

## Scientific computing in Statistical Mechanics: time decay of orientation order in 2D hard disk system

A.G. França<sup>1</sup>, R.S. Grisotto<sup>2</sup>, B.M. Rocha<sup>3</sup>, F.A.A.M.N. Soares<sup>3</sup>, H.A. Fernandes<sup>1</sup>, P.F. Gomes<sup>1</sup>

<sup>1</sup>Instituto de Ciência Exatas, Universidade Federal de Goiás  
Jataí, GO – Brasil

<sup>2</sup>Instituto de Computação – Universidade Estadual de Campinas  
Campinas, SP – Brasil

<sup>3</sup>Instituto de Informática – Universidade Federal de Goiás  
Goiânia, GO – Brasil

andreygfranca@gmail.com, paulofisicajatai@gmail.com

---

**Abstract.** *In this work we successfully applied diverse computational techniques to calculate important quantities in one problem in Statistical Mechanics: the hard disk system in two dimensions. We calculated the global and local orientation order and the time decay constant of the global orientation correlation function. We also computed the Voronoi construction to visualize the spatial distribution of the local orientation order.*

**Resumo.** *Neste trabalho apresentamos diversas técnicas computacionais no cálculo de grandezas de interesse em um sistema de Mecânica Estatística: discos rígidos em duas dimensões. Calculamos a ordem de orientação local e global e a constante de decaimento da função de correlação da orientação global. Fizemos também a diagramação de Voronoi para visualização da distribuição espacial da orientação local.*

### 1 Introduction

Solidification and vaporization of water are two common examples of phase transitions in physics. The temperature and pressure defines which phase the water is. Another parameter called order parameter is used to verify which phase the system is. In the case of the water it is the molecule organization: in the solid phase the molecules form an ordered hexagonal state, in the liquid state the molecules are close to each other but with no organization and in the gas state there is no correlation between the molecules. One interesting system to study phase transitions is the hard disks in two dimensions [Isobe and Krauth (2015)], composed of  $N$  disks of radius  $R$  confined in a 2D planar rectangular box of sides  $L_x$  and  $L_y$  (see Fig. 1(a)). Each disk  $k$  has its center at position  $(x_k, y_k)$ . The superposition between two disks is forbidden so the minimal distance between them is  $2R$ . One parameter is the local orientation of each disk  $k$  defined as [Deutschlander et al.

(2014); Engel et al. (2013)]:

$$\psi_k(x_k, y_k) = \frac{1}{N_k} \sum_{l=1}^6 e^{6i\theta_{kl}}, \quad (1)$$

where  $i = \sqrt{-1}$  and the sum is taken over the six nearest-neighbors. The angle  $\theta_{kl}$  is between a line passing through the center of the two discs and the  $x$  axis (see Fig. 1(b)). The  $\psi_k$  is defined so that  $\psi_k = 0$  in the liquid phase and  $\psi_k = 1$  in the solid one. The spatial distribution of the parameter through the volume is also important. The order parameter can have different ranges defined by its dependency on the distance  $r$  [Deutschlander et al. (2014)]: exponential decay  $e^{-r/\xi}$  means short range, algebraic one  $r^{-\xi}$  means quasi-long range and a constant variation means a long range. So the liquid state in the 2D hard disk system has short range and the solid state has long range for  $\psi_k$ . However, a third phase was theoretical and experimentally observed: the so called hexatic phase [Strandburg (1988)] (see Fig. 2(a)) and it has quasi-long range orientational order [Bernard and Krauth (2011)]. The hexatic phase is an exclusive feature of 2D systems [Isobe and Krauth (2015)].

