction between monomers takes on the same form

as that between charged particles, and a coulombs law-like potential emerges[11].

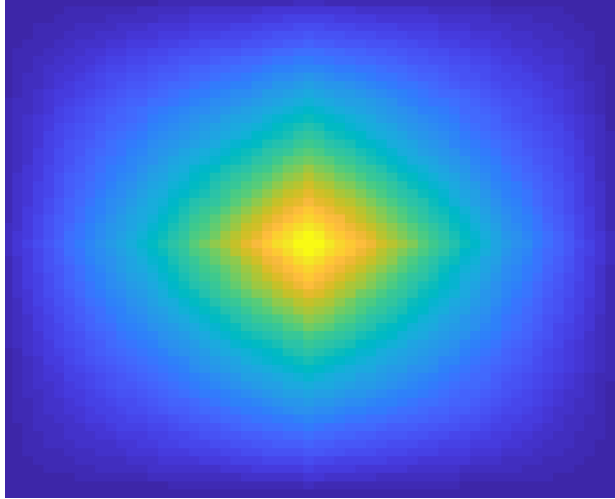


Figure 11: A heat-map representation of $G(R)$ in a three-dimensional system. To generate this image a two dimensional plane that cuts through the origin was taken. The square shape of the distribution for small R is a product of the square lattice structure.

5 Summary

In this report we have seen how the use of a Monte Carlo method and statistical analysis of dimer systems can form algorithms capable of extracting numerical properties of the system with physical accuracy. The plaquette flip algorithm was used to allow general problems to be solved initially. Then once they were solved the complexity of the algorithm was increased and the loop algorithm was used. Both these algorithms, although using loops was more efficient and accurate, produced values of n_h and $\text{var}(n_h)$ that agreed with analytical results. We also analysed the boundary conditions and type of space in which the simulation took place. This led us to find the value of $\text{var}(n_h)$ with periodic boundaries is slightly higher than that in the closed system. Using the implementation of the algorithms developed we were able to simulate a two-monomer system and calculate effective potentials between them. We then saw the emergence of an expression that was of the same form as the potential produced by coulombs law. Extending the system and algorithms to work

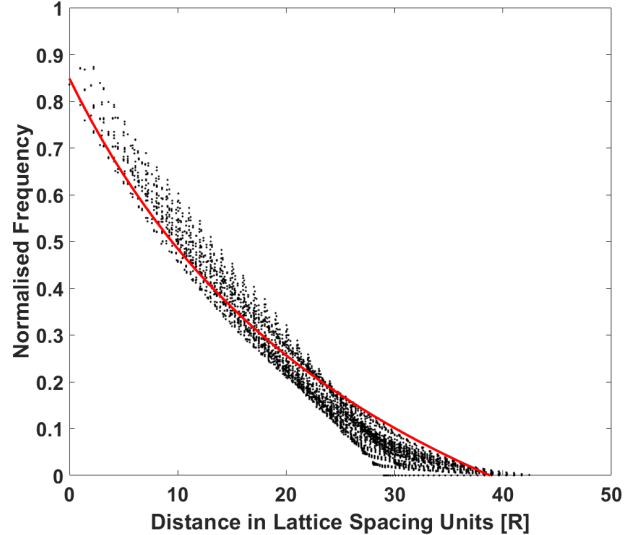


Figure 12: The distribution of distances collected from the three-dimensional array used to generate figure (11). Using the expected form a best fit curve was drawn, seen in red. The best fit curve does not seem as well fitted as in the two dimensional case. This could be due to the larger number of dimers in the system and an increased number of possible configurations.

in three-dimensions also confirmed this emergence.

A application of the simulations I would have liked to investigate, if I had more time, would be adding a random scattering of monomers to the two-monomer system and observing how this would affect the behaviour. Another development I would have liked to make, would be to investigate the addition of Boltzmann weights and for configurations of dimers to be energetically favourable.

Overall the simulations and results were a success. If I had the opportunity to repeat the project, I would have changed very little on the programming side - other than a few possible optimisations. I would have also spent less time on the plaquette flip algorithm; however, I understand its importance.

References

- [1] R. Kenyon. Exact methods in low-dimensional statistical physics and quantum computing. *Oxford Univ. Press* **89**, 341-361 (2008).

- [2] M. E. Fisher. Statistical mechanics of dimers on a plane lattice. *Phys. Rev.* **124**, 1664 (1961).
- [3] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika* **57**, 97-109 (1970).
- [4] Nicholas Metropolis. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* **21**, 6 (1953).
- [5] Stefan Wessel. Monte carlo simulations of quantum spin models. *Institute for Theoretical Solid State Physics (2011)*.
- [6] Charles Therrien and Murali Tummala. Probability and random processes for electrical and computer engineers. *CRC Press (2011)*.
- [7] P. W. Kasteleyn. Dimer statistics and phase transitions. *Journal of Mathematical Physics* **4**, 287 (1963).
- [8] Anders W. Sandvik and R. Moessner. Correlations and confinement in nonplanar two-dimensional dimer models. *Phys. Rev. B* **73**, 144504 (2006).
- [9] Christopher L. Henley. The coulomb phase in frustrated systems. *Annual Review of Condensed Matter Physics* **1**, 179-210 (2010).
- [10] Gregoire Misguich, Vincent Pasquier, and Fabien Alet. Correlations and order parameter at a coulomb-crystal phase transition in a three-dimensional dimer model. *Phys. Rev. B* **78**, 100402 (2008).
- [11] David A. Huse, Werner Krauth, R. Moessner, and S. L. Sondhi. Coulomb and liquid dimer models in three dimensions. *Phys. Rev. Lett.* **91**, 167004 (2003).