Numerical Methods for Mean Field Games

Lecture 3
Classical Numerical Methods: Part II
FBPDE and FBSDE systems

Mathieu LAURIÈRE

New York University Shanghai

UM6P Vanguard Center, Université Cadi AYYAD,
University Côte d'Azur, & GE2MI
Open Doctoral Lectures
July 5 – 7, 2023

Outline

Reminder: FB systems

- Here we will focus on the continuous time and space setting
- We have seen two types of forward-backward systems:
  - PDE systems: Kolmogorov-Fokker-Planck (KFP) and Hamilton-Jacobi-Bellman (HJB)
  - SDE systems of McKean-Vlasov (MKV) type
- We will describe methods based on both approaches
- In each case, there will be two questions to design a numerical method:
  - Discretization $\rightarrow$ numerical scheme
  - Computation $\rightarrow$ algorithm

MFG PDE System

Goal: (1) introduce and **(2)** solve a discrete version of the MFG PDE system:

$$
\begin{cases}
0 = -\dfrac{\partial u}{\partial t}(t,x) - \nu \Delta u(t,x) + H(x, m(t,\cdot), \nabla u(t,x)), \\
0 = \dfrac{\partial m}{\partial t}(t,x) - \nu \Delta m(t,x) - \mathrm{div}\left(m(t,\cdot)\partial_p H(\cdot, m(t), \nabla u(t,\cdot))\right)(x), \\
u(T,x) = g(x, m(T,\cdot)), \qquad m(0,x) = m_0(x)
\end{cases}
$$

MFG PDE System

Goal: **(1)** introduce and **(2)** solve a discrete version of the MFG PDE system:

$$\begin{cases} 0 = -\dfrac{\partial u}{\partial t}(t,x) - \nu \Delta u(t,x) + H(x, m(t,\cdot), \nabla u(t,x)), \\ 0 = \dfrac{\partial m}{\partial t}(t,x) - \nu \Delta m(t,x) - \operatorname{div}\left(m(t,\cdot)\partial_p H(\cdot, m(t), \nabla u(t,\cdot))\right)(x), \\ \ u(T,x) = g(x, m(T,\cdot)), \qquad m(0,x) = m_0(x) \end{cases}$$

Desirable properties for (1):

- **Mass** and **positivity** of distribution: $\int_{\mathcal{X}} m(t,x)dx = 1$, $m \geq 0$
- **Convergence** of discrete solution to continuous solution as mesh step $\to 0$

MFG PDE System

Goal: (1) introduce and **(2)** solve a discrete version of the MFG PDE system:

$$\begin{cases} 0 = -\dfrac{\partial u}{\partial t}(t,x) - \nu \Delta u(t,x) + H(x, m(t,\cdot), \nabla u(t,x)), \\ 0 = \dfrac{\partial m}{\partial t}(t,x) - \nu \Delta m(t,x) - \operatorname{div}\left(m(t,\cdot)\partial_p H(\cdot, m(t), \nabla u(t,\cdot))\right)(x), \\ \;u(T,x) = g(x, m(T,\cdot)), \qquad m(0,x) = m_0(x) \end{cases}$$

Desirable properties for (1):

- **Mass** and **positivity** of distribution: $\int_{\mathcal{X}} m(t,x)dx = 1$, $m \geq 0$
- **Convergence** of discrete solution to continuous solution as mesh step $\to 0$
- The KFP equation is the **adjoint** of the linearized HJB equation
- Link with **optimality condition** of a discrete problem

$\Rightarrow$ Needs a careful discretization

MFG PDE System

Goal: (1) introduce and **(2)** solve a discrete version of the MFG PDE system:

$$\begin{cases} 0 = -\dfrac{\partial u}{\partial t}(t,x) - \nu \Delta u(t,x) + H(x, m(t,\cdot), \nabla u(t,x)), \\ 0 = \dfrac{\partial m}{\partial t}(t,x) - \nu \Delta m(t,x) - \mathrm{div}\left(m(t,\cdot)\partial_p H(\cdot, m(t), \nabla u(t,\cdot))\right)(x), \\ \quad u(T,x) = g(x, m(T,\cdot)), \qquad m(0,x) = m_0(x) \end{cases}$$

Desirable properties for (1):

- **Mass** and **positivity** of distribution: $\int_{\mathcal{X}} m(t,x)dx = 1$, $m \geq 0$
- **Convergence** of discrete solution to continuous solution as mesh step $\to 0$
- The KFP equation is the **adjoint** of the linearized HJB equation
- Link with **optimality condition** of a discrete problem

$\Rightarrow$ Needs a careful discretization

For (2): Once we have a discrete system, how can we compute its solution?

Outline

Outline

Discretization

Semi-implicit finite difference scheme from [Achdou and Capuzzo-Dolcetta, 2010]
Discretization:

- For simplicity we consider the domain $\mathbb{T}$ = one-dimensional (unit) torus.
- Let $\nu = \sigma^2/2$.
- We consider N_h and N_T steps respectively in space and time.
- Let $h = 1/N_h$ and $\Delta t = T/N_T$. Let $\mathbb{T}_h$ = discretized torus.
- We approximate $m_0(x_i)$ by ρ_i^0 such that $h\sum_i \rho_i^0 = 1$.

Discretization

Semi-implicit finite difference scheme from [Achdou and Capuzzo-Dolcetta, 2010]
Discretization:

- For simplicity we consider the domain $\mathbb{T} =$ one-dimensional (unit) torus.
- Let $\nu = \sigma^2/2$.
- We consider N_h and N_T steps respectively in space and time.
- Let $h = 1/N_h$ and $\Delta t = T/N_T$. Let $\mathbb{T}_h =$ discretized torus.
- We approximate $m_0(x_i)$ by ρ_i^0 such that $h\sum_i \rho_i^0 = 1$.

Then we introduce the following **discrete operators** : for $\varphi \in \mathbb{R}^{N_T+1}$ and $\psi \in \mathbb{R}^{N_h}$

- **time derivative** : $$(D_t\varphi)^n := \frac{\varphi^{n+1} - \varphi^n}{\Delta t}, \qquad 0 \le n \le N_T - 1$$
- **Laplacian** : $$(\Delta_h \psi)_i := -\frac{1}{h^2}\left(2\psi_i - \psi_{i+1} - \psi_{i-1}\right), \qquad 0 \le i \le N_h$$
- **partial derivative** : $$(D_h\psi)_i := \frac{\psi_{i+1} - \psi_i}{h}, \qquad 0 \le i \le N_h$$
- **gradient** : $$[\nabla_h \psi]_i := \left((D_h\psi)_i, (D_h\psi)_{i-1}\right), \qquad 0 \le i \le N_h$$

Discrete Hamiltonian

For simplicity, we assume that the drift b and the costs f and g are of the form

$$b(x, m, \alpha) = \alpha, \qquad f(x, m, \alpha) = L(x, \alpha) + \mathtt{f}_0(x, m), \qquad g(x, m) = \mathtt{g}_0(x, m).$$

where $x \in \mathbb{R}^d$, $\alpha \in \mathbb{R}^d$, $m \in \mathbb{R}_+$. Then

$$H(x, m, p) = \max_{\alpha} \{-L(x, \alpha) - \langle \alpha, p \rangle\} - \mathtt{f}_0(x, m) = H_0(x, p) - \mathtt{f}_0(x, m)$$

where H_0 is the convex conjugate (also denoted L^*) of L with respect to α:

$$H_0(x, p) = L^*(x, p) = \sup_{\alpha} \{\langle \alpha, p \rangle - L(x, \alpha)\}$$

Discrete Hamiltonian

For simplicity, we assume that the drift b and the costs f and g are of the form

$$b(x, m, \alpha) = \alpha, \qquad f(x, m, \alpha) = L(x, \alpha) + \mathtt{f}_0(x, m), \qquad g(x, m) = \mathtt{g}_0(x, m).$$

where $x \in \mathbb{R}^d$, $\alpha \in \mathbb{R}^d$, $m \in \mathbb{R}_+$. Then

$$H(x, m, p) = \max_{\alpha} \{-L(x, \alpha) - \langle \alpha, p \rangle\} - \mathtt{f}_0(x, m) = H_0(x, p) - \mathtt{f}_0(x, m)$$

where H_0 is the convex conjugate (also denoted L^*) of L with respect to α:

$$H_0(x, p) = L^*(x, p) = \sup_{\alpha} \{\langle \alpha, p \rangle - L(x, \alpha)\}$$

Discrete Hamiltonian: $(x, p_1, p_2) \mapsto \tilde{H}_0(x, p_1, p_2)$ satisfying:

- Monotonicity: decreasing w.r.t. p_1 and increasing w.r.t. p_2
- Consistency with H_0: for every x, p, $\tilde{H}_0(x, p, p) = H_0(x, p)$
- Differentiability: for every x, $(p_1, p_2) \mapsto \tilde{H}_0(x, p_1, p_2)$ is $\mathcal{C}^1$
- Convexity: for every x, $(p_1, p_2) \mapsto \tilde{H}_0(x, p_1, p_2)$ is convex

