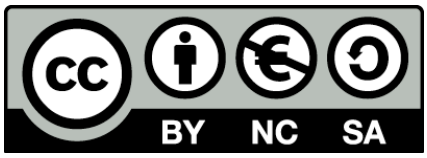
UNIVERSITAT DE BARCELONA

An experimental and theoretical view of autocatalytic colloids interacting with interfases

Lucas Santiago Palacios Ruiz



Tesi doctoral

An experimental and theoretical view of autocatalytic colloids interacting with interfases

Lucas Santiago Palacios Ruiz

UNIVERSITAT DE BARCELONA

An experimental and theoretical view of autocatalytic colloids interacting with interfases

Programa de doctorat en Física

Autor/a: Lucas Santiago Palacios Ruiz

Director/a: Samuel Sánchez Ordóñez,

Ignacio Pagonabarraga Mora

Tutor/a: Giancarlo Franzese

Facultat de Física

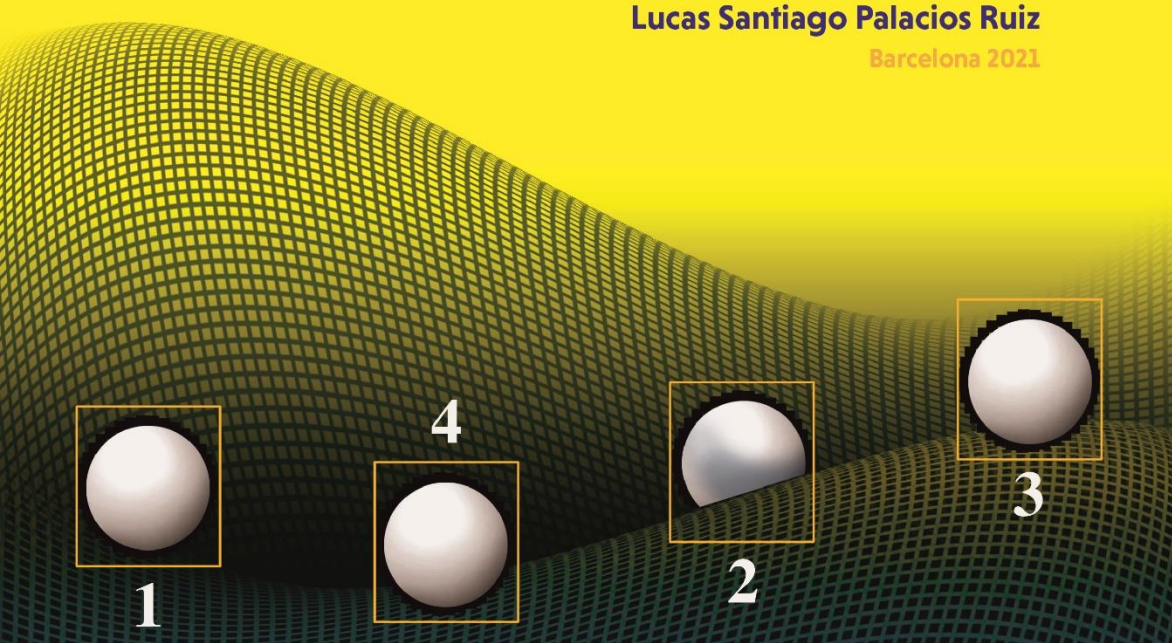
UNIVERSITAT DE BARCELONA

Doctoral Thesis

An experimental and theoretical view of autocataalythic colloid interacting with surfaces

Lucas Santiago Palacios Ruiz

Barcelona 2021



UNIVERSITAT DE BARCELONA

PHD program in Physics

An experimental and theoretical view of autocatalytic colloids interacting with interfaces

Lucas Santiago Palacios Ruiz

Ph.D. Thesis
Ph.D. advisors: Samuel Sánchez Ordóñez and Ignacio Pagonabarraga Mora
Barcelona, November 2021
Programa de Doctorat en Física
Facultat de Física
Universitat de Barcelona

Agradecimientos

How could you have a slogan like "freedom is slavery" when the concept of freedom has been abolished? The whole climate of thought will be different. In fact there will be no thought, as we understand it now. Orthodoxy means not thinking — not needing to think. Orthodoxy is unconsciousness.

One of these days, thought Winston with sudden deep conviction, Syme will be vaporized. He is too intelligent. He sees too clearly and speaks too plainly. The Party does not like such people. One day he will disappear. It is written in his face. [...]

To keep them in control was not difficult. A few agents of the Thought Police moved always among them, spreading false rumours and marking down and eliminating the few individuals who were judged capable of becoming dangerous; but no attempt was made to indoctrinate them with the ideology of the Party. It was not desirable that the proles should have strong political feelings. All that was required of them was a primitive patriotism which could be appealed to whenever it was necessary to make them accept longer working-hours or shorter rations. And even when they became discontented, as they sometimes did, their discontent led nowhere, because being without general ideas, they could only focus it on petty specific grievances. The larger evils invariably escaped their notice. The great majority of proles did not even have telescreens in their homes. Even the civil police interfered with them very little. There was a vast amount of criminality in London, a whole world-within-a-world of thieves, bandits, prostitutes, drug-peddlers, and racketeers of every description; but since it all happened among the proles themselves, it was of no importance. [...]

Freedom is the freedom to say that two plus two make four. If that is granted, all else follows.

Nineteen Eighty-Four

Resulta curioso ver como a mi alrededor ha habido más gente desencantada con la ciencia, que ilusionada. Gente joven, con toda su vida por delante, que entró con ilusión y ganas, y que año tras año se marchita, se desilusiona. Algunos saldrán, pero otros, con su intento de no querer aceptar lo que vivieron buscarán dubitativos un puesto posterior, aún dentro por temor a perder ese tren, poderoso, que anula a muchos. ¿Cómo es posible que gente tan cualificada acabe aguantando tanto? ¿Cómo es posible que lo que debería ser un sueño, después de tanto esfuerzo, se convierta en sus cadenas? ¿Cómo es posible que malvivan temiendo durante años? ¿En qué clase de sistema se ha convertido la ciencia en la actualidad?

Habrá, porque también he conocido, gente que haya tenido suerte, o que en su intento de no perder el tren lograron encontrar una mejor situación. No puedo decir que haya tenido la misma suerte que éstos, y no será por falta de paciencia. Mucho deberían de cambiar las cosas para que pudiese continuar, pero por mí, el tren puede marchar. Porque no es sensato vivir con tantos títulos, pero con tanta inseguridad en la vida. Tampoco lo es vivir con unas condiciones laborales tan precarias, sea por sueldo o derechos. Y menos, tener que aguantar el trato que muchos recibimos.

Algunos en este apartado dan las gracias por mera cortesía, porque deben, aunque no quieran. No seré tampoco uno de ellos. Hay quienes se consideran dioses en esta ciencia. Qué suerte no ser creyente.

Gracias a todos los que durante estos años habéis estado ahí, los de verdad, sea cual fuese vuestra posición porque queríais estar ahí y compartir conmigo momentos de la vida. He tenido la suerte de conocer grandes personas, aunque con muchas difiera en cuanto al postureo. Lo siento, nunca fui modelo.

A la familia, los amigos y compañeros, que, aunque sigan sin entender qué he hecho, más que ver lo vivido, han estado ahí.

A los colaboradores, que gracias a su empeño me dieron la oportunidad de poder terminar lo que un día quise hacer.

Al Estado, por haberme permitido durante estos años ser científico, a pesar de introducirnos lentamente derechos ya asentados para el resto de trabajadores.

Nihil novum sub sole.
Dente lupus, cornu taurus petit.
Acta est fabula.

A mamá, y a la yaya

Preface

This thesis presents my Ph.D. research in soft condensed matter physics, at the University of Barcelona (UB) between October 2016 and November 2021 while working in the Smart-Nanobiodevices group (IBEC) and Statistical physics group (UB). It covers the interaction of Active Janus particles with liquid-liquid and liquid-solid interfaces, not only from an experimental point of view but also from a theoretical one. The structure of this thesis is composed of three parts.

The first issue I address in this thesis is the need to obtain methods and setups for experimental analysis. This discussion is split into 3 different chapters. In chapter 2, I show how to create active Janus particles, but also the creation of new microfluidic setups needed for my thesis. After the setups are ready, it is important to properly record videos for the different experiments to perform, and to store them. However, these processes can be difficult depending on the characteristic parameters of the system, and we can record large quantities of data. Consequently, in chapter 3 I show how to record, store and compress the videos acquired in the microscope. Finally, once the videos are recorded, we must track the recorded particles. In chapter 4 I show how we can track these particles, from a basic tracking system based on traditional techniques to more advanced techniques based on neural networks.

The second issue I studied in this thesis addresses the interaction of these active particles with liquid-liquid interfaces. To study the basics of the problem I studied at a one particle level, or at most, a couple of particles. In chapter 5 I account my studies from an experimental point of view. The necessity of a better explanation of the experiments made me consider to develop a powerful model based on a Lattice-Boltzmann algorithm, which I show in chapter 6. In any case we learnt how active particles can be guided using these interfaces.

The third and last issue I studied explores how we can take use of the interaction of active particles with interfaces to obtain an accumulation system based on a complex geometry where the unit cells of the system do not break detailed balance. Because active particles behaved similar next to liquid-liquid interfaces compared to when they are next to solid-liquid interfaces, in chapter 7 I used solid-liquid interfaces to create the complex geometry.

Contents

Appendices

1

Introduction

Divide et vinces.

Gaius Julius Caesar

1.1 | Active matter

In the last years, *active matter*, a new field in physics, has emerged with the study of some *active entities* that create their own motion by transforming the energy they can take from around them. The fact that these entities rely on non-equilibrium systems has intrigued the scientific community and because their huge implications they can have in our daily lives, it has been clipped as one of the main studies in *soft matter* field. Because this thesis is based on active matter, in this section I will revise which are these active entities, why are they so special and why we should study them.

1.1.1 Surrounded by active matter

Motion. What is motion? Probably this question could seem trivial to any reader of today, but in fact, it has not been so obvious in the history of the humanity. Since millenniums ago, men have tried to answer to this question, no matter where they were living. Old manuscripts from different regions of the Americas [1], Europe [2] or Asia [3, 4] reveal different philosophies about motion along the history. Motion is a crucial question of man's curiosity because it surrounds us. Since we wake up, until we fall sleep, we can see motion. An incorrect answer to this question and all the physics will tumble. For example, Europe lived with Aristotle (384-322 BC) motion philosophy until Galileo Galilei (1564-1642 AD) discovered he was not so correct [5]. For Aristotle, motion was any kind of change, or the actuality of a potentiality, and he even differentiated the matter that moves as expected (*natural movement*) from that one that moves forced against its natural movement (*violent movement*) [2]. For him, a rock would fall because it is its natural movement, and it will fall faster than a feather because it possesses more "earth". Motion is essentialist. Opposite to him, Galileo showed how motion is conditional, that there is not such a distinction between natural and violent and that, if there is no air friction, the leather and the rock with the same weight will fall at the same time, dealing to a new era in the study of motion and physics [5]. Of course, since the Renaissance, science have advanced notably, and we have changed more its significance. Rationalism and the comprehension of mathematics have allowed us to say that, nowadays, any scientist could answer that motion is the change in the position of an object over time. Compared with the previous definitions, this definition is too mathematical. In some sense, it loses the idea of forces that push the objects.

It is due to physics that we can join the previous definition with the use of *forces*. As well as with motion, the concept of force has changed along history but no matter which definition we use, we could see in a glimpse that some forces are done over the objects, while others, are done by the own objects. Following with the rock example, a rock will move because something is acting over it, but it will never generate any force on its own. Opposite to it, there are objects that, although can be moved by other forces, they are capable to create their own necessary forces to move. Any object of this last group is what we can consider as *active matter*. Active matter is nothing more than an object, also referred as an *entity*, which transforms internal or ambient free energy into kinetic energy, and they do it locally, without being dragged by any global field [6]. Thus, becoming this entity as an *active entity*. Consequently, we can say that active matter is possible due to an out-of-equilibrium process since it needs a net flux of energy to move the entity. Following this description, we could think that many living organisms are nothing more but active matter. In this way, the study of active matter is crucial to understand us and our environment since we are surrounded by active matter.

1.1.2 Biomimicking

The history of men would have been totally different without these active entities. Since the beginning of the days, man have evolved coordinately with the dangers that surrounded him. Inventions such as flints, axes, weapons in general or the domain of fire would not have been as important if the animal kingdom would not have been as threatening to man. But animal kingdom is only the tip of iceberg. Although we cannot see them every day, there is a lot of life around us. As Feynman once said [7], there is plenty of room at the bottom. It is thought that 85% of the living species in the Earth are things we cannot distinguish with a naked eye, such as bacteria and protists [8]. It is true that it seems that these invisible entities represent only 15% of the total biomass distribution on Earth, but still, their biomass is around 1350 times the biomass that man represents on Earth [9]. With such amount of life, it is understandable that many processes we use to live better, or to enjoy life, are supported by them. Any kind of product where fermentation is involved, such as food, drinks, or other utilities, would not be possible without these invisible organisms. Even our digestion would not be the same without them. Of course, not everything they do is always good for man. In the Late Middle Ages, the Great Plague, a pandemic caused by *Y. pestis*, killed between 30% to 60% of Europe's population in that age [10] plunging the European Old World into a psychosis of terror and poverty. Nowadays, *Y. pestis* is not anymore important in our world but still, according to World Health Organization (WHO) [11], one of the top ten global causes of deaths by today is still due directly a bacterium, *M. tuberculosis*. Considering production and diseases, one could also think about smaller entities like viruses, but even if we could answer the traditional controversy of whether a virus is alive or not, we need to think that viruses cannot generate motion on their own and thus, they are not active matter. Even though,

we can still find smaller active entities such as the catalytic enzymes. Catalytic enzymes are found inside the cells and can be considered as active matter given that by their reaction they can self-propel [12]. Since their importance, if we want to take the maximum profit from living organisms, we need to study further how they behave.

Studying big animals can be very complex as they are strong and because their size, but it is very possible that they share common patterns with smaller organisms [13, 14, 15]. Thus, and because of how important they are, we will only focus on that active matter that is below the millimetre scale. In this scenario, taking care of them can be tough and weary and hence, if we can obtain what we want without growing and feeding them, it would be wonderful. Furthermore, even if taking care of them would be easy, they, as a system, are not. We cannot have a clear idea a priori if what we see in their behaviour is just because the underlying physics (due to hydrodynamics and phoretic fields) or there are some stimuli behind it. Luckily for us, advances in nanotechnology have opened the doors to build things at their scale and even replicate them [16, 17]. Of course, with the tools we have, we could invent any kind of machine at their scale, but nature does nothing in vain. Aristotle was wrong sometimes, but his idea of perfect nature was not so erroneous. If something is present in nature is due to some reason. Otherwise, natural selection would have erased something superfluous present in them. Since this process has been carried for very long time, the best we can do if we want to mimic this life to understand, improve and use differently the processes they do is to copy directly from nature. In the last years we have seen plenty of works[18, 19, 20, 21, 22] where many tiny artificial particles have been created to mimic this invisible life. Since life at this scale is active matter, these particles also had to be active matter. This is the reason why today, we can talk about artificial active matter, or *artificial active particles*. From among all the inventions related to artificial active particles, there are two properties in which the design has focused more: the shape of the particle and the mechanism of its propulsion [18].

Shapes and sizes

When looking at unicellular organisms, one finds few geometries in nature. Cylinders or spheres, this is what one mainly observes. Therefore, researchers focused the synthesis of artificial active particles on these two geometries (See Fig. 1.1A). Even though, we also did things differently. We have particles shaped as cylinders, but we also emptied them to have tubes [19]. We did the same with spheres to obtain empty spheres [20]. Even in the road to it, we did not removed everything inside, but just made them porous [21]. Furthermore, the surface of the particle can also be modified as reported widely. Many authors want to add either new abilities to these systems or to biomimic their surface as in the case of particles that biomimic virus capsids [22] (See Fig. 1.1B). Nonetheless, the capabilities nanotechnology give us has pushed us to go further, and we have even made very complex geometries such as ships[23].

Life in this world spans a couple of orders of magnitude, from tens of microns to hundreds of nanometres [18]. In the same way, many authors have conquered all sizes, and even they have been inspired in virus to go smaller [22]. From the world's record smallest nanotube jet [24] till the spherical microparticles I will use in this thesis. Remarkably, the size of the particle is very important to study the motion of the particles, and thus choosing a correct size for experiments is crucial. We will analyze this detail later, in section 1.4

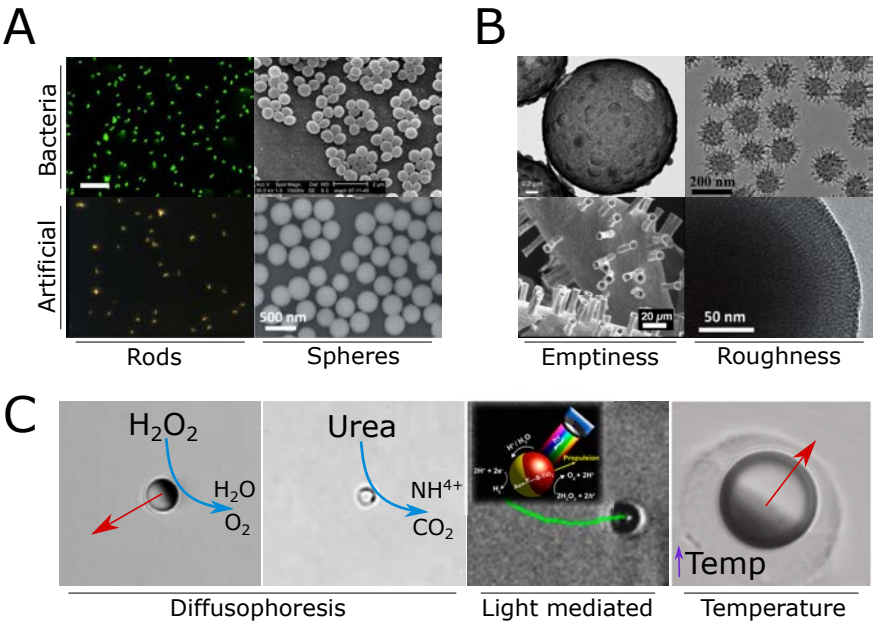


Figure 1.1: Shapes and mechanisms of propulsion. A) Artificial active matter looks like and behaves as bacteria. One could not distinguish from a glimpse which are bacteria such as *P. aeruginosa* (rods)[25] and *S. Aureus* (spheres) or artificial particles such as bimetallic Pt-Au rods[26] and enzymatic nanomotors (spheres)[21]. **B)** Our technology allow us to do several modifications. For example, we can void spheres[27] or rods[19] (emptiness images), creating holes along the particles, but we can also work on the surface by attaching different chemicals or modifying it as in the roughness pictures, where on top, a virus-like structure is presented[22] and in the bottom, a porous sphere[21] is presented. **C)** To add activity we can use different strategies, but many of them use the Janus-like structure as present in these images, except for the urea case. These mechanisms can involve diffusophoresis such in the Pt-SiO_2 particles (left) or enzymes such as urease to break urea. But also they can require light as in the Ti-Au particles[28] or temperature as in the case of using a a binary mixture of H_2O and propylene glycol n-propyl ether (PnP)[29]. Particles presented in C) are around a few micrometer size. *Image authorship. S. Aureus: Janice Haney, provided by CDC/Matthew J. Arduino, DRPH, 2001, public domain. Diffusophoretic particles: own images. Other images were taken from the cited literature. Images were cropped or no modified.*

Mechanisms of propulsion

Given the limited geometries life has in this length scale and, more importantly, due to the physics at this scale as we will refer in section 1.3.2, the number of mechanisms of propulsion are also limited. The typical mechanisms that are observed when prokaryote bacteria move inside a liquid are those found in *ciliated* organisms and in *flagellated* ones [30]. While the former use a system of very tiny hair-like structures called *cilia* that move in complex manners, the latter use bigger hair-like structures that move in a planar and wave-like manner to self-propel. In the case where the medium involves solid interfaces, we can also find a few extra mechanisms of motion including *swarming*[31], *twitching*[32] and *gliding*[33]. Swarming and twitching can occur on soft and moist surfaces. While in swarming a collective movement of many bacteria is needed, in twitching the phenomena occurs at a single bacterium level. Swarming occurs by the rotation of flagella of many bacteria, that also usually generate surfactants to lower the tension between the bacteria and the substrate. In contrast, twitching needs a special *pili*, the so-called Type IV pili, which binds to the substrate as a hook and retracts it to move towards where the pili is attached. In the case of the gliding motion, this motion usually occurs with firmer surfaces and implies a broad spectrum of mechanisms to work. Furthermore, an extra case is also typically found for eukaryotic cells given that they usually move over surfaces by using a ramification of their own body called *pseudopodia*. Eukaryotic cells are actively changing the pseudopodia by modifying its actin cytoskeleton to drag themselves over the surfaces [34].

In the case of artificial active particles, authors have tried to copy nature, but these mechanisms are difficult to replicate with the tools we have today, and simpler ways are much more efficient. For instance, different groups have tried to replicate the flagella motion. Dreyfus R. et Al. created a flagellum made of a chain of magnetic beads linked between them via DNA and attached to a red cell that move as a flagellum under the actuation of a magnetic field [35]. The addition of a magnetic field is not necessary to replicate flagella as Brian J. Williams et Al. showed how a spermatozoon made of polydimethylsiloxane (PDMS) can move by attaching cardiomyocytes that self-contract to it, reaching to speeds up to 10 μm/s [36]. In any case, because of its complexity, artificial particles do not typically base their motion in any kind of hair-like structures, neither on dragging over surfaces by modifying their body. Instead, researchers have simplified the level of complexity by using materials and alternatives to move that bacteria could not use but are easier to artificially build. The key ingredient required is to add an asymmetric motion mechanism to fulfil with conditions we will see in section 1.3.2. For example, many propulsion mechanisms are based on catalytic reactions, either by using metallic materials as Pt [37] or either using enzymes as urease [21] or lipase [38]. Reduction-oxidation (redox) reactions are also used and many times they require light to work [28]. Even there are particles which can need sound waves to work[39], special solutions to separate[29] or electric fields to work[40] (See Fig. 1.1C)). In all these cases, material usually remains constant in the surface of the particle but there are some particles that can degrade itself

to move [41]. In any case, it is common to see particles that have two areas: one where there is the donor material for redox or where the catalytic reaction occurs, and another one which is the acceptor material for redox reaction or simply, it is inert. This kind of composition is also referred as *Janus particle* due to the Roman God Janus [42], represented always as a biface head. Although the Janus structure is common, we can also build artificial particles without this structure. Examples of it are the new enzymatic particles where random clusters allow an asymmetry to move [43]. Finally, motion can also be achieved by bubble propulsion commonly seen with tubes. In this case, the use of surfactants is usually important to stabilize the bubbles. [19]

Pt-SiO_2 Janus particles are one of the most used ones among the different types of synthesized particles, not only because its easy fabrication but because they are easy to store, they last active for long times and given that they are Janus, one could in principle observe the direction of propulsion. These Janus particles displace due the catalysis performed by the Pt of the slow reaction of decomposition of hydrogen peroxide ($H_2O_2 \rightarrow H_2O + \frac{1}{2}O_2$) but there is still controversy why this reaction leads to the motion of these particles as discussed in section 1.4.3. The disadvantages to use these types of particles are due mainly two reasons: they are expensive to create since they need Pt and more important, the reaction of catalysis generate O_2, which is not well dissolved in water due to the low oxygen solubility in water. The accumulation of O_2 leads to the formation of bubbles in the solution. This is the main reason why these particles cannot be used for biocompatible systems. However, none of the objectives of this thesis rely on the biocompatibility of the particles, and only the bubbles by themselves were a limiting factor because they eventually burst, creating flows in the solution. Nontheless, the setups I developed and present in chapter 2 could manage this limitation, and hence I used this kind of particles.

1.1.3 Collective phenomena

One of the most exciting features that these active entities give rise is to *emerge* new phenomena when multiple entities are together, a phenomenon that by looking at each individual entity would not appear. This phenomenon is observed within living organisms at multiple scales with different levels of biological organization. At a microscopic level, cells of pluricellular organisms organize themselves to create *tissues*[44]. These tissues offer new phenomena that cells, by themselves, could not offer. For example, consider the epithelial and the muscular tissue. While in the epithelial tissue cells join by tight junctions providing a barrier, in the case of the muscular tissue, cells organize in a special distribution providing a mechanism for contraction and retraction. But these phenomena also occur at larger scales. At a macroscopic level, animals tend to move from one location to another one in populations. Typical examples are school of fishes [45] or flocks of birds [46]. Interestingly, when they move there is always some that lead the group. This *leading* is a behaviour that emerges from the collective.

As shown in the previous biological examples, these collective phenomena result from the interaction and organization of the active entities. In the case of artificial active particles these phenomena can also appear, and it is interesting to see whether which new properties can emerge and if natural collective phenomena behaviours can be reproduced without the proper life stimuli, but because of pure physics coming from symmetries and conservation laws. Indeed, natural phenomena at the macroscale such as *bird flocking* has been described by using a very simple model such the Vicsek model [47], where an entity just needs to align with their neighbours to emerge this phenomenon (See Fig. 1.2, first column). But flocking is not the only natural process scientists have recreated. Processes seen at the microscale such as *quorum sensing* and *tissue wound healing* have also been reported.

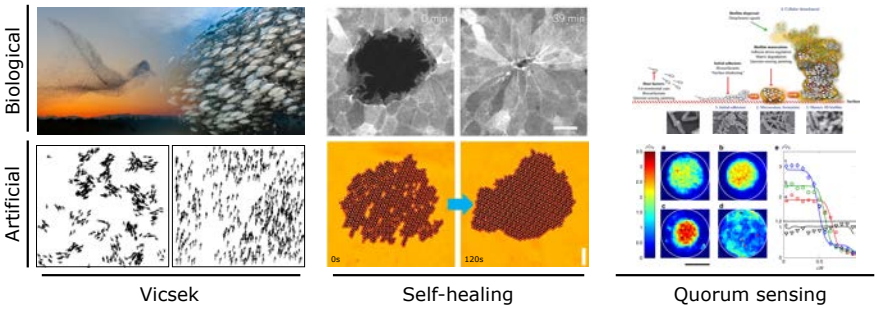


Figure 1.2: Collective phenomena is typical of active matter entities. This behaviour is found both in biological entities (upper row) and artificial entities (lower row). Examples of it include the collective motion of birds, known as swarms, and fishes, known as schooling, the closure of a tissue[48], or the development of a biofilm by using the quorum sensing mechanism[49]. These phenomena can also described with artificial entities as shown widely [47, 50, 51]. *Image authorship. Swarm: Daniel Biber, Germany, Shortlist, Professional competition, Natural World & Wildlife , 2018 Sony World Photography Awards. Schooling: Coral reef image bank, The Ocean agency, 2021. Other images were taken from the cited literature. Images were cropped or no modified.*

Tissue wound healing is an important biological process done by living tissues to self-repair themselves [48] since the loss of cells in a tissue can drag to the loss of function of the tissue and must be repaired. If we imagine a tissue such as the epithelia as a 2D lattice, in our synthetic world, if we control this process, we will have 2D lattices of particles without any defects that compare to crystals, where particles imitate the atoms. H. Massana et al. showed recently how by using paramagnetic beads and employing a couple of magnetic fields, it is possible to heal these lattices filling the defects with particles to obtain lattices without defects [50] (See Fig. 1.2, second column).

Quorum sensing is a communication mechanism present in microorganisms that is used to regulate their gene expression depending on the cell population density [52]. Typically, bacteria recognize products released by other bacteria and when these products are over a threshold, different genetic pathways are regulated. Thus, the size of the population will control the amount of product to overcome the threshold. Recently, researchers have reproduced the typical quorum sensing by using light activated active particles [51]. Since the chemical sensing is difficult to reproduce, they decided to hide this process and just look at the population around particles. These particles move actively depending on the radiation input given by a laser, which considers the density of particles around each particle. By using this interaction, T. Bäuerle et Al. have seen how active particles can self-organize and create clusters depending on the neighbourhood density (See Fig. 1.2, third column).

These examples help us to understand why the study of collective phenomena within active matter has gained interested over the past years, and what we can expect by studing a collection, instead of an individual entity. As Aristotle would say, it will be truth that the whole is greater than the sum of its parts.

1.1.4 Wet and dry active matter

If motion is inherent to the concept of active matter, there will be always a main concern surrounding every active entity: the medium that surrounds the active entity. Any active entity we are used to is surrounded by a *medium*. When active entities move inside a medium, the entities must move through the medium causing two consequences. On the one hand, since entities must displace part of the medium, they will feel a drag that opposes to their own movement, causing them to lose their kinetic energy since it is converted into heat. On the other hand, the motion due to these entities can change the medium itself, easing them or not, the motion through the medium. Thus, the presence of the medium is very important to describe the motion of our entities.

Because of the previous fact, when one models active entities it could be a must to model explicitly also the medium. In these scenarios, where we model the medium, we say we work with a *wet active matter model*. But could is not always a must. There are some scenarios where, even though its importance, the model of the medium is not important, and the effects of the medium can be expressed into our active entities as a simply drag in the equations of motion. In these last scenarios, we say we work with a *dry active matter model*. Since both scenarios can be possible, one may be intrigued in which are the main differences between them and when we should apply one of both possibilities. The key difference between both models is related with the conservation of momentum. When hydrodynamic interactions are important, fluids should be accounted for, and total momentum of the system (active entities + fluid) is conserved. But hydrodynamic flows can be neglected in general on length scales bigger than $\sqrt{\eta/\gamma}$ [53], being η the viscosity

of the medium and γ the drag[1] that act over these entities because moving inside a medium. In those cases, we can omit model the fluid and thus, the total momentum of the system is not conserved.

Dry active matter models

Dry active matter models are usually used when the drag made by the medium dominates and the effect made by the medium is just to provide friction. For example, for systems where medium is porous, the active entities are constraint to move next to a surface or in general, when length scales are bigger than $\sqrt{\eta/\gamma}$ and we can forget about hydrodynamic flows. There are dry models for a broad range of length scales. For instance, the famous Vicsek model [47] describes big crowds of birds flying together in the sky. But other models can also work for the microscale. Synthetic active particles that move in a simple liquid confined in 2D have been described using the famous Active Brownian Particle model or ABP[54], which I will present in section 1.4.

Wet active matter models

Wet active matter models are usually used when the hydrodynamic interactions are important, or the medium does something else than applying a simple drag (e.g. flows due to phoretic currents in the presence of walls or Marangoni flows). These type of models are usually found for the microscale environment, and a couple of scenarios can involve either living organisms such as bacteria swimming [55] or synthetic active particles such as the Pt-SiO_2 I will use in my thesis interacting via hydrodynamic and phoretically with solid walls [56]. For example, we can use the Lattice-Boltzmann algorithm to describe these systems as I will present in section 1.3.3.

[1] Notice that this drag has units of force density/velocity.

1.2 | Beyond simple fluids

In the previous section we have seen how active entities are surrounded by a medium. The medium that surrounds these entities can be very simple, as water or air, or much more complex, as a porous media or a gel. Typically, experiments with active particles are carried out in water, and if particles have a similar density to water, particles will move within the bulk of the system. But nature is more than just water and can be very complex. For example, a simple way to create complexity is to create boundaries, as I studied in this thesis. The introduction of these systems is very interesting not only because they are present in nature, but also because they give complexity to our system by introducing either solid-liquid or liquid-liquid interfaces. Although making experiments with liquid-liquid interfaces is more difficult, lately they are gaining terrain in this field due to its rich environment. Hence, I started studying these interfaces, to which I will present results in chapters 5 and 6, but by the end of the thesis I also used the solid-liquid as I will present in 7. In this section we will see why interfaces can introduce this complexity and what these interfaces can bring us.

1.2.1 Introducing interfaces: Definitions and properties

Any substance present in our daily life is composed of atoms or molecules. The interaction and the density of these atoms and molecules allow us to define regions of space where, at larger scales, different physical properties are uniform. When these properties are uniform, we say that the substance present in the region is in the same *phase*. Moreover, since inside the phase properties are uniform, we know that there will not be any exchange of energy or matter within it, and hence we can also say that the phase is in *thermodynamic equilibrium*.

Although there exist many kinds of phases [57, 58], it is typically said that matter can only be in four phases, named as *plasma*, *gas*, *liquid* or *solid*. Interestingly, from these four phases, plasma, gas and liquids are also encompassed under a same category recalled as *fluid*, because any substance in any of these three phases will deform or *flow* under an applied force [59].

In nature, phases can coexist together. When phases coexist, there are regions of the system where they contact. To these regions we call them phase boundaries or *interfaces*. Interfaces are very interesting because these are regions which do not have either all the properties of one or the other phase, and can bring new

phenomena to the system. For instance, at interfaces we can find a surface tension, which will generate wetting and capillarity effects, mainly seen for liquid-solid interfaces and marangoni flows, which born at liquid-liquid interfaces.

Surface tension

Due to the density and interactions[2] in liquids, atoms, or molecules (which I will refer to them as particles in general), have first order neighbouring, but without a fixed position since they are constantly moving randomly in the medium. Even though they are constantly moving, if we take a particular particle in the bulk of the liquid, we can think that the sum of forces exerted by its neighbours over time is zero on average, since there is *isotropy* in the medium: Everything is surrounded by the same kind of particles, no matter at which direction you look at. But isotropy is not found in all parts of the liquid. When one gets into the liquid surface, things are different because on the other side of the surface, either there is nothing (void) or either there are different particle species that interact differently[3] with the particle you are considering.

In this window of space, where we deal with an interface, particles will have an imbalance of forces and hence, they will inevitably feel a higher attraction to be in one side, pushing other inner particles and generating with it an internal pressure. Since the density of particles is not as high as in a solid, particles will have enough space to accommodate and hence they will move towards the bulk of the phase. But at the same time, since these particles push down, other particles inside the fluid may go to the surface to let them space, and a dynamic flow is stablished. At some moment, both fluxes will be balanced, and this will happen when particles at the surface are packed enough, originating two consequences. The first consequence is that, to pack more, they need to get closer, and hence, surface will be reduced. This is the reason why, for instance, water droplets in the space are spherical. The second consequence is that if we look the horizontal forces for a given particle in the surface, although the net force is zero on average, they have created more chemical bonds with their neighbours since they have less particles around. Thus, if we separate a bit any of these particles, they will want to return to its equilibrium position stronger than other particles in bulk. Hence, there is an inherent *tension* in the surface that is parallel to the surface that prevents other particles from outside the surface to move into the bulk. This is the reason why, for instance, if you put a steel clip on a surface it will not sink, and the same applies for mosquitos that do not sink even they are sit on top of the water. In Fig. 1.3 we can see a sketch of the process, and another example. We refer to this tension as the *surface tension*, which may found written with the symbol γ.

[2] Consider cohesive interactions such as Van der Waals, sulphur bonds or, especially for water, hydrogen bonds. In general, these forces are considered jointly with the Pauli exclusion principle as a unique potential, the Lennard-Jones potential.

[3] Consider adhesive interactions.

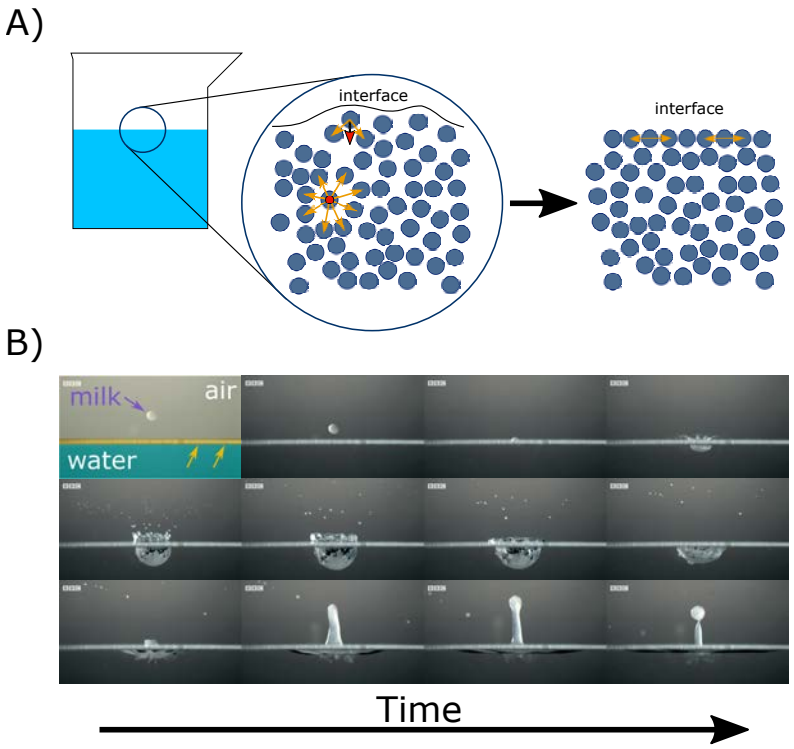


Figure 1.3: Surface tension is a consequence of spatial limits. **A)** At a molecular level, water such the one presented in this beaker can be modelled by these blue circles. Each circle represents a different water molecule. Water molecules are moving and feeling interactions (yellow arrows). Inside the bulk, the average interaction (red arrow) is zero, but in the frontier with the air, there is a net force towards the bulk. This force pulls molecules from interface to inside, and therefore, molecules at surface gets closer. Moreover, molecules at the interface will create more bounds between them since there are less particles around. This will originate a horizontal tension, and these molecules will want to strongly recover their equilibrium position to equilibrate forces after some hit. Thus, there is a surface tension parallel to the interface. **B)** As an example, I present some frames of a slow-motion video where a droplet of milk impacts water (blue area). Interface is marked with yellow arrows. The surface tension can stop the motion of the droplet and bounce it back as if it were an elastic membrane. *Image authorship. Milk-water frames: Obtained from [60]. Scheme: Own images.*

From a thermodynamic point of view, the idea of this internal pressure is the same as an excess of internal energy [4], which leads to an excess of free energy. The more surface area, the more particles will be able to move, increasing their speed, and hence this excess of energy will be higher. Therefore, we can define the variation of this excess of the Heltzmoth free energy F with respect the change

[4] See Annex B.1.2 for a brief refresh on basic thermodynamic concepts.

of surface A as the previous surface tension γ, which can be written for constant temperature T, volume V and n particles as:

$$\gamma = \left(\frac{\partial F}{\partial A}\right)_{T,V,n}. \tag{1.2.1}$$

Wetting

In the last example with water, we supposed that the interactions between water molecules were stronger than the interactions between water molecules and air. This hypothesis is appropiate because air is a gas, and thus air molecules will interact less strongly with water molecules. But what if the other medium interacts more with water, such with a solid? In this situation water molecules could interact stronger with molecules of the other media rather than with other water molecules. Therefore, depending on the repulsion or atraction of these interactions, water will enlarge or reduce its volume. Hence, we will observe that the fluid spreads over the solid or avoids contacting the solid. This phenomena is what is known as *wetting* and is described by a meaning of an angle, the so called *contact angle*.

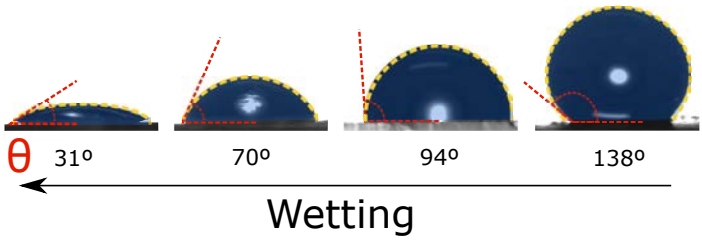


Figure 1.4: Wetting effect. The more spread is the fluid over a solid surface, the higher the wetting is. Contact angle θ (in red) helps us to define the quantity of wetting the system has. The contact angle is the angle between two surfaces. The first is the surface between the droplet and the solid. The second, between the droplet and the air surrounding the droplet, and must pass through the most extreme point of the droplet contacting the solid. *Image Authorship. Modified from [61], licensed as CC BY 4.0.*

To define the contact angle, we just need to consider a droplet of fluid in contact with a solid. If we take a perpendicular plane to their interface, we will see that the droplet will be spherical, but cut up to a fraction. The less strong the interaction with the solid is, the more fraction of a sphere will be. A very simple manner to define it is by drawing two surfaces and take the angle between them. The first surface is the one between the solid and the droplet. The second is between the droplet and the air surrounding the droplet. This surface must pass through the outermost point of the droplet in contact with the solid (See Fig. 1.4 for a few examples). There are deeper relations relating the contact angle as a function of different free energies, but these are not needed for this thesis.

Capillarity

In Fig. 1.4 we already saw what happens if we have a droplet in contact with a solid wall, but what if instead of a droplet we have much more liquid? And what if is it constraint to a fixed space? For instance, imagine we have a tube, which we put inside a glass full of water and parallel to gravity, as in Fig. 1.5. On the inner surface of the tube, molecules of water will try to be in contact to the surface or avoid it, depending on its wetting.

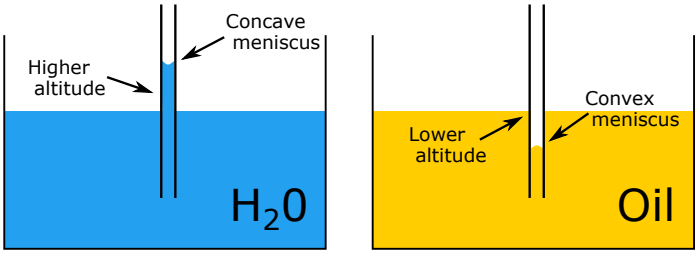


Figure 1.5: Capillarity effect. Because of wetting, liquids in capillary will mould their surface creating a meniscus. If the contact angle is acute, the meniscus will be concave, and liquid will push upwards (left). If the contact angle is obtuse (right), the meniscus will be convex, and the liquid will push downwards.

If the liquid wets the surface, molecules will try to contact the solid surface, and since the liquid flows, more and more molecules will be in contact, pushing the liquid to the top of the tube. If the tube is wide, only a little portion of the liquid will go up, and we will see as if the liquid pretends to climb in the inner surface. But if the tube is very narrow, e.g., we have a *capillary*, then this effect of "climbing" is much more important, and we will see two consequences. First, we will not have a flat surface of liquid inside the capillary. The surface is concave and is what we refer as *meniscus*. Second, the liquid will start to push towards the top of the capillary, until the gravity balances this force. This is typical for glass capillary with water.

Opposite to it, if the liquid does not want to wet the surface, particles of the liquid will try to avoid the surface of the inner capillary, and we will see not only a convex meniscus, but also that the column of water inside the tube is below the surface of water in the glass. For instance, you can see this if instead of water, you use oil in the same capillary. To these set of physical phenomena where surface tension and the wetting play a relevant role we call it as *capillarity*.

Marangoni flows

So far, I have explained phenomena that occur due to the surface properties, but at all moment, we have considered that the surface energy is constant along the surface. Although this is a good argument to start dealing with interfaces, there are

systems in which this hypothesis does not fulfil. For example, solutes in liquid will interact with molecules of the liquid, changing their surface energy and tension. If the concentration of solutes is not uniform in the liquid, this will lead to a gradient of surface energy. The same applies if, for example, other molecules which we refer as *surfactants* are present in the liquid. Surfactants are just compounds which will reduce the surface tension of an interface, such as detergents.

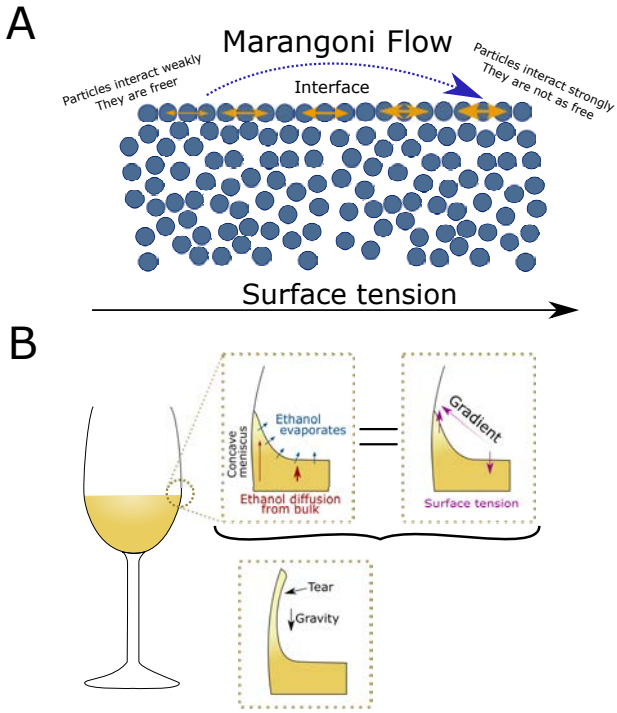


Figure 1.6: Marangoni flows are originated due to gradients of surface tension. A) Particles of liquid prefer to live in regions of higher surface tension, since they are stronger interacting with their neighbours. If a region with lower surface tension is present, particles will move to higher surface tension areas, creating a Marangoni flow. B) Classical description of alcohol tears in a glass of wine. These tears are a combination effect of capillarity forces and Marangoni flows. Alcohol evaporated will be replaced by alcohol coming from bulk, but it will take more time to diffuse from to the upper parts in contact with the glass, and thus these regions will have a higher surface tension. This will create a gradient of surface tension leading to a Marangoni flow. *Image authorship. Own images. Wine image based on [62].*

When a gradient of surface tension appears, a new phenomenon also appears, the so called *Marangoni flows*. Marangoni flows are flows that born at interfaces. Since there are areas where the tension is lower, particles of liquid of these parts will interact less with their neighbours than particles at higher surface tension. As

a consequence, particles contacting higher surface tension areas will be tighter to them than to lower surface tension areas and, little by little, a flow of liquid particles will be stablished from lower surface tension areas to higher surface tension areas, which it receives the name of Marangoni flow (See Fig. 1.6A).

Sometimes, Marangoni flows relate to capillarity effects. For instance, Marangoni flows are typically explained by looking at the alcohol tears that appear in a glass of wine. Wine in a glass presents capillarity, since there is enough wetting to see a concave meniscus. But wine can be sketched as water with alcohol. Alcohol at interfaces will evaporate faster than water, and thus, the less alcohol, the more interaction water molecules will have between them and the higher the surface tension will be. At the flat interface, although alcohol is evaporated, there is a fast incoming of alcohol from the bulk, but at the borders of the glass this income is lower, and thus the surface tension is higher. This will lead to a gradient of surface tension, letting the wine to climb more in a thin layer until gravity will oppose it and eventually falls in the shape of droplets, which people refer as tears (See Fig. 1.6B).

1.2.2 From "simple" to smart interfaces: Mixing active particle with interfaces

Creating an experimental setup where these active particles interact with solid surfaces is very easy since the active particles I will use sediment due to their density mismatch with water. Therefore, they will be always confined on top of a solid wall, moving in a 2D plane and hence, they will be always interacting with a solid-liquid interface which we will consider as the bulk. When particles sediment, it is easy to observe that without any fuel, they face with the Pt side towards the wall[5], and they remain still (See Fig. 1.7A) bulk, inactive). But when particles start moving, things change. Under an external flow, particles will roll, and we will see from the top how the Pt side appears and disappears as if it were the phases of the Moon [63]. But if we activate these particles by adding its fuel to the solution, the configuration is totally different. Under activity, particles start to selfpropel but interestingly, they always face with the same Moon phase and we can constantly see both sides of the Janus particle (See Fig. 1.7A) bulk, active), even if a flow is superposed to its own motion[64]. If we define the director vector $\boldsymbol{n}$ for the particle, which is normal to the plane that separates the inactive from the active side, it will be almost parallel to the wall along the experiments. The angle quenches[63].

As we will see in section 1.4.3, the interaction of the solutes surrounding these particles with themselves is the key point to understand their selfpropulsion when they are at bulk, but this interaction also help us to explain why particles with fuel

[5]The Pt side has a 10 nm thick layer of Pt in addition to the SiO_2, but both sides have the same quantity of SiO_2. Because of Pt density is much higher than the SiO_2 one, this layer represents about 12.5% of the total mass of the SiO_2 for 1 μm radii size or 5.0% for 2.5 μm radii size, which are the sizes I used. This is enough for any fluctuation to flip particles to this configuration.

have this configuration next to the walls. When the particle generates the solute, this solute will be more concentrated next to the wall, and there will be more in the active side rather in the inactive one. If we assume that the surface of the particle interacts repulsively with the solutes, and more importantly, that the active side does it stronger than the inactive, then this inhomogeneity in the solute concentration will lead to two consequences, both making the active side to leave the wall. On the one hand, since the interaction is stronger with the active side, the particle will feel a torque, placing the active side to the top. On the other hand, since on top of the wall there is a gradient of solutes, this gradient will imply a chemiosmotic gradient that will uplift the particle. But as we said before, the active side is heavier, and hence it will want to go closer to the wall by gravity. Thus, a counterbalance is created, and the particle achieves this stable orientation. Notice that, even we did not consider electrostatic interactions in this balance, they have been experimentally discarted as the main cause of rotational quenching[63].

The strong phoretic interaction also allow these particles to interact with steps they find (See Fig. 1.7A) interface and B)). Particles do not cross but stay for some time next to steps if they are high enough. Thus, in small spaces, particles can stay more time within this second solid-liquid surface rather than in the bulk, and hence one can see the importance of the study of these particles next to interfaces. For particles of $R = 1$ μm or bigger, this altitude is about 400 nm. If particles do not cease their activity while they are next to the steps, they will continue moving and therefore, one can use this property to guide them from point A to point B (See Fig. 1.7C)) without the needing of an external field such a magnetic one, which is always a complex structure [63, 56]. Just the shape of the step is enough. Of course, if a simple step is constructed, particles will be able to go from A to B or B to A, but we can also give a directionality to these steps as seen with the ratchet structures Jaideep designed (See Fig. 1.7D)). This directionality was further studied in chapter 7 by creating a special path that allowed us to reconstruct a topological insulator where particles behave as electrons in a classic topological insulator. This further step also helped us to understand how to create flows of particles even if in principle, the structure should not show a net directionality, and how the topology of the lattice where particles move can accumulate particles in one side of the lattice. This work mixed two different fields, where we could show phenomena of topological insulators using active particles, something that to my knowledge I am not aware of other similar studies. As with the previous definitions for interfaces, in the next section 1.2.3 I will explain a few details on what a topological insulator is.

Although guiding the particles with a solid-liquid interface can be useful, not all the interfaces in nature are solid-liquid interfaces as we could see with the example of the oil-water mixture. In the case of having two fluids, the phenomena are richer than the solid-liquid because the interface can be non static and different phenomena such Marangoni flows can appear. Moreover, the description of the entrapment of the particles at the interface is due to the accumulation of solutes

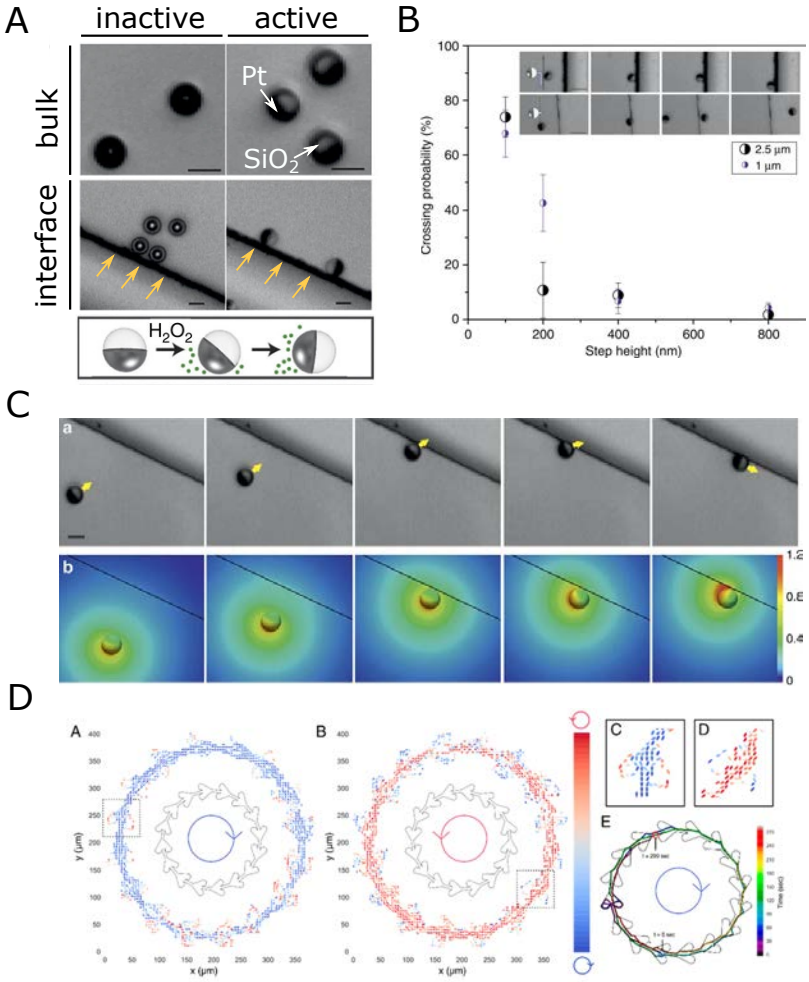


Figure 1.7: Janus particles at the presence of solid-liquid interfaces. A) When SiO_2-Pt particles sediment (Bulk, on top of a surface), they show different configurations depending on if they are active (Janus structure) or not (homogenous colour). When they find an extra solid-liquid interface such a solid step (interface), they move along them[56]. **B)** If the solid step is very small, particles will jump over them, but for steps higher than 400 nm and particles of $R = 1\mu$m and $R = 2.5\mu$m, they mostly do not cross them[56]. **C)** When particles find a step that cannot cross, they turn and move along the step[56]. **C.a)** Experimental data **C.b)** Simulated data showing the concentration of O_2 produced by the particles. **D)** By designing an intelligent shape for these interfaces, one can guide them as necessary. Here I show the implementation of the ratchet structure where we can guide particles to turn right (blue indicate clockwise motion) or left (red indicate anticlockwise motion)[65]. *Image authorship. All images were taken from the cited literature. Images were cropped or no modified.*

next to the surface. In the case of both fluids, one may think that there can be no accumulation, since in both mediums the solute can diffuse. Thus, in principle is not secure that the interactions of the particles with these interfaces will be the same. Following this reasoning, I decided to study the behaviour of active particles next to liquid-liquid interfaces, where the solid step shown in Fig. 1.7C) is now another liquid. Thus, my system had a solid-liquid-liquid interface. The experimental results will be shown in chapter 5.

This kind of system is a reasonable scenario for particles that sediment, but that at the same time can encounter with a liquid-liquid interface because the particle is in a channel and suddenly there is a clog by other fluid (e.g., lab on a chips) or simply, because there are colloidal droplets in the environment, such as oil droplets. Moreover, notice how in this setup gravity is parallel to interface. Commonly, studies with active particles at liquid-liquid interfaces are made at pure liquid-liquid interfaces, and hence gravity is perpendicular to the interface. Although it is true that in my case we have an interaction with the solid, in the case of a pure liquid-liquid interface particles are trapped to the interface due to gravity. Hence, effects of the interaction can be hidden by the gravity acting on the particles. For example, X. Wang et Al.[66] studied a water-air interface perpendicular to gravity, where Pt-SiO_2 particles move. Their most important findings were an increase in the speed of particles and a reduction of the rotational diffusion, making particles to move in straighter trajectories compared with when they swim free in the media. In a subsequent article, the same author [67] analysed for the same system what happens when your particles actively rotate. Their findings were similar, with a reduction of the rotational diffusion, except for fast actively rotators, where the rotational diffusion increases. Recently, K. Dietrich et Al. [68] studied a similar system where air was substituted by an oil. Opposite to them, their particles were based on polystyrene (PS) instead of SiO_2, and they found two kind of populations that move differently (high and low speeds), which depend on how the Pt cap is oriented, and gave two values of diffusivities. As Wang, they found a reduction in the rotational diffusion due to contact-line pinning, and hence a large persistency of particles motion as X. Wang obtained. Both authors agree that differences in the surface of the particles may lead to deviations on the behaviour in the same particles population.

To understand better the behaviour of active particles at liquid-liquid interfaces we can model them. Several studies have placed a colloid at a liquid-liquid interface, but the complexity of the fluid, the solute field generated by the particle and the particle interactions are not always fully resolved. For example, P. Malgaretti et Al.[69] considered that the activity of particles does not affect the surface tension, that this solute diffuses along the media without any preference to be at the interface or in one of the both liquids, that the catalytic part is only on one fluid, that the interface is flat and does not exert a force on the particle while moving neither a torque. Moreover, wetting of the particle is not considered. But A. Dominguez et Al. also showed that Marangoni flows may play an important role

on the motion of active particles near liquid-liquid interfaces[70]. Last year, T. Peter et Al.[71] used a fully resolve fluid method such as Lattice-Boltzmann to study these particles at fluid interfaces. However, this study lacks the possibility of having the products generated by the catalysis of the colloid more concentrated in one phase, and the control of the phoresis mechanism for the particle is not implemented. Since there are still many open questions, I decided to model this problem using a fully resolved fluid using Lattice-Boltzmann in coordination with a finite elements method to solve the phoretic field generated by the product because of the catalysis of the active particle, and to add as many elements on the colloids to model better their behaviour. The results of this work are found in chapter 6.

1.2.3 Thinking deeper: Playing with global properties

Probably one of the immediate ideas that comes after directionality is the ability to create and control flows. In the last chapter of this thesis, I analyze these questions by building a pattern composed of different ratchets. The interconnections between ratchets can stimulate the conduction and accumulation of particles even if the design would not allow us to think in a net flow. Thus, it is very interesting to explore how the topological properties of the pattern can affect the conduction and accumulation of particles. To explore these problems, we combined the fields of soft matter with topological insulators, where people in this last field experimentally work with electrons as their particles. Thus, with this study we were not only studying the directionality and accumulation of active particles, but we also were mimicking the behaviour of electrons in a topological insulator. To explain better what a topological insulator is, I will introduce next some key fundamental concepts.

Topological insulators

Among all the existing insulators in nature, there is a specific group which have interesting properties. Since they are insulators, we expect no conduction. However, if we focus on the edges of the material, we will see conduction. Overall, the conduction does not appear in the bulk, but appears at the edges of the material. Interestingly, in these insulators this edge conduction is topologically protected. Thus we refer to them as *topological insulators*.

Topology is a field of mathematics that studies how global properties of objects are maintained under continuous deformation, such stretching or bending, but not tearing or merging. For instance, if we take a sphere and a donut, both are topologically different because we cannot reshape a sphere into the donut without making a hole, which is not a continuous deformation. Nonetheless, we can reshape a donut into a mug, and a sphere into a cube. Thus, the cube will share a set of properties that are equal to the sphere, while the mug will share them with the donut. To define which objects have the same topological properties, we can define a set of numbers that we call *topological invariants*. For example, in the previous case,

the *genus* is an integer topological invariant that describes how many holes the object has (See Fig. 1.8A)).

In the same sense, a material could be stretched, or bent, but still, in a topological insulator we would see the conduction at the edges. This is a consequence of the topology applied on the band structure. Similar to how the donut is deformed into a mug, but it cannot be deformed to a sphere, the band gap in an insulator can be reshaped, but never transformed into a band gap of a topological insulator, or vice versa, because smoothly deforming a trivial insulator into a topological insulator requires closing the gap, which is analogous to make a hole in the sphere. It is topologically restricted. Depending on the symmetry of the problem we can use different invariants. However, a typical topological invariant that describes the band is the *Chern number*, which can be fractional.

At the edges, we can imagine that electrons propagate in a wave-like manner. Thus, in the edges they can escape from the full Valence band and reach a different energetic level as if there was no band gap. Consequently, we can describe these propagative states as propagative edge states, or simply, *edge states*. Because they connect the valence and conduction band, these edge states carry an electric current. Their topological nature implies that the current is dissipationless with a quantized conductance robust against disorder. Due to their utility, research aims to discover new materials with these properties[72]. Researchers have discovered analogues replacing electrons by light[73] or sound[74] for example. The experiments carried out in chapter 7 want to find also an analogue using active particles, which up to my knowledge has not reported in the literature[6].

A first approximation to an edge state is to consider a 2D gas of electrons, to which you apply a perpendicular magnetic field. From a classical perspective, these electrons will start moving in circular orbits. However, at the edges, the image of the orbit is broken, and they start to move along the edge in just one direction[7]. From a quantum perspective, we recover the idea of energy quantization because there are a discrete energy values available as if we were considering the energetic bands. In the bulk there is no conduction, as in a material with a full Valence band. However, at the edges we close this band gap. In this sense, we could think that this system is like an insulator in the bulk, but conductor at the edge, as the topological insulators. But electrons have a quantum property called spin, that can be 1/2 or -1/2, and that interacts with a magnetic field. Hence, when the magnetic field is applied in one sense, the spin of these electrons is defined as one of these two values. However, imagine we record this experiment and then we play it backwards. The first you will notice is that the magnetic field now goes backwards. Consequently, electrons will have the opposite spin. This means that if we play backwards the film, the electrons have changed their spin. Thus, this experiment depends on if we go to the future or the past, which means that we

[6] A close work could be ref. [75] because they use colloids, but these are not active colloids.
[7] This is the Hall effect.

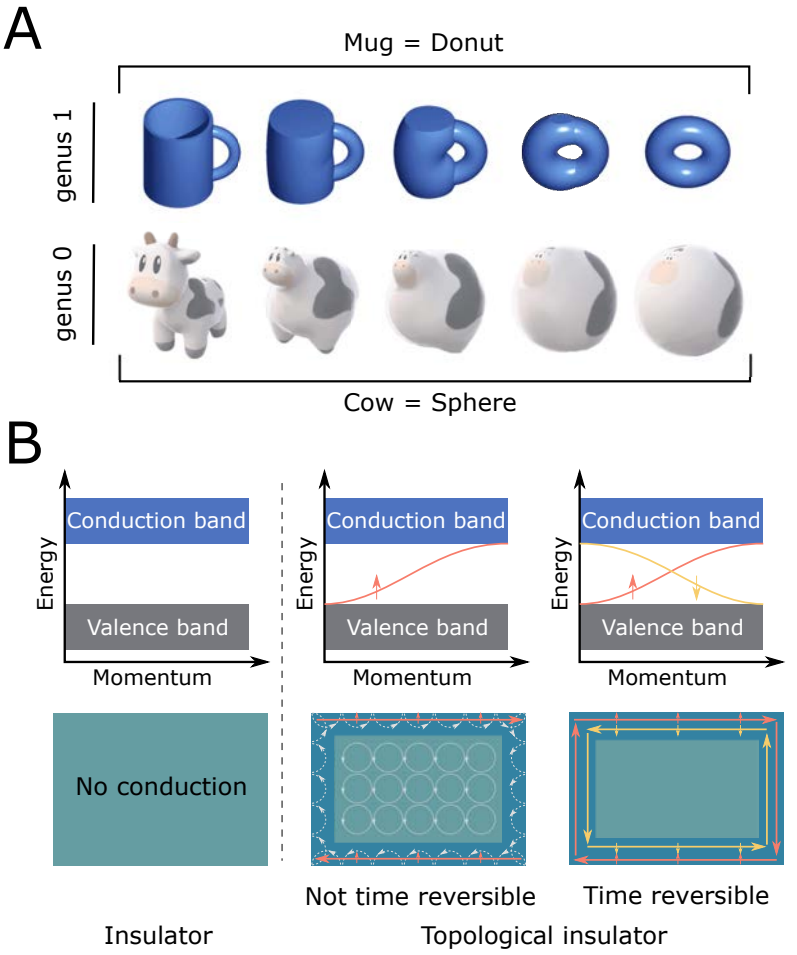


Figure 1.8: Topology applied to crystals. A) Topology is a mathematical field that differ objects by global properties. For example, the genus parameter identifies the number of holes. Zero genus means zero holes, and hence this cow and the sphere are in the same topological class, while the donut and the mug with one hole have different properties since their genius are 1. **B)** We can apply topology to the energy bands of a crystal. An insulator has a band gap and cannot conduct electricity within it (green blueish). In a 2D electron gas, if you apply a perpendicular magnetic field, electrons will do closed circular trajectories, and no conduction is found. But in the borders, it can conduct, with a spin (orange vertical arrows) defined by the magnetic field. This is the Hall effect, a topological insulator without time-reversal symmetry. To obtain a time-reversal symmetric topological insulator, we can have two conduction paths at the borders, each for a different spin of the electrons. The three cases are topologically different, as the genus in the previous example, here we have different paths to behave relative to the energy bands. *Image authorship. Mug and donut: Lucas Vieira, Wikimedia Commons, public domain. Cow and sphere: Keenan Crane, Wikimedia Commons, CC BY-SA 4.0. Rest: own images.*

broke time reversibly. Consequently, this class of topological insulators does not have time reversibly. Nonetheless, if instead of one edge state we have two, where one state has 1/2 for the spin value of the electrons and the other -1/2, we would see the same even if we play the film backwards, and therefore this would be a different class of topological insulator which has time reversibly (See Fig. 1.8B)).

The Su-Schrieffer-Heeger (SSH) model

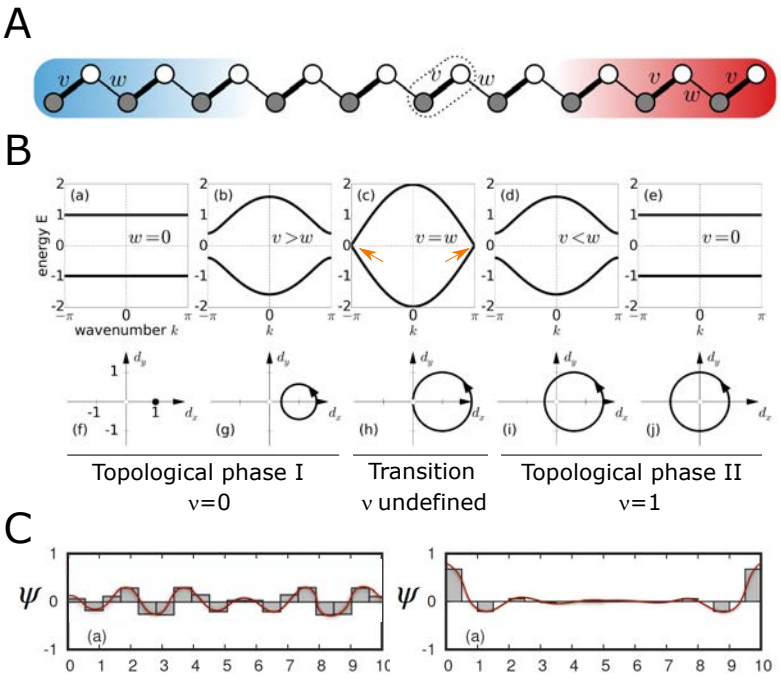


Figure 1.9: SSH model. A) The SSH model is a chain composed of N unit cells, where electrons can be in sub-lattice A or B. Electrons can hop from one to another one with a probability v, and to a different unit with probability w. **B)** In the bulk, the model has no conductivity, since there is a gap between both bands. However, when $v = w$ both bands join (orange arrows). Thus, in this case we have conduction. Although the energy bands seem symmetric for $v > w$ and $v < w$, this is not totally true. This symmetry appears at the level of the eigenvalues but looking at the eigenvectors ($\vec{d}$) the problem is different. The solution is drawn as a circle with radius w and shift to (v,0). In this plane, the conduction refers to point (0,0). The winding number ν of this curve using this point gives the topological phase. If the line crosses (0,0) we are in a phase transition. **C)** While in the conduction we see electrons moving along all the chain (left), in a topological scenario ($\nu = 1$, right) we only see this conduction at the edges. *Image authorship. Modified from ref. [76] (A,B) and from ref. [77] (C).*

One of the simplest models to introduce topological insulators is the Su-Schrieffer-Heeger (SSH) model[8]. This model describes fermions, such electrons, that hope on a chain of N unit cells. Each unit cell is composed of two sites (A and B), and hence we can also divide our problem into two sub-lattices (A and B). The electrons can hope with a probability v inside each unit cell, and probability w between different cells (See Fig. 1.9A)).

The inclusion of two sub-lattices introduces the two-band structure we already have seen (Valence-Conduction band). But to describe the topological insulator we need to consider several extra points. First, as in band theory, we will assume that electrons move as a plane wave. Consequently, they will not interact among them. Moreover, to solve the bulk behaviour we will select the same boundary condition as in the band theory, the Born-von Karman. Finally, as in the topological insulators behaviour, we will assume the spin is polarized, and hence we will not include the spin in all the procedure. With these key-ingredients the model can predict conduction in the bulk if $v = w$ and insulating otherwise[9] (See Fig. 1.9B)). In fact, the conduction mode can also appear because of a topological transition. We can see this transition using the *winding number*, which is a topological invariant. The winding number is commonly used when one has a curve in a plane and a singular point. If we centre in that point and then we start following the trajectory of the curve until we reach the starting point, the winding number would indicate how many times we have make a complete turn around the singular point. In this case, each topological phase has a different winding number, while in the conductor ($\nu = \omega$) is undefined (See Fig. 1.9B)). To the phase with $\nu = 0$ is called *trivial*, while $\nu = 1$ is called *topological*. The difference between both is the number of *edge states*. In the topological phase ($\nu = 1$) we can see electrons localizing at the edges, while this is not observable in the trivial ($\nu = 0$) case.

Skin effect and states of higher order

The SSH model is the simplest model to introduce topological insulators. However, it is not the only model that there exists. For example, the Hatano-Nelson model is another famous model that can introduce different concepts. From a mathematical perspective, the SSH model includes an *Hermitian hamiltonian*[10], while the Hatano-Nelson[78] includes a *Non-herminitian hamiltonian*. Non-hermiticity is necessary to see dissipative effects[79], and hence, it will be useful when studying our active particles.

As a consequence of non-hermiticity, different edge states will occur in the system (see Fig. 1.10), known as the *skin effect*. The main difference between hermitian and non-hermitian edge states is that non-hermitian modes are very sensitive to the boundary conditions. If L^d is the size of the system, being d the

[8] See ref. [76] for a full mathematical description.

[9] This is the Peierls transition.

[10] A square matrix which is equal to its own conjugate transpose.

dimension, then the number of hermitian edge states will be L^{d-1}, but there can be up to L^d skin states[80]. Moreover, when we increase the dimension of the system we can look at states that live on a different dimension. For example, in 2D systems we could find states that live on edges, but also some that live at the corners. In general, we will talk about states of order n, with $n \leq d$, being d the dimension of your system. We can find these $n-th$ order states in both hermitian and non-hermitian systems, but the difference in the number of states will remain.

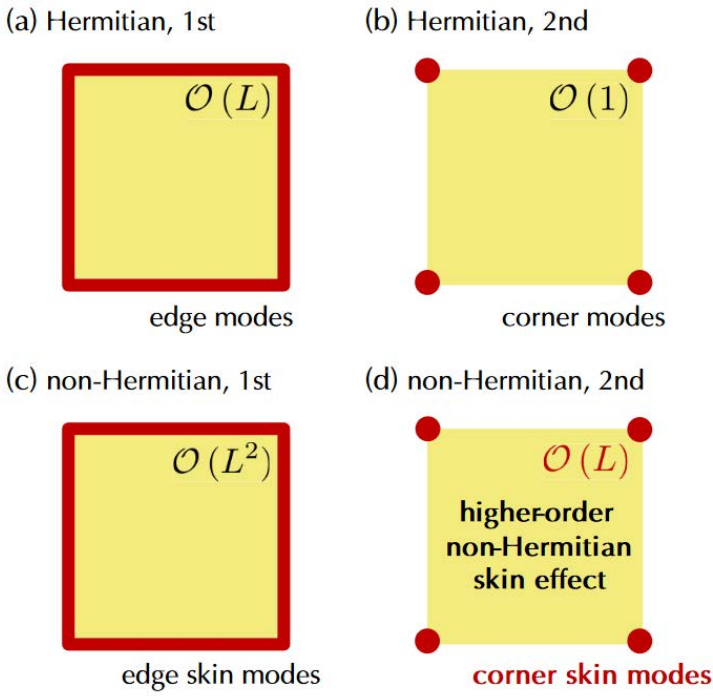


Figure 1.10: Skin effect. Different edge states appear depending on the order and if the Hamiltonian describing the model is hermitian or non-hermitian. *Image authorship. Taken from ref. [80]. No modifications were done.*

1.3 | Modeling the surrounding medium

Working with a wet active particle system imply the model of the surrounding medium, which is a fluid. To model a fluid, probably, the naivest idea we can have is to solve the equations of motion for each individual molecule that compose the fluid. But even in the simplest scenario, where there is a single liquid (e.g., water), this approach is not the more intelligent one and we should think whether if this approach is correct. Thus, should we solve the equations of motion for molecules? Which alternatives do we have?

In this section we will introduce different approaches to answer these questions. We will build a basic continuum approach where the fluid is a continuum field, and we will adequate this model to the experimental scale. Then, we will find a more complex but adequate model for modelling the experiments we perform, introducing liquid-liquid and solid-liquid interfaces.

1.3.1 A continuum approach for the surrounding medium: Navier-Stokes equations

At a microscopic level, fluids are composed by many particles. A simple underestimated calculus show how big is this amount: If we consider a low dense fluid (e.g. an ideal gas) at Standard Conditions[11], and we use the Avogadro number N_A[12] and the fact that 22.4L $=$ 1 mol for an ideal gas at Standard Conditions, then, for a similar volume as I will use in my experiments (10-100 µL) we have:

$$10\mu L \frac{10^{-6}\ L}{1\ \mu L} \frac{1\ mol}{22,4\ L} \frac{6.022 \cdot 10^{23}\ molecules}{1\ mol} = 2.7 \cdot 10^{17}\ molecules. \quad (1.3.1)$$

With such amount of molecules, if one aims to describe the fluid from the motion of these molecules will spectacularly fail because solving the equations of motion for such amount is extremely costly, and because at long runs, the system could have dynamical instabilities of phase-space due to small uncertainties in the initial conditions[13]. Just think about the number of interparticle interactions we should consider in such an enormous system. In fact, this approach is

[11] Temperature of 273.15 K and an absolute pressure of exactly 10^5 Pa.

[12] $N_A = 6.022 \cdot 10^{23}$ constituent particles/mol

[13] If δ_0 is an uncertainty in the initial momenta or position, the uncertainty will grow as $\delta(t) = \delta_0 e^{\lambda t}$, where λ is the Lyapunov exponent. At times greater than λ^{-1}, the uncertainty will grow enough to prevent any deterministic prediction.

what is known as *molecular dynamics* (MD) and nowadays is only applicable for much smaller volumes and for very short timescales[81] compared with the working scales of this thesis (ms to s). To avoid these computational problems, we can think our problem from a different point of view. For example, we could think that the concept of molecules is not needed for our studying purposes. If we could consider bigger constitutive particles for our system instead of molecules, we would reduce the number of particles we have to simulate. This idea is nothing strange but common in statistical physics, and it is called *coarse graining*. In fact, we already considered it by assuming we had to model the fluid from molecules, because molecules are made up of atoms, and at the same time, atoms are composed of subatomic particles such as electrons and quarks. The key point in this process is to consider the right scale we need to describe our problem. Describing the quark world will make even much more complex the molecular simulations but will apport nothing to the fluid, because the typical interactions at that level occurs at shorter time and length scales than what we will need to solve. For instance, while strong interactions inside the atom occurs at a time-scale of ys ($10^{-22} - 10^{-24}$ s) and in a range of fm (10^{-15} m), weak interactions occurs at a time-scale of almost ns-ps ($10^{-7} - 10^{-13}$ s) and in a range of am (10^{-18} m)[82], which still it is much lower than this thesis time (ms-s) and length (μm) scale.

In the case of the molecular level, interactions will occur at length-scales of Åm (10^{-10} m) and typically at time-scales of μs-ns ($10^{-6} - 10^{-9}$ s) that we will neither solve[83]. Thus, we should consider a less complex level, where we can describe the matter with bigger particles. Following this idea, physicists made the assumption of the *continuum fluid* [84]. In this assumption, we consider that our elementary particles in the system are a big packet of molecules of volume V. The size of V can be arbitrary chosen, but it must verify two limits. On the one hand, the size of the characteristic linear length for V ($V^{\frac{1}{3}}$), must be much larger than the characteristic length present in the molecular system, which we will refer as δ. For a simple liquid, such as water, we can take the typical interparticle distance (δ = 3Å). By assuming this limit, every property of the new elementary particles will be statistically a microscopical averaging. On the other hand, the size of $V^{\frac{1}{3}}$ cannot be bigger than the order of magnitude of the characteristic properties we want to see in our system, which we will refer as L. For example, active particles are around a few micrometres size and. Thus, to be conservative we could say that L = 0.1 μm.

But things look different when one zooms out. If the elementary particles are still very small, as we have, from a macroscopic point of view we can consider that they uniformly fill the space and compound a *continuum*. Thus, any of these particles can be treat as a mathematical point, for which different properties of the system can be calculated. Therefore, we refer to this hypothesis as the hypothesis of the continuum. A direct consequence of this hypothesis is that at a macroscopic point of view, we can assume that the medium will flow with a mean steady velocity field **u** with some fluctuations. Fluctuations arise by the fact of the

stochasticity of the particles at the microscopic level and give the idea of its *diffusion*[14]. Opposite to them, the **u** field describe the *convection* of the fluid. Hence, since we can define a field, we can forget about the elemental particles and look at this field, solving with this step the initial computational problem. This fact does not mean, of course, that this calculus is not computational costly, but at least, we can compute it in a reasonable time for large macroscopic sizes of fluid and long-time scales. By applying conservation of different quantities such as mass and linear momentum to each of these particles we can arrive to what is known as the *Navier-Stokes equations*[15], which model a fluid from this continuum point of view. For the scenario of an incompressible Newtonian fluid under the gravity acceleration **g**, these equations read as:

$$\begin{aligned} \rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) &= \rho\mathbf{g} - \nabla p + \mu\nabla^2\mathbf{u}, \\ \nabla\cdot\mathbf{u} &= 0, \end{aligned} \tag{1.3.2}$$

where p is the pressure of the fluid at point $\mathbf{r}$ and time t, μ is the viscosity of the fluid and ρ is the density of the fluid.

1.3.2 Reducing the scale to active particles scale: The Stokes equation

One of the main concerns regarding the Navier-Stokes equation is the fact that these are non-linear partial derivative equations (PDE) for which in general, there are no exact analytic solutions known. Consequently, if one wants to tackle this problem analytically, one may simplify the previous equations by looking over the magnitudes that play an important role in the system. To deal with this fact, the Navier-Stokes equation is often written dimensionless[16]. In this dimensionless form, there appear a few dimensionless numbers that are the ratio of some important variables of the problem. Hence, depending on the conditions, these variables will be small enough to vanish and we will obtain a simpler equation. With the correct dimensionless variables, Eq. 1.3.2 reads as follows:

$$\frac{\partial \boldsymbol{u}^*}{\partial t^*} + (\boldsymbol{u}^*\cdot\nabla)\,\boldsymbol{u}^* = \underbrace{\frac{L|\boldsymbol{g}|}{U^2}}_{1/\mathrm{Fr}^2}\boldsymbol{g}^* - \nabla^* p^* + \underbrace{\frac{\nu}{LU}}_{1/\mathrm{Re}}\nabla^{*2}\boldsymbol{u}^*, \tag{1.3.3}$$

where * expresses the dimensionless variable. In this new expression we can see two prefactors. The first one, associated with the gravity, is the so called *Froude number* (Fr), which measures how important is inertia with respect the gravity force. The second factor, associated with viscosity effects, is the *Reynolds number* (Re), which measures how important is inertia with respect the viscous forces.

[14] Although these particles are not molecules, at the size scale of these particles we can still think that our particles will move stochastically.

[15] See Appendix $A.1$ for a detailed derivation.

[16] See section A.1.1 for a detailed conversion

In the microscopic world, active entities will move at speeds of μm/s and sizes will be of μm order. In this sense, if we substitute these numbers in the dimensionless numbers, we can see that:

- For Froude, the gravity force dominates and thus, it will be present.

$$\frac{U^2}{L|\boldsymbol{g}|} = \frac{\mathcal{O}(10^{-12})\ m^2/s^2}{\mathcal{O}(10^{-6})\ m \cdot 10\ m/s^2} = 10^{-5} \tag{1.3.4}$$

- For Reynolds, the viscous forces dominate over inertia.

$$\frac{LU}{\nu} = \frac{\mathcal{O}(10^{-6})\ m\ \mathcal{O}(10^{-6})\ m/s}{\mathcal{O}(10^{-7})\ m^2/s} = 10^{-5} \tag{1.3.5}$$

Therefore, in the microscopic world both contributions must be present. But at the same time, we said that both contributions dominate over inertia. Thus, in this tiny world inertia will not play any important role and we can get rid of this term. In Navier-Stokes the inertia is introduced by the term $\boldsymbol{u} \cdot \nabla \boldsymbol{u}$ since it gives idea of the *advection* of fluid to an element of fluid of volume V. Consequently, we will remove this term[17]. Apart from dismissing the advection, the term $\partial_t \boldsymbol{u}$ neither appears in this tiny scale because we always consider that the fluid is in a *steady state*. This is, for a given point $\boldsymbol{r}$ in our fluid, the change in velocity only depends on the advection of fluid, but by itself, the speed of any point does not change with time. Consequently, Navier-Stokes can be written as:

$$\begin{aligned} \mu \nabla^2 \mathbf{u} &= \nabla p - \rho g, \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned} \tag{1.3.6}$$

which is also known as the *Stokes equation*.

The dimensionless of these equations imply that if the dimensionless ratios are the same, we can scale the system as we want. For example, many aerodynamic tests while manufacturing vehicles are done at smaller scales than the final product will be[85]. Thus, if we want to imagine what this Stokes equation means, we could put human daily orders of magnitude in Re number. If we want to have a $Re \approx 10^{-5}$ for a human walking, this would mean to have $\mathcal{O}(1)$ m and $\mathcal{O}(1)$ m/s, and therefore, we should have a kinematic viscosity of order $\mathcal{O}(10^5)$ m^2/s, which is really, really huge (See table 1.1 for the viscosities of different fluids). From the values I found in the literature[86, 87], only pitch was of this order of magnitude[18] (See Table 1.1).

[17] Note that this term can appear but linearized, which will end into an equation called the *Oseen equation*. Oseen equation gives some corrections on the Stokes equation, which are essential to explain the Stokes' paradox. Stokes' paradox says that, opposite to the scenario of a sphere that I will use in this thesis, an infinitely cylinder in a fluid will not have a non-trivial steady-state solution for the Stokes equations around it. The same applies for disks in two dimensions, where there cannot be stokes flows around them.

[18] Regarding the viscosity of pitch, it may be interesting to see the pitch drop experiment, one of the longest experiments ever run[88]. Since 1927, pitch in a controlled chamber is dropping, and due to its huge viscosity, only 9 droplets have fallen since the beginning of the experiment.

Fluid	Temperature (ºC)	μ (mPa·s)	ν (mm^{2}s)
Air	27	0.018	13.9
Water	25	0.894	0.894
Ethanol	25	1.074	1.36
Mercury	25	1.526	0.11
Honey	25	5000	3500
Pitch	25	$(2.3\pm0.5)\cdot 10^{11}$	$(2.1\pm0.5)\cdot 10^{11}$

Table 1.1: Some viscosity values for different fluids.

Translated to human scale, active entities that move in water is like for humans swimming in a pool full of pitch, which it is almost impossible to be aware of what this means. Just consider that honey, which is very viscous, is many orders of magnitude below the pitch. Even though, if we try to imagine this situation, we can understand that at the moment we would stop moving, we will stop immediately. Opposite to it, if we were swimming in a pool with water, at the moment we stop swimming we would still move a bit in the direction we were swimming. That scenario, which corresponds to high Re, is the one we are used to, opposite to this low Re scenario (See Fig. 1.11 for a graphic comparison). Therefore, one can see that the physics at this level will be different from what we are used to.

The first we must consider when looking at microscopic active entities is that active entities will only move either because they are dragged by flow currents or because they are actively creating their own motion. As us in a pool full of pitch, at the moment they stop their activity, they will stop moving. Furthermore, from Stokes equation we can see an interesting perspective regarding this motion. Since fluid is steady, and the equation is linear with $\boldsymbol{u}$, Stokes equation has time reversal symmetry [19]. This is, if the flow flows with speed $\boldsymbol{u}$ and suddenly we invert the forces and reverse the direction of the flow, $-\boldsymbol{u}$, we could reach to the same state as we had at initial time. This fact is very important for the active entities to move because if they have to move, they cannot move with *reciprocal movements*. This condition is also known as the *Scallop theorem*, after the work from Purcell[91], who stated that an animal with one degree of freedom in configuration space will not be able to move since all their movements will be reciprocal, but could move with two. He proposed a two hinge object, and to make clear his statement, he put the example of a scallop, therefore the name of the theorem. Since then, a few other works have also claimed new and very simple mechanisms to move, such as Ali Najafi et Al. using three linked spheres [92], or have been inspired in nature [93, 94, 95]. This is a very important fact since the evolution of life would have not been the same without the ability to move. Nonetheless, it is also true that we are only considering Newtonian fluids. In Non-Newtonian fluids, the viscous part of the Stokes equation would be different and thus, reciprocal movements could be valid as Tian Qiu et Al. show with their studies [96].

[19] See ref. [90] for a nice experimental demonstration of this phenomenon.

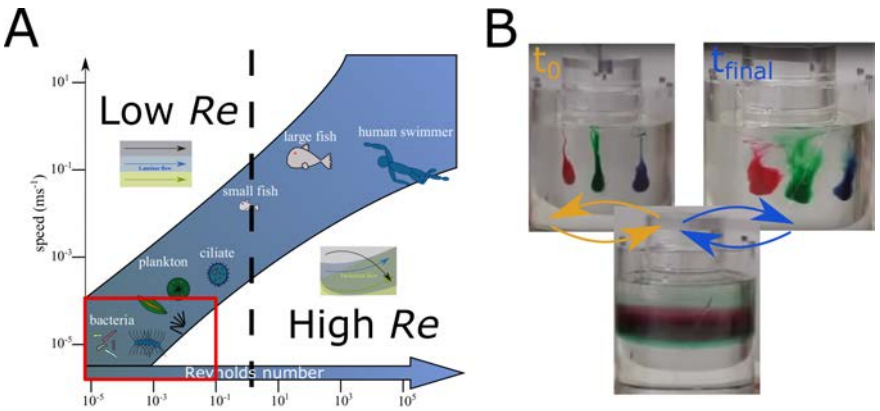


Figure 1.11: High and Low Re. The Re compares the inertial forces to viscous forces. If inertial forces are dominant, Re number is high. If viscous forces are dominant, Re is low. **A)** Re regimes for different things we live within our daily lives (in red our tiny world). At low Re, flows are laminar. Mixing of the compounds is not important, and sheets of flow are not mixed. At high Re it appears turbulence and everything gets mixed. **B)** At low Re one can recover the initial state by inversing forces and $\boldsymbol{u}$. Therefore, if the fluid is Newtonian, motion at this scale cannot be due to reciprocal motions. Opposite to it, at high Re we do not recover it, and hence we can move with reciprocal motion. At initial time t_0, three colourful droplets are unmixed. After rotating clockwise, droplets are mixed, and so colours are. But if one rotates counterclockwise, droplets are recovered, and we will recover each colour. *Image authorship. A) Modified from ref. [89]. B) Frames taken from ref. [90].*

1.3.3 Redefining the continuum: The Lattice-Boltzmann equation

In the previous sections we saw how we can model a simple liquid by using the Navier-Stokes equation or simply, the Stokes equation. With these equations we have simplified the problem of having millions of molecules by introducing a continuum velocity field $\boldsymbol{u}$ that characterizes the fluid motion. However, nature is not so simply, and these equations will need to present different terms and proper boundary conditions that may be complicated to formulate. For example, in this thesis I will work with active particles close to liquid-liquid interfaces and therefore, I will need to introduce these phenomena in the above equations. Moreover, once introduced, these equations will be numerically solved and thus, a proper discretization of the above should be correctly implemented. Furthermore, these equations will need to be parallelized, since this is how computer power has increased in the last years. Thus, if we could have a different algorithm that can be easily parallelizable and that at the same time, could be also easy to introduce different physical concepts, it would be very useful to model fluids.

Luckily for us, this method exists, and it is based on a much more statistical basis by solving the *Boltzmann equation*. The Boltzmann equation describes how particles evolve in a system of N equal particles, interacting or not, by looking at groups of particles instead of individual particles. The main parameter that it calculates is the distribution function $f(\boldsymbol{r}, \boldsymbol{v}, t)$, which expresses the number of particles at position $\boldsymbol{r}$, with speed $\boldsymbol{v}$ at time t. By looking at this parameter we can describe systems that do not necessary are at equilibrium[20]. Hence, we can study out of equilibrium, e.g., they have net fluxes of matter or energy. Of course, when dealing with a physical equation in a computer model one needs to discretize the space in a lattice, and therefore, it is understandable why this model has such an imaginative name: *The Lattice-Boltzmann (LB) model*[20].

In this new scenario, we need to imagine things differently from the idea how we based Navier-Stokes. Instead of solving differential equations for the $\boldsymbol{u}$ field, in LB we just play with a very naive system composed by N equal particles moving over the edges that connect the vertices of a lattice. These particles are not the same as the ones used to describe the Boltzmann equation but are useful to build a toy model that connects with the Boltzmann model. In fact, this kind of toy models have been used for many years in other algorithms such as the Lattice Gas Cellular Automata (LGCA).

In this toy model, for each time step particles move and collide, following a streaming and a collision operation respectively (See Fig. 1.12). First, in the streaming operation, every particle stays in its vertex or moves to a different one through one of the available edges. Thus, not only space and time are discretized, but speed is also discretized in a few i modes that depend on how the lattice is discretized. After particles have moved, the collision operator applies momentum and mass conservation for all particles at each vertex, which will decide the following streaming operation[21]. However, although in these models we use particles to explain them, we do not track these particles. Instead, because each vertex will have i different directions of motion, we store information of each direction for each vertex, which defines a *cell*. In older versions of this toy model such as in LGCA, each direction could be filled or not, being thus a binary value. Then, to have macroscopic values of each cell one should average information over each direction for different neighbours. However, LB goes a step forward, and instead of storing binary data, it stores a float value. This float value for each cell matches with the previous distribution function $f(\boldsymbol{r}, \boldsymbol{v}, t)$ from the Boltzmann equation. The following equation captures the overall process:

[20] See Appendix $B.3$ for a detailed derivation.

