University of Trento
Department of Mathematics

Doctoral Programme in Mathematics
in agreement with the University of Verona

36th cycle

PhD thesis

Particles methods for kinetic equations in plasma physics, collective behaviors and optimization.

Candidate
Federica Ferrarese

Supervisor
Prof. Giacomo Albi

March 2024

Contents

Abstract

In recent years, the development of efficient numerical methods for the simulation of kinetic dynamics plays an important role. Besides deterministic methods, such as Lagrangian schemes, Galerkin schemes and finite volume schemes, an important attention is given to particles schemes. These methods are able to produce a good approximation of the density function, and, by introducing stochasticity, they are able to capture the natural property of the system such as randomness and uncertainties. They can also be suitable to solve the issue of high dimensionality related to deterministic schemes and to substantially reduce the related computational costs. In this thesis we will focus on the application of particles schemes to different problems, developing novel methods to improve the efficiency of the existent ones. In Chapter 2 we will focus on a particular problem related to magnetic confinement in Plasma Physics. We will introduce an optimal control problem and discretize it by means of a particular particle scheme. In Chapter 3 we will formulate a follower-leader kinetic model to simulate the collective motion of birds. We will show how it is possible to improve the efficiency of classical stochastic algorithms in presence of non-locality. In Chapter 4 we will consider a similar setting in the context of optimization and we will develop a novel stochastic algorithm useful to minimize non-convex high dimensional functions. Finally, in Chapter 5 we will focus on a predator-prey model, developing an efficient version of classical approximated stochastic algorithms which is able to preserve the accuracy.

Chapter 1

Introduction

The investigation of mathematical models that explain the collective behavior of groups of animals, such as birds flocks [14, 64, 65, 89], fish schools [127], and insect swarms [22, 24, 35, 139], has been a considerable surge in interest. Beyond its significance in biology, emerging collective behaviors have a substantial impact on a wide array of applications that encompass the dynamics of numerous individuals or particles. These applications span from computer science [140, 172], physics [118], and engineering [141] to the social sciences and economics [12, 86, 168, 185]. Numerous conventional models describe these behaviours starting from the individuals or particles level. Among the different frameworks proposed, the three-zone model [45, 190] has been particularly used. Particles engage in three types of interactions: repulsion, alignment, and attraction, depending on the relative location of their neighboring agents. This is the case for example of opinion dynamics [10, 109], where people are affected by others for instance in social networks or during political elections. Another example is the case of flocking of animals, such as for instance fish schools, where preys may organize in huge groups to escape from predators.

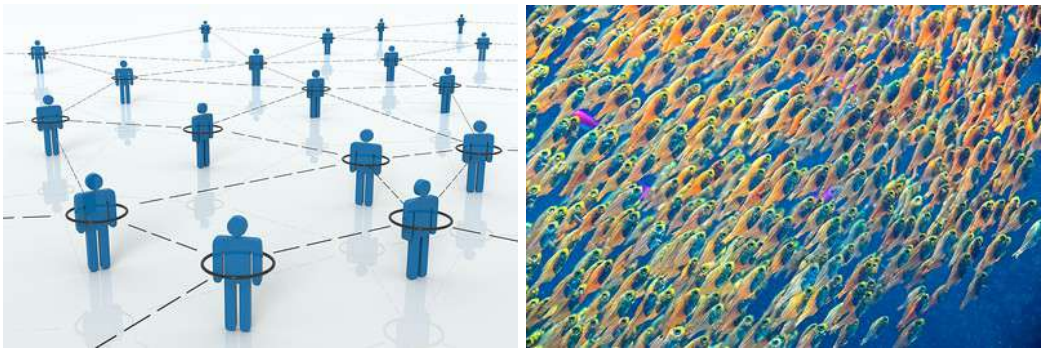

Figure 1.1: On the left a network diagram and on the right a fish school.

1.1 From ODEs to kinetic equations

From a mathematical view point, the dynamics at the individuals level can be described by microscopic models which take into account of the evolution of the position x_i and velocity

v_i of the i-th particle, for $i = 1, \dots, N$. The ODE system reads

$$\begin{cases} \dot{x}_i(t) = v_i, \\ \dot{v}_i(t) = \sum_{j=1}^{N} F_i(t, x_j(t), v_j(t)), \end{cases} \tag{1.1.1}$$

where $F_i(\cdot)$ models the interaction forces between each agent i and all the other agents. Among the classical models we recall the Cucker-Smale model [73, 74, 49] and the D'Orsogna-Bertozzi model [90] which assume symmetric interactions between all the agents. Some extensions which do not involve symmetric interactions are given for example by the Mostch-Tadin model [155], in which a weighted interaction force is considered. The weighted influence can be represented by a visual interaction cone [62, 94, 95], assuming agents can interact just with the ones inside their vision area, or by metric and topological balls [67, 31, 32, 123]. In metric interaction, each agent communicates just with the agents whose distance from it is lower than a certain threshold, while in topological interaction, each agent interacts just with its N^* nearest neighbors. Direct simulation of the ODE system can be extremely demanding, especially when the sample size is large and non-local interactions are considered. From a mathematical modeling perspective, these issues have been extensively explored within the kinetic research community [162, 50, 102, 123, 167, 185], with the development of kinetic equations to simplify the computational complexity. Kinetic equations are mathematical formulations that describe the behavior and evolution of systems with a large number of particles or constituents. These equations focus on understanding the evolution of the distribution of particles $f(\cdot)$ in phase space, considering factors such as position and velocity. In particular, each particle with position and velocity (x, v) is subjected to free transport, that is it change its position x with a position

$$x' \to x + (t - t_0)v, \tag{1.1.2}$$

where t_0 denotes the initial time and t the current time. In addition, each particle interacts with another randomly selected particle with velocity w and both particles change their velocity because of interaction according to the following *binary interactions rules*,

$$\begin{aligned} v^* &= v + \alpha \mathcal{F}(v, w), \\ w^* &= w + \alpha \mathcal{F}(w, v), \end{aligned} \tag{1.1.3}$$

where α represents the interaction propensity, i.e. the strength of interaction, and $\mathcal{F}(\cdot)$ is the interaction force. The evolution of the density function $f(\cdot)$ of particle position $x \in \mathbb{R}^d$ and velocity $v \in \mathbb{R}^d$ can be described by a Boltzmann type equation of the form

$$\partial_t f(x, v, t) + v \cdot \nabla_x f(x, v, t) = \mathcal{Q}(x, v, t). \tag{1.1.4}$$

In (1.1.4) the term

$$v \cdot \nabla_x f(x, v, t), \tag{1.1.5}$$

represents the effect of the free-transport of particles, while the term $\mathcal{Q}(f,f)$ represents the effect of binary collisions and takes the following form

$$\mathcal{Q}(x,v,t) = \int_{\mathbb{R}^{2d}} \frac{1}{J}(f(x,v^*,t)f(y,w^*,t) - f(x,v,t)f(y,w,t))dwdy \tag{1.1.6}$$

where (v^*, w^*) denotes the post interaction velocities of particles with velocity (v,w), $\mathcal{F}(\cdot)$ models the interaction force between the two agents, $\alpha > 0$, and J is the Jacobian of the transformation of (v,w) to (v^*, w^*).

1.2 Numerical methods for kinetic equations

From a numerical view point, the dynamics can be simulated by means of different numerical schemes such as the Lagrangian schemes [72, 175], the finite volume schemes [71, 194] and the Galerkin schemes [76, 151]. These methods produce an accurate approximation of the density but may be computationally expensive. Indeed, tackling the numerical approximations for the interaction term, especially in cases involving multi-dimensional integrals, proves to be very challenging. In addition, the kinetic dynamics is difficult to be approximated on regular grids since the long-time behaviour of the system is not compactly supported. In contrast to the mentioned numerical schemes we find particles methods which produce a good approximation of the density function with the advantage of reducing substantially the computational costs. Indeed, they can handle problems with a large number of dimensions without a significant increase in computational complexity which instead characterize the deterministic methods. Furthermore, by introducing stochasticity, they allow to formulate numerical schemes that retain the majority of the physical properties inherent in the original system such as for instance randomness and uncertainty. The idea of these methods is to approximate the density by a large number of particles with position and velocity $(x_i(t), v_i(t))$ for $i = 1, \dots, N$ as

$$f^N(x,v,t) = \frac{1}{N}\sum_{i=1}^{N} \delta(x - x_i(t))\delta(v - v_i(t)) \tag{1.2.1}$$

where $\delta(\cdot)$ is the usual Delta-function. At time $t = 0$ particles are sampled from an initial given distribution by generating a sequence of pseudo random numbers. To this aim one can refer for instance to inverse transport methods or an acceptance-rejection methods [162]. In the following we will present different particles methods.

1.2.1 Direct Simulation Monte Carlo (DSMC)

DSMC methods are an efficient class of numerical schemes used to simulate the probabilistic dynamics given by kinetic equations. Schemes of this type have been first developed by

Bird [28] and Nanbu [157, 158]. The method divides the simulation domain into computational cells and simulates the motion of particles through random sampling of collision and interaction events. This statistical approach allows DSMC to capture the probabilistic nature of collisions, however its accuracy strongly depend on the number of particles N, being of order $\mathcal{O}(N^{-1/2})$. In advance, due to its statistical nature, in low populated regions, simulations can exhibit a numerical noise which may affect the results. Despite these challenges, the DSMC is an indispensable tool to capture the individuals behaviour and to face the curse of dimensionality problem related to the computation of the interaction term. The fundamental idea of Monte Carlo methods is a decoupling between the transport and collision parts in the Boltzmann equation (1.1.4). The transport step solves the exact free flow of the sampled particles with position and velocity $(x_i(t), v_i(t))$ in a time interval Δt, for any $i = 1, \dots, N$

$$x_i(t) = x_i(t) + \Delta t v_i(t). \tag{1.2.2}$$

Then different variants of the Monte Carlo method can be used to solve the interaction step by giving a probabilistic reformulation of the dynamics described by (1.1.4).

Asymptotic Nanbu algorithm. The idea is to introduce the approximated density f^{n+1} at time $t^{n+1} = (n+1)\Delta t$ as

$$f^{n+1} = (1 + \Delta t)\, f^n + \Delta t \mathcal{Q}^+(f^n, f^n), \tag{1.2.3}$$

where $\mathcal{Q}^+(f^n, f^n)$ denotes the gain part of the discretized collision operator defined in (1.1.6). Equation (1.2.3) can be explained as follows. With a probability Δt particles interact and change their velocity, with probability $1 - \Delta t$ particles do not interact. At each time t the density is reconstructed as in (1.2.1). A simple algorithm for the interaction step is presented in the sequel.

Algorithm 1.2.1. [Asymptotic Nanbu algorithm]

1. Give N samples v_i^0 from the initial distribution f^0, set $N_t = T\Delta t$ to be the total number of iterations, where T represents the final time.

2. `for` $n = 0$ `to` N_t

   (a) `for` $i = 1$ `to` N

      i. select randomly an index $j \neq i$,

      ii. with probability Δt update the velocity v_i using the first relation in (1.1.3),

      iii. with probability $1 - \Delta t$ set $v_i^{n+1} = v_i^n$.

   `end for`

`end for`

There exists a symmetrized version of this algorithm in which once that two particles have been selected, with a certain probability both the velocities will be updated according to the rules described in (1.1.3).

Bird like algorithms. Another class of methods is the Bird like asymptotic schemes. The primary distinction from the previously outlined algorithm lies in the fact that particles are engaged in multiple interactions within a single time step. The method is based on the observation that binary interactions undergo at time intervals which are exponentially distributed. Hence, given a random number $r \sim U([0,1])$, we set

$$\Delta t_c = -\frac{ln(r)}{N}. \tag{1.2.4}$$

A simpler technique relies on a constant time step that corresponds to the average time between interactions and hence it is defined to be

$$\Delta t_x = \frac{\Delta t}{N_c} = \frac{1}{N} \tag{1.2.5}$$

where $N_c = N\Delta t$ is the average number of interactions per time step. The symmetrize version of this algorithm reads as follows.

Algorithm 1.2.2. [Symmetrize Bird like algorithm]

1. Give N samples v_i^0 from the initial distribution f^0, set $N_t = T\Delta t$ to be the total number of iterations, where T represents the final time, and $N_c = N\Delta t$.
2. `for` $n = 0$ `to` N_t
   (a) `for` $k = 0$ `to` N_c
      i. `for` $i = 1$ `to` N
         A. select randomly an index $j \neq i$,
         B. update the velocity v_i and v_j using the first and second relation in (1.1.3).
      `end for`
   `end for`

`end for`

Mean-field interaction algorithm. A method to directly simulate the Boltzmann equation is the mean-field Monte Carlo algorithm, which at the particle level is equivalent to microscopic simulations. At each step the idea is to select a random subset of particles with size M for interactions, as opposed to having each particle to interact with another randomly selected particle. The following algorithm briefly describes this procedure.

Algorithm 1.2.3. [Mean-field interaction algorithm]

1. Give N samples v_i^0 from the initial distribution f^0, set $N_t = T\Delta t$ to be the total number of iterations, where T is the final time, and set $M < N$ to be a subsample size.

2. `for` $n = 0$ `to` N_t

   (a) `for` $i = 1$ `to` N

      i. select randomly M indexes $j \neq i$,

      ii. update the velocity v_i as

$$v_i^{n+1} = v_i^n + \frac{\Delta t}{M} \sum_{j=1}^{M} \mathcal{F}(v_i^n, v_j^n). \tag{1.2.6}$$

   `end for`

`end for`

This algorithm allows to reduce the computational cost from $\mathcal{O}(N^2)$, which corresponds to the cost of a microscopic simulation, to $\mathcal{O}(NM)$. Note that if $M = 1$ we recover the Asymptotic Nanbu algorithm previously described. Other versions of the mean-field Monte Carlo algorithm can be derived by taking suitable averages in the algorithms for Boltzmann models. We refer to [162] for further the details.

1.2.2 Particle-in-Cell schemes (PIC)

Particle-in-Cell (PIC) methods represent a class of numerical techniques widely employed in the simulation of complex physical systems, particularly in the realm of plasma physics, astrophysics, and high-energy particle interactions. PIC methods originated in the 1950s and have since evolved significantly. The methods popularity stems from their ability to capture kinetic effects and self-consistent interactions, making them particularly well-suited for studying phenomena where the behavior of individual particles is crucial. These phenomena include also the study of magnetized plasma [54, 96, 97, 120] on which we are mainly interested in. In this context we assume the evolution of the plasma density function to be given by a particular collisionless kinetic equation, the Vlasov equation

$$\frac{\partial f(t,x,v)}{\partial t} + v \cdot \nabla_x f(t,x,v) + (\mathbf{E}(t,x) + v \times \mathbf{B}(t,x)) \cdot \nabla_v f(t,x,v) = 0, \tag{1.2.7}$$

with $\mathbf{E}(t,x)$ and $\mathbf{B}(t,x)$ representing the electric and external magnetic field respectively, which describes how the number of particles at a particular position and velocity in phase space change. The Vlasov equation can be coupled with the Poisson equation

$$\Delta_x \phi(x,t) = 1 - \rho(x,t), \qquad \mathbf{E}(x,t) = -\nabla_x \phi(x,t), \tag{1.2.8}$$

where $\phi(\cdot)$ represents the electric potential, $\rho(\cdot)$ the charge density and $\mathbf{E}(\cdot)$ the value of the electric field, which relates the electrostatic potential to the charge density, linking the dynamics of charged particles to the electric field they create. In the context of PIC

simulations for Vlasov-Poisson systems, the computational domain is discretized into a grid, and the evolution of the distribution function is tracked by representing it with a finite number of computational particles. These particles move through phase space along the characteristics of the system which are

$$\begin{aligned} \frac{dx_i(t)}{dt} &= v_i(t), \qquad x_i(0) = x_i^0, \\ \frac{dv_i(t)}{dt} &= v_i(t) \times \mathbf{B}(t, x_i(t)) + \mathbf{E}(t, x_m), \qquad v_i(0) = v_i^0, \end{aligned} \tag{1.2.9}$$

for any $i = 1, \dots, N$, with N representing the total number of particles. The approximated density is then computed from the particle distribution as in (1.2.1), and the Poisson equation is solved to obtain the electric field. There is also a modified version of the PIC method frequently employed when the physics under examination remains in proximity to an equilibrium state. For instance, this variant is commonly utilized in PIC simulations of tokamak plasmas or particle accelerators. This method is called the δf method [16]. It consists in expanding the distribution function in the neighborhood of a known equilibrium f_0 in $f = f_0 + \delta f$ and to approximate only the δf part with a PIC method. PIC methods are in general stable and easily parallelizable allowing the simulation with huge number of particles. They are also versatile and can be applied to a wide range of problems. As for DSMC, the presence of noise due to stochasticity may impact the accuracy of the simulation, especially in situations with low particles levels. In advance, we may pay attention in imposing proper boundary conditions. Incorrect boundary treatments may introduce artefacts into the simulation results. Another challenge consists in the particle-mesh coupling problem. Indeed, the computation of the approximated density $f^N(\cdot)$ as in (1.2.1) does not naturally give an expression for this function at all points of phase space. Thus for the coupling with the field solver, which is defined on the mesh, a regularizing step is necessary. To this aim we shall define convolution kernels which can be used for this regularization procedure, see [180] for a detailed description. In the following we will present different variants of the PIC methods.

First order semi-implicit scheme. The first simplest scheme that we might consider to simulate the dynamics given by (1.2.9) is a combination between the backward and forward Euler scheme. Given a fixed time step Δt, an electric field $\mathbf{E}(t, x)$ and external magnetic field $\mathbf{B}(t, x)$ we get

$$\begin{aligned} x_i^{n+1} &= x_i^n + \Delta t v_i^{n+1}, \\ v_i^{n+1} &= v_i^n + \Delta t v_i^{n+1} \times \mathbf{B}(t^n, x_i^n) + \Delta t \mathbf{E}(t^n, x_i^n), \end{aligned} \tag{1.2.10}$$

for any $i = 1, \dots, N$, and $n = 1, \dots, N_t$ with N_t representing the total number of iterations. This method is semi-implicit and hence it is stable for all the choices of Δt. Besides its simplicity in the implementation, it is a first order method, and hence it is not enough accurate to describe the long-term behavior of the solution. This approach can be generalized

to produce second and third order semi-implicit schemes using Runge-Kutta methods, see [97].

Boris scheme. It was originally presented by Boris in 1970, [121]. The algorithm reads as follows. Given a time step Δt, a fixed electric field $\mathbf{E}(t,x)$, an external magnetic field $\mathbf{B}(t,x)$ and the initial position and velocity (x_i^0, v_i^0) of each particle $i = 1, \dots, N$, we compute

$$v_i^{1/2} = v_i^0 + \frac{\Delta t}{2}\left(v_i^0 \times \mathbf{B}(t, x_i^0) + \mathbf{E}(t, x_i^0)\right). \tag{1.2.11}$$

Then, for any $n = 1, \dots, N_t$, with N_t total number of iterations we compute

$$\begin{aligned} x_i^{n+1} &= x_i^n + \Delta t v_i^{n+1/2}, \\ v_i^{n+1/2} &= v_i^{n-1/2} + \Delta t v_i^n \times \mathbf{B}(t^n, x_i^n) + \Delta t \mathbf{E}(t^n, x_i^n), \end{aligned} \tag{1.2.12}$$

where $v_i^{n+1} = 1/2(v_i^{n+1/2} + v_i^{n-1/2})$. The scheme has second order accuracy, it is numerically stable for any choice of Δt and it preserves the symplectic structure of Hamiltonian systems, which is beneficial for long-term accuracy in simulations. An alternative version of the Boris scheme has been presented by Jean-Luc Vay in 2008, [189].

Verlet scheme. The Verlet scheme takes the name by its creator, who developed it in the 1960s. The basic principle involves updating particle positions and velocities in two steps, using informations from the current and previous time steps. Given a time steps Δt, an electric field $\mathbf{E}(t,x)$ and an external magnetic field $\mathbf{B}(t,x)$ we get for any $n = 0, \dots, N_t$, with N_t the total number of iterations

$$\begin{aligned} v_i^{n+1/2} &= v_i^n + \frac{\Delta t}{2}\mathbf{F}(t, x_i^n, v_i^n), \\ x_i^{n+1} &= x_i^n + \Delta t v_i^{n+1/2}, \\ v_i^{n+1} &= v_i^{n+1/2} + \frac{\Delta t}{2}\mathbf{F}(t, x_i^{n+1}, v_i^{n+1}), \end{aligned} \tag{1.2.13}$$

for any $i = 1, \dots, N$, with

$$\mathbf{F}(t, x, v) = \mathbf{E}(t, x) + v \times \mathbf{B}(t, x).$$

It is a second order accurate symplectic scheme. However it may present some instabilities for large time step sizes or in presence of stiff potentials. Several variations and extensions of the basic Verlet scheme have been implemented, such as the velocity Verlet algorithm and the leapfrog Verlet algorithm, [119].

1.2.3 Stochastic algorithms for biochemical reactions

Another class of stochastic algorithms [146, 30, 88, 92] on which we might be interested in includes the methods used to simulate biochemical reactions [88, 126]. Inside cells molecules are constantly moving and colliding each other. The rate of collisions depends on the number of species and molecules involved in the process. Besides chemical reactions, this process can be observed also in interactions among individuals in biological systems, such as competition or predator preys models. The description at individual or stochastic level has been shown to be more realistic than the deterministic one. As mean-field models capture the mean behavior of the system, the random effect included in stochastic models lead the solution to oscillate around the equilibria and extinction may also occurs. For a formal mathematical description, the idea is to consider a biological system consisting of N species which interact according to specific reaction events. The aim of this class of stochastic algorithms is to simulate the occurrence of these reaction events. In the following we will present different versions of these algorithms, ranging from the so called "exact" algorithms to their approximated version.

Direct stochastic simulations. The Direct stochastic algorithm, also known as Gillespie algorithm [111, 114, 113, 112], simulates the time evolution of a system by considering individual reaction events and their associated rates. Each reaction is characterized by a propensity function representing the likelihood of that particular reaction occurring per unit time. The algorithm works as follows. At each time step, a reaction event is selected on the basis of its propensity. Then the state of the systems is updated accordingly to the reaction selected and the time steps size is tuned according to a random number generation technique. The process is repeated until the final time T is reached. Direct stochastic simulations provides an exact simulation, selecting which reaction must occur and at which time. However, they can be prohibitively expensive especially when the number of species is large. To face with this problem the idea is to consider approximated algorithms, such as the τ-leaping methods.

τ-leaping methods. Classical approximated algorithms, such as the τ-leaping methods, can speed up the simulations provided by the Gillespie algorithm by allowing multiple reactions to occur in the same time interval. However their efficiency and accuracy depends on the parameters choice [169, 176, 178, 125]. The algorithm works as follows. At each time step, which might also be considered to be fixed, the number of reactions that occur is sampled from a Poisson distribution. Then, the species counts are updated accordingly to the selected reactions events. The steps are repeated until the final time is reached.

Remark 1.
- Extensions and improvements of the τ-leaping methods are well exploited in [146].

- More details about the algorithms just mentioned and their implementation will be provided in Chapter 5.

1.3 Overview

The aim of this thesis is to study and propose novel particles schemes, taking inspiration from the ones described above, to simulate the dynamics in different contexts. We first focus on a particular PIC method, useful to simulate the evolution of the plasma density, and we design a control problem with the aim to let the plasma to assume a desiderated configuration, suppressing the instabilities that normally arise. Secondly, we focus on follower-leader dynamics which can be observed in different situations. We show how it is possible to improve the efficiency of a classical Asymptotic Nanbu algorithm in presence of non-locality by describing a model in which topological interactions plays an important role. Then, a similar setting is considered in the context of optimization. We develop a novel stochastic algorithm based on the Asymptotic Nanbu algorithm and which is useful to minimize non-convex high dimensional functions. Finally, we focus on a predator-prey model and we develop an efficient version of classical approximated stochastic algorithms which is able to preserve the accuracy. A more detailed description is given in the next sections.

1.3.1 Instantaneous control of plasma via an external magnetic field in a Vlasov-Poisson system

In Chapter 2, we focus on a model to describe the evolution of the plasma density in presence of strong magnetic field. We consider the Vlasov-Poisson equation (1.2.7)-(1.2.8). The study of magnetized plasma plays an important role in different physics situations such as for example in the problem of plasma confinement in huge devices as for instance the tokamak. Due to the presence of strong magnetic field different instabilities may arise leading to the formation of vortices. On the other hand, the magnetic field can be seen as a control aiming at leading the plasma density to assume a desiderated configuration. We will derive different instantaneous controls with the aim of reducing the mass which hits the boundaries in agreement with the results proposed in [138, 193, 15]. To simulate the dynamics we use a particular PIC method, the one introduced in [97], which is a semi-implicit scheme and hence has a huge stability area.

1.3.2 Follower-Leader dynamics

Secondly, we move to more intricate models which encompass the potential division of one population into two populations, namely the followers and leaders. The level of leadership may depend on different factors ranging from experience to prior knowledge. As an example of such a dynamics we can consider the case of the evacuation of a crowd of individuals

from an unknown environment [3, 8, 68, 7, 53, 19]. In this context, few informed agents, say the leaders, are introduced in the crowd to drive people, the followers, toward a safety region. It can be proven that by introducing leaders and by optimizing their strategies [66, 69] it is possible to enhance the evacuation time. As another example we may consider the dynamics at the cellular level where a group of cells, called leaders cells, can trigger a migration process [191, 177]. In breast cancer for instance the invasion of healthy tissues occurs due to the presence of particular cells, which can be distinguished from the other by their epithelial genes [60]. Follower-leader dynamics can be observed also in migrating animals groups. In honey bees swarms, only the so called "scout" bees know the final position of the new honeybee nest [22, 23, 27]. In fishes shoal, we observe the presence of "braver" or faster individual which are able to drive the whole group far away from a potential danger or toward a selected target [58].

Non-local kinetic dynamic with emergent leaders

In Chapter 3 we present a model to describe the collective motion of birds at the kinetic level. This model introduces spontaneous changes of direction within the swarm, independent of external factors but rather influenced by the interaction between two dynamics populations labelled as leaders and followers. The main idea is that each agent is potentially a turn initiator, becoming a leader whose influence acts on his nearest neighbors. The interest is on the phenomenon of transient leadership, which has been largely studied in [1, 12, 142, 143, 154]. At each time agents can change their label, according to a stochastic process. Each event is associate with a certain transition rate which, in general, is a non linear function of the state. The main novelty of this work consists in the study of efficient numerical methods to deal with the non-locality induced by topological types interactions [14, 20, 31, 32]. To address this computational issues, we introduce a novel stochastic simulation algorithm based on the classical Asymptotic Nanbu algorithm. This method is able to reduce the computational costs from quadratic to logarithmic by introducing a binary tree and a k-nearest neighbor search over it.

Follower-leader dynamics in optimization

Follower-leader dynamics can be a useful tool also in global optimization problems. In recent years, several numerical methods based on collective dynamics for the minimization of non-convex high dimensional functions and opinion formation have been developed [186, 103]. The advantage of these methods, which are also called gradient free methods, is that they can improve the efficiency of classical gradient based numerical optimization schemes such as the Newton method or gradient descent method [39]. Hence, they can be used in different applications, ranging from machine learning problems to engineering applications, where the evaluation of the cost function can be extremely costly [39, 52]. Among them we recall the simulated annealing, the particles swarm optimization algorithm (PSO) [117,

Figure 1.2: On the left a honey bees swarm and on the right a birds flock.

135, 166], the genetic algorithm (GA) [115, 153], and the Consensus based optimization algorithm (CBO) [48, 103, 104, 117, 165, 187]. In contrast with this last approach, whose dynamics is of mean field type, we can find other methods such as the kinetic based optimization (KBO) ones [21]. In Chapter 4 we introduce a novel particles method to bridge the gap between the KBO and the genetic algorithm. In the KBO algorithm the dynamics is described by a multidimensional Boltzmann equation and can be simulated towards Asymptotic Nanbu algorithms. Each agent is subjected to an attraction term toward the estimated position of the global minimizer at time t, which is a weighted mean over the particles position computed according to Laplace principle [82]. The genetic algorithm takes instead inspiration from the natural selection process that mimics biological evolution. A population of agents is divided into two sub-populations, called parents and children. Parents are assumed to be the agents occupying the best position over the cost function. Via crossover and mutation processes they produce children for the next generation. Over successive generation the population evolve toward an optimal solution. In this thesis, the idea is to consider a population which is divided into followers (children) and leaders (parents) and whose dynamics is similar to the one of the KBO. In particular, followers will explore the space searching for the position of the global minimizer and are attracted toward a randomly selected leader. Leaders instead move toward the estimated position of the global minimizer which is computed according to Laplace principle. Labels are changed in time by means of a stochastic process. Different numerical tests show that by developing a particle method involving a dynamically change of labels, it is possible to improve the efficiency of both the KBO and the genetic algorithm.

1.3.3 Biological competition models

Finally, the interest move to a different model involving two populations whose labels do not change in time. In particular, we focus on the description of a predator-prey dynamics starting from the individual or stochastic level [30, 33, 149, 150, 156]. Stochasticity, char-

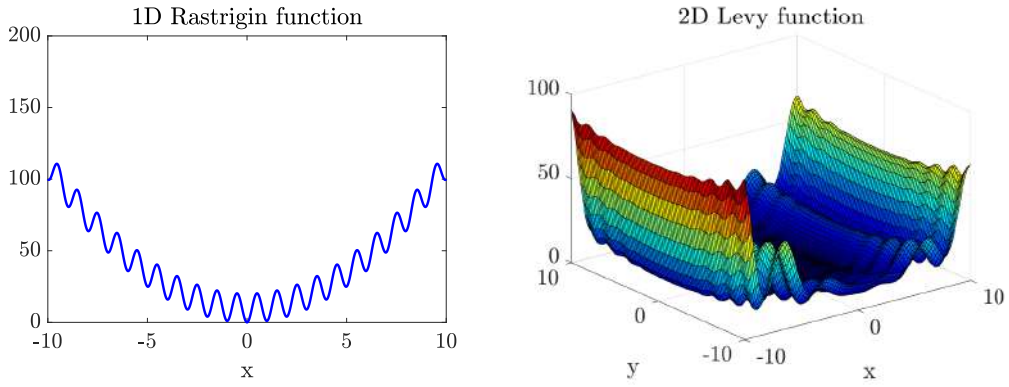

Figure 1.3: Two benchmark function for global optimization. On the left the 1d Rastrigin function and on the right the 2d Levy function.

acterized by inherent randomness and uncertainty, is ubiquitous in ecological systems and can have profound effects on population dynamics. As mentioned above, the approximation of the evolution of the density function via classical stochastic algorithms can be extremely demanding. Hence, in Chapter 5 we derive a novel stochastic algorithm which is able to preserve the accuracy of classical algorithms but which improves the efficiency.

Chapter 2

Instantaneous control strategies for magnetically confined fusion plasma

The principle behind magnetic fusion is to confine high temperature plasma inside a device in such a way that the nuclei of deuterium and tritium joining together can release energy. The high temperatures generated needs the plasma to be isolated from the wall of the device to avoid damages and the scope of external magnetic fields is to achieve this goal. In this chapter, to face this challenge from a numerical perspective, we study different instantaneous control mathematical approaches to steer a plasma into a given spatial region. From the modeling point of view, we focus on the Vlasov equation in a bounded domain with self induced electric field and an external strong magnetic field. The main feature of the control strategy employed is that it provides a feedback on the equation of motion based on an instantaneous prediction of the discretized system permitting to directly embed the minimization of a given cost functional into the particle interactions of the corresponding Vlasov model. The numerical results demonstrate the validity of our control approach and the capability of an external magnetic field, even if in a simplified setting, to lead the plasma far from the boundaries.

The results collected in this chapter are in preparation.

2.1 Introduction

Plasma is an electrically conducting fluid where temperature is so high that electrons are separated by their atoms and free to move [29]. This is called the fourth state of the matter and, in this state, a gas looks macroscopically neutral meaning that it is composed by roughly the same number of positive and negative charges while perturbation from neutrality arises only at the microscopic level [55]. Since most of the visible universe appears to be in the state of plasma, its behavior and properties are of intense interest to scientists in many disciplines ranging from physics, engineering to mathematics. In this framework, the development of numerical methods for solving plasma physics problems has attracted a lot of attention in the recent years[59, 110, 181, 72, 100, 194, 84, 98, 77, 78]. In particular, it is of great interest the study of magnetized plasma for its application in the so called fusion devices [170, 108, 93, 75, 78, 116] such as Tokamaks or Stellarators. In these machines, a strong magnetic field tries to contain the plasma during the fusion process of deuterium with

tritium and to avoid the direct contact of the charged particles with the walls of the devices. In order to describe such phenomenon, there exists many different models and associated numerical methods [108, 13, 26, 41, 81, 116] which are able to characterize the many features observed during the time evolution of a plasma. The choice of the appropriate model depends on the spatial or temporal scale at which one is interested when studying the fusion phenomenon [81]. An important role, in the context of the numerical approximation of these models, has been played by the so-called asymptotic preserving methods [77, 78, 96, 97, 85] in the recent past, which are able to efficiently deal with the different physical scales and by methods able to treat quasi-neutrality in plasma [70, 63, 132, 17]. In this work, we consider specifically one of the models commonly used to describe the evolution of a plasma when far from equilibrium, i.e. when collisions are not enough to reach an equilibrium state. The model is based on the Vlasov equation which describes the evolution of charged particles in an electromagnetic field which can either be self-consistent or externally applied. This reads for one single species of the plasma as [98]

$$\frac{\partial f(t,\mathbf{x},\mathbf{v})}{\partial t} + \mathbf{v}\cdot\nabla_{\mathbf{x}} f(t,\mathbf{x},\mathbf{v}) + F(t,\mathbf{x},\mathbf{v})\cdot\nabla_{\mathbf{v}} f(t,\mathbf{x},\mathbf{v}) = 0, \tag{2.1.1}$$

where

$$f = f(t,\mathbf{x},\mathbf{v}), \qquad f : \mathbb{R}_+ \times \mathbb{R}^{d_x} \times \mathbb{R}^{d_v}, \tag{2.1.2}$$

is the so-called distribution function giving the probability for the species of being in a certain position of the d_x dimensional in space and of the d_v dimensional in velocity space and where $F(t,\mathbf{x},\mathbf{v})$ represents a force field which can take different forms. For example, for the Vlasov-Poisson model, one has

$$F(t,\mathbf{x},\mathbf{v}) = \frac{q}{m}\mathbf{E}(t,\mathbf{x}), \qquad \mathbf{E}(t,\mathbf{x}) = -\nabla_{\mathbf{x}}\phi(t,\mathbf{x}), \qquad -\Delta_{\mathbf{x}}\phi(t,\mathbf{x}) = \frac{\rho(t,\mathbf{x}) - \rho_0(t,\mathbf{x})}{\varepsilon_0}, \tag{2.1.3}$$

where q is the elementary charge, m the mass of a a single particle, ε_0 the permittivity, $\mathbf{E}(t,\mathbf{x})$ represents the electric field, $\phi(t,\mathbf{x})$ the electric potential,

$$\rho(t,\mathbf{x}) = \int_{\mathbb{R}^{d_v}} f(t,\mathbf{x},\mathbf{v})d\mathbf{v}, \tag{2.1.4}$$

the charge density and $\rho_0(t,\mathbf{x})$ a static neutralizing background. On the other hand, for the Vlasov-Maxwell model one has

$$F(t,\mathbf{x},\mathbf{v}) = \mathbf{E}(t,\mathbf{x}) + \mathbf{v}\times\mathbf{B}(t,\mathbf{x}), \tag{2.1.5}$$

where $\mathbf{B}(t,\mathbf{x})$ represents the magnetic field and where the equation (2.1.1) is coupled with the solution of the Maxwell equation

$$\begin{aligned} &\frac{\partial \mathbf{E}(t,\mathbf{x})}{\partial t} = c^2\nabla_{\mathbf{x}}\times\mathbf{B}(t,\mathbf{x}) - \frac{\mathbf{J}}{\varepsilon_0}, \qquad \nabla_{\mathbf{x}}\cdot\mathbf{B}(t,\mathbf{x}) = 0, \\ &\frac{\partial \mathbf{B}(t,\mathbf{x})}{\partial t} = -\nabla_{\mathbf{x}}\times\mathbf{E}(t,\mathbf{x}), \qquad \nabla_{\mathbf{x}}\cdot\mathbf{E}(t,\mathbf{x}) = \frac{\rho(t,\mathbf{x}) - \rho_0(t,\mathbf{x})}{\varepsilon_0}. \end{aligned} \tag{2.1.6}$$

with $\mathbf{J} = \mathbf{q}\int_{\mathbb{R}^{\mathbf{dv}}} \mathbf{f}(\mathbf{t},\mathbf{x},\mathbf{v})\mathbf{dv}$ the current density c the speed of light and the compatibility condition which must hold true

$$\frac{\partial \rho}{\partial t} + \nabla_{\mathbf{x}} \cdot \mathbf{J} = \mathbf{0}. \tag{2.1.7}$$

The numerical solution of the fully three-dimensional Vlasov-Poisson (2.1.1)-(2.1.3) or Vlasov-Maxwell system (2.1.1)-(2.1.6) is already in itself a major challenge because of the huge size of the resulting system and the temporal and spatial scale related to its resolution. In this work, we set ourselves in an intermediate situation between the Poisson and the Maxwell model in which the magnetic field is given and external while the electric field is obtained from the solution of the Poisson equation, i.e.

$$\begin{aligned} -\Delta_{\mathbf{x}}\phi(t,\mathbf{x}) &= \frac{\rho(t,\mathbf{x}) - \rho_0(t,\mathbf{x})}{\varepsilon_0}, \quad \mathbf{B}(t,\mathbf{x}) = \mathbf{B}_{ext}(t,\mathbf{x}), \\ F(t,\mathbf{x},\mathbf{v}) &= \mathbf{E}(t,\mathbf{x}) + \mathbf{v}\times\mathbf{B}_{ext}(t,\mathbf{x}), \quad \mathbf{E}(t,\mathbf{x}) = -\nabla_{\mathbf{x}}\phi(t,\mathbf{x}). \end{aligned} \tag{2.1.8}$$

At the numerical level, such dynamics can be discretized by means of different numerical schemes, such as for example semi-Lagrangian schemes [72, 175, 194], finite difference-finite volume methods [71, 71] or Galerking schemes [76, 99], which are able to produce an accurate solution but in general have high computational costs. In contrast to the above recalled techniques, we find particle methods such as the famous particle-in-cell (PIC) methods [54, 96, 97, 121, 120, 80, 83, 87], which are able to produce a faster approximation of the evolution of the plasma density based on a representation of the solution as a sum of Dirac deltas or hybrid methods which tries to couple the advantages of deterministic and particle solvers in one single method [43, 174].

Besides on the numerical method used for approximating the Vlasov model (2.1.1), the main novelty of this work is to study efficient instantaneous control strategies based on the external magnetic field $\mathbf{B}_{ext}(t,\mathbf{x})$ obtained as a solution of an optimality principle, aiming at minimizing the mass which hits the boundaries and/or the energy close to the walls and to investigate the results of such strategy numerically. In literature, there exists some attempts of studying the magnetic confinement of a plasma by numerical simulations, however most of these research directions focus on hydrodynamic models [34] while the control of microscopic models describing systems of charged particles and based on kinetic equations is a direction which has almost not been explored in the past, we are aware of only two contributions to this case [91, 15]. The first refers to the Vlasov equation coupled with the Poisson equation while the second one refers to a case in which also the magnetic field is considered. In fact, the precise scope of an external magnetic field, which is the case we treat in this work, is to act as a control on the system, necessary to let the plasma assuming a desired configuration. Depending on the situation, its strength can in principle induces fast scale dynamics which have to be treated with ad-hoc numerical methods. In particular, in [96], a semi-implicit PIC method, which enjoys a wide stability area with respect to the spatial and temporal scales considered, has been proposed allowing the choice of larger

time steps for the evolution of the dynamics which is well suited for our scopes as precised later. The general form of the functional we aim to minimize reads as

$$\mathcal{J}(\mathbf{B}_{ext}; f^0) := \int_0^T \left(\mathcal{L}(f(t,\mathbf{x},\mathbf{v})) + \frac{\gamma}{2} \int_{\Omega_x} \|\mathbf{B}_{ext}(t,\mathbf{x})\|^2 d\mathbf{x} \right) dt, \tag{2.1.9}$$

where $\Omega_x \subseteq \mathbb{R}^{d_x}$ represents the space domain, $\mathcal{L}(f(t,\mathbf{x},\mathbf{v}))$ is a cost function, and γ acts as a weight penalizing the magnitude of the control $\|\mathbf{B}_{ext}\|$. For instance, by choosing

$$\mathcal{L}(f(t,\mathbf{x},\mathbf{v})) = \left\| \int_\Omega \psi(\mathbf{x},\mathbf{v}) \left(f(\mathbf{x},\mathbf{v},t) - \hat{f}(\mathbf{x},\mathbf{v},t) \right) d\mathbf{x} d\mathbf{v} \right\|^q, \tag{2.1.10}$$

where $\Omega = \Omega_x \times \Omega_v$ with $\Omega_v \subseteq \mathbb{R}^{d_v}$, for $q > 0$ means that we are looking to steer the moments of the distribution functions towards a desired state, while the specific case $\psi(\mathbf{x},\mathbf{v}) = 1$ corresponds to force the density of the plasma to be as close as possible to a given configuration. Then, the general form of the control problem we aim to study is given by

$$\begin{aligned} &\min_{\mathbf{B}_{ext} \in \mathcal{B}^{adm}} \mathcal{J}(\mathbf{B}_{ext}; f^0) \\ &\text{subject to} \\ &\quad \partial_t f(t,\mathbf{x},\mathbf{v}) + \mathbf{v} \cdot \nabla_{\mathbf{x}} f(t,\mathbf{x},\mathbf{v}) + (\mathbf{E}(t,\mathbf{x}) + \mathbf{v} \times \mathbf{B}_{ext}(t,\mathbf{x})) \cdot \nabla_{\mathbf{v}} f(t,\mathbf{x},\mathbf{v}) = 0, \\ &\quad -\Delta_{\mathbf{x}} \phi(t,\mathbf{x}) = \frac{\rho(t,\mathbf{x}) - \rho_0(t,\mathbf{x})}{\varepsilon_0}, \qquad \mathbf{E}(t,\mathbf{x}) = -\nabla_{\mathbf{x}} \phi(t,\mathbf{x}). \end{aligned} \tag{2.1.11}$$

The above problem has been analyzed theoretically in [138, 193, 46, 122], where the magnetic field has been computed as the superposition of the fields that are generated from different coils. A simplified configuration has been analyzed theoretically in [15] deriving the control following a Lagrangian approach by solving the optimality system. While another similar research direction has been recently explored in [91] where however the control is performed over the electric field. Up to our knowledge instantaneous control methods of the type described in [42, 6, 105] have never been investigated in this setting which is the direction we aim to explore in this work. In more details here, we assume, as it will be clarified later, that the external magnetic field may take different values on different parts of the domain and for this specific choice we derive an instantaneous feedback control for the plasma which permits to drive the system towards the desired state.

The rest of the chapter is organized as follows. In Section 2.2 we describe the geometrical setting and we introduce the details of the control problem. In Section 2.3, we give a general overview of the type of numerical scheme we choose to approximate our model. In particular, we will focus on particle-in-cell methods together with well suited semi-implicit discretizations. In Section 2.4, we discretize the control problem and we derive different control strategies. In Section 2.5- 2.6, we propose different numerical examples in which we

apply the control to the plasma dynamics. In Appendices 2.A and 2.B, we provide some detailed descriptions of the numerical methods not given in the main text and we test the numerical method on well known examples.

2.2 Problem setting

From now on, we restrict ourselves to a two dimensional setting, i.e. $d_x = d_v = 2$. This situation mimics a three dimensional axisymmetric toroidal device containing the plasma. To give a precise definition of our simplified setting, we focus on a two dimensional horizontal section of the three dimensional torus shown on the top of Figure 2.1. This is obtained from the intersection of the $x-y$ plane with the solid and it is shown on the bottom left of Figure 2.1. Finally, we restrict our analysis on a portion of this section which is approximated by a rectangle to avoid unnecessary complications as shown in the same Figure on the bottom right.

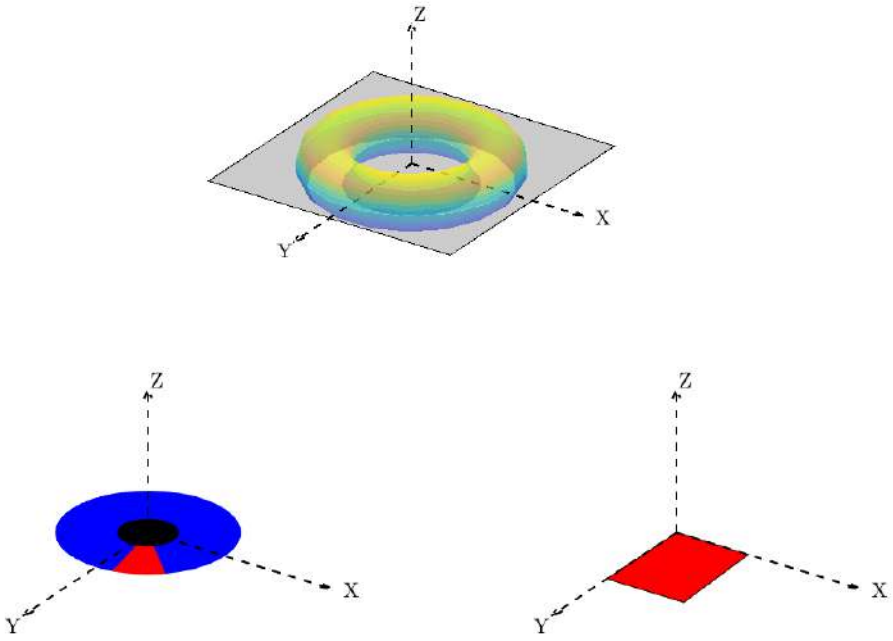

Figure 2.1: On the top the three dimensional torus, on the left bottom the toric section, on the bottom right the simplified geometry considered.

Over this depicted geometry, the external magnetic field is supposed to be such that

$$\mathbf{B}_{ext}(t,\mathbf{x}) = \left(0,0,B^{ext}(t,\mathbf{x})\right). \tag{2.2.1}$$

Hence, the two dimensional Vlasov-Poisson equations reads as

$$\begin{aligned}&\frac{\partial f(t,\mathbf{x}_\perp,\mathbf{v}_\perp)}{\partial t} + \mathbf{v}_\perp\cdot\nabla_{\mathbf{x}_\perp} f(t,\mathbf{x}_\perp,\mathbf{v}_\perp) + (\mathbf{E}(t,\mathbf{x}_\perp) + \mathbf{v}_\perp\times\mathbf{B}_{ext}(t,\mathbf{x}_\perp))\cdot\nabla_{\mathbf{v}_\perp} f(t,\mathbf{x}_\perp,\mathbf{v}_\perp) = 0,\\ &\mathbf{E}(t,\mathbf{x}_\perp) = -\nabla_{\mathbf{x}_\perp}\phi(t,\mathbf{x}_\perp), \qquad -\Delta_{\mathbf{x}_\perp}\phi(t,\mathbf{x}_\perp) = \rho(t,\mathbf{x}_\perp),\end{aligned} \tag{2.2.2}$$

where $\mathbf{x}_\perp = (x,y)$ and $\mathbf{v}_\perp = (v_x,v_y)$, and where we took $\rho_0 = 0$ and the scaling parameter $\varepsilon_0 = 1$ in the Poisson equation. In (2.2.2), $f(t,\mathbf{x}_\perp,\mathbf{v}_\perp)$ represents a set of charged particles lying on the $x-y$ plane where periodic boundary conditions are imposed in x direction while, if not otherwise stated, reflexive boundary conditions are imposed in y direction mimicking the wall of the device. The aim is to control the dynamics described by (2.2.2) with an external magnetic field in order to let particles assume a desired configuration and to stay as far as possible from the walls. In order to reach this goal, the idea consists in using the external magnetic field $\mathbf{B}_{ext}(t,\mathbf{x})$ as a control variable. However, in order to try to be realistic instead of considering $\mathbf{B}_{ext}(t,\mathbf{x})$ we introduce a space discretization grid with N_c cells, and we derive a control $B_k^{ext}(t)$ taking constant values in each of the cells C_k, $k=1,\dots,N_c$. In fact, from the practical point of view one cannot expect that the coils used to generate the magnetic field would be capable of realizing very complex pointwise field structures. Hence, we formulate the problem at the continuous level over a finite time horizon $[0,T]$ as follows

$$\min_{B^{ext}\in\mathcal{B}_{adm}} \sum_{k=1}^{N_c} \mathcal{J}_k(B_k^{ext};f^0), \qquad \text{s.t. (2.2.2) is verified} \tag{2.2.3}$$

where $B^{ext} = (B_1^{ext},\dots,B_{N_c}^{ext})$ now represents the vector of z components of $\mathbf{B}_{ext}(t,\mathbf{x}_\perp)$ defined in (2.2.1) within each cell C_k, $\mathcal{B}_{adm}$ is set of admissible controls such that $\mathcal{B}_{adm} = \{B_k^{ext} | B_k^{ext}\in[-M,M], M>0,\ k=1,\dots,N_c\}$, and the cost functional is defined as follows

$$\mathcal{J}_k(B_k^{ext};f^0) = \int_0^T \left(\mathcal{D}_k^{\mathrm{x}}(f,\psi_{\mathrm{x}})(t) + \mathcal{D}_k^{\mathrm{v}}(f,\psi_{\mathrm{v}})(t) + \frac{\gamma}{2}\|B_k^{ext}(t)\|^2\right) dt, \tag{2.2.4}$$

with $\gamma>0$ a penalization parameter of the control, and where the running cost aims at enforcing a specific velocity and position of the particles as follows

$$\mathcal{D}_k^{\ell}(f,\psi_\ell) = \frac{\alpha_\ell}{2}\|m_k[\psi_\ell](t) - \hat{\psi}_{\ell,k}\|^2 + \frac{\beta_\ell}{2}\sigma_k^2[\psi_\ell](t), \qquad \ell = \{\mathrm{x},\mathrm{v}\}, \tag{2.2.5}$$

with $\hat{\psi}_{\ell,k} \equiv \hat{\psi}_{\ell,k}(\mathbf{x}_\perp, \mathbf{v}_\perp)$ a target state, $\alpha_\ell, \beta_\ell \geq 0$, and mean and variance quantities

$$\begin{aligned} m_k[\psi](t) &= \int_{\Omega_k} \psi(\mathbf{x}_\perp, \mathbf{v}_\perp) f(t, \mathbf{x}_\perp, \mathbf{v}_\perp) d\mathbf{x}_\perp d\mathbf{v}_\perp, \\ \sigma_k^2[\psi](t) &= \int_{\Omega_k} \|\psi(\mathbf{x}_\perp, \mathbf{v}_\perp) - m_k[\psi](t)\|^2 f(t, \mathbf{x}_\perp, \mathbf{v}_\perp) d\mathbf{x}_\perp d\mathbf{v}_\perp, \end{aligned} \tag{2.2.6}$$

with $\Omega_k = C_k \times \mathbb{R}^2$, $k = 1, \ldots, N_c$. The purpose of the first term in (2.2.5) is to force the mean state of the particles in each cell C_k, such as velocity and position, to a desidered state $\hat{\psi}_k$, at same time the second term aims at reducing the variance of the particles distribution in the cell. In Section 2.4 we will discretize the problem in (2.2.3) and we will derive an instantaneous feedback control.

Remark 2. In the rest of the chapter we will use for simplicity the following notation $B(t, \cdot) = B^{ext}(t, \cdot)$, $\mathbf{B}(t, \cdot) = \mathbf{B}_{ext}(t, \cdot)$, $\mathbf{x} = \mathbf{x}_\perp$ and $\mathbf{v} = \mathbf{v}_\perp$.

2.3 Particle-in-cell (PIC) Methods

Particle-in-cell methods are used to study the motion of N particles, which in this context constitute the plasma density $f(t, \mathbf{x}, \mathbf{v})$. We approximate the plasma density at time t^{n+1} as

$$f^N(t^{n+1}, \mathbf{x}, \mathbf{v}) = \sum_{m=1}^{N} \omega_m \delta(\mathbf{x} - \mathbf{x}_m^{n+1}) \delta(\mathbf{v} - \mathbf{v}_m^{n+1}), \tag{2.3.1}$$

where $\delta(\cdot)$ is the Dirac-delta function and ω_m denotes the mass of each particle. The trajectory of each particle $m = 1, \ldots, N$ is computed from the characteristic curves corresponding to the Vlasov equation

$$\begin{aligned} \frac{d\mathbf{x}_m(t)}{dt} &= \mathbf{v}_m(t), \qquad \mathbf{x}_m(0) = \mathbf{x}_m^0, \\ \frac{d\mathbf{v}_m(t)}{dt} &= \mathbf{v}_m(t) \times \mathbf{B}(t, \mathbf{x}_m(t)) + \mathbf{E}(t, \mathbf{x}_m), \qquad \mathbf{v}_m(0) = \mathbf{v}_m^0, \end{aligned} \tag{2.3.2}$$

with $\mathbf{E}(t, \mathbf{x}_m)$ computed on the spatial mesh grid by solving the Poisson equation by means of a finite difference method (see Appendix 2.A for further details). At each time t, the approximated density is reconstructed considering the updated particles positions and velocities. Here we focus on the first order semi-implicit PIC method proposed in [96] to discretize the system of equations (2.3.2). Since the method is semi-implicit, it is unconditionally stable for a wider range of time step than an explicit method. The first step consists in sampling the initial position and velocity of each particle from a given initial distribution $f_0(\mathbf{x}, \mathbf{v})$. To do so, we compute the total mass and the mass of each particle as

$$m_{tot} = \int_{\mathbb{R}^4} f_0(\mathbf{x}, \mathbf{v}) d\mathbf{x} d\mathbf{v}, \qquad \omega_m = \frac{m_{tot}}{N}, \tag{2.3.3}$$

where N denotes the total number of particles. Then we consider a space discretization grid of size Δx with m_x nodes in the x direction and of size Δy with m_y nodes in the y direction. We determine the number of particles in each cell

$$N_k = \left\lfloor \frac{\rho_k}{N} \right\rceil, \tag{2.3.4}$$

where $\lfloor \cdot \rceil$ denotes the stochastic rounding, and

$$\rho_k = \int_{\mathbb{R}^2} f_0(\mathbf{x}_k, \mathbf{v}) d\mathbf{v}, \tag{2.3.5}$$

for any $k = (j-1)m_x + i$ with $i = 1, \dots, m_x$ and $j = 1, \dots, m_y$, is the density in each cell, with $\mathbf{x}_k$ denoting the nodes of the space discretization grid. Finally, we set uniformly the positions of the N_k particles inside each cell C_k, $k = (j-1)m_x + i$ for $i = 1, \dots, m_x$ and $j = 1, \dots, m_y$. Following a similar argument, we associate to each particle a certain velocity on a grid with $m_{v_x} \times m_{v_y}$ nodes, dependent of the initial distribution. Other methods to realized the initial sampling are for instance the Inverse Transform Sampling technique, or the Aceptance-Rejection algorithm, see [162]. Then, we consider a time interval $[0, T]$ divided into N_t intervals of size h, we define $t^n = nh$ and we write for any $m = 1, \dots, N$,

$$\begin{aligned} \mathbf{x}_m^{n+1} &= \mathbf{x}_m^n + h\mathbf{v}_m^{n+1}, \\ \mathbf{v}_m^{n+1} &= \mathbf{v}_m^n + h\mathbf{v}_m^{n+1} \times \mathbf{B}(t^n, \mathbf{x}_m^n) + h\mathbf{E}(t^n, \mathbf{x}_m^n), \end{aligned} \tag{2.3.6}$$

to simulate the evolution of the Vlasov equation. Finally we reconstruct the approximate density as in (2.3.1). A complete analysis of this family of PIC numerical methods have been proposed in [96].

2.4 Derivation of the instantaneous control

As mentioned in Section 2.2, we consider a space discretization grid with N_c cells of size $\Delta^c x \times \Delta^c y$ with $\Delta^c x \gg \Delta x$, and $\Delta^c y \gg \Delta y$, (see Figure 2.2), and we provide a feedback control B_k, $k = 1, \dots, N_c$, based on a one-step prediction of the dynamics, which takes constant values in each cell C_k.

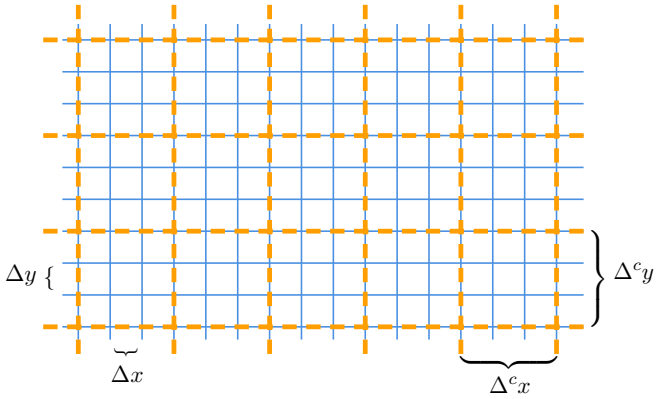

Figure 2.2: Space discretization grids. The finer continuous (blue) mesh represents the discretization grid on which we reconstruct the density, the dashed (orange) mesh the cell grid on which we will define the control B_k.