### 1.1 Global orientation order

The global orientation is the absolute value of the average of the local one:

$$\Psi = \left| \frac{1}{N} \sum_{k=1}^N \psi_k \right|. \quad (2)$$

The phase of the system depends on the density  $\eta = (N\pi R^2)/(L_x L_y)$  (see Fig. 2(a)). The solid phase has an hexagonal symmetry (see Fig. 4(a)) and the liquid phase has random positions for the disks (see Fig. 4(b)). We study the time behavior of the orientation order correlation function:

$$C(\delta) = \sum_{t=0}^Q \Phi(t)\Phi(t + \delta), \quad (3)$$

which should be normalized so  $C(0) = 1$ . The normalized orientation order is  $\Phi(t) = \Psi(t) - \Psi_\infty$  where  $\Psi_\infty = \lim_{t \rightarrow \infty} \Psi(t)$ <sup>1</sup>. The correlation function has an exponential decay [Krauth and Wilson (2009)] of the form  $C(\delta) \approx e^{-\delta/\tau}$  where the decay time  $\tau$  is a function of the number of disks,  $\tau = \tau(N)$ . This function depends of the phase:

$$\tau(N) = \begin{cases} N \log N, & \text{liquid phase} \\ cN^\alpha, & \text{hexatic phase} \\ N, & \text{solid phase} \end{cases} \quad (4)$$

In this work we study the time decay  $\tau$  of the global orientation in the liquid-hexatic transition as function of  $N$  in the 2D hard disk system. We use different computation technique to evaluate the time evolution and the orientation order. The goal is to

<sup>1</sup>In practical terms,  $\Psi_\infty = \Psi(t = Q)$  where  $Q$  is the number of evaluated Monte Carlo iterations.

verify the exponential decay of the orientation order correlation function and then obtain the behavior of the time constant as function of the number of disks  $N$ . To our knowledge, and complete study of the  $\tau$  as function of  $N$  in the hexatic-liquid coexistence phase ( $0.70 < \eta < 0.716$ ) has not been done.

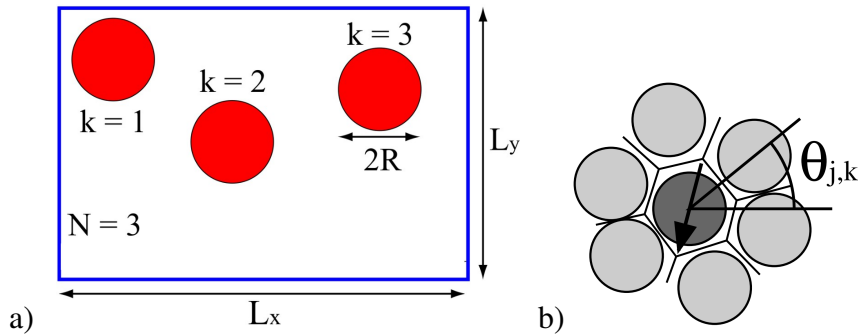


Figure 1: (a) Illustration of the 2D hard disk system. (b) Illustration of the angle  $\theta_{j,k}$  between two discs. This Fig. was taken from Ref. [Bernard (2011)].

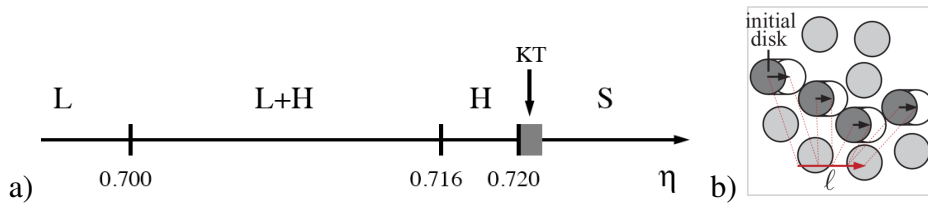


Figure 2: (a) Identification of the phase as function of the density  $\eta = (N\pi R^2)/(L_x L_y)$ . For  $\eta < 0.700$  the system is liquid (L), for  $0.700 < \eta < 0.716$  it is in the coexistence between the liquid and hexatic (L-H) phases and for  $0.716 < \eta \lesssim 0.720$  the phase is hexatic (H). For  $\eta \gtrsim 0.72$  the phase is solid (S). (b) Illustration of the movement in the Event-chain algorithm. Collective move with the total displacement  $\ell$ . The Figures were taken from Ref. [Bernard (2011)].