[21] Note how by splitting the evolution in these two steps, we can easily parallelize the problem. Collision step does not require information from other nodes and can be calculated locally. Thus, we can divide the problem into many individual problems that each processor will calculate. However, the amount of processors will be also limited because the more processors, the more information you will have to share during the streaming operation between nodes each time step, which can slower the process more than what one extra processor can faster it.

$$\underbrace{f_i^*(\boldsymbol{r}^* + \boldsymbol{v}_i^* \Delta t^*, t^* + \Delta t^*) - f_i^*(\boldsymbol{r}^*, t^*)}_{\text{Streaming operation}} = \underbrace{-\frac{f_i^*(\boldsymbol{r}^*, t^*) - f_i^{*0}(\boldsymbol{r}^*, t^*)}{\tau^*}}_{\text{Collision operation}}, \quad (1.3.7)$$

where $f_i^*(\boldsymbol{r}^*, t^*)$ is the equivalent for the distribution function $f(\boldsymbol{r}, \boldsymbol{v}, t)$, but since we only have a limited amount of i speeds modes, we have a limited amount of f^* functions and thus we can discretize them. In the streaming operation, we calculate the difference of f^* between time t^* and time $t^* + \Delta t^*$, where Δt^* is the time past between two-time steps. Since systems out of equilibrium tend to equilibrate, with

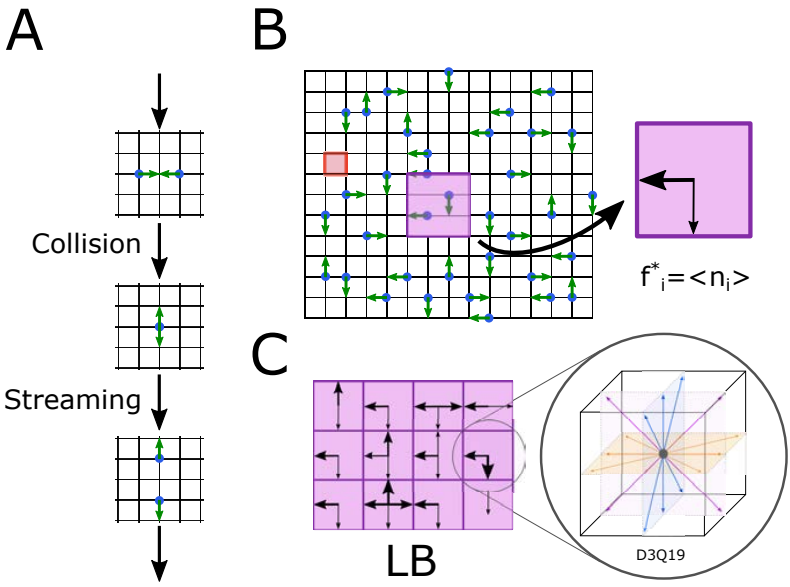


Figure 1.12: Basic scheme to understand the Lattice Boltzmann (LB) model. **A)** The model is based on the two-operation step: Collision-Streaming. During collision, momentum and mass are conserved. After the collision, particles stream towards different nodes using a discrete set of velocities. **B)** Landscape of these particles moving along the lattice (in red the unit cell). In older algorithms such as the Gas Automata, we can capture macroscopic values by averaging the occupation number in each direction over the neighbours (purple square). However, in LB these averages are directly the particles, which are equal to our f_i^*. **C)** The actual lattice is something closer to this lattice, where we use floats instead of boolean properties. Depending on how the lattice is constructed, distributions will have different number of directions. The overall picture here is in 2D, but each big square can have more directions available as shown in the inset which is commonly used in 3D scenarios. The available directions are written as $DnQm$, where n are the number of dimensions and m the number of available speed modes. *Image authorship. Own image.*

each collision we will be closer to this equilibrium, and that is why the equilibrium distribution for the distribution function appears here as f_i^{*0}. The decay into equilibrium will be modulated by a characteristic dimensionless relaxation time, τ^*. Finally, * indications refer to a dimensionless form of each parameter.

Although this scheme seems quite simple, under correct lattice discretization, one can derive the Navier-Stokes from the Lattice-Boltzmann equation[22] and hence, this is an equivalent to model fluids. The model presented in Eq. 1.3.7 is the simplest simplification of the LB model, but other complex expressions for the collision operator exist as shown widely in the literature[97]. Nonetheless, this was the model I followed during my thesis.

1.3.4 Liquid-Liquid interfaces in Lattice-Boltzmann: The Cahn-Hilliard equation

As previously remarked, the Boltzmann equation is developed considering we have N equal particles, and hence one chemical specie. However, if we want to model a fluid composed of s species we can have s distributions that evolve independently during the streaming, but interact in the collision step. Nontheless, in the case of a *binary mixture*, a mixture composed of two chemical species, we can also introduce the mixture using a Lattice-Boltzmann and a finite difference algorithm[98, 99], where the Lattice-Boltzmann develops the flow and the finite differences define the species present in the fluid.

The method chosen for the finite difference algorithm is based on the *Cahn-Hilliard equation*, an equation named after John W. Cahn and John E. Hilliard, who published a paper in 1958[100] with an interesting idea: the phase separation of a fluid composed by two miscible liquids, A and B, into two phases can be described by using a continuity equation such in Navier-Stokes[23]. The only key ingredients to develop the Cahn-Hilliard equation are the use of the spinodal decomposition of a binary system and the use of the free energy to define a system[24]. The development of this equation is rather simple, as one can see in Appendix B.2.3, and results in the following equation:

$$\frac{\partial c}{\partial t} = \nabla \cdot \left(M(c) \nabla \left(\mathcal{A}c + \mathcal{B}c^3 + \kappa \nabla^2 c \right) \right) - \nabla \cdot (\boldsymbol{u} c) \,, \tag{1.3.8}$$

where c is the difference of concentration between the relative concentration of both liquids, M is a parameter known as mobility, $\mathcal{A}$, $\mathcal{B}$, and κ are different real constants that describe the kind of mixture and $\boldsymbol{u}$ is the speed flow field calculated using the Lattice-Boltzmann algorithm. If $\mathcal{B} > 0$ and $\mathcal{A} < 0$ the model will describe the mixing of two immiscible liquids. In this case, κ will introduce

[22] See Appendix B.3.3 to find this derivation.
[23] See Appendix A.1.3 for a derivation of a continuity equation
[24] See Appendix B.1 and B.2 for a brief explanation of these concepts

an energy cost for interface creation. But if $\mathcal{B} = 0$ and $\mathcal{A} > 0$, the system will not be able to unmix and a homogenous phase will be seen[25].

As I will show in chapter 6, the modification of the previous equation will be very important to the development of the model I propose for my self-propelled particles interacting with liquid-liquid interfaces.

1.3.5 Walls in Lattice-Boltzmann

Once the fluid itself is defined, we also need to define how this fluid will interact with solid boundaries since particles are solid. To add solid nodes in LB we need to consider that some cells of the lattice are not accessible. The method chosen for this porpoise is known as *bounce-back on links* (BBL), and it was first proposed by Ladd [101].

In this method, when there is a solid-liquid interface between node $\boldsymbol{r}$ and node $\boldsymbol{r} + \boldsymbol{c}_i$, we consider as if at both sides we would have liquid, but the distributions are reflected during the collision step (See Fig. 1.13). If we denote with i' the opposite sense to direction i ($\boldsymbol{c}_i = -\boldsymbol{c}_{i'}$), then we have:

$$\begin{aligned} f_i(\boldsymbol{r} + \boldsymbol{c}_i, t+1) &= f_{i'}(\boldsymbol{r} + \boldsymbol{c}_i, t^+) + 6t^i \rho \boldsymbol{u}_b \cdot \boldsymbol{c}_i, \\ f_{i'}(\boldsymbol{r}, t+1) &= f_i(\boldsymbol{r}, t^+) - 6t^i \rho \boldsymbol{u}_b \cdot \boldsymbol{c}_i, \end{aligned} \tag{1.3.9}$$

where ρ is the density of the fluid, $\boldsymbol{u}_b$ is the velocity of the distribution node that acts at the boundary of the solid-liquid interface and t^i are some prefactors that depend on $\boldsymbol{c}_i$. This last correction is only necessary if the solid nodes are moving[98]. In the case of the walls this is not important but will be when dealing with particles since they move.

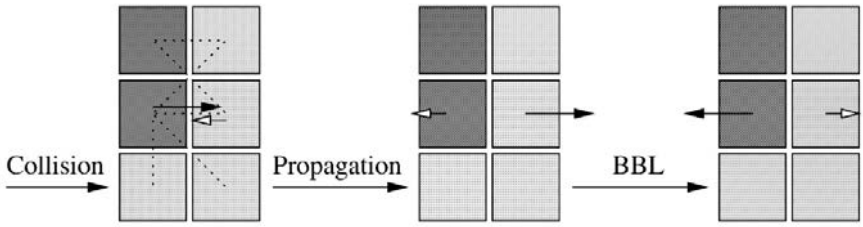


Figure 1.13: Bounce-back on links (BBL) method. In our model we can find a solid (dark nodes) - liquid (gray nodes) interface. The BBL method reflect the lattice-Boltzmann distributions on each side to model the solid wall. Dotted lines represent some of the links between solid and liquid nodes present at the interface. *Image authorship. Taken from ref. [98]. No modifications were performed.*

[25] In this case, *kappa* can be different from zero because there can be fluctuations and local variations, which can have an energetic cost.

1.4 | Modeling the particles

Probably, one of the most anguish and terrifying moments I have lived while working in the lab during this thesis has been when particles do not move, because without motion, no experimental thesis on this field can be done. To avoid these situations, and to improve better the capabilities of these systems, we need to know the propulsion mechanism of these particles.

As with the environment, there are several ways to model a particle, depending on whether we are working with a dry or a wet active system. In dry active systems the particle is usually modelled as a point mass particle that feels the medium by a drag and a noise term in its equations of motion, as in the Active Brownian Particle (ABP) model. However, in wet systems we can treat the particles much more realistic, with a much more improved interaction with the medium.

In this section we will see how we can model particles from different perspectives. First, I will present inactive particles by introducing their diffusion theory. Then, I will explain how we can model a dry model, following the well-known Active Brownian Particle (ABP) model. Afterwards, I will introduce different mechanisms of propulsion already known and I will show how we can introduce self-diffusophoretic particles in a wet system such as in the Lattice-Boltzmann framework.

1.4.1 Inactive particles

Almost two hundred years ago, the biologist and botanist Robert Brown observed under the microscope that tiny particles being ejected from some pollen grains of *Clarkia pulchella* were moving randomly[102]. At that moment he thought that these particles could be alive, but soon afterwards he could confirm they were not, since he observed the same motion in inclusions in quartz of millions of years old. After that, he could not explain what was happening, but his discoveries were left there, and were known as *Brownian motion*. It was not after around 80 years later that Einstein, in his marvellous year could give a theoretical explanation to this phenomenon in 1905[103]. A couple of years later, in 1909, the experimentalist Jean Perrin[26] published his quantitative measurements after reading Einstein's discoveries in his book "Atoms"[104], confirming Einstein's theory and obtaining

[26] He may be not as popular as Einstein, but he earned a Nobel Prize in Physics in 1926 because his work on the discontinuous structure of matter, with special attention to his discovery of sedimentation equilibrium.

an accurate value for the Avogadro's number. Today these discoveries could not be so remarkable, but we must situate ourselves in that epoch. The kinetic theory of gases was recently developed, and the atomistic model was coming back from the Ancient times, but nobody knew anything about the size of these molecules or atoms. Only ratios of number of atoms were known[27]. Therefore, it may be possible that the size of molecules and atoms would not be relevant and may be as a mathematical infinitesimal[28]. With the discoveries in the diffusion and Brownian motion, there was an estimation of the sizes of these molecules and atoms, and the kinetic theory of heat got a remarkable proof of validity. That is why, this theory was of real importance at that time. Molecules and atoms became definitely accepted.

So, why do those tiny particles move randomly? The answer for that motion is the fluid surrounding those particles. Although when we deal with fluids we are used to work with the Navier-Stokes equation, and thus we work with the continuum, we must remember that this is a model where we hided the molecules of the fluid. Not introducing the molecules into the equation does not mean that they have stop existing. Of course, they do and in fact, they are the answer to the random motion of what Brown discovered. Any object immersed into the fluid is feeling the tiny collisions of these molecules. They are just as random weak sparks that hit anybody. Thus, if the body is big, these interactions cannot be observed, because they will not be able to move the object noticeable. But as soon as you reduce your volume, the fraction of surface over volume increases, and the object perceives much more these interactions.

To start dealing this problem[29] we can start considering the grains of pollen as enormous molecules of gas that are in thermal equilibrium with the water that surrounds them. Given these grains, we can take one and just look at its motion over one of its axes (e.g., the x direction). The grain, of mass m, is moving because of numerous hits coming from the surrounding molecules. Therefore, we can introduce a random force F from which we do not know anything a priori. Since the grain is inside a fluid, it will feel a resistance to move. We can model this friction with the drag we obtained in Eq. (1.4.2). Consequently, if one applies Newton's second law:

$$m\ddot{x} = -\gamma\dot{x} + F, \tag{1.4.1}$$

[27]For instance, the kinetic theory for the ideal gas law, $PV = nRT$, was shown in 1856-1857, but this equation only records for n, which is the number of molecules divided per Avogadro constant, but it has no information about their size.

[28]In that time, there was a philosophical current that considered energy as the fundamental physics reality. For people who believed in this philosophy, atoms and molecules were just mathematical fictions. As an example, Mach, the scientist under the dimensionless Mach number, was one of the believers of that current.

[29]Einstein's demonstration is based on osmotic pressure and kinetic theory of gases. The procedure I will show here is not his approach but Langevin's approach, who obtained the same results a couple of years later but in a much simpler way[105].

where $\dot{x}$ indicates a time derivative (velocity), $\ddot{x}$ indicates a second time derivative (acceleration) and γ is the drag that the particle feels from the surrounding medium, which for a sphere is[30]:

$$\gamma = 6\pi\eta R, \tag{1.4.2}$$

where η is the viscosity of the medium and R is the radius of the particle. Now, we can take Eq. 1.4.1 and multiply all the equation per x, and later substitute the acceleration introducing: $\frac{d(x\dot{x})}{dt} = x\ddot{x} + \dot{x}^2$. Thus, Eq. 1.4.1 reads as:

$$m\left(\frac{d\left(x\dot{x}\right)}{dt} - \dot{x}^2\right) = -\gamma\dot{x}x + Fx. \tag{1.4.3}$$

To clear the previous equation, we need to do several assumptions:

- The first one is about the F term. This random force was introduced to give idea of the molecules hitting our grain of pollen. But if this process is randomly performed, the number of hits coming from both sides should be equal on average, since there is no a preferred direction. Therefore, in average this term must be zero.

- The second assumption is about thermal equilibrium. Boltzmann showed how two systems interacting reach the same temperature after enough time. When both are at the same temperature, we say they are at *thermal equilibrium*. Furthermore, in a system in thermal equilibrium, all its energy is shared equally among all the degrees of freedom of the system, with a kinetic energy of $1/2k_BT$ per direction of motion, where k_B is the Boltzmann constant and T is the temperature. Here, we suppose it has passed enough time so the grain is in thermal equilibrium and we will consider it as one more "molecule" of the whole system. Therefore, in average, the kinetic value $m\dot{x}^2$ will be $1/2k_BT$.

Taking these considerations, averaging the previous equation, and making the following change of variable $y = x\dot{x}$, we can obtain:

$$m < \dot{y} >= \frac{K_BT}{2} - 6\pi\eta R < y >, \tag{1.4.4}$$

from which, by integrating and considering that the grain was at $x = 0$ at $t = 0$, we can have:

$$2 < x\dot{x} >= \frac{d < x^2 >}{dt} = \frac{k_BT}{6\pi\eta R}\left(1 - e^{-t/t_0}\right), \tag{1.4.5}$$

where $t_0 = \frac{m}{6\pi\eta R}$ is a characteristic time. We can integrate this expression once again:

$$< x^2 >= \frac{k_BT}{6\pi\eta R}\left(t + t_0\left(e^{-t/t_0} - 1\right)\right), \tag{1.4.6}$$

[30] See section A.1.2 for more details on the calculus of this drag.

which for times much longer than this characteristic time gives us:

$$< x^2 >= \frac{k_B T}{6\pi\eta R} t. \tag{1.4.7}$$

This is a characteristic relation between space and time for particles that move randomly in a fluid as the grain of pollen Brown saw. The surprising result of this equation is that the length that these grains travel is not $x \propto t$, but $x \propto \sqrt{t}$. If we pay attention to the units, we can see that this relation gives us the area that the grain of pollen explores with time. But in fact, since the hits on the grain are random, on average there is no preferred direction and hence $< x >$ must be zero. This is the typical behaviour of particles under Brownian motion and we simply say that these particles *diffuse*. The relation we obtained is quite simple, but showed the behaviour Brown saw, including the dependence on temperature, viscosity, and size it was already known. Furthermore, when Perrin did the experiments, he could see that the relation fitted really well, and hence the molecular kinetic theory had to exist and with them, the molecules and atoms.

Usually, to the prefactor of this equation $\frac{k_B T}{6\pi\eta R}$ we call it as translational diffusion constant D_t since defines how particles diffuse around one degree of freedom for translation. Given that the pollen could have moved in two more directions, the above equation is usually known as $< r^2 >= dD_t t$, where d is the number of dimensions in your system. The above equation can also be computed for a rotational degree of freedom. In this case, the result also give us an expression of this kind, but with a prefactor that we call rotational diffusion constant D_R, which is $D_R = \frac{k_B T}{8\pi\eta R^3}$. In the case of the rotational diffusion constant, this constant has units of frequency. Sometimes, D_R is shown as the inverse, since it defines a characteristic time τ_R that we call it as rotational diffusion time:

$$\tau_r = \frac{8\pi\eta R^3}{k_B T}. \tag{1.4.8}$$

Finally, a couple of remarks about Eq. 1.4.7:

- To obtain this expression I supposed I was working at times much longer than the characteristic time, but what is the scale of this characteristic time? Let us consider the particles I am using in my thesis. They can be of 2 sizes: $R = 1$ µm and $R = 2.5$ µm. Since they are spheres made of SiO_2, which has a density of 2650 kg/m^3, we know that the mass per particle will be $1.11 \cdot 10^{-14}$ kg and $1.73 \cdot 10^{-13}$ kg respectively. It is true that half of the particles are covered by Pt, but this is just a thin layer of 10 nm, and hence we can neglect it. Experiments are done in water at room temperature and therefore, $T = 298$ K and $\eta = 8.9 \cdot 10^{-4}$ Pa·s. These give us a $t_{0,R=1\mu m} = 0.7$µs and $t_{0,R=2.5\mu m} = 4.1$µs. At a maximum, in my experiments I see an image every 20 ms. Hence, at any moment I will see that my inactive particles fulfil this relation.

- The expressions given for both D_t and D_R are also called as *Einstein-Stokes relation*. The Einstein-Stokes relation is a case of the relation Einstein – Smoluchowski, which relates fluctuations with dissipations. When the particle diffuses, it losses energy into the liquid because of the friction. The friction is the term given in the denominator by the Stokes relation. But this energy that is provided to the liquid will increase the temperature of the liquid, which will increase the speed of the molecules of the liquid, and hence molecules will hit stronger. Therefore, fluctuations given by the numerator will be larger and the cycle will continue.
- For the particle sizes I have, at room temperature and in water, the above relations give me the following numbers:

R (μm)	D_t ($\mu m^2/s$)	D_R (Hz)	τ_R (s)
1.0	0.245	0.184	5.435
2.5	0.098	0.118	84.926

Table 1.2: Values for D_t, D_R and τ_R for the size of particles I used in my experiments considering working in water at 25 C.

1.4.2 Adding activity in a dry active matter system

Without any knowledge on the mechanism of propulsion, and without considering all the hydrodynamic interactions, one could still have a model to mimic the motion of active particles. The most famous model for dry active matter is the *Active Brownian Particle* (ABP) model. The ABP is the result of applying the Langevin equation shown in Eq. 1.4.1 under overdamp regime and with a constant force that is proportional to the constant speed of the particle given by its activity. As shown in the previous section, the Langevin equation gives a relation between the space explored by a particle and the time it took it to explore this space. In the case of the ABP, this relation is also true but modified. This modification is useful since, even we are not considering the hydrodynamics, it still can show us a way to proof that our active particle has activity.

Therefore, I will start my explanation from the Langevin equation to get the classical ABP equations and after, I will follow with the fitting analysis required to verify this relation.

ABP model

We will suppose that our active particles move at a constant speed v_p given by their activity. Thus, we can start our derivation using Eq. 1.4.1, but adding a new constant term which gives the constant speed produced by the activity of the particle. Moreover, from the assumptions I did to obtain Eq. 1.4.7, we can directly define the force as a Gaussian distribution $\xi(t)$ with the following properties:

- $< \xi(t) >= 0$.
- $< \xi_i(t)\xi_j(t') >= 2\gamma^2\lambda\delta_{ij}\delta(t-t')$, where λ is D_t if the noise is for a translational degree of freedom or D_R if it is for a rotational one.

Adding the new term into the equation we have:

$$m\ddot{\boldsymbol{r}} = -\gamma\dot{\boldsymbol{r}} + \sqrt{2D_t}\gamma\boldsymbol{\xi(t)} + \gamma\boldsymbol{v_p}. \tag{1.4.9}$$

But as I said, our particle will move mainly due to the speed given by the activity and will have a constant speed. Also, consider that we are working at low Re. Thus, we can say that the inertial term is zero and now:

$$\dot{\boldsymbol{r}} = \sqrt{2D_t}\boldsymbol{\xi(t)} + \boldsymbol{v_p}. \tag{1.4.10}$$

If the particle is also diffusing in its rotational axis, we can also have a similar equation but without any speed related to the activity, since our particles only translate actively. For a 2D system this will read as:

$$\left.\begin{aligned} \dot{x}(t) &= \sqrt{2D_t}\xi(t)_x + v_p cos(\theta(t)), \\ \dot{y}(t) &= \sqrt{2D_t}\xi(t)_y + v_p sin(\theta(t)), \\ \dot{\theta}(t) &= \sqrt{2D_R}\xi(t)_\theta. \end{aligned}\right\} \tag{1.4.11}$$

Which is the classical ABP model. As with the pure Langevin equation, we also calculate the average of the square of the distance, which we also call as *mean square displacement* or MSD. The solution for the translational part in 2D reads as:

$$\text{MSD}(t)_{active} = 4D_t t + 2v_p^2\tau_r^2\left(\frac{t}{\tau_r} + e^{-\frac{t}{\tau_r}} - 1\right), \tag{1.4.12}$$

Fitting the MSD

The MSD can be understood as the mean area covered by a particle after a given time. In the MSD it is important to have clear that t is the *time lag*, due to MSD construction, because when we build the MSD, we are averaging the square of the subtraction of particles position at different times. Time lag refers to these differences in time rather than an absolute time. If the biggest time lag you can build is shorter enough compared with the rotational diffusion time τ_R of the particle, MSD can be fitted to a parabolic equation[37], where the speed can be extracted from the quadratic term following:

$$\text{MSD}(t)_{active} = 4D_t t + v_p^2 t^2. \tag{1.4.13}$$

If on the other side, we can get much more longer time lags compared with the τ_R, the approximation will read[37]:

$$\text{MSD}(t)_{active} = 4\left(D_t + \frac{v_p^2\tau_R}{2}\right)t = 4D_{eh}t, \tag{1.4.14}$$

where typically a new diffusive term is obtained, the enhanced diffusion D_{eh}, since it incorporates the effect of the activity over the normal diffusion of an inactive particle. Thus, by analysing the MSD of our particles and fitting it to a parabola or a lineal fit, we can obtain the D_t, the v_p or an effective D_{eh} of our particles, and of course, we will know if our particles are really active.

But to have these approximations we need to play with the ratio of the rotational diffusion time compared with the maximum time lag. The rotational diffusion time depends cubically on the size of the particle as shown in section 1.4.1, and thus, the size of the particle will mainly limit our maximum time lag. The smaller this maximum, the less time a particle must be recorded. But at the same time, if time is reduced, we will need more points per second to do good statistics. Furthermore, because of how MSD is constructed, only the first points have enough statistics to rely on. Typically, only the first 10% of the points in the MSD are used for fitting it. Thus, the number of points we have per second will be a very important parameter to play with. Although typically theoreticians suppose that this is just an order of magnitude difference, experimentally sometimes it can be difficult to reach. Moreover, some scientists may not understand totally how to apply this approximation. Given the problems I saw in the literature and given that I neither knew exactly which parameters to use in the beginning of this thesis, at half of the trajectory of my thesis I decided to tackle this problem to give a better approach. In chapter 3 I develop in more detail this problem, especially for the v_p and τ_r calculation.

1.4.3 Propelling particles: Methods of propulsion

The ABP model is a good method to have a minimum information regarding active particles, but if we aim to know more about the mechanism of propulsion or we need to model them in a much more realistic manner, we will need to know how these particles move. In section 1.1.2 I briefly showed some examples of particles moving with supposedly different mechanisms, but few comments were given about what was the true mechanism behind these particles. For many of those particles, the main mechanisms for particle propulsion are related with *phoresis*. The term phoresis comes from the Greek and, in fact, it was originally related to the biology field to refer to the behaviour of some animals that attach over other animals just to be displaced by them, without doing anything by themselves.

For inactive particles we could imagine that they are "attached" to the flow when they flow within to move from one side to another, but this can be more complex. A flow, in fact, can be a gradient of some field. For instance, if we have a gradient of solutes, we can have *diffusophoresis*. For example, B. Abécassis showed this effect with inactive SiO_2 particles[106]. If instead we have a gradient of an electric field, we can have *electrophoresis*. Of course, active particles are active not because there is a global gradient, but because they are producing by themselves these gradients. Hence, we have particles that move due to a self-

diffusophoretic mechanism and particles that move due to a self-electrophoresis mechanism, among others.

In my case, the motion of the Pt-SiO_2 particles I used during this thesis has used to be explained by self-diffusophoretic models[107, 108, 109][31]. However, Aidan Brown and Wilson Poon also showed that the mechanism cannot be totally due to self-diffusophoresis but might include self-electrophoresis[110] due the effects of salts on the motion of the particles. In the next pages I will explain the basics of both mechanisms.

Self-Diffusophoresis

Self-diffusophoretic active particles are those that generate a non-uniform solute concentration field around them providing a mechanism for motion. To generate this field, particles speed up chemical reactions asymmetrically by using some catalysers that are on some parts of their surfaces. Typical catalysers can be a metal like platinum or an enzyme like urease. For example, these last catalysers give the following reactions:

$$\begin{aligned} H_2O_2 &\xrightarrow{\text{Pt}} \frac{1}{2}O_2 + H_2O, \\ Urea &\xrightarrow{\text{Urease}} NH_4^+ + HCO_3^-. \end{aligned} \tag{1.4.15}$$

While the former reaction is one of the possible mechanisms to explain the motion of Pt-SiO_2 particles, the latter is typically used for drug delivery systems. Moreover, these examples are good to provide us an important fact. In some diffusiophoretic particles, the products of the reactions are ionic species, while in others are neutral. Therefore, we can distinguish between neutral and ionic self-diffusophoresis. However, in this section I will mainly focus in the neutral self-diffusophoresis given the fact that is the traditional mechanism to explain the motion of Pt-SiO_2 particles.

To develop a neutral self-diffusophoretic model, we assume that the catalytic reaction only occurs at one region of the particle surface (e.g., like a Janus particle). Since the production is on the surface, on top of the surface there will be a c_0 concentration of products. However, as we move far from the surface, this concentration will dilute in the bulk, until a c_∞ is achieved. Hence, in between both values there is a gradient of products that spans for a distance that we will refer as λ. We can assume that products inside this region will interact with the surface by a potential $U(r)$. But the distance λ will be only around a few molecules' length, which will be very small compared with the radius of the particle. Therefore, we can zoom in into the surface and consider it as a flat surface rather than spherical. From this perspective, we will consider that the y-direction is the same as increasing r, while the x-direction would be the θ direction. Thus, we will consider the

[31] Some studies used polystyrene (PS) instead of SiO_2, but both materials are insulators and should act similar.

potential $U(r)$ as $U(y)$. Consequently, we can assume that the concentration of solutes c around the particle will be $c = c(U(y))$. In Fig. 1.14 we display the schematics of the process.

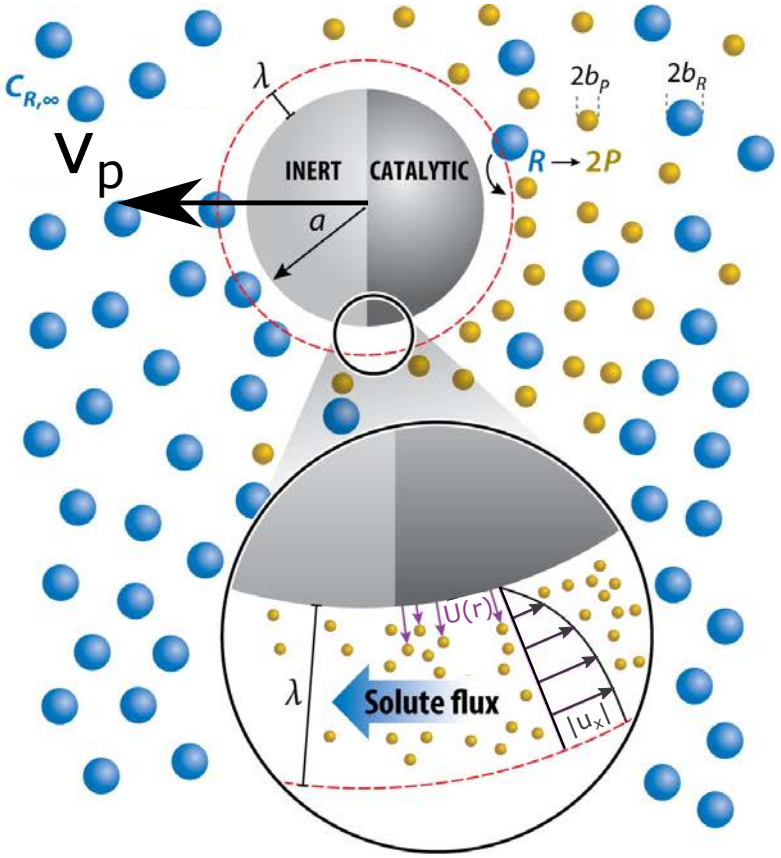


Figure 1.14: Scheme to understand the self-diffusophoretic model. A Janus particle move from the catalytic side towards the inactive side. In the medium there are R molecules, which reacts in the catalytic surface to generate P molecules. Next to the particle surface, these P molecules interact with the particle surface by a $U(r)$ potential that spans for a lenght λ. Because the anisotropy of the problem, there is a gradient of these P molecules along the particles that generates a velocity field u_s around the particle. The motion of the particle, with speed v_p, will be opposite to this velocity. *Image authorship. Adapted from Ref. [111].*

Since there is catalysis only in one side of the particle, we will have a gradient of these solutes also along the surface of the particle. In this view, along the x-direction. But the gradient in x-direction will take much larger distances than the one in the y-direction. Hence, we can consider that the dependence on y-direction will equilibrate very fast in comparison with the x-direction, and therefore we

can also assume that in the y-direction it will follow a Boltzmann distribution, e.g. $c(r) \sim e^{-\beta U(y)}$, where $\beta = k_b T$ being k_b the Boltzmann's constant and T the temperature. But we also have a gradient on the x-direction, and hence $c = c(x, y)$. If we consider two points, x_1 on the active surface, and x_2 on the inactive surface, we could think that the concentration at which it will equilibrate next the surface will be different in point x_1 and x_2. Thus, we can consider that $c_\infty = c_\infty(x)$ and that locally, the concentration can follow:

$$c(x, y) \sim c_\infty(x) e^{-\beta U(y)}. \tag{1.4.16}$$

Because the equilibration in the y-direction is fast, we do not expect to see a net flow in the y-direction, but there will be a counterbalance between the fluid pressure P and the force density due to the presence of the solute $-c\partial_y U(y)$. Hence, we can relate both terms by using Navier-Stokes, or simply, conservation of momentum. At the same time, due to the gradient of solutes in the x-direction we also expect to see a counterbalance. In this case though, since the equilibration is much slower, we expect to see flow. In principle, no force is acting since the only force is given by $U(y)$ and we do not have inertia because we work at low Re. Therefore, we have the following system:

$$\left.\begin{aligned} \frac{\partial P}{\partial y} &= -c\frac{\partial U}{\partial y}, \\ \frac{\partial P}{\partial x} &= \frac{\partial}{\partial y}\left(\eta \frac{\partial u_x}{\partial y}\right), \end{aligned}\right\} \tag{1.4.17}$$

where η is the viscosity of the fluid. To solve these partial derivatives we need to know a couple of boundary conditions. It is evident that if $U(y)$ is a short-range potential that applies only for λ region, at $y \to \infty$ must be $U(y) = 0$. Moreover, in the bulk we also expect to see everything uniform. Hence, there will not be velocity gradients and $\partial_y u_x = 0$ for $y \to \infty$. Finally, next to the surface we can apply non-slip boundary conditions. Therefore, $u_x(y = 0) = 0$. Introducing Eq. 1.4.16 into Eq. 1.4.17, when needed, and integrating as many times as required, we have:

$$\left.\begin{aligned} p(x, y) &= p_\infty + \frac{c_\infty(x)}{\beta}\left(e^{-\beta U(y)} - 1\right), \\ u_x &= -\frac{1}{\eta\beta}\frac{\partial c_\infty}{\partial x}\int_0^y dY \int_Y^\infty dy' \left(e^{-\beta U(y')} - 1\right). \end{aligned}\right\} \tag{1.4.18}$$

As we can see, the result of the speed depends on an integrand we could not resolve. In fact, we can introduce a couple of terms and simplify to give a closer form. Relative to the new terms, we can define:

- The distance in which the potential $U(y)$ is acting over the solutes, λ, can be expressed as a weighted length with the surface-interaction potential $U(y)$ as:

$$\lambda = \frac{\int_0^\infty y dy \left(e^{-\beta U(y)} - 1\right)}{\int_0^\infty dy \left(e^{-\beta U(y)} - 1\right)}. \tag{1.4.19}$$

- The normalization of the previous integral as κ, which can be seen as the excess surface concentration normalized by the concentration at infinite or an adsorption length. A simple way to understand κ is by looking at Fig. 1.14. If we take all the excess of solute (γ)next to the surface with respect the one found in the bulk, and we divide it by the solute at the bulk, this would be the module of κ. It would represent the width of a layer of the bulk containing as many solute molecules per unit area as γ.

$$\kappa = \frac{1}{c_\infty}\int_0^\infty dy\,(c(x,y) - c_\infty(x)) = \int_0^\infty dy\left(e^{-\beta U(y)} - 1\right). \quad (1.4.20)$$

About the simplification, we see in these integrals that the limits are between 0 and infinite, but in Eq. 1.4.18 the first integral does not end in infinite. Even though, since this integrand is over the y-direction, we know that from some distance (λ), the potential is not going to act and hence, the speed is going to be the same at the infinite than at the end of λ. Therefore, we can say $u_s = u_x(y \to \infty)$. Considering $g(Y) = \int_0^y dY \int_Y^\infty dy' \left(e^{-\beta U(y')} - 1\right)$ and integrating by parts, we have:

$$v_{diffuso} = \frac{\lambda\kappa}{\beta\eta}\nabla c_\infty \quad (1.4.21)$$

where we have redefined u_s as $v_{diffuso}$. The reason to redefine it is because essentially, to have this speed we need some interaction between the solute and the surface, but the distance where this potential act is very small in comparison with the radii of the particle. Therefore, from a length-scale of the particle, this is just as if the particle would have a slip condition over its surface with this velocity. Thus, this is the reason why a self-diffusophoretic particle can move. Over this derivation, there can be other assumptions, including the extrapolation for ionic self-diffusophoresis as seen in ref. [108].

Finally, two last facts about the expression 1.4.21. The first fact is about a redefinition of 1.4.21. Although in the literature is possible to find a description of the parameters preceding the ∇c_∞, in the wet model I used during this thesis, the constants are redefined as a new constant, which I will refer as a diffusophoretic mobility μ. More details about this model will be presented in section 1.4.4. The second fact is about the ∇c_∞ itself. Experimentally, to see these particles moving one just needs to add H_2O_2 into the water solution with the particles. Thus, experimentally we do not work with ∇c_∞, but $[H_2O_2]$. For instance, in Fig. **??** we can see how the speed of these particles change depending on the $[H_2O_2]$. However, the term ∇c_∞ can be expressed as a function of $[H_2O_2]$. The details of the process are explained in ref. [108], but briefly, one assumes that the diffusion of the products follows Fick's first law. Thus, the number of molecules N that diffuses over time from the reactive surface of area a^2 will be $\partial_t N = -|D\nabla c_\infty|a^2$. But this term must equilibrate with the incoming reactive molecules. The units of the previous term are the inverse of time units. Hence, we can define for the incoming

as the inverse of a characteristic time τ. This time is connected via the speed of the reaction to the $[H_2O_2]$ and therefore, the speed can be written as a linear function of $[H_2O_2]$. Depending how one defines the chemical reaction, and which is the limiting step, there can be several equations for the speed.

Self-Electrophoresis

Self-electrophoretic active particles are those that generate non-uniform surface charges over their surface. Consequently, around these particles there will be a cloud of ions that counteract the surface charge and hence, an electric E field is generated. Notice that this mechanism is different from ionic self-diffusophoresis because the key point in these particles is to generate this asymmetrical surface charge. The typical particles that produce this mechanism are those related with reduction-oxidation (redox) reactions, where a metal gives electrons to another. For instance, it is famous the bimetallic Au-Pt particles:

$$\left.\begin{array}{ll} H_2O_2 & \xrightarrow{\text{Pt}} O_2 + 2H^+ + 2e^-, \\ H_2O_2 + 2H^+ + 2e^- & \xrightarrow{\text{Au}} 2H_2O. \end{array}\right\} \tag{1.4.22}$$

The layer of ions around the particles will extend for a typical length called as the *Debye length*. Inside this region, one can calculate the electric potential given the charge by the Poisson's equation. From this potential we can derive the electric force, which considers the E field, and hence counterbalance the fluid flow as I showed with the diffusophoretic mechanism. Here, the E would replace the ∇c_∞. Hence, the following result for the speed can be given:

$$v_{electro} = \frac{\epsilon\epsilon_0\zeta}{\eta}E. \tag{1.4.23}$$

where ϵ and ϵ_0 are the permeabilities for the void and the material, η is the viscosity of the fluid and ζ is the zeta potential, or potential at the surface. These particles will move in the direction of $-E$, since the fluid slips in the direction of E.

Finally, a last remark about these particles. Since these particles depend on the cloud of ions also to move, they will be sensitive to the addition of salts into the medium, since they can alter the cloud of ions.

1.4.4 Adding activity in a wet active matter system

The previous fundamental findings are very useful to improve the model of our particles. For example, if we need to also model the fluid surrounding the particle, this knowledge let us write alternative methods. For example, a more complex model than the ABP model can be built with a Lattice-Boltzmann method. A. Scagliarini and I. Pagonabarraga introduced a model of self-propelled particles months before I started this thesis[112] using this knowledge and the results from

what I explained in section 1.3.4. In the following subsections I will explain this implementation, from inactive to active particles in Lattice-Boltzmann.

Adding inactive particles into Lattice-Boltzmann

The discretization of the LB method implies to use a lattice where nodes are cubes. Consequently, the particles will be objects made of cubes, being impossible to have a sphere. However, if we increase the radius of the particle, the surface of our sphere will be softer, and we will have a better particle model. Typically, particles radius are about 3-4 unit cells to decrease this problem. In principle, we could make them bigger, but we will have problems when parallelizing the simulation. When we parallelize the simulation, our simulation box is cut in different sub-boxes that run in different processors. But for each sub-box, we need at least a sub-box size equal to a particle diameter plus the length at which particles interactions will compute. Therefore, although increasing the particle size will improve the surface, it will decrease our parallelization process[32]. In my simulations I ended up working with particles of $R = 4.5$ lattice nodes.

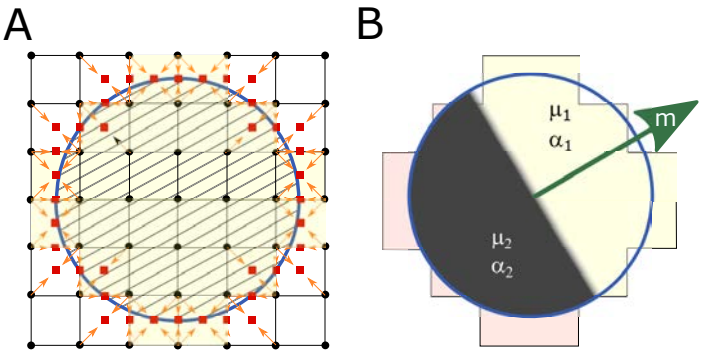


Figure 1.15: Particle implementation in LB method. A) LB discretization implies we only have lattice nodes (black spheres) on the vertices of squares. When a particle is set (blue line, black stripes, $R = 2.5$ nodes), the particle in this lattice looks as the yellow area. At the boundary nodes (red squares, half path between black solids), we calculate the Bounce-back on links (BBL) method, where we used different velocities (orange arrows) that cut the boundary. **B)** To add activity to the particle we divide them in two sides, each with a mobility constant μ which describes how the node interacts with the phoretic field and an activity rate α. The perpendicular vector to the surface that separates both sides determines the $\boldsymbol{m}$ orientation vector. For each time step, we add α quantity to the phoretic field in each boundary node. For example, each node in red will add a quantity α_2, while nodes in yellow will add a quantity α_1. *Image authorship. A) Taken from ref. [114]. Added colours. B) Own image.*

Internally, the implementation of the particles follows the solid nodes idea from

[32] See Ref. [113] for more details on the parallelization

section 1.3.5, applying Eq. 1.3.9 in the surface nodes (See Fig. 1.15A)). Here, $\boldsymbol{u_b}$ is:

$$\boldsymbol{u}_b = \boldsymbol{U} + \boldsymbol{\Omega} \times \left(\boldsymbol{r} + \frac{1}{2}\boldsymbol{c}_i - \boldsymbol{R} \right), \tag{1.4.24}$$

where $\boldsymbol{U}$ and $\boldsymbol{\Omega}$ are the velocity and angular velocity respectively of a particle with its center of mass at $\boldsymbol{R}$. This equation arises because although we say we have solid nodes, inside the particle we still have fluid nodes that do not exchange the distribution functions with the exterior. Since the size of the particles is usually small, these distributions will relax fast and will give the rigid-body motion behaviour[114]. Hence, the nodes of fluid in contact with the particle will have the same local velocity as the solid nodes in contact with them[33].

While performing these updates on the nodes, the total momentum of the particle and the fluid is conserved, although locally is exchanged. This local transfer allows us to calculate the forces and torques on each node. Summing over all the nodes of the particles we obtain the total force and torque applied to a particle[115], which we can sum to other forces we added to the system such as gravity or a lubrication force. The lubrication force solves a problem appearing when two particles are in contact. In this case, the fluid between the particles is squeezed out of the gap between the particles, generating high pressures in the gap. Moreover, two particles may share a same lattice node, causing a gain or loss of fluid from the interior of the particle which do not allow the system to reach a steady state[115].

Adding activity into particles

To add activity, A. Scagliarini and I. Pagonabarraga introduced a scalar phoretic field C such in section 1.3.4 with $\mathcal{B} = \kappa = 0$ and $\mathcal{A} > 0$. This field represents the product generated by the active particles such with the O_2 in the particles I use experimentally. The particle is then divided in two areas (see Fig. 1.15B)), each one with a couple of μ and α values. The constant μ represents the phoretic surface mobility, which in equation 1.4.21 would be the prefactor $\lambda\kappa/\beta\eta$ and hence we can introduce a slip velocity particle:

$$\boldsymbol{U}_s = \mu(\boldsymbol{x}_s)(\mathbf{1} - \hat{\boldsymbol{n}} \times \hat{\boldsymbol{n}}) \cdot \nabla C, \tag{1.4.25}$$

where $\boldsymbol{x}_s$ is a point on the surface of the particle, $\hat{\boldsymbol{n}}$ is the normal to the surface in $\boldsymbol{x}_s$ and $\mu(\boldsymbol{x}_s)$ is the phoretic mobility at point $\boldsymbol{x}_s$ which carries the molecular details of the solute-colloid interaction. When phoretic mobility is uniform, $\mu(\boldsymbol{x}_s) = \mu$, particles gain a net propulsion velocity $\boldsymbol{v_p} \sim -\mu\nabla C$, i.e. if $\mu_0 < 0$, we obtain a positive diffusiophoresis and they are "chemoattractive" colloids, while if $\mu_0 > 0$ we have a negative diffusiophoresis and "chemorepulsive" colloids.

Because our particles generate their own product, they also create ϕ product, an $\alpha(\boldsymbol{x}_s)$ quantity, at each step. Because we model our particles as Janus particles,

[33] These are called *Stick boundary conditions*.

only half of its surface, the active side, will generate this product, while the inactive will not produce. Moreover, in the active side the production is homogeneous and therefore we have:

$$\alpha(\boldsymbol{x}_s) = \begin{cases} \alpha & \text{if } \hat{\boldsymbol{m}} \cdot \hat{\boldsymbol{n}} \leq 0 \\ 0 & \text{if } \hat{\boldsymbol{m}} \cdot \hat{\boldsymbol{n}} > 0 \end{cases}, \tag{1.4.26}$$

where α is the constant production rate and $\hat{\boldsymbol{m}}$ is the particle characteristic unit vector (see the sketch in Fig.1.15B)). For an isolated free Janus particle with the previous activity and uniform mobility we expect a motion with constant speed $v_p = |\mu|\alpha/(4D)$ [116, 117, 118]. If we distinguish from an active and an inactive side, we can have μ_A and μ_I being the surface mobilities for the active side and the inactive side respectively.

Finally, to avoid accumulation of the order parameter over time, a proportional quantity to the equilibrium value of the order parameter is removed every time step per node in the lattice.

Interacting with liquid-liquid interfaces in Lattice-Boltzmann: Wetting

A last ingredient is needed when working with particles, liquid-liquid interfaces, and LB: the wetting behaviour. To add the wetting, we must introduce a binary mixture. Therefore, now equation 1.3.8 will have $\mathcal{A} < 0$, $\mathcal{B} > 0$ and $\kappa > 0$. But this free energy defines what happens in the bulk. Since wetting is happening on the surface of the particle, we will add an extra cost term f_s to this equation, which we will integrate over the surface of the particle. Hence the free energy will be:

$$F = F_{bulk}[\phi] + \int f_s(\phi_s) dS \tag{1.4.27}$$

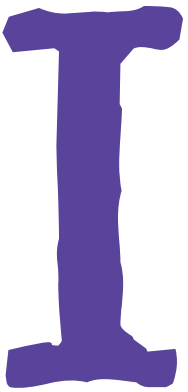
The simplest relation for f_s will be:

$$f_s(\phi_s) = \frac{C}{2}\phi_s^2 + H\phi_s \tag{1.4.28}$$

for simplicity, we can make $C = 0$. If $H = 0$, we will not add an extra cost to be at the interface, and particles will be at interface (90º contact angle). But if $H \neq 0$, then we will have a different contact angle following[98]:

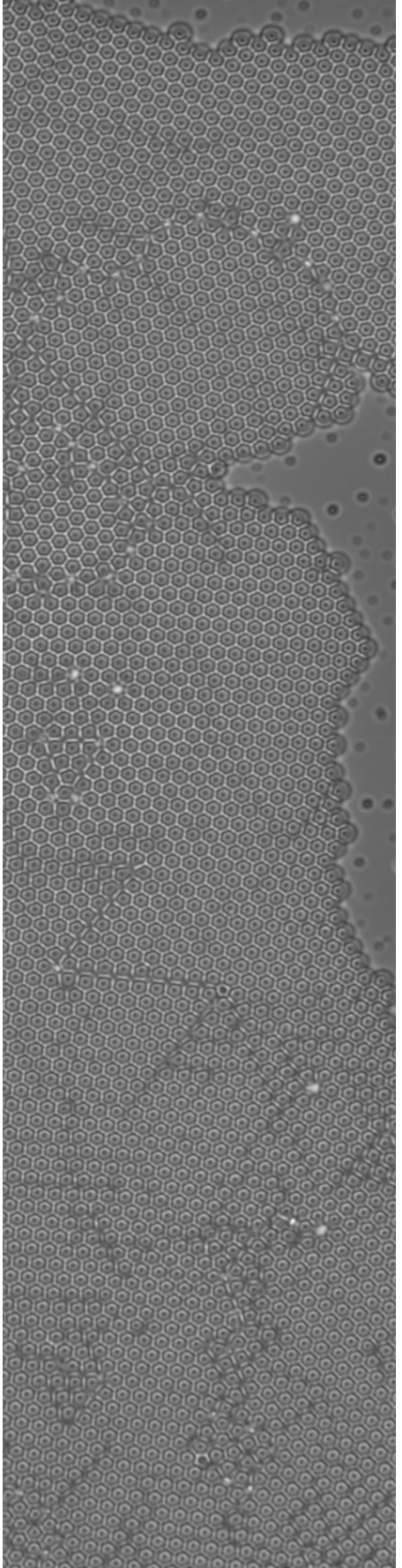
$$cos(\theta) = \frac{1}{2}\left[-(1-h)^{3/2} + (1+h)^{3/2}\right], \tag{1.4.29}$$

where $h = H\sqrt{1/(\kappa B)}$.



Methods and video analysis

The first part of this thesis is dedicated to all the side projects that paved the way for the development of the following parts of my thesis. The part starts with a chapter dedicated to the creation of experimental setups such as particles and microfluidic devices. Next, a second chapter is dedicated to video recording. In this second chapter I optimize different parameters for video recording to limit the digital size of these videos while being able to correctly analyse particle parameters such as their enhanced diffusion constant, translational diffusion constant, rotational diffusion time or speed. The chapter includes the equations needed to extract these parameters and to differentiate when particles are drifting. This chapter ends with a couple of tools to compress the videos. Finally, a last chapter is included to process the recorded videos. This chapter includes tips and tools to track particles using different algorithms. The chapter is especially dedicated to a tracking system based on machine learning algorithms.



2

Fabrication of particles and microfluidic setup

"A designer knows he has achieved perfection not when there is nothing left to add, but when there is nothing left to take away."

Antoine de Saint-Exupry

None of the experiments carried out in this thesis would have been possible without a proper setup, and of course, without particles. The particles I used for this thesis are very simple to create, and the steps to produce them are widely available in the literature. However, these particles are basic for the following experiments, and therefore in this chapter I present the steps to create them.

The chapter follows with the creation process of several setups developed in this thesis. The final setups I obtained are useful to work with these particles for long-time experiments of one hour. This fact is an incredible evolution of my working setups given the problems I had in the beginning of my thesis due to the bubbles generated by the particles after 10 minutes.

The present chapter is based on the setups I developed for the experiments carried out in chapters 5 and 7. I would like to especially thank Jaideep Katuri for sharing with me his knowledge to create Janus particles, which I think by the end of the thesis I improved. I would also like to especially thank Maria Guix for her help during the use of the Inkredible+ 3D printer and the design of the microfluidic designs for chapter 7. Finally, I would also like to thank the people from MicroFabSpace and Microscopy Characterization Facility, Unit 7 of ICTS "NANBIOSIS" from CIBER-BBN at IBEC, who carried the process of printing the microfluidic design into a mould that I could later use.

2.1 | Creating Janus particles

Pt-SiO_2 Janus particles are SiO_2 particles that we half cover with a thin layer of Pt. SiO_2 particles are easy to chemical synthetize them, but precisely because it is easy and because they have many utilities, you can also find them commercially available. In my case, a little bottle with 5 ml of an aqueous solution containing SiO_2 particles was sufficient for this thesis, and the cost of it was under 200€. We purchased microparticles from Sigma-Aldrich, and we bought particles of 2 and 5 μm diameter size.

To build the Janus structure, there are different methods that we can follow, but the easiest is to *sputter deposit* the platinum. Sputter deposition is a physical process where ions are bombarded onto a layer of a material to free atoms of this material. Subsequently, these atoms fly onto a substrate where they will deposit as a layer of this material. Here, the ions bombarded are He^+, the material is Pt and the substrate is the SiO_2 particles. Sputter deposition is completely made under vacuum by using Leica EM ACE600 sputter machine. During this process, particles must be static and dried, and hence we need to attach them to someplace. I attached the SiO_2 particles to 25x25 mm^2 squared glass cover slides (Deltalab). To attach particles to the slides we can also use different strategies, but the simplest is to *drop casting* the particles.

Drop casting is a technic where the solution containing the particles is deposited on top of a surface, and after waiting for liquid evaporation, particles get attached by electrostatic forces to the surface. But to attach, the surface must be cleaned and activated.

To clean the surface, which in this case is the glass slides, we sonicate them with acetone for two minutes, following two minutes more with isopropanol. Acetone will remove organics, if any present, while isopropanol will remove any leftover of acetone. Then, we dry them using a compressed N_2 or air gun.

To activate the surface, we do an O_2 plasma for 10 minutes. Plasma will attack the slides surface, breaking chemical bounds and leaving ions on the surface. These ions can make better hydrogen bounds when water is present, which in turn makes the slides more hydrophilic. If the slide is hydrophilic, water will flow easily on top of the slide, which is a critical step for us because we want particles to get attached, but also, we want to have monolayers of particles. If liquid flows better, we will be able to disperse better the particles and hence, to have more area

with monolayers. If particles cannot flow easily, there will be several layers of particles, one on top of each other. Hence, only particles on the outermost layer will receive the Pt during the sputter process, and therefore, only particles of this layer will be active particles. Particles on other layers will be inactive even the sputter process. Initially, I used to do two minutes of plasma, but I found that if you apply it for 10 minutes, water could flow easily, covering all the surface and obtaining better monolayers.

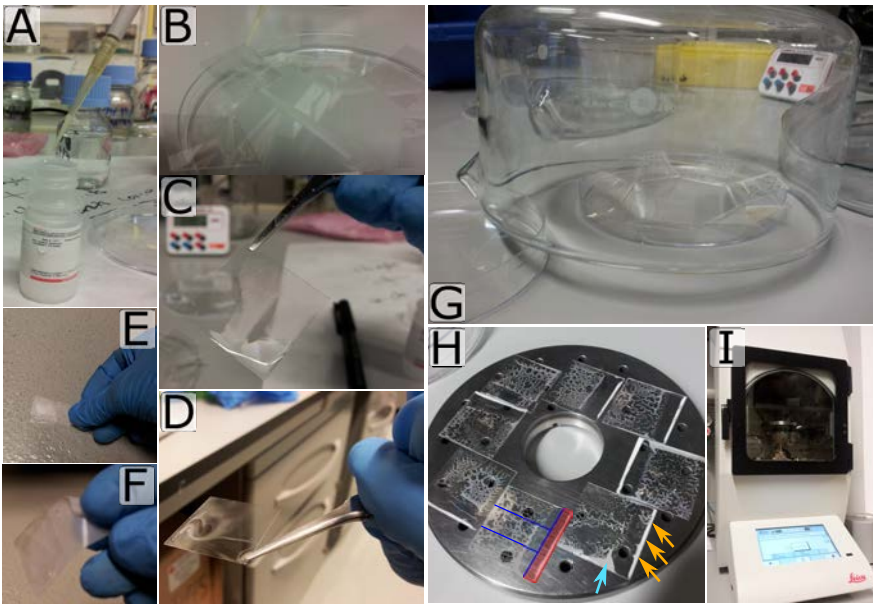


Figure 2.1: Janus creation. A) Take 20-25 µl of the commercial particles and place it on top of the slides as in **B)**. Next move the slide with a tweezer as in **C)**. Alternatively, you can also flip the slide as in **D)**. If during this process the slide falls to the floor as in **E)**, it is easy to clean it by pulling the slide with a sticky tape **(F)**. Let everything dry for an hour **(G)** and place them in the supporter for the sputter **(H)**. Make the Pt sputter **(I)** and storage closed and at room temperature. **Orange arrows in H)** indicate an area with multilayers. **Blue arrow in H)** indicate an area of a very dense monolayer. **Blue lines in H)** indicate where I usually cut the slide before doing an experiment. **Red zone** is always discarded because it contains many inactive particles.*Image authorship. Own images.*

After slides are cleaned and activated, we can pipette 20-25 µl of the commercial aqueous particle solution per slide (See Fig. 2.1 A) and B))[1]. To spread the liquid over all the slide, we can carefully move it with a tweezer softly in the air during 10s, as if you were opening and closing a lock (See Fig. 2.1 C) and D).[2].

[1] Pipette only on 3 slides and finish their spread process before pipetting 3 more. Otherwise, slides will dry and we will not be able to do the spread process.

[2] It is easy that the slide falls to the ground at this moment. If so, lots of tiny pieces will be attached

If the slide presents dried gaps after moving it, we can touch the liquid next to these gaps with a clean pipette to let liquid in. Alternatively, we can put downside the slide in the air. Because we have a few microliters on the slide and it is hydrophilic, liquid will not fall by gravity, but capillarity forces will maintain it.

Next, we can leave the slides with one edge supported on the case of a plastic petri dish. Let it dry completely for an hour covered to protect from dust (See Fig. 2.1 G). After one hour, slides are introduced in the sputter (See Fig. 2.1 H) and I) with a Pt target, and we start sputtering. The parameters used for the sputter machine are the following:

- Vacuum: 1 purge cycle, with a pre-process vacuum of $2 \cdot 10^{-5}$ mbar and a sputter vacuum of $5 \cdot 10^{-2}$ mbar.
- Geometry: Slides are supported on top of a stage inside the sputter machine (The metallic cylinder in Fig. 2.1 H). This stage is at a distance from the Pt, which is defined as *the working distance*, of 45.0 mm. To better adjust this distance, the software from the sputter allows to introduce the sample height, which we set to 0.2 mm. Stage can rotate, and we imposed 2 rotations. At any moment, the stage will not be tilted.
- Current: set to 35 mA.
- Thickness: 10 nm. Until a thickness of 10 nm[119], the thicker the layer, the faster particles will move. From 10 nm on, this effect is lost.

Finally, we store them closed at room temperature until we use them. At the moment of usage, we take one slide and do a short Ar plasma treatment. This process will clean the Pt surface, removing organic matter without oxidizing the Pt, and thus increasing the particle mobility. Then, we cut[3] the outer 3 mm of the edge that was supporting the slides for drying with a diamond pen (red area in Fig. 2.1 H)). This edge is easily to differentiate from the others because it has a visible white band (orange arrows in Fig. 2.1 H)). Within this area there can be multiple layers of particles, and if we do not remove it, we will have lots of inactive particles in the experiments. Other areas with the same white tone are also multi-layered, and we should remove them too. However dense but a bit translucent areas are perfect monolayer wide areas, do not cut them (blue arrows in Fig. 2.1 H)). After cutting this edge, we can cut the slide in 3 equal pieces (blue lines in Fig. 2.1 H) indicate where to cut the slide). Insert one piece in a 5 ml snap cap vial (VWR) and add 1 mL of miliQ water. Sonicate for 3 seconds to detach particles from the slide and place the dilution in an Eppendorf. You can play with the amount of slide and water you sonicate, but these numbers were useful for my experiments.

to the floor by capillary forces due to the liquid. The easiest way to clean this mess-up is by attaching sticky tape on the floor and pull it. Then, dry it with some paper. (See Fig. 2.1 E) and F)

[3] When you cut the slide it is possible to see a small aerosol against a backlight. SiO_2 inhalation can produce silicosis, a lung disease. Consider cutting the slide under a fume hood.

2.2 | Creating wells for particles

To record particles under a microscope, we need to put them on a surface or in a container. For example, the easiest procedure is to mix them with their fuel and put a droplet of this solution on top of a cover glass slip. However, setups are usually more complex than a droplet. First, because our particles generate oxygen as a product of the chemical reaction on the Pt surface. Tiny imperfections on the surface where particles move lead to oxygen nucleation, creating oxygen bubbles that create inner flows and gaps of liquid. Eventually, these bubbles burst, and all particles will suddenly move. Second, because the droplet will evaporate easily, which limits the time for recording. Moreover, if the droplet is reduced, phenomena occurring at the borders such as coffee rings[120] will interact with our experiments. Third, because a droplet is easily affected by current air flows, which can also add drift to the system. And forth, because if the surface is hydrophilic, the droplet will splash, and we will not be able to record anything. Hence, to sort these problems I started by using some simple microfluidic channels, but finally I decided to build my own wells where to introduce the particles solution.

In chapter 5 I used a microfluidic chip commercially available from IBIDI (μ-Slide VI $^{0.4}$ ibiTreat) which consists in closed channels of size 75.5 x 25.5 x 0.4 mm^3 opened at the borders (See Fig. 2.2 A)). These channels avoid problems of evaporation and external flows, but because they are closed, after 10 minutes they will be full of oxygen bubbles (See Fig. 2.2 B)). To overcome this problem, I decided to have an open system, where I could also have enough liquid to avoid evaporation and external flows too. The easiest approach was to build wells.

To build these wells, I decided to use Polydimethylsiloxane (PDMS) and a 3D printer (Form 2) because it is easy, cheap, and fast. PDMS (Sylgard 184, Dow Corning) is a silicone, and a polymer, consisting in two elements: a monomer and a crosslinker. Commercially, PDMS comes in two separated bottles, each containing the monomer and the crosslinker respectively in liquid form. After mixing both solutions[4] (See Fig. 2.2 C)), and heating it at 65ºC for 4-6h, the silicone gets crosslinked, which you can notice because it is solid. Hence, we can build as many wells as we want if we have a mould (See Fig. 2.2 D)). To build the mould we can use SketchUp software. This mould will be the inverted object of the wells, a pool with cylinders.

[4]The ratio between monomer and crosslinker defines the stiffness of the silicone after crosslinking. During all the thesis I used the standard 90% monomer, 10% crosslinker. Consider doing 1h of vacuum to the mixture before heating it to remove tiny air bubbles.

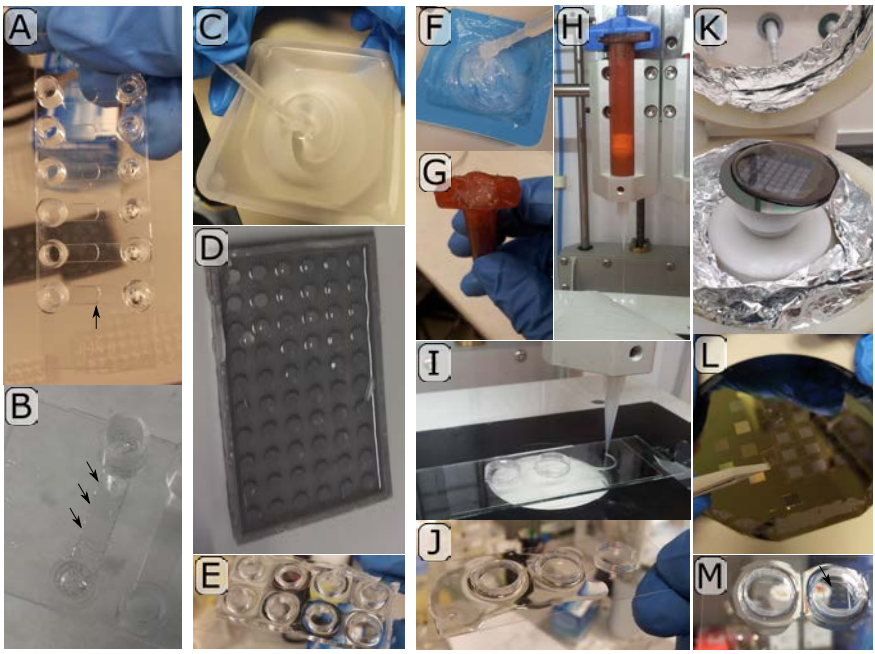


Figure 2.2: Setups used for experiments. A) Ibidi microfluidic setup used for the experiments carried out in chapter 5 **Arrow:** front of the liquid inside the channel **B)**. Because the system will have bubbles of oxygen, these will affect the experiment creating visible bubbles and big gaps of liquid **(arrows) C)**. Alternatively, we can create a setup made by PDMS, a mixture of a crosslinker and a monomer. **D)** We put this PDMS in a mould created with a 3D printer and we heat it for 4-6 h at 65ºC. **E)** Then, we peel the PDMS from the mould and attach to glass slides. Before attaching, we must do an O_2 plasma cleaning for 30 s to both the PDMS and the slides. **F)** We can also attach PDMS to the glass by 3D printing it directly to the glass. 3D printer will need a more viscous PDMS than the commonly used. **G)** PDMS is introduced in a cartridge, which is placed in the 3D printer **(H)**. **I)** The 3D printer can directly print wells of PDMS by pneumatic extrusion of the PDMS in the cartridge. **J)** As a result, we can obtain this glass with 3 wells, which we finally cover with more PDMS around the wells to fix any tiny hole. **K)** Inside these wells we can introduce topographic features, which are present on top of a wafer (black disk). A spin coating of PDMS will make a thin layer of PDMS. **L)** After 4-6 h at 65ºC, we obtain a thin layer of PDMS that we can cut and transfer to our wells **(M)** after applying a 30 s O_2 plasma cleaning on both the slides and the PDMS. *Image authorship. Own images.*

After the PDMS crosslinks, it is easy to detach from the mould and cut it in pieces that I attach to 24 x 60 mm^2 glass cover slips (Deltalab) (See Fig. 2.2 E)). To attach them, it is important to previously apply an O_2 plasma cleaning step to both the slips and the wells. Moreover, the texture of the bottom layer of the wells must be flat and smooth. Otherwise, the adhesion force between the glass and the well will not be sufficient, and they will deattach or leak the containing liquid. For example, after some time of using my mould it broke, and when I printed again it had a strip texture due to software and ink changes. This texture was transfered to the wells, and I could never recover the previous mould. Thus, I changed the procedure.

To solve the PDMS attachment problem to the glass, I decided to directly print the PDMS on top of the crystal with a 3D bioprinter (Inkredible+ 3D Bioprinter, from Cellink). This printer can print PDMS while it is not crosslinked, but it uses a much more viscous PDMS (SE 1700 Clear W/C, Dowsil. Compare Fig. 2.2 C) with Fig. 2.2 F)). [5] After mixing the PDMS, it is inserted into the printer, which extrudes it at a pressure of 200 kPa through a nozzle by pneumatic extrusion (See Fig. 2.2 G) and Fig. 2.2 H)). Using the file instructions we made with the software Slic3r, we were capable to print 3 circular wells of 2 mm height and 1.5 cm diameter of PDMS on top of a 24 x 60 mm^2 cover slip glass every 5 minutes (See Fig. 2.2 I)). After printing them, we crosslinked the PDMS by heating them at 65ºC for a couple of hours. Next, we added Sylgard 184 PDMS around the wells (See Fig. 2.2 J)), to close any little hole that would leak the liquid inside and we heated them at 65ºC for a couple of hours again to crosslink this PDMS. Glass slips with PDMS are then stored at room temperature until use. Before we use these setups, we apply an O_2 plasma cleaning for 30 s to make hydrophilic the PDMS. Otherwise, water inside the well would try to avoid contacting the walls.

These wells can contain around 120 µl, allowing us to record videos for long times without appreciable effects of evaporation, air flows and bubbles explotion. Because our particles sediment, the air flows in contact with the well will not be appreciable at the bottom of the well, where our particles move. Moreover, because the well is wide and hydrophilic, any generated bubble will move towards the walls of the well, where it can explote, far from the recording area. Thus, we do not see drift from these events. As a consequence, I could record videos for an hour without bubble problems as I did for chapter 7.

Furthermore, these wells are large enough to easily introduce other structures inside. For instance, also in chapter 7, I needed particles to move within a micrometric pattern. To build this pattern I used soft lithography. First, the design is created with a computer-aided design (CAD) software (AutoCAD, Autodesk), and then is transferred to a resist master (AZ 1512HS, Microchemicals GmbH) of 1.5 µm thickness on a silicon wafer using the Direct writing laser lithography (DWL 66FS, Heidelberg Instruments). The result of the process is a mould with

[5] For this PDMS we used a monomer-crosslinker ratio of 20:1.

the 3D design on top of the wafer. Thus, we can make as many PDMS replica as we want. But we need a replicas of 1 mm thick[6]. To create this layer, we need to put the PDMS on top of the wafer and then do a spin coating for 10 s and 1000 rpm (See Fig. 2.2 K)). The spin coating will create a thin homogenous layer of PDMS by spinning the wafer. Afterwards, we heat it at 65ºC for a couple of hours. Next, by applying an O_2 plasma cleaning to both the glass cover and the wafer for 30 s, we can place the replica inside the wells (See Fig. 2.2 L)), obtaining the desired open microchannel designs with the sub-micrometre step-like topographic (See Fig. 2.2 M)).

[6]For reusability purposes, the master is sylanized with trichloro (1H,1H,2H,2H-perfluorooctyl)sylane (Sigma-Aldrich) by vapor phase for one hour at room temperature. This operation reduces the adhesion of PDMS to the substrate, making very easy to peel the PDMS layer from the wafer, especially when it is very thin.

3

Dealing with digital videos: Limited by the size of the video

"Measure what is measurable, and make measurable what is not so."

Antoine-Augustin Cournot and Thomas-Henri Martin

It has been more than a century ago that the conception of a video was introduced in the men knowledge with Lumière brother's films production, but the foundations of what a video is are still hidden for part of the society, specially, since the introduction of the digital video. Nowadays, almost everybody carries a smartphone that can record videos. When somebody wants to record a video, they just need to take their smartphone from their pocket and shot it. If any of us record a video, probably this video will last for a few seconds and it will be storage until we change our smartphone, we notice the memory is full or simply, we lose it. Thus, most of the users do not think in the implications recording a video has, neither how this process works. Since recording a video is simple and we do not find any trouble when we usually do it, we have the perception that this should be always easy, and that no problem associated to this process should be. But when a scientist wants to record videos, this process can be tedious and show unexpected problems for an average user. For instance, since experiments imply the repetition of many conditions and/or samples, lots of videos will be recorded and memory problems can appear. Moreover, even if this problem is affordable, these videos can be big and if one wants to treat them, the computers will need to process a lot of information, which in turn can make the post-analysis very slow. And even if this problem is solved, no one ensures we are taking the correct time scale. Thus, when one needs to record videos, a basic knowledge is needed when mentally preparing an experimental setup.

In this chapter, I will show the problems I had during video recording and the solutions I found to solve them. These solutions not only include the software I coded, but also the analysis I did for extracting the speed and rotational diffusion time of the particles.

The present chapter is based on the software I coded and an expansion of the paper "Extraction of the propulsive speed of catalytic nano- and micro-motors under different motion dynamics"[121], from which I am first co-author, and to which I did the analysis and wrote the paper. At the same time, this paper is a deeper analysis on the work I did in the paper "Fundamental Aspects of Enzyme-Powered Micro-and Nanoswimmers"[122], where I am forth author. I would like to special thank Rafael Mestre, with who I shared the analysis of both previous papers.

3.1 | Size matters: Understanding the size of a digital video

A digital video is an electronic representation of a collection of images, also referred as *frames*, that are reproduced once and again and typically, jointly with the reproduction of some audio, which can be transcript into subtitles. Since its electronic representation, videos must be stored in electronic devices, as happens with any digital file. Nonetheless, videos can be very heavy files, and a proper management should be carried out. To understand why they can be as heavy, one needs to understand the data a video has. Since in this thesis I will only record mute videos without subtitles, I will forget about these components and I will focus just on the image part.

We already commented that videos are a succession of images or frames. Therefore, the size of our videos will be the result of the size of one of these frames per the number of frames our video has, which not only will depend on the seconds we are recording, but on the time required to change a frame. The time required to change from frame to frame depends on what one wants to see and typically, one does not refer to it in seconds but rather in the inverse of seconds. Although the inverse of seconds are the hertz, it is also typical to refer to them as *FPS* or frames per second. The FPS are essential to have the sense of good motion in a video. For example, if we want to see a smooth video without any problem and be able to see a movie rather than pictures in motion, the FPS should be equal or bigger to what is known as the flicker-fusion rate[123, 124], which is related with eye cells and neuron speed activity. Even though, some visual tricks can allow to see smoothness with smaller values and traditionally, films have taken advantage of it and have reduced to 25 FPS.

If one records a video without any compression, the size of an image will be the size of a raw image, which can be easily calculated. First, an image can be visualized as a lattice, with i columns and j rows, which creates a total number of cells or *pixels*. The size of each pixel will depend on the amount of information it is capable to carry out itself. Typically, colour images are saved in RGB format, and thus, each pixel has 3 values: one for red, one for blue and one for green. For each colour, a total of 256 levels of colour are available. This number of levels is chosen because is the maximum information one byte (8-bit) can have, and in computational world is known as a *char or byte*. Nonetheless, we can have pixels with a higher amount of information available. For example, some cameras record

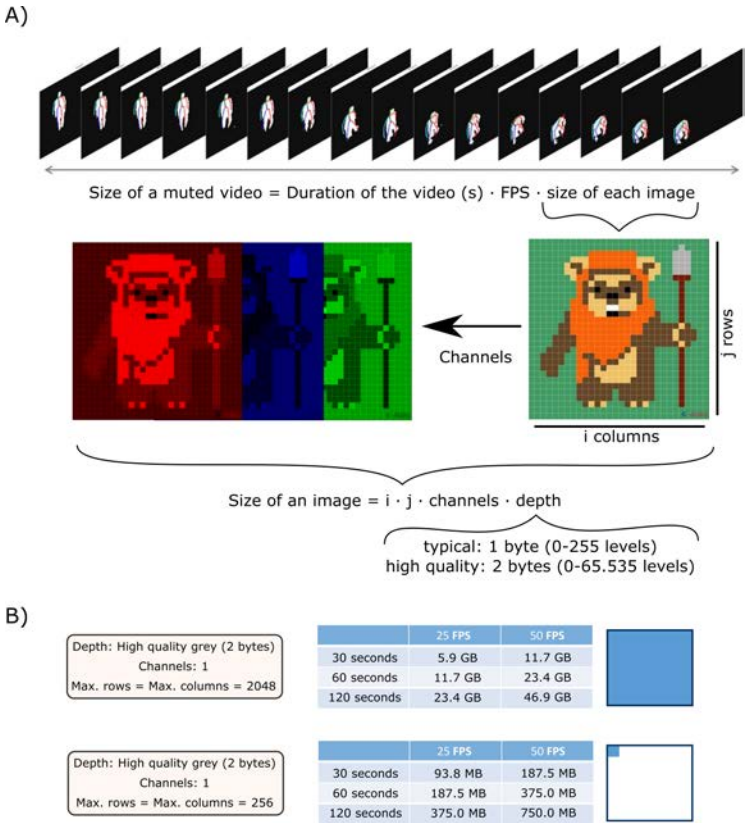


Figure 3.1: Size of a video. A) The size of a mute raw video is a function of the number of frames per the size of each image. **B)** In this thesis we only work with 16-bit black and white videos. Here we present an example of size for these videos considering that the camera we have in the microscope can record up to 2048x2048 pixels (4.2 Mpx). *Image authorship. Ewok: Ahooka'migurumi. No license was present. No changes were performed.*

with 2 bytes (16-bit, also known as *short*) colours, which make possible to record up to 65.536 levels of colours. This is the case for example of the camera installed in our microscope, where videos are only recorded in black and white, and thus only one channel is recorded. To have a better idea of the size a video can have, in Fig 3.1A) we can see a complete diagram for calculating the size of a video. Moreover, in Fig 3.1B) we can see the size of the video for different conditions. Usually, team members from my team record videos at 50 FPS for 30s. Thus, one can imagine the size of the videos we are recording in the lab. If one does not correctly choose the adequate parameters, tons of information will be saved, and data storage problems will appear. Therefore, it is important to choose the correct area to record, the number of frames one records and the duration of this video.

3.2 | Choosing the correct frames per second

As we saw in Fig. 3.1B), the correct chose of the frames per second has a big impact in the size of our videos. To correctly choose this parameter, in our experiments we need to rely on the time scale of the physical processes we want to study. Lower FPS will imply less video size and consequently, easily video management during post-analysis and less data storage. But lower FPS can also avoid our proper analysis. Thus, the right choose of this parameter is essential for any experiment we will do in this thesis.

3.2.1 Frames per second required for speed calculation

In the case of studying the individual behaviour of active particles, a good parameter to obtain is their speed, since active particles, by definition, generate their own motion. Obtaining a good value for the speed is essential to understand the fundamental mechanisms of particle propulsion since it can explain us fundamental mechanisms of particle propulsion, i.e. salts can change its speed[119, 110, 125]. In section 1.4.2 we already commented a simple procedure to obtain the speed by fitting the particle MSD. Splitting Eq. 3.5 in two regimes is always valid independently on other considerations, but as previously commented, the time lags at which one can use one or another approximation will depend on the rotational diffusion time τ_r, which will change depending on the radius of the particle and the temperature and viscosity of the medium. Thus, τ_r will limit our experiments.

The easiest way to obtain the speed of the particle is to use the regime where $t \ll \tau_r$. In this case, this condition implies that our FPS will be much higher than the inverse of τ_r. At the same time, if our FPS depends on τ_r, we can immediately deduce that any parameter that describe τ_r will modify the FPS we are taken. If we dive into the theoretical value for τ_r (see Eq. 1.4.8), we notice that τ_r grows cubically with the radius of the particle. Thus, the smaller the particle, the higher FPS we will have to take. But τ_r also depends on properties of the medium such as its temperature and its viscosity. Although these properties are also important, typically, experiments are performed in water or water with low quantity of salts at room temperature ($T = 298$ K), and little changes of these values will be negible for τ_r calculation. Thus, to define our FPS we will focus on the size of our particles.

If our particles are big enough, of $R = 3\,\mu$m or more, τ_r can be higher than the typical observable times ($\tau_r = 149.29$ s) and the enhanced diffusive regime might be too far in time to be visible. Thus, the ballistic regime will be easily observed. Following the regime approximation requirement, one can understand that the limit $t \ll \tau_r$ refers up to a time where we record the video for $\tau_r/10$ s, but this does not mean we should record up to $\tau_r/10$ s. In fact, we should record up to τ_r s because not all the data from this trajectory is useful to fit the MSD to obtain the speed. Usually, only the first 10% of the recorded trajectory is used due to the increasing error in the MSD construction. Therefore, if we record up to τ_r s, we will be able to select up to $\tau_r/10$ s to fit Eq. 1.4.13. If we consider that within this region, we will need at least 10 lag points for the MSD, we can approximate the FPS needed to around $100/\tau_r$ s$^-1$. In the case of the $R = 3\,\mu$m, this approximation gives us the condition to record at a minimum of 0.7 FPS. Recording with less than 0.7 FPS will give us inaccurate results. Of course, we can always take more than 0.7 FPS, but the pros will not beat the cons, since we will not obtain much more accuracy for the fitting, and the size of the video will grow unnecessary.

In the opposite scenario, if our particles are small, τ_r will be close to zero, and the temporal resolution of the camera of our microscope will not be as small, in general, to observe the ballistic regime. This will be the case for particles approximately below $R = 0.5\,\mu$m, where $\tau_r = 0.69$ s for water at room temperature. In this case, if we apply the previous hint, we should record at 145 FPS, but to have that high value one must has a very special high-speed camera, that is not usually found in labs since average commercial cameras only record up to 30 FPS. In these cases, we should only compare the diffusion values by linearly fitting the MSD. Because the speed cannot be extracted from the MSD and the instantaneous speed does not represent a real value, we should avoid using the speed of the particle to characterize its motion. Therefore, in these cases, FPS is not a critical value, and we just need to verify that the time within we record, particles are still active.

However, for particles of around $R = [0.5, 3]\,\mu$m of radius, τ_r can be within the observable regime and hence, we could also obtain data for the ballistic regime, although it may be difficult to capture the enough data points to fit our MSD due to the temporal resolution of the camera. At this level, one can ask himself if the safe recommendation we gave for the FPS can be better estimated and if in fact, the error of considering that approximation is appropriate. For instance, we obtained the previous recommendation as a factor of 2 orders of magnitude: one to have enough points for the fitting and one to accomplish with the approximation regime. For a parabolic fitting, we will assume that 10 points are good enough to fit it. Then, we could verify what $t \ll \tau_r$ means. To investigate this problem, we find that expanding the MSD approximation to higher polynomial orders when doing the Taylor expansion leads to more accurate and interpretable results. If we expand the MSD equation for the active case (Eq. 3.5) up to fourth order:

$$\mathrm{MSD}(t) = \underbrace{\underbrace{\underbrace{4D_t t + v_p^2 t^2}_{\text{n=2}} - \frac{v_p^2}{3\tau_r} t^3}_{\text{n=3}} + \frac{v_p^2}{12\tau_r^2} t^4}_{\text{n=4}} + \mathcal{O}(t^5), \tag{3.2.1}$$

we obtain different expressions depending on the desired degree of accuracy. In particular, we recover the already proposed expression for the second-degree approximation n=2 of section 1.4.2, but we could expand up to higher orders, theoretically infinite. Therefore, we might wonder what are the limits that we encounter when we apply this equation up to a specific term and what is the error associated.

Cutting this approximation up to the n-th order will always give us an intrinsic mathematical error, which decreases for higher orders but increases as we calculate for a time lag far from 0 s, since the Taylor expansion is done around 0 s. If we do the following change of variable, $x = t/\tau_r$, expression inside parenthesis in Eq. 3.5 can be rewritten as $m = -1 + x + e^{-x}$. From this equation, we can know this error depending on the n_0 terms we included in the expansion:

$$\begin{aligned} \%\ error &= 100 \cdot \left[1 - \frac{\left| \overbrace{\sum_{n=2}^{\infty} \frac{(-1)^n}{n!} x^n}^{m} - \overbrace{\sum_{n=2}^{n_0} \frac{(-1)^n}{n!} x^n}^{Taylor\ expansion} \right|}{\underbrace{\sum_{n=2}^{\infty} \frac{(-1)^n}{n!} x^n}_{m}} \right] \\ &= 100 \cdot \left[1 - \frac{\left| \overbrace{\sum_{n=n_0+1}^{\infty} \frac{(-1)^n}{n!} x^n}^{Taylor\ residual} \right|}{\underbrace{\sum_{n=2}^{\infty} \frac{(-1)^n}{n!} x^n}_{m}} \right]. \end{aligned} \tag{3.2.2}$$

Although we know which is the error for each of our points we commit when we cut the series at n_0 order, we must remember that to extract the value of the speed we are doing a fitting, which imply to take 10 points as we referred before. This means that since the error of this expression grows with x, the further x from 0, the more error this point will have, but the total error of this fit will not coincide

with the one we could extract from the residual of the Taylor expansion since the fitting is adjusted to the 10 points, and the firsts points have less error than the lasts ones. Since calculating this error can be complex, one could be tempted also to increase the order of Taylor polynomial to reduce the error and forget to calculate it, but this might not be appropriate. The main reason is that the prefactors of the power series (D_t, v_p, τ_r) are unknown and the equation is always fitted to experimental data. Having too many terms to fit could create overfitting of the results. Hence, the number of terms added to the equation should be kept as low as possible, while still allowing an accurate representation of the experimental MSD with minimum error. Thus, in some cases can be useful to add one or, as a maximum, two more terms. For instance, in the case of adding the third term (n=3) to this expansion, the overfitting problems will not be important and new information beyond reducing the error is obtained, as we can indirectly obtain an experimental τ_r.

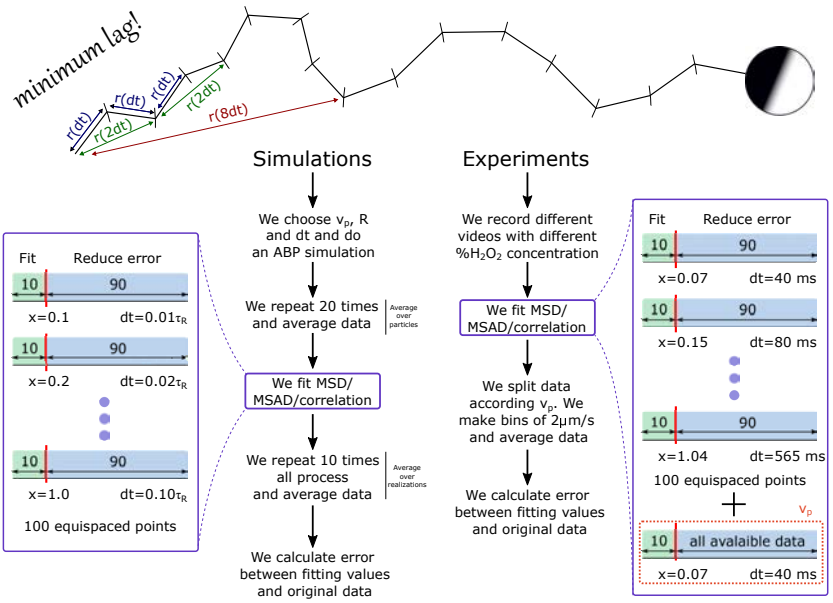


Figure 3.2: Overview of the analysis performed to the simulations and experiments.

In order to check the accuracy of these equations for different examples, and to properly check the validity of Eq. 1.4.13 as a function of the FPS, we analogue the optical microscopy tracking experiments by simulating them with a simple ABP model (Eq. 1.4.11) (See Fig. 3.2). Then, we construct the MSD from the particles simulated and we estimate the parameters fitting the MSD to Eq. 3.2.1. These parameters are compared to the actual values used in the simulations. Consequently, we can obtain the relative error to them.

To compare the parameters provided to the ABP model and the fitted ones, we averaged the fitted parameters values of 10 full simulations. For each full simulation, we obtain the fitted parameters by fitting the average MSD of 20 non-interactive active particles which move with the same active speed v_p and are subject to Brownian fluctuations.

To study the effect of the FPS, we repeated the previous calculation simulating particle trajectories for different time steps dt, to which the inverse of the time step is the simulated FPS. Moreover, we set that 10 points are enough to fit a parabolic curve, and hence, for each simulation we will fit data up to $10 \cdot dt$ s. If we divide $10 \cdot dt$ per τ_r of the particle we will obtain a corresponding x value. The validity $t << \tau_r$ corresponds to $t/\tau_r = x << 1$, and thus we can also compare the validity of this expression for different meanings of what $t << \tau_r$ means. Nontheless, the total time simulated will be $100 \cdot dt$ s because we consider that we only fit the first 10% datapoints of the MSD. This model does not include other sources of error such as errors during the tracking of the particles. Thus, the errors associated to the estimation of the parameters only come from over- or under-fitting of the equations and by the Taylor expansion.

As a result, in Fig. 3.3 we present the errors calculated when compairing the exact values to the fitted values. The addition of a third term in the equation lead to a better estimation of the speed of the particle, with a fast convergence to the exact value. For almost all the comparisons, the error for estimating the speed using the MSD method is lower than 5%. However, if we use the second order approximation, the error can reach up to 15% or more. Moreover, Fig. 3.3 demonstrates the meaning of $t << \tau_r$. In fact, if we use $t \sim \tau_r$ we could use the third order approximation and still, obtain almost the exact value. For example, this could be useful in the case of a small τ_r and being limited by camera limitations or, if one wants to track these particles for longer times but still, be able to track its speed individually. Furthermore, Fig. 3.3 is useful for any beginner in the field that wants to know which FPS and for how long it needs to record their videos to properly estimate the speed of their particles, optimizing thus the amount of data to be recorded.

Interesting, particles smaller than $R = 0.6$ μm show high errors on calculating the speed using these approximations, breaking the behaviour of the rest of particles simulated. Specially, if you use the third order approximation. Hence, we need to be careful in this limit. To complete the information we can extract from these simulations, in Fig. 3.4 we plot the D_t for these fittings. The results for D_t are shocking since the large error obtained, except for particles smaller than $R = 1$μm if third order approximation and $x < 0.15$ is used. For bigger particles, the value obtained should be carefully used because of the large error it can contain. Therefore, it is better to focus on the speed rather than on D_t value.

Although these results could be enough to conclude, I also worked experimentally with these particles and hence, I checked the above data with Pt-SiO_2 of

$R = 1\mu$m for different fuel concentrations. In this simple experiment, I mixed a suspension of particles in miliQ water with hydrogen peroxide to obtain different solutions at different hydrogen peroxide concentrations $(1\%, 2\%, 3\%, 4\%, 5\%)$. Then, for each solution I put 10 µL on top of a glass slide and recorded with a microscope at 25 FPS for a few minutes. Each particle recorded is tracked, and then analysed individually. Although I had data from different fuel concentrations, I finally did not consider the fuel concentration to split data information but joined all data together. I decided to join all the information because when dealing with a batch of particles, particles can have a dispersion in its size, can have defects or its surface can have a different condition to do the catalysis, thus affecting the speed of particles as seen in experimental studies[68]. Therefore, I preferred to average particles within the same speed range calculated rather than by its fuel concentration.

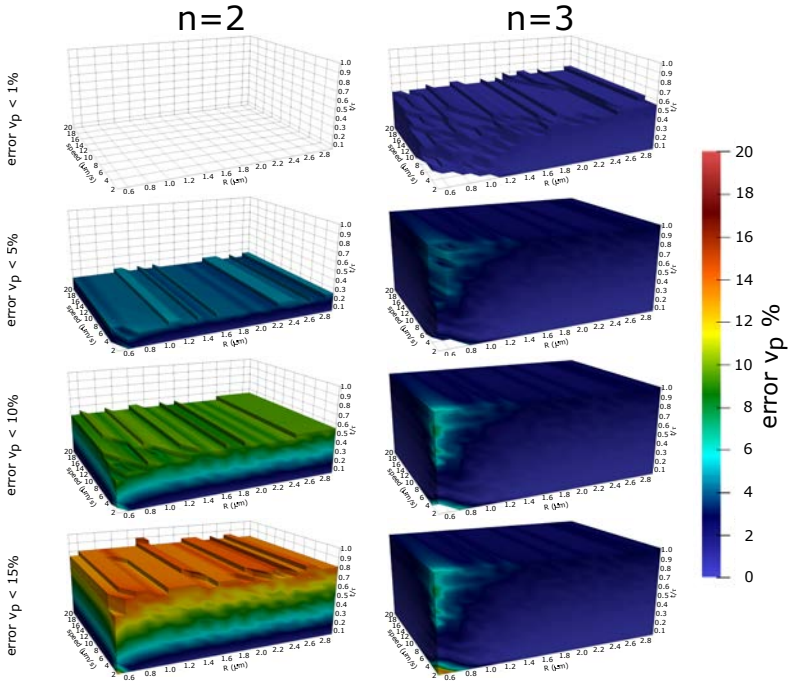


Figure 3.3: Error in % of the estimated particles speed after fitting its MSD to a Taylor expansion (Eq. 3.2.1) of n=2 terms or n=3 terms as a function of the particle radius, speed and fitted region $x = t/\tau_r$**.** We obtain the relative error between the speed introduced in the ABP model and the speed obtained from fitting the MSD of ABP simulations. First row shows all the simulations where the fitted speed v_p differs less than 1% to the values introduced in the ABP simulation. The following rows shows data for <5%, <10% and <15%. Left column uses n=2 approximation, while right column uses n=3 approximation.