Hence, we formulate the discretized version of (2.2.3) in a short time horizon $[t; t+h]$ as follows

$$\min_{B\in\mathcal{B}_{adm}} \sum_{k=1}^{N_c} \mathcal{J}_k^N(B_k), \tag{2.4.1}$$

subject to the discretize dynamics

$$\begin{aligned} x_i^{n+1} &= x_i^n + h v_{x_i}^{n+1}, \\ y_i^{n+1} &= y_i^n + h v_{y_i}^{n+1}, \\ v_{x_i}^{n+1} &= v_{x_i}^n + h v_{y_i}^n B_k + h E_{x_i}^n, \\ v_{y_i}^{n+1} &= v_{y_i}^n - h v_{x_i}^n B_k + h E_{y_i}^n, \end{aligned} \tag{2.4.2}$$

where $B = (B_1, \dots, B_{N_c})$, and with B_{adm} a set of admissible controls such that $B_{adm} = \{B_k | B_k \in [-M, M], M > 0, k = 1, \dots, N_c\}$. By considering the discretized version of (2.2.4), the functional in (2.4.1) reads as follows

$$\mathcal{J}_k^N(B_k) = h\left(\mathcal{D}_k^{\mathrm{x}}(f^N, \psi_{\mathrm{x}})(t^{n+1}) + \mathcal{D}_k^{\mathrm{v}}(f^N, \psi_{\mathrm{v}})(t^{n+1}) + \frac{\gamma}{2}\|B_k\|^2\right), \tag{2.4.3}$$

where

$$\mathcal{D}_k^{\ell}(f^N, \psi_\ell) = \frac{\alpha_\ell}{2}\|m_k[\psi_\ell](t^{n+1}) - \hat{\psi}_{\ell,k}\|^2 + \frac{\beta_\ell}{2}\sigma_k^2[\psi_\ell](t^{n+1}), \qquad \ell = \{\mathrm{x}, \mathrm{v}\}, \tag{2.4.4}$$

with $\hat{\psi}_{\ell,k} \equiv \hat{\psi}_{\ell,k}(\mathbf{x}_\perp, \mathbf{v}_\perp)$ a target state, $\alpha_\ell, \beta_\ell \geq 0$, f^N the empirical density (2.3.1), and

where the mean and variance quantities reads as follows

$$\begin{aligned} m_k[\psi](t^{n+1}) &= \int_{\Omega_k} \psi(\mathbf{x}_\perp, \mathbf{v}_\perp) f^N(t^{n+1}, \mathbf{x}_\perp, \mathbf{v}_\perp) d\mathbf{x}_\perp d\mathbf{v}_\perp, \\ \sigma_k^2[\psi](t^{n+1}) &= \int_{\Omega_k} \|\psi(\mathbf{x}_\perp, \mathbf{v}_\perp) - m_k[\psi](t^n)\|^2 f^N(t^{n+1}, \mathbf{x}_\perp, \mathbf{v}_\perp) d\mathbf{x}_\perp d\mathbf{v}_\perp, \end{aligned} \tag{2.4.5}$$

with $\Omega_k = C_k \times \mathbb{R}^2$, $k = 1, \dots, N_c$. By computing explicitly the terms in (2.4.5), and by setting $\psi_{\mathrm{x}} = y^{n+1}$, $\psi_{\mathrm{v}} = v_y^{n+1}$, $\hat{\psi}_{\mathrm{x},k} = \hat{y}_k$ and $\hat{\psi}_{\mathrm{v},k} = \hat{v}_{y_k}$, we can rewrite the functional in (2.4.3) as

$$\begin{aligned} \mathcal{J}_k^N(B_k) = \frac{\alpha_{\mathrm{v}}}{2}\|\bar{v}_{y,k}^{n+1} - \hat{v}_{y_k}\|^2 + \frac{\beta_{\mathrm{v}}}{2N_k}\sum_{i\in C_k}\|v_{y_i}^{n+1} - \bar{v}_{y,k}^n\|^2 \\ + \frac{\alpha_{\mathrm{x}}}{2}\|\bar{y}_k^{n+1} - \hat{y}_k\|^2 + \frac{\beta_{\mathrm{x}}}{2N_k}\sum_{i\in C_k}\|y_i^{n+1} - \bar{y}_k^n\|^2 + \frac{\gamma}{2}\|B_k^n\|^2, \end{aligned} \tag{2.4.6}$$

with

$$\bar{y}_k = \frac{1}{N_k}\sum_{j\in C_k} y_j, \qquad \bar{v}_{y,k} = \frac{1}{N_k}\sum_{j\in C_k} v_{y_j}, \tag{2.4.7}$$

denoting the mean position and velocity over cell C_k, $k = 1, \dots, N_c$, and where we assumed $\alpha_\ell h = \alpha_\ell$, $\beta_\ell h = \beta_\ell$ for any $\ell = \{\mathrm{x}, \mathrm{v}\}$, and $\gamma h = \gamma$. The following Proposition holds.

Proposition 1. *The feedback control at cell C_k associated to (2.4.6) reads as follows*

$$B_k = \mathbb{P}_{[-M,M]}\left(\frac{\mathcal{N}_{\mathrm{v}}^{n,k} + \mathcal{N}_{\mathrm{x}}^{n,k}}{\gamma + \mathcal{D}_{\mathrm{v}}^{n,k} + \mathcal{D}_{\mathrm{x}}^{n,k}}\right), \tag{2.4.8}$$

with $\gamma > 0$, and

$$\begin{aligned} \mathcal{N}_{\mathrm{v}}^{n,k} &= \alpha_{\mathrm{v}}(\bar{v}_{y,k}^n + h\bar{E}_{y,k}^n - \hat{v}_{y_k})\bar{v}_{x,k}^n + \frac{\beta_{\mathrm{v}}}{N_k}\sum_{i=1}^{N_k}\left[(v_{y_i}^n + hE_{y_i}^n - \bar{v}_{y,k}^n)v_{x_i}^n\right], \\ \mathcal{N}_{\mathrm{x}}^{n,k} &= \alpha_{\mathrm{x}}(\bar{y}_k^n + h(\bar{v}_{y,k}^n + h\bar{E}_{y,k}^n) - \hat{y}_k)\bar{v}_{x,k}^n + \frac{\beta_{\mathrm{x}}}{N_k}\sum_{i=1}^{N_k}\left[(y_i^n + h(v_{y_i}^n + hE_{y_i}^n) - \bar{y}_k^n)v_{x_i}^n\right], \\ \mathcal{D}_{\mathrm{v}}^{n,k} &= h\left(\alpha_{\mathrm{v}}(\bar{v}_{x,k}^n)^2 + \frac{\beta_{\mathrm{v}}}{N_k}\sum_{i=1}^{N_k}(v_{x_i}^n)^2\right), \qquad \mathcal{D}_{\mathrm{x}}^{n,k} = h^2\left(\alpha_{\mathrm{x}}(\bar{v}_{x,k}^n)^2 + \frac{\beta_{\mathrm{x}}}{N_k}\sum_{i=1}^{N_k}(v_{x_i}^n)^2\right), \end{aligned} \tag{2.4.9}$$

$\mathbb{P}_{[-M,M]}(\cdot)$ denoting the projection over the interval $[-M, M]$, and where we assumed the parameters to scale as follows

$$\alpha_{\mathrm{x}} \to \frac{\alpha_{\mathrm{x}}}{h}, \qquad \beta_{\mathrm{x}} \to \frac{\beta_{\mathrm{x}}}{h}, \qquad \gamma \to \gamma h. \tag{2.4.10}$$

In the limit $h \to 0$ the control at the continuos level reads,

$$B_k(t) = \mathbb{P}_{[-M,M]}\left(\frac{1}{\gamma}\left(\mathcal{N}_{\mathrm{v}}^k(t) + \mathcal{N}_{\mathrm{x}}^k(t)\right)\right), \tag{2.4.11}$$

with

$$\begin{aligned}
\mathcal{N}_{\mathrm{v}}^k(t) &= \alpha_{\mathrm{v}}(\bar{v}_{y,k}(t) - \hat{v}_{y_k})\bar{v}_{x,k}(t) + \frac{\beta_{\mathrm{v}}}{N_k}\sum_{i\in C_k}(v_{y_i}(t) - \bar{v}_{y,k}(t))v_{x_i}(t),\\
\mathcal{N}_{\mathrm{x}}^k(t) &= \alpha_{\mathrm{x}}(\bar{y}_k(t) - \hat{y}_k)\bar{v}_{x,k}(t) + \frac{\beta_{\mathrm{x}}}{N_k}\sum_{i\in C_k}(y_i(t) - \bar{y}_k(t))v_{x_i}(t)
\end{aligned} \tag{2.4.12}$$

Proof. We introduce the augmented Lagrangian

$$\mathcal{L}(B, \lambda^0, \lambda^1) = -\mathcal{J}_k^N(B_k) - \sum_{k=1}^{N_c}[\lambda_k^0(B_k - M) + \lambda_k^1(-B_k - M)], \tag{2.4.13}$$

with λ^0, λ^1 Lagrangian multipliers. For any $k = 1, \dots, N_c$ we solve the optimality system

$$\begin{cases}
\partial_{B_k}\mathcal{L} = 0,\\
\lambda_k^0(B_k - M) = 0,\\
\lambda_k^1(-B_k - M) = 0,\\
\lambda_k^0 \geq 0, \qquad \lambda_k^1 \geq 0,\\
-M \leq B_k \leq M.
\end{cases} \tag{2.4.14}$$

By scaling the parameters as in (2.4.10), by setting $\lambda_k^j \to \lambda_k^j h$, for any $j = 0, 1$, $k = 1, \dots, N_c$ from the first equation in (2.4.14) we get

$$\begin{aligned}
\partial_{B_k}\mathcal{L} = -\Big[&\alpha_{\mathrm{v}}\left(\bar{v}_{y,k}^n - h\bar{v}_{x,k}^n B_k + h\bar{E}_{y,k}^n - \hat{v}_{y_k}\right)\left(-\bar{v}_{x,k}^n\right) +\\
&+ \frac{\beta_{\mathrm{v}}}{N_k}\sum_{i\in C_k}\left\{\left(v_{y_i}^n - hv_{x_i}^n B_k + hE_{y_i}^n - \bar{v}_{y,k}^n\right)\left(-v_{x_i}^n\right)\right\} +\\
&+ \alpha_{\mathrm{x}}\left(\bar{y}_k^n + h(\bar{v}_{y,k}^n - h\bar{v}_{x,k}^n B_k + h\bar{E}_{y,k}^n) - \hat{y}_k\right)\left(-\bar{v}_{x,k}^n\right) +\\
&+ \frac{\beta_{\mathrm{x}}}{N_k}\sum_{i\in C_k}\left\{\left(y_i^n + h(v_{y_i}^n - hv_{x_i}^n B_k + hE_{y_i}^n) - \bar{y}_k^n\right)\left(-v_{x_i}^n\right)\right\} + \gamma B_k\Big] - \lambda_k^0 + \lambda_k^1 = 0.
\end{aligned}$$

Then, if $\lambda_k^0 = 0$ and $\lambda_k^1 \geq 0$, we get from the third equation in (2.4.14) $B_k = -M$. From

the first equation in (2.4.14), setting $B_k = -M$ we get

$$\begin{aligned}\lambda_k^1 =&\alpha_{\mathrm{v}}\left(\bar{v}_{y,k}^n + h\bar{v}_{x,k}^n M + h\bar{E}_{y,k}^n - \hat{v}_{y_k}\right)\left(-\bar{v}_{x,k}^n\right) + \\ &+ \frac{\beta_{\mathrm{v}}}{N_k}\sum_{i\in C_k}\left\{\left(v_{y_i}^n + hv_{x_i}^n M + hE_{y_i}^n - \bar{v}_{y,k}^n\right)\left(-v_{x_i}^n\right)\right\} + \\ &+ \alpha_{\mathrm{x}}\left(\bar{y}_k^n + h(\bar{v}_{y,k}^n + h\bar{v}_{x,k}^n M + h\bar{E}_{y,k}^n) - \hat{y}_k\right)\left(-\bar{v}_{x,k}^n\right) + \\ &+ \frac{\beta_{\mathrm{x}}}{N_k}\sum_{i\in C_k}\left\{\left(y_i^n + h(v_{y_i}^n + hv_{x_i}^n M + hE_{y_i}^n) - \bar{y}_k^n\right)\left(-v_{x_i}^n\right)\right\} - \gamma M \geq 0\end{aligned} \tag{2.4.15}$$

from which we get

$$\frac{\mathcal{N}_{\mathrm{v}}^{n,k} + \mathcal{N}_{\mathrm{x}}^{n,k}}{\gamma + \mathcal{D}_{\mathrm{v}}^{n,k} + \mathcal{D}_{\mathrm{x}}^{n,k}} \leq -M, \tag{2.4.16}$$

with $\mathcal{N}_{\mathrm{v}}$, $\mathcal{N}_{\mathrm{x}}^{n,k}$, $\mathcal{D}_{\mathrm{v}}^{n,k}$, $\mathcal{D}_{x}^{n,k}$ defined as in (2.4.9). If $\lambda_k^0 \geq 0$ and $\lambda_k^1 = 0$, we get from the third equation in (2.4.14) $B_k = M$. Following a similar argument from the lower bound we have

$$\frac{\mathcal{N}_{\mathrm{v}}^{n,k} + \mathcal{N}_{\mathrm{x}}^{n,k}}{\gamma + \mathcal{D}_{\mathrm{v}}^{n,k} + \mathcal{D}_{\mathrm{x}}^{n,k}} \geq M. \tag{2.4.17}$$

If $\lambda_k^0 = \lambda_k^1 = 0$, from the first equation in (2.4.14) we get

$$B_k = \frac{\mathcal{N}_{\mathrm{v}}^{n,k} + \mathcal{N}_{\mathrm{x}}^{n,k}}{\gamma + \mathcal{D}_{\mathrm{v}}^{n,k} + \mathcal{D}_{\mathrm{x}}^{n,k}}, \tag{2.4.18}$$

and from the forth equation we get

$$-M \leq B_k \leq M. \tag{2.4.19}$$

All in all we get B_k defined as in (2.4.8). In the limit $h \to 0$ we get B_k defined as in (2.4.11). □

Remark 3.

- The feedback control obtained from (2.4.11) can be equivalently derived as an instantaneous control of the following functional on the interval $[t, t+h]$, without introducing the rescaling (2.4.10),

$$\begin{aligned}\mathcal{J}_k(B_k) = &\frac{\alpha_{\mathrm{v}}}{2}\|\bar{v}_{y,k}(t+h) - \hat{v}_{y_k}\|^2 + \frac{\beta_{\mathrm{v}}}{2N_k}\sum_{i\in C_k}\|v_{y_i}(t+h) - \bar{v}_{y,k}(t)\|^2 \\ &+ \int_t^{t+h}\left(\frac{\alpha_{\mathrm{x}}}{2}\|\bar{v}_{y,k}(\tau) - \hat{\mathcal{V}}_k(\tau)\|^2 + \frac{\beta_{\mathrm{x}}}{2N_k}\sum_{i\in C_k}\|v_{y_i}(\tau) - \bar{\mathcal{V}}_{k,i}(\tau)\|^2\right)d\tau \\ &+ \frac{\gamma}{2}\int_t^{t+h}\|B_k(\tau)\|^2\,d\tau,\end{aligned} \tag{2.4.20}$$

where we introduced the term

$$\hat{\mathcal{V}}_k(\tau) = \hat{\eta}(\hat{y}_k - \bar{y}_k(\tau)), \quad \bar{\mathcal{V}}_{k,i}(\tau) = \bar{\eta}(\bar{y}_k(\tau) - y_i(\tau)), \qquad \hat{\eta}, \bar{\eta} > 0.$$

Hence,we consider the following discretization

$$\begin{aligned}\mathcal{J}_k(B_k) = &\frac{\alpha_{\mathrm{v}}}{2}\|\bar{v}_{y,k}^{n+1} - \hat{v}_{y_k}\|^2 + \frac{\beta_{\mathrm{v}}}{2N_k}\sum_{i\in C_k}\|v_{y_i}^{n+1} - \bar{v}_{y,k}^{n}\|^2 \\ &+ \frac{h\alpha_{\mathrm{x}}}{2}\|\bar{v}_{y,k}^{n+1} - \hat{\mathcal{V}}_k^n\|^2 + \frac{h\beta_{\mathrm{x}}}{2N_k}\sum_{i\in C_k}\|v_{y_i}^{n+1} - \bar{\mathcal{V}}_{k,i}^n\|^2 + \frac{h\gamma}{2}\|B_k\|^2,\end{aligned} \tag{2.4.21}$$

and we observe that the following reformulation is viable for the third term

$$\frac{h\alpha_{\mathrm{x}}}{2}\|\bar{v}_{y,k}^{n+1} - \hat{\mathcal{V}}_k^n\|^2 = \frac{h\alpha_{\mathrm{x}}}{2}\left\|\frac{\bar{y}_k^{n+1} - \bar{y}_k^n}{h} - \frac{\tilde{y}_k^n - \bar{y}_k^n}{h}\right\|^2 = \frac{\alpha_{\mathrm{x}}}{2h}\left\|\bar{y}_k^{n+1} - \tilde{y}_k^n\right\|^2$$

where $\tilde{y}_k^n = \bar{y}_k^n + h\hat{\eta}(\hat{y}_k - \bar{y}_k^n)$. The same argument holds for the forth term of (2.4.21), and assuming $h\hat{\eta} = 1$ and $h\bar{\eta} = 1$ we recover the cost functional in (2.4.6).

- To derive the different controls we consider an explicit discretization scheme. Once that we have derived the value of B_k we will plug it into the semi-implicit scheme that is used to evolve the dynamics.

2.5 Numerical experiments: validation test

The aim of this section is to show how to control the instabilities by means of a magnetic field derived as in (2.4.8). We consider a two dimensional domain and we impose periodic boundary conditions in x and either reflexive or Dirichlet boundary conditions in y. We consider as initial density

$$f_0(x,y) = \frac{1}{\sqrt{2\pi\sigma}}\left(\exp\left(-\frac{(y-c_y)^2}{2\sigma^2}\right) + \exp\left(-\frac{(y+c_y)^2}{2\sigma^2}\right)\right), \tag{2.5.1}$$

with $c_y = 1$ and $\sigma = 0.3$, and we assume $(x,y) \in [0,40] \times [-1.5, 1.5]$. We sample $N = 10^6$ particles from the initial distribution with position (x_i^0, y_i^0), and we set $v_{x_i}^0 = 1$ and $v_{y_i}^0 = 10$ for any particle i s.t. $y_i^0 \geq 0$, and $v_{y_i}^0 = -10$ for any particle i s.t. $y_i^0 < 0$. In Figure 2.3 the initial configuration.

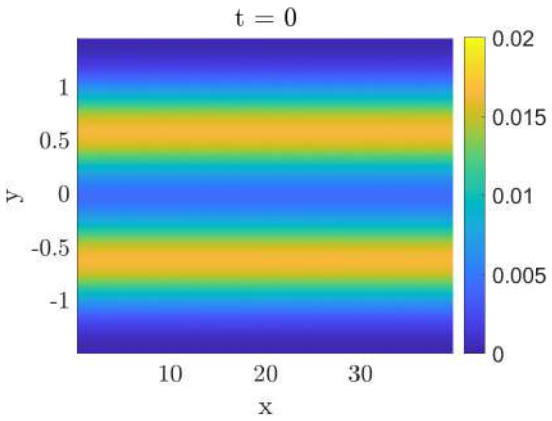

Figure 2.3: Approximation of the initial density.

We first assume to have no control and a constant magnetic field $B = 1$ acting on the system. Then we test the problem supposing that the magnetic field is derived as the solution of a control problem aiming at minimizing the percentage of mass which hits the lower and upper boundaries. We suppose the domain on which the control is active to be divided into N_c cells C_k, $k = (j_x - 1)K_x + i_k$, with $i_x = 1, \dots, K_x$ and $j_x = 1, \dots, K_y$. In the following tests we set $K_x = 1$ and $K_y = 2$. We let the dynamics to evolve up to time $T = 100$, choosing $h = 0.1$ as time steps. We set $m_x \times m_y$, with $m_x = m_y = 64$, as number of cells for the space discretization and $m_{v_x} \times m_{v_y}$, with $m_{v_x} = m_{v_y} = 128$, number of cells in the velocity space. We then reconstruct the density at each time as in (2.3.1). To test the control, we compute also the percentage of mass at the boundaries and inside the domain and the total kinetic energy. In particular, the mass at the boundaries at time t^n is given by

$$m_b(t^n) = \int_{\Omega_b} f^N(t^n, \mathbf{x}, \mathbf{v}) d\mathbf{x} d\mathbf{v}, \tag{2.5.2}$$

with $f^N(\cdot)$ defined as in (2.3.1) and $\Omega_b = [-1.5, -0.625] \cup [0.625, 1.5]$, while the mass inside the domain at time t^n is defined to be

$$m_i^n(t^n) = 1 - m_b^n(t^n). \tag{2.5.3}$$

The kinetic energy at the boundaries at time t^n is instead computed as

$$\mathcal{E}(t^n) = \frac{1}{2} \int_{\Omega_b} \mathbf{v}^2 f^N(t^n, \mathbf{x}, \mathbf{v}) d\mathbf{x} d\mathbf{v}, \tag{2.5.4}$$

with $f^N(\cdot)$ defined as in (2.3.1) and $\Omega_b = [-1.5, -0.625] \cup [0.625, 1.5]$.

2.5.1 Test with reflexive boundary conditions in y

Test without control. In Figure 2.4 three snapshots of the dynamics at time $t = 5$, $t = 50$ and $t = 100$. In red the velocity vector field. The mass oscillates around the centre of the domain due to the fact that we impose reflexive boundary conditions in y.

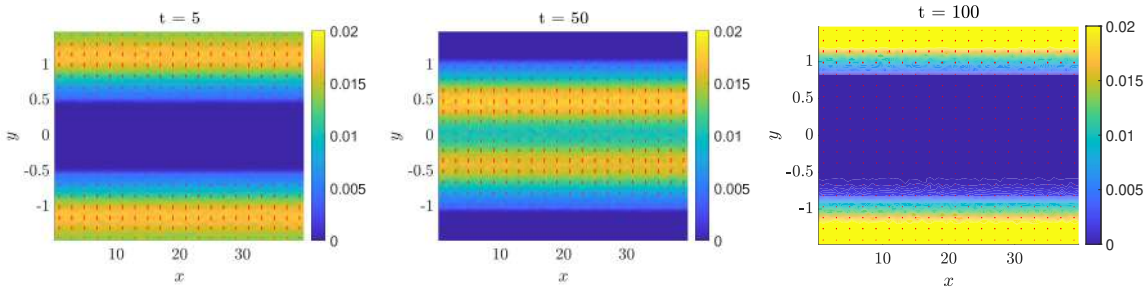

Figure 2.4: Test without control with $B = 1$. Three snapshots of the dynamics taken at time $t = 5$, $t = 50$ and $t = 100$, described in equation (2.2.2). In red the velocity vector field.

In Figure 2.5 we show on the left the mass percentages in time and on the right the kinetic energy at the boundaries as a function of time computed as in (2.5.2)-(2.5.3), and (2.5.4), respectively. Once that particles hits the boundaries, the energy increases.

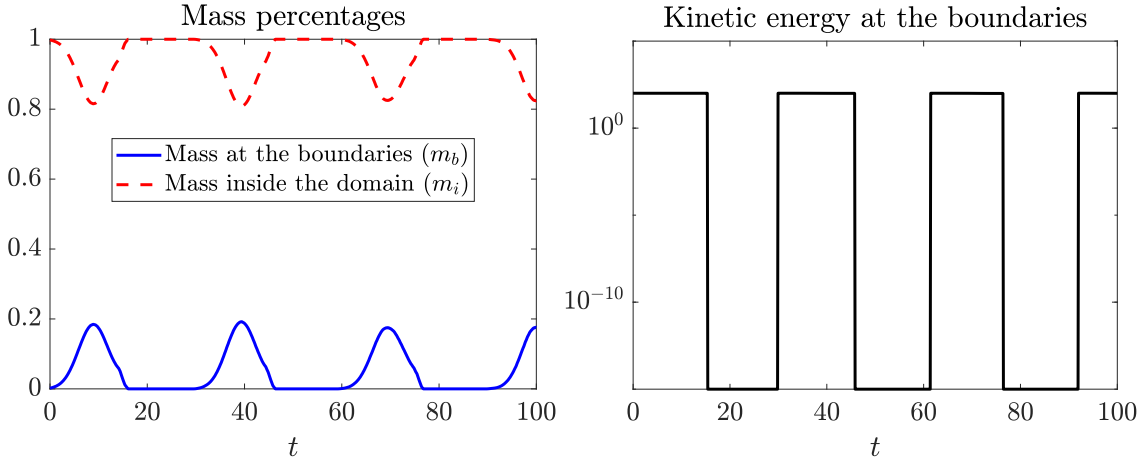

Figure 2.5: Test without control with $B = 1$. On the left the mass percentages defined as in equation (2.5.2)-(2.5.3) and on the right the kinetic energy at the boundaries defined as in (2.5.4).

Test with an instantaneous control of the particles velocity mean and variance. We consider the case of an instantaneous control of the particles velocity mean and variance derived as in (2.4.8) by setting $\alpha_{\mathrm{v}} = 1.5$, $\beta_{\mathrm{v}} = 0.1$, $\alpha_{\mathrm{x}} = \beta_{\mathrm{x}} = 0$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). We choose $\hat{v}_{y_1} = 1$ and $\hat{v}_{y_2} = -1$ as targets points in our control problem to guarantee that the velocity points toward the centre of the domain. In Figure 2.6 in the first row three snapshots of the dynamics described by equation (2.2.2) at time $t = 5$, $t = 50$ and $t = 100$. In red the velocity vector field. In the second row the correspondent value of the magnetic field. The mass is forced to move toward the centre of the domain.

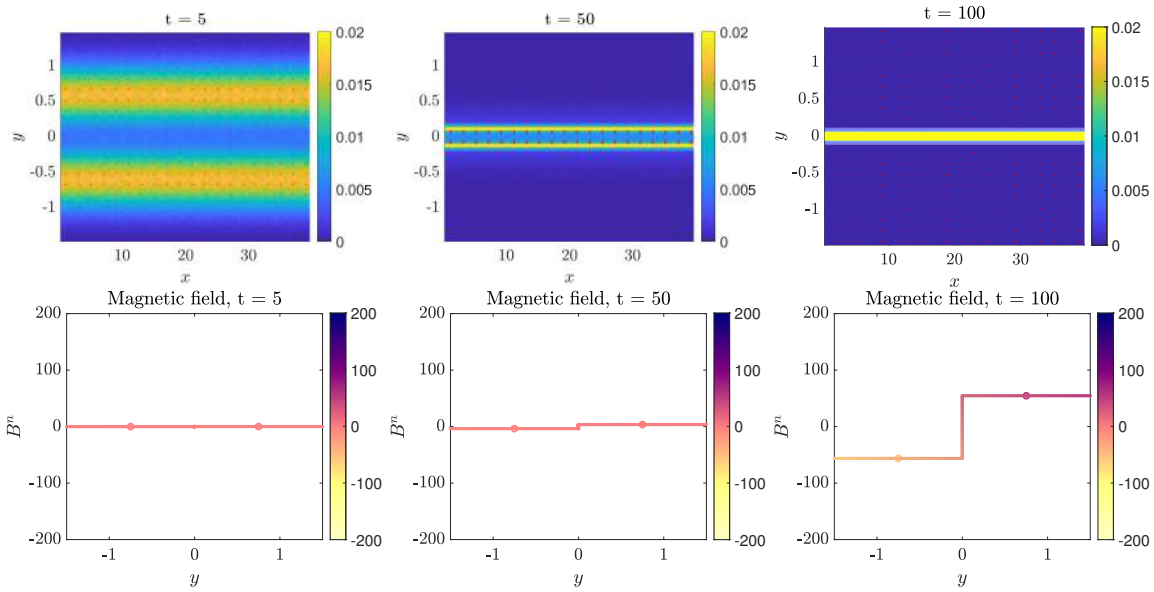

Figure 2.6: Test with an instantaneous control of the particles velocity mean and variance derived as in (2.4.8) by setting $\alpha_{\mathrm{v}} = 1.5$, $\beta_{\mathrm{v}} = 0.1$, $\alpha_{\mathrm{x}} = \beta_{\mathrm{x}} = 0$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). In the first row, three snapshots of the dynamics taken at time $t = 5$, $t = 50$ and $t = 100$, described in equation (2.2.2). In red the velocity vector field. In the second row, the value of the magnetic field in each cell.

In Figure 2.7 the value of the magnetic field in the cells C_k, $k = 1, 2$ as a function of time. The control takes small values in early times and it oscillate in late time.

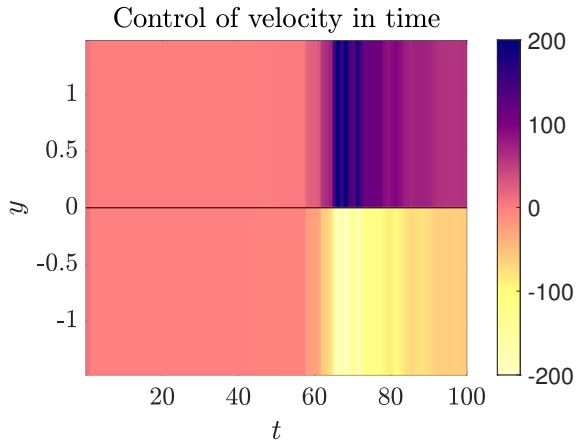

Figure 2.7: Test with an instantaneous control of the particles velocity mean and variance derived as in (2.4.8) by setting $\alpha_v = 1.5$, $\beta_v = 0.1$, $\alpha_x = \beta_x = 0$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). Value of the control B_k in each cell C_k, $k = 1, 2$ as a function of time.

In Figure 2.8 we plot the mass percentages in time, computed as in (2.5.2)-(2.5.3), and the kinetic energy at the boundaries as a function of time computed as in (2.5.4). Furthermore we zoom and plot in logarithmic scale the mass at the boundaries to highlight the fact that the energy increases if a small percentage of mass hits the boundary.

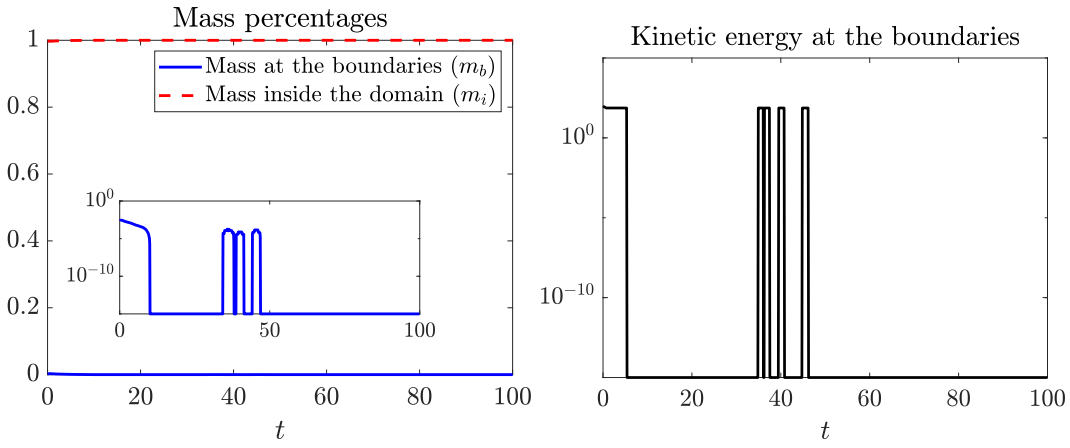

Figure 2.8: Test with an instantaneous control of the particles velocity mean and variance derived as in (2.4.8) by setting $\alpha_v = 1.5$, $\beta_v = 0.1$, $\alpha_x = \beta_x = 0$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). On the left the mass percentages defined as in equation (2.5.2)-(2.5.3), and the mass at the boundaries in logarithmic scale, and on the right the kinetic energy at the boundaries defined as in (2.5.4).

Test with an instantaneous control of the particles position and velocity mean and variance. We consider an instantaneous control of the particles position and velocity mean and variance derived as in (2.4.8) by setting $\alpha_{\mathrm{x}} = \alpha_{\mathrm{v}} = 1.5$, $\beta_{\mathrm{x}} = \beta_{\mathrm{v}} = 0.1$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). We choose $\hat{y}_k = 0$ for any $k = 1, 2$ and $\hat{v}_{y_1} = 1$, $\hat{v}_{y_2} = -1$ as target points, in such a way to force the mass to move toward the centre of the domain. In Figure 2.9 in the first row three snapshots of the dynamics described by equation (2.2.2) at time $t = 5$, $t = 50$ and $t = 100$. In red the velocity vector field. In the second row the correspondent value of the magnetic field. The mass is forced to move toward the centre of the domain.

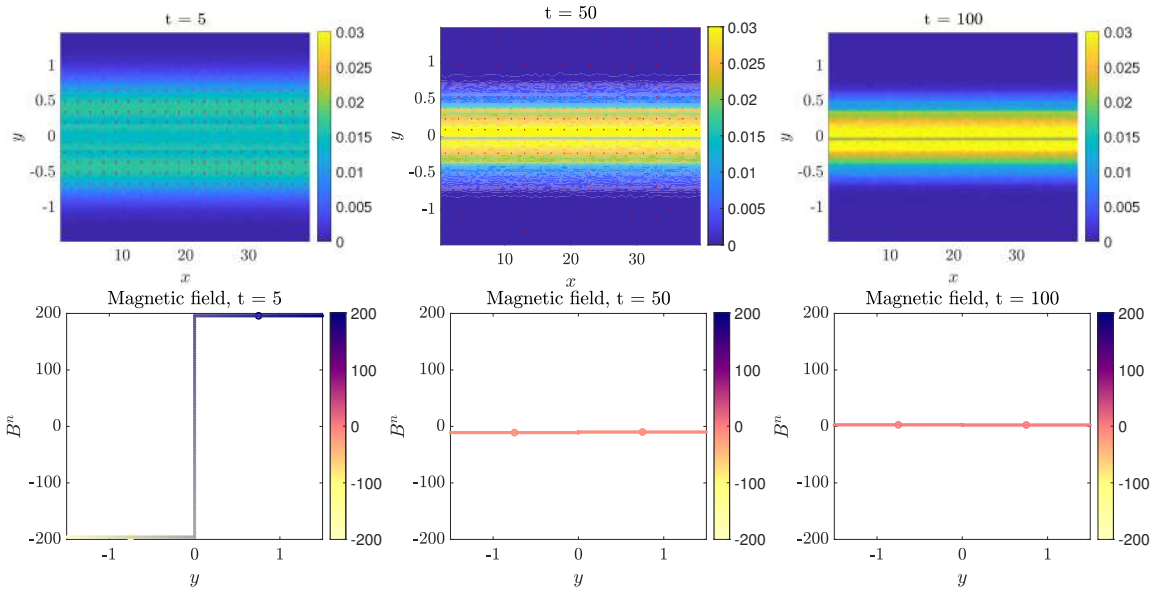

Figure 2.9: Test with an instantaneous control of the particles position and velocity mean and variance derived as in (2.4.8) by setting $\alpha_{\mathrm{x}} = \alpha_{\mathrm{v}} = 1.5$, $\beta_{\mathrm{x}} = \beta_{\mathrm{v}} = 0.1$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). In the first row, three snapshots of the dynamics taken at time $t = 10$, $t = 70$ and $t = 150$, described in equation (2.2.2). In red the velocity vector field. In the second row, the value of the magnetic field in each cell.

In Figure 2.10 the value of the magnetic field in the cells C_k, $k = 1, \ldots, N_k$ as a function of time. The control activates once that the mass move far away from the centre of the domain.

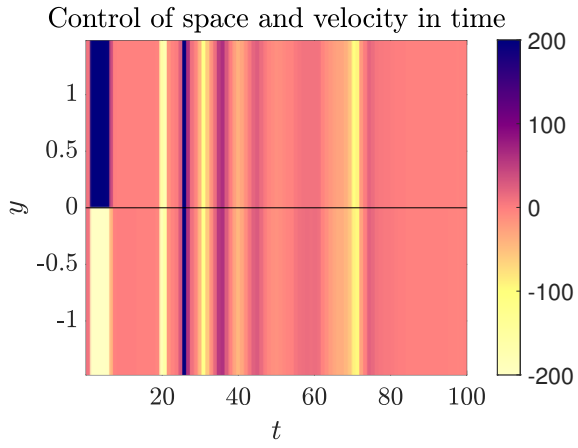

Figure 2.10: Test with an instantaneous control of the particles position and velocity mean and variance derived as in (2.4.8) by setting $\alpha_x = \alpha_v = 1.5$, $\beta_x = \beta_v = 0.1$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). Value of the control B_k in each cell C_k, $k = 1, 2$ as a function of time.

In Figure 2.11 we plot the mass percentages in time, computed as in (2.5.2)-(2.5.3), and the kinetic energy at the boundaries as a function of time computed as in (2.5.4). We add the plot of the mass at the boundaries in logarithmic scale. Even if part of the mass hits the boundaries the energy decreases toward zero.

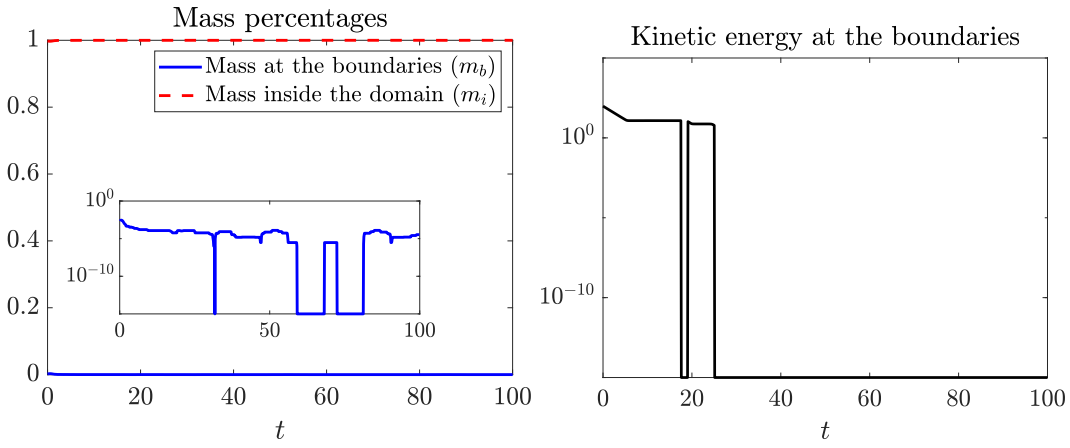

Figure 2.11: Test with an instantaneous control of the particles position and velocity mean and variance derived as in (2.4.8) by setting $\alpha_x = \alpha_v = 1.5$, $\beta_x = \beta_v = 0.1$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). On the left the mass percentages defined as in equation (2.5.2)-(2.5.3) and the mass at the boundaries in logarithmic scale, and on the right the kinetic energy at the boundaries defined as in (2.5.4).

2.5.2 Test with Dirichlet boundary conditions in y

Test without control. In Figure 2.12 three snapshots of the dynamics at time $t = 5$, $t = 50$ and $t = 100$. In red the velocity vector field. The mass splits into two sub-masses that reach the boundaries and get trapped due to the fact that we impose Dirichlet boundary conditions in y.

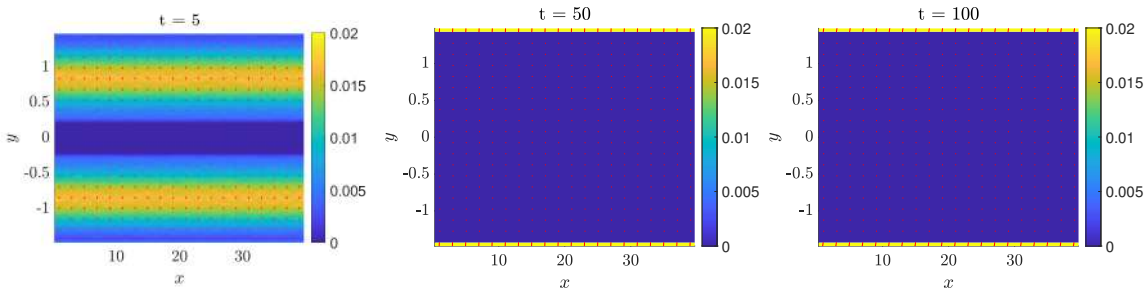

Figure 2.12: Test without control with $B = 1$. Three snapshots of the dynamics taken at time $t = 5$, $t = 50$ and $t = 100$, described in equation (2.2.2). In red the velocity vector field.

In Figure 2.13 we show on the left the mass percentages in time and on the right the kinetic energy at the boundaries as a function of time computed as in (2.5.2)-(2.5.3), and (2.5.4), respectively. Once that particles hits the boundaries, the energy increases.

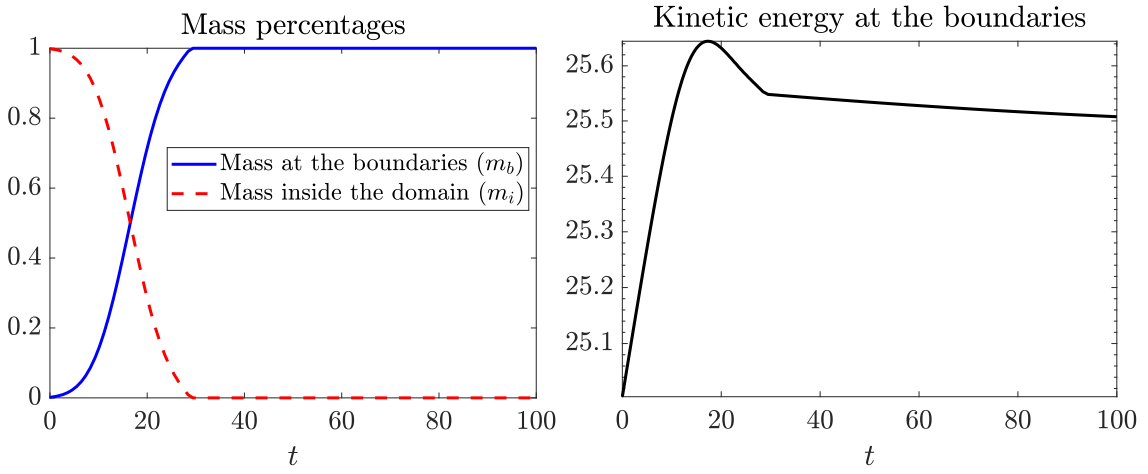

Figure 2.13: Test without control with $B = 1$. On the left the mass percentages defined as in equation (2.5.2)-(2.5.3) and on the right the kinetic energy at the boundaries defined as in (2.5.4).

Test with an instantaneous control of the particles velocity mean and variance. We consider the case of an instantaneous control of the particles velocity mean and variance derived as in (2.4.8) by setting $\alpha_{\mathrm{v}} = 1.5$, $\beta_{\mathrm{v}} = 0.1$, $\alpha_{\mathrm{x}} = \beta_{\mathrm{x}} = 0$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). We choose $\hat{v}_{y_1} = 5$ and $\hat{v}_{y_2} = -5$ as targets points in our control problem to guarantee that the velocity points toward the centre of the domain. In Figure 2.14 in the first row three snapshots of the dynamics described by equation (2.2.2) at time $t = 5$, $t = 50$ and $t = 100$. In red the velocity vector field. In the second row the correspondent value of the magnetic field. The mass is forced to move toward the centre of the domain.

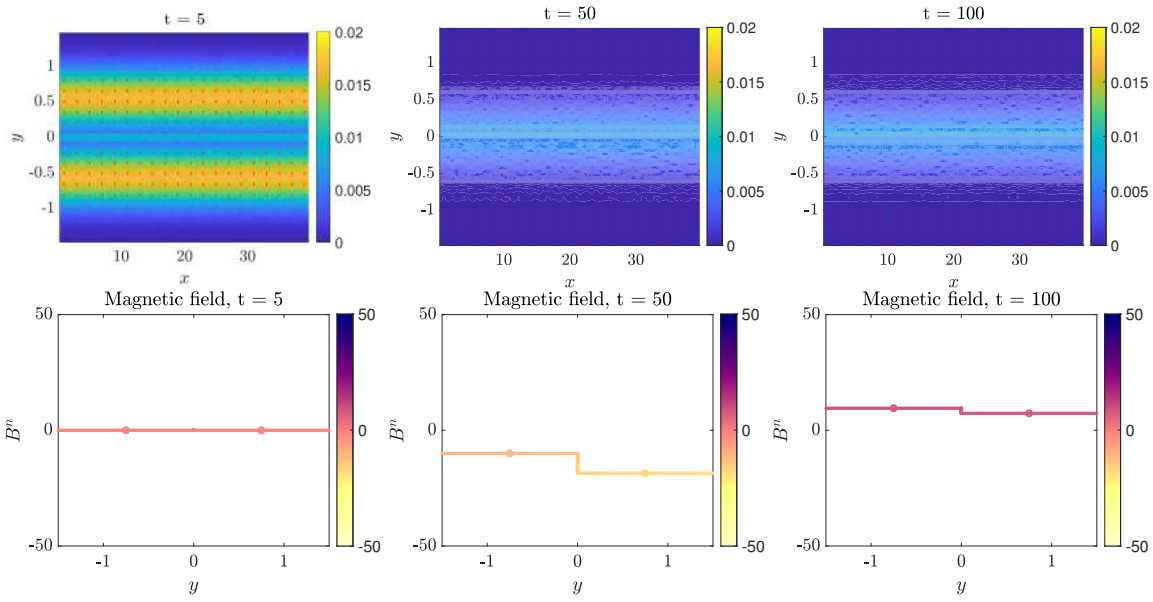

Figure 2.14: Test with an instantaneous control of the particles velocity mean and variance derived as in (2.4.8) by setting $\alpha_{\mathrm{v}} = 1.5$, $\beta_{\mathrm{v}} = 0.1$, $\alpha_{\mathrm{x}} = \beta_{\mathrm{x}} = 0$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). In the first row, three snapshots of the dynamics taken at time $t = 5$, $t = 50$ and $t = 100$, described in equation (2.2.2). In red the velocity vector field. In the second row, the value of the magnetic field in each cell.

In Figure 2.15 the value of the magnetic field in the cells C_k, $k = 1, 2$ as a function of time. The control takes small values in early times and it oscillate in late time.

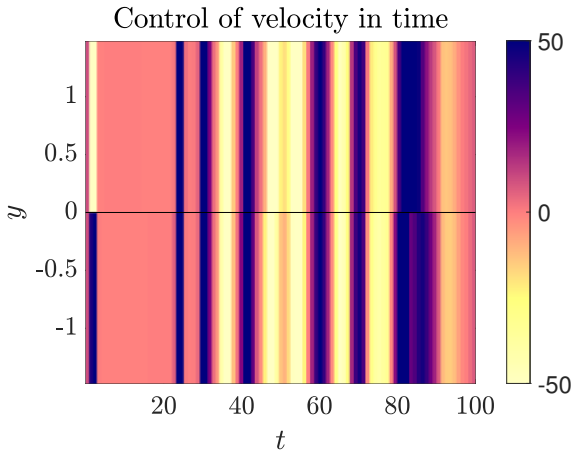

Figure 2.15: Test with an instantaneous control of the particles velocity mean and variance derived as in (2.4.8) by setting $\alpha_v = 1.5$, $\beta_v = 0.1$, $\alpha_x = \beta_x = 0$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). Value of the control B_k in each cell C_k, $k = 1, 2$ as a function of time.

In Figure 2.16 we plot the mass percentages in time, computed as in (2.5.2)-(2.5.3), and the kinetic energy at the boundaries as a function of time computed as in (2.5.4). Furthermore we zoom and plot in logarithmic scale the mass at the boundaries to highlight the fact that it does not hit the boundaries.

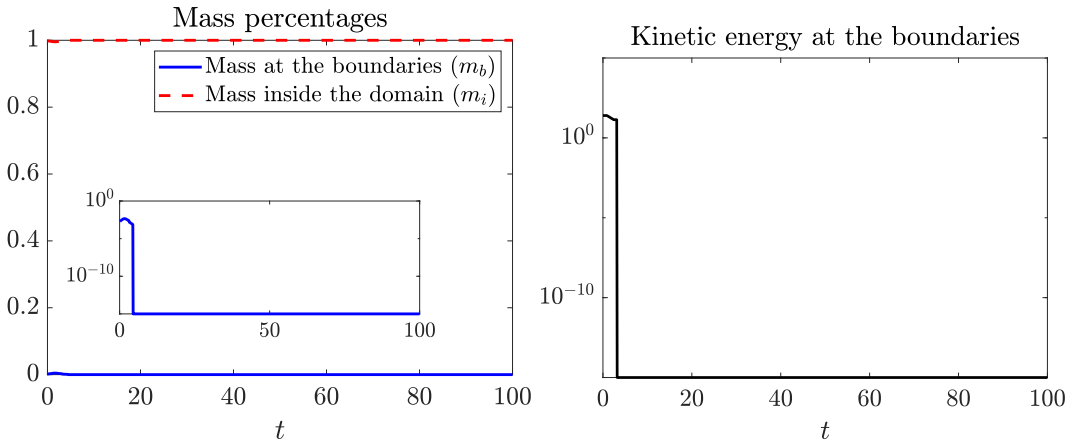

Figure 2.16: Test with an instantaneous control of the particles velocity mean and variance derived as in (2.4.8) by setting $\alpha_v = 1.5$, $\beta_v = 0.1$, $\alpha_x = \beta_x = 0$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). On the left the mass percentages defined as in equation (2.5.2)-(2.5.3), and the mass at the boundaries in logarithmic scale, and on the right the kinetic energy at the boundaries defined as in (2.5.4).

Test with an instantaneous control of the particles position and velocity mean and variance. We consider an instantaneous control of the particles position and velocity mean and variance derived as in (2.4.8) by setting $\alpha_{\mathrm{x}} = \alpha_{\mathrm{v}} = 1.5$, $\beta_{\mathrm{x}} = \beta_{\mathrm{v}} = 0.1$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). We choose $\hat{y}_k = 0$ for any $k = 1, 2$ and $\hat{v}_{y_1} = 5$, $\hat{v}_{y_2} = -5$ as target points, in such a way to force the mass to move toward the centre of the domain. In Figure 2.17 in the first row three snapshots of the dynamics described by equation (2.2.2) at time $t = 5$, $t = 50$ and $t = 100$. In red the velocity vector field. In the second row the correspondent value of the magnetic field. The mass is forced to move toward the centre of the domain.

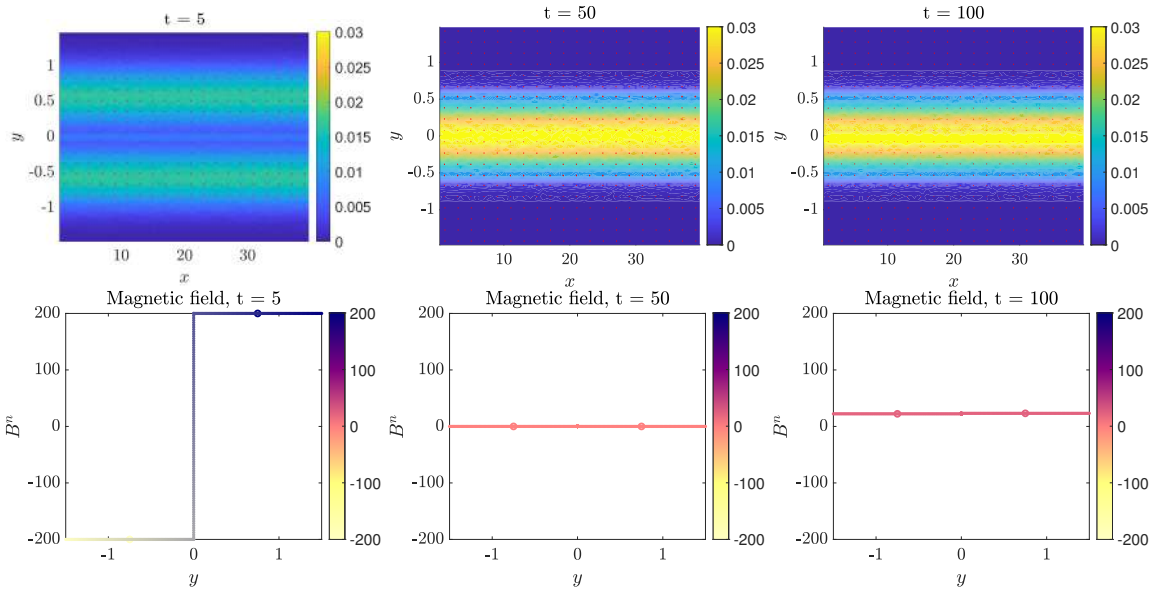

Figure 2.17: Test with an instantaneous control of the particles position and velocity mean and variance derived as in (2.4.8) by setting $\alpha_{\mathrm{x}} = \alpha_{\mathrm{v}} = 1.5$, $\beta_{\mathrm{x}} = \beta_{\mathrm{v}} = 0.1$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). In the first row, three snapshots of the dynamics taken at time $t = 10$, $t = 70$ and $t = 150$, described in equation (2.2.2). In red the velocity vector field. In the second row, the value of the magnetic field in each cell.

In Figure 2.18 the value of the magnetic field in the cells C_k, $k = 1, \ldots, N_k$ as a function of time. The control activates once that the mass moves far away from the centre of the domain.

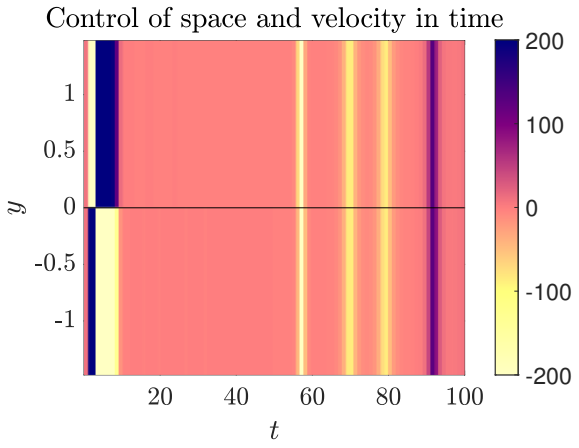

Figure 2.18: Test with an instantaneous control of the particles position and velocity mean and variance derived as in (2.4.8) by setting $\alpha_x = \alpha_v = 1.5$, $\beta_x = \beta_v = 0.1$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). Value of the control B_k in each cell C_k, $k = 1, 2$ as a function of time.

In Figure 2.19 we plot the mass percentages in time, computed as in (2.5.2)-(2.5.3), and the kinetic energy at the boundaries as a function of time computed as in (2.5.4). We add the plot of the mass at the boundaries in logarithmic scale. The behaviour is similar to the one of the previous test.

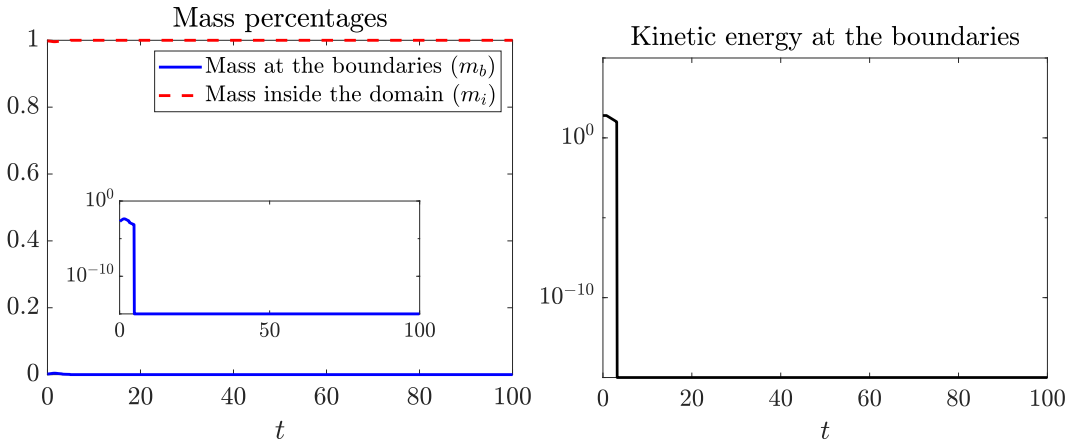

Figure 2.19: Test with an instantaneous control of the particles position and velocity mean and variance derived as in (2.4.8) by setting $\alpha_x = \alpha_v = 1.5$, $\beta_x = \beta_v = 0.1$, and $\gamma = 10^{-3}$, before scaling as in (2.4.10). On the left the mass percentages defined as in equation (2.5.2)-(2.5.3) and the mass at the boundaries in logarithmic scale, and on the right the total energy defined as in (2.5.4).

2.6 Numerical experiments: Kelvin-Helmholtz instability

Here we analyze the Kelvin-Helmholtz instability [72, 181, 54, 110] which is the analogues of the diocotron instability in an ideal fluid. We first show the evolution of the dynamics without control in a section of a $3D$ torus imposing periodic boundary conditions in x and Dirichlet boundary conditions in y, and assuming to have a strong magnetic field B acting on the orthogonal plane. Then we test the instantaneous control derived in Section 2.4. We consider the initial density

$$f_0(x,y) = \frac{1.5}{2\pi}\mathrm{sech}\Big(\frac{y}{0.9}\Big)(1+\epsilon_0\cos(3k_0x+\epsilon_1\sin(k_0x))), \tag{2.6.1}$$

with $k = 0.15$, $\epsilon_0 = 0.1$, $\epsilon_1 = 0.001$, $x \in [0, 40]$, $y \in [-5, 5]$, which is plotted in Figure 2.20.

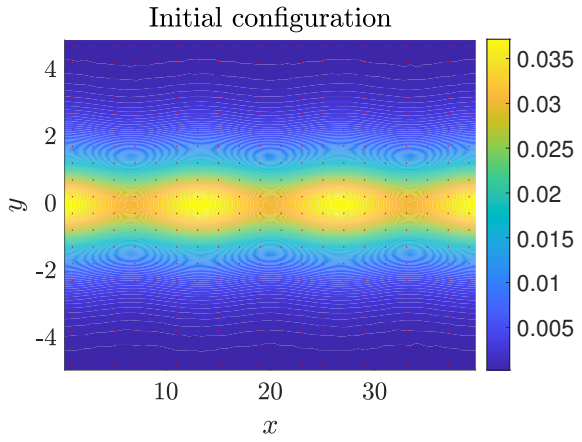

Figure 2.20: Approximation of the initial density.

We sample $N = 10^6$ particles from the initial density with position (x_i^0, y_i^0) and we assume each particle i s.t. $y_i^0 \geq 0$ to have velocity $(v_{x_i}^0, v_{y_i}^0) = (1, 0)$ and each particle i s.t. $y_i^0 < 0$ to have velocity $(v_{x_i}^0, v_{y_i}^0) = (-1, 0)$. We let the dynamics to evolve up to time $T = 100$, choosing $h = 0.1$ as time steps. We set $m_x \times m_y$, with $m_x = m_y = 64$, as number of cells for the space discretization and $m_{v_x} \times m_{v_y}$, with $m_{v_x} = m_{v_y} = 128$, number of cells in the velocity space. We then reconstruct the density at each time as in (2.3.1). We assume the control to be active in the whole domain which is divided into N_c cells C_k, $k = (j_x - 1)K_x + i_k$, with $i_x = 1, \ldots, K_x$ and $j_x = 1, \ldots, K_y$. In the following tests we set $K_x = 1$ and $K_y = 10$. As in the previous example, to test the efficiency of the control we compute the percentage of mass at the boundaries and inside the domain as in (2.5.2)-(2.5.3), and the energy as in (2.5.4) with $\Omega_b = [-5, -4.8] \cup [4.8, 5]$.

2.6.1 Test without control

We consider the case in which the external magnetic field is fixed to be $B = 1.5$. In Figure 2.21 three snapshots of the dynamics at time $t = 5$, $t = 10$ and $t = 100$. In red the velocity vector field. An instability occurs.

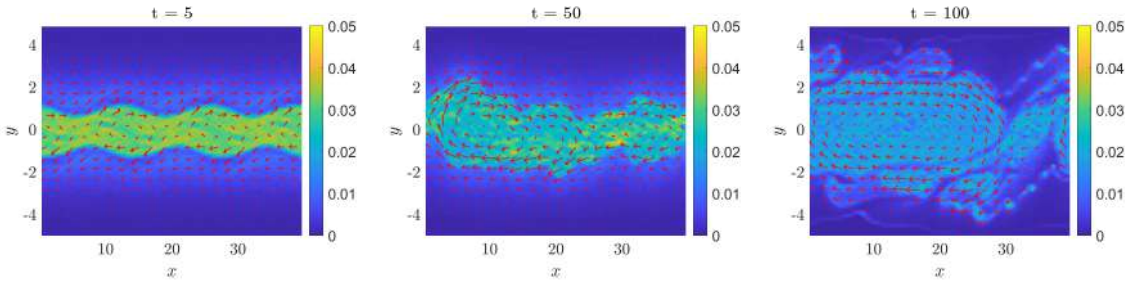

Figure 2.21: Test without control with $B = 1.5$. Three snapshots of the dynamics taken at time $t = 5$, $t = 50$ and $t = 100$, described in equation (2.2.2). In red the velocity vector field.

In Figure 2.22 we show on the left the mass percentages in time and on the right the kinetic energy at the boundaries as a function of time computed as in (2.5.2)-(2.5.3), and (2.5.4), respectively. We add also the plot of the mass at the boundaries in logarithmic scale. The mass hits the upper and lower boundary and the energy increases.

