



HAL
open science

Mean Field Games of Controls: Finite Difference Approximations

Y Achdou, Z Kobeissi

► **To cite this version:**

Y Achdou, Z Kobeissi. Mean Field Games of Controls: Finite Difference Approximations. 2020.
hal-02501366

HAL Id: hal-02501366

<https://hal.science/hal-02501366>

Preprint submitted on 6 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mean Field Games of Controls: Finite Difference Approximations

Y. Achdou^{*}; Z. Kobeissi[†]

March 6, 2020

Abstract

We consider a class of mean field games in which the agents interact through both their states and controls, and we focus on situations in which a generic agent tries to adjust her speed (control) to an average speed (the average is made in a neighborhood in the state space). In such cases, the monotonicity assumptions that are frequently made in the theory of mean field games do not hold, and uniqueness cannot be expected in general.

Such model lead to systems of forward-backward nonlinear nonlocal parabolic equations; the latter are supplemented with various kinds of boundary conditions, in particular Neumann-like boundary conditions stemming from reflection conditions on the underlying controlled stochastic processes.

The present work deals with numerical approximations of the above mentioned systems. After describing the finite difference scheme, we propose an iterative method for solving the systems of nonlinear equations that arise in the discrete setting; it combines a continuation method, Newton iterations and inner loops of a bigradient like solver.

The numerical method is used for simulating two examples. We also make experiments on the behaviour of the iterative algorithm when the parameters of the model vary.

1 - INTRODUCTION

The theory of mean field games, (*MFGs* for short), aims at studying deterministic or stochastic differential games (Nash equilibria) as the number of agents tends to infinity. It supposes that the rational agents are indistinguishable and individually have a negligible influence on the game, and that each individual strategy is influenced by some averages of quantities depending on the states (or the controls as in the present work) of the other agents. MFGs have been introduced in the pioneering works of J-M. Lasry and P-L. Lions [17, 18, 19]. Independently and at approximately the same time, the notion of mean field games arose in the engineering literature, see the works of M.Y. Huang, P.E. Caines and R.Malhamé [14, 15].

The present work deals with numerical approximations of mean field games in which the agents interact through both their states and controls; it follows a more theoretical work by the second author, [16], which is devoted to the mathematical analysis of the related systems of nonlocal partial differential equations. There is not much literature on MFGs in which the agents also interact through their controls, see [13, 12, 8, 10, 7, 16]. To stress the fact that the latter situation is considered, we will sometimes use the terminology *mean field games of control* and the acronym *MFGC*.

^{*}Univ. Paris Diderot, Sorbonne Paris Cité, Laboratoire Jacques-Louis Lions, UMR 7598, UPMC, CNRS, F-75205 Paris, France. achdou@ljl.univ-paris-diderot.fr

[†]Univ. Paris Diderot, Sorbonne Paris Cité, Laboratoire Jacques-Louis Lions, UMR 7598, UPMC, CNRS, F-75205 Paris, France. zkobeissi@math.univ-paris-diderot.fr

The MFGC that is considered in the present work is described by the following system of nonlocal partial differential equations:

$$(1.1) \quad \begin{cases} -\partial_t u - \nu \Delta u + H(x, \nabla_x u(t, x), \mu(t)) = f(x, m(t)) & \text{in } [0, T] \times \Omega, \\ \partial_t m - \nu \Delta m - \operatorname{div}(H_p(x, \nabla_x u(t, x), \mu(t)) m) = 0 & \text{in } (0, T] \times \Omega, \\ \mu(t) = \left(I_d, -H_p(\cdot, \nabla_x u(t, \cdot), \mu(t)) \right) \# m(t) & \text{in } [0, T], \\ u(T, x) = \phi(x) & \text{in } \Omega, \\ m(0, x) = m_0(x) & \text{in } \Omega. \end{cases}$$

Here, the state space Ω is a bounded domain of \mathbb{R}^d with a piecewise smooth boundary, while the controls are vectors of \mathbb{R}^d . The time horizon is $T > 0$, and the parameter $\nu > 0$ is linked to the level of noise in the trajectories (the dynamics of a given agent is described by $dX_t = \alpha_t dt + \sqrt{2\nu} dB_t$ where (B_t) is a standard d -dimensional Brownian motion and $\alpha_t \in \mathbb{R}^d$ is the control at time t). A special feature of the present model is that the third argument of the Hamiltonian H is a probability measure $\mu(t) \in \mathcal{P}(\Omega \times \mathbb{R}^d)$, which stands for the joint probability of the states and optimal controls of the agents at time t . In (1.1), $u : [0, T] \times \Omega \rightarrow \mathbb{R}$ and $m : [0, T] \times \Omega \rightarrow \mathbb{R}_+$ respectively stand for the value function of a representative agent and the distribution of states. The first, respectively second line in (1.1) is the Hamilton-Jacobi-Bellman equation (*HJB* for short) which leads to the optimal control of a representative agent, respectively the Fokker-Planck-Kolmogorov equation (*FP* for short) which describes the transport-diffusion of the distribution of states by the optimal control law. The HJB equation is a backward w.r.t. time parabolic equation and is supplemented with a condition at $t = T$ which involves the terminal cost function $\phi : \Omega \rightarrow \mathbb{R}$, whereas the FP equation is forward w.r.t. time and is supplemented with an initial condition, which translates the fact that the initial distribution of states is known. The HJB equation also involves the so-called *coupling* function $f : \Omega \times \mathbb{R}_+ \rightarrow \mathbb{R}$, or, in other words, $f(X_t, m(t, X(t)))$ is part of the running cost of a representative agent. We shall discuss later the boundary conditions on $(0, T) \times \partial\Omega$ associated with the first two equations in (1.1). The third equation in (1.1) gives the connection between $\mu(t)$ and $m(t)$, and shows in particular that $\mu(t)$ is supported by a d -dimensional geometrical object, namely the graph of the feedback law: $\Omega \rightarrow \mathbb{R}^d, x \mapsto -H_p(x, \nabla_x u(t, x), \mu(t))$.

1.1 A brief discussion on the mathematical analysis of (1.1)

Recall that the Hamiltonian of the problem is $(x, p, \mu) \mapsto H(x, p, \mu)$, $(x, p, \mu) \in \Omega \times \mathbb{R}^d \times \mathcal{P}(\Omega \times \mathbb{R}^d)$.

From the viewpoint of mathematical analysis, a priori estimates for (1.1) are more difficult to obtain than in the case when the agents interact only via the distribution of states m . Indeed, in the latter case, if for example the costs f and ϕ are uniformly bounded, then a priori estimates on $\|u\|_\infty$ stem from the maximum principle for second-order parabolic equations. By contrast, since the Hamiltonian in (1.1) depends non-locally on $\nabla_x u$, the maximum principle applied to the HJB equation only permits to bound $\|u\|_\infty$ by a quantity which depends (quadratically under standard assumptions on H) on $\|\nabla_x u\|_\infty$, and this information may be useless without additional arguments.

If the agents interact only through the distribution of states and if the Hamiltonian depends separately on p and m , a natural assumption is that the latter is monotone with respect to m , see [17, 18, 19]; it implies existence and uniqueness of solutions, see in [20, 19]. Such an assumption is quite sensible in many situations, since it models the aversion of the agents to highly crowded regions of the state space. It is possible to extend these arguments to MFGCs, see [10] for a probabilistic point of view and [8] for a PDE point of view, and the monotonicity assumption then means that the agents favor controls that are opposite to the main stream. In [16] and in the present work, we prefer to avoid such an assumption, because it is generally not satisfied, at least in models of crowd motion: indeed, in models of traffic or pedestrian flows, a generic agent would rather try to adjust her speed (control) to the average speed in a neighborhood of her position.

The third equation in (1.1) can be seen as a fixed point problem for μ given u and m , which turns to be well-posed under the Lasry-Lions monotonicity assumption adapted to MFGC, provided that u and m are smooth enough. We shall replace this assumption by a new structural condition which has been introduced in [16], namely that H_p depends linearly on the variable μ and is a contraction with respect to μ (using a suitable distance on probability measures). In the context of crowd motion, this structural condition is satisfied if the representative agent targets controls that are proportional to an average of the controls chosen by the other agents nearby, with a positive proportionality coefficient smaller than one. Were this coefficient equal to or larger than one, it would be easy to cook up examples in which there is no solution to (1.1) or even to the N -agent game, see Remark 4.3 below.

In [16], the focus is put on the existence of solutions rather than on their uniqueness. Indeed, without the monotonicity condition, uniqueness is unlikely in general if T is not small. Consider for example a game in which the function ϕ has two perfectly symmetrical minima (the targets), $f = 0$ and where the initial distribution of states has the same symmetry. If H depends on $|\nabla u(t, x)|$ only (no interaction through the controls), then a representative agent will simply travel to the minimum which is closest to her initial position. On the contrary, if the representative agent favors a control close to the average one, then there are at least two symmetrical solutions in which the whole distribution moves toward one of the two minima.

Going back to existence results, it is now frequent in the MFG literature to obtain energy estimates by testing the HJB equation by m , the FP equation by u , summing the resulting equations, integrating in the time and state variables then making suitable integrations by parts. If the value function is uniformly bounded from below (which is often the case even if there are interactions through the controls), this results in a relationship between the $L^\infty([0, T]; L^1(\Omega))$ -norm of the positive part of u and the $L_m^2([0, T] \times \Omega; \mathbb{R}^d)$ -norm of $\nabla_x u$; this observation can then be used to obtain additional a priori estimates, which may be combined with the ones obtained from the maximum principle and discussed above, and finally with Bernstein method. This strategy has been implemented in [16] and leads to the existence of a solution to (1.1) under suitable assumptions. By and large, existence was proved in [16] for periodic problems in any of the following cases:

- short time horizon
- monotonicity
- small enough parameters (in particular the contraction factor mentioned above, see the parameters λ and θ in (1.7)-(1.10) below)
- weak dependency of the average drift on the state variable

Note that in [11], existence and uniqueness have been proved with probabilistic arguments in the case where the Hamiltonian depends separately on p and μ .

1.2 A more detailed description of the considered class of MFGCs

For any $x \in \partial\Omega$, let $n(x)$ be the outward pointing unit normal vector to $\partial\Omega$ at x . The dynamics of a representative agent is given by

$$(1.2) \quad dX_t = \alpha_t dt + \sqrt{2\nu} dB_t - 2\nu n(X_t) dL_t, \quad X_t \in \bar{\Omega}, \quad 0 \leq t \leq T,$$

where $(B_t)_{t \in [0, T]}$ is a standard d -dimensional Brownian motion, $(\alpha_t)_{t \in [0, T]}$ is the control, a stochastic process adapted to (B_t) with values in \mathbb{R}^d , L_t is the local time of the process X_t on $\partial\Omega$. We assume that $X_{|t=0}$ is a random variable in $\bar{\Omega}$, independent of (B_t) for all $t > 0$, whose law is $\mathcal{L}(X_{|t=0}) = m_0$. In what follows, the part of the running cost which models the interactions via the controls will involve an average drift $V \in \mathbb{R}^d$:

$$(1.3) \quad V(t, x) = \frac{1}{Z(t, x)} \int_{(y, \alpha) \in \Omega \times \mathbb{R}^d} \alpha K(x, y) d\mu(t, y, \alpha),$$

for $(t, x) \in [0, T] \times \Omega$, where K is a nonnegative kernel, $Z(t, x) = \int_{\Omega} K(x, y) dm(t, y)$ is a normalization factor, $\mu(t, \cdot, \cdot)$ is the law of the joint distribution of the states and the controls, and $m(t, \cdot)$ is the law of the distribution of states.

Hereafter, we are going to focus in MFGCs in which the cost to be minimized by a representative agent is

$$(1.4) \quad J(\alpha) = \mathbb{E} \left[\int_0^T \left(\frac{\theta}{2} |\alpha_t - \lambda V(t, X_t)|^2 + \frac{1-\theta}{2} |\alpha_t|^2 + cm(t, X_t) + f_0(X_t) \right) dt + \phi(X_T) \right],$$

where λ, θ are real numbers such that $-1 < \lambda < 1$ and $0 < \theta \leq 1$. This leads us to define the Lagrangian

$$(1.5) \quad L(\alpha, V) = \frac{\theta}{2} |\alpha - \lambda V|^2 + \frac{1-\theta}{2} |\alpha|^2, \quad (\alpha, V) \in \mathbb{R}^d \times \mathbb{R}^d,$$

which is convex with respect to α , and its Legendre transform (with respect to α):

$$(1.6) \quad H(p, V) = \frac{1}{2} |p - \lambda \theta V|^2 - \frac{\lambda^2 \theta}{2} |V|^2, \quad (p, V) \in \mathbb{R}^d \times \mathbb{R}^d.$$

With this definition of the running cost, the first three lines of (1.1) can be written as follows:

$$(1.7) \quad -\partial_t u - \nu \Delta u + \frac{1}{2} |\nabla_x u - \lambda \theta V|^2 - \frac{\lambda^2 \theta}{2} |V|^2 = cm + f_0(x), \quad \text{in } [0, T] \times \Omega,$$

$$(1.8) \quad \partial_t m - \nu \Delta m - \operatorname{div}((\nabla_x u - \lambda \theta V) m) = 0, \quad \text{in } (0, T] \times \Omega,$$

$$(1.9) \quad V(t, x) = -\frac{1}{Z(t, x)} \int_{\Omega} (\nabla_x u(t, y) - \lambda \theta V(t, y)) K(x, y) dm(t, y), \quad \text{in } (0, T] \times \Omega,$$

$$(1.10) \quad Z(t, x) = \int_{\Omega} K(x, y) dm(t, y), \quad \text{in } [0, T] \times \Omega,$$

$$(1.11) \quad u(T, x) = \phi(x), \quad \text{in } \Omega,$$

$$(1.12) \quad m(0, x) = m_0(x), \quad \text{in } \Omega.$$

We can now specify the boundary conditions on $(0, T) \times \partial\Omega$. First, assuming that u is smooth enough, the optimal control of a representative player is given by the feedback law:

$$\alpha^*(t, x) = -\left(\nabla_x u(t, x) - \lambda \theta V(t, x) \right).$$

The reflection condition at the boundary translates into the Neumann boundary conditions:

$$(1.13) \quad \frac{\partial u}{\partial n}(t, x) = 0, \quad \text{on } [0, T] \times \partial\Omega.$$

Now, from the definition of m , $\int_{\Omega} m(t, x) \psi(x) dx = \mathbb{E} \left(\int_{\Omega} m_0(x) \psi(X_{t,x}) dx \right)$ for all smooth enough test-function ψ such that $\frac{\partial \psi}{\partial n}(x) = 0$ on $\partial\Omega$. Taking the time derivative of the latter equality, and using (1.13), we deduce that

$$(1.14) \quad \nu \frac{\partial m}{\partial n}(t, x) - \lambda \theta m(t, x) V(t, x) \cdot n(x) = 0, \quad \text{on } (0, T] \times \partial\Omega.$$

Note that (1.8), (1.13) and (1.14) imply that the total mass $\int_{\Omega} m(t, x) dx$ is conserved.