## 1.2 Simulation Method

We used two methods for the time evolution of the system. The first one is the Event Chain Monte Carlo method [Michel et al. (2014)] where many disks move in sequence in each iteration. Each disk moves until it reaches another disk (when their distance =  $2R$ ). Then the first disk stops and the second one starts to move until it reaches the third disk. The process repeats until the sum of the distances of all disks are equal to an input value  $\ell$  [Krauth and Wilson (2009)]. The method is efficient because many disks move in one iteration. The other method is the parallel Markov Chain Monte Carlo one, calculated with the Hard Particle Monte Carlo module (HPMC) [Anderson et al. (2016)] of HOOMD-BLUE package<sup>2</sup> [Anderson et al. (2008); Glaser et al. (2015)]. This implementation has been used in many type of problems including disks [Anderson et al. (2017)].

<sup>2</sup>[glotzerlab.engin.umich.edu/hoomd-blue](http://glotzerlab.engin.umich.edu/hoomd-blue).

To calculate  $\psi_k$  we need the distances between the disk  $k$  and all other disks. Using brute force one requires  $N - 1$  calculations. Then, to calculate the average  $\Psi$ , another  $N$  calculations will be done with total of  $(N - 1)N \simeq N^2$  calculations when  $N$  is large. The distances are also required in the time evolution of the system. We implemented this algorithm in C language. However, for large  $N$  we used the method List of Cells [Frenkel and Smit (2002)] which is faster. In this one a grid of  $N$  cells is created so there is one disk per cell. When we need to get the neighbor of a disk, we just look into the neighbor cells. The scaling of this method is  $N$ . We implemented this list of cells method in C++.

## 2 Results

Our C++ implementation is illustrated in the Algorithm 1. The distance calculations are performed in the functions `newL` and `abs_Psi`, where the `listcell` structure is used to identify the neighbor disk. The matrix `LxLy` keeps the positions  $x(t), y(t)$  of all disks in the iteration  $t^3$ . In the Fig. 3(a) there is a simple example of the list with 9 cells. To get the neighbor of a disk, just check the neighbor cells (each cell has pointer pointing to them and a linked list to keep the id of the disks). The Fig. 3(b) shows a comparison of execution time between brute force and list of cells methods. For small  $N$ , brute force is better because there is a cost to create the cell grid. However, for  $N = 128^2$  the list of cells is already one order of magnitude better, and this improvement increases with  $N$ .

---

### Algorithm 1 Time evolution with Event Chain Monte Carlo

---

```

1: procedure MCCE
2:   Input:  $N, \ell, \eta, S, Q, L_x$ 
3:   listcell  $\leftarrow$  CREATE_LIST(N)
4:   for  $k \in S$  do
5:     LxLy  $\leftarrow$  INITIAL(N,  $\eta$ ,  $L_x$ )
6:     ADD_DISKS(N, listcell)
7:     seed  $\leftarrow$   $k$ 
8:     for  $t \in Q$  do
9:       LxLy  $\leftarrow$  NEWL(LxLy, N,  $\eta$ ,  $\ell$ , seed, listcell)
10:      Psi(t)  $\leftarrow$  ABS_PSI( $\eta$ , LxLy, listcell)
11:   file  $\leftarrow$  time_Psi.csv
12:   write(t, Psi, file)

```

---

In total, the code has six input parameters: (1) number of disks  $N$ , (2) the density  $\eta$ , (3) the number of iterations  $Q$ , (4) the side  $L_x$  of the box, (5) the number  $S$  of samples in order to take the average (and decrease the error) and (6) the total distance  $\ell$ . We used  $S = 5$ ,  $L_x = 100$  and  $\ell = \sqrt{N}$  and  $Q = 30\,000$  in all simulations. All the graphics presented here were done in Python (Matplotlib package). The way to achieve the liquid state configuration shown in Fig. 4(b) was to evaluate the system with  $Q = 200$  iterations and  $\eta = 0.5$  using our C++ implementation and an initial hexagonal configuration (Fig. 4(a)).

---

<sup>3</sup>Although the concepts of the method and the definition of  $\Psi$  are not too complicated, the code itself presents many numerical challenges [Bernard (2011)].