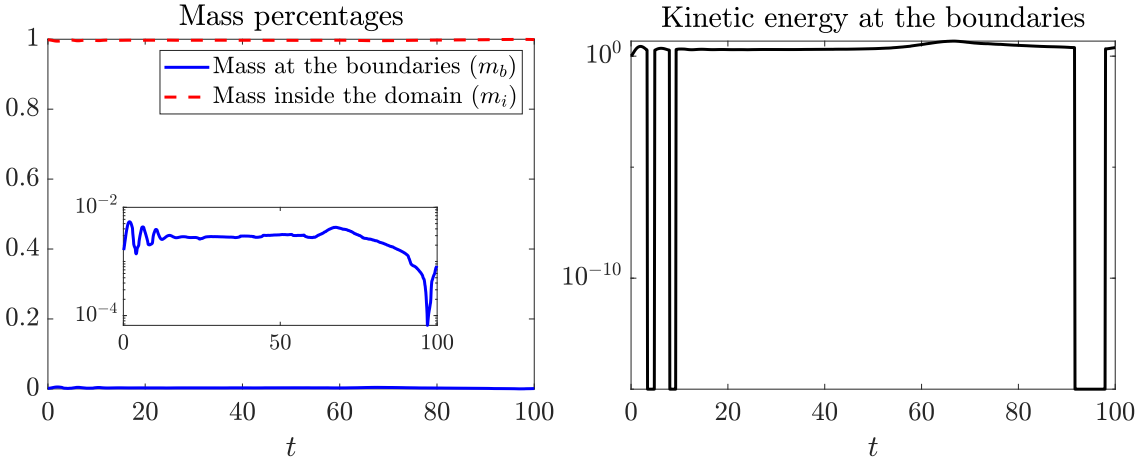

Figure 2.22: Test without control with $B = 1.5$. On the left the mass percentages defined as in equation (2.5.2)-(2.5.3) and the mass percentage at the boundaries in logarithmic scale, and on the right the kinetic energy at the boundaries defined as in (2.5.4).

2.6.2 Test with an instantaneous control of the particles velocity and position mean and variance

We now assume to derive an instantaneous control of the particles velocity and position mean and variance defined as in (2.4.8) by setting $\alpha_x = \alpha_v = 1.5$, $\beta_x = \beta_v = 0.1$, and $\gamma = 10^{-1}$, before scaling as in (2.4.10). We choose $\hat{y}_k = 0$, for any $k = 1, \dots, 10$, and $\hat{v}_{y_j} = 1$ for any $j = 1, \dots, 5$ and $\hat{v}_{y_j} = -1$ for any $j = 6, \dots, 10$ as target points to confine the mass at the centre of the domain. In Figure 2.23 in the first row three snapshots of the dynamics described by equation (2.2.2) at time $t = 5$, $t = 50$ and $t = 100$. In red the velocity vector field. The mass is confined in the centre of the domain. In the second row the correspondent value of the magnetic field.

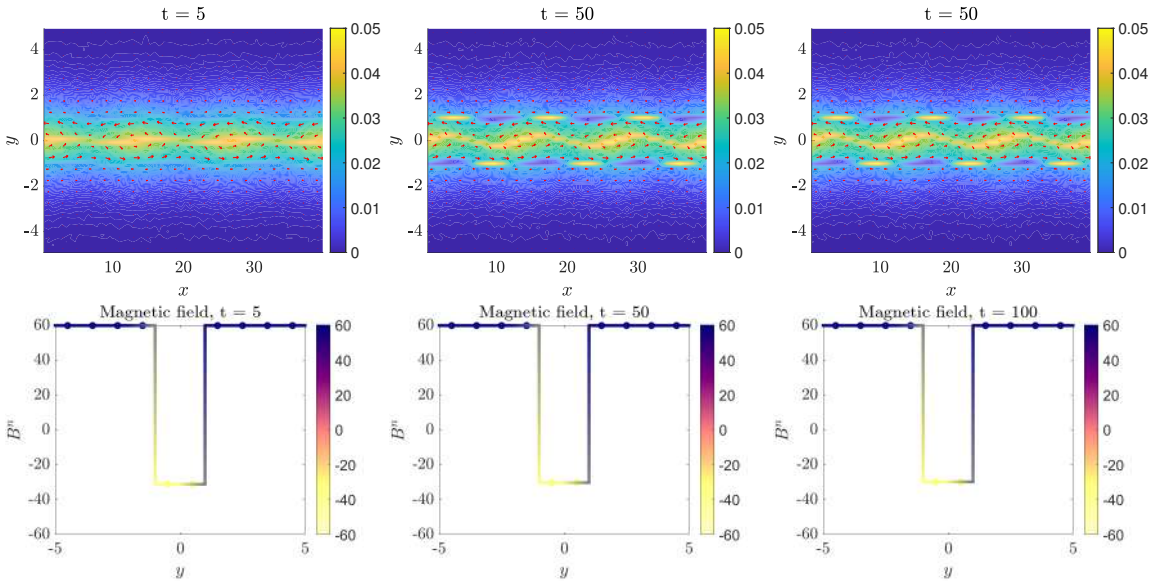

Figure 2.23: Test with a an instantaneous control of the particles velocity and position mean and variance defined as in (2.4.8) by setting $\alpha_x = \alpha_v = 1.5$, $\beta_x = \beta_v = 0.1$, before scaling as in (2.4.10). In the first row, three snapshots of the dynamics taken at time $t = 5$, $t = 50$ and $t = 100$, described in equation (2.2.2). In red the velocity vector field. In the second row, the value of the magnetic field in each cell.

In Figure 2.24 the value of the magnetic field in the cells C_k, $k = 1, \dots, 10$ as a function of time. The control takes positive values at the boundaries of the domain and negative values at the centre of the domain.

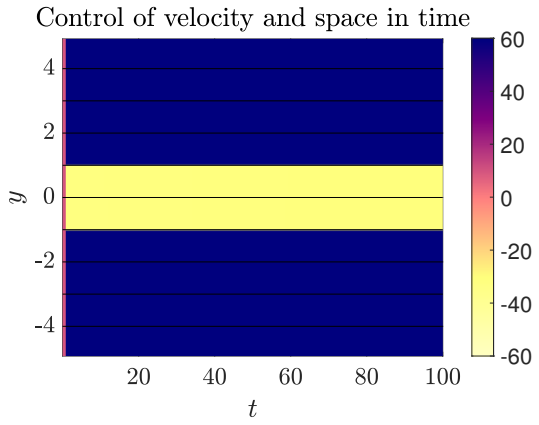

Figure 2.24: Test with an instantaneous control of the particles velocity and position mean and variance defined as in (2.4.8) by setting $\alpha_x = \alpha_v = 1.5$, $\beta_x = \beta_v = 0.1$, and $\gamma = 10^{-1}$, before scaling as in (2.4.10). Value of the control B_k in each cell C_k, $k = 1, \dots, 10$ as a function of time.

In Figure 2.25 we plot the mass percentages in time, computed as in (2.5.2)-(2.5.3) and the mass at the boundaries in logarithmic scale, and the kinetic energy at the boundaries as a function of time computed as in (2.5.4).

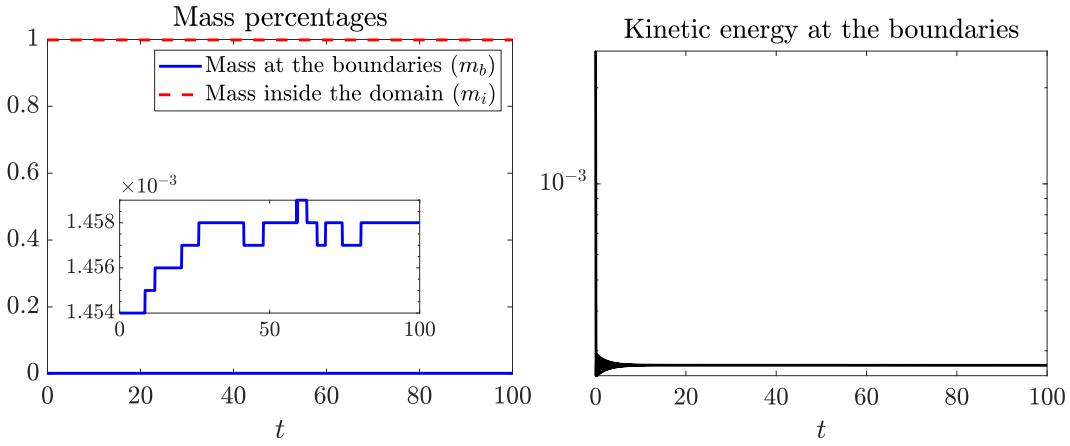

Figure 2.25: Test with an instantaneous control of the particles velocity and position mean and variance defined as in (2.4.8) by setting $\alpha_x = \alpha_v = 1.5$, $\beta_x = \beta_v = 0.1$, and $\gamma = 10^{-1}$, before scaling as in (2.4.10). On the left the mass percentages defined as in equation (2.5.2)-(2.5.3), and the mass percentage at the boundaries in logarithmic scale, and on the right the kinetic energy at the boundaries defined as in (2.5.4).

Finally, in Figure 2.26 on the left we compare the results in terms of kinetic energy at the

boundaries for different values of γ up to time $t = 3$. The energy decreases proportionally to γ. On the right the value of the kinetic energy at the boundaries up to time $t = 5$, assuming to have no control and assuming to activate the control at time $t^{act} = 0$, $t^{act} = 10$ and $t^{act} = 20$. The kinetic energy starts decreasing once that the control is switched on.

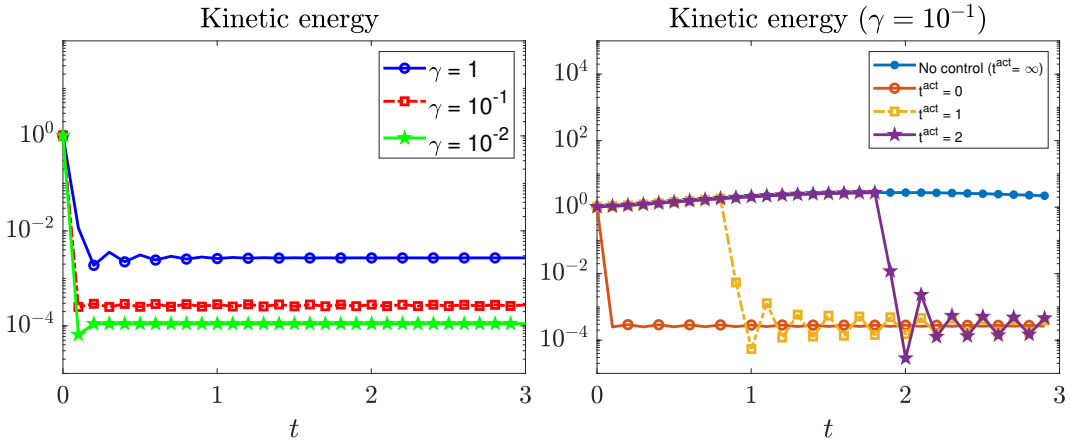

Figure 2.26: Test with an instantaneous control of the particles velocity and position mean and variance defined as in (2.4.8) by setting $\alpha_x = \alpha_v = 1.5$, $\beta_x = \beta_v = 0.1$, and $\gamma = 10^{-1}$, before scaling as in (2.4.10). On the left the value of the kinetic energy at the boundaries as γ varies. On the right the kinetic energy at the boundaries for $\gamma = 10^{-1}$ and assuming to activate the control at different times, where t^{act} denotes the activation time of the control.

Appendix 2.A Discretization of Poisson equation

In order to calculate the electric field, we solve the Poisson equation in the interval $[a_x, b_x] \times [a_y, b_y]$, assuming $\mathbf{x} = (x, y)$, with periodic boundary conditions in x and Dirichlet boundary conditions in y, that is

$$\begin{cases} -\Delta\phi(x,y) = \rho(x,y), & \text{for } (x,y) \in (a_x, b_x) \times (a_y, b_y), \\ \phi(x, a_y) = \phi(x, b_y) = 0, & \text{for } x \in [a_x, b_y], \\ \phi(a_x, y) = \phi(b_x, y), & \text{for } y \in [a_y, b_y], \end{cases} \tag{2.A.1}$$

We compute an approximation of the charge density $\rho(x, y)$ at time t in each cell C_k, $k = (j-1)m_x + i$, for $i = 1, \dots, m_x$ and $j = 1, \dots, m_y$. Secondly, we discretize equation (2.A.1) using central finite difference and we write for any $i = 1, \dots, m_x$, $j = 1, \dots, m_y$,

$$\begin{cases} -\frac{1}{\Delta_y^2}\left(\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}\right) - \frac{1}{\Delta_x^2}\left(\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}\right) = \rho_{i,j}, \\ \phi_{i,1} = \phi_{i,m_y} = 0, \quad \text{for } i = 1, \dots, m_x, \\ \phi_{1,j} = \phi_{m_x,j}, \quad \text{for } j = 1, \dots, m_y, \end{cases} \tag{2.A.2}$$

where

$$\phi_{i,j} = \phi(x_i, y_j), \qquad \rho_{i,j} = \rho(x_i, y_j), \tag{2.A.3}$$

for any $i = 1, \dots, m_x$ and $j = 1, \dots, m_y$. We introduce two discretization matrices

$$A_x = \frac{1}{\Delta_x^2}\begin{bmatrix} * & * & * & * & * & * \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & -2 & 1 \\ * & * & * & * & * & * \end{bmatrix}, \qquad A_y = \frac{1}{\Delta_y^2}\begin{bmatrix} * & * & * & * & * & * \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & -2 & 1 \\ * & * & * & * & * & * \end{bmatrix} \tag{2.A.4}$$

where, the first and last row of A_x and A_y should be opportunely modified to impose periodic and Dirichlet boundary conditions, respectively. Then we solve

$$-A\vec{\phi} = \vec{\rho}, \tag{2.A.5}$$

where we denote by $\tilde{\phi}_k = \phi_{i,j}$ each component of the vector $\vec{\phi}$ and by $\tilde{\rho}_k = \rho_{i,j}$ each component of the vector $\vec{\rho}$, for any $k = (j-1)m_x + i$, $i = 1 \dots, m_x$ and $j = 1, \dots, m_y$ and where

$$A = (Id_{m_y} \otimes A_x) + (A_y \otimes Id_{m_x}),$$

with Id_{m_x}, Id_{m_y} representing the identity matrix of size m_x and m_y, respectively. Once that we have computed the electric potential, we can compute the electric field which can be written in discretized form as

$$\vec{E}_x = -(Id_{m_x} \otimes C_x)\vec{\phi}, \qquad \vec{E}_y = -(C_y \otimes Id_{m_x})\vec{\phi}, \tag{2.A.6}$$

where C_x, C_y are discretization matrices for the first derivative. Finally, we associate to each particle the values $\vec{E}_x, \vec{E}_y$ which corresponds to their position over the grid.

Appendix 2.B Classical numerical tests

We consider here different classical tests (see for example [151, 96]). We assume to evolve the dynamics up to time T. We define an initial density, we approximate it with $N = 10^6$ particles and we solve the Vlasov-Poisson equation with the particle-in-cell method described in Section 2.3.

Linear Landau damping. We suppose to be in the one dimensional setting and we consider as initial condition

$$f_0(t, x, v) = \frac{1}{\sqrt{2\pi}}(1 + \alpha \cos(k_0 x))e^{-\frac{v^2}{2}}, \tag{2.B.1}$$

with $\alpha = 0.05$ and $k_0 = 0.5$, $x \in [0, 4\pi]$, $v \in [-6, 6]$. We apply periodic boundary conditions both in x and v. We choose a space discretization with $m_x = 32$ and $m_{v_x} = 128$ nodes. We evolve the dynamics up to time $T = 10$ with a time step $h = 0.1$. We set $B = 0$ in (2.2.2). In Figure 2.27 the evolution in time of the potential energy

$$\mathcal{E}_{pot}(t) = \sqrt{\int_{\mathbb{R}^d} E^2(x,t)dx}, \tag{2.B.2}$$

where $E(\cdot,\cdot)$ is the electric field in position x at time t. We observe the expected damping effect whose oscillations decreases exponentially with rate $\gamma_d = -0.1533$ as reported in [151].

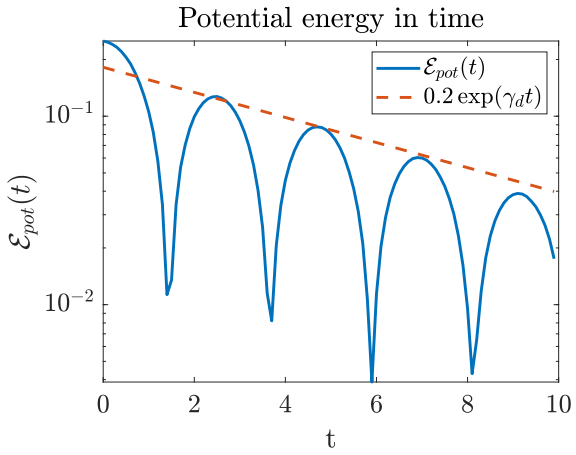

Figure 2.27: Linear Landau damping: potential energy as a function of time defined as in (2.B.2) and theoretical decrease of the oscillations with rate $\gamma_d = -0.1533$

Non linear Landau damping. We consider the initial condition (2.B.1) as in the linear case, with a larger perturbation $\alpha = 0.2$. We let the dynamics to evolve up to time $T = 60$ with a time step $h = 0.1$. In Figure 2.28, the time evolution of the potential energy (2.B.2). After an exponentially decreasing phase which occurs with rate $\gamma_d = -0.2920$ we observe an exponentially increasing phase which occurs with rate $\gamma_i = 0.085$. The decreasing and increasing rates are chosen as reported in [151].

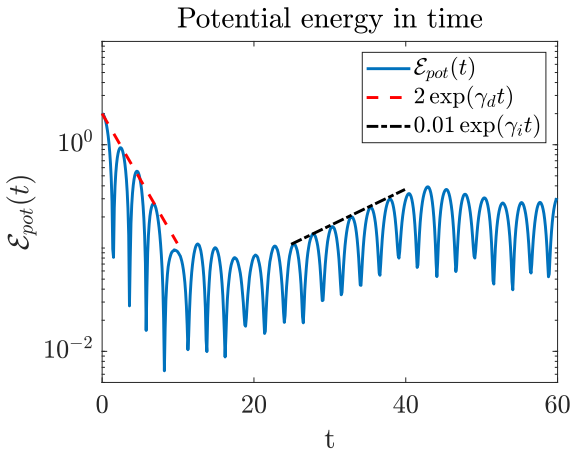

Figure 2.28: Non linear Landau damping: potential energy as a function of time defined as in (2.B.2), theoretical decay of the oscillations ($\gamma_d = -0.2920$) and theoretical increase of oscillations ($\gamma_i = 0.085$)

Two stream instability. We consider the $1D$ case both in space and in velocity where we impose periodic boundary conditions. We choose a space discretization with $m_x = 64$ and $m_{v_x} = 128$ nodes. We suppose $B = 0$ in the Vlasov-Poisson equation (2.2.2). We define the initial density to be

$$f_0(t,x,v) = \frac{1}{\sqrt{2\pi}}(1+\alpha\cos(k_0 x))\left(e^{-\frac{(v-2.4)^2}{2}} + e^{-\frac{(v+2.4)^2}{2}}\right), \tag{2.B.3}$$

with $\alpha = 3\times 10^{-3}$, $k_0 = 0.2$, $x \in [0, 2\pi/k_0]$, $v_x, v_y \in \mathbb{R}^{2d}$. We let the dynamics to evolve up to $T = 50$ with time step $h = 0.1$. In Figure 2.29 the initial configuration and in Figure 2.30 three snapshots of the evolution of the dynamics at time $t = 5$, $t = 20$, $t = 50$. We note that an instability occurs.

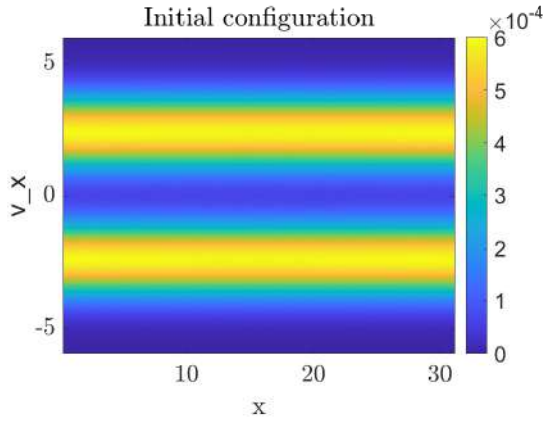

Figure 2.29: Approximation of the initial configuration.

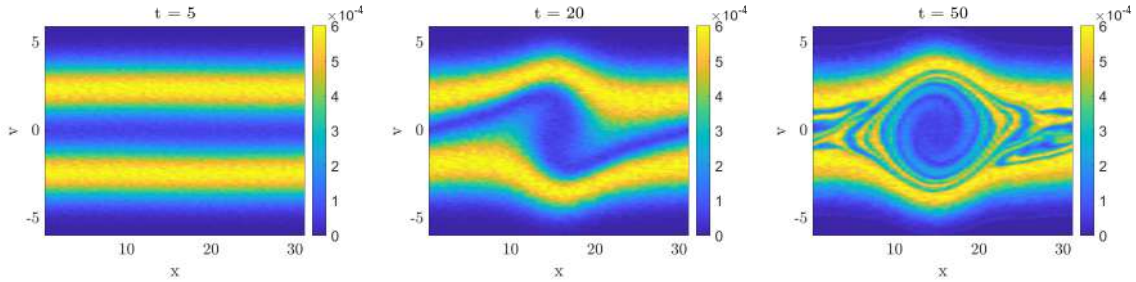

Figure 2.30: Two stream instability. Three snapshots of the dynamics taken at time $t = 5$, $t = 20$ and $t = 50$, described in equation (2.2.2) with $B = 0$.

In Figure 2.31 the potential energy defined as in (2.B.2) as a function of time. The energy increases exponentially with rate $\gamma_i = 0.2258$, in good agreement with the theoretical results presented in [151].

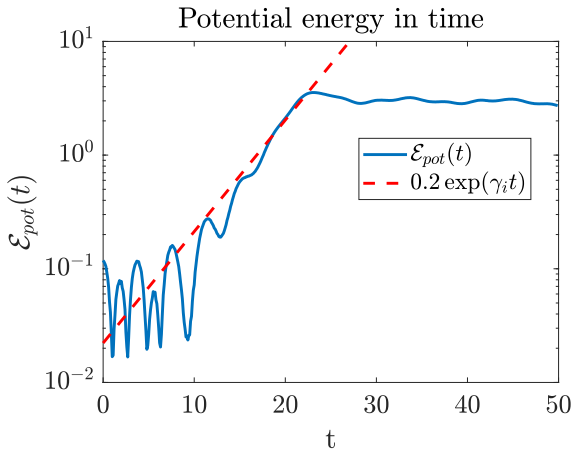

Figure 2.31: Two stream instability: potential energy as a function of time defined as in (2.B.2) and theoretical increase of the oscillations ($\gamma_i = 0.2258$).

Diocotron instability. We now focus on the solution of the Vlasov-Poisson equation with $B = 10$. We consider the $2D$ case in both space and velocity, with initial density

$$f_0(\mathbf{x},\mathbf{v}) = \frac{1}{2\pi}\left(1+\alpha\cos\left(k\arctan\left(\frac{y}{x}\right)\right)\right)e^{\left(-4\left(\sqrt{x^2+y^2}-6.5\right)^2\right)}e^{\left(-\frac{v_x^2+v_y^2}{2}\right)}, \qquad (2.B.4)$$

where $\alpha = 0.2$, $l = 7$, $(x,y) \in [-10,10]\times[-10,10]$, $v_x, v_y \in \mathbb{R}^{2d}$. We set Dirichlet boundary conditions both in x and in y. We let the dynamics to evolve up to time $T = 200$ with time step $h = 1$ performing $N_t = 2000$ iterations. In Figure 2.32 the initial configuration. In Figure 2.33 three snapshots of the dynamics taken at time $t = 50$, $t = 100$, $t = 200$. The presence of a strong magnetic field leads to the formation of vortices, as expected.

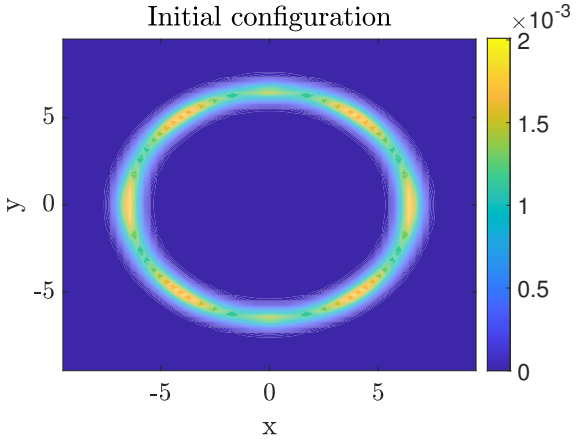

Figure 2.32: Approximation of the initial configuration.

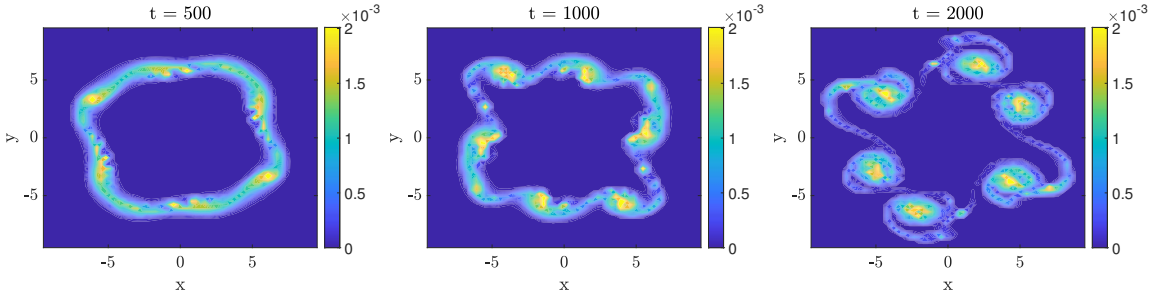

Figure 2.33: Diocotron instability. Three snapshots of the dynamics taken at time $t = 50$, $t = 100$ and $t = 200$, described in equation (2.2.2) with $B = 10$.

Chapter 3

Kinetic description of swarming dynamics with topological interaction and emergent leaders

In this chapter, we present a model describing the collective motion of birds. We explore the dynamic relationship between followers and leaders, wherein a select few agents, known as leaders, can initiate spontaneous changes in direction without being influenced by external factors like predators. Starting at the microscopic level, we develop a kinetic model that characterizes the behaviour of large crowds with transient leadership. One significant challenge lies in managing topological interactions, as identifying nearest neighbors in extensive systems can be computationally expensive. To address this, we propose a novel stochastic particle method to simulate the mesoscopic dynamics and reduce the computational cost of identifying closer agents from quadratic to logarithmic complexity using a k-nearest neighbours search algorithm with a binary tree. Lastly, we conduct various numerical experiments for different scenarios to validate the algorithm's effectiveness and investigate collective dynamics in both two and three dimensions.

The results presented in this chapter have been collected in [**?**] and they have been submitted for publication.

3.1 Introduction

In the last decades, there has been a notable surge in interest regarding the study of mathematical models describing collective behaviour of animals such as bacterial swarm [139], self-organization in insects [89, 35], bird flocking [73, 163, 14, 145], and fish schooling [127, 90]. This captivating area of investigation has garnered substantial interest, with researchers increasingly delving into the complexities of emergent behaviours exhibited by natural systems, but also spanned to a wider range of applications such as swarm of robots [129, 61, 136], as well as social sciences and economics, [12, 168, 185, 86], vehicular and pedestrian traffic [68, 2, 40].

These large ensemble of models incorporate rules governing the behaviour of individual entities within the system. By integrating such mechanisms, these models effectively

capture the impact of each entity on others, taking into account their relative positions and velocities. In this manuscript, our focus centres around the dynamics governing animal swarms, building upon the recent model proposed in [65]. This model introduces spontaneous changes of direction within the swarm, independent of external factors like predators, but rather influenced by a hierarchical interaction structure comprising two dynamic subpopulations labelled as leaders and followers. The key concept is that each bird possesses the potential to initiate turns and become a leader, consequently influencing its nearest neighbours who adopt follower status. This change of labels is characterized as a stochastic process, where each occurrence represents a random event. Our primary interest lies in exploring the phenomenon of *transient leadership*, wherein agents can alter their labels over time, as examined for example in [4, 143], and also at different scales in [142, 154, 23, 64]. Here, we examine a simplified version of the second-order stochastic differential equations presented in [65]. Specifically, we remove delay effects and consider that each agent (bird) can interact with a maximum of M nearest neighbours, consistent with observations in [14]. Our objective is to study such dynamics at the mesoscopic scale, formally introducing a kinetic model of the swarms with topological-type interaction dynamics and deriving the associated mean-field limit. For alternative mean-field and kinetic models with topological interactions, we refer to [32, 31, 123], and for rigorous derivations, we refer specifically to [79, 20]. Another primary objective of this study is to efficiently perform numerical simulations of high-dimensional non-local dynamics. One of the main challenges arises from the presence of topological-type interactions, necessitating the implementation of ad-hoc methods to reduce the complexity of the nearest neighbour search process. To address this computational burden at the mesoscopic scale, we introduce a novel stochastic simulation algorithm for the simulation of kinetic models such as, [162, 11, 47], in particular introducing the label switching feature, and adopting $k-$nearest neighbours search strategy, following the approach proposed in [107]. By implementing this method, we successfully reduce the computational complexity of the nearest neighbour search from quadratic to logarithmic scale, significantly enhancing the efficiency of numerical simulations.

The chapter is organized as follows. In Section 3.2 we introduce the microscopic model describing which are the forces that act on followers and on leaders. In Section 3.3, we extend the study to the kinetic level, describing the evolution of the densities and how the change of labels occurs. In Section 3.4, we introduce the algorithms that can be used to simulate the binary interaction rules and the change of labels. In Section 3.4.2 we perform two different validations experiments, testing both the accuracy and the efficiency of the numerical methods introduced. In particular, we show how it is possible to reduce the computational costs in dealing with non-locality. In Section 3.5 we simulate the dynamics at the microscopic and kinetic level for both the two and three dimensional cases.

3.2 Swarming models with leaders-followers dynamics

We consider a large system of N interacting agents represented by points moving in a d-dimensional space with an evolving hierarchy of interactions ruled by follower-leaders dynamics. For every $i = 1, \dots, N$, let $(x_i(t), v_i(t)) \in \mathbb{R}^{2d}$ denote position and velocity of the i-th agent at time t, with $d = 1, 2, 3$, and $\lambda_i(t) \in \Lambda \equiv \{0, 1\}$ the space of labels indicating at time t the status of agent i to be either *follower* (F) for $\lambda_i(t) = 0$, or *leader* (L) for $\lambda_i(t) = 1$. Moreover we account for N_{src} fixed target positions located at $x_k^{src} \in \mathbb{R}^d$ for $k = 1, \dots, N_{src}$, indicating positions of interested for the swarm such as nest, or foraging areas [22, 27].

We assume the system of agents evolving according to ODEs system,

$$\begin{aligned} \dot{x}_i &= v_i, \\ \dot{v}_i &= \frac{1}{M} \sum_{\{j \,:\, x_j \in \mathcal{B}_M(x_i;\mathbf{x})\}} \Big[A^{rep}(x_i, x_j) + \Big(A^{ali}(v_i, v_j) + A^{att}(x_i, x_j) \Big) (1 - \lambda_i(t)) \Big] \\ &\qquad + \big[A^{src}(x_i) + A^{ctr}(x_i) + S(v_i) \big] \lambda_i(t), \qquad i = 1, \dots, N, \end{aligned} \tag{3.2.1}$$

where we denoted by $\mathcal{B}_M(x_i; \mathbf{x})$ the ball centred at x_i, with $\mathbf{x} = (x_1, \dots, x_N)$, containing the M nearest neighbors to i-agent, assuming that in case of ambiguity, e.g. more than one agent is at the same distance from agent in position x_i, we select the first M agents giving priority according to the indexing order. Hence, the dynamics encodes different behaviours according to the value of the label $\lambda_i(t)$.

- For $\lambda_i(t) = 0$, we have follower-type interactions characterized by
  - repulsion force
  $$A^{rep}(x, x') = -C_{rep} \frac{x' - x}{\|x' - x\|^2}, \tag{3.2.2}$$
  - alignment force
  $$A^{ali}(v, v') = C_{ali} \left(v' - v \right), \tag{3.2.3}$$
  - and attraction force
  $$A_i^{att}(x, x') = C_{att} \left(x' - x \right), \tag{3.2.4}$$

  where $C_{rep} \geq 0$, $C_{ali} \geq 0$ and $C_{att} \geq 0$ are non-negative constants.
- For $\lambda_i(t) = 1$, we have leaders-type dynamics characterized by a repulsion force defined as in equation (3.2.2) and by a self-propulsion friction term $S(\cdot)$ defined as
$$S(v) = C_v(s - \|v\|^2)v, \tag{3.2.5}$$

where s is a given characteristic speed and $C_v \geq 0$. In presence of sources terms, leaders are driven by

$$A^{src}(x) = C_{src} \sum_{k=1}^{N_{src}} \varphi_\epsilon(\|x_k^{src} - x\|; \overline{r}) \frac{x_k^{src} - x}{\|x_k^{src} - x\|}, \tag{3.2.6}$$

where $C_{src} \geq 0$, x_k^{src} denotes the position of the attraction source (nest, or food) and $\varphi_\epsilon(\cdot)$ is a sigmoid function of the following type

$$\varphi_\epsilon(r; \overline{r}) := \frac{1}{1 + \exp\{(r - \overline{r})/\epsilon\}}, \tag{3.2.7}$$

with regularization parameter $\epsilon > 0$, modelling a perception area around the source activating when the distance of the agent is below the threshold value $\overline{r} > 0$. Furthermore, leaders can be forced to move toward the centre of mass x_c according to the force

$$A^{ctr}(x) = C_{ctr}\Big(1 - \varphi_\epsilon(\|x_c - x\|; \underline{r})\Big) \frac{x_c - x}{\|x_c - x\|}, \tag{3.2.8}$$

where $C_{ctr} \geq 0$, and when the distance with respect to the centre of mass is larger than $\underline{r}$.

3.2.1 Stochastic process for leaders emergence

Agents have the ability to switch between being leaders and followers, and vice versa. Such a change in status is treated as a stochastic process, where each occurrence represents a random event governed by an assigned probability distribution. Each event is associated with a transition rate, which quantifies the probability of its occurrence per unit time. Therefore, for $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_N)$, each label $\lambda_i(t)$ will follow a jump process in this manner

- if $\lambda_i(t) = 1$ then it switches to 0 with rate $\pi_{L\to F}(t, x_i, v_i, \lambda_i; \mathbf{x}, \boldsymbol{v}, \boldsymbol{\lambda})$,
- if $\lambda_i(t) = 0$ then it switches to 1 with rate $\pi_{F\to L}(t, x_i, v_i, \lambda_i; \mathbf{x}, \boldsymbol{v}, \boldsymbol{\lambda})$,

where the transition rates $\pi_{F\to L}(\cdot), \pi_{L\to F}(\cdot)$ in general are non-linear functions of the state of the system. In what follows we will consider different choices for the labels' switching rules, ranging from random, density dependent and aiming at organizing agents toward a common target. These choices will be detailed in Section 3.3.2.

3.3 Kinetic modelling of swarming dynamics

In this section, we will provide a kinetic description of the swarming model with leader emergence and topological interaction, we refer to [12, 3, 143] for related studies in the context of kinetic models.

Thus, we associate to each agent a position and velocity $(x, v) \in \mathbb{R}^d \times \mathbb{R}^d$ and a leadership-level λ, as a discrete binary variable in the label space $\Lambda = \{0, 1\}$. We are interested in the evolution of the probability density function

$$f = f(x, v, \lambda, t), \qquad f : \mathbb{R}^d \times \mathbb{R}^d \times \{0, 1\} \times \mathbb{R}_+ \to \mathbb{R}_+ \tag{3.3.1}$$

where $t \in \mathbb{R}^+$ denotes as usual the time variable. For each time $t \geq 0$, $\lambda \in \{0, 1\}$, we have the following marginal density

$$\rho(\lambda, t) = \int_{\mathbb{R}^d \times \mathbb{R}^d} f(x, v, \lambda, t) d(x, v), \tag{3.3.2}$$

which defines the quantity of agents with label λ at time t. In the sequel, we will assume that the total number of agents is conserved, namely

$$\rho(1, t) + \rho(0, t) = 1. \tag{3.3.3}$$

Likewise, we define the marginal density for agents in space and velocity

$$g(x, v, t) = \sum_{\lambda} f(x, v, \lambda, t), \qquad \lambda \in \{0, 1\}. \tag{3.3.4}$$

Next, we assume the density $f(x, v, \lambda, t)$ to be solution of a kinetic equation accounting for pairwise interactions among agents, and for labels transition.

Notational convention. To ease the writing, we will use an equivalent notation for functions depending on λ, where we introduce the indexing given by the discrete label space Λ, as follows

$$F_\lambda(\cdot) := F(\cdot, \lambda).$$

Then, for example, the density $f(x, v, t, \lambda)$ will be denoted by $f_\lambda(x, v, t)$ or the mass $\rho(\lambda, t)$ by $\rho_\lambda(t)$.

3.3.1 Povzner-Boltzmann-type model

We assume that each agent modifies its velocity through a binary interaction occurring with an other agent within the topological ball $B_{r^*}(x, t)$, the ball centred in x whose radius is defined, for a fixed $t \geq 0$, by the following variational problem

$$r^*(x, t) = \arg\min_{\alpha > 0} \left\{ \sum_{\lambda} \int_{B_\alpha(x,t) \times \mathbb{R}^d} f_\lambda(x_*, v_*, t) dx_* dv_* \geq \rho^* \right\}, \tag{3.3.5}$$

where $\rho^* \in (0, 1]$ is the target topological mass, namely the ratio $\rho^* = M/N$ associated to the microscopic model (3.2.1).

Hence, we consider pairwise interactions among an agent with state $(x, v, \lambda) \in \mathbb{R}^{2d} \times \{0, 1\}$ and $(x_*, v_*, \lambda_*) \in B_{r^*}(x, t) \times \mathbb{R}^d \times \{0, 1\}$, where the post-interaction velocities are given by

$$\begin{cases} v' &= v + \alpha \mathcal{F}_\lambda(x, x_*, v, v_*), \\ v'_* &= v_*, \end{cases} \tag{3.3.6}$$

where $v, v_* \in \mathbb{R}^d$ denote the pre-interaction velocities and v', v'_* the velocities after the exchange of information between the two agents. In (3.3.6) we assume

$$\begin{aligned} \mathcal{F}_\lambda(x, x_*, v, v_*) = A^{rep}(x, x_*) + \big[A^{ali}(v, v_*) + A^{att}(x, x_*)\big](1 - \lambda) \\ + \big[A^{src}(x) + A^{ctr}(x) + S(v)\big]\lambda. \end{aligned} \tag{3.3.7}$$

For $\lambda \in \{0, 1\}$, the evolution in time of the density function $f_\lambda(x, v, t)$ is described by a integro-differential equation of the Povzner-Boltzmann type [167, 102] as follows

$$\partial_t f_\lambda(x, v, t) + v \cdot \nabla_x f_\lambda(x, v, t) - \mathcal{T}_\lambda[f](x, v, t) = Q_\lambda(f, f)(x, v, t), \tag{3.3.8}$$

where $\mathcal{T}_\lambda[f](\cdot)$ accounts for the evolution of the agents in the discrete label space and $Q_\lambda(\cdot, \cdot)$ is the interaction operator defined as follows

$$Q_\lambda(f, f)(x, v, t) = \eta \sum_{\lambda_*} \int_{B_{r^*(x,t)} \times \mathbb{R}^d} \left(\frac{1}{J_\lambda} f_\lambda(x, {}'v, t) f_{\lambda_*}(x_*, {}'v_*, t) - f_\lambda(x, v, t) f_{\lambda_*}(x_*, v_*, t) \right) d(x_*, v_*), \tag{3.3.9}$$

where $({}'v, {}'v_*)$ are the pre-interaction velocities, and the term J_λ denotes the Jacobian of the transformation $(v, v_*) \to (v', v'_*)$ with (v', v'_*) the post-interaction velocities, and $\eta > 0$ is a constant relaxation rate representing the interaction frequency.

3.3.2 Master equation for leaders transition

In the previous section, we have introduced the transition operator $\mathcal{T}_\lambda[f](x, v, t) = \mathcal{T}[f](x, v, \lambda, t)$ characterizing the evolution of the agents in the discrete space of labels $\Lambda = \{0, 1\}$ (leaders/followers). Such operator is defined as follows

$$\begin{aligned} \mathcal{T}_0[f](x, v, t) =& \pi_{L \to F} f_1(x, v, t) - \pi_{F \to L} f_0(x, v, t), \\ \mathcal{T}_1[f](x, v, t) =& \pi_{F \to L} f_0(x, v, t) - \pi_{L \to F} f_1(x, v, t), \end{aligned} \tag{3.3.10}$$

where $\pi_{F \to L} := \pi_{F \to L}(x, v, t; f)$ and $\pi_{L \to F} := \pi_{L \to F}(x, v, t; f)$ are certain transition rates.

Thus the evolution of the transition process of labels can be described by the evolution equation for $\rho_\lambda(t) = \rho(\lambda, t)$,

$$\frac{d}{dt} \rho_\lambda(t) - \int_{\mathbb{R}^{2d}} \mathcal{T}_\lambda[f](x, v, t) \, d(x, v) = 0. \tag{3.3.11}$$

From the definition of the transition operator $\mathcal{T}_\lambda[\cdot]$ and (3.3.3) it follows the conservation of the mass,

$$\frac{d}{dt}\sum_\lambda \rho_\lambda(t) = \sum_\lambda \int_{\mathbb{R}^{2d}} \mathcal{T}_\lambda[f](x,v,t)\, d(x,v) = 0. \tag{3.3.12}$$

In the sequel we list possible choices of transition rates in (3.3.10).

Constant rates. Leaders emerge with rate $q_{FL} > 0$ and return to the followers status with rate $q_{LF} > 0$. Hence, the transition rates write as follows

$$\pi_{L\to F} = q_{LF}, \qquad \pi_{F\to L} = q_{FL}. \tag{3.3.13}$$

Thus, we can rewrite equation (3.3.11) as

$$\begin{aligned} \partial_t \rho_1(t) &= q_{FL}\rho_1(t) - q_{LF}\rho_0(t), \\ \partial_t \rho_0(t) &= q_{LF}\rho_1(t) - q_{FL}\rho_0(t), \end{aligned} \tag{3.3.14}$$

and find the stationary solution of equation (3.3.14) that is

$$\rho_1^\infty = \frac{q_{FL}}{q_{LF}+q_{FL}}, \qquad \rho_0^\infty = \frac{q_{LF}}{q_{LF}+q_{FL}}. \tag{3.3.15}$$

Density-dependent rates. Leaders emerge with higher probability where the followers density is higher and the leaders one is lower and they return to the followers status with higher probability if the followers concentration around them is lower, similarly to [1]. The transition rates reads

$$\pi_{L\to F} = q_F\ (1 - \mathcal{D}_F[f](x,t)), \qquad \pi_{F\to L} = q_L\ (1 - \mathcal{D}_L[f](x,t)), \tag{3.3.16}$$

where q_F, q_L are constant parameters and the functions $\mathcal{D}_F[f](x,t)$ and $\mathcal{D}_L[f](x,t)$ represent the concentration of leaders and followers in position x and are defined as

$$\begin{aligned} \mathcal{D}_F[f](x,t) &= S_F(t)\int_{\mathbb{R}^{2d}} e^{-\frac{|x-y|^2}{\delta^2}} f_0(y,w)d(y,w), \\ \mathcal{D}_L[f](x,t) &= S_L(t)\int_{\mathbb{R}^{2d}} e^{-\frac{|x-y|^2}{\delta^2}} f_1(y,w)d(y,w), \end{aligned} \tag{3.3.17}$$

with $S_F(t)$, $S_L(t)$ normalization constants to ensure that the above quantities are bounded by one and with $\delta > 0$.

Target-oriented rates. Leaders emerge when their direction is oriented in the correct direction toward a target position, $\bar{x}$, such as the nesting or foraging area. We consider the following rates

$$\pi_{F\to L} = \begin{cases} 0, & \text{if } \alpha(x,v,t;f) < \overline{\alpha}, \\ 1, & \text{if } \alpha(x,v,t;f) \geq \overline{\alpha}, \end{cases} \qquad \pi_{L\to F} = \begin{cases} 0, & \text{if } \alpha(x,v,t;f) \geq \underline{\alpha}, \\ 1, & \text{if } \alpha(x,v,t;f) < \underline{\alpha}, \end{cases} \tag{3.3.18}$$

with $\underline{\alpha}, \overline{\alpha} \in [-1,1]$ and

$$\alpha(x,v,t;f) = \cos\left(\angle\left(\bar{x}-x, \mathcal{G}[f](x,v,t)\right)\right), \tag{3.3.19}$$

with $\angle(\cdot,\cdot)$ denoting the angle between two vectors. The functional $\mathcal{G}[f](\cdot)$ accounts for the directional information of agents according to

$$\mathcal{G}[f](x,v,t) = S(v) - \mathcal{X}_c[f](x,t) - \mathcal{V}_c[f](x,v,t), \tag{3.3.20}$$

where $S(v)$ is the self-propulsion term, and the terms $\mathcal{X}_c[f](\cdot), \mathcal{V}_c[f](\cdot)$ account for the average influence induced by neighbours as follows

$$\mathcal{X}_c[f](x,t) = \int_{B_{r^*(x,t)}\times\mathbb{R}^d} A^{att}(x,x_*) f_\lambda(x_*,v_*,t) dx_* dv_*,$$
$$\mathcal{V}_c[f](x,v,t) = \int_{B_{r^*(x,t)}\times\mathbb{R}^d} A^{ali}(v,v_*) f_\lambda(x_*,v_*,t) dx_* dv_*,$$

with $A^{ali}(\cdot,\cdot)$, $A^{att}(\cdot,\cdot)$ defined as in equation (3.2.3)-(3.2.4). Note that in (3.3.20), when the term $\mathcal{G}[f]$ is partially aligned with the target direction $\bar{x}-x$, i.e., $\alpha(x,v,t;f) \geq \overline{\alpha}$, agents switch to, or remain in, leader status, naturally steering their dynamics towards the target $\bar{x}$. Conversely, if $\alpha(x,v,t;f) \leq \underline{\alpha}$, the agent with position and velocity (x,v) remains in, or is switched to, follower status. Figure 3.1 illustrates two possible configurations.

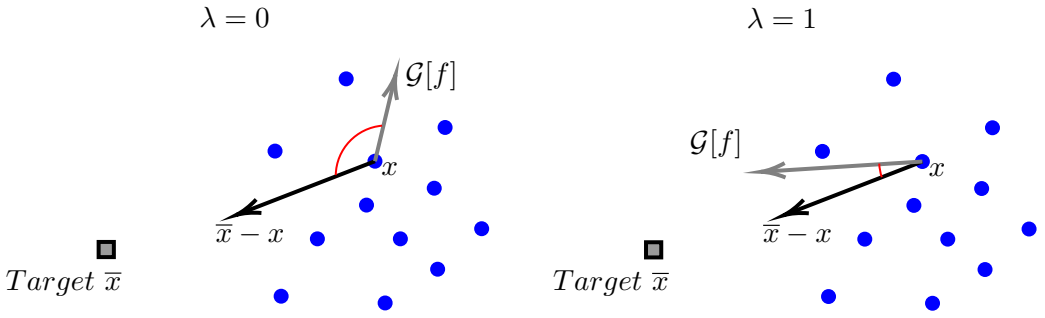

Figure 3.1: On the left the case in which agent x remains or switches to follower status ($\lambda = 0$), and on the right the case in which it remains or switches to leader status ($\lambda = 1$).

Remark 4 (Multiple-label case and continuous limit)**.** We observe that the previous formulation can be extended to include multiple levels of leadership, up to a continuous space of labels [64, 4]. Hence, we consider $\lambda \in \Lambda = \{\lambda_1, \dots, \lambda_{N_\ell}\}$ such that $\lambda_k = k\Delta\lambda$ with $\lambda_1 = 0$ and $\lambda_{N_\ell} = 1$. The transition operator $\mathcal{T}_k[\cdot]$ in the multiple-label case for $k = 2, \dots, N_\ell - 1$ reads

$$\mathcal{T}_k[f](t) = \Big(\mathcal{T}^+_{k+1}[f](t) - \mathcal{T}^+_k[f](t)\Big) - \Big(\mathcal{T}^-_k[f](t) - \mathcal{T}^-_{k-1}[f](t)\Big), \tag{3.3.21}$$

and for the boundary values $\lambda_1 = 0, \lambda_{N_\ell} = 1$ we have

$$\begin{aligned} \mathcal{T}_1[f](t) &= \mathcal{T}^+_2[f](t) - \mathcal{T}^+_1[f](t), \\ \mathcal{T}_{N_\ell}[f](t) &= \mathcal{T}^-_{N_\ell}[f](t) - \mathcal{T}^-_{N_\ell - 1}[f](t). \end{aligned} \tag{3.3.22}$$

In the above expressions

$$\mathcal{T}^+_k[f](t) := \pi_{\lambda_k \to \lambda_{k-1}} f_k(t), \qquad \mathcal{T}^-_k[f](t) = \pi_{\lambda_k \to \lambda_{k+1}} f_k(t),$$

where we denoted by $\pi_{\lambda_m \to \lambda_n} := \pi_{\lambda_m \to \lambda_n}(x, v, t; f)$ the transition rates from the state λ_m to the state λ_n, and we used the shorten notation for the density $f_k(t) := f(x, v, \lambda_k, t)$ and the transition operator $\mathcal{T}_k[f](t) := \mathcal{T}[f](x, v, \lambda_k, t)$. Hence, the evolution of the density in the label space is ruled by

$$\frac{d}{dt} f_k(t) = \mathcal{T}_k[f](t), \quad k = 1, \dots, N_\ell. \tag{3.3.23}$$

Furthermore, from (3.3.21) we can retrieve a transition operator for a continuous label space, $\lambda \in [0, 1]$ by scaling the time by $1/\Delta\lambda$ and by considering the limit for $N_\ell \to \infty$ and $\Delta\lambda \to 0$ in equation (3.3.23),

$$\partial_t f_\lambda(t) = \partial_\lambda \Big[\mathcal{T}^+_\lambda[f](t) - \mathcal{T}^-_\lambda[f](t)\Big], \tag{3.3.24}$$

for any $\lambda \in [0, 1]$ where

$$\mathcal{T}^+_\lambda[f](t) = \lim_{\Delta\lambda \to 0} \frac{\mathcal{T}^+_{k+1}[f](t) - \mathcal{T}^+_k[f](t)}{\Delta\lambda}, \qquad \mathcal{T}^-_\lambda[f](t) = \lim_{\Delta\lambda \to 0} \frac{\mathcal{T}^-_k[f](t) - \mathcal{T}^-_{k-1}[f](t)}{\Delta\lambda}.$$

Since at the boundary we have no inflow and outflow of mass, thanks to (3.3.22), we have

$$\partial_\lambda \Big[\mathcal{T}^+_\lambda[f](t) - \mathcal{T}^-_\lambda[f](t)\Big] = 0, \qquad \lambda \in \{0, 1\}. \tag{3.3.25}$$

Finally, we can write the master equation for the density integrating (3.3.24) as follows

$$\partial_t \rho(\lambda, t) = \partial_\lambda \int_{\mathbb{R}^{2d}} \Big[\mathcal{T}^+_\lambda[f](x, v, t) - \mathcal{T}^-_\lambda[f](x, v, t)\Big] d(x, v), \tag{3.3.26}$$

where, for transition operators of type $\mathcal{T}^\pm_\lambda[f](x, v, t) = \kappa^\pm(\lambda) f_\lambda(x, v, t)$, we retrieve the transport equation for the density $\rho_\lambda(t)$ in the label space in the following form

$$\partial_t \rho_\lambda(t) = \partial_\lambda \Big[\big(\kappa^+(\lambda) - \kappa^-(\lambda)\big)\, \rho_\lambda(t)\Big]. \tag{3.3.27}$$

3.3.3 Mean-field asymptotics

In order to retrieve asymptotic behaviour of the Boltzmann-type equation (3.3.8), we resort on a mean-field approximation of the interaction dynamics. Thus we introduce a grazing collision limit for the interaction operator (3.3.9), following the approach in [162, 50]. Thus, we rescale the interaction frequency η and the interaction propensity α to maintain asymptotically the memory of the microscopic interactions, as follows

$$\alpha = \varepsilon, \qquad \eta = \frac{1}{\varepsilon}, \tag{3.3.28}$$

for $\varepsilon > 0$, which corresponds to the case where the interaction kernel concentrates on binary interactions producing very small changes in the agents velocity but at the same time the number of interactions becomes very large. For now on, for simplicity we remove the dependence on time t. We introduce the test function $\psi(x,v) \in C_0^1(\mathbb{R}^d \times \mathbb{R}^d)$ and we write the weak form of the scaled kinetic equation (3.3.8)

$$\begin{aligned} &\int_{\mathbb{R}^{2d}} \left(\partial_t f_\lambda(x,v) + v \cdot \nabla_x f_\lambda(x,v)\right) \psi(x,v) d(x,v) - \int_{\mathbb{R}^{2d}} \mathcal{T}_\lambda[f](x,v)\psi(x,v)d(x,v) = \\ &\frac{1}{\varepsilon} \sum_{\lambda_*} \int_{\mathbb{R}^{2d}} \int_{B_{r_*(x,t)}\times\mathbb{R}^d} \left(\psi(x,v') - \psi(x,v)\right) f_{\lambda_*}(x_*,v_*) f_\lambda(x,v) d(x,v) d(x_*,v_*), \end{aligned} \tag{3.3.29}$$

with scaled interactions (3.3.6) as follows

$$v' - v = \varepsilon \mathcal{F}_\lambda(x, x_*, v, v_*). \tag{3.3.30}$$

Since as $\varepsilon \to 0$, we have $v' \to v$ we can expand $\psi(x,v')$ in Taylor series centred in (x,v) up to second order and rewrite the right hand side of equation (3.3.29) as

$$\begin{aligned} &\frac{1}{\varepsilon} \sum_{\lambda_*} \int_{\mathbb{R}^{2d}} \int_{B_{r_*(x,t)}\times\mathbb{R}^d} \left(\psi(x,v') - \psi(x,v)\right) df_{\lambda_*} df_\lambda = \\ &\frac{1}{\varepsilon} \sum_{\lambda_*} \int_{\mathbb{R}^{2d}} \int_{B_{r_*(x,t)}\times\mathbb{R}^d} \nabla_v \psi(x,v) \cdot (v' - v) df_* df + R(\varepsilon), \end{aligned} \tag{3.3.31}$$

where we used the shorten notation $df_{\lambda_*} = f(x_*, v_*, \lambda_*) d(x_*, v_*)$, $df_\lambda = f(x,v,\lambda) d(x,v)$, and where $R(\varepsilon)$ indicates the remainder which is given by

$$R(\varepsilon) = \frac{1}{2\varepsilon} \sum_{\lambda_*} \int_{\mathbb{R}^{2d}} \int_{B_{r_*(x,t)}\times\mathbb{R}^d} \left[\sum_{i,j=1}^{d} \partial_v^{(i,j)} \psi(x,\bar{v}) (v' - v)_i (v' - v)_j \right] df_{\lambda_*} df_\lambda, \tag{3.3.32}$$

with

$$\bar{v} = \gamma v + (1 - \gamma) v',$$

for some $\gamma \in [0,1]$. Therefore, the scaled binary interaction term (3.3.31) reads

$$\sum_{\lambda_*} \int_{\mathbb{R}^{2d}} \int_{B_{r_*(x,t)}\times\mathbb{R}^d} \nabla_v \psi(x,v) \cdot \mathcal{F}_\lambda(x,x_*,v,v_*) df_{\lambda_*} df_\lambda + R(\varepsilon). \tag{3.3.33}$$

Integrating equation (3.3.33) by parts and taking the limit $\varepsilon \to 0$ we have

$$\begin{aligned} &\sum_{\lambda_*} \int_{\mathbb{R}^{2d}} \int_{B_{r_*(x,t)}\times\mathbb{R}^d} \nabla_v \psi(x,v) \cdot \mathcal{F}_\lambda(x,x_*,v,v_*) df_{\lambda_*} df_\lambda = \\ &- \sum_{\lambda_*} \left\langle \nabla_v \cdot \Big[f_\lambda(x,v) \int_{B_{r_*(t,x)}\times\mathbb{R}^d} \mathcal{F}_\lambda(x,x_*,v,v_*) df_{\lambda_*} \Big], \psi(x,v) \right\rangle, \end{aligned} \tag{3.3.34}$$

where we denoted the inner scalar product

$$\langle h, \phi \rangle := \int_{\mathbb{R}^{2d}} h(x,v)\phi(x,v) d(x,v), \tag{3.3.35}$$

for any function $h(x,v)$, $\phi(x,v)$ for which the integral in (3.3.35) is well defined. By similar arguments of [3], it can be shown rigorously that $R(\varepsilon) \to 0$, as $\varepsilon \to 0$. Thus, we can rewrite equation (3.3.29) as follows

$$\begin{aligned} &\langle \partial_t f_\lambda(x,v) + v \cdot \nabla_x f_\lambda(x,v) - \mathcal{T}_\lambda[f](x,v), \psi(x,v) \rangle = \\ &- \sum_{\lambda_*} \left\langle \nabla_v \cdot \Big[f_\lambda(x,v) \int_{B_{r_*(x,t)}\times\mathbb{R}^d} \mathcal{F}_\lambda(x,x_*,v,v_*) df_* \Big], \psi(x,v) \right\rangle. \end{aligned} \tag{3.3.36}$$

Finally, we retrieve the mean-field equation as the strong form of (3.3.36)

$$\begin{aligned} &\partial_t f_\lambda(x,v) + v \cdot \nabla_x f_\lambda(x,v) - \mathcal{T}_\lambda[f](x,v) = \\ &\quad - \nabla_v \cdot \Big[f_\lambda(x,v) \int_{B_{r_*(x,t)}\times\mathbb{R}^d} \mathcal{F}_\lambda(x,x_*,v,v_*) \sum_{\lambda_*} f_{\lambda_*}(x_*,v_*) d(x_*,v_*) \Big]. \end{aligned} \tag{3.3.37}$$

Summing over the values of λ in equation (3.3.37) the transition operator vanishes as in (3.3.12) and we obtain the mean-field model for the total density $g(x,v)$ as

$$\partial_t g(x,v) + v \cdot \nabla_x g(x,v) = -\nabla_v \cdot \Big[\sum_\lambda f_\lambda(x,v) \int_{B_{r_*(x,t)}\times\mathbb{R}^d} \mathcal{F}_\lambda(x,x_*,v,v_*) g(x_*,v_*) d(x_*,v_*) \Big]. \tag{3.3.38}$$

Remark 5. Note that the continuous mean-field model (3.3.37) and the microscopic one (3.2.1) are equivalent when we consider the empirical distribution of the N-particles

$$f^N(x,v,\lambda,t) = \frac{1}{N} \sum_{i=1}^N \delta(x - x_i(t)) \delta(v - v_i(t)) \delta(\lambda - \lambda_i(t)), \tag{3.3.39}$$

where $\delta(\cdot)$ indicates the Dirac-delta function.

3.4 Stochastic particle-based approximation

We aim at solving the large system of agent (3.2.1) for $N \gg 1$, solving the mean-field model (3.3.37) by means of the scaled Boltzmann equation in the asymptotic regime (3.3.28). In particular, we aim at developing asymptotic stochastic algorithms for the simulation of the swarming dynamics, such as in [11, 162]. These approaches, based on Monte-Carlo algorithms are based of direct simulation Monte-Carlo methods (DSMCs) for kinetic equations [158, 161]. We mention also Random Batch Methods (RBMs) which, similarly, have been devised for simulating large systems of interacting agents [131].