To imitate the simulations with these experiments, I decided to fit the MSD both for Taylor orders $n = 2$ and $n = 3$ up to different values of x. Originally, movies were recorded at 25 FPS but to accomplish with the same structure as in the simulations I removed data in between. Therefore, I treated movies as if they were multiples of 25 FPS. Thus, for each FPS used I will have a different estimation of the speed, obtaining a collection of different speeds v that I named after v_x, since each FPS is coupled to a specific value of x as calculated for simulations. But of course, opposite to the simulations, we do not know a priori which of these v_x is the correct v_p of the particle. Since with the simulations we saw that third order approximation has lower error, I assumed v_p will be closest to the real value using this approximation. Moreover, to reduce the error of the MSD, I decided to use the full data instead of using only 100 points as with every v_x. Hence, the higher available FPS (25 FPS) will give the maximum amount of data points, and

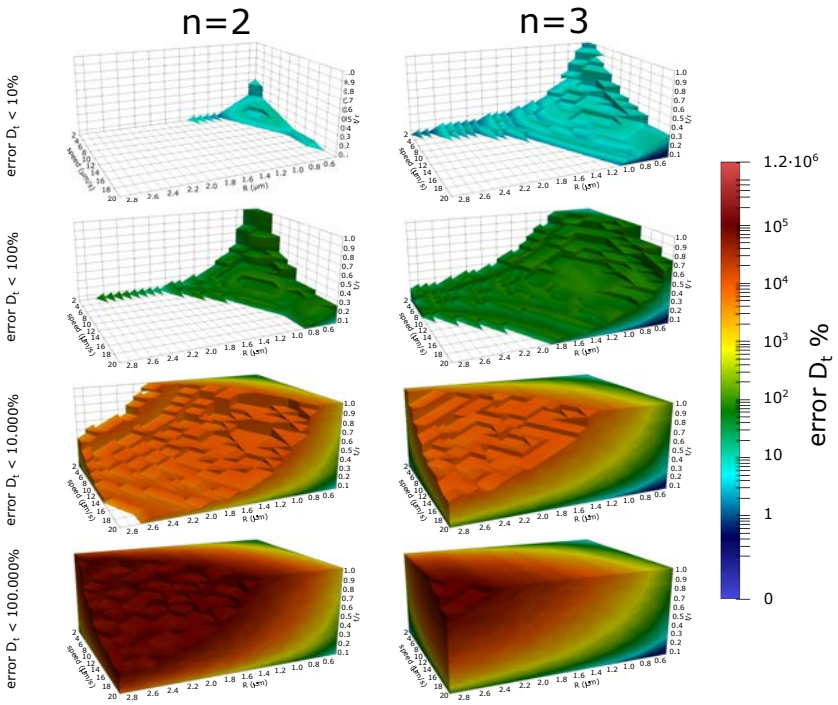


Figure 3.4: Error in % of the estimated particles D_t after fitting its MSD to a Taylor expansion (Eq. 3.2.1) of n=2 terms or n=3 terms as a function of the particle radius, speed and fitted region $x = t/\tau_r$. We obtain the relative error between the D_t calculated theoretically and the D_t obtained from fitting the MSD of ABP simulations. First row shows all the simulations where the fitted D_t differs less than 10% to the values introduced in the ABP simulation. The following rows shows data for $<10^2\%$, $<10^4\%$ and $<10^5\%$. Left column uses n=2 approximation, while right column uses n=3 approximation.

thus I considered this case as the one to obtain v_p. Then, to have better statistics, data was averaged over bins of 2 µm/s. Finally, we calculated the error of v_x with respect v_p (See Fig. 3.2 for a complete overview of the process).

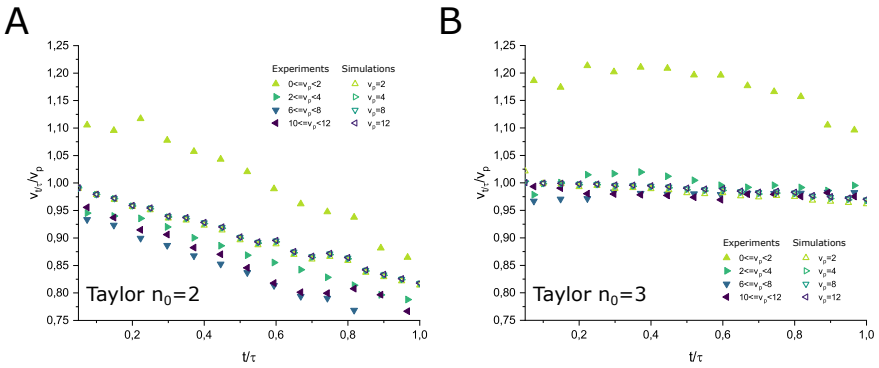


Figure 3.5: Comparison between experimental and simulation data for the error extracted from speed fitting using MSD at different Taylor expansions and different $x = t/\tau_r$. A) Taylor approximation to second order. B) Taylor approximation to third order. Empty points correspond to simulations, while full points are obtained from experiments.

In Fig. 3.5 we plot the estimated error obtained experimentally, and we overlap the simulated series for these speeds, size and x. Experimental and computational data agree, except for low-speed values since for this range the error associated to the tracking is very important.

While the second order approximation is only good for low values of x, the $n = 3$ order is also good for higher values of x. In the case of $n = 2$, the error associated to speed calculation grows linearly with higher x, both experimentally and theoretical. Moreover, experimental data presents a shift compared with simulations. In contrast, the simulations for $n = 3$ order do not present these problems. To explain this underestimation there could be different answers, yet there is a fundamental one we still have not considered. What if experimentally our particles do not match the theoretical τ_r? In this case, the experimental line will be displaced, and that could explain this underfitting. To better explain this problem, and since τ_r is an important characteristic time, in the next subsection I also do a similar calculus to properly calculate τ_r.

3.2.2 Frames per second required for τ_r calculation

In the previous subsection we observed that the theoretical τ_r could be under- or over-estimated experimentally. Therefore, a subsequent question arises: how can we estimate τ_r without using the theoretical value?

Although from a theoretical point of view the calculus of τ_r is simple, experimentally different questions arise. For example, there are different methods to use, each of them with different errors implicated. Which one has less error? And how can we calculate each of them? Furthermore, some of these methods rely on obtaining the particle orientation, but sometimes we cannot differentiate both parts of the Janus, and hence defining the orientation is not directly possible. And of course, as with the speed calculation, the fundamental question arises: how many FPS do I need to use while recording my videos?

To answer all these questions, I analysed 3 different methods to calculate τ_r:

- Fitting the MSD with Eq. 3.2.1 with at least $n = 3$ or higher order.
- Fitting the MSAD instead of the MSD.
- Fitting the velocity autocorrelation.

Using the experiments and simulations already realized in the previous subsection, I also calculated τ_r using the above methods. In the next pages I will show the results obtained for each method.

Fitting the MSD with Eq. 3.2.1 with $n = 3$ and $n = 4$

We present the results in Fig.3.6 for the estimated τ_r for simulations using the MSD fitting to cubic and forth order. In general, the errors estimated for τ_r are orders of magnitude higher than the ones calculated for the speed.

In the cubic approximation case, if we fit the MSD up to $x < 0.1$, we can obtain the estimated τ_r with less than 10% error if we have particles bigger than $R > 1.9$ µm. For smaller particles, this estimation can also be good enough (less than 25%), but the more you reduce the size, the faster the particle must move. However, as soon as we increase x, the error for τ_r increases, and we can have up to 75% error on the estimation. Nonetheless, for particles smaller than $R < 1.2$ µm, the x used for fitting is very important because we find a region for low x where the estimated τ_r can have errors up to $1.2 \cdot 10^5$% the real value.

In the fourth approximation case, we can also obtain estimations with less than 10% error if $R < 1.2$ µm, but only if the speed is high enough. Interesting, we can explore more values of x to have this low error. Values for smaller speeds have a similar trend with the cubic order, while values for bigger particles have an erratic trend. Depending on the region of interest, it may better to use the cubic or the

fourth order approximation. Thus, the size of the particle is very important if we decide to estimate τ_r using this method. In the case of the particles I used for these experiments ($R = 1\mu$m), the fourth approximation is better.

As in the study of the speed of the particles, we can also test this method directly with experiments. However, the procedure for comparing experimental and theoretical errors of τ_r must be different. In the case of the speed, the estimation obtained by fitting the simulated MSD had a relative error of less than 5% with

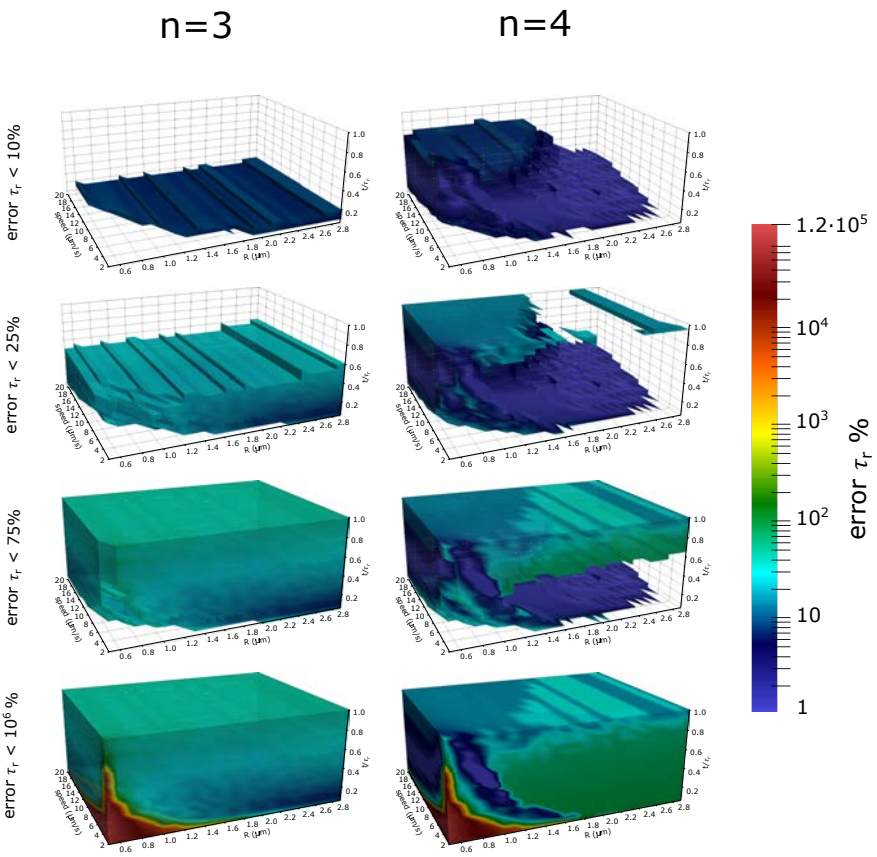


Figure 3.6: Error in % of the estimated particles τ_r after fitting its MSD to a Taylor expansion (Eq. 3.2.1) of n=3 terms or n=3 terms as a function of the particle radius, speed and fitted region $x = t/\tau_r$. We obtain the relative error between the τ_r introduced in the ABP model and the τ_r obtained from fitting the MSD of ABP simulations. First row shows all the simulations where the fitted τ_r differs less than 10% to the values introduced in the ABP simulation. The following rows shows data for <25%, <75% and <10^6%. Left column uses n=2 approximation, while right column uses n=3 approximation.

the parameters introduced in the simulations, with independence of x. Hence, we could assume that in the case of the experiments, we could have a reference value for the speed if we use the same order approximation, the lowest x and the longest MSD available. However, notice that in the case of extracting τ_r, we have an important dependence on the x parameter, which in turn depends on τ_r. Thus, in this case we cannot proceed as before. Instead, I decided to directly estimate the value of τ_r to compare with the results from simulations. But we started with the hypothesis that the experimental value can be different to the theoretical one. Hence, instead of comparing directly the experimental and simulated values, I compare the variability in the data.

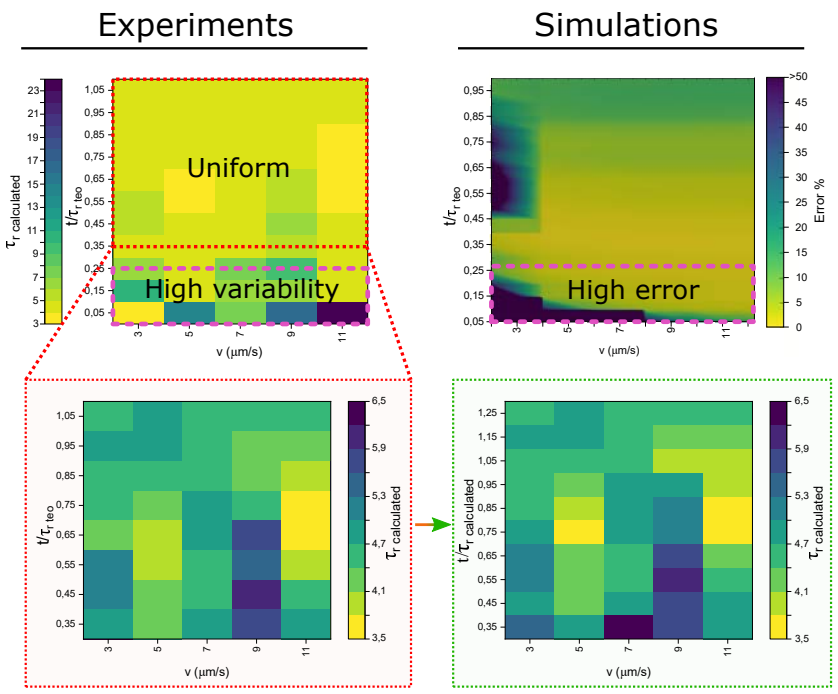


Figure 3.7: Experimental results for τ_r. Top row: τ_r calculated by fitting the MSD to Eq. n=4. Left: Experimental data. Right: Simulated data. Colormap for simulated data is cut from 9000% error to 50% to present better regions with low error. Simulated results are presented as the relative error to the real value of τ_r. **Bottom row:** Experimental data from Uniform area. Left: Colormap limited to the available range of data. Right: Axis t/τ_r recalculated using the average τ_r in the Uniform area as the real τ_r ($\tau_{r\ calculated} = 4.50 \pm 0.07$s).

To start with the experimental data, I assume that the theoretical τ_r is the correct value. Hence, I can create a map using v_p and x (See Fig. 3.7Top left)). The values obtained present a high variability for low x, but high consistency for higher values of x. If we compare this data to the extracted from simulations

(See Fig. 3.7Top right)), we see that there is also a large variability for low x, but the rest of values have an homogenous relative error of 5-15%. Thus, we can consider that the homogeneous area in the experimental data will also have a low relative error. When we zoom into this region (See Fig. 3.7Bottom left)), we can assume this region contains the real value for τ_r. Therefore, if we average all data points included in this region we can obtain an estimation of our τ_r, which is $\tau_{r\ experimental} = 4.50 \pm 0.07$s. This value is incompatible with the theoretical data ($\tau_{r\ theoretical} = 5.4$s). If we use $\tau_{r\ experimental}$ for the calculus of x we can rebuild the x axis (See Fig. 3.7Bottom right)). The new axis is still in the range of the simulated area with homogenous error, and hence we can conclude that our estimated value is the best estimation we can give with this method.

Fitting the MSAD

The next method to study is the calculus of the MSAD. To calculate the MSAD we need to calculate the MSD but using the rotational components instead of the translational components. In an ABP model the MSAD can be related with τ_r given:

$$MSAD = 2dD_r t + \omega^2 t^2, \tag{3.2.3}$$

where d is the number of rotational dimensions, D_r is the rotational diffusion coefficient, which is the inverse of the rotational diffusion time τ_r, ω is the active angular speed and t is the lag time. To have $\omega \neq 0$, we need chiral particles. For example, we can obtain chiral particles modifying the body geometry of the particles[126], fabricating them with non-uniform parameters[67], or by clustering several active particles[127]. However, my active particles are not chiral and hence $\omega = 0$. Because in my system there is only one degree of freedom for rotation, $d = 1$.

In the case of the simulations, I have the values for the angle already calculated, and hence I can use them directly. But in the case of the experiments, we need to distinguish any asymmetry in the particles to draw their director vector. Because Janus particles have two different sides, if we can distinguish them, we will obtain the angle of the particle. But in the experiments I did, the magnification was low to capture more area of motion, and hence I could not track their orientation by simply differentiating the active from the inactive side. Instead, I took the angle from its instantaneous speed vector, and I assumed that this angle is the same one I could obtain by distinguishing both Janus parts (See Fig. 3.8A)). Thus, to better compare experiments with simulations, I will also present data for simulations obtaining the angle from the instantaneous speed. However, if possible, it is better not to use this method because it is very sensible to the FPS used. The instantaneous speed includes a random component from Brownian motion (See Fig. 3.8B)), which we do not average and hence we introduce it in the calculus.

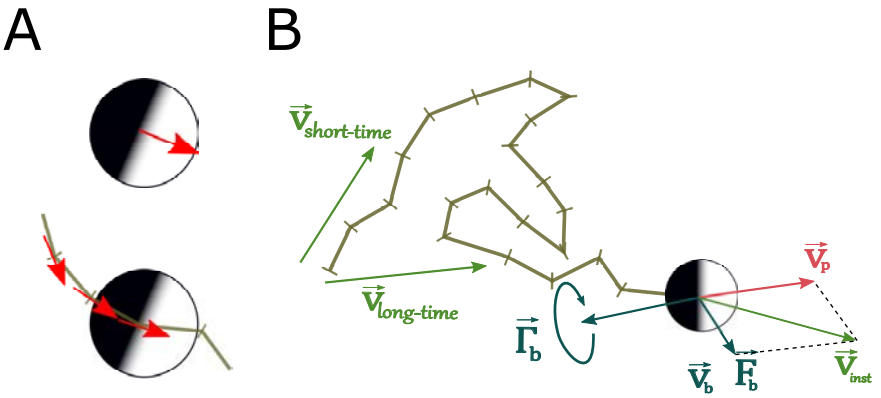


Figure 3.8: A. Approaches to calculate the angle of a particle. The angle of the particle (top) is obtained from the vector (red) perpendicular to the separation between the inactive (white) and active (black) areas of the Janus. We can measure this vector using image processing or alternatively, we can calculate it from its instantaneous speed (bottom), tangent to its trajectory (green olive). **B. Components of the instantaneous speed.** When we calculate the instantaneous speed we are including not only the propulsive speed of the particle ($\boldsymbol{v}_p$), but also we include a Brownian component ($\boldsymbol{v}_b$) arising from the random Brownian force ($\boldsymbol{F}_b$). *Image authorship. Own image.*

One question that also arises when we calculate τ_r is the method to average over the collection of particles we have. Each particle has its own MSAD, which we will fit to obtain its own τ_r. Thus, we can average over the population of τ_r obtained. Nonetheless, we could also average the MSAD, and then fit once to obtain the averaged τ_r. Both methods are equivalent, but maybe one produces less error. Because errors when calculating τ_r can be very large, I decided to use both approaches too.[1].

In Fig. 3.9 I present the results for the simulations using the four options[2]. First, these results confirm a difference between both averages. However, this difference is less than 10% relative error, being the errors bigger if we average over τ_r instead of over the MSAD. Second, we observe a contained relative error for the simulated parameters. Notice that when we calculated τ_r using the MSD we obtained errors up to 10^5%, while in this case the maximum relative error is 100%. Third, as we predicted, if possible we should avoid using the instantaneous speed.

[1]Indeed, when I did the approach for the speed, I averaged the MSD and then fit this value, but when I was analysing the experimental data I was fitting each MSD and then averaging the different speeds. However, the error obtained in the simulations was very small, as in the case of using the ABP angle here, and hence I discarded that analysis

[2]Notice that the angle must be in radians, and not limited to be between 0 and 2π. In python, you can extend the angles between 0 and 2π to increasing values using the method *numpy.unwrap(angle)* from numpy library. Also, if one takes the angle from the instantaneous speed, there can be 0 values because the speed at that moment is zero, but the previous and following steps are not. Thus, you could have had a different value. Consider filling these zeros with the previous angle values.

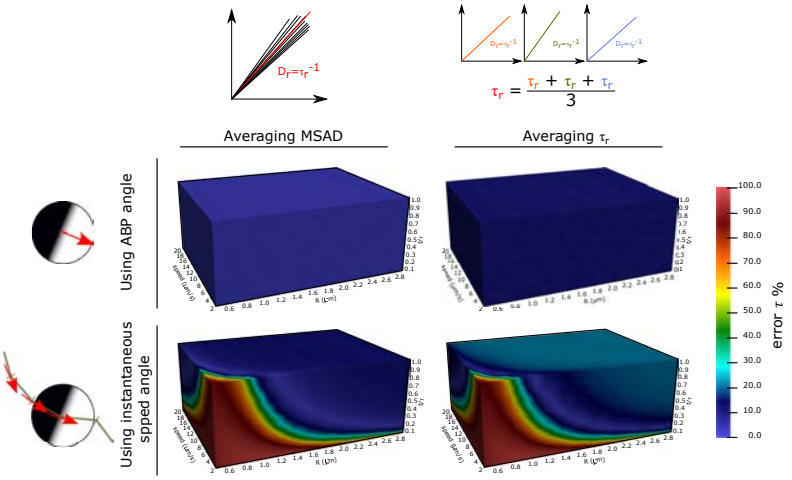


Figure 3.9: Relative error of the estimated τ_r after fitting its MSAD. Top row. MSAD calculated using the angles obtained in the ABP model. Left: First average the MSAD, then fit. Right: First fit, then average τ_r. **Bottom row.** MSAD calculated using the angles obtained from the instantaneous speed. Left: First average the MSAD, then fit. Right: First fit, then average τ_r.

When we use the angle from the correct source (e.g., optical angle tracking) the error can be almost negligible, of 10% maximum, and we do not observe a dependence on x, which can ease the calculus. However, if we use the instantaneous speed, we will have errors of up to 100% for particle radii $R < 1.0\mu$m. Brownian effects should be less noticeable if particles are bigger or move faster, and in fact we note that the error shrinks as the particles are bigger or move faster. Moreover, to calculate the vertical axis (x) we use 10 points. Thus, to increase x we enlarge the time step, and hence the less noticeable should be the influence from Brownian as we observe (See particle trajectory at Fig. 3.2).

We consider now the the experimental data, which we plot in Fig. 3.10. To obtain these results I repeated the same process as when obtaining τ_r from the MSD. We find a similar trend between simulated (left column) and experimental data (Middle column), with an area of high error for smaller speeds and x ratio. Therefore, we will exclude this data from the analysis. Although simulations present an area with lower error (yellow region), obtaining the experimental value for this area is difficult because it depends on the vertical axis (x), which includes the term that we want to calculate (τ_r). Hence, I preferred to take the values for an area (area inside orange dashed lines) further from the regions with high error. The average of τ_r over this orange dashed lines area gives $\tau_{r\ calculated} = 3.26 \pm 0.10$ s if we average over MSAD and $\tau_{r\ calculated} = 4.20 \pm 0.13$ s if we average over τ_r. These values are not compatible, which is surprising given we use the same data.

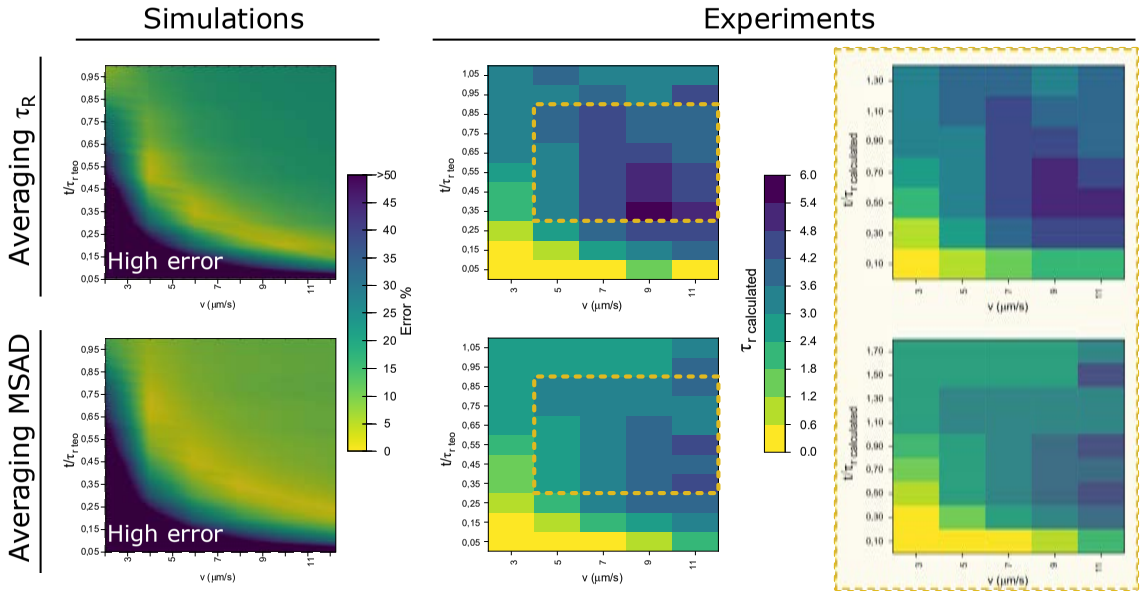


Figure 3.10: Experimental results for τ_r calculated experimentally using the MSAD. We fitted Eq. 3.2.3 up to different values of x and speeds. **Top row**. First, we fit MSAD. Then we average τ_r. **Bottom row**. First, we average MSAD. Then we fit τ_r. **Left column.** We calculate the relative error between the fitted τ_r and the theoretical τ_r using the simulated data. Colormap for simulations is cut at 50% error to present better the regions with lower error (Maximum relative error is 100%.) **Middle column.** We calculate τ_r from experimental data. Inside the dashed orange lines area, we calculate $\tau_r = 3.26 \pm 0.10$ s for the case of averaging first the MSAD and $\tau_r = 4.20 \pm 0.13$ s for averaging the different τ_r. **Right column.** Middle column with axis x updated using the previous measured τ_r.

Fitting the velocity autocorrelation

The last method to study uses the velocity autocorrelation. Velocity autocorrelation is a measure that is implicitly calculated when we calculate the MSD, prior to integrations present in the MSD. The velocity autocorrelation follows:

$$< \boldsymbol{v} \cdot \boldsymbol{v} >=< cos^2(\theta) >= e^{-t/\tau_r} \tag{3.2.4}$$

But to ease the fitting, we can also apply the log:

$$log(< \boldsymbol{v} \cdot \boldsymbol{v} >) = log(< cos^2(\theta) >) = -\frac{t}{\tau_r} \tag{3.2.5}$$

The calculus of the autocorrelation can lead to fitting problems because under some circumstances, some points can be negative, and thus *Not a Number (NaN)* when plotting the log. [3] Moreover, because we also use the angle, we will find the same problems presented when calculating the MSAD.

In Fig. 3.11 we present the results using the simulated results. These results are very similar to the ones we obtained for the MSAD, and hence, both methods can be used undistinguished. If we can obtain the angles directly from the observation of the particles, the results are perfect, but if we need to use the instantaneous speed we will obtain large errors for small speeds and particles.

[3] Nonetheless, this method was the one I used for the analysis present in chapter 5.

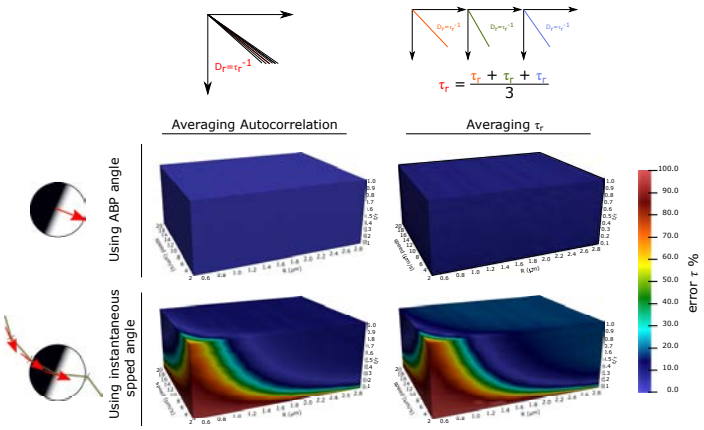


Figure 3.11: Relative error of the estimated τ_r after fitting its velocity autocorrelation. Top row. Velocity autocorrelation calculated using the angles obtained in the ABP model. Left: First average the velocity autocorrelation, then fit. Right: First fit, then average τ_r. **Bottom row.** Velocity autocorrelation calculated using the angles obtained from the instantaneous speed. Left: First average the velocity autocorrelation, then fit. Right: First fit, then average τ_r.

In Fig. 3.12 we plot the results for the experiments, repeating the same procedure as in the previous studies. Experimental values for τ_r using the autocorrelation method are very close to the values using the MSAD method. In fact, the calculated τ_r between both approaches are compatible. In this case, I obtained a value of $\tau_r = 3.41 \pm 0.07$ s when first averaging the autocorrelation and then fitting it and $\tau_r = 4.49 \pm 0.14$ s when averaging the calculated τ_r. Again, as in the calculus of the MSAD we obtained two results, depending on the average, that are not compatible.

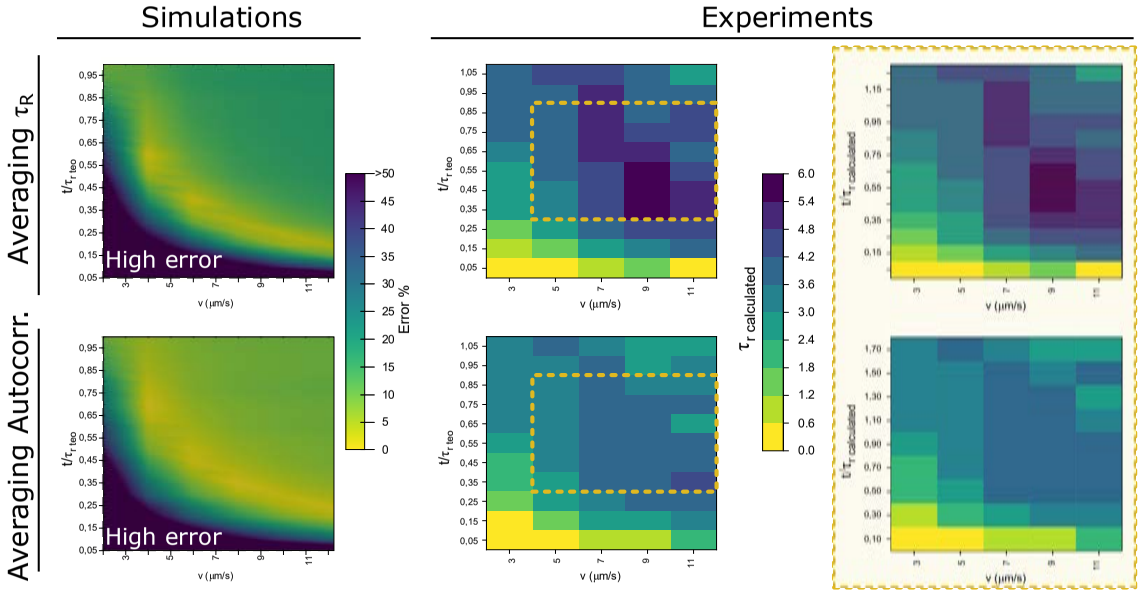


Figure 3.12: Experimental results for τ_r calculated experimentally using the velocity autocorrelation. We fitted Eq. 3.2.5 up to different values of x and speeds. **Top row**. First, we fit velocity autocorrelation. Then we average τ_r. **Bottom row**. First, we average velocity autocorrelation. Then we fit τ_r. **Left column.** We calculate the relative error between the fitted τ_r and the theoretical τ_r using the simulated data. Colormap for simulations is cut at 50% error to present better the regions with lower error (Maximum relative error is 100%.) **Middle column.** We calculate τ_r from experimental data. Inside the dashed orange lines area, we calculate $\tau_r = 3.26 \pm 0.10$ s for the case of averaging first the velocity autocorrelation and $\tau_r = 4.20 \pm 0.13$ s for averaging the different τ_r. **Right column.** Middle column with axis x updated using the previous measured τ_r.

Updating the speed measures with the experimental τ_r

When we compared the computational and experimental results in Fig. 3.5 A), we observed a gap between the experimental and computational values. However, in this plot we used the theoretical τ_r but experimentally we obtained different values. Thus, we can use these values to plot again that figure. The values already obtained experimentally for τ_r are:

Average method	MSD $n = 4$	MSAD	Autocorrelation
Average first, then fit.	-	3.26 ± 0.10 s	3.41 ± 0.07 s
Fit first, average then.	4.50 ± 0.07 s	4.20 ± 0.13 s	4.49 ± 0.14 s

We mainly obtained two different values for τ_r, depending on the average used. Although the method can be different, the values are compatible if we use the same average approach. Thus, we can average these values and obtain 4.4 ± 0.2 s for when first fitting and then averaging, and 3.3 ± 0.1 s for when first averaging and then fitting. In Fig. 3.13 we present an updated version of Fig. 3.5 A) considering these two values for τ_r. Now, experimental and computational values match, and hence we have a possible reasoning for the difference we previously found. The best set is when we average first, and then we fit, as computationally we already observed.

3.2.3 Conclusions

To summarize, choosing the FPS and the n-degree equation to fit the MSD is clearly context-dependent, and the choice depends on the size of the particle, its speed, and the fitting interval. In general, a reliable value for the FPS can be estimated as FPS=100/τ_r, although we can set smaller FPS if needed.

For particles smaller than $R = 0.5\mu$m, we will usually obtain a linear MSD, which is proportional to the enhanced diffusion constant (D_{eh}). Therefore, the best parameter to present is D_{eh}.

Bigger particles will present a non-linear MSD. For these particles, a third-degree approximation provides a great improvement with respect to a quadratic approximation for speed calculation, with estimated relative errors of less than 10% as compared as to the real value. Nonetheless, translational diffusion constant and rotational diffusion time provide estimations with large error. Translational diffusion constant presents high error if we use bigger particles than $R = 1\mu$m, and if their speeds are higher because Brownian fluctuations play a smaller role than the active motion. To report diffusion, the linear part of the MSD, long after the rotational diffusion time, should be fitted to a linear equation to extract the enhanced diffusion coefficient.

If possible, we should analyse experimentally τ_r. When calculating τ_r, it is better to directly use the tracked angle from the particle orientation, but if not, it

can be calculated by using the angle obtained with the instantaneous speed. This method will enlarge the error associated, but it can give good results. The best method to calculate τ_r will depend on the values of particle and speed size, and the FPS, but consider using the autocorrelation or the MSAD approach. Finally, depending how we do the average over the different particles, the result can differ. Hence, it is important to indicate which approach is used.

Following the guidelines in this section videos can have the adequate FPS and duration, which in turn will solve the possible problems of data storage you could have.

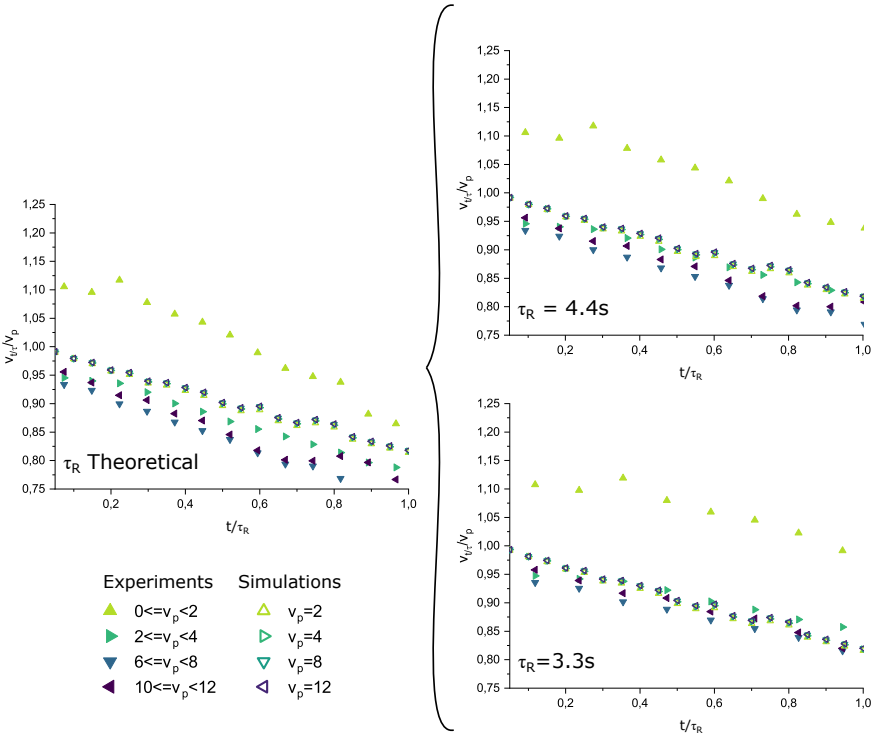


Figure 3.13: Comparison between experimental and simulated data for the relative error extracted from speed fitting using MSD at Taylor expansion $n = 2$ and different $x = t/\tau_r$. Left: Original data using theoretical τ_r. **Right:** We used the experimental τ_r obtained by different averaging instead of the theoretical value. Top: Fit first, average then. Bottom: Average first, then fit.

3.3 | Choosing the correct size of your region of interest

One of the challenges when recording a video is to fit the complete scene. The first limitation to fit is the *field of view* or *FOV* of our microscope. If the FOV is not big enough, it may happen that we observe the initial part of a process, but we cannot observe the final. Thus, it can conditionate the experimental setups we can have. For example, imagine we need a microstructure as we presented in chapter 2. The size of this microstructure will be limited by the FOV. In the insets of Fig. 3.14 we can better understand this effect. To overcome this problem, one can analyse the parameters the FOV depends on. For normal lenses (e.g., not fisheye lens), the FOV can be estimated as $FOV = 2arctan\left(\frac{Sensor\ size}{2 \cdot focal\ length}\right)$[128], and thus, to enlarge this area we can reduce the magnification of the objective, change the kind of lens of the objective or buy a new camera with a bigger sensor if needed, but this would cost much more. Although reducing the magnification can be a good idea, one should be aware of the resolution the particles will have with such objectives. For example, in the microscope I used (Sensor size = 2048x2048 px), a $R = 2.5$ µm particle under a 10x objective fits in just 9x9 px. At this resolution, checking visually the orientation of these particles is not possible, but one can its position. Thus, depending on our aim, we can play with the magnification or not.

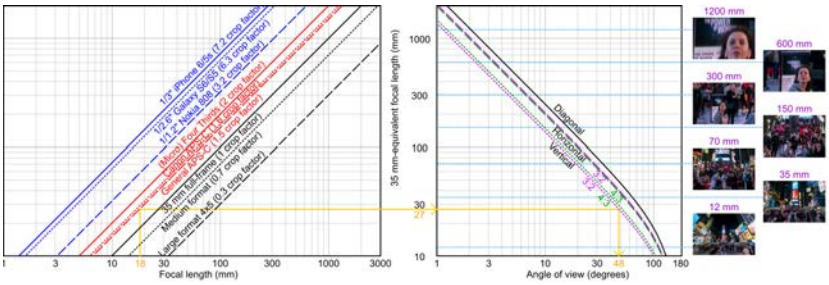


Figure 3.14: Example of different FOV (right image) defined with the diagonal or one side of the sensors size (left image) depending on the focal length of the objective. In yellow: A 3:2 APS-C format sensor coupled with an objective with 18 mm for focal length is equivalent to a 35 mm format sensor coupled with an objective of 27 mm focal length and shows 48 degrees for the vertical FOV. *Image authorship: cmglee, chensiyuan, Wikimedia Commons. Used under Creative Commons Attribution-Share Alike 4.0 International license. No changes were performed.*

In general, though, recording the complete FOV is not necessary. For example, in the case of the microstructures I was commenting on, it may happen that a higher magnification we cannot observe the complete structure, but at a lower, there is more visible area than needed. In these cases, one should record only the interesting region. To this area we refer as the region of interest or *ROI*. Thus, the ROI will be an important constant parameter to define the size of each video. Of course, if the ROI is not defined by the user, the computer will take the maximum area available by our FOV as the ROI, but one should avoid this fact because it enlarges video size. For instance, in Fig. 3.1B) we observed the difference between choosing all the available space or just choosing a little area. For the microscope I used, a ROI of 2048x2048 px would weight 64 times more than one of 256x256 px, which may be a problem if we need to record many videos, especially if they have many frames. Therefore, as a general recommendation, before enlarging the ROI it is preferable to reduce the magnification of the objective if what we want to observe is still visible and trackable by reducing the magnification.

If we cannot reduce the magnification and we do not want to enlarge the ROI, an alternative option is to develop some technique in which the platform of the microscope moves as the video is being recorded following a particle[129]. However, this solution is not usually commercially available and requires a good know-how to implement it. As a result, they are typically found in a few labs. For instance, I could not have this setup during the development of my thesis. Nonetheless, reducing the ROI if possible, it is a good idea, but it should never be extremely reduced because some phenomena can be masked. For example, sometimes when one wants to study the individual behaviour of active particles, a small ROI is selected around a single particle. If the ROI is small enough, we could mask the effect of drift, which is commonly observed under the microscope.

Drift is one of the main problems regarding motion analysis of active particles as the key point when analysing its motion is to distinguish whether if what we have is active motion or its motion just comes from advection [130, 131]. To have active motion means that the particles are producing their own motion, which is not the case when they are being dragged by the currents of the surrounding medium. For this reason, it should be checked if the overall population of particles is moving in the same direction or in random directions as reflected in Fig 3.15A. The first case would indicate that there is drift in the system, while the former will be a sign that there is not. But if the ROI is small enough and only one particle is observed, without proper analysis, a particle under drift could seem that is active.

To test if a recorded particle is moving under drift, we can analyse the MSD since the MSD should follow the ABP model with a small variation:

$$\begin{aligned}\dot{x}(t) &= v_p \cos(\theta(t)) + \sqrt{2D_t}\xi_x(t) + v_d \cos(\theta_d),\\ \dot{y}(t) &= v_p \sin(\theta(t)) + \sqrt{2D_t}\xi_y(t) + v_d \sin(\theta_d),\\ \dot{\theta}(t) &= \sqrt{2D_R}\xi_\theta(t),\end{aligned} \tag{3.3.1}$$

where here v_d refers to the speed of the particle due to drift and θ_d is its drifting angle. Because of the drift, the new formula for the MSD is the same as in Eq. 3.5 but with the addition of an extra quadratic term:

$$\mathrm{MSD}(t) = 4D_t t + 2\frac{v^2}{D_R^2}\left(D_R t + e^{-D_R t} - 1\right) + v_d^2 t^2 = \mathrm{MSD}_{\text{No drift}} + v_d^2 t^2. \tag{3.3.2}$$

The extra quadratic term that appears in Eq. 3.3.2 implies that, unlike in the scenario without drifting, at times longer than the rotational diffusion time, the MSD will look quadratic instead of linear with an enhanced diffusion. As an example, we present in Fig. 3.15 experimental data of one inactive silica micro-particle of $R = 1$ µm where drift was induced. In Fig. 3.15B we observe that the direction of this particle is constant in time. When the MSD is plotted (see Fig. 3.15C), its curve fits very well with a parabolic function even at large times compared with its

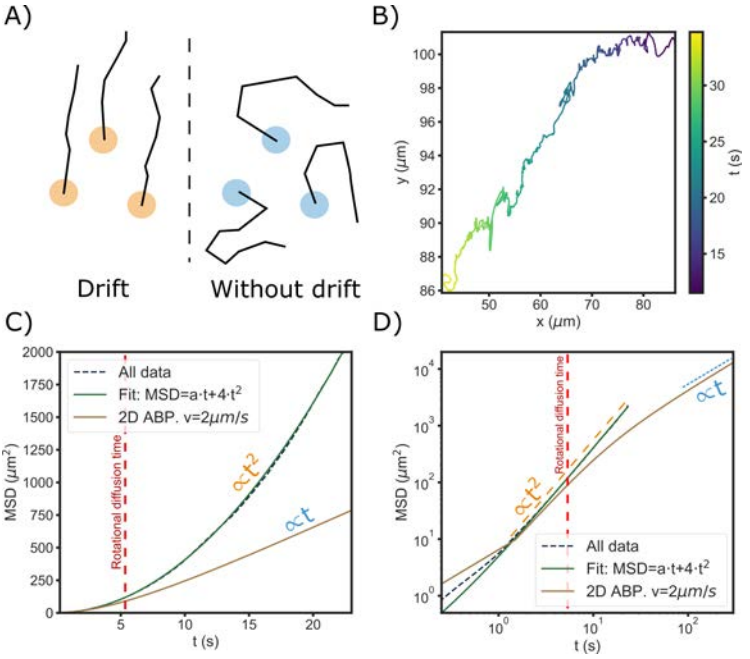


Figure 3.15: Particles with drift. A) Particles that experiment drift move all in the same direction (left) while particles not affected by drift experiment movement in different directions (right). **B)** Trajectory of a recorded passive particle of $R = 1$ µm undergoing drift. **C)** MSD plot of the particle recorded and presented in B) (dashed and green lines) and the theoretical equation of an active particle (brown) following Eq. 3.5 with the same speed as calculated with the drift case. **D)** Log-log of C) plot. The log-log view can help us also distinguishing scenarios between propulsive and Brownian easier.

rotational diffusion time. However, comparing it with a simulated active particle of the same size, $v_p = v_d$ and $v_d = 0$, we observe that after the rotational diffusion time, the MSD of the simulated active particle stops being quadratic and grows linearly, as the limits of Eq. 3.5 indicate. Following a quadratic tendency after the rotational diffusion time can be a clear indication of a drifting-like behaviour in an active particle system.

To check the effects of drifting in the motion of micro- and nano-motors, the long-term behaviour of the MSD should be analysed as indicated in the previous paragraph. An estimation of the rotational diffusion time, τ_r, should already be known. Then, a trajectory much longer than this time should be obtained, and the MSD calculated. By plotting the MSD in a double log-log scale, we should observe a change in slope at the rotational diffusion time, as in Fig. 3.15D. In this figure, the simulated active particle follows a quadratic tendency during the first seconds, before reaching the rotational diffusion time, and its slope decreases to 1 after a longer time. In general, doing a log-log plot is a good approach to evaluate the presence of drift in the system, as well as to investigate different motion from the MSD of micro- or nano-motors that move with sub- or super-diffusive dynamics not collected in any of our previous examples [122, 132]. Moreover, it is also important to notice that when a particle is showing drift-like behaviour, the error of the MSD at large lags will be small, even at times compared to the actual trajectory length. This is due to the high directionality of the motion, which is deterministic, compared to when it is being dominated by Brownian fluctuations. In that case, the full MSD should be smooth and highly parabolic at long times, meanwhile the one of an active particle without drift would show non-uniform or sharp variations in its shape.

Finding the drift by recording long videos could be an incorrect strategy because of the length and size of the final videos. Hence, it could be useful to record only a couple of videos during the experiment where it is observable the regime where $t \gg \tau_r$. But even though, if you can record more than a particle in a ROI, it will be better for your analysis, especially if there are the tracers to justify that one does not have drift in the system. For instance, colloidal inactive particles that are visually different from the active can actuate as tracers that visually help you to understand if we have or not drift in the system without a long analysis afterwards. Hence, consider adding them if possible.

Finally, as a last remark, the directionality of particles towards one point can also be possible because of a chemotactic behaviour of the particles. However, the chemotactic behaviour is a very specific scenario that would require an intelligent particle creation, while drifting can be common when doing experiments. Thus, one should think in drifting before chemotactic behaviours.

3.4 | Saving bytes: From raw to compressed data

In the previous sections I presented a methodology to adequate and reduce the amount of data generated when recording videos. Although these solutions can be helpful while performing the experiments and can reduce the amount of time invested in the experiments, at the moment of analysis we need videos, and our videos do not have to be in raw but can be compressed.

When one records with a digital camera, a raw video or picture is not usually given. Typically, the image is post-processed to reduce the amount of data and a compressed image is given. When the users use the ".tiff" format, normally a lossless data compression is performed. This is, the size of the image is reduced by different algorithms, but no detail is lost. However, this type of compression is not usually used, since it barely drops the size of the file and because many times, a soft lossy compression (or irreversible compression) can reduce the size of the video without any visual important changes. Only fine details are lost, but these are not commonly needed to see well the image. For example, we are well used to see pictures in ".jpeg" format. The "JPEG" name born from the acronym of the organization that defined the standard for compressing the images, the Joint Photographic Experts Group. It is not the porpoise of this thesis to show how these algorithms work, but just to show an example, in Fig. 3.16 I present an image taken by myself with a different jpeg compression ratio applied. Even that one can reduce the size of an image importantly, this does not mean we will not distinguish the objects we have in the video. If this effect is not noticeable for such a complex image, imagine what we can do when working with a black and white picture where we are only focussing on Janus particles.

It is true that the more complex is the algorithm to compress the image, the more powerful a personal computer (PC) needs to be, but our daily PCs are powerful enough to run videos that are highly encoded. Moreover, having a big raw video also needs a powerful PC with huge quantity of random-access memory (RAM) since it needs to move more data. And the worst about all is that any modification realized over the videos will imply that new videos of the same or bigger size will be created. Consequently, the data storage will be more problematic. Due to this reason, and because of the special format I had as an output from the microscope I was using during the realization of this thesis, during the first stages of my thesis I had to develop a software to convert the format and compress these videos. Indeed, by the end of this thesis I finally developed a second more powerful software.

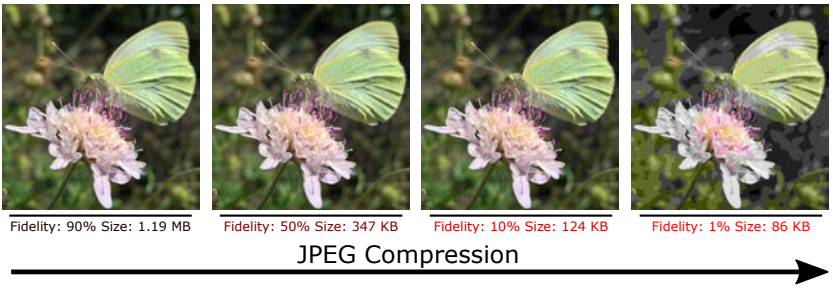


Figure 3.16: An image of a butterfly is saved with the jpeg compressor method at different fidelities. The lower the fidelity, the lower the quality and the size of the image. Even though one can lose image quality, the difference between a 90% and 50% are not very appreciable when comparing them visually, but the size of the image is reduced by a factor of 62%.*Image authorship. Own images.*

If any reader has tried to use the software I was using (LAS X, from Leica Microsystems [133]), probably knows very well its own proprietary file system. For those readers that are not used to use this software, I must clarify that any picture or video that one takes with the software does not create a picture or video itself. Instead, a unique file per project is created as described in Fig. 3.17. This special format is a container, where in the first bytes there is the metadata of the pictures and videos recorded in an extensible markup language (XML) format followed by all the pictures and videos in raw.

Nonetheless, we need to access to these videos and images, and therefore we need to extract them from the container. At the moment I started the thesis, LASX had a method for extracting the data, but the user had to extract videos and images one per one and had to wait until the process ended to save the next one. Moreover, while this was processed, a different user could not use the microscope and consequently, many hours of recording videos could be lost by the team. Furthermore, the extracting method from the software had problems when extracting big videos, which was the kind of sizes we could have in our dataset.

To avoid these problems, and to compress data, I developed the tools Lif2Avi and Lif2Mkv.

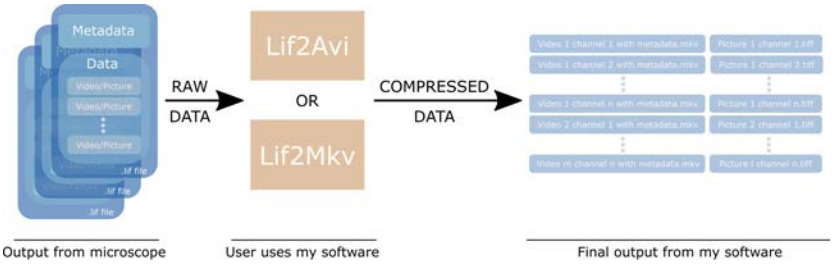


Figure 3.17: Usage of my software. When a user records videos with the microscope, a .lif file which contains all the videos, pictures and metadata is created. Data inside this file is saved in raw, but to export all files to videos or images is tedious. Just opening them with my plug-in Lif2Avi for Fiji or my program Lif2Mkv is enough for the user to have the data extracted from the lif file. Data is compressed by reducing pixel depth to 8 bits and a lossy compression is applied. Metadata for each video is also stored in a new XML file in the case of using Lif2Avi or in each of the own videos in the case of using Lif2Mkv. If channels are present in the videos/pictures, a new file will be created per channel.

3.4.1 Lif2Avi software

The first tool I created was a plug-in for the image software Fiji[134], which is based on the software ImageJ[135]. The reason to do this plug-in was because there was already a plug-in for Fiji called BioFormats[136] that could open images and videos from Leica's files inside Fiji. I also asked support to Leica, to whom I thank for sharing with me a minimal code to open their files, but the process seemed to me shorter and easier by using the previous plug-in. Therefore, I developed a plug-in for Fiji in Java since Fiji is developed in Java. In Fig. 3.18 I present the different sections of the plug-in developed. Since I was using Fiji, and Fiji has many options, I decided to use in my plug-in some of the already implemented options. For instance, the user can add a scale bar and stamp the time in each frame for each video. Both options come as independent methods in Fiji with its own properties, and therefore these properties could also be modified in my plug-in. If a user always uses the same configuration, this configuration can be saved and loaded from the main tab ("General") directly each time it opens the plug-in. Since in the lab we could record many .lif files, instead of doing this action per .lif file, I also added the bulk option to let users indicate to the software the files they want to convert and do the process automatically one after another one. Data needed as FPS, magnification or binning is selected individually per video after reading its own metadata information, but user can establish a different one if needed.

The action of saving is carried by the own Fiji software itself and videos compressed in jpeg format are created with avi extension. Nevertheless, it is possible that an alternative tool must be used. This alternative tool is FFmpeg soft-

ware[137], which is an open source software for video record, convert and do streaming of video and audio. The main reason to use FFmpeg is because when Fiji opens an image or video it has two modes for opening. On the one side, it can open the data in a "virtual" mode. In this scenario, the user can open big files, but it cannot modify the image or video. If anything is performed, the action will be lost after changing from one frame to another one. Thus, if the options for adding the time stamp or the scale bar are activated, they will not work. To use them, the user must open the files in the non-virtual mode, which essentially, it loads all the images or videos in the RAM. Thus, if the video is long enough, since it is a raw video, it will consume a lot of memory and the user will not be able to extract the video. Instead, what the software does is to load in virtual mode the video and then, split the video in sub-videos and convert one of them from virtual to non-virtual mode. Once it finishes to apply the extra options and saves this sub-video, it will perform the same action for the rest of the sub-videos. Finally, when all of them are converted, the software will call the FFmpeg software to fuse all the sub-videos in just one final video. This action can be avoided if the user unchecks the "Divide videos" option in the main tab, but it is checked by default.

As an example of its performance, this plug-in could extract data from a 29.6 GB .lif file in 15 min and reduce the total video size of all the videos present in the .lif file to 321 MB. This is a 98.94% less of data storage. If you want to obtain and install this software, please read appendix C.1.1.

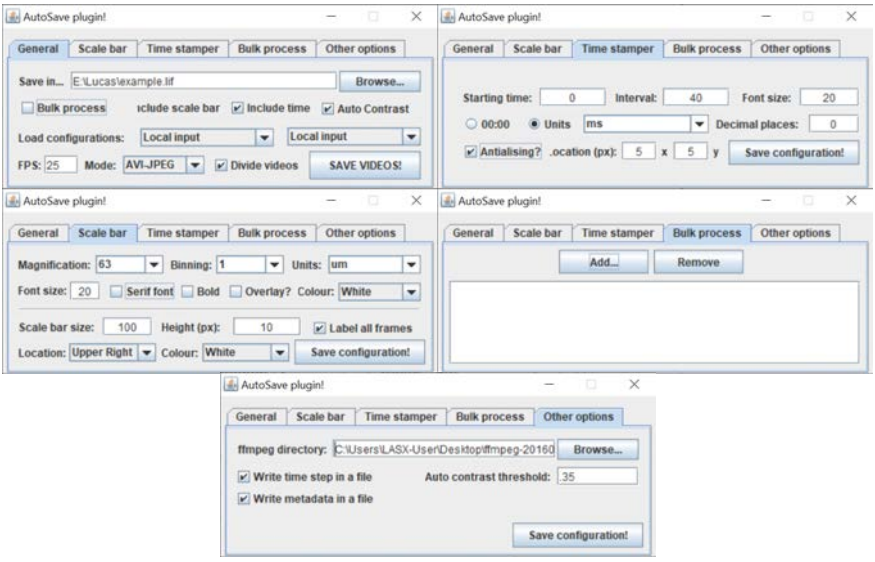


Figure 3.18: Lif2Avi software. This little plug-in for Fiji/ImageJ lets the user to extract all the videos and images from the .lif files. User can add scale bar and stamp the time in each frame if needed.

3.4.2 Lif2Mkv software

Although the Lif2Avi software works well, there were some drawbacks that decide me to create a second software, which I called Lif2Mkv. The first and main reason to develop this new software was because for some .lif files, the videos were blinking. I never understood totally why this problem was appearing, but the only reason I could find is that the inner Fiji was performing a contrast adjust for each individual frame. Thus, I tried to add an option called "Auto contrast" to perform an auto contrast calibration for each frame, but the problem was still appearing. Moreover, the time needed to extract the videos was long, and the metadata was lost for each video, since it was all written in just a single XML file.

To avoid these problems, I decided to start a new application in Java since I was still using Bioformat software. This time though, I created a Maven project [138], where the source code from Bioformat and FFmpeg are embedded into the project. Hence, the final user does not need to download these software programs by itself and I can call the methods present in these libraries internally in my code. In Fig. 3.19 I present this software. In this case, it is much easier for the user to select the videos that wants to extract from the beginning, although I did not include the options for time and scale bar stamp because I lost these functionalities not working with Fiji and because no one in the group was using them anymore. In any case, it should not be very difficult to add them into the project.

Using this software, any user can see all the files from each .lif project, and select which wants to extract by clicking the checkbox associated with the video. Furthermore, it can change the output route for all the project or individually. When the user press “Extract videos”, the videos are extracted, and a progress bar shows in blue the progress of the extraction. When all the video is extracted, the progress bar will be full and green. Videos with different channels can also be used with this software. The software will create individual videos per each channel, which the user can change their name by clicking at the button "Channels...". Quality for compressing the videos or Z-stack images is easily to change too introducing the proper number in the text fields.

The only final drawback for the user when using this software is that these videos cannot be opened in Fiji, if needed, since Fiji cannot open videos in mp4 format. There is also a limitation for PowerPoint software, but I created a similar software that changes the encoder of the videos to open them with PowerPoint.

Internally, the software works as follows. Bioformats opens each frame from each video or image, and then the frame is transferred to an object from the FFmpeg library. If we treat a video, the frame is written in a .mkv file using a mp4 encoder. If the image is from a Z-stack, it will be saved in a .tiff image. In the case of the videos, I decided to use a .mkv extension because it is a container, where it saves not only the video but other useful information. In my case it allowed to save any kind of metadata, and thus, each video file contain the details of creation

during the recording with LASX. For instance, details such as the objective or light intensity are stored by video. In the case of the Z-files I was not able to add this information, but an xml file is created with all this information.

The processing of the files is faster, no blinking effect is observed, and the compression is better since I am using mp4 format for the videos. As an example of its performance, this software could extract data from a 29.6 GB .lif file in 5 min and reduce the total video size of all the videos present in the .lif file to 67.2 MB. This is a 99.78% less of data storage. If you want to obtain and install this software, please read appendix C.1.2.

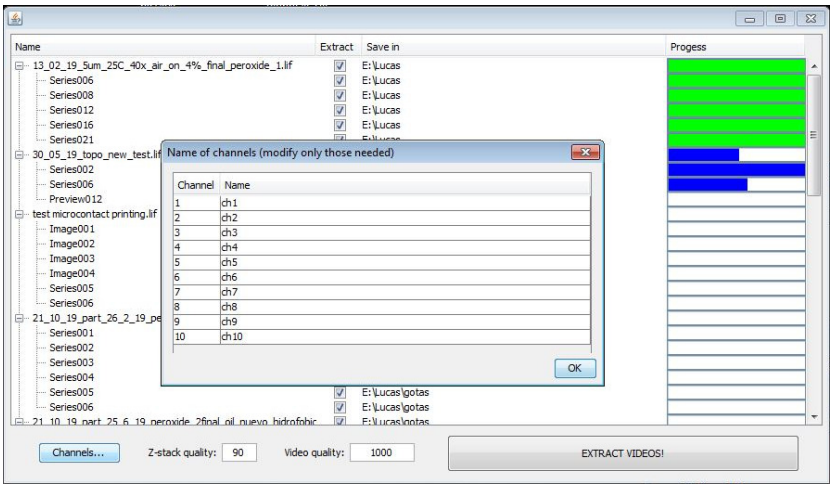


Figure 3.19: Lif2Mkv software. This little software lets the user to extract all the videos and images from the .lif files faster and with a better compression ratio than with Lif2Avi software.

3.5 | Conclusions

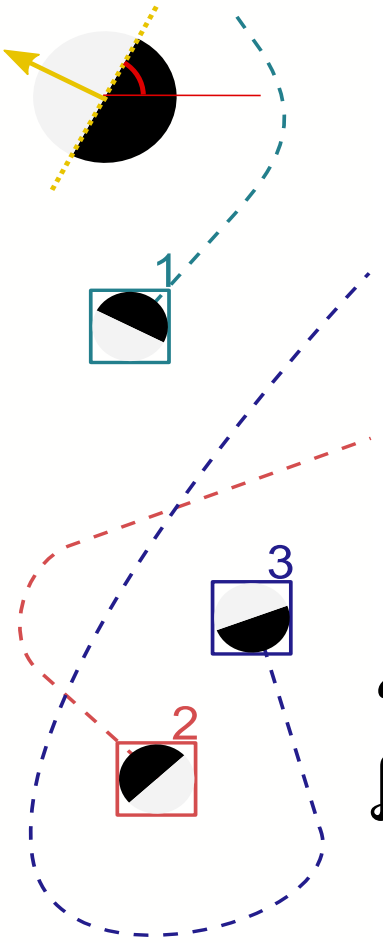
In this chapter we presented several useful tools when recording videos with a microscope. Some of these tools are consequence of the recording software used during my thesis, but others were born to improve the precision in the calculation of some particle's properties such as its active speed and rotational diffusion time.

First, the video basic knowledge and the programming of the couple of software tools provided allows any user to decide if their videos should be compressed. Moreover, if the user uses a LEICA microscope, they directly allow to compress their videos. Nonetheless, for other users this information can caveat the path to do an analogue of the present codes.

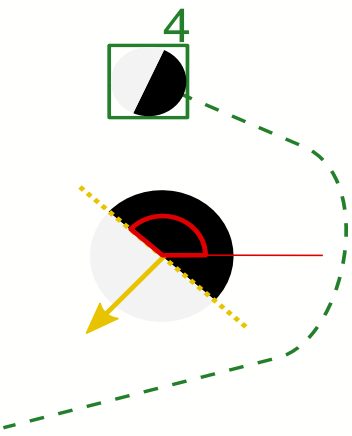
Second, this chapter advises any user on the ROI, length, and FPS to use when analyse the velocity or rotational diffusion time of active particles. While setting the ROI is important to avoid drifting, the lengths and FPS are important to obtain a good value of particles properties. In any case, if the user cannot select a proper ROI, this chapter have aid on how to notice if a particle has drifting, i.e., if the MSD at longer times than the rotational diffusion time grows quadratically with time.

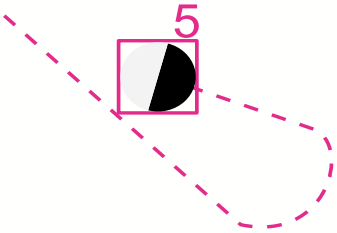
At the same time, we gave indications to improve the results obtained using equation . For particles smaller than $R = 0.5\mu$m, a linear fit gives D_{eh}, which is the value to look for. For bigger particles, we can obtain speed and D_t. D_t should be avoided for particles bigger than 1 μm due to its large error when extracting it. In any case, we obtain better results for both parameters if we use the third order approximation $n = 3$.

The present analysis also presents three different methods to experimentally calculate the rotational diffusion time τ_r. If possible, we should avoid using the angle taken from its instantaneous speed to calculate τ_r, and indicate how the average was done.



Using neural networks for tracking of particles





"Any sufficiently advanced technology is indistinguishable from magic."

Arthur C. Clarke

If any of us ask a physicist what would analyse from a video full of particles, probably, the first they will answer is their position, or at least, something that needs to know the position of the particles. In our tiny world, the only option to know where the particles are, is to *detect* and to *track* them from the videos we have recorded. To detect a particle means that we know which is the centre of mass of the specific particle in our setup. To track a particle means that for each frame of the video, we are not only able to detect the same particle, but to give the same identity and to do not lose this identity in the next frames. Although this can be easy to say, the reality is that depending in your system it could be complex to obtain a good result. There are plenty of software programs and methods that brag about their good detection and tracking, but this problem is very specific for what one wants to detect. Some rules can in general be followed, but there are situations where these are not enough for the problems we have, and we need new solutions. During the realization of this thesis, I could use from the begging a software developed in the experimental group by Albert Miguel López. However, this software was insufficient when I embraced new projects. For instance, in chapter 7 I could not analyse any of the videos I recorded properly because I could not track these particles as I wanted. The results given in that chapter would not have been possible if I would not invest my effort in building a new platform for tracking particles. In this chapter I will show what I had for tracking particles, and what I did to solve the drawbacks I was experiencing.

The present chapter is based on the software I coded for chapters 5 and 7. I would like to special thank Albert Miguel, who provide me in my first stages his software to track my experiments. I would also like to thank Xavier Arquè, who provide me with the videos for MOF tracking. Finally, I would also thank the GitHub users AlexeyAB and YashasSamaga. The first, because he provided the YOLO package I used and answered me some questions related with it. The second, because without his effort on time I could not speed up my software using GPU processing for image detection using OpenCV library.

4.1 | A traditional tracking

Tracking of objects is a research field with big interest since decades ago[139, 140]. There are different methods to track objects, but the basis of any tracking system is to have a good detection system.

Traditionally, to detect objects one can play with changes in the images, or with patterns. As we already explained, an image is a collection of pixels. From the distance, one usually does not see these pixels, but sees a continuous image. But when one zooms in, the continuous image starts to be discontinuous. Thus, when from the distance one sees a change in colour in the image (e.g., due to different illumination, border of an object...), from the pixel world this change can be very sharp. And since pixels are anything more than numbers, one can always use different mathematical formulas to apply to these numbers. Hence, it is not surprising to think that we can calculate derivatives in different directions, or average pixel intensities.

These mathematical transformations can be realized directly to the pixels, but it is also common to calculate the Fourier transform of the image, operate in the transformed image and then do the inverse Fourier transform[141]. By applying intelligently these operations, one can detect objects in images. For instance, it is well-known the Viola-Jones detector to detect faces[142], the Histogram of Oriented Gradients (HOG) to detect pedestrians[143] or much simpler, the Circle Hough Transform (CHT) to detect circles in the images[144].

These transformations have been used widely in the last years, and they have evolved to detect every time better different objects. However, in all the cases, these algorithms do not know what the images we are analysing are, neither what we want to detect. Instead, they just use the parameters we feed them to any image we want to process. Therefore, the detection of objects performed with these algorithms are very user and parameter dependent.

Luckily, new advances in image recognition have changed this paradigm. There are a new set of algorithms based on neural networks that can learn by themselves what we have in images, what we want to detect, and which parameters are better to use for each image we provide them. Thus, I will differentiate detectors into traditional detectors, explained in this section, from detectors based in neural network, explained in section 4.2.

4.1.1 Particle position detection

In the case of the available tracking software implemented by Albert[145], the particle detection is very simple, yet it works really well with circular particles that are alone, and under no changes in contrast in the image. To detect particles, he first transforms the image into a *binarized image*, this is, an image where pixels can either be 0 (black) or 255 (white). To obtain the binarized image, he uses a threshold: for each pixel, if its intensity is higher than the value, we will return white, or black if it is below. Before this method is applied, there are a couple of important filters he introduced that the user could use:

- Sobel filter. It calculates both vertical and horizontal derivatives to each pixel by operating on the pixel itself and its neighbours. If two pixels have similar information, the derivative will be zero. But if two pixels have different information (such in a border of an object) the derivative will be large. Therefore, this filter returns an image where basically, only pixels that creates borders have high intensities and the rest are near zero.

- Background extraction filter. It subtracts a referenced imaged to the image at frame t. The reference image can be built by averaging all the frames in the video or taking the first frame of the video for example. Hence, areas with no change in the image will be close to zero, while areas with high dynamism will not. Thus, we remove constant backgrounds, and we only have things under motion.

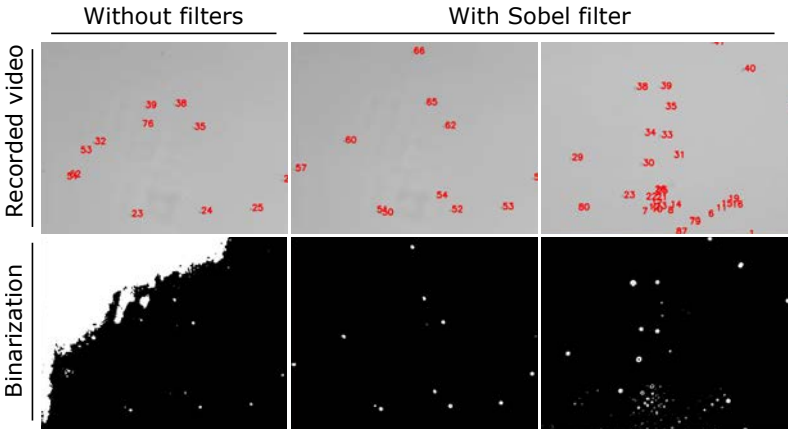


Figure 4.1: Detection of 5.0 μm diameter Pt-SiO_2 particles under at 20x objective. We can detect particles by converting the images to a binarized image and then calculate centre of mass of objects. This method can give problems since of different contrast and hence we can apply previously other filters as the Sobel filter. Nonetheless, for the same frame there can be other problems as appear in the last column, which was taken from the same frame as the other pictures.

Once binarized, he analyses the pixels in white and calculates the centre of mass per each continuous white pixel area. The centre of mass will be the centre of each of the objects detected. To better detect these areas, other filters such as a gaussian filter which smooths the image can be used, and only areas in the range of interest can be detected. These simple methods are enough to detect particles as demonstrated with Fig. 4.1.

4.1.2 Particle orientation detection

Albert's tracking code also allows us to calculate the particles orientation. In this case, he tried different strategies, but only kept the one with the finest result. For each time step and particle, his software draws a line of the size inputted by the user, which should be of the optical size of the particle in the video analysed. This line crosses along all the particle, and its centre is placed on the particle centre already calculated by the centre of mass. Then, a standard deviation of the pixels' values along the line is calculated. This process is repeated for 180º, using a line per degree. If the line is not on the line separating both faces, the deviation will be large, because some pixels will be black (0 intensity) and some will be white (255 intensity). But if the line is on the line that separates both faces, then all the intensity values should be close, and hence the deviation will be low.

The lowest value will define the best face-face separation and, therefore, a perpendicular line to this line will define the director vector for our particles. Thus, to correctly apply this method we need a good image of the Janus, which means that the highest the size and magnification, the better results we will obtain. If the particle size is not big, the lines we will draw will be small, and hence the standard deviation method will not be valid any longer. In these cases, we can estimate the angle from the instantaneous velocity vector, which his code can also calculate, but that as we already commented in section 3.2.2, if possible we should avoid this method to calculate the orientation of the particle due to low precision.

4.1.3 Particle orientation detection from a different frame of reference

The previous method can detect the angle of a Janus particle, but it is always referenced from the same axis, e.g., 0º is parallel to each row of pixels in an image. However, sometimes the frame of reference can be different, and we could want to obtain different angles. For instance, in chapter 5 I needed to calculate a couple of angles defined by particles and liquid-liquid interfaces. In that problem, each video could have one interface, but they were randomly oriented. Hence, I needed a method to define this frame.

To calculate this reference per video, I coded a software using Java where the user must introduce the recorded video files and the tracked positions and angles after using Albert's code (See Fig. 4.2A)). Then, for each video I marked with a straight line the visible line that defines the interface between both liquids (See Fig. 4.2B) red line). This line is easily defined by clicking twice at the interface at different points. In this project, a straight line was sufficient to capture the geometry of the interface at the magnifications used since we did not observe any curvature of the interface. Once we have this reference line, we can calculate the orientation of the particles with respect the interface while they move along the interface (See Fig. 4.2B) left, red arrow).

A second orientation I needed to calculate was the angle between particles trajectories and the interface while they were approaching or leaving the interface.

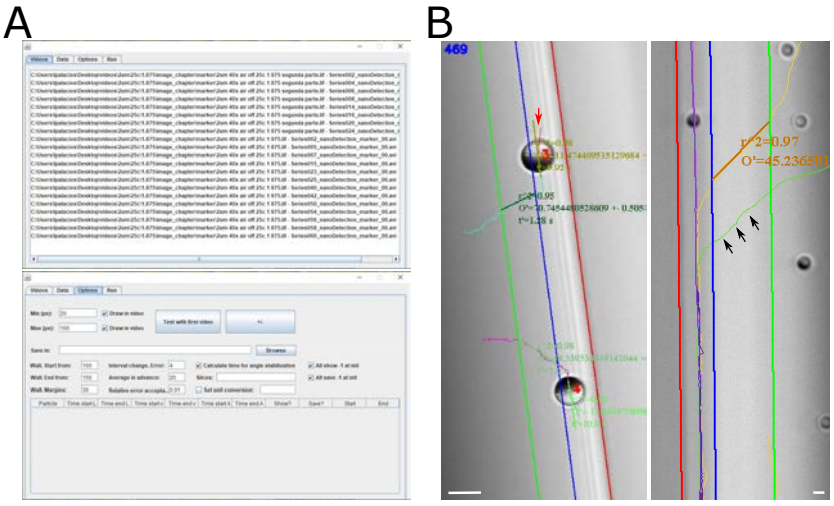


Figure 4.2: Tracking orientations from a different frame of reference. A) The software needs a list of the videos (Videos tab) and the csv containing the trajectories of the particles at these videos generated by Albert's code (Data tab). At Options tab, the user can modify a few parameters to calculate the areas where different angles will be calculated. **B)** When the user presses the Run tab, a video opens, and the user can click twice in the video. After the second click, a red line is drawn. This line will be the new reference line we will use for our calculus. In the case of the project at chapter 5, this line will match the liquid-liquid interface. Once drawn, two lines (blue and green) are drawn parallel to the red line. We will use data in between red and blue lines to calculate the average orientation of the particle while moving along the interface with respect the interface (red arrow, left image). We will use data in between blue and green lines to fit the trajectory of the particle into a line since its trajectory is mainly linear as seen with the black arrows (right image).

Even though these particles move actively Brownian, we can define a characteristic length at which particles move straight before turning due to rotational Brownian motion. This length is defined as the persistent length L_p, and it is the product of the particle propulsive speed v_p and its rotational Brownian time τ_r:

$$L_p = v_p \tau_r. \tag{4.1.1}$$

For particles with a diameter size of 2 μm and 5μm, and speeds between 1 and 15 μm/s, L_p is in the range of [5.4, 81.0] μm and [84.9, 1273.5] μm respectively using the theoretical values for τ_r. However, we know from section 3.2.2 that τ_r will be lower. Nonetheless, even if we consider a real τ_r of half the theoretical one, these L_p extends at least for a region of a few particle radii. Thus, we can define a rectangular ROI parallel to the interface contact line of 3-5 particle radii width. To define this ROI, we click twice on the image, one per line drawn, and two parallel lines to the interface contact line will appear (See Fig. 4.2B) blue and green lines). The region inside these two lines is our ROI.

For each of the particles found in this ROI, a linear fit of its trajectory was performed, and the director vector of each line was obtained, both for the incoming and outgoing particle to/from the liquid-liquid interface. A simple scalar product between these director vectors and the vector defining the interface line gave us the angle of incoming/outgoing. Since particles move straight in these regions, we can also compute an average velocity as a ratio of the displacement over the time interval.

4.1.4 Tracking of particles

We can calculate some of the previous angles because we already tracked our particles. To track particles, you need to stablish a unique identity per each particle detected, and do not lose it over time.

Albert's code compare positions of particles between two consecutive frames. For each particle in a frame, he finds the smallest distance to a particle in the previous frame and supposes that both particles are the same if this distance does not exceed a threshold that the user can change. But a particle present in the previous frame could be not detected in the next frame, even if it is present. Hence, he also introduced the possibility to find particles that did not encounter any match yet since frames ago. If particles are well detected, and do not cross over, this simple rule can work well. But as soon as the detection is poor, even with this last addition the tracking will not work, and we will lose the possibility to track.

Another problem that his tracking has is when particles contact. When contacting, the software only detects one particle, and therefore we will lose the identity of at least one. Consequently, we will need a better approach.

4.1.5 Tracking of particles from a different frame of reference

Sometimes, we might want to track particles from a different frame of reference. For example, as a step further of the research I did in chapter 5, I needed to track the relative motion of active particles that were contacting moving oil droplets.

To solve this issue, I decided to track first the droplets, and then to track the particles. But in between, I modified the videos to centre them at the droplet frame of reference (See Fig. 4.3A)). To centre the videos we must do the following transformation for each time step t and pixel (x, y):

$$(x, y) \rightarrow (w/2+max\{|w/2-x_d| : x_d \in X\}), h/2+max\{|h/2-y_d| : y_d \in Y\}), \tag{4.1.2}$$

where w, h are the original width and height of the video, max is the maximum value and X, Y are the collection of points that compose the droplet trajectory. Then, we only need to track the particles with these new videos[1].

Because the video is centred at the droplet, we can calculate extra parameters for the particles such as the incoming and outgoing angle to/from the droplet and calculate the travelled length and number of turns made by the particle around the droplet and the total time for the particle contacting the droplet (See Fig. 4.3B)).

In section C.2.1 I share the code to shift the frames of the video.

[1] After the transformation, videos will have a different size: $(w + 2max(|w/2 - x|), h + 2max(|h/2 - y|))$.

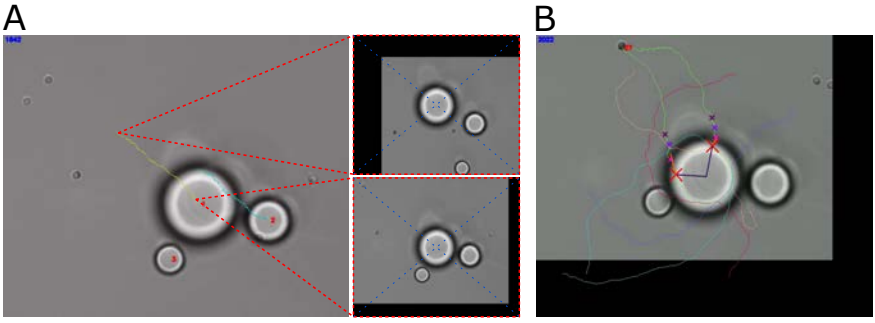


Figure 4.3: Tracking particles from a different frame of reference. A) After tracking oil droplets we can centre the video at the tracked trajectory. Each colour line is a different tracked droplet. Small images refer to different frames of the final video. **B)** Particles are tracked with the previous video (each colour line is a different tracked particle). Red crosses are the points where particle contact and leave the particle. Orange lines show the incoming/outgoing angle for particles as I showed in section 4.1.3

4.1.6 Tracking of particles from a different system of coordinates

When we track particles, we obtain its trajectory in the (x, y) pixel coordinates. However, depending on the porpoise of the project we could need a different system coordinate. For example, in chapter 7 I discarded using the (x,y) pixel coordinates because to understand the physics of my problem, these coordinates only added noise into my analysis. Instead, I needed to map the structure where the particles were moving (See Figs. 4.4A) and E)) into different cells, and then transform from the (x,y) coordinates into these cells coordinates, as if particles would be moving on a chessboard.

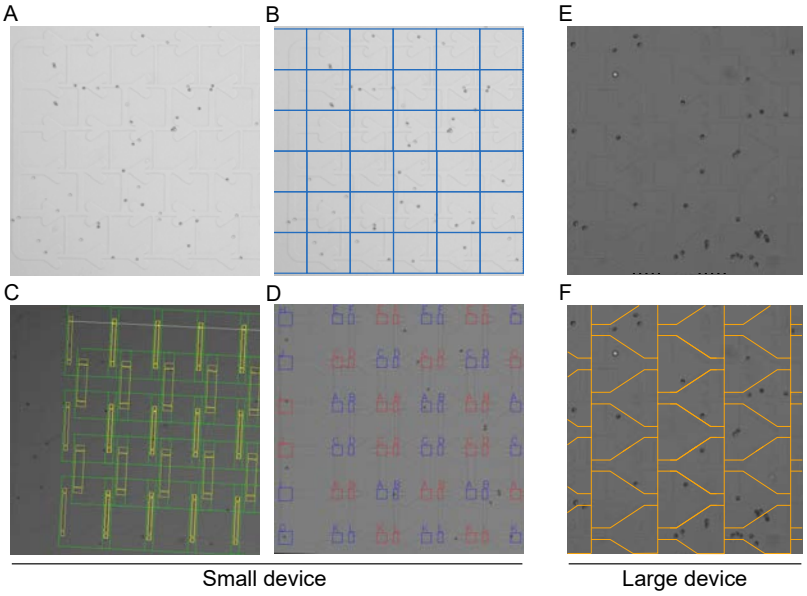


Figure 4.4: From Cartesian coordinates to Cell coordinates. A) First device to build the cells. **B)** Final cells coordinates for device presented in A). **C)** and **D)** alternative Cell coordinates for device presented in A). **E)** Second device to build the cells. **F)** Final Cell coordinates for the device presented in E).

To transform these coordinates, we need to define the new coordinates in the videos. All videos will have the same definition of cells, which will be constant over time. However, videos can have the structure shift, rotated, or stretched. Thus, after defining the cell structure we need to adapt the code for all the videos. This might require adding to the code mouse and keyboard events. In the case of the first design I used, the definition of the cells was a chessboard pattern, as seen in Fig. 4.4B), which is very easy to implement. Nonetheless, we previously tried different strategies such as the ones presented in Figs. 4.4C) and D). But cell structures can be more complex, as the one we defined for the second design presented in Fig. 4.4F).

To define these cells we need to know the vertexes of each cell. If we work with Python, we can use the OPENCV library to open the videos and obtain these vertexes by clicking on the image and saving the position of the mouse when the user clicked. Then, we can transform these vertexes into a Shapely polygon. Shapely is a library for geometry manipulation. It let the user to define objects (polygons) and check if a coordinate is inside the object or not. Thus, once we define the cells as shapely objects, we can check to which object belongs each particle coordinate and obtain the transformed trajectory.

In section C.2.2 I share the code to show how to do this step. The example includes code to move a cell over the image using the keyboard. For example, if you press "w" once, all the vertexes of the cell could move one pixel up, and hence your cell has been displaced one pixel vertically. This edition can be implemented by individual cells or also by rows and columns as I did when working in my project.

4.2 | Basics of neural networks

Under optimal conditions, the simple rules Albert applied can fit our necessities. But when things start to be more complex or simply, they are not under optimal conditions, the above methods can be insufficient and new tools are needed. For example, in Fig. 4.1 we can observe that although the system is simple, particles are detected at someplace, but the light conditions are not homogenous and fails to detect correctly in other regions. If the user tries to track a particle, probably this will not be a problem if the particle does not escape from the region where it is well detected. But if the user wants to track many particles, it may be impossible to take a set of values for the detection parameters to detect everything. Furthermore, this problem increases if by default, the distinction of what a particle is rely on only on a few pixels. As the user would understand, it is not the same to consider that a square of 50x50 px^2 is a particle, rather than a square of 2x2 px^2. In the first case you have more data, and you could distinguish the particle from the noise while in the second, any grain of noise in the image will cause a detection or difficult the ones you want to detect.

In the case of the videos I recorded for the experiments carried in chapter 7 I had both problems. To fit all the microstructure, I had to reduce the magnification of the objective, and particles became as small as 9x9 px^2. Then, particles could touch between them and the solid interfaces, and the light was not totally homogenous. These conditions created me new identities over time for the same quantity of particles, and do not allow me to track them. To track these particles, I decided to start a new tracking software based on a different detection: neural networks.

Opposite to the algorithms described in the previous section, the major advantage of using neural networks is that the algorithm learns what kind of object we want to detect, even if the image conditions are different from the ideal condition. Thus, this kind of detection can improve a lot the image detection. However, traditionally, this kind of algorithms have been considered as using a hammer to crack a nut due to the high PC resources needed. But as time passes, hardware also evolves and today there are many problems that in the past were impossible due to PC lack of resources but today are not. Thus, detection of images by neural networks is something we should consider for the experiments of the future, since its results can be surprising and can be really useful for the Active Matter community [146][147].

4.2.1 Silicon neurons and neural networks

In the last years, neural networks have been a new paradigm in computer science, with many fields involved [148, 149, 150] but specially, in the field of image object detection[151]. Its name come after the idea of the biological neuron networks.

Biological neurons are cells which frequently have a cell membrane depolarization[152]. This depolarization can lead to the transmission of an electric potential, which will activate the realising of some chemical molecules, called neurotransmitters, at the end of some specific neuron ramifications, in what is known as synaptic terminals. Neurotransmitters will attach to other close neurons at their dendrites, another kind of neuron ramifications close to their nuclei, via the synaptic link. If these new neurons receive enough neurotransmitters from one or more than one neuron, a new electric potential will be fired in this second neuron, and the process will be repeated in this neuron, proceeding thus to the transmission of the electric pulse, or directly, information. Therefore, biological neurons create networks by their own nature. In the same sense, a silicon analogy can be thought. For example, we can consider that the neuron is an abstract entity that does a mathematical calculus. As in the biological neuron, the silicon neuron will be connected to others, and their result will be transmitted. These results will be used by the next neuron to do its own calculation, and hence, transmitting or not information. Therefore, since we have connected silicon neurons, by analogy, we call to these networks as neural networks.

Neural networks allow us to create better artificial intelligence which can solve problems that up today were not possible. The key point of all neural network is to find those mathematical calculus and connections to perform the desired action. If we look at how biological neuron works, we can glimpse that the main idea is to overpass a threshold to continue the transmission[152]. Thus, since their result is binary (either there is transmission or not), mathematically they could be described by a simple Heaviside function.

Commonly, these functions are nonlinear because we need to solve nontrivial problems by using few neurons. Examples of nonlinear functions include the hyperbolic tangent, the sigmoid function, the Heaviside function, and the rectified linear unit function (ReLU), which under a threshold the result is zero, but over the threshold is linear. Since these functions are *activating* neurons, as in the biological neurons, we also call them as the *activation functions*. See Fig. 4.5 for a graphical overview.

4.2.2 Layers, weights and learning

In a neural network, not all the neurons must have the same activation function. Since they create a network, there can be a diversity of neurons using different activation functions. Thus, the second important ingredient in neural networks is the interconnection between them. As more than one biological neuron can be close

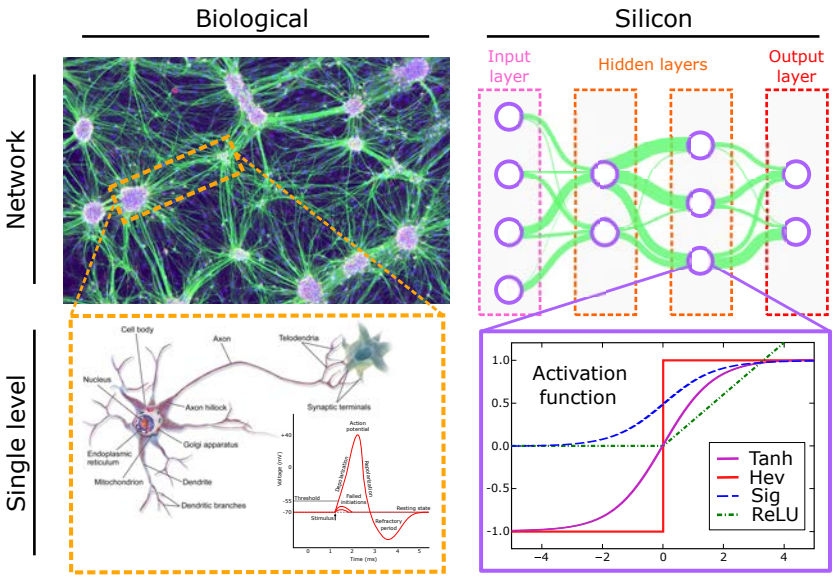


Figure 4.5: Biological and silicon neurons. Biological neurons (Neuron nuclei in purple) are connected to other neurons through their axons (lines in green). At the end of the axons, in the synaptic terminals they can send molecules to the following neurons, the neurotransmitters. These transmitters can produce changes in potential in their cellular membrane. If the change is over a threshold, they create an electric signal, which is also known as the action potential. When a neuron has an action potential, it will release their neurotransmitters to the next neuron. Otherwise, it will not release them. Several neurons can communicate between them creating a network of neurons. At the silicon level, a neuron (purple circle) is a mathematical function, the activation function, which translates the idea of the electric signal. There are different activation functions such as hyperbolic tangent (tanh), Heaviside (Hev), Sigmoid (Sig) or rectified linear unit (ReLU) function. The output of a silicon neuron will be one of the inputs of the following neuron (green lines). Since several neurons can give information to one neuron, these inputs will be modulated by a weight (the wider the green line, the higher the weight). Neurons sharing the same connections create a layer. The first layer is the input layer. The last, the output layer. Any layer in between is a hidden layer. *Image authority. Biological network: Laura Struzyna, Cullen Laboratory, Perelman School of Medicine, University of Pennsylvania, Philadelphia, NIH Image Gallery. Used under Creative Commons Attribution-NonCommercial 2.0 Generic license, cropped. Biological single level neuron: BruceBlaus, Wikimedia Commons. Used under Creative Commons Attribution 3.0 Unported without making any change. Action potential: Chris 73, Wikimedia Commons. Used under Creative Commons Attribution-ShareAlike 3.0 Unported, without making any change.*

between them, in silicon neurons many can connect between them. In fact, with silicon neurons one might consider that they can have more connections than biological neurons since they do not occupy a physical space but are just mathematical entities. However, not all neurons are connected between them. We usually group different neurons that do not connect between them but connect to the same set in what we call a *layer*. In all neural networks there is an *input layer* to which we feed with the data we give to the network, and an *output layer*, which give us the result of all the process performed by the neural network. Nonetheless, between both layers there can be as many layers as we want. To these layers we call them as *hidden layers*[2].