We have studied or will sometimes consider other kinds of boundary conditions on some parts of the boundary, for example:

- periodic conditions, i.e. we set (1.7)-(1.12)) in the flat torus $\Omega = \mathbb{R}^d/\mathbb{Z}^d$
- At a part of the boundary standing for an exit, there is an exit cost. This yields a Dirichlet condition on u . The Dirichlet condition on m : $m = 0$ at exits, means that the agents stop taking part in the mean field game as soon as they reach the exit. Such boundary conditions will be simulated numerically in paragraph 4.2 below.
- If a part of the boundary stands for an entrance, then it is natural to impose a Dirichlet condition on u to prevent the agents from crossing the entrance outward, and a flux-like condition on m to specify the entering flux of agents, see also paragraph 4.2 below.

1.3 Organization of the paper

Section 2 is devoted to the description of the finite difference scheme; it is based on a monotone upwind scheme for the HJB equation, and a scheme for the FP equation which is obtained by differentiating the discrete HJB equation and taking the adjoint. It is an extension of the finite difference schemes proposed and studied by the first author and I. Capuzzo-Dolcetta in [3, 2] for simpler MFGs. Designing an efficient strategy for solving the system of non linear equations arising in the discrete version of MFGCs is a challenging task, in particular because

- the underlying system of PDEs is forward-backward and cannot be solved by simply marching in time
- there is no underlying variational structure
- Equation (1.9) is non local.

In Section 3, we propose a strategy for solving the system of non linear equations: it is a continuation method in which the viscosity parameter (for instance) is progressively decreased to the desired value, and for each value of the latter parameter, the system of non linear equations is solved by a Newton method with inner iterations of a bi-gradient like algorithm.

Finally, in Section 4, we discuss the results of numerical simulations in two cases. In the first example, we show in particular that there exist multiple solutions and we discuss how the iteration count of the solver is affected by the variation of the parameters in the model. The second case is a model for a crowd of agents crossing a rectangular hall from an entrance to an exit: we consider situations in which queues occur, and we show how the solution depends on the parameters.

2 - FINITE DIFFERENCE METHODS

In order to approximate (1.7)-(1.12) numerically, we propose a finite difference method reminiscent of that introduced in [3, 2] for MFGs without coupling through the controls. The important features of such methods are as follows

- they are based on monotone finite difference schemes for (1.7). Hence, a comparison principle still holds at the discrete level.
- the special structure of (1.7)-(1.8) is preserved at the discrete level, namely that the FP equation can be obtained by differentiating the HJB equation w.r.t. u and by taking the adjoint of the resulting equation. This results in a monotone approximation of (1.8), which ensures that the discrete version of m remains non negative at all time if m_0 is non negative. The discrete FP equation will also preserve mass.

To simplify the discussion, let us focus on the case when $d = 2$ and $\Omega = (0, 1)^2$. More complex domains (even with holes) can be handled by the present method, but an additional effort would be needed if the domain had boundaries not aligned with the axes of the underlying grid that will be introduced soon. We also assume for simplicity that the boundary conditions are of the type (1.13)-(1.14) on the whole $\partial\Omega$.

2.1 Notations and definitions

For two positive integers N_T, N_h , we set $\Delta t = T/N_T$, the time step, and $h = 1/N_h$, the step related to the state variables. Consider the set of discrete times $\mathcal{T}_{\Delta t} = \{t_k = k\Delta t, k = 0, \dots, N_T\}$ and the grid $\Omega_h = \{x_{i,j} = (ih, jh), i, j = 0, \dots, N_h\}$.

The goal will be to approximate $u(t_k, x_{i,j})$ and $m(t_k, x_{i,j})$ respectively by $u_{i,j}^k$ and $m_{i,j}^k$, for all $k \in \{0, \dots, N_T\}$ and $(i, j) \in \{0, \dots, N_h\}^2$, by solving the discrete version of (1.7)-(1.14) arising from the finite difference scheme.

Let us define the discrete time derivative of $y : \mathcal{T}_{\Delta t} \times \Omega_h \rightarrow \mathbb{R}$ by the collection of real numbers:

$$(2.1) \quad D_t y_{i,j}^k = \frac{y_{i,j}^{k+1} - y_{i,j}^k}{\Delta t},$$

for $k = 0, \dots, N_T - 1, i, j = 0, \dots, N_h$.

Given a grid function $y : \Omega_h \rightarrow \mathbb{R}$, we introduce the first order right sided difference operators

$$(2.2) \quad \begin{aligned} (D_1^+ y)_{i,j} &= \frac{y_{i+1,j} - y_{i,j}}{h}, \\ (D_2^+ y)_{i,j} &= \frac{y_{i,j+1} - y_{i,j}}{h}, \end{aligned}$$

and the five point discrete laplace operator:

$$(2.3) \quad (\Delta_h y)_{i,j} = -\frac{1}{h^2} (4y_{i,j} - y_{i+1,j} - y_{i-1,j} - y_{i,j+1} - y_{i,j-1}),$$

for all (i, j) such that $0 \leq i, j \leq N_h$. Note that the definition of these operators at boundary nodes of Ω_h needs to extend the grid function y on a layer of nodes outside Ω_h . This is done by using discrete versions of the Neumann conditions (1.13)- (1.14): assume that the boundary condition for y is of the type $\frac{\partial y}{\partial n} = z$, where z is a given continuous function. Let $(z_{i,j}^k)$ is a suitable grid function approximating z . We then choose the discrete version of the latter equation (first order scheme):

$$(2.4) \quad \begin{aligned} y_{-1,j}^k &= y_{0,j}^k + h z_{0,j}^k & \text{and} & & y_{N_h+1,j}^k &= y_{N_h,j}^k + h z_{N_h,j}^k, \\ y_{i,-1}^k &= y_{i,0}^k + h z_{i,0}^k & \text{and} & & y_{i,N_h+1}^k &= y_{i,N_h}^k + h z_{i,N_h}^k, \end{aligned}$$

for $i, j = 0, \dots, N_h$ and $k = 0, \dots, N_T$.

Remark 2.1. *We have chosen a first order scheme for the boundary condition in order to preserve the monotonicity of the discrete Hamiltonian at boundary nodes, see later. Since the overall scheme is monotone as it was already mentioned above, thus first order, it would be pointless to choose a higher order scheme for the boundary conditions.*

Remark 2.2. *Note that the previously mentioned discrete operators are not needed at the nodes of $\partial\Omega_h$ at which Dirichlet boundary conditions are imposed.*

For a grid function $V : \Omega_h \rightarrow \mathbb{R}$, we define the value of V at half-integer indices by linear interpolation:

$$(2.5) \quad \begin{aligned} V_{i+\frac{1}{2},j} &= \frac{V_{i+1,j} + V_{i,j}}{2}, & V_{-\frac{1}{2},j} &= V_{0,j}, \text{ and } V_{N_h+\frac{1}{2},j} = V_{N_h,j}, & 0 \leq i \leq N_h - 1, 0 \leq j \leq N_h, \\ V_{i,j+\frac{1}{2}} &= \frac{V_{i,j+1} + V_{i,j}}{2}, & V_{i,-\frac{1}{2}} &= V_{i,0}, \text{ and } V_{i,N_h+\frac{1}{2}} = V_{i,N_h}, & 0 \leq i \leq N_h, 0 \leq j \leq N_h - 1. \end{aligned}$$

In order to define the Godunov scheme for (1.7), we introduce the map Φ : for two grid functions $y : \Omega_h \rightarrow \mathbb{R}$ and $V : \Omega_h \rightarrow \mathbb{R}^2$, the grid function $\Phi(y, V) : \Omega_h \rightarrow \mathbb{R}^4$ is defined by

$$(2.6) \quad \Phi(y, V)_{i,j} = \left\{ \begin{array}{l} \left[(D_1^+ y)_{i,j} - \lambda \theta V_{i+\frac{1}{2},j,1} \right]_- \\ - \left[(D_1^+ y)_{i-1,j} - \lambda \theta V_{i-\frac{1}{2},j,1} \right]_+ \\ \left[(D_2^+ y)_{i,j} - \lambda \theta V_{i,j+\frac{1}{2},2} \right]_- \\ - \left[(D_2^+ y)_{i,j-1} - \lambda \theta V_{i,j-\frac{1}{2},2} \right]_+ \end{array} \right\} \quad \text{for } i, j = 0, \dots, N_h.$$

Finally, let us introduce the function $g : \mathbb{R}^4 \times \mathbb{R}^2 \rightarrow \mathbb{R}$:

$$(2.7) \quad g(q, V) = \frac{1}{2} |q|^2 - \frac{\lambda^2 \theta}{2} |V|^2.$$

2.2 The scheme

With the ingredients defined in paragraph 2.1, we are ready to propose the discrete version of (1.7)-(1.10):

$$(2.8) \quad -D_t u_{i,j}^k - \nu \left(\Delta_h u^k \right)_{i,j} + g \left(\Phi \left(u^k, V^k \right)_{i,j}, V_{i,j}^k \right) = c m_{i,j}^{k+1} + f_0(x_{i,j}),$$

$$(2.9) \quad D_t m_{i,j}^k - \nu \left(\Delta_h m^{k+1} \right)_{i,j} - \mathcal{T} \left(\Phi \left(u^k, V^k \right), m^{k+1} \right)_{i,j} = 0,$$

$$(2.10) \quad V_{i,j,1}^k = -\frac{1}{Z_{i,j}^k} \sum_{r,s} \left(\frac{u_{r+1,s}^k - u_{r-1,s}^k}{2h} - \lambda \theta V_{r,s,1}^k \right) K(x_{i,j}, x_{r,s}) m_{r,s}^{k+1},$$

$$(2.11) \quad V_{i,j,2}^k = -\frac{1}{Z_{i,j}^k} \sum_{r,s} \left(\frac{u_{r,s+1}^k - u_{r,s-1}^k}{2h} - \lambda \theta V_{r,s,2}^k \right) K(x_{i,j}, x_{r,s}) m_{r,s}^{k+1},$$

$$(2.12) \quad Z_{i,j}^k = \sum_{r,s} K(x_{i,j}, x_{r,s}) m_{r,s}^{k+1},$$

for $i, j = 0, \dots, N_h$, $k = 0, \dots, N_T - 1$, where the discrete transport operator \mathcal{T} is defined by

$$(2.13) \quad \mathcal{T}(q, m)_{i,j} = \frac{1}{h} \left(-q_{i,j,1} m_{i,j} + q_{i-1,j,1} m_{i-1,j} + q_{i,j,2} m_{i,j} - q_{i+1,j,2} m_{i+1,j} \right. \\ \left. - q_{i,j,3} m_{i,j} + q_{i,j-1,3} m_{i,j-1} + q_{i,j,4} m_{i,j} - q_{i,j+1,4} m_{i,j+1} \right),$$

for any $i, j = 0, \dots, N_h$. Note that (2.6) is not enough in order to fully determine $\mathcal{T}(\Phi(u, V), m)$ at the boundary nodes. We also need the following quantities:

$$(2.14) \quad \begin{aligned} \Phi(u, V)_{-1,j,1} &= \left[(D_1^+ u)_{-1,j} - \lambda \theta V_{0,j,1} \right]_- = [\lambda \theta V_{0,j,1}]_+, \\ \Phi(u, V)_{N_h+1,j,2} &= - \left[(D_1^+ u)_{N_h,j} - \lambda \theta V_{N_h,j,1} \right]_+ = - [\lambda \theta V_{N_h,j,1}]_-, \\ \Phi(u, V)_{i,-1,3} &= \left[(D_2^+ u)_{i,-1} - \lambda \theta V_{i,0,2} \right]_- = [\lambda \theta V_{i,0,2}]_+, \\ \Phi(u, V)_{i,N_h+1,4} &= - \left[(D_2^+ u)_{i,N_h} - \lambda \theta V_{i,N_h,2} \right]_+ = - [\lambda \theta V_{i,N_h,2}]_-, \end{aligned}$$

for $i, j = 0, \dots, N_h$, where the last identity in each line comes from (2.16) below. The discrete version of (1.11) and (1.12) is

$$(2.15) \quad u_{i,j}^{N_T} = \phi(x_{i,j}), \text{ and } m_{i,j}^0 = m_0(x_{i,j}), \quad 0 \leq i, j \leq N_h.$$

The discrete version of (1.13)- (1.14) is

$$(2.16) \quad \begin{cases} u_{-1,j}^k = u_{0,j}^k & \text{and} & u_{N+1,j}^k = u_{N,j}^k, \\ u_{i,-1}^k = u_{i,0}^k & \text{and} & u_{i,N+1}^k = u_{i,N}^k, \end{cases}$$

$$(2.17) \quad \begin{cases} \nu \left(m_{-1,j}^{k+1} - m_{0,j}^{k+1} \right) - h \left[\lambda \theta V_{0,j,1}^k \right]_+ m_{0,j}^{k+1} + h \left[\lambda \theta V_{0,j,1}^k \right]_- m_{-1,j}^{k+1} = 0, \\ \nu \left(m_{N,j}^{k+1} - m_{N+1,j}^{k+1} \right) + h \left[\lambda \theta V_{N,j,1}^k \right]_+ m_{N,j}^{k+1} - h \left[\lambda \theta V_{N,j,1}^k \right]_- m_{N+1,j}^{k+1} = 0, \\ \nu \left(m_{i,-1}^{k+1} - m_{i,0}^{k+1} \right) - h \left[\lambda \theta V_{i,0,1}^k \right]_+ m_{i,0}^{k+1} + h \left[\lambda \theta V_{i,0,1}^k \right]_- m_{i,-1}^{k+1} = 0, \\ \nu \left(m_{i,N}^{k+1} - m_{i,N+1}^{k+1} \right) + h \left[\lambda \theta V_{i,N,1}^k \right]_+ m_{i,N}^{k+1} - h \left[\lambda \theta V_{i,N,1}^k \right]_- m_{i,N+1}^{k+1} = 0, \end{cases}$$

for $i, j = 0, \dots, N_h$ and $k = 0, \dots, N_T - 1$.

Note that (2.8) is an implicit scheme for (1.7), (recall that (1.7) is backward w.r.t. time), whereas (2.9) is an implicit scheme for (1.8), (recall that (1.8) is forward w.r.t. time). This explains why no restriction is made on the time step.