3.4.1 Asymptotic Nanbu-type algorithm

In order to solve the mean-field dynamics we consider the Boltzmann-type equation (3.3.8) in the scaling limit (3.3.28), and we split the dynamics evaluating in three different steps the free transport, the label evolution and the interaction process, as follows (3.3.8)

$$\partial_t f_\lambda(x,v) = -v \cdot \nabla_x f_\lambda(x,v) \tag{3.4.1}$$
$$\partial_t f_\lambda(x,v) = \mathcal{T}_\lambda[f](x,v) \tag{3.4.2}$$
$$\partial_t f_\lambda(x,v) = Q^\varepsilon_\lambda(f_\lambda, f_\lambda)(x,v). \tag{3.4.3}$$

In order to approximate the time evolution of the density $f_\lambda(x,v,t)$ we assume to sample N_s particles $(x_i^0, v_i^0, \lambda_i^0)$ from the initial distribution. We consider a time interval $[0,T]$ discretized in N_t intervals of size Δt.

Transport step. First, we focus on the transport step in equation (3.4.1) and we approximate the solution at time t^{n+1} by

$$x_i^{n+1} = x_i^n + \Delta t v_i^n, \qquad i = 1, \dots, N_s \tag{3.4.4}$$

Labels switching. Secondly, we simulate how the labels change denoting by f_λ^n the approximation of $f_\lambda(x,v,n\Delta t)$, and writing the discrete version of the equation (3.4.2), for the transition operator (3.3.10) as follows

$$\begin{aligned} f_0^{n+1} &= (1 - \Delta t\ \pi^n_{F\to L})\ f_0^n + \Delta t\ \pi^n_{L\to F}\ f_1^n, \\ f_1^{n+1} &= (1 - \Delta t\ \pi^n_{L\to F})\ f_1^n + \Delta t\ \pi^n_{F\to L}\ f_0^n, \end{aligned} \tag{3.4.5}$$

The following Algorithm 3.4.1 describes how to simulate equation (3.4.5) in a time interval $[0,T]$ divided into N_t time steps.

Algorithm 3.4.1. [Labels switching]

1. Given N_s samples $(x_i^0, v_i^0, \lambda_i^0)$ from the initial distribution f_λ^0;

2. `for` $n = 0$ `to` N_t

   (a) `for` $i = 1$ `to` N_s

      i. compute the following probabilities rates

$$p_L = \Delta t\ \pi_{F\to L}, \qquad p_F = \Delta t\ \pi_{L\to F},$$

      ii. `if` $\lambda_i^n = 0$,
      with probability p_L agent i becomes a leader: $\lambda_i^{n+1} = 1$,

      iii. `if` $\lambda_i^n = 1$,
      with probability p_F agent i becomes a follower: $\lambda_i^{n+1} = 0$,

   `end for`

`end for`

Interaction step. Finally, we consider the interaction step (3.4.3) decomposing the interaction operator (3.3.9) in its gain and loss part,

$$Q_\lambda^\varepsilon(f_\lambda, f_\lambda) = \frac{1}{\varepsilon}\left[Q_\lambda^{\varepsilon,+}(f_\lambda, f_\lambda) - \rho^* f_\lambda\right],$$

where $\rho^* = M/N$ is the topological mass. Considering a forward discretization we obtain

$$f_\lambda^{n+1} = \left(1 - \frac{\rho^*\Delta t}{\varepsilon}\right) f_\lambda^n + \frac{\rho^*\Delta t}{\varepsilon}\frac{Q^{\varepsilon,+}(f_\lambda^n, f_\lambda^n)}{\rho^*}. \tag{3.4.6}$$

Equation (3.4.6) can be interpreted as follows. With probability $1 - \rho^*\Delta t/\varepsilon$ an individual in position x, velocity v and label λ will not interact with other individuals and, with probability $\rho^*\Delta t/\varepsilon$, it will interact with another individual according to

$$v_i^{n+1} = v_i^n + \varepsilon\mathcal{F}_{\lambda_i^n}(x_i^n, x_j^n, v_i^n, v_j^n), \tag{3.4.7}$$

for any $i = 1, \dots, N_s$, and where (x_j^n, v_j^n) is selected randomly among the nearest neighbours belonging to the topological ball $B_{r^*}(x_i, t)$. We will assume $\rho^*\Delta t = \varepsilon$ to maximize the total number of interactions and ensure that at each time step all agents interact with another individual with probability one.

Note that the sampling procedure of agents from the topological ball $B_{r^*}(x_i, t)$ can have extremely high computational costs, especially when the sample size is large, since it requires the explicit computation of the distances between each agent i and all the others agents. In order to improve the computational efficiency of this step we propose a procedure based on two steps: *a*) an approximation of the topological ball, *b*) k–Nearest Neighbours (k–NN) search.

a) *Topological ball approximation.* To avoid the expensive procedure of computing the topological ball over the whole sample, we consider a subsample of size N_c of the N_s selected

particles such that $N_c < N_s$, and we define the approximation to radius of the topological ball as follows

$$\tilde{r}_*(x_i, t) = \arg\min_{r>0} \left\{ \frac{1}{N_c} \sum_{k=1}^{N_c} \chi_{B_r(x_i)}(x_k) \geq \rho^* \right\}, \tag{3.4.8}$$

where ρ^* is the target topological mass.

b) k–NN search. We perform a *k–NN search* over a *k-d* binary tree. First, we construct the binary tree on the subsample of size N_c in such a way to partition the space and organize the points optimally dividing them according to their medians. We assume that every leaf-node contains at most N_l points. Then, we use a k-NN algorithm to find the $\rho^* N_c$ nearest neighbours to a given agent i, using the tree structure. We will show in the numerical experiments that this algorithm reduces the computational costs from the original quadratic to logarithmic. We refer to [107] for further details about this procedure.

Algorithm 3.4.2 describes how to solve equation (3.4.6) in a time interval $[0, T]$ divided into N_t time steps.

Algorithm 3.4.2. [Asymptotic Nanbu algorithm]

1. Give N_s samples $(x_i^0, v_i^0, \lambda_i^0)$ from the initial distribution f_λ^0;
2. Set the value of the topological mass ρ^* and of the subsample size N_c;
3. `for` $n = 0$ `to` N_t
   (a) select a subsample of size N_c,
   (b) construct a binary tree over the subsample, with leaf-nodes of size at most N_l
   (c) `for` $i = 1$ `to` N_s
      i. find the $\rho^* N_c$ nearest agents using a k–NN search algorithm on the tree,
      ii. select randomly an index j among the nearest neighbors,
      iii. compute the velocity change v_i^{n+1} as in equation (3.4.7),
      iv. Update the position x_i according to (3.4.4), with $\rho^* \Delta t = \varepsilon$.

   `end for`

`end for`

3.4.2 Numerical validation

In this section we perform different numerical experiments to test both the accuracy and the efficiency of the Asymptotic Nanbu Algorithm 3.4.2 with k–NN search.

Accuracy. Consider a model in which N_s agents with position x_i and velocity v_i interact with their nearest M neighbours without changing their labels and their position. Assume agents are subjected just to alignment forces. Hence, their dynamics at the microscopic level is governed by the following ODE for $i = 1, \dots, N_s$,

$$\dot{v}_i = \frac{1}{\rho^*}\frac{1}{N_s}\sum_{j=1}^{N_s}(v_j - v_i)\chi_{\mathcal{B}_M(x_i;\mathbf{x})}(x_j), \tag{3.4.9}$$

where $\rho^* = M/N_s$ is the target topological mass. At the kinetic level, suppose that agents modify their velocity according to binary interactions. Assume that at any time step an agent with position and velocity (x, v) meets another agent with position and velocity $(x_*, v_*) \in B_{r_M^*}(t, x)$ where r_M^* is defined as in (3.4.8). Its post-interaction velocity is given by

$$v' = v + \varepsilon(v_* - v), \tag{3.4.10}$$

where $\varepsilon > 0$ is a small parameter. Recall that the ball $B_{r_M^*}(t, x)$ by definition contains a certain percentage of mass, that we suppose to be ρ^*. If we denote by $f(x, v, t)$ the density of agents at time t with position x and velocity v, then the kinetic equation describing its evolution reads

$$\partial_t f(x, v, t) = -\nabla_v \cdot \Big[f(x, v, t) \int_{B_{r^*(t,x)} \times \mathbb{R}^d} (v_* - v) f(x_*, v_*, t) dx_* dv_* \Big]. \tag{3.4.11}$$

The microscopic model in (3.4.9) can be solved exactly and the evolution of the velocity is given by

$$v_i(t) = v_i(0)e^{-t} + e^{-t}\int_0^t \bar{v}_i(t)e^t dt, \qquad \text{with} \qquad \bar{v}_i(t) = \frac{1}{\rho^*}\frac{1}{N_s}\sum_{j=1}^{N_s} v_j(t)\chi_{\mathcal{B}_M(x_i;\mathbf{x})}(x_j). \tag{3.4.12}$$

We choose as initial distribution the sum of two 2d-Gaussian in the plane (x, v) one with mean $(-0.33, -0.16)$ and the other with mean $(0.33, 0.16)$ and both with standard deviation $(0.12, 0.06)$. The dynamics at the kinetic level is simulated with Algorithm 3.4.2, where we compute the velocity change as in equation (3.4.10). We suppose $N_s = 10^5$, and $\varepsilon = 10^{-5}, \dots, 10^0$. We perform the computations assuming the subsample is made with the $p = 100N_c/N_s\%$ of the total mass, for a certain p. In Figure 3.2 we plot the initial distribution in the v-x plane and the marginals in x and v.

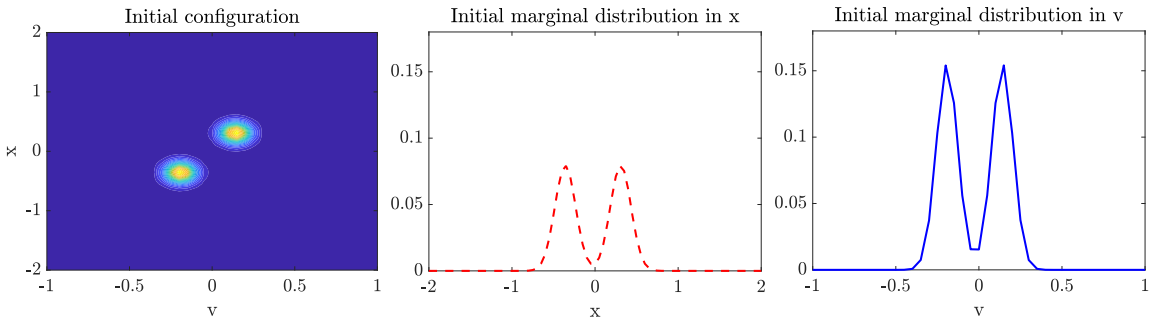

Figure 3.2: Validation test: initial configuration and its marginals in x and v.

Then, we run $S = 100$ simulations and we plot in figure 3.3 the mean and the standard deviation as a shaded area of the velocity distribution at time $T = 3$ for $N_s = 10^5$, $\rho^* = 0.01, 0.35, 0.75$ and $\varepsilon = 10^{-3}$ for different values of p.

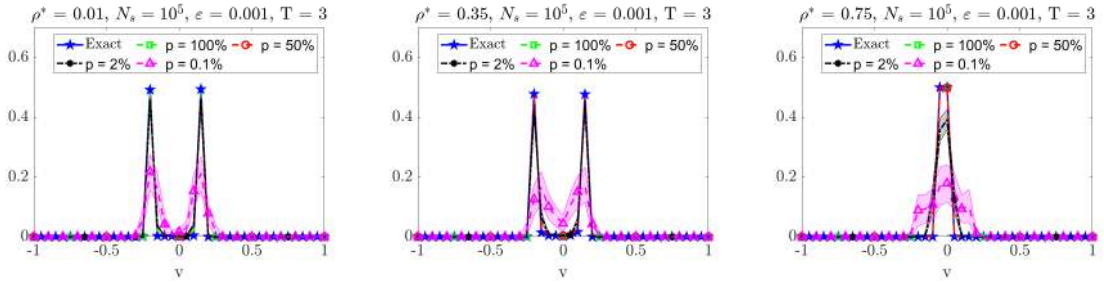

Figure 3.3: Validation test: comparison between the solution to the kinetic equation in (3.4.11) computed by means of Asymptotic Nanbu algorithm 3.4.2 and the exact solution in (3.4.12). Mean (dashed line) and standard deviation (shaded area) of the velocity distribution computed over $S = 100$ simulations for different values of p. From the left to the right $\rho^* = 0.01, 0.35, 0.75$. Markers have been added just to indicate different lines.

In Figure 3.4 for different values of ρ^*, the L_2-norm of the error between the solution to the kinetic equation in (3.4.11), simulated by means of the Asymptotic Nanbu algorithm 3.4.2 (one simulation) for different values of p, and the exact solution in (3.4.12). Note that we observe a saturation effect for $\varepsilon \approx 10^{-2}$.

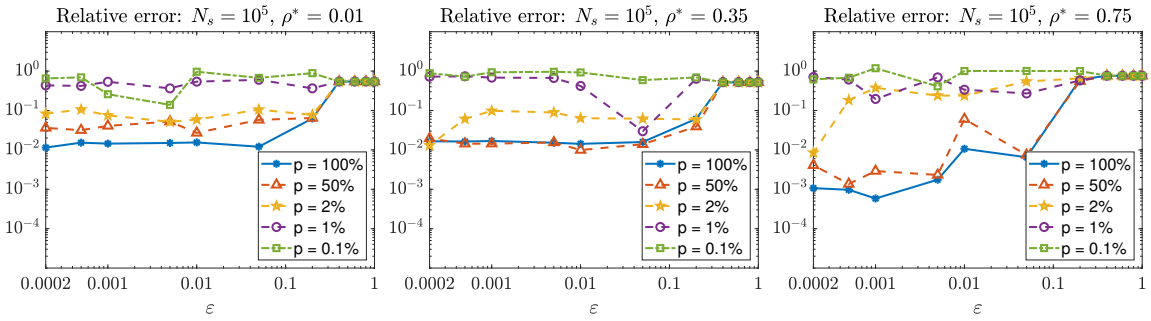

Figure 3.4: Validation test: L_2- norm of the error between the solution to the kinetic equation in (3.4.11) simulated by means of the Asymptotic Nanbu algorithm 3.4.2 (one simulation) and the exact solution in (3.4.12). From the left to the right $\rho^* = 0.01, 0.35, 1$. Markers represent the error for the different values of ε.

Computational costs. We now compare the computational costs of the exhaustive search and the k–NN search. The computational cost of an exhaustive search is $\mathcal{O}(dN_s^2)$, where d is the space dimension and N_s the number of particles. Indeed, first one needs to compute the distances between each point and all the others, with a cost of $\mathcal{O}(dN_s^2)$, and then to sort them, with a cost of $\mathcal{O}(N_s^2 log(N_s))$. The cost of a k–NN search is logarithmic in time. First one needs to organize agents optimally with a k-d tree. The cost of this operation is proportional to $N_s log(N_s)$. Then the idea is to perform a search over the tree to select which are the nearest agents. It can be shown (see [107]) that the k–NN search algorithm examines the nodes in optimal order, that is in order of increasing dissimilarities, and that the number of nodes that should be examined is proportional to $((\rho^* N_s)^{1/d} + 1)^d$. Hence, the total cost to construct a k-d tree and to perform the search over it is

$$\mathcal{O}(max(((\rho^* N_s)^{1/d} + 1)^d log(N_s), N_s log(N_s)). \tag{3.4.13}$$

In Figure 3.5 we see the comparison between the computational cost to perform one exhaustive and one k–NN search as N_s varies for different values of ρ^*. We set $N_s = 5 \times 10^3, \ldots, 1.5 \times 10^4$. The k–NN computational cost increases as ρ^* increases. In Figure 3.6 we see a comparison between the computational costs of a k–NN search as N_s varies for different subsamples percentage size p. We see that the computational cost decreases proportionally to the subsample size.

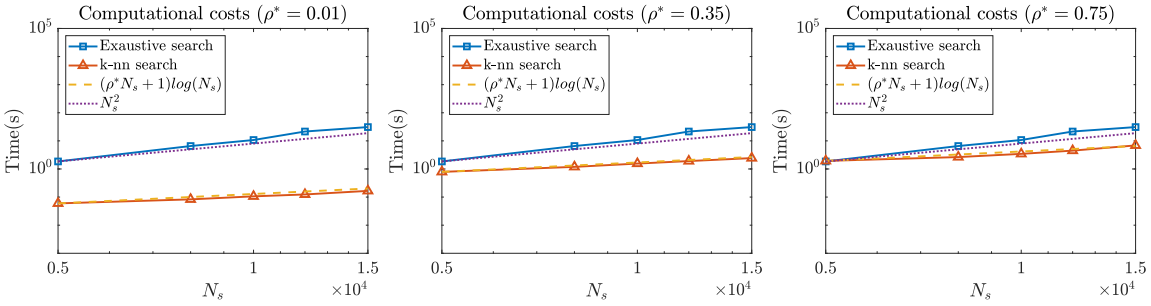

Figure 3.5: Comparison between the computational costs of the exhaustive search and the k–NN search for different values of N_s. From the left to the right $\rho^* = 0.01, 0.35, 0.75$. Markers represent the computational costs relative to the different values of N_s.

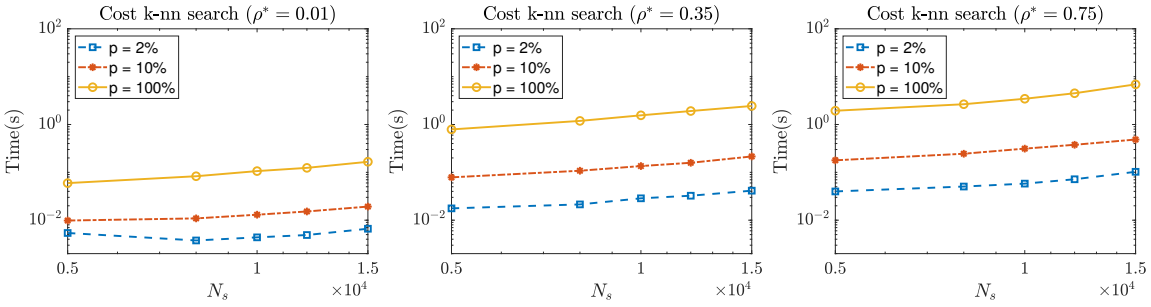

Figure 3.6: Comparison between the computational costs of the exhaustive search and the k–NN search for different values of N_s and of the percentage p of the subsample size. From the left to the right $\rho^* = 0.01, 0.35, 0.75$. Markers represent the computational costs relative to the different values of N_s.

3.5 Numerical experiments

We present different numerical experiments simulating the two and three dimensional dynamics both at the microscopic and mesoscopic levels. The dynamics at microscopic level is discretized by a forward Euler scheme with a time step $\Delta t = 0.01$, whereas the evolution of the kinetic dynamics is approximated by the Asymptotic Nanbu algorithm described in 3.4.2 with $\varepsilon = 0.01$. The time evolution of the labels is computed with Algorithm 3.4.1 at both the microscopic and mesoscopic levels. In the microscopic case we set $N = 400$. In the mesoscopic case we choose a sample of $O(N_s)$ particles, with $N_s = 5 \times 10^5$, and a subsample of $O(N_c)$ particles, with $N_c = 10^4$ that corresponds to a percentage $p = 2\%$ of the total mass, for the approximation of the density. We assume the topological target mass to be $\rho^* = 0.01$. Table 3.1 reports the parameters of the model that remain unchanged in the various scenarios. The other parameters will be specified later.

Table 3.1: Model parameters for the different scenarios.

	C_{rep}	C_{ali}	C_{att}	C_v	s	$\overline{r}$	$\underline{r}$	ϵ
2D model	100	12	0.7	5	10	200	1	200
3D model	100	12	0.7	5	10	350	20	150

3.5.1 Numerical test in two spatial dimensions

We consider the swarming dynamics evolving on the spatial space $(x, y) \in \mathbb{R}^2$ and velocity space $(v_x, v_y) \in \mathbb{R}^2$.

Test 2D with no food sources

Suppose the model includes no food sources, i.e. $C_{src} = 0$, and no attraction to the centre of mass, i.e. $C_{ctr} = 0$. We simulate the dynamics up to time $T = 500$, and we report in Figure 3.7 the initial configuration for both the microscopic and mesoscopic dynamics.

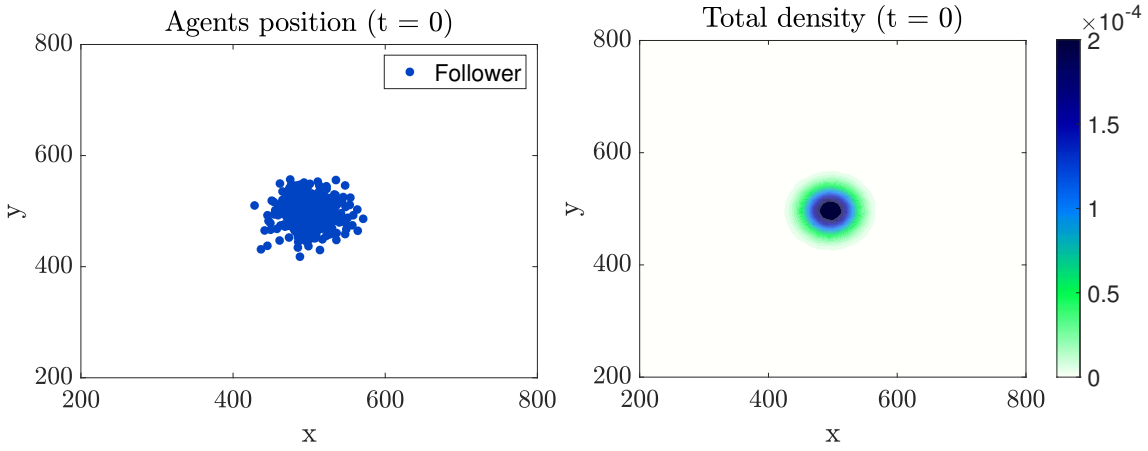

Figure 3.7: Initial configuration in the spatial 2D case with no food sources.

At time $t = 0$, agents are normally distributed with mean $\mu = 500$ and variance $\sigma^2 = 25^2$ and are in the followers status. Labels change according to the transition rates defined in (3.3.13) with $q_{FL} = 2 \times 10^{-4}$ and $q_{LF} = 4 \times 10^{-3}$.

Microscopic case. In Figure 3.8 we report three snapshots of the dynamics at time $t = 50$, $t = 300$ and $t = 500$, for the dynamics without leaders' emergence (top row) and with leaders' emergence (bottom row). We observe that, without leaders, agents align and form a compact swarm, whereas with leaders' emergence we observe the formation of

different groups. The splitting is not symmetric since leaders' emergence occurs randomly and this is reflected in the cluster formation.

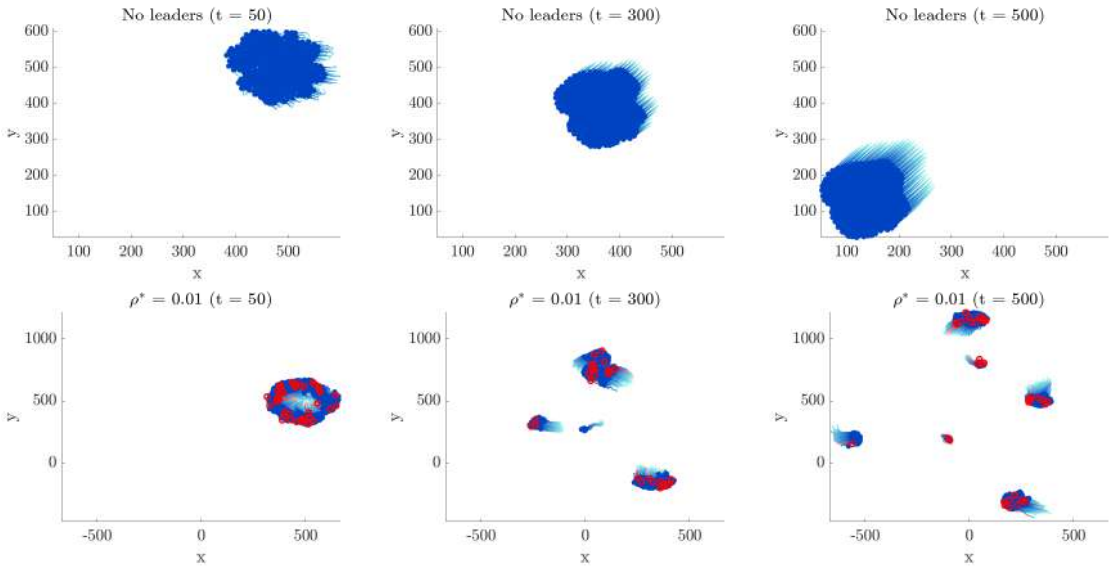

Figure 3.8: Three snapshots of the 2D dynamics at the microscopic level described in (3.2.1) with λ evolving with rates (3.3.13) taken at time $t = 50$, $t = 300$ and $t = 500$ and assuming $C_{ctr} = C_{src} = 0$. In the first row, the dynamics without leaders. In the second row, the dynamics with leaders assuming $\rho^* = 0.01$. We represent in blue the agents in the followers status and in red the ones in the leaders status.

Mesoscopic case. In Figure 3.9 we report three snapshots of the dynamics at time $t = 50$, $t = 300$, $t = 500$. In the first row, the time evolution of the total density and in red the velocity vector field of the leaders. In the second row, the evolution of the leaders' density. The behavior is similar to the one of the microscopic case, where we observe the formation of various clusters, and the emergence of leaders uniformly over the swarm density.

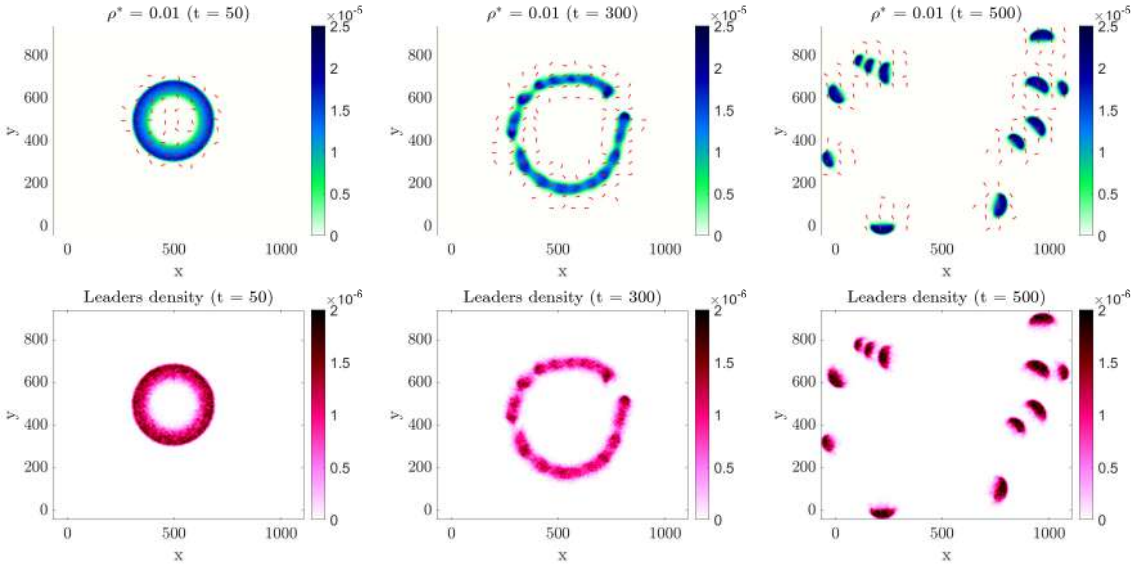

Figure 3.9: Three snapshots of the 2D dynamics at the mesoscopic level described in (3.3.37) and simulated by means of the Asymptotic Nanbu algorithm 3.4.2 with λ evolving with rates (3.3.13) as in Algorithm 3.4.1, taken at time $t = 50$, $t = 300$ and $t = 500$ and assuming $C_{ctr} = C_{src} = 0$. In the first row, the dynamics of the total mass and in red the velocity vector field of the leaders. In the second row, the leaders' dynamics.

In Figure 3.10 the agents percentages for the dynamics with leaders.

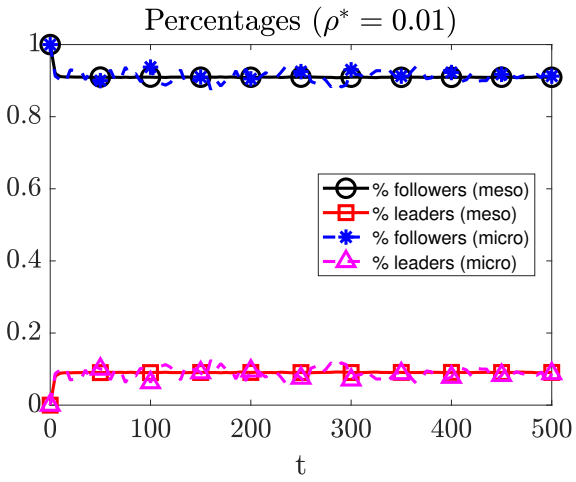

Figure 3.10: Agents percentages. Markers have been added just to distinguish the different lines.

The videos of the simulations of this subsection are available at [VIDEO].

Test 2D: two food sources

Assume the model includes two food sources located in $x_1^{src} = (300, 500)$ and $x_2^{src} = (1000, 500)$. Assume $C_{ctr} = C_{src} = 0.75$. Run the dynamics until time $T = 200$. In Figure 3.11 the initial configuration for the microscopic and mesoscopic case. We assume that initially the 87.5% of agents is in the followers status. Among them the 75% is normally distributed with mean $\mu = 550$ and variance $\sigma^2 = 10^2$ while the 12.5% is normally distributed with mean $\mu = 650$ and variance $\sigma^2 = 50^2$. The remaining 12.5% is in the leaders status and it is normally distributed with mean $\mu = 800$ and variance $\sigma^2 = 10^2$. New leaders emerge with higher probability where the followers concentration is higher. Leaders return in the follower status with higher probability if the followers concentration around their position is lower. Hence we consider density dependent transition rates defined in equation (3.3.16) with $q_L = 4 \times 10^{-3}$ and $q_F = 3 \times 10^{-3}$.

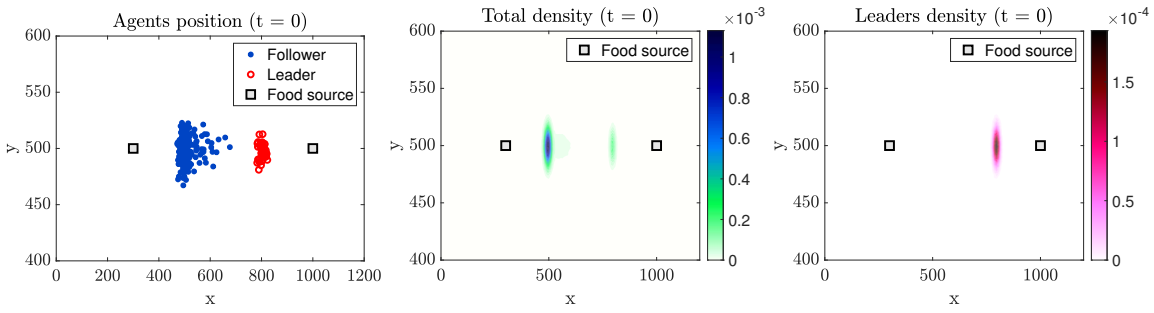

Figure 3.11: Initial configuration in the 2D case with two food sources.

Microscopic case. In Figure 3.12 three snapshots of the dynamics at time $t = 50$, $t = 100$, $t = 200$. Agents that at time $t = 0$ were in the leaders status change immediately their labels since no followers are positioned around them. A large group is attracted by one of the two food sources while the remaining part moves subjected just to attraction, repulsion and alignment forces without being attracted by the other food source. Once this smaller group moves far away from the main group, leaders start to be attracted to the centre of mass. In late time, all agents join and move toward one of the two food sources.

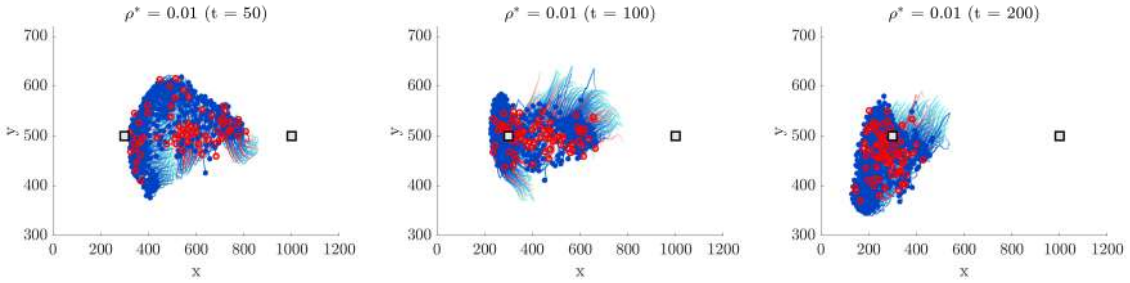

Figure 3.12: Three snapshots of the 2D dynamics at the microscopic level described in (3.2.1) with λ evolving with rates (3.3.16) taken at time $t = 50$, $t = 100$ and $t = 200$ and assuming $C_{ctr} = C_{src} = 0.75$. We represent in blue the agents in the followers status and in red the ones in the leaders status.

Mesoscopic case. In Figure 3.13 three snapshots of the dynamics at time $t = 50$, $t = 100$, $t = 200$. In the first row the time evolution of the total density and in red the velocity vector field of the leaders. In the second row the time evolution of the leaders' density. The behaviour is similar to the one observed in the microscopic case.

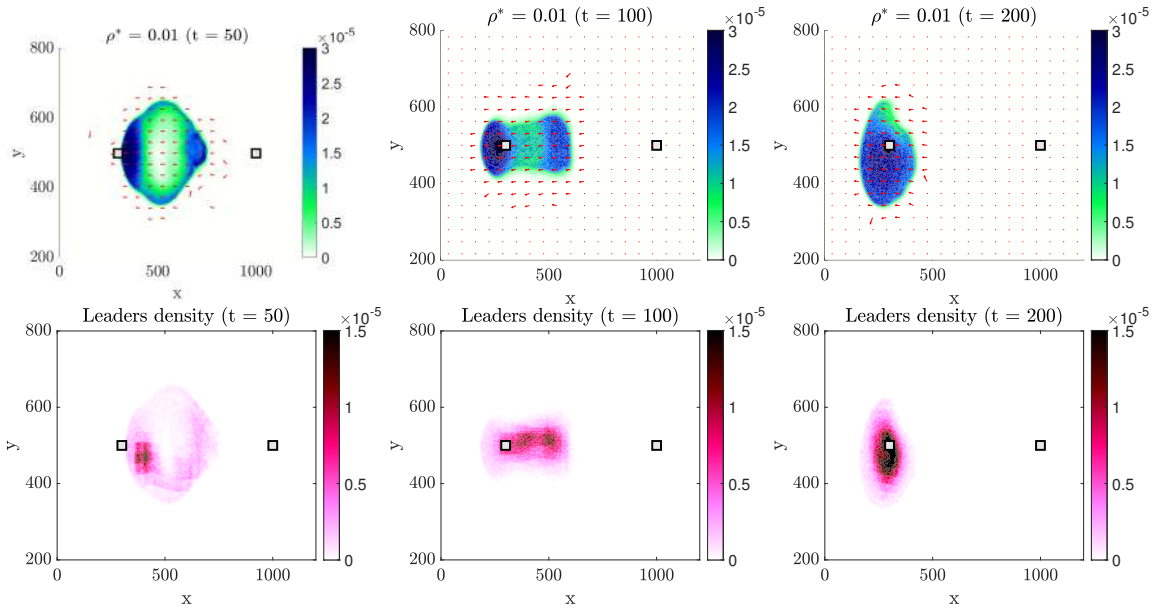

Figure 3.13: Three snapshots of the 2D dynamics at the mesoscopic level described in (3.3.37) and simulated by means of the Asymptotic Nanbu algorithm 3.4.2 with λ evolving with rates (3.3.16) as in Algorithm 3.4.1, taken at time $t = 50$, $t = 100$ and $t = 200$ and assuming $C_{ctr} = C_{src} = 0.75$. First row: evolution of the total density and in red the velocity vector field of the leaders' density. Second row: evolution of the leaders' density.

In Figure 3.14 the time evolution of the percentages of leaders and followers for the two dimensional spatial test with two food sources. The agents percentages have been computed both by counting the effective number of followers and leaders per time steps, and as stationary solution to the master equation (3.3.14). The densities reach the positive equilibrium defined in equation (3.3.15). Indeed, the transition rates defined in (3.3.16) can be approximated by constant values. In particular, for any fixed time $t > 0$, for any $\lambda \in \{0, 1\}$ and for any $x \in \mathbb{R}^d$ we have

$$\pi_{L\to F}(x, \lambda; f, t) = \bar{\alpha}(t), \qquad \pi_{F\to L}(x, \lambda; f, t) = \bar{\beta}(t) \tag{3.5.1}$$

where

$$\bar{\alpha}(t) = \mathbb{E}_x\left(\pi_{L\to F}(\cdot)\right), \qquad \bar{\beta}(t) = \mathbb{E}_x\left(\pi_{F\to L}(\cdot)\right), \tag{3.5.2}$$

with $\mathbb{E}_x(\cdot)$ denoting the mean value with respect to x.
Similar results can be obtained for the 2D model without food sources since the transition rates are constants values, by definition.

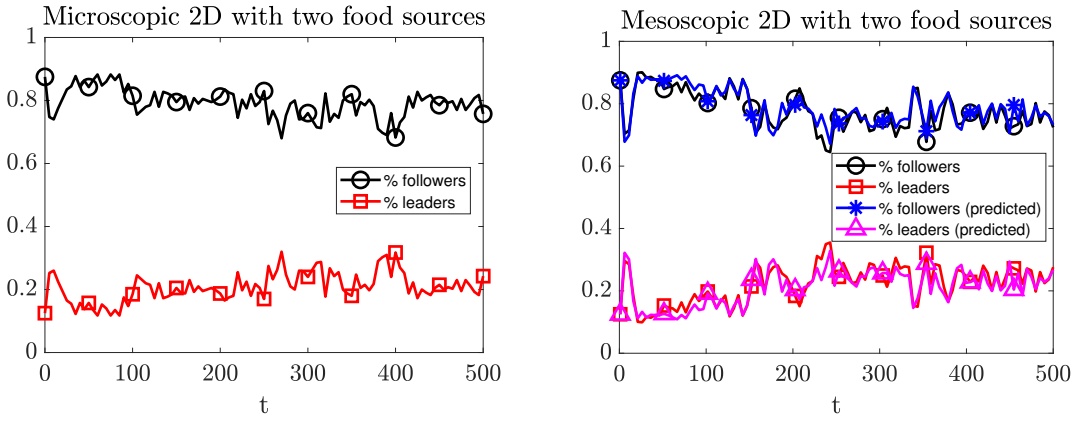

Figure 3.14: Agents percentages: microscopic (on the left) and mesoscopic (on the right) two dimensional model with two food sources. In black and in red the agents percentages computed by counting the effective number of followers and leaders per time steps. In blue and in magenta the stationary solution to the master equation (3.3.14).

The videos of the simulations of this subsection are available at [VIDEO].

Test 2D: one food source

Assume the model includes one food source located in $x_1^{src} = (300, 500)$. Run the simulation until time $T = 120$. Suppose labels change aiming at organizing agents toward a common target, that in this case is supposed to be the food source x_1^{src}. In particular, assume λ varies with rates (3.3.18) with $\bar{\alpha} = 0.7$ and $\underline{\alpha} = 0.3$. In Figure 3.15 the initial configuration for both the microscopic and mesoscopic case.

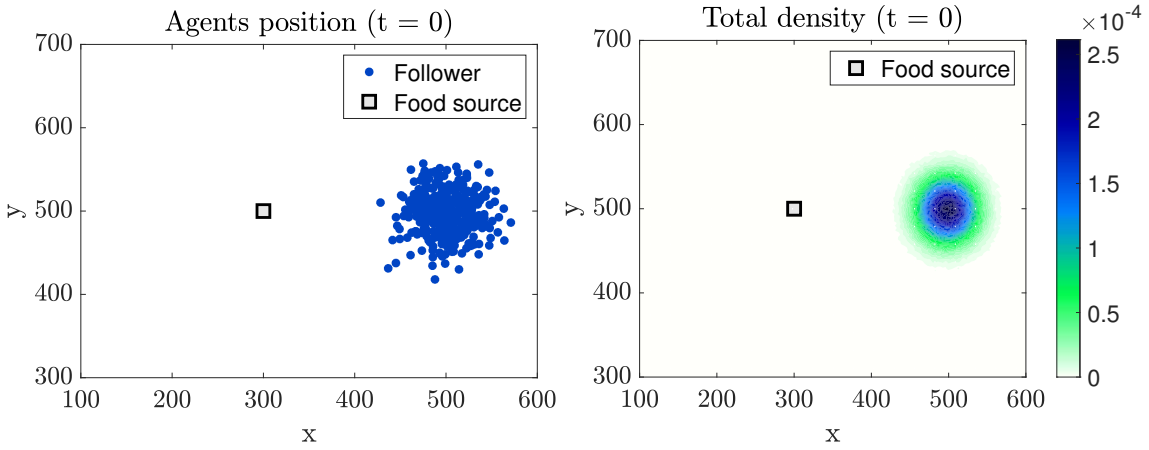

Figure 3.15: Initial configuration in the 2D case with one food source.

Microscopic case. In Figure 3.16 three snapshots of the dynamics at time $t = 5$, $t = 20$ and $t = 50$ with leaders and with $\mathcal{G}[f](\cdot)$ chosen as in (3.3.20). Followers are driven by leaders and reach the target position.

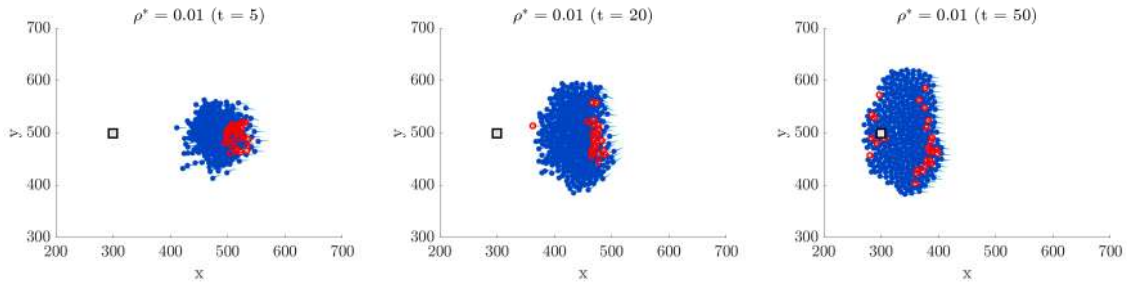

Figure 3.16: Three snapshots of the 2D dynamics at the microscopic level described in (3.2.1) with λ evolving with rates (3.3.18) taken at time $t = 5$, $t = 20$ and $t = 50$ and assuming $C_{ctr} = C_{src} = 0$. We represent in blue the agents in the followers status and in red the ones in the leaders status.

Mesoscopic case. In Figure 3.17 three snapshots of the dynamics at time $t = 5$, $t = 20$, $t = 50$ with transition rates depending on the orientation according to (3.3.18). In the first row, the evolution of the whole mass, and in red the velocity vector field of the leaders. In the second row, the evolution of the leaders mass.

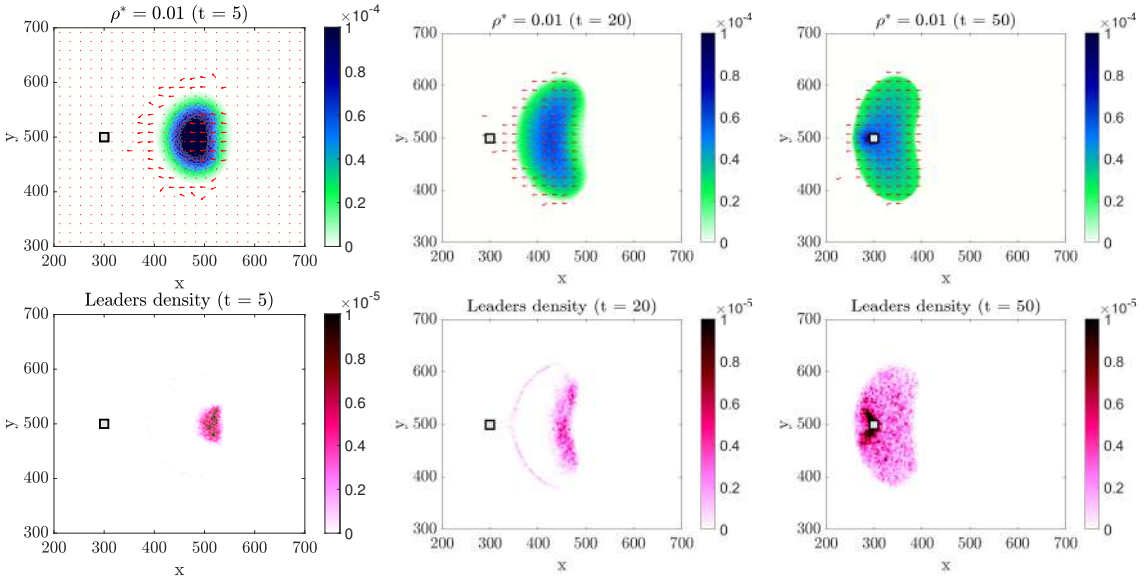

Figure 3.17: Three snapshots of the 2D dynamics at the mesoscopic level described in (3.3.37) and simulated by means of the Asymptotic Nanbu algorithm 3.4.2 with λ evolving with rates (3.3.18) as in Algorithm 3.4.1, taken at time $t = 5$, $t = 20$ and $t = 50$ and assuming $C_{ctr} = C_{src} = 0$. First row: evolution of the total density and in red the velocity vector field of the leaders. Second row: evolution of the leaders density.

In Figure 3.18 we report in the first row the angle velocity distribution at time $t = 100$, $t = 120$, $t = 180$ and in the second row the correspondent velocity vector field, outlining the milling behaviour around the target positions $\overline{x} = x_1^{src}$. The videos of the simulations of this subsection are available at [VIDEO].

3.5.2 Numerical test in 3D with two food sources

We consider the three dimensional model in space and velocity, simulating the swarming dynamics up time $T = 200$. Initially agents are normally distributed with mean $\mu = 500$ and variance $\sigma^2 = 25^2$ in both spatial and velocity dimension, and are all in the followers status. We report in Figure 3.19 the initial configuration for both the microscopic and mesoscopic case. For the mesoscopic case, we also depict on the (x, y) plane the projection of the spatial density.

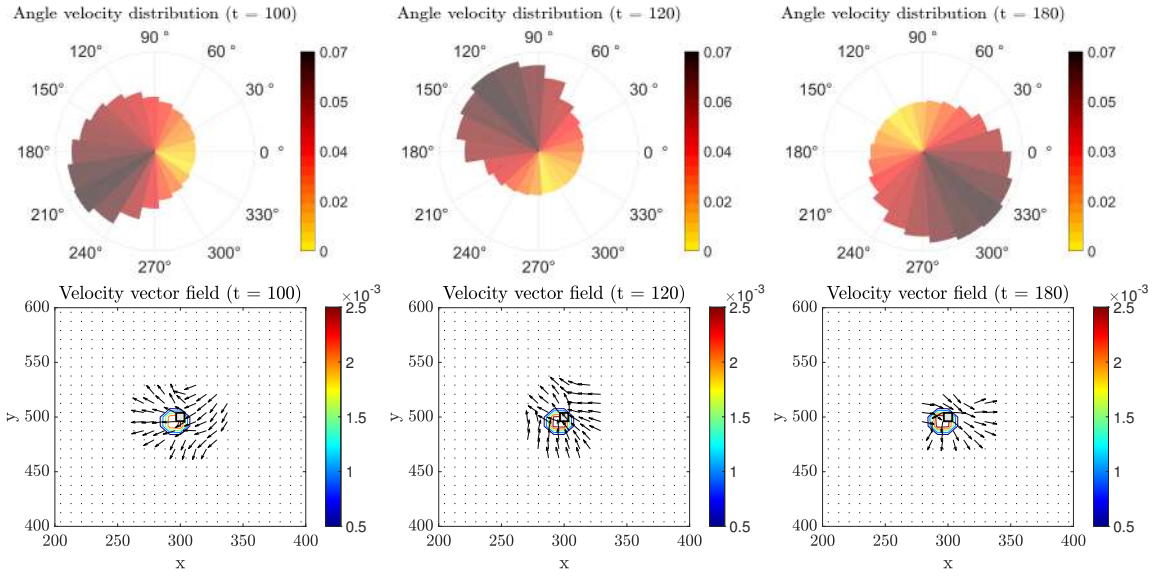

Figure 3.18: Angle velocity distribution (first row) and velocity vector field (second row) at time $t = 100$, $t = 120$, $t = 180$ obtained by simulating the dynamics in (3.3.37) by means of the algorithm 3.4.2 with λ evolving with rates (3.3.18) as in algorithm 3.4.1.

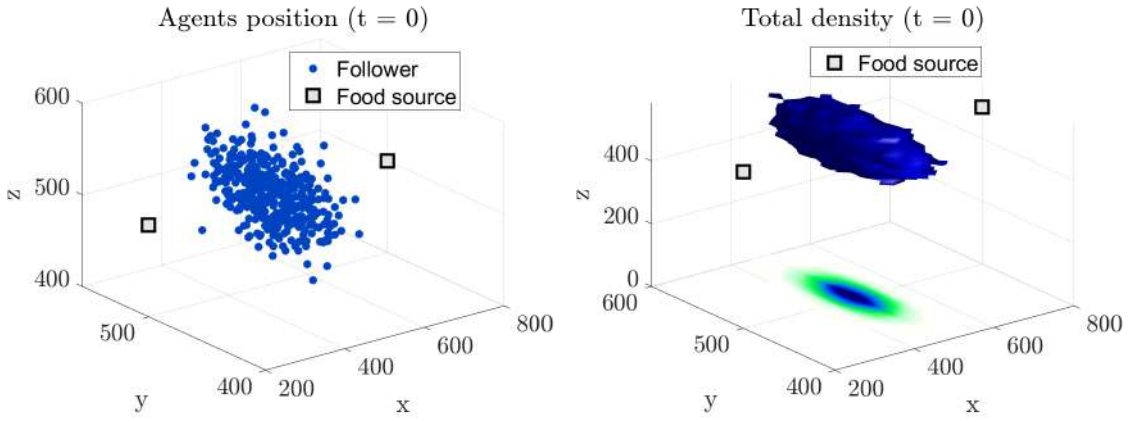

Figure 3.19: Initial configuration in the 3D case with food sources.

Assume the two food sources to be located in $x_1^{src} = (200, 500, 500)$ and $x_2^{src} = (800, 500, 500)$. Suppose that leaders emerge with density-dependent transition rates as defined in (3.3.16) and where we assume the constants to be $q_L = 4 \times 10^{-3}$ and $q_F = 3 \times 10^{-3}$.

Microscopic case. In Figure 3.20 three snapshots of the dynamics at time $t = 50$, $t = 100$, $t = 200$. First row: $C_{centre} = 0$, $C_{food} = 0.75$. Agents split in two groups moving toward the two food sources. Second row: $C_{ctr} = 4$, $C_{src} = 0.75$. At final time agents move toward one of the two food sources.

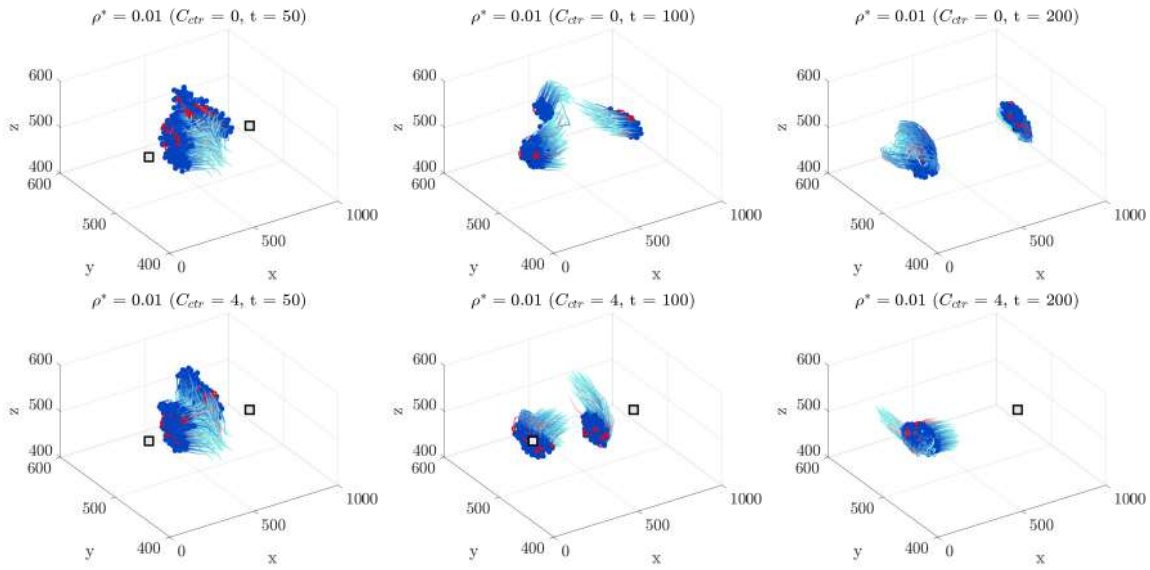

Figure 3.20: Three snapshots of the 3D dynamics at the microscopic level described in (3.2.1) with λ evolving with rates (3.3.16) taken at time $t = 50$, $t = 100$ and $t = 200$. First row, $C_{ctr} = 0$, $C_{src} = 0.75$. Second row, $C_{ctr} = 4$, $C_{src} = 0.75$. We represent in blue the agents in the followers status and in red the ones in the leaders status.

Mesoscopic case. In Figure 3.21 we report three snapshots of the dynamics at time $t = 50$, $t = 100$, $t = 200$ and in red the velocity vector field. We add also the density distribution of the whole flock projected over the plane (x, y) and in red the leaders velocity vector field. First row: $C_{crt} = 0$, $C_{src} = 0.75$. Second row: $C_{ctr} = 4$, $C_{src} = 0.75$. The behaviour is similar to the one in the microscopic case.

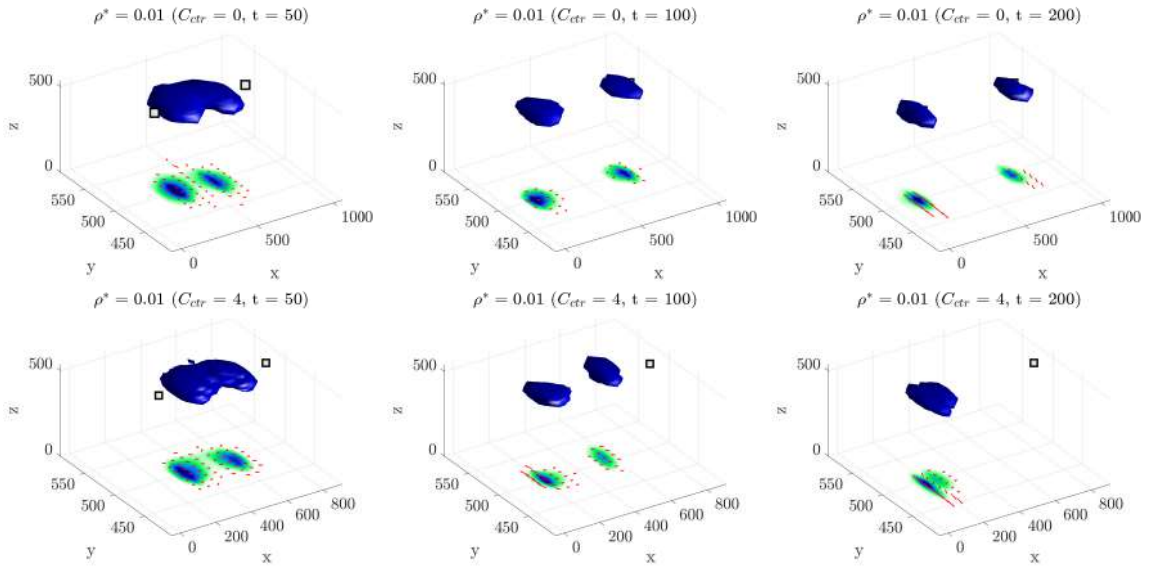

Figure 3.21: Three snapshots of the 3D dynamics at the mesoscopic level described in (3.3.37) and simulated by means of the Asymptotic Nanbu algorithm 3.4.2 with λ evolving with rates (3.3.16) as in Algorithm 3.4.1, taken at time $t = 50$, $t = 100$ and $t = 200$. First row, $C_{ctr} = 0$, $C_{src} = 0.75$. Second row, $C_{ctr} = 4$, $C_{src} = 0.75$. In red the velocity vector field.

The videos of the simulations of this subsection are available at [VIDEO].

3.6 Conclusions

In this chapter, we have studied collective behaviour of birds under a follower-leaders dynamics, starting from a simplified version of the model presented in [65].Through the emergence of leaders, we recover the ability to split the initial configuration and initiate directional changes without the need of external influences. We derived a kinetic model to effectively depict the motion of a large swarm with transient leadership and topological interactions, and subsequently we simulated the dynamics introducing a novel stochastic particle method. A significant emphasis was placed on studying topological interactions. We tackled the issue of the numerical evaluation of nearest Neighbors reducing the computational costs of the search from quadratic to logarithmic by optimally organizing agents in a binary tree and performing a k-NN search. Moreover, we directed our attention to transient leadership, showcasing how labels can change over time, particularly for driving agents towards a common target. Various strategies for leaders' emergence were explored, including continuous leadership levels, as introduced in Remark 4. Additionally, it would be intriguing to describe the original model from a kinetic viewpoint, reintroducing delay,

which, as demonstrated in [65], appears to play a crucial role in achieving desired configurations. Finally, several questions arise concerning the study of non-local terms in high dimensions. For instance, it could be beneficial to further enhance the numerical scheme implemented, focusing on other useful strategies for approximating topological interactions.

Chapter 4

Kinetic based optimization enhanced by genetic dynamics

In this chapter we propose and analyse a variant of the recently introduced kinetic based optimization method that incorporates ideas like survival-of-the-fittest and mutation strategies well-known from genetic algorithms. Thus, we provide a first attempt to reach out from the class of consensus/kinetic-based algorithms towards genetic metaheuristics. Different generations of genetic algorithms are represented via two species identified with different labels, binary interactions are prescribed on the particle level and then we derive a mean-field approximation in order to analyse the method in terms of convergence. Numerical results underline the feasibility of the approach and show in particular that the genetic dynamics allows to improve the efficiency, of this class of global optimization methods in terms of computational cost.

The results presented in this chapter have been published in [9] .

