Simulation and Visualization of collective dynamics

Author: Aitor Martin Gomez
*Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.**

Advisor: Dr. Ignacio Pagonabarraga and Dr. Albert Diaz-Guilera

Abstract: I present a computational simulation and the corresponding interactive visualization of a two-dimensional system of hard disk oscillators obeying Langevin dynamics and interacting. The visualization is carried out through a real-time 3D graphics package from Python called Visual Python, which allows visualization with a higher degree of interactivity. The Kuramoto model will be implemented as a mathematical model used to describe the synchronization of a large set of weakly coupled oscillators. The simulation results confirm that in the absence of hard core interactions the time required to synchronize increases linearly with the particle density. Excluded volume interactions lead to a more involved dependence of synchronization time with particle density.

I. INTRODUCTION

Synchronization is a collective phenomenon occurring in interacting systems and is ubiquitous in nature, society and technology. There are many systems in which cooperative dynamics emerge such as molecules, cells, flocks of animals[1], even remote-control devices[2] or people. Individuals interact in some way with the environment and with each other, exchanging and sharing information with the whole ensemble.

A marvelous example of this emerging phenomena has been studied in fireflies in South-East Asia forests. In places like Thailand, Malaysia or Borneo, a beautiful cooperative behaviour among male fireflies can be seen every night among the Mangrove trees that grow along the river banks. Specifically, male fireflies modify back and forth it inner clock in order to sparkle in perfect synchrony and, therefore, to reinforce their message to females[3].

Some scientists tried to find how this synchronization could be understood and modelled. The most successful attempt was proposed by Kuramoto, who studied a model of phase oscillators running at arbitrary frequencies which were coupled depending proportionally to the sine of their phase diference. Depending on the distribution of natural frequencies, different synchronization scenarios can develop. Fortunately, the Kuramoto model is simple enough to be mathematically manageable despite being a non-linear dynamic equation. Nonetheless, the model is rich enough to display a large variety of synchronization patterns and rather flexible to be adapted to many different contexts[4].

Tipically, the Kuramoto model has no spatial depencences and it assumes an infinite range interaction between the oscillator. Otherwise, it is also common the study of fixed coupled oscillators in a lattice with a interaction between the nearest neighbours. We depart from previous studies in order to consider a system with moving oscillators in a topology that is changing constantly. Taking into account these relevant mechanisms of synchronization introduces a higher degree of complexity into our system.The use of a mean-field interaction will provide us a good approximation to this complexity. For that reason, the developement of an interactive tool of visualizations allow us to tackle the problem of the study of these new scenarios in a more flexible way.

Recent studies have enlightened the relevance of these synchronization states thus making them a clear target of study. Hard-disks' potential, temperature, and the volume fraction will be our control parameters in order to analyse the dependence of the synchronization in different interaction regimes.

To conclude, we have seen that in our simulations the dispersion of the dispersion of the average phase difference decays exponentially over a long time when synchronyzation is reached. So, we will be able to define a characteristic time T and estimate this parameter as a function of the volume fraction of the system[6]. This fitting will show us how synchronization depends on volume fractions and how the results will vary with the relevant control parameters.

II. KINETIC MODEL FOR HARD DISKS MOVING OSCILLATORS

A. Hard disk model: Volume fraction ϕ

My system consists of a bidimensional assembly of N hard disks of radius σ in a rectangular box of linear and variable sizes $2L_x$ and $2L_y$, using periodic boundary conditions. The hard interaction between the disks implies that their overlap is forbidden. Therefore, the system is thus uniquely characterized by the packing (or area) fraction,

$$\phi = \frac{\pi N \sigma^2}{4 L_x L_y} \tag{1}$$

We have implemented an option that allow us to toggle the potential. It is very useful to compare what really

*Electronic address: aitormg93@gmail.com

happens to the system with hard disks or point particles.