The discrete Hamiltonian introduced in (2.7) $g : \mathbb{R}^4 \times \mathbb{R}^2 \rightarrow \mathbb{R}$, $(q, V) \mapsto g(q, V)$, has the following properties:

H1 (monotonicity) if $q = (q_1, q_2, q_3, q_4)$, then g is nonincreasing with respect to q_1 and q_3 and nondecreasing with respect to q_2 and q_4

H2 (consistency) $g(q_1, q_1, q_3, q_3, V) = H(q_1, q_3, V)$

H3 (regularity) g is C^1 .

Remark 2.3. We aim at using Newton iterations in order to solve the system of nonlinear equations arising from the discrete scheme. For that purpose, we need to linearize the discrete version of the Fokker-Planck equation. Therefore, for another small positive parameter ε , we may replace the definition (2.6) of Φ by the following:

$$(2.18) \quad \Phi(y, V)_{i,j} = \begin{cases} \left[(D_1^+ y)_{i,j} - \lambda \theta V_{i+\frac{1}{2},j,1} \right]_{-\varepsilon} \\ - \left[(D_1^+ y)_{i-1,j} - \lambda \theta V_{i-\frac{1}{2},j,1} \right]_{+\varepsilon} \\ \left[(D_2^+ y)_{i,j} - \lambda \theta V_{i,j+\frac{1}{2},2} \right]_{-\varepsilon} \\ - \left[(D_2^+ y)_{i,j-1} - \lambda \theta V_{i,j-\frac{1}{2},2} \right]_{+\varepsilon} \end{cases} \quad \text{for } i, j = 0, \dots, N_h,$$

where the C^1 approximation $a \mapsto a_{+,\varepsilon}$ of $a \mapsto a_+ = a \mathbf{1}_{a>0}$ is defined by

$$(2.19) \quad a_{+,\varepsilon} = \frac{1}{2} \left(e^{-\frac{|a|}{\varepsilon}} - 1 \right) \varepsilon + \mathbf{1}_{a>0} a,$$

and $a_{-,\varepsilon} = a_{+,\varepsilon} - a$. We may modify (2.14) and (2.17) accordingly.

Lemma 2.4. Assume that (u, m, V) satisfies (2.8)-(2.12), (2.15), and (2.16), (2.17).

The total mass of m is preserved, i.e.

$$(2.20) \quad \sum_{1 \leq i,j \leq N_h} m_{i,j}^k = \sum_{1 \leq i,j \leq N_h} m_{i,j}^0,$$

for any $k = 0, \dots, N_T$.

Proof. From (2.3), we deduce that

$$(2.21) \quad \sum_{0 \leq i, j \leq N_h} \nu \left(\Delta m^{k+1} \right)_{i,j} = \frac{\nu}{h^2} \left[\sum_{j=0}^{N_h} \left(m_{N_h+1,j}^{k+1} - m_{N_h,j}^{k+1} - m_{-1,j}^{k+1} + m_{0,j}^{k+1} \right) + \sum_{i=0}^{N_h} \left(m_{i,N_h+1}^{k+1} - m_{i,N_h}^{k+1} - m_{i,-1}^{k+1} + m_{i,0}^{k+1} \right) \right].$$

Then we sum (2.13) for $i, j = 0, \dots, N_h$:

$$(2.22) \quad \sum_{0 \leq i, j \leq N_h} \mathcal{J}(q, m)_{i,j} = \frac{1}{h} \sum_{j=1}^{N_h} -q_{N_h,j,1} m_{N_h,j} + q_{-1,j,1} m_{-1,j} + q_{0,j,2} m_{0,j} - q_{N_h+1,j,2} m_{N_h+1,j} + \frac{1}{h} \sum_{i=1}^{N_h} -q_{i,N_h,3} m_{i,N_h} + q_{i,-1,3} m_{i,-1} + q_{i,0,4} m_{i,0} - q_{i,N_h+1,4} m_{i,N_h+1}.$$

From (2.6), (2.14), (2.16) and (2.17), the sums of the right hand sides of (2.22) and (2.21) vanish if $q = \Phi(u^k, V^k)$, for any $k = 0, \dots, N_T - 1$.

From the observations above, summing (2.9) on $i, j = 0, \dots, N_h$ yields

$$\sum_{0 \leq i, j \leq N_h} m_{i,j}^{k+1} = \sum_{0 \leq i, j \leq N_h} m_{i,j}^k,$$

then the desired result. □

2.3 Solving the discrete version of the Hamilton-Jacobi-Bellman equation

For brevity, we will use the notation $(y_{i,j}^k)$ for a grid function, omitting that the indices i, j take their values in $\{0, \dots, N_h\}$ and that k takes its values in $\{0, \dots, N_T\}$. This paragraph is devoted to solving the system of nonlinear equations satisfied by the grid function $(u_{i,j}^k)$, given the grid functions $(m_{i,j}^k)$ and $(V_{i,j}^k)$. Let $\tilde{f} = (\tilde{f}_{i,j}^k)$ denote the grid function defined by

$$(2.23) \quad \tilde{f}_{i,j}^k = c m_{i,j}^{k+1} + f_0(x_{i,j}).$$

We may then rewrite (2.8), (2.16) and the first identity in (2.15) in the compact form

$$(2.24) \quad u = \mathcal{F}_u(\tilde{f}, V).$$

Finding u given m and V amounts to solving a discrete version of a nonlinear parabolic equation posed backward in time with Neumann boundary conditions. This is much simpler than solving the complete forward-backward system for (u, m, V) , because a backward time-marching procedure can be used. Since, as it was already observed above, the scheme is implicit, each time step consists of solving the discrete version of a nonlinear elliptic partial differential equation which is local because V is given. Starting from the terminal time step N_T for which (2.15) gives an explicit formula for u^{N_T} , the k -th step of the backward loop consists of computing u^k by solving (2.8) and (2.16) given u^{k+1} , m^{k+1} and V^k . This is done by means of Newton iterations.

For completeness, let us give a few details on Newton iterations. We introduce the operator \mathcal{R}_u ,

$$(2.25) \quad \mathcal{R}_u(u', v', f', V')_{i,j} = u'_{i,j} - v'_{i,j} + \Delta t \left[-\nu (\Delta_h u')_{i,j} + g \left(\Phi(u', V')_{i,j}, V'_{i,j} \right) - f'_{i,j} \right],$$

for $u', v', f' : \Omega_h \rightarrow \mathbb{R}$, $V' : \Omega_h \rightarrow \mathbb{R}^2$, and $i, j = 0, \dots, N_h$. Note that in (2.25), u' is extended outside Ω_h thanks to (2.16). We aim at approximating the solution of

$$(2.26) \quad \mathcal{R}_u(u', u^{k+1}, \tilde{f}^k, V^k) = 0.$$

Starting from an initial guess noted $u^{k,0} : \Omega_h \rightarrow \mathbb{R}$, the Newton iterations consist of computing by induction a sequence $u^{k,\ell}$ of approximations of the solution to (2.26): given $u^{k,\ell}$, $u^{k,\ell+1}$ is found by solving the system of linear equations

$$d_u \mathcal{R}_u(u^{k,\ell}, u^{k+1}, \tilde{f}^k, V^k)(u^{k,\ell+1} - u^{k,\ell}) = -\mathcal{R}_u(u^{k,\ell}, u^{k+1}, \tilde{f}^k, V^k).$$

In the latter equation, the Jacobian matrix $d_u \mathcal{R}_u(u^{k,\ell}, u^{k+1}, \tilde{f}^k, V^k)$ is sparse since the PDE is local, and invertible since the scheme is monotone. Note that $d_u \mathcal{R}_u(u^{k,\ell}, u^{k+1}, \tilde{f}^k, V^k)$ depends neither on u^{k+1} nor on \tilde{f}^k , so we can write it $d_u \mathcal{R}_u(u^{k,\ell}, \cdot, \cdot, V^k)$. The system of linear equations is solved by using special algorithms for sparse matrices. In the numerical simulations presented below, we use the C-library UMFPACK, see [1], implementing unsymmetric multifrontal method and direct sparse LU factorization.

Note that one may choose the initial guess $u^{k,0} = u^{k+1}$. The Newton iterations are stopped when the residual $|\mathcal{R}_u(u^{k,\ell}, u^{k+1}, \tilde{f}^k, V^k)|$ is small enough, say for $\ell = \ell_0$. Then we set $u^k = u^{k,\ell_0}$. It is well known that the Newton algorithm for (2.8) is equivalent to an optimal policy iteration algorithm, that it is convergent for any initial guess, and that the convergence is quadratic.

2.4 Solving the discrete version of the Fokker-Planck-Kolmogorov equation

This paragraph is devoted to solving the system of linear equations satisfied by the grid function $(m_{i,j}^k)$, given the grid functions $(u_{i,j}^k)$ and $(V_{i,j}^k)$. We may then rewrite (2.9), (2.17) and the second identity in (2.15) in the compact form

$$(2.27) \quad m = \mathcal{F}_m(u, V).$$

Finding m given u and V amounts to solving a discrete version of an linear parabolic equation posed forward in time with Neumann boundary conditions. Since the scheme is implicit, each time step consists of solving the discrete version of a linear elliptic partial differential equation which is local. Starting from the terminal time step 0 for which (2.15) gives an explicit formula for m^0 , the k -th step of the forward loop consists of computing m^{k+1} by solving (2.9), (2.17) given m^k , u^{k+1} and V^k . It is easy to see that the matrix of the latter system of linear equations is exactly the transposed of $d_u \mathcal{R}_u(u^k, \cdot, \cdot, V^k)$ which has been introduced in the previous paragraph. Therefore, it is sparse and invertible. In the numerical simulations presented below, we use the C-library UMFPACK again.

3 - NEWTON ALGORITHMS FOR SOLVING THE WHOLE SYSTEM (2.8)-(2.17)

The solution of the system of nonlinear equations (2.8)-(2.17) is not easy because

1. the system is the discrete version of a system of forward-backward equations, which precludes a simple time marching algorithm
2. The easiest instances of MFGs can be interpreted as optimal control problems driven by a partial differential equation, which opens the way to algorithms based on the variational structure. In the MFGC considered here, there is no underlying variational principle and these algorithms cannot be used.

Following previous works of the first author, see [2], we choose to use a continuation method (for example with respect to the viscosity parameter ν) in which every system of nonlinear equations (given the parameter of the continuation method) is solved by means of Newton iterations. With Newton algorithm, it is important to have a good initial guess of the solution; for that, we take advantage of the continuation method by choosing the initial guess as the solution obtained with the previous value of the parameter. Alternatively, we have sometimes taken the initial guess from the simulation of the same problem on a coarser grid, using interpolation.

It is also important to implement the Newton algorithm on a “well conditioned” system. Therefore, we shall not directly address (2.8)-(2.17), but we shall rather eliminate the unknowns u and m by using the time-marching loops described in paragraphs 2.3 and 2.4 and see (2.8)-(2.17) as a fixed point problem for (\tilde{f}, V) .

Before describing the algorithm, we need to provide a numerical method for obtaining the average drift given u and m , i.e. for solving (2.10)-(2.11) at least approximately.

3.1 The coupling cost and the average drift

Let us introduce \mathcal{F}_f which maps the grid function m defined on $\mathcal{T}_{\Delta t} \times \Omega_h$ to the grid function \tilde{f} given in (2.23). We also define the maps \mathcal{Z} and \mathcal{V} by

$$(3.1) \quad \mathcal{Z}(m')_{i,j} = \sum_{r,s} K(x_{i,j}, x_{r,s}) m'_{r,s},$$

$$(3.2) \quad \mathcal{V}(u', m', V')_{i,j,1} = - \sum_{r,s} \left(\frac{u'_{r+1,s} - u'_{r-1,s}}{2h} - \lambda \theta V'_{r,s,1} \right) K(x_{i,j}, x_{r,s}) \frac{m'_{r,s}}{\mathcal{Z}(m')_{i,j}},$$

$$(3.3) \quad \mathcal{V}(u', m', V')_{i,j,2} = - \sum_{r,s} \left(\frac{u'_{r,s+1} - u'_{r,s-1}}{2h} - \lambda \theta V'_{r,s,2} \right) K(x_{i,j}, x_{r,s}) \frac{m'_{r,s}}{\mathcal{Z}(m')_{i,j}},$$

for $u', m' : \Omega_h \rightarrow \mathbb{R}$, $V' : \Omega_h \rightarrow \mathbb{R}^2$ and $i, j = 0, \dots, N_h$.

It can be proved exactly in the same manner as in the continuous case, see [16, Lemma 2.4], that $V' \mapsto \mathcal{V}(u', m', V')$ is a contraction in the maximum-norm for instance, with a contraction factor $|\lambda|\theta$.

For a positive integer L , we define the map \mathcal{F}_V by :

$$(3.4) \quad \mathcal{F}_V(u, m, V) = \tilde{V},$$

$$(3.5) \quad \tilde{V}^k = \tilde{V}^{k,L}, \quad \text{for } k = 0, \dots, N_T - 1,$$

where $\tilde{V}^{k,0} = V^k$ and the sequence $\tilde{V}^{k,\ell}$ is defined by the following induction:

$$(3.6) \quad \tilde{V}^{k,\ell} = \mathcal{V}\left(u^k, m^{k+1}, \tilde{V}^{k,\ell-1}\right), \quad 1 \leq \ell \leq L.$$

Remark 3.1. 1. With a slight abuse of notation, if $L = \infty$, then \tilde{V}^k is the fixed point of the map $V' \mapsto \mathcal{V}(u^k, m^{k+1}, V')$.

2. The induction (3.6) corresponds to Jacobi iterations and can be easily parallelised. It is also possible to implement Gauss-Seidel iterations, which are a little more complex to write. They consist of using the components of $\tilde{V}^{k,\ell}$ as soon as they are obtained (instead of those of $\tilde{V}^{k,\ell-1}$) in order to compute the components of $\tilde{V}^{k,\ell}$ which have not been obtained yet. In our implementation, we have in fact used the Gauss-Seidel iterations with a lexicographic ordering of the components of the grid functions.

Remark 3.2. As we shall see in paragraph 4.1.4, the choice of L does not impact the convergence of the overall iterative algorithm. Therefore, a good choice turns out to be $L = 1$.