4.1 Introduction

In recent years a new perspective on gradient-free methods for global optimization of non-convex high-dimensional functions was established based on collective motion of particles. The mathematical depiction of self-organizing dynamics has significant implications in various applications, including traffic flow, pedestrian dynamics, flock behavior of birds, cell migration, and its primary focus lies in comprehending emergent properties of large interacting systems across different scales, see for example [2, 18, 19, 50, 155]. Hence, this new class of methods aims to exploit the collective dynamics of swarms to find global optimizers as consensus points emerging from a system of interacting agents, where each agent represents an evaluation of the objective function. In this context, Consensus-based optimization (CBO) is a global optimization method allowing rigorous proofs that the swarms concentrate arbitrarily close to the unique global minimizer of the objective function, we refer to [48, 103, 104, 117, 165] and the overview in [186]. In CBO, differently from gradient-based methods, such as stochastic gradient descent[39], the landscape of the objective functions is explored via function evaluations only. Indeed, the objective value at the current position and the current position of the agents is exchanged, with the help of a weighted mean value which is constructed such that the Laplace principle [82] applies. The dynamics is tailored

such that the agents confine towards the weighted-mean on the one hand, and randomly explore the landscape on the other hand. This already indicates that the two components of the dynamics need to be well-balanced to obtain desirable results. CBO was a first step towards the mathematical understanding of metaheuristics for global optimization, such as particle swarm optimization (PSO) methods, where second order dynamics is used for the evolution of the particles [135, 166]. Recently, the gap between the first-order method CBO and PSO was bridged in [117], further extensions were provided for systems with memory or momentum effects [57, 187], for constrained optimization [37], multi-objective problems [36, 137] and jump-diffusion processes [134]. More recently, kinetic-based optimization (KBO) methods have been proposed in[21], where each agent with position x moves subject to the following interaction rules

$$x' = x + \nu_F(\hat{x}(t) - x) + \sigma_F D(x)\xi, \tag{4.1.1}$$

where x' denotes the post iteration position, σ_F, ν_F are positive parameters which allow to balance the exploitation and exploration of the swarm, ξ is a random perturbation term, $D(x)$ is a diffusion matrix and $\hat{x}(t)$ denotes the global estimate of the position of the global minimizer at time t. In addition to this dynamics a drift towards the local best and a local diffusion term is proposed in [21]. The corresponding dynamics is described by a multidimensional Boltzmann equation and can be simulated with the help of Monte Carlo algorithms [11, 162].

With this work, we aim at extending KBO algorithm reaching out towards genetic algorithms (GA) [152, 183], a very popular class of metaheuristics that is widely used in engineering. The GA models a natural selection process based on biological evolution [115, 153]. To this end, individuals (parents) from the current population are selected and their objective values (gene information) is combined to generate the next generation (children). The selection process is usually driven by a survival-of-the-fittest idea, hence over successive generations, the system is assumed to evolve towards an optimal solution. Agents in promising positions, i.e., with small objective values, are labeled as parents and the others are labeled as children. Parents do not modify their position, hence they survive the iteration like in a survival-of-the-fittest strategy. In contrast, a child in position x interacts with a randomly chosen parent in position x_* and updates the position according to the rules

$$\begin{aligned} x' &= x_*, \quad \text{with rate } \nu_F, \\ x' &= x, \quad \text{with rate } 1 - \nu_F. \end{aligned} \tag{4.1.2}$$

Here x' denotes the post interaction position and $\nu_F > 0$ is the jump rate. Furthermore, mutations can occur, that means a child in position x encounters a random perturbation of the form

$$x' = x + \sigma_F \xi, \tag{4.1.3}$$

where σ_F is a positive parameter and ξ random vector drawn from a normal distribution. Slight modifications of the mutation process can be considered, assuming that all children encounter a random perturbation of the type

$$x' = x + \sigma_F D(x)\xi, \tag{4.1.4}$$

with diffusion matrix $D(x)$.

Thus, to establish a relation between KBO and GA, we divide the swarm into two species called followers (children) and leaders (parents) via a labeling strategy, evolving according to different transition processes [4, 144]. In this setting, leaders are selected analogously to parents in genetic algorithms, whereas the process of child's generation is substituted by the relaxation of the followers toward the leaders' position. Then, the dynamics of KBO (4.1.1) is split between the leaders, which move toward the global optimum estimate $\hat{x}$, and the followers, which explore the minimization landscape. Furthermore, we aim to improve the performance of these algorithms by relying on the higher flexibility of the particle dynamics.

More details are presented in the following sections which are organized as follows: in Section 4.2 we introduce the Genetic Kinetic based optimization methods focusing on the description of the binary interactions rules which describes the dynamics. In Section 4.3, we derive the mean field, in particular the evolution equations of the density functions of the two species. In Section 4.2.2, we discuss different strategies of how to assign labels. In Section 4.4, we provide a theoretical analysis including the exponential decay of the variance and the convergence of the method to the global minimum. In Section 4.5, we describe Nanbu's algorithm which is used to obtain the numerical results presented in Section 4.6. Here, we show different numerical experiments, testing the efficiency in terms of success rate and number of iterations and compare the results of the GKBO algorithm, to KBO and genetic algorithms.

4.2 Genetic kinetic based optimization (GKBO)

The GKBO method we propose in the following enhances kinetic based optimization, which belongs to the class of consensus based algorithms, with ideas from genetic algorithms. To this end, we assume to have a population divided into two groups, similar to the parent and children populations in genetic algorithms. The two groups are specified with the help of labels leading to a modified KBO dynamic with followers and leaders. The dynamics is tailored in such a way that the the population clusters at the unique global minimum of the possibly non-convex objective function $\mathcal{E}\colon \mathbb{R}^d \to \mathbb{R}$. Hence, in the long time limit the dynamics solves the global optimization problem given by

$$\min_{x\in\mathbb{R}^d} \mathcal{E}(x), \tag{4.2.1}$$

where $\mathcal{E}$ is assumed to have a unique global minimizer. In more detail, each agent is described by its position $x \in \mathbb{R}^d$ varying continuously and a binary variable for the leadership-level $\lambda \in \{0,1\}$. In the following we identify leaders with $\lambda = 1$ and followers with $\lambda = 0$. We are interested in the evolution of the density function

$$f = f(x, \lambda, t), \qquad f : \mathbb{R}^d \times \{0,1\} \times \mathbb{R}_+ \to \mathbb{R}_+ \tag{4.2.2}$$

where $t \in \mathbb{R}^+$ denotes as usual the time variable. In the rest of the chapter we denote $f(x, \lambda, t)$ as $f_\lambda(x, t)$ and define

$$g(x,t) = \sum_{\lambda \in \{0,1\}} f_\lambda(x,t), \tag{4.2.3}$$

to be the density of the whole population at time t. We assume that $g(x,t)$ is normalized, hence a probability measure and introduce the fractions $\rho_\lambda \in [0,1]$ with $\lambda \in \{0,1\}$ s.t. $\rho_0 + \rho_1 = 1$ and $f_\lambda(x,t)/\rho_\lambda$ are probability measures as well.

4.2.1 Binary interaction between agents

A binary interaction of agents with state (x, λ) and (x_*, λ_*) is described by their post-interaction positions given by

$$\begin{aligned} x' &= x + (\nu_F(x_* - x) + \sigma_F D(x)\xi)\,\lambda_*(1-\lambda) + \nu_L(\hat{x}(t) - x)\lambda, \\ x'_* &= x_*, \end{aligned} \tag{4.2.4}$$

where σ_F, ν_F, ν_L, are positive parameters, ξ is a normally distributed random number and $D(x)$ is the diffusion matrix, defined to be either

$$D(x) = |\hat{x}(t) - x| Id_d, \tag{4.2.5}$$

in the case of isotropic diffusion [165], or

$$D(x) = diag\{(\hat{x}(t) - x)_1, \dots (\hat{x}(t) - x)_d\}, \tag{4.2.6}$$

in the case of anisotropic diffusion [52]. In equation (4.2.4)- (4.2.5)-(4.2.6) the term $\hat{x}(t)$ is the global estimate of the best position of the minimizer. The term $\hat{x}(t)$ is computed as a convex combination of particle locations weighted by the cost function according to Laplace principle [82]. In case we consider the whole population, we have

$$\hat{x}(t) = \frac{\int_{\mathbb{R}^d} x e^{-\alpha \mathcal{E}(x)} g(x,t) dx}{\int_{\mathbb{R}^d} e^{-\alpha \mathcal{E}(x)} g(x,t) dx}, \tag{4.2.7}$$

and Laplace principle yields

$$\lim_{\alpha \to \infty} \Big(-\frac{1}{\alpha} \int_{\mathbb{R}^d} e^{-\alpha \mathcal{E}(x)} g(x,t) dx \Big) = \inf_{x \in \text{supp } g(x,t)} \mathcal{E}(x). \tag{4.2.8}$$

In the section on numerical results we will also consider variants, where the weighted mean is computed with information of leaders or followers only. In (4.2.4) we use a compact form to describe both the followers and leaders dynamics. In particular, if we assume $\lambda = 0$ (and $\lambda_* = 1$) we retrieve the followers dynamics,

$$\begin{aligned} x' &= x + (\nu_F(x_* - x) + \sigma_F D(x)\xi)\,, \\ x'_* &= x_*, \end{aligned}$$

where x, x_* denote the pre-interaction positions of a follower and a leader, respectively. Followers are attracted toward a randomly selected leader and explore the space, searching for a possible position of the global minimizer. On the contrary, if $\lambda = 1$, we retrieve the leaders dynamics,

$$\begin{aligned} x' &= x + \nu_L(\hat{x}(t) - x), \\ x'_* &= x_*, \end{aligned}$$

where x, x_* denote the pre-interaction positions of two leaders. Hence, leaders do not explore the space but they relax their position toward the estimated position of the global minimizer at time t, which is given by $\hat{x}(t)$, defined in (4.2.7).

Remark 6. Note that (4.2.4) implies that no follower-follower interactions are considered, since if both λ and λ_* are equal to zero the agents keep their positions.

Remark 7. The goal of these algorithms is to find the global minimizer by letting agents concentrate on it. The exploration part slows down the convergence, especially for high value of the diffusion parameters. In the GKBO algorithm, just one of the two populations, the followers, is involved in the exploration process while the other agents, the leaders, are reaching quite immediately consensus on the estimated position of the global minimizer. Under these assumptions, the convergence process is speeded up, as we will see in the numerical experiments in Section 4.6.1.

4.2.2 Emergence of leaders and followers

The emergence of leaders and followers is realized with the help of a transition operator which acts as follows

$$\begin{aligned} \mathcal{T}[f_0](x,t) &= \pi_{L\to F}(x,\lambda;f)f_1(x,t) - \pi_{F\to L}(x,\lambda;f)f_0(x,t), \\ \mathcal{T}[f_1](x,t) &= \pi_{F\to L}(x,\lambda;f)f_0(x,t) - \pi_{L\to F}(x,\lambda;f)f_1(x,t), \end{aligned} \tag{4.2.9}$$

where $\pi_{F\to L}(\cdot)$ and $\pi_{L\to F}(\cdot)$ are certain transition rates, possibly depending on the current states. In the simplest case, if we assume that leaders emerge with fixed rate $\pi_{FL} > 0$ and return to the followers status with fixed rate $\pi_{LF} > 0$ then the transition rates reduce to

$$\pi_{L\to F} = \pi_{LF}, \qquad \pi_{F\to L} = \pi_{FL}. \tag{4.2.10}$$

However, we also cover more general cases, as for example proposed in [123], where each agent in position x is associated with a weight

$$\begin{aligned}\omega(x,t) &= \frac{1}{N}\#\left\{y \in \mathcal{A}(t) : |\mathcal{E}(x_{min}) - \mathcal{E}(y)| < |\mathcal{E}(x_{min}) - \mathcal{E}(x)|\right\} \\ &= \frac{1}{N}\sum_{\lambda\in\{0,1\}}\int_{\mathbb{R}^d} \chi_{[0,1)}\left(\frac{|\mathcal{E}(x_{min}(t)) - \mathcal{E}(y)|}{|\mathcal{E}(x_{min}(t)) - \mathcal{E}(x)|}\right) f(y,\lambda,t)dy,\end{aligned}$$

with $\chi_{[0,1)}(\cdot)$ denoting the characteristic function of the interval $[0,1)$ and

$$x_{min}(t) = \arg\min_{x\in\mathcal{A}(t)} \mathcal{E}(x),$$

where $\mathcal{A}(t)$ is the set of agents at time t. Assuming that agents with weight smaller than a certain threshold $\bar{\omega}$, which depends on the amount of leaders that we would like to generate, are in the leaders status while the others are in the followers status, then we can write the transition rates as follows

$$\pi_{L\to F} = \begin{cases} 1, & \text{if } \omega(x,t) > \bar{\omega}, \\ 0, & \text{if } \omega(x,t) \le \bar{\omega}, \end{cases} \qquad \pi_{F\to L} = \begin{cases} 0, & \text{if } \omega(x,t) \ge \bar{\omega}, \\ 1, & \text{if } \omega(x,t) < \bar{\omega}. \end{cases} \tag{4.2.11}$$

The evolution of the emergence and decay of leaders can be described by the master equation

$$\frac{d}{dt}\rho_\lambda(t) + \int_{\mathbb{R}^{2d}} \mathcal{T}[f](x,\lambda,t)\,dx = 0, \tag{4.2.12}$$

for $\lambda \in \{0,1\}$, with $\rho_\lambda(t) = \int_{\mathbb{R}^d} f_\lambda(x,t)dx$. From the above definition of the transition operator $\mathcal{T}[\cdot]$, it follows that

$$\frac{d}{dt}\sum_{\lambda\in\{0,1\}} \rho_\lambda(t) = 0. \tag{4.2.13}$$

In case of constant transition rates

$$\pi_{L\to F}(\cdot) = \pi_{LF}, \qquad \pi_{F\to L}(\cdot) = \pi_{FL}, \tag{4.2.14}$$

we can rewrite equation (4.2.12) as

$$\begin{aligned}\partial_t\rho_1(t) &= \pi_{FL}\rho_0(t) - \pi_{LF}\rho_1(t), \\ \partial_t\rho_0(t) &= \pi_{LF}\rho_1(t) - \pi_{FL}\rho_0(t).\end{aligned} \tag{4.2.15}$$

which allows us to calculate its stationary solution explicitly as

$$\rho_1^\infty = \frac{\pi_{FL}}{\pi_{LF}+\pi_{FL}}, \qquad \rho_0^\infty = \frac{\pi_{LF}}{\pi_{LF}+\pi_{FL}}. \tag{4.2.16}$$

Remark 8. The weighted strategy is inspired from the selection criterion of GA, where parents are chosen to be the agents in best position w.r.t. the cost function. In the numerical experiments we will also consider a mixed strategy, assuming that a certain percentage $\bar{p}$ of the total amount of leaders change their label according to the weighted strategy and the remaining ones changes their label randomly.

4.3 Derivation of the mean-field equation

Combining the interaction and transition dynamic described in the previous section, we obtain the evolution of the density function $f_\lambda(x,t)$ which is described by the integro-differential equation of Boltzmann-type

$$\partial_t f_\lambda(x,t) - \mathcal{T}[f_\lambda](x,t) = \sum_{\lambda_* \in \{0,1\}} Q(f_\lambda, f_{\lambda_*})(x,t), \tag{4.3.1}$$

where $\mathcal{T}[\cdot]$ is the transition operator and $Q(\cdot,\cdot)$ is the binary interaction operator defined as follows

$$Q(f_\lambda, f_{\lambda_*}) = \eta \int_{\mathbb{R}^{2d}} \left(\frac{1}{J} f_\lambda(x',t) f_{\lambda_*}(x'_*,t) - f_\lambda(x,t) f_{\lambda_*}(x_*,t) \right) dx dx_*, \tag{4.3.2}$$

where (x', x'_*) are the pre-interaction positions generated by the couple (x, x_*) after the interaction (4.2.4). The term J denotes the Jacobian of the transformation $(x, x_*) \to (x', x'_*)$ and $\eta > 0$ is a constant relaxation rate representing the interaction frequency. To obtain a weak-formulation, we consider a test function $\phi(x)$ and rewrite the collision operator

$$\int_{\mathbb{R}^d} Q(f_\lambda, f_{\lambda_*})(x,t)\phi(x)dx = \eta \int_{\mathbb{R}^{2d}} \left(\phi(x') - \phi(x)\right) f_{\lambda_*}(x_*,t) f_\lambda(x,t) dx dx_*. \tag{4.3.3}$$

Hence, the weak form of (4.3.1) reads

$$\begin{aligned} \frac{\partial}{\partial t} \int_{\mathbb{R}^d} f_\lambda(x,t)\phi(x)dx - \int_{\mathbb{R}^d} \mathcal{T}[f_\lambda](x,t)\phi(x)dx = \\ \eta \sum_{\lambda_* \in \{0,1\}} \left\langle \int_{\mathbb{R}^{2d}} \Big[\phi(x') - \phi(x)\Big] f_\lambda(x,t) f_{\lambda_*}(x_*,t) dx dx_* \right\rangle. \end{aligned} \tag{4.3.4}$$

To simplify the computations, we assume to have constant transition rates (4.2.14) and to be in the quasi-stationary state ρ_λ^∞ i.e. $\rho_\lambda \approx \rho_\lambda^\infty$ for any $\lambda \in \{0,1\}$ as in (4.2.16). Moreover, we introduce the scaling parameter $\varepsilon > 0$ and consider

$$\nu_F \to \frac{\nu_F}{\rho_1}\varepsilon, \qquad \nu_L \to \frac{\nu_L}{\rho_1}\varepsilon, \qquad \sigma_F \to \frac{\sigma_F}{\sqrt{\rho_1}}\sqrt{\varepsilon}, \qquad \eta \to \frac{1}{\varepsilon}. \tag{4.3.5}$$

This scaling corresponds to the case where the interaction kernel concentrates on binary interactions producing very small changes in the agents position but at the same time the number of interactions becomes very large.

To obtain the mean-field equation, we consider the Taylor expansion of the test function $\phi(x')$ centred in x given by

$$\phi(x') - \phi(x) = \nabla_x \phi(x) \cdot (x' - x) + \frac{1}{2}\Delta_x \phi(x)(x' - x)^2 + \mathcal{O}(\varepsilon^2), \tag{4.3.6}$$

and use it to rewrite (4.3.4) as follows

$$\frac{\partial}{\partial t}\int_{\mathbb{R}^d} f_\lambda(x,t)\phi(x)dx - \int_{\mathbb{R}^d} \mathcal{T}[f_\lambda](x,t)\phi(x)dx = \sum_{\lambda_*\in\{0,1\}}\left\{\int_{\mathbb{R}^{2d}}\left(\frac{\nu_F}{\rho_1}(x_*-x)\lambda_*(1-\lambda)+\frac{\nu_L}{\rho_1}(\hat{x}(t)-x)\lambda\right)\cdot\nabla_x\phi(x)df_\lambda df_{\lambda_*} + \frac{\sigma_F^2}{2}\int_{\mathbb{R}^{2d}} D^2(x)(1-\lambda)^2\lambda_*^2\Delta_x\phi(x)df_\lambda df_{\lambda_*}\right\}+\mathcal{O}(\varepsilon), \tag{4.3.7}$$

where for simplicity we write $df_\lambda = f_\lambda(x,t)dx$ and $df_{\lambda_*} = f_{\lambda_*}(x_*,t)dx_*$. Now, taking the limit $\varepsilon\to 0$, integrating by parts and rewriting the equation in strong form yields

$$\begin{aligned}\frac{\partial}{\partial t}f_0(x,t)-\mathcal{T}[f_0](x,t) &= \frac{\sigma_F^2}{2}\Delta_x\Big[D^2(x)f_0(x,t)\Big]-\nu_F\nabla_x\cdot\Big[\Big(\frac{m_1(t)}{\rho_1}-x\Big)f_0(x,t)\Big],\\ \frac{\partial}{\partial t}f_1(x,t)-\mathcal{T}[f_1](x,t) &= -\frac{\nu_L}{\rho_1}\nabla_x\cdot\Big[\Big(\hat{x}(t)-x\Big)f_1(x,t)\Big],\end{aligned} \tag{4.3.8}$$

where $D(x)$ is the diffusion matrix defined in (4.2.6)-(4.2.5), $\hat{x}(t)$ is the global estimate of the global minimizer at time t defined in equation (4.2.7) and

$$m_1(t) = \int_{\mathbb{R}^{2d}} x f_1(x,t)dx \tag{4.3.9}$$

denotes the centre of mass of the leaders at time t.

Remark 9. Multiplying both side of the second equation in (4.3.8) by x/ν_L integrating and taking the formal limit $\nu_L\to+\infty$, we get

$$\frac{m_1(t)}{\rho_1} = \hat{x}(t).$$

Plugging it into the first equation in (4.3.8), assuming $\mathcal{T}[f_0](x,t)=0$, we recover the equation that governs the dynamics in absence of leaders that is

$$\frac{\partial}{\partial t}f_0(x,t) = \frac{\sigma_F^2}{2}\Delta_x\Big[D(x)^2 f_0(x,t)\Big]-\nu_F\nabla_x\cdot\Big[\Big(\hat{x}(t)-x\Big)f_0(x,t)\Big]. \tag{4.3.10}$$

To summarize, the diagram in Figure 4.1 describes the relation between the three algorithms at the particle and mean field level.

4.4 Moments estimates and convergence to the global minimum

Following the idea introduced in [21, 165] we provide moments estimates, showing that the variance decreases exponentially to zero, and we prove the convergence of the method toward the position of the global minimum. In this section we study the behaviour of the two population dynamic, we therefore assume throughout this section $\rho_0,\rho_1>0$.

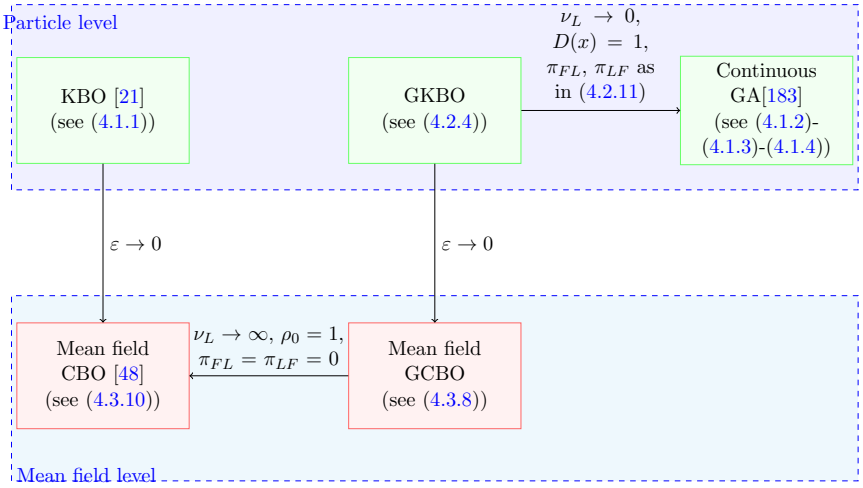

Figure 4.1: Diagram describing the relation between the KBO, the GKBO with weighted and random strategies and the GA.

4.4.1 Evolution of the moment estimates

We define the first two moments of the total population by

$$m(t) = m_0(t) + m_1(t), \qquad e(t) = e_0(t) + e_1(t), \tag{4.4.1}$$

respectively, where

$$\begin{aligned} m_0(t) &= \int_{\mathbb{R}^d} x\ f_0(x,t)dx, \qquad e_0(t) = \int_{\mathbb{R}^d} |x|^2 f_0(x,t)dx, \\ m_1(t) &= \int_{\mathbb{R}^d} x\ f_1(x,t)dx, \qquad e_1(t) = \int_{\mathbb{R}^d} |x|^2 f_1(x,t)dx, \end{aligned} \tag{4.4.2}$$

are the first two moments of the subpopulations $f_\lambda(x,t)$ for $\lambda \in \{0,1\}$ and

$$V(t) = v_0(t) + v_1(t), \tag{4.4.3}$$

the sum of the variances of the subpopulations given by

$$v_0(t) = \int_{\mathbb{R}^d} \left|x - \frac{m_0}{\rho_0}\right|^2 f_0(x,t)dx, \qquad v_1(t) = \int_{\mathbb{R}^d} \left|x - \frac{m_1}{\rho_1}\right|^2 f_1(x,t)dx. \tag{4.4.4}$$

Remark 10. For the following computations it is helpful to have in mind that

$$m(t) = \int_{\mathbb{R}^d} x(f_0(x,t) + f_1(x,t))dx, \qquad e(t) = \int_{\mathbb{R}^d} |x|^2(f_0(x,t) + f_1(x,t))dx,$$

but due to the nonlinearity

$$V(t) \neq \int_{\mathbb{R}^d} |x - m(t)|^2 (f_0(x,t) + f_1(x,t)) dx.$$

Proposition 2. *Let us assume the transitions have equilibrated, that is, $\rho_0 \equiv \rho_0^\infty$ and $\rho_1 \equiv \rho_1^\infty$. Furthermore let $\mathcal{E}(x)$ positive and bounded for all $x \in \mathbb{R}^d$, in particular, there exist constants $\underline{\mathcal{E}}, \overline{\mathcal{E}} > 0$ such that*

$$\underline{\mathcal{E}} := \inf_x \mathcal{E}(x) \leq \mathcal{E}(x) \leq \sup_x \mathcal{E}(x) := \overline{\mathcal{E}}, \tag{4.4.5}$$

and define $\tilde{\sigma} = k\sigma_F^2 b_{\underline{\mathcal{E}}}$, with $b_{\underline{\mathcal{E}}} = \exp(\alpha(\bar{\mathcal{E}} - \underline{\mathcal{E}}))$, where $k = d$ in the case of isotropic diffusion and $k = 1$ in the case of anisotropic diffusion. If

$$\nu_F = \nu_L, \qquad \nu_F > \max\left\{\frac{\tilde{\sigma}}{2}, \frac{\rho_1}{2}\right\}, \tag{4.4.6}$$

it holds

$$\begin{aligned} &\frac{d}{dt} m(t) = \nu_F (\hat{x} - m)(t), \\ &\frac{d}{dt} V(t) \leq (-2\nu_F + \tilde{\sigma}) V(t) + (\tilde{\sigma}\rho_0\rho_1 - \pi_{LF}\rho_1 - \pi_{FL}\rho_0) \left(\frac{m_0(t)}{\rho_0} - \frac{m_1(t)}{\rho_1}\right)^2. \end{aligned} \tag{4.4.7}$$

Proof. Let us define for simplicity

$$\begin{aligned} M_\lambda(t) = \frac{1}{\rho_\lambda} \int_{\mathbb{R}^d} x \; f_\lambda(x,t) dx, \qquad E_\lambda(t) = \frac{1}{\rho_\lambda} \int_{\mathbb{R}^d} |x|^2 f_\lambda(x,t) dx, \\ V_\lambda(t) = \frac{1}{\rho_\lambda} \int_{\mathbb{R}^d} \left|x - M_\lambda\right|^2 f_\lambda(x,t) dx, \end{aligned} \tag{4.4.8}$$

for any $\lambda \in \{0, 1\}$ such that

$$\begin{aligned} m(t) = \rho_0 M_0(t) + \rho_1 M_1(t), \qquad e(t) = \rho_0 E_0(t) + \rho_1 E_1(t), \\ V(t) = \rho_0 V_0(t) + \rho_1 V_1(t). \end{aligned}$$

We begin by computing the evolution of the first moments

$$\frac{d}{dt} m(t) = \rho_0 \frac{d}{dt} M_0(t) + \rho_1 \frac{d}{dt} M_1(t). \tag{4.4.9}$$

For the first term of (4.4.9) we obtain

$$\begin{aligned} \rho_0 \frac{d}{dt} M_0(t) = &\int_{\mathbb{R}^d} x \partial_t f_0 = \\ &= \int_{\mathbb{R}^d} x \Big[- \pi_{LF} f_1 + \pi_{FL} f_0 + \\ &- \nabla_x \cdot (\nu_F (M_1 - x) f_0) + \frac{\sigma_F^2}{2} \Delta_x \left(D^2(x) f_0\right) \Big] dx = \\ &= -\pi_{LF}\rho_1 M_1(t) + \pi_{FL}\rho_0 M_0(t) + \rho_0 \nu_F (M_1 - M_0)(t). \end{aligned} \tag{4.4.10}$$

and for the second term in (4.4.9) it holds

$$\rho_1 \frac{d}{dt} M_1(t) = \int_{\mathbb{R}^d} x \partial_t f_1 = \int_{\mathbb{R}^d} x \Big[\pi_{LF} f_1 - \pi_{FL} f_0 - \nabla_x \cdot \left(\frac{\nu_L}{\rho_1} (\hat{x} - x) f_1 \right) \Big] dx = \\ = \pi_{LF} \rho_1 M_1(t) - \pi_{FL} \rho_0 M_0(t) + \nu_L (\hat{x} - M_1)(t). \tag{4.4.11}$$

Together this yields

$$\frac{d}{dt} m(t) = \nu_F \rho_0 (M_1 - M_0)(t) + \nu_L (\hat{x} - M_1)(t), \tag{4.4.12}$$

and recalling the definition of $M_0(t)$ and $M_1(t)$ in (4.4.8) we get

$$\frac{d}{dt} m(t) = \nu_L \hat{x}(t) - \nu_F m(t) + \left(\nu_F \frac{\rho_0}{\rho_1} + \nu_F - \frac{\nu_L}{\rho_1} \right) m_1(t). \tag{4.4.13}$$

By the first assumption in (4.4.6) we can recover the first equation of the statement.

For $V(t)$ we have

$$\frac{d}{dt} V(t) = \rho_0 \frac{d}{dt} V_0(t) + \rho_1 \frac{d}{dt} V_1(t). \tag{4.4.14}$$

We investigate the terms separately. First, we obtain

$$\frac{d}{dt} V_0(t) = \frac{1}{\rho_0} \frac{d}{dt} \int_{\mathbb{R}^d} |x - M_0(t)|^2 df_0 = \\ \underbrace{\frac{2}{\rho_0} \int_{\mathbb{R}^d} \left(x - M_0(t), -\frac{d}{dt} M_0(t) \right) df_0}_{=: I_0} + \underbrace{\frac{1}{\rho_0} \int_{\mathbb{R}^d} |x - M_0(t)|^2 \partial_t f_0}_{=: I_1}. \tag{4.4.15}$$

We note that I_0 vanishes, since $2\rho_0^{-1} \int_{\mathbb{R}^d} x - M_0(t) df_0 = 0$. We divide I_1 into its drift, diffusion and transition parts to obtain

$$I_1 =: I_1^0 + I_1^1 + I_1^2, \tag{4.4.16}$$

with

$$I_1^0 = \frac{1}{\rho_0} \int_{\mathbb{R}^d} |x - M_0(t)|^2 \left(-\nu_F \nabla_x \cdot \left((M_1(t) - x) f_0 \right) \right) dx = \\ = \frac{2\nu_F}{\rho_0} \int_{\mathbb{R}^d} (x - M_0(t))(M_1(t) - x) df_0 = \\ = 2\nu_F \left(M_0(t) M_1(t) - E_0(t) - M_0(t) M_1(t) + M_0^2(t) \right) = -2\nu_F V_0, \tag{4.4.17}$$

and, by an application of Jensen inequality we get

$$\begin{aligned}
I_1^1 =& \frac{1}{\rho_0}\int_{\mathbb{R}^d} |x - M_0(t)|^2 \Delta_x \left(\frac{\sigma_F^2}{2} D^2(x) f_0\right) dx = \\
&= \frac{\sigma_F^2}{2\rho_0}\int_{\mathbb{R}^d} k|\hat{x}(t) - x|^2 df_0 = \\
&= \frac{\sigma_F^2}{2\rho_0}\int_{\mathbb{R}^d} k \int_{\mathbb{R}^d} \left|\frac{\int_{\mathbb{R}^d}(y-x)e^{-\alpha\mathcal{E}(y)}g(y)dy}{\int_{\mathbb{R}^d} e^{-\alpha\mathcal{E}(y)}g(y)dy}\right|^2 df_0 = \\
&\leq \frac{\tilde{\sigma}}{\rho_0}\int_{\mathbb{R}^{2d}} |y-x|^2 g(y) f_0(x) dx dy = \\
&= \frac{\tilde{\sigma}}{\rho_0}\left(\rho_0 E_0(t) + \rho_1 E_1(t) - (\rho_0 M_0(t) + \rho_1 M_1(t))^2\right) = \\
&= \frac{\tilde{\sigma}}{\rho_0}\left(V(t) + \rho_0\rho_1 \left(M_0(t) - M_1(t)\right)^2\right),
\end{aligned} \tag{4.4.18}$$

and finally,

$$\begin{aligned}
I_1^2 =& \frac{1}{\rho_0}\int_{\mathbb{R}^d} |x - M_0(t)|^2 \left(-\pi_{LF} f_1 + \pi_{FL} f_0\right) dx = \\
&- \frac{\pi_{LF}}{\rho_0}\int_{\mathbb{R}^d} \left(x^2 + M_0^2(t) - 2x M_0(t)\right) df_1 + \pi_{FL} V_0(t) = \\
&- \frac{\pi_{LF}}{\rho_0}\left(\rho_1 V_1(t) + \rho_1 (M_0 - M_1)^2(t)\right) + \pi_{FL} V_0(t),
\end{aligned} \tag{4.4.19}$$

where we add and subtract $\rho_1 M_1^2(t)$ in the last term of I_1^2.

For $V_1(t)$ we have

$$\begin{aligned}
\frac{d}{dt}V_1(t) =& \frac{1}{\rho_1}\frac{d}{dt}\int_{\mathbb{R}^d} |x - M_1(t)|^2 df_1 = \\
&\underbrace{\frac{2}{\rho_1}\int_{\mathbb{R}^d}\left(x - M_1(t), -\frac{d}{dt}M_1(t)\right) df_0}_{=:I_2} + \underbrace{\frac{1}{\rho_1}\int_{\mathbb{R}^d} |x - M_1(t)|^2 \partial_t f_1}_{=:I_3}.
\end{aligned} \tag{4.4.20}$$

Similarly to the case I_0 one can easily conclude that I_2 vanishes. We divide I_3 into the drift and transition part to obtain

$$I_3 = I_3^0 + I_3^1, \tag{4.4.21}$$

with

$$\begin{aligned}
I_3^0 =& \frac{1}{\rho_1}\int_{\mathbb{R}^d} |x - M_1(t)|^2 \left(-\frac{\nu_L}{\rho_1}\nabla_x \cdot \left(\left(\hat{x}(t) - x\right) f_1\right)\right) dx = \\
&= \frac{2\nu_L}{\rho_1^2}\int_{\mathbb{R}^d} (x - M_1(t))(\hat{x}(t) - x) df_1 = \\
&= \frac{2\nu_L}{\rho_1}\left(M_1(t)\hat{x}(t) - E_1(t) - M_1(t)\hat{x}(t) + M_1^2(t)\right) = -2\frac{\nu_L}{\rho_1}V_1,
\end{aligned} \tag{4.4.22}$$

and

$$
\begin{aligned}
I_3^1 =& \frac{1}{\rho_1}\int_{\mathbb{R}^d} |x - M_1(t)|^2 \left(\pi_{LF} f_1 - \pi_{FL} f_0\right) dx = \\
&= -\frac{\pi_{FL}}{\rho_1}\int_{\mathbb{R}^d} \left(x^2 + M_1^2(t) - 2xM_1(t)\right) df_0 + \pi_{LF} V_1(t) = \\
&= -\frac{\pi_{FL}}{\rho_1}\left(\rho_0 V_0(t) + \rho_0 (M_0 - M_1)^2(t)\right) + \pi_{LF} V_1(t),
\end{aligned} \tag{4.4.23}
$$

where we add and subtract $\rho_0 M_0^2(t)$ in the last term of I_3^2.

Altogether, we get

$$
\begin{aligned}
\frac{d}{dt}V(t) \leq& \rho_1 \left(2\nu_F + \rho_0 - \frac{2\nu_F}{\rho_1}\right) V_0(t) + \left(-2\nu_F + \tilde{\sigma}\right) V(t) + \\
&+ \left(\tilde{\sigma}\rho_0\rho_1 - \pi_{LF}\rho_1 - \pi_{FL}\rho_0\right) \left(M_0 - M_1\right)^2 (t).
\end{aligned} \tag{4.4.24}
$$

Using the assumptions, we recover the second inequality in (4.4.7). □

Corollary 1. *Let the assumptions of Proposition 2 hold, and in addition suppose that*

$$
\nu_F > \max\left\{\frac{\pi_{LF}\rho_1}{\rho_0(1 - b_{\bar{\mathcal{E}}}\rho_1)}, \frac{\pi_{FL}}{b_{\bar{\mathcal{E}}}\rho_0}\right\}, \qquad \text{with } b_{\bar{\mathcal{E}}} = e^{\alpha(\underline{\mathcal{E}} - \bar{\mathcal{E}})}. \tag{4.4.25}
$$

Then it holds

$$
\left|\frac{m_0(t)}{\rho_0} - \frac{m_1(t)}{\rho_1}\right|^2 \to 0, \quad V(t) \to 0, \qquad \text{as } t \to \infty. \tag{4.4.26}
$$

Proof. Let us first study the behavior of $|M_0 - M_1|^2(t)$. We have

$$
\begin{aligned}
&\frac{d}{dt}|M_0 - M_1|^2(t) = 2\left(M_0 - M_1\right)(t)\frac{d}{dt}\left(M_0 - M_1\right)(t) \leq \\
&\leq -2\nu_F |M_0 - M_1|^2(t) + 2(M_0 - M_1)(t)\left(\mathcal{C}_1 M_1(t) - \mathcal{C}_0 M_0(t)\right) = \\
&= -2\nu_F |M_0 - M_1|^2(t) - 2\begin{bmatrix} M_0(t) \\ M_1(t) \end{bmatrix}^T \begin{bmatrix} \mathcal{C}_0 & -\mathcal{C}_0 \\ -\mathcal{C}_1 & \mathcal{C}_1 \end{bmatrix} \begin{bmatrix} M_0(t) \\ M_1(t) \end{bmatrix},
\end{aligned} \tag{4.4.27}
$$

with

$$
\mathcal{C}_0 = -\frac{\pi_{FL} - \nu_F\rho_0 b_{\bar{\mathcal{E}}}}{\rho_1}, \qquad \mathcal{C}_1 = \frac{-\pi_{LF}\rho_1 + \nu_F\rho_0(1 - \rho_1 b_{\bar{\mathcal{E}}})}{\rho_0\rho_1}, \tag{4.4.28}
$$

and we used equations (4.4.10)-(4.4.11) and the estimate

$$
\hat{x}(t) = \frac{\int_{\mathbb{R}^d} x e^{-\alpha\mathcal{E}(x)} g(x,t) dx}{\int_{\mathbb{R}^d} e^{-\alpha\mathcal{E}(x)} g(x,t) dx} \geq \frac{e^{\alpha\underline{\mathcal{E}}}}{e^{\alpha\bar{\mathcal{E}}}}\int_{\mathbb{R}^d} x\ g(x,t) dx := b_{\bar{\mathcal{E}}}\ m(t). \tag{4.4.29}
$$

Note that, if condition (4.4.25) holds then $\mathcal{C}_0$, $\mathcal{C}_1 > 0$ and so

$$-2\begin{bmatrix} M_0(t) \\ M_1(t) \end{bmatrix}^T \begin{bmatrix} \mathcal{C}_0 & -\mathcal{C}_0 \\ -\mathcal{C}_1 & \mathcal{C}_1 \end{bmatrix} \begin{bmatrix} M_0(t) \\ M_1(t) \end{bmatrix} \leq 0, \tag{4.4.30}$$

since the above matrix is weakly diagonal dominant and hence positive semidefinite. Altogether, we obtain the estimate

$$\frac{d}{dt}|M_0 - M_1|^2(t) \leq -2\nu_F |M_0 - M_1|^2(t). \tag{4.4.31}$$

and an application of Grönwall lemma yields

$$|M_0 - M_1|^2(t) \leq |M_0 - M_1|^2(0) e^{-2\nu_F t}, \tag{4.4.32}$$

which allow us to conclude $|M_0 - M_1|^2(t) \to 0$ as $t \to \infty$.

In particular, this implies

$$|M_0 - M_1|^2(t) \leq |M_0 - M_1|^2(0), \tag{4.4.33}$$

which helps us to show the second statment. Indeed, we rewrite the second inequality in (4.4.7) in integral form as

$$V(t) \leq V(0) + \mathcal{C}_v^0 |M_0 - M_1|^2(0) \int_0^t ds - \mathcal{C}_v \int_0^t V(s) ds, \tag{4.4.34}$$

with $\mathcal{C}_v^0 = \tilde{\sigma}\rho_0\rho_1 - \pi_{LF}\rho_1 - \pi_{FL}\rho_0$ and $\mathcal{C}_v = 2\nu_F - \tilde{\sigma}$. Moreover, we note that

$$t \to V(0) + \mathcal{C}_v^0 |M_0 - M_1|^2(0) t,$$

is a non-decreasing function. Hence, again using Grönwall lemma, we get

$$V(t) \leq \Big[V(0) + \mathcal{C}_v^0 |M_0 - M_1|^2(0) t \Big] e^{-\mathcal{C}_v t}, \tag{4.4.35}$$

which implies $V(t) \to 0$ as $t \to \infty$. □

The fact that $V(t)$ vanishes in the limit $t \to \infty$ allows us to conclude that the crowd concentrates. However, the position of the concentration point is unknown. This position is quantified in the following section.

4.4.2 Convergence to the global minimum

In this section, we determine the conditions under which the mean value of the population is a reasonable approximation of the global minimizer.

Proposition 3. *Suppose the assumptions of Proposition 2 hold. Further, we assume that* $\mathcal{E} \in C^2(\mathbb{R}^d)$ *and that there exist constants* $c_1, c_2 > 0$ *such that*

$$\sup_{y\in\mathbb{R}^2} |\nabla\mathcal{E}(y)| \leq c_1, \qquad \sup_{y\in\mathbb{R}^2} |\Delta\mathcal{E}(y)| \leq c_2, \tag{4.4.36}$$

and that the initial condition is well-prepared in the sense that the minimizer of $\mathcal{E}$ *is in the support of the initial population and*

$$\frac{\mu}{M_\alpha^2(0)} \leq \frac{3}{4}, \tag{4.4.37}$$

is satisfied with

$$M_\alpha(t) = \int_{\mathbb{R}^d} e^{-\alpha\mathcal{E}(x)} g(x)dx, \tag{4.4.38}$$

and

$$\begin{aligned}\mu =& 2\alpha e^{-\alpha\underline{\mathcal{E}}}\left[c_1\sqrt{2}\left(\nu_F + \frac{\nu_F}{\rho_1}\right) + c_2\sigma_F^2 k\right]\cdot \\ &\cdot\left[\max\left\{\frac{1}{\mathcal{C}_v}V(0) + \gamma_m\left(\frac{\mathcal{C}_v^0}{\mathcal{C}_v^2} + \frac{\rho_0\rho_1}{2\nu_F}\right), \frac{2}{\mathcal{C}_v}V(0) + \frac{4\mathcal{C}^*}{\mathcal{C}_v} + \frac{\sqrt{\rho_0\rho_1\gamma_m}}{\nu_F}\right\}\right],\end{aligned} \tag{4.4.39}$$

with

$$\mathcal{C}_v^0 = \tilde{\sigma}\rho_0\rho_1 - \pi_{LF}\rho_1 - \pi_{FL}\rho_0, \qquad \mathcal{C}_v = 2\nu_F - \tilde{\sigma}, \qquad \gamma_m = \left(\frac{m_0(0)}{\rho_0} - \frac{m_1(0)}{\rho_1}\right)^2,$$

and $\mathcal{C}^*$ *is the maximal value of*

$$t \to e^{-\frac{\mathcal{C}_v}{4}t}\sqrt{\mathcal{C}_v^0\gamma_m t}.$$

Then there exists $\tilde{x} \in \mathbb{R}^d$ *such that* $m(t) \to \tilde{x}$ *as* $t \to \infty$ *and*

$$\mathcal{E}(\tilde{x}) = \underline{\mathcal{E}} + r(\alpha) + \frac{\log 2}{\alpha}, \tag{4.4.40}$$

where $r(\alpha) = -\frac{1}{\alpha}\log(M_\alpha(0)) - \underline{\mathcal{E}} \to 0$, *as* $\alpha \to \infty$.

Proof. First, we show

$$\left|\frac{d}{dt}m(t)\right| \to 0, \tag{4.4.41}$$

as $t \to \infty$. To this end, we rewrite

$$\Big|\frac{d}{dt}m(t)\Big| = \Big|\nu_F \int_{\mathbb{R}^d} \left(\frac{e^{-\alpha\mathcal{E}(x)}}{M_\alpha(t)} - 1\right) x\, g(x)dx\Big|, \tag{4.4.42}$$

where we use the first estimate in (4.4.7) and the definition of $\hat{x}(t)$. Applying Jensen inequality and using the estimate

$$\hat{x}(t) \leq e^{-\alpha(\underline{\mathcal{E}}-\bar{\mathcal{E}})} \int_{\mathbb{R}^d} xg(x,t)dx := b_{\underline{\mathcal{E}}} m(t),$$

we get

$$\begin{aligned}
\Big|\frac{d}{dt}m(t)\Big| &= \frac{\nu_F}{M_\alpha(t)}\Big|\int_{\mathbb{R}^{2d}} xe^{-\alpha\mathcal{E}(x)}g(x)g(x_*)dxdx_* - \int_{\mathbb{R}^{2d}} x_* e^{-\alpha\mathcal{E}(x)}g(x)g(x_*)dxdx_*\Big| = \\
&= \frac{\nu_F}{M_\alpha(t)}\Big|\int_{\mathbb{R}^{2d}} (x-x_*)e^{-\alpha\mathcal{E}(x)}g(x)g(x_*)dxdx_*\Big| \\
&\leq \frac{\nu_F}{M_\alpha(t)}\int_{\mathbb{R}^{2d}} |x-x_*|e^{-\alpha\mathcal{E}(x)}g(x)g(x_*)dxdx_* \leq \\
&\leq b_{\underline{\mathcal{E}}}\nu_F\left(\int_{\mathbb{R}^{2d}} |x-x_*|^2 g(x)g(x_*)dxdx_*\right)^{1/2} = \\
&= b_{\underline{\mathcal{E}}}\nu_F\sqrt{2}\left(\rho_0 E_0(t) + \rho_1 E_1(t) - (\rho_0 M_0(t) + \rho_1 M_1(t))^2\right)^{1/2} = \\
&= b_{\underline{\mathcal{E}}}\nu_F\sqrt{2}\left(V(t) + \rho_0\rho_1(M_0 - M_1)^2(t)\right)^{1/2} \to 0, \qquad \text{as } t \to \infty,
\end{aligned} \tag{4.4.43}$$

since both, $V(t)$ and $(M_0 - M_1)^2(t)$, go to zero as $t \to \infty$. Thus, there exists $\tilde{x} \in \mathbb{R}^d$ such that

$$\tilde{x} = m(0) + \int_0^t \frac{d}{ds}m(s)ds = \lim_{t\to\infty} m(t). \tag{4.4.44}$$

Let us now focus on the term $M_\alpha(t)$

$$\frac{d}{dt}M_\alpha^2(t) = 2M_\alpha(t)\frac{d}{dt}M_\alpha(t) = 2M_\alpha(t)\int_{\mathbb{R}^d} e^{-\alpha\mathcal{E}(x)}\partial_t g(x,t)dx, \tag{4.4.45}$$

with

$$\begin{aligned}
\partial_t g(x,t) = \partial_t f_0(x,t) + \partial_t f_1(x,t) &= -\nu_F\nabla_x\cdot\Big[(M_1 - x)f_0(x,t)\Big] + \\
&+ \frac{\sigma_F^2}{2}\Delta_x\Big[D^2(x)f_0(x,t)\Big] - \frac{\nu_F}{\rho_1}\nabla_x\cdot\Big[(\hat{x}(t) - x)f_1(x,t)\Big],
\end{aligned} \tag{4.4.46}$$

where we recall that we assume $\nu_L = \nu_F$ and

$$\sum_{\lambda_\in\{0,1\}} \mathcal{T}[f_\lambda](x,t) = 0.$$

We consider the terms separately to obtain

$$\begin{aligned}
I_1 =& -\nu_F \int_{\mathbb{R}^d} e^{-\alpha\mathcal{E}(x)} \nabla_x \cdot \Big[(M_1 - x) f_0\Big] dx = \\
&= -\nu_F \alpha \int_{\mathbb{R}^{2d}} e^{-\alpha\mathcal{E}(x)} \nabla\mathcal{E}(x)(x_* - x) df_0 df_1 \geq \\
&\geq -\nu_F \alpha e^{-\alpha\underline{\mathcal{E}}} c_1 \frac{M_\alpha(t)}{M_\alpha(t)} \int_{\mathbb{R}^{2d}} |x_* - x| dg dg_* \geq \\
&\geq -\nu_F \alpha \frac{e^{-2\alpha\underline{\mathcal{E}}}}{M_\alpha(t)} c_1 \left(\int_{\mathbb{R}^{2d}} |x_* - x|^2 dg dg_* \right)^{1/2} \geq \\
&\geq -\nu_F \alpha \frac{e^{-2\alpha\underline{\mathcal{E}}}}{M_\alpha(t)} c_1 \sqrt{2} \left[V(t) + \rho_0\rho_1 (M_0 - M_1)^2(t) \right]^{1/2}, \qquad (4.4.47)
\end{aligned}$$

$$\begin{aligned}
I_2 =& \frac{\sigma_F^2}{2} \int_{\mathbb{R}^d} e^{-\alpha\mathcal{E}(x)} \Delta_x \Big[D^2(x) f_0 \Big] dx = \\
&= -\frac{\sigma_F^2}{2} \alpha \int_{\mathbb{R}^d} e^{-\alpha\mathcal{E}(x)} \Delta\mathcal{E}(x) k |\hat{x}(t) - x|^2 df_0 + \\
&+ \frac{\sigma_F^2}{2} \alpha^2 \int_{\mathbb{R}^{2d}} e^{-\alpha\mathcal{E}(x)} \nabla_x\mathcal{E}(x) \otimes \nabla_x\mathcal{E}(x) k |\hat{x}(t) - x|^2 df_0 \geq \\
&\geq -\frac{\alpha\sigma_F^2}{2} k c_2 e^{-\alpha\underline{\mathcal{E}}} \int_{\mathbb{R}^d} |\hat{x}(t) - x|^2 dg \geq \\
&\geq -\frac{\alpha\sigma_F^2}{2} k c_2 e^{-\alpha\underline{\mathcal{E}}} \int_{\mathbb{R}^{2d}} \int |x_* - x|^2 \frac{e^{-\alpha\mathcal{E}(x_*)}}{M_\alpha(t)} dg dg_* \geq \\
&\geq -\frac{\alpha\sigma_F^2}{2} k c_2 \frac{e^{-2\alpha\underline{\mathcal{E}}}}{M_\alpha(t)} \int_{\mathbb{R}^{2d}} |x_* - x|^2 dg dg_* \geq \\
&\geq -\frac{\alpha\sigma_F^2}{2} k c_2 \frac{e^{-2\alpha\underline{\mathcal{E}}}}{M_\alpha(t)} \Big[V(t) + \rho_0\rho_1 (M_0 - M_1)^2(t) \Big], \qquad (4.4.48)
\end{aligned}$$

and

$$
\begin{aligned}
I_3 = &- \frac{\nu_F}{\rho_1}\int_{\mathbb{R}^d} e^{-\alpha\mathcal{E}(x)}\nabla_x\cdot\Big[(\hat{x}(t)-x)f_1\Big]dx \geq \\
&\geq -\frac{\alpha\nu_F}{\rho_1}c_1 e^{-\alpha\underline{\mathcal{E}}}\int|\hat{x}(t)-x|dg \geq \\
&\geq -\frac{\alpha\nu_F}{\rho_1}c_1\frac{e^{-2\alpha\underline{\mathcal{E}}}}{M_\alpha(t)}\left(\int_{\mathbb{R}^{2d}}|x_*-x|^2 dgdg_*\right)^{1/2} \geq \\
&\geq -\frac{\alpha\nu_F}{\rho_1}c_1\frac{e^{-2\alpha\underline{\mathcal{E}}}}{M_\alpha(t)}\Big[V(t)+\rho_0\rho_1(M_0-M_1)^2(t)\Big]^{1/2}, \qquad (4.4.49)
\end{aligned}
$$

where we use assumption (4.4.36), we integrate by parts, use Jensen inequality and the previous estimates. Altogether, we estimate (4.4.45) as follows

$$
\begin{aligned}
\frac{dM_\alpha(t)}{dt} \geq &-2\alpha e^{-2\alpha\underline{\mathcal{E}}}\Big[c_1\sqrt{2}\nu_F\left(1+\frac{1}{\rho_1}\right)\left(V(t)+\rho_0\rho_1(M_0-M_1)^2(t)\right)^{1/2}+ \\
&+c_2\sigma_F^2 k\left(V(t)+\rho_0\rho_1(M_0-M_1)^2(t)\right)\Big]. \qquad (4.4.50)
\end{aligned}
$$

Using the estimates for the mean and variance in (4.4.32)-(4.4.35) and integrating equation (4.4.50) we get

$$
\begin{aligned}
M_\alpha^2(t) \geq &M_\alpha^2(0) - 2\alpha e^{-\alpha\underline{\mathcal{E}}}\left[c_1\sqrt{2}\nu_F\left(1+\frac{1}{\rho_1}\right)+c_2\sigma_F^2 k\right]\cdot \\
&\cdot\max\left\{\int_0^t\left[V(0)+\mathcal{C}_v^0\gamma_m s\right]e^{-\mathcal{C}_v s}+\rho_0\rho_1\gamma_m e^{-2\nu_F s}ds,\right. \\
&\left.\int_0^t\sqrt{\left[V(0)+\mathcal{C}_v^0\gamma_m s\right]e^{-\mathcal{C}_v s}+\rho_0\rho_1\gamma_m e^{-2\nu_F s}}ds\right\}. \qquad (4.4.51)
\end{aligned}
$$

We integrate the first integral in (4.4.51) by parts to get

$$
\int_0^t\left[V(0)+\mathcal{C}_v^0\gamma_m s\right]e^{-\mathcal{C}_v s}+\rho_0\rho_1\gamma_m e^{-2\nu_F s}ds \leq \frac{V(0)}{\mathcal{C}_v}+\gamma_m\left(\frac{\mathcal{C}_v^0}{\mathcal{C}_v^2}+\frac{\rho_0\rho_1}{2\nu_F}\right).
$$

Moreover, applying Hölder inequality to the second integral in (4.4.51) yields

$$
\begin{aligned}
&\int_0^t\sqrt{\left[V(0)+\mathcal{C}_v^0\gamma_m s\right]e^{-\mathcal{C}_v s}+\rho_0\rho_1\gamma_m e^{-2\nu_F s}}ds \leq \\
&\leq \frac{2\sqrt{V(0)}}{\mathcal{C}_v}+\|\sqrt{\mathcal{C}_v^0\gamma_m s}e^{\frac{-\mathcal{C}_v s}{4}}\|_\infty\int_0^t e^{\frac{-\mathcal{C}_v s}{4}}ds+\frac{\sqrt{\rho_0\rho_1\gamma_m}}{\nu_F} \leq \\
&\leq \frac{2\sqrt{V(0)}}{\mathcal{C}_v}+\frac{4\mathcal{C}^*}{\mathcal{C}_v}+\frac{\sqrt{\rho_0\rho_1\gamma_m}}{\nu_F},
\end{aligned}
$$

where $\mathcal{C}^* := \max_{s\in\mathbb{R}} \sqrt{\mathcal{C}_v^0 \gamma_m} s e^{\frac{-\mathcal{C}_v s}{4}}$ and we use the fact that

$$\sqrt{a+b} \leq \sqrt{a} + \sqrt{b},$$

for and $a, b \geq 0$. Altogether, using assumption (4.4.37) we obtain

$$M_\alpha^2(t) \geq M_\alpha^2(0) - \mu \geq \frac{1}{4} M_\alpha^2(0), \tag{4.4.52}$$

with μ defined as in equation (4.4.39). Thus

$$M_\alpha(t) \geq \frac{1}{2} M_\alpha(0). \tag{4.4.53}$$

In addition, since $m(t) \to \tilde{x}$ and $V(t) \to 0$ as $t \to \infty$ it holds,

$$M_\alpha(t) = \int_{\mathbb{R}^d} e^{-\alpha \mathcal{E}(x)} g(x) dx \to e^{-\alpha \mathcal{E}(\tilde{x})}, \tag{4.4.54}$$

as $t \to \infty$ as a consequence of Chebishev inequality (see [48]). Thus

$$0 \geq e^{-\alpha \mathcal{E}(\tilde{x})} \geq \frac{1}{2} M_\alpha(0) \quad \Longleftrightarrow \quad 0 \geq -\alpha \mathcal{E}(\tilde{x}) \geq \log\left(\frac{M_\alpha(0)}{2}\right), \tag{4.4.55}$$

that is

$$0 \leq \mathcal{E}(\tilde{x}) \leq -\frac{1}{\alpha} \log(M_\alpha(0)) + \frac{\log(2)}{\alpha}. \tag{4.4.56}$$

Finally, $0 \leq \mathcal{E}(\tilde{x}) \leq \underline{\mathcal{E}}$ as $\alpha \to \infty$, since the first term tends to $\underline{\mathcal{E}}$ thanks to Laplace principle and $\log(2)/\alpha$ vanishes in the limit. □

Remark 11. We emphasize the following observations:

- In order to satisfy condition (4.4.37), $V(0)$ and $m(0)$ need to be small.
- Note that if we assume to have anisotropic diffusion the convergence is guaranteed independently of the parameters choice and, in particular, of the dimension d. For this reason, all numerical examples of the next section consider the anisotropic noise.

4.5 Numerical methods

In order to approximate the time evolution of the density $f_\lambda(x,t)$ we sample N_s particles $(x_i^0, \lambda_i^0), i = 1, \ldots, N_s$ from the initial distribution. We consider a time interval $[0, T]$ discretized in N_t intervals of length h. The interaction step is solved by means of binary interaction algorithms, see [160, 162] for details.

We denote the approximation of $f_\lambda(x, nh)$ at time t^n by $f_\lambda^n(x)$. For any $\lambda \in \{0, 1\}$ fixed, the next iterate is given by

$$f_\lambda^{n+1}(x) = \left(1 - \frac{h}{\varepsilon}\right) f_\lambda^n(x) + \frac{h}{\varepsilon} \sum_{\lambda_* \in \{0,1\}} Q_\alpha^+(f_\lambda^n, f_{\lambda_*}^n)(x), \tag{4.5.1}$$

where $\varepsilon > 0$ is a frequency parameter and $Q^+(f_\lambda^n, f_{\lambda_*}^n)$ is the gain part of the collision operator defined in (4.3.3). Equation (4.5.1) can be interpreted as follows: with probability $1 - h/\varepsilon$ an individual in position x does not interact with other individuals and with probability h/ε it interacts with another randomly selected individual. In the following we will assume $h = \varepsilon$.

In order to simulate changes of the label λ, we discretize equation (4.2.12). For any fixed $x \in \mathbb{R}^d$, we obtain

$$\begin{aligned} f_0^{n+1}(x) &= (1 - \varepsilon\ \pi_{F\to L})\ f_0^n(x) + \varepsilon\ \pi_{L\to F}\ f_1^n(x), \\ f_1^{n+1}(x) &= (1 - \varepsilon\ \pi_{L\to F})\ f_1^n(x) + \varepsilon\ \pi_{F\to L}\ f_0^n(x), \end{aligned} \tag{4.5.2}$$

where $\pi_{F\to L}(\cdot)$ and $\pi_{L\to F}(\cdot)$ are the transition rates as defined in (4.2.10)-(4.2.11). The details of the numerical scheme are summarized in Algorithm 4.5.1. Here, the parameters δ_{stall} and j_{stall} are used to check if consensus has been reached in the last j_{stall} iterations within a tolerance δ_{stall}. In more detail, we stop the iteration if the distance of the current and previous mean $\hat{x}$ is smaller then the tolerance δ_{stall} for at least j_{stall} iterations. In this case, the evolution is stopped before the total number of iterations has been reached.