Example: if $H_0(x, p) = |p|^2$, a possible choice is $\tilde{H}_0(x, p_1, p_2) = ({p_1}^-)^2 + ({p_2}^+)^2$

Discrete HJB

Discrete solution: We replace $u, m : [0,T] \times \mathbb{T} \to \mathbb{R}$ by vectors

$$U, M \in \mathbb{R}^{(N_T+1)\times N_h}$$

Discrete HJB

Discrete solution: We replace $u, m : [0,T] \times \mathbb{T} \to \mathbb{R}$ by vectors

$$U, M \in \mathbb{R}^{(N_T+1)\times N_h}$$

The **HJB equation**

$$\begin{cases} \partial_t u(t,x) + \nu \Delta u(t,x) + H_0(x, \nabla u(t,x)) = \mathtt{f}_0(x, m(t,x)) \\ u(T,x) = \mathtt{g}_0(x, m(T,x)) \end{cases}$$

is discretized as:

$$\begin{cases} -(D_t U_i)^n - \nu(\Delta_h U^n)_i + \tilde{H}_0(x_i, [D_h U^n]_i) = \mathtt{f}_0(x_i, M_i^{n+1}) \\ U_i^{N_T} = \mathtt{g}_0(x_i, M_i^{N_T}) \end{cases}$$

Discrete KFP

The **KFP equation**

$$\partial_t m(t,x) - \nu\Delta m(t,x) + \mathrm{div}\Big(m(t,x)\partial_q H(x, m(t), \nabla u(t,x))\Big) = 0, \qquad m(0,x) = m_0(x)$$

is discretized as

$$(D_t M_i)^n - \nu(\Delta_h M^{n+1})_i - \mathcal{T}_i(U^n, M^{n+1}) = 0, \qquad M_i^0 = \rho_i^0$$

Discrete KFP

The **KFP equation**

$$\partial_t m(t,x) - \nu \Delta m(t,x) + \operatorname{div}\Big(m(t,x) \partial_q H(x, m(t), \nabla u(t,x)) \Big) = 0, \qquad m(0,x) = m_0(x)$$

is discretized as

$$(D_t M_i)^n - \nu(\Delta_h M^{n+1})_i - \mathcal{T}_i(U^n, M^{n+1}) = 0, \qquad M_i^0 = \rho_i^0$$

Here we use the **discrete transport operator** $\approx -\operatorname{div}(\dots)$

$$\mathcal{T}_i(U,M) := \frac{1}{h} \begin{pmatrix} M_i \partial_{p_1} \tilde{H}_0(x_i, [\nabla_h U]_i) - M_{i-1} \partial_{p_1} \tilde{H}_0(x_{i-1}, [\nabla_h U]_{i-1}) \\ + M_{i+1} \partial_{p_2} \tilde{H}_0(x_{i+1}, [\nabla_h U]_{i+1}) - M_i \partial_{p_2} \tilde{H}_0(x_i, [\nabla_h U]_i) \end{pmatrix}$$

Discrete KFP

The **KFP equation**

$$\partial_t m(t,x) - \nu \Delta m(t,x) + \mathrm{div}\Big(m(t,x)\partial_q H(x, m(t), \nabla u(t,x))\Big) = 0, \qquad m(0,x) = m_0(x)$$

is discretized as

$$(D_t M_i)^n - \nu(\Delta_h M^{n+1})_i - \mathcal{T}_i(U^n, M^{n+1}) = 0, \qquad M_i^0 = \rho_i^0$$

Here we use the **discrete transport operator** $\approx -\mathrm{div}(\dots)$

$$\mathcal{T}_i(U,M) := \frac{1}{h}\left(\begin{array}{l} M_i \partial_{p_1} \tilde{H}_0(x_i, [\nabla_h U]_i) - M_{i-1}\partial_{p_1}\tilde{H}_0(x_{i-1}, [\nabla_h U]_{i-1}) \\ \quad + M_{i+1}\partial_{p_2}\tilde{H}_0(x_{i+1}, [\nabla_h U]_{i+1}) - M_i \partial_{p_2}\tilde{H}_0(x_i, [\nabla_h U]_i) \end{array} \right)$$

Intuition: weak formulation & integration by parts

$$\int_{\mathbb{T}} \mathrm{div}\,(m\partial_p H_0(x, \nabla u)) w = -\int_{\mathbb{T}} m \partial_p H_0(x, \nabla u) \cdot \nabla w$$

is discretized as

$$-h \sum_i \mathcal{T}_i(U,M) W_i = h \sum_i M_i \nabla_q \tilde{H}_0(x_i, [\nabla_h U]_i) \cdot [\nabla_h W]_i$$

Discrete System – Properties

Discrete forward-backward system:

$$\begin{cases} -(D_t U_i)^n - \nu(\Delta_h U^n)_i + \tilde{H}_0(x_i, [D_h U^n]_i) = \mathtt{f}_0(x_i, M_i^{n+1}), & \forall n \leq N_T - 1 \\ (D_t M_i)^n - \nu(\Delta_h M^{n+1})_i - \mathcal{T}_i(U^n, M^{n+1}) = 0, & \forall n \leq N_T - 1 \\ M_i^0 = \rho_i^0, \qquad U_i^{N_T} = \mathtt{g}_0(x_i, M_i^{N_T}), \qquad i = 0, \dots, N_h & \end{cases}$$

Discrete System – Properties

Discrete forward-backward system:

$$
\begin{cases}
-(D_t U_i)^n - \nu(\Delta_h U^n)_i + \tilde{H}_0(x_i, [D_h U^n]_i) = \mathtt{f}_0(x_i, M_i^{n+1}), & \forall n \leq N_T - 1 \\
(D_t M_i)^n - \nu(\Delta_h M^{n+1})_i - \mathcal{T}_i(U^n, M^{n+1}) = 0, & \forall n \leq N_T - 1 \\
M_i^0 = \rho_i^0, \qquad U_i^{N_T} = \mathtt{g}_0(x_i, M_i^{N_T}), \qquad i = 0, \dots, N_h
\end{cases}
$$

This scheme enjoys many nice properties, among which:

- It yields a monotone scheme for the KFP equation: **mass** and **positivity** are preserved
- **Convergence** to classical solution if monotonicity [Achdou and Capuzzo-Dolcetta, 2010, Achdou et al., 2012]
- Can sometimes be used to show existence of a **weak** solution [Achdou and Porretta, 2016]
- The discrete KFP operator is the **adjoint** of the linearized Bellman operator
- **Existence** and **uniqueness** result for the discrete system
- It corresponds to the **optimality condition** of a discrete optimization problem (details later)

Outline

Algo 1: Fixed Point Iterations

Input: Initial guess $(\tilde{M}, \tilde{U})$; damping $\delta(\cdot)$; number of iterations $\mathtt{K}$

Output: Approximation of $(\hat{M}, \hat{U})$ solving the finite difference system

Initialize $M^{(0)} = \tilde{M}^{(0)} = \tilde{M}, U^{(0)} = \tilde{U}$

for $\mathtt{k} = 0, 1, 2, \ldots, \mathtt{K} - 1$ **do**

Let $U^{(\mathtt{k}+1)}$ be the solution to:

$$\begin{cases} -(D_t U_i)^n - \nu(\Delta_h U^n)_i + \tilde{H}_0(x_i, [D_h U^n]_i) = \mathtt{f}_0(x_i, \tilde{M}_i^{(\mathtt{k}), n+1}), \quad n \leq N_T - 1 \\ U_i^{N_T} = \mathtt{g}_0(x_i, \tilde{M}_i^{(\mathtt{k}), N_T}) \end{cases}$$

Let $M^{(\mathtt{k}+1)}$ be the solution to:

$$\begin{cases} (D_t M_i)^n - \nu(\Delta_h M^{n+1})_i - \mathcal{T}_i(U^{(\mathtt{k}+1), n}, M^{n+1}) = 0, \quad n \leq N_T - 1 \\ M_i^0 = \rho_i^0 \end{cases}$$

Let $\tilde{M}^{(\mathtt{k}+1)} = \delta(\mathtt{k}) \tilde{M}^{(\mathtt{k})} + (1 - \delta(\mathtt{k})) M^{(\mathtt{k}+1)}$

return $(M^{(\mathtt{K})}, U^{(\mathtt{K})})$

Algo 1: Fixed Point Iterations

The HJB equation is **non-linear**

- **Idea 1:** replace $\tilde{H}_0(x_i, [D_h U^n]_i)$ by $\tilde{H}_0(x_i, [D_h U^{(\mathtt{k}),n}]_i)$

Algo 1: Fixed Point Iterations

The HJB equation is **non-linear**

- **Idea 1:** replace $\tilde{H}_0(x_i, [D_h U^n]_i)$ by $\tilde{H}_0(x_i, [D_h U^{(\mathtt{k}),n}]_i)$
- **Idea 2:** use non linear solver to find a zero of

$$\varphi : \mathbb{R}^{N_h \times (N_T+1)} \to \mathbb{R}^{N_h \times N_T},$$

with:

$$\varphi(U) = \Big(-(D_t U_i)^n - \nu(\Delta_h U^n)_i + \tilde{H}_0(x_i, [D_h U^n]_i) - \mathtt{f}_0(x_i, \tilde{M}_i^{(\mathtt{k}),n+1})\Big)_{i=0,\dots,N_h-1}^{n=0,\dots,N_T-1}$$