Mathematically, the connection of each neuron is proposed as a weight. If we consider a neuron, different neurons will transfer their outputs to this neuron, and every input will be modulated by their own weight. As a result, we could think that on each neuron we have:

$$output = f(b + \sum_i w_i x_i) \tag{4.2.1}$$

where $f()$ is the activation function, b is a bias we can introduce in each neuron with a known value at a priori in the model and w_i is the weight applied to the input x_i which is given by the i neuron feeding the neuron of interest. See Fig. 4.5 for a graphical overview.

As one can deduce, all parameters are set when one creates the scheme of the neural network. However, there is still an important concept to learn. To explain it, we could start considering that we want to detect a specific object, like a chair. Biologically, when we are born none of us know what a chair is, but with time, the plasticity associated to our neurons let us to create new connections that allows us to say what is a chair, even if it is the first time we see a specific model. For instance, if we enter in a furniture shop, any of us can distinguish what is a chair even though it is the first time we see that model. The nice concept behind silicon neural networks is that this plasticity process also happens. Initially, the network will not detect what we want, because the connections between neurons are not well stablished. But they can learn if we train them. In this case, we do not create new connections, but we modulate the weights between connections. If all the weights between connections are well estimated, the neural network will give incredible results because it will be able to extract the intrinsic characteristics of the object to detect. Thus, a neural network could detect if the photo we give to it is mainly a chair. Once a neural network can differentiate objects, we can build other algorithms to even present an image with many objects and locate, if present, the object of interest in the picture.

[2]If the activation function would be linear, we could simplify all hidden layers to have a neural network with only the input and output layers.

4.2.3 Convolutional Neural Network (ConvNet or CNN)

The previous example of the chair can help us to understand the main idea behind object detection with neural networks, but in fact, for a better comprehension of the problem, apart from considering the training process, it is also important to focus on the layered structure of the network. As I detailed, the output of a layer is the input of the following layer. This means that in the network we are stablishing a hierarchy, where each layer extracts some important features and the next one processes these features to look for higher detail features.

This kind of structure can work because indeed, is how our visual system works as we can see in Fig. 4.6. In 1959, Hubel and Wiesel[153] carried some experiments with cats where they studied their visual primary cortex and saw that the system is hierarchical, with different layers in between and that mainly, two kind of neurons work on the system: ones that they called *simple* because they respond directly to the stimulus, and another ones that they called *complex*, because it was not clear how they respond to stimulus. A couple of years later, in 1979 Kunihiko Fukushima took their conclusions and created the first neural network for image recognition[154]. With his system, which he called the *neocognitron*, he could recognise numbers and some patterns. He based his system also on a hierarchy, combining different layers. Indeed, for him a layer was composed of two kinds of cells, where he could combine the idea of the simple and complex neurons. First, data would transfer to the simple cells, which have plasticity, and where each cell works for a small area of the image. Later, this information would transfer to the complex cells, which do not have plasticity but with the ability of extracting the main features of the output from simple cells. In somehow, we could think that these complex cells do a coarse graining of their input, and hence they remove the sensitive to shift in position of the pattern that the simple cells considered. Thus, the complex cells here simplify the connections of the network, allowing to have less quantity of weights. If we have less weights, we will have less free parameters in the system and hence, it will be more difficult to have overfitting problems, which could return us a bad detector.

After Fukushima's work, different works came, but there was one main drawback: the computational power was low. With the huge development of *graphic processing units (GPU)* in this century, these algorithms gained again big interest. Today, the scheme we follow is very similar to Fukushima's work. In our neural networks there are some layers that have plasticity and hence they are composed of neurons, but there are others that just do the coarse graining, and there can be several in between a neural layer as we can observe in Fig. 4.6. In these last layers what we do is a *mathematical convolution*, and hence, since we include this calculus in the network, to this kind of neural networks we call them as *convolutional neural networks (CNN)*. Thus, if your aim is to detect objects in an image, you will need a CNN.

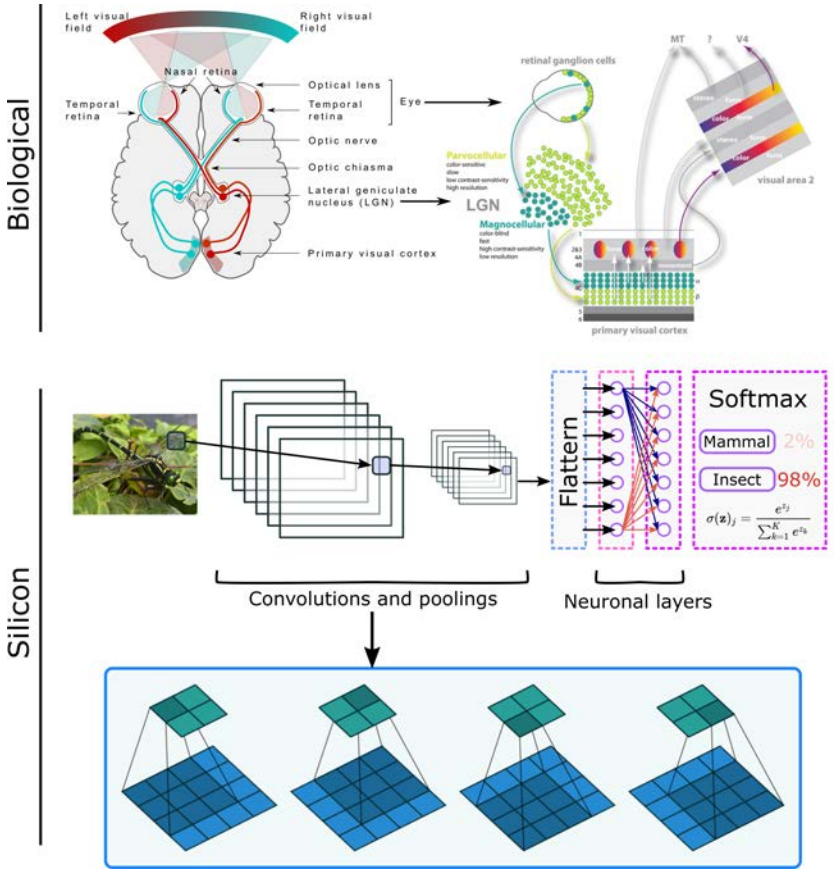


Figure 4.6: Copying the animal visual system. The visual system of animals is a hierarchical system where the information that arrives to the eyes passes through many neurons distributed in layers. In primates, for instance, the information passes from the eyes to the lateral geniculate nucleus (LGN), which connects to the first primary visual cortex (or V1). Information from this area will pass to different areas named after Vx, where x is the number of the area. V5 also receive the name of middle temporal visual area or MT visual area. We can model this hierarchy as a sequence of mathematical functions, where we start doing convolutions and poolings. This information is prepared to feed the neural layers by the flatten operation. After our neurons work, we will classify the information by some function as a SoftMax. This model is what we name after Convolutional Neural Network (CNN). *Image authority. Brain: Miquel Perello Nieto, Wikimedia Commons. Used under Creative Commons Attribution-ShareAlike 4.0 International, without making any change. Visual cortexes: Markus Bongard, Webvision. Used under Creative Commons Attribution-NonCommercial 4.0 International without making any change. Convolutions: see Ref. [155].*

Mathematical convolutions give the amount of overlapping of one function while it shifts along another function. In the case of images, a small picture composed of a few pixels called *kernel* is shift along the picture that arrives into the operation. Convolutions are often used as filters when performing a traditional tracking, since as example, it can give us information of borders in the images as with the Sobel filter or smooth images as with the Gaussian filter. When we apply a convolution in a convolutional neural network, we also get similar information for the first convolutions, but if we apply more and more, we might extract also some higher-level features that allow the network to understand what the important information in the object is as we do biologically.

Convolutional layers also allow us to modify the size of the image depending on the *padding* and the *stride* we use for them. Adding a padding means to add a border of blank pixels around the picture. Stride is the number of pixels we move the kernel each time. No stride means we move one pixel, and hence, we calculate the convolution on each pixel. Depending on the size of the padding, the stride and the size of the kernel[3], the image might get reduced. Reducing the image is important because it allows us to reduce more the number of weights to calculate and thus, the number of free parameters our system has. But at the same time, we keep the main features. With the idea of reducing pixels, there is also a special kind of convolution called *pooling*. Pooling uses a configuration for always reducing the size of the image. There are two kinds of pooling: the *average pooling* and the *max pooling*. While the first average the values of the pixels in the kernel, the second gives only the maximum value of the kernel. Pooling layers are often calculated after a neural layer to keep the features but remove fine details. In fact, the complex cells in Fukushima's network were performing an average pooling.

Finally, in CNN we will probably see a *flatten* layer and a *SoftMax* layer. Flatten layers restructures the image into a 1D array by adding each row of pixels after another one, which is the format accepted by the neural layer to feed the neurons. SoftMax layer is usually done at the end of the process to classify. SoftMax is an activation function that transform the input into a number between 0 and 1, thus giving the idea of a probability. For instance, the final output of our neural network could be an array with as much elements as kind of objects we want to detect in our CNN model. SoftMax will take this array and transform it into probabilities, which allow us to interpret statistically which are the chances that we have detected either one kind or another kind of object. Mathematically, SoftMax remembers to statistical mechanics. The input would be as the energy of a state, and the SoftMax is the probability of appearing this state given a Boltzmann's distribution.

[3] See [155] for a visual understanding of these concepts.

4.2.4 Region Based Convolutional Neural Networks (R-CNN), fast R-CNN, faster R-CNN and You only look once (YOLO)

The result of a CNN allows us to classify images among the objects we trained it for, but the classification is over all the image. Instead, my aim for using neural networks is to detect objects inside images. Thus, we need an extra step. For example, we could divide our image into areas, and give each area to the CNN. This would give us a map for the entire image of where is probable to find the object we want to detect and hence, to detect inside the image the objects we wanted. But depending on the size of the areas chosen and the size of the image, this process could be slow. Furthermore, our objects could have different ratio size, and this would be an extra problem. To solve these problems, a few years ago it was proposed the R-CNN algorithm[156], which is based on the *selective search* algorithm[157]. The selective search algorithm solves the problem of the areas by doing a high *segmentation* of the image following the method given by Felzenszwalb and Huttenlocher[158], where they find areas with similar pixel intensity by using graphs. Once the image is segmented in many pieces, the selective search tries to join areas with similar colours, textures, or sizes. All the possible collection of areas will be used as regions where to find objects.

The R-CNN algorithm uses around 2000 of these areas per image where to do the detection. Once the areas are defined, it crops the image on those areas and warps them to a fixed size that depends on the initial layer of the CNN. Then, it performs the CNN per area and classify the areas per object detected. Finally, it combines the different areas overlapping the same object and gives a unique object per unified area. Although this algorithm could detect objects in images, it was slow, and the selective search method cannot learn since it does not have neurons. This means that the areas to use could not be as good as they could be. To improve the speed of the algorithm, the same author created the *fast R-CNN*[159]. In the fast R-CNN, instead of doing the selective search on the original image, it first does a CNN of all the image to create a feature map and then calculates the selective search on this map. Then, it feeds these areas into a CNN composed of a pooling and a neuron layer, which ends with a SoftMax layer. In this way, you avoid repeating the convolutional process for all the areas and calculate it only once. But fast R-CNN still uses the selective search algorithm. To avoid this slow method, Shaoqing Ren et al. propose a change in fast R-CNN, creating the *faster R-CNN method*[160]. In the faster R-CNN, they exchange the selective search per a different neural network. Their idea is that this network can be trained to look for better areas where to detect the objects. While their approach improves the results, it also made faster R-CNN capable to do real-time object detection.

At the time faster R-CNN was published, I was in my first's months of PhD, without any knowledge on neural networks, neither thinking that I would use something relative in my near future. At that time, it also appeared another model to detect objects that was really fast on the object detection, more than faster R-CNN. This algorithm evolved, and by the time I started with this tracking project,

this algorithm was already in its third version. This model is named *You only look once (YOLO)*[161]. Opposite to the previous methods, YOLO only gives the image once to the neural network. YOLO divides the image in a lattice of $n \; x \; n$ cells, and for each cell it calculates the probability to contain an object and the probability of this object to be of a specific kind. Moreover, by using a few predefined areas, which we also know as *anchors*, it finally estimates the final area where the object is, to which we refer as *bounding box*. Here the weights are estimated properly to move from the anchors to the bounding boxes.

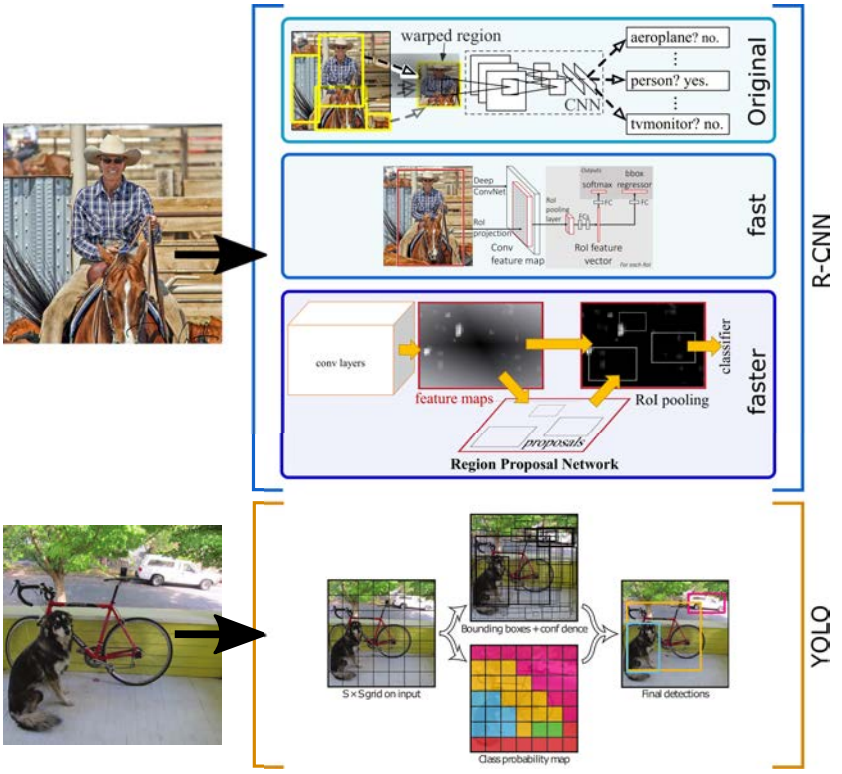


Figure 4.7: Different strategies to detect objects in images by using a CNN. One of the famous algorithms is the R-CNN, which have had several evolutions that improved its performance. In the case of this thesis, I decided to use YOLO, which is a faster alternative to R-CNN algorithms without losing accuracy. YOLO divides the image in $n \; x \; n$ squared regions and then proposes up to 3 bounding boxes per each cell. For each cell it decides which is the highest probability of this cell to be an object and assign to this cell that object. Later, it composes the detection by joining both layers. *Image authority. R-CNN images are originally from Ref. [156, 159, 160]. YOLO images are originally from Ref. [161].*

4.2.5 Training and variables that indicate a good training

To train a neural network we need to feed the network with images that contain the objects to detect and, if possible, also with images without these objects to reduce false positives during the training. Thus, technically we need to prepare a dataset with images which the neural network will use to learn. Of course, initially the network does not know what objects we want to detect. Therefore, complementary to this dataset we also need a dataset to explain to the neural network where the objects are to detect in each image. To define each object, we need to draw a polygon on top of each object to detect present in these images. We will refer to this polygon as the *ground truth bounding box*. Then, we store the coordinates of these polygons in plain text files, which we include in our dataset jointly with the original images. The characteristics and format of these text files will depend on the software we will use for our network.

With this dataset we can start training the network, which means that we are correctly estimating our weights. Basically, the network will give a bounding box per object detected, and will compare these boxes with the ones we fed. The more boxes coincide, and the more they overlap, the better estimated will be our weights, since that means that what the network is detecting is what we said there was in the dataset. Thus, to help us with this decision it is very useful to introduce the *intersection over union (IOU)* and the *mean average precision (mAPs)*.

The IOU estimates the area of intersection between the bounding box and the ground-truth bounding box with respect the union of both areas (See Fig. 4.8A)). This ratio is confined between 0 and 1, and we can set a threshold to determine if the bounding box is well estimated. IOU with higher values than the threshold will be considered as a true positive, while lower would be a false positive. If we could not detect anything but there was an object in the dataset, then this means we have a false negative. If we did not detect anything and there was nothing there, we have a true negative, but these are not used when training the data. From these values, we can extract two new parameters, the *precision* and the *recall*. The precision measures which fraction from among all the bounding boxes determined were in fact real objects or simply, the accuracy of the predictions. Recall measures how good the network found all the positives from among all the ground-truth bounding boxes (See Fig. 4.8B)). This is:

$$\begin{aligned} precision &= \frac{True\ positive}{True\ positive + false\ positives} \\ recall &= \frac{True\ positive}{True\ positive + false\ negatives} \end{aligned} \tag{4.2.2}$$

Both values are constrained to be in the region [0,1].

When the network is training, for a given step we will train the network using some images from the dataset. For each image, we will have the detections calculated by the network, and the predicted confidence level (the output from the SoftMax layer) for each object detected. We can order each image by the predicted confidence level and draw the curve precision-recall line (See Fig. 4.8B)). The area under this curve is what we call as *average precision (AP)*. Usually, before calculating the area, the line is corrected. When recall improves, accuracy usually drops, because false positives increase too, but it will increase later because true positives will increase too. To avoid these drops, the precision is taken as the first maximum decision towards higher recall values. If we average the AP using different thresholds for IOU, we will have the mAP. The convention is to average using IOU from 0.05 to 0.95 with steps of 0.05. Higher mAP will indicate better object detection.

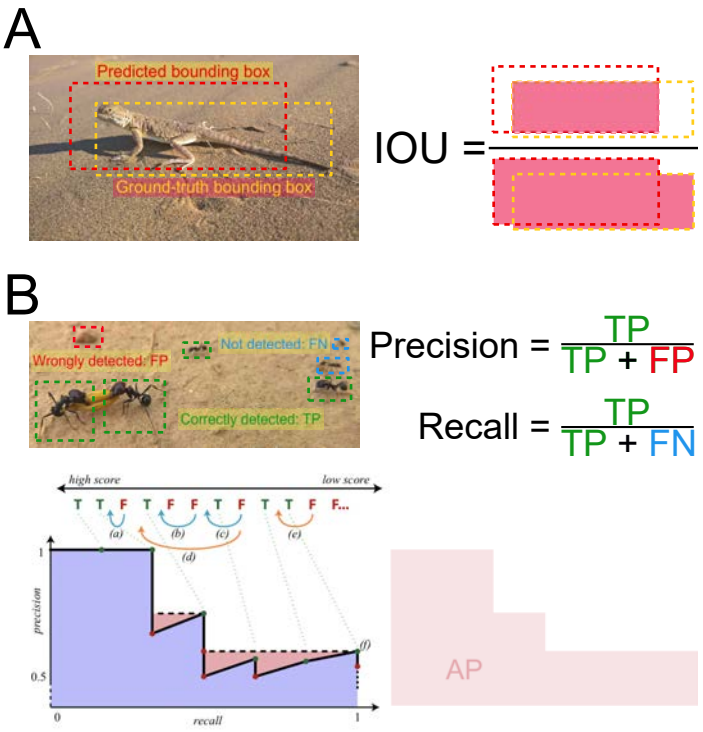


Figure 4.8: Different variables for training. **A)** IOU is the area of intersection between detection and real object, with respect the union of both areas. **B)** Precision is the quantity of real positives from among all the positives detected, while recall is the number of real positives detected from among all the objects. Both parameters are used to calculate the precision-recall curve for each step. This curve is made by ordering the detections by confidence (violet area), and then is corrected (red area). This corrected area is the Average precision (AP), another indicator for training images. *Image authority. Own images. Image for AP curve is originally from Ref. [162].*

The software will train the network until we stop it, or it ends all the cycles indicated to train the network. Training for longer times can give us a better output, but it can also produce overfitting, which will make our detections worse. Hence, if after sufficient cycles the results do not improve, it is better to stop training the network. After training it, we will obtain an output file which include all the parameters fitted for the network. This file will be loaded afterwards by the network to do the detection.

Notice that when training the network, we use a characteristic dataset. If we want to detect an object under very different conditions, we should train again the network adding images from these new conditions. Otherwise, the network will probably still be detecting the object of interest, but its detection will be worse than expected. The more the training dataset is close to the real images you want to detect, the better will be the output. Thus, add to your dataset as many images from as many possible conditions as you will have in your experiments.

4.3 | Classifying data to train a neural network

For this thesis, I decided to use YOLO at its third and fourth versions because several reasons. First, this software is one of the best and faster detectors at the moment I started with this project. Second, it has an active community working on it. This fact implies that any new characteristic for convolutional neural networks will be implemented, and any bug found will be debugged fast. But also, even if new stable versions are not released, little changes in the schemes for neural networks can be implemented by the community (e.g., by adding more layers or changing their properties), and they are easy to modify by any user. Finally, the software also carries with the corresponding methods for loading the data, calculating anchors from the dataset we feed the network and to do *data augmentation* without needing to code anything. Data augmentation is the process of creating new images from the dataset which we use to feed the network. Over the images used, we can do different transformations such as shift, rotation, scaling, blurring, or joining images into one image. These transformations multiply the original dataset available and reduce the number of images needed to hundreds or thousands.

After selecting the candidate software for object detection, we need to prepare the dataset accordingly. On the Internet, there are some open-source projects that work with images, but not videos, and that allow us to create bounding boxes with many vertexes. But YOLO only needs rectangular bounding boxes and my data will come from the videos I will record. Thus, I preferred to code a software for easily creating the dataset.

The software I developed is based on Python and uses OpenCV library to manipulate the videos. The main idea of this software is to present different frames per each video to the user. Then, the user can draw every bounding box with a maximum of 3 clicks per object. Since a video is a collection of frames, we could use just one video and extract many frames from it. But when training, the dataset must include as much diversity as we can. Otherwise, the detection will be very specific and will not work properly. Thus, if we have a collection of videos, the best we can do is to select a few frames per video. Furthermore, since frames close in time will have similar information, it is useless to analyse frames that are close between them. Instead, it is better to skip many frames in between two frames of the same video. Thus, my software needs the path of the folder where all the

videos are and where the output will be, the number of frames to skip between two frames to analyse and the number of different kinds of objects we want detect because YOLO can detect different kinds of object while detecting. After these parameters are set, the software starts loading a video, and shows the first frame to the user. At that moment, the user can click with the mouse for every object there is in the scene. Left click will add a coordinate for the bounding box. Right click will define a new bounding box. After the third left click, a rectangle with the minimum size but containing all the points is drawn. From the third click on, the rectangle will be modified to continue enclosing all the points but using the minimum space possible. Alternatively, if all the objects have the same size, it is also possible to add a rectangle per each left click with the predefined sizes of the rectangle. The software will use the coordinate of the click as the centre of the rectangle. To change from mode to mode, the user just needs to press the "m" key (See Fig. 4.9). After the user draws all the rectangles, by clicking a key the software closes the frame, saves it as a new image and saves a text file with the coordinates of all the bounding boxes accordingly to YOLO format in a plain text file. Next, the code will skip the determined number of frames defined by the user and will load another frame from the same video. The code will repeat this process until it cannot load more frames from the same video, and it will load a new video automatically.

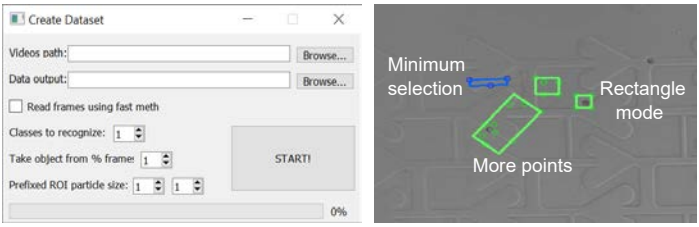


Figure 4.9: Creating the dataset. Self developed software graphic interface (left) and example of its use (right). Each green or blue square is a different object that the user has signalled as an object.

During my thesis, I used the tracking algorithm on two different kinds of particles. The first kind of particles were my Janus particles at a small magnification for the project in chapter 7. Later, I decided to use it on other projects running in the experimental lab. For instance, I tried to detect amorphous particles that generate too many bubbles and are not possible to detect by the traditional available tracking at the lab. Thus, I had to obtain two different datasets with different characteristics. In the following subsections I will show a few remarks on how to properly choose the dataset when you have similar problems.

In appendix C.3.3 I share this software, with a brief explanation to install it.

4.3.1 Dataset for small and constant size objects

One of the biggest challenges to detect the particles in chapter 7 was the size of these particles. Because of the size of the circuit where my particles had to move, I had to record videos at a small magnification. This implied that particles became very small, and thus, difficult to detect. Although there is always a small variability in the size of the particles I had, on average, any particle could fit in a square of 9x9 px^2. Detecting these particles with the traditional method was almost impossible due to the variable contrast of videos, the interfaces present there and because interparticle contacts. Thus, neural networks were a perfect solution. But to detect things as small using neural networks is neither easy. The third version of YOLO improved the detection for small objects compared with previous versions, but still, it is not easy. Furthermore, traditionally one expects to have a dataset where there are one or two objects to detect per image, and hence it is recommended to have hundreds of images. Instead, in my videos I have many particles. This means that if in every frame I can easily have 80 particles, with a few frames I can achieve thousands of particles, yet I will only have a few images to present to the network. With as many particles per image, the process to obtain hundreds of frames can be tedious and a better approach is needed.

In any case, since I could not find information about what happens if one has many objects per image, I thought that maybe even I only had around 58 images, this approach could work due to the number of particles per image. Although the network learnt something, the results were not very good. At that moment, I tried with the 3 schemes available[4], from the simplest tiny scheme to the more complex SPP scheme, passing through the complete scheme for version 3 (See Fig. 4.10). But these trials were very useful because at the beginning I thought that by making more complex the scheme of the network, the results would be better but instead, it is not only about the complexity of the network but about how we present the data to the network.

To improve these results, the first we need to know is that the neural network starts its training by wrapping every image in the dataset to a fixed size, which is a parameter we can change in every scheme. In my case, the frames of my videos had a size of 1600x800 px^2 approximately, but the biggest value by default to wrap the images was 608x608 px^2. Therefore, if the initial size to wrap the image is smaller than the image present in the dataset, at the moment the network wraps the image it will reduce more the size of the particles. Since my particles were 9x9 px^2, this could make even worse the detection. Thus, a possible solution to detect better my particles could be to increase the size at which the network wraps the images from the dataset. Images will enlarge and hence particles will be bigger and easier to detect too. However, this approach is only valid if one has unlimited PC resources, which is not usually the case. The main reason is that the size of the network scales with the initial wrapping size, and hence, the bigger is the size to

[4] I refer as a scheme to the amount, kind and order of layers in the convolutional neural network.

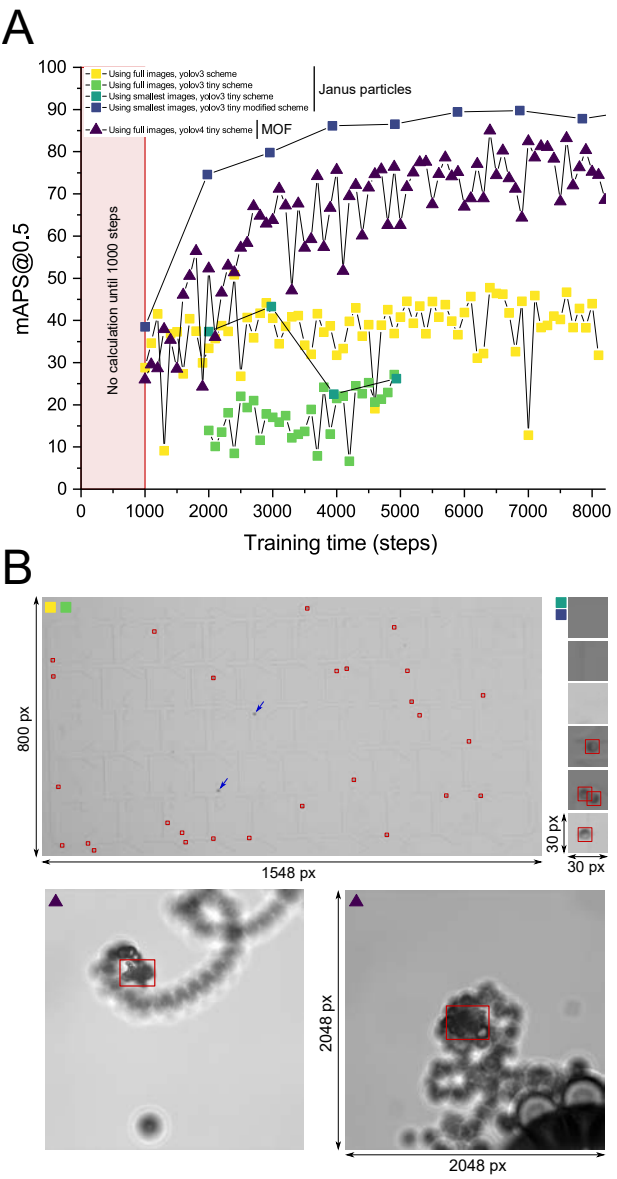


Figure 4.10: Training the network. A) Training of Janus particles and metal-organic framework (MOF) particles using different YOLO schemes. Using the tiny v4 scheme is good enough to detect MOF (violet triangles). However, for Janus particles (squares) I used the third version because the fourth was unavailable at that moment. None of the schemes were good for training them, except when using a modification of images and scheme. **B)** Images used for the dataset. Top: Janus particles. Bottom: MOF. Red squares: Ground truth bounding box. Blue arrows show particles to detect without the red squares for better visualization.

wrap the image, the bigger will be the network. But when one trains the network, the PC reserves as much memory in the *random-access memory (RAM)* as the size of the network. Therefore, if we want to increase the initial wrapping size, we need a considerable amount of RAM.

Although nowadays is not rare to have a PC with 32 or 64 GB, the truth is that when training the network, we are much more limited. To train the network one can use the *Central Processing Unit (CPU)*, in which that case we would use the main RAM of the PC, but these algorithms are very slow to apply using the CPU and we could spend weeks to train the network using a CPU. Instead, one needs to use the GPU, which it is at least an order of magnitude faster because the algorithms used are very easy to parallelize, and hence the calculation is boosted. But if we use the GPU we will have to use the RAM present within the GPU instead of the main one the computer has. Increasing the RAM in the computer is easy, and it is relative cheap. But one cannot modify the RAM within a GPU. Thus, when applying these software's it is very important to buy[5] a good GPU for testing. Furthermore, if we increase the size of the image, the detection will slow down, and depending on what we want to do later, it can be impossible due to the speed of the process.

Therefore, if increasing this size can be a problem, I decided to modify the images I already had. Instead of presenting this dataset, I presented smaller images, which at the same time increased my dataset. This is easy to achieve by just cutting the images in smaller pieces. I tried by just cutting them in 8 pieces, but finally I decided to reduce even more the image. For that purpose, I coded a new python software to cut the images with a small ROI around each particle. The size of this ROI is of 3 diameters the size of my particles. Thus, I could present many images to the network while increasing their size, but still allowing the network to learn that there can be particles in contact. Notice that if all ROI are centred at the centre of a particle, this will imply that all images have a particle at the centre of the image. Although the network should be translational invariant, I simply centred the ROI at the centre of the particle plus a small random quantity just to avoid any possible artifact. Moreover, I also implemented a function where, for each small new image containing a particle, I try to take another small image without containing any particle from the original image. This addition is very useful to teach better the network what is not a particle. From 58 images, I finally obtained 8700 small images containing at least a particle and 8700 small images without containing any particle, which I used to train the network (See Fig. 4.10)).

[5] Almost every year a new series of GPU is presented in the market. New GPU series often increase the RAM available, but the more RAM, the more expensive they will be. For example, a GPU with 11GB can cost around 1000€ (2020). If the user cannot afford these prices, maybe it would be useful to rent a server with such GPUs for the number of hours needed. I will avoid any publicity here, but it is easy to find them by searching on the net. It is also important to check the GPU needed for the software one uses. For instance, there are different libraries to parallelize calculus using GPUs. Some are opensource, and can be used with any GPU, but in this market, there is a preference for CUDA libraries, which is a library one can only use with NVIDIA GPUs. Remember to check if all the software and libraries you will use in your project can parallelize the work with the GPU you will use.

Before training the network with this dataset, I also did a few modifications on the scheme. First, one of the YOLO forks available in GitHub allow the user to calculate the anchors given the dataset associated. Usually, when one presents objects to the networks, they will have different sizes but in my case all particles have the same size. Therefore, one can expect to have anchors very similar to final bounding boxes. Anchors calculation is processed within a minute, and in my case gave me two sizes which were almost equal, and around 25x25px [2]. As I previously explained in section 4.2.4, YOLO divides the image in a lattice of nxn cells but in fact, it does not only divide the original image once, but three times with different n. For each division, there are 3 different anchors sizes. In the step with a small n it will detect the larger objects, while in the bigger, the smaller objects since increasing n means that you divide your image in more cells and these will have a smaller size. For the bigger n, YOLO already have anchors of the size of the ones I calculated. Therefore, any other cell division is useless here because of the constant size of the particles. Hence, I removed those calculus and I changed the anchors to the ones I had, removing also one since I only had two. Finally, I also activated the different options for data augmentation, and I also changed colour images to grayscale images. Grayscale images is important to reduce the amount of RAM needed per each image. Since colourful images have three channels and a grey image only one, by converting the image to grey scale you save 2/3 of the amount of RAM needed to do the calculus. In my case, since the images are now small, this is not an important step, but I was already doing it since in the past I needed to save RAM. In any case, my videos are white and black, but when they are stored, the algorithm store them in colour. Thus, the colour is not important, and the benefits are bigger than the cons.

With these changes, I decided to use the simplest scheme available because for improving the tracking I needed the fastest detection available. The results were excellent, as after a short time I got a mAP-50 larger than 95%, in comparison with the rest of cases where I only got around 40% (See Fig. 4.10). Moreover, this modified scheme was faster than the other ones, which means that in the tracking this process will be faster, something important when one can have lots of particles in a single frame (See Table 4.1)

Scheme	Images type	mAP@50%	Time
YOLOv3	Complete image	40%	10 h 1 min 17 s
YOLOv3 tiny	Complete image	27%	4 h 13 min 18 s
YOLOv3 tiny	Smallest image	27%	2 h 17 min 45 s
YOLOv3 tiny modified	Smallest image	87%	0 h 3 min 38 s

Table 4.1: Training results and time spent using an NVIDIA GTX 2080 Ti graphic card and an Intel Core i9 10940X after 5000 training steps.

4.3.2 Dataset for large and different size objects

At the time I created this dataset, YOLO was in its fourth version, and hence the scheme used was for the fourth version. In particular, I still used the tiny scheme but for the fourth version. In this case particles are amorphous, bigger and present different sizes (See Fig. 4.10B)). Hence, here the difficulty does not come from the size. Instead, the main problem comes from their propulsion because they create a lot of bubbles while moving and makes impossible to track them with the available tracking in the experimental lab. In fact, the PhD who carried these experiments had to track them manually.

To train the network with this kind of particles I measured the anchors, but I did not remove any layer for calculus because of the different sizes they can have. New anchors sizes were put on those layers with of the same range of anchors sizes. I modified the scheme for grey images and to do data augmentation, but I did not cut the images due to their big size. Furthermore, cutting images here almost would not increase the number of images because almost all videos only had one particle per video. To reduce the area of the bounding boxes I also implemented a function to create rectangles that fit the particles and that their edges do not have to be parallel to the borders of the images (see Fig. 4.9), but later I noticed that YOLO needs a rectangle with their edges parallel to the borders of the frame. Therefore, I discarded this output and saved the data with rectangles with two of their axis parallel to the frame rows.

The results obtained were good enough to track particles. After 5000 training steps I had a 75% mAP@50%, which took 45 min 47 s using the same machine as the one used for training the detection of Janus particles.

4.3.3 Conclusions to choose a proper dataset

As a conclusion for any user that wants to detect their particles, here there is a list of steps to check every time one creates a dataset:

- Use as many videos as possible, with different visual properties. They should have a representation of what you will see in the rest of videos to detect.
- Only use frames that are temporally separated among them.
- If particles are very tiny, cut the images. For example, create images of a few times their characteristic length. Do not put a particle in the middle of each image. If cutting the image is not possible, apply some stride in the scheme as proposed in https://github.com/AlexeyAB/darknet/#how-to-improve-object-detection
- Transform all images to grey images, if possible.
- Try first with a tiny scheme.

- Modify the scheme to include only grey images and data augmentation.
- Calculate the anchors.
- Modify the anchors in the scheme with the ones the software proposed. If YOLO has a layer with anchors that have a size totally different from the ones proposed, remove that layer.
- Train the network. If the result is deficient, try with a better scheme.

To speed up this process and make it user friendly, I decided to implement all these procedures in a new python software (see Fig. 4.11). See section C.3.3 for the source code.

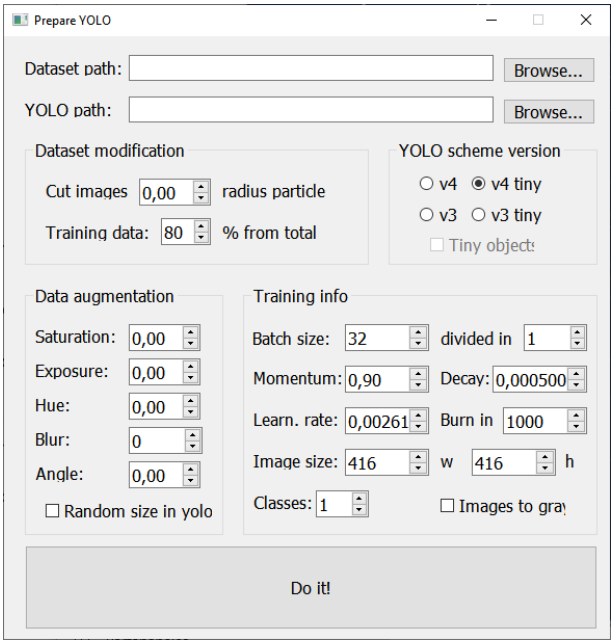


Figure 4.11: Preapering scheme and polishing the dataset. The user can choose directly with this graphic interface different options to modify its dataset to train the network and prepare the scheme to be used for YOLO. This software can use different YOLO scheme versions (YOLOv3 and YOLOv4) with its full scheme or simpler one (tiny). Then, the user can select the options for data augmentation to make bigger their dataset at the runtime of the training. The dataset will also be bigger if the user decides to cut the images in smaller images. If this option is not 0, the software will also find images without particles. Finally, the user can select options for the training, such the number of images to be used for training/to test the trained network, the number of images to train the network at each step (Batch size) or the size of the images it will be feed to the network (Image size), if we will convert the images to grayscale and other properties of the training.

4.4 | Tracking of particles using the trained network

YOLO software allows to do detection in images or videos using users own trained network but is not capable yet of doing tracking. Therefore, we need to build an alternative software to do the tracking. To solve this issue, I decided to build a tracking software in python because I needed to implement it fast, and because I already have experience on treating videos with the OPENCV library. Furthermore, OPENCV library also includes a module dedicated to neural networks that is capable to read pretrained networks with YOLO [6] and do the detection process. Therefore, we can do the tracking software easily with python and OPENCV.

To track particles, one needs first to detect and give an identity number to each detected particle. If the network is well trained, almost all the objects of the scene will be detected, but we can always have false positives and false negatives. For instance, one of the problems I could see when detecting particles in a video using YOLO was that particles were not detected always at every frame, and it was happening frequently for different particles. If particles are alone, this could not be a problem if we implement some kind of memory as Albert did for his tracking, but if we have many particles next each other, particles could exchange identities. Thus, this conditioned my approach for doing the tracking algorithm.

The tracking software I created is easy to use, and any user can use it (See Fig. 4.12A)). The only requirement for the user is to correctly train the network and have particles that do not overlap each other. Once the network is trained, the user must introduce the paths for both the output of the training and the neural scheme used during the training. The user also needs to introduce the folder where videos are. The software will load the first video to track, and once finishes tracking that video, it will load the next video until the user tracks all the videos from the selected folder. When the software loads a video, it will show the first frame to the user and will do a detection of particles using all the frame. Since the size of the dataset can be differently (e.g., if you cut the images to feed the network), to improve this detection I let the user to change the size of this frame for a better object detection. In any case, the user can avoid changing this parameter because after the detection, the software will show all the bounding boxes on the frame and the user can edit the position of the bounding boxes, remove them, or add news

[6] OPENCV is capable to read networks from different software as pytorch, cafe or tensorflow among others. Check dnn module for more information.

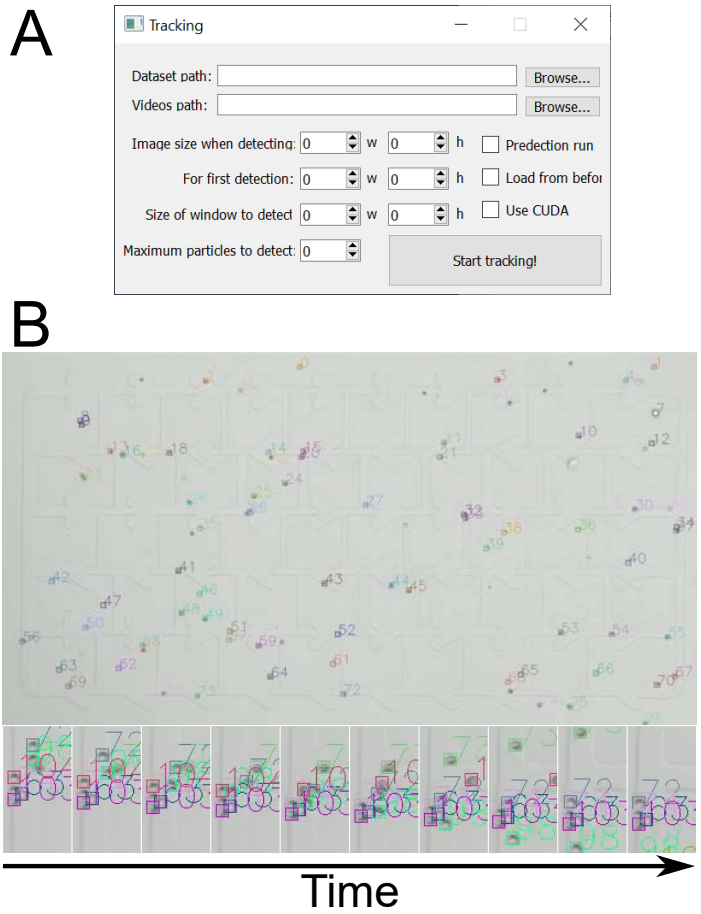


Figure 4.12: Tracking of particles. A) The user can choose directly with this graphic interface different options to track particles. **B)** Example of a frame tracked with particles of size 9x9 px^2. Under the main image there is an example of time evolution of the tracking with multiple particles colliding.

by doing a click with the mouse. After finishing, the user can press a key and the software will load the next frame. From this new frame until the last frame, the software will repeat the following process:

- For each particle and time step t, we crop a frame around particle's position. The size of this cropped window is selected by the user when initializes the software, but I used 2 particle diameter.
- The software will resize the cropped image to improve the detection, and then it will perfom the detection. To detect Janus particles I enlarged 8 times each axis of the image, but this is a parameter that the user can select when initializing the software.

- The number of detections do not have to match with the number of particles in the cropped image. Thus, to select the best detection we calculate its closest particle among all the particles in the main frame, not the cropped frame. Because each particle has an identity (id), we will also keep the id for the closest particle to each detection.

- We filter distances by id. We select the previous set of detection-distance-particle id and filter per id. We only keep those distances where the id corresponds to the particle id we want to track. Among these filtered detections, we will assume that the new position of the particle is the closest detection.

- If there are not distances with the same id as the particle we are tracking, we will take the detection with the shortest distance to the centre of the cropped image.

- If there are not any detection, we will keep the centre of the cropped window as the new position for the particle.

- We repeat this process for every known particle. Meanwhile this is processed, the user can click on the frame to add a new particle to the collection, which will be added in the next frame.

- Those particles that have been several frames in the borders of the video without changes in their position are particles that left the ROI of the video. Thus, we will stop counting them.

- Before accessing to the next step, we will update particles position for the next frame with their speed to have a better detection. If the particle is located at position r_t at time t, we estimate that its location r_{t+1} at a time $t+1$ is:

$$r_{t+1} = r_t + \frac{r_t - r_{t-1}}{2}. \tag{4.4.1}$$

This algorithm avoids particle exchange for particles that cannot overlap. If particles hit between them or are next together, they will conserve their id without exchanges (See Fig. 4.12B) small images). When more than 5-6 particles are in contact, such in big clusters, this algorithm starts to fail. However, I discarded exploring solutions to this problem because when I tracked videos for chapter 7 I could notice that the code was already slow. For example, I could track 200 particles simultaniously for a total of 7200 frames every 15 minutes. However, for videos with a couple of particles this software runs very fast.

Improving the tracking software

The main drawback relative to this software resides in its linearity, but many parts can be parallelized. For example, the software used to detect particles using the CPU. Using the CPU for neural network processing is not recommended, but there was not other option in OPENCV library. However, by October 2019 the community started to implement the detection process with GPU processing and just by changing a flag, I could improve the speed of the detection calculus by a factor of 10 times[7]. After I solved this issue, I started looking at different areas.

The next idea I had was to parallelize the main loop of the software. Instead of detecting for a particle, calculate its distances and repeat for the next particle in a loop, this process could be done simultaneously. In Python one can parallelize code using the NUMBA library. However, this library requires numpy arrays along the code. Although when coding numeric arrays in Python one usually uses numpy arrays, I thought that simple lists could faster processes and therefore I had to adapt the code to numpy arrays. Numpy arrays did not slow the software but improved it. Specially because one of the steps required to convert a list into a numpy array, which is a slow method. Because these couple of improvements made the software faster and comfortable to use for my requirements, I stopped introducing NUMBA, but any user can implement it. As a consequence of NUMBA, now arrays cannot be dynamic, and hence the user must set the maximum number of particles to detect when initializing the software.

The user can also improve the execution speed by improving the calculus for distances, which is the slowest part when tracking each particle. To improve this part, I tried to divide the image in pieces, and record which particles are in each piece. Nonetheless, this process slowed more the process and I discarded it. In fact, to solve this problem we need to parallelize the calculus of the distances using the GPU. Mainly, we need to solve a *k-nearest neighbours algorithm* . This is a classic computer science problem, and therefore there must be solutions avalaible for Python using GPUs.

Finally, in case the process is still slow, I added an option to run the movie without tracking. In this mode the user must click once the first time they see a particle. The next time the user opens the tracking software, the software will use these positions at the frame clicked to start tracking particles. Therefore, the user could run the tracking in a disregarded mode.

[7] At the time I wrote the software, the OPENCV installed from Ubuntu repository was compiled for CPU only. If we want to use GPU, we need to compile it for CUDA. See appendix C.3 for the instructions.

4.5 | Conclusions

Image processing is a stepping stone to analyse better our experiments. In this chapter I have explained several tricks and possibilities this processing allows us. Moreover, I have provided with a collection of codes and software for this porpoise.

Traditional tools such as mathematical convolutions can help us to smooth images or detect borders, processes that can be useful when tracking. However, if needed, there are more powerful tools to track objects such the case of CNN, where convolutions are mixed with neural networks. A tracking system based on the neural network for image detection named YOLO is presented, and different hints to improve image dataset and network training is provided. Then, a self-developed tracking system based on YOLO detection is given. All the codes related with neural networks are joined under a single software with a graphic user interface and are available for any user to download and using it.

Sometimes image processing must be done after tracking. The addition of lines or polygons in the images let us to do several transformations. For example, we can change the frame of reference, the coordinates system or add different methods to obtain new parameters. To proceed with this manipulation, we can code several algorithms in a language such as Java, but python provides useful libraries such as OPENCV and Shapely that makes faster the development of these algorithms.

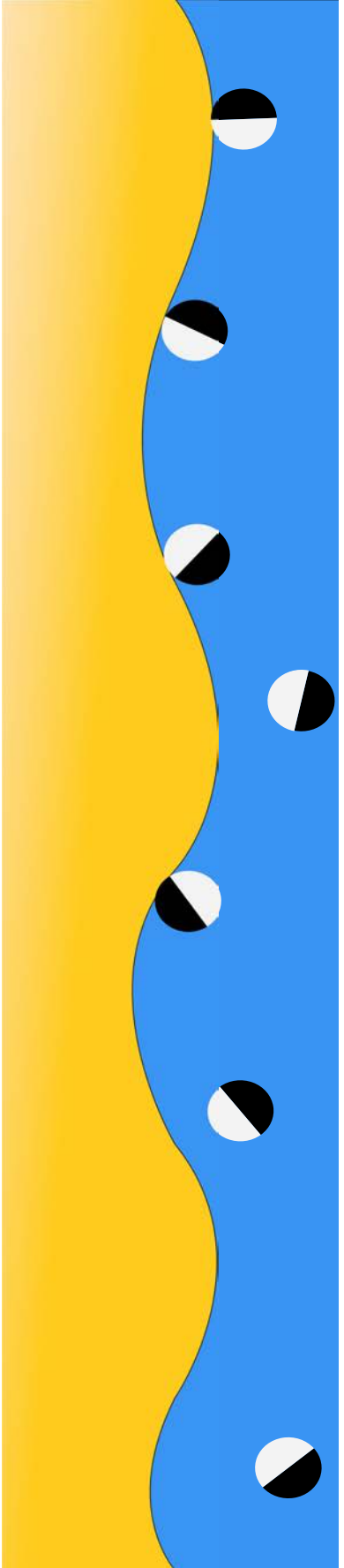
I hope the codes shown here will be useful for the community.

II

Active particles next to liquid-liquid interfaces

After describing in the previous part methods and tools useful to create setups and to record and analyse videos, in this second part I will demonstrate the behaviour of active particles when interacting with liquid-liquid interfaces.

This part contains two chapters. In the first chapter, I will analyse this issue from an experimental overview. In this chapter particles are not only contacting the liquid-liquid interface but also are on top of a solid interface. Hence, we work on a liquid-liquid-solid interface. In the second, I will create a model to describe the previous system. The model can also introduce the solid interface, but we considered that for the community would be useful to answer first some questions that are not easy to explain at pure liquid-liquid interfaces and later to introduce the solid interface and continue the research. Moreover, both studies are realized on the diluted regime, explicitly, at one single particle level. However, at the end of the first chapter there are some experiments with more than a particle, and the model can run with thousands of particles.



Guidance of active particles at liquid-liquid interfaces near surfaces

"The man who moves a mountain begins by carrying away small stones."

Confucius

Artificial microswimmers have the potential for applications in many fields, ranging from targeted cargo delivery and mobile sensing to environmental remediation. In many of these applications, the artificial swimmers will operate in complex media necessarily involving liquid-liquid interfaces. In this chapter, I experimentally study the motion of chemically powered phoretic active colloids close to liquid-liquid interfaces while swimming next to a solid substrate. In a system involving this complex geometry, we find that the active particles have an alignment interaction with both the neighbouring solid and liquid interfaces, allowing for a robust guiding mechanism along the liquid interface. We compare with minimal active Brownian simulations to show that these phoretically active particles stay along the interfaces for much longer times and lengths than expected for standard active Brownian particles. We also track the propulsion speeds of these particles and find a reduced speed close to the liquid-liquid interface. We report an interesting non-linear dependence of this reduction on the particle's bulk speed. The analysis of this interaction will help us to understand not only the behaviour of these particles when they move next to these interfaces but also, it will help us to model this problem in the next chapter.

The present chapter is based on the paper "Guidance of active particles at liquid–liquid interfaces near surfaces"[163], from which I am first author, and to which I did the experiments and model, analysed and wrote the paper. I would like to special thank Jaideep Katuri, who helped me with this project, which was the first experimental I did. I also want to thank Marco De Corato for his discussions on the ABP model, and to Lei Wang, who showed me how to create droplets of oil for further research after the work of this paper.

5.1 | Introduction

Biological microswimmers necessarily operate in complex media involving viscosity changes and liquid-liquid interfaces[164]. At interfaces, their swimming behaviour can change remarkably and exhibit new phenomena that are not observed when they are swimming in the bulk. In section 1.1.2 I already explained some examples, but we can even find more. For instance, bacterial trajectories can change from run and tumble behaviour in the bulk to circular trajectories when close to a solid interface[165]. At higher densities, bacteria can create biofilms when an interface is present[166, 167, 168] but not in the bulk. These changes in behaviour are very important not only as natural processes that offer an evolutionary advantage[169, 170, 171], but also because of the numerous industrial applications[172, 173, 174] that have been developed for several microswimmers in the last few decades.

Recently, new advances in materials science have allowed us to create new kinds of synthetic microswimmers that can mimic natural microswimmers such as bacteria[18] in remarkable ways (see section 1.1.2 for examples). These synthetic swimmers consume energy from their surroundings and engage in out-of-equilibrium behavior such as self-propulsion, much like their biological counterparts. A prominent example of these synthetic microswimmers is that of self-diffusiophoretic colloidal particles. These particles have an asymmetric coating of a catalytic material on their surface and have the ability to locally generate chemical gradients, which leads to their self-propulsion (See section 1.4.3). In addition to engaging in self-propulsion, these particles can also interact with each other and their surroundings via hydrodynamic and phoretic fields[51]. Recent experiments have shown that synthetic microswimmers tend to accumulate near surfaces[56, 65], similar to motile bacteria. While hydrodynamic interactions are thought to be primarily responsible for the accumulation of bacteria near surfaces[175, 176], in synthetic microswimmers, it is indeed a combination of both phoretic and hydrodynamic interactions[56, 66, 177].

A broad range of applications have been envisioned for synthetic microswimmers, from drug delivery[21] to water remediation[178]. In all these applications, the synthetic swimmers will inevitably interact with different complex fluids and interfaces, making it of significant interest to understand their behaviour in these conditions. To date, there are studies describing the interaction of these microswimmers with liquid-solid[56, 65, 177] and liquid-liquid interfaces[66, 67, 179, 68, 180, 69, 181], both from an experimental and theoretical point of view.

While liquid-solid interfaces introduce a strict boundary condition for hydrodynamics and phoretic fields, liquid-liquid interfaces relax this boundary, leading to different scenarios[69].

At liquid-liquid interfaces, effects such as capillarity or Marangoni flows (see section 1.2.1) due to the difference in surface tension between the two liquids can influence the particle swimming behaviour[70, 182]. The difference in viscosity between the two liquids can also modify the velocity and orientation of the particles close to these interfaces. And finally, since synthetic microswimmers create phoretic fields around them which couple back to the particle motion, the differential diffusion of the chemical product at the interface can give rise to an additional contribution that is not observed for solid-liquid interfaces as we already saw in section 1.2.2. For instance, in a system involving only a solid interface, we have observed an active quenching of the particle's Brownian rotation close to a solid surface, leading to a stable orientation that is parallel to the adjacent surface, effectively restricting the particle trajectory to a 2D plane. We have previously elucidated the phenomenon of the angular attraction of self-phoretic particles to this configuration close to confining boundaries[56]. Briefly, we could show that a combination of gravitational and hydrodynamic torques tends to push the catalytic cap of the particle towards the bottom wall, whereas the contributions from phoretic interactions arising from the accumulation of the chemical product at the solid substrate and the resultant wall slip have the opposite effect, leading to a stable angle that is parallel to the bottom surface. But in the case of liquid-liquid interfaces, we might see a different behaviour since the system is different.

Different experimental studies performed with active particles at pure liquid-liquid interfaces[66, 68] have shown how pinning of the in plane angle plays a prominent role in the resulting particle behaviour at the interface. The initial pinning angle will determine the angle that the particles will have at interfaces and hence their motion, allowing for a sub-population of particles to have an increased characteristic rotational time and persistence length. As opposed to a pure liquid-liquid interface, in this chapter we study a configuration in which active particles can interact with fluid-fluid interfaces without needing to be in between them. Our set up consists of a liquid-liquid interface that is perpendicular to a solid interface created in a simple single channel geometry (Fig. 5.1). This configuration contains both the solid substrate along which the particles accumulate and swim, and a liquid-liquid interface that is perpendicular to it.

An active particle swimming on the solid substrate will eventually encounter the liquid-liquid interface, at which point it is effectively interacting with a solid-liquid-liquid interface. This is a very likely scenario in many application settings, as most synthetic swimmers are density mismatched and quickly sediment close to the bottom solid surface, along which they accumulate and explore the system. As an example of an application involving these interfaces, it has already been reported that chemically functionalized active particles can be used to clean oil

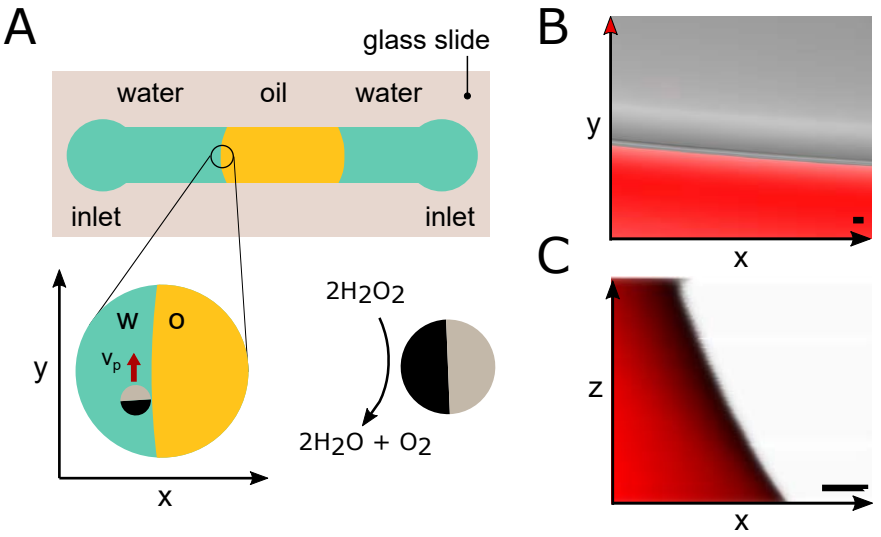


Figure 5.1: Schematics of our experimental setup. **A)** A microfluidic channel (Ibidi VI$^{0.4}$) is filled with oil (yellow) and a solution of water, hydrogen peroxide and Pt-SiO2 Janus particles (blue). Janus particles are propelled because of the O_2 gradient that they create. When they touch the liquid-liquid interface, the particles travel along this interface. **B)** Water-oil seen from the top. The water was differentiated from the oil phase by addition of Rhodamine-B into the water. Top view of the interface. **C)** Side projection of the reconstruction of the interface in 3D. In red, water mixed with Rhodamine-B. Z = 0 µm is taken as the bottom glass layer. Scale bars: 5 µm.

contaminants in water where a physical interaction between the active particle and the oil droplet is necessary[38]. Another interesting avenue to be considered is that of lab on a chip systems which are highly confined and the use of both immiscible droplets and active particles is becoming increasingly common. At the liquid-liquid interface, the chemical product from the catalytic reaction on the particle surface can diffuse through the interface, and any local accumulation or depletion is dependent on its solubility in both the liquids. It is a priori unclear what the net effect will be on the particle swimming at the interfaces due to this partial diffusion of the chemical product into the other medium, the difference in viscosity between the two media and surface tension related effects such as Marangoni flows at the liquid-liquid interface.

In this study, we aim to understand if the introduction of both a solid and liquid interface presents similarities in behaviour with the previous studies performed at purely solid interfaces and if we can find a simple effective model to capture our experimental results.

5.2 | Experimental setup and video analysis

We use self-diffusophoretic Pt-SiO_2 Janus particles of 5 μm and 2 μm diameter size as model synthetic microswimmers (See section 2.1 for particles synthesis). When Pt-SiO_2 particles are suspended in an aqueous solution of hydrogen peroxide, peroxide is decomposed into O_2 and water on the catalytic Pt side. This anisotropy of reaction products creates a gradient of oxygen across the particle surface, resulting in a net flow along the particle surface and a resultant self-propulsion velocity as described in section1.4.3.

Our experimental cell to build the surface adjacent liquid-liquid interface is composed of a single microfluidic channel (Ibidi $VI^{0.4}$) of dimensions l = 17.0, w = 3.8, and h = 0.4 mm. To create the liquid-liquid interfaces, we first introduce a droplet of oil in the center of this channel (18 μl). Then from either side of the channel we introduce a droplet of a dilute suspension of Janus colloids mixed in an aqueous solution of hydrogen peroxide (35 μl, concentration from 0.6% to 2.5% by volume[1]). This creates two oil-water interfaces in our cell where we can observe the interactions of Janus colloids with these interfaces (Fig. 5.1). This configuration also allow us to remove easily any excess of oxygen from the system through the outlets on either side without interfering with the experimental observations. Nontheless, we record our experiments using a contrast microscope (Leica) with a 40x objective for an entire duration of 10 minutes when the interface is stable. Over longer durations we have noticed that the interface can move due to interference from the accumulated oxygen in the water phase, which can generate bubbles of oxygen that stuck in the channel or burst.

The resulting videos from the observations were tracked using the tracking software developed by Albert Miguel López (See section 4.1). Orientations were obtained directly from the image in the case of 5 μm particles or by extracting them from the velocity vector due to lack of resolution in the case of 2 μm particles. In this last case, we had to skip data contained in every 4-6 frames to get a more stable value of this director vector. Once the raw videos were processed to find the Janus particle position and absolute orientation, the relative incidence, reflection and orientation angles with respect to the oil-water interface were calculated using the software referenced in section 4.1.3.

[1] To avoid decomposition of H_2O_2, H_2O_2 was stored at 4 °C.

5.3 | Encounter of an active particle with an oil-wall interface

As we are primarily interested in the behaviour of these particles at interfaces, we set our microscopy region of interest (ROI) to an area containing the solid-liquid-liquid interface, within a region of both oil and water phases (circled region in Fig. 5.1). Initially, the particles approach the interface from random orientations. During this approach, we do not see any long range attractive or repulsive interactions and the particle speed remains constant until it contacts the interface (Fig. 5.2A; see also the ESI S1.mp4 in ref. [163]).

We track θ, the relative angle between the orientation vector $\vec{m}$ of the particle and the interface (Fig. 5.2B), throughout the experiment. θ_0 is the approach angle, i.e., the particle angle at the point of contact. For all θ_0, we find that the particle eventually reorients until its orientation vector is parallel to the interface, $\theta_f \sim 0°$ (Fig. 5.2C), and the particle begins to move along the interface. The time that the particle takes to reorient to reach θ_f is in fact dependent on the approach angle θ_0 with particles approaching at higher angles requiring more time to reorient with the interface (Fig. 5.2D), as expected. Plotting the probability distribution of θ as the particle moves along the interface, we obtain a Gaussian spread with a mean value of -6 $\pm$ 10°, indicating no clear preference for the particle to point into or away from the oil phase (for individual fittings see Fig. 5.11 at Appendix 5.A).

In an effort to understand the role of the physical properties of the oil phase, we also repeat these experiments for different oil viscosities (5, 50 and 500 cst), but find no strong dependence on any of the parameters analysed, indicating the robustness of the reorientation mechanism.

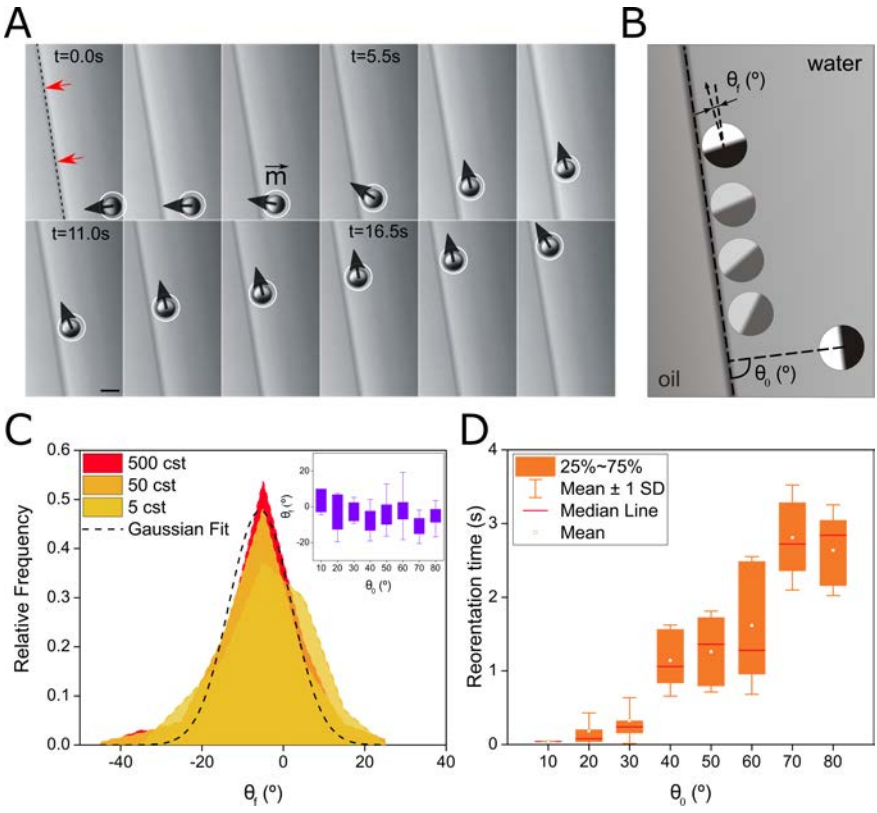


Figure 5.2: Particle approaching the liquid-liquid-solid interface and angle reorientation. **A)** Particles arriving at the liquid-liquid-solid interface reorient when they contact the interface and follow the interface with a stable angle alignment with respect to the interface (dotted lines). Arrows: interface (red) and particle orientation ($\vec{m}$) (black). The particle is circled. Scale bar: 5 μm. (See ESI S1.mp4 in ref. [163].) **B)** We captured the angle defined between the interface and the particle orientation at the initial time (θ_0) and when it stabilizes (θ_f). **C)** The particles' orientation changes after they contact the interface. They obtain a stable orientation angle that does not depend on the oil viscosity or initial angle. **D)** The time they need for completing the reorientation increases with the initial angle, showing us that the particles rotate with a constant angular velocity at the interface.

5.4 | Change in particle speed at the interface

We next study the influence of the interface on the propulsion speed of the particles. We classify the particles based on their propulsion speeds into two categories of $0 < v_{int} < 5$ μm/s and $5 < v_{int} < 10$ μm/s. We then compare the bulk speed of the particles with their speed at the interface (Fig. 5.3). For both the categories we observe a clear reduction in propulsion speed at the interface compared to their bulk speed. However, this effect is more pronounced for the particles with lower propulsion velocities $0 < v_{int} < 5$ μm/s as compared with the particles with $5 < v_{int} < 10$ μm/s. The ratio v_{bulk}/v_{int} for the former is 1.19 whereas for the latter it is 1.04 (Fig. 5.3B). To further study this speed reduction at the interface we plot the data with smaller bins of 2 μm/s and we see that the same trend is retained with increasing v_p (Fig. 5.3C; see also Fig. 5.11 at Appendix 5.B).

In general, the faster the velocity of the particles at the interface, the lower the loss in speed as compared to the particle's bulk velocity. This reduction in speed at the interface originates from the fact that although the particle aligns with the interface on average, the orientation angle of the particle is still subject to Brownian fluctuations and occasionally the particle orients into the oil interface, where a component of the propulsion velocity is lost (Fig. 5.3D). Since the effect of Brownian fluctuations is more significant for slower particles, the reduction in speed at the interface is more pronounced for these particles. Although in most cases the particle recovers its orientation from facing into the oil interface to the initial orientation in which it was propelling, we occasionally observe that the particle can change its propulsion direction and start travelling in the opposite direction as shown in Fig. 5.3E and in the (See ESI S2.mp4 in ref. [163]. This is observed solely for the smaller r = 1 μm particles where the Brownian fluctuations have a stronger effect. For these particles we also notice that occasionally (in 15% of the cases observed) the particles start scaling the interface in 3D and move away from the bottom plane. Both of these effects were not seen in our previous work[56]. We think that the less strong interaction between solid-liquid and liquid-liquid interfaces could explain these phenomena.

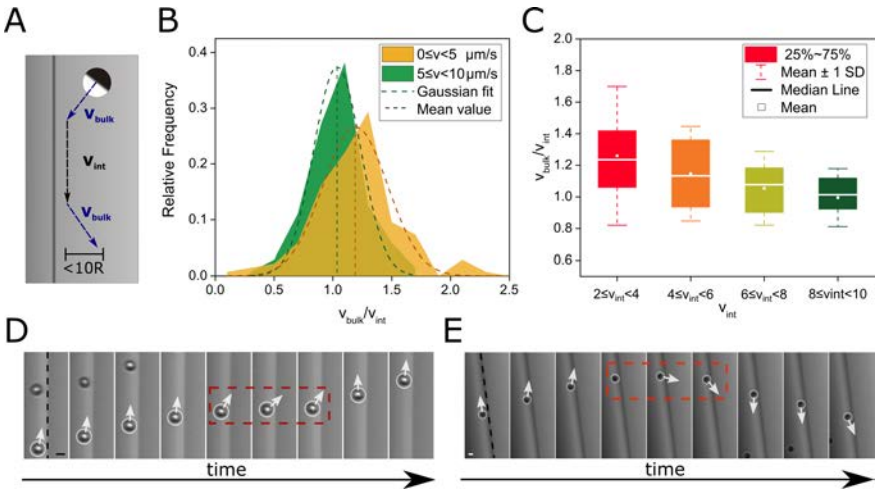


Figure 5.3: Speed reduction at the interface. We analysed the speeds indicated in (**A**)). Particles that move with a speed at the interface between 0 and 5 μm/s (**B**), orange) decrease their speed by around 19% with respect to their speed in the bulk, but if particles move at the interface with a speed between 5 and 10 μm/s **B**), green) their loss is around 4%. After fitting to a Gaussian, the mean values are statistically different after comparing 2 SD. (**C**)) We separated the data more accurately by using bins of 2 μm/s. There are no statistically significant differences between the groups, but we clearly see a tendency. The faster the particle moves close to the interface, the less speed is lost. (**D**)) We show one effect seen several times that could explain our decrease of speed. Sometimes the particles reorient with the silica cap looking towards the oil, giving extra time until they recover their speed. (**E**)) The last effect can induce particles to reverse their direction of motion. Scale bar (**D**) and **E**)): 2 μm. (See ESI S2.mp4 in ref. [163].) Time between frames: 0.44 s (**D**)), 0.76 s (**E**)).

5.5 | Alignment effect induced by the interfaces

As stated previously, we occasionally observe that the smaller R = 1 µm particles leave the interface. We calculate the retention time of the particles at the interface for different propulsion velocities and compare it to the bulk rotational diffusion time. Since the particles are on top of a glass slide with which there exists an active alignment interaction, we do not use the theoretical formula for the rotational diffusion time as we saw in section 3.2.2. Originally, in the article published, we calculated it from the angular autocorrelation of the particles moving far from the interface:

$$< \vec{m}(t) \cdot \vec{m}(t+dt) > \propto e^{-t/\tau_r}, \tag{5.5.1}$$

where we obtained an average rotational diffusion time (τ_r) of 3.54 $\pm$ 0.06 s and we noticed that τ_r does not depend on the propulsion speed of the particle, consistent with previous observations for similar systems and as in section 3.2.2. However, after the study present in section 3.2.2 we also obtained a different value for τ_r. Nontheless, I will keep this value in the chapter as originally presented in the paper, but I will also use the other value obtained and present the data in the appendix.

When observing the particle trajectories at the interface, because we have a limited ROI and a linear interface, we can only capture videos for a limited period of their motion at the interface (in obtaining these data, we maximized our ROI in the existing set up and recorded the videos for 120 s). Consequently, not all the particles that we analysed include the incoming and the outgoing phases to/from the interface. We only ascertain that we need to have captured both the incoming and outgoing events if the particle stays at the interface for a time less than τ_r (see Fig. 5.4).

Similarly to our previous analysis we classify our particles based on their propulsion velocities, $0 < v_p < 5$ µm/s, $5 < v_p < 10$ µm/s and $10 < v_p < 15$ µm/s. Among these categories we see that there is no difference in the distribution of retention times at the interface. However, when compared to a scenario of a purely active Brownian particle with no effective interactions with the interface, this retention time distribution is markedly different. In order to perform this comparison, we simulate an active Brownian particle (ABP), represented by a point-mass particles at a position (x, y) and orientation θ with respect to the normal as

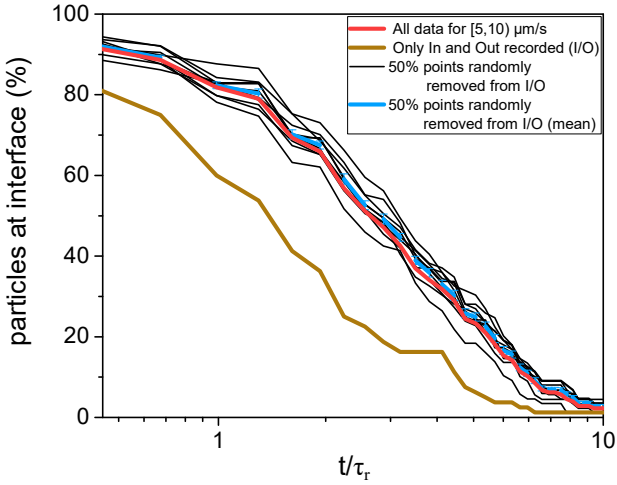


Figure 5.4: Effect of including only particles with incoming (I) and outgoing (O) events travelling for shorter times than τ_r. If one tries to remove videos from where we do not have these events, we make a big bias.

in section 1.4.2, but with reflective boundary conditions at the interface[183],i.e., when the active Brownian particle reaches the interface, it stays along the interface until random fluctuations orient it away from the interface and it moves away into the bulk. We also disregard translational diffusion because experimental results show a clear directional movement close to the interface. We discretize the equations as in ref. [183].

To model our interface, we define a line at x = $\pm$w where if a particle travels beyond this point it is transferred to the nearest x = |w| point. We accept that a particle has reached the interface if the particle position is |x| $\geq$ |w|. We also define the line x = $\pm w_0$ as the nearest point to the wall that defines a region of interest (ROI). We accept that particle has reached the interface once it has contacted it and $\vec{m}$ points to the interface. We regard the particle to have left the interface when it has left this ROI, which is at least 0.2 μm away from the interface to mimic the experimental resolution.

We see from the plot in Fig. 5.5A that at t = τ_r, less than 20% of the self-propelled particles have left the interface in our experiments, whereas nearly all the particles have left the interface in the case of the simulated active Brownian particles with reflective boundary conditions. We plot the distance travelled by the particles at the interface with respect to the particle's bulk persistence length $L_p = v_p\tau_r$ in Fig. 5.5B. In order to capture the longer retention times observed in our experiments with simple Brownian simulations we can model the angular reorientation due to the interface-particle interaction to follow the Hookean form: $\Omega_y = -k(\theta - \theta_0)$[63], where Ω_y is the angular velocity, θ_0 is the stationary angle

of the interface, and $k > 0$ gives the effective restoring torque that stabilises the particle angle along the interface. In fact, in our simulations we use the functional form $\Omega_y = ksin(2(\theta - \theta_0))$ in order to more accurately represent the physical picture of the experiments. Specifically, we take $\theta_0 = \pi/2$, which ensures that for a particle that approaches the interface either at a normal angle or that moves parallel to it, the presence of the interface does not introduce any torques as the system is axi-symmetric. To introduce this torque in our simulation we substitued the $\dot{\theta}$ equation per:

$$\dot{\theta} = \sqrt{2D_r(t)} + ksin\left(2\left(\theta - \frac{\pi}{2}\right)\right) H\left((w_0 - w) \pm x\right), \tag{5.5.2}$$

where H is the Heaviside function, for which the sign of x changes for the lower $(-w)$ and upper $(+w)$ walls.

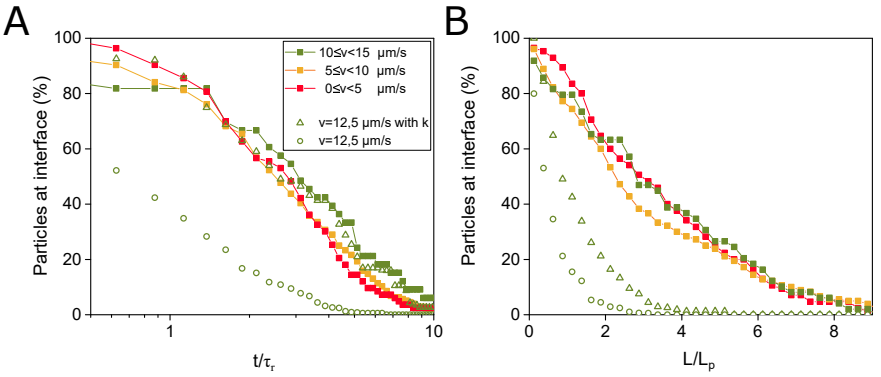


Figure 5.5: Particles align with interfaces. Particles have a bigger residence time associated with the Brownian rotational diffusion time at liquid-liquid interfaces than in the bulk. As a consequence, particles also have a bigger persistence length at these interfaces compared with the intrinsic persistence length that they have in the bulk. Squares: experimental data. Circles: ABP model. Triangles: ABP model with extra torque (**A**)). Particles increase their rotational diffusion time close to liquid-liquid interfaces, but this effect does not depend on their activity. We try the simplest standard model for ABP (circles) but we cannot recover the experimental data. We also add an extra torque to align particles close to interfaces. Modulating the constant associated with the torque, we can recover the experimental results. (**B**)) Particles have a bigger persistence length than expected in the bulk with no difference in their activity. The ABP model with and without torque could not approximate our experimental results.

As soon as the particle rotates, there is an effective restoring torque that aligns the particle with the wall since this is a stable point while the perpendicular orientation is unstable. When the particle begins to rotate towards the interface, a maximum magnitude will be found at $\theta - \theta_0 = 45°$. With this condition at the interface, for $k = -360$ rad/s, the distribution of particle retention times at the interface in

our simulations closely represents the distribution obtained experimentally (green squares in Fig. 5.5A; see also Fig. 5.12 at Appendix 5.C) although it does not in terms of length travelled (Fig. 5.5B).

Effectively, we note that active particles tend to move along liquid-liquid interfaces, similar to the previous observations made for solid-liquid interfaces, and these interfaces can also be used as effective guiding mechanisms for chemically active particles in microfluidic systems.

5.6 | Conclusions

In conclusion, we have presented how liquid-liquid interfaces adjacent to a solid surface can be used for effectively guiding active colloids. The particle in-plane orientation is quenched when in contact with the water-oil interface, similar to what was reported for solid-liquid interfaces. In this case, however, we do not find a dependency of the retention time with particle speed.

In comparison with simple active Brownian simulations the particles in our experiments have significantly longer retention times due to the active alignment effect of the interface. This retention time can be modulated if one introduces an interface-particle interaction, although we did not recover the experimental results for the path travelled. While this alignment effect is likely due to a combination of phoretic and hydrodynamic interactions as we have previously reported in our study with solid interfaces, a further full study is required to understand the possible contributions from capillarity, Marangoni flows, partial diffusion of the chemical product into the oil medium etc. In contrast to some previous studies of active particles at pure fluid-fluid interfaces, where the particle speed was enhanced at the interface, we find a speed reduction at interfaces adjacent to a solid surface[66, 68]. This reduction results from the tendency of the particle to be occasionally oriented into the interface because of Brownian fluctuations.

Our understanding of the behaviour of active particles at liquid-liquid interfaces could pave the way for their application in various microfluid and lab-on-a-chip applications.

5.7 | Future perspectives

After publishing the previous results, I still worked experimentally with similar systems. However, the experiments were not reproducible, and I finally stopped working on these systems. Here I present two of the projects related and ideas for future approaches.

5.7.1 Curvature effect

The experiments carried out in this chapter used linear liquid-liquid interfaces. However, it would be interesting to repeat these experiments with curved interfaces. These experiments could help us to understand better the possible contribution of each force present in the system. For instance, Desai N. et Al [184] and Saverio E. et Al. [185] showed that hydrodynamics can retain or scatter particles at circular interfaces depending on their curvature, and a characteristic radius for the droplets define how particles will behave. But phoretic interactions could change these interactions and hence, we could extract more information from the results we already obtained. Simmchen J. et Al. [56] already studied these particles at circular interfaces, but in their case these were solid-liquid, which their nature is different from the liquid ones. Moreover, solid interfaces can have imperfections not visible for us but that could help particles to escape from them. Hence, this effect could interfere in the results obtained. But in the case of the liquid-liquid interfaces, there are not imperfections, and therefore we would avoid this problem.

To build curved liquid-liquid interfaces I decided to make droplets of oil. However, natural, and artificial oils such the silicon oil already used for flat interfaces are less dense than water, and they would move to the top of the setup due to density mismatch with water. Instead, I created immiscible droplets by mixing Tributyrin with water at a ratio 1/10 and shaking the solution by hand for a few seconds [38]. These droplets are stable for long times and sediment. Hence, particles could interact with the droplets. In general, droplets were moving over the surface as the particles were, but in this case, they moved due to external currents and/or Brownian motion. Sometimes, these droplets could attach to the solid surface, changing their appearance (See Fig. 5.6). However, this was a rare event to see, and only occurred on a few occasions even the experimental process carried out was the same always. To control better these events, I tried to modify the surface by applying an O_2 plasma for a few minutes or sylanizing the surface. These processes can improve the fixation of the droplets, but the control was neither possible. For instance, these methods could make my droplets to splash more the surface.

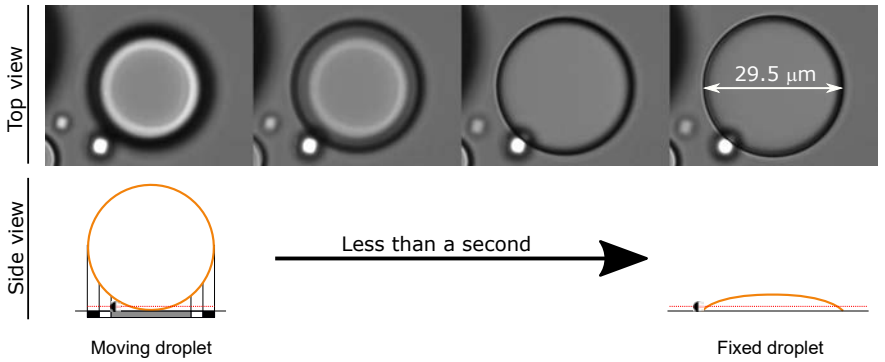


Figure 5.6: Droplets under the microscope. In general, droplets are moving, and present this black and white shade around them. In a few cases, droplets can attach to the surface. In this moment, the shade disappears, and we can see the droplet with a different contrast. This process is very fast and lasts for less than a second. Because of the change in geometry, we could expect particles to behave differently in contact with both cases. Particle at scale (R=1μm). Red dotted line: Plane focused.

When I observed the particles interacting with these droplets, I could see two completely different behaviours, that I related on the fact of the droplet being fixed or not to the substrate surface (See Fig. 5.6). However, when I obtained more samples with fixed droplets using the previous methods, this hypothesis failed. Instead, I could observe particles interacting with fixed droplets as if droplets were moving too. Hence, I was not able to understand when these two scenarios can happen, and what is the difference that makes them appear.

Particles trapped in oil droplets

The most interesting scenario happened with droplets fixed spontaneously, without doing the plasma or sylanizing. In this case I could see that 2 μm diameter size particles could move around the droplets for long times compared with their τ_r, and the size of the droplet was not a problem for this behaviour. I could see particles moving around smaller, the same size and bigger droplets than themselves for minutes. For instance, once I could record a particle moving in circles around the droplet for 10 minutes, but at some moment I had to stop recording, even the particle was still moving around the droplet. This time is two orders of magnitude bigger than τ_r. Nonetheless, it is also true that some of these particles could have something at their surface, since optically I could see something on a few of them as if they would have had oil attached to their surface, but this was not always true neither. Therefore, the surface modification of particles could have an influence on the behaviour of the particles next to these interfaces.

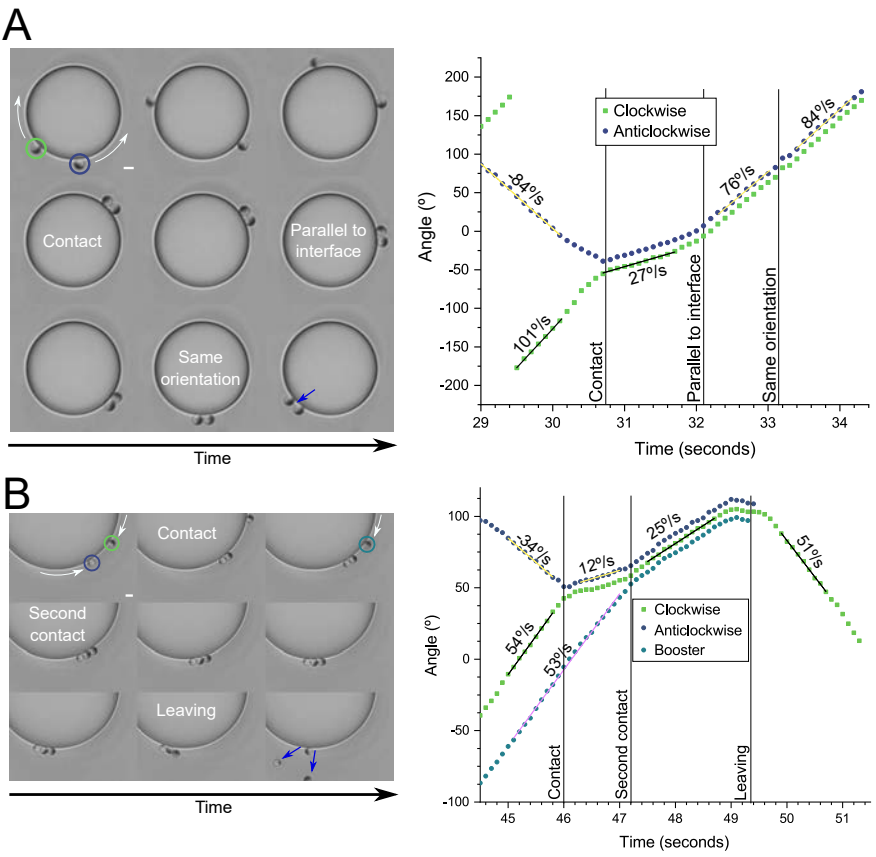


Figure 5.7: Particles contacting fixed droplets. **A)** Two particles contacting a fixed droplet turn around it with opposite sense of motion. At time 30.8 s after starting the video both particles contact. The slowest particle changes its sense of motion and starts moving in the same sense as the other particle with the same speed module as it had before the contact. While this particle is changing its orientation, the speed of the pack of particles drops, but as soon as the particle orients parallel to the interface, the package speeds up. Blue arrow: Gap between both particles. **B)** If we add a second particle (booster) and there is not a change of sense of motion, we can see that the addition of a second particle in the same sense of motion to one of them speeds up the package made of 3 particles. Blue arrow: particles leave the droplet. White arrows indicate the sense of motion of particles. Scale bar: 2 μm.

When these particles move around fixed droplets, it is common to see multiple particles moving around the same droplet (See Fig. 5.7). There are multiple phenomena that we can observe in this situation. For example, particles can collide if they are coming from different sense of motions (See Fig. 5.7A)). Depending

on the strength of each particle, after collision, the system of particles will move in one sense if one particle moves faster than the other one. But if both particles have similar strengths, they will face with their silica face together and will bounce slightly from an equilibrium point as if they were a spring. At this moment, a particle can change its orientation by 180º. During this process, both particles will move at constant speed until the silica part from the particle changing its orientation is facing totally to the oil droplet. Just now, its resistance to move will decay, and the other particle will push the system stronger in one sense of motion.

After the particle changes its orientation by 180º, the system will evolve at a different speed. As seen in Fig. 5.7A), the package of particles do not necessary move at the speed of the sum of the individual speeds, but with the speed of the front particle. The reason that explains this phenomena is because particles are not in solid contact, but there is a gap between them, caused by the products generated in the cap of the particle as already recognized in the literature [65]. In this case, both particles had similar speeds when they moved freely, and the gap prevents the addition of speeds. However, there are other situations where speeds can sum, and the package of particles can move faster if there are more particles. As seen in Fig. 5.7B), we start the experiment with a similar configuration, where one particle moves faster than the other one and after collision the package moves in one sense of motion. In this case, the particle does not reorient, and all the silica cap is facing towards the other particle, creating the maximum possible of resistance. If another particle comes in the same sense of motion of the package, and it is much faster than the package of two particles, the particle coming from behind can push the other two particles, increasing the speed of the package. This phenomenon is also visible if particles in the front move much slower than the one pushing from behind, no matter how they are oriented.

Additionally, when we have more than one particle, it is possible that the reorientation happens in a particle not in the borders of the package. These phenomena often occurs if all the package moves around an equilibrium point because the strengths towards both senses are similar. Particles next to the one changing its orientation can feel the products generated by this particle, creating an extra torque that can change their orientation and even leave the droplet as detected in Fig. 5.7B).