B. Langevin dynamics

Langevin dynamics is a mechanical model in the limit where the friction with the environment rules the dynamics of the particles. This dynamics describes the random motion of large particles surrounded by a great amount of tiny particles which make them collide and divert their paths. This equation has the contribution of the frictional, potential and the noise term. The thermal noise ensures the system remains in thermodynamical equilibrium in the absence of external forces.

$$m\dot{\mathbf{v}}_i = -\gamma m\mathbf{v}_i(t) - \sum_{j=1} \boldsymbol{\nabla} U(\mathbf{r}_i - \mathbf{r}_j) + \mathbf{F}_i^r(t) \quad (2)$$

where γ denotes the damping coefficient associated to the firtcion force, $U(\mathbf{r}_{ij})$ is the particle interaction potential and $\mathbf{F}^r(t) = \eta\boldsymbol{\xi}$ a random force composed by the amplitude $\eta = \sqrt{2\gamma K_B T}$ and a random vector, $\boldsymbol{\xi}$, with components independently drawn from a flat distribution in the interval [-1,1]. This uniform distribution function is equivalent to the Gaussian random noise since both have the same second moment. The random noise satisfies the fluctuation-dissipation theorem,

$$< F_{i,\alpha}^r(t) \cdot F_{j,\beta}^r(t') >= 2\gamma K_B T \delta_{\alpha\beta}\delta_{ij}\delta(t-t') \quad (3)$$

By Stokes-Einstein's law, the short-time diffusion coefficient for a single particle will be $D_0 = \frac{K_B T}{\gamma m}$ and the relaxation time of the friction coefficient $\tau = \frac{m}{\gamma}$.

Then, τ quantifies the time scale in which the intertia is relevant. We must compare this time scale with the diffusion time in which the particle covers its size. The comparison will indicate us whether the inertia is relevant or whether we are in the over damped regime. This regime is described as Brownian dynamics. Therefore, we are interested in long-time dynamics which allows us to remove the inertial term that corresponds to the momentum derivative.

C. The Kuramoto Model

This is a widely used synchornization model in physics that makes several assumptions, including that there is a great amount of weakly coupled oscillators, that these oscillators are identical or nearly identical and that interactions depend on the sine of the phase difference between each pair. Following these simple rules, Kuramoto came to the conclusion of this refined model,

$$\frac{d\varphi_i(t)}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N} \sin(\varphi_j(t) - \varphi_i(t)) \quad (4)$$

where N is the total number of oscillators (the model is solvable exactly, at least in the $N \to \infty$). K is the coupling strength (considered equal for every one) and w_i are the natural frequencies.

In this paper the dynamics of the oscillators does not have an infinite range and it is considered a finit range interaction in which the oscillators interact with different local neighbours any time step. Hence, I have introduced an interaction range based on the center-to-center distance $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, for which only the pairwise within a distance between each other $d_{ij} < d_{link}$ can interact to approach their phases.

$$\dot{\varphi}_i = \omega_i + \sum_{j, d_{ij}<d_{link}}^{N} \frac{K}{N_{ij}} \sin(\varphi_j - \varphi_i) \quad (5)$$

where N_{ij} stands for the sum of the j neighbors of the oscillator i. below a treshold, critical value ,K_c, the system does not synchronize. Increasing K it reaches the threshold, where a phase transition emerges spontaneously from incoherence to synchronization. An order parameter is defined as well, which is conveniently used to measure synchronization,

$$re^{i(\Psi-\varphi_i)} = \frac{1}{N}\sum_{j=1}^{N} e^{i(\varphi_j-\varphi_i)} \quad (6)$$

where r measures the coherence of the oscillator population and Ψ is the average phase. In the limit of $N = \infty$, the amplitude of the parameter vanishes when the system does not synchronize, and it is positive and close to 1 in synchronized states, reaching the maximum value $r = 1$ when the system is perfectly synchronized (this case occurs solely if the oscillators are identical, with the same natural frequencies)[4].

In my simulations, the dispersion of the average phase difference defined as,

$$< \Delta\varphi >= \sqrt{\frac{2}{N(N-1)}\sum_{j<i} (\varphi_j - \varphi_i)^2}, \quad (7)$$

is calculated globally and we find that it decays exponentially after an initial transient. I am able to study this asymptotic behavior, defined as the characteristic time T (related to sync) in such a way that $\langle\Delta\varphi\rangle \propto e^{-t/T}$ and estimate T by fitting the numerical data of $< \Delta\varphi >$[6].