Example: Newton's method

Sample code

Code

Sample code to illustrate: IPython notebook

`https://colab.research.google.com/drive/1shJWSD2MA5Fo7_rB625dAvNTdZS1a7bG?usp=sharing`

- Finite difference scheme
- Solved by (damped) fixed point approach

Algo 2: Newton's Method for FD System

Idea: Directly look for a zero of $\varphi = (\varphi_{\mathcal{U}}, \varphi_{\mathcal{M}})^\top$ with $\varphi_{\mathcal{U}}$ and $\varphi_{\mathcal{M}}$ s.t.

$$\begin{cases} \varphi_{\mathcal{U}}(U, M) = 0 & \Leftrightarrow \ (U, M) \text{ solves discrete HJB equation} \\ \varphi_{\mathcal{M}}(U, M) = 0 & \Leftrightarrow \ (U, M) \text{ solves discrete KFP equation} \end{cases}$$

- Let $X^{(k)} = (U^{(k)}, M^{(k)})^\top$
- Iterate: $X^{(k+1)} = X^{(k)} - J_\varphi(X^{(k)})^{-1} \varphi(X^{(k)})$

Algo 2: Newton's Method for FD System

Idea: Directly look for a zero of $\varphi = (\varphi_{\mathcal{U}}, \varphi_{\mathcal{M}})^\top$ with $\varphi_{\mathcal{U}}$ and $\varphi_{\mathcal{M}}$ s.t.

$$\begin{cases} \varphi_{\mathcal{U}}(U, M) = 0 & \Leftrightarrow \ (U, M) \text{ solves discrete HJB equation} \\ \varphi_{\mathcal{M}}(U, M) = 0 & \Leftrightarrow \ (U, M) \text{ solves discrete KFP equation} \end{cases}$$

- Let $X^{(k)} = (U^{(k)}, M^{(k)})^\top$
- Iterate: $X^{(k+1)} = X^{(k)} - J_\varphi(X^{(k)})^{-1}\varphi(X^{(k)})$
- Or rather: $J_\varphi(X^{(k)})Y = -\varphi(X^{(k)})$, then $X^{(k+1)} = Y + X^{(k)}$

Algo 2: Newton's Method for FD System

Idea: Directly look for a zero of $\varphi = (\varphi_{\mathcal{U}}, \varphi_{\mathcal{M}})^\top$ with $\varphi_{\mathcal{U}}$ and $\varphi_{\mathcal{M}}$ s.t.

$$\begin{cases} \varphi_{\mathcal{U}}(U, M) = 0 & \Leftrightarrow \ (U, M) \text{ solves discrete HJB equation} \\ \varphi_{\mathcal{M}}(U, M) = 0 & \Leftrightarrow \ (U, M) \text{ solves discrete KFP equation} \end{cases}$$

- Let $X^{(k)} = (U^{(k)}, M^{(k)})^\top$
- Iterate: $X^{(k+1)} = X^{(k)} - J_\varphi(X^{(k)})^{-1} \varphi(X^{(k)})$
- Or rather: $J_\varphi(X^{(k)}) Y = -\varphi(X^{(k)})$, then $X^{(k+1)} = Y + X^{(k)}$

Key step: Solve a linear system of the form

$$\begin{pmatrix} A_{\mathcal{U},\mathcal{U}} & A_{\mathcal{U},\mathcal{M}} \\ A_{\mathcal{M},\mathcal{U}} & A_{\mathcal{M},\mathcal{M}} \end{pmatrix} \begin{pmatrix} U \\ M \end{pmatrix} = \begin{pmatrix} G_{\mathcal{U}} \\ G_{\mathcal{M}} \end{pmatrix}$$

where $A_{\mathcal{U},\mathcal{M}}(U, M) = \nabla_U \varphi_{\mathcal{M}}(U, M), \quad A_{\mathcal{U},\mathcal{U}}(U, M) = \nabla_U \varphi_{\mathcal{U}}(U, M), \quad \dots$

Newton Method – Implementation

Linear system to be solved: $\begin{pmatrix} A_{\mathcal{U},\mathcal{U}} & A_{\mathcal{U},\mathcal{M}} \\ A_{\mathcal{M},\mathcal{U}} & A_{\mathcal{M},\mathcal{M}} \end{pmatrix} \begin{pmatrix} U \\ M \end{pmatrix} = \begin{pmatrix} G_{\mathcal{U}} \\ G_{\mathcal{M}} \end{pmatrix}$

Structure: $A_{\mathcal{U},\mathcal{M}}, A_{\mathcal{M},\mathcal{U}}$ are block-diagonal, $A_{\mathcal{U},\mathcal{U}} = A^{\top}_{\mathcal{M},\mathcal{M}}$, and

$$A_{\mathcal{U},\mathcal{U}} = \begin{pmatrix} D_1 & 0 & \dots & \dots & 0 \\ -\frac{1}{\Delta t}\mathrm{Id}_{N_h} & D_2 & \ddots & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & -\frac{1}{\Delta t}\mathrm{Id}_{N_h} & D_{N_T} \end{pmatrix}$$

where D_n corresponds to the discrete operator

$$Z = (Z_{i,j})_{i,j} \mapsto \left(\frac{1}{\Delta t} Z_{i,j} - \nu(\Delta_h Z)_{i,j} + [\nabla_h Z]_{i,j} \cdot \nabla_p \tilde{H}_0(x_{i,j}, [\nabla_h U^{(k),n}]_{i,j}) \right)_{i,j}$$

Newton Method – Implementation

Linear system to be solved: $\begin{pmatrix} A_{\mathcal{U},\mathcal{U}} & A_{\mathcal{U},\mathcal{M}} \\ A_{\mathcal{M},\mathcal{U}} & A_{\mathcal{M},\mathcal{M}} \end{pmatrix} \begin{pmatrix} U \\ M \end{pmatrix} = \begin{pmatrix} G_{\mathcal{U}} \\ G_{\mathcal{M}} \end{pmatrix}$

Structure: $A_{\mathcal{U},\mathcal{M}}, A_{\mathcal{M},\mathcal{U}}$ are block-diagonal, $A_{\mathcal{U},\mathcal{U}} = A_{\mathcal{M},\mathcal{M}}^\top$, and

$$A_{\mathcal{U},\mathcal{U}} = \begin{pmatrix} D_1 & 0 & \cdots & \cdots & 0 \\ -\frac{1}{\Delta t}\mathrm{Id}_{N_h} & D_2 & \ddots & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & -\frac{1}{\Delta t}\mathrm{Id}_{N_h} & D_{N_T} \end{pmatrix}$$

where D_n corresponds to the discrete operator

$$Z = (Z_{i,j})_{i,j} \mapsto \left(\frac{1}{\Delta t} Z_{i,j} - \nu(\Delta_h Z)_{i,j} + [\nabla_h Z]_{i,j} \cdot \nabla_p \tilde{H}_0(x_{i,j}, [\nabla_h U^{(k),n}]_{i,j}) \right)_{i,j}$$

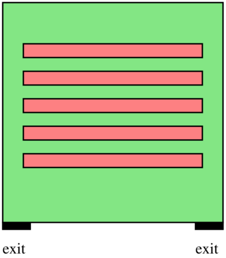
Rem. Initial guess $(U^{(0)}, M^{(0)})$ is important for Newton's method

- Idea 1: initialize with the ergodic solution (see e.g., [Achdou et al., 2021])
- Idea 2: continuation method w.r.t. ν (converges more easily with a large viscosity)

See [Achdou, 2013] for more details.

Example: Exit of a Room – Distribution

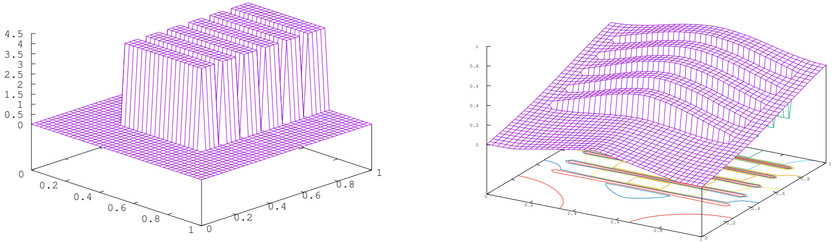
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2020]



Geometry of the room

Example: Exit of a Room – Distribution

Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2020]



Initial density (left) and final cost (right)

Example: Exit of a Room – Crowd model

- Crowd motion with ocal interactions; see *e.g.* [Lachapelle and Wolfram, 2011, Achdou and Lasry, 2019, Achdou and Porretta, 2018, Achdou and Laurière, 2016a] for other models of this type and [Aurell and Djehiche, 2018, Achdou and Laurière, 2015] for crowd motion models with non-local interactions.