Particles scattering oil droplets

The second and most general behaviour I could notice was specially found when particles contact moving droplets (See Fig. 5.8A) for a scheme of the process). Thus, the analysis of this experiment was not easy, and I had to develop the tools presented in section 4.1.5. In this case, after particles contacting the droplets, they made less than half a turn around the droplet and then escape. This second behaviour was tested with different droplet sizes, from particles size to 6-7 times their size (See Fig. 5.8B)). Overall, once particles touch the droplet, they move along

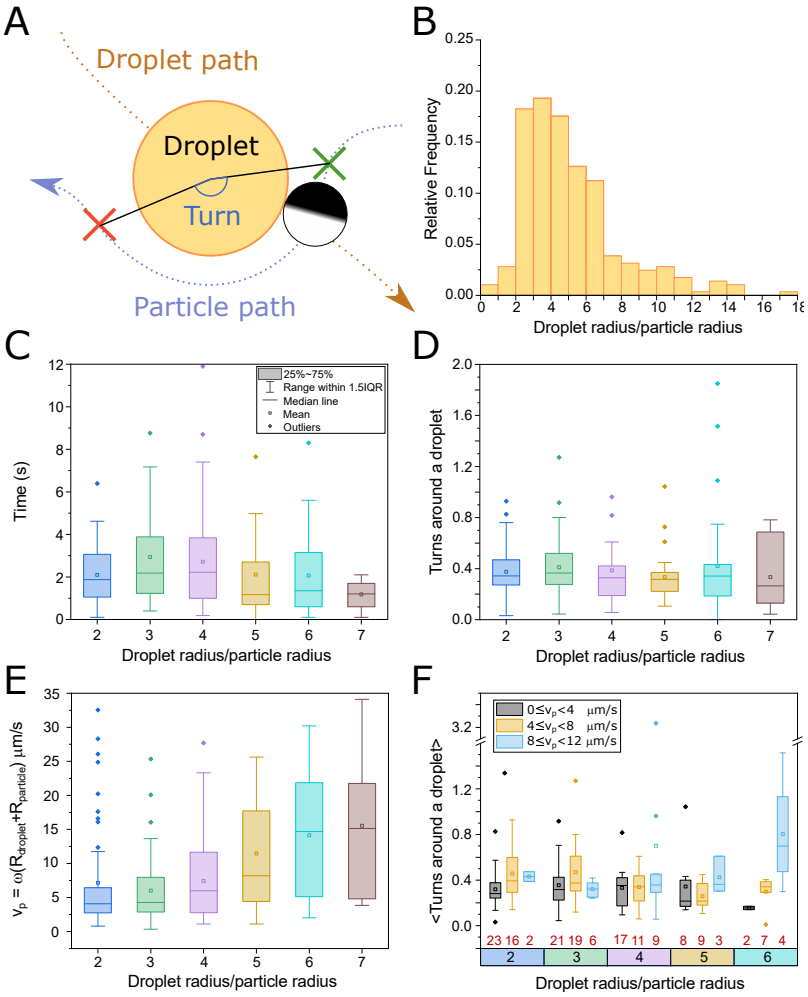


Figure 5.8: Particles contacting moving droplets. **A)** A particle contacting a moving droplet turns around it for a specific angle, and then escapes from the droplet. Crosses are the moment of contacting (green) and escaping (red). Particle motion is always tracked from the droplet frame of reference. **B)** Relative frequency of droplets sizes. Most droplets radius is between 2- and 7-times particles radius. **C)** Average time of particles moving around droplets. In average, particles move for two second along the interfaces. Error is big enough to show time $\not\propto$ droplet size. **D)** Average number of turns that particles do around droplets. In average, particles move for 0.35 turns, which are 130º, and neither depend on the size of the droplet. **E)** Linear speed for particles while moving around the droplets. ω is the angular speed for each particle (angle travelled divided by the time spent). R_i is the radius for the droplet or particle. Particles moving around bigger droplets move faster. **F)** The turns do neither depend on the particle speed, but the number of particles for each case (in red) is very reduced, and the error high.

the droplet for a couple of seconds (See Fig. 5.8C) to turn less than half the droplet (See Fig. 5.8D)) before escaping from the droplet (about 130º in average[2]). These measurements, time and turns, do not depend on the droplet size. If we measure the angular speed of these particles (angle travelled divided per time) and extract the linear speed (angular speed per the sum of the droplet and particle radius), we see that for higher droplets size, particles move faster around the interface. This behaviour is expected after seeing that particles do the same turn and stay for the same time independently on the size of the droplet, but the physics underlying is not clear[3]. To complete the view, I tried to divide the number of turns depending on size of the droplet and the speed of the particle (See Fig. 5.8F)). However, here the dataset is reduced (in red the number of samples taken for the mean), and the data should be read carefully. In fact, here I only present data up to 12 μm/s, but if we see at Fig. 5.8E) we can see that the population for droplet size/particle size $= 5$ and 6 is over this limit. This data does not show a clear dependency of the speed for the turns done by particles, but more data should be included since the dispersity is elevated for all the data. In any case, almost all particles do less than a turn around the droplet for a couple of seconds.

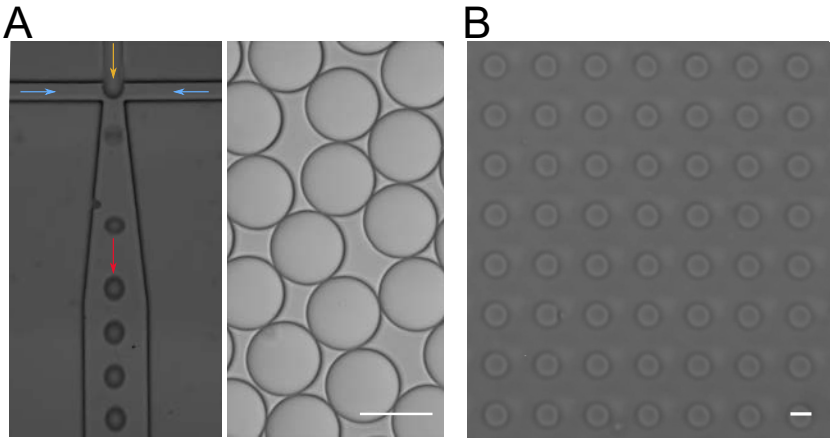


Figure 5.9: Alternative setups to study droplets system. **A)** A microfluidic chip can create droplets of oil, but it needs a surfactant to avoid all droplets to combine after its creation. Arrows indicate direction of flow. Yellow: oil. Blue: Water with surfactants. Red: Final output. Scale Bar: 60 μm. **B)** An array of holes can be filled with oil, creating oil droplets of the same size. Scale bar: 5 μm.

To increase the dataset, we should create droplets of the same size per experiment instead of random sizes every time. To create these droplets I tried to create particles using a microfluidic chip (See Fig. 5.9A)). However, droplets generated using this microchip needed a surfactant to avoid droplets combining

[2]Notice that in these experiments measurements are highly scattered.
[3]Notice that this speed is in the droplet frame of reference.

between them. We should avoid using surfactants because they introduce more physics, and particles move slower. Even they can stop [186]. Hence, I stopped using these devices. A second approach was to create holes in an array following a lithographic process and posterior PDMS moulding (See Fig. 5.9B)). With the holes in the PDMS, we could spread oil, wash with water, and expect to have oil in the holes. This second approach could be useful, but it was not possible to do these experiments due to lack of time.

Given these results, more research in this line is needed. As a step forward, it could be interesting to use the array of holes to see the collective behaviour of these particles. For example, if some phenomena of synchronization could appear between different droplets nearby.

5.7.2 Surface modification

The modification of Janus particles or interface surface can change the interaction of active particles at interfaces. For instance, silica surface of the particles can be modified by attaching different chemical groups, which can change its wetting properties. Moreover, liquid-liquid interfaces can present chemical molecules at the interface such as surfactants or other molecules that could stabilize the droplets in water.

Consequently, one future project to explore could be the modification of either particles or interfaces. From the interfaces side, we started making droplets using BSA protein with gold nanoparticles to stabilize them, but we finally discarded this technique because these droplets were stable enough by themselves without the BSA and because particles in water were moving slower since of the BSA used for the mixture. However, after adding the BSA to the droplets, particles could not move around the droplets. In fact, when particles where at a particle diameter from the droplet, it was very easy to see how they were being adsorbed by the oil, making particles to stuck at the interface. Hence, these modifications could be interesting to further study in next projects.

Appendix

5.A | Particle orientation next to interfaces

To analyse the orientation of particles next to interfaces we fit the data obtained at section 5.3 with a Gaussian distribution. In the case of low oil viscosity, the data seemed not symmetrically distributed, and hence we also fitted it with a skewed normal distribution function. The skewed normal distribution includes the possibility of having an asymmetrically (or skewed) distributed dataset. To determine if the distribution is skewed, the distribution includes the shape parameter α. Positive values for α indicate that the distribution has a longer tail towards higher values, but lower values are more important, while negative values indicate the opposite. A zero value means that the population is symmetric, and hence we obtain a simple Gaussian distribution.

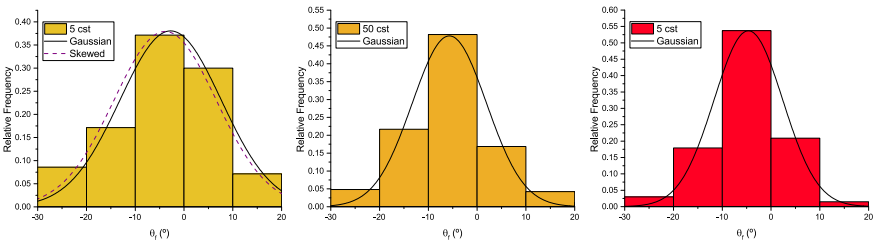


Figure 5.10: Particle orientation change after they contact the interface. Particles obtain a stable orientation angle no matter on the oil viscosity. We fit our experimental data to Gaussian distributions. All of them are compatible with a mean value of -6$\pm$10º indicating no clear preference for the particle to point into or away from the oil phase. Since data for 5 cst could seem left-skewed, we also fitted it to a skewed normal distribution. Value of the shape parameter is $\alpha = 0.25$, and therefore we do think that still is compatible with a normal distribution.

5.B | Speed distribution

We present in Fig. 5.11 the relative frequency for the ratio between particle speed at the bulk (v_{bulk}) and the interface (v_{int}). We split the data in bins of particles moving at the interface at different speeds. This data correspond to the averages present in Fig. 5.3.

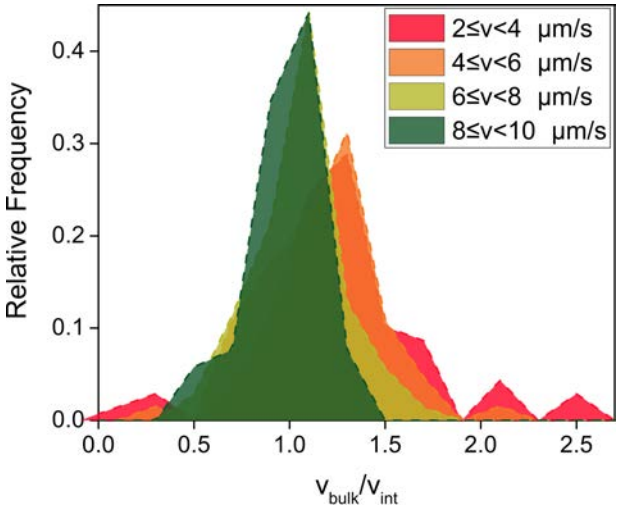


Figure 5.11: Speed distributions. We present speed distributions for each bin of Fig. 5.3C.

5.C | ABP with different spring constants

We present in Fig. 5.12 a comparison of different k spring constant to fit experimental data in Fig. 5.5

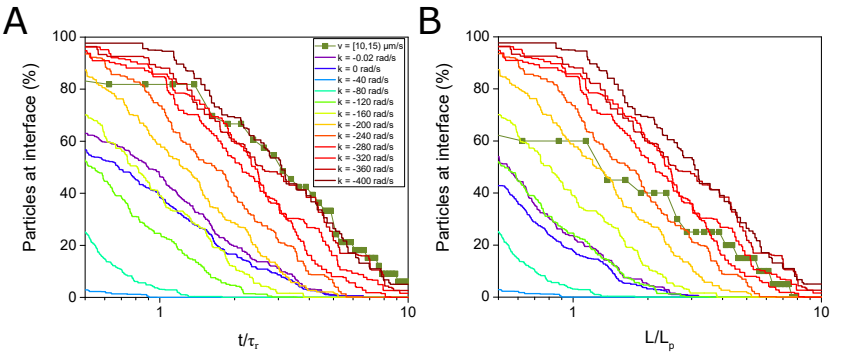


Figure 5.12: Comparison for spring constant determination. We present how our model fits for our experimental values with different k values. We compare data only to experimental values for $10 < v_p < 15$ μm/s. ***A***) We found a good agreement for retention times using $k = -360$ rad/s. ***B***) When we look for length values, data for $k = -360$ rad/s was not the best fit, but $k = -240$ rad/s was.

5.D | On the update of τ_r for the results obtained in the paper

In section 3.2.2 we learnt that depending on the method to extract the τ_r we can obtain two statistically different values. In the original paper I extracted this method using the autocorrelation, which gave me the smallest value between both values obtained. However, if we rebuild Fig. 5.5 using the larger value, we can still see the same effect. Particles stay more time than expected and travel longer straight trajectories than expected. Hence, we do not see that the update of the τ_r value can change the results shown in the original paper.

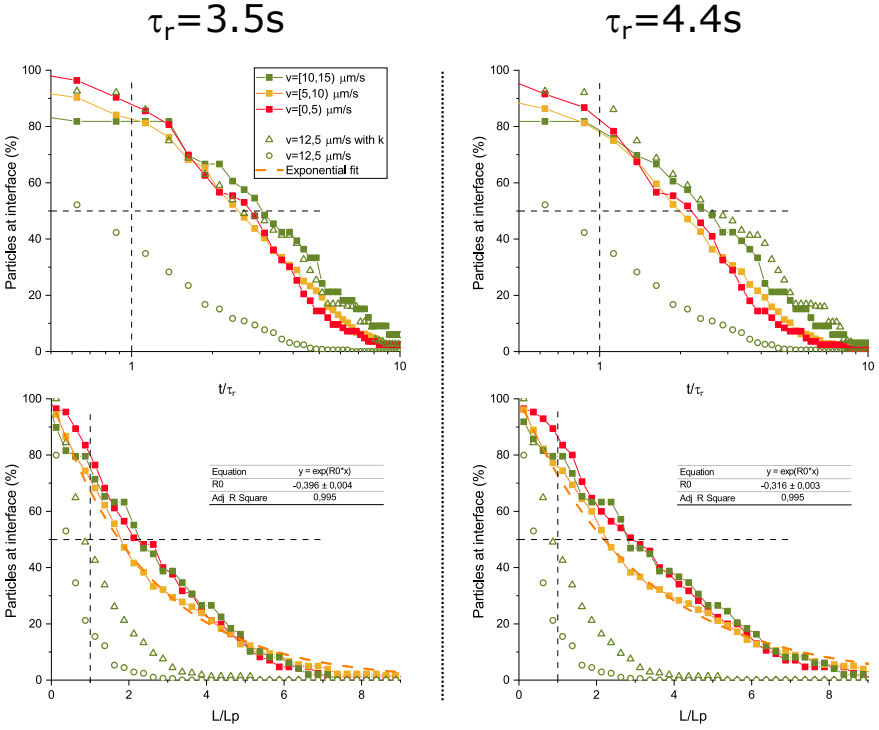
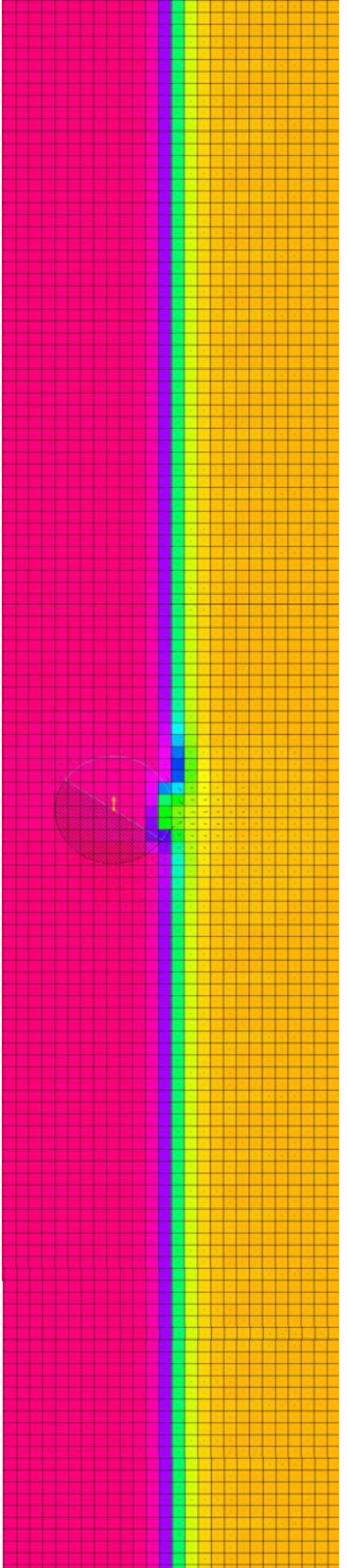


Figure 5.13: Comparison for particle retention at the interface and length travelled during the retention using both values for τ_r obtained in section 3.2.2. The lowest value was used in the original paper but results still are compatible: particles stay longer than expected at interfaces and travel in straighter trajectories than expected while moving next to them.



6

A lattice boltzmann model for self-diffusiophoretic particles next to liquid-liquid interfaces

"Sometimes the wheel turns slowly, but it turns."

Lorne Michaels

The analysis of the interaction of active particles at liquid-liquid interfaces requires of a powerful model capable of introducing interactions from active particles and from the medium. A very naive model was presented in the previous chapter, and as a following step, in this chapter I model the motion of chemically powered phoretic active colloids close to liquid-liquid interfaces by means of a hybrid finite differences-Lattice-Boltzmann method. The study presented here includes a comparison between capillarity and diffusiophoretic forces for particles, inactive or active, that are or approach to liquid-liquid interfaces. Moreover, this model allows to accumulate differently the product generated by particles at both sides of the interface.

For inactive particles trapped at the interface, diffusiophoretic forces are important when the wetting is neutral, and its behaviour depends on the surface mobility of the particle and the accumulation of solute in both sides of the interface. When activity is added, particles align when they are close to the interface and can move along interfaces as previously observed in the previous chapter. Thus, we recover the torque we introduced in the previous simple model naturally.

The present chapter is based on the paper "A Lattice Boltzmann model for self-diffusiophoretic particles next to liquid-liquid interfaces", from which I am first author, and to which I developed the model, analysed and wrote the paper. I would like to special thank Andrea Scagliarini, who helped me working with this project.

6.1 | Introduction

In the previous chapter we observed the behaviour of active particles interacting with liquid-liquid interfaces. These experiments demonstrate that particles experiment a torque when they contact the interface which quenches their director vector almost parallel to the interface's contact line. This new alignment makes particles to move along the interface, although they move slower as compared as when they swim in the bulk. To describe these effects, we based our theoretical comprehension on an ABP model with an ad hoc torque added. Even that this simple model could reproduce the new alignment of particles and their increase in retention time next to interfaces, it could not reproduce other effects such as the longer paths travelled along the interface. Moreover, the model was very naive and did not include any of the forces present at the interface. For example, our particles will experiment a wetting and a phoretic force. While in the experiments the wetting property was constant, the change in the wetting and phoretic attraction/repulsion can modify radically its behaviour at the interface. For example, the addition of BSA to the surface of the particles attached them to the interfaces. On the other hand, a change in phoretic activity can also change its behaviour next to interfaces. Slower particles in the bulk were decreasing more their speed while travelling along the interface than faster ones. These couple of examples demonstrate the importance of having a model where these interactions are present, and where other forces inherent with these interfaces such as Marangoni flows are also present.

To answer many of the questions that can arise from the interaction of these particles with free liquid-liquid interfaces, here we implement a hybrid Lattice-Boltzmann method to model active particles that move due to a self-diffusiophoretic motion in a media with liquid-liquid interfaces. In our model, liquid-liquid interfaces are not static, and particles have a natural wetting. Furthermore, our particles create solutes asymmetrically to move. The region of the particle that creates the solutes is easy to change, and both the region that creates or not these solutes can have different interaction with them. Furthermore, the solute created can accumulate more in one phase of both liquids, which can be useful to study different combinations of immiscible fluids as solutes can accumulate differently depending on the combinations of fluids used. This last effect could not happen with a solid-liquid interface and could modify the behaviour of active particles interacting with liquid-liquid interfaces as compared with solid-liquid interfaces.

With this model we aim to study the importance of the capillarity forces arising from the wetting in contrast with the active motion of our particles. Moreover, we open the possibility to study this system under different regimes where the concentration of the solute created by our particles accumulate differently in both sides of the liquid phases. Although the current model has been tested only for one particle problems, the algorithm is fully implemented to run with as many particles as computational power there is, and each particle can have different configurations.

6.2 | Computational model

A suspension of active particles in presence of liquid-liquid interfaces consists of a fluid phase (solvents + solute) and a solid phase (the active particles). To model such a multiphase (and multicomponent) system we resort to a mesoscopic approach, based on the lattice Boltzmann (LB) method [187, 188] in the phase field formulation [189, 99].

6.2.1 Phase field model: the free energy functional

The fluid phase is a ternary mixture made of two immiscible liquids (say, water and oil) and a solute, which is the product of the reaction occurring at the catalytic site on the particle surface (the oxygen). We associate to the water-oil system a scalar field $\psi(\mathbf{r},t)$ standing for the local composition, that is $\psi = \frac{\rho_W - \rho_O}{\rho_W + \rho_O}$, where ρ_W and ρ_O are the density fields of water and oil, respectively. As in the standard Cahn-Hilliard theory, the thermodynamics of the oil-water mixture is controlled by a quartic in ψ double-well free energy density of Landau type, $f_{\text{ow}}[\psi] = \frac{A}{4}\psi^4 + \frac{B}{2}\psi^2$ (with $A > 0$ and $B < 0$). This free energy has to be extended to embrace the dynamics of the oxygen, that is, in principle, miscible with each of the two other components; therefore we need to add a term characterized by a single minimum that disregards the energetic cost associated to the concentration gradients [112], such that, in the case of a single component solvent ($f_{\text{ow}} = 0$ identically), a diffusive equation for the solute is recovered. A simple parabolic potential is appropriate to this aim (as we will show shortly), namely $f_{O_2}[\phi] = \frac{C}{2}\phi^2$ ($C > 0$), having introduced the field $\phi(\mathbf{r},t)$. In actual systems, though, the oxygen may display, in general, a greater affinity for one of the two liquids (i.e. it can be more soluble in water than in oil, or vice versa). To account for this preferential concentration an "interaction" term, coupling ϕ and ψ, has to be included. We propose to do so by simply shifting the global minimum of $f_{O_2}^{(0)}$ in $\phi = 0$ to a ψ-dependent minimum, i.e.

$$f_{O_2}^{(0)}[\phi] = \frac{C}{2}\phi^2 \rightarrow f_{O_2}[\phi,\psi] = \frac{C}{2}\left(\phi - \phi_0(\psi)\right)^2 . \tag{6.2.1}$$

For $\phi_0(\psi)$ we choose the form $\phi_0(\psi) = \phi_{b0} + E\tanh(\psi)$, where the parameter E tunes the oxygen solubility and ϕ_b sets the average oxygen concentration. The full free energy functional then reads:

$$F[\psi,\phi]=\int \mathbf{dr}\left[\frac{A}{4}\psi^4+\frac{B}{2}\psi^2+\frac{\kappa}{2}|\nabla\psi|^2+\frac{C}{2}(\phi-E\tanh(\psi)-\phi_{b0})^2\right]. \tag{6.2.2}$$

Hereafter we set $B=-A$, such that the ψ minima, corresponding to the bulk water and oil phases, are located in $\psi=\pm 1$. The minimization of the functional (6.2.2) yields the chemical potentials $\mu_\phi=\frac{\delta F}{\delta\phi}$ and $\mu_\psi=\frac{\delta F}{\delta\psi}$, i.e.

$$\left.\begin{aligned}\mu_\phi &= C(\phi-E\tanh(\psi)-\phi_{b0})\\ \mu_\psi &= A(\psi^3-\psi)-\kappa\nabla^2\psi-C\,E\frac{(\phi-E\tanh(\psi)-\phi_{b0})}{\cosh^2(\psi)}\end{aligned}\right\}. \tag{6.2.3}$$

The dynamics of the ternary mixture system is, then, described by the following advection-diffusion-reaction equations:

$$\left.\begin{aligned}\partial_t\phi+\nabla\cdot(\boldsymbol{u}\phi) &= D_\phi\nabla^2\left(\phi-\phi_0(\psi)\right)+\mathcal{D}_\phi-k_d(\phi-\phi_0(\psi))\\ \partial_t\psi+\nabla\cdot(\boldsymbol{u}\psi) &= D_\psi\nabla^2\left(\psi^3-\psi-\frac{\kappa}{A}\nabla^2\psi-E\frac{\phi-\phi_0(\psi)}{\cosh^2(\psi)}\right)\end{aligned}\right\}, \tag{6.2.4}$$

where $D_\phi=C\,M_\phi$ and $D_\psi=A\,M_\psi$ are the diffusivities for the oxygen and the water-oil mixture, respectively, and M_ϕ and M_ψ are the mobility constants for ϕ and ψ. The source term, $\mathcal{D}_\phi$, differs from zero only at the active site on the particles surfaces and accounts for the continuous generation of a product (e.g., the oxygen) in the associated catalytic reaction. This production needs to be balanced by a sink term, $k_d(\phi-\phi_0(\psi))$, in order to allow the attainment of a steady state. Physically, the sink mimics the degradation of the production or its loss in the environment [112]. Fig. 6.1A) displays the effect of changing the solubility parameter from negative to positive values, by plotting the average oxygen concentration, $\langle\phi\rangle_\psi$ in oil or water as a function of E, at equilibrium and in the absence of particles (hence of oxygen production). Since, by virtue of Eq. (6.2.3), the equilibrium profile of ϕ is $\phi=E\tanh(\psi)$ (the background value having been set to zero here, $\phi_{b0}=0$), the average is $\langle\phi\rangle_\psi\approx\pm E\tanh(1)$ (depending on whether the oil, $\psi>0$, or water, $\psi<0$, phase are considered), which is reported in Fig. 6.1A) with the solid lines and agrees well with the numerical data.

If the background value is not set to zero, $\phi_{b0}\neq 0$, we recover the equilibrium values only if ϕ_b is present in Eq. 6.2.2 (See Fig. 6.1B)). Moreover, the selection of ϕ_{b0} also modulates the asymmetric concentration of ϕ in both liquid phases. For a given value of E, and if $\phi_{b0}>0$, we can have a state where the equilibrium value for ϕ is positive in one phase and negative in the other, or positive in both phases.

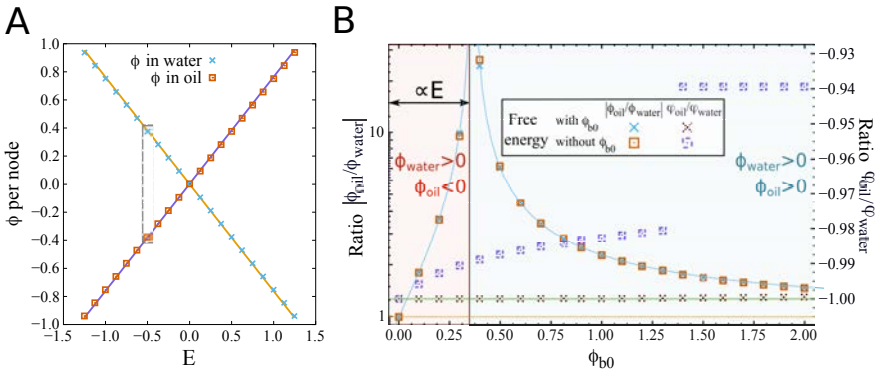


Figure 6.1: Equilibrium values for order parameters as a function of E or ϕ_{b0}. A) When E changes, the equilibrium values for ϕ vary linearly, as expected. Here we show the value for ϕ in water (blue crosses) and oil (orange square) bulk phases. These values fit with the expected values from the thermodynamic model, both for ϕ in water phase (yellow line) and for oil phase (violet line). Thus, by modifying E we can have a greater affinity of ϕ for one of the two liquids (equilibrium phases of ψ). In these simulations $\phi_{b0} = 0$. Changing ϕ_{b0} at constant $E = -1/2$ (**B**)) the system behaves as expected theoretically. Nonetheless, to fit the values with the equilibrium values (blue and green lines), we need to add the parameter ϕ_{b0} in our free energy model. This inclusion is necessary to stabilize the values of equilibrium for the binary mixture at ± 1. Depending on the initial ϕ_{b0} we distinguish two different areas: one with positive ϕ in both sides, and another with a positive and negative value of ϕ, one for each phase. The higher is the E parameter, the bigger will be this region.

6.2.2 Suspended Active Particles

Particles are modelled as solid spheres defined by a set of boundary "links" between inner and outer nodes. The fluid-solid coupling is realized by means of the so called "bounce-back-on-links" algorithm that guarantees the proper momentum-torque exchange between particles and solvent [101, 190, 191, 192]. Thus, we modelled the particles as introduced in section 1.4.4. We also follow the description given at section 1.4.4 and introduced the colloidal phoresis by imposing at the particle surface an effective slip velocity profile which depends on the local solute concentration [193] as:

$$\boldsymbol{v}_s = \mu(\boldsymbol{r}_s)(\mathbf{1} - \hat{\boldsymbol{n}} \times \hat{\boldsymbol{n}}) \cdot \nabla\phi, \tag{6.2.5}$$

where $\boldsymbol{r}_s$ is a point on the surface of the particle, $\hat{\boldsymbol{n}}$ is the normal to the surface in $\boldsymbol{r}_s$ and $\mu(\boldsymbol{r}_s)$ is the phoretic mobility at $\boldsymbol{r}_s$, which carries the molecular details of the solute-colloid interaction[193]. As a consequence, in the presence of concentration gradients, particles gain a net propulsion velocity $\boldsymbol{V}_p \sim -\mu\nabla\phi$ (for uniform phoretic mobility $\mu(\boldsymbol{r}_s) \equiv \mu$), i.e. if $\mu < 0$ they are attracted by the solute, else if $\mu > 0$ they are repelled. To achieve self-propulsion, particles are,

then, endowed with the property of generating solute [117]; this is done by simply adding a production term that injects ϕ with a given rate at nodes neighbouring the particles surfaces, thus modelling the catalytic activity of Pt-coated colloids. In particular, the following Janus activity profile is chosen:

$$\Pi(\boldsymbol{r}_s) = \begin{cases} \alpha & \text{if } \hat{\boldsymbol{m}} \cdot \hat{\boldsymbol{n}} \leq 0 \\ 0 & \text{if } \hat{\boldsymbol{m}} \cdot \hat{\boldsymbol{n}} > 0 \end{cases}, \tag{6.2.6}$$

where α is the constant production rate and $\hat{\boldsymbol{m}}$ is the particle characteristic unit vector (see the sketch in Fig. 6.2A)). Notice that the superposition of such activities associated to the various particles is precisely what gives rise to the production term $\mathcal{D}_\phi$ appearing in Eq. (6.2.4). Because of the sink term appearing also in Eq. (6.2.4), this production term extends for a characteristic length L, which depends on a degradation ratio κ_d and its diffusion constant D_ϕ:

$$L = \sqrt{\frac{D_\phi}{k_d}}. \tag{6.2.7}$$

If this length scale is larger than any other relevant scale in the system, the chemical degradation will have a minimal impact in the system dynamics.

Analogously, for the phoretic mobility $\mu(\boldsymbol{r}_s)$ we set:

$$\mu(\boldsymbol{r}_s) = \begin{cases} \mu_A & \text{if } \hat{\boldsymbol{m}} \cdot \hat{\boldsymbol{n}} \leq 0 \\ \mu_I & \text{if } \hat{\boldsymbol{m}} \cdot \hat{\boldsymbol{n}} > 0 \end{cases}, \tag{6.2.8}$$

where, with the subscripts 'A' and 'I' it is meant that the associated mobilities correspond to the active and inactive hemispheres, respectively. For an isolated Janus particle with the above activity and mobility profiles we expect a motion with constant velocity of magnitude [116, 117, 118]

$$v_p = \frac{|(\mu_A + \mu_I)|\alpha}{8D}. \tag{6.2.9}$$

In the presence of interfaces, a specific treatment of the interaction of the two liquids with solid boundaries, which determines the particles wetting properties, needs to be included. To this aim, we follow the description given at section 1.4.4 and added an extra boundary term to the free energy functional, such that:

$$F_{\text{tot}}[\phi, \psi] = F[\phi, \psi] + \int_S H\psi(\boldsymbol{r}_s) d\boldsymbol{r}_s \tag{6.2.10}$$

where the integral is over the solid surface. The parameter H controls the wetting through the following boundary condition, that can be derived by minimization of the surface term in (6.2.10),

$$H = \kappa |\nabla\psi \cdot \hat{\boldsymbol{n}}| \tag{6.2.11}$$

and, therefore, it sets the particle contact angle θ, to which it is related by [98]

$$cos(\theta) = \frac{1}{2}\left[-(1-h)^{3/2} + (1+h)^{3/2}\right] \tag{6.2.12}$$

where $h = H\sqrt{1/(\kappa A)}$.

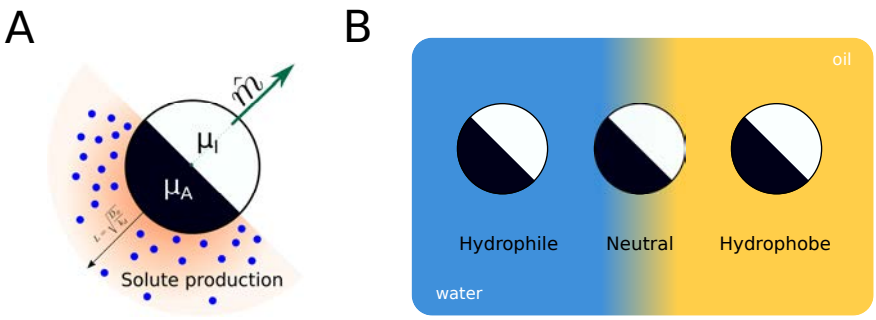


Figure 6.2: A) Sketch of a spherical self-diffusiophoretic particle. The particle is characterized by a Janus profile of both the activity (it produces solute only over one hemisphere) and the phoretic mobility, whose value is μ_A over the active cap and μ_I otherwise. **B)** Active particle in the presence of an interface: depending on the value of the contact angle, θ, it tends to stay preferentially in the bulk of the water phase ($\theta > 90°$), of the oil phase ($\theta < 90°$) or absorbed at the interface ($\theta = 90°$).

6.2.3 Numerical details

All simulations were calculated either by simulating none or one particle of radius $R = 4.5$ lattice spacing. The size of the particles is small compared by the lattice size (32^3 lattice spacing for speed and degradation tests and $64x32x32$ or $64x64x32$ lattice spacing for the rest). Simulations run at least for 60.000 lattice time steps. We study different regimes values, but some were constant. See table 6.1 for different values used in our simulations.

We first started our simulations by creating the liquid-liquid interfaces. Order parameters were set to $\phi = 0$ for the whole simulation box and ψ followed as 3 blocks of $\psi = +1$, $\psi = -1$ and $\psi = +1$, with a smooth transition between phases given by an hyperbolic tangent. Next, we let the system to reach the equilibrium, until we got the stable distribution for different values of E parameter. These stable configurations were used each time a simulation with a Janus was started, except for the case of speed and degradation studies, where just one phase was introduced with the corresponding ϕ at equilibrium value. Particles were initialized with zero speed.

Table 6.1: Parameters used for simulation

Parameter	Symbol	Value
Simulation box size	L_x x L_y x L_z	min: 32 x 32 x 32 max: 64 x 96 x 96
Time steps	t	60.000 to 300.000
Fluid kinematic viscosity	ν	0.167
Fluid density	ρ_f	1.0
ϕ diffusive	M_ϕ	0.4 and 0.8
ψ diffusive	M_ψ	0.05 to 3.2
Parameters free energy	A, A'	0.0625
Parameters free energy	κ, E	0.04, 0.0625, -1.0 to 1.0
Particle radius	R	4.5
Surface activity	α	0 to 0.01
Degradation rate	k_d	0.00001 to 0.01
Phoretic mobility inactive side	μ_I	0.00 to 0.60
Phoretic mobility active side	μ_A	0.00 to 0.60
Gravity force	F_b	0.0

6.3 | Results and discussion

6.3.1 Motion of a Janus particle in free media

For the sake of validation of the model, we first consider the motion of an isolated Janus in the bulk of a single phase fluid.

The motion of Janus particles will depend on the media properties, characterized by the thermodynamic constants for the medium described in table 6.1 but also, on several colloidal properties. These properties are the ratio of creation of the product per node in one face of the particle, α, and the surface mobilities μ_A and μ_I. The more product they create by time step, the fastest the product will accumulate in the media. Hence, a degradation of ϕ is also introduced, where for each time step and node its ϕ value is checked and a quantity proportional to its distance from equilibrium ($\phi - \phi_0$) is removed. This degradation is modulated by a proportional constant k_d. The presence of a linear degradation term implies that the concentration field does not decay purely algebraically with the distance from the source (the particle surface) but it is modulated by an exponential factor, i.e. $\phi(r) \sim e^{-r/\ell}/r$, with *screening length* $\ell = \sqrt{D/k_d}$. In the remainder of the chapter the value of the degradation rate is kept fixed to $k_d = 10^{-3}$, which gives a screening length of approximately one particle diameter, $\ell \approx 2R$. Fig. 6.3A) displays the change of the global average of ϕ when considering a Janus particle in a single fluid phase, as a function of the activity and degradation rate.

The steady state values are used to identify reference parameters for the average ϕ and k_d. Fig. 6.3B displays the modulation of the Janus particle speed when modifying its surface interaction with the product. If both mobilities are zero, the particle cannot interact with the product and hence their speed vanishes. However, once the surface mobility is not zero, even if the mobility for the active side is zero, the particle will interact with the product and will start to move. Because we focus in the low Péclet number regime, the increase of these mobilities causes a linear increase with the particle speed [116, 117, 112].

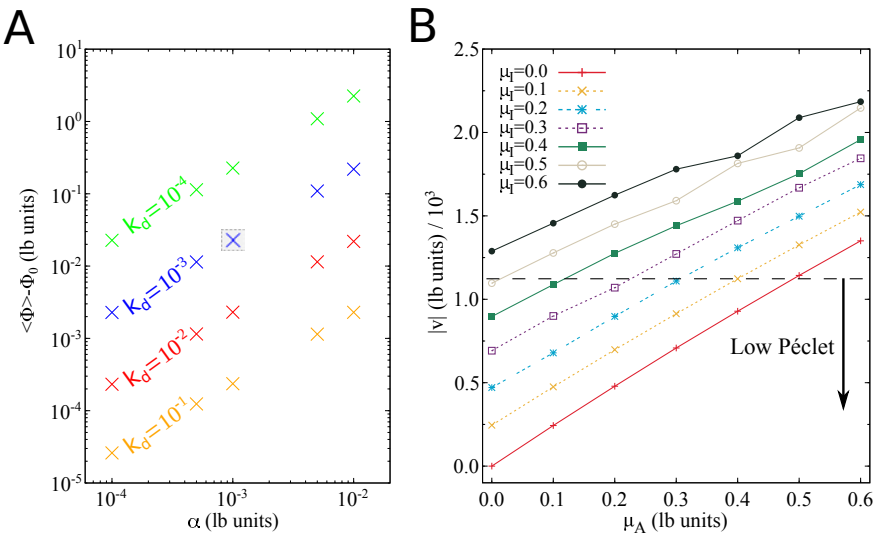


Figure 6.3: Janus parameters explored. A) Since our Janus creates ϕ at a ratio of α per each active node in its surface, we remove at each step a quantity of ϕ proportional to k_d for each time step. After a short period, the total ϕ in our simulation is stable, and hence we can calculate the average ϕ in the box. We subtract to this value the equilibrium ϕ_0 value and plot as a function of α and k_d for a Janus particle moving at water regions ($\psi = +1$). We chose $k_d = 10^{-3}$ and $\alpha = 10^{-3}$ as our working values for the rest of the simulations. **B)** For the previous couple of parameters, we plot the speed of particles in just water phase, for different values of μ_A and μ_I.

6.3.2 Inactive Janus particles and liquid-liquid interfaces

The calibration of the particle motion is now exploited to analyse the interaction of Janus particles with liquid-liquid interfaces. However, as a previous step, we analyse how capillarity forces overcome diffusiophoretic forces. To this end, first the study focuses on inactive Janus particles, with $\alpha = 0$ and uniform diffusiophoretic mobilities, $\mu = \mu_A = \mu_I$, for simplicity' sake (See Fig. 6.4).

To modulate the diffusiophoretic forces simulations are run as a function of μ and E. Capillarity forces are modulated by changing the wetting of the particle. We only study those wetting values that are extreme (totally hydrophobic, totally hydrophilic, and neutral), and place the particle in the oil, water, or interface region respectively. Because of the imbalance of capillary forces, particles will relax from their initial position toward or away from the interface. Therefore, we observe at which position particles move because of the previous forces. Figs. 6.4A and 6.4B display the change in equilibrium position of the Janus particle as a function of E and μ, respectively.

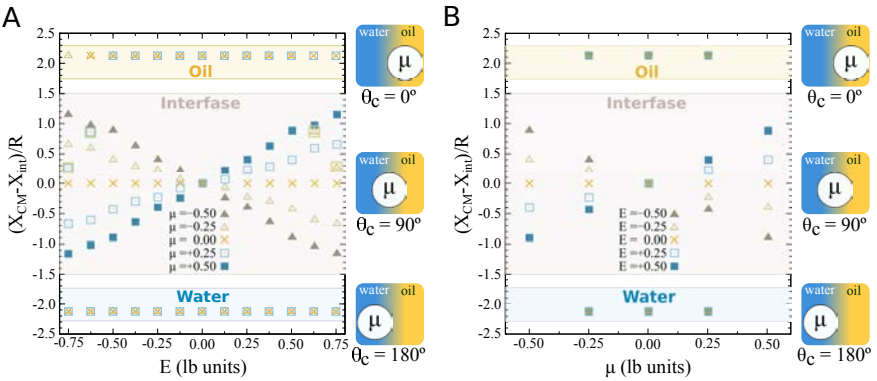


Figure 6.4: Interaction of inactive particles with uniform diffusiophoretic mobilities with liquid-liquid interfaces. Our particles interact with liquid-liquid interfaces via capillarity and diffusiophoretic forces. We explore this mechanism by introducing inactive particles with uniform diffusiophoretic mobility. We put these particles either in the oil, water or at the interface with a wetting parameter to be at each place. When particles are not at the interface, capillarity forces leave the particles at their adequate region (Water or oil). However, if particles have neutral capillarity, because of the difference of solute at each side of the liquid, particles start to move due to diffusiophoretic forces. The stronger is the diffusiophoretic mobility, the stronger is this force and hence particle move further from the interface. If the diffusiophoretic force is zero, the particle will stay at the interface because of the capillarity force. **A)** Constant μ. **B)** Constant E.

When particles are placed in the oil or water regions because their solute surrounding is homogeneous, diffusiophoretic forces vanish, μ and E do not affect the particle's motion, and consequently wetting forces gain the competition, placing the particle away from the interface.

However, when particles are placed at the interface, they can be surrounded by an inhomogeneous solute field, and diffusiophoretic forces will become relevant. To create the inhomogeneous field, E is modulated. For $E = 0$, ψ is homogeneously distributed across the fluid phases homogenous solute field, implying that wetting determines the motion and equilibrium location of the Janus particle at the liquid interface. A non-vanishing E induces an asymmetric solute field, which saturates to different values in the two coexisting liquid phases. The bigger is the difference of solute in both phases ($|E|$ is bigger) the stronger are these forces, and the further they displace the particle from the interface. At the same time, these forces will depend on the interaction of the particle with the solute, and hence a stronger μ has the same effect as increasing $|E|$.

We next consider the impact that inhomogeneous diffusiophoretic mobilities, $\mu_A \neq \mu_I$, have when diffusiophoretic forces compete with capillarity, $|E| > 0$.

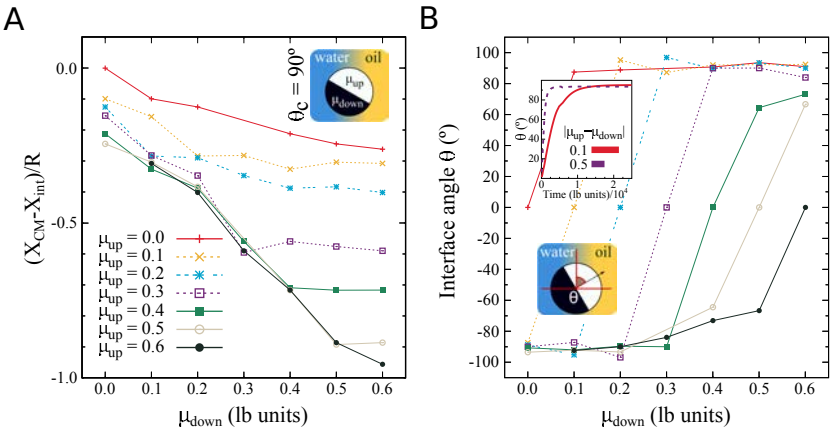


Figure 6.5: Asymmetric inactive particles at liquid-liquid interfaces. We study the neutral wetting problem introducing an asymmetric diffusiophoretic mobility in the inactive particle. We record the final position of the particle **A)** and the final angle **B)**. Stronger mobilities move the particle further from the interface. However, the minimum mobility dominates. This a consequence of a torque appearing in the particle, which faces the stronger diffusiophoretic mobility towards the liquid phase with less solute concentration. This torque does not appear if particles have the same mobility in both sides and is stronger if the difference in surface mobilities between both sides is higher.

Since Janus are inactive, to avoid confusion, we will refer to the two sides of the colloid as μ_{up} and μ_{down}, instead of μ_A and μ_I. Although diffusiophoretic forces also depend on the magnitude of E, because the larger the asymmetry in ψ the stronger the effect, we focus on the impact that μ has on particle motion, and fix $E = -1/2$ that leads to a larger concentration of the solute in the oil phase (see Fig. 6.4).

Fig. 6.5A shows that the Janus particle, initially aligned with its director vector $\hat{m}$ parallel to the interface, moves further away from the interface as the diffusiophoretic mobilities increase their magnitude. Results show that the particle saturate at a finite distance from the interface when the value of the mobility in the leading side of the particle is larger than the one in the rear part, except for $\mu_{down} = 0$. To better understand these results, Fig. 6.5 plots the saturation angle the particle director makes with the liquid interface. Initially the Janus particles are always facing the interface with larger mobility (more repulsive) toward the water region (region with less solute). Therefore, a natural torque appears. The rotation undergone by the particle is faster if the difference between both mobilities is larger, as expected (See Fig. 6.5B)inset). Because particles orient fast with the stronger diffusiophoretic mobility facing the water region, the side facing the oil is the more important input to displace the particle from the interface. Hence, this explains why in Fig. 6.5A) we reach a saturation when μ_{up} is stronger than μ_{down}.

6.3.3 Active Janus particles and liquid-liquid interfaces

Once the inherent behaviour of capillarity and wetting properties on passive Janus colloids has been established, we analyse the behaviour of active Janus colloids, $\alpha \neq 0$, with equal surface mobilities. Initially, we consider a neutrally-wetting particle with homogeneous surface mobility at the interface, and compare their motion for different surface mobilities ($\mu = 1/3$ and $\mu = 1/2$) and different ratios of product concentrated at both sides of the interface ($E = -1/2$, $E = -1/4$ and no difference, $E = 0$). We run simulations starting with different particle orientations defined by θ_p, which is the angle between the particle characteristic vector and the interface (See Fig. 6.6).

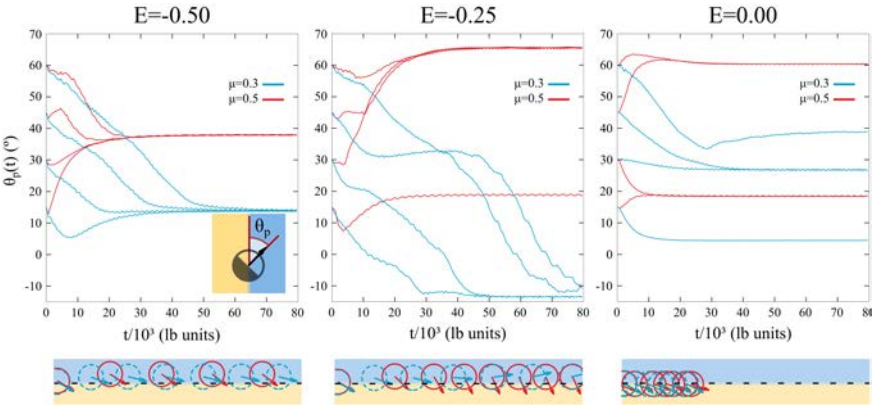


Figure 6.6: Evolution over time of the angle between the particle director vector $\hat{m}$ and the interface (θ_p) as a function of E and μ for a trapped active particle at the liquid-liquid interface with neutral wetting. The datasets correspond to various initial θ_p, two different surface mobilities ($\mu = 1/3$ and $\mu = 1/2$) and three different oxygen-oil/water affinities: $E = -1/2$ (left panel), $E = -1/4$ (middle panel) and $E = 0$ (right panel). The insets below the curves indicate the progression of the particles along the interface for the case with initial $\theta_p = 60$º over 80.000 steps (lb units) simulated.

When the difference of product in both phases is strong ($E = -1/2$) Janus particles move along the interface. Interestingly, if the simulations are initialized with different θ_p, particles stabilizes at a unique angle, which depends on the surface mobility of the particle. When the ratio of products in both phases is closer to 1 ($E = -1/4$), particles continue their motion at interfaces, although they are slower, and this unique angle starts to disappear. For some mobilities, as in $\mu = 1/3$, a single angle is observed, but for others, as in $\mu = 1/2$ we observe the appearance of competing attractors. Moreover, the asymptotic θ_p changed from the previous case. Thus, both the asymmetric accumulation of the product in both sides and the surface mobility of the particle change the torque the particle feels at the interface and that stabilizes at a certain θ_p. Finally, in the last scenario where

both phases are symmetric with respect to the solute solubility, $E = 0$, particles move very slow along the interface, and additional attractors appear for θ_p. Consequently, the asymmetric accumulation of product is also responsible for the speed of particles at the interface.

We also analyze the impact of wetting in trapped particles when diffusiophoretic forces compete with capillarity forces. To quantify such a copetition, particles are placed at a prescribed position and, as shown in Fig.6.7A), the position at which they stabilize is displayed.

First, we consider inactive particles trapped at the interface with vanishing diffusiophoretic forces, $\mu = 0$, at different fields configuration by modulating E ($E = 0$, red line and $E = -1/2$, orange line). Consequently, phoretic fields do not affect particles, and both lines coincide. Furthermore, as previously found in Figs. 6.4 and 6.5, the stronger the wetting , the further the particle displaces away from the interface.

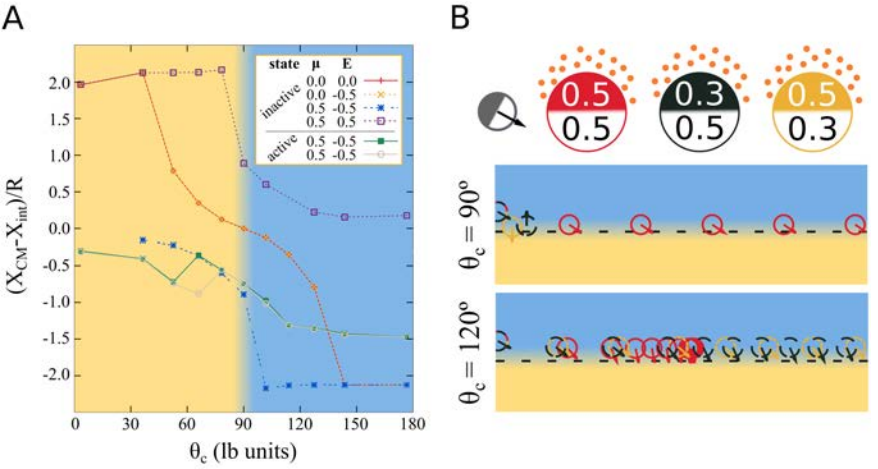


Figure 6.7: Motion of active particles next to liquid-liquid interfaces. A) Equilibrium position of particles with respect the interface as a function of the wetting of the particle and forces present. Red and orange lines are for inactive particles without phoretic forces, while blue and violet have phoretic forces. Green and grey lines are for active particles, with the difference of the initial angle condition ($\theta_p = 60$º for gray and $\theta_p = 30$º for green). **B)** Trajectory (in a plane orthogonal to the interface) of an active colloid initially placed in water and oriented towards the interface (indicated with the dashed line); the initial orientation angle is $\theta = 30°$ and the colloid is either neutral ($\theta_c = 90°$, middle panel) or partially hydrophilic ($\theta = 120°$, bottom panel). Three combinations of phoretic mobilities, (μ_I, μ_A), are considered, namely: $(0.5, 0.5)$ (red), $(0.5, 0.3)$ (yellow) and $(0.3, 0.5)$ (black).

Second, we activate diffusiophoretic forces ($\mu = 1/2$) and test at different E fields ($E = -1/2$, blue line and $E = 1/2$, violet lines). Particles with repulsive (positive) mobilities (blue line, $E < 0$) move towards low product concentration regions even if their wetting prefers regions of high product concentration ($\theta_c < 90$º) because diffusiophoretic forces overcome wetting forces. When both forces have opposite directions ($\theta_c < 90$º), particles can stabilize at the interface instead of being in one side of the interface. But if forces have the same direction ($\theta_c > 90$º), particles leave the interface, and there is no dependence on the wetting force. This effect also occurs in the opposite direction if we make attractive (negative) the mobility or if we maintain the repulsion of the particle, but we change which phase concentrate more solute (violet line). In any case, the addition of diffusiophoretic forces make the particles move away from the phase with higher solute concentration (oil phase, yellow area).

Third, we make particles active (green and grey lines). Active particles cannot enter into areas of high product concentration neither, even if particles move perpendicular to the interface. However, for hydrophilic particles ($\theta_c > 90$º) activity approaches them to the interface. To understand if the initial orientation θ_p has any effect, simulations are repeated placing the colloid with $\theta_p = 60$º (grey line) and $\theta_p = 30$º (green line). The absence of significant differennces indicates θ_p does not impact the subsequent particle motion.

We finally place particles away from the interface, in the low solute concentration region (water, blue area), moving towards the interface, as displayed in Fig.6.7B) and as experimentally demonstrated in the previous chapter (See Fig.6.7B)). We consider both a partially hydrophilic ($\theta_c = 120$º) and a neutral particle ($\theta_c = 90$º) with homogeneous diffusiophoretic ($\mu = \mu_I = \mu_A = 0.5$, red) and non-homogeneous surface mobilities ($\mu_I = 0.5, \mu_A = 0.3$ (black) and $\mu_I = 0.3, \mu_A = 0.5$ (yellow)). In all cases particles get trapped next to the interface, acquire a new angle and, in some cases, start moving along the interface. We found the same results in the previous chapter too, although here we do not see detachment of particles. However, this system does not introduce thermal fluctuations, which can be responsible for the detachment.

Neutrally wetting particles display an opposite behavior to partially hydrophilic ones. Neutrally wetting particles align to face the more repulsive side towards the region with low solute concentration (water, blue area). Thus, particles with a softer mobility in the front (yellow particle) face the front towards the region with high solute concentration (oil, yellow area), while particles with softer mobility in the rear (black particles) face the opposite. Because they align with their director vector $\hat{m}$ perpendicular to the interface, the activity cannot move particles along the interface. Moreover, this force is not capable to move particles away from the interface and hence particles get trapped. However, if particles present uniform diffusiophoretic mobilities (red particle), interfacial alignement is lacking, and particles displace along the interface.

Partially hydrophilic particles partially move further away from the interface. Particles with non-uniform mobilities align to a new angle when they are close to the interface and move along the interface. Particles with softer mobility in the front (yellow) align more parallel to the interface's contact line than particles with softer mobility in the rear (black). Hence, particles with softer mobility in the front move faster. Opposite to neutral wetting, here we see that uniform mobilities causes the particle to face perpendicular to the interface and stop their motion. A possible explanation to this fact is that since particle are less trapped than in neutral wetting, particles do not feel the torque appearing in Fig. 6.5, and hence particles with non-uniform mobilities are freer to change their orientation. The combination of non-uniformity and activity aligns the particle to an angle that, without this non-uniformity, would orient the particle perpendicular to the interface as in the uniform case.

6.4 | Conclusions

In this work we have introduced a new model based on Lattice-Boltzmann to study the interaction of active particles with liquid-liquid interfaces. The model facilitates the study of the full hydrodynamics of the system, on the same footing as diffusiophoretic and wetting forces suspended particles are subject to. The model allows to switch on and off easily these forces, and to modify the particle properties such as wetting and the diffusiophoretic force, differentiating for this last scenario two parts on the particle with its own activity and mobility. These contributions are formulated locally, and can then be adapted to particles of arbitrary shape, with a general inhomogeneous treatment of their surfaces. Moreover, the liquid mixture can show asymmetric solubility to the chemicals produced by the particles.

We have tested the interaction of inactive particles trapped at the interface under different wetting angles θ_c (0º, 90º and 180º) and different particle surface mobility. We have seen that while wetting dominates over diffusiophoretic forces, when the wetting is neutral (90º), diffusiophoretic properties are important, and inactive particles with homogeneous surface mobility displace from the interface. This interaction is proportional to the surface mobility, and to the different of products between both phases. When the surface has an asymmetric mobility, particles reorient to have its more repulsive face towards the liquid phase with less product, and displace from the phase of high accumulation of product. The reorientation depends on the strength of the mobility. The more repulsive, the fastest reorient.

Active particles at the interface with neutral wetting move along the interface. The more asymmetry between product accumulation in both phases and the more repulsive is their surface to products, the fastest particles move. Particles reorient themselves to a specific angle, no matter the angle at which particles are placed. However, if the asymmetry of products between both phases decays, particles find different equilibrium positions depending on the initial angle. This effect is seen for different surface mobilities. If wetting is changed, particles will stay closer to the interface rather if they would not have the activity. If particles with high wetting for the side of initial motion move towards the interface, they will contact the interface, reorient, and move along the interface. Depending on the ratio of the surface mobilities, particles can stop, or continue their motion.

Overall, the proposed model has huge capabilities to explain many phenomena occurring at these interfaces, and that sets a new start line where to study these and more complex systems.

6.5 | Further perspectives

The study presented in this chapter is an initial step to study systems of active particle at liquid-liquid interfaces. In this study we have focused on the effect of the wetting and diffusiophoretic forces for single particles. However, the code is capable to introduce as many particles as hardware capabilities we have. Moreover, each particle is divided in two surface areas, that we can modify its proportion, and have different surface mobility and activity for each side and particle. We can also change their wetting individually, and we can start them at different positions, velocities, and orientations. Thus, we can create lots of different initial distributions to study these systems.

Moreover, all the study has been carried out with flat interfaces as I did with the previous chapter. Nonetheless, while working in this chapter I also introduced an initial distribution for the liquid-liquid interface in the shape of a cylinder, which was stable along the simulation. Therefore, as I did in the previous chapter, we could also study the effect of interface curvature to the motion of the particles.

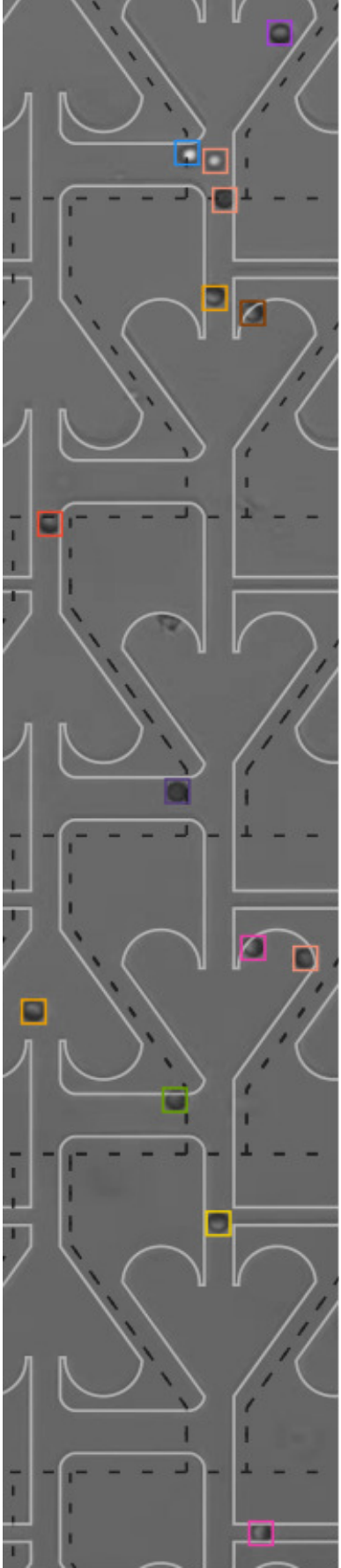
Two more interesting studies may require attention with this code. On the one hand, the code also allows us to introduce solid boundaries, either by flat walls or by selecting the nodes we want to be solid. Because the code also allows to introduce forces such as gravity, we could follow this work by studying the solid-liquid-liquid interfaces as I did experimentally. On the other hand, this software can add small fluctuations for the hydrodynamics and the phoretic fields, which we can turn on/off individually and, particles can also have a direct kick that aligns them every some steps. These methods introduce Brownian motion to the system, and it would be interesting to study these systems with the inherent Brownian motion they should have.

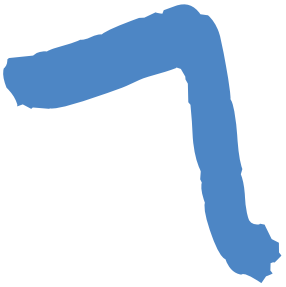
Finally, these studies probably will need stronger tools to analyse the system. This could be an opportunity to work and expand the tools I created while analysing these simulations, and that can be found in section C.4.

III

Active particles next to solid interfaces

The third part of this thesis is related with the study of many active particles next to interfaces. In this part we focus on the guidance of particles using interfaces to obtain new phenomena such as spontaneous accumulation in a microfluidic device with a pattern that at a priori does not break detailed balance. Because of the easy creation of solid-liquid interfaces, and the similarities we found with liquid-liquid interfaces at an experimental level, we decided to do these experiments with solid-liquid interfaces. This part is of considering beauty, because it connects two fields that we could consider as far from each other as active matter and topology. The results of this part do not only show the possibilities of collective phenomena for active matter but also how we can use active matter to build a toy model to explain further fields of physics.





Guided accumulation of active particles by topological design of a second-order skin effect

"We choose to go to the moon in this decade and do the other things, not because they are easy, but because they are hard."

John F. Kennedy

Collective guidance of out-of-equilibrium systems without using external fields is a challenge of paramount importance in active matter, ranging from bacterial colonies to swarms of self-propelled particles. Designing strategies to guide active matter using microfabricated circuits will provide insights for applications from sensing, drug delivery to water remediation [194, 195, 196]. However, achieving directed motion without breaking detailed balance, for example by asymmetric topographical patterning, is challenging. Here we engineer a two-dimensional periodic topographical design with detailed balance in its unit cell where we observe spontaneous particle edge guidance and corner accumulation of self-propelled particles. This emergent behaviour is guaranteed by a second-order [197, 198, 199, 200, 201, 202] non-Hermitian skin effect [203, 204, 205, 206, 207, 208], a topologically robust non-equilibrium phenomenon, that we use to dynamically break detailed balance. Our stochastic circuit model predicts, without fitting parameters, how guidance and accumulation can be controlled and enhanced by design: a device guides particles more efficiently if the topological invariant characterizing it is non-zero. Our work establishes a fruitful bridge between active and topological matter, and our design principles offer a blueprint to design devices that display spontaneous, robust, and predictable guided motion and accumulation, guaranteed by out-of-equilibrium topology.

The present chapter is based on the paper "Guided accumulation of active particles by topological design of a second-order skin effect"[209], from which I am first author, and to which I did the experiments and analysed the data, including the software development as the tracking system shown in section 4.4. I would like to special thank Adolfo Grushin who had the idea of this project and supervised it with many inputs to develop the work. I also want to thank Serguei Tchoumakov, who helped me with the analysis, and that finally developed a better analysis than the one I was doing. Finally, I also want to thank Maria Guix, Jaideep Katuri and the MicroFabSpace and Microscopy Characterization Facility, Unit 7 of ICTS "NANBIOSIS" from CIBER-BBN at IBEC. Without their inputs and help, the design of the setup used would have not been possible.

7.1 | Introduction

In the previous chapters we studied individual, or a couple of, particles interacting with liquid-liquid interfaces. However, as described in section 1.1.3, new phenomena can emerge in active matter when we work with many active entities. Hence, as a step further for the development of this thesis, we started new studies on the interactions of hundreds of these particles with the same interfaces. For example, we could address this problem by studying the possible synchronization of particles moving around droplets that conform a square lattice as proposed in section 5.7. Nonetheless, these experiments could arise some experimental problems, such of reproducibility or stability of the interface for long times. Furthermore, they will also require a simple interface pattern, because building liquid-liquid interfaces with the desired complex geometry is not trivial. However, a complex geometry could be useful to exploit the guidance of particles at these interfaces, and obtain new phenomena from the guidance. To make a complex geometry we could use solid-liquid interfaces, since they are very easy to build. Because solid-liquid interfaces also guide particles (See section 1.2.2) we could think that, even if the underlying physics can be different, ultimately particles had a similar behaviour at a single level. Thus, we could expect to see similar collective behaviour. Hence I decided to start this project using solid-liquid interfaces, with the aim of guiding and accumulating these particles. These processes will be useful not only to study the physics, but for the development of new devices such as lab-on-a-chip for screening test, where a preconcentration of an analyte, that particles could take, would be necessary prior its detection, and hence accumulation of particles with the analyte could work as a concentration method.

To aid this accumulation problem, we decided to explore new fields such as topology. Specially, because the last decades of research in condensed matter physics have revealed that exceptionally robust electronic motion occurs at the boundaries of a class of insulators known as topological insulators [210, 211], which I introduced in section 1.2.3. These ideas extend beyond solid-state physics, and predict guided boundary motion in systems including photonic [212], acoustic and mechanical systems [213]. Therefore, they could be useful for us too, especially after the recent discovery that topological properties emerge in the class of out-of-equilibrium systems described by non-Hermitian matrices, which includes active matter systems [214]. Hence, these studies have opened the possibility to engineer new robust behaviour out of equilibrium [215, 216].

Interesting, non-Hermitian matrices introduce some new phenomena in these systems. While in equilibrium topological boundary states are predicted by a non-zero bulk topological invariant, a feature known as the bulk-boundary correspondence, in non-Hermitian systems, this correspondence is broken by the skin-effect [203, 204, 205, 203, 207, 208]. For example, in a one-dimensional (1D) chain of hopping particles, the first-order non-Hermitian skin effect arises from the asymmetry between left and right hopping probabilities, which results in an accumulation of a macroscopic number of modes, of the order of the system size, on one side of the system. This 1D effect occurs in systems without an inversion centre, and has been observed in photon dynamics [217], mechanical metamaterials [218, 219, 220], optical fibers [221] and topoelectrical circuits [222, 223]. In 1D, the skin-effect occurs if a topological invariant, the integer associated to the winding of the complex spectrum of the normal modes, is non-zero [224, 225, 226].

Higher-dimensional versions of the skin effect can display a considerably richer and subtle phenomenology [227, 228, 229, 230, 231, 232, 197, 202, 198, 199, 200, 201, 233]. For instance, it has been predicted only in out-of-equilibrium systems in two dimensions (2D) [197, 202, 198, 199, 200, 201] the second-order non-Hermitian skin effect appears. It differs from the first-order skin effect because:

- (i) it can occur in inversion symmetric systems, accumulating modes at opposing corners rather than edges [223, 234]
- (ii) the number of accumulated modes is of order of the system boundary L, rather than its area L^2.

While the first-order non-Hermitian skin effect requires inversion to be broken, e.g. due to an applied field, the emergence of the second-order non-Hermitian skin effect is guaranteed by the presence of certain symmetries [199, 200]. However, predicting the second-order non-Hermitian skin effect is challenging in general, and it remains unobserved. Dissipation, which drives a system out of equilibrium, is hard to control experimentally in quantum electronic devices, therefore calling for other platforms to realize the second-order non-Hermitian skin effect.

Active matter systems [194] are a natural platform to explore non-Hermitian topological physics, since these systems absorb and dissipate energy [214]. Often, the hydrodynamic equations that describe their flow can be mapped to a topological Hamiltonian. This strategy predicts topologically protected motion of topological waves in active-liquid metamaterials [235, 236, 237], skin-modes in active elastic media [234], and emergent chiral behaviour for periodic arrays of defects [238]. Non-Hermitian topology in active matter has been demonstrated experimentally in active nematic cells [239], and robotic [219] and piezoelectric metamaterials [220].

Overall, 2D systems showing the second-order non-Hermitian skin effect can be a solution to accumulate particles. As a consequence, we designed microfabricated devices that display a controllable second-order non-Hermitian skin effect by creating two designs, the trivial and the topological, where we see a large difference over the trivial scenario. These devices are designed to satisfy detailed balance on their unit cell, such that the flow of particles through a unit cell vanishes. The non-Hermitian skin effect dynamically breaks this detailed balance on the top and bottom edges, and we use it to guide and accumulate self-propelled Janus particles. In contrast to hydrodynamic descriptions, the topological particle dynamics in our devices is quantitatively described by a stochastic circuit model [240] without fitting parameters. It establishes that topological circuits, where a topological invariant $\nu = 1$, display the second-order non-Hermitian skin effect that guides and accumulates particles more efficiently than the topologically trivial circuits, with $\nu = 0$. This phenomenon occurs without external stimuli, e.g. electrical or magnetic fields, a useful feature for active matter applications [194, 195, 196], and to extend our design principles to metamaterial platforms [212, 213, 241].

7.2 | Design and stochastic model

7.2.1 Coupled-wire device design

Our design realizes the coupled-wire construction, a theoretical tool to construct topological phases [242]. This is possible by the precise engineering of microchannel devices (see section 2.2 and 1.2.2). Each device contains two types of horizontal microchannels, the wires, which are coupled vertically, forming a 2D mask (see Fig. 7.1A), B)). The horizontal microchannels are consecutive left or right oriented hearth-shaped ratchets, that favour a unidirectional motion towards their tip as seen in section 1.2.2. Their left-right orientation alternates vertically. These horizontal microchannels are coupled by vertical microchannels that are straight, designed to imprint a symmetric vertical motion to the nanoparticles. The vertical microchannels alternate in width, with successive narrow and wide channels. Wide channels are more likely to be followed by the active particles than narrow channels. Using these principles, we design two types of devices that we coin trivial and topological, depicted in Figs. 7.1A) and B), respectively. They only differ in that the narrow and wide channels exchange their roles along the vertical direction (see central inset of Figs. 7.1A) and B)). Both trivial and topological designs have the same number of left and right oriented ratchets, so they satisfy global balance.

Active particles are injected within the device and move along the microchannels walls, see Figs. 7.1C) and D). We study devices built out of $(L_x, L_y) = (12, 6)$ and $(L_x, L_y) = (13, 14)$ unit-cells in the horizontal (x) and vertical (y) directions, with either a small or a large density of injected active particles, to compare how the density of active particles qualitatively affects our experiment. Indeed, a larger density of particles enhances the Brownian motion, due to the many collisions, but also decreases the velocity of particles and leads to more clusters. The two densities are obtained by injecting particles in either one of the two solutions:

1. High density. We dilute $17.5\ \mu$L of the solution of Janus particles prepared as detailed in section 2.1 with $35\ \mu$L of H_2O_2 at 2% per volume and $47.5\ \mu$L of miliQ water.
2. Low density. We do the same dilution as high density, but we previously dilute the solution of Janus particles prepared as detailed in section 2.1 in miliQ water (3/10).

This way we expect a ratio of concentrations of a third to a half between high and low densities of active particles.

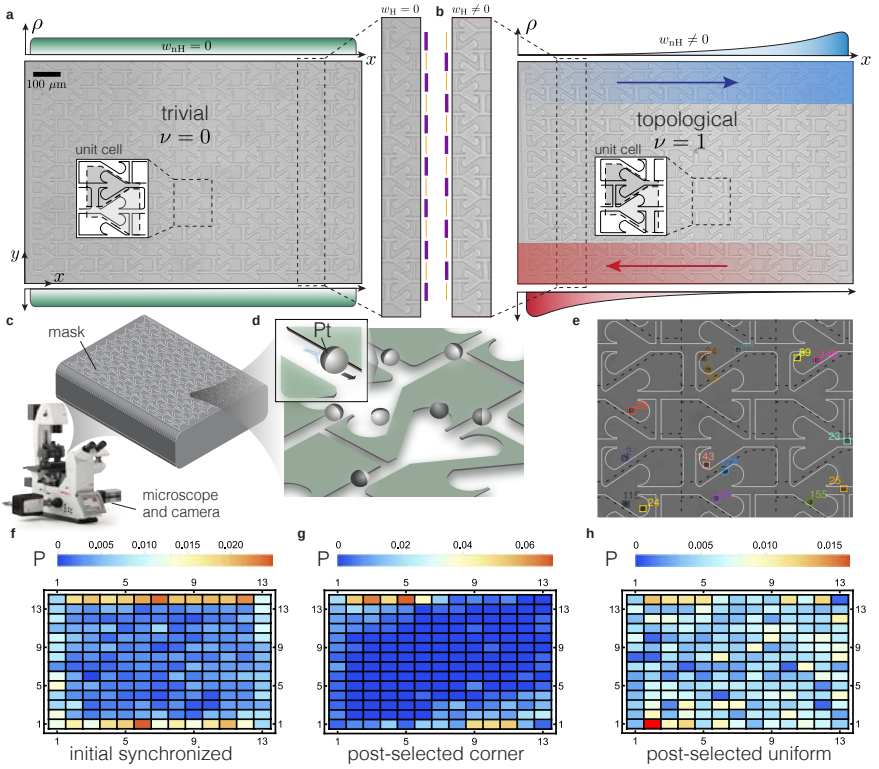


Figure 7.1: Experimental designs. A) and **B)** show the trivial and the topological devices respectively, with $(L_x, L_y) = (13, 14)$ unit cells. The insets depict the unit cells which are the same for both devices except the vertical wide and narrow channels are reversed, as shown in the zoomed central inset and emphasized with orange and purple lines. The topological device displays chiral edge modes at the top and bottom, sketched in blue and red respectively, and which are related to a non-vanishing Hermitian topological invariant, $w_{\rm H}$. These topological edge modes have a non-vanishing non-Hermitian winding number, $w_{\rm nH}$, responsible for the accumulation of active particles at the corners. This accumulation at the corners is expected to vanish for the trivial device (see the uniform density in green) because the topological invariant $\nu = w_{\rm H} w_{\rm nH}$ vanishes. **C)** and **D)** show a schematic of the experimental set up, where Pt-coated SiO_2 Janus particles self-propel when hydrogen peroxide is added, following the topographic features of each design. **E)** shows a portion of the device to illustrate the tracking of particles by using a neural network, where each particle is uniquely identified within our algorithm. We locate particles within the cells outlined by the dashed lines. **F)**-**H)** show the probability distribution data, $\mathbf{P}$, of particles on the lattice when trajectories are tracked and synchronized to start at the same initial time **F)**, post-selected to start at opposing corners **G)**, or post-selected to have an initial uniform distribution **H)**. The particle distributions in **G)** and **H)** are shown three minutes after synchronization.

We then track the particles from recorded videos using the neural tracking system presented in section 4.4. We locate the particles within a grid of regularly spaced cells (dashed lines in Fig. 7.1E)) by using the methods explained in section 4.1.6 to eliminate any intrinsic motion of a particle within a cell and thus to reduce noise. In Fig. 7.1 F) we can see the initial distribution of particles obtained from the experiments. We observe more particles at the borders than in the bulk because of the particle flux from outside the device. Since this flux cannot be controlled, we cannot directly choose the initial particle distribution. We can overcome this limitation in order to study the average particle motion by post-selecting trajectories, which is a differentiating aspect between active particle and electronic systems. For the next analysis we either post-select particles from the corner or uniformly:

- Post-selection corner. We select particles starting from the top left and bottom right corners (see Fig. 7.1G)) in two steps. We first select trajectories that go through one of the two corners. Then, among the selected trajectories, we remove all frames before the one where the particles first enter one of the two corners.

- Post-selected uniform. We select particles uniformly scattered over the sample based on simulated annealing. We focus on the probability distribution $P_{\sigma ij}(t=0)$ of all trajectories on their first frame. We scan trajectories from the longest to the shortest and remove the first frames of a trajectory until the newly generated distribution of particles $P'_{\sigma ij}(t=0)$ is more uniform than the one, $P_{\sigma ij}(t=0)$, at the previous step. We consider the distribution is more uniform when:

$$C(P') = \sqrt{\sum_{\sigma ij} \left(P'_{\sigma ij}(t=0) - 1/(L_x L_y)\right)^2} < C(P). \tag{7.2.1}$$

  During this procedure, we remove trajectories that have less than 6 minutes of video to remove noise in the first frames.

In Fig. 7.1G) and H) we can see the result of doing these post-selections compared with the original data in Fig. 7.1F). In the main text, we focus on the larger device, with $(L_x, L_y) = (13, 14)$, see Fig. 7.1. All other devices and the corresponding results are shown in the Supplementary Information 7.C.

7.2.2 Stochastic model

Because of the irregular microchannel walls and the collisions between particles, we model the collective motion of active particles as a Brownian motion in a stochastic network with transition probabilities between the cells introduced in Fig. 7.1E). In our model, these cells are represented by nodes as we can see in Fig. 7.2. Hence, within each device we describe the motion of active particles as a random walk described by the continuous-time Markov master equations:

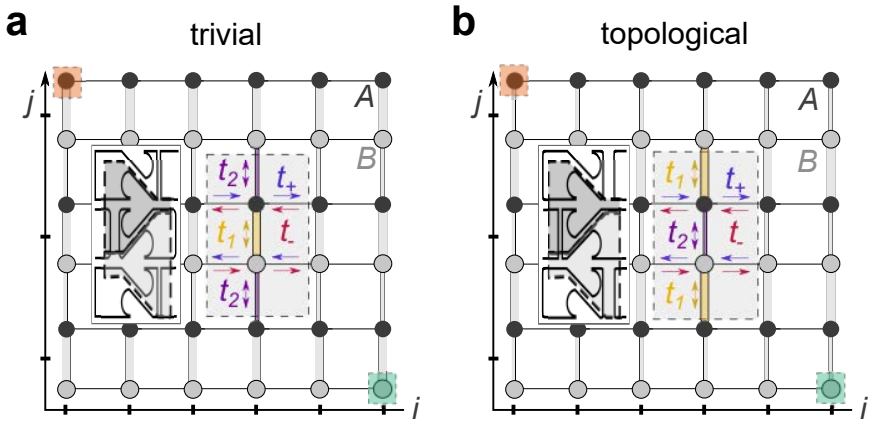


Figure 7.2: Stochastic model. Illustration of the stochastic network in Eq. (7.2.2) for our trivial device **A)** and our topological device **B)**. We superimpose the unit cell of the experimental device (see Figs. 7.1 A), B)). Each dot corresponds to a cell of the device and each link represents one of the transition probabilities (t_1, t_2, t_+, t_-) to move in each cardinal direction. The unit cell of this lattice is the rectangle in gray, which contains black and gray nodes that are labelled by the sub-cell indexes A, B, respectively, introduced in Eq. (7.2.2). The orange and green areas in **A), B)** represent the starting points for particle motion, following our post-selection corner method in our experiments.

$$\begin{cases} \tau \dfrac{dP_{A,ij}}{dt} = t_1(P_{B,i,j+1} - P_{A,ij}) \\ \qquad\qquad + t_2(P_{B,ij} - P_{A,ij}) + t_+ P_{A,i-1,j} \\ \qquad\qquad - t_- P_{A,ij} + t_- P_{A,i+1,j} - t_+ P_{A,ij}, \\ \tau \dfrac{dP_{B,ij}}{dt} = t_1(P_{A,i,j-1} - P_{B,ij}) \\ \qquad\qquad + t_2(P_{A,ij} - P_{B,ij}) + t_+ P_{B,i+1,j} \\ \qquad\qquad - t_- P_{B,ij} + t_- P_{B,i-1,j} - t_+ P_{B,ij}, \end{cases} \tag{7.2.2}$$

where we subdivide the lattice on two sublattices, A and B, which include the idea of same hearth direction. The lattice also has its natural discrete coordinates, (i, j), which describe the column and row, respectively, of the cell in the system. Including the three variables, we construct the probability distribution to observe a particle, $P_{\sigma,ij}$, with $\sigma \in (A, B)$ and $(i, j) \in \mathbb{N}^2$. We also insert four transition probabilities ($t_\pm$, t_1 and t_2) for the motion of the particles. $t_\pm$ is the transition probability for the motion along or against the ratchet-like microchannels, while t_1 and t_2 are the transition probability for the motion along the wide and narrow microchannels, respectively. Finally, we introduce a timescale for a particle to move

between adjacent sites, which we describe as τ. The time scale τ and the conditional probabilities are extracted from our experimental data in Supplementary Information 7.A. We found that these values depend on the device and density, but in any case, they always fulfil the same behaviour: a large ratio t_+/t_- and t_2/t_1. The large ratio t_+/t_- shows the motion is unidirectional on the horizontal axis and $t_2/t_1 \approx 1.5 \neq 1$ confirms that we can explore the difference between trivial and topological devices. However, notice that as we wrote the model, we have neglected correlations between particles (e.g., jamming). This can be a problem for experiments at high densities, as we will see in the next analysis.

Eq. (7.2.2) is also balanced to ensure probability conservation, $\sum_{\sigma ij} P_{\sigma,ij} = 1$, and also, since the equation is irreducible, it has a unique stationary probability distribution with $dP_{\mathrm{st.}}/dt = 0$ [243]. Moreover, it also has the form of the Schrödinger equation with a non-Hermitian Hamiltonian. Finally, it is convenient to introduce the transition matrix $\hat{W}$ such that:

$$\tau \frac{d\mathbf{P}}{dt} = \hat{W}\mathbf{P}, \tag{7.2.3}$$

where $(\mathbf{P})_{\sigma ij} = P_{\sigma,ij}$.

7.3 | Topological chiral edge motion

We first use the corner post-selection technique to explore the edge dynamics of an ensemble of trajectories synchronized to start either at the top left or the bottom right corners (see Fig. 7.1G)). We select 281 trajectories of about 17 minutes each, for the trivial device, and 327 trajectories, of about 16 minutes each, for the topological case. In Figs. 7.3A) and D) we show the density of active particles as a function of time for trivial (A), B)) and topological (D),E)) devices.

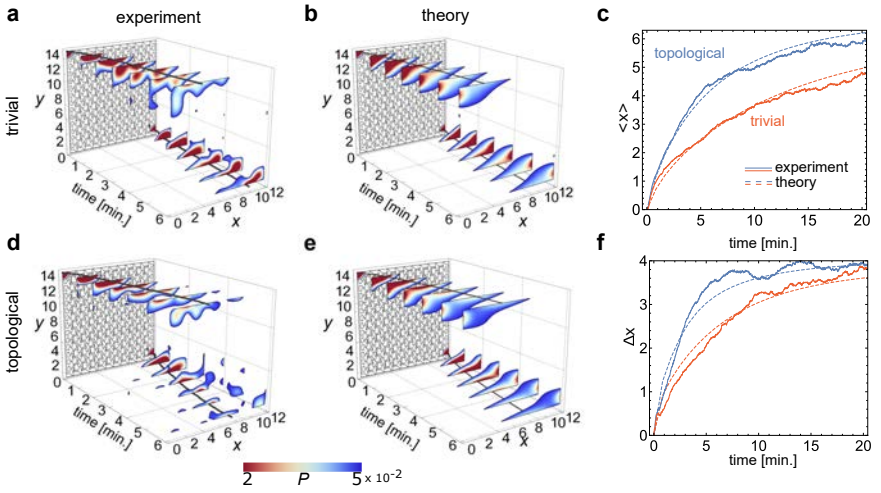


Figure 7.3: Propagation of particles starting from corners. We show both the experimental (first column) and theoretical (second column) behaviour for (L_x, L_y) = (13, 14) using the trivial **A), B)** and the topological **D), E)** device. The solid lines in **A), B), D), E)** follow the average position of the particles starting from top left and bottom right corners. **C)** Average position and **F)** standard deviation in units of the cell index of a set of trajectories starting from either top left or bottom right corners of the device, which we compare with our model. The active particles of a topological device propagate faster than in a trivial device.

Qualitatively, the particle distribution propagates unidirectionally along the edge, faster in the topological device than in the trivial one. Quantitatively, the time-dependent average displacement $\langle x \rangle$ and spread $\Delta x \equiv \sqrt{\langle (x - \langle x \rangle)^2 \rangle}$ confirms this behaviour (See Figs. 7.3C) and F)). We observe that active particles in the topological device are ahead of the trivial one, by one unit cell after 3 minutes.

Next, we compare these experimental results with our theoretical model. We initialize Eq. (7.2.3) with a probability distribution localized at the top left and bottom right corner. We compare the experimental results with theoretical predictions in Figs. 7.3B), C), E), F), using the parameters t_1, t_2, t_+, t_- and τ set by our statistical analysis of the experimental data.

The theoretical curves qualitatively reproduce the experimental trends without any fitting parameter, for both topological and trivial devices. The above observations are reproduced for smaller devices and larger densities of actives particles, see Supplementary Information 7.C. We find that for larger densities, the particles are slower than theory predicts, an effect we attribute to particle jamming, neglected in our model. A lower density of particles prevents jamming, in which case the motion compares better with our model.

To explain the difference shown in our model, we need to refer to the edge states appearing in both devices. The motion of active particles is understood decomposing Eq. (7.2.3) into the normal modes, $\mathbf{P_k}$, defined by:

$$\hat{W} \cdot \mathbf{P_k} = \lambda_{\mathbf{k}} \mathbf{P_k}, \tag{7.3.1}$$

where $\lambda_{\mathbf{k}}$ is a complex scalar which depends on wavevector $\mathbf{k} = (k_x, k_y)$ for periodic lattices. The real part of $\lambda_{\mathbf{k}}$ sets the lifetime of the normal mode, and its curvature $\partial_k^2 \mathrm{Re}(\lambda_{\mathbf{k}})$ at $\mathbf{k} = 0$ sets its diffusion coefficient. The slope of the imaginary part $\partial_k \mathrm{Im}(\lambda_{\mathbf{k}})$ at $\mathbf{k} = 0$ sets the velocity of the normal mode (see Supplementary Information 7.B).

For open boundaries along y and periodic along x, the spectrum $\lambda_{\mathbf{k}}$ can be represented as a function of k_x. The boundary condition is such that the probability distribution vanishes outside the lattice. The resulting spectrum is shown in Figs. 7.4 A), B) for the topological device and C), D) for the trivial one. The spectrum is colored according to the localization of the normal modes, where red and blue colours denote states at the top and bottom edges, respectively. These edge modes have a chiral group velocity, shown by the slope of the imaginary part of the spectrum at $k_x = 0$ in Fig. 7.4B), and are absent for the trivial device (see Figs. 7.4C),D)). If in addition to enforcing a vanishing probability distribution outside the lattice we impose that detailed balance is preserved at the boundary, an additional edge potential partially hybridizes the edge modes with bulk modes, but does not remove them (see Supplementary Information 7.B).

The existence of edge modes is guaranteed by a bulk Hermitian topological invariant. In the vertical direction the couplings, t_1 and t_2, alternate between weak and strong (see Fig. 7.1B)). This is the stochastic equivalent of the Su-Schrieffer-Heeger model of the polyacetylene chain (see section 1.2.3), which is characterized by a winding number w_{H} in the y direction (see Supplementary Information 7.B). $w_{\mathrm{H}} = 1$ and $w_{\mathrm{H}} = 0$ for the topological and trivial device models, respectively.

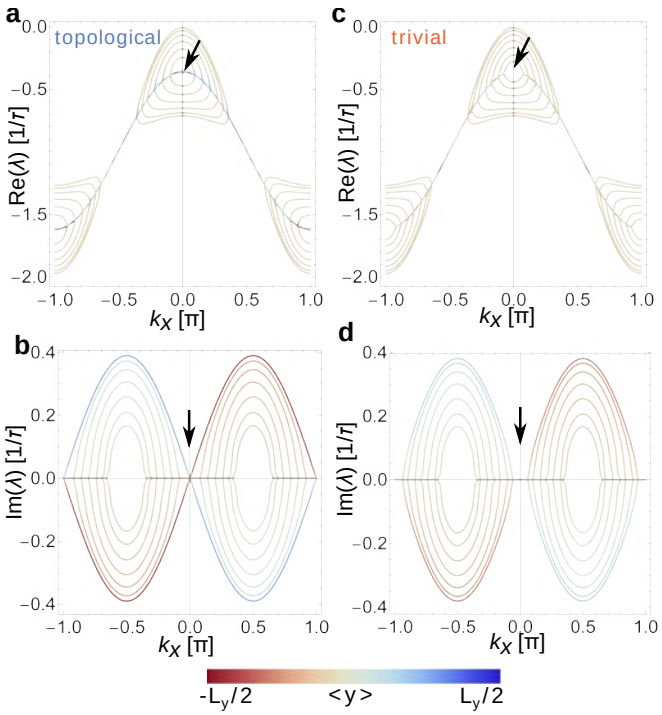


Figure 7.4: Stochastic chiral edge states. Real **A),C)** and **B),D)** imaginary part of the normal modes of the rate matrix $\hat{W}$ for open boundary conditions in the y direction, for the **A),B)** topological and **C),D)** trivial devices. The colour denotes the average $\langle y \rangle$ position of a normal mode. Modes in green are delocalized and correspond to bulk states. Modes in blue and red correspond are strongly localized on the top or bottom edges, respectively. The arrows in **C),D)** highlight the presence or absence of topological edge modes close to $\mathbf{k} = 0$, where their lifetime $\mathrm{Re}(\lambda)$ is largest.

This implies that they respectively have, or not, topological edge modes for a boundary in the y direction, indicated by black arrows in Figs. 7.4B), D). Thus, this demonstrates that the displacement in the topological device is larger because of the topological edge modes.

7.4 | Corner accumulation

The detailed balance of the unit cell implies that $\hat{W}$ has no strong topological invariant [244] (see Supplementary Information 7.B). Moreover, $\hat{W}$ has inversion symmetry, implying that the first-order skin-effect vanishes. However, our model has a second-order skin effect. To derive the second-order skin effect of $\hat{W}$ we consider the topological edge modes, $\mathbf{P}_{s,\chi}$, at the top ($\chi = +$) and bottom ($\chi = -$) described by a 1D equation $\tau\partial_t \mathbf{P}_{s,\pm} = H_{s,\pm}\mathbf{P}_{s,\pm}$ (see Supplementary Information 7.B). These modes locate on either A (for $\chi = +$) or B (for $\chi = -$) sub-lattice, have a ballistic propagation $\langle x\rangle_\pm = \pm(t_+ - t_-)t/\tau$, and diffuse by an amount $\Delta x = \sqrt{(t_+ + t_-)t/\tau}$ after a time t (see Supplementary Information 7.B). The real part of $H_{s,\pm}$ is finite and indicates that the contribution of the topological edge modes to the total probability decays over a timescale $\tau_d = \tau/(t_1 + t_2)$, where τ_d sets how far the particles propagate due to the topological edge modes. The edge modes $H_{s,\pm}$ have a finite 1D winding number $w_{\mathrm{nH}} = \pm 1$ that implies a 1D non-Hermitian skin effect [224, 225, 226]. Since the edge modes are spatially separated, active particles can accumulate at the top and bottom corners.