3.2 The linearized operators

3.2.1 Notation

Let $u, m, f : \mathcal{T}_{\Delta h} \times \Omega_h \rightarrow \mathbb{R}$ be generic grid functions standing respectively for discrete versions of the value function, the law of the distribution of states, the right hand side of the discrete HJB equation (2.8). Let $V : \mathcal{T}_{\Delta h} \times \Omega_h \rightarrow \mathbb{R}^2$ be a generic grid function standing for the average drift. Let us introduce the operators obtained by differentiation of the maps $\mathcal{F}_u, \mathcal{F}_m$ and \mathcal{F}_V :

$$(3.7) \quad \begin{aligned} B_{u,f} &= D_f \mathcal{F}_u(f, V), & B_{u,V} &= D_V \mathcal{F}_u(f, V), \\ B_{m,V} &= D_u \mathcal{F}_m(u, V), & B_{m,V} &= D_V \mathcal{F}_m(u, V), \\ C_{f,m} &= D_m \mathcal{F}_f(m), & C_{V,u} &= D_u \mathcal{F}_V(u, m, V), \\ C_{V,m} &= D_m \mathcal{F}_V(u, m, V), & C_{V,V} &= D_V \mathcal{F}_V(u, m, V). \end{aligned}$$

In the three paragraphs below, we explain how these differential operators can be computed.

3.2.2 Linearized Hamilton-Jacobi-Bellman equation

The variation du of $u = \mathcal{F}_u(f, V)$ induced by variations df and dV of f and V is given by

$$(3.8) \quad du = \begin{pmatrix} B_{u,f} & B_{u,V} \end{pmatrix} \begin{pmatrix} df \\ dV \end{pmatrix}.$$

More explicitly, du is obtained by linearizing (2.8), (2.16) and (2.15). It satisfies

$$(3.9) \quad -D_t du_{i,j}^k - \nu \left(\Delta_h du^k \right)_{i,j} + \Phi \left(u^k, V^k \right)_{i,j} \cdot \left(d\Phi \left(u^k, V^k \right) \left(du^k, dV^k \right) \right)_{i,j} - \lambda^2 \theta V_{i,j}^k \cdot dV_{i,j}^k = df_{i,j}^k,$$

for $i, j = 0, \dots, N_h$ and $k = 0, \dots, N_T$. The first (respectively second) inner product appearing in (3.9) takes place in \mathbb{R}^4 (respectively \mathbb{R}^2). The set of equations (3.9) is supplemented with the terminal condition

$$(3.10) \quad du_{i,j}^{N_T} = 0,$$

and the boundary conditions

$$(3.11) \quad \begin{cases} du_{-1,j}^k = du_{0,j}^k, & du_{N_h+1,j}^k = du_{N_h,j}^k, \\ du_{i,-1}^k = du_{i,0}^k, & du_{i,N_h+1}^k = du_{i,N_h}^k. \end{cases}$$

Given df and dV , the variation du is found by solving (3.9), (3.10) and (3.11). This is done by marching backward in time and at each time step solving a system of linear equations with a sparse and invertible matrix (of the same form as in the Newton iterations described in 2.3). Again, we use the library UMFPACK for that.

3.2.3 Linearized Kolmogorov-Fokker-Planck equation

The variation dm of $m = \mathcal{F}_m(u, V)$ induced by variations du and dV of u and V is given by

$$(3.12) \quad dm = \begin{pmatrix} B_{m,u} & B_{m,V} \end{pmatrix} \begin{pmatrix} du \\ dV \end{pmatrix}.$$

The grid function dm is obtained by linearizing (2.9), (2.17) and (2.15). It satisfies

$$(3.13) \quad D_t dm_{i,j}^k - \nu \left(\Delta_h dm^{k+1} \right)_{i,j} - \mathcal{T}_{i,j} \left(d\Phi \left(u^k, V^k \right) \left(du^k, dV^k \right) \right), m^{k+1} - \mathcal{T}_{i,j} \left(\Phi \left(u^k, V^k \right), dm^{k+1} \right) = 0,$$

for $i, j = 0, \dots, N_h$ and $k = 0, \dots, N_T$. The set of equations (3.13) is supplemented with the initial condition

$$(3.14) \quad dm_{i,j}^0 = 0, \quad i, j = 0, \dots, N_h,$$

and the boundary conditions

$$(3.15) \quad \begin{cases} \nu \left(dm_{-1,j}^{k+1} - dm_{0,j}^{k+1} \right) - h \left[\lambda \theta V_{0,j,1}^k \right]_+ dm_{0,j}^{k+1} + h \left[\lambda \theta V_{0,j,1}^k \right]_- dm_{-1,j}^{k+1} = 0, \\ \nu \left(dm_{N,j}^{k+1} - dm_{N+1,j}^{k+1} \right) + h \left[\lambda \theta V_{N,j,1}^k \right]_+ dm_{N,j}^{k+1} - h \left[\lambda \theta V_{N,j,1}^k \right]_- dm_{N+1,j}^{k+1} = 0, \\ \nu \left(dm_{i,-1}^{k+1} - dm_{i,0}^{k+1} \right) - h \left[\lambda \theta V_{i,0,1}^k \right]_+ dm_{i,0}^{k+1} + h \left[\lambda \theta V_{i,0,1}^k \right]_- dm_{i,-1}^{k+1} = 0, \\ \nu \left(dm_{i,N}^{k+1} - dm_{i,N+1}^{k+1} \right) + h \left[\lambda \theta V_{i,N,1}^k \right]_+ dm_{i,N}^{k+1} - h \left[\lambda \theta V_{i,N,1}^k \right]_- dm_{i,N+1}^{k+1} = 0. \end{cases}$$

Here again, (3.13), (3.14) and (3.15) is solved by marching in time and solving a system of linear equations at each time step (using UMFPAK).

3.2.4 Linearized coupling costs and average drifts

The variation of $\tilde{f} = \mathcal{F}_f(m)$ induced by a variation dm of m is obviously given by $d\tilde{f}_{i,j}^k = c dm_{i,j}^k$, hence $C_{f,m} dm = c dm$.

Let us turn to the variation of $\tilde{V} = \mathcal{F}_V(u, m, V)$ induced by the variations du , dm and dV of u , m and V . Differentiating (3.4)-(3.6) leads to

$$\begin{aligned} d\tilde{V}^{k,0} &= dV^k, \\ d\tilde{V}^{k,\ell} &= \mathcal{V} \left(du^k, m^{k+1}, d\tilde{V}^{k,\ell-1} \right) + d_m \mathcal{V} \left(u^k, m^{k+1}, \tilde{V}^{k,\ell-1} \right) dm^{k+1}, \\ d\tilde{V}^k &= d\tilde{V}^{k,L}, \end{aligned}$$

see (3.6) for the definition of $\tilde{V}^{k,L}$.

This permits to compute

$$d\tilde{V} = \begin{pmatrix} C_{V,u} & C_{V,m} & C_{V,V} \end{pmatrix} \begin{pmatrix} du \\ dm \\ dV \end{pmatrix}.$$

To summarize,

$$\begin{pmatrix} d\tilde{f} \\ d\tilde{V} \end{pmatrix} = C \begin{pmatrix} du \\ dm \\ dV \end{pmatrix}, \quad \text{with} \quad C = \begin{pmatrix} 0 & C_{f,m} & 0 \\ C_{V,u} & C_{V,m} & C_{V,V} \end{pmatrix}.$$

3.3 The algorithm for solving (2.8)-(2.17)

We see (2.8)-(2.17) as a fixed point problem for the pair (f, V) , which we write

$$(3.16) \quad \mathcal{G}_f(f, V) = 0, \quad \text{and} \quad \mathcal{G}_V(f, V) = 0,$$

where $\mathcal{G}_f, \mathcal{G}_V$ are defined by

$$(3.17) \quad \mathcal{G}_f(f, V) = f - \mathcal{F}_f \circ \mathcal{F}_m \left(\mathcal{F}_u(f, V), V \right),$$

$$(3.18) \quad \mathcal{G}_V(f, V) = V - \mathcal{F}_V \left(\mathcal{F}_u(f, V), \mathcal{F}_m \left(\mathcal{F}_u(f, V), V \right), V \right).$$

Remark 3.3. Given u and m , $V \mapsto W = \mathcal{F}_V(u, m, V)$ is obtained as follows: for each k , computing W^k consists of iterating $W^k \times \mathcal{V}^L(u^k, m^{k+1}, W^k)$ L times starting from $W^k = V^k$. Recall that $\mathcal{V}(u^k, m^{k+1}, \cdot)$ is in (3.2) and (3.3), and is a contraction with a unique fixed point. Therefore, $\mathcal{F}_V(u, m, \cdot)$ has also a unique fixed point which does not depend on L . Since \mathcal{F}_u and \mathcal{F}_m do not depend on L , neither do the solutions of (3.16).

The Newton iterations involve the Jacobian $A(f, V)$ of the map $(f, V) \mapsto (\mathcal{G}_f(f, V), \mathcal{G}_V(f, V))$. We set

$$(3.19) \quad \begin{cases} A_{f,f}(f, V) = D_f \mathcal{G}_f(f, V), & A_{f,V}(f, V) = D_V \mathcal{G}_f(f, V), \\ A_{V,f}(f, V) = D_f \mathcal{G}_V(f, V), & A_{V,V}(f, V) = D_V \mathcal{G}_V(f, V), \end{cases}$$

and the blocks of the Jacobian $A(f, V)$ are defined by:

$$(3.20) \quad A_{f,f}(f, V) = I_N - C_{f,m} B_{m,u} B_{u,f},$$

$$(3.21) \quad A_{f,V}(f, V) = -C_{f,m} (B_{m,u} B_{u,V} + B_{m,V}),$$

$$(3.22) \quad A_{V,f}(f, V) = -C_{V,u} B_{u,f} - C_{V,m} B_{m,u} B_{u,f},$$

$$(3.23) \quad A_{V,V}(f, V) = I_{2N} - C_{V,u} B_{u,V} - C_{V,m} (B_{m,u} B_{u,V} + B_{m,V}) - C_{V,V},$$

where $B_{u,f}$, $B_{u,V}$, $B_{m,u}$, $B_{m,V}$, $C_{V,u}$, $C_{V,m}$, $C_{V,V}$ are given by (3.7) with $u = \mathcal{F}_u(f, V)$ and $m = \mathcal{F}_m(u, V)$. In an equivalent manner, we can write

$$(3.24) \quad A(f, V) = I_{3N} - \begin{pmatrix} C_{f,m} B_{m,u} B_{u,f} & C_{f,m} (B_{m,u} B_{u,V} + B_{m,V}) \\ C_{V,u} B_{u,f} + C_{V,m} B_{m,u} B_{u,f} & C_{V,u} B_{u,V} + C_{V,m} (B_{m,u} B_{u,V} + B_{m,V}) + C_{V,V} \end{pmatrix},$$

or

$$(3.25) \quad A(f, V) = I_{3N} - \begin{pmatrix} 0_N & C_{f,m} & 0_{N,2N} \\ C_{V,u} & C_{V,m} & C_{V,V} \end{pmatrix} \begin{pmatrix} I_N & 0_{N,2N} \\ B_{m,u} & B_{m,V} \\ 0_{2N,N} & I_{2N} \end{pmatrix} \begin{pmatrix} B_{u,f} & B_{u,V} \\ 0_{2N,N} & I_{2N} \end{pmatrix}.$$

Every Newton iteration for solving (3.16) consists of solving a system of linear equations of the form

$$(3.26) \quad A(f^\ell, V^\ell) \begin{pmatrix} f^{\ell+1} - f^\ell \\ V^{\ell+1} - V^\ell \end{pmatrix} = - \begin{pmatrix} \mathcal{G}_f(f^\ell, V^\ell) \\ \mathcal{G}_V(f^\ell, V^\ell) \end{pmatrix}.$$

In our implementation, this system is solved iteratively by BiCGStab algorithm, see [21]. Note that BiCGStab algorithm only requires a function that computes

$$A(f^\ell, V^\ell) \begin{pmatrix} \hat{f} \\ \hat{V} \end{pmatrix},$$

for any grid functions $\hat{f} : \mathcal{T}_{\Delta t} \times \Omega_h \rightarrow \mathbb{R}$ and $\hat{V} : \mathcal{T}_{\Delta t} \times \Omega_h \rightarrow \mathbb{R}^2$. This is done using (3.25) and combining the methods described in paragraph 3.2. The assembly of the Jacobian matrix is not needed.

4 - NUMERICAL SIMULATIONS

We are going to discuss the results of numerical simulations in two cases, both related to crowd motion.

4.1 First example

4.1.1 Description of the model

The state space is the square $\Omega = (-0.5, 0.5)^2$ and the time horizon is $T = 1$.

We consider the MFGC described by (1.7)-(1.14), in which

1. The kernel k in (1.9) is radial, i.e. of the form $K(x, y) = K_\rho(|x - y|)$. Here $r \mapsto K_\rho(r)$ is a non-increasing C^1 function defined on \mathbb{R}_+ , with

$$(4.1) \quad K_\rho(r) = 0, \quad \text{if } r \geq \rho, \quad \text{and} \quad K_\rho(r) = 1, \quad \text{if } r \leq 0.9\rho,$$

for a positive number $\rho > 0$, which is the radius of the disc in the state space that a representative agent uses for computing the average of the controls.

2. The parameter c in (1.7) is chosen as $c = 10^{-3}$. Recall that the cost $cm(t, x)$ reflects the aversion of a representative player to crowded regions of the state space.
3. At time $t = 0$, the agents are distributed in the top-right and the bottom-left corners of the domain Ω , see the left part of Figure 1. The density is piecewise constant, with values appearing on Figure 1.
4. The terminal cost is also piecewise constant. It takes a small value in the top-left and the bottom-right corners of the domain Ω , see the right part of Figure 1 for the chosen values. This cost attracts the agents to the latter two corners of Ω .
5. The function f_0 in (1.7) is chosen to be proportional to the terminal cost, with a factor 0.1. This term is linked to the running cost and has the same effect as the terminal cost.
6. Recall that the boundary conditions (1.13)-(2.17) rule out the entry or exit of agents. The total mass of m is conserved.

Remark 4.1. *Note that the problem is symmetric with respect to the two diagonals of the square domain. The grid of the domain is chosen to have the same symmetry.*

Remark 4.2. *The two parts of the running cost: $cm(t, X_t)$ and $\frac{\theta}{2} |\alpha - \lambda V(t, X_t)|^2$ may have competing effects: indeed, the former cost may incitate the agents to spread all over the state space, whereas the latter may result in the agents selecting the same controls therefore staying grouped. Although the constant $c = 10^{-3}$ seems small, it is chosen in such a way that the above-mentioned two costs have the same orders of magnitude for e.g. $\theta = 1$ and $\lambda = 0.9$.*

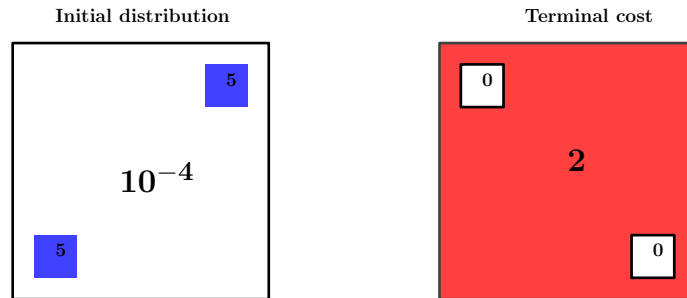


Figure 1: Example 1. Left: the initial distribution of states. Right: the terminal cost.