- Here, control = velocity:

$$dX_t = \alpha(t, X_t)dt + \sigma dW_t$$

- Congestion through the cost: higher density $\Rightarrow$ higher price to move

- Hamiltonian:

$$H(x, m, p) = \frac{8|p|^2}{(1+m)^{\frac{3}{4}}} - \frac{1}{3200}\,.$$

Exercise

What is the cost function leading to this Hamiltonian?

Example: Exit of a Room – Crowd model

- MFG PDE system:
  1. Mean field games: the MFG PDE system is:

$$\begin{cases} -\dfrac{\partial u}{\partial t} - 0.05\ \Delta u + \dfrac{8}{(1+m)^{\frac{3}{4}}}\ |\nabla u|^2 = \dfrac{1}{3200}\,, \\ \dfrac{\partial m}{\partial t} - 0.05\ \Delta m - 16\ \mathrm{div}\left(\dfrac{m \nabla u}{(1+m)^{\frac{3}{4}}}\right) = 0\,. \end{cases}$$

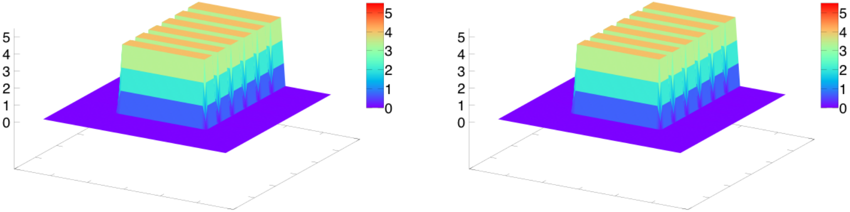
  2. Mean field control: the HJB becomes:

$$-\frac{\partial u}{\partial t} - 0.05\ \Delta u + \left(\frac{2}{(1+m)^{\frac{3}{4}}} + \frac{6}{(1+m)^{\frac{7}{4}}}\right)\ |\nabla u|^2 = \frac{1}{3200}.$$

- We choose a small ν (e.g. 0.05) so the diffusion is not too strong
- No terminal cost: $g \equiv 0$
- Boundary has several parts.
  - Doors: Dirichlet condition $u = 0$ (exit cost), $m = 0$ ($m = 0$ outside the domain)
  - Walls: for u, Neumann condition: $\frac{\partial u}{\partial n} = 0$ (velocity is tangential to the walls); for m: $\nu \frac{\partial m}{\partial n} + m \frac{\partial H}{\partial p}(\cdot, m, \nabla u) \cdot n = 0$, therefore $\frac{\partial m}{\partial n} = 0$
- Initial density m_0: piecewise constant with two values 0 and 4 people/m^2
- Interpretation: At $t = 0$, there are 3300 people in the hall. $T = 50$ minutes

Example: Exit of a Room – Evolution

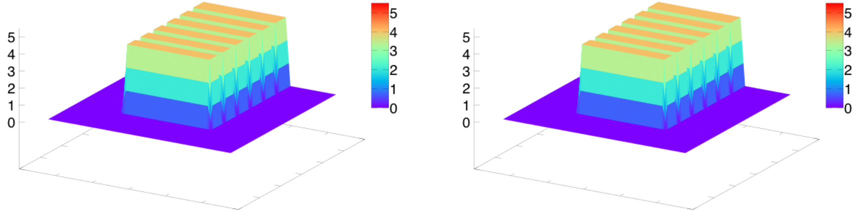
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

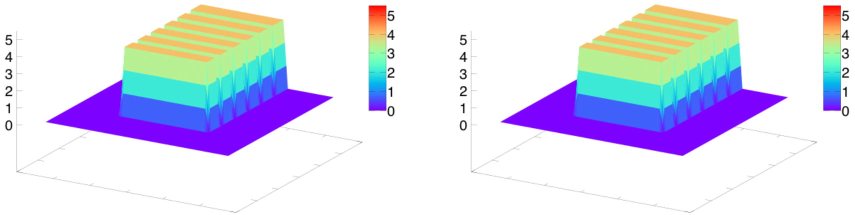
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

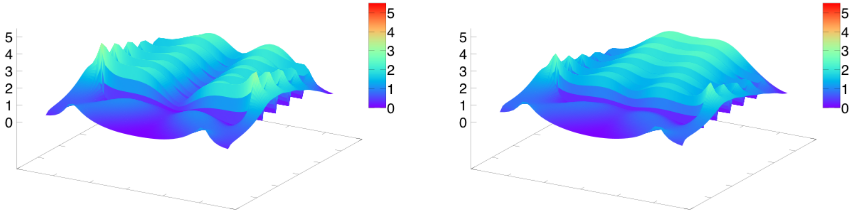
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

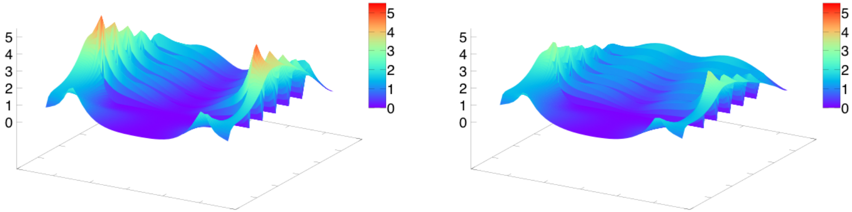
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

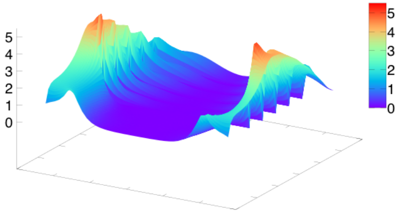
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

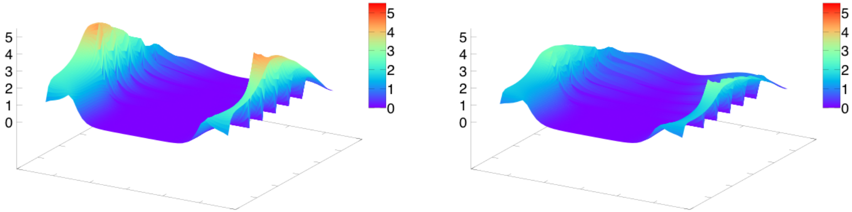
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

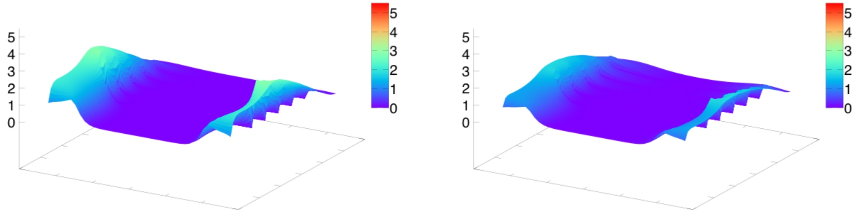
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

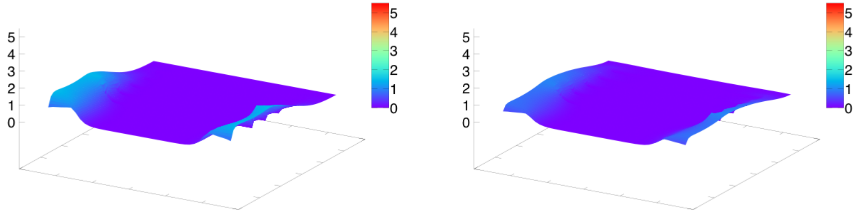
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

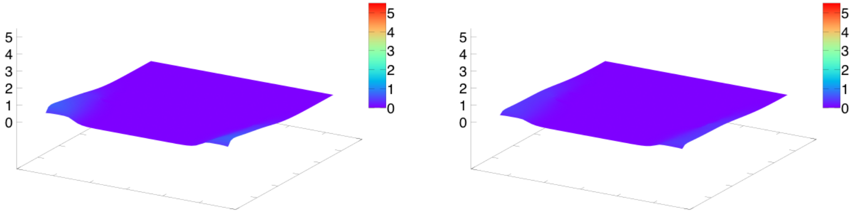
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

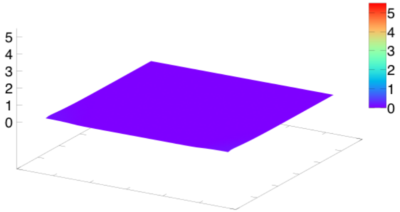
Example: Exit of a Room – Evolution

Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

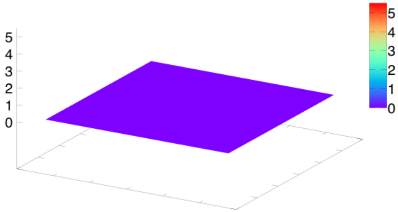
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]

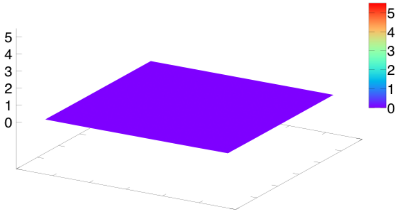


Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]