We detect the accumulation of active particles experimentally by post-selecting trajectories that start from a uniform configuration (see Fig. 7.1 H)). We observe an accumulation of active particles at the corners which is larger in the topological device (Figs. 7.5 A), C)) and that qualitatively compares with our model (Fig. 7.5 B), D)). This observation can be made quantitative using the Shannon entropy of the particle distribution $S = -\sum_{ij\sigma} P_{ij\sigma} \ln(P_{ij\sigma})$. The entropy is maximal for a uniform distribution of particles, $S < S_{\mathrm{uniform}} = \ln(L_x L_y)$, and it decreases if particles localize. We average out other sources of particle localization unrelated to the non-Hermitian skin-effect by averaging the probability distribution over neighbouring cells (see Supplementary Information 7.C).

The experimental and theoretical entropies are depicted in Fig. 7.5 H). Both figures show a smaller entropy in the topological device than in the trivial one. They depart from each other at the same rate, yet the absolute values of the experimental entropies are a factor 13 smaller than theory. Our model thus captures the difference between trivial and topological devices but underestimates the accumulation that occurs in the experiment. A potential reason is that, in the experiment, a local increase in the number of active particles leads to particle jamming, neglected in our model. When we decrease the density of particles to reduce jamming, the entropy still drops but is similar for topological and trivial devices. This suggests that there is a critical density of particles to observe corner accumulation.

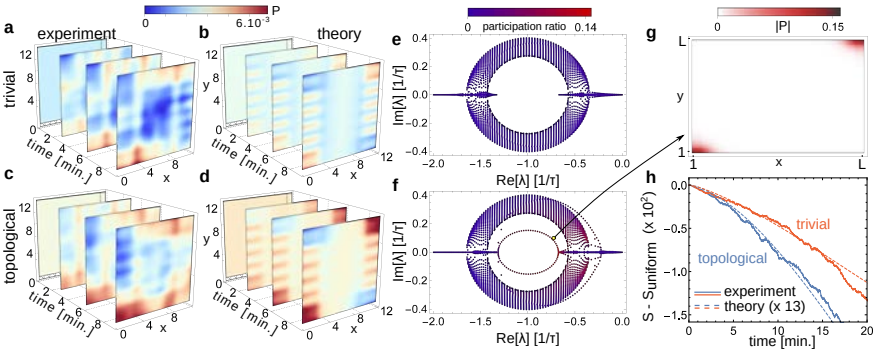


Figure 7.5: Second-order non-Hermitian skin effect of the active particles. A), B), C), D) We compare the distribution of particles in the trivial **A)**,**B)** and topological **C)**,**D)** devices with $(L_x,\, L_y) = (13, 14)$, observed experimentally **A)**,**C)** and predicted theoretically **B)**,**D)**. We observe that more particles locate on the top right and bottom left corners. **E)**,**F)** Parametric representation of the real and imaginary parts of the spectrum of normal modes for the model of the trivial **E)** and topological **F)** devices with $(L_x, L_y) = (L, L) = (70, 70)$, coloured with the participation ratio of each normal mode $\sum_{\sigma ij} |P_{\sigma ij}|^4$, which is small for a delocalized mode. The gap in the periodic band structure is within the dashed circle (see Supplementary Information 7.B). The number of localized modes within the point gap is proportional to the size of the device, L, and are localized at the corners as shown in **G)**. **H)** Shannon entropy of the particle distribution over time. The smaller entropy in the topological device is associated to an accumulation of particles at the corners, signalling the second-order non-Hermitian skin effect. The theoretical figures match the experimental curves when multiplied by a factor $\times 13$, suggesting that particle jamming, that occurs frequently for higher densities of active particles, contributes to enhance the accumulation.

The corner accumulation we observe is a consequence of the chiral motion of topological edge modes, that accumulate at the corners for long times. In the topological device the chiral edge motion occurs because $w_{\mathrm{H}} \neq 0$, and the corner accumulation occurs because $w_{\mathrm{nH}} \neq 0$ for $H_{s,\pm}$. The two devices thus have a different topological invariant:

$$\nu = w_{\mathrm{H}} w_{\mathrm{nH}}, \tag{7.4.1}$$

which equals one or zero for the topological and trivial devices, respectively. In the Supplementary Information 7.B we show that this invariant is equivalent to that in Ref. [199], and that it signals a second-order skin-effect as follows. First, $\hat{W}$ has inversion symmetry, and a point gap spectrum in which corner modes appear only for open-boundary conditions in both directions (Figs. 7.5 F), G)). Second, the number of corner modes scales as the edge length L, rather than the system size L^2, a defining characteristic of the second-order the skin-effect [200].

7.5 | Conclusions

Overall, our work establishes a strategy to design circuits that spontaneously break detailed balance to guide and accumulate active matter, enforced by robust out-of-equilibrium topological phenomena.

We have demonstrated these results both experimentally, and theoretically, via a stochastic model that is able to describe the underlying physics. Both experiments and theory match extraordinary well considering we have not used any fitting parameters. This match allows us to prove that these results are a consequence of extra edge states in the topological device with respect the trivial one and have made also possible to observe a second-order non-Hermitian skin effect, a topologically robust non-equilibrium phenomenon unobserved so far. However, our model also has its own limitations, as we have not introduced other effects as jamming occurring in the system at high densities. This could be a reason why the model can differ quantitatively, but not quantitatively, when high concentrations of particles are used. Nonetheless, the experiments have been carried with different system sizes and particles density, and still we can observe the topological effects over all the cases analysed.

As a final remark, this work has shown how topological matter can be connected with soft matter, and vice versa, and how both fields can teach each other to improve the global knowledge of both fields.

7.6 | Future perspectives

The results presented in this chapter should be considered as a starting point for the development of future structures to accumulate particles and to obtain more experimental proofs of topological insulators theory. During the development of the experiments, we considered using other devices, which we finally did not include in the final analysis because the material we already had was sufficient to demonstrate the hypothesis.

However, we also proposed the existence of broken rows. Since the topological properties should not be removed by stretching the mask, these changes in the mask should not affect the results. In Figs. 7.6 B), C) and D) I show the possible editions we considered for the trivial scenario and large device (Fig. 7.6 A)). We also had the same designs for the topological device. For the small device, we only had the analogue to design B). These designs could be useful to develop more complex patterns. We also included the possibility of adding a flower shape surrounding the designs to maintain a more constant density of particles. The idea of the flower is to act as a wall, and hence, there will not be a flow of particles between the inner and the outer part of the flower, avoiding the introduction of particles with time to our device. In our designs, we had up to 6 patterns per microfluidic well (see section 2.2, Fig. 2.2), and hence we considered at first to draw each flower around each 6 patterns. Finally, we thought it was a better idea to consider them at the individual level because of a better control of particles, even that this design would reduce the number of patterns to have per microfluidic well.

As a step further, we have considered the idea of building these microchannels using a 3D printer instead of a lithographic system. This setup would require using a 3D printer able to print at the resolution level of the micrometres, but we think it could be technically possible, if not now, in the near future. This could allow us not only to further move and accumulate the particles, which could take analytes from the solution, in 3D but also, to study topological properties of 3D systems, something which is at a higher level of complexity.

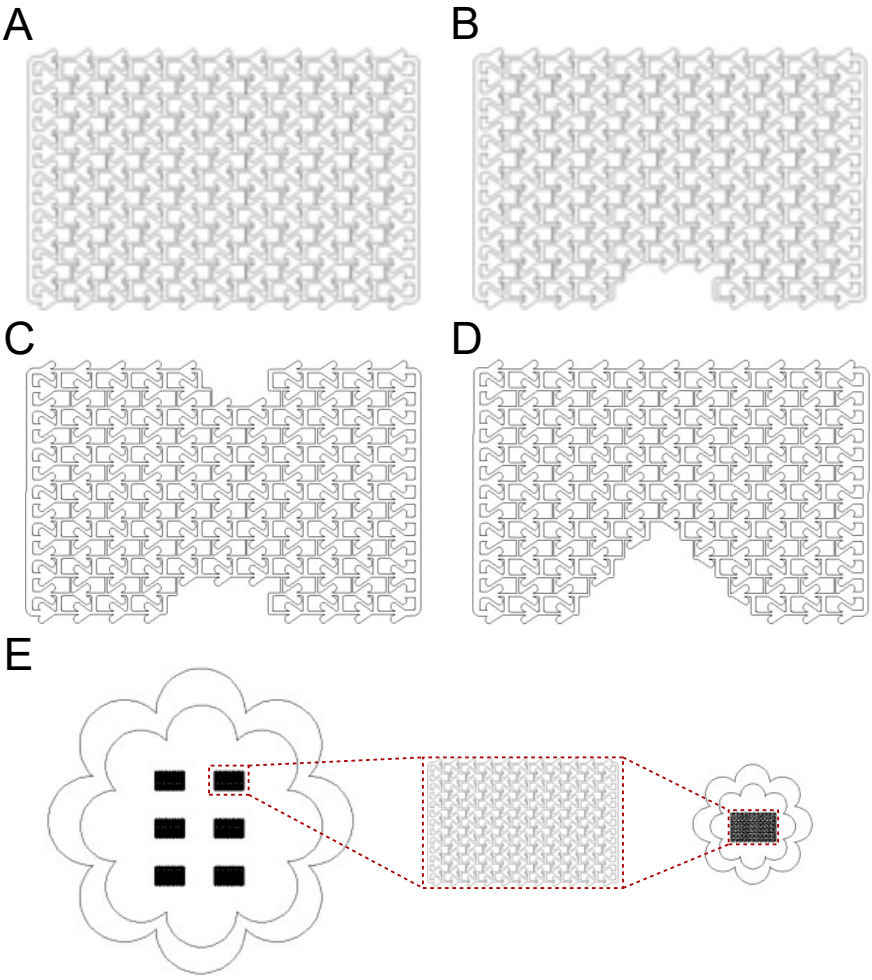


Figure 7.6: Next designs to be studied. A) Original trivial design. **B)**, **C)**, **D)** designs with removed unit cells at the borders of the device. **E)** Designs to have constant density of particle.

Appendix

7.A | Experimental values

We fix the values of the parameters in Eq. (7.2.2) by counting the number of times we see a particle moving along the four types of bulk links in the experiment. We have performed the experiment for two device sizes, $(L_x, L_y) = (12, 6)$ and $(L_x, L_y) = (13, 14)$ (see Figs. 7.10 A), B), C), D)) and, for a low and a high density of active particles (see section 7.2.1). For each situation we obtain the following transition probabilities:

1. For our largest device and a low density of active particles:

   - $(t_1, t_2, t_+, t_-) = (0.154, 0.212, 0.512, 0.122)$ for the trivial device.
   - $(t_1, t_2, t_+, t_-) = (0.214, 0.128, 0.545, 0.112)$ for the topological device.

   These are the values we use to compare our experiment with the model in Figs. 7.4 C), D).

2. For our largest device and a high density of active particles:

   - $(t_1, t_2, t_+, t_-) = (0.126, 0.206, 0.461, 0.207)$ for the trivial device,
   - $(t_1, t_2, t_+, t_-) = (0.203, 0.119, 0.466, 0.212)$ for the topological device.

   These are the values we use to compare our experiment with the model in Figs. 7.5 E).

3. For our smallest device and a low density of active particles:

   - $(t_1, t_2, t_+, t_-) = (0.150, 0.199, 0.519, 0.132)$ for the trivial device,
   - $(t_1, t_2, t_+, t_-) = (0.179, 0.123, 0.579, 0.119)$ for the topological device.

4. For our smallest device and a high density of active particles:

   - $(t_1, t_2, t_+, t_-) = (0.143, 0.191, 0.514, 0.152)$ for the trivial device,
   - $(t_1, t_2, t_+, t_-) = (0.215, 0.125, 0.541, 0.120)$ for the topological device.

We observe an asymmetry between vertical and horizontal motions since $t_1 + t_2 = 0.33 < 0.67 = t_+ + t_-$; the motion along the horizontal axis is easier because the ratchets are aligned while vertical micro-channels are not. This asymmetry in the design is a consequence of the spatial constraints during device fabrication and it also helps to distinguish better topological and trivial devices in the experiment. Indeed, the contribution of topological edge modes to the displacement of particles is largest for an asymmetric network with $t_1 + t_2 < t_+ + t_-$, because it increases the decay time $\tau_d = \tau/(t_1 + t_2)$ of the edge modes. This condition is satisfied in the present experiment because the ratchets favour the horizontal motion of the Janus particles. Also, one can expect to observe a ballistic regime after a time

$$\tau_b = (t_+ + t_-)\tau/(t_+ - t_-)^2, \tag{7.A.1}$$

after which $\langle x \rangle > \Delta x$. Since in our experiment $\tau_b/\tau_d = 1.3 > 1$, it allows us to observe the ballistic regime.

Another parameter that enters our model is the typical time, τ, for a particle to move from one cell to another. We evaluate this time separately for each active particle by dividing the total time of its trajectory by the number of times it goes from one cell to another. This time is different for each active particle for a variety of reasons, for example because of the differences in particle sizes or local chemical environment. We evaluate the probability distribution of τ, $\mathcal{P}(\tau)$, for each initial configuration we pick up (see Fig 7.7). When we compare our model with the experiment, we average all the quantities over the probability distribution of τ. For example, for the average position we compute:

$$\langle\langle x \rangle\rangle = \sum_{\tau} \langle x \rangle \mathcal{P}(\tau), \tag{7.A.2}$$

where the average position $\langle x \rangle = \sum_{\sigma ij} x_{\sigma ij} P_{\sigma ij}$ is evaluated with the probability distribution of our model, in Eq. (7.2.2), for a given value of τ. In the text we omit the double bracket notation, but this averaging procedure is always performed.

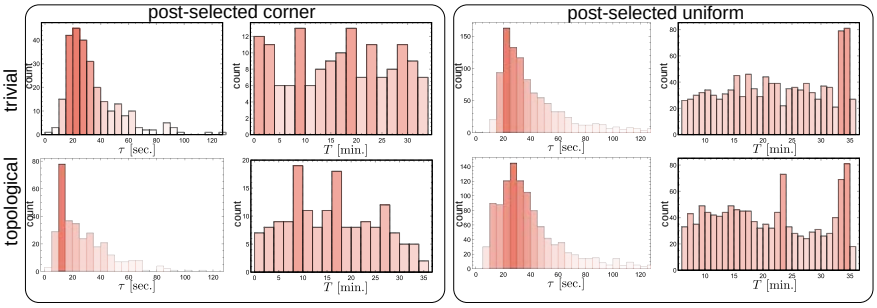


Figure 7.7: Distribution of characteristic times for each selection of particles for the devices with $(L_x, L_y) = (13, 14)$ and a small density of active particles, for both topological and trivial devices. The timescale τ corresponds to the average time for a particle to go from one cell to the next. The timescale T corresponds to the total length of a trajectory.

7.B | Developing the model

7.B.1 Bulk solution

In the situation of a infinite or periodic lattice, we can decompose the solution over the basis of Bloch solutions such that $P_{\sigma ij} = \sum_{k_x,k_y} P_\sigma(\mathbf{k}) e^{i(k_x i + k_y j)}$ where $\{k_x, k_y\} = \{n\pi a/L, m\pi a/L\}$ with a the lattice spacing, L the lattice size and $\{n, m\} \in \mathbb{N}^2$. In this basis Eq. (7.2.2) can be written for each $\mathbf{k}$ independently, $\tau d\mathbf{P}_\mathbf{k}/dt = \hat{W}_\mathbf{k}\mathbf{P}_\mathbf{k}$, with:

$$\tau \frac{d}{dt}\begin{pmatrix} P_A(\mathbf{k},t) \\ P_B(\mathbf{k},t) \end{pmatrix} = \begin{pmatrix} -1 + t_+ e^{-ik_x} + t_- e^{ik_x} & t_1 e^{ik_y} + t_2 \\ t_1 e^{-ik_y} + t_2 & -1 + t_+ e^{ik_x} + t_- e^{-ik_x} \end{pmatrix} \begin{pmatrix} P_A(\mathbf{k},t) \\ P_B(\mathbf{k},t) \end{pmatrix}, \tag{7.B.1}$$

which we can diagonalize to write:

$$\tau \partial_t P_\eta(\mathbf{k},t) = \lambda_\eta(\mathbf{k}) P_\eta(\mathbf{k},t), \tag{7.B.2}$$

where $\eta = \pm$ denotes the two eigensolutions with normal mode P_η and eigenvalue λ_η. The eigenvalues are:

$$\lambda_\eta = -1 + (t_+ + t_-)\cos(k_x) \pm \sqrt{(t_2 + t_1\cos(k_y))^2 + t_1^2 \sin^2(k_y) - (t_- - t_+)^2 \sin^2(k_x)}. \tag{7.B.3}$$

The time evolution of the normal modes is then $P_\eta(\mathbf{k},t) = P_\eta(\mathbf{k}) e^{\lambda_{\eta\mathbf{k}} t}$, so if we define the distribution at $t = 0$ by:

$$P(\mathbf{x}, t=0) = \sum_\mathbf{k} \sum_\eta c_\eta(\mathbf{k}) P_\eta(\mathbf{k}) e^{i(k_x x + k_y y)}, \tag{7.B.4}$$

then the probability distribution at a time t is:

$$P(\mathbf{x},t) = \sum_\mathbf{k} \sum_\eta c_\eta(\mathbf{k}) P_\eta(\mathbf{k}) e^{\lambda_{\eta\mathbf{k}} t + i(k_x x + k_y y)}. \tag{7.B.5}$$

Since $\mathrm{Re}(\lambda_{\eta\mathbf{k}}) \leq 0$, with a maximum at $\mathbf{k} = 0$ with $\mathrm{Re}(\lambda_{\eta\mathbf{k}}) = 0$, the infinite or periodic lattice tends towards the uniform distribution in space at long times.

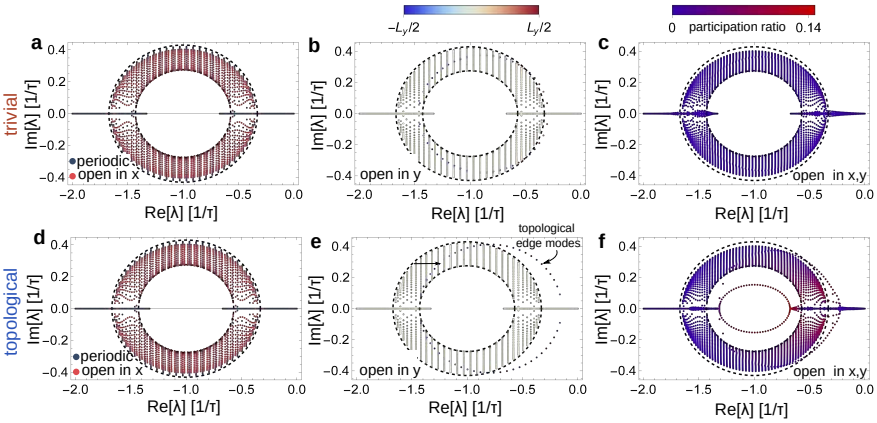


Figure 7.8: Parametric representation of the real and imaginary part of the spectrum of the normal modes for the trivial **A)**, **B)**, **C)** and topological **D)**, **E)**, **F)** devices, for (L_x,L_y)=(L,L) = (70,70) lattice sites. **A)**, **D)** Spectrum for periodic boundary conditions in both directions (in blue) superimposed with that for open boundary conditions in x (in red). The two spectra coincide and indicate no edge mode. **B)**, **E)** Spectrum for open boundary conditions in y. The points are coloured with respect to the average position $\langle y\rangle$ of the normal mode. The topological edge modes (7.B.22) partly hybridize with bulk modes, this is seen as a gap opening for larger values of $\mathrm{Re}[\lambda]$. **C)**, **F)** Spectrum for open boundary conditions in x and y. The points are coloured with respect to the participation ratio $\sum_{\sigma ij}|P_{\sigma ij}|^4$ of the normal mode, a quantity which is small for delocalized modes. As shown in the main text, the modes in the inner dashed circle are localized at the top right and bottom left corners and are related to the second order non-Hermitian skin-effect.

7.B.2 Real and imaginary eigenvalues

The imaginary and real parts of the eigenvalues λ_n determine the average position and variance of the active particles. Indeed, for an initial probability distribution that is uniform and that we decompose over the eigensolution, $P(\mathbf{x},t=0) = \sum_{\mathbf{k}}\sum_{\eta} c_\eta(\mathbf{k})P_\eta(\mathbf{k})e^{i(k_x x+k_y y)}$, we have:

$$\langle r_i\rangle_t = \sum_{\mathbf{x}} r_i P(\mathbf{r},t) = -\sum_{\eta=\pm} c_\eta(\mathbf{k}=0)\partial_{k_i}\mathrm{Im}\left[\lambda_\eta(\mathbf{k}=0)\right]t, \tag{7.B.6}$$

$$\Delta_t r_i^2 = \langle r_i^2 - \langle r_i\rangle_t^2\rangle_t = -\sum_{\eta=\pm} c_\eta(\mathbf{k}=0)\partial_{k_i}^2\mathrm{Re}\left[\lambda_\eta(\mathbf{k}=0)\right]t, \tag{7.B.7}$$

where $\mathbf{r}=(x,y)$ is the position over cells (see Fig. 7.1 E)). These two relations show that for an initially uniform distribution, only the eigenvalues at $\mathbf{k}=0$ matter. As explained in the main text, the slope of the imaginary part is related to the ballistic motion and the curvature of the real part is related to diffusion.

7.B.3 Edge modes

In this section we derive the spectrum of edge modes for open boundary conditions in the y direction, and periodic in x. Because the lattice is periodic in the x direction, we can still apply the procedure in Sec. 7.B.1 for the horizontal (x) direction. This leads to the equation:

$$\tau \frac{d}{dt}\begin{pmatrix} P_{A,j}(k_x,t) \\ P_{B,j}(k_x,t) \end{pmatrix} = t_1 \begin{pmatrix} P_{B,j+1}(k_x,t) \\ P_{A,j-1}(k_x,t) \end{pmatrix} + \begin{pmatrix} -1+t_+e^{-ik_x}+t_-e^{ik_x} & t_2 \\ t_2 & -1+t_+e^{ik_x}+t_-e^{-ik_x} \end{pmatrix} \begin{pmatrix} P_{A,j}(k_x,t) \\ P_{B,j}(k_x,t) \end{pmatrix}, \tag{7.B.8}$$

where j labels the lattice site in the vertical (y) direction. Since the bulk spectrum (7.B.3) is invariant by the transformation $k_y \to -k_y$, the solutions for open boundaries can be expanded as:

$$P_{A,j} = a_+e^{ik_yj} + a_-e^{-ik_yj} = A_1\cos(k_yj) + A_2\sin(k_yj), \tag{7.B.9}$$

$$P_{B,j} = b_+e^{ik_yj} + b_-e^{-ik_yj} = B_1\cos(k_yj) + B_2\sin(k_yj). \tag{7.B.10}$$

The value of k_y is not necessarily real and is set by the boundary condition. Also, if we write $k_y = \chi + i\mu$, we have that:

$$\cos(k_yj) = \cos(\xi j)\cosh(\mu j) - i\sin(\xi j)\sinh(\mu j), \tag{7.B.11}$$

$$\sin(k_yj) = \sin(\xi j)\cosh(\mu j) + i\cos(\xi j)\sinh(\mu j), \tag{7.B.12}$$

and when replacing these expressions in Eq. (7.B.3) we have to check that the real part of λ_n is negative, so the solutions stay normalized. We consider two types of open boundary conditions in y: (1) one that preserves chiral symmetry and, (2) one that preserves detailed balance.

As we show next, the boundary condition (1) is a useful approximation; its chiral symmetry allows us to show in a simple way why the edge states have a non-Hermitian skin-effect in the next section. The price to pay is that we do not recover the exact numerical edge spectrum, but only an approximation to it. The boundary condition (2) is more involved analytically but recovers the exact numerical edge spectrum with the same topological information, which is discussed in the next section.

(1) Boundary condition with chiral symmetry

A way to describe open boundary conditions is to solve Eq. (7.2.2) with the constraints $P_{B,L_y+1} = 0$ and $P_{A,0} = 0$. These constraints prevent a particle from moving away from the region $y \in [1, L_y]$. The constraint $P_{B,L_y+1} = 0$ implies that:

$$P_{B,j} = B_1 \frac{\sin(k_y(L_y + 1 - j))}{\sin(k_y(L_y + 1))}. \tag{7.B.13}$$

This is input in the equation for $P_{A,0}$ in Eq. (7.B.8) to give the consistency relation:

$$\frac{t_1}{t_2} = -\frac{\sin(k_y(L_y + 1))}{\sin(k_y L_y)} \xrightarrow[Ly\to\infty]{k_y=\pi+i\mu} e^{\mu}. \tag{7.B.14}$$

This transcendental equation has L_y *real* solutions for $k_y \in [0, \pi)$ if only $t_2/t_1 < 1$, this corresponds to the trivial device which has no edge mode. If $t_1/t_2 > 1$, as in the topological device, this equation has only $L_y - 1$ real solutions for $k_y \in [0, \pi)$. The missing real solution is that at $k_y = \pi$. Note that it would be missing at $k_y = 0$ if $t_1/t_2 < -1$, but this does not occur since conditional probabilities are positive. This missing mode is actually substituted by an evanescent mode, with $k_y = \pi + i\mu$, solution to Eq. (7.B.14). In the limit where L_y is large, the first term in the square-root in Eq. (7.B.3) vanishes and the eigenvalues are:

$$\lambda_{\chi=\pm} = -1 + (t_+ + t_-)\cos(k_x) + \chi i(t_- - t_+)\sin(k_x). \tag{7.B.15}$$

They correspond to the top ($\chi = +$) and bottom ($\chi = -$) interface, respectively. Compared to Eq.(7.B.15) the numerical edge modes seen in Fig. 7.8E) are shifted by a constant. To obtain the exact spectrum we discuss the second type of boundary conditions, that preserve detailed balance.

(2) Boundary condition with detailed balance

The previous boundary condition does not respect detailed balance because it removes the hopping term to outside the lattice without tuning-off the associated on-site sink. This on-site sink is the $-1 = -(t_1 + t_2 + t_+ + t_-)$ term in (7.B.1). Imposing detailed balance at the boundary leads to the following condition at the $j = L_y$ boundary,

$$\lambda P_{A,L_y} = (-1 + t_1 + t_+ e^{-ik_x} + t_- e^{ik_x}) P_{A,L_y} + t_2 P_{B,L_y}. \tag{7.B.16}$$

This equation looks similar to the previous one, where we impose $P_{B,L_y+1} = 0$, but with an additional $t_1 P_{A,L_y}$ contribution. This boundary condition can be

worked out with (7.2.2) and (7.B.3) to give:

$$\left[2\cos(k_y) - \underbrace{\frac{t_2}{\lambda - (-1 + t_1 + t_+ e^{-ik_x} + t_- e^{ik_x})}}_{\equiv z}\right] P_{B,L_y} = P_{B,L_y-1}. \tag{7.B.17}$$

The second term in the bracket would be zero in the boundary condition (1). We combine (7.B.17) with Eq (7.B.9) and get:

$$P_{B,j} = B_1 \left(\cos(kj) - \frac{\cos(k(L_y+1)) - z\cos(kL_y)}{\sin(k(L_y+1)) - z\sin(kL_y)} \sin(kj)\right). \tag{7.B.18}$$

We can then insert this in the equation for the $j = 1$ boundary, leading to:

$$\lambda P_{B,1} = (-1 + t_1 + t_+ e^{ik_x} + t_- e^{-ik_x}) P_{B,1} + t_2 P_{A,1}, \tag{7.B.19}$$

and that can be transformed to:

$$\frac{1}{z} = 2\cos(k_y) - \frac{P_{B,2}}{P_{B1}} = \frac{\sin(k_y(L_y+1)) - z\sin(kL_y)}{\sin(k_y L_y) - z\sin(k(L_y-1)} \xrightarrow[Ly\to\infty]{k_y=\pi+i\mu} -e^{\mu}. \tag{7.B.20}$$

This transcendental equation is similar to Eq. (7.B.14) and depending on the value of z, now also a function of λ and k_x, we lose a real-valued solution for k_y at $k_y \approx \pi$. This solution is replaced by an evanescent solution with $k_y = \pi + i\mu$ and in the limit of $L_y \to \infty$, we obtain:

$$e^{\mu} = -1/z = -\frac{\lambda - (-1 + t_1 + t_+ e^{-ik_x} + t_- e^{ik_x})}{t_2}. \tag{7.B.21}$$

Then replacing $\cos(k) \to_{k\to\pi+i\mu} -\cosh(\mu) = \frac{1}{2}(z + 1/z)$ in the equation for λ (7.B.3), we find the two solutions:

$$\lambda_{\chi=\pm} = h_0 + t_1 + \chi\sqrt{h_z^2 + t_2^2}, \tag{7.B.22}$$

where $h_0(k_x) = -1 + (t_+ + t_-)\cos(k_x)$ and $h_z(k_x) = i(t_- - t_+)\sin(k_x)$. These solutions coincide with the numerical spectrum in Fig. 7.8e. The edge mode dispersion, $\lambda_\pm$ is close to that in Eq. (7.B.15) but with a spectrum shifted by t_1. For some values of k_x the spectrum of edge modes (7.B.22) hybridize with bulk modes. Compared to the boundary condition (1), this can be traced back to imposing detailed balance, which adds the t_1 term in (7.B.16), the sink potential.

7.B.4 Topological properties

The probability distribution of active particles is determined by the non-Hermitian matrix $W_{\mathbf{k}}$ in Eq. (7.B.1). Non-Hermitian operators are classified according to their symmetries, to the gap structure of their complex energy spectra and to their dimension [244]. Depending on the discrete symmetries of the system, and excluding crystal symmetries, the model falls in one of the 38 distinct topological classes that define strong topological insulators. The gaps are or three types: real or imaginary line gaps or point gaps. If the class is topologically non-trivial for the given gap type and dimension, then a topological invariant exists to classify the possible topological phases of the system in that class. In our case, the matrix $\hat{W}$ is non-Hermitian with time-reversal symmetry, $W^*_{\mathbf{k}} = W_{-\mathbf{k}}$, and inversion symmetry, $\hat{\sigma}_x \hat{W}_{\mathbf{k}} \hat{\sigma}_x = \hat{W}_{-\mathbf{k}}$. It falls in the real Altland-Zirnbauer (AZ) symmetry class AI. Since $\hat{W}$ has a point gap (see Fig 7.8) and describes a two-dimensional system, this class is trivial and has no strong topological invariant [244].

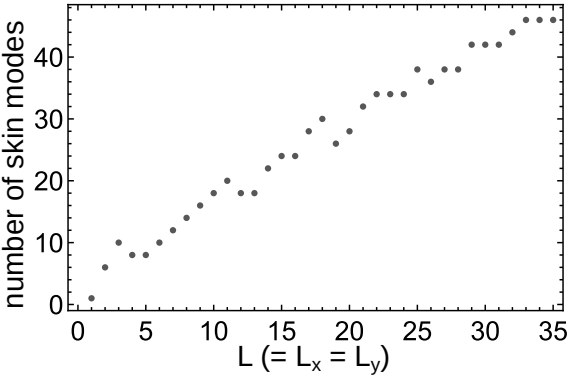


Figure 7.9: Number of skin modes for open boundary conditions in both x and y directions for the topological device, as a function of system size L where $(L_x, L_y) = (L, L)$. This number corresponds to the number of states for open boundary conditions within the point gap found with periodic boundary conditions (i.e. within the inner circle in Fig. 7.8F)). The number of skin modes increases linearly with the perimeter, which is characteristic of a second-order non-Hermitian skin effect [200].

Our model can instead be better understood by analogy to weak topological insulators. Weak topological insulators in d dimensions can be constructed by coupling $d-1$ strong topological insulators. Following this principle $\hat{W}$ is constructed by coupling strong one-dimensional topological insulators defined in the y direction with non-reciprocal hoppings along the x direction. In this way $\hat{W}$ in a direct sum of two terms $\hat{W}_{\mathbf{k}} = \hat{W}_x + \hat{W}_y$ with:

$$\hat{W}_x = \left(-1 + (t_+ + t_-)\cos(k_x)\right)\hat{\sigma}_0 + i(t_- - t_+)\sin(k_x)\hat{\sigma}_z, \tag{7.B.23}$$

$$\hat{W}_y = \left(t_2 + t_1\cos(k_y)\right)\hat{\sigma}_x - t_1\sin(k_y)\hat{\sigma}_y. \tag{7.B.24}$$

In these equations, $\hat{W}_y$ describes a strong hermitian topological insulator in the y direction. It is the Su, Schrieffer and Heeger model [245]. $\hat{W}_y$ has chiral symmetry, represented by $\hat{\sigma}_z$ satisfying $\{\hat{W}_y, \hat{\sigma}_z\} = 0$, and it is time-reversal symmetric, represented by complex conjugation. Therefore, $\hat{W}_y$ belongs to the real AZ symmetry class BDI. Since $\hat{W}_y$ is one-dimensional and has a real line gap (it is Hermitian), it can be classified by a strong topological invariant [246], the total Berry phase on the Brillouin zone. It can be calculated for each sub-band $\eta = \pm$ in (7.B.2) as [247]:

$$\gamma_{y\eta} = \frac{1}{2}\left(\gamma_{y\eta}^{LR} + \gamma_{y\eta}^{RL}\right) = \frac{1}{2\pi}\int_{-\pi}^{\pi} dk_y\, \partial_{k_y}\phi, \tag{7.B.25}$$

where we have introduced the Berry phase on the left (P_η^L) and right (P_η^R) normal modes of $\hat{W}$:

$$\gamma_{y\eta}^{\alpha\beta} = \frac{i}{2\pi}\oint_{\mathcal{C}} dk_y\, P_\eta^{\alpha *}(\mathbf{k})\partial_{k_y} P_\eta^{\beta}(\mathbf{k}). \tag{7.B.26}$$

In this last expression, the normal modes of $\hat{W}$ are:

$$P_\eta^\alpha = \frac{1}{\sqrt{1+e^{2i\alpha\theta}}}\begin{pmatrix} 1 \\ \eta e^{i(\phi+\alpha\theta)} \end{pmatrix}, \tag{7.B.27}$$

with $\alpha = +1$ for right- and $\alpha = -1$ for left-eigenstates and:

$$\phi = \arg\left(t_2 + t_1\cos(k_y) + it_1\sin(k_y)\right), \tag{7.B.28}$$

$$\theta = \arg\left(-\sum_i t_i + (t_+ + t_-)\cos(k_x) + i(t_- - t_+)\sin(k_x)\right). \tag{7.B.29}$$

The topological phase occurs when $t_1 > t_2$ because $\gamma_{y\pm} = 1$, and the system is trivial otherwise, with $\gamma_{y\pm} = 0$. This winding number is independent on $\hat{W}_x$ in (7.B.24), it is the same for both $\hat{W}$ and $\hat{W}_y$. Since the winding number $\gamma_{y\eta}$ defined in Eq. (7.B.25) is the winding number of the Hermitian operator $\hat{W}_y$, we call it the Hermitian winding number, $w_{\mathrm{H}} = \gamma_y$.

The non-Hermitian matrix $\hat{W}_x$ in Eq. (7.B.24) couples each copy of $\hat{W}_y$ in the x direction. Because $\hat{W}_x$ is time-reversal symmetric and has no chiral symmetry, it can be classified in the real AZ symmetry class AI. Also, since it describes a one-dimensional system with an imaginary line gap, it can be classified by a topological invariant. As we now show, both the winding of the spectrum in the complex plane and the total Berry phase vanish for $\hat{W}_x$, which is thus always trivial [247]. The reason the winding numbers of $\hat{W}_x$ vanish can be derived from the fact that $\hat{W}_x$ is composed of two independent copies, one for each eigenvalue $\chi = \pm$ of $\hat{\sigma}_z$,

$$H_{s,\chi}(k_x) = -1 + (t_+ + t_-)\cos(k_x) + \chi i(t_- - t_+)\sin(k_x). \tag{7.B.30}$$

These two copies are non-Hermitian matrices with the same symmetries than $\hat{W}_x$ and each with opposite winding numbers of their spectrum in the complex plane [225, 247, 243]:

$$w_{\mathrm{nH},\pm} = \frac{1}{2\pi i}\int_{-\pi}^{\pi} dk_x \, \frac{d\log\left(H_{s,\pm}(k_x) - E\right)}{dk_x} = -w_{\mathrm{nH},\mp}. \tag{7.B.31}$$

These winding number are $w_{\mathrm{nH},\chi} = \chi(= \pm 1)$ for modes within an ellipse in the complex plane, centred at $\lambda_C = -1$ and with radii $t_+ + t_-$ on the real axis and $|t_+ - t_-|$ on the imaginary axis (the outer dashed black line in Fig. 7.5F)). In general, these two winding numbers cancel out in $\hat{W}$ since the two independent modes of $\hat{W}_x$ are coupled by $\hat{W}_y$ in the y direction. As a consequence, the spectrum for periodic and open boundary conditions in the x direction are similar to each other, so the non-Hermitian skin effect is absent (see Figs. 7.8 B),D)).

As derived in Sec. 7.B.3, in the topological device a y boundary that preserves chirality has two chiral edge modes, described by the two copies (7.B.30) on the top ($\chi = +$) and bottom ($\chi = -$) edges, since the chirality of $\hat{W}_y$ is represented by $\hat{\sigma}_z$. In this case, each edge displays a one-dimensional first-order non-Hermitian skin effect because of their non-zero w_{nH}, resulting in a second-order non-Hermitian skin effect of the two-dimensional system.

Although the boundary condition that preserves chiral symmetry is a useful approximation, it is necessary to impose detailed balance at the top and bottom edge to obtain exactly the edge theory that we obtain numerically, as discussed in Sec. 7.B.3. The main change between these two boundary conditions is that the spectrum of the edge modes shifts along the real axis, and as a result some of the edge modes partially hybridize with bulk modes (7.B.22), see Fig. 7.8E). This effect can be interpreted as an edge potential on the topological edge states, which, despite breaking chiral symmetry, is not expected to change their skin-effect, for moderate parameter values. We confirm this expectation numerically, using the conditional probability parameters relevant for our experiments in Fig. 7.8F).

With the above, our analysis implies that, if chiral edge modes exist with open boundary conditions in the y direction (i.e. when $t_1/t_2 > 1$), these display a non-Hermitian skin effect with open boundary conditions in the x direction, resulting in an accumulation of active particles at the corners. This effect is a second-order non-Hermitian skin-effect [197, 198, 199, 200, 201], which exists when the topological number:

$$\nu = w_{\mathrm{H}} w_{\mathrm{nH}}, \tag{7.B.32}$$

is non-zero. This is the case for our topological devices, but $\nu = 0$ for our trivial devices. The Hermitian topology of $\hat{W}_y$ when $w_{\mathrm{H}} \neq 0$ spatially separates the topological non-Hermitian modes of $\hat{W}_x$ with opposite and non-vanishing w_{nH}, realizing a second-order skin effect.

7.B.5 Topological invariant of the second-order non-Hermitian skin-effect: connection to Hermitian topology

By following Ref. [199] it is possible to understand the second-order non-Hermitian skin effect by connecting with topological Hermitian systems as follows. First we construct an Hermitian matrix from $\hat{W}$ as:

$$H_{\text{eff}} = \begin{pmatrix} 0 & \hat{W} - \lambda_C \\ \hat{W}^\dagger - \lambda_C & 0 \end{pmatrix}, \tag{7.B.33}$$

where λ_C is a base energy defined by the centre of the point gap. This is in general a complex number, but, as can be seen from Fig. 7.8, in our case $\lambda_C = -1$. When λ_C is real, as in our case, it is instructive to perform a unitary transformation of H_{eff} as $\tilde{H}_{\text{eff}} = U H_{\text{eff}} U^\dagger$ by using the unitary matrix:

$$U = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{7.B.34}$$

such that:

$$\tilde{H}_{\text{eff}} = \tilde{H}_x \otimes \tau_0 + \sigma_0 \otimes \tilde{H}_y \tag{7.B.35}$$

$$\tilde{H}_x = (-1 - \lambda_C + (t_+ + t_-)\cos(k_x))\,\hat{\sigma}_x - (t_- - t_+)\sin(k_x)\hat{\sigma}_y, \tag{7.B.36}$$

$$\tilde{H}_y = (-t_2 - t_1\cos(k_y))\,\hat{\tau}_x - t_1\sin(k_y)\hat{\tau}_y. \tag{7.B.37}$$

Both $\tilde{H}_x$ and $\tilde{H}_y$ have chiral symmetry represented by $\hat{\sigma}_z$ and $\hat{\tau}_z$, respectively. Each symmetry is associated to the winding numbers $\tilde{\omega}_{x,y}$. Consequently, $\tilde{H}_{\text{eff}}$ has the symmetry $\tilde{\Gamma} = \hat{\sigma}_z \otimes \hat{\tau}_z$. Additionally it displays inversion symmetry given by $\tilde{I} = \hat{\sigma}_x \otimes \hat{\tau}_x$. Note that $\tilde{\Gamma}$ commutes with $\tilde{I}$.

As discussed in Sec 7.B.4, the present model has trivial non-Hermitian topology in the bulk but non-trivial on the edges. In the nomenclature of Ref. [199] its second-order non-Hermitian skin effect is thus extrinsic, since chirality and inversion symmetries commute and the non-Hermitian topology is only characterized by chiral symmetry $\tilde{\Gamma}$. The corresponding Hermitian winding numbers of $\tilde{H}_x$ and $\tilde{H}_y$ are non-trivial and can be computed from (7.B.25). We have that:

1. the normal modes of $\tilde{H}_x$ are:

$$P_{\eta=\pm} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \eta e^{i\phi_x} \end{pmatrix}, \tag{7.B.38}$$

where $\phi_x = \arg\left[-(1+\lambda_C) + (t_+ + t_-)\cos(k_x) + i(t_+ - t_-)\sin(k_x)\right]$. So the winding number w_x is

$$w_x = \frac{1}{2\pi}\int_{-\pi}^{\pi} dk_x \partial_{k_x}\phi_x. \tag{7.B.39}$$

$w_x = 1$ for $\lambda_C \in -1 \pm (t_+ + t_-)$ and it vanishes otherwise.

2. the normal modes of $\tilde{H}_y$ are:

$$P_{\eta=\pm} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \eta e^{i\phi_y} \end{pmatrix}, \tag{7.B.40}$$

where $\phi_y = \arg\left[-(t_2 + t_1 \cos(k_y)) - i t_1 \sin(k_y)\right]$. So the winding number w_y is:

$$w_y = \frac{1}{2\pi} \int_{-\pi}^{\pi} dk_y \partial_{k_x} \phi_y. \tag{7.B.41}$$

$w_y = 1$ for $t_1/t_2 > 1$, which defines the topological device, and it vanishes in the trivial device.

Comparing to our discussion in the previous section, we see here that the Hermitian winding number, w_{H}, of $\hat{W}$ coincides with the total Berry phase w_y of $\tilde{H}_y$. Moreover, the non-Hermitian winding number, w_{nH}, coincides with the total Berry phase w_x of $\tilde{H}_x$ for eigenvalues λ_C on the real axis, as in our case. The topological invariant we introduced in the main text Eq. (7.4.1) (Eq. (7.B.32) in this Supplementary Material) thus coincides with the one proposed in Ref. [199]:

$$\nu = w_{2D} \equiv w_x w_y, \tag{7.B.42}$$

for $\lambda_C \in \mathbb{R}$. Note that if we had considered $\lambda_C \in \mathbb{C}$, then there would be an additional chirality-breaking contribution $-\mathrm{Im}[\lambda_C]\hat{\sigma}_2 \otimes \hat{\tau}_3$ that enters (7.B.24), that would have prevented this topological classification [199].

7.B.6 Discussion on the topological invariant ν

In Sections 7.B.4 and 7.B.5 we computed the same topological invariant, ν, from two different arguments. One may wonder if this invariant is truly a 2D invariant. First, we note that ν is closer to invariants that characterize weak topological insulators, as it is defined by a combination of lower dimensional (in our case 1D) topological invariants. In this sense, the topological invariant describing our system is not a bulk 2D topological invariant. Rather, the present higher-order topological behaviour emerges fromthe superposition of Hermitian and non-Hermitian 1D topologies in transverse directions, that combined lead to a second-order non-Hermitian skin-effect.

We note as well that ν is not strictly a 1D invariant either since a first order non-Hermitian skin-effect cannot occur in the presence of inversion. As discussed in the main text, in our experiments we enforce detailed balance over the unit cell, so the lattice does not break inversion symmetry, and thus does not strictly allow a first-order skin-effect. Each edge independently may break 1D inversion symmetry, and cause the accumulation. The parameters for which our 2D model allows this to happen are captured by ν

7.C | Comparisons between devices with different dimensions and densities

In this section we review all our results for the trivial and topological devices, in small ($(L_x, L_y) = (12, 6)$) and large ($(L_x, L_y) = (13, 14)$, main text) devices (see Fig. 7.10 A), B), C), D)) and for low and high density of active particles (see section 7.2.1).

7.C.1 Topological chiral edge motion

We initiate each device with particles on the top left corner and compare our experimental data with our model in Fig. 7.10E). In order to decrease the statistical uncertainty, we use the symmetry between opposite corners to include trajectories from the bottom right corner in the figures.

At long times, fewer particles contribute to the average because some have shorter trajectories than others. The agreement between model and theory is generally better for shorter times because the experimental average then includes more particles, from 150 to 300 particles.

7.C.2 Non-Hermitian topology : corner localization

We initiate each device with a uniform density of particles on every cell. The time evolution of the density resembles that in Figs. 7.5 A), B), drawn for the large device with a large density of active particles, resulting in an accumulation of active particles at the corners. This increase in the density of particles at the corners is compared quantitatively using the Shannon entropy, in Fig. 7.10G).

Because the entropy is sensitive to the noise in the density of particles, we reduce it by averaging the density over adjacent cells. The range for this averaging is estimated by a Fourier transform, to determine how much the distribution of particles spreads away from uniformity. We obtain the spread Δk from which we deduce the range of fluctuations $\Delta r = 1/(2\Delta k)$. We then perform the averaging procedure over $3\Delta r \approx 3$ cells for the small devices, and ≈ 2 cells for the large devices. We perform this averaging procedure on both experimental and theoretical data.

The entropy we compute from our model is usually larger than what is observed experimentally. This can be caused by the residual noise from the fluctuations in the number of particles or by correlations between particles, that tend obstruct the motion of each other and lead to jamming. Also, we see a difference between the entropies of the topological and trivial devices when the density of active particles is large. In this case the larger number of particles per cell enables the use of the entropy as a probe of the non-Hermitian skin effect.

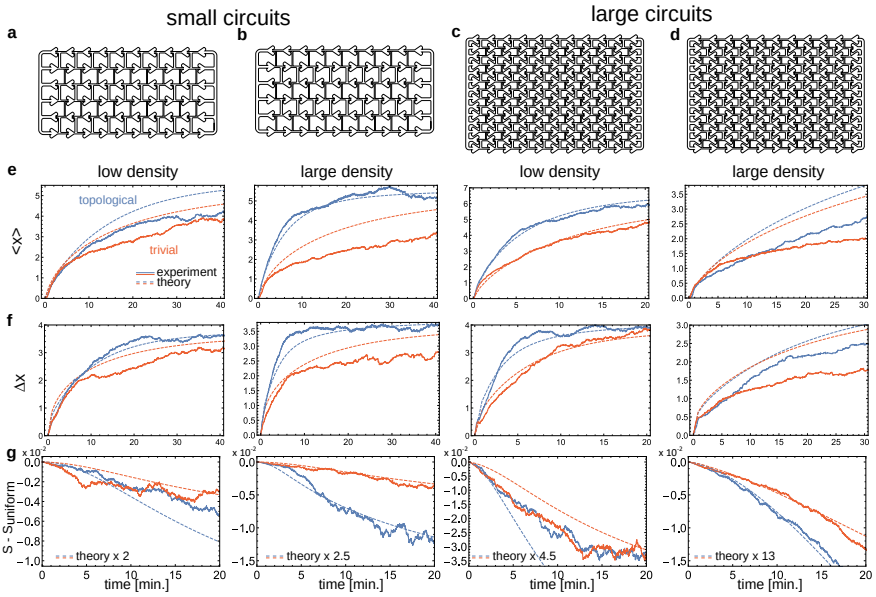


Figure 7.10: A),B),C),D) Depiction of the microfluidic devices used in our experiments. The trivial **A)** and topological **B)** small, $(L_x, L_y) = (12, 6)$, designs, and the trivial **C)** and topological **D)** large, $(L_x, L_y) = (13, 14)$, designs. In the main text we focus on the results obtained for the designs **C)** and **D)**. We consider four experimental situations with small $(L_x, L_y) = (12, 6)$ and large $(L_x, L_y) = (13, 14)$ microfluidic devices, and with low and large density of active particles. A larger density of particles enhances the Brownian motion, but also decreases the velocity of particles and leads to more clusters. **E),F)** Complementing Figs. **??**C), F), we show the the time evolution of the average position and spread for all devices, when tracking particles initially at the top left corner (see Section 7.2.1). We compare these quantities for the trivial and topological devices between experiments and theory. The model parameters are assigned separately for each case, using the experimental data (see Section 7.A), and there are no fitting parameters. **G)** Complementing Figs. 7.5 E), we show the time evolution of the entropy when tracking particles initially spread uniformly over the device (see Section 7.2.1).

Conclusions

The results presented in this thesis demonstrate how active particles behave next to liquid-liquid interfaces and how can we generate collective behaviour by introducing interfaces in the system. These results were only possible after building a set of different tools for carrying out the simulations and experiments needed. These tools will be useful for other researchers too.

In the first part I create different setups for the following experiments, which I will have to record and analyse. To analyse these experiments, it is necessary to obtain different particle parameters, which can include their position, angle, speed, translational diffusion constant, enhanced diffusion constant and rotational diffusion time. Therefore, I also explore which video parameters are appropriate to correctly obtain these parameters, and I create the necessary tools to obtain them. These tools include software to manage digital space and to track particles and the appropriate equations to use. In particular, in this part I demonstrate that:

- 3D printing systems are a great technology for creating reproducible setups easily, quickly, and economically. These systems allow the user to create setups for active particles for long-term experiments even if they generate oxygen.

- When recording a video, its length must be adequate to avoid digital space problems. First, we need to adjust the ROI, time length and FPS of the video accordingly to the characteristic scales of our system. ROI should be limited, but if we cannot select very small ROI because we could mask effects such drifting. Afterwards, we can compress the digital videos using a mp4 format. This conversion might need a specific software depending on the video capture software, as the one developed during this thesis for LeicaX software.

- To analyse particles in videos we need to track their positions. Simple methods can work and can be sufficient under some conditions. However, if we cannot detect particles correctly, we can use a more advanced method to detect particles such a neural network. Neural networks allow us to detect and track particles even if they are complex or difficult to detect by traditional algorithms. The code here provided is an example that can use any researcher.

- To use neural networks we must adequate the dataset and the scheme. Consider adding as many different cases as possible to the dataset, as well as images with no objects to detect to improve network results. If needed, cut the images to a different ROI. Schemes should be as simple as possible: always adapt the anchors and remove layers if possible. Activate always data augmentation.

- After particles are tracked, we can use different algorithms to transform coordinates appropriately such changing the frame of reference or drawing

new cell coordinates visually. A simple solution can be based in python + OPENCV API.

- Once we obtain the position, we can obtain other particle parameters such as its characteristic translational diffusion constant D_t, its enhanced diffusion constant D_{eh}, the speed originated by its activity (v_p), or its rotational diffusion time (τ_r). To obtain these parameters we can fit its MSD to different equations, but before we need to analyse its size. The size of the particle defines its theoretical rotational diffusion time, which determines the proper equation to use. Moreover, to correctly fit this equation we will need to adjust the length and the FPS of the recorded video. For common microscope cameras we have:

  - If particles are smaller than R=0.5 μm, the MSD of the trajectories will adjust to a linear fit, $MSD(\Delta t) = 4D_{eh}\Delta t$, where Δt is the time lag. The enhanced diffusion is the correct parameter to give at this scale.
  - If particles are bigger than R=1.50 μm, the MSD of the trajectories will adjust to a parabolic fit: $MSD(\Delta t) = 4D_t\Delta t + v^2(\Delta t)^2$. At this scale v_p is the correct parameter to give. Although we could present the translational diffusion constant, we should avoid for particles bigger than $R = 1\mu$m because when calculating, the value obtained can be several orders of magnitude bigger than its real value.
  - For particles with sizes between R=0.5 μm and R=3.0 μm, the general expression to fit the MSD can be approximated to a third or fourth order as presented in Eq.3.2.1. As in the previous case, we should avoid using D_t due to its large error when calculating it, at least for particles bigger than R=0.5 μm. In any case, it is better to use the fourth-degree MSD approximation to obtain D_t because it reduces the error of its calculation. Instead of using D_t, it is better to obtain the speed v_p of these particles. To calculate their speed, it is enough to use a third order approximation because the low error when calculating this parameter. Even if we fit data up to the rotational diffusion time, the third order approximation gives accurate results. We can also obtain an experimental rotational diffusion time by adjusting the MSD to the fourth order, adjusting the MSAD or the log of the speed autocorrelation to a linear fit. If possible, both last methods are better, and especially if we can optically track the angles of the particles. The experimental value of the rotational diffusion time can be statistically incompatible with the theoretical value for a free particle, and hence it should be calculated if particles move next to a plane or in complex media.

The conclusions obtained in the first part are used widely for the experiments present in the following parts, but also were used by other colleagues to ease their work.

In the second part I demonstrate the behaviour of active particles next to liquid-liquid interfaces. First, I verify experimentally these systems. To understand better the experimental results, I model them with a simple ABP model, but this model is not powerful enough to explain the results and I create a complex model based on the Lattice-Boltzmann algorithm. As a conclusion:

- When active particles adjacent to a solid surface contact these interfaces, they acquire a common in-plane orientation of -6$\pm$10º deviated from the interface, facing more its inert side towards the oil, and follow the interface's contact line.
- While moving along the interface we find a particle speed reduction. This reduction increases if particle swim slower in the bulk. Hence, we hypothesize this speed reduction is a consequence of Brownian fluctuations reorientations.
- The retention time of particles next to liquid-liquid interfaces is longer than expected, as they stay for longer times compared to the rotational diffusion time. We hypothesize this is a consequence of the active alignment effect of particles next to interfaces. Moreover, the time they stay does not depend on the speed of the particle.
- Particles next to liquid-liquid interfaces move for longer straight trajectories as compared as when they swim in the bulk, but the ratio between both lengths does not depend on the speed of the particle.
- An ABP model with an extra torque applying when particles are near the interface can capture the increase in residence time obtained experimentally. However, if we recover the residence time, we cannot recover the length travelled experimentally with the same model parameters.
- To improve the physical knowledge of these systems we use a Lattice-Boltzmann model. This model allows us to parametrize several forces such as capillarity forces and phoretic forces of the particle, but also to concentrate asymmetrically the product generated by particles in both sides of the interface.
- The modulation of these parameters prove that capillarity forces overcome phoretic forces when an inactive particle is totally hydrophobic (contact angle is 0º) or hydrophilic (contact angle is 180º). However, for a neutral wetting the phoretic forces can move away particles from the expected position set by its wetting. The stronger this force, the further the particle from the expected position will be. This effect is even seen if particles have asymmetric phoretic mobilities. At the same time, the more product generated by the particle is accumulated in one of the liquid phases, the stronger is this phoretic effect.
- When inactive particles with asymmetric phoretic mobilities are set with neutral wetting, particles will orient with the stronger repulsive phoretic

mobility in contact with the liquid phase with less product. The stronger the mobilities, the less time takes for particles to achieve this orientation.

- When activity is added, particles at the interface change their orientation to a stable orientation, which depends on the initial orientation and surface mobility of the particle and the concentration of particle product in both liquid phases. If the asymmetrical concentration of product in both phases is strong enough, particles with the same surface mobility will rotate to the same final angle independently of its initial orientation. However, if the asymmetry decays, different stable orientations are reached depending on the initial orientation. Once they reach this final angle they may move along the interface. The more parallel is their director vector to the interface's contact line, the faster particles move along the interface. If the asymmetry of product in both phases is higher, particles will also move faster.

- If active particles with asymmetric surface mobilities collide with a liquid-liquid interface, we observe that after contacting the interface, they change their orientation, stay at the interface and depending on the wetting and surface mobilities they move along the interface or stop at the interface. Thus, with our Lattice-Boltzmann method we recover the artificial torque introduced manually in the previous ABP model.

After studying the interaction of active particles next to liquid-liquid interfaces we conclude that these interfaces can guide active particles. Guidance is a step crucial for many applications such as microfluidic systems. However, in the second part we only study the behaviour of single particles. But nature is more complex, and we expect to find more than one particle. If we work with multiple particles, we can obtain new phenomena. For instance, with the previous example of the microfluidic device it could be interesting to accumulate particles. Because we obtain guidance either with solid-liquid or liquid-liquid interfaces, and it is easier to build complex structures with solid-liquid interfaces, in the third part I change the interfaces to solid-liquid interfaces to exploit collective phenomena appearing because of the guidance. The experiments in the third part let me conclude that:

- We can obtain accumulation of particles in a microfluidic setup composed of channels even if we start with a pattern that at a priori should not have a net preference for moving particles towards any point of the structure.

- Such system can be composed of units cells with detailed balance, but globally we spontaneously break detailed balance to accumulate active matter. This effect depends on global properties of the setup, and hence we can use topology theory to explain these experiments. Consequently, we can build different structures such as the topological and trivial ones we built.

- When a stochastic model is built, this model can match the experiments extraordinal well considering that we have not used fitting parameters. This model proves that the experimental results are a consequence of extra edge

states in the topological device with respect the trivial one. However, this model also has its own limitations as we did not introduce jamming effects.

- The accumulation of particles is possible as a consequence of a second-order non-Hermitian skin effect, a topologically robust non-equilibrium phenomenon unobserved so far.
- The experiments realized connect Topological matter to soft matter, and vice versa.

As a result, with this thesis we have had an experimental and theoretical view of autocatalytic colloids interacting with interfaces.

"For everything there is a season and a time for every purpose under heaven."

Ecclesiastes

Resumen en castellano

La tesis que actualmente estás leyendo es la culminación de un trabajo que se ha prolongado a lo largo de mis últimos 5 años de vida. Durante estos 5 años me he enfocado en el estudio de la materia activa, un campo de la física en el que se estudia todo aquel ente que es capaz de generar su propio movimiento. Para ello, estos entes utilizan o bien su energía interna o bien capturan energía del exterior, y la convierten en energía cinética. Aunque dicho así pueda sonar algo exótico, en realidad lo que acabas de leer es algo que vemos cada día. De hecho, no sólo lo vemos, pues lo hacemos. Todos los seres vivos en este planeta consumen energía para vivir, y algunos usan parte de esta energía para poderse mover, con lo que estos seres vivos son ejemplos de estos entes que estudia la materia activa. Y no solamente me refiero a los que vemos a nuestro alrededor, o nosotros mismos, sino también de toda esa colección de vida microscópica de la que estamos rodeados y que comúnmente no tenemos presente en el pensamiento, pese a ser el sustento de nuestra vida. Y es precisamente en esta escala microscópica, con entes microscópicos artificiales, en la que me centro en esta tesis.

Seguramente el hecho de que este campo de la física esté tan interconectado con la vida es algo que hizo que acabara decidiéndome por estudiarlo. De algún modo, mi interés a lo largo de mi vida académica no solamente se ha centrado en la física, sino también en la biología. Los conceptos aprendidos durante mis estudios de Biotecnología me han dado un bagaje de fondo que han complementado a los aprendidos en Física. Quizás sea por ello por lo que, cuando entré en el doctorado, no quería hacer algo meramente teórico, sino que quería toquetear, hacer cosas que pudiese ver, que fuesen reales. No digo ni mucho menos que la teoría sea irreal, pero, al fin y al cabo, el lápiz y el papel es un espejo que sirve para describir la realidad, realidad que en algún momento tienes que mirar. Ahora bien, mi parte de física me decía que también se pueden abstraer los conceptos que ves y describirlos con un modelo. Por eso, cuando empecé esta tesis decidí hacer algo mixto. Haría experimentos, sí, pero no me alejaría tampoco de una fuerte abstracción. En consecuencia, la tesis que leerás a continuación tiene algo de ambos mundos. Así, mientras el capítulo 5 versa sobre experimentos, el capítulo 6 lo hace sobre un modelo teórico de lo visto experimentalmente en el anterior capítulo. Dada la complejidad que tuvo hacer dicho modelo, preferí seguir una línea más experimental, y de ahí que el último capítulo lo acabase cerrando con experimentos de nuevo.

En estos tres capítulos me centro en lo que considero la piedra fundamental de mis estudios de doctorado: el estudio de estos entes microscópicos en presencia de interfases. Aunque estos entes podrían haber sido microorganismos, cultivarlos es esclavizarse al laboratorio. Por ello, cuando supe que podía describir estos organismos con entes artificiales me interesó. Es así como para esta tesis acabé trabajando con unas micropartículas esféricas, de 2 y 5 μm de diámetro, de sílice y medio recubiertas por una fina capa de 10 nm de platino (Pt). Cuando estas partícu-

las están en presencia de agua oxigenada, el platino lo descompone rápidamente a agua y oxígeno, y el campo de oxígeno que se crea alrededor de la partícula es responsable de propulsar estas partículas mediante un proceso que llamamos auto difusoforesis. Por supuesto hay muchos más tipos de partículas, pero estas son ideales como modelo teórica como experimentalmente, pues su geometría es simple, funcionan bien, son fáciles de hacer y se conservan bien. Así pues, se resta dificultad en el proceso de creación y me permitía estudiar un sistema más complejo que incluye las interacciones con interfases.

¿Por qué interfases? Pues porque la naturaleza está repleta de ellas. A veces, por simplificar los estudios, los experimentos se realizan en un medio libre de cualquier interacción buscando demostrar que has encontrado una partícula artificial que se mueve mejor, o diferente. Estos estudios son necesarios como primer paso, pero para reflejar la totalidad del movimiento de estas partículas hay que incluir su interacción con interfases. Al menos, sus análogos vivos continuamente interaccionan con ellas. Así pues, empecé mi tesis estudiando sus interacciones en interfases. Como estas partículas se mueven en un medio líquido, una de las dos fases que forma la interfase debe ser un líquido. Cuando entré en el laboratorio había un estudiante que centraba su tesis en la interacción de interfases líquido-solido, por lo que se decidió que yo seguiría su estudio con interfases líquido-líquido, experimentos que describo en el capítulo 5. En presencia de interfases líquido-líquido, estas partículas se comportaban de manera similar que en presencia de interfases líquido-solida, a pesar de la diferente naturaleza de estas interfases. Al tiempo que realizaba estos experimentos empecé a desarrollar un modelo computacional que pudiese explicar lo que estaba viendo, que concluyó con lo expuesto en el capítulo 6. Con este modelo intentábamos capturar todos los efectos hidrodinámicos y foréticos que hubiese en nuestro problema, pese a que con ello tuve que descubrir lo que significan dos palabras: Lattice-Boltzmann.

Una de las conclusiones más evidentes que pudimos obtener es que las interfases pueden ser útiles para guiar partículas, pues al llegar a las interfases se acumulan y se mueven a lo largo de ellas por un tiempo mayor al predicho. Este hecho puede ser muy interesante tanto para explotarlo a nivel de aplicaciones comerciales como para estudiar interacciones de múltiples partículas de los que se pueden obtener diversos efectos colectivos. En el primer caso, una utilidad práctica pudiera ser el uso de estas partículas en canales microfluídicos, necesarios por ejemplo para lab-on-a-chips que pudiesen detectar analitos contaminantes de líquidos o enfermedades a partir del análisis de muestras biológicas. En cuanto a los efectos colectivos, dado que los análisis de los capítulos 5 y 6 los realicé a nivel de partículas individuales, creía conveniente dar un paso más allá y buscar algún resultado colectivo. Sin embargo, construir estas interfases con una geometría algo más compleja a lo que había usado resultaba difícil a nivel experimental, por lo que pensé en retomar las interfases sólido-líquidas.

En consecuencia, en el capítulo 7 doy un vuelco y uso interfases líquido-sólidas

en presencia de muchas partículas. Por el camino tuve la suerte de poder colaborar con un grupo externo localizado en Grenoble con el que, a pesar de la distancia, pude trabajar desde Barcelona con ellos y realizar los experimentos que consideraban necesarios para dar respuesta a su teoría. En realidad, aunque con este último capítulo me centré en la parte experimental, no puedo negar que me sirvió también a nivel teórico, pues aprendí sobre topología y aislantes topológicos, de los cuales desconocía. Con este último trabajo pudimos ver como el uso de las interacciones de estas partículas con interfases nos permite estudiar con materia activa otros campos de la física tan alejados como el de la topología, y como es posible conseguir ver efectos predichos en la teoría de aislantes topológicos experimentalmente con un modelo de "juguete" basado en partículas activas.

Aunque estos tres capítulos centran el grueso de mis estudios, por el camino ha habido otros proyectos fructíferos, y otros que por desgracia no dieron ningún resultado. Por ejemplo, una vía de avanzar en estas interacciones era con el uso de líquidos no newtonianos, pero continuamente veía que las partículas eran arrastradas por el medio y no podía hacer un estudio de su interacción propiamente. La línea de investigación con más de una partícula en interfase líquido-líquido también la intenté usando gotas de lípidos en agua, pero por una parte tuve problemas para crear el dispositivo, y por otra tenía un sistema que no era capaz de hacerlo reproducible. Más adelante ideé otra idea, pero ya consideré que era tarde para seguir con esta investigación dado el año en el que estaba y debido a la COVID-19. Estos últimos intentos sin embargo están recogidos al final del capítulo 5.

El estudio con microcanales también tuvo sus problemas. Por ejemplo, en un principio pensé en estudiar fenómenos de embotellamiento en microcanales, tanto a nivel experimental como con el modelo computacional, lo cual es algo muy interesante a tener en cuenta a la hora de tener un dispositivo experimental. Una de las cosas que me frenaron fue la acumulación de oxígeno que podría dar problemas de deriva con el uso de estas partículas. Como alternativa intenté usar partículas de sílice con estreptavidina en superficie a las que enganchar de forma muy sencilla ureasa biotinilada (por unión biotina-estreptavidina), lo cual podría ser también muy útil para enganchar otras cosas como mRNA biotinilado con fluoróforos, útiles para detectar analitos en suspensión, pero las partículas caían por su peso y se adherían a la superficie en lugar de moverse. Otro tipo de partículas que intenté usar fueron las basadas en dióxido de titanio, pero nunca logré una muestra que fuese capaz de moverse.

A nivel más teórico, también intenté estudiar las interacciones foréticas de estas partículas entre sí con pocillos en los que colocar partículas de forma individual, pero también tuve que detenerlo debido a la COVID y estar ya en mis últimas etapas de mi doctorado.

Además de estos intentos, el camino hacia la resolución de los 3 capítulos principales también ha llevado hacia otros caminos inesperados que finalmente

han dado éxito. Prueba de ello queda reflejado en los capítulos 2, 3, 4 y el anexo C, que aunque no sigan la temática por la que empecé el doctorado, han sido trabajos paralelos consecuencia de los anteriores. Cuando empecé mis investigaciones del capítulo 5 uno de los problemas encontrados fue la grabación de vídeos por el tamaño que tenían. Problema recurrente entre los demás usuarios del equipo de grabación del microscopio. Todos los vídeos tomados los graba en un único fichero en formato RAW. A consecuencia de ello, y dado que me gusta programar, acabé haciendo dos programas con los que convertir estos vídeos de RAW a formato comprimido y ahorrar más del 98% de espacio en memoria. Así mismo, a fin de grabar sólo lo necesario, y dado que podíamos optimizar los parámetros a usar para estudiar el movimiento de partículas activas (tiempo de grabación, imágenes por segundo y tiempo de análisis) empecé a investigar en cómo podían afectar estos parámetros al estudio de la velocidad y el tiempo rotacional de las partículas. El estudio a nivel de velocidad no era algo que me afectase tanto a mí, pero mientras estuve trabajando en el capítulo 5 tuve dudas en como calcular, el tiempo rotacional de mis partículas, y aunque tarde, pensé que esto podría ser interesante para otras personas que se enfrenten al mismo problema. Todos estos resultados los enseño en el capítulo 3.

Por otra parte, una de las cosas que observé en el capítulo 5 era la problemática de la acumulación de partículas por el oxígeno que producían, y por tanto trabajar con muchas como hice en el capítulo 7 sería complicado. Sin embargo, pude mejorar los dispositivos experimentales de los que partía y solucionar este problema como presento en el capítulo 2.

En cuanto al capítulo 6, en ocasiones tuve problemas con la representación tridimensional de lo que veía y en como graficarlo. Ninguna de las soluciones que encontré me fueron adecuadas y finalmente acabé desarrollando una herramienta de visualización en 3D que puede ser útil para la comunidad que usa el mismo software de Lattice-Boltzmann que yo usé. A fin de cuentas, acabé introduciéndome en la programación de OPENGL, algo que siempre había querido tratar.

Curiosamente, algo en lo que también había querido trabajar desde que lo descubrí hace años era la detección de objetos con inteligencia artificial. Este tema siempre me resultó difícil para ser autodidacta, pero ante los problemas del capítulo 7 me representaba, no me quedó otra alternativa que intentar volver a investigar en este ámbito. Aunque disponía de un programa con el que hacer el seguimiento de las partículas de mis vídeos, los vídeos del capítulo 7 eran imposibles de analizar, y acabé programando desde cero un programa de detección y seguimiento de objetos usando inteligencia artificial. Quizás cuando supe de esta tecnología era algo demasiado novedoso. A fin de cuentas, me he apoyado en proyectos e investigaciones de hace 3 o 4 años. El cómo se llega a este código, junto a otras manipulaciones de seguimiento que tuve que hacer a lo largo del doctorado, es algo que muestro en el capítulo 4 y en el anexo C.

En consecuencia de todo lo anteriormente expuesto, he considerado dividir esta tesis en 3 partes, precedidas por una introducción general algo extensa y finalizan por unos anexos algo extensos también. Aunque extensos, debido a todos los conceptos presentes en esta tesis es positivo explicar todo con claridad, en especial para otras generaciones a las que le pudiera ser útil la síntesis que he hecho en mi camino. En cuanto a las partes, en la primera doy idea de cómo hacer partículas y dispositivos, los mejores parámetros para grabar, cómo reducir el espacio de memoria y consejos y un sistema avanzado de seguimiento de partículas. En la segunda parte me centro en el estudio de las interfases líquido-líquido a nivel de partícula individual. Y finalmente en la última parte muestro un proyecto a nivel de muchas partículas en el que uso interfase líquido-sólidas.

Espero que seáis capaces de apreciar el esfuerzo depositado, y en especial, que disfrutéis su contenido.

Summary in English

The thesis you are currently reading is the culmination of a work that has lasted throughout the last 5 years of my life. During these 5 years I have focused on the study of active matter, a field of physics in which any entity that is capable of generating its own movement is studied. To do this, these entities use either their internal energy or they capture energy from the outside, and convert it into kinetic energy. Although said like that it may sound something exotic, in reality what you have just read is something we see every day. In fact, we not only see it, we do. All living beings on this planet consume energy to live, and some use part of this energy to be able to move, so these living beings are examples of these entities that study active matter. And I am not only referring to those that we see around us, or ourselves, but also of all that collection of microscopic life that we are surrounded by and that we do not usually have in our minds, despite being the sustenance of our life. And it is precisely on this microscopic scale, with artificial microscopic entities, that I focus on in this thesis.

Surely the fact that this field of physics is so interconnected with life is something that made me decide to study it. Somehow, my interest throughout my academic life has not only been in physics, but also in biology. The concepts learned during my Biotechnology studies have given me a background knowledge that has complemented those learned in Physics. Perhaps that is why, when I entered my doctorate, I did not want to do something merely theoretical, but rather wanted to fiddle around, do things that I could see, that were real. I am not saying by any means that the theory is unreal, but, after all, the pencil and paper is a mirror that serves to describe reality, a reality that at some point you have to look at. But my physics part also told me that one can also abstract the concepts one sees and describe them with a model. So when I started this thesis I decided to do something mixed. I would do experiments, yes, but I would not distance myself from a strong abstraction either. Consequently, the thesis that you will read next has something of both worlds. Thus, while the chapter 5 deals with experiments, the chapter 6 deals with a theoretical model of what was seen experimentally in the previous chapter. Given the complexity of making this model, I preferred to follow a more experimental line, and that is why the last chapter ended up closing it with experiments again.

In these three chapters I focus on what I consider to be the cornerstone of my doctoral studies: the study of these microscopic entities in the presence of interfaces. Although these entities could have been microorganisms, cultivating them is enslaving the laboratory. Therefore, when I learned that I could describe these organisms with artificial entities, I was interested. This is how, for this thesis, I ended up working with spherical microparticles, 2 and 5 μ in diameter, made of silica and a medium coated by a thin 10 nm layer of platinum (Pt). When these particles are in the presence of hydrogen peroxide, platinum quickly breaks it down to water

and oxygen, and the oxygen field that is created around the particle is responsible for propelling these particles through a process we call self-diffusiophoresis. Of course there are many more types of particles, but these are ideal as a theoretical and experimental model, since their geometry is simple, they work well, they are easy to make and they are well preserved. Thus, the difficulty in the creation process was reduced and it allowed me to study a more complex system that includes interactions with interfaces.

Why interfaces? Because nature is full of them. Sometimes, to simplify the studies, the experiments are carried out in a medium free of any interaction, seeking to show that you have found an artificial particle that moves better, or different. These studies are necessary as a first step, but to reflect the totality of the movement of these particles it is necessary to include their interaction with interfaces. At least their living analogs continually interact with them. So, I started my thesis by studying their interactions at interfaces. Since these particles move in a liquid medium, one of the two phases that forms the interface must be a liquid. When I entered the laboratory there was a student who focused his thesis on the interaction of liquid-solid interfaces, so it was decided that I would continue his study with liquid-liquid interfaces, experiments that I describe in the chapter 5. In the presence of liquid-liquid interfaces, these particles behaved in a similar way as in the presence of liquid-solid interfases, despite the different nature of these interfaces. While conducting these experiments, I began to develop a computational model that could explain what I was seeing, which concluded with the discussion in chapter 6. With this model we were trying to capture all the hydrodynamic and pheric effects that were in our problem, although with this I had to discover what two words mean: Lattice-Boltzmann.

One of the most obvious conclusions that we were able to obtain is that interfaces can be useful to guide particles, since when they reach the interfaces they accumulate and move along them for a longer time than predicted. This fact can be very interesting both to exploit it at the level of commercial applications and to study interactions of multiple particles from which various collective effects can be obtained. In the first case, a practical utility could be the use of these particles in microfluidic channels, necessary for example for lab-on-a-chips that could detect analytes contaminating liquids or diseases from the analysis of biological samples. As for the collective effects, given that the analyzes of the chapters 5 and 6 were carried out at the level of individual particles, I thought it appropriate to go a step further and look for some collective result. However, building these interfaces with a somewhat more complex geometry than what I had used was difficult at an experimental level, so I thought about going back to the solid-liquid interfaces.

Consequently, in chapter 7 I flip and use liquid-solid interfaces in the presence of many particles. Along the way I was lucky enough to be able to collaborate with an external group located in Grenoble with which, despite the distance, I was able to work with them from Barcelona and carry out the experiments they consid-

ered necessary to answer their theory. Actually, although with this last chapter I focused on the experimental part, I cannot deny that it also served me on a theoretical level, since I learned about topology and topological insulators, of which I was unaware. With this last work we were able to see how the use of the interactions of these particles with interfaces allows us to study with active matter other fields of physics as far away as topology, and how it is possible to see predicted effects in the theory of topological insulators. experimentally with a toy model based on active particles.

Although these three chapters focus the bulk of my studies, there have been other fruitful projects along the way, and others that unfortunately did not yield any results. For example, one way to advance in these interactions was with the use of non-Newtonian liquids, but he continually saw that the particles were dragged through the medium and he could not make a study of their interaction properly. I also tried the line of research with more than one particle in liquid-liquid interface using droplets of lipids in water, but on the one hand I had problems creating the device, and on the other I had a system that was not capable of making it reproducible. Later I came up with another idea, but I already considered that it was too late to continue with this investigation given the year I was in and due to COVID-19. These last attempts, however, are listed at the end of chapter 5.

The microchannel study also had its problems. For example, at first I thought of studying bottleneck phenomena in microchannels, both at an experimental level and with the computational model, which is something very interesting to take into account when having an experimental device. One of the things that slowed me down was the accumulation of oxygen that could cause drift problems with the use of these particles. As an alternative I tried to use silica particles with streptavidin on the surface to which to bind very easily biotinylated urease (by binding biotin-streptavidin), which could also be very useful to bind other things like biotinylated mRNA with fluorophores, useful for detecting analytes in suspension, but the particles fell under their weight and clung to the surface instead of moving. Another type of particle I tried to use were those based on titanium dioxide, but I never got a sample that was able to move.

On a more theoretical level, I also tried to study the phoretic interactions of these particles with each other with wells in which to place particles individually, but I also had to stop it due to COVID-19 and be already in my last stages of my PhD.

In addition to these attempts, the road to solving the 3 main chapters has also led to other unexpected paths that have finally been successful. Proof of this remains refllisted in the chapters 2, 3, 4 and the annex C, that although they do not follow the theme for which I started my doctorate, they have been parallel works consequence of the previous ones. When I started my research on chapter 5 one of the problems encountered was recording videos because of their size. Recurring

problem among other users of the microscope recording equipment. All the videos taken are recorded in a single file in RAW format. As a result, and since I like programming, I ended up making two programs with which to convert these videos from RAW to compressed format and save more than 98 % of memory space. Likewise, in order to record only what is necessary, and since we could optimize the parameters to be used to study the movement of active particles (recording time, images per second and analysis time), I began to investigate how these parameters could affect the study of the speed and rotational time of the particles. The study at the speed level was not something that affected me so much, but while I was working on the chapter 5 I had doubts about how to calculate the rotational time of my particles, and although it was late, I thought that this It could be interesting for other people who are facing the same problem. All these results are shown in the chapter 3.

On the other hand, one of the things that I observed in the chapter 5 was the problem of the accumulation of particles due to the oxygen they produce, and therefore work with many as I did in chapter 7 would be tricky. However, I was able to improve the experimental devices from which I started and solve this problem as presented in the chapter 2.

Regarding the chapter 6, sometimes I had problems with the three-dimensional representation of what I saw and how to graph it. None of the solutions I found were suitable for me and I eventually ended up developing a 3D visualization tool that can be useful to the community that uses the same Lattice-Boltzmann software that I used. Ultimately, I ended up getting introduced to OPENGL programming, something I had always wanted to try.

Interestingly, something I had also wanted to work on since I discovered it years ago was object detection with artificial intelligence. This topic was always difficult for me to be self-taught, but given the problems of the chapter 7 represented me, I had no alternative but to try to investigate again in this area. Although I had a program with which to track the particles in my videos, the videos in the chapter 7 were impossible to analyze, and I ended up programming a program to detect and track objects using artificial intelligence from scratch . Perhaps when I learned about this technology it was something too novel. After all, I have relied on projects and research from 3 or 4 years ago. How to get to this code, along with other follow-up manipulations that I had to do throughout the doctorate, is something that I show in the chapter 4 and in the annex C.

As a result of all the above, I have considered dividing this thesis into 3 parts, preceded by a somewhat extensive general introduction and ending with somewhat extensive annexes as well. Although extensive, due to all the concepts present in this thesis, it is positive to explain everything clearly, especially for other generations to whom the synthesis that I have made on my way could be useful. As for the parts, in the first one I give an idea of how to make particles and devices, the

best parameters to record, how to reduce memory space and tips and an advanced particle tracking system. In the second part I focus on the study of liquid-liquid interfaces at the individual particle level. And finally, in the last part, I show a project at the level of many particles in which I use the liquid-solid interface.

I hope you are able to appreciate the effort put into it, and especially that you enjoy its content.

Appendices

4

Fluid mechanics

"The real problem is not whether machines think,
but whether men do."

B. F. Skinner

A.1 | The Continuum approach: Navier-Stokes equation

To start with Navier-Stokes equations derivation we will consider the pseudo particles I described in section 1.3. These pseudo particles are a big packet of molecules of volume V that verifies two limits. First, the size of the characteristic linear length for V ($V^{\frac{1}{3}}$), must be much larger than the characteristic length present in our molecular system, which we will refer as δ. For a simple liquid, such as water, we can take the typical interparticle distance (δ = 3Å). By assuming this limit, every property of our new elementary particles will be statistically a microscopical averaging. On the other hand, the size of $V^{\frac{1}{3}}$ cannot be bigger than the order of magnitude of the characteristic properties we want to see in the system, which we will refer as L. For example, particles size in this thesis is around a few micrometres. Thus, to be conservative we could set L = 0.1 μm.

Once the elementary particles are defined, we have two approaches to proceed. The better physical interpretation of the fluid would imply to use some conservation equations to these particles under the framework recalled as *Lagrangian*. Even though, and because how computers solve the fluid, we can also derive a mathematical equivalent by looking at fixed regions of space, under the framework recalled as *Eulerian*. Since it is also easier to derive from the Eulerian perspective, in this thesis I will present shortly the derivation in this framework[1], but I will change to the Lagrangian perspective to give a better idea when necessary. In the Eulerian framework, because we work under the continuum hypothesis, we can divide the space where the fluid is into tiny fixed regions of space of volume V'. In all these regions, conservation of different quantities is applied. To start with this methodology, we need to derive what is known as the *continuity equation*. This equation expresses the accumulation of a quantity inside any fixed region as a function of the quantity that enters or leaves the region and if the quantity is conserved or not.

If we refer to the quantity we want to study as ϕ, one can understand that since the fluid is moving at speed **u**, there will be a net flux of ϕ flowing with speed **u** through a unit of area per unit of time over all the surface S' of the fixed region. To this flux we will refer as **j**. Therefore, $\mathbf{j} = \phi\mathbf{u}$. Because of the flux **j**, inside the fixed region V' there will be a change of ϕ over time. Furthermore, it can

[1] See ref. [84] for the alternative derivation in the Lagrangian framework.

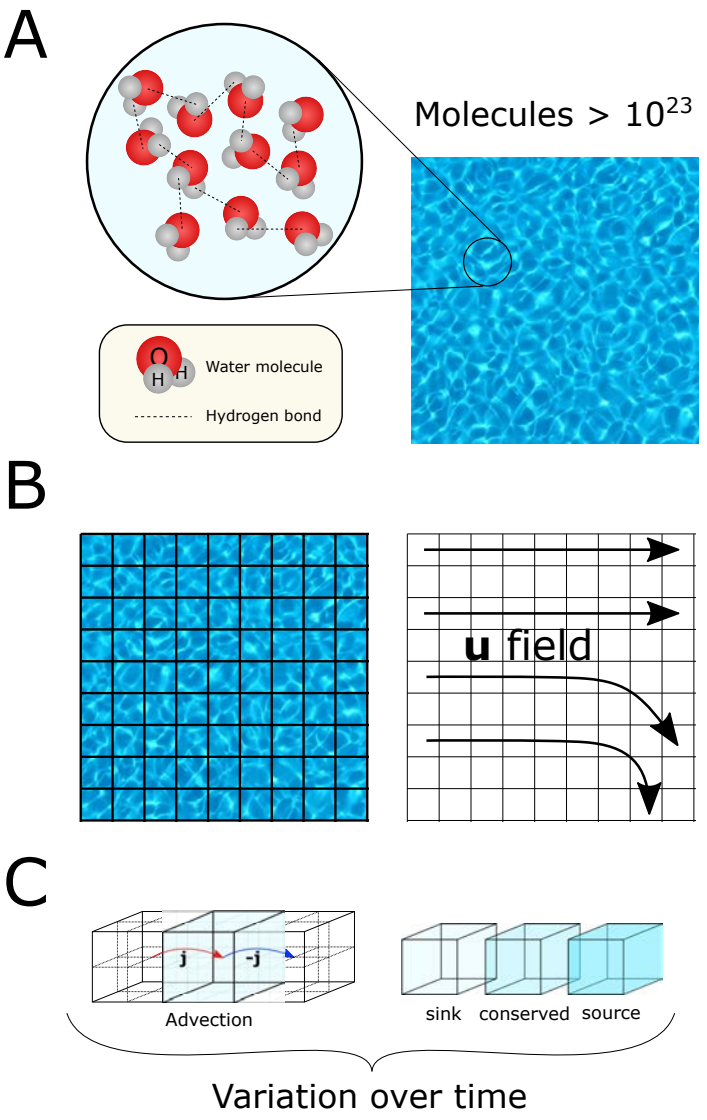


Figure A.1: Model of a fluid. A) A fluid is composed of an incredible number of molecules. In the case of water, water molecules are moving randomly. These molecules can feel other molecules by different potentials such as the hydrogen bonds. **B)** To avoid modelling such number of molecules, we can define it as a continuum and divide the box into a lattice. A great example of this procedure is by solving the Navier-stokes, where the fluid is defined by a vector field $\boldsymbol{u}$ which defines the speed of the fluid. **C)** To solve the Navier-Stokes equation, one needs to apply the continuity equation, which relates the variability of a quantity in one of the discretized cells of the lattice as a factor of the advection (flow coming/leaving from/to other cells) and the creation/annihilation of this substance inside the cell. *Image authorship. Own images.*

be possible that inside V', ϕ is being created or destroyed at a ratio σ per unit of volume and time. If $\sigma > 0$ we say V' is a *source*, while if $\sigma < 0$ it is called a *sink*. In the case that $\sigma = 0$, the quantity ϕ is conserved. Summing all these components, we have:

$$\int_{V'} \frac{\partial \phi}{\partial t} dV + \int_{S'} \mathbf{j} \cdot \mathbf{n} dS = \int_{V'} \sigma dV, \tag{A.1.1}$$

where **n** is a perpendicular vector to the surface S' of the fixed region of space with volume V' and $\partial/\partial t$ is the partial temporal derivative. Equation $A.1.1$ can be rewritten using the *divergence theorem* [248] as:

$$\int_{V'} \left[\frac{\partial \phi}{\partial t} + \nabla \cdot \mathbf{j} - \sigma \right] dV = 0. \tag{A.1.2}$$

Since V' is arbitrary chosen, the previous equation can only be satisfied always if the integrand is 0. Therefore, we have derived the continuity equation:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \mathbf{j} = \sigma. \tag{A.1.3}$$

Now, by using the continuity equation we can apply the conservation of mass and the conservation of the momentum to obtain the well-known Navier-Stokes equation, which models the **u** field and thus, the fluid.

Conservation of mass

We will assume that mass is neither created neither destroyed in the fluid, and thus $\sigma = 0$. Because V' is infinitesimal small due to continuum approach, if we apply this conservation to every region of volume V', we can directly take the density of the fluid instead of the mass. Thus, here $\phi = \rho$. Applying both conditions to Eq. $A.1.3$ we obtain:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0. \tag{A.1.4}$$

Although we may use Eq. $A.1.4$ as it is, it is usually simplified. Density is a property of systems that is usually related with pressure (p) and temperature (T) by means of an equation of state. When one uses systems where the fluid rest at the same temperature, the relation $\rho = \rho(p, T)$ becomes $\rho = \rho(p)$. But now, consider we have a box filled with a fluid inside where we apply a uniform force perpendicularly into each of the walls of the box from outside into inside. In other words, consider we are applying a pressure into the box. If the fluid is a gas, by applying pressure into the box the volume of the box can be reduced because interparticle space between molecules of gas is large, and thus, molecules can get closer each other. Consequently, density is increased. If we stop applying these forces, the volume of the box will be restored, and hence, density will decrease. Thus, gases can be *compressed* or *uncompressed* and therefore, they are *compressible.* But for

liquids, the space between molecules is not as high as for gases, and they are usually not very compressible[2]. Thus, when dealing with liquids we can approximate that $\rho \neq \rho(p)$, or that $\partial_p \rho \approx 0$. Under this scenario we will have that density is almost constant in time and therefore:

$$\nabla \cdot \mathbf{u} = 0. \tag{A.1.5}$$

This is the reason why for many fluids, when Eq. $A.1.5$ is true, we say that the fluid is *incompressible*.

Conservation of momentum

Newton's second law affirms that the change in linear momentum is equal to the forces applied to the body. We can also apply Newton's second law to the fluid. The proper derivation should be applied to every elemental particle, but one can also apply it using the continuity equation in the Eulearian framework since both scenarios are equivalent. To introduce the linear momentum, we can consider that it can be described as $\rho\mathbf{u}$ since we refer always to density instead of mass as already explained. In this case, σ is not strictly 0, because forces are elements of sink and sources. For example, we will always have the gravity force ρg. Furthermore, if we imagine that the fluid is a continuum full of boxes, these boxes will have a normal force between them. This force considers the microscopically diffusion of particles over boxes. Typically, this last term is described as a tensor called *Stress tensor* [3] and it is written as $\mathbf{T}$. Thus, applying Eq. $A.1.3$ we have:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \cdot \mathbf{u}) = \rho g + \nabla \cdot \mathbf{T}. \tag{A.1.6}$$

It is important to note that the Stress tensor gives more unknown quantities than relations we have in the previous equation. To reduce this number of unknown quantities, we can also apply conservation of angular momentum. By applying this conservation, we can find that $\mathbf{T}$ is a symmetric tensor. This will be always true except in the cases where there is a body couple per unit mass such in the case of ferrofluids. Nevertheless, we will have more unknown quantities than relations. To have more relations, we need to apply the conservation of energy to obtain what is known as the *constitutive equations*, but I will not introduce them since they are not needed to obtain the Navier-Stokes equation.

In general, $\mathbf{T}$ is decomposed in a general pressure p acting over each cartesian axis of the volume[4] plus another component called the *deviatoric stress* $\boldsymbol{\tau}$, which

[2] Mostly, liquids moving at speeds much lower than the speed of sound in the fluid are incompressible. This condition is often characterized by what is known as *Mach number (M)*, a dimensionless number which is the ratio between the speed of the fluid ($|\boldsymbol{u}|$) and the speed of sound in the fluid (u_{sound}): $M \equiv \frac{|\boldsymbol{u}|}{u_{sound}}$

[3] See ref. [84] for a good derivation of the Stress tensor.