Algorithm 4.5.1. [GKBO]

1. Draw $(x_i^0, \lambda_i^0)_{i=1,\dots,N_s}$ from the initial distribution $f_\lambda^0(x)$ and set $n = 0$, $j = 0$.
2. Compute $\hat{x}^0$ as in equation (4.2.7).
3. `while` $n < N_t$ `and` $j < j_{stall}$
   (a) `for` $i = 1$ `to` N
   - Select randomly a leader with position y_k^n, $k \neq i$.
   - Compute the new positions

$$\begin{aligned} x_i^{n+1} &= x_i^n + \nu_F \varepsilon \left(y_k^{n+1} - x_i^n\right) + \sigma_F \sqrt{\varepsilon} D\xi \left(1 - \lambda_i^n\right) + \varepsilon \nu_L (\hat{x}^n - x_i^n)\lambda_i^n, \\ y_k^{n+1} &= y_k^n + \nu_L \varepsilon (\hat{x}^n - y_k^n). \end{aligned} \tag{4.5.3}$$

   - Compute the following probabilities rates

$$p_L = \varepsilon\ \pi_{F\to L}(x_i^{n+1}, \lambda_i^n), \qquad p_F = \varepsilon\ \pi_{L\to F}(x_i^{n+1}, \lambda_i^n).$$

- `if` $\lambda_i^n = 0$,
  with probability p_L agents i becomes a leader: $\lambda_i^{n+1} = 1$.
- `if` $\lambda_i^n = 1$,
  with probability p_F agents i becomes a follower: $\lambda_i^{n+1} = 0$.

`end for`

(b) Compute $\hat{x}^{n+1}$ as in equation (4.2.7).

(c) `if` $\|\hat{x}^{n+1} - \hat{x}^n\|_\infty \leq \delta_{stall}$

$j \leftarrow j + 1$

`end if`

$n \leftarrow n + 1$

`end while`

The above algorithm is inspired from Nanbu's method[157], for larger class of direct simulation Monte-Carlo algorithm for interacting particle dynamics we refer to[11, 162].

4.6 Validation tests

In this section we test the performance of the GKBO algorithm in terms of success rate and number of needed iterations. We consider the translated Rastrigin function with global minimum in $\bar{x} = 1$ for the vast majority of the tests. In the last experiment we compare the results for different benchmark functions (see [130] for a complete list). If not explicitly specified, we run $M = 20$ simulations and, according to [21, 48], we consider a simulation successful if

$$\|\hat{x}(t) - \bar{x}\|_\infty \leq 0.25. \tag{4.6.1}$$

We set $\alpha = 5 \cdot 10^6$ and we adopt the numerical trick described in [104] to allow for arbitrary large values of α. We assume $N = 200$ and that agents are initially uniformly distributed in the hypercube $[-4.12, 0]^d$, which does not contain the global minimum. At time $t = 0$ we suppose all agents are in the followers status and they change their label according to equation (4.2.12). For the GKBO algorithm we set the total percentage of leaders is $\rho_1^\infty = 0.5$, if not specified explicitly. Hence, the transition rates are defined as in equation (4.2.10), with $\pi_{LF} = \pi_{FL} = 0.2$, if the emergence of leaders is random or as defined in equation (4.2.11) if the labels change according to the weighted criterion defined in Section 4.2.2. We will consider also a mixed strategy with $\bar{p} = 0.5$, that is, among the total amount of generated leaders, 50% change their labels according to the weighted strategy and the remaining ones change their labels randomly. We let the dynamics in (4.2.4) to evolve for $N_t = 10000$ iterations with $\varepsilon = 0.1$, where differently specified. We set $j_{stall} = 1000$, $\delta_{stall} = 10^{-4}$. We assume $\nu_F = 1$, $\nu_L = 10$ while the diffusion parameter and the dimension change in the different tests and will be specified later.

4.6.1 Test 1: Comparison of different followers / leaders ratios

Suppose $\sigma_F = 4$, $d = 20$. Table 4.1 reports the mean of the number of iterations and success rate (in parenthesis) for the GKBO algorithm tested on the translated Rastrigin function as the leaders mass at the equilibrium ρ_1^∞, defined as in equation (4.2.16), varies. The success rate and number of iterations for the KBO algorithm are 1 and 10000 respectively. GKBO outperforms KBO in terms of the number of iterations. However, the success rate of GKBO with random leader emergence deteriorates for $\rho_1^\infty = 0.75$.

Table 4.1: Mean of the number of iterations (success rate) for the GKBO algorithm tested on the translated Rastrigin function as the leaders mass at the equilibrium ρ_1^∞ varies.

	GKBO random	GKBO $\bar{p} = 0.5$	GKBO weighted
$\rho_1^\infty = 0.25$	3008 (1)	3588 (1)	6421 (1)
$\rho_1^\infty = 0.5$	2898 (1)	3477 (1)	6612 (1)
$\rho_1^\infty = 0.75$	4252 (0.6)	4566 (1)	7741 (1)

4.6.2 Test 2: GKBO for different choices of $\hat{x}$

We compare the results of the algorithm with $\hat{x}$ as in (4.2.7) and slight modifications given by

$$\hat{x}_F = \frac{\int_{\mathbb{R}^d} x e^{-\alpha \mathcal{E}(x)} f_0(x,t) dx}{\int_{\mathbb{R}^d} e^{-\alpha \mathcal{E}(x)} f_0(x,t) dx}, \qquad \text{or} \qquad \hat{x}_L = \frac{\int_{\mathbb{R}^d} x e^{-\alpha \mathcal{E}(x)} f_1(x,t) dx}{\int_{\mathbb{R}^d} e^{-\alpha \mathcal{E}(x)} f_1(x,t) dx}, \tag{4.6.2}$$

which corresponds to the cases where the weighted mean depends only on the followers or only on the leaders, respectively.

In Figure 4.2 the success rate and number of iterations as σ_F and d varies for $\hat{x}$ (left), $\hat{x}_f$ (middle) and $\hat{x}_L$(right). In the first row, results for the case with random leaders generation are shown, in the second row the mixed leaders generation with $\bar{p} = 0.5$ and in the third row the case with weighted leaders generation. Note that the performance of the random strategy, especially for large values of the dimension d is higher if $\hat{x}_F(t)$ is used for the estimate of the global minimizer. This can be explained by a better exploration phase of the particles during the evolution, whereas the leaders position $\hat{x}_L$ may result in a less accurate estimate, since labels change randomly. The weighted strategy with $\hat{x}_L(t)$ has computational advantages since leaders are chosen to be the agents with optimal position and the computation of the $\hat{x}_L(t)$ requires a lower number of evaluations of the cost function. This may be advantageous in particular if the evaluation of the cost function is numerically expensive.

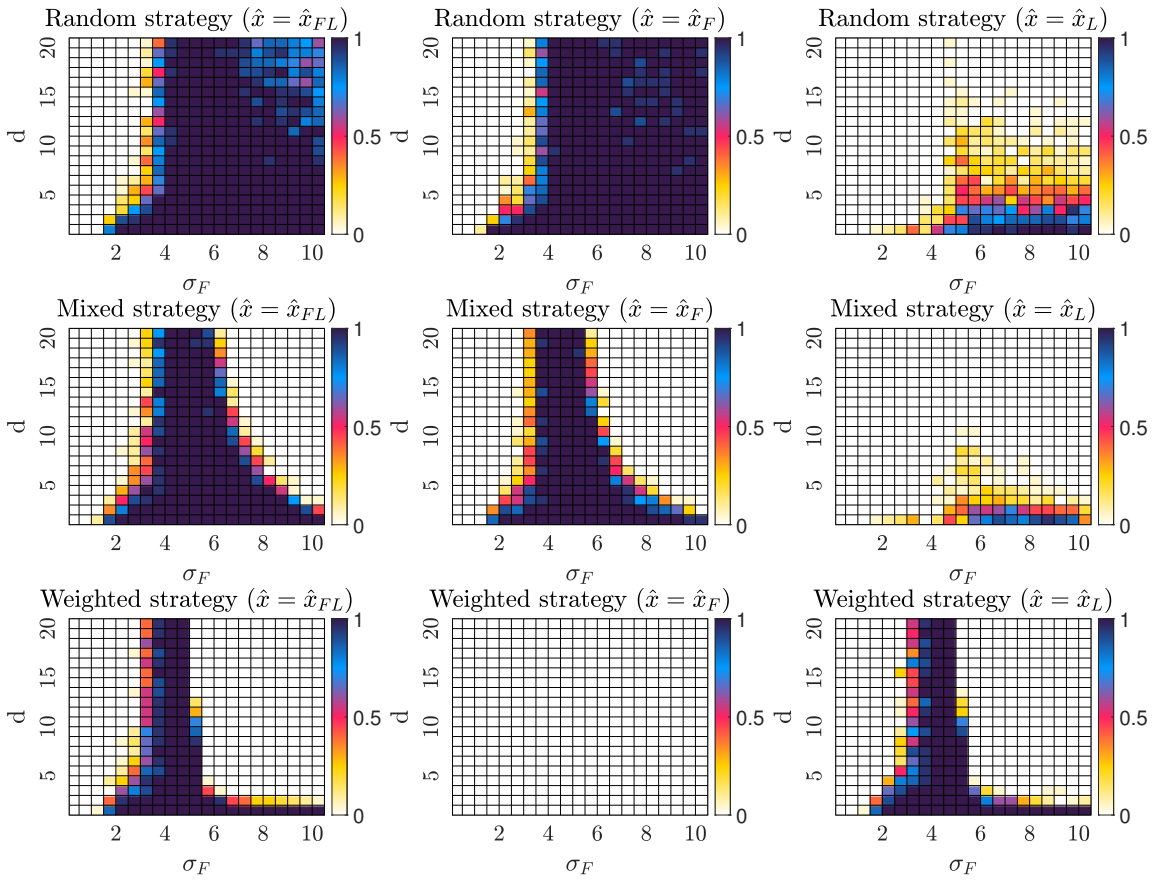

Figure 4.2: Success rates for varying σ_F and d for the translated Rastrigin function with dynamics simulated with the GKBO method for $\hat{x}$ (left), $\hat{x}_F$ (middle), $\hat{x}_L$ (right). The first row is with random leader emergence, second row with mixed strategy $\bar{p} = 0.5$ and third row with weighted leader emergence.

4.6.3 Test 3: Comparison in $d = 20$ dimensions for varying σ_F

We fix $d = 20$ and let σ_F vary from $\sigma_F = 0.1$ to $\sigma_F = 10$ to compare the performance of GKBO (equation (4.2.4)), standard GA (equation (4.1.2)-(4.1.3)), the modified GA (equation (4.1.2)-(4.1.4)) and the KBO (equation (4.1.1)).

In Figure 4.3 the success rates and means of the number of iterations obtained with the different algorithms in the case of the translated Rastrigin function is shown. Here, test GKBO with $\hat{x}$, $\hat{x}_F$ and $\hat{x}_L$ as defined above and study random leader emergence (left), mixed leader emergence with $\bar{p} = 0.5$ (middle), and weighted leader emergence (right).

Altough the success rates of KBO and the variants of GKBO behave similar, the GKBO versions required less iterations. Moreover, we remark that the behavior of the GKBO with weighted leaders generation and with $\hat{x}$ as in equation (4.2.7) and of the KBO is similar, as expected from our analysis.

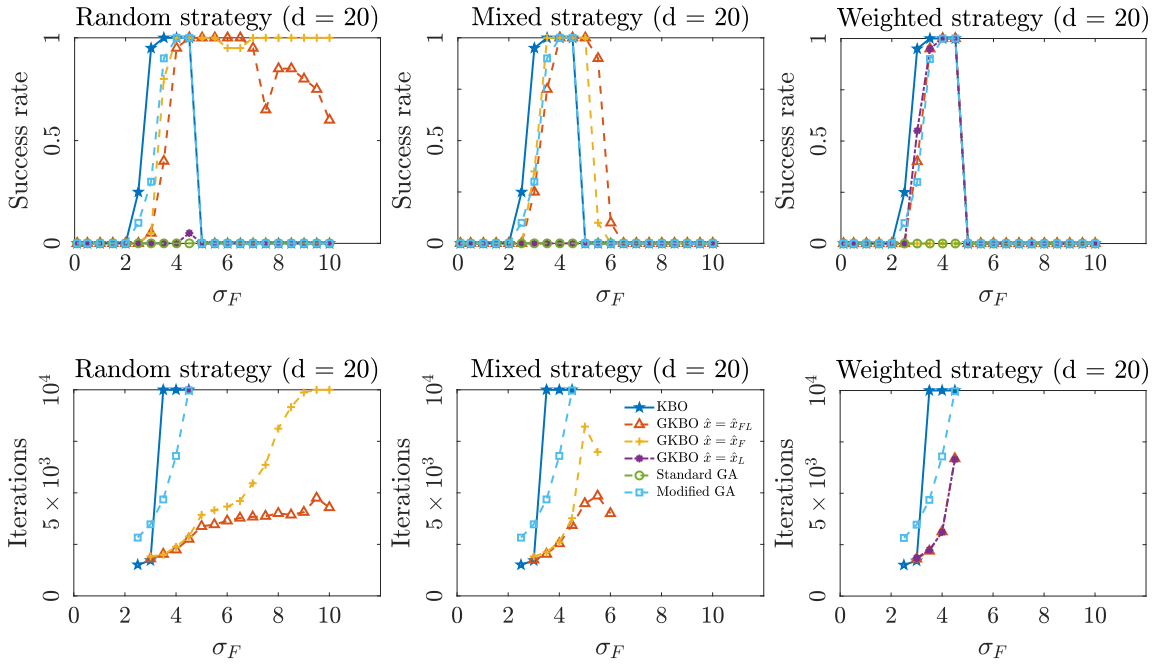

Figure 4.3: Success rates and means of the number of iterations as σ_F varies and $d = 20$ for the translated Rastrigin function obtained with the different algorithms. On the left leaders emerge randomly, in the, we consider mixed leader emergence with $\bar{p} = 0.5$, and on the right, we have weighted leader emergence. The markers denote the value of the success rates and numbers of iterations for different σ_F.

4.6.4 Test 4: Comparison of different leader emergence strategies

Let us fix $d = 20$ and consider the mixed leader emergence strategies as discussed in Remark 8. In Figure 4.4 on the left we see the success rates for different values of σ_F and $\bar{p}$, on the right the number of iterations for different values of $\bar{p}$ and for $\sigma_F = 4, 5$. In Figure 4.5 the success rate and minimum, maximum and mean iterations number for the GKBO method with $\hat{x}$ is shown for $d = 20$ as $\bar{p}$ and σ_F vary.

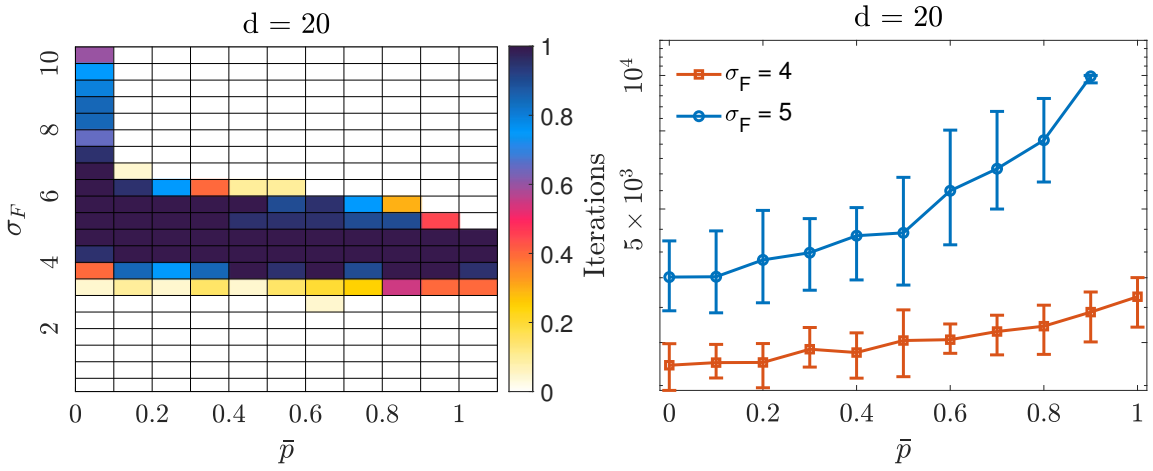

Figure 4.4: Different leader emergence strategies. On the left, success rates for varying σ_F and $\bar{p}$ and $d = 20$. On the right, max, min and mean number of iterations obtained in the different simulations for $d = 20$ and $\sigma_F = 4, 5$ as $\bar{p}$ varies. The markers denote the number of iterations needed for different $\bar{p}$ and tested on the translated Rastrigin function.

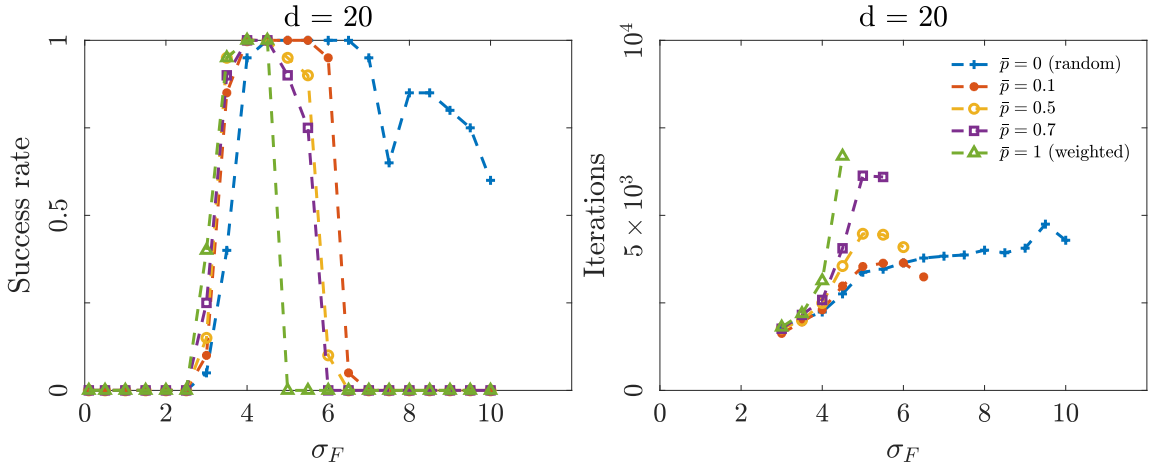

Figure 4.5: Different leader emergence strategies. Success rates and mean number of iterations for $d = 20$ as σ_F varies and for different values of $\bar{p}$, tested on the translated Rastigin function. The markers denote the value of the success rates and number of iterations for different σ_F.

4.6.5 Test 5: Comparison of different methods for varying d

We fix $\sigma_F = 4$ and vary the dimension d from 1 to 20. Figure 4.6 shows the success rates and means of the number of iterations of the different methods in the case of the translated Rastrigin function. GKBO uses $\hat{x}$ as in (4.2.7).

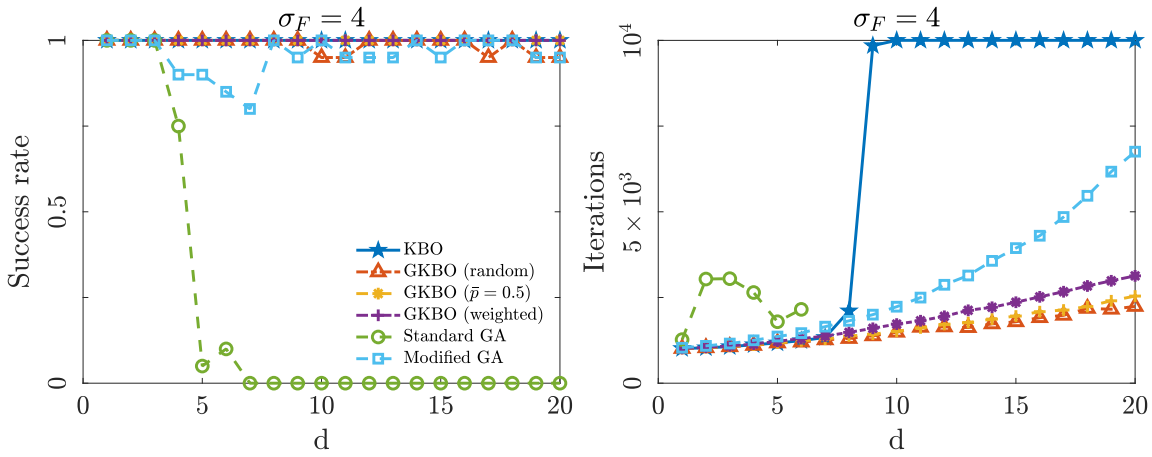

Figure 4.6: Success rates and number of iterations as d varies for the translated Rastrigin function obtained with the different algorithms. The markers denote the value of the success rates and mean number of iterations.

4.6.6 Test 6: Comparison of the accuracy for varying frequency ε

Here we study the influence of the frequency parameter ε by comparing the accuracy of the KBO and GKBO with weighted and random leader emergence. We run the test for $M = 100$ simulations assuming the initial data to be normally distributed in the hypercube $[-4.12, 0]^d$, $d = 20$. The accuracy is computed as

$$\|\hat{x}(t) - \bar{x}\|_\infty, \tag{4.6.3}$$

where $\bar{x}$ is the actual value of the minimum. In Figure 4.7 the accuracy of the KBO algorithm (left) for GKBO algorithm with random leader emergence (middle) and weighted leader emergence (right) with $\varepsilon = 0.01$ (first row) and $\varepsilon = 0.1$ (second row). Note that in both cases, the values of σ_F for which the method converges with the weighted GKBO and the KBO algorithm is almost the same. If $\varepsilon = 0.01$ the accuracy of the weighted GKBO is higher than the one of the KBO. If $\varepsilon = 0.1$ the random strategy performs better than the other methods since the algorithm converges for almost all the values of σ_F considered.

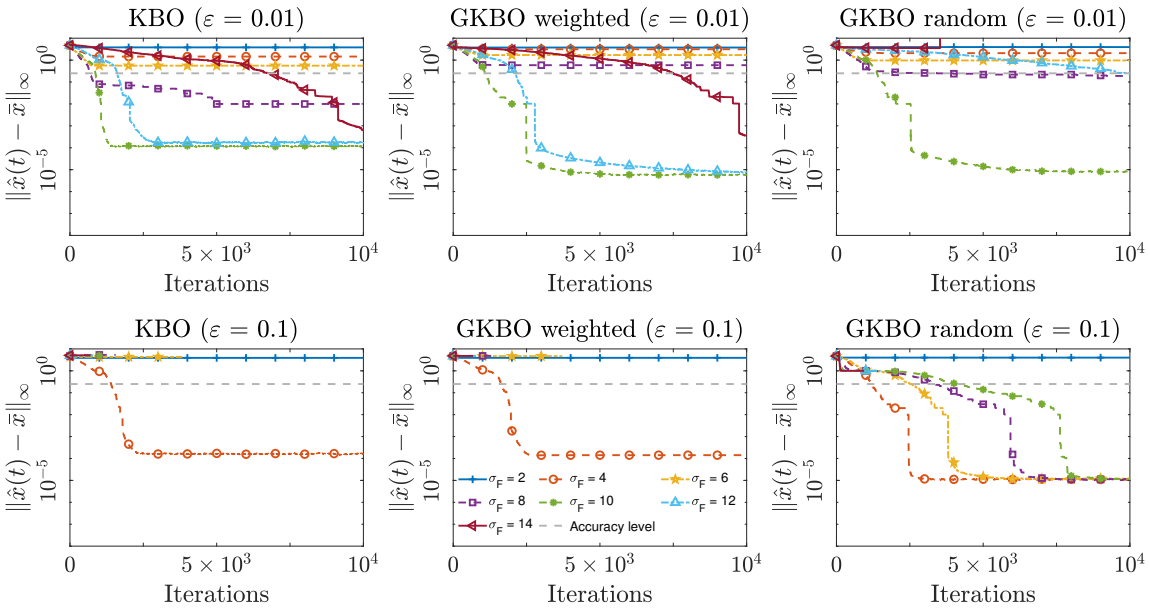

Figure 4.7: Accuracy of the KBO and of the GKBO algorithm as σ_F varies for $d = 20$ and $\varepsilon = 0.01$ (first row), $\varepsilon = 0.1$ (second row) for the translated Rastrigin function. From the left to the right, KBO, GKBO with random leader emergence and GKBO with weighted leader emergence. The markers have been added to distinguish the lines.

Furthermore, if we look at the case $\sigma_F = 4$, all the methods converge but the random strategy reaches higher levels of accuracy.

In Figure 4.8 the results in terms of success rate and number of iterations needed for different values of ε are shown. If $\varepsilon = 0.01$ the success rate are of the GKBO methods is smaller than the one of the KBO but the number of needed iterations is reduced. If $\varepsilon = 0.1$, the success rate area is enlarged for the strategy with random leader emergence. With this test we confirm the results obtained in Figure 4.7. Moreover, the number of iterations is reduced with respect to the KBO and the GKBO method with weighted leader emergence.

4.6.7 Test 7: Comparison of different benchmark functions

In the previous subsection we tested the different algorithms and different parameter sets with the translated Rastrigin function. Now, we choose σ_F such that both the KBO and the GKBO algorithms have success rate equal to one in the previous studies and test different benchmark functions in 20 dimensions. In Figure 4.9 the comparison of KBO and GKBO in terms of success rate and mean number of iterations are shown. GKBO with both variants

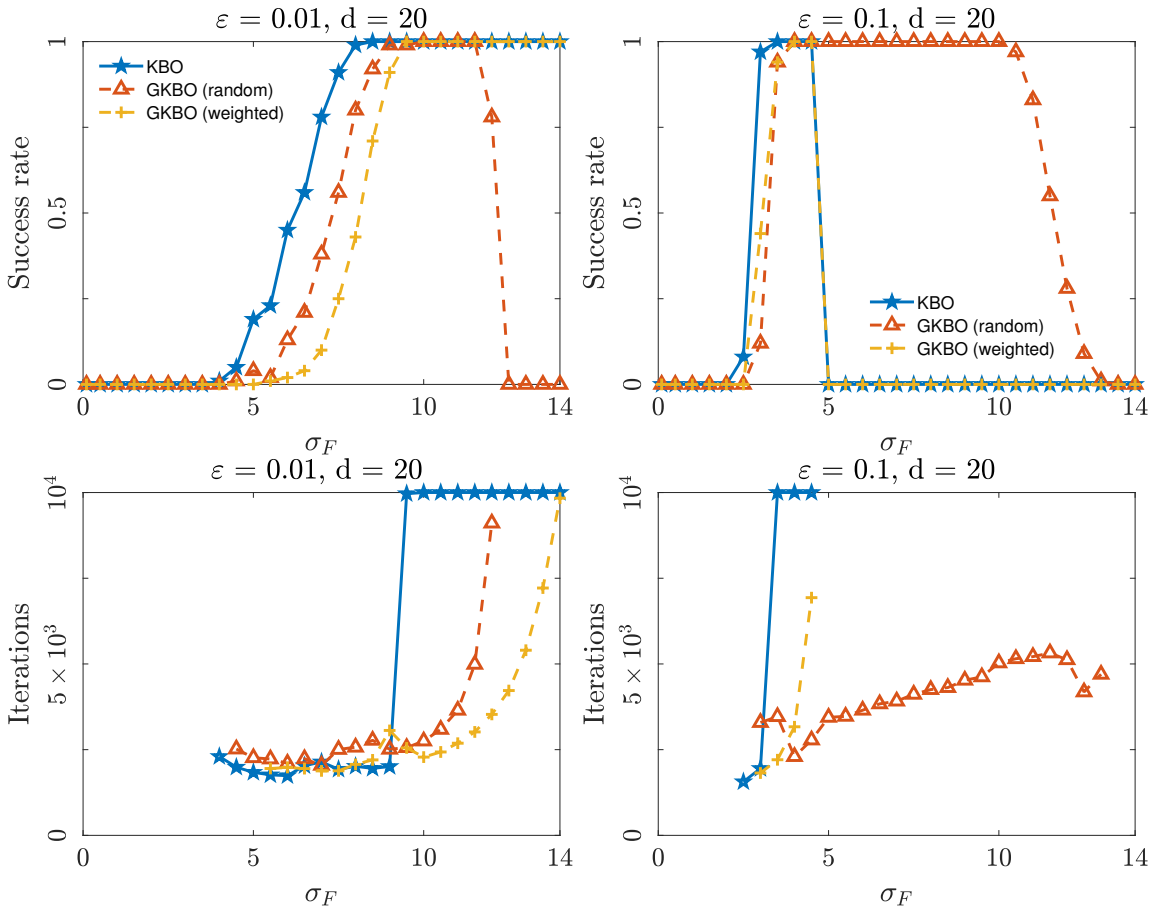

Figure 4.8: Success rates and means of the number of iterations as σ_F varies for $\varepsilon = 0.01$ (first column), $\varepsilon = 0.1$ (second column) for the translated Rastrigin function. The markers have been added to distinguish the lines.

of leader emergence outperforms KBO in terms of the number of iterations.

4.7 Conclusion

We propose a variant of the KBO method for global optimization which is enhanced by a transition process, inspired by genetic dynamics. These lead to a population divided into two species which we call followers and leaders. We adapt the convergence analysis to the new method and show in particular that the population concentrates in the long-time limit arbitrarily close to the global minimizer of the cost function. Numerical results

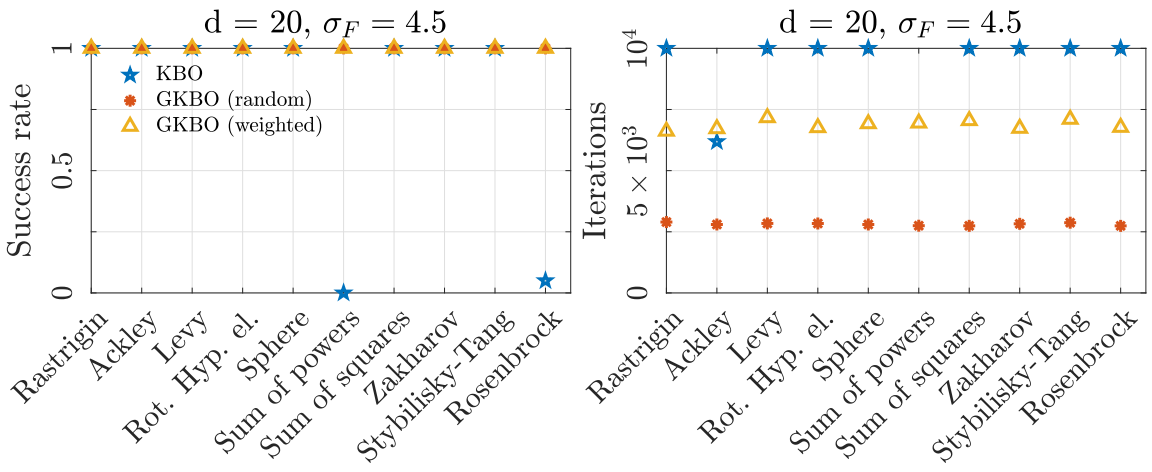

Figure 4.9: Success rate and mean iterations number for the different benchmark function for fixed $d = 20$ and $\sigma_F = 4.5$. Markers denotes the value of the success rate and of the iterations number for the different benchmark functions.

show the feasibility of the approach and the improvement of the proposed generalization in terms of numerical effort. In particular, we measure the numerical effort in terms of the number of iterations which are proportional to the number of function evaluations used in the optimization process. Since GKBO outperforms KBO in this measure, the proposed variant is especially attractive for optimization problems with expensive cost function evaluations. This can be the case for example in engineering applications where each cost function evaluation requires a time-consuming simulation, such as the solution of high-dimensional PDEs with commercial codes.

Chapter 5

Efficient ensemble stochastic algorithms for agent-based models with spatial predator-prey dynamics

Experiments in predator-prey systems show the emergence of long-term cycles. Deterministic model typically fails in capturing these behaviors, which emerge from the microscopic interplay of individual based dynamics and stochastic effects. However, simulating stochastic individual based models can be extremely demanding, especially when the sample size is large. Hence, we propose an alternative simulation approach, whose computation cost is lower than the one of the classic stochastic algorithms. First, we describe the agent-based model with predator-prey dynamics, and its mean-field approximation. Then, we provide a consistency result for the novel stochastic algorithm at the microscopic and mesoscopic scale. Finally, we perform different numerical experiments in order to test the efficiency of the proposed algorithm, focusing also on the analysis of the different nature of oscillations between mean-field and stochastic simulations.

The results presented in this chapter have been published in [5].

5.1 Introduction

The description of how biological populations interact and evolve in both space and time is central to theoretical ecology, [106, 164, 148, 171, 156, 30]. To this aim, several mathematical models have been proposed to approximate the evolution of large ensembles of interacting species, starting from microscopic stochastic agent-based dynamics, passing from mesoscopic equations, up to macroscopic systems, such as reaction-diffusion systems. This hierarchy of descriptions, at different scales, has been used extensively to model interactions among species and their spatial spread beyond theoretical ecology, and it has a profound impact in several fields such as epidemiology, [38, 192], biology, physiology and medicine, [51, 179, 173], chemical reactions[125, 88], and in socio-economic systems opinions, markets, veichular traffic and crowds [162, 68, 2].

In the present manuscript, we will focus on the efficient simulation of predator-prey dynamics in population biology, such as [156, 56, 148]. One striking feature of population

biology is the emergence of temporal cycles of species densities, [156, 159, 25]. Various mechanisms can cause populations to oscillate, the most investigated being the cyclic dynamics that arise from the interactions between predators and preys (see ref. [33] and references cited therein). Traditionally, these interactions have been studied in the context of deterministic Lotka-Volterra-type models, [156, 159, 25]. In their simplest form, these are coupled deterministic equations that include specific birth, death, competition and predation processes, and they can show limit cycles behaviors for appropriate choice of parameters values. Long-term persistence of predator-prey cycles have been recently observed in a series of well controlled microcosm experiments with freshwater organisms, [33]. Planktonic rotifers cultured together with their prey unicellular green algae showed oscillatory behaviors of both prey and predator densities that persisted for up to approximately 50 cycles or ∼300 generations, [33]. Interestingly, dominant dynamics characterized by coherent oscillations were interrupted by shorter episodes of irregular non-coherent oscillations, [33]. The experiments clearly demonstrate that sustained oscillations in population dynamics do arise even in simple well-controlled ecosystems. They strongly indicate that stochastic forces are at work to drive the reversible shift from coherent to non-coherent oscillatory behaviours of observed populations, [33], but their role in driving population dynamics can not be investigated within the context of deterministic predator-prey models.

This novel experimental evidence, therefore, calls for new theoretical explanations, and we note that these might be found within the theoretical framework developed by McKane and Newman (see [149, 150]). They studied individual based predator-prey models described by simple stochastic processes of mortality, reproduction and predation, and showed that large cycles of species densities persisted unless the number of individuals was taken to be strictly infinite, [149, 150]. Biological cycles have been found to arise from a novel resonance effect that include the statistical fluctuations of a given finite-size population, an effect that fade away when different realization of the model are averaged to recover the mean-field behavior described by the classic deterministic predator-prey equations, [149, 150]. Finding the solution of stochastic individual-based models, however, can easily become computationally challenging, above all when the final goal is to investigate the dynamics of large, albeit finite, population sizes over hundreds of species generations. Moreover, depending on the context of application, ad-hoc methods are of paramount importance to capture the essential features of phenomena at various scale, see for example [101, 126, 92, 182].

In the present work, to cope with the increasing complexity of the model we rely on Monte Carlo methods, introducing a novel stochastic algorithm for the simulation of the population dynamics with individual based models. We investigate the efficiency of the new method on spatial predator-prey models and we compare the computational costs, also by exploring in part the space of parameters, to that of different stochastic algorithms such as the classic Monte Carlo, the direct method or the $\tau-$leaping approach considered as benchmarks [146, 111, 44].

The fundamental idea behind our work is that with classic approaches the whole popu-

lation sample must be reconstructed any time interaction between individuals occurs with the updated number of individuals. However, we show that there exists a maximum number of interactions that can take place before updating the sample, and this allows us to reduce the total number of time steps required for the whole simulation. As the consequence, the new algorithm has a computational cost that is up to 2 orders of magnitude less than those of the classic stochastic algorithms considered for a wide range of parameter values. The simulations obtained with the new algorithm converge to the mean-field solutions for increasing sample size N with error $\propto 1/\sqrt{N}$, as also observed for classic algorithms, while individual realizations for finite population sizes oscillates as observed by McKane and Newman.

In Section 5.2 we present an individual-based stochastic model where predators compete with preys and migration of individuals is allowed. We derive the master equation and the mean-field equations which are obtained when the number of individuals is taken to be strictly infinite. In Section 5.3 we present a novel ensemble stochastic algorithm and we prove its consistency with the classical formulation. In Section 5.4 we show different numerical experiments aimed at testing the new algorithm and at comparing it with the classical approaches. We also partly investigate the stochastic persistency of the oscillatory behavior shown by both predator and prey dynamics. Overall our approach can be exploited to efficiently simulate stochastic agent-based models and thus to explore the emergence of long-lasting persistent resonant effects in population dynamics. In 5.A we show how to reduce the model to the homogeneous case, assuming that individuals are not allowed to migrate. In 5.B we briefly report some of the classical exact and approximated stochastic algorithms and we describe them in the homogeneous case.

5.2 Agent-based models with predator-prey interactions

We consider an agent based stochastic model of predation between two species with migration. We will show that for considerably large number of individuals the evolution of this model reduces to a system of partial differential equations.

5.2.1 Spatially heterogeneous predator-prey model

We consider a stochastic process describing the evolution of individuals distributed in the spatial domain $\Omega = [0, L]$ with $L > 0$, divided in M_c spatial cells. We assume that every cell C_ℓ, with $\ell = 1, \dots, M_c$, has N_c components, where each component is realized in one of the following states: predator ($s_{A,\ell}$), prey ($s_{B,\ell}$), or empty ($s_{E,\ell}$). To describe the stochastic dynamics, we consider cell C_ℓ and its nearest cells $C_{\ell\pm1}$, for a fixed time $t > 0$ we define different interaction events.

- *Competition & Birth events.* We sample a component in cell C_ℓ and with probability $q_1 \in [0, 1]$ we let it interacts with another component, randomly chosen among the

other $N_c - 1$ components in cell C_ℓ, according to the following birth and competition rules with rates b^r, p_1^r, p_2^r

$$s_{B,\ell}s_{E,\ell} \xrightarrow{b^r} s_{B,\ell}s_{B,\ell}, \quad s_{A,\ell}s_{B,\ell} \xrightarrow{p_1^r} s_{A,\ell}s_{A,\ell}, \quad s_{A,\ell}s_{B,\ell} \xrightarrow{p_2^r} s_{A,\ell}s_{E\ell}, \tag{5.2.1}$$

where $b^r, p_1^r, p_2^r > 0$ are constant parameters. We assume that if interactions occur among two empty components then no change in the populations sizes is accounted.

- *Migration event.* We sample one component in cell C_ℓ and we assume that with probability $q_2 \in [0, 1]$ such that $q_1 + q_2 \leq 1$ it interacts with another component, randomly chosen among the N_c positions in one of the nearest cells $C_{\ell\pm1}$. Changes in the state happen according to migration rates m_1^r, m_2^r

$$\begin{aligned} &s_{A,\ell}s_{E,\ell\pm1} \xrightarrow{m_1^r} s_{E,\ell}s_{A,\ell\pm1}, \quad s_{B,\ell}s_{E,\ell\pm1}, \xrightarrow{m_2^r} s_{E,\ell}s_{B,\ell\pm1}, \\ &s_{A,\ell\pm1}s_{E,\ell} \xrightarrow{m_1^r} s_{E,\ell\pm1}s_{A,\ell}, \quad s_{B,\ell\pm1}s_{E,\ell} \xrightarrow{m_2^r} s_{E,\ell\pm1}s_{B,\ell}, \end{aligned} \tag{5.2.2}$$

  where $m_1^r, m_2^r > 0$. If interactions occur among predator/prey and empty components or among two empty components then no changes in the populations sizes happen.

- *Death event.* We sample a component in cell C_ℓ, and we assume that with probability $(1 - q_1 - q_2)$ it changes according to death rates d_1^r, d_2^r

$$s_{A,\ell} \xrightarrow{d_1^r} s_{E,\ell}, \quad s_{B,\ell} \xrightarrow{d_2^r} s_{E,\ell}, \tag{5.2.3}$$

  where $d_1^r, d_2^r > 0$. We suppose that if the selected component is empty then no change is accounted.

We assume that at most one birth/competition, one migration and one death event can occur at each time step. In order to introduce a master equation for such process, we consider the vector state $\mathbf{x} = (\mathbf{A}, \hat{\beta}, \mathbf{E}) \in \mathbb{R}^{3M_c}$, where $\mathbf{A} = (A_1, A_2, \ldots, A_{M_c})$, $\hat{\beta} = (B_1, B_2, \ldots, B_{M_c})$, $\mathbf{E} = (E_1, E_2, \ldots, E_{M_c})$ account respectively the number of predators, prey and empty spaces in each cell.

We first define the stoichiometry matrices $\hat{V}$, associated to single competition/birth and death events, and $\hat{V}_M$, associated to migration events

$$\hat{V} = \begin{bmatrix} 0 & 1 & -1 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix}, \quad \hat{V}_M = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}.$$

Hence, the stoichiometry matrix of the full process over the lattice of M_c cells is defined as follows,

$$\tilde{V} = \begin{bmatrix} \hat{V} \otimes I \\ \hat{V}_M \otimes M_- \\ \hat{V}_M \otimes M_+ \end{bmatrix}, \tag{5.2.4}$$

where $\otimes$ denotes the Kronecker product, and I is the identity matrix of size $M_c \times M_c$, and M_-, M_+ are squared are square matrices of size $M_c \times M_c$ defined as

$$M_- = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & -1 & 0 & & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -1 & 0 \\ 0 & \dots & 0 & 1 & -1 \end{bmatrix}, \quad M_+ = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -1 & 0 \\ 0 & \dots & 0 & 0 & 0 \end{bmatrix}.$$

Here, each row of the stoichiometry matrix (5.2.4) represents the changes in the populations sizes due to the occurrence of the events described in (5.2.1)-(5.2.2)-(5.2.3) and each column represents the predators, preys and empty components in each cell. Then, for any $\ell = 1, \dots, M_c$, we write the associated transition rates as follows

$$\begin{aligned}
&\pi_{\mathbf{v}_{\ell_0}}(\mathbf{x}) = 2b^r q_1 \frac{B_\ell}{N_c}\frac{E_\ell}{N_c - 1}, && \pi_{\mathbf{v}_{\ell_1}}(\mathbf{x}) = 2p_1^r q_1 \frac{A_\ell}{N_c}\frac{B_\ell}{N_c - 1}, \\
&\pi_{\mathbf{v}_{\ell_2}}(\mathbf{x}) = 2p_2^r q_1 \frac{A_\ell}{N_c}\frac{B_\ell}{N_c - 1}, && \pi_{\mathbf{v}_{\ell_3}}(\mathbf{x}) = d_1^r(1 - q_1 - q_2)\frac{A_\ell}{N_c}, \\
&\pi_{\mathbf{v}_{\ell_4}}(\mathbf{x}) = d_2^r(1 - q_1 - q_2)\frac{B_\ell}{N_c}, && \pi_{\mathbf{v}_{\ell_5}}(\mathbf{x}) = m_1^r q_2 \frac{A_\ell}{N_c}\frac{E_{\ell-1}}{N_c}, \\
&\pi_{\mathbf{v}_{\ell_6}}(\mathbf{x}) = m_1^r q_2 \frac{A_{\ell-1}}{N_c}\frac{E_\ell}{N_c}, && \pi_{\mathbf{v}_{\ell_7}}(\mathbf{x}) = m_2^r q_2 \frac{B_\ell}{N_c}\frac{E_{\ell-1}}{N_c}, \\
&\pi_{\mathbf{v}_{\ell_8}}(\mathbf{x}) = m_2^r q_2 \frac{B_{\ell-1}}{N_c}\frac{E_\ell}{N_c}, && \pi_{\mathbf{v}_{\ell_9}}(\mathbf{x}) = m_1^r q_2 \frac{A_\ell}{N_c}\frac{E_{\ell+1}}{N_c}, \\
&\pi_{\mathbf{v}_{\ell_{10}}}(\mathbf{x}) = m_1^r q_2 \frac{A_{\ell+1}}{N_c}\frac{E_\ell}{N_c}, && \pi_{\mathbf{v}_{\ell_{11}}}(\mathbf{x}) = m_2^r q_2 \frac{B_\ell}{N_c}\frac{E_{\ell+1}}{N_c}, \\
&\pi_{\mathbf{v}_{\ell_{12}}}(\mathbf{x}) = m_2^r q_2 \frac{B_{\ell+1}}{N_c}\frac{E_\ell}{N_c}, &&
\end{aligned} \tag{5.2.5}$$

where we define the operators $\pi_{\mathbf{v}_{\ell_j}}(\mathbf{x}) = \pi(\mathbf{x}+\mathbf{v}_{\ell_j}|\mathbf{x})$ for any $j = 0, \dots, M$. Here, $M+1 = 13$ is the total number of events described in (5.2.1)-(5.2.2)-(5.2.3) and $\mathbf{v}_{\ell_j}$ is the ℓ_j-th row of the stoichiometry matrix $\tilde{V}$ defined in (5.2.4), $\ell_j = \ell + jM_c$ for $j = 0, \dots, M$. We assume that migrations events are not allowed at the boundaries. Hence we introduce $A_0, E_0, B_0, A_{M_c+1}, E_{M_c+1}, B_{M_c+1}$ to be equal to zero, in order to properly define the transition rates associated to the boundary cells. The density $P(\mathbf{x}, t)$ describing the probability of the state $\mathbf{x}$ evolves according to the master equation as follows

$$\frac{dP(\mathbf{x},t)}{dt} = \sum_{\ell_j \in \mathcal{J}} \Big[\pi(\mathbf{x}|\mathbf{x} - \mathbf{v}_{\ell_j})P(\mathbf{x} - \mathbf{v}_{\ell_j}, t) - \pi(\mathbf{x} + \mathbf{v}_{\ell_j}|\mathbf{x})P(\mathbf{x}, t) \Big], \tag{5.2.6}$$

where

$$\mathcal{J} = \{\ell + jM_c | j = 0, \dots, M \text{ and } \ell = 1, \dots, M_c\}. \tag{5.2.7}$$

Remark 12.

- We assume that each spatial cell contains at most N_c components that can be either preys or predators or empty spaces. Hence, even if predators and preys are restricted to move only to empty neighbor sites, the model can include regions occupied both by predators and preys. In the case of a lattice consisting of a single spatial cell, we refer to spatial homogeneous model, where migration events are neglected and maximal density per cell is $N_c = N$. We report in 5.A the corresponding agent-based dynamics, and the associated master equation (5.2.6).

- We observe that the vector state $\mathbf{x}$ can be dimensionaly reduced if we assume to remove the empty components, considering the relation $\mathbf{E} = N_c - \mathbf{A} - \hat{\beta}$. However, we prefer to give a more general description, to allow for further generalization, such as models that may include empty sites with different nature.

5.2.2 Mean-field approximation

In order to recover the mean-field behavior of the stochastic process, we introduce the empirical densities $f_\ell^{N_c}(t), g_\ell^{N_c}(t)$, for $\ell = 1, \dots, M_c$, as the averaged quantities

$$\begin{aligned} f_\ell^{N_c}(t) &= \frac{\langle A_\ell \rangle}{N_c} = \frac{1}{N_c} \sum_{A_\ell=0}^{N_c} \sum_{B_\ell=0}^{N_c} \sum_{E_\ell=0}^{N_c} A_\ell P(\mathbf{x}, t), \\ g_\ell^{N_c}(t) &= \frac{\langle B_\ell \rangle}{N_c} = \frac{1}{N_c} \sum_{A_\ell=0}^{N_c} \sum_{B_\ell=0}^{N_c} \sum_{E_\ell=0}^{N_c} B_\ell P(\mathbf{x}, t), \end{aligned} \tag{5.2.8}$$

where $\langle \cdot \rangle$ denotes the expected value, and the empirical density for the empty component can be recovered in each cell computing $1 - f_\ell^{N_c}(t) - g_\ell^{N_c}(t)$.

Hence, multiplying the master equation by A_ℓ and by B_ℓ, and summing over all the values of A_ℓ, B_ℓ and E_ℓ for any $\ell = 1 \dots, M_c$ leads to the following Proposition.

Proposition 4. *By standard assumptions of the mean-field limit we assume that for any* $\ell = 1 \dots, M_c$, $\langle A_\ell B_\ell \rangle = \langle A_\ell \rangle \langle B_\ell \rangle$, $\langle A_\ell^2 \rangle = \langle A_\ell \rangle^2$, $\langle B_\ell^2 \rangle = \langle B_\ell \rangle^2$, $\langle A_{\ell \pm 1} B_\ell \rangle = \langle A_{\ell \pm 1} \rangle \langle B_\ell \rangle$ *and* $\langle A_\ell B_{\ell \pm 1} \rangle = \langle A_\ell \rangle \langle B_{\ell \pm 1} \rangle$ *then the time evolution of the empirical population densities*

$f_\ell^{N_c}$, $g_\ell^{N_c}$ is given by

$$\begin{aligned}\frac{df_\ell^{N_c}}{d\tau} =& 2\tilde{p_1^r}\frac{\langle A_\ell\rangle}{N_c}\frac{\langle B_\ell\rangle}{N_c-1} - \tilde{d_1^r}\frac{\langle A_\ell\rangle}{N_c} \\ &+ \tilde{m}_1^r\left(\frac{\Delta_\epsilon\langle A_\ell\rangle}{N_c} + \frac{\langle A_\ell\rangle}{N_c}\frac{\Delta_\epsilon\langle B_\ell\rangle}{N_c} - \frac{\langle B_\ell\rangle}{N_c}\frac{\Delta_\epsilon\langle A_\ell\rangle}{N_c}\right), \\ \frac{dg_\ell^{N_c}}{d\tau} =& r\frac{\langle B_\ell\rangle}{N_c}\left(1-\frac{\langle B_\ell\rangle}{qN_c}\right) - \alpha\frac{\langle A_\ell\rangle}{N_c}\frac{\langle B_\ell\rangle}{N_c} \\ &+ \tilde{m}_2^r\left(\frac{\Delta_\epsilon\langle B_\ell\rangle}{N_c} + \frac{\langle B_\ell\rangle}{N_c}\frac{\Delta_\epsilon\langle A_\ell\rangle}{N_c} - \frac{\langle A_\ell\rangle}{N_c}\frac{\Delta_\epsilon\langle B_\ell\rangle}{N_c}\right),\end{aligned} \tag{5.2.9}$$

where

$$\Delta_\epsilon h_\ell = \sum_{s\in\{\ell-1,\ell+1\}} \frac{h_s - h_\ell}{\epsilon^2},$$

for any function h is the discrete Laplace operator, ϵ is the lattice spacing, $\tau = t/N_c$, the parameters are defined according to the following scaling

$$\begin{aligned}&\tilde{b}^r = b^r q_1 \quad \tilde{p_1}^r = p_1^r q_1 \quad \tilde{p_2}^r = p_2^r q_1 \quad \tilde{d_1^r} = (1-q_1-q_2)d_1^r \\ &\tilde{d_2}^r = (1-q_1-q_2)d_2^r \quad \tilde{m}_1^r = q_2\epsilon^{-2}m_1^r \quad \tilde{m}_2^r = q_2\epsilon^{-2}m_2^r,\end{aligned} \tag{5.2.10}$$

and

$$r = 2\tilde{b}^r - \tilde{d_2}^r, \qquad q = 1 - \frac{\tilde{d_2}^r}{2\tilde{b}^r}, \qquad \alpha = 2(\tilde{p_1^r} + \tilde{p_2^r} + \tilde{b}^r).$$

Finally the mean-field behavior is recovered for $N_c \gg 1$ as the result of the following Theorem.

Theorem 5.2.1. *Consider the discrete mean-field model (5.2.9) for N_c individuals. Then taking the limit for $N_c \to \infty$, $\epsilon \to 0$, the mean-field equations for the densities $f(x,t)$ and $g(x,t)$,*

$$\begin{aligned}\partial_\tau f &= 2\tilde{p}_1 fg - \tilde{d}_1 f + \tilde{m}_1(f\Delta g + (1-g)\Delta f), \\ \partial_\tau g &= rg\left(1-\frac{g}{q}\right) - \alpha fg + \tilde{m}_2(g\Delta f + (1-f)\Delta g),\end{aligned} \tag{5.2.11}$$

where each entry of the vectors $f(\mathbf{x},t)$ and $g(\mathbf{x},t)$ represents the predators and preys densities in each cell C_ℓ, $\ell = 1,\dots,M_c$ and

$$\Delta h(x,t) = \lim_{\epsilon\to 0}\Delta_\epsilon h$$

for any function h.

We refer to [149, 150, 188], for a detailed proof of the mean-field limit.

Remark 13. Note that under condition $(1-q_1-q_2)d_2^r < (2q_1b^r)$ model (5.2.11) corresponds to the well known Lotka-Volterra equations with logistic growth term and diffusion, [156].

5.3 Efficient ensemble stochastic algorithms

The dynamics described by model (5.2.11) can be properly simulated with stochastic algorithms. Exact stochastic algorithms predict which is the next firing event and at which time it will fire, as for example in [111]. However, a direct implementation is often prohibitively expensive since the final goal is the simulation of a large stochastic process. On the other hand, classic approximated algorithms, such as the Monte Carlo algorithm and the τ-leaping method, can speed up the simulations but their efficiency and accuracy are strongly related to the choice of the time step that can vary in time or be inversely proportional to the sample size. The main idea of the procedure is to fix a constant time step, and to allow multiple events to happen at the same time. Hence, we describe the associated stochastic process for the spatial heterogeneous predator-prey dynamics and show its consistency with the previous description. Second we describe the ensemble stochastic algorithm in the spatial homogeneous and heterogeneous setting.

5.3.1 Predator-prey model with ensemble interactions

At each time step $t \geq 0$ consider two fractions $q_1, q_2 \in [0,1]$ such that $q_1 + q_2 \leq 1$. Assume that the following events can occur.

- *Competition & Birth.* Sample $\lfloor q_1 N_c \rfloor$ components in each cell C_ℓ, $\ell = 1, \dots, M_c$ assuming that each one interacts with another component, randomly chosen among the N_c positions in cell C_ℓ without repetition, according to the rules described in (5.2.1).

- *Migration.* Sample other $\lfloor q_2 N_c \rfloor$ components in any cell C_ℓ, different from the ones previously selected, and let each one to interact with another state, randomly chosen in cell $C_{\ell \pm 1}$ without repetition, according to the migration rules defined in (5.2.2).

- *Death.* The remaining $N_c - \lfloor q_1 N_c \rfloor - \lfloor q_2 N_c \rfloor$ components in any cell C_ℓ change according to death rules defined in equation (5.2.3).

Hence, in any infinitesimal time interval $[t, t + d\tau_{N_c}]$, where $d\tau_{N_c} := dt N_c$, a transition can occur from the state with $\mathbf{x} = (\mathbf{A}, \hat{\beta}, \mathbf{E})$ individuals to the states with $\mathbf{x} + k\mathbf{v}_{\ell_j}$ individuals for any $k = 1, \dots, N_c$, where $\mathbf{v}_{\ell_j}$ is the ℓ_j-th row of the stoichiometry matrix $\tilde{V}$ defined in (5.2.4) for any $\ell = 1, \dots, M_c$, $j = 0, \dots, M$. For $k = 1, \dots N$, the transition rates write as follows

$$\pi(\mathbf{x} + k\mathbf{v}_{\ell_j} | \mathbf{x}) = \prod_{i=1}^{k} \pi^i_{\mathbf{v}_{\ell_j}}(\mathbf{x}) \tag{5.3.1}$$

where

$$
\begin{aligned}
\pi^i_{\mathbf{v}_{\ell_0}}(\mathbf{x}) &= 2b^r q_1 \frac{\tilde{B}^i_\ell}{\tilde{N}^i_c} \frac{\tilde{E}^i_\ell}{\tilde{N}^i_c}, & \pi^i_{\mathbf{v}_{\ell_1}}(\mathbf{x}) &= 2p^r_1 q_1 \frac{\tilde{A}^i_\ell}{\tilde{N}^i_c} \frac{\tilde{B}^i_\ell}{\tilde{N}^i_c},\\
\pi^i_{\mathbf{v}_{\ell_2}}(\mathbf{x}) &= 2p^r_2 q_1 \frac{\tilde{A}^i_\ell}{\tilde{N}^i_c} \frac{\tilde{B}^i_\ell}{\tilde{N}^i_c}, & \pi^i_{\mathbf{v}_{\ell_3}}(\mathbf{x}) &= d^r_1 (1-q_1-q_2) \frac{\tilde{A}^i_\ell}{\tilde{N}^i_c},\\
\pi^i_{\mathbf{v}_{\ell_4}}(\mathbf{x}) &= d^r_2 (1-q_1-q_2) \frac{\tilde{B}^i_\ell}{\tilde{N}^i_c}, & \pi^i_{\mathbf{v}_{\ell_5}}(\mathbf{x}) &= m^r_1 q_2 \frac{\tilde{A}^i_\ell}{\tilde{N}^i_c} \frac{\tilde{E}^i_{\ell-1}}{\tilde{N}^i_c},\\
\pi^i_{\mathbf{v}_{\ell_6}}(\mathbf{x}) &= m^r_1 q_2 \frac{\tilde{A}^i_{\ell-1}}{\tilde{N}^i_c} \frac{\tilde{E}^i_\ell}{\tilde{N}^i_c}, & \pi^i_{\mathbf{v}_{\ell_7}}(\mathbf{x}) &= m^r_2 q_2 \frac{\tilde{B}^i_\ell}{\tilde{N}^i_c} \frac{\tilde{E}^i_{\ell-1}}{\tilde{N}^i_c},\\
\pi^i_{\mathbf{v}_{\ell_8}}(\mathbf{x}) &= m^r_2 q_2 \frac{\tilde{B}^i_{\ell-1}}{\tilde{N}^i_c} \frac{\tilde{E}^i_\ell}{\tilde{N}^i_c}, & \pi^i_{\mathbf{v}_{\ell_9}}(\mathbf{x}) &= m^r_1 q_2 \frac{\tilde{A}^i_\ell}{\tilde{N}^i_c} \frac{\tilde{E}^i_{\ell+1}}{\tilde{N}^i_c},\\
\pi^i_{\mathbf{v}_{\ell_{10}}}(\mathbf{x}) &= m^r_1 q_2 \frac{\tilde{A}^i_{\ell+1}}{\tilde{N}^i_c} \frac{\tilde{E}^i_\ell}{\tilde{N}^i_c}, & \pi^i_{\mathbf{v}_{\ell_{11}}}(\mathbf{x}) &= m^r_2 q_2 \frac{\tilde{B}^i_\ell}{\tilde{N}^i_c} \frac{\tilde{E}^i_{\ell+1}}{\tilde{N}^i_c},\\
\pi^i_{\mathbf{v}_{\ell_{12}}}(\mathbf{x}) &= m^r_2 q_2 \frac{\tilde{B}^i_{\ell+1}}{\tilde{N}^i_c} \frac{\tilde{E}^i_\ell}{\tilde{N}^i_c}.
\end{aligned}
\tag{5.3.2}
$$

Here we define the operators $\pi^i_{\mathbf{v}_{\ell_j}}(\mathbf{x}) = \pi(\mathbf{x}+i\mathbf{v}_{\ell_j}|\mathbf{x}+(i-1)\mathbf{v}_{\ell_j})$ for any $i = 1, \dots, N_c$, we use the following notation $\tilde{Y}^i = Y - i + 1$ for any state Y in the set $\{A_\ell, B_\ell, E_\ell, A_{\ell\pm1}, B_{\ell\pm1}, E_{\ell\pm1}, N_c\}$ and since migration outside the boundaries is not allowed, we introduce $A_0, E_0, B_0, A_{M_c+1}, E_{M_c+1}, B_{M_c+1}$ to be equal to zero. The master equation associated to this process is derived in the following Proposition.

Proposition 5 (Consistency)**.** *The master equation can be written as*

$$
\frac{dP(\mathbf{x},s)}{ds} = \sum_{\ell_j \in \mathcal{J}} \Big[\pi(\mathbf{x}|\mathbf{x}-\mathbf{v}_{\ell_j}) P(\mathbf{x}-\mathbf{v}_{\ell_j}, s) - \pi(\mathbf{x}+\mathbf{v}_{\ell_j}|\mathbf{x}) P(\mathbf{x}, s) \Big], \tag{5.3.3}
$$

where $s = tN_c$, the set $\mathcal{J}$ is defined as in (5.2.7) and $|\mathcal{J}| = M_c(M+1)$ is the number of events that can occur.