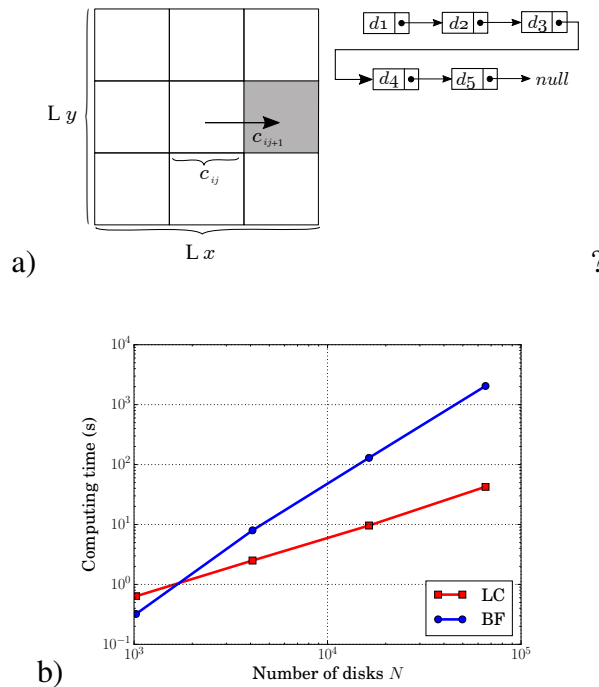


Figure 3: (a) System with 9 cells, each one has pointer pointing to the neighbors and a linked list. (b) Comparison between the two methods to calculate the distances between the disks: brute force in blue and list of cells in red.

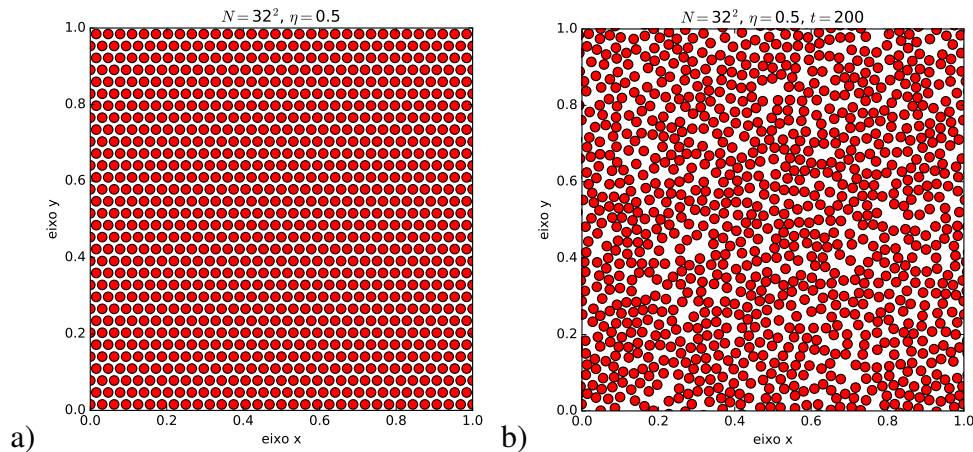


Figure 4: Simulation with  $N = 32^2$  and  $\eta = 0.5$  using our C++ implementation. (a) Initial configuration with triangular symmetry. (b) Configuration with  $t = 200$ .

The correlation functions  $C(\delta)$  were calculated for  $N = 64^2, 128^2, 256^2$  and  $512^2$ . Due to time limitations we were not able to perform simulations with a larger number of  $S$ . The results shown in the Fig. 5(a) were noisy. To find the decay constant it is needed more samples. Another result is shown in the Fig. 5(b), which is a kind of graphic used to visually identify the phases [Bernard and Krauth (2011); Engel et al. (2013)]. First a Voronoi construction is created where each cell contains only one disk. Then, the local orientation order  $\psi_k$  is calculated for each disk and a color coding is created. The disk in each cell are removed and each cell are painted with the color of its disk orientation

order [Bernard (2011)]. If  $\psi_k$  is aligned with  $\Psi$  we have  $\cos \theta_k = 1$  and if they are perpendicular  $\cos \theta_k = 0$ , using the associated vectors with  $\psi_k$  and  $\Psi$ . Note that in Fig. 5(b) there are  $N = 128^2$  disks. The number of sides of each cell is the number of its neighbors cells. As the solid phase of the hard disk system has hexagonal symmetry, the average number of neighbors cell is 6, which explains why most of the cells are hexagons. As the density is  $\eta = 0.7$ , the configuration of Fig. 5(b) is liquid.