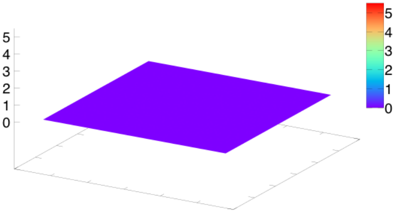




Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

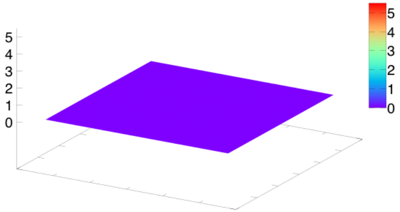
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

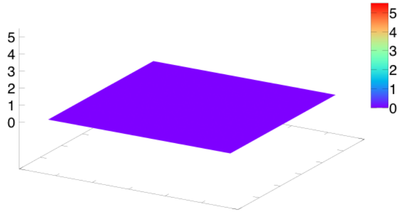
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Evolution

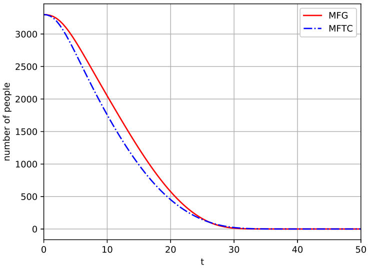
Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2015]



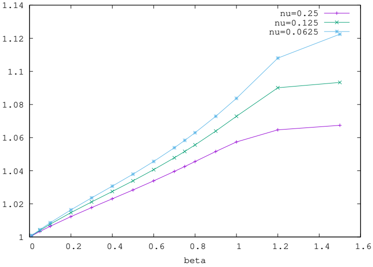
Density in MFGame (left) and MFControl (right)

Example: Exit of a Room – Remaining Mass

Evacuation of a room with obstacles & congestion [Achdou and Laurière, 2020]



Remaining mass inside the room



Price of Anarchy

($\beta =$ exponent)

Outline

MFG Setup

- Scheme introduced by [Carlini and Silva, 2014]
- For simplicity: $d = 1$, domain $\mathcal{X} = \mathbb{R}$, $\mathcal{A} = \mathbb{R}$
- $\nu = 0$, degenerate second order case also possible; see [Carlini and Silva, 2015]
- Model:

$$b(x, m, \alpha) = \alpha$$
$$f(x, m, \alpha) = \frac{1}{2}|\alpha|^2 + f_0(x, m), \qquad g(x, m)$$

where f_0 and g depend on $m \in \mathcal{P}_1(\mathbb{R})$ in a potentially non-local way

MFG Setup

- Scheme introduced by [Carlini and Silva, 2014]
- For simplicity: $d = 1$, domain $\mathcal{X} = \mathbb{R}$, $\mathcal{A} = \mathbb{R}$
- $\nu = 0$, degenerate second order case also possible; see [Carlini and Silva, 2015]
- Model:

$$b(x, m, \alpha) = \alpha$$
$$f(x, m, \alpha) = \frac{1}{2}|\alpha|^2 + f_0(x, m), \qquad g(x, m)$$

  where f_0 and g depend on $m \in \mathcal{P}_1(\mathbb{R})$ in a potentially non-local way
- MFG PDE system:

$$\begin{cases} -\dfrac{\partial u}{\partial t}(t,x) + \dfrac{1}{2}|\nabla u(t,x)|^2 = f_0(x, m(t,\cdot)), & \text{in } [0,T) \times \mathbb{R}, \\ \dfrac{\partial m}{\partial t}(t,x) - \operatorname{div}\left(m(t,\cdot)\nabla u(t,\cdot)\right)(x) = 0, & \text{in } (0,T] \times \mathbb{R}, \\ u(T,x) = g(x, m(T,\cdot)), \qquad m(0,x) = m_0(x), & \text{in } \mathbb{R}. \end{cases}$$

Representation of the Value Function

- Dynamics:

$$X_t^{\alpha} = X_0^{\alpha} + \int_0^t \alpha(s)ds, \qquad t \geq 0.$$

- **Representation formula** for the value function given $m = (m_t)_{t\in[0,T]}$:

$$u[m](t,x) = \inf_{\alpha\in L^2([t,T];\mathbb{R})} \left\{ \int_t^T \left[\frac{1}{2}|\alpha(s)|^2 + f_0(X_s^{\alpha,t,x}, m(s,\cdot))\right] ds + g(X_T^{\alpha,t,x}, m(T,\cdot)) \right\},$$

where $X^{\alpha,t,x}$ starts from x at time t and is controlled by α

Discrete HJB equation

Discrete HJB: Given a flow of densities m,

$$\begin{cases} U_i^n = S_{\Delta t,h}[m](U^{n+1}, i, n), & (n,i) \in [\![N_T - 1]\!] \times \mathbb{Z}, \\ U_i^{N_T} = g(x_i, m(T,\cdot)), & i \in \mathbb{Z}, \end{cases}$$

where

- $S_{\Delta t,h}$ is defined as

$$S_{\Delta t,h}[m](W,n,i) = \inf_{\alpha \in \mathbb{R}} \left\{ \left(\frac{1}{2}|\alpha|^2 + f_0(x_i, m(t_n,\cdot)) \right) \Delta t + I[W](x_i + \alpha\, \Delta t) \right\},$$

Discrete HJB equation

Discrete HJB: Given a flow of densities m,

$$\begin{cases} U_i^n = S_{\Delta t,h}[m](U^{n+1}, i, n), & (n,i) \in [\![N_T - 1]\!] \times \mathbb{Z}, \\ U_i^{N_T} = g(x_i, m(T,\cdot)), & i \in \mathbb{Z}, \end{cases}$$

where

- $S_{\Delta t,h}$ is defined as

$$S_{\Delta t,h}[m](W,n,i) = \inf_{\alpha \in \mathbb{R}} \left\{ \left(\frac{1}{2}|\alpha|^2 + f_0(x_i, m(t_n,\cdot)) \right) \Delta t + I[W](x_i + \alpha\, \Delta t) \right\},$$

- with $I : \mathcal{B}(\mathbb{Z}) \to \mathcal{C}_b(\mathbb{R})$ is the **interpolation operator** defined as

$$I[W](\cdot) = \sum_{i \in \mathbb{Z}} W_i \beta_i(\cdot),$$

- where $\mathcal{B}(\mathbb{Z})$ is the set of bounded functions from $\mathbb{Z}$ to $\mathbb{R}$
- and $\beta_i = \left[1 - \frac{|x - x_i|}{h}\right]_+$: triangular function with support $[x_{i-1}, x_{i+1}]$ and s.t. $\beta_i(x_i) = 1$.

Discrete HJB equation – cont.

Before moving to the KFP equation:

- **Interpolation:** from $U = (U_i^n)_{n,i}$, construct the function $u_{\Delta t,h}[m](x,t) : [0,T] \times \mathbb{R} \to \mathbb{R}$,

$$u_{\Delta t,h}[m](t,x) = I[U^{[\frac{t}{\Delta t}]}](x), \qquad (t,x) \in [0,T] \times \mathbb{R}.$$

Before moving to the KFP equation:

- **Interpolation:** from $U = (U_i^n)_{n,i}$, construct the function $u_{\Delta t,h}[m](x,t) : [0,T] \times \mathbb{R} \to \mathbb{R}$,

$$u_{\Delta t,h}[m](t,x) = I[U^{[\frac{t}{\Delta t}]}](x), \qquad (t,x) \in [0,T] \times \mathbb{R}.$$

- **Regularization of HJB solution** with a mollifier ρ_ϵ:

$$u^\epsilon_{\Delta t,h}[m](t,\cdot) = \rho_\epsilon * u_{\Delta t,h}[m](t,\cdot), \qquad t \in [0,T].$$

Discrete KFP equation: intuition

- **Eulerian** viewpoint:
  - focus on a location
  - look at the flow passing through it
  - evolution characterized by the velocity at (t, x)
- **Lagrangian** viewpoint:
  - focus on a fluid parcel
  - look at how it flows
  - evolution characterized by the position at time t of a particle starting at x

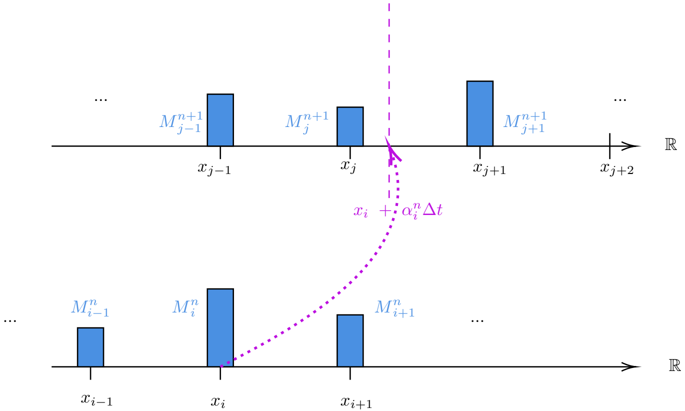
Discrete KFP equation: intuition

- **Eulerian** viewpoint:
  - focus on a location
  - look at the flow passing through it
  - evolution characterized by the velocity at (t, x)
- **Lagrangian** viewpoint:
  - focus on a fluid parcel
  - look at how it flows
  - evolution characterized by the position at time t of a particle starting at x
- Here, in our model:

$$X_t^{\alpha} = X_0^{\alpha} + \int_0^t \alpha(s) ds, \qquad t \geq 0.$$

- Time and space discretization?