III. METHODOLOGY

A. Implementation

It is crucial to set up the system units in order to do the data analysis in reduced units. Distance is measured in units of the size of the cell R_c with which makes up the whole grid of size $4L_xL_y$(it will be developed later). R_c must be a number inversely proportional to N and

smaller than Lx and Ly. The energy expressed in units of K_BT and mass in units of the mass of a single particle m.

On the one hand, numerical integration for the computational implementation requires the use of finite time steps for the diffusive motion of the particles, Δt.The evolution of Eq.2 must be expressed in terms of discretized increments,

$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) - \tfrac{\gamma}{m}\mathbf{v}_i(t) - \tfrac{1}{m}\sum_{j=1}\nabla U(\mathbf{r}_{ij}) + \tfrac{\sqrt{\Delta t}}{m}\cdot\eta\boldsymbol{\xi}_i(t) \quad (8)$$

from which we can update the particle positions, $r_i(t+\Delta t) = \boldsymbol{r}_i(t) + \boldsymbol{v}_i(t)\Delta t$.
On the other hand, the numerical integration of finite time steps for the evolution of the phases of the oscillators, τ_p, derives from Eq.5 to the following expression,

$$\varphi_i(t+\tau_p) = \varphi_i(t) + \omega_i\tau_p + \sum_{j,d_{ij}<d_{link}}^{N} \tfrac{K\tau_p}{N_{ij}}\sin(\varphi_j(t)-\varphi_i(t)) \quad (9)$$

Since mobile agents emit signals at discrete time lapses, each phase is updated at discrete time steps $\tau_p = \Delta t$ to simplify the model.

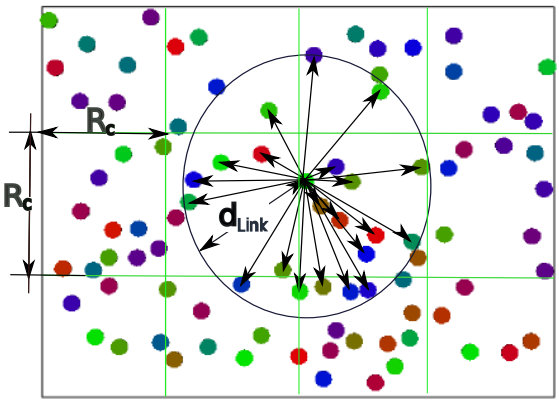

FIG. 1: The pairwise interactions with a cut-off distance d_{link} for a single particle is computed by dividing the domain into the green cells of linear size R_c. Here it is represented a zoom of the grid.

The hard-disk's potential is not continuous and, therefore, neither derivable. Thus, it cannot be implemented in Eq. 8. Conveniently I add the contribution of the overlap directly calculating the distance between the superposition of the areas and adding the half part of it to each particle in the collision to the opposite directions. Velocities are also modified by the equation obtained from the two-body elastic collision equations in two dimension.

$$\Delta\boldsymbol{v}_i = \frac{(\boldsymbol{r}_i-\boldsymbol{r}_j)\cdot(\boldsymbol{v}_i-\boldsymbol{v}_j)}{(|\boldsymbol{r}_i-\boldsymbol{r}_j|)^2}(\boldsymbol{r}_i-\boldsymbol{r}_j) \quad (10)$$

It is very common in molecular dynamics simulations to implement algorithms to reduce the computational cost due to the scaling of the program as $\theta(N^2)$ generated by the $N(N-1)$ interactions of the whole ensemble every time step. For solving this issue the "Cell lists" algorithm is suitable (See [7]).