The functions of HOOMD-BLUE package are organized in modules, which should be installed as regular Python packages (within Conda environment). The results for  $C(\delta)$  are shown in the Fig. 5(c) for  $\eta = 0.705$  and  $N = 64^2, 128^2, 256^2$  and  $512^2$ . To calculate the decay time  $\tau$  we adjust an exponential function  $f(\delta) = A_0 + A_1 e^{-\delta/\tau}$  to the data. The blue circles in Fig. 5(d) are the obtained values of  $\tau$ . The second step is to determine  $\alpha$  from Eq. 4. So we take the natural logarithm  $\ln \tau = B + \alpha \ln N$  (with  $B = \ln c$ ) and make a linear fit for  $\ln \tau$  vs.  $\ln N$ . The slope will be the constant  $\alpha$ . Indeed, they follow a linear behavior as the red solid line indicates in Fig. 5(d)<sup>4</sup>.

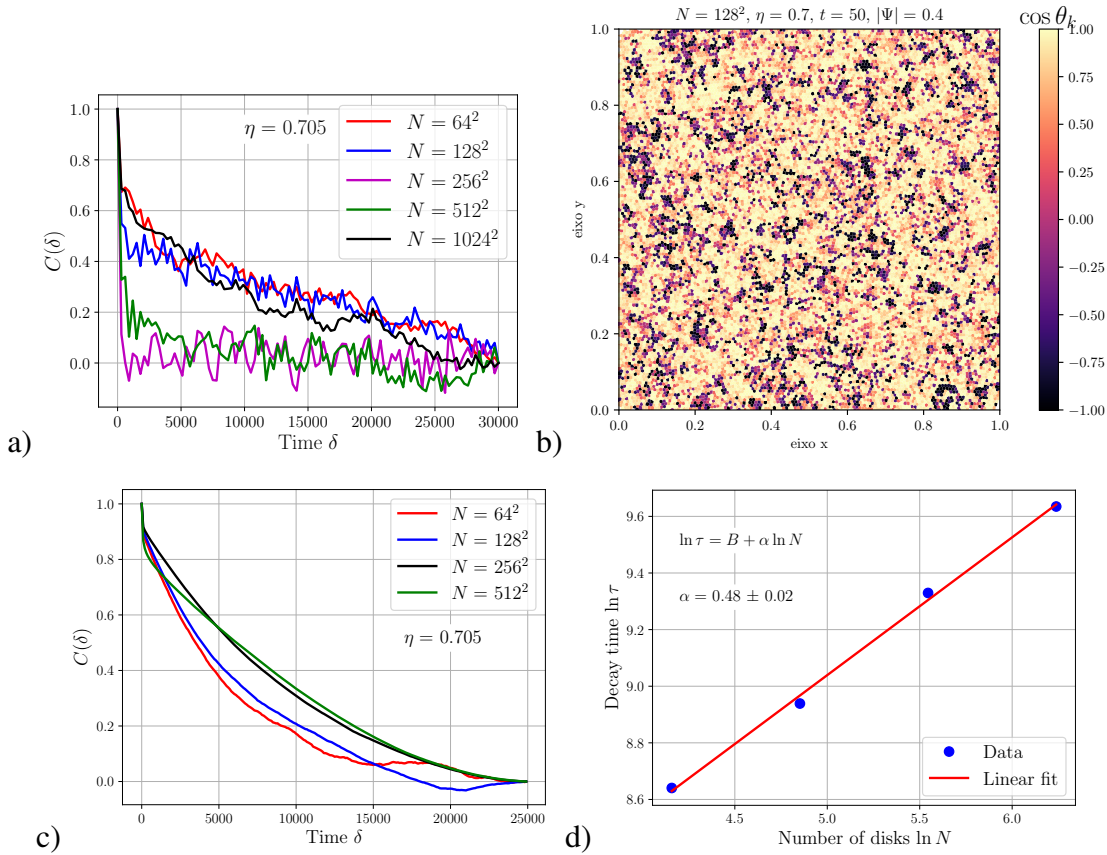


Figure 5: Time behavior of correlation function  $C(\delta)$  (Eq. 3). (a) Using our C++ implementation. (b) Voronoi construction where each cell contains only one disk. (c) Using HOOMD. (d) Linear fit of the decay  $\tau$  vs. the number of disks  $N$  using the results from HOOMD case.

<sup>4</sup>The code that generated the results presented here are available at [github: github.com/AndreyGFranca/mcec](https://github.com/AndreyGFranca/mcec).



### 3 Conclusions

We compute the correlation function of the orientation order in the 2D hard disk system and verified its exponential behavior. We then obtain the expected behavior of the decay constant  $\tau$  as function of the number of disks  $N$  for  $\eta = 0.705$ . We successfully applied various computation techniques to this Statistical mechanical problem, showing the great benefit which one can have from the interplay between Statistical Mechanics and computational modelling.

### 4 Acknowledgements

This work was funded in part by CNPq (A. G. França and R.S. Grisotto). We thank W. Krauth for the suggestions.