[4] It is a diagonal 3x3 tensor

takes into account different sources of pressure that apply to the volume in any direction[5]. Thus, we can write:

$$\mathbf{T} = -p\mathbf{I} + \boldsymbol{\tau}, \tag{A.1.7}$$

where $\mathbf{I}$ is the unitary matrix. Although $\boldsymbol{\tau}$ can be very complex, it is often described as a linear function of another tensor called the *rate-of-strain tensor* or $\mathbf{E}$. To understand $\mathbf{E}$, we need to think that when we apply forces to the walls of the volume V, this volume may deform, and this deformation can be different in each axis. From a Lagragian perspective, the reasoning is more intuitive. In the Lagragian perspective, the fluid is defined by *material volumes* and *material points*. A material point is a point that moves with the local continuum $\boldsymbol{u}$. These material points are enclosed into regions of space from where they cannot escape, but these regions can deform with time, and are called as material volumes. The ratio at how this deformation occurs is explained by this tensor $\mathbf{E}$, and mathematically, it is also defined as the symmetric part of $\nabla\mathbf{u}$[6]. For some fluids, the rate of deformation in each direction is proportional to the force applied in that direction. In those cases, we say that the fluid is Newtonian and the constant that relates them, which can be seen as a resistance of a volume to be deformed, is called *shear viscosity* and it is written as μ. Thus, the concept of viscosity of fluids is introduced by the volume deformations that the volume can have because of different forces acting over the surfaces of the volume. In these cases, where also the fluids are incompressible, $\mathbf{T}$ is written as:

$$\mathbf{T} = -p\mathbf{I} + 2\mu\mathbf{E}, \tag{A.1.8}$$

where in this case, p is the thermodynamic pressure of the fluid. For instance, simple fluids, such as water, are Newtonian fluids, and will follow this relation, but others, such as a blood and in general, colloidal systems, are Non-Newtonian and Eq. $A.1.8$ will be more complex.

Combining mass and linear momentum: The Navier-Stokes equation

When one uses the previous equations ($A.1.4$ and $A.1.6$) we say we work with the Navier-Stokes equation, which are the typical equations a physicist would think to model any fluid. In the particular scenario of an incompressible Newtonian fluid, these equations read as:

$$\begin{gathered}\rho\left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = \rho g - \nabla p + \mu\nabla^2\mathbf{u}, \\ \nabla\cdot\mathbf{u} = 0.\end{gathered} \tag{A.1.9}$$

Sometimes, the kinematic viscosity ν is placed in Eq. $A.1.9$. This constant represents the contribution to acceleration of the fluid element due to viscous stresses and is defined as $\nu = \mu/\rho$. Also notice that in some texts, μ is written as η.

[5] It is a 3x3 tensor

[6] The antisymmetric part of $\nabla\mathbf{u}$ is defined as the vorticity tensor $\boldsymbol{\Omega}$, and represents the local rigid-body rotation of two adjacent material points.

A.1.1 Dimensionless Navier-Stokes equation

To obtain a dimensionless Navier-Stokes equation we need to change all variables by dimensionless variables. The typical dimensionless variables are:

- For length, $\boldsymbol{r}^* = \frac{\boldsymbol{r}}{L}$ and $\nabla^* = L\nabla$, where $\boldsymbol{r}^*$ is the dimensionless spatial flow vector, $\boldsymbol{r}$ is the spatial flow vector, L is the characteristic length of our system, ∇^* is the dimensionless gradient operator and ∇ is the gradient operator.
- For flow velocity, $\boldsymbol{u}^* = \frac{\boldsymbol{u}}{U}$, where $\boldsymbol{u}^*$ is the dimensionless flow velocity vector, $\boldsymbol{u}$ is the flow velocity vector and U is the modulus of the characteristic speed of our system.
- For time, $t^* = \frac{t}{L/U}$, where t^* is the dimensionless time and t is the non-dimensionless time.
- For pressure, $p^* = \frac{p}{\rho U^2}$, where p^* is the dimensionless pressure, p is the non-dimensionless pressure and ρ is the density of the fluid.
- For forces, $\boldsymbol{f}^* = \frac{\boldsymbol{f}}{f_0}$, where $\boldsymbol{f}^*$ is the dimensionless force, $\boldsymbol{f}$ is the force and f_0 is the modulus of a characteristic force associated with our force in our system. In the case of the gravity, we would have $\boldsymbol{g}^* = \frac{\boldsymbol{g}}{|\boldsymbol{g}|}$

Introducing these new variables we obtain the dimensionless Navier-Stokes equation:

$$\frac{\partial \boldsymbol{u}^*}{\partial t^*} + (\boldsymbol{u}^* \cdot \nabla)\,\boldsymbol{u}^* = \underbrace{\frac{L|\boldsymbol{g}|}{U^2}}_{1/\mathrm{Fr}^2}\boldsymbol{g}^* - \nabla^* p^* + \underbrace{\frac{\nu}{LU}}_{1/\mathrm{Re}}\nabla^{*2}\boldsymbol{u}^*, \tag{A.1.10}$$

A.1.2 Drag of a sphere moving at constant speed in a flow

Particles at the scale of μm are orders of magnitud bigger than water molecules that surround them[7]. Thus, if we want to analyse how these molecules affect as a drag to the microparticles, it is intelligent to tackle this problem from the continuum point of view of Navier-Stokes, since it is the simplest approach we can use. However, if we look at eq. (1.3.2) we will not be able to see anything related to a particle. Despite of their appearance, the inclusion of an inactive particle is already implicit there. The reason is because this equation gives you an expression for the flow field $\boldsymbol{u}$. Obviously, if there is an impediment that does not allow the fluid to flow, the flow field will have this information. In our case, the inactive particle will act as this impediment, and hence, by introducing the proper boundary conditions we will be able to capture the consequences that introducing an inactive particle has over the flow.

[7] They are at Å scale

If the fluid is flowing, a particle settled in the fluid will flow within it. In such conditions, one could think that there is not much to say about the particle, but for instance, we know that particles do not always have the same density as the fluid has. Hence, the gravity and buoyancy force will act over the particle and it could move with a non-zero speed relative to the fluid, just as active particles will do. Thus, in this scenario, the particle will feel a drag, which can be estimated from the flow field $\boldsymbol{u}$ by calculating the stress of the fluid around the particle. The calculus of this problem was first introduced by Stokes in 1850[249]. To obtain the drag we will consider a few assumptions:

- We will set the reference point of view in the particle of radius R, which is moving at a constant speed V relative to the fluid. Hence, far from the sphere the speed we will see is this V, but on the walls the speed will be zero.
- Since we deal with a sphere, the problem is symmetric along the ψ coordinate, and both V_ψ and the derivates along this direction will be zero.
- We will suppose that the flow is steady and that we are working at low Re. Thus, we will start by using $1.3.6$ without any external forces. The viscous term can be expanded following some mathematical algebra to: $\nabla^2 \boldsymbol{u} = \nabla(\underbrace{\nabla \cdot \mathbf{0}}_{=0}) - \nabla \times (\nabla \times \boldsymbol{u})$
- We will start from the divergence of the speed and the previous rotational. Once we have an equation for $\boldsymbol{u}$, we will obtain the pressure from Stokes equation.
- Finally, we will obtain the stress and we integrate over the surface of the sphere to obtain the force.

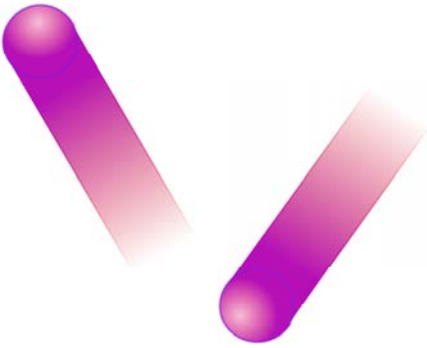
After performing this procedure, we obtain that the drag over the sphere is:

$$\gamma = 6\pi\eta R. \tag{A.1.11}$$

This result will yield valid in case of a particle moving at constant speed and at low Reynolds, which is the condition considered in this thesis. Because the Janus particles used in this thesis move at a constant speed, no matter which is the mechanism of motion behind them, we can freely use this result. If we consider that the relative constant speed is the one due to the activity, we can reuse this drag every time we need to insert a drag force to work with the active particles. Therefore, this drag will appear along this thesis.

B

Thermodynamics



"Everything here appears calculated to inspire kind and happy feelings, for everything is delicate and beautiful."

Washington Irving

B.1 | Macroscopic thermodynamics

The basis of thermodynamics was developed along the XIX century. Before this century, concepts such as free energy or thermodynamic potentials were undefined, and thus it would be impossible to find anything related with them under these names. Today it may be something fundamental that any student can learn in his bachelor but still, although fundamental, the clear concept itself may be forgotten with time if not used. Therefore, here I present a reminder of the basic concepts I considered useful to fully understand this thesis.

B.1.1 Equation of state, state variables and state functions

From a thermodynamic point of view, all the properties of a system can be defined at a specific time by obtaining the value of some measurable variables. These variables are what we know as *state variables* and experimentally, it is observed that one can express a state variable as a function of other state variables. In this sense, we could say that these variables can be *state functions*. Moreover, we can also construct a function which defines the state of the system to which we refer as *equation of state*.

For fluid systems, the typical variables set are the temperature T, the pressure P and the volume V or the volume per mass $v = 1/\rho$, where ρ is the density. As an example of state function we could refer to $P = P(T, v)$ and for equation of state we can refer to the ideal gas law $pV = nRT$, where R is the gas constant[1] and n is the number of moles of the substance.

Nonetheless, state variables, state functions and the equation of state describe the system in *equilibrium*, this is, the system does not have net fluxes of matter or energy, and they do not depend on the path followed by the system to achieve the equilibrium. When a property or a function depends on the path, we say they are process quantities and process functions, respectively. For example, heat and work are process quantities. To avoid confusions, when recalling infinitesimal quantities, we note state variables with d, while process variables are marked with δ. E.g., work would be δW, but pressure is dP.

[1] 8.314 $JK^{-1}mol^{-1}$

B.1.2 Thermodynamic potentials and free energy functionals

There are a few state functions that we refer to them as *thermodynamic potentials*. These state functions are not usually considered as observables but can be constructed from observables as a state function.

The discussion of these potentials started by the work of Thomson in 1855[250], and soon was extended by Massieu in 1869[251]. In fact, there are different thermodynamic potentials, but all of them are connected following *Legendre transformations*, which transform independent variables into dependent and vice versa. Depending on the problem we look for, different conditions are given, and hence different natural variables are used. Therefore, will decide which thermodynamic potential to use depending on these conditions. The common thermodynamic potentials are:

- The internal energy U
- The Gibbs free energy G
- The Helmholtz free energy F or A[2]
- The enthalpy H

We usually look for the difference of these potentials between two thermodynamic states because they give us relevant information for the system.

Internal energy

Suppose we have a glass full of water, that we put on a table. Apparently, the total energy of this glass is zero, because it is not moving and hence, its kinetic energy is zero. Moreover, if we choose the zero reference for the gravitational potential at the level of the table, the glass neither has potential energy. But the glass with the water is full of molecules that are moving randomly and interacting each other. And each molecule has its own translational, rotational, and vibrational kinetic energy. Plus, each molecule has its own potential energy coming from intermolecular forces. In fact, the glass is full of energy!

To refer to this energy, that apparently one could not see from the macroscopic point of view, we use the term *internal energy* or U. The amount of internal energy of a system is very important because limits other properties of the system. For instance, the energy one needs to increase one degree of temperature a substance, also known as the *heat capacity*, depends on U. If one compares water, which has higher potential energy from its interactions, and copper, which has less, one can see that the specific heat capacity[3] for water is 10 times more than for copper[252].

[2] From the German *Arbeit* , which means work.
[3] Heat capacity of a sample divided by its mass.

Alternatively, one could also imagine that the internal energy is the total energy needed to create a specific system, without changing temperature or volume.

Free energy: Gibbs and Helmholtz potentials

Traditionally, two definitions to define the term free energy exist, depending on the working conditions of the system:

- The *Gibbs free energy* , represented by G, gives the amount of non-expansionary work a system can do, this is, the work that can do not related with pressure-volume work, at a constant temperature and pressure. For example, if in a reaction the system changes its volume, this free energy will not consider the work done for expanding/compressing the system. Mathematically:

$$G = U - TS + pV, \tag{B.1.1}$$

  where T is the temperature, S the entropy, p the pressure and V the volume.

- The *Helmholtz free energy* , represented by F, or sometimes A[4], gives the amount of non-expansionary work a system can do, this is, at constant temperature and volume. Mathematically:

$$F = U - TS. \tag{B.1.2}$$

B.1.3 Chemical potential

In the previous subsections I already used the term potential energy because it is common in physics, but I did not define it. However, to understand well the concept of chemical potential this concept should be clear. The concept of potential energy was born just a few years before the Gibb's energy was defined, around 1850 by W. Rankine[253], but in its essence, one even could see Aristotle's philosophy. Aristotle used to use the concept of *potentially* as what a body can be in the future, but it is not now, and soon you will understand why both terms are related.

In nature, bodies can interact with forces over long ranges. Hence, they can be displaced by these forces, and consequently, a total energy in concept of work will be done. But the work caused by some of these forces only depends on the initial and final positions of the bodies. When we have such a kind of forces, we say they are *conservative*, and the energy used as work to displace these objects relative to each other is a *potential energy*. This potential energy is an intensive property of the system, since is an energy divided per unit of matter. For example, gravitational and electric forces are conservatives, and we can define both gravitational and electric potentials, which uses the mass and the charge respectively as

[4]Because of the German *Arbeit* , which means *work* .

unit of matter. Thus, if we increase the size of the system, these properties will not change. In somehow, one could think that this energy is stored in the system, and it will be used at some moment. For example, if we hit slightly a rock in a cliff, it will suddenly acquire a large kinetic energy, which comes from the stored gravitational potential energy. The gravitational potential had the potential of becoming a kinetic energy. We can observe two interesting facts about potential energy from the previous example. The first is that the potential energy depends on the relative distance between bodies (e.g., Earth-rock). The second is that in those systems where potentials can be defined, there can be regions of different potentials levels. The potential energy on top of the cliff was bigger than when it fell. When there are gradients of potential, bodies move from regions of high potentials to low potentials.

As we can define a potential for gravity, in thermodynamics it can be defined a potential for the Gibbs free energy, which can be also redefined as a chemical energy. This potential defines what the chemical compounds of the system can be, and it is called *chemical potential* or μ. Here, the unit of matter is the N molecules in the system. For example, molecules in a system can react and become new species. Thus, the chemical potential expresses the potentially of having a chemical reaction, but also, since mathematically $\mu N = G = U - TS + pV$, it is often expressed as the necessary energy to introduce a molecule in the system under constant pressure and temperature. Therefore, by construction, one can see that the chemical potential is the derivative of Gibbs free energy when changing the number of molecules under constant T, p. But one can also define it for T, V constants by using the Helmholtz free energy.

Although one can always define a mean value of the chemical potential for a system, this value will always come from the individual chemical potentials present in the system. Each different molecular specie will have its own chemical potential. When a system reaches the equilibrium, the different molecular species necessarily will equal their chemical potentials since the net chemical reaction must be zero.

B.2 | Phase transitions

B.2.1 Spinodal decomposition of a binary mixture

In section 1.2.1 I already introduced the concept of a phase and that we can move from one phase to another phase. In fact, the last is what is known as a *phase transition*. From among all kinds of different phase transition we can have, we can distinguish two types:

- First phase order transition. In these transitions we will see a discontinuity in some macro thermodynamical quantities. During these transitions, the process evolves in a transition period where both phases coexist. In this period, temperature is constant, and a quantity of heat is absorbed or released. To this heat we refer as *latent heat* .
- Second order phase transition. In these transitions we will not see discontinuities and they will happen at a specific configuration of the thermodynamics quantities, which we refer as the *critical point*. At the critical point we will see an infinite correlation length, a power law decay of correlations and a susceptibility will diverge. No latent heat is seen.

As an example of the previous transitions, we can take a system where its state can be defined by the pressure, volume, and temperature. These kinds of systems are the ones we use to refer to common substances that change between gas-liquid-solid states. In these systems, it is typical to draw a scheme of temperature vs pressure, where different areas referring to different phases can be drawn, as seen in Fig. B.1A). The separation of these regions defines a curve, *the phase diagram curve*, in which first order transitions occur. In the case of the gas-liquid states, the diagram curve is not infinite, but ends at a point. This point is the *critical point*, which changes from first to second order transition, and after it we will never see a phase transition[5], since there is only a unique phase, the *supercritical fluid*.

[5]The necessity of an infinite curve can be seen as a matter of symmetries. When two phases lie in the same symmetry group, it is not necessary to have a phase transition to move from one to another phase, but when they are not, it is necessary. Because of atomic arrangement, solids have symmetries, which liquids and gases do not have. Therefore the liquid-solid curve is infinite, but not the liquid-gas. This is the *Landau symmetry principle*.

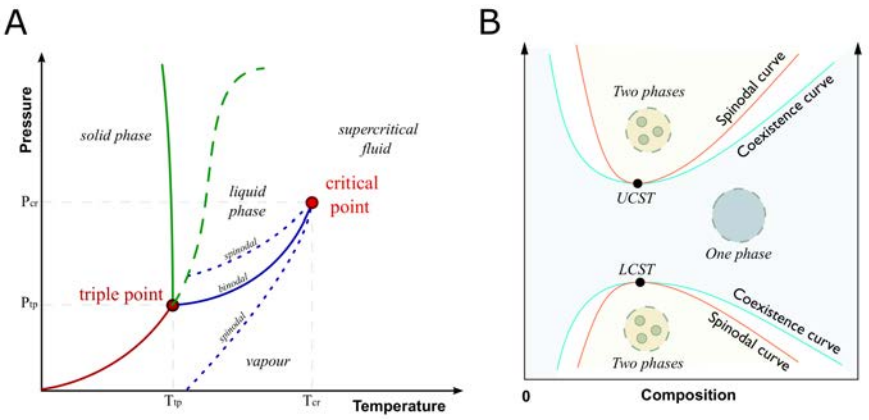


Figure B.1: Phase diagrams for phase transition. A) Typical diagram for a gas/liquid/solid system. Solid/gas transition will happen if we cross the red line. Solid/liquid transition will happen if we cross the green line. Liquid/gas transition will happen if we cross the blue line. In water, the less temperature the more pressure you need to make the transition (solid green line), but usually it is the opposite (dashed green line). The point where all lines crosses is the triple point. These transitions are first order, but there is a point, the critical point, which is a second order phase transition. Notice that these lines are the binodal. As an example, I draw the spinodal for the liquid/gas transition. A system in between the two spinodal may be in the opposite phase as expected. **B)** When dealing with binary mixtures, it is often presented a temperature-composition plot, where composition is the normalized percentage of one element from the two of the mixture. The same idea as in A) applies here. LCST is the lower critical solution temperature, while UCST is the upper critical solution temperature. *Image authorship. A) Matthieumarechal, Wikimedia Commons, used under CC BY-SA 3.0 license. Adapted. B) Locke9k, Martin Strand, Wikimedia Commons, used under CC BY-SA 3.0 license. Adapted.*

However, there a couple of things that usually are not explained about this diagram, or little is said when these plots are shown. The phase diagram curve is in fact what is known as the *binodal line*, where the transition between phases should start occurring. But to start, you need a nucleation point. If no point is found, the phase can still survive even if you would not expect it. For example, have you even seen any video where a bottle with liquid water inside is hit, and suddenly the water becomes ice? Well, this is the reason. Liquid water at atmospheric pressure and a bit under 0ºC should have been ice, because in the phase diagram is in the ice region, but it is still in the metastable area. When you hit it, suddenly the metastable state looks for the stable state, which is the one of the ice. To this liquid water is what you refer as *supercooled water*. If instead of being at the solid phase would have been in the gas coming from the liquid, you would have had *superheated water* . But these metastable regions end at some moment. As with the case of the binodal line, you could also draw a new line. We call to this line the *spinodal line*, and it is the line that separates the metastable region from the unstable one. Any fluctuation entering the spinodal will change the phase.

Another thing that is usually not commented is about the possibility of having critical points in between other phases. Maybe, because these kind of diagrams are usually shown just for a simple substance, but not for a mixture of different species (See Fig. B.1B)). For instance, we could have a critical point in between two liquid phases[254, 255]. For a simple substance it could be weird to think that we can have two different liquid phases, but even they can exist. In the case of having a mixture between two liquids that are miscible, one of the possibilities that may happen is that, as with the water can be converted from liquid to ice, both liquids unmix. For example, imagine we mix liquid A with liquid B. There will be a set of thermodynamical variables that will unmix the solution into a solution where both liquids are not mixed. When one works with these kinds of systems, it is usually painted a plot of molar fraction of one of the components vs temperature. In these plots, it is also possible to draw both the spinodal and the binodal curves. Where both curves coincide, we have a critical point, and since the nature of a critical point is different, no metastable gap is seen. Thus, around the critical point we can unmix the mixing without metastable regions. In any case, the area inside the binodal curve is called the miscibility gap, because we can have both phases unmixed. Technically, to represent these plots we would need to paint the Gibbs free energy vs the molar ratio. In this plot, for the miscibility gap we would see two minima, corresponding to both phases of liquid A and liquid B unmixed. But outside this region, the Gibbs free energy would look like a single minimum, and thus we can only have a unique phase, the mixed phase.

B.2.2 Landau free energy functional

Typically, the Gibbs and the Helmholtz are the two free energies defined in classical thermodynamics, but just a year after the starting of the Spanish Civil War, Lev Landau introduced his own free energy[256], which has units of energy as well, but lacks the complete significance of the previous free energies. Landau saw that around a critical point, the phenomena could be described by a power law for many thermodynamic variables, which give you some characteristics exponents depending on the system. Different systems can share same exponents, and hence, same physics, arising to the concept of *universality*. He took this idea in mind and decided to develop a function, which is expressed as a power law series also, but taking into account a parameter that is characteristic of the system, the *order parameter*, which we will refer for now as ϕ. The order parameter can be whatever, not necessary a scalar or a vector, but it must differentiate two phases. One, the *disordered phase*, will have order parameter zero, and the other one, the *ordered phase*, will have non-zero. With this parameter, Landau postulates that one can build the *Landau free energy* or Landau functional $\mathcal{F}$, where the state of the system is described by the global minimum of the functional with respect the order parameter. There are a few remarks $\mathcal{F}$ must obey:

- The symmetries of $\mathcal{F}$ must be consistent with the ones from the system.
- Around the critical point, defined by the critical temperature T_c, $\mathcal{F}$ is expandable in a power series. Moreover, in a uniform spatially system of volume V, it can be written also the *Landau free energy density* as:

$$f = \frac{\mathcal{F}}{V} = \sum_{n=0}^{\infty} a_n \phi^n, \tag{B.2.1}$$

  where a_n are different prefactors for each power of ϕ.
- If the system is inhomogeneous, e.g. being $\boldsymbol{r}$ the spatial position vector, $\phi = \phi(\boldsymbol{r})$, f is a local function. In such cases, f will only depend, as a maximum, on $\phi(\boldsymbol{r})$ and a finite number of derivatives.
- Around T_c, ϕ is zero in the disordered phase and non-zero in the ordered phase, but in the last, it is a very small number. Hence, depending on the dimensions of our system and the codimension of the critical point, f can be truncated. For a homogenous system, it is usually enough to develop the power series up to the forth power:

$$f = \frac{\mathcal{F}}{V} = \sum_{n=0}^{4} a_n \phi^n. \tag{B.2.2}$$

B.2.3 The Cahn-Hilliard equation

Using the ideas of the spinodal decomposition of a binary system and the Landau free energy, we can develop a simple model for the phase separation of a fluid composed by two miscible liquids, A and B, into two phases. The original development of this model[100], which is called as the Cahn-Hilliard equation, starts by using the mole fraction of one of the two liquids, but for our interpretation we will use the difference instead. If in our system we have n_A molecules of liquid A and n_B molecules of liquid B, then their molar fraction at position $\boldsymbol{r}$ and time t will be

$$c_A(\boldsymbol{r}, t) = \frac{n_A}{n_A + n_B},$$

and

$$c_B(\boldsymbol{r}, t) = \frac{n_B}{n_A + n_B},$$

respectively. Of course, $c_A(\boldsymbol{r}, t) + c_B(\boldsymbol{r}, t) = 1$. Here, we will consider that the important parameter is the difference between both concentrations $c(\boldsymbol{r}, t) = c_A(\boldsymbol{r}, t) - c_B(\boldsymbol{r}, t)$. Thus, for any $\boldsymbol{r}$ at any time t, if no A is present, $c = -1$. If no B is present, $c = 1$. If both are at the same quantity (mixed), $c = 0$. Therefore, c is a perfect candidate for being an order parameter as we will see. If mass is conserved, we can write:

$$\frac{\partial c(\boldsymbol{r}, t)}{\partial t} = -\nabla \cdot \boldsymbol{j}. \tag{B.2.3}$$

But, what is the flux $\boldsymbol{j}$ here? In section $A.1$ we can see that the flux is something the fluid is transporting. This is an advection term, but in the Cahn-Hilliard equation we do not consider, in principle, that there is an advection of c due to a liquid. This means that if we imagine c as solutes in a liquid, the solutes are moving inside it by themselves, diffusing, but not because the liquid is transporting the solutes itself. These are two different mechanisms. When liquids have regions of different solute concentrations, the solutes diffuse from regions of high concentration to lower concentration. One can say that phenomenological, the flux that transport these solutes are proportional to their concentration gradient[6]. Thus, $\boldsymbol{j} = -D\nabla c$, where D is a constant we call it as *diffusion constant*. But if there is a gradient of solutes is because there is a force acting over them. If the driving force that is acting is small enough, we can consider that this force is linearly related with the gradient. In our case, what leads to this change of concentration is the difference of chemical potential. As with the gravity force which can be derived from the gravitational potential, the chemical potential can also derive a force, which in our case is this driving force. Thus, we can write:

$$\frac{\partial c}{\partial t} = -\nabla \cdot (M(c)\nabla\mu)\,, \tag{B.2.4}$$

where $M(c)$ is a new parameter we know as the *mobility* and can depend on c. Hence, if we can obtain the chemical potential, we will be able to describe our system. The chemical potential can be derived from a free energy as already explained, and in the case of considering the critical point, we can develop the Landau free energy. Thus, we can develop a Landau free energy for this system using $c(\boldsymbol{r}, t)$ as the order parameter and we will assume that locally, the free energy will depend on both the direct local environment and the local concentration. Therefore, the free energy will be a series of power of ∇c and c. By introducing a few symmetries in the system we can simplify this series. For an isotropic medium or a cubic lattice, as the authors of the equation looked for, we need a rotational symmetry about a fourfold axis ($r_i \to r_j$) and a reflection symmetry ($r_i \to -r_i$), being $r_{i,j}$ the components of $\boldsymbol{r}$. After adding these symmetries, the free energy function has a shape of:

$$\begin{aligned}\mathcal{F} &= N_v \int_V dV \left(f_0(c) + \kappa_1 \nabla^2 c + \kappa_2 \left(\nabla c\right)^2 \right) \\ &= N_v \int_V dV \left(f_0(c) + \kappa \left(\nabla c\right)^2 \right).\end{aligned} \tag{B.2.5}$$

Here, N_v are the number of molecules per volume unit cell and $f_0(c)$ is the free energy density per molecule of a system with uniform composition c, which will depend on a power series of c. This power series should be reduced to the minimum, and by symmetry, we should ask for a solution of the shape[7] $f_0(c) = Ac^2 + Bc^4$. The fact of neglecting odds powers here is because we do not want that one phase has more or less energy than the other one. Both states are identically

[6] This is also known as the Fick's first law.

[7] The original work defined f_0 differently, but this alternative is simpler.

energetical and probable. On the other hand, the fourth term is the lowest that let to have two minima, regarding each to each phase, but only if $B > 0$ and $A < 0$. If f_0 expresses the homogenous part then, the derivative must be interpreted as the inhomogeneities of the system. This association is straightforward since gradients are only non-zero terms if the system is not homogenous, such as in the presence of interfaces. Hence, this term will include the role of surface tension into our free energy. Finally, the second step follows after applying the divergence theorem and choosing a boundary of integration in such a manner that $\nabla c \cdot \boldsymbol{n}$ is zero, being $\boldsymbol{n}$ a normal vector to the surface of element. This boundary selection can be done because we do not care about effects on the external surface. Therefore, Eq. $B.2.5$ is often written as:

$$\mathcal{F} = \int_V dV \left(\frac{A}{2} c^2 + \frac{B}{4} c^4 + \frac{\kappa}{2} |\nabla c|^2 \right), \tag{B.2.6}$$

where the parameters A, B, κ must be properly chosen as already discussed. The prefactors in the denominator in Eq. $B.2.6$ are chosen to simplify later derivatives of the free energy density. The gradient is often written in terms of the modulus, which is mathematically equal[8] to the previous term.

Once we obtained a free energy function, we can derive the chemical potential as if it were the Gibb's free energy from the free energy density f ($\mu = \partial_c f$) and hence, we developed the Cahn-Hilliard equation:

$$\frac{\partial c}{\partial t} = \nabla \cdot \left(M(c) \nabla \left(Ac + Bc^3 + \kappa \nabla^2 c \right) \right). \tag{B.2.7}$$

Finally, a last comment about the previous equation. For simplicity, it is often used $B = -A$. This is because the values of equilibrium for c are those that makes the derivative of f zero, or directly, the chemical potential equal 0. The solutions then read as $c^* = \pm\sqrt{-\frac{A}{B}}$, where c^* are the values of equilibrium. Thus, if $B = -A$, $c^* = \pm 1$.

Connecting the Cahn-Hilliard equation with Lattice-Boltzmann

In the development of the previous subsection I already discussed that c diffuses. But we know our fluid is not static, and that it transports this c. Thus, to connect both the fluid and c we just need to add one thing into Eq. $B.2.7$: an advection term. If $\boldsymbol{u}$ is the velocity of the fluid we can calculate from the Lattice-Boltzmann algorithm, then we will solve:

$$\frac{\partial c}{\partial t} = \nabla \cdot \left(M(c) \nabla \left(Ac + Bc^3 + \kappa \nabla^2 c \right) \right) - \nabla \cdot (\boldsymbol{u} c). \tag{B.2.8}$$

The discretization of this equation does not have anything special, as with the Lattice-Boltzmann, but just that the cells where it will be discretized will be the same to be used for the fluid.

[8] $(\nabla c)^2 = (\nabla c) \cdot (\nabla c) = \left(\frac{dc}{dx}\right)^2 + \left(\frac{dc}{dy}\right)^2 + \left(\frac{dc}{dz}\right)^2 = |\nabla c|^2$

B.3 | Microscopic thermodynamics: Kinetic theory

The concept of atoms and molecules has not ever been important along human history[257]. The firsts from which is typically said to think about indivisible particles are the followers of Leucippus and his pupil Democritus, from the old Greece about the 5th century BC, although the idea was also in other territories such as India before them. In fact, the word atom still comes from the Greek ἄτομον, which means that it cannot be cut. In the Medieval era, the influence of Aristotle, who was against this idea, hide the atomist current. In the renaissance, the idea came back, and intellectuals such as Galileo Galilei, René Descartes and Isaac Newton started supporting similar ideas. In the 19th century, the idea expanded until was definitely stablished in the first years of the 20th century, with the work of Einstein and Jean Perrin for the Brownian motion. Some years before their discoveries, while the atoms were supposed to exist, a new important theory was developed: The Kinetic theory, developed by Maxwell, but specially Ludwig Boltzmann. They explained thermodynamics using the idea of atoms with this theory. Their theory was further developed, and today we can simulate fluids using the kinetic theory. In this section I will show how it is possible.

B.3.1 The phase-space

When any object moves, it is easy to think about its trajectory over the $\mathbb{R}^n$ space where it is moving since we live in a $\mathbb{R}^3$ world, and to imagine n dimensions of the space it is very easy if $1 \leq n \leq 3$. By just fixing an axis of n coordinates in this $\mathbb{R}^n$ space, one can draw as many trajectories for as many objects as one wants. Although this naive idea for drawing trajectories can work, sometimes these coordinates are not the best to deal with a problem and it is better to use other alternatives. For example, the Hamiltonian mechanics shows us the possibilities of the phase-space, which is what Boltzmann used during its reasoning for obtaining the Boltzmann Equation. Under Hamiltonian mechanics, the particles of the system will follow the Hamiltonian, which for a system composed of N particles each of them with mass m under the action of an external force $\boldsymbol{F} = -\nabla V$ and a two-body interaction between particles $U(\boldsymbol{r}_i - \boldsymbol{r}_j)$ will be written as:

$$H = \frac{1}{2m}\sum_{i=1}^{N} \boldsymbol{p}_i^2 + \sum_{i=1}^{N} V(\boldsymbol{r}_i) + \sum_{i<j} U(\boldsymbol{r}_i - \boldsymbol{r}_j), \tag{B.3.1}$$

where $\boldsymbol{r}_i$ is the position of the i-th particle and $\boldsymbol{p}_i$ is the momentum of the i-th particle. We assume U is a short-range potential, which for a distance between particles much higher than the atomic distance scale, the potential U is zero. Since this equation only depends on $\boldsymbol{r}$ and $\boldsymbol{p}$, and explicitly could also depend on t as well, it is common to work in a space composed of the $\boldsymbol{r}$ and $\boldsymbol{p}$ coordinates, which is what we call as the *phase-space*. Thus, the phase-space is a space of $2nN$ dimensions, where the 2 is because for each particle we add the axis related with its momentum $\boldsymbol{p}$ and its position $\boldsymbol{r}$. Consequently, the typical idea of a gas composed by N molecules in a $\mathbb{R}^3$ world will lead to a space of $6N$ dimensions. To imagine such a huge number of dimensions can be of course difficult, and indeed, we cannot draw it, but one can get used to it because its advantage to deal with big number of particles. In this phase-space, it is very important to understand that a point and a trajectory is not anymore, the position of a particle and its trajectory. Instead, a point in this phase-space is a microstate of the system. To explain what a microstate is, it is easier to think from the $\mathbb{R}^3$ world perspective. For any given time, any particle of our gas of particles will have a position and a momentum. If we take a photo of the system at a given time, all the positions and momentums pairs of the whole system will be a *microstate* of the system. Thus, all the information related with momenta and positions of all particles will collapse as a single point in this phase-space as shown in Fig. $B.2$. If the particles are distinguishable, each point will be a microstate. If particles are not distinguishable, then there will be N! points that will refer to the same microstate since we cannot differentiate from the exchange of particles.

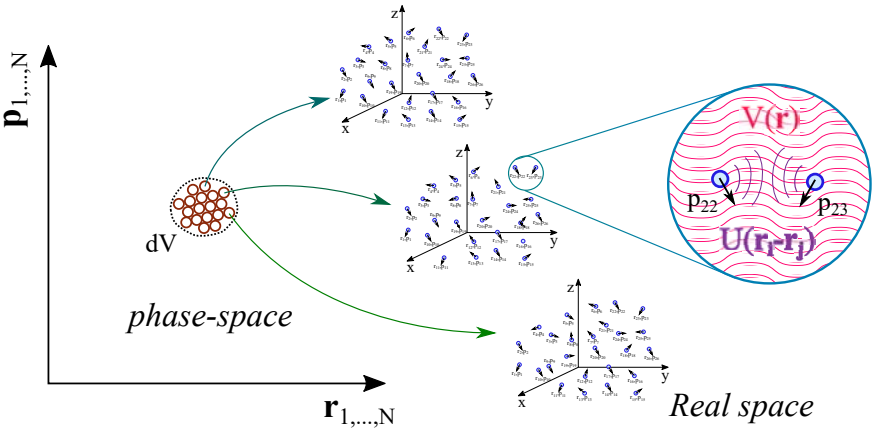


Figure B.2: In the phase-space description, a point is a microstate of the whole N particle system, which is different from the point in the Real space, where a point is the position of a single particle. Tiny changes in the Real space will lead to close microstates with same macrostate values. Thus, the phase-space can be treated as a continuum with a density of microstates, which can be treat as mathematical points in this new space.

At a macroscopic perspective, each microstate will define macroscopic values such as the temperature. But of course, changes in position and momentum of particles with such number of particles can lead at the end with a microstate for which its macroscopic values cannot be distinguished from the one coming from a different microstate. These microstates in fact, will be points that are close in the phase-space, and thus, for a given macroscopic value, we can have plenty of points close together in this phase-space creating a cloud of points. Therefore, a system of N particles such a gas should be thought as a cloud of points instead of just one point, because it is not possible to know all the initial conditions of every single particle, but we know the averaged properties. When the system evolves, the system will move towards a new microstate and thus, by connecting different microstates with a line in this phase-space you can draw the evolution of the system. This is what a trajectory means in this phase-space.

B.3.2 The Boltzmann equation

The Boltzmann equation was developed by Ludwig Boltzmann in 1872[258] to deal with systems composed by millions of particles, such as gases, under non-equilibrium processes that will move towards equilibrium. His idea is based on trying to understand the system by looking at a special set of probabilities that can occur in the system. In his mind, the principal problem of these systems was to understand which is the probability of finding a particle with momentum $\boldsymbol{p}$ around position $\boldsymbol{r}$ at time t. Thus, instead of working on the real space, he worked on the phase-space.

To start with the Boltzmann equation explanation, we need first to discretize our space in infinitesimal cubes of 6N dimensions and size $\prod_{i=1}^{N} d\boldsymbol{p_i} d\boldsymbol{r_i}$. Now, once the space is discretized, in principle we can always build a probability density function $f(\boldsymbol{r_1}, .., \boldsymbol{r_N}, \boldsymbol{p_1}, .., \boldsymbol{p_N}, t)$ which is the probability of finding the system around the point $(\boldsymbol{r}, \boldsymbol{p})$ for a given time. Of course, if you integrate this probability density function over all the space it must be 1:

$$\int f(\boldsymbol{r_1}, .., \boldsymbol{r_N}, \boldsymbol{p_1}, .., \boldsymbol{p_N}, t) \prod_{i=1}^{N} d\boldsymbol{p_i} d\boldsymbol{r_i} = 1. \tag{B.3.2}$$

But we will not work directly with $f(\boldsymbol{r_1}, .., \boldsymbol{r_N}, \boldsymbol{p_1}, .., \boldsymbol{p_N}, t)$ since working with this probability density function means to work with a 6N dimensional space system. To avoid such number of dimensions, we can work with less dimensions by introducing the n-th particle distribution function:

$$f_n(\boldsymbol{r}_1, .., \boldsymbol{r}_n, \boldsymbol{p}_1, .., \boldsymbol{p}_n, t) = \frac{N!}{(N-n)!} \int \prod_{i=n+1}^{N} d\boldsymbol{p_i} d\boldsymbol{r_i} f(\boldsymbol{r_1}, .., \boldsymbol{r_N}, \boldsymbol{p_1}, .., \boldsymbol{p_N}, t), \tag{B.3.3}$$

where we have supposed that in this system, particles are indistinguishable and therefore, we cannot differentiate one particle from another one. That is the reason why we have the factorial prefactor, since in fact we have N particles and there is a total of different $\frac{N!}{(N-n)!}$ different combinations we could have. Associated with this n-th distribution function, we can also define the n-th particle Hamiltonian as:

$$H_n = \sum_{i=1}^{n} \left(\frac{\boldsymbol{p}_i^2}{2m} + V(\boldsymbol{r}_i) \right) + \sum_{i<j\leq n} U(\boldsymbol{r}_i - \boldsymbol{r}_j), \tag{B.3.4}$$

which is the Hamiltonian given by Eq. $B.3.1$ associated with up to n particles. Once we have defined these equations, we can proceed in the search for the Boltzmann equation. What Boltzmann saw is that in fact, to deal with a gas of N particle system, even if it is out-of-equilibrium, it is enough to work with the 1-particle distribution function:

$$f_1(\boldsymbol{r}, \boldsymbol{p}, t) \equiv f(\boldsymbol{r_1}, \boldsymbol{p_1}, t) = N \int f(\boldsymbol{r_1}, .., \boldsymbol{r_N}, \boldsymbol{p}, .., \boldsymbol{p_N}, t) \prod_{i=2}^{N} d\boldsymbol{p_i} d\boldsymbol{r_i}, \tag{B.3.5}$$

where of course, if we integrate $f_1(\boldsymbol{r}, \boldsymbol{p}, t)$ over all the space we should have N, since we have N particles. The reason to work with just the one particle probability density function is because it is the only probability density function we will need to describe the system as a consequence of a strong assumption we will do afterwards, the molecular chaos hypothesis. To see this fact, we can see how $f_1(\boldsymbol{r}, \boldsymbol{p}, t)$ evolves with time:

$$\frac{\partial f_1}{\partial t} = N \int \prod_{i=2}^{N} d\boldsymbol{p_i} d\boldsymbol{r_i} \frac{\partial f}{\partial t} = N \int \prod_{i=2}^{N} d\boldsymbol{p_i} d\boldsymbol{r_i} \{H, f\}, \tag{B.3.6}$$

where here, we have introduced the Hamiltonian given in Eq. $B.3.1$, the Poisson bracket $\{A, B\} \equiv \frac{\partial A}{\partial \boldsymbol{r}_i} \cdot \frac{\partial B}{\partial \boldsymbol{p}_i} - \frac{\partial A}{\partial \boldsymbol{p}_i} \cdot \frac{\partial B}{\partial \boldsymbol{r}_i}$ and we have used the Liouville theorem[9]. If we insert the complete Hamiltonian (Eq. B.3.1) we will find:

$$\frac{\partial f_1}{\partial t} = N \int \prod_{i=2}^{N} d\boldsymbol{p_i} d\boldsymbol{r_i} \left[-\sum_{j=1}^{N} \frac{\boldsymbol{p}_j}{m} \cdot \frac{\partial f}{\partial \boldsymbol{r}_j} + \sum_{j=1}^{N} \frac{\partial V}{\partial \boldsymbol{r}_j} \cdot \frac{\partial f}{\partial \boldsymbol{p}_j} + \sum_{j=1}^{N} \sum_{k<l} \frac{\partial U(\boldsymbol{r}_k - \boldsymbol{r}_l)}{\partial \boldsymbol{r}_j} \cdot \frac{\partial f}{\partial \boldsymbol{p}_j} \right]. \tag{B.3.7}$$

[9] As I previously explained, a gas of N particles will be seen as a cloud of microstates given the uncertainty of the initial conditions and given that many will give the same macrostate. With this perspective, this cloud of packed dots could be treated as a continuous and therefore, treat it as if it were a fluid following Eq. $A.1.3$. Consequently, it can be seen that the "fluid" is incompressible and hence, under Hamiltonian evolution, if you follow some region of the phase space, it will have constant volume $\left(\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial \boldsymbol{r}_i} \cdot \dot{\boldsymbol{r}}_i + \frac{\partial f}{\partial \boldsymbol{p}_i} \cdot \dot{\boldsymbol{p}}_i = 0 \right)$. This is known as the *Liouville theorem*. Furthermore, since the density is conserved, from this theorem we can see two consequences. The first is that because Noether's theorem, there is a symmetry under time translations, being the Hamiltonian of the system the generator of the symmetry. The second is that for a f_N under equilibrium, the $\partial_t f = 0$, and hence $\{H, f\} = 0$, which means that $f(H)$, as in the case of equilibrium where $f \approx e^{-\beta H}$.

We can remove the sums for $j = 2, ..., N$ by integrating per parts. Some elements will depend on the moment, but the derivatives will depend on the position, or vice versa, and hence will be zero. Other terms will need to evaluate f at infinity, and because is a probability distribution, to be normalizable it must be zero at the infinity. Hence, we have:

$$\frac{\partial f_1}{\partial t} = N \int \prod_{i=2}^{N} d\boldsymbol{p_i} d\boldsymbol{r_i} \left[-\frac{\boldsymbol{p}}{m} \cdot \frac{\partial f}{\partial \boldsymbol{r}} + \frac{\partial V}{\partial \boldsymbol{r}} \cdot \frac{\partial f}{\partial \boldsymbol{p}} + \sum_{k=2} \frac{\partial U(\boldsymbol{r} - \boldsymbol{r}_k)}{\partial \boldsymbol{r}} \cdot \frac{\partial f}{\partial \boldsymbol{p}} \right], \tag{B.3.8}$$

where we adopted $\boldsymbol{r}_1 \equiv \boldsymbol{r}$ and $\boldsymbol{p}_1 \equiv \boldsymbol{p}$. If we use Eq. $B.3.4$ for one particle, we can define the one-particle Hamiltonian as:

$$H_1 = \frac{p^2}{2m} + V(\boldsymbol{r}), \tag{B.3.9}$$

which simplifies more Eq. B.3.8:

$$\frac{\partial f_1}{\partial t} = \{H_1, f_1\} + \left(\frac{\partial f_1}{\partial_t} \right)_{coll}. \tag{B.3.10}$$

In the last equation, we call to the first term the *streaming* term and, to the last the *collision* term. Since the one-particle Hamiltonian does not include the 2-body interaction potential U, we can clearly see that the streaming term refers to how particles move in the absence of collisions between particles. Opposite to the streaming, the collision term is the term which includes this potential and thus, gives information about these collisions. Because all particles are indistinguishable, the collision term can be written as:

$$\begin{aligned} \left(\frac{\partial f_1}{\partial_t} \right)_{coll} &= N(N-1) \int d^3 r_2 d^3 p_2 \frac{\partial U(\boldsymbol{r} - \boldsymbol{r_2})}{\partial \boldsymbol{r}} \cdot \\ &\frac{\partial}{\partial \boldsymbol{p}} \int \prod_{i=3}^{N} \Delta \boldsymbol{p_i} \Delta \boldsymbol{r_i} f(\boldsymbol{r_1}, .., \boldsymbol{r_N}, \boldsymbol{p_1}, .., \boldsymbol{p_N}, t) = \\ &N(N-1) \int d^3 r_2 d^3 p_2 \frac{\partial U(\boldsymbol{r} - \boldsymbol{r_2})}{\partial \boldsymbol{r}} \frac{\partial f_2}{\partial \boldsymbol{p}}, \end{aligned} \tag{B.3.11}$$

where f_2 is the two-particle distribution function following Eq. $B.3.3$. As we can see, we were looking for how 1-particle distribution evolves, and we reached to an equation where this evolution depends on the 2-particles distribution. In fact, one can always repeat the previous process reaching to an equation where:

$$\frac{\partial f_n}{\partial t} = \{H_n, f_n\} + \sum_{i=1}^{n} \int d^3 r_{n+1} d^3 p_{n+1} \frac{\partial U(\boldsymbol{r_i} - \boldsymbol{r_{n+1}})}{\partial \boldsymbol{r}_i} \frac{\partial f_{n+1}}{\partial \boldsymbol{p}_i}. \tag{B.3.12}$$

Eq. $B.3.12$ is also known as the BBGKY hierarchy[10]. As we can see, this hierarchy is a sum of n elements. This means that, if we find that from one of these elements all are negligible, then we can truncate the series at that element. Boltzmann saw that the simplest truncation, where only f_1 is important, is enough. To affirm this, we need first, to admit a few assumptions before doing the reasoning:

1. Collisions are punctual in time. The total time that a collision lasts (τ_{coll}) is much lower than the time needed to occur a collision (τ_{st}), which is also known as the relaxation time or scattering time. Thus, $\tau_{st} \gg \tau_{coll}$.

2. Collisions are local in space. Both particles are at the same point when they collide.

3. If particles are affected by an external potential $V(\boldsymbol{r})$ as the global Hamiltonian in Eq. $B.3.1$ reflects, this potential only changes in a macroscopic length scale.

Under these assumptions, we can treat the problem of the collision as the typical problem that one would solve in a classical mechanics lecture: we must apply conservation of momenta and energy:

$$\begin{aligned} \boldsymbol{p_1} + \boldsymbol{p_2} &= \boldsymbol{p_1'} + \boldsymbol{p_2'} \\ p_1^2 + p_2^2 &= p_1'^2 + p_2'^2 \end{aligned} \tag{B.3.13}$$

where $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ are the momenta of both particles before the collision and $\boldsymbol{p_1'}$ and $\boldsymbol{p_2'}$ is their respective momenta after the collision. From the previous equation we can see that there is going to be a probability for the particles to go from the state $(\boldsymbol{p_1}, \boldsymbol{p_2})$ towards the state $(\boldsymbol{p_1'}, \boldsymbol{p_2'})$, or simply $(\boldsymbol{p_1}, \boldsymbol{p_2}|\boldsymbol{p_1'}, \boldsymbol{p_2'})$. But at the same time, other particles can also evolve following $(\boldsymbol{p_1'}, \boldsymbol{p_2'}|\boldsymbol{p_1}, \boldsymbol{p_2})$. For both scenarios we can define a scattering function ω. We can rewrite the collision term as a function of this scattering function[259]:

$$\begin{aligned} \left(\frac{\partial f_1}{\partial t}\right)_{coll} = \int d^3p_2 d^3p_1' d^3p_2' \, [&\omega(\boldsymbol{p_1}, \boldsymbol{p_2}|\boldsymbol{p_1'}, \boldsymbol{p_2'}) f_2(\boldsymbol{r}, \boldsymbol{r}, \boldsymbol{p_1'}, \boldsymbol{p_2'}, t) \\ &- \omega(\boldsymbol{p_1'}, \boldsymbol{p_2'}|\boldsymbol{p_1}, \boldsymbol{p_2}) f_2(\boldsymbol{r}, \boldsymbol{r}, \boldsymbol{p_1}, \boldsymbol{p_2}, t)] = \\ \int d^3p_2 d^3p_1' d^3p_2' \, &\omega(\boldsymbol{p_1}, \boldsymbol{p_2}|\boldsymbol{p_1'}, \boldsymbol{p_2'}) \cdot \\ &[f_2(\boldsymbol{r}, \boldsymbol{r}, \boldsymbol{p_1'}, \boldsymbol{p_2'}, t) - f_2(\boldsymbol{r}, \boldsymbol{r}, \boldsymbol{p_1}, \boldsymbol{p_2}, t)] \end{aligned} \tag{B.3.14}$$

where we have considered that the scattering is parity invariant and time reversal invariant:

$$\begin{aligned} \omega(\boldsymbol{p_1}, \boldsymbol{p_2}|\boldsymbol{p_1'}, \boldsymbol{p_2'}) &= \omega(-\boldsymbol{p_1}, -\boldsymbol{p_2}| - \boldsymbol{p_1'}, -\boldsymbol{p_2'}) \\ \omega(\boldsymbol{p_1}, \boldsymbol{p_2}|\boldsymbol{p_1'}, \boldsymbol{p_2'}) &= \omega(-\boldsymbol{p_1'}, -\boldsymbol{p_2'}| - \boldsymbol{p_1}, -\boldsymbol{p_2}) \end{aligned} \tag{B.3.15}$$

[10]The name of the hierarchy comes from the initials of several people who worked dealing this n-particle distributions and kinetic theory. In particular: Bogoliubov, Born, Green, Kirkwood and Yvon.

Now, it is the moment to break the series and introduce the concept Boltzmann thought: we will assume the *molecular chaos hypothesis* or *Stosszahlansatz*. When two particles, A and B, collide, their momenta get correlated. Therefore, the next time they will collide both together, their momenta should be also correlated before the new collision. But the time between a new collision, the relaxation time τ_{st}, is large and whenever a new collision occurs, this new collision probably will be between rather A or B with a different particle C, so momenta of these two particles do not have to be correlated in principle. In this sense, we can say that before the collision, the momenta are uncorrelated. This is what is known as the molecular chaos hypothesis and mathematically means that:

$$f_2(\boldsymbol{r}, \boldsymbol{r}, \boldsymbol{p_1}, \boldsymbol{p_2}) = f_1(\boldsymbol{r}, \boldsymbol{p_1}) f_1(\boldsymbol{r}, \boldsymbol{p_2}) \tag{B.3.16}$$

Therefore, the collision term has the shape:

$$\left(\frac{\partial f_1}{\partial t}\right)_{coll} = \int d^3p_2 d^3p_1' d^3p_2' \, \omega(\boldsymbol{p_1}, \boldsymbol{p_2} | \boldsymbol{p_1'}, \boldsymbol{p_2'}) \cdot \\ [f_1(\boldsymbol{r}, \boldsymbol{p_1'}) f_1(\boldsymbol{r}, \boldsymbol{p_2'}) - f_1(\boldsymbol{r}, \boldsymbol{p_1}) f_1(\boldsymbol{r}, \boldsymbol{p_2})] \tag{B.3.17}$$

and with this new result, we can close all our problem by just using f_1 instead of the f_N. The Boltzmann equation is nothing more than Eq. $B.3.10$ given the collision term expressed as in Eq. $B.3.17$. Finally, it is worth it to say that by accepting the molecular chaos hypothesis, we have introduced an arrow of time. This assumption breaks the reversibility one could think when dealing with particles at molecular level and brings the irreversibility seen in the macroscopic view (e.g. if you have a closed box with a lot of molecules in one corner, and you release them, they will always fill uniformly the box, but they will never go back all of them to the point from where you left them).

B.3.3 The Lattice-Boltzmann equation

We will start the development of the Lattice-Boltzmann equation by expanding the terms in the Boltzmann equation:

$$\frac{\partial f(\boldsymbol{r}, \boldsymbol{p}, t)}{\partial t} + \frac{\boldsymbol{p}}{m} \nabla_r f(\boldsymbol{r}, \boldsymbol{p}, t) + F \nabla_p f(\boldsymbol{r}, \boldsymbol{p}, t) = \left(\frac{\partial f}{\partial t}\right)_{coll} \tag{B.3.18}$$

where f is the one particle distribution or f_1, which is the probability of finding a particle of mass m around position $\boldsymbol{r}$ with momentum $\boldsymbol{p}$ at time t, F is any force applied to this particle and ∇_r and ∇_p are the ∇ operator applied only to $\boldsymbol{r}$ and $\boldsymbol{p}$ respectively. Terms on the left side of the equation are the so-called stream because they represent how one particle moves without interacting with other particles, and the term on the right side is the collision operator because represents how a particle interact with other particles in the system.

Before starting with the derivation, we will simplify the collision part since in the form we obtained in Eq. $B.3.18$ it is complex to discretize it. To simplify it, we can assume that with each collision, the system tends towards the equilibrium as thermodynamics shows us. Therefore, we could think that with each collision we are an amount proportional to the f^{eq} closer, which is the f at equilibrium. In average, this process will last for a characteristic time giving us a relaxation time towards the equilibrium τ. In this sense, the collision term can be replaced, and the Boltzmann equation has the shape of:

$$\frac{\partial f(\boldsymbol{r},\boldsymbol{p},t)}{\partial t}+\frac{\boldsymbol{p}}{m}\nabla_r f(\boldsymbol{r},\boldsymbol{p},t)+F\nabla_p f(\boldsymbol{r},\boldsymbol{p},t)=\frac{f^{eq}(\boldsymbol{r},\boldsymbol{p},t)-f(\boldsymbol{r},\boldsymbol{p},t)}{\tau} \tag{B.3.19}$$

This approximation is known as the *BGK model*[260]. There are other simplifications of this collision term that take more than one relaxation time[97], but this is the simplest simplification and I did not work with those during my thesis. Now that we have simplified the equation, we can also dimensionless as we did with Navier-Stokes at section 1.3.2. The set of dimensionless parameters are the same as with Navier-Stokes, but writing the force in capital letters, the p now means the momentum and introducing a few new changes:

- A reduced form of the relaxation time τ, which is $\tau^*=\frac{\tau}{\tau_{st}}$, where τ_{st} is the time between collisions.
- A reduced form of f_i, which is $f_i^*=\frac{f_i}{n_r}$, where n_r is a reference density.
- A dimensionless parameter ϵ that arises from the dimensionless form as the Re was in the case of Navier-Stokes. This parameter is $\epsilon=\tau_{st}U/L$ and can be interpreted as the dimensionless Knudsen number, which is the ratio of mean free path to the characteristic length or the ratio of collision time to flow time.

With these modifications, we have:

$$\frac{\partial f^*}{\partial t}+\boldsymbol{v^*}\nabla_r^* f^*+\frac{F_0 L}{p_0 U}\frac{F^*}{m^*}\nabla_{\boldsymbol{v^*}} f^*=\frac{f^{eq*}-f^*}{\epsilon\tau^*} \tag{B.3.20}$$

Now we may start the discretization of this equation to be computed. Notice that, opposite to the common discretizations we usually do when dealing with equations where only space and time are discretized, here the $\boldsymbol{p}$ is part of the axis. This means that $\boldsymbol{p}$, or the velocity $\boldsymbol{v}$, must also be discretized by itself. That is why, when one discretizes this equation, the first that is shown is that we start discretizing the velocities in a set of i velocities. Therefore, we could in principle associate each $\boldsymbol{v}_i$ to a different distribution function f so now we will have a set of i distribution functions $f_i^*(\boldsymbol{r^*},t^*)$, each of them verifying equation $B.3.20$. Usually, when this discretization is performed, the force term is neglected. If later one needs to add any force, some terms can be introduced in an intelligent manner. With all these changes, we can discretize the equation by applying an implicit upwind scheme[261], which uses a backward difference in space and a backward

implicit Euler method in time. Consequently, the derivative over time is calculated at position $\boldsymbol{r}^*$, but the derivative over $\boldsymbol{r}^*$ is calculated at time $t^* + \Delta t^*$. Different schemes could be used, but by using this scheme, the equation can be simplified if we choose a $\frac{\Delta \boldsymbol{r}^*}{\Delta t^*} = \boldsymbol{v}_i^*$. Moreover, if $\Delta t^* = \tau_{st}^*$ the Knudsen number disappears, and we get:

$$f_i^*(\boldsymbol{r}^* + \boldsymbol{v}_i^* \Delta t^*, t^* + \Delta t^*) - f_i^*(\boldsymbol{r}^*, t^*) = -\frac{f_i^*(\boldsymbol{r}^*, t^*) - f_i^{eq*}(\boldsymbol{r}^*, t^*)}{\tau^*} \quad \text{(B.3.21)}$$

Thus, for each velocity we will do a local collision event and only one stream operation. Eq. $B.3.21$ is what is known as the Lattice-Boltzmann equation.

About the speed discretization in the Lattice-Boltzmann equation

In the expansion of the Lattice-Boltzmann I did there was something shady: the velocity discretization. Because if the velocity can span over orders of magnitude, how many equations of the style $B.3.21$ will we have to solve?

In fact, the key point here is that the number of velocities we discretize this space is limited. And by limited I refer to less than 30, since are enough to recover the hydrodynamics of Navier-stokes. To understand this, we need to know that the Lattice-Boltzmann equation is seen as an upgrade over the Lattice Gas Cellular Automata (LGCA) model. The LGCA model is a model that evolve from an older model, the Cellular Automata (CA), and that got famous because although its simplicity, it could recover the dynamics of the Navier-Stokes equation[262]. In this model, one looks for the position of N indistinguishable particles of mass $m = 1$ that can only live in the nodes of a lattice. Although these particles can only live in the nodes, at each time step they are moving along the i edges that connect these nodes. But of course, since they can only live in the nodes, each time they move they do with a velocity within a set of available velocities. If one chooses $\Delta t = 1$ and sets that the distance between all nodes is $|\Delta \boldsymbol{r}_i| = 1$, then the velocities must be $\boldsymbol{c}_i = \Delta \boldsymbol{r}_i / \Delta t$. Therefore, depending how you build the lattice given these constraints you will be able to have more or less velocities, but they will always be a very reduced number of velocities because of geometry construction. When particles in this model move to a new node (see Fig. $B.3$A), you say you are doing a *streaming* operation. But when particles arrive to the new node, another operation is done: the *collision* operation, because only one particle can sit in each node, introducing then a kind of exclusion principle (see Fig. $B.3$A). This principle reminds to the idea of Pauli principle, and hence is not weird to see that the equilibrium distribution you recover in the model is of Fermi-Dirac type. During the collision step, one introduces the conservation of mass and momentum, which are the key ingredients in the development of Navier-Stokes equation and, surprisingly, if the lattice has a proper symmetry[11] the dynamics of Navier-Stokes

[11] To recover Navier-Stokes dynamics you need rotational invariance. For example, in 2D this implies that a square lattice, such the HPP LGCA model, is not valid. A hexagonal lattice, such the FHP LGCA model, is enough to recover this symmetry.

are recovered. The only you need to have the macroscopic values is to average over many neighbors the amount of particles with different velocities $\boldsymbol{c}_i$ (See Fig. B.3B).

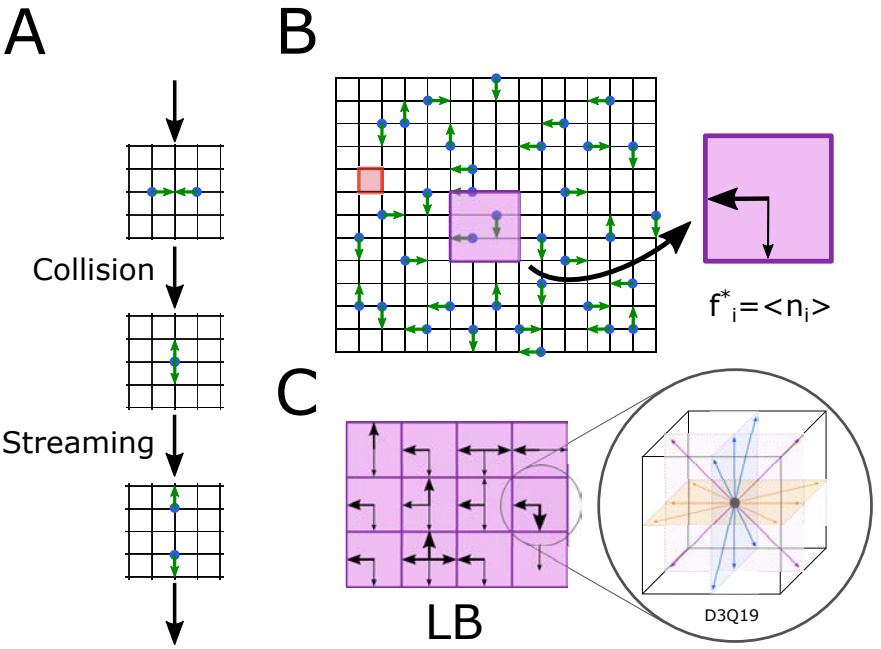


Figure B.3: Basic scheme to understand the Lattice Boltzmann (LB) model. A) The model is based on the two-operation step: Collision-Streaming. During collision, momentum and mass are conserved. After the collision, particles stream towards different nodes using a discrete set of velocities. **B)** Landscape of these particles moving along the lattice (in red the unit cell). Macroscopic values are captured by averaging the occupation number in each direction over the neighbours (purple square). In LB these averages are directly the particles, which are equal to our f_i^*. **C)** The actual lattice is something closer to this lattice, where we use floats instead of boolean properties, and we will use only the f_i from each unit cell to obtain the macroscopic values of that cell. Depending on how the lattice is constructed, distributions will have different number of directions. The overall picture here is in 2D, but each big square can have more directions available as shown in the inset which is commonly used in 3D scenarios. The available directions are written as $DnQm$, where n are the number of dimensions and m the number of available speed modes. *Image authorship. Own image.*

In the LGCA model, the only you need to save is the state of occupation of each node. Since there can come up to i particles to each node per time step, you only need a binary number of lengths i to know if a particle from each direction arrived or not. Thus, for each node you have a value for $n_k \equiv |n_1, n_2, ..., n_j, ..., n_i >$, where n_k is the state of occupation of node k and n_j is a boolean that indicates if

there is a particle or not in the direction j. Essentially, the rule that governs this model is an equation such Eq. $B.3.21$ but with a different collision term. Hence, it is understandable why Lattice-Boltzmann is seen as an upgrade of the LGCA and now, which speeds we must take. For instance, the 3D model of the lattice I will use limits the number of velocities to 19 (See Fig. $B.3$C)[12].

The main difference between LGCA and the firsts models of Lattice-Boltzmann is that in LGCA, to have the macroscopic values at some cell, you need to average the occupation state over many neighbours. Instead, in the Lattice-Boltzmann you consider that your particles are directly these averages and consequently, the structure of the model is not based in boolean values but in float values. Now, the floats represent the distribution f, and macroscopic values are recuperated by operating in each cell without averaging over neighbour cells. This fact solved one of the main troubles in LGCA: the reduction of noise.

From Lattice-Boltzmann equation to Navier-Stokes

Up to now, I have showed the Lattice-Boltzmann equation but, I have not linked yet with Navier-Stokes equation mathematically. To show this transition, one needs to bring the Chapman-Enskog (CE) expansion[263]. This expansion is basically applying the mathematical perturbation theory upto order two to the Lattice-Boltzmann equation. If we expand the elements in equation B.3.21:

$$\begin{aligned} f_i(\boldsymbol{r},t) &= \sum_{n=0}^{\infty} \epsilon^n f_i^{(n)}(\boldsymbol{r},t), \\ \partial t &= \sum_{n=0}^{\infty} \epsilon^n \partial_{t_n}, \\ f_i(\boldsymbol{r}+\boldsymbol{v}_i\Delta t, t+\Delta t) &= \sum_{n=0}^{\infty} \frac{\epsilon^n}{m!} D_t^m f_i(\boldsymbol{r},t), \end{aligned} \tag{B.3.22}$$

where we have applied the perturbative method on f_i and its derivatives using a very small parameter ϵ, we have used Taylor series to expand the forward step in the streaming operator with $D_t^m = \partial_t + \boldsymbol{c}_i \cdot \nabla$ being the material derivative and assuming $\Delta t \sim \epsilon$, and we have omitted the * to simplify the notation. We will further remove $(\boldsymbol{r}, t)$ in the following lines to simplify more our notation since all the f work with the same set of parameters now. After inserting these expansions and cutting to ϵ order two, we obtain:

[12] In this field it is common to describe the lattice as DnQc, where n refers to the dimension of the system and c to the total amount of different velocities you have for your particles. In my case I use D3Q19.

$$
\begin{aligned}
\epsilon^0: &\quad f_i^{(0)} = f_i^{eq}, \\
\epsilon^1: &\quad D_{t_0} f_i^{(0)} = (\partial_{t_0} + \boldsymbol{c}_i \cdot \nabla) f_i^{(0)} = -\frac{\Delta t}{\tau} f_i^{(1)}, \\
\epsilon^2: &\quad \partial_{t_1} f_i^{(0)} + (\partial_{t_0} + \boldsymbol{c}_i \cdot \nabla) f_i^{(1)} + \frac{1}{2}(\partial_{t_0} + \boldsymbol{c}_i \cdot \nabla)^2 f_i^{(0)} + \frac{\Delta t}{\tau} f_i^{(2)} = 0.
\end{aligned}
\tag{B.3.23}
$$

Combining orders first and zero, reordering the Del operator, and multiplying by the speeds, we obtain the following two equations:

$$
\begin{aligned}
\partial_{t_0} f_i^{(eq)} + \nabla \cdot \boldsymbol{c}_i f_i^{(eq)} &= -\frac{\Delta t}{\tau} f_i^{(1)}, \\
\partial_{t_0} \boldsymbol{c}_{i,m} f_i^{(eq)} + \nabla \cdot \boldsymbol{c}_{i,m} \boldsymbol{c}_{i,n} f_i^{(eq)} &= -\frac{\Delta t}{\tau} \boldsymbol{c}_{i,m} f_i^{(1)}.
\end{aligned}
\tag{B.3.24}
$$

Now, if we sum over the i directions, we see that:

$$
\begin{aligned}
\partial_{t_0} \sum_i f_i^{(eq)} + \nabla \cdot \sum_i \boldsymbol{c}_i f_i^{(eq)} &= -\frac{\Delta t}{\tau} \sum_i f_i^{(1)}, \\
\partial_{t_0} \sum_i \boldsymbol{c}_{i,m} f_i^{(eq)} + \nabla \cdot \sum_i \boldsymbol{c}_{i,m} \boldsymbol{c}_{i,n} f_i^{(eq)} &= -\frac{\Delta t}{\tau} \sum_i \boldsymbol{c}_{i,m} f_i^{(1)}.
\end{aligned}
\tag{B.3.25}
$$

If we remember the concept of f, we said that it was the probability of finding a particle with such position and velocity at time t. If we sum over all the directions of a node, then what we find is the total mass per node, or the total density ρ. If the sum of f_i is the density, the sum of $f_i c_i$ will be the momentum $\rho \boldsymbol{u}$, being $\boldsymbol{u}$ the velocity of the fluid at that node. As a constraint, we will impose that all the non-equilibrium term of f_i, this is all $f_i^{(n)}$ but $f_i^{(0)}$, will be zero. Therefore, we have:

$$
\begin{aligned}
\partial_{t_0} \rho + \nabla \cdot \rho \boldsymbol{u} &= 0, \\
\partial_{t_0} (\rho \boldsymbol{u}) + \nabla \cdot \Pi_{m,n}^{(0)} &= 0,
\end{aligned}
\tag{B.3.26}
$$

where we expressed the second term as the tensor $\Pi_{m,n}^{(0)}$. We can do the same for the second order in ϵ. Before but, we can obtain a new equation combining all the equations in B.3.23, which if we multiply again per the velocity we obtain a new couple of equations:

$$
\begin{aligned}
\partial_{t_1} f_i^{(eq)} + \frac{2\tau - \Delta t}{2\tau} \left(\frac{\partial}{\partial t_0} f_i^{(1)} + \nabla \cdot \boldsymbol{c}_i f_i^{(1)} \right) &= -\frac{\Delta t}{\tau} f_i^{(2)}, \\
\partial_{t_1} \boldsymbol{c}_{i,m} f_i^{(eq)} + \frac{2\tau - \Delta t}{2\tau} \left(\frac{\partial}{\partial t_0} \boldsymbol{c}_{i,m} f_i^{(1)} + \nabla \cdot \boldsymbol{c}_{i,m} \boldsymbol{c}_{i,n} f_i^{(1)} \right) &= -\frac{\Delta t}{\tau} \boldsymbol{c}_{i,m} f_i^{(2)}.
\end{aligned}
\tag{B.3.27}
$$

summing again these expressions we will obtain:

$$\begin{aligned} \partial_{t_1}\rho &= 0, \\ \partial_{t_1}(\rho \boldsymbol{u}) + \frac{2\tau - \Delta t}{2\tau}\nabla \cdot \Pi^{(1)}_{m,n} &= 0, \end{aligned} \tag{B.3.28}$$

where we expressed again a term as a tensor $\Pi^{(1)}_{m,n}$. Now, if we sum the first equation for each couple of expressions at B.3.26 and B.3.28, and considering how we expanded ∂t at Eq. B.3.22, we obtain the conservation of mass (Eq. A.1.4), which is the first requirement to obtain Navier-Stokes:

$$\partial_t \rho + \nabla \cdot \rho \boldsymbol{u} = 0. \tag{B.3.29}$$

The same can be done with the second equation of both B.3.26 and B.3.28, which gives an equivalent of conservation of momentum (Eq. A.1.6), which is the second requirement to obtain Navier-Stokes:

$$\partial_t \rho \boldsymbol{u} + \nabla \cdot \Pi = 0, \tag{B.3.30}$$

where $\Pi = \Pi^{(0)} + \frac{2\tau - \Delta t}{2\tau}\Pi^{(1)}$. The specific sums of $\boldsymbol{c}_i$ depend on the scheme that you use for LB[13]. Finally, the expression of f_i^{eq} also depend on some parameters that depend on your discretization. However, in all cases it is written as a Taylor expansion of $\boldsymbol{u}/c_s$, being c_s the Mach number. Consequently, the traditional LB only works with fluids at low Mach numbers, although there some LB built for high Mach numbers[265].

[13] As an example, for D2Q9 you can see [264]. There, $\Pi_{\alpha\beta}$ in Eq. (20) is equal to Π in my Eq. B.3.30.

B.4 | Colloidal systems

Typically it is said that matter can be found as a plasma, gas, liquid or solid. Although it is true that plasma is also encompassed as a fluid, I will omit this state in the present thesis since the presence of it is reduced to areas far from the topic of this thesis (mainly, naturally found in cosmos and atmospheric phenomena [266]).

For the rest of the three phases (see Fig. B.4), we can clearly differentiate them:

- Gas. Atoms or molecules are very far from each other, and the interactions between them are almost negligible. Consequently, there is no spatial order, and they will expand among all the available space they have. We can imagine gases as balls in a billiard: each ball moves in a different random direction, and the interactions are limited to direct hits between them.
- Liquid. Atoms or molecules are closer than in a gas. Because they are closer, they can have a first neighbour order, but not longer. Hence, there are more interactions, but they are weak. Liquids will adapt to the shape of their container but will have a fixed volume. From a global perspective, we still have the billiard image, but now we need to add longer distance interactions.
- Solid. Atoms or molecules are highly dense and interact strongly. They have a long neighbour order and thus, they will not adapt to the shape of the container.

Clearly, in nature we see these phases every day. And even we can see changes from one phase to another one as when ice melts. Although these definitions are very clear, there are substances that are not easy to classify among these phases. For example, consider the foam you could find in your cold beer. In which category will you fit it? Foam can adapt to the shape of its container, but no one will consider it as a liquid. Foam is full of air, but neither is a gas, because it is much denser than a gas. But still, there is something that reminds you to a fluid. Of course, is nothing like a solid. Why is so difficult to classify the foam? The truth behind it is that foam is not a simply fluid where atoms or molecules of the same or different species are simply well mixed and are miscible, but it is a much more complex fluid. Strictly, it is what is known as a *colloidal system*, a system where a microscopically sized substance, also refereed as a *colloid*, is suspended throughout another major substance. In the case of the foam of your cold beer, a gas is dispersed into a liquid, but we can have different combinations of phases

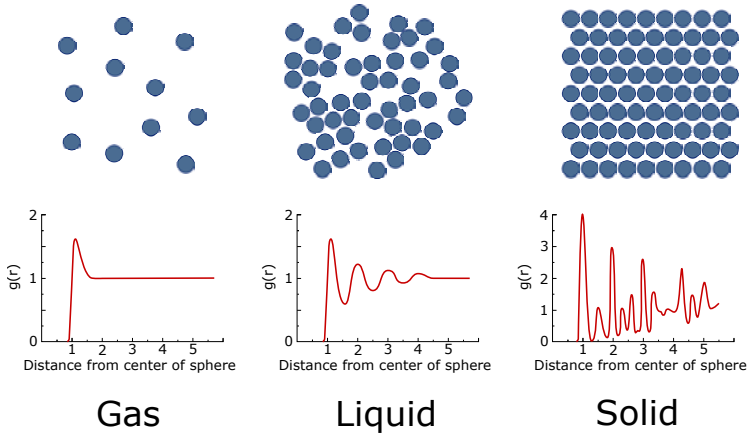


Figure B.4: Common states of nature. Gases are low dense systems, while liquids are denser, and solids are compact. States are easily to differentiate by looking at the atomic pair distribution functions g(r), where peaks inform about the relative order of atoms or molecules (described as a solid sphere) between themselves and their neighbours. *Images authorship. Own image, inspired on [59].*

too as shown in Fig. B.5. Depending on the phases we combine, colloidal systems will have different properties. For example, gels present networks giving them non-Newtonian properties while solid foams can be very light and elastic[14].

If we attend to the definition of these colloidal systems, we can see how our active particles will also be a colloidal system, where the major substance is the liquid where they move and the minor is a solid, themselves. Since life is intrinsically related with water, and we are biomimicking nature, most of our motors will move in a water-based system and thus, the major substance will be water. The fluid composed of the water and the particles will be a complex fluid, but in this thesis I did not look from the macroscale, but the microscale. Hence, if we zoom at the size of the particles, the environment is not complex at all. To add more complexity one can think about the spatial limits of motion of these particles, and the interactions they can have with them or other colloids present in the system. The spatial limits of motion for these particles are those surfaces where two phases are in contact. To these regions we call them phase boundaries or *interfaces*. Interfaces are very interesting because these are regions which do not have either all the properties of one or the other phase, and can bring new phenomena to the system. Furthermore, if we want one day to use these particles in some real application, our systems will be full of interfaces since they are common in nature. For example, there are two applications where active particles are promising. One is for environmental applications, and the other one for drug delivery.

[14]For instance, the metallic microlattice[267] is compound of 99.99% air and if you only consider the solid part, its density is just of 0.9 mg/cm^3 at room temperature (air density is 1.18 mg/cm^3). If you compressed it to half of its original height, it will recover up to the 98% of its original height.

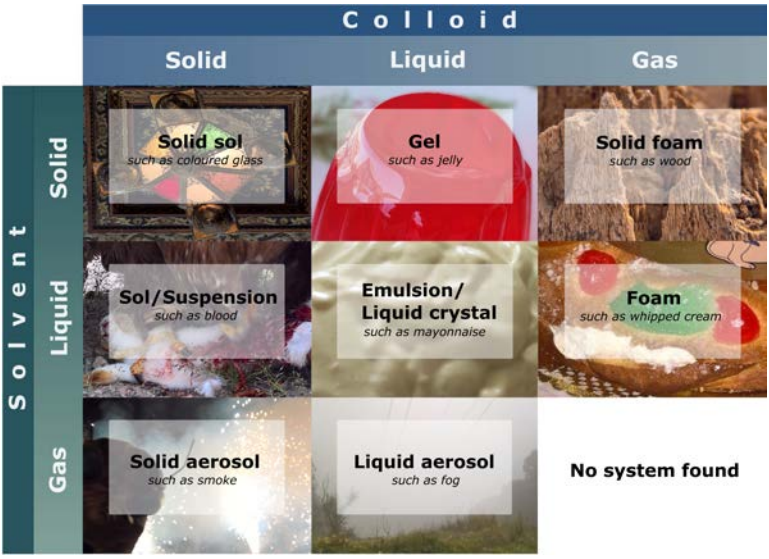


Figure B.5: Types of colloidal systems. We can create colloidal systems mixing substances within the same or different phase, arising to different material properties. The only exception is for the gas-gas colloidal system, to which no colloids are known, although helium and xenon can be immiscible under some conditions[268]. *Image authorship. Own images.*

Environmental applications are related with water cleaning since water is essential for life and we pollute it every day. For instance, some studies show how active particles can remove heavy metals such as lead from water[19], but also to remove organic pollutants[269]. Both of them can really damage the environment and our health[270, 271], and hence it is very important to remove them. These substances are usually found as industrial and agricultural waste, and specific wastewater treatment is made, but companies do not always fulfil good environmental policies. Hence, this research can be useful for future wastewater treatment. In the same line, something that common wastewater treatment for domestic water usually treat first is for oil removal from water[272]. Thus, imagine how important must be this process. Although the removal of oil may work for wastewater treatments, sometimes it is not possible to use a wastewater treatment. As with the heavy metals or organic pollutants, there can be spills. Hence, there will appear a large quantity of oil-water interfaces. As an extreme case, imagine we would need to clean an oil spill, as the one that happened a decade ago in the Gulf of Mexico, where between 500-800 million of litres of oil were released[273]. Can you imagine how many oil droplets could be there? Of course, we are far to use these particles in such disasters, but we have just started removing oil with them[38].

On the other hand, drug delivery systems also seem suitable for using these particles, but in our bodies, everything is surrounded by an interface. Blood vessels are just like a tube, but the tube can be porous or straight and hence we can have different topology of interfaces[274]. Moreover, inside the channels, there can be many things, such as fat molecules, which can join the walls of the tube creating a different interface and of course, arteriosclerosis[275], which can led into a coronary artery disease. If instead of using the blood vessels, we inject these particles in other regions, then we will probably find more complex media, since the body is full of a viscoelastic media or gels. Between our cells there is a rich mixture we refer as the *extra cellular matrix* (ECM), which provide biochemical and structural support to the cells[274]. This ECM is full of fibers, proteins and minerals, but it may be different depending on the cells that create them. In any case, it is a gel where our particles would move differently and will stack with many things. Thus, we will find many interfaces around the body.

These have been just a couple of examples, but as the comprehension and production of these systems will develop, the number of relevant applications will increase fast.

```
# Properties for output video
fourcc = cv2.VideoWriter_four
video_name = video_name_origi

# Open new video
cap = cv2.VideoCapture(video_
height = cap.get(cv2.CAP_PROP
width = cap.get(cv2.CAP_PROP_
# NOTE:
# fps = cap.get(cv2.CAP_PROP_
height2 = height/2
width2 = width/2
out = cv2.VideoWriter(video_n
cap.set(1,0) # Start in the f
frame_now = 0

# Properties to be calculated
init_time=[-1,-1] # [contact
final_time=[-1,-1] # [end con
slope = [float('nan'), float(
intercept = [float('nan'), fl
r_value = [0, 0, 0, 0]
p_value = [0, 0, 0, 0]
std_err = [0, 0, 0, 0]

# For each frame
while(cap.isOpened()) and cou
  # Read the frame
  ret, frame = cap.read()

  # If there's no frame, exit
  if hasattr(frame, 'shape') :
    break

  # Obtain info from the fram
  rows,cols,d = frame.shape
  dst = frame

  # Calculate distance of par
  distance = math.sqrt((x_dat

  # Calculate incoming approx
  #   We consider a distance
  # Obtain first time and pri
  if init_time[0]<0 and init_
    init_time[1] = count
    dst = cv2.drawMarker(fram
  elif init_time[1]>-1:
    dst = cv2.drawMarker(fram

  # We consider a distance of
  if init_time[0]<0 and dista
    init_time[0] = count

    # Do linear fits!
    if init_time[0]-init_time
```

Software

"The wisest men follow their own direction."

Euripides

C.1 | Compressing videos

C.1.1 Lif2Avi software

This software is available at https://github.com/Scolymus/Lif2Avi. To use it, simply import the git project in Eclipse software https://www.eclipse.org/downloads/ following the instructions that there are in the repository link. In principle, this is a plugin for Fiji, and one could extract the plugin to install it in any personal version.

C.1.2 Lif2Mkv software

This software is available at https://github.com/Scolymus/Lif2Mkv. To use it, simply import the git project in Eclipse software https://www.eclipse.org/downloads/ following the instructions that there are in the repository link.

C.2 | Transforming particles coordinates

C.2.1 Moving the frame of reference

The code used for my thesis was adapted to the output from the Albert's tracking code used. Hence, since every user could have a different output, I decided to clean the code and avoid the sections for reading the csv files where the data is stored. A clean version of these codes are found at https://github.com/Scolymus/Droplets. There are a few videos examples in the same folder.

First, track the droplet in your video. Once we obtain the coordinates for a droplet, we can use "droplet_centering.py" code. This file reads all the videos in a folder, and then for each one will move each frame to have all frames centred at the position of the (x,y) coordinates of your droplet. As an example, the code shows how to do this for a folder, where each video has one droplet to centre, but each video could have many droplets. Just introduce a new loop to do a new video per droplet if needed. Once finished, track the particles in the videos created by this code.

Next, use "droplet_particles_contact.py". This file will use the videos from the last output. It needs to feed the positions of the particles tracked before. As an example, here I just considered one video, but many particles for each video. The software will output a video showing the direction of incoming/outgoing to the droplet and the positions where it entered and left the droplet. For each video we need to introduce manually how many turns did each particle as a function of half turns. For example, if the particle did less than half turn, we will have to introduce 0. If it did more than half but less than 1, 0.5. If it did more than 1 but less than 1.5, 1. If it did more than 1.5 but less than 2, 1.5, and so on. The exact angle will be then calculated by the software and added to this number. At the end of the code, we will see how we can print different important variables calculated by the software.

C.2.2 From Cartesian coordinates to Cell coordinates

To change from (x,y) coordinates to cell coordinates we need to create the cell boxes. We can implement the geometry desired, but the basics are shown in https://github.com/Scolymus/Change_coordinates_shapely. This project includes a simplified version of the code I used during the thesis.

Cell boxes are created by creating Shapely Polygon objects. These objects are stored in an array, which later is used to check the position of the particle. For example, in function "draw_box" we have the code to draw boxes as a chessboard. Then, in function "trajectories_with_indexes" we transform the (x,y) coordinates to cell coordinates using the previous boxes created.

To edit properly later the boxes position we can also manage a keyboard call as shown in the "keyboard_control_boxes" function, where only the motion of the selected box to the left is shown.

C.3 | Tracking using neural networks

This software was coded in Python 3 under Ubuntu 18.04. Because the required software to run this tracking code is also avalaible for Microsoft Windows, it should also run under this operative system, but I never tried. The following sections are an instructions manual to install the tracking software for an Ubuntu system. Notice that when copying the instructions from the pdf, some lines may be cut.

C.3.1 How to compile OPENCV with CUDA enabled

The tracking software detects particles using the OPENCV library. To boost this process, OPENCV must be compiled with CUDA and cuDNN, but this step is only avalaible if we use a NVIDIA GPU. CUDA is a library to run code in parallel, while cuDNN is a library to calculate faster algorithms for neuronal networks.

However, not all NVIDIA GPUs can boost OPENCV for our requirements. Each NVIDIA GPU has a *Compute capability* number. This number informs the user what can do the GPU to parallelize calculus, and depends on the hardware of the graphic card. Thus, if we need a higher capability we need to buy a new GPU. The minimum version for our porpoises is 5.3. Given the name of the graphic card, we can check at https://developer.nvidia.com/cuda-gpus#compute which is the compute capability. To know the GPU model one can run in the terminal:

Check NVIDIA Compute capability

```
nvidia-smi
```

If the Compute capability is 5.3 or higher, we can install CUDA and cuDNN. Instructions to install both software are well explained at:

- CUDA. Download CUDA from https://developer.nvidia.com/cuda-downloads and the installation instructions from https://docs.nvidia.com/cuda/ cuda-installation-guide-linux/index.html.
- cuDNN. Download cuDNN from https://developer.nvidia.com/cuDNN and the installation instructions from https://docs.nvidia.com/deeplearning/ cudnn/install-guide/index.html

After installing NVIDIA's software, we need to install a few libraries to compile OPENCV:

Installing packages necessary to compile OPENCV

```
sudo apt-get install build-essential
sudo apt-get install cmake git libgtk2.0-dev
    pkg-config libavcodec-dev libavformat-dev
    libswscale-dev python-dev python3-dev
    python-numpy python3-numpy libtbb2 libtbb-dev
    libjpeg-dev libpng-dev libtiff-dev
    libjasper-dev libdc1394-22-dev
```

Next, we will obtain the OPENCV source code and the OPENCV contrib source code, which contains some modules we will need for our OPENCV compilation:

Downloading OPENCV source code

```
cd ~/
mkdir OPENCV
cd OPENCV
wget
    https://github.com/opencv/opencv/archive/master.zip
unzip master.zip
rm master.zip
wget
    https://github.com/opencv/opencv_contrib/archive/
    master.zip
unzip master.zip
rm master.zip
mv opencv-master opencv
mv opencv_contrib-master opencv_contrib
cd opencv
mkdir build
cd build
```

Now we can run make. Change the number at CUDA_ARCH_BIN with your compute capability. Also, we might need to change the 8 at $make - j8$. This 8 is the number of CPU cores to use during compilation. This step is very slow; therefore, it would be useful to change the number by the maximum number of cores the CPU has.