4.1.2 The results of the simulations: discussion

In the present simulation, we choose the parameters as follows:

$$\nu = 10^{-3}, \quad \theta = 1, \quad \lambda = 0.9, \quad \rho = 0.2, \quad c = 10^{-3}.$$

On Figure 2, we display snapshots of m , u and the optimal feedback law at several times.

Since both the problem and the grid are symmetric with respect to the two diagonals, m , u and the feedback have the same symmetry for all times. Let us describe the evolution of the distribution of states, that we can name “*gathering-kissing-splitting*” referring to the “*drafting-kissing-tumbling*” phenomenon in fluid mechanics (for the interaction of particles in a fluid).

- The term $cm(t, X_t)$ in the running cost prevents the part of the distribution initially supported in one of the opposite corners (say the bottom-left corner) to travel directly to one of the targets (say the bottom-right corner). On the other hand, the interaction through controls prevents this part of the distribution to split into two equal parts which would travel directly to the two targets, because the agents favor controls close to the local average. Therefore, the part of the distribution initially supported in one of the opposite corners first travels to the center of the domain. This is the *gathering* phase of the evolution.
- The two parts of the distribution reach the center of the domain, where the local average of the velocity becomes small; the dominating cost then becomes the one which attracts the agents to the bottom-right and top-left corners. Because of the repulsive effect due to the part $cm(t, X_t)$ of the running cost, the part of the distribution initially supported in one of the opposite corners (say the bottom-left corner) and having traveled toward the center of the domains splits into two parts which make each a ninety-degrees turn. This is the *kissing* phase of the evolution.
- After the *kissing* phase, the distribution of states *splits* into parts which travel to the targets.

Finally, we see that the paths followed by the agents is far from being a shortest path to the targets.

Let us compare these results with a simulation of the MFG obtained by cancelling either θ or λ in (1.7)-(1.14) while keeping all the other parameters and the grid unchanged. Since the problems remains symmetric with respect to the two diagonals, the obtained results have the same symmetry. On Figure 3, we display snapshots of m , u and the optimal feedback law at several times. We see that the evolution is quite different from that displayed on Figure 2, since the two parts of the distribution initially supported in two opposite corners of the domain symmetrically split into two parts each, which travels directly to the targets. The initial splitting phase is due to the coupling cost $cm(t, X_t)$ which favors the spread of the distribution. The path followed by the agents is close to a shortest path to the targets.

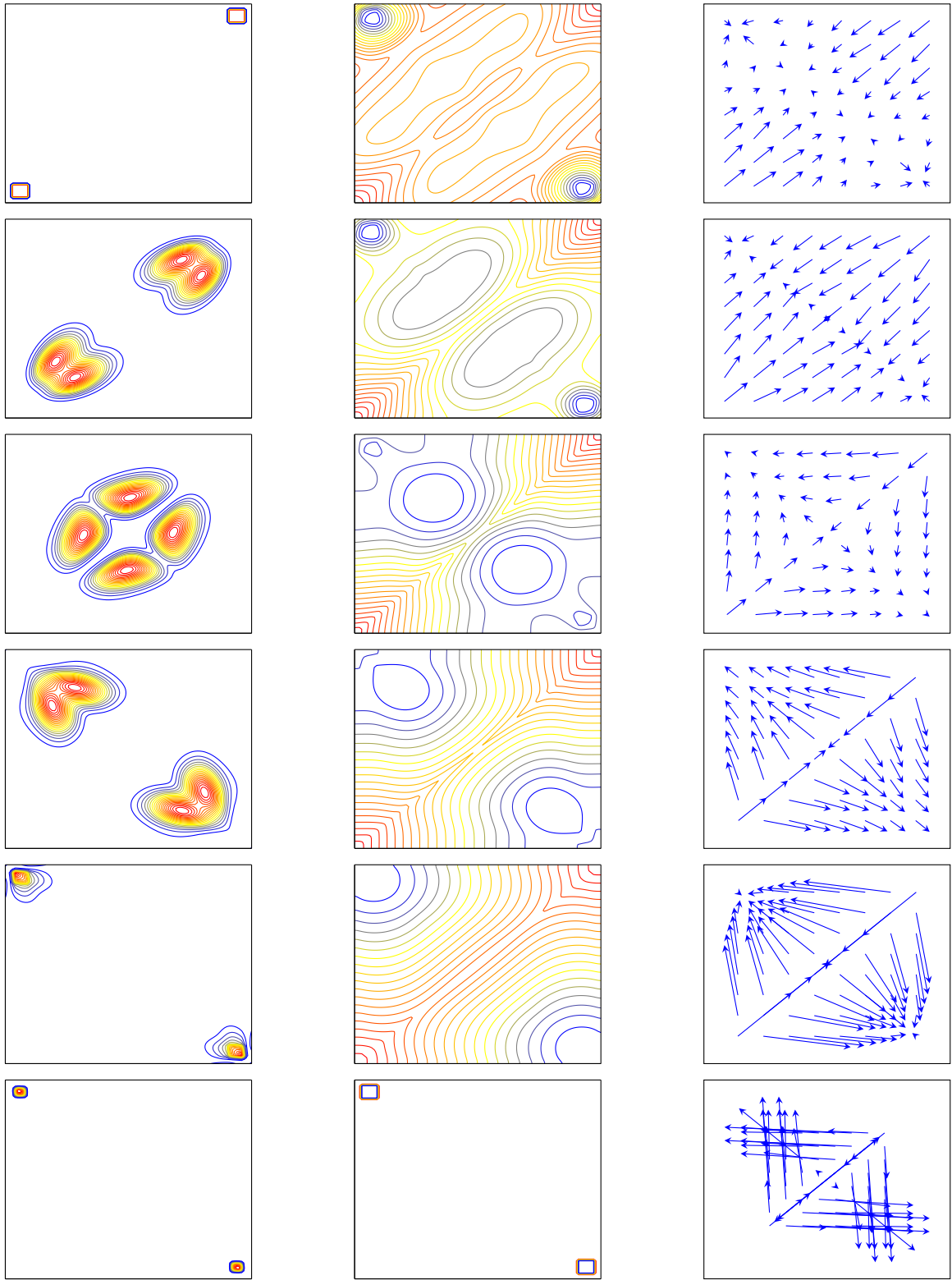


Figure 2: Example 1. Gathering-kissing-splitting; snapshots at $t = 0, 0.2, 0.4, 0.6, 0.8, 1$. Left: contours of m . Center: contours of u . Right: optimal feedback α .

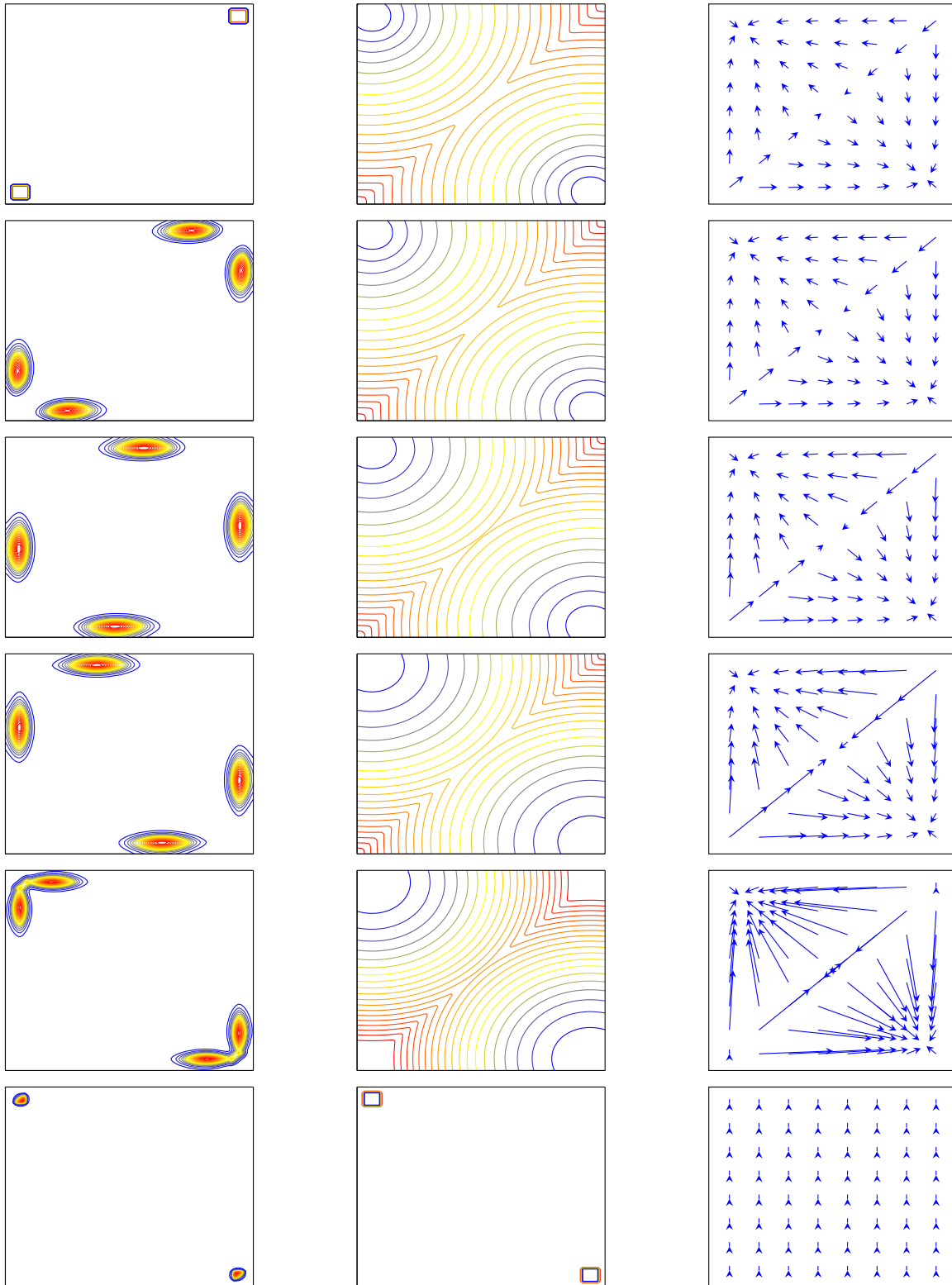


Figure 3: Example 1. Same parameters except $\theta\lambda = 0$; snapshots at $t = 0, 0.2, 0.4, 0.6, 0.8, 1$. Left: contours of m . Center: contours of u . Right: optimal feedback α .

4.1.3 Non-uniqueness of solutions

The solution of the MFGC displayed on Figure 2 is likely to be unstable because the paths followed by the agents are significantly longer than the shortest paths to the targets. We expect that there

are other solutions. We are going to see that this is indeed the case. For that purpose, we are going to introduce a vanishing perturbation of the initial distribution which breaks the symmetry of the problem. This will lead to different solutions to (1.7)-(1.14). An example of perturbation is displayed on Figure 4. It consists of adding very little mass at the top and the bottom of the domain; the perturbed distribution is no longer symmetric with respect to the diagonals. We expect that the agents initially distributed near the bottom target will go to the right, and that all the agents initially distributed at the top of the domain will follow them, because of the interaction through controls. Similarly, we expect that all the agents initially located at the top of the domain will go to the left.

In our simulations, we use a continuation method, i.e. we consider perturbations corresponding to a decreasing sequence of nonnegative parameters $(\pi_n)_{n=0,\dots,N}$. The last value $\pi_N = 0$ corresponds to the distribution displayed on the left of Figure 1. For each new value π_{n+1} of the parameter, we initialize the Newton iterations described in Section 3 by the solution corresponding to the preceding value π_n . For positive values of π_n , the simulated solution is not symmetric with respect to the diagonals, and this property is preserved for the last value $\pi_N = 0$.

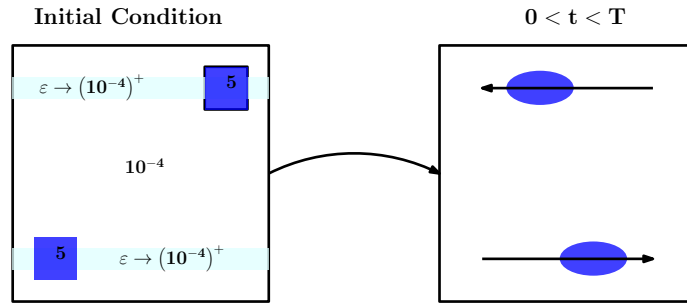


Figure 4: Example 1. A continuation method leading to a different solution. Left: a vanishing perturbation of the initial distribution. Right: the expected evolution of the distribution.

On Figure 5, we display three different solutions (the distribution of states at different times) obtained with three different sequences of vanishing perturbations of the initial distribution displayed on Figure 1: the solution displayed on the left corresponds to the sequence of perturbations displayed on Figure 4; applying to the perturbation a symmetry with respect to a diagonal, we obtain the solution displayed on the center; the solution displayed on the right is obtained by another kind of perturbation symmetric with respect to one diagonal only, located at the top and at the left of the domain: for this solution, we again notice gathering and kissing phases, and that the paths followed by the agents deviate from the shortest ones.