Cell lists work by subdividing the main domain into M cells with an edge lenght R_c greater or equal to the cut-off radius of the interaction (or influence) d_{link}. Foremost the particles are randomly placed into these cells of the built grid and the interactions and phases influence are computed only between the particles in the same or neighbouring cells.(See Fig.1)

At the first step in a simulation, an array named `NUMPARTA` of length `len(NUMPARTA)= Num-cellmax` is constructed of all the particles that belong to each cell. At the same time, a matrix `CELLA` runs over the ensemble and is built of the `INDEX`(the label of the belonging cell), in the x-axis, and of the number of particles per cell, `NUMPARTA[INDEX[i]]`, in the y-axis. Thus, the construction happens to be `CELLA[INDEX[i],NUMPARTA[INDEX[i]]] = i`. Thereafter, it is only necessary to run loop for each molecule i and the program identifies all the neighbouring particles in each cell as `max-j = int(NPARTA[INDEX])`. Now we can run all the j neighbours taking into account that only are necessary the values of `(jj = cella[index,j]) > i` to avoid repetitions. Over the next few time steps, the lists are updated and used in the interaction evaluation routine.

B. Visual Python

Visual Python is an IDLE (Interactive development environment) based on Python Programming Language used to create very interactive and navigable 3D displays, toolbars and real-time graphs and animations.(To see different examples[8]).

We chase to develop an environment in which we are able to visualizate the motion of the particles, their inner phases and the connectivity. We considered quite important to vary the relevant physical parameters of the model "on the fly" in order to analyse interactively the impact of the range, the interaction, the strenght K, the diffusion, etc. Moreover, visualization is powerful enough to enable us to see how synchronization emerges and how it is spatially distributed and how does it evolve in time.

The main tools that have been required along the visualization are: **Objects** , **Widgets**, **Displays** and **Visual Graphics**.At the beginning I started creating N cylinders randomly settled (avoiding overlaps) and confined into a flat box of area $4L_xL_y$. The command, `particle[i] = cylinder(pos=Pos[i], radius=R,color=col[i])` produces the i-th disk centred at location `Pos[i] = (x[i],y[i])` with radius R and color `col[i]`. The positions and colors of the disks are updated simply by modifying the objects, `particle[i].pos = Pos[i]` and `particle[i].color = col[i]`. For a proper and clarifying interpretation I

have made up a RGB round chromatic scale and, accordingly, I have assigned to each disk a color as a function of its inner phase (between $[0, 2\pi]$).

After creating a controls window, you can set in the following control objects that will appear in that window. You can drag "Sliders" to enter numeric values graphically (the coupling strength K or K_BT) and put "Buttons" to toggle links uniquely between the interacting oscillator within the cut-off distance d_{link}.

We must also introduce the display where the entire "scene" will be developed, `display(title="Brownian Motion",width= w3, height= h3,...)`. It is also possible to place a graph in a VPython window by specifying `VGraph= gdisplay( title= ...)` and plot functions on it.

IV. RESULTS

Given a simulation with N particles, the number M of cells provided by the "Cell Lists" algorithm must be proportional to N (if N increases, so does the number of cells) and inversely proportional to R_c. Then, the cost of iterating two cells is $\theta((\frac{N}{M})^2)$ and the number of cell pairs will be proportional to the number of cells, which is again proportional to N. Hence, the total cost of sweeping all pairwise distances within a cut-off distance became $\theta(\frac{N^2}{M}) \sim \theta(N)$, which is more efficient to run all the particles twice and see which one interacts with each other.

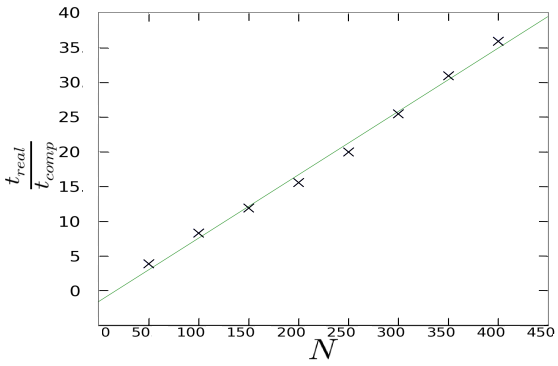

FIG. 2: Representation of the $\frac{t_{real}}{t_{comp}}$, where $t_{comp} = N_{steps}\Delta t$, versus the number of particles. I have adjusted a linear regression which seems to fit with the plotted values.

Notice that in Fig.2 the computational cost has been studied as a function of the number of particles in order to prove the effectiveness of the Cell Lists' algorithm.