Proof. The probability to be in the state with $\mathbf{x}$ individuals at time $t + \tau_{N_c}$ is given by two contributions:

- the probability of staying in the state $\mathbf{x}$ at time $t + \tau_{N_c}$ that is

$$
1 - \sum_{\ell_j \in \mathcal{J}} \sum_{k=1}^{N_c} \pi(\mathbf{x} + k\mathbf{v}_{\ell_j}|\mathbf{x}) d\tau^k_{N_c}; \tag{5.3.4}
$$

- the probability of moving from the state with $\mathbf{x}$ individuals to the state with $\mathbf{x}+k\mathbf{v}_{\ell_j}$ individuals in the time interval $[t, t+\tau_{N_c}]$, for any $k = 1, \dots, N_c$ and $\ell_j \in \mathcal{J}$ that is

$$\sum_{\ell_j \in \mathcal{J}} \sum_{k=1}^{N_c} \pi(\mathbf{x}|\mathbf{x} - k\mathbf{v}_{\ell_j}) d\tau_{N_c}^k. \tag{5.3.5}$$

Indeed, recall that for any $i = 1, \dots, N_c$ a transition from the state with $\mathbf{x}+i\mathbf{v}_{\ell_j}$ individuals to the state with $\mathbf{x}+(i-1)\mathbf{v}_{\ell_j}$ individuals in the infinitesimal time interval $[t, t+\tau_{N_c}]$ occurs with probability $\pi^i_{\ell_j}(\mathbf{x})d\tau_{N_c}$ and hence the probability of moving from the state with $\mathbf{x}$ to the state with $\mathbf{x} + k\mathbf{v}_{\ell_j}$ individuals, for any $k = 1, \dots, N_c$, is

$$\prod_{i=1}^{k} \left\{ \pi^i_{\mathbf{v}_{\ell_j}}(\mathbf{x}) d\tau_{N_c} \right\} = \pi(\mathbf{x} + k\mathbf{v}_{\ell_j}|\mathbf{x}) d\tau_{N_c}^k,$$

that is exactly the expression that appears in equation (5.3.4). Similarly we can derive equation (5.3.5). Hence,

$$\begin{aligned} P(\mathbf{x}, t + d\tau_{N_c}) = &\sum_{\ell_j \in \mathcal{J}} \sum_{k=1}^{N_c} \left[P(\mathbf{x} - k\mathbf{v}_{\ell_j}, t) \pi(\mathbf{x}|\mathbf{x} - k\mathbf{v}_{\ell_j}) d\tau_{N_c}^k \right] \\ &+ P(\mathbf{x}, t) \left(1 - \sum_{\ell_j \in \mathcal{J}} \sum_{k=1}^{N_c} \left[\pi(\mathbf{x} + k\mathbf{v}_j|\mathbf{x}) d\tau_{N_c}^k \right] \right). \end{aligned} \tag{5.3.6}$$

Rewrite explicitly the term for $k = 1$ in equation (5.3.6) to have

$$\begin{aligned} P(\mathbf{x}, t + d\tau_{N_c}) - P(\mathbf{x}, t) = d\tau_{N_c} \sum_{\ell_j \in \mathcal{J}} \left[P(\mathbf{x} - \mathbf{v}_j, t) \pi(\mathbf{x}|\mathbf{x} - \mathbf{v}_j) - P(\mathbf{x}, t) \pi(\mathbf{x} + \mathbf{v}_j|\mathbf{x}) \right] \\ + d\tau_{N_c}^k \sum_{\ell_j \in \mathcal{J}} \sum_{k=2}^{N_c} \left[P(\mathbf{x} - k\mathbf{v}_j, t) \pi(\mathbf{x}|\mathbf{x} - k\mathbf{v}_j) - P(\mathbf{x}, t) \pi(\mathbf{x} + k\mathbf{v}_j|\mathbf{x}) \right]. \end{aligned} \tag{5.3.7}$$

Dividing both sides of equation (5.3.7) by $d\tau_{N_c}$ and letting $d\tau_{N_c} \to 0$ we obtain the consistency with the master equation (5.2.6), where the time variable is scaled by a factor N_c, i.e. $s = tN_c$. □

Remark 14. We observe that thanks to the consistency result of Proposition 5, the corresponding mean-field approximation is equivalent to (5.2.11). According to (5.2.11) the time scale of the mean-field dynamics is such that

$$\tau = s/N_c = N_c t/N_c = t,$$

then the time scales of the ensemble agent-based dynamics and of the mean-field dynamics are equivalent.

5.3.2 Efficient Monte-Carlo methods

The idea of the efficient Monte Carlo algorithm is to allow multiple events to occur at the same time step that is supposed to be fixed and constant for any choice of N_c. Therefore, as we will see in the numerical experiments, its accuracy is comparable with the one of classic approximated algorithms but its efficiency is improved. In the following we will focus first on the spatial homogeneous case, where a single cell is accounted ($M_c = 1$) and $N_c = N$, second we will consider the full dynamics with spatial interaction. We refer to 5.A for further details on the homogeneous case.

Spatially homogeneous case Divide a priori the time interval considering a constant time step τ that is independent of the sample size. Introduce a parameter $\mu \in [0, 1]$ and at each time t assume that simultaneously $\lfloor \mu N \rfloor$ individuals interact two by two. If the two individuals are

- a prey and an empty space: with probability $b^r\tau$ a new prey born and occupies the empty space;
- a predator and a prey: with probability $p_1^r\tau$ then a new predator born and with probability $p_2^r\tau$ the prey dies and a new empty space is added to the system.

Assume that the remaining $N - \lfloor \mu N \rfloor$ individuals are subjected to death events that happen with probability $d_1^r\tau$ in the predators population and with probability $d_2^r\tau$ in the preys one. Algorithm 5.3.1 defines the details of the efficient Monte Carlo algorithm.

Algorithm 5.3.1 (Efficient ensemble stochastic algorithm - homogeneous)**.**

1. Define the sample, the initial time $t = 0$, the final time T, the time step τ and a parameter $\mu \in [0, 1]$.
2. `while` $t < T$
   (a) Birth and competition events happen between $\lfloor \mu N \rfloor$ individuals selected two by two
   $$s_B s_E \xrightarrow{b} s_B s_B, \quad s_A s_B \xrightarrow{p_1} s_A s_A, \quad s_A s_B \xrightarrow{p_2} s_A s_E,$$
   according to probability $b = b^r\tau$, $p_1^r\tau$ and $p_2^r\tau$, respectively.
   (b) The remaining $N - \lfloor \mu N \rfloor$ individuals are subjected to death events
   $$s_A \xrightarrow{d_1} s_E, \quad s_B \xrightarrow{d_2} s_E,$$
   according to probability $d_1 = d_1^r\tau$ and $d_2 = d_2^r\tau$, respectively.
   (c) Update the sample.
   (d) Set $t \leftarrow t + \tau$.

   `repeat`

Spatially heterogeneous case We extend Algorithm 5.3.1 to the spatially heterogeneous case. Hence, the sample is divided in cells and populations are subjected also to migration events. Migrations can occur either between cell C_ℓ and cell $C_{\ell+1}$ or between cell C_ℓ and cell $C_{\ell-1}$ for any $\ell = 1, \dots, M_c$. The spatially heterogeneous Algorithm 5.3.1 reads as follows

Algorithm 5.3.2 (Efficient ensemble stochastic algorithm)**.**

1. Define the sample, the initial time $t = 0$, the final time T, the time step τ and two parameters $q_1, q_2 \in [0, 1]$ such that $q_1 + q_2 \leq 1$.

2. `while` $t < T$

   (a) In each cell C_ℓ, $\lfloor q_1 N_c \rfloor$ individuals interact two by two and are subjected to birth and competition events (5.2.1) that occur with probabilities $b = b^r\tau$, $p_1 = p_1^r\tau$ and $p_2 = p_2^r\tau$;

   (b) In each cell C_ℓ, $\lfloor q_2 N_c/2 \rfloor$ individuals interact with $\lfloor q_2 N_c/2 \rfloor$ individuals sampled in cell $C_{\ell+1}$ and other $\lfloor q_2 N_c/2 \rfloor$ interact with $\lfloor q_2 N_c/2 \rfloor$ individuals sampled in cell $C_{\ell-1}$ according to (5.2.2), and migrate with probability $m_1 = m_1^r\tau$ for predators and $m_2 = m_2^r\tau$ for preys;

   (c) In each cell C_ℓ, the remaining $N_c - \lfloor q_1 N_c \rfloor - \lfloor q_2 N_c \rfloor$ individuals are subjected to death events (5.2.3) according to probabilities $d_1 = d_1^r\tau$ and $d_2 = d_2^r\tau$;

   (d) Update the sample;

   (e) Set $t \leftarrow t + \tau$.

5.4 Numerical experiments

5.4.1 Test 1: Validation

In this section we present a comparison between the numerical solutions of the mean-field equations and the stochastic simulations. In both the homogeneous and heterogeneous case, stochastic simulations have been performed with the efficient version of the Monte Carlo algorithm presented in Section 5.3. In the homogeneous case, the numerical solutions of the mean-field equations are computed using the Matlab function `ode45`, [147], while in the heterogeneous case with a combination of finite difference methods and numerical methods for ODEs assuming periodic boundary conditions, [128]. The parameters choice for all the tests in Section 5.4.1 is specified in Table 5.1. The sample size N and the total number of individuals N_c in any cell change in any test and will be defined later.
Figure 5.1 shows a comparison between stochastic and mean-field solutions in the homogeneous case assuming the sample size to be $N = 1000$ and the initial predators and preys

Table 5.1: Model parameters for the different scenarios.

	b^r	d_1^r	d_2^r	p_1^r	p_2^r	m_1^r	m_2^r	μ	q_1	q_2
Homogeneous	0.1	0.1	0	0.25	0.05	-	-	0.5	-	-
Heterogeneous	0.1	0.1	0	0.25	0.05	0.5	0.5	-	0.3	0.3

density to be $A_0 = N/4$ and $B_0 = N/2$ respectively. Note that the solution of the mean-field equations (5.2.11), without spatial dependence, converges to the stable equilibrium

$$f^* = \frac{2\tilde{b}^r\tilde{p}_1^r - \tilde{b}^r\tilde{d}_1^r - \tilde{p}_1^r\tilde{d}_2^r}{2\tilde{p}_1^r(\tilde{p}_1^r + \tilde{p}_2^r + \tilde{b}^r)}, \qquad g^* = \frac{\tilde{d}_1^r}{2\tilde{p}_1^r}, \tag{5.4.1}$$

that for the parameters choice of Table 5.1 takes the values $f^* = g^* = 0.2$.

Figure 5.2 shows that the error of the efficient Monte Carlo algorithm is proportional to $1/\sqrt{N}$, as the one of classic algorithms in both the predators (left) and preys (right) cases. The errors e_f^N and e_g^N are defined as

$$e_f^N = \|f(t) - f^N(t)\|_\infty, \qquad e_g^N = \|g(t) - g^N(t)\|_\infty, \tag{5.4.2}$$

where $f(t)$, $g(t)$ denote the mean-field solutions and $f^N(t)$, $g^N(t)$ denote the stochastic solutions obtained with the efficient Monte Carlo algorithm at time t for a certain value of N. Recall that $f(t)$, $f^N(t)$ refer to the predators population while $g(t)$, $g^N(t)$ to the preys one.

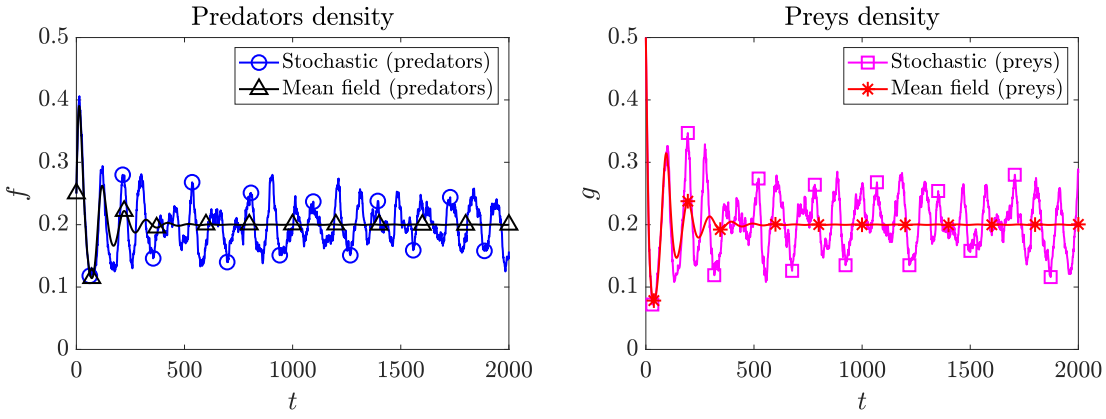

Figure 5.1: Homogeneous predator-prey model: simulation of the processes described in (5.A.1)-(5.A.2) with the efficient Monte Carlo algorithm and solutions of the spatial homogeneous mean-field equations (5.2.11) for $N = 1000$. Markers have been added just to indicate different lines.

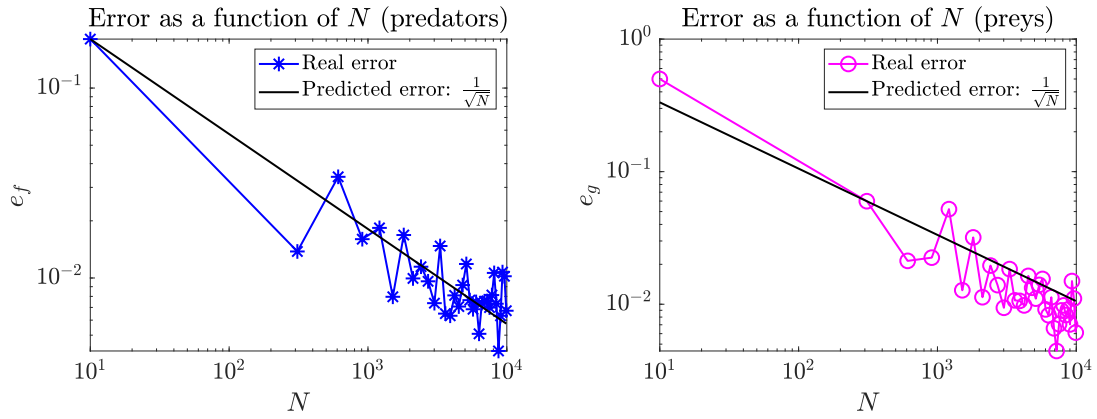

Figure 5.2: Homogeneous predator-prey model: efficient Monte Carlo algorithm's error computed as in equations (5.4.2) for $N = [10, \dots, 10^4]$. The error is proportional to $1/\sqrt{N}$ as the one of the classic Monte Carlo algorithm. Markers correspond to the values e_f^N, e_g^N for a fixed N.

Let us now focus on the heterogeneous model and consider first the one dimensional case. Assume that the dynamics evolves in an interval area of land $[0, L_x]$, $L_x > 0$, divided in M_c cells. Figure 5.3 shows three snapshots taken at time $t = 5$, $t = 50$ and $t = 100$ in which the stochastic and mean-field solutions are compared. At time $t = 5$, $A_0 = N_c/4$ predators and $B_0 = N_c/2$ preys are concentrated in the same central cells. At time $t = 50$, preys migrate in regions where the predators concentration is lower and predators decrease their size and start to migrate to reach the regions occupied by preys. At time $t = 100$, preys continue their migration increasing their size in the regions where the predators concentration is lower. Figure 5.4 shows the asymptotic behavior of the two populations

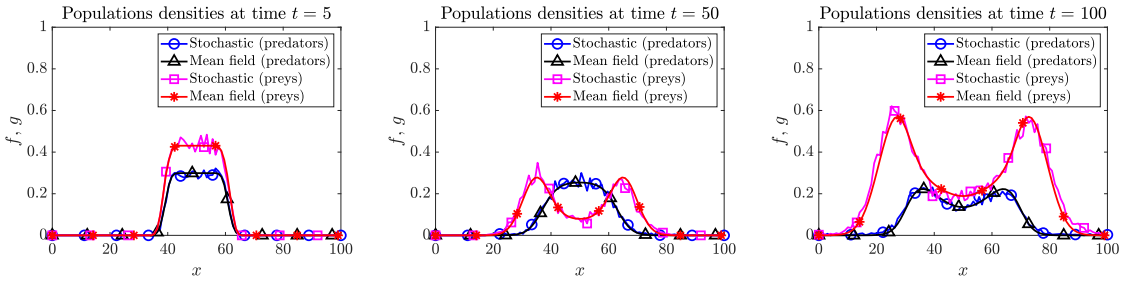

Figure 5.3: Heterogeneous one dimensional predator-prey model: simulation of the processes described in (5.2.1)-(5.2.2)-(5.2.3) with the efficient Monte Carlo algorithm and solutions of the mean-field equations (5.2.11) for $N_c = 1000$. This figure shows three snapshots taken at time $t = 5$ (left), $t = 50$ (centre), $t = 100$ (right). Markers have been added just to indicate different lines.

at the mean-field and stochastic level. Predators and preys migrate in the whole available space reaching in each cell the value given by the equilibrium (5.4.1). Figure 5.5 shows

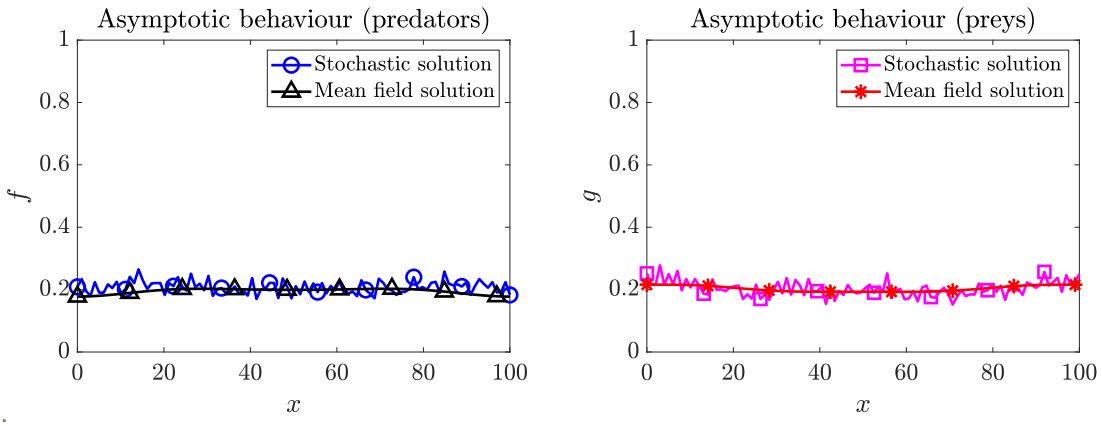

Figure 5.4: Heterogeneous one dimensional predator-prey model: asymptotic behavior (at time $t = 500$) of predators (on the left) and preys (on the right) populations at the mean-field and stochastic level for $N_c = 1000$. Markers have been added just to indicate different lines.

that the error of the efficient Monte Carlo algorithm as a function of N_c is proportional to $1/\sqrt{N_c}$, as the one of classic algorithms. The errors are computed as

$$e_f^{N_c} = \left\langle \max_t |f(x,t) - f^{N_c}(x,t)| \right\rangle_x, \quad e_g^{N_c} = \left\langle \max_t |g(x,t) - g^{N_c}(x,t)| \right\rangle_x, \tag{5.4.3}$$

where $\langle \cdot \rangle$ denotes the expected value with respect to x.

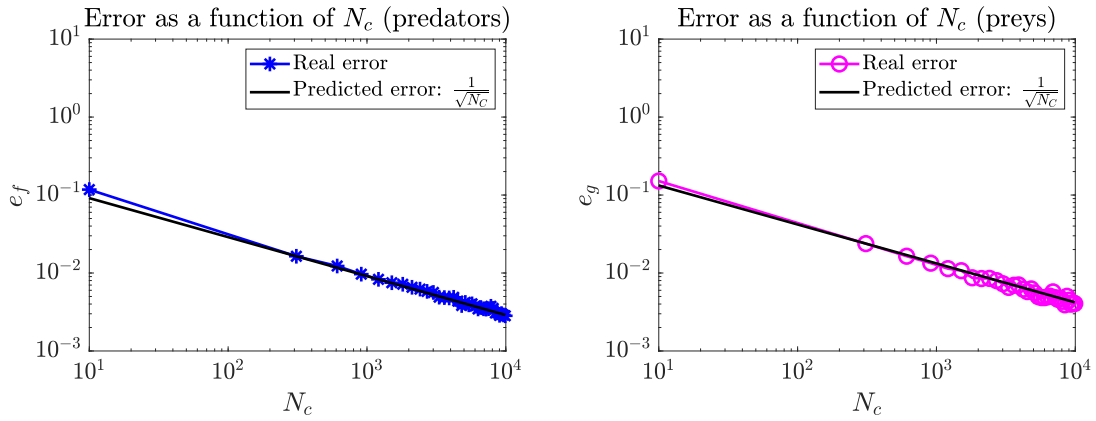

Figure 5.5: Heterogeneous one dimensional predator-prey model: efficient Monte Carlo algorithm's error computed as in equations (5.4.3) for $N_c = [10, \dots, 10^4]$. The error is proportional to $1/\sqrt{N_c}$, as the one of the classic Monte Carlo algorithm. Markers correspond to the values $e_f^{N_c}$, $e_g^{N_c}$ for a fixed N_c.

Let us now focus on the two dimensional case. Assume that the dynamics evolves in a square area of land $[0, L_x] \times [0, L_y]$, $L_x, L_y > 0$ divided in $C_{\ell_x \ell_y}$ cells, for $\ell_x = 1, \dots, M_c^x$ and $\ell_y = 1, \dots, M_c^y$. Populations can born, compete, die and migrate in one of the four nearest cells. The error computed as in equation (5.4.3) assuming $x = (x, y)$ is still proportional to $1/\sqrt{N_c}$, as shown in Figure 5.6.

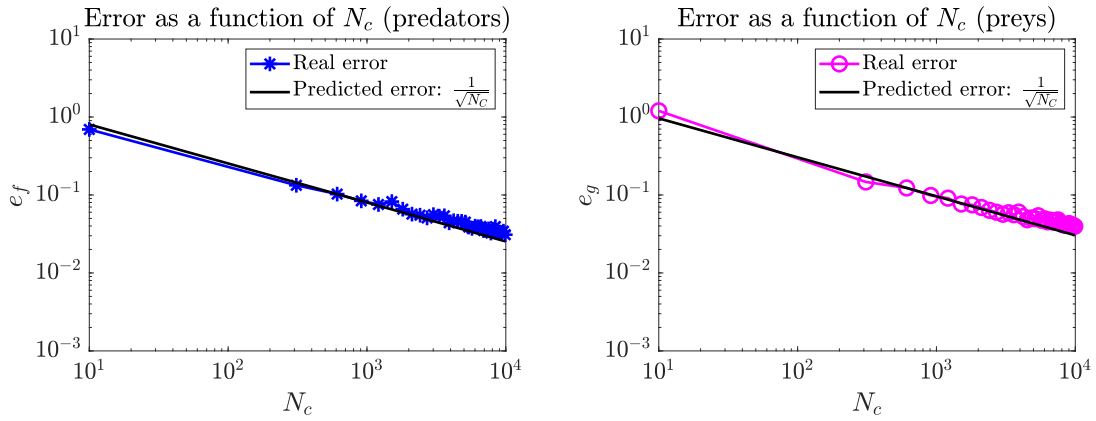

Figure 5.6: Heterogeneous two dimensional predator-prey model: efficient Monte Carlo algorithm's error computed as in equations (5.4.3) for $N_c = [10, \dots, 10^4]$. The error is proportional to $1/\sqrt{N_c}$, as the one of the classic Monte Carlo algorithm. Markers correspond to the values $e_f^{N_c}$, $e_g^{N_c}$ for a fixed N_c.

Figure 5.7 and Figure 5.8 show three snapshots describing the time evolution of preys and predators densities at the mean-field and stochastic level in the heterogeneous two dimensional case for $N_c = 1000$. At time $t = 5$, $B_0 = N_c/2$ preys are concentrated in the central cells and surrounded by $A_0 = N_c/4$ predators. At time $t = 100$ predators migrate in the central cells while preys reduce their size and start to migrate in the regions where the predators concentration is lower. At time $t = 150$ preys are still migrating and increasing their size. Predators on the contrary are reducing their size and migrating to reach the regions in which preys are mainly concentrated.

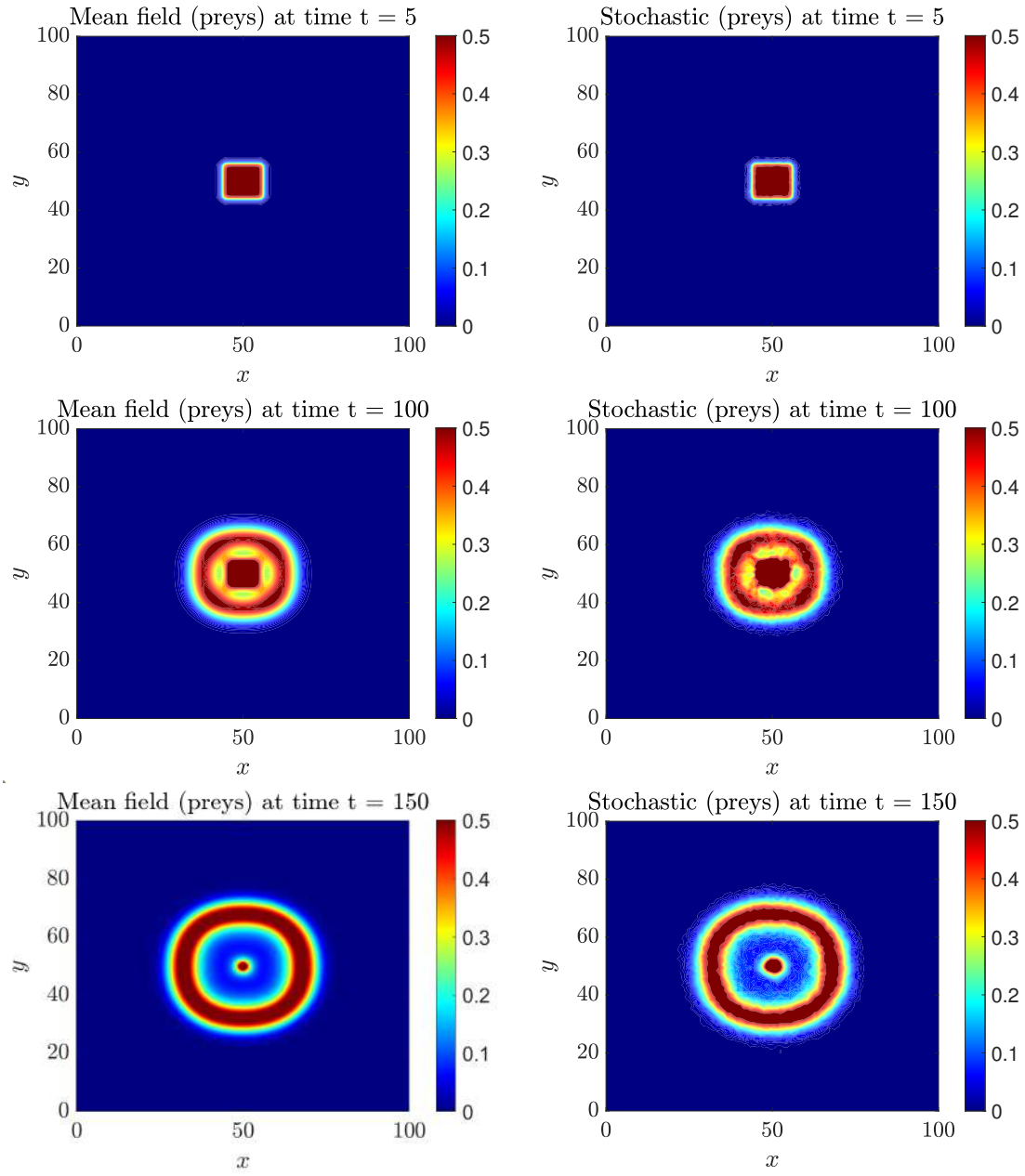

Figure 5.7: Heterogeneous two dimensional predator-prey model (preys population): simulation of the processes described in (5.2.1)-(5.2.2)-(5.2.3) with the efficient Monte Carlo algorithm and solutions of the mean-field equations (5.2.11) in the two dimensional case for $N_c = 1000$. This figure shows three snapshots taken at time $t = 5$ (top), $t = 100$ (middle), $t = 150$ (bottom). On the left, mean-field solutions and on the right, stochastic simulations.

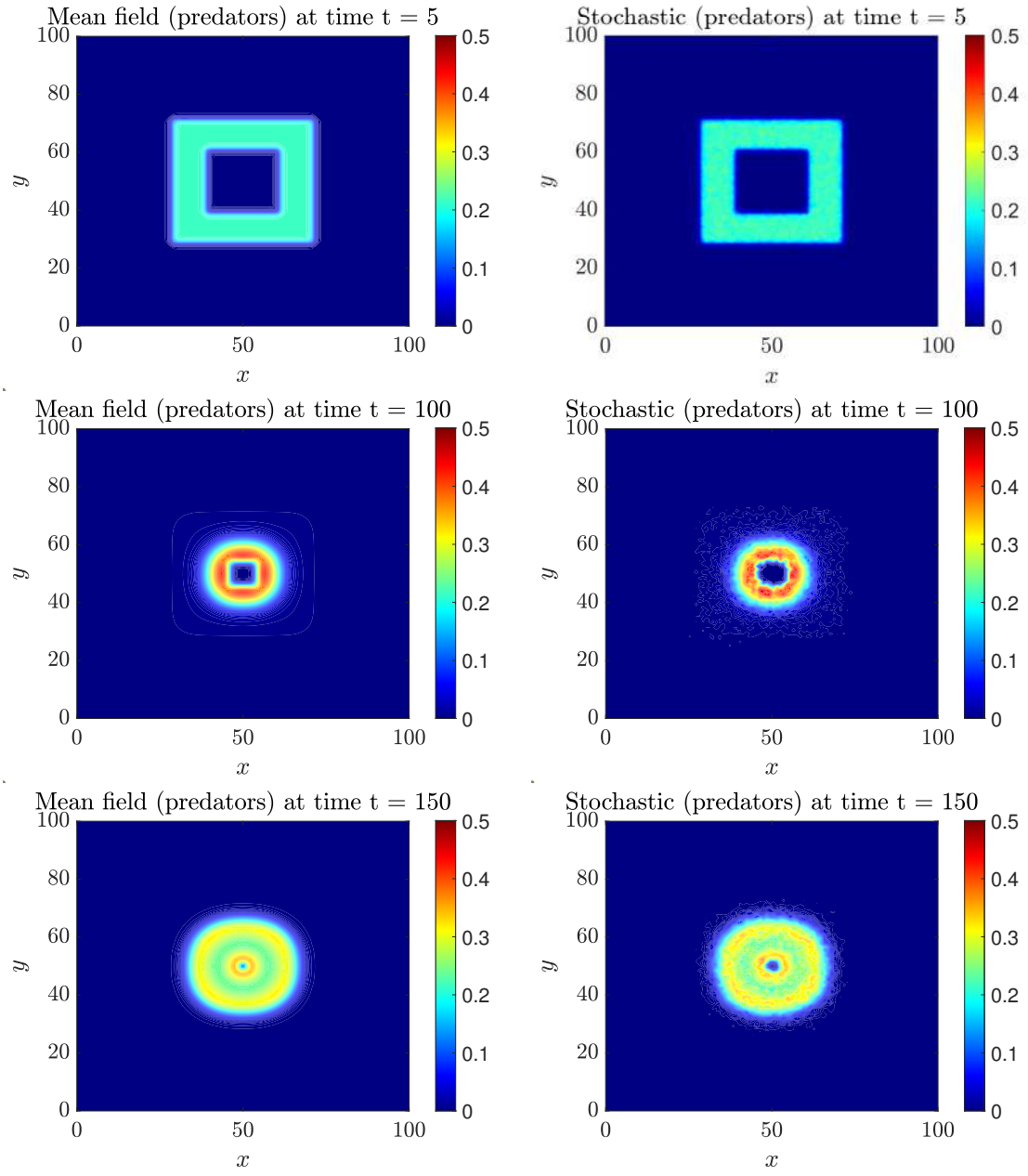

Figure 5.8: Heterogeneous two dimensional predator-prey model (predators population): simulation of the processes described in (5.2.1)-(5.2.2)-(5.2.3) with the efficient Monte Carlo algorithm and solutions of the mean-field equations (5.2.11) in the two dimensional case for $N_c = 1000$. This figure shows three snapshots taken at time $t = 5$ (top), $t = 100$ (middle), $t = 150$ (bottom). On the left, mean-field solutions and on the right, stochastic simulations.

In Figure 5.9 we see for simplicity just the asymptotic behavior of the predators population. One can show that both populations in long time migrate in the whole available space reaching in each cell the value given by the equilibrium (5.4.1).

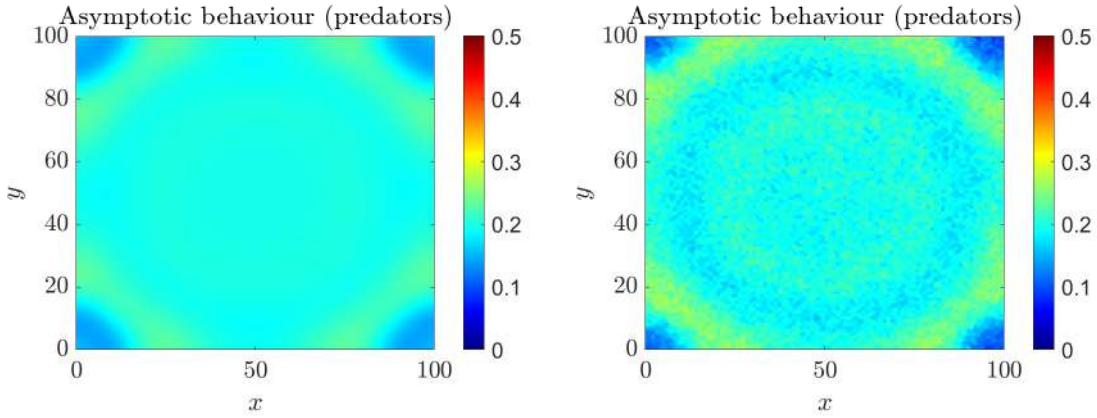

Figure 5.9: Heterogeneous two dimensional predator-prey model: asymptotic behaviour (at time $t = 500$) of predators population at the mean-field (on the right) and stochastic (on the left) level for $N_c = 1000$.

Remark 15. Note that the mean-field solutions of equations (5.2.11) present a damped behavior in time while the stochastic solutions obtained with the efficient Monte Carlo algorithm have a persistent behavior in time. One can prove that the stochastic persistency is due to a resonant effect, [150]. We refer to section 5.4.4 for a dedicated test to analyzed this behavior.

5.4.2 Test 2: Computational cost

Let us consider the homogeneous case and assume to fix the parameters $\mu = 0.5$, $b^r = 1$, $d_1^r = d_2^r = 0.3$, and to let the competition parameters p_1^r, p_2^r to vary between 0.1 and 0.9. Figure 5.10 shows that the computational cost of the efficient Monte Carlo algorithm is lower than the one of the other algorithms. Figure 5.11 shows that the efficient Monte Carlo algorithm, the direct method and the τ-leaping algorithm have a computational complexity of order N in time while the one of the classic Monte Carlo algorithm is of order N^2. Figure 5.12 shows the comparison between the computational costs of the stochastic algorithms in the one dimensional heterogeneous case. The dynamics in (5.2.1)-(5.2.2)-(5.2.3) is simulated for $N_c = 100$, $q_1 = 0.3$, $q_2 = 0.3$ and $b^r = 0.1$, $d_1^r = 0.1$, $d_2^r = 0$, $p_1^r = 0.25$, $p_2^r = 0.05$, letting the migration parameters m_1^r, m_2^r to vary between 0.1 and 0.9. The computational cost of the efficient Monte Carlo algorithm is lower than the one of classic algorithms. Figure 5.13 shows the computational cost of the stochastic algorithms for different values of the

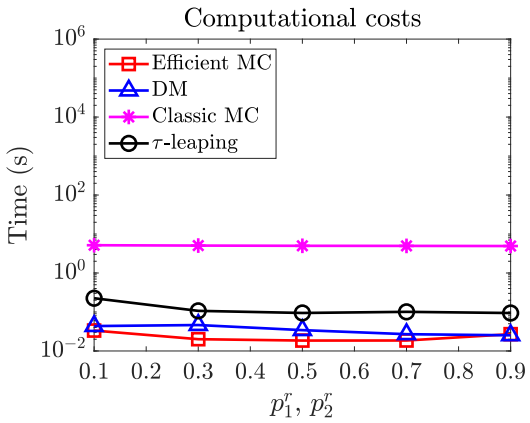

Figure 5.10: Homogeneous predator-prey model: computational cost of efficient and classic Monte Carlo algorithms, direct method and τ-leaping method as the competition parameters vary for N fixed. The dynamics in (5.A.1)-(5.A.2) is simulated for $N = 500$, $\mu = 0.5$, $b^r = 1$, $d_1^r = d_2^r = 0.3$, $p_1^r = p_2^r = 0.1, \ldots, 0.9$. Markers represent the computational costs relative to the parameters choice indicated.

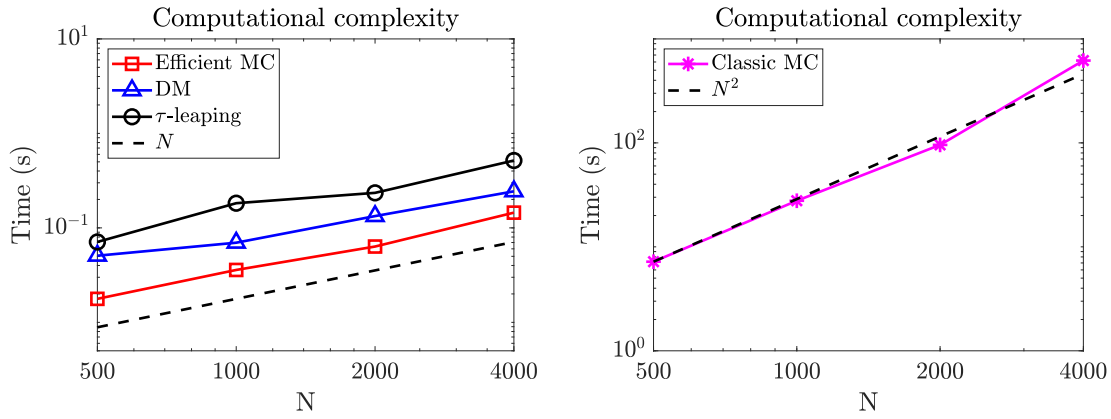

Figure 5.11: Homogeneous predator-prey model: computational complexity of efficient Monte Carlo algorithm, direct method and τ-leaping method (on the left), and classic Monte Carlo algorithms (on the right) as N varies.The dynamics in (5.A.1)-(5.A.2) is simulated for $N = 500, \ldots, 4000$, $\mu = 0.5$, $b^r = 1$, $d_1^r = d_2^r = 0.3$, $p_1^r = p_2^r = 0.5$. Markers represent the computational costs relative to the parameters choice indicated.

migration rates as N_c varies. On the left the computational costs for fixed migration rates $m_1^r = m_2^r = 0.1$ and on the right for $m_1^r = m_2^r = 0.9$. Note that the computational cost of the efficient Monte Carlo algorithm is always lower than the one of the classic algorithms.

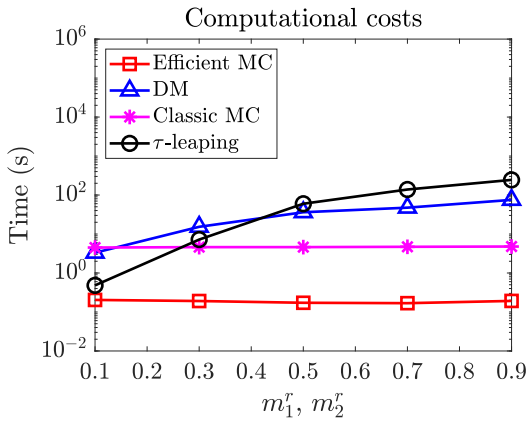

Figure 5.12: Heterogeneous one dimensional predator-prey model: computational cost of efficient and classic Monte Carlo algorithms, direct method and τ-leaping method as the migration parameters vary, for N_c fixed. The dynamics in (5.2.1)-(5.2.2)-(5.2.3) is simulated for $N_c = 100$, $q_1 = 0.3$, $q_2 = 0.3$, $b^r = 0.1$, $d_1^r = 0.1$, $d_2^r = 0$, $p_1^r = 0.25$, $p_2^r = 0.05$, $m_1^r = m_2^r = 0.1, \ldots, 0.9$. Markers represent the computational costs relative to the parameters choice indicated.

In Figure 5.14 a comparison between the computational costs of the stochastic algorithms

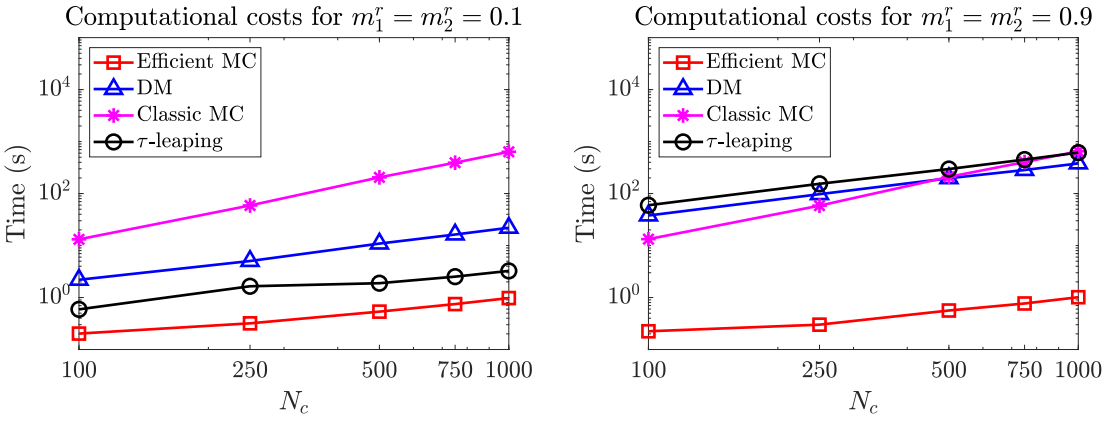

Figure 5.13: Heterogeneous one dimensional predator-prey model: computational cost of efficient and classic Monte Carlo algorithms, direct method and τ-leaping method as N_c varies. The dynamics in (5.2.1)-(5.2.2)-(5.2.3) is simulated for $N_c = 100, \ldots, 1000$, $q_1 = 0.3$, $q_2 = 0.3$, $b^r = 0.1$, $d_1^r = 0.1$, $d_2^r = 0$, $p_1^r = 0.25$, $p_2^r = 0.05$, $m_1^r = m_2^r = 0.1$(on the left), $m_1^r = m_2^r = 0.9$ (on the right). Markers represent the computational costs relative to the parameters choice indicated.

in the two dimensional heterogeneous case. The dynamics is simulated for $N_c = 50$ fixed, $q_1 = 0.3$, $q_2 = 0.3$, $b^r = 0.1$, $d_1^r = 0.1$ $d_2^r = 0$, $p_1^r = 0.25$, $p_2^r = 0.05$, letting the migration parameters to vary between 0.1 and 0.9. Also in this case the efficient Monte Carlo algorithm has a computational cost lower than the one of classic algorithms.

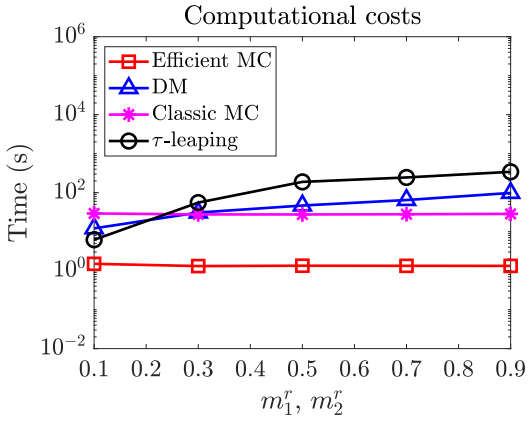

Figure 5.14: Heterogeneous two dimensional predator-prey model: computational cost of efficient and classic Monte Carlo algorithms, direct method and τ-leaping method as the migration parameters vary, for N_c fixed. The dynamics in (5.2.1)-(5.2.2)-(5.2.3) is simulated for $N_c = 50$, $q_1 = 0.3$, $q_2 = 0.3$, $b^r = 0.1$, $d_1^r = 0.1$, $d_2^r = 0$, $p_1^r = 0.25$, $p_2^r = 0.05$, $m_1^r = m_2^r = 0.1, \ldots, 0.9$. Markers represent the computational costs relative to the parameters choice indicated.

5.4.3 Test 3: Accuracy & performances

In the following we will compare the accuracy of the direct method with the one of the efficient Monte Carlo algorithm and of the τ-leaping method. We first focus on the homogeneous case and we define the errors as

$$E_f^N = \|f_{DM}^N(t) - f^N\|_\infty, \quad E_g^N = \|g_{DM}^N(t) - g^N\|_\infty, \tag{5.4.4}$$

where $f_{DM}^N(t)$, $g_{DM}^N(t)$ denote the simulations obtained with the direct method and $f^N(t)$, $g^N(t)$ denote the simulations obtained either with the efficient Monte Carlo or with the τ-leaping algorithms for the predators and preys populations, respectively. Figure 5.15 shows the errors E_f^N, E_g^N as N varies and for fixed birth, competition and death parameters (see Table 5.1). Note that the error related to the simulations obtained with the efficient Monte Carlo algorithm is lower than the one related to the simulated τ-leaping solutions for almost every N. Note also that both errors decrease as the sample size increases reaching a constant value when N is large enough.

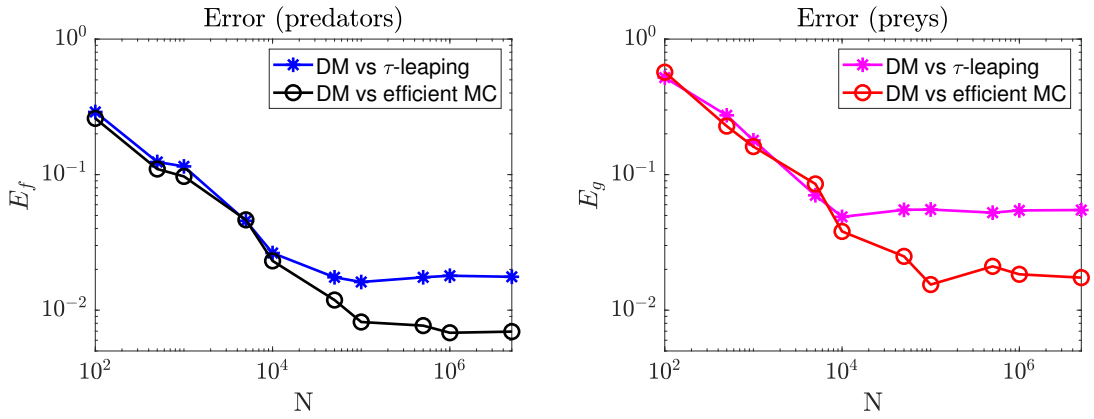

Figure 5.15: Homogeneous predator-prey model: accuracy of the efficient Monte Carlo and of the τ-leaping algorithms with respect to the direct method as N varies. The dynamics in (5.A.1)-(5.A.2) is simulated for $N = 10^2, \dots, 5 \times 10^6$, $\mu = 0.5$, $b^r = 0.1$, $d_1^r = 0.1$, $d_2^r = 0$, $p_1^r = 0.25$, $p_2^r = 0.05$. Markers correspond to the values E_f^N, E_g^N computed as in equations (5.4.4) for a fixed N.

The same results can be obtained in the heterogeneous case with the parameters choice reported in Table 5.1, as show in Figure 5.16. Here, for simplicity we focus on the one dimensional case and we define the errors as

$$E_f^{N_c} = \left\langle \max_t |f^{N_c}(x,t) - f^{N_c}(x,t)| \right\rangle_x, \quad E_g^{N_c} = \left\langle \max_t |g^{N_c}(x,t) - g^{N_c}(x,t)| \right\rangle_x, \quad (5.4.5)$$

where $\langle \cdot \rangle$ denotes the expected value with respect to x.

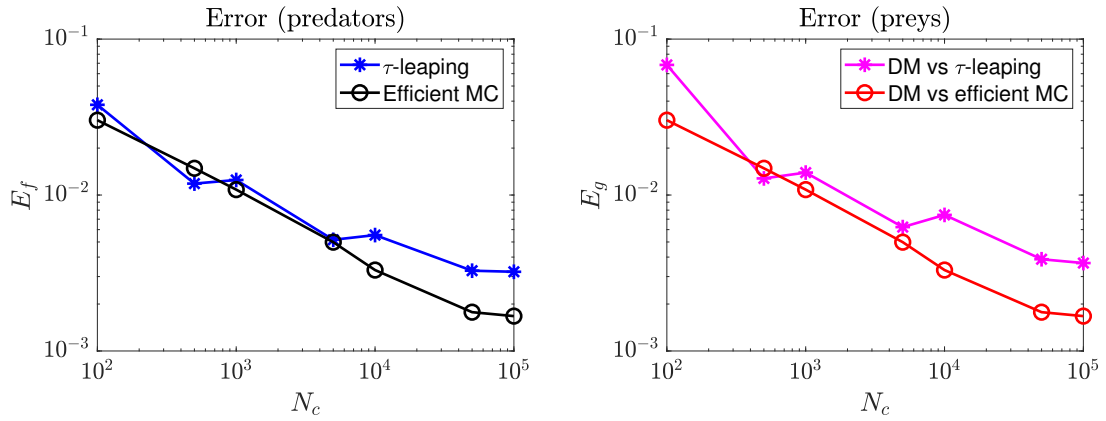

Figure 5.16: Heterogeneous predator-prey model: accuracy of the efficient Monte Carlo and of the τ-leaping algorithms with respect to the direct method as N_c varies. The dynamics in (5.2.1)-(5.2.2)-(5.2.3) is simulated for $N_c = 10^2, \dots, 10^5$, $q_1 = q_2 = 0.3$, $b^r = 0.1$, $d_1^r = 0.1$, $d_2^r = 0$, $p_1^r = 0.25$, $p_2^r = 0.05$, $m_1 = 0.5$, $m_2 = 0.5$. Markers correspond to the values $E_f(N_c), E_g(N_c)$ computed as in equations (5.4.5) for a fixed N_c.

Figure 5.17 shows the trade off between the error E_f and the computational costs of the efficient Monte Carlo algorithm and of the τ-leaping method in both the homogeneous and heterogeneous case for the predators dynamics. Note that both errors decrease as the computational time increases, that is as the sample size increases. Note also that in the homogeneous case the computational cost is lower but the error is higher than in the heterogeneous case. Here, for simplicity we focus on the predators dynamics but the same results can be obtained if we focus on the preys dynamics.

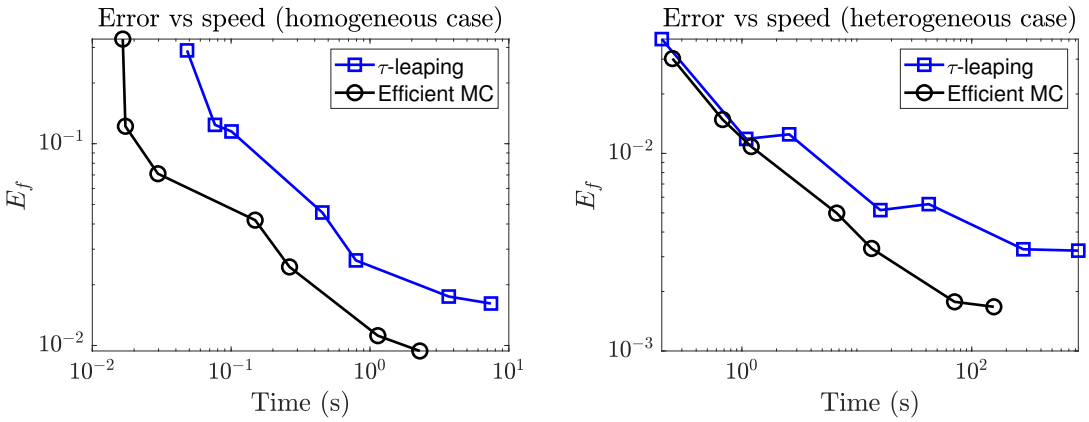

Figure 5.17: Homogeneous and heterogeneous one dimensional predator-prey model: trade-off of error $E_f^{N_c}$ computed as in equation (5.4.4)-(5.4.5) vs computational cost of the efficient Monte Carlo and of the τ-leaping algorithms as the computational cost varies. On the left the trade-off in the homogeneous case, on the right in the heterogeneous case. Markers correspond to the values $E_f^{N_c}$ in relation to the computational costs.

5.4.4 Test 4: Stochastic persistency

We have observed in Figure 5.1 that both mean-field solutions and stochastic simulations present an oscillatory behavior. However, the nature of the oscillations seems to be different. Following the idea in [150], we can prove that oscillations in predators and preys stochastic simulations should be of order $1/\sqrt{N}$, as the error of the Monte Carlo method, but indeed are amplified by a large factor due to noise resonant effect.

In order to analyze the nature of oscillations we introduce the following linearized model corresponding to a perturbation of the mean-field dynamics (5.2.9) in absence of spatial diffusion, see also Appendix 5.A,

$$\frac{d}{dt}\begin{bmatrix} f^N \\ g^N \end{bmatrix} = \Psi \begin{bmatrix} f^N \\ g^N \end{bmatrix} + \Phi\xi, \tag{5.4.6}$$

where Ψ is the Jacobian matrix of the mean-field equations evaluated at the equilibrium (f^*, g^*), defined in (5.4.1), Φ corresponds to the following matrix

$$\Phi = \begin{bmatrix} 0 & \sqrt{2\tilde{p}_1^r f^* g^*} & 0 & -\sqrt{\tilde{d}_1^r f^*} \\ \sqrt{2\tilde{b}^r g^*(1-f^*-g^*)} & -\sqrt{2\tilde{p}_1^r f^* g^*} & -\sqrt{2\tilde{p}_2^r f^* g^* + \tilde{d}_2^r g^*} & 0 \end{bmatrix},$$

associated to the white noise vector $\xi = [\xi_1, \xi_2, \xi_3, \xi_4]^T$, with ξ_i i.i.d distributed random numbers with zero mean and variance $1/\sqrt{N}$, [113],[184]. We focus on the predator density,

we compute the Fourier transform of the mean-field density $f(t)$, i.e. $\tilde{f}(\omega)$, and we derive the power spectrum

$$P(\omega) = \left\langle |\tilde{f}(\omega)|^2 \right\rangle = \frac{\Theta + \Lambda\omega^2}{(\omega^2 - \Omega_0^2)^2 + \Gamma^2\omega^2}, \tag{5.4.7}$$

where $\langle\cdot\rangle$ is the expected value w.r.t. ξ. The parameters $\Theta, \Lambda, \Omega_0, \Gamma$ are defined as follows

$$\Theta = \left\langle [\Psi_{12}(\Phi_{21}\xi_1 + \Phi_{22}\xi_2 + \Phi_{23}\xi_3) - \Psi_{22}(\Phi_{12}\xi_2 + \Phi_{14}\xi_4)]^2 \right\rangle$$
$$\Lambda = \left\langle [(\Phi_{12}\xi_2 + \Phi_{14}\xi_4]^2 \right\rangle, \quad \Omega_0^2 = \Psi_{12}|\Psi_{21}|, \quad \Gamma = |\Psi_{22}|.$$

Note that $P(\omega)$ resembles the one of a damped harmonic oscillator, [124]. As shown in Figure 5.18 on the left, the solution $f(t)$ of the system of equations (5.4.6) without the influence of noise behaves as an under-damped harmonic oscillator. The damped term is $\Gamma < 1$ and oscillations decrease exponentially as $\tilde{A}\exp\left(-\frac{\Gamma}{2}t\right)$ where $\tilde{A}$ is the maximum oscillations amplitude. This behavior recalls the one of the mean-field solutions. If we introduce the noise term, we see in Figure 5.18 on the right that the oscillations amplitude is influenced both by damped and resonant effects and the oscillations become sustained as the one of the stochastic simulations shown in Section 5.4. Hence the resonant effect is a consequence of the white noise that is not an external factor but it is due to the stochasticity of birth, death and competitions events.

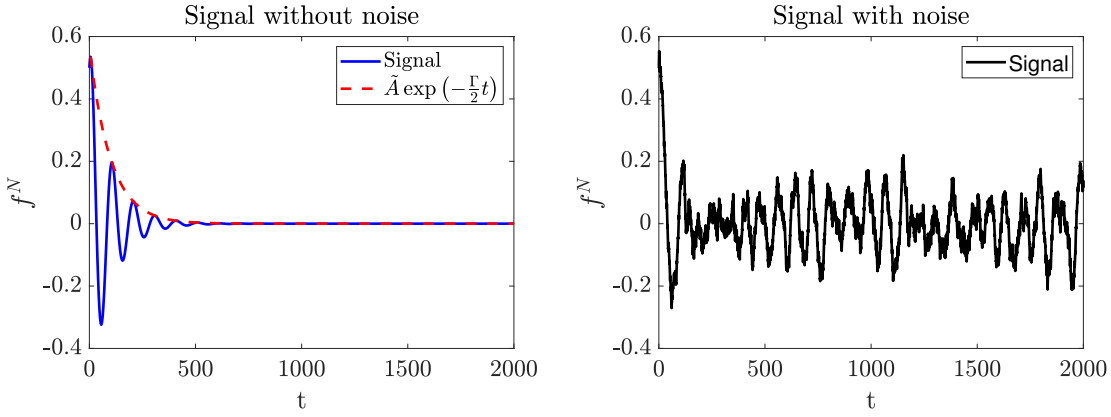

Figure 5.18: Predators density without noise (on the left) and with noise (on the right) as solution of equations (5.4.6).

In Figure 5.19 the power spectrum computed as in equation (5.4.7) and the one obtained averaging the results of 500 simulations of the processes described in (5.A.1)-(5.A.2) computed with the efficient Monte Carlo algorithm and with the direct method for different values of N and for the parameters choice reported in the first line of Table 5.1. Note that the two simulated power spectra agree with the one computed as in equation (5.4.7),

especially when N is large. Note also that the amplitude of the oscillations decreases as N increases.