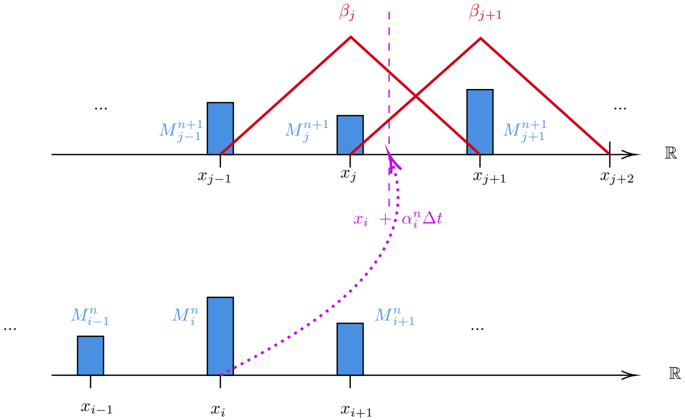
Discrete KFP equation: intuition – diagram



Movement of the mass when using control $v(t_n, x_i) = \alpha_i^n$.

Bottom: time t_n; top: time t_{n+1}.

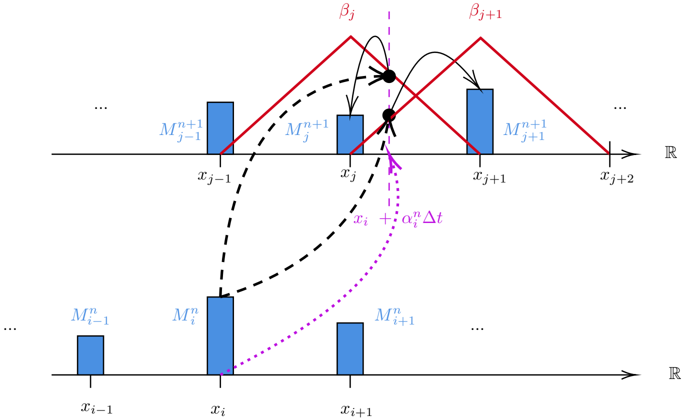
Discrete KFP equation: intuition – diagram



Movement of the mass when using control $v(t_n, x_i) = \alpha_i^n$.

Bottom: time t_n; top: time t_{n+1}.

Discrete KFP equation: intuition – diagram



Movement of the mass when using control $v(t_n, x_i) = \alpha_i^n$.

Bottom: time t_n; top: time t_{n+1}.

Discrete KFP equation

- **Control** induced by value function:

$$\hat{\alpha}^{\epsilon}_{\Delta t,h}[m](t,x) = -\nabla u^{\epsilon}_{\Delta t,h}[m](t,x),$$

and its discrete counter part: $\hat{\alpha}^{\epsilon}_{n,i} = \hat{\alpha}^{\epsilon}_{\Delta t,h}[m](t_n, x_i)$.

- **Discrete flow:**

$$\Phi^{\epsilon}_{n,n+1,i}[m] = x_i + \hat{\alpha}^{\epsilon}_{\Delta t,h}[m](t_n, x_i)\Delta t \,.$$

Discrete KFP equation

- **Control** induced by value function:

$$\hat{\alpha}^{\epsilon}_{\Delta t,h}[m](t,x) = -\nabla u^{\epsilon}_{\Delta t,h}[m](t,x),$$

and its discrete counter part: $\hat{\alpha}^{\epsilon}_{n,i} = \hat{\alpha}^{\epsilon}_{\Delta t,h}[m](t_n, x_i)$.

- **Discrete flow:**

$$\Phi^{\epsilon}_{n,n+1,i}[m] = x_i + \hat{\alpha}^{\epsilon}_{\Delta t,h}[m](t_n, x_i)\Delta t\,.$$

- **Discrete KFP equation:** for $M^{\epsilon}[m] = (M^{\epsilon,n}_i[m])_{n,i}$:

$$\begin{cases} M^{\epsilon,n+1}_i[m] = \sum_j \beta_i \left(\Phi^{\epsilon}_{n,n+1,j}[m]\right) M^{\epsilon,n}_j[m], & (n,i) \in [\![N_T - 1]\!] \times \mathbb{Z}, \\ M^{\epsilon,0}_i[m] = \displaystyle\int_{[x_i - h/2, x_i + h/2]} m_0(x) dx, & i \in \mathbb{Z}. \end{cases}$$

- **Function** $m^{\epsilon}_{\Delta t,h}[m] : [0,T] \times \mathbb{R} \to \mathbb{R}$ defined as: for $n \in [\![N_T - 1]\!]$, for $t \in [t_n, t_{n+1})$,

$$m^{\epsilon}_{\Delta t,h}[m](t,x) = \frac{1}{h} \left[\frac{t_{n+1} - t}{\Delta t} \sum_{i \in \mathbb{Z}} M_i^{\epsilon,n}[m] \mathbf{1}_{[x_i - h/2, x_i + h/2]}(x) \right. \\ \left. + \frac{t - t_n}{\Delta t} \sum_{i \in \mathbb{Z}} M_i^{\epsilon,n+1}[m] \mathbf{1}_{[x_i - h/2, x_i + h/2]}(x) \right] .$$

Fixed Point Formulation

- **Function** $m^{\epsilon}_{\Delta t,h}[m] : [0,T] \times \mathbb{R} \to \mathbb{R}$ defined as: for $n \in [\![N_T - 1]\!]$, for $t \in [t_n, t_{n+1})$,

$$m^{\epsilon}_{\Delta t,h}[m](t,x) = \frac{1}{h} \left[\frac{t_{n+1} - t}{\Delta t} \sum_{i \in \mathbb{Z}} M_i^{\epsilon,n}[m] \mathbf{1}_{[x_i - h/2, x_i + h/2]}(x) \right.$$
$$\left. + \frac{t - t_n}{\Delta t} \sum_{i \in \mathbb{Z}} M_i^{\epsilon,n+1}[m] \mathbf{1}_{[x_i - h/2, x_i + h/2]}(x) \right].$$

- **Goal: Fixed-point problem:** Find $\hat{M} = (\hat{M}_i^n)_{i,n}$ such that:

$$\hat{M}_i^n = M_i^n \left[m^{\epsilon}_{\Delta t,h}[\hat{M}] \right].$$

Fixed Point Formulation

- **Function** $m^{\epsilon}_{\Delta t,h}[m] : [0,T] \times \mathbb{R} \to \mathbb{R}$ defined as: for $n \in [\![N_T - 1]\!]$, for $t \in [t_n, t_{n+1})$,

$$m^{\epsilon}_{\Delta t,h}[m](t,x) = \frac{1}{h}\left[\frac{t_{n+1}-t}{\Delta t}\sum_{i\in\mathbb{Z}} M_i^{\epsilon,n}[m]\mathbf{1}_{[x_i-h/2,x_i+h/2]}(x) + \frac{t-t_n}{\Delta t}\sum_{i\in\mathbb{Z}} M_i^{\epsilon,n+1}[m]\mathbf{1}_{[x_i-h/2,x_i+h/2]}(x)\right].$$

- **Goal: Fixed-point problem:** Find $\hat{M} = (\hat{M}_i^n)_{i,n}$ such that:

$$\hat{M}_i^n = M_i^n\left[m^{\epsilon}_{\Delta t,h}[\hat{M}]\right].$$

- **Solution strategy:** Fixed point iterations for example
- See [Carlini and Silva, 2014] for more details

Numerical Illustration

Costs:

$$g \equiv 0, \qquad f(x, m, \alpha) = \frac{1}{2}|\alpha|^2 + (x - c^*)^2 + \kappa_{MF} V(x, m),$$

with

$$V(x, m) = \rho_{\sigma_V} * \big(\rho_{\sigma_V} * m\big)(x),$$

Numerical Illustration

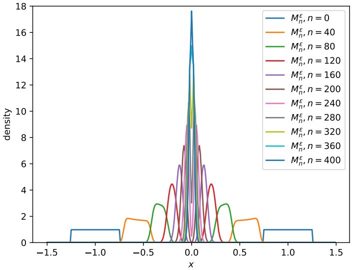
Costs:

$$g \equiv 0, \qquad f(x, m, \alpha) = \frac{1}{2}|\alpha|^2 + (x - c^*)^2 + \kappa_{MF} V(x, m),$$

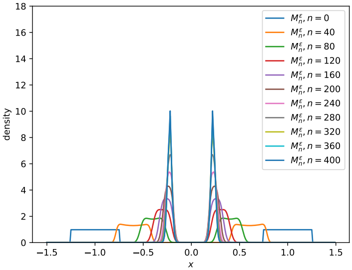
with

$$V(x, m) = \rho_{\sigma_V} * \big(\rho_{\sigma_V} * m\big)(x),$$

Experiments: target $c^* = 0$, m_0 = unif. on $[-1.25, -0.75]$ and on $[0.75, 1.25]$



$\kappa_{MF} = 0.5$



$\kappa_{MF} = 0.9$

See [Laurière, 2021] for more details on these experiments

Sample code

Code

Sample code to illustrate: IPython notebook

```
https://colab.research.google.com/drive/1ZikqKh-DlIGNJhhgzPQV0_gIu1jOP78g?usp=sharing
```

- Semi-Lagrangian scheme
- Solved by damped fixed point approach

Exercise

Exercise

Implement the previous finite difference scheme on the same MFG model.