### 5 References

#### Referências

- J. A. Anderson, C. D. Lorenz, and A. Travesset. *J. Comput. Phys.*, 227:5342, 2008.
- J. A. Anderson, M. E. Irrgang, and S. C. Glotzer. *Comput. Phys. Commun.*, 204:21–30, 2016. doi: <http://dx.doi.org/10.1016/j.cpc.2016.02.024>.
- J. A. Anderson, M. E. Irrgang, and S. C. Glotzer. *PHYSICAL REVIEW X*, 7:021001, 2017. doi: <https://doi.org/10.1103/PhysRevX.7.021001>.
- E. P. Bernard and W. Krauth. *Physical Review Letters*, 107:155704, 2011. doi: <http://dx.doi.org/10.1103/PhysRevLett.107.155704>.
- Etienne Bernard. PhD thesis, Université Pierre et Marie Curie - Paris VI, The address of the publisher, 2011. <https://tel.archives-ouvertes.fr/tel-00637330/en>.
- S. Deutschlander, A. M. Puertas, G. Maret, and P. Keim. *Physical Review Letters*, 113:127801, 2014. doi: <http://dx.doi.org/10.1103/PhysRevLett.113.127801>.
- M. Engel, J. A. Anderson, S. C. Glotzer, M. Isobe, E. P. Bernard, and W. Krauth. *Physical Review E*, 87:042134, 2013. doi: <http://dx.doi.org/10.1103/PhysRevE.87.042134>.
- D. Frenkel and B. Smit. *Understanding Molecular Simulation*. Academic Press, 2002.
- J. Glaser, T. Dac Nguyen, J. A. Anderson, P. Lui, F. Spiga, J. A. Millan, D. C. Morse, , and S. C. Glotzer. *Comput. Phys. Commun.*, 192:97, 2015. doi: <http://dx.doi.org/10.1016/j.cpc.2016.02.024>.
- M. Isobe and W. Krauth. *The Journal of Chemical Physics*, 143:084509, 2015. doi: <http://dx.doi.org/10.1063/1.4929529>.
- E. P. Bernard W. Krauth and D. B. Wilson. *Physical Review Letters*, 80:056704, 2009. doi: <http://dx.doi.org/10.1103/PhysRevE.80.056704>.
- M. Michel, S. C. Kapfer, and W. Krauth. *THE JOURNAL OF CHEMICAL PHYSICS*, 140:054116, 2014. doi: <http://dx.doi.org/10.1063/1.4863991>.
- K. J. Strandburg. *Review of Modern Physics*, 60:161, 1988. doi: <http://dx.doi.org/10.1103/RevModPhys.60.161>.