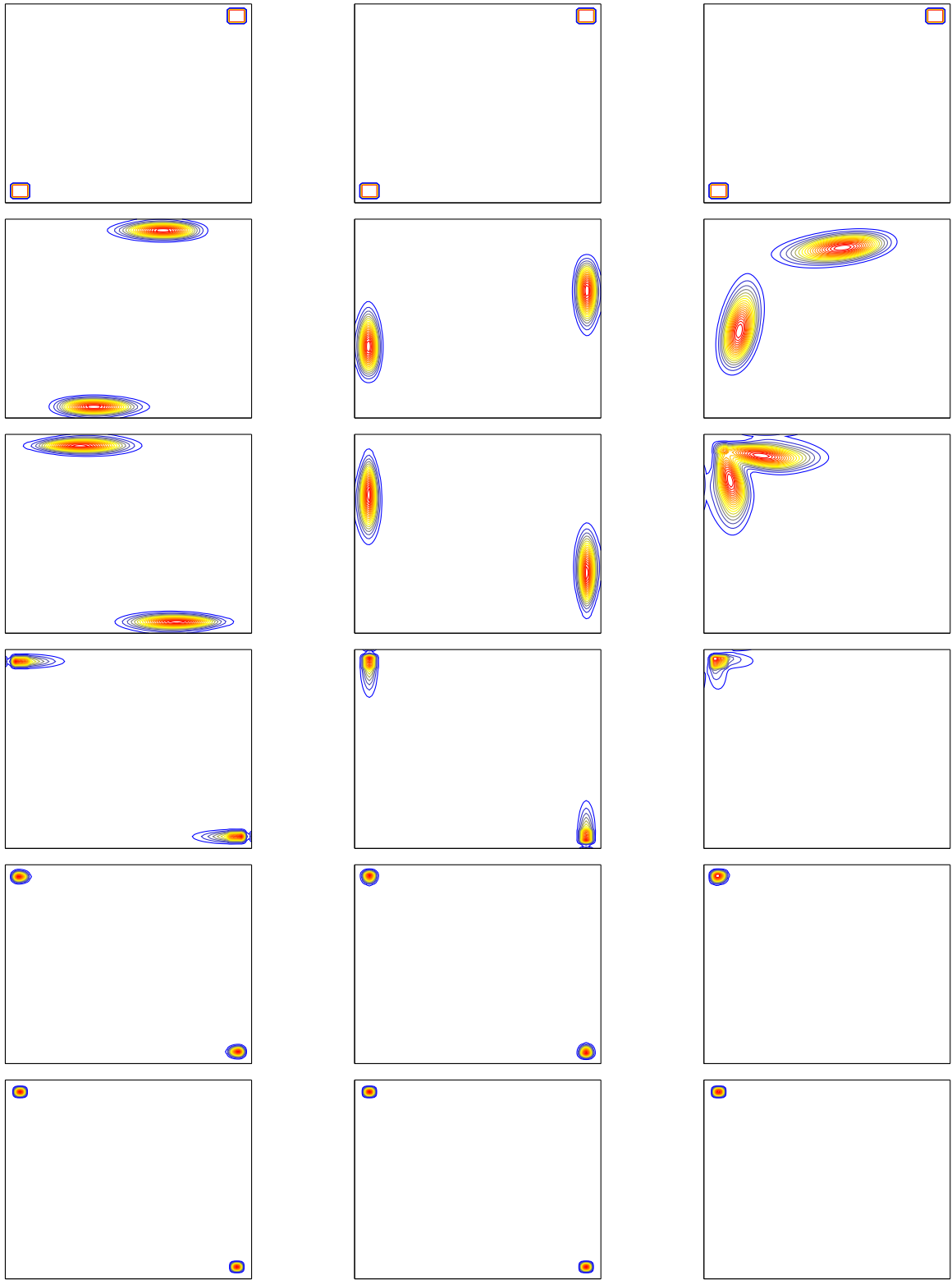


Figure 5: Example 1. Different solutions obtained by adding vanishing perturbations to the initial distribution of states. The distribution at time $t = 0, 0.2, 0.4, 0.6, 0.8, 1$. The three columns correspond to three different sequences of perturbations.

4.1.4 Behaviour of the algorithm

In this paragraph, we investigate how the behaviour of the algorithm described in Section 3 is affected by the variations of the parameters in the model. Recall that the algorithm is based on Newton iterations with a BiCGStab inner loop at each step.

We start by the grid size and the viscosity parameter ν : in the experiments reported below, the stopping criterion for the outer Newton iteration is that the normalized Euclidean norm of the residual is smaller than 10^{-8} . The stopping criterion for the inner BiCGStab iterations is that the ratio of the norms of the current and the initial residuals is smaller than 10^{-7} (hence, it involves a relative error). The parameters of the model are set to

$$\theta = 1, \quad \lambda = 0.9, \quad c = 10^{-3}, \quad \rho = 0.2, \quad L = 1.$$

Table 1 displays the iterations counts for the outer Newton and the inner BiCGStab loops for different viscosities and grid sizes. We notice that the number of iterations is not very sensitive to the grid size, as expected because the map $\mathcal{G} - I_d$, with \mathcal{G} defined in (3.17) (3.18), is the discrete version of a map which has compactness properties. The iteration counts increase as ν is decreased.

Table 1: Number of outer Newton iterations and average number of inner BiCGStab iterations per Newton step (the stopping criteria are given in the text) with different viscosities and grid sizes.

ν	$26 \times 26 \times 26$		$51 \times 51 \times 51$		$76 \times 76 \times 76$		$101 \times 101 \times 101$	
	Newton	BiCGStab	Newton	BiCGStab	Newton	BiCGStab	Newton	BiCGStab
0.5	3	13.67	3	12.67	3	15	3	14.33
0.1	9	14.67	10	21.9	12	15.08	12	18.25
0.05	4	16.5	4	17.25	4	16.25	4	19.25
0.01	4	42.5	4	29	5	30.4	5	28.8

Table 2 displays the iteration counts for different choices of λ and θ , and fixed values the other parameters:

$$\nu = 10^{-3}, \quad c = 10^{-3}, \quad \rho = 0.2, \quad L = 1,$$

with a 101×101 nodes grid and 101 time steps.

In fact, the numbers in Table 2 are obtained by running a continuation method in θ and λ : for each cell of the table, the solution corresponding either to the left or the upper neighboring cell is used as an initial guess for the Newton iterations. If a cell has two such neighbors, the choice of the initial guess is made as follows: in the top-right triangular part of the table, strictly above the diagonal, we choose the initial guess corresponding to the left neighboring cell. In the bottom-left triangular part of the table, including the diagonal, we choose the initial guess corresponding to the cell immediately above.

We see that changing λ and θ mostly impacts the number of iterations of the inner loop. A reason for that is that V is obtained by solving a linear fixed point problem whose contraction factor is $\lambda\theta$. Hence, it is sensible that the number of iterations necessary to solve the systems of linear equations arising in the Newton steps increases as $\lambda\theta$ tends to 1.

Recall that we use a continuation method, consisting of decreasing progressively ν until it reaches the desired value; in Figure 6 (respectively Figure 7), we plot the average number of BiCGStab iterations versus ν , (respectively the number of Newton iterations versus ν) for different values of λ , θ and c . The stopping criteria have been described above. The grid contains 101×101 nodes and there are 101 time steps. We recover the information contained in Tables 1 and 2, i.e. that ν mostly impacts the number of iterations in the inner loop. We also notice that when ν is large, the number of iterations seems to depend more on λ and θ than on c and that it becomes highly sensitive to c when ν is small, but we do not really know how to explain this. The number of iterations in

Table 2: Sensitivity to λ and θ .

(a) Average number of BiCGStab iterations.

$\lambda \backslash \theta$	0.2	0.4	0.6	0.8	1
0.2	3.5	4.5	7	8	8
0.4	4.5	7	8	9	10.67
0.6	5	8	9	12	15.67
0.8	7	8.67	10.67	15.67	27.25
0.9	7.33	8.67	12	18	40

(b) number of Newton iterations.

$\lambda \backslash \theta$	0.2	0.4	0.6	0.8	1
0.2	2	2	2	2	2
0.4	2	3	2	3	3
0.6	2	3	3	3	3
0.8	3	3	3	3	4
0.9	3	3	3	3	6

the outer Newton loop seems much less sensitive than the number of inner BiCGStab iterations. Looking at the overall number of BiCGStab iterations, they are not very different for the choices $(\lambda, \theta, c) = (0.9, 1, 0)$ and $(\lambda, \theta, c) = (0, 0, 0.1)$, since the latter case needs more Newton iterations but less BiCGStab iterations per Newton steps.

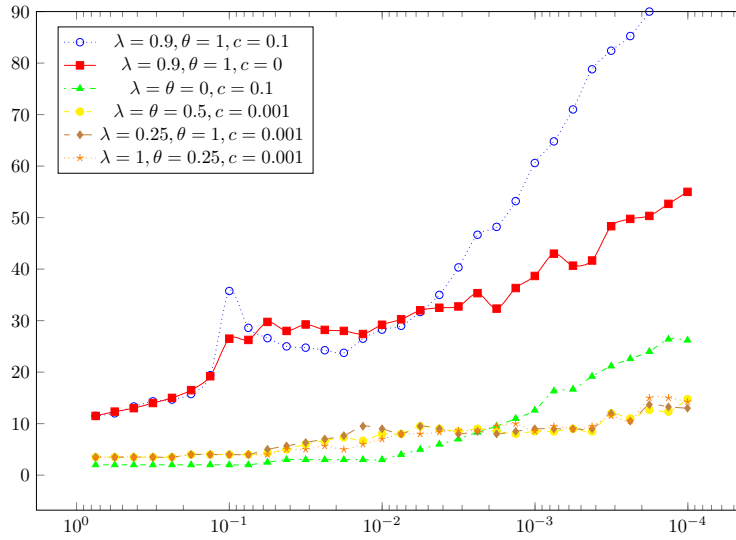


Figure 6: Average number of iterations of BiCGStab iterations per Newton step versus ν for different choices of λ , θ and c .

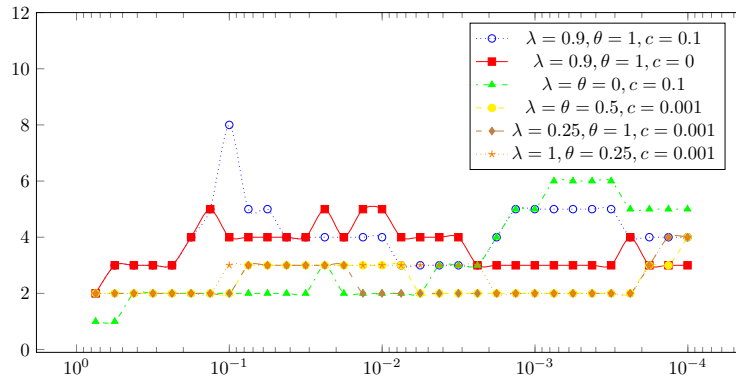


Figure 7: Number of iterations of Newton iterations versus ν for different choices of λ , θ and c .

Finally, we investigate the sensitivity of the algorithm to L , the number of fixed point iterations yielding a proxy of V , see paragraph 3.1. Table 3 contains the iteration counts of the inner and outer

loops for different values of ν and L , and two choices of (λ, θ) (all the other parameters are fixed $c = 10^{-3}$, $\rho = 0.2$). We notice that the choice of L seems to have little impact on the iteration count. For these reasons, $L = 1$ appears to be a good choice.

Table 3: Number of outer Newton iterations and average number of inner BiCGStab iterations per Newton step (the stopping criteria are given in the text) with different viscosities and L .

$\lambda = \theta = 0.8$								
$\nu \backslash L$	1		2		3		5	
	Newton	BiCGStab	Newton	BiCGStab	Newton	BiCGStab	Newton	BiCGStab
0.1	3	7.33	3	6	3	6	3	5.33
0.01	3	13.67	3	13	3	15.33	3	13.67
0.001	2	14.5	2	14.5	2	17.5	2	16
0.0001	3	22	3	22.67	3	22.33	3	24.67
$\lambda = 0.9, \theta = 1$								
$\nu \backslash L$	1		2		3		5	
	Newton	BiCGStab	Newton	BiCGStab	Newton	BiCGStab	Newton	BiCGStab
0.1	4	26.75	4	19.75	4	19.5	4	19.25
0.01	4	29	4	25.75	5	28	5	29
0.001	3	36.67	3	34	3	38.67	3	37.67
0.0001	3	55	3	58.33	3	64	3	63.33

4.2 Second example

As a second example, we consider a model for a crowd crossing a hall, with an entrance at the left and an exit at the right. Roughly speaking, a generic agent will interact with those located in a cone ahead of her.

4.2.1 Description of the model

The state space is the rectangle $\Omega = (-1, 1) \times (-0.1, 0.1)$ and the time horizon is $T = 8$. Let ℓ denote the length of the bottom and upper sides of $\partial\Omega$: $\ell = 2$.

The boundary of Ω is split into three parts Γ_{ND} , Γ_{NN} and Γ_{DD} which respectively stand for an entrance, some walls and an exit:

$$(4.2) \quad \Gamma_{DD} = \{1\} \times [-0.05, 0.05], \quad \Gamma_{ND} = \{-1\} \times [-0.05, 0.05], \quad \Gamma_{NN} = \partial\Omega \setminus (\Gamma_{DD} \cup \Gamma_{ND}),$$

and we consider the following boundary conditions:

$$(4.3) \quad \left. \begin{aligned} \frac{\partial u}{\partial n}(t, x) &= 0 \\ \nu \frac{\partial m}{\partial n}(t, x) - \lambda \theta m(t, x) V(t, x) \cdot n(x) &= 0 \end{aligned} \right\} \quad \text{on } \Gamma_{NN},$$

$$(4.4) \quad \left. \begin{aligned} u(t, x) &= 6 \\ \nu \frac{\partial m}{\partial n}(t, x) + m(t, x) H_p(\nabla u(t, x), V(t, x)) \cdot n(x) &= -2 \end{aligned} \right\} \quad \text{on } \Gamma_{ND},$$

$$(4.5) \quad \left. \begin{aligned} u(t, x) &= -4 \\ m(t, x) &= 0 \end{aligned} \right\} \quad \text{on } \Gamma_{DD},$$

where H is defined below.

The system of PDEs is still (1.7)-(1.10), with the following new features:

- compared to (1.5), the part of the running cost accounting for the interaction through controls is multiplied by a positive normalization factor a :

$$(4.6) \quad L(\alpha, V) = a \frac{\theta}{2} |\alpha - \lambda V|^2 + a \frac{1 - \theta}{2} |\alpha|^2,$$

and the associated Hamiltonian by the Legendre's transform is

$$(4.7) \quad H(p, V) = \frac{a}{2} \left| \frac{p}{a} - \lambda \theta V \right|^2 - a \frac{\lambda^2 \theta}{2} |V|^2.$$

We choose the normalization factor a as follows:

$$(4.8) \quad a = \tilde{a}(1 - \lambda^2 \theta)^{-1},$$

where \tilde{a} is a positive constant (independent of λ or θ). As we shall see later, the normalization allows us to compare the solutions obtained with different values of λ and θ . We shall also see that this normalization is specific to the example under consideration and may not be relevant in other situations.

- The kernel K is given by

$$(4.9) \quad K(x, y) = \mathbf{1}_{y_1 \geq x_1} K_\rho(|x - y|) \kappa(|\omega|),$$

where ω is the angle made by the vector $y - x$ with the vector $(1, 0)$ and

1. k_ρ is defined as in the first example
2. κ is a nonincreasing C^1 function defined such that

$$\kappa = \begin{cases} 1, & \text{if } 0 \leq \omega \leq 0.9 \omega_0, \\ 0, & \text{if } \omega \geq \omega_0, \end{cases}$$

for a given angle $\omega_0 \in (0, \frac{\pi}{2})$.