Compile OPENCV

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D INSTALL_C_EXAMPLES=OFF \
-D OPENCV_ENABLE_NONFREE=ON \
-D WITH_CUDA=ON \
-D WITH_CUDNN=ON \
-D OPENCV_DNN_CUDA=ON \
-D ENABLE_FAST_MATH=1 \
-D CUDA_FAST_MATH=1 \
-D CUDA_ARCH_BIN=7.0 \
-D WITH_CUBLAS=1 \
-D
    OPENCV_EXTRA_MODULES_PATH=~/OPENCV/opencv_contrib
    /modules \
-D HAVE_opencv_python3=ON \
-D
    PYTHON_EXECUTABLE=~/.virtualenvs/opencv_cuda/bin/
    python \
-D BUILD_EXAMPLES=ON -D
    OPENCV_GENERATE_PKGCONFIG=YES ..
make -j8
sudo make install
```

Now we should have OPENCV installed correctly. If we find any problem, before compiling again, please remove the build folder and make it again.

C.3.2 How to install YOLO to train networks

First, we will download YOLO:

Downloading YOLO source code

```
cd ~/
mkdir YOLO
cd YOLO
wget https://github.com/AlexeyAB/darknet/archive/
    master.zip
unzip master.zip
rm master.zip
cd darknet-master
```

Now, we need to modify the file $Makefile$ with a text editor. Change the 0 at the first lines to 1 to activate these components during the compiling:

Changing parameters for compilation

```
GPU=1
CUDNN=1
CUDNN_HALF=1
OPENCV=1
AVX=1
OPENMP=1
```

Notice that not all GPU support CUDNN_HALF. If the GPU does not support it, it should be 0. The GPU will support it if it has *tensor cores*. After these lines, depending on the YOLO version there might appear some commented line (they have a # at the beginning) that start by "ARCH". If we can find the GPU architecture, remove the # before the ARCH for your architecture. Finally, we just need to compile it:

Compiling YOLO

```
make
```

If at this moment you receive an error about nvcc, please open the file at $/home/yourusername/.bashrc$ and add the following lines at the end of the file. Do change XYZ by the version of your CUDA, eg.: cuda-XYZ -> cuda-11.3.

Solving nvcc error

```
export CUDA_HOME=/usr/local/cuda-XYZ
export LD_LIBRARY_PATH=${CUDA_HOME}/lib64

PATH=${CUDA_HOME}/bin:${PATH}
export PATH
```

C.3.3 Software to create the training dataset, train YOLO with your dataset and track

By the end of this thesis, I decided to package all the tracking codes under a single python software. This software is available at https://github.com/Scolymus/Yolo_Tracking.

There might be some libraries that we should install such as numpy, urllib3, cmapy, lxml, shapely, scipy or ctypes before we can run the code. Depending on the python installation, an error will indicate which library we will need to install. To install a python3 library use:

Installing python3 libraries

```
sudo apt-get install python3-pip
pip3 install package-names
```

where package-names are the libraries we want to install.

To run the software enter:

Running the software

```
python3 ./run.py
```

Once running, we will see a window as in Fig. C.1. This window allows to access the different windows present in sections 4.3 and 4.4. Follow from left to right the buttons to create the dataset, prepare it, train the network, and track the particles.

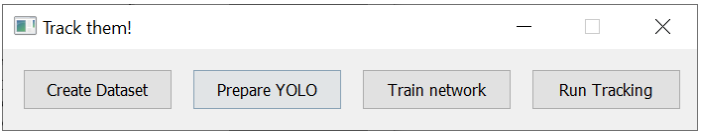


Figure C.1: Complete software to train and track particles. Main window that allows the user to choose between different python scripts.

C.4 | Visualizing Ludwig output

The complexity of the calculus performed with Ludwig software made the developers to focus on the simulation process. Hence, it is understandable that the software runs without any visual interaction with the user. Only after a simulation is performed, the user can process the results to visualize them.

The output given by Ludwig consists of several binary files, each containing its own structure, for each time that the user selected to print a file. Therefore, after the simulation we need to process these files to obtain the interested data. There are some .c utility files coming with Ludwig that allow the user to extract binary data into plain text files, but sometimes even with this readable data is difficult to visualize them using other software such as Gnuplot. One of these util files include an extraction to use the data with Paraview software, but I only found it after some years of using Ludwig, and it was neither easy nor adaptable to my porpoises.

Consequently, as part of my PhD I had to develop different tools to understand Ludwig output. Some of the files were based on these .c utilities, and hence I do not find them necessary to show, but there were a couple of tricks that were very useful for me. Some imply an advance usage of Gnuplot, but I also developed a complete software for visualizing the results in 3D, while interacting with them.

C.4.1 Gnuplot script for movie creation

One of the main problems while introducing a new free energy was to see the evolution in time of the simulating box, especially for the order parameters evolution. To see easier the evolution of the order parameters I decided to visualize a plane of the simulating box and record it for all the simulated time.

However, when plotting with Gnuplot software, one usually plots series, where the axis of a series can be the time evolution. Here what I needed to plot was a surface, but a surface moving in time. To achieve this effect, we can plot a surface with "$w\ image$" command, and then repeat this process for every file using a loop inside the Gnuplot script. Once we have the pictures, we can call an external software such as ffmpeg to join all the pictures into a single .mp4 movie. ffmpeg allows to join pictures with the same name of the file but differing by a successive number, e.g., picture1.png, picture2.png, ..., pictureN.png, where N is the number of pictures you have.

In https://github.com/Scolymus/Ludwig_gnuplots/ there are the scripts I used.

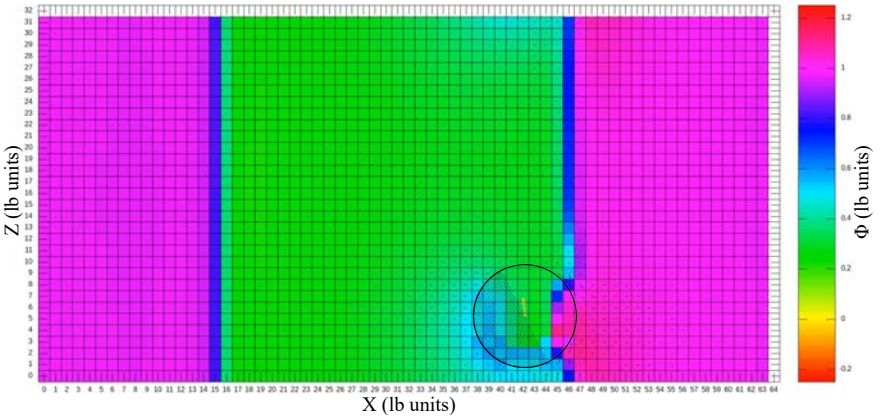


Figure C.2: Example of plane visualization. All lattice cells are represented with squares, where each square has as a colour representing the quantity of order parameter in that lattice cell. The particle is represented by a circle. Because of its orientation, the particle is a perfect Janus seen from this direction and can be drawn as half circle with lines (active side) and half circle without (inactive side). Cells inside these regions are also shown for mainly computational reasons, but order parameter for cells inside do not have a physical meaning. Velocity of the particle centre of mass is represented with an orange arrow.

C.4.2 3D Visualizer

At some moment of the thesis, it was necessary to see the configuration of the whole box and how the particle orients in 3D. Although one can print in a plot the orientation of the particle by plotting its spherical coordinates, sometimes it is not easy to understand what is happening, even more if there are many particles in the system. To avoid any misinterpretation, I created a software in Java using Open Graphics Library (OPENGL). OPENGL is a programming interface (API) for rendering 2D and 3D vectors graphics, usually used for computer-aided design (CAD), game development, virtual reality and in general, anything related to rendering 2D and 3D graphics. It is a cross-language, cross-platform, and open-source project. For this porpoise, I used the last major version 4, launched in 2010.

With this software, one can represent from Ludwig output:

- All the particles in 3D. It is prepared to draw half-half of the Janus particles in different colours (not necessary if you do not want).
- All the walls in the simulation box if you need them.
- All the lattice nodes as a coloured points. The colour represents the quantity of the order parameter to see how oxygen or liquid/liquid is distributed. All

points can be coloured with the same opacity or increasing opacity for those around the particles for a better visualization.

The software will read all the files of your output and will run the results as an interactive movie because while playing the results, the user can move inside the movie in 3D, as if it were a character in a game. We can move the camera with the keyboard (up, down, left, right keys) and the mouse to point the direction (left click while moving the mouse).

To use it, simply install Eclipse software https://www.eclipse.org/downloads/ following the instructions available in the repository link. Then, import the git project present at https://github.com/Scolymus/Ludwig_visualization/ and follow the instructions present in the repository.

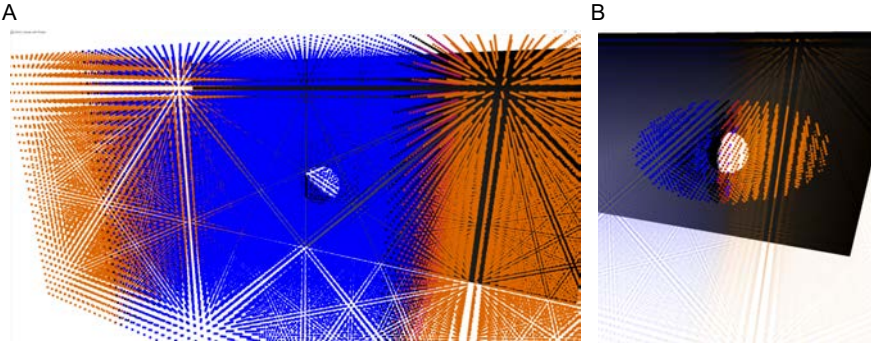


Figure C.3: Example of 3D visualizer software. All lattice cells are represented with points, where each point has as a colour representing the quantity of order parameter in that lattice cell. After the lattice cells we can see a black rectangle representing a solid wall. Inside the lattice cells there is a Janus particle. **A)** All points have the same transparency. **B)** Only those surrounding the particle are opaque.

Index

Bibliography

[1] Miguel Luis León-Portilla. *Aztec Thought and Culture: A Study of the Ancient Nahuatl Mind*. University of Oklahoma Press, 1990, p. 272. ISBN: 978-0806122953.

[2] Hasan S. Padamsee. *Unifying the Universe: The Physics of Heaven and Earth*. Ed. by CRC Press. 2002, p. 668. ISBN: 978-0750307581.

[3] Kenneth Dorter. "Metaphysics and Morality in Neo-Confucianism and Greece: Zhu Xi, Plato, Aristotle, and Plotinus". In: *Dao* 8.3 (Sept. 2009), pp. 255–276. ISSN: 1569-7274. DOI: 10.1007/s11712-009-9125-x. URL: https://doi.org/10.1007/s11712-009-9125-x.

[4] Nuri Adıgüzel. "The Concept of 'Nature' in Peripatetic Islamic Philosophers [Meşşâî İslam Filozoflarında 'Tabiat' Kavramı]". In: (July 2018). DOI: 10.5281/ZENODO.1421981. URL: https://zenodo.org/record/1421981.

[5] Galileo Galilei and Henry Crew. *Dialogues Concerning Two New Sciences*. Ed. by Martino Fine Books. 2015, p. 328. ISBN: 978-1614277941.

[6] Sriram Ramaswamy. "The Mechanics and Statistics of Active Matter". In: *Annual Review of Condensed Matter Physics* 1.1 (July 2010), pp. 323–345. ISSN: 1947-5454. DOI: 10.1146/annurev-conmatphys-070909-104101. URL: https://doi.org/10.1146/annurev-conmatphys-070909-104101.

[7] Richard P. Feynman. "There's Plenty of Room at the Bottom". In: *Engineering and Science* 23 (5) (1960), pp. 22–36. URL: http://calteches.library.caltech.edu/1976/1/1960Bottom.pdf.

[8] Brendan B. Larsen et al. "Inordinate Fondness Multiplied and Redistributed: the Number of Species on Earth and the New Pie of Life". In: *The Quarterly Review of Biology* 92.3 (Sept. 2017), pp. 229–265. ISSN: 0033-5770. DOI: 10.1086/693564. URL: https://www.journals.uchicago.edu/doi/10.1086/693564.

[9] Yinon M. Bar-On, Rob Phillips, and Ron Milo. “The biomass distribution on Earth”. In: *Proceedings of the National Academy of Sciences* 115.25 (June 2018), pp. 6506–6511. ISSN: 0027-8424. DOI: 10.1073/PNAS.1711842115. URL: https://www.pnas.org/content/115/25/6506.

[10] Suzanne Austin Alchon. *A pest in the land: new world epidemics in a global perspective*. University of New Mexico Press, 2003. ISBN: 978-0-8263-2871-7.

[11] WHO. *Global Health Observatory (GHO) data: Top 10 causes of death*. 2018. URL: https://www.who.int/gho/mortality%7B%5C_%7Dburden%7B%5C_%7Ddisease/causes%7B%5C_%7Ddeath/top%7B%5C_%7D10/en/.

[12] Ah-Young Jee et al. “Catalytic enzymes are active matter”. In: *Proceedings of the National Academy of Sciences* 115.46 (Nov. 2018), E10812–E10821. ISSN: 0027-8424. DOI: 10.1073/pnas.1814180115. URL: http://www.pnas.org/lookup/doi/10.1073/pnas.1814180115.

[13] Pamela Lyon. “The cognitive cell: bacterial behavior reconsidered”. In: *Frontiers in Microbiology* 6 (Apr. 2015). ISSN: 1664-302X. DOI: 10.3389/fmicb.2015.00264. URL: http://journal.frontiersin.org/article/10.3389/fmicb.2015.00264/abstract.

[14] Despoina A.I. Mavridou et al. “Bacteria Use Collective Behavior to Generate Diverse Combat Strategies”. In: *Current Biology* 28.3 (Feb. 2018), 345–355.e4. ISSN: 09609822. DOI: 10.1016/j.cub.2017.12.030. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960982217316639.

[15] Tamás Vicsek and Anna Zafeiris. “Collective motion”. In: *Physics Reports* 517.3-4 (Aug. 2012), pp. 71–140. ISSN: 03701573. DOI: 10.1016/j.physrep.2012.03.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0370157312000968.

[16] Jamel Ali et al. “Bacteria-inspired nanorobots with flagellar polymorphic transformations and bundling”. In: *Scientific Reports* 7.1 (Dec. 2017), p. 14098. ISSN: 2045-2322. DOI: 10.1038/s41598-017-14457-y. URL: http://www.nature.com/articles/s41598-017-14457-y.

[17] D. G. Gibson et al. “Creation of a Bacterial Cell Controlled by a Chemically Synthesized Genome”. In: *Science* 329.5987 (July 2010), pp. 52–56. ISSN: 0036-8075. DOI: 10.1126/science.1190719. URL: https://www.sciencemag.org/lookup/doi/10.1126/science.1190719.

[18] Samuel Sánchez, Lluís Soler, and Jaideep Katuri. “Chemically Powered Micro- and Nanomotors”. In: *Angewandte Chemie International Edition* 54.5 (Jan. 2015), pp. 1414–1444. ISSN: 14337851. DOI: 10.1002/anie.201406096. URL: http://doi.wiley.com/10.1002/anie.201406096.

[19] Diana Vilela et al. “Graphene-Based Microbots for Toxic Heavy Metal Removal and Recovery from Water”. In: *Nano Letters* 16.4 (Apr. 2016), pp. 2860–2866. ISSN: 1530-6984. DOI: 10.1021/acs.nanolett.6b00768. URL: https://pubs.acs.org/doi/10.1021/acs.nanolett.6b00768.

[20] Tania Patino et al. “Self-Sensing Enzyme-Powered Micromotors Equipped with pH-Responsive DNA Nanoswitches”. In: *Nano Letters* 19.6 (June 2019), pp. 3440–3447. ISSN: 1530-6984. DOI: 10.1021/acs.nanolett.8b04794. URL: https://pubs.acs.org/doi/10.1021/acs.nanolett.8b04794.

[21] Ana C. Hortelão et al. “Enzyme-Powered Nanobots Enhance Anticancer Drug Delivery”. In: *Advanced Functional Materials* 28.25 (June 2018), p. 1705086. ISSN: 1616301X. DOI: 10.1002/adfm.201705086. URL: http://doi.wiley.com/10.1002/adfm.201705086.

[22] Wenxing Wang et al. “Facile Synthesis of Uniform Virus-like Mesoporous Silica Nanoparticles for Enhanced Cellular Internalization”. In: *ACS Central Science* 3.8 (Aug. 2017), pp. 839–846. ISSN: 2374-7943. DOI: 10.1021/acscentsci.7b00257. URL: https://pubs.acs.org/doi/10.1021/acscentsci.7b00257.

[23] Rachel P. Doherty et al. “Catalytically propelled 3D printed colloidal microswimmers”. In: *Soft Matter* 16.46 (2020), pp. 10463–10469. ISSN: 1744-683X. DOI: 10.1039/D0SM01320J. URL: http://xlink.rsc.org/?DOI=D0SM01320J.

[24] Guinness world Records. *Smallest nanotube jet engine*. 2016. URL: https://www.guinnessworldrecords.com/world-records/smallest-jet-engine/ (visited on 2020).

[25] Alexander K Epstein et al. “Biofilm attachment reduction on bioinspired, dynamic, micro-wrinkling surfaces”. In: *New Journal of Physics* 15.9 (Sept. 2013), p. 095018. ISSN: 1367-2630. DOI: 10.1088/1367-2630/15/9/095018. URL: https://iopscience.iop.org/article/10.1088/1367-2630/15/9/095018.

[26] W. Wang et al. “Catalytically powered dynamic assembly of rod-shaped nanomotors and passive tracer particles”. In: *Proceedings of the National Academy of Sciences* 110.44 (Oct. 2013), pp. 17744–17749. ISSN: 0027-8424. DOI: 10.1073/pnas.1311543110. URL: http://www.pnas.org/cgi/doi/10.1073/pnas.1311543110.

[27] Xavier Arqué et al. “Intrinsic enzymatic properties modulate the self-propulsion of micromotors”. In: *Nature Communications* 10.1 (Dec. 2019), p. 2826. ISSN: 2041-1723. DOI: 10.1038/s41467-019-10726-8. URL: http://www.nature.com/articles/s41467-019-10726-8.

[28] Bumjin Jang et al. “Multiwavelength Light-Responsive Au/B-TiO 2 Janus Micromotors”. In: *ACS Nano* 11.6 (June 2017), pp. 6146–6154. ISSN: 1936-0851. DOI: 10.1021/acsnano.7b02177. URL: https://pubs.acs.org/doi/10.1021/acsnano.7b02177.

[29] Juan Ruben Gomez-Solano et al. “Tuning the motility and directionality of self-propelled colloids”. In: *Scientific Reports* 7.1 (Dec. 2017), p. 14891. ISSN: 2045-2322. DOI: 10.1038/s41598-017-14126-0. URL: http://www.nature.com/articles/s41598-017-14126-0.

[30] L Sherwood, J Willey, and C J Woolverton. *Prescott's Microbiology*. McGraw-Hill Education, 2013. ISBN: 9780073402406. URL: https://books.google.es/books?id=5dK0MwEACAAJ.

[31] Daniel B. Kearns. “A field guide to bacterial swarming motility”. In: *Nature Reviews Microbiology* 8.9 (Sept. 2010), pp. 634–644. ISSN: 1740-1526. DOI: 10.1038/nrmicro2405. URL: http://www.nature.com/articles/nrmicro2405.

[32] Mark J. McBride. “Bacterial Gliding Motility: Multiple Mechanisms for Cell Movement over Surfaces”. In: *Annual Review of Microbiology* 55.1 (Oct. 2001), pp. 49–75. ISSN: 0066-4227. DOI: 10.1146/annurev.micro.55.1.49. URL: http://www.annualreviews.org/doi/10.1146/annurev.micro.55.1.49.

[33] Beiyan Nan and David R. Zusman. “Novel mechanisms power bacterial gliding motility”. In: *Molecular Microbiology* 101.2 (July 2016), pp. 186–193. ISSN: 0950382X. DOI: 10.1111/mmi.13389. URL: http://doi.wiley.com/10.1111/mmi.13389.

[34] F. J. Siemensma. *Microworld, world of amoeboid organisms.* URL: https://www.arcella.nl/2421-2/ (visited on 2020).

[35] Rémi Dreyfus et al. “Microscopic artificial swimmers”. In: *Nature* 437.7060 (Oct. 2005), pp. 862–865. ISSN: 0028-0836. DOI: 10.1038/nature04090. URL: http://www.nature.com/articles/nature04090.

[36] Brian J. Williams et al. “A self-propelled biohybrid swimmer at low Reynolds number”. In: *Nature Communications* 5.1 (May 2014), p. 3081. ISSN: 2041-1723. DOI: 10.1038/ncomms4081. URL: http://www.nature.com/articles/ncomms4081.

[37] Jonathan R Howse et al. “Self-Motile Colloidal Particles: From Directed Propulsion to Random Walk”. In: *Physical Review Letters* 99.4 (July 2007), p. 048102. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.99.048102. arXiv: 0706.4406. URL: https://journals-aps-org.sire.ub.edu/prl/pdf/10.1103/PhysRevLett.99.048102%20https://link.aps.org/doi/10.1103/PhysRevLett.99.048102.

[38] Lei Wang et al. “Lipase-Powered Mesoporous Silica Nanomotors for Triglyceride Degradation”. In: *Angewandte Chemie International Edition* 58.24 (June 2019), pp. 7992–7996. ISSN: 1433-7851. DOI: 10.1002/anie.201900697. URL: https://doi.org/10.1002/anie.201900697.

[39] Wei Wang et al. “Autonomous Motion of Metallic Microrods Propelled by Ultrasound”. In: *ACS Nano* 6.7 (July 2012), pp. 6122–6132. ISSN: 1936-0851. DOI: 10.1021/nn301312z. URL: https://pubs.acs.org/doi/10.1021/nn301312z.

[40] Antoine Bricard et al. “Emergence of macroscopic directed motion in populations of motile colloids”. In: *Nature* 503.7474 (Nov. 2013), pp. 95–98. ISSN: 0028-0836. DOI: 10.1038/nature12673. URL: http://www.nature.com/articles/nature12673.

[41] Berta Esteban-Fernández de Ávila et al. “Micromotor-enabled active drug delivery for in vivo treatment of stomach infection”. In: *Nature Communications* 8.1 (2017), p. 272. ISSN: 2041-1723. DOI: 10.1038/s41467-017-00309-w. URL: https://doi.org/10.1038/s41467-017-00309-w.

[42] Pierre-Gilles de Gennes. “Soft Matter (Nobel Lecture)”. In: *Angewandte Chemie International Edition in English* 31.7 (July 1992), pp. 842–845. ISSN: 0570-0833. DOI: 10.1002/anie.199208421. URL: http://doi.wiley.com/10.1002/anie.199208421.

[43] Tania Patiño et al. “Influence of Enzyme Quantity and Distribution on the Self-Propulsion of Non-Janus Urease-Powered Micromotors”. In: *Journal of the American Chemical Society* 140.25 (June 2018), pp. 7896–7903. ISSN: 0002-7863. DOI: 10.1021/jacs.8b03460. URL: https://pubs.acs.org/doi/10.1021/jacs.8b03460.

[44] Anthony L. Mescher. *Junqueira’s basic histology : text and atlas*. 14th ed. McGraw-Hill Medical, 2016, p. 560. ISBN: 9780071842709.

[45] Dmitrii Pavlov and A Kasumyan. “Patterns and mechanisms of schooling behavior in fish: A review”. In: *Journal of Ichthyology* 40 (Dec. 2000), S163–S231.

[46] Andrea Cavagna and Irene Giardina. “Bird Flocks as Condensed Matter”. In: *Annual Review of Condensed Matter Physics* 5.1 (Mar. 2014), pp. 183–207. ISSN: 1947-5454. DOI: 10.1146/annurev-conmatphys-031113-133834. URL: https://doi.org/10.1146/annurev-conmatphys-031113-133834.

[47] Tamás Vicsek et al. “Novel Type of Phase Transition in a System of Self-Driven Particles”. In: *Phys. Rev. Lett.* 75.6 (1995), pp. 1226–1229. DOI: 10.1103/PhysRevLett.75.1226. URL: https://link.aps.org/doi/10.1103/PhysRevLett.75.1226.

[48] Agustí Brugués et al. “Forces driving epithelial wound healing”. In: *Nature Physics* 10.9 (Sept. 2014), pp. 683–690. ISSN: 1745-2473. DOI: 10.1038/nphys3040. URL: http://www.nature.com/articles/nphys3040.

[49] Olaya Rendueles and Jean-Marc Ghigo. “Multi-species biofilms: how to avoid unfriendly neighbors”. In: *FEMS Microbiology Reviews* 36.5 (Sept. 2012), pp. 972–989. ISSN: 1574-6976. DOI: 10.1111/j.1574-6976.2012.00328.x. URL: https://academic.oup.com/femsre/article-lookup/doi/10.1111/j.1574-6976.2012.00328.x.

[50] Helena Massana-Cid et al. “Tunable self-healing of magnetically propelling colloidal carpets”. In: *Nature Communications* 10.1 (Dec. 2019), p. 2444. ISSN: 2041-1723. DOI: 10.1038/s41467-019-10255-4. URL: http://www.nature.com/articles/s41467-019-10255-4.

[51] Tobias Bäuerle et al. “Self-organization of active particles by quorum sensing rules”. In: *Nature Communications* 9.1 (2018), p. 3232. ISSN: 2041-1723. DOI: 10.1038/s41467-018-05675-7. URL: https://doi.org/10.1038/s41467-018-05675-7.

[52] Melissa B. Miller and Bonnie L. Bassler. “Quorum Sensing in Bacteria”. In: *Annual Review of Microbiology* 55.1 (Oct. 2001), pp. 165–199. ISSN: 0066-4227. DOI: 10.1146/annurev.micro.55.1.165. URL: http://www.annualreviews.org/doi/10.1146/annurev.micro.55.1.165.

[53] M. C. Marchetti et al. “Hydrodynamics of soft active matter”. In: *Reviews of Modern Physics* 85.3 (July 2013), pp. 1143–1189. ISSN: 0034-6861. DOI: 10.1103/RevModPhys.85.1143. URL: https://link.aps.org/doi/10.1103/RevModPhys.85.1143.

[54] Clemens Bechinger et al. “Active Particles in Complex and Crowded Environments”. In: *Reviews of Modern Physics* 88.4 (Nov. 2016), p. 045006. ISSN: 0034-6861. DOI: 10.1103/RevModPhys.88.045006. URL: https://link.aps.org/doi/10.1103/RevModPhys.88.045006.

[55] Eric Lauga and Thomas R Powers. “The hydrodynamics of swimming microorganisms”. In: *Reports on Progress in Physics* 72.9 (Sept. 2009), p. 096601. ISSN: 0034-4885. DOI: 10.1088/0034-4885/72/9/096601. URL: http://stacks.iop.org/0034-4885/72/i=9/a=096601?key=crossref.736a5c13368e75b7395f94099aead8e4.

[56] Juliane Simmchen et al. “Topographical pathways guide chemical microswimmers”. In: *Nature Communications* 7.1 (Apr. 2016), p. 10598. ISSN: 2041-1723. DOI: 10.1038/ncomms10598. URL: http://www.nature.com/articles/ncomms10598.

[57] Wener Massa. *Crystal Structure Determination*. Second edi. Springer-Verlag Berlin Heidelberg New York, 2004. ISBN: 3540206442.

[58] X. Chen et al. “Symmetry-Protected Topological Orders in Interacting Bosonic Systems”. In: *Science* 338.6114 (Dec. 2012), pp. 1604–1606. ISSN: 0036-8075. DOI: 10.1126/science.1227224. URL: https://www.sciencemag.org/lookup/doi/10.1126/science.1227224.

[59] Jean-Louis Barrat and Jean-Pierre Hansen. *Basic Concepts for Simpleand Complex Liquids*. The press syndicate of the University of Cambridge, 2003. ISBN: 0-521-78344-5.

[60] Richard Hammond. *What is Surface Tension? | Richard Hammond's Invisible Worlds*. 2021. URL: https://www.youtube.com/watch?v=5NCOnr3VSAY.

[61] Philip S. Brown and Bharat Bhushan. “Mechanically durable liquid-impregnated honeycomb surfaces”. In: *Scientific Reports* 7.1 (Dec. 2017), p. 6083. ISSN: 2045-2322. DOI: 10.1038/s41598-017-06621-1. URL: http://www.nature.com/articles/s41598-017-06621-1.

[62] Brianne Christopher. *Tears of Wine and the Marangoni Effect*. 2015. URL: https://www.comsol.com/blogs/tears-of-wine-and-the-marangoni-effect/ (visited on 2021).

[63] Sambeeta Das et al. “Boundaries can steer active Janus spheres”. In: *Nature Communications* 6 (Dec. 2015), p. 8999. URL: https://doi.org/10.1038/ncomms9999%20http://10.0.4.14/ncomms9999%20https://www.nature.com/articles/ncomms9999%7B%5C#%7Dsupplementary-information.

[64] Jaideep Katuri et al. “Cross-stream migration of active particles”. In: *Science Advances* 4.1 (Jan. 2018), eaao1755. ISSN: 2375-2548. DOI: 10.1126/sciadv.aao1755. URL: https://advances.sciencemag.org/lookup/doi/10.1126/sciadv.aao1755.

[65] Jaideep Katuri et al. “Directed Flow of Micromotors through Alignment Interactions with Micropatterned Ratchets”. In: *ACS Nano* 12.7 (July 2018), pp. 7282–7291. ISSN: 1936-0851. DOI: 10.1021/acsnano.8b03494. URL: https://doi.org/10.1021/acsnano.8b03494.

[66] Xiaolu Wang et al. “Enhanced active motion of Janus colloids at the water surface”. In: *Soft Matter* 11.37 (July 2015), pp. 7376–7384. ISSN: 1744-683X. DOI: 10.1039/C5SM01111F. URL: http://xlink.rsc.org/?DOI=C5SM01111F.

[67] Xiaolu Wang et al. “Janus Colloids Actively Rotating on the Surface of Water”. In: *Langmuir* 33.48 (Dec. 2017), pp. 13766–13773. ISSN: 0743-7463. DOI: 10.1021/acs.langmuir.7b02353. URL: https://pubs.acs.org/doi/10.1021/acs.langmuir.7b02353.

[68] Kilian Dietrich et al. “Two-dimensional nature of the active Brownian motion of catalytic microswimmers at solid and liquid interfaces”. In: *New Journal of Physics* 19.6 (June 2017), p. 065008. ISSN: 1367-2630. DOI: 10.1088/1367-2630/aa7126. URL: https://iopscience.iop.org/article/10.1088/1367-2630/aa7126.

[69] P Malgaretti, M N Popescu, and S Dietrich. “Active colloids at fluid interfaces”. In: *Soft Matter* 12.17 (2016), pp. 4007–4023. DOI: 10.1039/C6SM00367B. URL: http://dx.doi.org/10.1039/C6SM00367B.

[70] Alvaro Domínguez et al. “Effective Interaction between Active Colloids and Fluid Interfaces Induced by Marangoni Flows”. In: *Physical Review Letters* 116.7 (Feb. 2016), p. 078301. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.116.078301. URL: https://link.aps.org/doi/10.1103/PhysRevLett.116.078301.

[71] T. Peter et al. “Numerical simulations of self-diffusiophoretic colloids at fluid interfaces”. In: *Soft Matter* 16.14 (2020), pp. 3536–3547. ISSN: 1744-683X. DOI: 10.1039/C9SM02247C. URL: http://xlink.rsc.org/?DOI=C9SM02247C.

[72] Barry Bradlyn et al. “Topological quantum chemistry”. In: *Nature* 547.7663 (July 2017), pp. 298–305. ISSN: 0028-0836. DOI: 10.1038/nature23268. URL: http://www.nature.com/articles/nature23268.

[73] Mikael C. Rechtsman et al. “Photonic Floquet topological insulators”. In: *Nature* 496.7444 (Apr. 2013), pp. 196–200. ISSN: 0028-0836. DOI: 10.1038/nature12066. URL: http://www.nature.com/articles/nature12066.

[74] Romain Fleury, Alexander B Khanikaev, and Andrea Alù. “Floquet topological insulators for sound”. In: *Nature Communications* 7.1 (Sept. 2016), p. 11744. ISSN: 2041-1723. DOI: 10.1038/ncomms11744. URL: http://www.nature.com/articles/ncomms11744.

[75] Johannes Loehr et al. “Colloidal topological insulators”. In: *Communications Physics* 1.1 (Dec. 2018), p. 4. ISSN: 2399-3650. DOI: 10.1038/s42005-017-0004-1. URL: http://www.nature.com/articles/s42005-017-0004-1.

[76] János K. Asbóth, László Oroszlány, and András Pályi. “The Su-Schrieffer-Heeger (SSH) Model”. In: 2016, pp. 1–22. DOI: 10.1007/978-3-319-25607-8_1. URL: http://link.springer.com/10.1007/978-3-319-25607-8%7B%5C_%7D1.

[77] Navketan Batra and Goutam Sheet. “Physics with Coffee and Doughnuts”. In: *Resonance* 25.6 (June 2020), pp. 765–786. ISSN: 0971-8044. DOI: 10.1007/s12045-020-0995-x. URL: http://link.springer.com/10.1007/s12045-020-0995-x.

[78] Naomichi Hatano and David R. Nelson. “Localization Transitions in Non-Hermitian Quantum Mechanics”. In: *Physical Review Letters* 77.3 (July 1996), pp. 570–573. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.77.570. URL: https://link.aps.org/doi/10.1103/PhysRevLett.77.570.

[79] Yuto Ashida, Zongping Gong, and Masahito Ueda. “Non-Hermitian Physics”. In: *Arxiv* (2020). arXiv: 2006.01837.

[80] Kohei Kawabata, Masatoshi Sato, and Ken Shiozaki. “Higher-order non-Hermitian skin effect”. In: *Physical Review B* 102.20 (Nov. 2020), p. 205118. ISSN: 2469-9950. DOI: 10.1103/PhysRevB.102.205118. URL: https://link.aps.org/doi/10.1103/PhysRevB.102.205118.

[81] Louis Lagardère et al. “Tinker-HP: a massively parallel molecular dynamics package for multiscale simulations of large complex systems with advanced point dipole polarizable force fields”. In: *Chemical Science* 9.4 (2018), pp. 956–972. ISSN: 2041-6520. DOI: 10.1039/C7SC04531J. URL: http://xlink.rsc.org/?DOI=C7SC04531J.

[82] Brian R. Martin and Graham Shaw. *Particle Physics*. Wiley, 2017, p. 474. ISBN: 9781118911907.

[83] Maricarmen Hernández-Rodríguez et al. “Current Tools and Methods in Molecular Dynamics (MD) Simulations for Drug Design”. In: *Current Medicinal Chemistry* 23.34 (Nov. 2016), pp. 3909–3924. ISSN: 09298673. DOI: 10.2174/0929867323666160530144742. URL: http://www.eurekaselect.com/openurl/content.php?genre=article%7B%5C&%7Dissn=0929-8673%7B%5C&%7Dvolume=23%7B%5C&%7Dissue=34%7B%5C&%7Dspage=3909.

[84] L. Gary Leal. *Advanced Transport Phenomena*. Cambridge: Cambridge University Press, 2007. ISBN: 9780511800245. DOI: 10.1017/CBO9780511800245. URL: http://ebooks.cambridge.org/ref/id/CBO9780511800245.

[85] K.R. Cooper. “The wind-tunnel simulation of surface vehicles”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 17.2 (Aug. 1984), pp. 167–198. ISSN: 01676105. DOI: 10.1016/0167-6105(84)90055-2. URL: https://linkinghub.elsevier.com/retrieve/pii/0167610584900552.

[86] RheoSense. *What Units Should I Use for Viscosity?* URL: https://www.rheosense.com/basics/viscosity-units (visited on 2021).

[87] R Edgeworth, B J Dalton, and T Parnell. “The pitch drop experiment”. In: *European Journal of Physics* 5.4 (Oct. 1984), pp. 198–200. ISSN: 0143-0807. DOI: 10.1088/0143-0807/5/4/003. URL: https://iopscience.iop.org/article/10.1088/0143-0807/5/4/003.

[88] The University of Queensland. *Pitch Drop experiment*. URL: https://smp.uq.edu.au/pitch-drop-experiment (visited on 2021).

[89] Maria Salta et al. “Designing biomimetic antifouling surfaces”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368.1929 (Oct. 2010), pp. 4729–4754. ISSN: 1364-503X. DOI: 10.1098/rsta.2010.0195. URL: https://royalsocietypub org/doi/10.1098/rsta.2010.0195.

[90] UNM Physics and Astronomy. *Laminar Flow*. 2007. URL: https://youtu.be/p08%7B%5C_%7DKlTKP50 (visited on 2021).

[91] E. M. Purcell. “Life at low Reynolds number”. In: *American Journal of Physics* 45.1 (Jan. 1977), pp. 3–11. ISSN: 0002-9505. DOI: 10.1119/1.10903. URL: http://aapt.scitation.org/doi/10.1119/1.10903.

[92] Ali Najafi and Ramin Golestanian. “Simple swimmer at low Reynolds number: Three linked spheres”. In: *Physical Review E* 69.6 (June 2004), p. 062901. ISSN: 1539-3755. DOI: 10.1103/PhysRevE.69.062901. URL: https://link.aps.org/doi/10.1103/PhysRevE.69.062901.

[93] J E Avron, O Kenneth, and D H Oaknin. “Pushmepullyou: an efficient micro-swimmer”. In: *New Journal of Physics* 7 (Nov. 2005), pp. 234–234. ISSN: 1367-2630. DOI: 10.1088/1367-2630/7/1/234. URL: https://iopscience.iop.org/article/10.1088/1367-2630/7/1/234.

[94] Eric Lauga and Thomas R Powers. “The hydrodynamics of swimming microorganisms”. In: *Reports on Progress in Physics* 72.9 (Sept. 2009), p. 096601. ISSN: 0034-4885. DOI: 10.1088/0034-4885/72/9/096601. URL: https://iopscience.iop.org/article/10.1088/0034-4885/72/9/096601.

[95] Eric Lauga. “Life around the scallop theorem”. In: *Soft Matter* 7.7 (2011), pp. 3060–3065. ISSN: 1744-683X. DOI: 10.1039/C0SM00953A. URL: http://xlink.rsc.org/?DOI=C0SM00953A.

[96] Tian Qiu et al. “Swimming by reciprocal motion at low Reynolds number”. In: *Nature Communications* 5.1 (Dec. 2014), p. 5119. ISSN: 2041-1723. DOI: 10.1038/ncomms6119. URL: http://www.nature.com/articles/ncomms6119.

[97] Li-Shi Luo et al. “Numerics of the lattice Boltzmann method: Effects of collision models on the lattice Boltzmann simulations”. In: *Physical Review E* 83.5 (May 2011), p. 056710. ISSN: 1539-3755. DOI: 10.1103/PhysRevE.83.056710. URL: https://link.aps.org/doi/10.1103/PhysRevE.83.056710.

[98] Jean-Christophe Desplat, Ignacio Pagonabarraga, and Peter Bladon. “LUDWIG: A parallel Lattice-Boltzmann code for complex fluids”. In: *Computer Physics Communications* 134.3 (Mar. 2001), pp. 273–290. ISSN: 00104655. DOI: 10.1016/S0010-4655(00)00205-8. URL: https://linkinghub.elsevier.com/retrieve/pii/S0010465500002058.

[99] VIVIEN M. KENDON et al. “Inertial effects in three-dimensional spinodal decomposition of a symmetric binary fluid mixture: a lattice Boltzmann study”. In: *Journal of Fluid Mechanics* 440 (Aug. 2001), pp. 147–203. ISSN: 0022-1120. DOI: 10.1017/S0022112001004682. URL: https://www.cambridge.org/core/product/identifier/S0022112001004682/type/journal%7B%5C_%7Darticle.

[100] John W. Cahn and John E. Hilliard. “Free Energy of a Nonuniform System. I. Interfacial Free Energy”. In: *The Journal of Chemical Physics* 28.2 (Feb. 1958), pp. 258–267. ISSN: 0021-9606. DOI: 10.1063/1.1744102. URL: http://aip.scitation.org/doi/10.1063/1.1744102.

[101] Anthony J. C. Ladd. “Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation”. In: *Journal of Fluid Mechanics* 271 (July 1994), pp. 285–309. ISSN: 0022-1120. DOI: 10.1017/S0022112094001771. URL: https://www.cambridge.org/core/product/identifier/S0022112094001771/type/journal%7B%5C_%7Darticle.

[102] Robert Brown. “XXVII. A brief account of microscopical observations made in the months of June, July and August 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies”. In: *The Philosophical Magazine* 4.21 (Sept. 1828), pp. 161–173. ISSN: 1941-5850. DOI: 10.1080/14786442808674769. URL: https://www.tandfonline.com/doi/full/10.1080/14786442808674769.

[103] A. Einstein. “Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen”. In: *Annalen der Physik* 322.8 (1905), pp. 549–560. ISSN: 00033804. DOI: 10.1002/andp.19053220806. URL: http://doi.wiley.com/10.1002/andp.19053220806.

[104] J Perrin and D L Hammick. *Atoms*. D. Van Nostrand Company, 1916. URL: https://archive.org/details/atoms00perrgoog/page/n8/mode/2up.

[105] Don S. Lemons and Anthony Gythiel. “Paul Langevin’s 1908 paper “On the Theory of Brownian Motion” [“Sur la théorie du mouvement brownien,” C. R. Acad. Sci. (Paris) 146 , 530–533 (1908)]”. In: *American Journal of Physics* 65.11 (Nov. 1997), pp. 1079–1081. ISSN: 0002-9505. DOI: 10.1119/1.18725. URL: http://aapt.scitation.org/doi/10.1119/1.18725.

[106] B. Abécassis et al. “Boosting migration of large particles by solute contrasts”. In: *Nature Materials* 7.10 (Oct. 2008), pp. 785–789. ISSN: 1476-1122. DOI: 10.1038/nmat2254. URL: http://www.nature.com/articles/nmat2254.

[107] Stephen Ebbens et al. “Size dependence of the propulsion velocity for catalytic Janus-sphere swimmers”. In: *Physical Review E* 85.2 (Feb. 2012), p. 020401. ISSN: 1539-3755. DOI: 10.1103/PhysRevE.85.020401. URL: https://link.aps.org/doi/10.1103/PhysRevE.85.020401.

[108] W C K Poon. “From Clarkia to Escherichia and Janus: the physics of natural and synthetic active colloids”. In: (2013). ISSN: 0074784X. DOI: 10.3254/978-1-61499-278-3-317. arXiv: 1306.4799. URL: http://arxiv.org/abs/1306.4799.

[109] Joost de Graaf, Georg Rempfer, and Christian Holm. “Diffusiophoretic Self-Propulsion for Partially Catalytic Spherical Colloids”. In: *IEEE Transactions on NanoBioscience* 14.3 (Apr. 2015), pp. 272–288. ISSN: 1536-1241. DOI: 10.1109/TNB.2015.2403255. URL: http://ieeexplore.ieee.org/document/7052389/.

[110] Aidan Brown and Wilson Poon. “Ionic effects in self-propelled Pt-coated Janus swimmers”. In: *Soft Matter* 10.22 (2014), pp. 4016–4027. ISSN: 1744-683X. DOI: 10.1039/C4SM00340C. URL: http://xlink.rsc.org/?DOI=C4SM00340C.

[111] Jeffrey L. Moran and Jonathan D. Posner. “Phoretic Self-Propulsion”. In: *Annual Review of Fluid Mechanics* 49.1 (Jan. 2017), pp. 511–540. ISSN: 0066-4189. DOI: 10.1146/annurev-fluid-122414-034456. URL: http://www.annualreviews.org/doi/10.1146/annurev-fluid-122414-034456.

[112] Andrea Scagliarini and Ignacio Pagonabarraga. “Unravelling the role of phoretic and hydrodynamic interactions in active colloidal suspensions”. In: *Soft Matter* 16.38 (2020), pp. 8893–8903. ISSN: 1744-683X. DOI: 10.1039/C8SM01831F. URL: http://xlink.rsc.org/?DOI=C8SM01831F.

[113] Kevin Stratford and Ignacio Pagonabarraga. “Parallel simulation of particle suspensions with the lattice Boltzmann method”. In: *Computers & Mathematics with Applications* 55.7 (Apr. 2008), pp. 1585–1593. ISSN: 08981221. DOI: 10.1016/j.camwa.2007.08.018. URL: https://linkinghub.elsevier.com/retrieve/pii/S089812210700644X.

[114] Anthony J. C. Ladd. “Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation”. In: *Journal of Fluid Mechanics* 271 (July 1994), pp. 285–309. ISSN: 0022-1120. DOI: 10.1017/S0022112094001771. URL: https://www.cambridge.org/core/product/identifier/S0022112094001771/type/journal%7B%5C_%7Darticle.

[115] A J C Ladd and R Verberg. “Lattice-Boltzmann Simulations of Particle-Fluid Suspensions”. In: *Journal of Statistical Physics* 104.5 (2001), pp. 1191–1251. ISSN: 1572-9613. DOI: 10.1023/A:1010414013942. URL: https://doi.org/10.1023/A:1010414013942.

[116] Ramin Golestanian, Tanniemola B. Liverpool, and Armand Ajdari. “Propulsion of a Molecular Machine by Asymmetric Distribution of Reaction Products”. In: *Physical Review Letters* 94.22 (June 2005), p. 220801. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.94.220801. URL: https://link.aps.org/doi/10.1103/PhysRevLett.94.220801.

[117] R Golestanian, T B Liverpool, and A Ajdari. “Designing phoretic micro- and nano-swimmers”. In: *New Journal of Physics* 9.5 (May 2007), pp. 126–126. ISSN: 1367-2630. DOI: 10.1088/1367-2630/9/5/126. URL: https://iopscience.iop.org/article/10.1088/1367-2630/9/5/126.

[118] M. Popescu et al. “Phoretic motion of spheroidal particles due to self-generated solute gradients ”. In: *European Physical Journal E: Soft Matter and Biological Physics* 31.4 (2010), pp. 351–367. ISSN: 1292-895X. DOI: 10.1140/epje/i2010-10593-3. URL: https://doi.org/10.1140/epje/i2010-10593-3.

[119] S. Ebbens et al. “Electrokinetic effects in catalytic platinum-insulator Janus swimmers”. In: *EPL (Europhysics Letters)* 106.5 (June 2014), p. 58003. ISSN: 0295-5075. DOI: 10.1209/0295-5075/106/58003. URL: https://iopscience.iop.org/article/10.1209/0295-5075/106/58003.

[120] Tugba Andac et al. “Active matter alters the growth dynamics of coffee rings”. In: *Soft Matter* 15.7 (2019), pp. 1488–1496. ISSN: 1744-683X. DOI: 10.1039/C8SM01350K. URL: http://xlink.rsc.org/?DOI=C8SM01350K.

[121] Rafael Mestre et al. “Extraction of the propulsive speed of catalytic nano- and micro-motors under different motion dynamics”. In: (July 2020). arXiv: 2007.15316. URL: http://arxiv.org/abs/2007.15316.

[122] Tania Patiño et al. “Fundamental Aspects of Enzyme-Powered Micro- and Nanoswimmers”. In: *Accounts of Chemical Research* 51.11 (Nov. 2018), pp. 2662–2671. ISSN: 0001-4842. DOI: 10.1021/acs.accounts.8b00288. URL: https://pubs.acs.org/doi/10.1021/acs.accounts.8b00288.

[123] Ernst Simonson and Josef Brozek. “Flicker Fusion Frequency: Background and Applications”. In: *Physiological Reviews* 32.3 (July 1952), pp. 349–378. ISSN: 0031-9333. DOI: 10.1152/physrev.1952.32.3.349. URL: https://www.physiology.org/doi/10.1152/physrev.1952.32.3.349.

[124] Diana Umeton, Jenny C. A. Read, and Candy Rowe. “Unravelling the illusion of flicker fusion”. In: *Biology Letters* 13.2 (Feb. 2017), p. 20160831. ISSN: 1744-9561. DOI: 10.1098/rsbl.2016.0831. URL: https://royalsocietypublishing.org/doi/10.1098/rsbl.2016.0831.

[125] Marco De Corato et al. “Self-Propulsion of Active Colloids via Ion Release: Theory and Experiments”. In: *Physical Review Letters* 124.10 (Mar. 2020), p. 108001. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.124.108001. URL: https://link.aps.org/doi/10.1103/PhysRevLett.124.108001.

[126] Felix Kümmel et al. “Circular Motion of Asymmetric Self-Propelling Particles”. In: *Physical Review Letters* 110.19 (May 2013), p. 198302. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.110.198302. URL: https://link.aps.org/doi/10.1103/PhysRevLett.110.198302.

[127] Stephen Ebbens et al. “Self-assembled autonomous runners and tumblers”. In: *Physical Review E* 82.1 (July 2010), p. 015304. ISSN: 1539-3755. DOI: 10.1103/PhysRevE.82.015304. URL: https://link.aps.org/doi/10.1103/PhysRevE.82.015304.

[128] John E. Greivenkamp. *Field Guide to Geometrical Optics*. Spie Press Book, 2004, p. 128. ISBN: 9780819452948.

[129] T. Darnige et al. “Lagrangian 3D tracking of fluorescent microscopic objects in motion”. In: *Review of Scientific Instruments* 88.5 (May 2017), p. 055106. ISSN: 0034-6748. DOI: 10.1063/1.4982820. URL: http://aip.scitation.org/doi/10.1063/1.4982820.

[130] Gary Dunderdale et al. “Importance of Particle Tracking and Calculating the Mean-Squared Displacement in Distinguishing Nanopropulsion from Other Processes”. In: *Langmuir* 28.30 (July 2012), pp. 10997–11006. ISSN: 0743-7463. DOI: 10.1021/la301370y. URL: https://pubs.acs.org/doi/10.1021/la301370y.

[131] Young-Moo Byun et al. “Distinguishing advective and powered motion in self-propelled colloids”. In: *Journal of Physics: Condensed Matter* 29.44 (Nov. 2017), p. 445101. ISSN: 0953-8984. DOI: 10.1088/1361-648X/aa88f1. URL: https://iopscience.iop.org/article/10.1088/1361-648X/aa88f1.

[132] Mingjun Xuan et al. “Noncontinuous Super-Diffusive Dynamics of a Light-Activated Nanobottle Motor”. In: *Angewandte Chemie International Edition* 57.23 (June 2018), pp. 6838–6842. ISSN: 14337851. DOI: 10.1002/anie.201801910. URL: http://doi.wiley.com/10.1002/anie.201801910.

[133] Leica. *LASX*. 2021. URL: https://www.leica-microsystems.com/products/microscope-software/.

[134] Johannes Schindelin et al. “Fiji: an open-source platform for biological-image analysis”. In: *Nature Methods* 9.7 (July 2012), pp. 676–682. ISSN: 1548-7091. DOI: 10.1038/nmeth.2019. URL: http://www.nature.com/articles/nmeth.2019.

[135] Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. “NIH Image to ImageJ: 25 years of image analysis”. In: *Nature Methods* 9.7 (July 2012), pp. 671–675. ISSN: 1548-7091. DOI: 10.1038/nmeth.2089. URL: http://www.nature.com/articles/nmeth.2089.

[136] Melissa Linkert et al. “Metadata matters: access to image data in the real world”. In: *Journal of Cell Biology* 189.5 (May 2010), pp. 777–782. ISSN: 1540-8140. DOI: 10.1083/jcb.201004104. URL: https://rupress.org/jcb/article/189/5/777/35828/Metadata-matters-access-to-image-data-in-the-real.

[137] FFmpeg Developers. *FFmpeg*. 2021. URL: http://ffmpeg.org/.

[138] Apache Software Foundation. *Apache Maven*. 2021. URL: https://maven.apache.org.

[139] Zhengxia Zou et al. “Object Detection in 20 Years: A Survey”. In: (May 2019). arXiv: 1905.05055. URL: http://arxiv.org/abs/1905.05055.

[140] Zahra Soleimanitaleb, Mohammad Ali Keyvanrad, and Ali Jafari. “Object Tracking Methods:A Review”. In: *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE, Oct. 2019, pp. 282–288. ISBN: 978-1-7281-5075-8. DOI: 10.1109/ICCKE48569.2019.8964761. URL: https://ieeexplore.ieee.org/document/8964761/.

[141] Ann Maria John et al. “A Review on Application of Fourier Transform in Image Restoration”. In: *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. IEEE, Oct. 2020, pp. 389–397. ISBN: 978-1-7281-5464-0. DOI: 10.1109/I-SMAC49090.2020.9243510. URL: https://ieeexplore.ieee.org/document/9243510/.

[142] Monali Chaudhari, Shanta Sondur, and Gauresh Vanjare. “A review on Face Detection and study of Viola Jones method”. In: *International Journal of Computer Trends and Technology* 25.1 (July 2015), pp. 54–61. ISSN: 22312803. DOI: 10.14445/22312803/IJCTT-V25P110. URL: http://www.ijcttjournal.org/archives/ijctt-v25p110.

[143] Chi Qin Lai and Soo Siang Teoh. “A review on pedestrian detection techniques based on Histogram of Oriented gradient feature”. In: *2014 IEEE Student Conference on Research and Development*. IEEE, Dec. 2014, pp. 1–6. ISBN: 978-1-4799-6428-4. DOI: 10.1109/SCORED.2014.7072948. URL: http://ieeexplore.ieee.org/document/7072948/.

[144] Allam Shehata Hassanein et al. “A Survey on Hough Transform, Theory, Techniques and Applications”. In: (Feb. 2015). arXiv: 1502.02160. URL: http://arxiv.org/abs/1502.02160.

[145] Albert Miguel López. “Development of a tracking code for active micro and nanomotors”. PhD thesis. Universitat Politècnica de Catalunya, 2015. URL: http://hdl.handle.net/2117/78805.

[146] Benjamin Midtvedt et al. “Quantitative digital microscopy with deep learning”. In: *Applied Physics Reviews* 8.1 (Mar. 2021), p. 011310. ISSN: 1931-9401. DOI: 10.1063/5.0034891. URL: https://aip.scitation.org/doi/10.1063/5.0034891.

[147] Frank Cichos et al. “Machine learning for active matter”. In: *Nature Machine Intelligence* 2.2 (Feb. 2020), pp. 94–103. ISSN: 2522-5839. DOI: 10.1038/s42256-020-0146-9. URL: http://www.nature.com/articles/s42256-020-0146-9.

[148] A.A. Chojaczyk et al. “Review and application of Artificial Neural Networks models in reliability analysis of steel structures”. In: *Structural Safety* 52 (Jan. 2015), pp. 78–89. ISSN: 01674730. DOI: 10.1016/j.strusafe.2014.09.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S016747301400085X.

[149] Yiming Jiang et al. “A Brief Review of Neural Networks Based Learning and Control and Their Applications for Robots”. In: *Complexity* 2017 (2017), pp. 1–14. ISSN: 1076-2787. DOI: 10.1155/2017/1895897. URL: https://www.hindawi.com/journals/complexity/2017/1895897/.

[150] AG Farizawani et al. “A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches”. In: *Journal of Physics: Conference Series* 1529 (Apr. 2020), p. 022040. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1529/2/022040. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1529/2/022040.

[151] F. Sultana, A. Sufian, and P. Dutta. “A Review of Object Detection Models Based on Convolutional Neural Network”. In: 2020, pp. 1–16. DOI: 10.1007/978-981-15-4288-6_1. URL: http://link.springer.com/10.1007/978-981-15-4288-6%7B%5C_%7D1.

[152] Dale Purves et al. *Neuroscience*. 2nd. Sunderland (MA): Sinauer Associates, 2001. ISBN: 978-0-87893-742-4.

[153] D. H. Hubel and T. N. Wiesel. “Receptive fields of single neurones in the cat’s striate cortex”. In: *The Journal of Physiology* 148.3 (Oct. 1959), pp. 574–591. ISSN: 00223751. DOI: 10.1113/jphysiol.1959.sp006308. URL: http://doi.wiley.com/10.1113/jphysiol.1959.sp006308.

[154] Kunihiko Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological Cybernetics* 36.4 (Apr. 1980), pp. 193–202. ISSN: 0340-1200. DOI: 10.1007/BF00344251. URL: http://link.springer.com/10.1007/BF00344251.

[155] Vincent Dumoulin and Francesco Visin. “A guide to convolution arithmetic for deep learning”. In: (Mar. 2016). arXiv: 1603.07285. URL: http://arxiv.org/abs/1603.07285.

[156] Ross Girshick et al. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2014, pp. 580–587. ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.81. URL: http://ieeexplore.ieee.org/document/6909475/.

[157] J. R. R. Uijlings et al. “Selective Search for Object Recognition”. In: *International Journal of Computer Vision* 104.2 (Sept. 2013), pp. 154–171. ISSN: 0920-5691. DOI: 10.1007/s11263-013-0620-5. URL: http://link.springer.com/10.1007/s11263-013-0620-5.

[158] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. “Efficient Graph-Based Image Segmentation”. In: *International Journal of Computer Vision* 59.2 (Sept. 2004), pp. 167–181. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000022288.19776.77. URL: http://link.springer.com/10.1023/B:VISI.0000022288.19776.77.

[159] Ross Girshick. “Fast R-CNN”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Dec. 2015, pp. 1440–1448. ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.169. URL: http://ieeexplore.ieee.org/document/7410526/.

[160] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (June 2017), pp. 1137–1149. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2577031. URL: http://ieeexplore.ieee.org/document/7485869/.

[161] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016, pp. 779–788. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.91. URL: http://ieeexplore.ieee.org/document/7780460/.

[162] Paul Henderson and Vittorio Ferrari. “End-to-end training of object class detectors for mean average precision”. In: (July 2016). arXiv: 1607.03476. URL: http://arxiv.org/abs/1607.03476.

[163] Lucas S. Palacios et al. “Guidance of active particles at liquid–liquid interfaces near surfaces”. In: *Soft Matter* 15.32 (2019), pp. 6581–6588. ISSN: 1744-683X. DOI: 10.1039/C9SM01016E. URL: http://xlink.rsc.org/?DOI=C9SM01016E.

[164] Savage Dwayne C. and Madilyn Fletcher, eds. *Bacterial Adhesion*. Springer, Boston, MA, 1985. ISBN: 9781461565147. DOI: 10.1007/978-1-4615-6514-7.

[165] Eric Lauga et al. “Swimming in Circles: Motion of Bacteria near Solid Boundaries”. In: *Biophysical Journal* 90.2 (Jan. 2006), pp. 400–412. ISSN: 00063495. DOI: 10.1529/biophysj.105.069401. URL: https://linkinghub.elsevier.com/retrieve/pii/S0006349506722214.

[166] Ya-Wen Chang et al. “Biofilm formation in geometries with different surface curvature and oxygen availability”. In: *New Journal of Physics* 17.3 (2015), p. 033017. ISSN: 1367-2630. DOI: 10.1088/1367-2630/17/3/033017. URL: http://stacks.iop.org/1367-2630/17/i=3/a=033017?key=crossref.4ca35e38b4d8e9054c56607952a3a96b

[167] P A Rühs et al. “Studying bacterial hydrophobicity and biofilm formation at liquid–liquid interfaces through interfacial rheology and pendant drop tensiometry”. In: *Colloids and Surfaces B: Biointerfaces* 117 (2014), pp. 174–184. ISSN: 0927-7765. DOI: https://doi.org/10.1016/j.colsurfb.2014.02.023. URL: http://www.sciencedirect.com/science/article/pii/S0927776514000903.

[168] Diego Lopez and Eric Lauga. “Dynamics of swimming bacteria at complex interfaces”. In: *Physics of Fluids* 26.7 (July 2014), p. 71902. ISSN: 1070-6631. DOI: 10.1063/1.4887255. URL: https://doi.org/10.1063/1.4887255.

[169] David Davies. “Understanding biofilm resistance to antibacterial agents”. In: *Nature Reviews Drug Discovery* 2 (Feb. 2003), p. 114. URL: http://dx.doi.org/10.1038/nrd1008%20http://10.0.4.14/nrd1008.

[170] Jeroen S Dickschat. “Quorum sensing and bacterial biofilms”. In: *Natural Product Reports* 27.3 (2010), pp. 343–369. ISSN: 0265-0568. DOI: 10.1039/B804469B. URL: http://dx.doi.org/10.1039/B804469B.

[171] Molin Søren and Tolker-Nielsen Tim. “Gene transfer occurs with enhanced efficiency in biofilms and induces enhanced stabilisation of the biofilm structure”. In: *Current Opinion in Biotechnology* 14.3 (2003), pp. 255–261. DOI: 10.1016/S0958-1669(03)00036-3.

[172] S L Percival and J T Walker. “Potable water and biofilms: A review of the public health implications”. In: *Biofouling* 14.2 (Sept. 1999), pp. 99–115. ISSN: 0892-7014. DOI: 10.1080/08927019909378402. URL: https://doi.org/10.1080/08927019909378402.

[173] Sokunrotanak Srey, Iqbal Kabir Jahid, and Sang-Do Ha. “Biofilm formation in food industries: A food safety concern”. In: *Food Control* 31.2 (2013), pp. 572–585. ISSN: 0956-7135. DOI: https://doi.org/10.1016/j.foodcont.2012.12.001. URL: http://www.sciencedirect.com/science/article/pii/S0956713512006536.

[174] José L Balcázar, Jéssica Subirats, and Carles M Borrego. “The role of biofilms as environmental reservoirs of antibiotic resistance”. In: *Frontiers in Microbiology* 6 (2015), p. 1216. ISSN: 1664-302X. DOI: 10.3389/fmicb.2015.01216. URL: https://www.frontiersin.org/article/10.3389/fmicb.2015.01216.

[175] Eric Lauga. “Bacterial Hydrodynamics”. In: *Annual Review of Fluid Mechanics* 48.1 (Jan. 2016), pp. 105–130. ISSN: 0066-4189. DOI: 10.1146/annurev-fluid-122414-034606. URL: https://doi.org/10.1146/annurev-fluid-122414-034606.

[176] Junghyun Kim et al. “Hydrodynamic effects on bacterial biofilm development in a microfluidic environment”. In: *Lab on a Chip* 13.10 (2013), pp. 1846–1849. ISSN: 1473-0197. DOI: 10.1039/C3LC40802G. URL: http://dx.doi.org/10.1039/C3LC40802G.

[177] W E Uspal et al. “Self-propulsion of a catalytically active particle near a planar wall: from reflection to sliding and hovering”. In: *Soft Matter* 11.3 (2015), pp. 434–438. ISSN: 1744-683X. DOI: 10.1039/C4SM02317J. URL: http://dx.doi.org/10.1039/C4SM02317J.

[178] Jemish Parmar et al. “Micro- and Nanomotors as Active Environmental Microcleaners and Sensors”. In: *Journal of the American Chemical Society* 140.30 (Aug. 2018), pp. 9317–9331. ISSN: 0002-7863. DOI: 10.1021/jacs.8b05762. URL: https://doi.org/10.1021/jacs.8b05762.

[179] Xiaolu Wang et al. *Wetting and orientation of catalytic Janus colloids at the surface of water*. Vol. 191. Apr. 2016. DOI: 10.1039/C6FD00025H.

[180] Priyanka Sharan et al. “Study of Active Janus Particles in the Presence of an Engineered Oil–Water Interface”. In: *Langmuir* 37.1 (Jan. 2021), pp. 204–210. ISSN: 0743-7463. DOI: 10.1021/acs.langmuir.0c02752. URL: https://pubs.acs.org/doi/10.1021/acs.langmuir.0c02752.

[181] “Active colloidal particles at fluid-fluid interfaces”. In: *Current Opinion in Colloid & Interface Science* (2017). URL: http://dx.doi.org/10.1016/j.cocis.2017.10.001.

[182] Alvaro Domínguez, Martin Oettel, and S. Dietrich. “Dynamics of colloidal particles with capillary interactions”. In: *Physical Review E* 82.1 (July 2010), p. 011402. ISSN: 1539-3755. DOI: 10.1103/PhysRevE.82.011402. URL: https://link.aps.org/doi/10.1103/PhysRevE.82.011402.

[183] Giorgio Volpe, Sylvain Gigan, and Giovanni Volpe. “Simulation of the active Brownian motion of a microswimmer”. In: *American Journal of Physics* 82.7 (June 2014), pp. 659–664. ISSN: 0002-9505. DOI: 10.1119/1.4870398. URL: https://doi.org/10.1119/1.4870398.

[184] Nikhil Desai, Vaseem A. Shaik, and Arezoo M. Ardekani. “Hydrodynamics-mediated trapping of micro-swimmers near drops”. In: *Soft Matter* 14.2 (2018), pp. 264–278. ISSN: 1744-683X. DOI: 10.1039/C7SM01615H. URL: http://xlink.rsc.org/?DOI=C7SM01615H.

[185] Saverio E Spagnolie et al. “Geometric capture and escape of a microswimmer colliding with an obstacle”. In: *Soft Matter* 11.17 (2015), pp. 3396–3411. DOI: 10.1039/C4SM02785J. URL: http://dx.doi.org/10.1039/C4SM02785J.

[186] Juliane Simmchen et al. “Effect of surfactants on the performance of tubular and spherical micromotors – a comparative study”. In: *RSC Adv.* 4.39 (2014), pp. 20334–20340. ISSN: 2046-2069. DOI: 10.1039/C4RA02202E. URL: http://xlink.rsc.org/?DOI=C4RA02202E.

[187] R. Benzi, S. Succi, and M. Vergassola. “The lattice Boltzmann equation: theory and applications”. In: *Physics Reports* 222.3 (Dec. 1992), pp. 145–197. ISSN: 03701573. DOI: 10.1016/0370-1573(92)90090-M. URL: https://linkinghub.elsevier.com/retrieve/pii/037015739290090M.

[188] D. Wolf-Gladrow. *Lattice-gas cellular automata and lattice Boltzmann models: an introduction*. Springer, 2000.

[189] Michael R. Swift et al. “Lattice Boltzmann simulations of liquid-gas and binary fluid systems”. In: *Physical Review E* 54.5 (Nov. 1996), pp. 5041–5052. ISSN: 1063-651X. DOI: 10.1103/PhysRevE.54.5041. URL: https://link.aps.org/doi/10.1103/PhysRevE.54.5041.

[190] Anthony J. C. Ladd. “Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 2. Numerical results”. In: *Journal of Fluid Mechanics* 271 (July 1994), pp. 311–339. ISSN: 0022-1120. DOI: 10.1017/S0022112094001783. URL: https://www.cambridge.org/core/product/identifier/S0022112094001783/type/journal%7B%5C_%7Darticle.

[191] N.-Q. Nguyen and A. J. C. Ladd. “Lubrication corrections for lattice-Boltzmann simulations of particle suspensions”. In: *Physical Review E* 66.4 (Oct. 2002), p. 046708. ISSN: 1063-651X. DOI: 10.1103/PhysRevE.66.046708. URL: https://link.aps.org/doi/10.1103/PhysRevE.66.046708.

[192] CYRUS K. AIDUN, YANNAN LU, and E.-JIANG DING. “Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation”. In: *Journal of Fluid Mechanics* 373 (Oct. 1998), pp. 287–311. ISSN: 0022-1120. DOI: 10.1017/S0022112098002493. URL: https://www.cambridge.org/core/product/identifier/S002211209800249type/journal%7B%5C_%7Darticle.

[193] J L Anderson. “Colloid Transport by Interfacial Forces”. In: *Annual Review of Fluid Mechanics* 21.1 (Jan. 1989), pp. 61–99. ISSN: 0066-4189. DOI: 10.1146/annurev.fl.21.010189.000425. URL: http://www.annualreviews.org/doi/10.1146/annurev.fl.21.010189.000425.

[194] Clemens Bechinger et al. “Active Particles in Complex and Crowded Environments”. In: *Reviews of Modern Physics* 88.4 (Nov. 2016), pp. 1–50. URL: https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.88.045006.

[195] Wei Wang et al. “From one to many: Dynamic assembly and collective behavior of self-propelled colloidal motors”. In: *Accounts of chemical research* 48.7 (2015), pp. 1938–1946. DOI: 10.1021/acs.accounts.5b00025. URL: https://pubs.acs.org/doi/abs/10.1021/acs.accounts.5b00025.

[196] Gerhard Gompper et al. “The 2020 motile active matter roadmap”. In: *Journal of Physics: Condensed Matter* 32.19 (2020), p. 193001. DOI: 10.1088/1361-648x/ab6348. URL: https://iopscience.iop.org/article/10.1088/1361-648X/ab6348/meta.

[197] Ching Hua Lee, Linhu Li, and Jiangbin Gong. “Hybrid Higher-Order Skin-Topological Modes in Nonreciprocal Systems”. In: *Phys. Rev. Lett.* 123 (1 July 2019), p. 016805. DOI: 10.1103/PhysRevLett.123.016805. URL: https://link.aps.org/doi/10.1103/PhysRevLett.123.016805.

[198] Yuhao Ma and Taylor L. Hughes. “The Quantum Skin Hall Effect”. In: *arXiv e-prints*, arXiv:2008.02284 (Aug. 2020), arXiv:2008.02284. arXiv: 2008.02284 [cond-mat.mes-hall].

[199] Ryo Okugawa, Ryo Takahashi, and Kazuki Yokomizo. “Second-order topological non-Hermitian skin effects”. In: *arXiv e-prints*, arXiv:2008.03721 (Aug. 2020), arXiv:2008.03721. arXiv: 2008.03721 [cond-mat.mes-hall].

[200] Kohei Kawabata, Masatoshi Sato, and Ken Shiozaki. “Higher-order non-Hermitian skin effect”. In: *Phys. Rev. B* 102 (20 Nov. 2020), p. 205118. DOI: 10.1103/PhysRevB.102.205118. URL: https://link.aps.org/doi/10.1103/PhysRevB.102.205118.

[201] Yongxu Fu and Shaolong Wan. “Non-Hermitian Second-Order Skin and Topological Modes”. In: *arXiv e-prints*, arXiv:2008.09033 (Aug. 2020), arXiv:2008.09033. arXiv: 2008.09033 [cond-mat.mes-hall].

[202] Linhu Li, Ching Hua Lee, and Jiangbin Gong. “Topological Switch for Non-Hermitian Skin Effect in Cold-Atom Systems with Loss”. In: *Physical Review Letters* 124.25 (June 2020), p. 250402.

[203] V. M. Martinez Alvarez, J. E. Barrios Vargas, and L. E. F. Foa Torres. “Non-Hermitian robust edge states in one dimension: Anomalous localization and eigenspace condensation at exceptional points”. In: *Phys. Rev. B* 97 (12 Mar. 2018), p. 121401. DOI: 10.1103/PhysRevB.97.121401. URL: https://link.aps.org/doi/10.1103/PhysRevB.97.121401.

[204] Ye Xiong. “Why does bulk boundary correspondence fail in some non-hermitian topological models”. In: *Journal of Physics Communications* 2.3 (Mar. 2018), p. 035043. DOI: 10.1088/2399-6528/aab64a. URL: https://doi.org/10.1088%2F2399-6528%2Faab64a.

[205] Flore K. Kunst et al. “Biorthogonal Bulk-Boundary Correspondence in Non-Hermitian Systems”. In: *Phys. Rev. Lett.* 121 (2 July 2018), p. 026808. DOI: 10.1103/PhysRevLett.121.026808. URL: https://link.aps.org/doi/10.1103/PhysRevLett.121.026808.

[206] V. M. Martinez Alvarez et al. “Topological states of non-Hermitian systems”. In: *European Physical Journal Special Topics* 227.12 (Dec. 2018). URL: https://link.springer.com/article/10.1140%2Fepjst%2Fe2018-800091-5.

[207] Shunyu Yao and Zhong Wang. “Edge States and Topological Invariants of Non-Hermitian Systems”. In: *Phys. Rev. Lett.* 121 (8 Aug. 2018), p. 086803. DOI: 10.1103/PhysRevLett.121.086803. URL: https://link.aps.org/doi/10.1103/PhysRevLett.121.086803.

[208] Ching Hua Lee and Ronny Thomale. “Anatomy of skin modes and topology in non-Hermitian systems”. In: *Phys. Rev. B* 99 (20 May 2019), p. 201103. DOI: 10.1103/PhysRevB.99.201103. URL: https://link.aps.org/doi/10.1103/PhysRevB.99.201103.

[209] Lucas S. Palacios et al. “Guided accumulation of active particles by topological design of a second-order skin effect”. In: *Nature Communications* 12.1 (Dec. 2021), p. 4691. ISSN: 2041-1723. DOI: 10.1038/s41467-021-24948-2. URL: https://www.nature.com/articles/s41467-021-24948-2.

[210] M. Z. Hasan and C. L. Kane. “Colloquium: Topological insulators”. In: *Rev. Mod. Phys.* 82 (4 Nov. 2010), pp. 3045–3067. DOI: 10.1103/RevModPhys.82.3045. URL: https://link.aps.org/doi/10.1103/RevModPhys.82.3045.

[211] Xiao-Liang Qi and Shou-Cheng Zhang. “Topological insulators and superconductors”. In: *Rev. Mod. Phys.* 83 (4 Oct. 2011), pp. 1057–1110. DOI: 10.1103/RevModPhys.83.1057. URL: https://link.aps.org/doi/10.1103/RevModPhys.83.1057.

[212] Tomoki Ozawa et al. “Topological photonics”. In: *Rev. Mod. Phys.* 91 (1 Mar. 2019), p. 015006. DOI: 10.1103/RevModPhys.91.015006. URL: https://link.aps.org/doi/10.1103/RevModPhys.91.015006.

[213] CL Kane and TC Lubensky. “Topological boundary modes in isostatic lattices”. In: *Nature Physics* 10.1 (2014), pp. 39–45. URL: https://www.nature.com/articles/nphys2835.

[214] Suraj Shankar et al. “Topological active matter”. In: *arXiv e-prints*, arXiv:2010.00364 (Oct. 2020), arXiv:2010.00364. arXiv: 2010.00364 [cond-mat.soft].

[215] Emil J. Bergholtz, Jan Carl Budich, and Flore K. Kunst. “Exceptional Topology of Non-Hermitian Systems”. In: *arXiv e-prints*, arXiv:1912.10048 (Dec. 2019), arXiv:1912.10048. arXiv: 1912.10048 [cond-mat.mes-hall].

[216] Luis E F Foa Torres. “Perspective on topological states of non-Hermitian lattices”. In: *Journal of Physics: Materials* 3.1 (Nov. 2019), p. 014002.

[217] Lei Xiao et al. “Non-Hermitian bulk–boundary correspondence in quantum dynamics”. In: *Nature Physics* 16.7 (Mar. 2020), pp. 761–766. URL: https://www.nature.com/articles/s41567-020-0836-6.

[218] Ananya Ghatak et al. “Observation of non-Hermitian topology and its bulk–edge correspondence in an active mechanical metamaterial”. In: *Proceedings of the National Academy of Sciences* 117.47 (2020), pp. 29561–29568. ISSN: 0027-8424. DOI: 10.1073/pnas.2010580117. eprint: https://www.pnas.org/content/117/47/29561.full.pdf. URL: https://www.pnas.org/content/117/47/29561.

[219] Martin Brandenbourger et al. “Non-reciprocal robotic metamaterials”. In: *Nature Communications* 10.1 (2019), p. 4608. URL: https://www.nature.com/articles/s41467-019-12599-3.

[220] Yangyang Chen et al. “Self-sensing metamaterials with odd micropolarity”. In: *arXiv e-prints*, arXiv:2009.07329 (Sept. 2020), arXiv:2009.07329. arXiv: 2009.07329 [physics.app-ph].

[221] Sebastian Weidemann et al. “Topological funneling of light”. In: *Science* 368.6488 (2020), pp. 311–314. ISSN: 0036-8075. DOI: 10.1126/science.aaz8727. URL: https://science.sciencemag.org/content/368/6488/311.

[222] T Helbig et al. “Generalized bulk–boundary correspondence in non-Hermitian topolectrical circuits”. In: *Nature Physics* 16.7 (July 2020), pp. 747–750. URL: https://www.nature.com/articles/s41567-020-0922-9.

[223] Tobias Hofmann et al. “Reciprocal skin effect and its realization in a topolectrical circuit”. In: *Phys. Rev. Research* 2 (2 June 2020), p. 023265. DOI: 10.1103/PhysRevResearch.2.023265. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.2.023265.

[224] Dan S Borgnia, Alex Jura Kruchkov, and Robert-Jan Slager. “Non-Hermitian Boundary Modes and Topology”. In: *Physical Review Letters* 124.5 (Feb. 2020), p. 056802.

[225] Nobuyuki Okuma et al. “Topological Origin of Non-Hermitian Skin Effects”. In: *Phys. Rev. Lett.* 124 (8 Feb. 2020), p. 086801. DOI: 10.1103/PhysRevLett.124.086801. URL: https://link.aps.org/doi/10.1103/PhysRevLett.124.086801.

[226] Kai Zhang, Zhesen Yang, and Chen Fang. “Correspondence between Winding Numbers and Skin Modes in Non-Hermitian Systems”. In: *Phys. Rev. Lett.* 125 (12 Sept. 2020), p. 126402. DOI: 10.1103/PhysRevLett.125.126402. URL: https://link.aps.org/doi/10.1103/PhysRevLett.125.126402.

[227] Yang Yu, Minwoo Jung, and Gennady Shvets. “Zero-energy corner states in a non-Hermitian quadrupole insulator”. In: *Phys. Rev. B* 103 (4 Jan. 2021), p. L041102. DOI: 10.1103/PhysRevB.103.L041102. URL: https://link.aps.org/doi/10.1103/PhysRevB.103.L041102.

[228] Elisabet Edvardsson, Flore K. Kunst, and Emil J. Bergholtz. “Non-Hermitian extensions of higher-order topological phases and their biorthogonal bulk-boundary correspondence”. In: *Phys. Rev. B* 99 (8 Feb. 2019), p. 081302. DOI: 10.1103/PhysRevB.99.081302. URL: https://link.aps.org/doi/10.1103/PhysRevB.99.081302.

[229] Tao Liu et al. “Second-Order Topological Phases in Non-Hermitian Systems”. In: *Phys. Rev. Lett.* 122 (7 Feb. 2019), p. 076801. DOI: 10.1103/PhysRevLett.122.076801. URL: https://link.aps.org/doi/10.1103/PhysRevLett.122.076801.

[230] Motohiko Ezawa. “Non-Hermitian boundary and interface states in nonreciprocal higher-order topological metals and electrical circuits”. English. In: *Physical Review B* 99.12 (Mar. 2019), p. 121411. DOI: 10.1103/PhysRevB.99.121411. URL: https://link.aps.org/doi/10.1103/PhysRevB.99.121411.

[231] Hong Wu, Bao-Qin Wang, and Jun-Hong An. "Floquet second-order topological insulators in non-Hermitian systems". In: *Phys. Rev. B* 103 (4 Jan. 2021), p. L041115. DOI: 10.1103/PhysRevB.103.L041115. URL: https://link.aps.org/doi/10.1103/PhysRevB.103.L041115.

[232] Jien Wu et al. "Observation of corner states in second-order topological electric circuits". In: *Phys. Rev. B* 102 (10 Sept. 2020), p. 104109. DOI: 10.1103/PhysRevB.102.104109. URL: https://link.aps.org/doi/10.1103/PhysRevB.102.104109.

[233] Kai Zhang, Zhesen Yang, and Chen Fang. "Universal non-Hermitian skin effect in two and higher dimensions". In: *arXiv e-prints*, arXiv:2102.05059 (Feb. 2021), arXiv:2102.05059. arXiv: 2102.05059 [cond-mat.mes-hall].

[234] Colin Scheibner, William T. M. Irvine, and Vincenzo Vitelli. "Non-Hermitian Band Topology and Skin Modes in Active Elastic Media". In: *Phys. Rev. Lett.* 125 (11 Sept. 2020), p. 118001. DOI: 10.1103/PhysRevLett.125.118001. URL: https://link.aps.org/doi/10.1103/PhysRevLett.125.118001.

[235] Anton Souslov et al. "Topological sound in active-liquid metamaterials". In: *Nature Physics* 13.11 (Nov. 2017), pp. 1091–1094. URL: https://www.nature.com/articles/nphys4193.

[236] Suraj Shankar, Mark J. Bowick, and M. Cristina Marchetti. "Topological Sound and Flocking on Curved Surfaces". In: *Phys. Rev. X* 7 (3 Sept. 2017), p. 031039. DOI: 10.1103/PhysRevX.7.031039. URL: https://link.aps.org/doi/10.1103/PhysRevX.7.031039.

[237] Anton Souslov et al. "Topological Waves in Fluids with Odd Viscosity". In: *Phys. Rev. Lett.* 122 (12 Mar. 2019), p. 128001. DOI: 10.1103/PhysRevLett.122.128001. URL: https://link.aps.org/doi/10.1103/PhysRevLett.122.128001.

[238] Kazuki Sone and Yuto Ashida. "Anomalous Topological Active Matter". In: *Phys. Rev. Lett.* 123 (20 Nov. 2019), p. 205502. DOI: 10.1103/PhysRevLett.123.205502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.123.205502.

[239] Lisa Yamauchi et al. "Chirality-driven edge flow and non-Hermitian topology in active nematic cells". In: *arXiv e-prints*, arXiv:2008.10852 (Aug. 2020), arXiv:2008.10852. arXiv: 2008.10852 [cond-mat.soft].

[240] Arvind Murugan and Suriyanarayanan Vaikuntanathan. "Topologically protected modes in non-equilibrium stochastic systems". In: *Nature Communications* 8.1 (Jan. 2017), pp. 1–6. URL: https://www.nature.com/articles/ncomms13881.

[241] Ching Hua Lee et al. "Topolectrical Circuits". In: *Communications Physics* 1.1 (2018), p. 39. URL: https://www.nature.com/articles/s42005-018-0035-2.

[242] Tobias Meng. “Coupled-wire constructions: a Luttinger liquid approach to topology”. In: *European Physical Journal Special Topics* 229.4 (Feb. 2020), pp. 527–543. URL: https://link.springer.com/article/10.1140/epjst/e2019-900095-5.

[243] Kinjal Dasbiswas, Kranthi K. Mandadapu, and Suriyanarayanan Vaikuntanathan. “Topological localization in out-of-equilibrium dissipative systems”. In: *Proceedings of the National Academy of Sciences* 115.39 (2018), E9031–E9040. ISSN: 0027-8424. DOI: 10.1073/pnas.1721096115. URL: https://www.pnas.org/content/115/39/E9031.

[244] Kohei Kawabata et al. “Symmetry and Topology in Non-Hermitian Physics”. In: *Phys. Rev. X* 9 (4 Oct. 2019), p. 041015. DOI: 10.1103/PhysRevX.9.041015. URL: https://link.aps.org/doi/10.1103/PhysRevX.9.041015.

[245] W. P. Su, J. R. Schrieffer, and A. J. Heeger. “Solitons in Polyacetylene”. In: *Phys. Rev. Lett.* 42 (25 June 1979), pp. 1698–1701. DOI: 10.1103/PhysRevLett.42.1698. URL: https://link.aps.org/doi/10.1103/PhysRevLett.42.1698.

[246] Ching-Kai Chiu et al. “Classification of topological quantum matter with symmetries”. In: *Rev. Mod. Phys.* 88 (3 Aug. 2016), p. 035005. DOI: 10.1103/RevModPhys.88.035005. URL: https://link.aps.org/doi/10.1103/RevModPhys.88.035005.

[247] Ananya Ghatak and Tanmoy Das. “New topological invariants in non-Hermitian systems”. In: *Journal of Physics: Condensed Matter* 31.26 (Apr. 2019), p. 263001. DOI: 10.1088/1361-648x/ab11b3. URL: https://doi.org/10.1088%2F1361-648x%2Fab11b3.

[248] Murray R. Spiegel and Seymour Lipschutz. *Schaum’s Outline of Vector Analysis*. McGraw-Hill, 2009. ISBN: 9780071615457.

[249] George Gabriel Stokes. “On the Effect of the Internal Friction of Fluids on the Motion of Pendulums”. In: *Mathematical and Physical Papers*. Cambridge: Cambridge University Press, pp. 1–10. DOI: 10.1017/CBO9780511702266.002. URL: https://www.cambridge.org/core/product/identifier/CBO9780511702266A006/type/book%7B%5C_%7Dpart.

[250] Crosbie W. Smith. “William Thomson and the creation of thermodynamics: 1840?1855”. In: *Archive for History of Exact Sciences* 16.3 (1977), pp. 231–288. ISSN: 0003-9519. DOI: 10.1007/BF00328156. URL: http://link.springer.com/10.1007/BF00328156.

[251] Roger Balian. “François Massieu and the thermodynamic potentials”. In: *Comptes Rendus Physique* 18.9-10 (Nov. 2017), pp. 526–530. ISSN: 16310705. DOI: 10.1016/j.crhy.2017.09.011. URL: https://linkinghub.elsevier.com/retrieve/pii/S1631070517300671.

[252] Paul A. Tipler and Gene Mosca. *Physics for Scientists and Engineers*. 6th. W. H. Freeman, 2007. ISBN: 9780716789642.

[253] John Roche. “What is potential energy?” In: *European Journal of Physics* 24.2 (Mar. 2003), pp. 185–196. ISSN: 0143-0807. DOI: 10.1088/0143-0807/24/2/359. URL: https://iopscience.iop.org/article/10.1088/0143-0807/24/2/359.

[254] Osamu Mishima and H. Eugene Stanley. “Decompression-induced melting of ice IV and the liquid–liquid transition in water”. In: *Nature* 392.6672 (Mar. 1998), pp. 164–168. ISSN: 0028-0836. DOI: 10.1038/32386. URL: http://www.nature.com/articles/32386.

[255] Vishwas V. Vasisht, Shibu Saw, and Srikanth Sastry. “Liquid–liquid critical point in supercooled silicon”. In: *Nature Physics* 7.7 (July 2011), pp. 549–553. ISSN: 1745-2473. DOI: 10.1038/nphys1993. URL: http://www.nature.com/articles/nphys1993.

[256] D. TER HAAR. “On the Theory of Phase Transitions”. In: *Men of Physics: L.D. Landau*. Elsevier, 1969, pp. 61–84. DOI: 10.1016/B978-0-08-006451-2.50013-3. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780080064512500133.

[257] Alan Chalmers. *The Scientist's Atom and the Philosopher's Stone*. Dordrecht: Springer Netherlands, 2009. ISBN: 978-90-481-2361-2. DOI: 10.1007/978-90-481-2362-9. URL: http://link.springer.com/10.1007/978-90-481-2362-9.

[258] LUDWIG BOLTZMANN. “Further Studies on the Thermal Equilibrium of Gas Molecules”. In: July 2003, pp. 262–349. DOI: 10.1142/9781848161337_0015. URL: http://www.worldscientific.com/doi/abs/10.1142/9781848161337%7B%5C_%7D0015.

[259] David Tong. *Kinetic Theory*. URL: http://www.damtp.cam.ac.uk/user/tong/kintheory/two.pdf (visited on 2021).

[260] P. L. Bhatnagar, E. P. Gross, and M. Krook. “A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems”. In: *Physical Review* 94.3 (May 1954), pp. 511–525. ISSN: 0031-899X. DOI: 10.1103/PhysRev.94.511. URL: https://link.aps.org/doi/10.1103/PhysRev.94.511.

[261] James D. Sterling and Shiyi Chen. “Stability Analysis of Lattice Boltzmann Methods”. In: *Journal of Computational Physics* 123.1 (Jan. 1996), pp. 196–206. ISSN: 00219991. DOI: 10.1006/jcph.1996.0016. URL: https://linkinghub.elsevier.com/retrieve/pii/S0021999196900169.

[262] Sauro Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Princeton U. Press, 2001, p. 288. ISBN: 0198503989.

[263] Sydney Chapman and T. G. Cowling. *The Mathematical Theory of Non-uniform Gases*. Cambridge University Press, 1991, p. 448. ISBN: 9780521408448.

[264] Shiyi Chen and Gary D. Doolen. “LATTICE BOLTZMANN METHOD FOR FLUID FLOWS”. In: *Annual Review of Fluid Mechanics* 30.1 (Jan. 1998), pp. 329–364. ISSN: 0066-4189. DOI: 10.1146/annurev.fluid.30.1.329. URL: http://www.annualreviews.org/doi/10.1146/annurev.fluid.30.1.329.

[265] Huidan Yu and Kaihua Zhao. “Lattice Boltzmann method for compressible flows with high Mach numbers”. In: *Physical Review E* 61.4 (Apr. 2000), pp. 3867–3870. ISSN: 1063-651X. DOI: 10.1103/PhysRevE.61.3867. URL: https://link.aps.org/doi/10.1103/PhysRevE.61.3867.

[266] Michael Keidar, Dayun Yan, and Jonathan H Sherman. *Plasma as a fourth state of matter*. 2019. DOI: 10.1088/2053-2571/aafb9cch1. URL: http://dx.doi.org/10.1088/2053-2571/aafb9cch1.

[267] T. A. Schaedler et al. “Ultralight Metallic Microlattices”. In: *Science* 334.6058 (Nov. 2011), pp. 962–965. ISSN: 0036-8075. DOI: 10.1126/science.1211649. URL: https://www.sciencemag.org/lookup/doi/10.1126/science.1211649.

[268] J. de Swaan Arons and G. A. M. Diepen. “Immiscibility of gases. The system He-Xe: (Short communication)”. In: *Recueil des Travaux Chimiques des Pays-Bas* 82.8 (Sept. 2010), pp. 806–806. ISSN: 01650513. DOI: 10.1002/recl.19630820810. URL: http://doi.wiley.com/10.1002/recl.19630820810.

[269] Katherine Villa et al. “Metal-Oxide-Based Microjets for the Simultaneous Removal of Organic Pollutants and Heavy Metals”. In: *ACS Applied Materials & Interfaces* 10.24 (June 2018), pp. 20478–20486. ISSN: 1944-8244. DOI: 10.1021/acsami.8b04353. URL: https://pubs.acs.org/doi/10.1021/acsami.8b04353.

[270] Ab Latif Wani, Anjum Ara, and Jawed Ahmad Usmani. “Lead toxicity: a review”. In: *Interdisciplinary Toxicology* 8.2 (June 2015), pp. 55–64. ISSN: 1337-9569. DOI: 10.1515/intox-2015-0009. URL: http://content.sciendo.com/view/journals/intox/8/2/article-p55.xml.

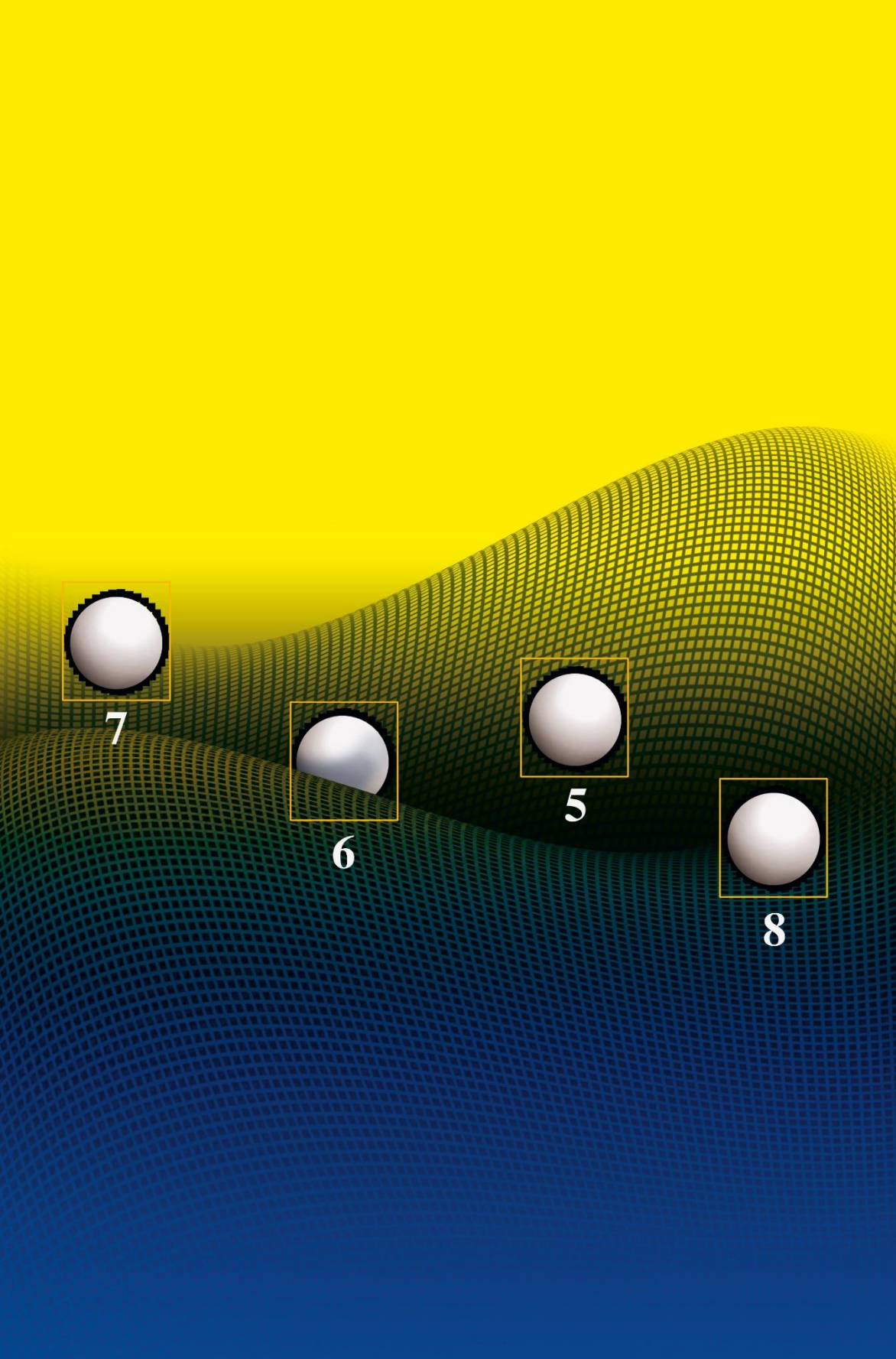
[271] Takanori Harada et al. “Toxicity and Carcinogenicity of Dichlorodiphenyltrichloroethane (DDT)”. In: *Toxicological Research* 32.1 (Jan. 2016), pp. 21–33. ISSN: 1976-8257. DOI: 10.5487/TR.2016.32.1.021. URL: http://www.toxicolres.org/journal/view.html?doi=10.5487/TR.2016.32.1.021.

[272] Eddy Metcalf. *Wastewater engineering: treatment and reuse*. 4th. McGraw-Hill, 2002, p. 1830. ISBN: 9780071122504.

[273] Craig R. McClain, Clifton Nunnally, and Mark C. Benfield. “Persistent and substantial impacts of the Deepwater Horizon oil spill on deep-sea megafauna”. In: *Royal Society Open Science* 6.8 (Aug. 2019), p. 191164. ISSN: 2054-5703. DOI: 10.1098/rsos.191164. URL: https://royalsocietypublishing.org/doi/10.1098/rsos.191164.

[274] OpenStax College. *Anatomy and Physiology*. 2021. URL: http://cnx.org/contents/14fb4ad7-39a1-4eee-ab6e-3ef2482e3e22@11.1..

[275] Peter Libby et al. “Atherosclerosis”. In: *Nature Reviews Disease Primers* 5.1 (Dec. 2019), p. 56. ISSN: 2056-676X. DOI: 10.1038/s41572-019-0106-z. URL: http://www.nature.com/articles/s41572-019-0106-z.


7
6
5
8