If the algorithm fails to converge with $\nu = 0$, try with $\nu > 0$ but small.

Outline

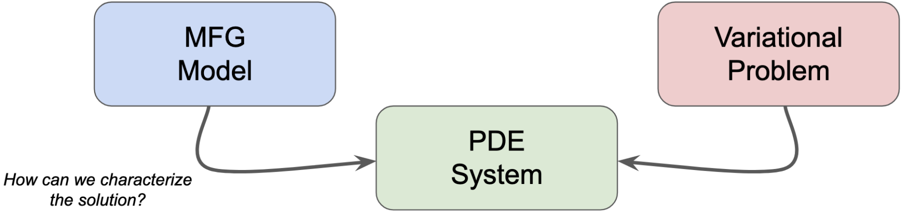
Outline

Variational MFGs

Key ideas:

- Variational MFG
- Duality
- Optimization techniques

Variational MFGs



In some cases, the MFG PDE system can be interpreted as the optimality conditions for a variational problem

MFG PDE system $\Leftrightarrow$ *optimality condition of two optimization problems in duality*

See [Lasry and Lions, 2007], [Cardaliaguet, 2015], [Cardaliaguet and Graber, 2015], [Cardaliaguet et al., 2015], [Benamou et al., 2017], ...

A Variational MFG

- $d = 1$, domain = $\mathbb{T}$
- drift and costs:

$$b(x, m, \alpha) = \alpha, \qquad f(x, m, \alpha) = L(x, \alpha) + \mathtt{f}_0(x, m), \qquad g(x, m) = \mathtt{g}_0(x).$$

where $x \in \mathbb{R}^d, \alpha \in \mathbb{R}^d, m \in \mathbb{R}_+$.

- Then

$$H(x, m, p) = \sup_{\alpha} \{-L(x, \alpha) - \alpha p\} - \mathtt{f}_0(x, m) = H_0(x, p) - \mathtt{f}_0(x, m)$$

- where H_0 is the convex conjugate (also denoted L^*) of L with respect to α:

$$H_0(x, p) = L^*(x, p) = \sup_{\alpha}\{\ \alpha p - L(x, \alpha)\}$$

- Further assume (for simplicity)

$$L(x, \alpha) = \frac{1}{2}|\alpha|^2, \qquad H_0(x, p) = \frac{1}{2}|p|^2$$

A Variational Problem

- At equilibrium, $\mathcal{L}(X_t) = \hat{\mu}_t$ and

$$J(\hat{\alpha};\hat{m}) = \mathbb{E}\left[\int_0^T f(X_t,\hat{m}(t,X_t),\hat{\alpha}(t,X_t))dt + g(X_T)\right]$$

$$= \int_0^T \int_{\mathbb{T}} \underbrace{f(x,\hat{m}(t,x),\hat{\alpha}(t,x))}_{=L(x,\hat{\alpha}(t,x))+f_0(x,\hat{m}(t,x))} \hat{m}(t,x)dxdt + \int_{\mathbb{T}} g(x)\hat{m}(T,x)dx$$

subject to:

$$0 = \frac{\partial \hat{m}}{\partial t}(t,x) - \nu\Delta\hat{m}(t,x) + \mathrm{div}\left(\hat{m}(t,\cdot)\underbrace{b(\cdot,\hat{m}(t),\hat{\alpha}(t,\cdot))}_{=\hat{\alpha}(t,\cdot)}\right)(x), \qquad \hat{m}_0 = m_0$$

A Variational Problem

- At equilibrium, $\mathcal{L}(X_t) = \hat{\mu}_t$ and

$$J(\hat{\alpha}; \hat{m}) = \mathbb{E}\left[\int_0^T f(X_t, \hat{m}(t, X_t), \hat{\alpha}(t, X_t))dt + g(X_T)\right]$$

$$= \int_0^T \int_{\mathbb{T}} \underbrace{f(x, \hat{m}(t,x), \hat{\alpha}(t,x))}_{=L(x,\hat{\alpha}(t,x))+\mathtt{f}_0(x,\hat{m}(t,x))} \hat{m}(t,x)dxdt + \int_{\mathbb{T}} g(x)\hat{m}(T,x)dx$$

subject to:

$$0 = \frac{\partial \hat{m}}{\partial t}(t,x) - \nu\Delta\hat{m}(t,x) + \operatorname{div}\Big(\hat{m}(t,\cdot)\underbrace{b(\cdot, \hat{m}(t), \hat{\alpha}(t,\cdot))}_{=\hat{\alpha}(t,\cdot)}\Big)(x), \qquad \hat{m}_0 = m_0$$

- Change of variable:

$$\hat{w}(t,x) = \hat{m}(t,x)\hat{\alpha}(t,x)$$

$$\mathcal{B}(\hat{m}, \hat{w}) = \int_0^T \int_{\mathbb{T}} \left[L\left(x, \frac{\hat{w}(t,x)}{\hat{m}(t,x)}\right) + \mathtt{f}_0(x, \hat{m}(t,x))\right]\hat{m}(t,x)dxdt + \int_{\mathbb{T}} g(x)\hat{m}(T,x)dx$$

subject to:

$$0 = \frac{\partial \hat{m}}{\partial t}(t,x) - \nu\Delta\hat{m}(t,x) + \operatorname{div}\Big(\hat{w}(t,\cdot)\Big)(x), \qquad \hat{m}_0 = m_0$$

- Reformulation:

$$\mathcal{B}(\hat{m},\hat{w}) = \int_0^T \int_{\mathbb{T}} \Big[\underbrace{L\Big(x, \frac{\hat{w}(t,x)}{\hat{m}(t,x)}\Big)\hat{m}(t,x)}_{\widetilde{L}(x,\hat{m}(t,x),\hat{w}(t,x))} + \underbrace{\mathtt{f}_0(x,\hat{m}(t,x))\hat{m}(t,x)}_{\widetilde{F}(x,\hat{m}(t,x))} \Big] dxdt + \int_{\mathbb{T}} \underbrace{g(x)\hat{m}(T,x)}_{\widetilde{G}(x,\hat{m}(t,x))} dx$$

$$= \int_0^T \int_{\mathbb{T}} \Big[\widetilde{L}(x,\hat{m}(t,x),\hat{w}(t,x)) + \widetilde{F}(x,\hat{m}(t,x)) \Big] dxdt + \int_{\mathbb{T}} \widetilde{G}(x,\hat{m}(t,x))dx$$

subject to:

$$0 = \frac{\partial \hat{m}}{\partial t}(t,x) - \nu\Delta\hat{m}(t,x) + \mathrm{div}\Big(\hat{w}(t,\cdot)\Big)(x), \qquad \hat{m}_0 = m_0$$

Reformulation

- Reformulation:

$$\mathcal{B}(\hat{m},\hat{w}) = \int_0^T \int_{\mathbb{T}} \Big[\underbrace{L\Big(x, \frac{\hat{w}(t,x)}{\hat{m}(t,x)}\Big)\hat{m}(t,x)}_{\widetilde{L}(x,\hat{m}(t,x),\hat{w}(t,x))} + \underbrace{\mathtt{f}_0(x,\hat{m}(t,x))\hat{m}(t,x)}_{\widetilde{F}(x,\hat{m}(t,x))} \Big] dx dt$$

$$+ \int_{\mathbb{T}} \underbrace{g(x)\hat{m}(T,x)}_{\widetilde{G}(x,\hat{m}(t,x))} dx$$

$$= \int_0^T \int_{\mathbb{T}} \Big[\widetilde{L}(x,\hat{m}(t,x),\hat{w}(t,x)) + \widetilde{F}(x,\hat{m}(t,x)) \Big] dx dt + \int_{\mathbb{T}} \widetilde{G}(x,\hat{m}(t,x)) dx$$

subject to:

$$0 = \frac{\partial \hat{m}}{\partial t}(t,x) - \nu \Delta \hat{m}(t,x) + \operatorname{div}\Big(\hat{w}(t,\cdot)\Big)(x), \qquad \hat{m}_0 = m_0$$

- **Convex** problem under a **linear** constraint, provided $\widetilde{L}, \widetilde{F}, \widetilde{G}$ are convex