- The parameter c in (1.7) is chosen to be zero
- We shall consider various functions f_0 modeling the cost for staying at a given point in the domain.

The initial condition is $m_0(x) = 10^{-4}$ for all $x \in \Omega$, and the terminal cost is 0.

4.2.2 The case when f is constant: comparisons with a one-dimensional problem

In this paragraph, we choose $f_0(x) = F$ for all $x \in \Omega$.

Simplification: a one-shot one-dimensional game We approximate the MFGC by a one-shot one-dimensional game. The state space is the interval $[0, \ell]$. The points $x = 0$ and $x = \ell$ respectively stand for an entrance and an exit. When she enters the domain, a representative agent chooses her drift $\alpha \in \mathbb{R}_+$ once and for all, and her dynamics is given by $x'(t) = \alpha$ for $t = \ell/\alpha$. The distribution of drifts is a probability measure μ on \mathbb{R}_+ and the average drift is $V = \int_{\alpha \in \mathbb{R}_+} \alpha d\mu(\alpha)$. There is no entry or exit costs.

With the running cost $L(\alpha, V) + F$, where L is given by (4.6), the mean field Nash equilibrium reads

$$\text{support}(\mu) \subset \operatorname{argmin}_\alpha \frac{\ell}{\alpha} \left(a \left(\frac{\theta}{2} (\alpha - \lambda V)^2 + \frac{1 - \theta}{2} \alpha^2 \right) + F \right).$$

Given V , one checks that the unique solution of the minimization problem in α is

$$(4.10) \quad \alpha^*(V) = \left(\frac{2F + a\theta\lambda^2 V^2}{a} \right)^{\frac{1}{2}}.$$

Hence $\mu = \delta_{\alpha^*(V)}$, and the static mean field Nash equilibrium reads

$$V = \left(\frac{2F + a\theta\lambda^2 V^2}{a} \right)^{\frac{1}{2}},$$

which yields

$$(4.11) \quad \alpha^* = V = \left(\frac{2F}{a(1 - \theta\lambda^2)} \right)^{\frac{1}{2}}.$$

This leads us to choose $a = \tilde{a}/(1 - \theta\lambda^2)$, which yields $\alpha^* = V = \sqrt{\frac{2F}{\tilde{a}}}$. Note that the value corresponding to the mean field Nash equilibrium is

$$(4.12) \quad J_{\text{MFG}} = \frac{\ell}{\alpha^*} \left(a \left(\frac{\theta}{2} (\alpha^* - \lambda\alpha^*)^2 + \frac{1 - \theta}{2} (\alpha^*)^2 \right) + F \right) = \ell \frac{1 - \lambda\theta}{1 - \lambda^2\theta} \sqrt{2F\tilde{a}}.$$

Remark 4.3. Note that if $\theta = \lambda = 1$, then (4.10) implies that $\alpha^*(V) > V$, while an equilibrium corresponds to $V = \alpha^*(V)$. Thus, there is no mean field equilibrium if $\theta = \lambda = 1$.

Remark 4.4 (Comparison with the one-shot mean field type control problem). *The Nash equilibrium must be distinguished from the situation in which the drift of the agents is determined in order to minimize the global cost; the latter problem belongs to the class of mean field type control problems (MFTC for short) studied by A. Bensoussan and his collaborators, [6] and R. Carmona and F. Delarue, [9].*

In this case, the problem reads

$$\inf_{\mu} \int_{\alpha \in \mathbb{R}_+} \frac{\ell}{\alpha} \left(a \left(\frac{\theta}{2} \left(\alpha - \lambda \int_{\beta \in \mathbb{R}_+} \beta d\mu(\beta) \right)^2 + \frac{1 - \theta}{2} \alpha^2 \right) + F \right) d\mu(\alpha).$$

Looking for the solution as the Dirac mass $\delta_{\alpha^{**}}$, we see that α^{**} minimizes

$$\alpha \mapsto \frac{\ell}{\alpha} \left(a \left(\frac{\theta}{2} (\alpha - \lambda\alpha)^2 + \frac{1 - \theta}{2} \alpha^2 \right) + F \right),$$

thus

$$\alpha^{**} = \left(\frac{2F}{a(1 - 2\theta\lambda + \theta\lambda^2)} \right)^{\frac{1}{2}}.$$

The value of the problem is

$$J_{\text{MFTC}} = \ell \sqrt{2Fa(1 - \lambda\theta(2 - \lambda))} \leq \frac{\ell}{2} \sqrt{2Fa} \frac{1 - \lambda\theta(2 - \lambda)}{\sqrt{1 - \lambda^2\theta}} + \frac{\ell}{2} \sqrt{2Fa(1 - \lambda^2\theta)} = J_{\text{MFG}},$$

from Young's inequality $y \leq \frac{y^2}{2z} + \frac{z}{2}$, with $y = \sqrt{1 - \lambda\theta(2 - \lambda)}$ and $z = \sqrt{1 - \lambda^2\theta}$. The value of the MFTC is lower than the value of the MFG, which is natural since in the former case, the agents are collaborating to minimize the global cost. Moreover, the values of the two problems coincide only if $\lambda\theta = 0$, i.e. when there are no interactions through controls.

Comparison with the one-shot one-dimensional problem We have made numerical simulations of the MFG described in paragraph 4.2.1 with $f_0(x) = F = 1$ for all $x \in \Omega$, $a = \tilde{a}(1 - \lambda^2\theta)^{-1}$ and $\tilde{a} = 2$. We wish to compare the results with the explicit formulas obtained above for the one-shot one-dimensional MFG. If the latter problem is a good approximation of the former one, we should find α_1 close to $\sqrt{\frac{2F}{a}} = 1$, therefore mostly independent of λ and θ . Note that the approximation by the one-shot one-dimension mean field game is sensible only if the time remaining to the horizon is large enough, i.e. significantly larger than $\ell/\alpha^* = \ell$. If $T - t$ is small, then the optimal strategy for a representative player located on the left part of the domain is not to move.

On Figure 8, we display the norm of the optimal feedback for several values of λ and θ . We see that the optimal drift is close to 1 in agreement with the explicit formula found for the simplified one-shot one-dimensional MFG. We also see that the agents located at the bottom and top of Ω head toward the center of the domain (i.e. $x_2 = 0$) before reaching the exit. Thus, the second coordinate of α gets large and there are singularities on both sides of the exit, whose amplitude increases as $\lambda\theta$ tends to 1.

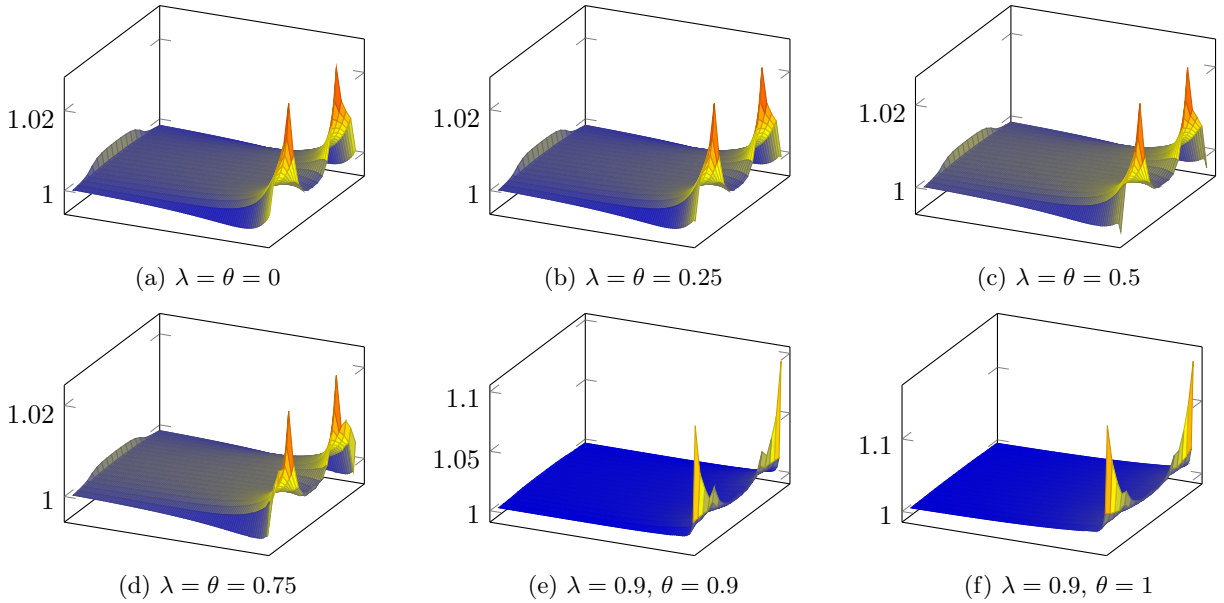


Figure 8: Example 2, $f_0(x) = 1$. The optimal feedback α for different choices of λ and θ at $t = T/2$.

4.2.3 Queues

Here, we choose f as follows:

$$f_0(x) = \begin{cases} 4, & \text{if } x \leq -0.1, \\ 1, & \text{if } x \geq 0.1, \\ 2.5 - 15x, & \text{otherwise;} \end{cases}$$

hence, an agent pays a cost for staying in Ω which is higher in the left part of the domain.

The parameters are

$$\nu = 0.001, \quad \tilde{a} = 2, \quad \lambda = 0.95, \quad \theta = 0.95, \quad \rho = 0.25, \quad \omega_0 = \frac{\pi}{3}.$$

The grid has 101×21 nodes and there are 101 time steps.

The evolution of the distribution of states and of the optimal feedback is displayed on Figure 9. We compare these results with a simulation of the MFG obtained by cancelling either θ or λ while keeping all the other parameters and the grid unchanged, see Figure 10. On Figure 9, we see that a

well distributed queue takes place from the entrance to the exit, by contrast with Figure 10 where we see that the agent rush and accumulate in the right part of the domain. Similarly, the deceleration is much stiffer in the latter case. It is not surprising that the interactions through controls have the effect of smoothing the distribution of states and the optimal feedback law. On the bottom of Figure 10, we see that when t is close to the horizon, the distribution is mainly concentrated near the middle of the domain but slightly on the right: this corresponds to the agents that have reached the zone where f_0 has the smaller value, i.e. 1, but for which reaching the exit before T becomes too costly. There is also a smaller bump near the entrance corresponding to agents that would pay too high a cost to reach the right part of the domain before T . These phenomena are clearly a side effects due to the finite horizon. These side effects are also present on Figure 9, but they are attenuated by the interaction through the controls.

Remark 4.5. *In models accounting for congestion effects, the cost of motion depends on the density of the distribution of states and gets higher in the crowded regions. We refer to [20] for a pioneering discussion of MFG models including congestion, to [5] for the analysis of the system of PDEs that arise with these models, and to [4] for numerical simulations. Such models also permit to describe queueing phenomena, because the agents located in crowded regions pay a large cost for moving. In [4], since the congestion effects are taken into account in a local manner, the queues take place in small regions of the state domain. By contrast, the present nonlocal model accounts for the fact that the agents anticipate low speed regions, making the traffic more fluid and the distribution of state smoother. Although it is quite possible to do, we have not incorporated congestion effects in the present model. We plan to do it in forthcoming works.*

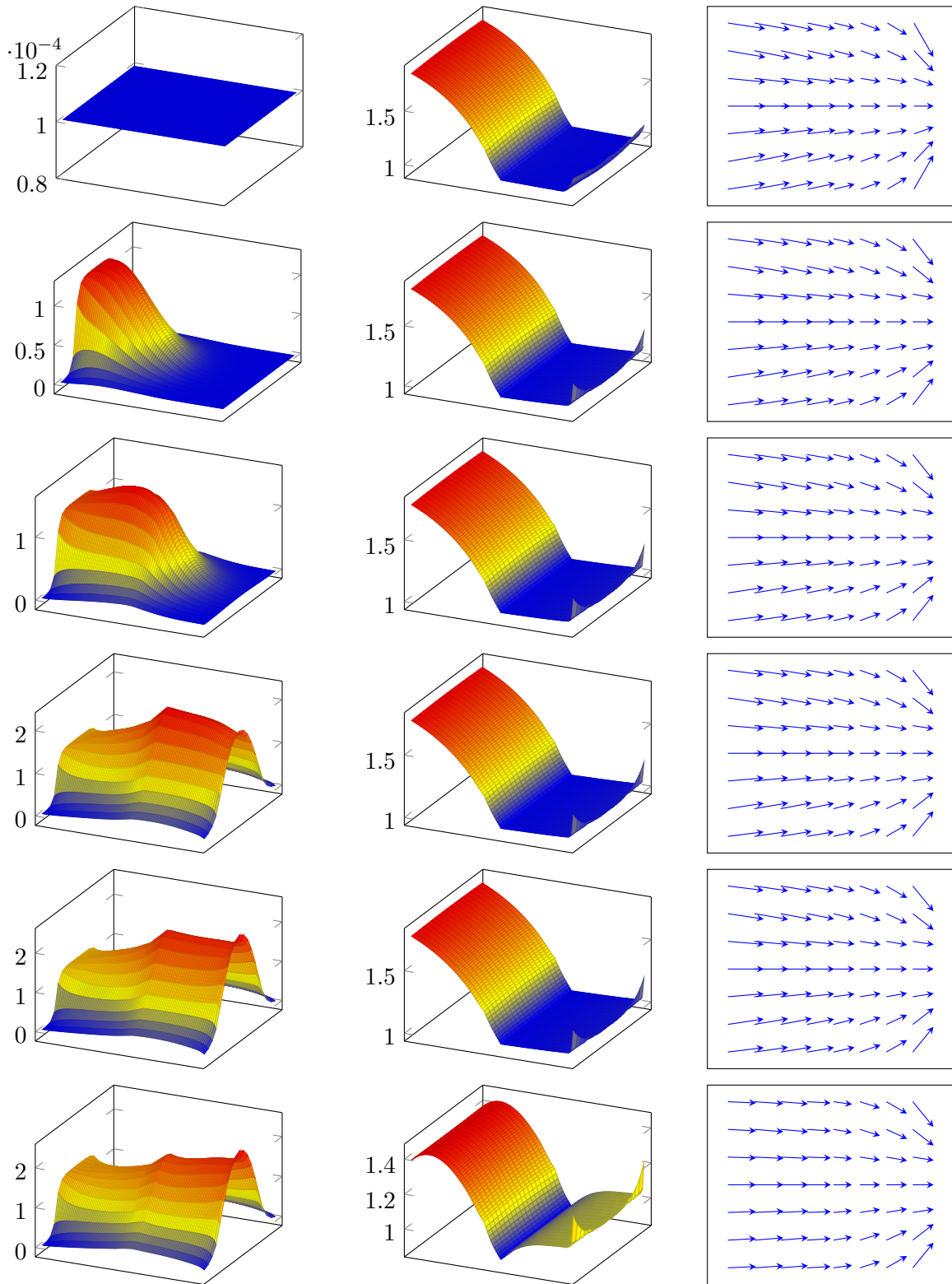


Figure 9: Example 2. Interaction via controls: snapshots at $t = 0, 0.4, 0.8, 2, 4, 7$. Left: the distribution of states. Center: the norm of the optimal feedback. Right: the optimal feedback.