The following part consists on verifying the exponential decay of the dispersion of the average phase difference [7] after the initial transient. The results showed that, for the different values of the volume fraction, the decay at long time(due to the persistent normal modes of synchronization) was clearly exponential. The assymptotic decay is exponential regardless of the type of interaction between oscillators. Accordingly, we represent it setting the log scale in the y-axis and demonstrated an almost linear behaviour as is shown in Fig.3. It is easy to understand that, for a fixed volume fraction, the non-interacting system has a higher diffusion than the interacting one, which encourages the system to a faster synchronization. So, in the first case the slopes are steeper than in the second one.

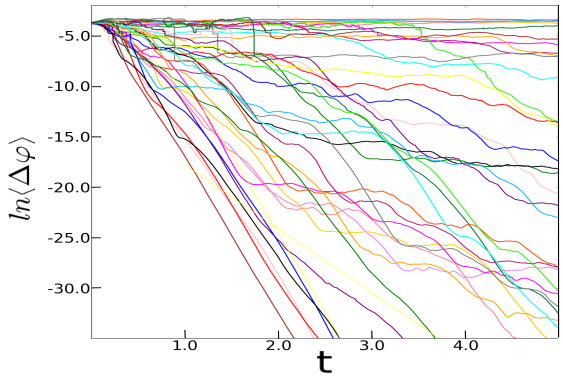

FIG. 3: The logarithm of the dispersion of the average phase difference of a non-interacting system (unitless) in front of the computational time (in units of $\sqrt{\frac{mR_c^2}{K_BT}}$), defined as $t = N_{steps}\Delta t$. The slopes are steeper for increasing volume fractions. The set up variables: $N = 150$, $K_BT = 100$, $\tau = 1$ and $K = 8$.

Going further, we noticed that for point partilces the higher the density is, the faster the ensemble synchronizes and hence, the steeper the slope becomes. In Fig.4 we can see the trend line, $\frac{1}{T} = a + c\phi$. Faster synchronyzation for higher densities occurs due to that the proximity of neighbours is notably reduced and the range of influence (cut-off distance) remains the same. Thereby, the oscillators have more neighbours to be in contact with and the whole flock reaches synchronization sooner. However, the linear dependence of the synchronization time breaks down with density. The system does not synchronize as quickly as expected and does it in a non-linear way. For high densities hard disks do not diffuse as easier as in lower densities. Consequently, there is a comptetion between the disk mobility, which determines how it samples space and changes neighbours, and the Kuramoto interaction.

In the Kuramoto models normally it is considered an interaction range interaction, corresponding to the limit $d_{link} \to \infty$. In our case we have analysed how affects the range of the interaction to the synchronization. The figure5 shows a first step into this direction. We have defined a value as a proportional part of the linear size of the cells, $d_{link} = nR_c$, where $0.5 < n \leq 1$.

V. CONCLUSIONS

Along this work I have tried to bring all my programming and physics skills into play and make the most of these resources. I have been studying different frameworks of synchronization of mobile agents in a system of Langevin Dynamics in an on-cushioned regime. In this

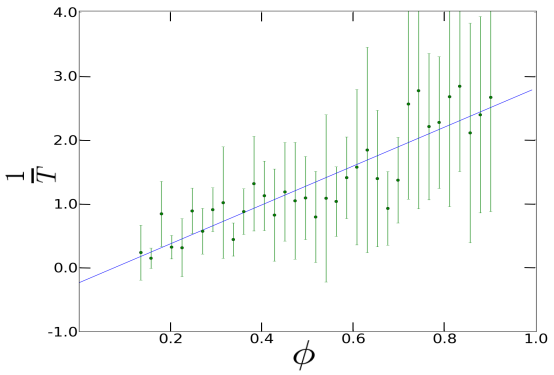

FIG. 4: The absolute value of the slopes of the lines represented in fig.3 as a function of the volume fraction. Where $1/T$ is in units of $\sqrt{\frac{K_BT}{mR_c^2}}$ and ϕ. I have fitted a straight line to emphasize the linear dependence. The set up variables: $N = 150$, $K_BT = 100$, $\tau = 1$ and $K = 8$.