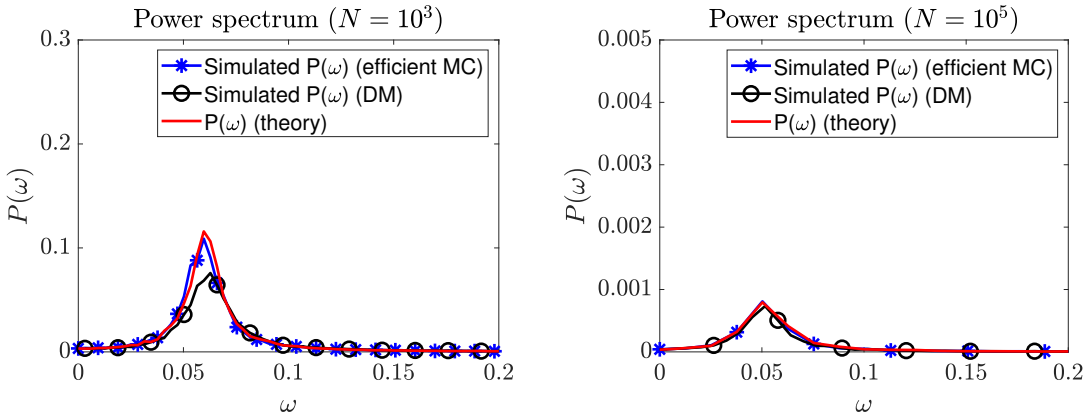

Figure 5.19: Homogeneous case: power spectrum computed as in equation (5.4.7) in red, averaging 500 simulations of the processes described in (5.A.1)-(5.A.2) computed with the efficient Monte Carlo algorithm in blue and with the direct method in black. On the left $N = 10^3$ and on the right $N = 10^5$. Markers have been added just to indicate different lines.

The same results can be obtained in the heterogeneous case in each cell C_ℓ, $\ell = 1, \dots, M_c$, as the time t varies. In Figure 5.20, we see an example of persistent oscillatory behavior in stochastic simulations compared with the damped one of the mean-field solutions in cell C_ℓ, $\ell = 25, 50$ for $N_c = 100$, $M_c = 100$.

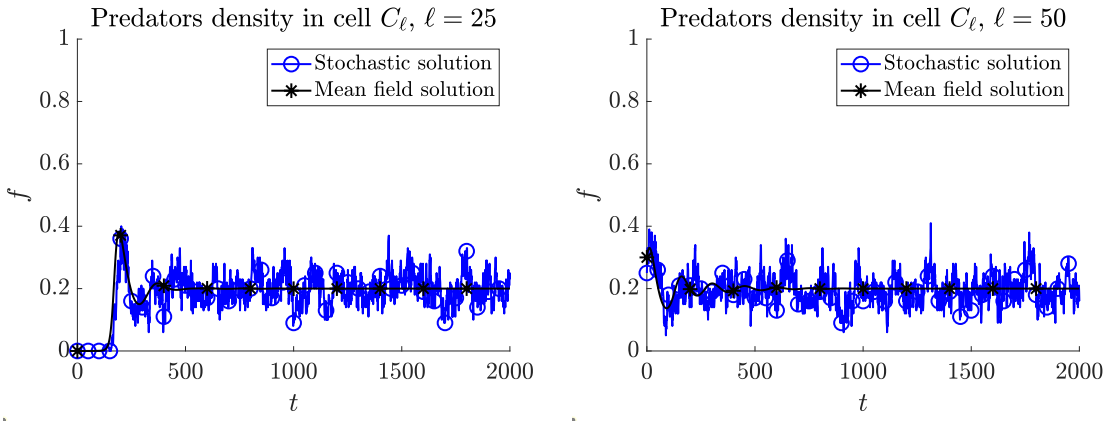

Figure 5.20: Heterogeneous case: time evolution of the predators density in cell C_ℓ, $\ell = 25, 50$. Parameters: $N_c = 100$, $b^r = 0.1$, $d_1^r = 0.1$, $d_2^r = 0$, $p_1^r = 0.25$, $p_2^r = 0.05$, $m_1^r = m_2^r = 0.5$, cell C_ℓ, $\ell = 25, 50$. Markers have been added just to indicate different lines.

In Figure 5.21 on the left the power spectrum computed averaging the results of 500 simulations of the processes described in (5.2.1)-(5.2.2)-(5.2.3) in cell C_ℓ, $\ell = 25, 50$ with the efficient Monte Carlo algorithm for the parameters choice reported in the second line of Table 5.1 for $N_c = 100$ and $T = 2000$. On the right, the average between the power spectra computed in cell C_ℓ, $\ell = 1, \ldots, M_c$ obtained with the efficient Monte Carlo algorithm and with the direct method. Note that also in the heterogeneous case the power spectrum recalls the one of a damped harmonic oscillator. Note also that the two average power spectra agree and so that the oscillations that characterize the simulations obtained with the two algorithms are of the same nature.

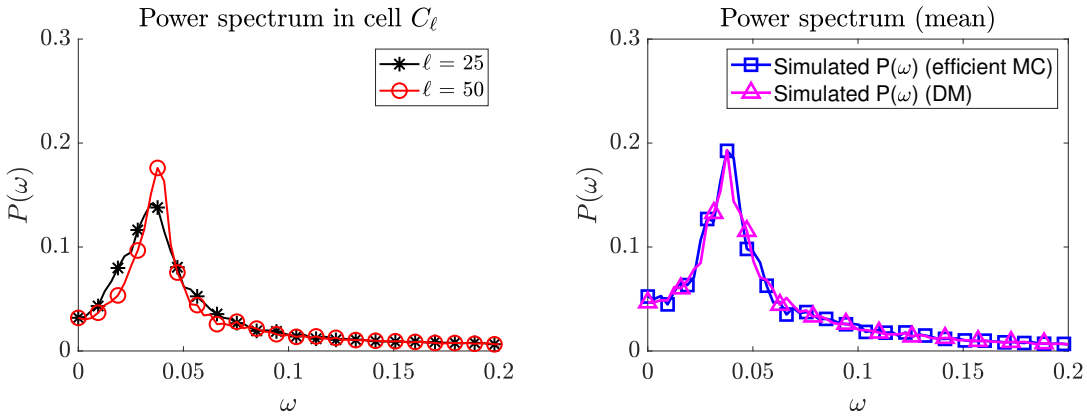

Figure 5.21: Heterogeneous case: power spectrum. On the left, power spectrum computed averaging 500 simulations of the processes described in (5.2.1)-(5.2.2)-(5.2.3) in cell C_ℓ, $\ell = 25, 50$ obtained with the efficient Monte Carlo algorithm. On the right the average between the power spectra computed in cell C_ℓ, $\ell = 1, \dots, M_c$ obtained with the efficient Monte Carlo algorithm (in blue) and with the direct method (in magenta). Markers have been added just to indicate different lines.

5.5 Conclusion

Starting from the microscopic level it is possible to derive the master equation that describes the time evolution of the probability of being in a certain state and the mean-field equations that provide an average description of the behavior of the system which is valid only when the sample size N is arbitrary large. Stochastic models instead describe the individuals behavior of interacting species in a more realistic context. However, classic stochastic algorithms have high computational costs, especially when the sample size increases. In this chapter, we have presented a valid efficient stochastic algorithm that, compared to other algorithms, has lower computational costs, and equivalent, or even better performances. First we have proved its consistency providing a reformulation of the model and showing that the mean-field equations are still valid. Then with different numerical experiments we have proved that the efficient Monte Carlo algorithm produces solutions that oscillate around the mean-field ones and that the error between stochastic simulations and mean-field solutions as a function of N is still proportional to $1/\sqrt{N}$. We have also shown which is the main advantage of this new method proving that its computational cost is lower than the one of classical algorithms for a huge set of parameters but its accuracy is preserved. In the end, we have compared the oscillatory behavior of the stochastic and mean-field solutions underling the differences in their oscillatory behavior. Due to the flexible formulation of the ensemble stochastic algorithm it can be used not only to simulate the microscopic dynamics of interacting biological entities, but further generalized to other agent-based

models, with different interaction dynamics and different macroscopic limiting models.

Appendix 5.A Spatial homogeneous predator-prey model

We report here the agent-based model associated to spatial homogeneous dynamics.

Agent-based dynamics At each instant of time we select one individual and with probability $\mu \in [0,1]$ we assume that an interaction occurs with another individual, randomly chosen among the remaining $N-1$, according to the following rules

$$s_B s_E \xrightarrow{b^r} s_B s_B, \quad s_A s_B \xrightarrow{p_1^r} s_A s_A, \quad s_A s_B \xrightarrow{p_2^r} s_A s_E, \tag{5.A.1}$$

where b^r, p_1^r, p_2^r are birth, and competition rates, and when interaction occurs among two empty components no change is accounted. Otherwise, with probability $1-\mu$, we assume that the sampled individual changes according to death events with rates d_1^r, d_2^r, as follows

$$s_A \xrightarrow{d_1^r} s_E, \quad s_B \xrightarrow{d_2^r} s_E. \tag{5.A.2}$$

We suppose that if the selected component is empty then no changes in the populations sizes happen. We assume that at most one birth/competition and one death event can occur at each time step.

We focus on the evolution of the total number of predators and preys considering the transition rates from state $\mathbf{x} = (A, B, E)$ to the states $\mathbf{x} + \mathbf{v}_j$, where $\mathbf{v}_j$ denotes the j-th row of the stoichiometry matrix

$$V = \begin{bmatrix} 0 & 1 & -1 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix}, \tag{5.A.3}$$

whose columns represent predators, preys and empty components respectively and whose rows represent the changes in the populations sizes due to the birth, competition and death events described in (5.A.1)-(5.A.2). For such predator-prey dynamics the transition rates write as follows

$$\begin{aligned} &\pi_{\mathbf{v}_1}(\mathbf{x}) = 2\mu b^r \frac{B}{N}\frac{E}{N-1}, \quad \pi_{\mathbf{v}_2}(\mathbf{x}) = 2\mu p_1^r \frac{A}{N}\frac{B}{N-1}, \quad \pi_{\mathbf{v}_3}(\mathbf{x}) = 2\mu p_2^r \frac{A}{N}\frac{B}{N-1}, \\ &\pi_{\mathbf{v}_4}(\mathbf{x}) = (1-\mu) d_1^r \frac{A}{N}, \quad \pi_{\mathbf{v}_5}(\mathbf{x}) = (1-\mu) d_2^r \frac{B}{N}, \end{aligned} \tag{5.A.4}$$

where we define the operators $\pi_{\mathbf{v}_j}(\mathbf{x}) = \pi(\mathbf{x} + \mathbf{v}_j | \mathbf{x})$ for any $j = 1, \ldots, M$. Here, $M = 5$ denotes the total number of events described in (5.A.1)-(5.A.2). In Figure 5.22 we depicted

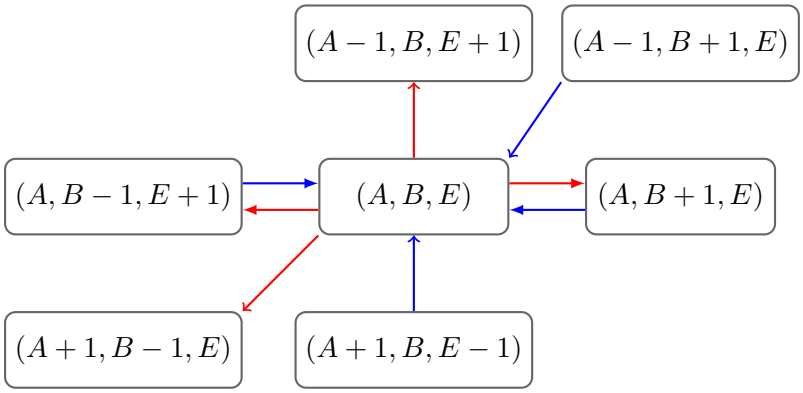

Figure 5.22: Neighbor states are connected to state (A, B, E) either with outgoing arrows (red) for loss events, or with incoming arrows (blue) for gain events.

for A predators, B preys and E empty states all possible states $\mathbf{x} + \mathbf{v}_j$ after one step of process (5.A.1)-(5.A.2). Neighbor states $\mathbf{x} + \mathbf{v}_j$ are connected to state $\mathbf{x}$ either with outgoing arrows (red) for loss events, or with incoming arrows (blue) for gain events. Given the probability $P(\mathbf{x}, t)$ to be in the state $\mathbf{x}$ with A predators, B preys and E empty spaces at time t, its time evolution is governed by the associated master equation as follows

$$\frac{dP(\mathbf{x},t)}{dt} = \sum_{j=1}^{M} \left[\left(\mathcal{E}^{\mathbf{v}_j} - 1 \right) \left(P(\mathbf{x},t) \pi(\mathbf{x} + \mathbf{v}_j | \mathbf{x}) \right) \right], \tag{5.A.5}$$

where the step operator $\mathcal{E}^{\mathbf{v}}$ acts on an arbitrary function f according to

$$\mathcal{E}^{\mathbf{v}_j} \left(f(\mathbf{x}) \right) = f(\mathbf{x} - \mathbf{v}_j). \tag{5.A.6}$$

Mean-field type approximations In analogy to Section 5.2, the spatial homogeneous mean-field approximation (5.2.11) can be derived from the master equation (5.A.5) introducing the empirical densities f^N, g^N as in (5.2.8).

A different approximation is obtained considering a Taylor expansion of (5.A.6) up to second order. In this framework, the master equation (5.A.5) is approximated by the following Fokker-Planck equation

$$\begin{aligned} \frac{dP(\mathbf{x},t)}{dt} = &- \sum_{i=1}^{3} \partial_{x_i} \left(\sum_{j=1}^{M-1} A_{ij}(\mathbf{x}) P(\mathbf{x},t) \right) + \frac{1}{2} \sum_{i=1}^{3} \partial_{x_i x_i} \left(\sum_{j=1}^{M-1} D_{ij}(\mathbf{x}) P(\mathbf{x},t) \right) + \\ &\sum_{i=1}^{3} \sum_{\substack{k=1 \\ k \neq i}}^{3} \partial_{x_i x_k} \left(\sum_{j=1}^{M-1} C_{ikj}(\mathbf{x}) P(\mathbf{x},t) \right), \end{aligned} \tag{5.A.7}$$

where

$$A_{ij}(\mathbf{x}) = \mathbf{v}_j^i \Pi_{\mathbf{v}_j}(\mathbf{x}), \quad D_{ij}(\mathbf{x}) = (\mathbf{v}_j^i)^2 \Pi_{\mathbf{v}_j}(\mathbf{x}), \quad C_{ikj}(\mathbf{x}) = \mathbf{v}_j^i \mathbf{v}_j^k \Pi_{\mathbf{v}_j}(\mathbf{x}). \tag{5.A.8}$$

Here $\mathbf{v}_j^i$ denotes the (i,j) component of the stoichiometry matrix V and $\Pi_{\mathbf{v}_l}(\mathbf{x}) = \pi_{\mathbf{v}_l}(\mathbf{x})$ for $l \in \{1,2,4\}$ and $\Pi_{\mathbf{v}_3}(\mathbf{x}) = \pi_{\mathbf{v}_3}(\mathbf{x}) + \pi_{\mathbf{v}_5}(\mathbf{x})$. As stated in [113], [184], [112], equation (5.A.7) is the Fokker-Planck equation associated to the Langevin equations

$$\frac{dx_i}{dt} = \sum_{j=1}^{M-1} \left(A_{ij}(\mathbf{x}) + \sqrt{D_{ij}(\mathbf{x})}\xi_j \right), \tag{5.A.9}$$

for $i = 1, \dots, 3$, where ξ_j, $j = 1, \dots, M-1$ are normally distributed random numbers with zero mean and variance equal to $1/\sqrt{N}$. Hence, multiplying both sides of equation (5.A.9) by $P(\mathbf{x},t)/N$ and summing over all the values of x_i, $i = 1, \dots, 3$ we can write

$$\begin{aligned}
\frac{df^N}{d\tau} &= (\beta g^N - \delta) f^N + \sqrt{2\tilde{p}_1^r f^N g^N}\xi_2 - \sqrt{\tilde{d}_1^r f^N}\xi_4, \\
\frac{dg^N}{d\tau} &= r g^N \left(1 - \frac{g^N}{K}\right) - \alpha f^N g^N + \sqrt{2\tilde{b}^r g^N (1 - f^N - g^N)}\xi_1 \\
&\quad - \sqrt{2\tilde{p}_2^r f^N g^N + \tilde{d}_2^r}\xi_3 - \sqrt{2\tilde{p}_1^r f^N g^N}\xi_2,
\end{aligned} \tag{5.A.10}$$

where we have substituted the value A_{ij} and D_{ij} defined in (5.A.8). Time is scaled according to $\tau = t/N$, and parameters are defined as $\beta = 2\mu p_1^r$, $\delta = (1-\mu)d_1^r$, $\alpha = 2\mu(p_1^r + p_2^r + b^r)$,$r = 2\mu b^r - (1-\mu)d_2^r$, and $K = 1 - \frac{(1-\mu)d_2^r}{2\mu b^r}$. Note that in the limit $N \to \infty$ the Langevin equations in (5.A.10) collapse to the spatial homogeneous mean-field equations (5.2.11).

Appendix 5.B Monte Carlo algorithms for agent-based dynamics

We report some of the classic stochastic algorithms for agent-based models with predator-prey interaction without spatial interaction. We refer to [111, 114, 56] for classical approaches, and to [146] for a comprehensive collection on stochastic algorithms for agent-based dynamics.

Direct method The main idea of the direct method is to estimate which is the next firing event j and at which time τ it will occur from the probability density function

$$p(\tau, j|\mathbf{x}, t) = a_j(\mathbf{x}) e^{-a_0(\mathbf{x})\tau}. \tag{5.B.1}$$

Here $a_j(\mathbf{x}) = N\pi(\mathbf{x} + \mathbf{v}_j|\mathbf{x})$ represents the propensity of the event j and a_0 is the sum of all the propensities. At each time t suppose to split the probability density function defined

in equation (5.B.1) into the product of two probability functions, one for the firing time τ and the other for the event j that occurs at time $t+\tau$, as follows

$$p(\tau, j|\mathbf{x}, t) = p_1(\tau|\mathbf{x}, t)p_2(j|\tau, \mathbf{x}, t), \tag{5.B.2}$$

where

$$p_1(\tau|\mathbf{x}, t) = a_0(\mathbf{x})e^{-a_0(\mathbf{x})\tau}, \quad p_2(j|\tau, \mathbf{x}, t) = \frac{a_j(\mathbf{x})}{a_0(\mathbf{x})}. \tag{5.B.3}$$

Consider two uniformly distributed random numbers $r_1, r_2 \sim U(0,1)$, the index of the next firing event j can be computed as the smallest index such that

$$\sum_{k=1}^{j} a_k(\mathbf{x}) \geq r_1 a_0(\mathbf{x}). \tag{5.B.4}$$

The time in which the next event happens can be computed from the first equation in (5.B.3) as

$$\tau = \frac{1}{a_0(\mathbf{x})} \ln(r_2^{-1}). \tag{5.B.5}$$

Algorithm 5.B.1 outlines the details of the direct method.

Algorithm 5.B.1 (Direct method)**.**

1. Define the stoichiometry matrix V, the initial state $\mathbf{x}$, the initial time $t = 0$ and the final time T.

2. `while` $t < T$

   (a) Compute the propensities a_j, $j = 1, \ldots, M$.

   (b) Consider two uniformly distributed random numbers $r_1, r_2 \sim U(0,1)$.

   (c) Select the index of the next firing event as in equation 5.B.4.

   (d) Compute the time in which the next event fires as in equation 5.B.5.

   (e) Set $\mathbf{x} = \mathbf{x} + \mathbf{v}_j$ where $\mathbf{v}_j$ is the j-th row of the stoichiometry matrix V defined as in equation 5.A.3.

   (f) Set $t \leftarrow t + \tau$.

   `repeat`

Classic Monte Carlo algorithm The classic Monte Carlo method is an approximated stochastic algorithm. It allows multiple events to occur at the same time step that is supposed to be inversely proportional to the sample size N. Its computational costs increase as the sample size increases and can be comparable with the ones of exact algorithms. The main idea of the classic Monte Carlo algorithm is the following. Suppose to divide a priori the time interval considering a constant time step τ/N. At each time t select two components and assume that with probability $\mu \in [0,1]$ a birth or a competition event can occur. If the two selected components are

- a prey and an empty space: with probability $b_r\tau$ a new prey born and occupies the empty space;
- a predator and a prey: with probability $p_1^r\tau$ a new predator born and with probability $p_2^r\tau$ the prey dies and a new empty space is added to the system.

Extract another component and assume that with probability $1-\mu$ a death event can occur. If the selected component is occupied by a predator then with probability $d_1^r\tau$ the predator dies and if it is occupied by a prey then with probability $d_2^r\tau$ the prey dies. Algorithm 5.B.2 defines the details of the classic Monte Carlo algorithm.

Algorithm 5.B.2 (Classic Monte Carlo algorithm)**.**

1. Define the sample, the initial time $t = 0$, the final time T, the time step as τ/N and a parameter $\mu \in [0,1]$.
2. `while` $t < T$
   (a) Select two components inside the sample.
   (b) Assume that with probability μ birth and competition events happen
   $$BE \xrightarrow{b} BB, \quad AB \xrightarrow{p_1} AA, \quad AB \xrightarrow{p_2} AE,$$
   where $b = b^r\tau$, $p_1^r\tau$ and $p_2^r\tau$.
   (c) Select another component in the sample.
   (d) Assume that with probability $1-\mu$ death events happen
   $$A \xrightarrow{d_1} E, \quad B \xrightarrow{d_2} E,$$
   with $d_1 = d_1^r\tau$ and $d_2 = d_2^r\tau$.
   (e) Update the sample.
   (f) Set $t \leftarrow t + \tau/N$.

   `repeat`

τ-leaping method The τ-leaping method is another approximated stochastic algorithm. Its main idea is to assume that multiple events can happen in the same time interval that is suppose to vary in time. Therefore this algorithm, as the Monte Carlo and the direct method, has to cope with high computational costs. In the following we will describe more in details how the τ-leaping algorithm works. Assume that more than one event can fire simultaneously at any time step. The probability that k_j events fires in the time interval $(t, t+\tau)$ follows a Poisson distribution of parameter $a_j(\mathbf{x})\tau$ where $a_j(\mathbf{x})$ is the propensity of event j and τ the time step. The number of firing events k_j is generated by sampling its corresponding Poisson distribution, $Poi(a_j(\mathbf{x})\tau)$. The state can be updated according to the following rule

$$\mathbf{x}(t+\tau) = \mathbf{x}(t) + \sum_{j=1}^{M} k_j \mathbf{v}_j = \mathbf{x} + \sum_{j=1}^{M} Poi(a_j(\mathbf{x})\tau)\mathbf{v}_j. \tag{5.B.6}$$

The time step τ is updated according to a leap selection:

- choose an initial time step τ;
- let $0 < \epsilon < 1$ and until $\Delta a_j(\mathbf{x}) := |a_j(\mathbf{x}(t+\tau)) - a_j(\mathbf{x}(\tau))| > \epsilon$ set $\tau = \tau/2$.

Algorithm 5.B.3 outlines the details of the τ-leaping method.

Algorithm 5.B.3 (τ - leaping method)**.**

1. Define the stoichiometry matrix V, the initial state $\mathbf{x}$, the initial time $t = 0$ and the final time T, the time step τ.
2. `while` $t < T$
   (a) Compute the propensities a_j, $j = 1, \ldots, M$.
   (b) Consider M uniformly distributed random numbers $k_1, \ldots, k_M \sim Poi(a_j(\mathbf{x})\tau)$.
   (c) Update the state as in equation (5.B.6).
   (d) Choose τ with leap selection.
   (e) Set $t \leftarrow t + \tau$.

   `repeat`

Chapter 6

Conclusion and perspectives

In this thesis we have exploited different particles schemes to simulate the dynamics of huge groups of agents through the derivation of kinetic equations and through stochastic description. As we have seen, the numerical methods introduced in Chapter 1 can be used to simulate the dynamics in different contexts. These methods present different advantages, as for instance they can handle the problem of dimensionality in computing the interaction term, or, thanks to their stochastic nature, they can be useful to preserve the natural aspects which characterize the dynamics, such as randomness and uncertainty. However, their accuracy depends on the total number of particles used for simulations and in low populated regions they can produce distorted results. For what concern the control of plasma instability exploited in Chapter 2 we might be interested in providing a theoretical study of the controllability of the system, and in testing the results with high order schemes. We might be also interested in deriving a finite horizon control of the system, and in studying the Vlasov-Poisson equation with uncertainties via machine learning methods. Then, several questions arise concerning the study of non-local interactions. As we have seen in Chapter 3, topological interactions which induce non-locality can be efficiently simulated by a k-nn search algorithm. However, the proposed numerical scheme can be further improved by studying different ways to approximate this kind of interactions. Furthermore, the numerical scheme proposed for global optimization in Chapter 4 can be also studied in different contexts such as for instance in constrained optimization [37] or in models involving fractional diffusion [133]. Finally, the approximated stochastic algorithms introduced in Chapter 5 might be further improved and generalized to other agent-based models.

Acknowledges

Vorrei innanzitutto ringraziare il mio supervisore di dottorato Prof. Giacomo Albi per avermi guidata in questi anni, introducendomi al mondo della ricerca. Grazie per avermi supportato e avermi dato la possibilità di viaggiare nonostante il brutto periodo in seguito alla pandemia. Grazie anche per avermi fatto conoscere diversi collaboratori, tra i quali Prof. Lorenzo Pareschi e Prof. Giacomo Dimarco che mi hanno accolta a Ferrara offrendomi la mia posizione attuale. In secondo luogo ringrazio i referee della mia tesi e i membri della commisione di dottorato Prof. Claudia Totzeck, Prof. Emiliano Cristiani e Prof. Dante Kalise per essersi offerti di leggere la mia tesi e avermi dedicato il loro tempo. Ringrazio poi tutti i colleghi di Verona per i diversi momenti condivisi insieme e per avermi anche aiutata nelle difficoltà e sopportata qualche volta. Fabio per il suo supporto con diversi problemi tecnici e per essere sempre stato il "pezzo di arredo" dell'ufficio. Poi Martina per le lunghe chiaccherate, Chiara per essere la mia sorella maggiore e aver condiviso insieme parte della ricerca, e infine Rossana, Elisa, Giulio e Ramon. Ringrazio le nuove colleghe di Ferrara, Elisa e Giulia, per tutti i pranzi e le serate trascorse insieme. Ringrazio poi tutte le mie amiche e compagne di corso. Anche se ormai ci vediamo poco sono contenta che siate ancora presenti nella mia vita e di aver condiviso parte del mio percorso con voi. Non per ultimo, ringrazio la mia famiglia per avermi sempre appoggiata nelle mie scelte e aver sempre creduto in me. Grazie a tutti voi per aver contribuito a farmi crescere e farmi diventare la persona che sono ora.

Bibliography

[1] G. Albi, S. Almi, M. Morandotti, and F. Solombrino. Mean-field selective optimal control via transient leadership. *Applied Mathematics & Optimization*, 85(2):22, 2022.

[2] G. Albi, N. Bellomo, L. Fermo, S.-Y. Ha, J. Kim, L. Pareschi, D. Poyato, and J. Soler. Vehicular traffic, crowds, and swarms: From kinetic theory and multiscale methods to applications and research perspectives. *Mathematical Models and Methods in Applied Sciences*, 29(10):1901–2005, 2019.

[3] G. Albi, M. Bongini, E. Cristiani, and D. Kalise. Invisible control of self-organizing agents leaving unknown environments. *SIAM Journal on Applied Mathematics*, 76(4):1683–1710, 2016.

[4] G. Albi, M. Bongini, F. Rossi, and F. Solombrino. Leader formation with mean-field birth and death models. *Mathematical Models and Methods in Applied Sciences*, 29(04):633–679, 2019.

[5] G. Albi, R. Chignola, and F. Ferrarese. Efficient ensemble stochastic algorithms for agent-based models with spatial predator–prey dynamics. *Mathematics and Computers in Simulation*, 199:317–340, 2022.

[6] G. Albi, Y.-P. Choi, M. Fornasier, and D. Kalise. Mean field control hierarchy. *Applied Mathematics & Optimization*, 76:93–135, 2017.

[7] G. Albi, E. Cristiani, L. Pareschi, and D. Peri. Mathematical models and methods for crowd dynamics control. *Crowd Dynamics, Volume 2: Theory, Models, and Applications*, pages 159–197, 2020.

[8] G. Albi, F. Ferrarese, and C. Segala. Optimized leaders strategies for crowd evacuation in unknown environments with multiple exits. In *Crowd Dynamics, Volume 3: Modeling and Social Applications in the Time of COVID-19*, pages 97–131. Springer, 2021.

[9] G. Albi, F. Ferrarese, and C. Totzeck. Kinetic based optimization enhanced by genetic dynamics. *Mathematical Models and Methods in Applied Sciences*, 33(14), 2023.

[10] G. Albi, M. Herty, and L. Pareschi. Kinetic description of optimal control problems and applications to opinion consensus. *arXiv preprint arXiv:1401.7798*, 2014.

[11] G. Albi and L. Pareschi. Binary interaction algorithms for the simulation of flocking and swarming dynamics. *Multiscale Modeling & Simulation*, 11(1):1–29, 2013.

[12] G. Albi, L. Pareschi, and M. Zanella. Opinion dynamics over complex networks: Kinetic modelling and numerical methods. *Kinetic & Related Models*, 10(1):1, 2017.

[13] M. Badsi, M. Campos-Pinto, B. Després, and L. Godard-Cadillac. A variational sheath model for stationary gyrokinetic Vlasov–Poisson equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 55(6):2609–2642, 2021.

[14] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, et al. Empirical investigation of starling flocks: a benchmark study in collective animal behaviour. *Animal behaviour*, 76(1):201–215, 2008.

[15] J. Bartsch, P. Knopf, S. Scheurer, and J. Weber. Controlling a Vlasov-Poisson plasma by a Particle-In-Cell method based on a Monte Carlo framework. *arXiv preprint arXiv:2304.02083*, 2023.

[16] W. B. Bateson and D. W. Hewett. Grid and particle hydrodynamics: Beyond hydrodynamics via fluid element Particle-In-Cell. *Journal of Computational Physics*, 144(2):358–378, 1998.

[17] R. Belaouar, N. Crouseilles, P. Degond, and E. Sonnendrücker. An asymptotically stable semi-Lagrangian scheme in the quasi-neutral limit. *Journal of Scientific Computing*, 41:341–365, 2009.

[18] N. Bellomo, D. Burini, G. Dosi, L. Gibelli, D. Knopoff, N. Outada, P. Terna, and M. E. Virgillito. What is life? a perspective of the mathematical kinetic theory of active particles. *Mathematical Models and Methods in Applied Sciences*, 31(09):1821–1866, 2021.

[19] N. Bellomo, L. Gibelli, A. Quaini, and A. Reali. Towards a mathematical theory of behavioral human crowds. *Mathematical Models and Methods in Applied Sciences*, 32(02):321–358, 2022.

[20] D. Benedetto, E. Caglioti, and S. Rossi. Mean-field limit for particle systems with topological interactions. *Mathematics and Mechanics of Complex Systems*, 9(4):423–440, 2022.

[21] A. Benfenati, G. Borghi, and L. Pareschi. Binary interaction methods for high dimensional global optimization and machine learning. *Applied Mathematics & Optimization*, 86(1):9, 2022.

[22] S. Bernardi, A. Colombi, and M. Scianna. A particle model analysing the behavioural rules underlying the collective flight of a bee swarm towards the new nest. *Journal of biological dynamics*, 12(1):632–662, 2018.

[23] S. Bernardi, R. Eftimie, and K. J. Painter. Leadership through influence: what mechanisms allow leaders to steer a swarm? *Bulletin of Mathematical Biology*, 83(6):69, 2021.

[24] S. Bernardi, G. Estrada-Rodriguez, H. Gimperlein, and K. J. Painter. Macroscopic descriptions of follower-leader systems. *Kinetic & Related Models*, 14, 2021.

[25] A. Berryman. *Population cycles: the case for trophic interactions*. Oxford University Press, 2002.

[26] C. Besse, P. Degond, F. Deluzet, J. Claudel, G. Gallice, and C. Tessieras. A model hierarchy for ionospheric plasma modeling. *Mathematical Models and Methods in Applied Sciences*, 14(03):393–415, 2004.

[27] S. Bidari, O. Peleg, and Z. P. Kilpatrick. Social inhibition maintains adaptivity and consensus of honeybees foraging in dynamic environments. *Royal Society open science*, 6(12):191681, 2019.

[28] G. Bird. Approach to translational equilibrium in a rigid sphere gas. *Phys. fluids*, 6:1518–1519, 1963.

[29] J. A. Bittencourt. *Fundamentals of plasma physics*. Springer Science & Business Media, 2013.

[30] A. J. Black and A. J. McKane. Stochastic formulation of ecological models and their applications. *Trends in ecology & evolution*, 27(6):337–345, 2012.

[31] A. Blanchet and P. Degond. Topological interactions in a Boltzmann-type framework. *Journal of Statistical Physics*, 163:41–60, 2016.

[32] A. Blanchet and P. Degond. Kinetic models for topological nearest-neighbor interactions. *Journal of Statistical Physics*, 169(5):929–950, 2017.

[33] B. Blasius, L. Rudolf, G. Withoff, U. Gaedke, and G. F. Fussmann. Long-term cyclic persistence in an experimental predator-prey system. *Nature*, 577:226–230, 2020.

[34] J. Blum. *Numerical simulation and optimal control in plasma physics*. New York, NY; John Wiley and Sons Inc., 1989.

[35] E. Bonabeau, G. Theraulaz, J.-L. Deneubourg, S. Aron, and S. Camazine. Self-organization in social insects. *Trends in ecology & evolution*, 12(5):188–193, 1997.

[36] G. Borghi, M. Herty, and L. Pareschi. A consensus-based algorithm for multi-objective optimization and its mean-field description. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 4131–4136. IEEE, 2022.

[37] G. Borghi, M. Herty, and L. Pareschi. Constrained consensus-based optimization. *SIAM Journal on Optimization*, 33(1):211–236, 2023.

[38] W. Boscheri, G. Dimarco, and L. Pareschi. Modeling and simulating the spatial spread of an epidemic through multiscale kinetic transport equations. *Mathematical Models and Methods in Applied Sciences*, pages 1–39, 2021.

[39] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

[40] A. Bressan, S. Čanić, M. Garavello, M. Herty, and B. Piccoli. Flows on networks: recent results and perspectives. *EMS Surveys in Mathematical Sciences*, 1(1):47–111, 2014.

[41] S. Brull, P. Degond, and F. Deluzet. Degenerate anisotropic elliptic problems and magnetized plasma simulations. *Communications in Computational Physics*, 11(1):147–178, 2012.

[42] M. Burger, R. Pinnau, C. Totzeck, O. Tse, and A. Roth. Instantaneous control of interacting particle systems in the mean-field limit. *Journal of Computational Physics*, 405:109181, 2020.

[43] R. Caflisch, C. Wang, G. Dimarco, B. Cohen, and A. Dimits. A hybrid method for accelerated simulation of Coulomb collisions in a plasma. *Multiscale Modeling & Simulation*, 7(2):865–887, 2008.

[44] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta numerica*, 7:1–49, 1998.

[45] F. Cao, S. Motsch, A. Reamy, and R. Theisen. Asymptotic flocking for the three-zone model. *Mathematical Biosciences and Engineering*, 17(6):7692–7707, 2020.

[46] S. Caprino, G. Cavallaro, and C. Marchioro. Time evolution of a Vlasov-Poisson plasma with magnetic confinement. *Kinetic and Related Models*, 5(4):729–742, 2012.

[47] J. Carrillo, L. Pareschi, and M. Zanella. Particle based gPC methods for mean-field models of swarming with uncertainty. *Communications in Computational Physics*, 25(2):508–531, 2019.

[48] J. A. Carrillo, Y.-P. Choi, C. Totzeck, and O. Tse. An analytical framework for consensus-based global optimization method. *Mathematical Models and Methods in Applied Sciences*, 28(06):1037–1066, 2018.

[49] J. A. Carrillo, M. Fornasier, J. Rosado, and G. Toscani. Asymptotic flocking dynamics for the kinetic Cucker–Smale model. *SIAM Journal on Mathematical Analysis*, 42(1):218–236, 2010.

[50] J. A. Carrillo, M. Fornasier, G. Toscani, and F. Vecil. Particle, kinetic, and hydrodynamic models of swarming. *Mathematical modeling of collective behavior in socio-economic and life sciences*, pages 297–336, 2010.

[51] J. A. Carrillo, Y. Huang, and M. Schmidtchen. Zoology of a nonlocal cross-diffusion model for two species. *SIAM Journal on Applied Mathematics*, 78(2):1078–1104, 2018.

[52] J. A. Carrillo, S. Jin, L. Li, and Y. Zhu. A consensus-based global optimization method for high dimensional machine learning problems. *ESAIM: Control, Optimisation and Calculus of Variations*, 27:S5, 2021.

[53] P. Centorrino, A. Corbetta, E. Cristiani, and E. Onofri. Managing crowded museums: Visitors flow measurement, analysis, modeling, and optimization. *Journal of Computational Science*, 53:101357, 2021.

[54] E. Chacon-Golcher, S. A. Hirstoaga, and M. Lutz. Optimization of Particle-In-Cell simulations for vlasov-poisson system with strong magnetic field. *ESAIM: Proceedings and Surveys*, 53:177–190, 2016.

[55] F. F. Chen. *Introduction to Plasma Physics and Controlled Fusion*. Springer New York, NY, 1984.

[56] S. Chen and U. C. Täuber. Non-equilibrium relaxation in a stochastic lattice Lotka–Volterra model. *Physical biology*, 13(2):025005, 2016.

[57] S. Chen, JingrunJin and L. Lyu. A consensus-based global optimization method with adaptive momentum estimation. *Communications in Computational Physics*, 31(4):1296–1316, 2022.

[58] Y. Chen and T. Kolokolnikov. A minimal model of predator–swarm interactions. *Journal of The Royal Society Interface*, 11(94):20131208, 2014.

[59] C. Cheng and G. Knorr. The integration of the Vlasov equation in configuration space. *Journal of Computational Physics*, 22(3):330–351, 1976.

[60] K. J. Cheung, E. Gabrielson, Z. Werb, and A. J. Ewald. Collective invasion in breast cancer requires a conserved basal epithelial program. *Cell*, 155(7):1639–1651, 2013.

[61] Y.-P. Choi, D. Kalise, J. Peszek, and A. A. Peters. A collisionless singular Cucker–Smale model with decentralized formation control. *SIAM Journal on Applied Dynamical Systems*, 18(4):1954–1981, 2019.

[62] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks. Collective memory and spatial sorting in animal groups. *Journal of theoretical biology*, 218(1):1–11, 2002.

[63] P. Crispel, P. Degond, and M.-H. Vignal. Quasi-neutral fluid models for current-carrying plasmas. *Journal of Computational Physics*, 205(2):408–438, 2005.

[64] E. Cristiani, N. Loy, M. Menci, and A. Tosin. Kinetic and macroscopic description of flocking dynamics with continuous leader-follower transition. *preprint*, 2023.

[65] E. Cristiani, M. Menci, M. Papi, and L. Brafman. An all-leader agent-based model for turning and flocking birds. *Journal of Mathematical Biology*, 83(4):1–22, 2021.

[66] E. Cristiani and D. Peri. Robust design optimization for egressing pedestrians in unknown environments. *Applied Mathematical Modelling*, 72:553–568, 2019.

[67] E. Cristiani, B. Piccoli, and A. Tosin. Modeling self-organization in pedestrians and animal groups from macroscopic and microscopic viewpoints. In *Mathematical modeling of collective behavior in socio-economic and life sciences*, pages 337–364. Springer, 2010.

[68] E. Cristiani, B. Piccoli, and A. Tosin. *Multiscale modeling of pedestrian dynamics*, volume 12. Springer, 2014.

[69] E. Cristiani, F. S. Priuli, and A. Tosin. Modeling rationality to control self-organization of crowds: an environmental approach. *SIAM Journal on Applied Mathematics*, 75(2):605–629, 2015.

[70] N. Crouseilles, G. Dimarco, and M.-H. Vignal. Multiscale schemes for the bgk–Vlasov–Poisson system in the quasi-neutral and fluid limits. stability analysis and first order schemes. *Multiscale Modeling & Simulation*, 14(1):65–95, 2016.

[71] N. Crouseilles and F. Filbet. Numerical approximation of collisional plasmas by high order methods. *Journal of Computational Physics*, 201(2):546–572, 2004.

[72] N. Crouseilles, M. Mehrenberger, and E. Sonnendrücker. Conservative semi-Lagrangian schemes for Vlasov equations. *Journal of Computational Physics*, 229(6):1927–1953, 2010.

[73] F. Cucker and S. Smale. Emergent behavior in flocks. *IEEE Transactions on automatic control*, 52(5):852–862, 2007.

[74] F. Cucker and S. Smale. On the mathematics of emergence. *Japanese Journal of Mathematics*, 2:197–227, 2007.

[75] I. Davies. Magnetic control of tokamak plasmas through deep reinforcement learning. *Bulletin of the American Physical Society*, 2023.

[76] B. A. De Dios, J. A. Carrillo, and C.-W. Shu. Discontinuous Galerkin methods for the multi-dimensional Vlasov–Poisson problem. *Mathematical Models and Methods in Applied Sciences*, 22(12):1250042, 2012.

[77] P. Degond. Asymptotic-Preserving schemes for fluid models of plasmas. Société mathématique de France, 2013.

[78] P. Degond and F. Deluzet. Asymptotic-preserving methods and multiscale models for plasma physics. *Journal of Computational Physics*, 336:429–457, 2017.

[79] P. Degond and M. Pulvirenti. Propagation of chaos for topological interactions. *The Annals of Applied Probability*, 29(4):2594–2612, 2019.

[80] F. Deluzet, G. Fubiani, L. Garrigues, C. Guillet, and J. Narski. Efficient parallelization for 3d-3v sparse grid Particle-in-Cell: Shared memory architectures. *Journal of Computational Physics*, 480:112022, 2023.

[81] F. Deluzet, J. Narski, M. Ndiaye, G. Hagelaar, and J.-P. Boeuf. Numerical methods and macroscopic models of magnetically confined low temperature plasmas. *Kinetic and Related Models*, 16(5):624–653, 2023.

[82] A. Dembo. Zeitouni, 0. large deviations techniques and applications. *Applications of Mathematics*, 38, 1998.

[83] G. Dimarco, R. Caflisch, and L. Pareschi. Direct simulation Monte Carlo schemes for Coulomb interactions in plasmas. *Communications in Applied and Industrial Mathematics*, 1(1):p–72, 2010.

[84] G. Dimarco, Q. Li, L. Pareschi, and B. Yan. Numerical methods for plasma physics in collisional regimes. *Journal of Plasma Physics*, 81(1):305810106, 2015.

[85] G. Dimarco, L. Mieussens, and V. Rispoli. An asymptotic preserving automatic domain decomposition method for the Vlasov–Poisson–bgk system with applications to plasmas. *Journal of Computational Physics*, 274:122–139, 2014.

[86] G. Dimarco, L. Pareschi, G. Toscani, and M. Zanella. Wealth distribution under the spread of infectious diseases. *Physical Review E*, 102(2):022303, 2020.

[87] A. M. Dimits, B. I. Cohen, R. E. Caflisch, M. S. Rosin, and L. F. Ricketson. Higher-order time integration of Coulomb collisions in a plasma using Langevin equations. *Journal of Computational Physics*, 242:561–580, 2013.

[88] A. Duncan, R. Erban, and K. Zygalakis. Hybrid framework for the simulation of stochastic chemical kinetics. *Journal of Computational Physics*, 326:398–419, 2016.

[89] A. Dussutour, S. C. Nicolis, J.-L. Deneubourg, and V. Fourcassié. Collective decisions in ants when foraging under crowded conditions. *Behavioral Ecology and Sociobiology*, 61:17–30, 2006.

[90] M. R. D,ÄôOrsogna, Y.-L. Chuang, A. L. Bertozzi, and L. S. Chayes. Self-propelled particles with soft-core interactions: patterns, stability, and collapse. *Physical review letters*, 96(10):104302, 2006.

[91] L. Einkemmer, Q. Li, L. Wang, and Y. Yunan. Suppressing instability in a Vlasov–Poisson system by an external electric field through constrained optimization. *Journal of Computational Physics*, 498:112662, 2024.

[92] S. Engblom, L. Ferm, A. Hellander, and P. Lötstedt. Simulation of stochastic reaction-diffusion processes on unstructured meshes. *SIAM Journal on Scientific Computing*, 31(3):1774–1797, 2009.

[93] A. Fasoli, S. Brunner, W. Cooper, J. Graves, P. Ricci, O. Sauter, and L. Villard. Computational challenges in magnetic-confinement fusion physics. *Nature Physics*, 12(5):411–423, 2016.

[94] E. Fernández-Juricic, J. T. Erichsen, and A. Kacelnik. Visual perception and social foraging in birds. *Trends in Ecology & Evolution*, 19(1):25–31, 2004.

[95] E. Fernández-Juricic, S. Siller, and A. Kacelnik. Flock density, social foraging, and scanning: an experiment with starlings. *Behavioral ecology*, 15(3):371–379, 2004.

[96] F. Filbet and L. M. Rodrigues. Asymptotically stable Particle-In-Cell methods for the Vlasov–Poisson system with a strong external magnetic field. *SIAM Journal on Numerical Analysis*, 54(2):1120–1146, 2016.

[97] F. Filbet and L. M. Rodrigues. Asymptotically preserving Particle-In-Cell methods for inhomogeneous strongly magnetized plasmas. *SIAM Journal on Numerical Analysis*, 55(5):2416–2443, 2017.

[98] F. Filbet and E. Sonnendrücker. Numerical methods for the Vlasov equation. In *Numerical Mathematics and Advanced Applications: Proceedings of ENUMATH 2001 the 4th European Conference on Numerical Mathematics and Advanced Applications Ischia, July 2001*, pages 459–468. 2003.

[99] F. Filbet and T. Xiong. Conservative discontinuous galerkin/hermite spectral method for the Vlasov–Poisson system. *Communications on Applied Mathematics and Computation*, pages 1–26, 2020.

[100] F. Filbet and C. Yang. Numerical simulations to the Vlasov-Poisson system with a strong magnetic field. *HAL*, 2018.

[101] M. B. Flegg, S. Hellander, and R. Erban. Convergence of methods for coupling of microscopic and mesoscopic reaction–diffusion simulations. *Journal of computational physics*, 289:1–17, 2015.

[102] M. Fornasier, J. Haskovec, and G. Toscani. Fluid dynamic description of flocking via the Povzner–Boltzmann equation. *Physica D: Nonlinear Phenomena*, 240(1):21–31, 2011.

[103] M. Fornasier, H. Huang, L. Pareschi, and P. Sünnen. Consensus-based optimization on hypersurfaces: Well-posedness and mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 30(14):2725–2751, 2020.

[104] M. Fornasier, H. Huang, L. Pareschi, and P. Sünnen. Consensus-based optimization on the sphere: Convergence to global minimizers and machine learning. *The Journal of Machine Learning Research*, 22(1):10722–10776, 2021.

[105] M. Fornasier and F. Solombrino. Mean-field optimal control. *ESAIM: Control, Optimisation and Calculus of Variations*, 20(4):1123–1152, 2014.

[106] J. Friedman and J. Gore. Ecological systems biology: The dynamics of interacting populations. *Current Opinion in Systems Biology*, 1:114–121, 2017.

[107] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

[108] P. R. Garabedian. Computational mathematics and physics of fusion reactors. *Proceedings of the National Academy of Sciences*, 100(24):13741–13745, 2003.

[109] J. Garnier, G. Papanicolaou, and T.-W. Yang. Consensus convergence with stochastic effects. *Vietnam Journal of Mathematics*, 45:51–75, 2017.

[110] A. Ghizzo, P. Bertrand, M. Shoucri, E. Fijalkow, and M. R. Feix. An Eulerian code for the study of the drift-kinetic Vlasov equation. *Journal of Computational Physics*, 108(1):105–121, 1993.

[111] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434, 1976.

[112] D. T. Gillespie. The multivariate Langevin and Fokker–Planck equations. *American Journal of Physics*, 64(10):1246–1257, 1996.

[113] D. T. Gillespie. The chemical Langevin equation. *The Journal of Chemical Physics*, 113(1):297–306, 2000.

[114] D. T. Gillespie. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*, 58:35–55, 2007.

[115] D. E. Golberg. *Genetic algorithms in search, optimization, and machine learning*, volume 1989. 1989.

[116] V. Grandgirard and Y. Sarazin. Gyrokinetic simulations of magnetic fusion plasmas. pages 91–176. 2013.

[117] S. Grassi and L. Pareschi. From particle swarm optimization to consensus based optimization: stochastic modeling and mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 31(08):1625–1657, 2021.

[118] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of computational physics*, 73(2):325–348, 1987.

[119] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten. Generalized verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Molecular Simulation*, 6(1-3):121–142, 1991.

[120] A. Gu, Y. He, and Y. Sun. Hamiltonian Particle-In-Cell methods for Vlasov–Poisson equations. *Journal of Computational Physics*, 467:111472, 2022.

[121] E. Hairer and C. Lubich. Energy behaviour of the Boris method for charged-particle dynamics. *BIT Numerical Mathematics*, 58:969–979, 2018.

[122] D. Han-Kwan. On the confinement of a tokamak plasma. *SIAM journal on mathematical analysis*, 42(6):2337–2367, 2010.

[123] J. Haskovec. Flocking dynamics and mean-field limit in the Cucker–Smale-type model with topological interactions. *Physica D: Nonlinear Phenomena*, 261:42–51, 2013.

[124] B. Hauer, J. Maciejko, and J. Davis. Nonlinear power spectral densities for the harmonic oscillator. *Annals of Physics*, 361:148–183, 2015.

[125] A. Hellander and P. Lötstedt. Hybrid method for the chemical master equation. *Journal of Computational Physics*, 227(1):100–122, 2007.

[126] S. Hellander, A. Hellander, and L. Petzold. Mesoscopic-microscopic spatial stochastic simulation with automatic system partitioning. *The Journal of chemical physics*, 147(23):234101.

[127] C. K. Hemelrijk and H. Hildenbrandt. Self-organized shape and frontal density of fish schools. *Ethology*, 114(3):245–254, 2008.

[128] W. Hundsdorfer and J. G. Verwer. *Numerical solution of time-dependent advection-diffusion-reaction equations*, volume 33. Springer Science & Business Media, 2013.

[129] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control*, 48(6):988–1001, 2003.

[130] M. Jamil and X.-S. Yang. A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194, 2013.

[131] S. Jin, L. Li, and J.-G. Liu. Random batch methods (RBM) for interacting particle systems. *Journal of Computational Physics*, 400:108877, 2020.

[132] A. Jüngel and Y.-J. Peng. A hierarchy of hydrodynamic models for plasmas. quasi-neutral limits in the drift-diffusion equations. *Asymptotic Analysis*, 28(1):49–73, 2001.

[133] D. Kalise, A. Sharma, and M. V. Tretyakov. Consensus based optimization via jump-diffusion stochastic differential equations. *arXiv preprint arXiv:2205.04880*, 2022.

[134] D. Kalise, A. Sharma, and M. V. Tretyakov. Consensus-based optimization via jump-diffusion stochastic differential equations. *Math. Models Methods Appl. Sci.*, 33(2):289–339, 2023.

[135] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

[136] A. J. King, S. J. Portugal, D. Strömbom, R. P. Mann, J. A. Carrillo, D. Kalise, G. de Croon, H. Barnett, P. Scerri, R. Groß, et al. Biologically inspired herding of animal groups by robots. *Methods in Ecology and Evolution*, 14(2):478–486, 2023.

[137] K. Klamroth, M. Stiglmayr, and C. Totzeck. Consensus-based optimization for multi-objective problems: A multi-swarm approach. *arXiv preprint*, 2022.

[138] P. Knopf and J. Weber. Optimal control of a Vlasov–Poisson plasma by fixed magnetic field coils. *Applied Mathematics & Optimization*, 81:961–988, 2020.

[139] A. L. Koch and D. White. The social lifestyle of myxobacteria. *Bioessays*, 20(12):1030–1038, 1998.

[140] T. R. Krogstad. Attitude synchronization in spacecraft formations: Theoretical and experimental results. 2010.

[141] J. M. Lee, S. H. Cho, and R. A. Calvo. A fast algorithm for simulation of flocking behavior. In *2009 International IEEE Consumer Electronics Society's Games Innovations Conference*, pages 186–190. IEEE, 2009.

[142] Z. Li and S.-Y. Ha. On the Cucker-Smale flocking with alternating leaders. *Quarterly of Applied Mathematics*, 73(4):693–709, 2015.

[143] N. Loy and A. Tosin. Boltzmann-type equations for multi-agent systems with label switching. *Kinetic & Related Models*, 14, 2021.

[144] N. Loy and A. Tosin. Boltzmann-type equations for multi-agent systems with label switching. *Kinetic and Related Models*, 14(5):867–894, 2021.

[145] R. Lukeman, Y.-X. Li, and L. Edelstein-Keshet. Inferring individual rules from collective behavior. *Proceedings of the National Academy of Sciences*, 107(28):12576–12580, 2010.

[146] L. Marchetti, C. Priami, and V. H. Thanh. *Simulation algorithms for computational systems biology*. Springer, 2017.

[147] The Mathworks, Inc., Natick, Massachusetts. *MATLAB version 9.3.0.713579 (R2017b)*, 2017.

[148] S. J. Maynard. Models in ecology. *Cambridge UP, Cambridge*, 1974.

[149] A. J. McKane and T. J. Newman. Stochastic models in population biology and their deterministic analogs. *Physical Review E*, 70(4):041902, 2004.

[150] A. J. McKane and T. J. Newman. Predator-prey cycles from resonant amplification of demographic stochasticity. *Physical review letters*, 94(21):218102, 2005.

[151] A. Medaglia, L. Pareschi, and M. Zanella. Stochastic Galerkin particle methods for kinetic equations of plasmas with uncertainties. *Journal of Computational Physics*, 479:112011, 2023.

[152] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1996.

[153] M. Mitchell. Genetic algorithms: An overview. *Complexity*, 1(1):31–39, 1995.

[154] M. Morandotti and F. Solombrino. Mean-field analysis of multipopulation dynamics with label switching. *SIAM Journal on Mathematical Analysis*, 52(2):1427–1462, 2020.

[155] S. Motsch and E. Tadmor. Heterophilious dynamics enhances consensus. *SIAM review*, 56(4):577–621, 2014.

[156] J. D. Murray. *Mathematical biology: I. An introduction*, volume 17. Springer Science & Business Media, 2007.

[157] K. Nanbu. Direct simulation scheme derived from the Boltzmann equation. i. monocomponent gases. *Journal of the Physical Society of Japan*, 49(5):2042–2049, 1980.

[158] K. Nanbu. Theoretical basis of the direct simulation Monte Carlo method. In *15th International Symposium on Rarefied Gas Dynamics*, volume 1, pages 369–383, 1986.

[159] R. M. Nisbet and W. Gurney. *Modelling fluctuating populations: reprint of first Edition (1982)*. 2003.

[160] L. Pareschi and G. Russo. An introduction to Monte Carlo method for the Boltzmann equation. In *ESAIM: Proceedings*, volume 10, pages 35–75. EDP Sciences, 2001.

[161] L. Pareschi and G. Russo. Time relaxed Monte Carlo methods for the boltzmann equation. *SIAM Journal on Scientific Computing*, 23(4):1253–1273, 2001.

[162] L. Pareschi and G. Toscani. *Interacting multiagent systems: kinetic equations and Monte Carlo methods*. OUP Oxford, 2013.

[163] J. K. Parrish and L. Edelstein-Keshet. Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science*, 284(5411):99–101, 1999.

[164] E. C. Pielou et al. An introduction to mathematical ecology. *An introduction to mathematical ecology.*, 1969.

[165] R. Pinnau, C. Totzeck, O. Tse, and S. Martin. A consensus-based model for global optimization and its mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 27(01):183–204, 2017.

[166] R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization: An overview. *Swarm intelligence*, 1:33–57, 2007.

[167] A. Y. Povzner. On the Boltzmann equation in the kinetic theory of gases. *Matematicheskii Sbornik*, 100(1):65–86, 1962.

[168] H. Rainer and U. Krause. Opinion dynamics and bounded confidence: models, analysis and simulation. 2002.

[169] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie. Consistency and stability of tau-leaping schemes for chemical reaction systems. *Multiscale Modeling & Simulation*, 4(3):867–895, 2005.

[170] S. Ravindran. Real-time computational algorithm for optimal control of an mhd flow system. *SIAM Journal on Scientific Computing*, 26(4):1369–1388, 2005.

[171] E. Renshaw. *Modelling biological populations in space and time*, volume 11. Cambridge University Press, 1993.

[172] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987.

[173] T. Roose, S. J. Chapman, and P. K. Maini. Mathematical models of avascular tumor growth. *SIAM review*, 49(2):179–208, 2007.

[174] M. S. Rosin, L. F. Ricketson, A. M. Dimits, R. E. Caflisch, and B. I. Cohen. Multilevel Monte Carlo simulation of coulomb collisions. *Journal of Computational Physics*, 274:140–157, 2014.

[175] G. Russo and F. Filbet. Semi-Lagrangian schemes applied to moving boundary problems for the bgk model of rarefied gas dynamics. *Kinetic and related models*, 2(1):231–250, 2009.

[176] W. Sandmann. Streamlined formulation of adaptive explicit-implicit τ-leaping with automatic tau selection. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 1104–1112. IEEE, 2009.

[177] L. J. Schumacher, P. M. Kulesa, R. McLennan, R. E. Baker, and P. K. Maini. Multidisciplinary approaches to understanding collective cell migration in developmental biology. *Open biology*, 6(6):160056, 2016.

[178] M. Sehl, A. V. Alekseyenko, and K. L. Lange. Accurate stochastic simulation via the step anticipation τ-leaping (sal) algorithm. *Journal of Computational Biology*, 16(9):1195–1208, 2009.

[179] N. Shigesada. Spatial distribution of dispersing animals. *Journal of mathematical biology*, 9(1):85–96, 1980.

[180] E. Sonnendrücker and K. Kormann. Numerical methods for vlasov equations. *Lecture notes*, 2013.

[181] E. Sonnendrücker, J. Roche, P. Bertrand, and A. Ghizzo. The semi-Lagrangian method for the numerical resolution of the Vlasov equation. *Journal of computational physics*, 149(2):201–220, 1999.

[182] F. Spill, P. Guerrero, T. Alarcon, P. K. Maini, and H. Byrne. Hybrid approaches for multiple-species stochastic reaction–diffusion models. *Journal of computational physics*, 299:429–445, 2015.

[183] C. F. M. Toledo, L. Oliveira, and P. M. França. Global optimization using a genetic algorithm with hierarchically structured population. *Journal of Computational and Applied Mathematics*, 261:341–351, 2014.

[184] R. Toral and P. Colet. *Stochastic numerical methods: an introduction for students and scientists*. John Wiley & Sons, 2014.

[185] G. Toscani. Kinetic models of opinion formation. *Communications in mathematical sciences*, 4(3):481–496, 2006.

[186] C. Totzeck. Trends in consensus-based optimization. In *Active Particles, Volume 3: Advances in Theory, Models, and Applications*, pages 201–226. Springer, 2021.

[187] C. Totzeck and M.-T. Wolfram. Consensus-based global optimization with personal best. *Mathematical Biosciences and Engineering*, 17(5):6026–6044, 2020.

[188] N. G. Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.

[189] J.-L. Vay. Simulation of beams or plasmas crossing at relativistic velocity. *Physics of Plasmas*, 15(5), 2008.

[190] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical review letters*, 75(6):1226, 1995.

[191] M. Vishwakarma, J. Di Russo, D. Probst, U. S. Schwarz, T. Das, and J. P. Spatz. Mechanical interactions among followers determine the emergence of leaders in migrating epithelial cell collectives. *Nature communications*, 9(1):3469, 2018.

[192] G. Webb. A reaction-diffusion model for a deterministic diffusive epidemic. *Journal of Mathematical Analysis and Applications*, 84(1):150–161, 1981.

[193] J. Weber. Optimal control of the two-dimensional Vlasov-Maxwell system. *ESAIM: Control, Optimisation and Calculus of Variations*, 27:S19, 2021.

[194] C. Yang and F. Filbet. Conservative and non-conservative methods based on hermite weighted essentially non-oscillatory reconstruction for Vlasov equations. *Journal of Computational Physics*, 279:18–36, 2014.