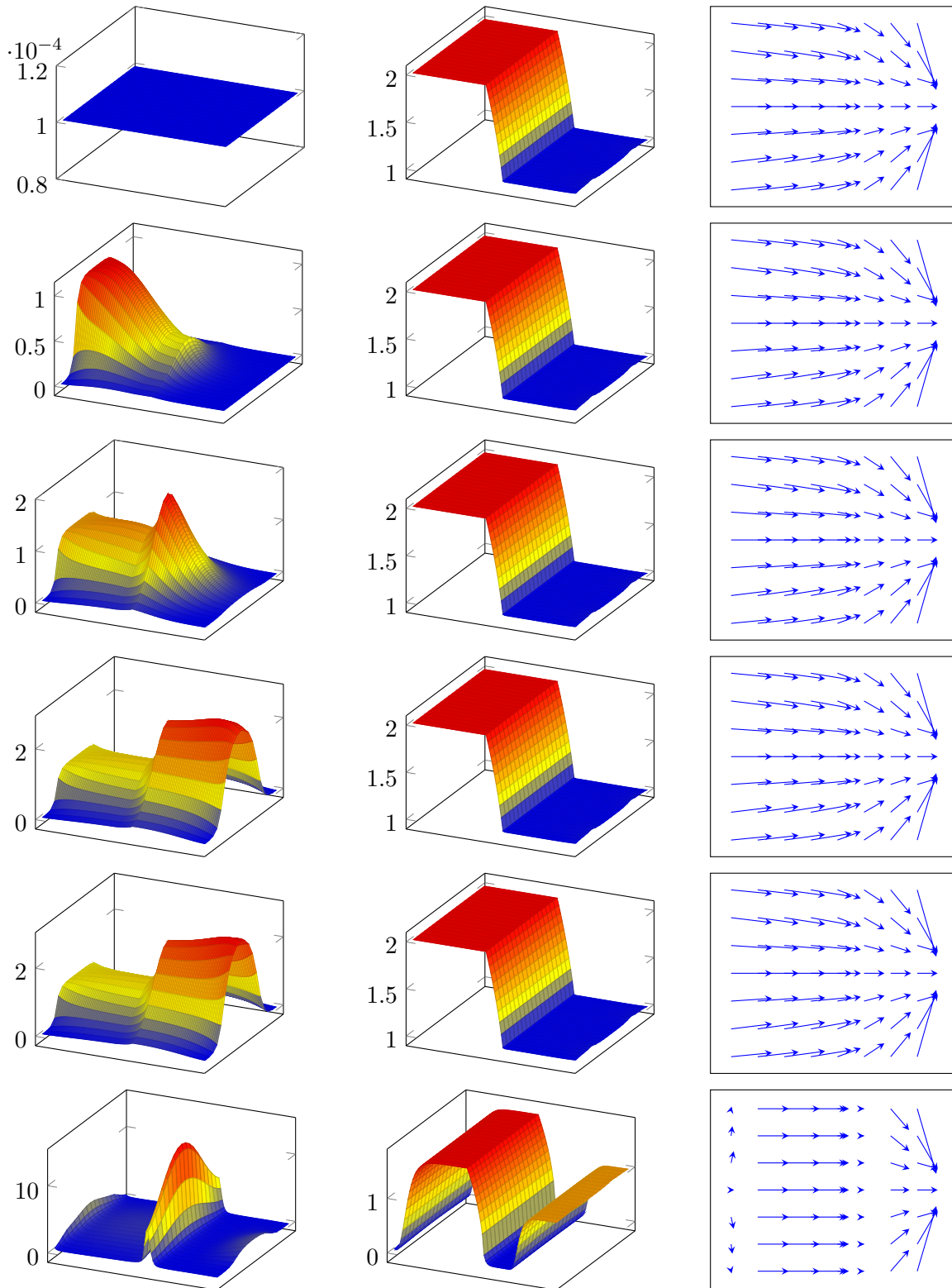


Figure 10: Example 2. Same parameters except $\lambda\theta = 0$: snapshots at $t = 0, 0.4, 0.8, 2, 4, 7$. Left: the distribution of states. Center: the norm of the optimal feedback. Right: the optimal feedback.

4.2.4 Stationary regime

We now look for a stationary equilibrium. For that, we use an iterative method in order to progressively diminish the effects of the initial and terminal conditions: starting from (u^0, m^0, V^0) , the

numerical solution of the finite horizon problem described above, we construct a sequence of approximate solutions $(u^\ell, m^\ell, V^\ell)_{\ell \geq 1}$ by the following induction: $(u^{\ell+1}, m^{\ell+1}, V^{\ell+1})$ is the solution of the finite horizon problem with the same system of PDEs in $(0, T) \times \Omega$, the same boundary conditions on $(0, T) \times \partial\Omega$, and the new initial and terminal conditions as follows:

$$(4.13) \quad u^{\ell+1}(T, x) = u^\ell\left(\frac{T}{2}, x\right), \quad x \in \Omega,$$

$$(4.14) \quad m^{\ell+1}(0, x) = m^\ell\left(\frac{T}{2}, x\right), \quad x \in \Omega.$$

As ℓ tends to $+\infty$, we observe that (u^ℓ, m^ℓ, V^ℓ) converge to time-independent functions. At the limit, we obtain a steady solution of

$$\begin{aligned} -\nu\Delta u + \frac{a}{2} \left| \frac{1}{a} \nabla_x u - \lambda\theta V \right|^2 - a \frac{\lambda^2 \theta}{2} |V|^2 &= cm + f_0(x), & \text{in } \Omega, \\ -\nu\Delta m - \operatorname{div} \left(\left(\frac{1}{a} \nabla_x u - \lambda\theta V \right) m \right) &= 0, & \text{in } \Omega, \\ V(x) = -\frac{1}{Z(x)} \int_{\Omega} \left(\frac{1}{a} \nabla_x u(y) - \lambda\theta V(y) \right) K(x, y) dm(y), & \text{in } \Omega, \\ Z(x) &= \int_{\Omega} K(x, y) dm(y), & \text{in } \Omega, \end{aligned}$$

with the boundary conditions on $\partial\Omega$ given by (4.3)-(4.5).

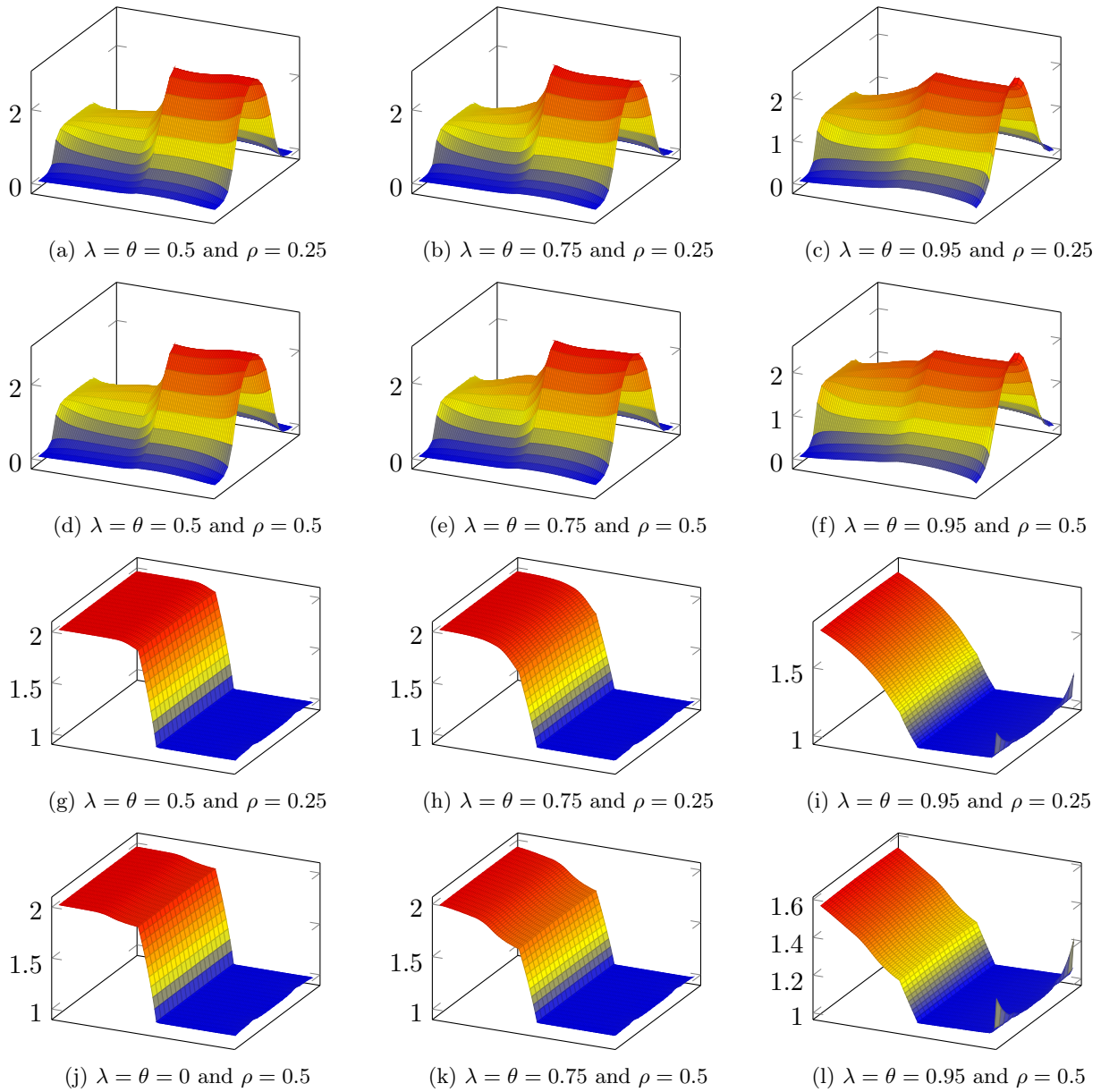


Figure 11

Example 2. Solutions of the stationary problem for different choices of λ, θ and ρ . Top six figures: the distribution of states m . Bottom six figures: norm of the optimal controls.

Acknowledgement 1. *This research was partially supported by the ANR (Agence Nationale de la Recherche) through MFG project ANR-16-CE40-0015-01.*

4 - REFERENCES

- [1] UMFPACK. <http://www.cise.ufl.edu/research/sparse/umfpack/current/>.
- [2] Yves Achdou. Finite difference methods for mean field games. In *Hamilton-Jacobi equations: approximations, numerical analysis and applications*, volume 2074 of *Lecture Notes in Math.*, pages 1–47. Springer, Heidelberg, 2013.
- [3] Yves Achdou and Italo Capuzzo-Dolcetta. Mean field games: numerical methods. *SIAM J. Numer. Anal.*, 48(3):1136–1162, 2010.

- [4] Yves Achdou and Jean-Michel Lasry. Mean field games for modeling crowd motion. In *Contributions to partial differential equations and applications*, volume 47 of *Comput. Methods Appl. Sci.*, pages 17–42. Springer, Cham, 2019.
- [5] Yves Achdou and Alessio Porretta. Mean field games with congestion. *Ann. Inst. H. Poincaré Anal. Non Linéaire*, 35(2):443–480, 2018.
- [6] Alain Bensoussan, Jens Frehse, and Phillip Yam. *Mean field games and mean field type control theory*. SpringerBriefs in Mathematics. Springer, New York, 2013.
- [7] Frédéric J. Bonnans, Saeed Hadikhanloo, and Laurent Pfeiffer. Schauder Estimates for a Class of Potential Mean Field Games of Controls. *arXiv e-prints*, page arXiv:1902.05461, Feb 2019.
- [8] Pierre Cardaliaguet and Charles-Albert Lehalle. Mean field game of controls and an application to trade crowding. *Math. Financ. Econ.*, 12(3):335–363, 2018.
- [9] René Carmona and François Delarue. Forward-backward stochastic differential equations and controlled McKean-Vlasov dynamics. *Ann. Probab.*, 43(5):2647–2700, 2015.
- [10] René Carmona and François Delarue. *Probabilistic theory of mean field games with applications. I*, volume 83 of *Probability Theory and Stochastic Modelling*. Springer, Cham, 2018. Mean field FBSDEs, control, and games.
- [11] René Carmona and Daniel Lacker. A probabilistic weak formulation of mean field games and applications. *Ann. Appl. Probab.*, 25(3):1189–1231, 06 2015.
- [12] D. A. Gomes and V. K. Voskanyan. Extended mean field games. *Izv. Nats. Akad. Nauk Armenii Mat.*, 48(2):63–76, 2013.
- [13] Diogo A. Gomes, Stefania Patrizi, and Vardan Voskanyan. On the existence of classical solutions for stationary extended mean field games. *Nonlinear Anal.*, 99:49–79, 2014.
- [14] Minyi Huang, Peter E. Caines, and Roland P. Malhamé. Large-population cost-coupled LQG problems with nonuniform agents: individual-mass behavior and decentralized ϵ -Nash equilibria. *IEEE Trans. Automat. Control*, 52(9):1560–1571, 2007.
- [15] Minyi Huang, Roland P. Malhamé, and Peter E. Caines. Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Commun. Inf. Syst.*, 6(3):221–251, 2006.
- [16] Z Kobeissi. On Classical Solutions to the Mean Field Game System of Controls. *arXiv e-prints*, page arXiv:1904.11292, Apr 2019.
- [17] Jean-Michel Lasry and Pierre-Louis Lions. Jeux à champ moyen. I. Le cas stationnaire. *C. R. Math. Acad. Sci. Paris*, 343(9):619–625, 2006.
- [18] Jean-Michel Lasry and Pierre-Louis Lions. Jeux à champ moyen. II. Horizon fini et contrôle optimal. *C. R. Math. Acad. Sci. Paris*, 343(10):679–684, 2006.
- [19] Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. *Jpn. J. Math.*, 2(1):229–260, 2007.
- [20] Pierre-Louis Lions. Théorie des jeux à champs moyen. *video lecture series at Collège de France*, 2011-2019. <https://www.college-de-france.fr/site/pierre-louis-lions/index.htm>.
- [21] H. A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 13(2):631–644, 1992.