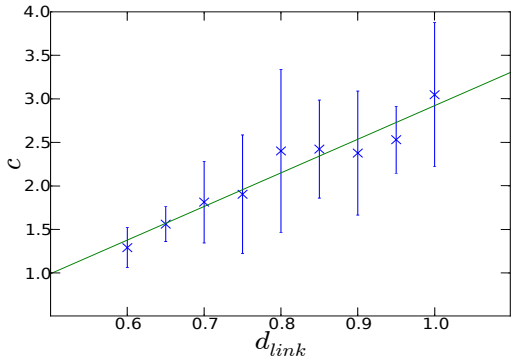

FIG. 5: The dependence of the slope of the regressions adjusted in figure 4 (the units of c are $\sqrt{\frac{K_BT}{mR_c^2}}$) respect to the value of the cut-off distance d_{link} (in units of R_c). I run the simulation varying the value of n for the same initial conditions.$N = 150$, $K_BT = 100$, $\tau = 1$ and $K = 8$

regime, the so-called Brownian motion, the agents are considered with no acceleration. The Kuramoto model was also implemented, which sets the phase transition from the incoherent state to synchronization. This model allowed us to study the evolution of the inner phases of the mobile agents.

Nonetheless, the main part of this work has been dealing with VPython and being able to create an interactive and real-time animation in order to address the problem from a complementary perspective that brings more flexibility to asses how the collective behaviour of the system depends on relevant control parameters. A tough part about it was to put in common different time scales. In the first place, I had to define a time step consistently with the diameter of the particles and the size of the box. In the second place, I had to make the visualization fast enough to be compatible at the same time with the evolution of the synchronization and the dynamics.

The system has been modelled for the two extreme cases of point particles and hard disks. We have shown that the interaction between particles plays an important role in sync. Particles with no interaction are able to go through each other, increasing their mean squared displacement and spreading all over the system. An infinite range interaction is more likely to happen and particles have more neighbour to which influence, making a global synchronization easier. Additionally, we have seen how the system synchronizes faster when we increase this range of influence.

Trying to fit some mathematical model to the dependence of the characteristic time T as a function of ϕ for the point particles(we do know at least that linearity is satisfied) and the case with hard disks' potential would be the next steps. A natural extension would be to study the implementation of self-propelled particles where much other scenarios of synchronization can occur. To conclude, I consider that this work has been very phenomenological but clarifying, and overall, very inspiring.

Acknowledgments

I give humongous gratefulness to my advisers Ignasi and Albert for their commitment and patience. Working simultaneously for two experts in their respective fields has been a very rewarding experience. I would also give my most sincere thank you to my colleagues Oscar, Laia, Enric and Araceli for their inspiring advices and moral support.

[1] J. Buhl, D. J. T. Sumpter, I. D. Couzin, J. J. Hale, E. Despland, E. R. Miller, and S. J. Simpson, Science **312**, 1402 (2006).

[2] A. Buscarino, L. Fortuna, M. Frasca, and A. Rizzo, Chaos **16**, 015116 (2006).

[3] Steven H. Strogatz, The emerging science of spontaneous order, New York, U.S, (2003).

[4] Juan A. Acebron, L.L. Bonilla, Conrad J. Perez Vicente, Felix Ritort, Renato Spigler, *The Kuramoto model: A simple paradigm for synchronization phenomena*, Reviews of Modern Physics, Volume 77, (2005).

[5] Demian Levis and Ludovic Berthier,*Clustering and heterogeneous dynamics in a kinetic Monte-Carlo model of self-propelled hard disks*, Phys. Rev. E **89**, 062301 (2014).

[6] Naoya Fujiwara, Jrgen Kurths, and Albert Daz-Guilera, *Synchronization in networks of mobile oscillators*, Phys. Rev. E **83** (2011) 025101(R).

[7] Allen, M. P.; D. J. Tildesley (1987). Computer Simulation of Liquids. Oxford: Clarendon Press.

[8] Equilibrium Stat. in Python, Leonard M. Sander (2013)