Primal Optimization Problem

Primal problem: Minimize over $(m, w) = (m, m\alpha)$:

$$\mathcal{B}(m, w) = \int_0^T \int_{\mathbb{T}} \left(\widetilde{L}(x, m(t,x), w(t,x)) + \widetilde{F}(x, m(t,x)) \right) dxdt + \int_{\mathbb{T}} \widetilde{G}(x, m(T,x)) dx$$

subject to the constraint:

$$\partial_t m - \nu \Delta m + \mathrm{div}(w) = 0, \qquad m(0,x) = m_0(x)$$

Primal Optimization Problem

Primal problem: Minimize over $(m, w) = (m, m\alpha)$:

$$\mathcal{B}(m,w) = \int_0^T \int_{\mathbb{T}} \Big(\widetilde{L}(x, m(t,x), w(t,x)) + \widetilde{F}(x, m(t,x)) \Big)\, dx dt + \int_{\mathbb{T}} \widetilde{G}(x, m(T,x)) dx$$

subject to the constraint:

$$\partial_t m - \nu \Delta m + \operatorname{div}(w) = 0, \qquad m(0,x) = m_0(x)$$

where

$$\widetilde{F}(x,m) = \begin{cases} \int_0^m \tilde{f}(x,s)ds, & \text{if } m \geq 0, \\ +\infty, & \text{otherwise,} \end{cases} \qquad \widetilde{G}(x,m) = \begin{cases} m\, \mathrm{g}_0(x), & \text{if } m \geq 0, \\ +\infty, & \text{otherwise,} \end{cases}$$

and

$$\widetilde{L}(x,m,w) = \begin{cases} mL\left(x, \frac{w}{m}\right), & \text{if } m > 0, \\ 0, & \text{if } m = 0 \text{ and } w = 0, \\ +\infty, & \text{otherwise} \end{cases}$$

where $\mathbb{R} \ni m \mapsto \tilde{f}(x,m) = \partial_m(m\, \mathrm{f}_0(x,m))$
is non-decreasing (hence $\widetilde{F}$ convex and l.s.c.) provided $m \mapsto m\, \mathrm{f}_0(x,m)$ is convex.

Dual problem: Maximize over ϕ such that $\phi(T,x) = \mathrm{g}_0(x)$

$$\mathcal{A}(\phi) = \inf_m \mathcal{A}(\phi, m)$$

with $\mathcal{A}(\phi, m) = \int_0^T \int_{\mathbb{T}} m(t,x)\Big(\partial_t \phi(t,x) + \nu \Delta \phi(t,x) - H(x, m(t,x), \nabla \phi(t,x))\Big) dx dt$

$$+ \int_{\mathbb{T}} m_0(x) \phi(0,x) dx.$$

Duality

Dual problem: Maximize over ϕ such that $\phi(T,x) = \mathrm{g}_0(x)$

$$\mathcal{A}(\phi) = \inf_{m} \mathcal{A}(\phi, m)$$

with $\mathcal{A}(\phi, m) = \int_0^T \int_{\mathbb{T}} m(t,x)\Big(\partial_t \phi(t,x) + \nu\Delta\phi(t,x) - H(x, m(t,x), \nabla\phi(t,x))\Big) dxdt$

$$+ \int_{\mathbb{T}} m_0(x)\phi(0,x)dx.$$

Duality relation: $\mathcal{A}$ and $\mathcal{B}$ satisfy: **(A)** $= \sup_{\phi} \mathcal{A}(\phi) = \inf_{(m,w)} \mathcal{B}(m,w) =$ **(B)**

Duality

Dual problem: Maximize over ϕ such that $\phi(T, x) = \mathrm{g}_0(x)$

$$\mathcal{A}(\phi) = \inf_{m} \mathcal{A}(\phi, m)$$

with $\mathcal{A}(\phi, m) = \int_0^T \int_{\mathbb{T}} m(t,x)\Big(\partial_t \phi(t,x) + \nu\Delta\phi(t,x) - H(x, m(t,x), \nabla\phi(t,x))\Big) dx dt$

$$+ \int_{\mathbb{T}} m_0(x)\phi(0,x) dx.$$

Duality relation: $\mathcal{A}$ and $\mathcal{B}$ satisfy: **(A)** $= \sup_{\phi} \mathcal{A}(\phi) = \inf_{(m,w)} \mathcal{B}(m, w) =$ **(B)**

Proof idea: Fenchel-Rockafellar duality theorem and observe:

$$\textbf{(A)} = -\inf_{\phi} \Big\{ \mathcal{F}(\phi) + \mathcal{G}(\Lambda(\phi)) \Big\}, \qquad \textbf{(B)} = \inf_{(m,w)} \Big\{ \mathcal{F}^*(\Lambda^*(m,w)) + \mathcal{G}^*(-m, -w) \Big\}$$

where $\mathcal{F}^*, \mathcal{G}^*$ are the convex conjugates of $\mathcal{F}, \mathcal{G}$, and Λ^* is the adjoint operator of Λ, and $\Lambda(\phi) = \left(\frac{\partial \phi}{\partial t} + \nu\Delta\phi, \nabla\phi\right)$,

$$\mathcal{F}(\phi) = \chi_T(\phi) - \int_{\mathbb{T}^d} m_0(x)\phi(0,x) dx, \qquad \chi_T(\phi) = \begin{cases} 0 & \text{if } \phi|_{t=T} = \mathrm{g}_0 \\ +\infty & \text{otherwise,} \end{cases}$$

$$\mathcal{G}(\varphi_1, \varphi_2) = -\inf_{0 \le m \in L^1((0,T)\times\mathbb{T}^d)} \int_0^T \int_{\mathbb{T}^d} m(t,x) \left(\varphi_1(t,x) - H(x, m(t,x), \varphi_2(t,x))\right) dx dt.$$

Outline

Augmented Lagrangian

Reformulation of the primal problem:

$$\textbf{(A)} = -\inf_{\phi}\Big\{\mathcal{F}(\phi) + \mathcal{G}(\Lambda(\phi))\Big\} = -\inf_{\phi}\ \inf_{q}\Big\{\mathcal{F}(\phi) + \mathcal{G}(q), \text{ subj. to } q = \Lambda(\phi)\Big\}.$$

- The corresponding **Lagrangian** is

$$\mathcal{L}(\phi, q, \tilde{q}) = \mathcal{F}(\phi) + \mathcal{G}(q) - \langle \tilde{q}, \Lambda(\phi) - q\rangle.$$

Augmented Lagrangian

Reformulation of the primal problem:

(A) $= -\inf_{\phi}\Big\{\mathcal{F}(\phi) + \mathcal{G}(\Lambda(\phi))\Big\} = -\inf_{\phi}\,\inf_{q}\Big\{\mathcal{F}(\phi) + \mathcal{G}(q),$ subj. to $q = \Lambda(\phi)\Big\}$.

- The corresponding **Lagrangian** is

$$\mathcal{L}(\phi, q, \tilde{q}) = \mathcal{F}(\phi) + \mathcal{G}(q) - \langle \tilde{q}, \Lambda(\phi) - q \rangle.$$

- We consider the **augmented Lagrangian** (with parameter $r > 0$)

$$\mathcal{L}^r(\phi, q, \tilde{q}) = \mathcal{L}(\phi, q, \tilde{q}) + \frac{r}{2}\|\Lambda(\phi) - q\|^2$$

- Goal: find a **saddle-point** of $\mathcal{L}^r$.

Alternating Direction Method of Multipliers (ADMM)

Reminder: $\mathcal{L}^r(\phi, q, \tilde{q}) = \mathcal{F}(\phi) + \mathcal{G}(q) - \langle \tilde{q}, \Lambda(\phi) - q \rangle + \frac{r}{2}\|\Lambda(\phi) - q\|^2$

Input: Initial guess $(\phi^{(0)}, q^{(0)}, \tilde{q}^{(0)})$; number of iterations $\mathtt{K}$
Output: Approximation of a saddle point $(\phi, q, \tilde{q})$ solving the finite difference system

Initialize $(\phi^{(0)}, q^{(0)}, \tilde{q}^{(0)})$
for $\mathtt{k} = 0, 1, 2, \dots, \mathtt{K} - 1$ **do**
    (a) Compute

$$\phi^{(\mathtt{k}+1)} \in \operatorname*{argmin}_{\phi} \left\{ \mathcal{F}(\phi) - \langle \tilde{q}^{(\mathtt{k})}, \Lambda(\phi) \rangle + \frac{r}{2}\|\Lambda(\phi) - q^{(\mathtt{k})}\|^2 \right\}$$

References: ALG2 in the book of [Fortin and Glowinski, 1983]
→ in MFG: [Benamou and Carlier, 2015a], [Andreev, 2017]
→ in MFC:[Achdou and Laurière, 2016b], [Baudelet et al., 2023]

Alternating Direction Method of Multipliers (ADMM)

Reminder: $\mathcal{L}^r(\phi, q, \tilde{q}) = \mathcal{F}(\phi) + \mathcal{G}(q) - \langle \tilde{q}, \Lambda(\phi) - q \rangle + \frac{r}{2}\|\Lambda(\phi) - q\|^2$

Input: Initial guess $(\phi^{(0)}, q^{(0)}, \tilde{q}^{(0)})$; number of iterations $\mathtt{K}$
Output: Approximation of a saddle point $(\phi, q, \tilde{q})$ solving the finite difference system

Initialize $(\phi^{(0)}, q^{(0)}, \tilde{q}^{(0)})$
for $\mathtt{k} = 0, 1, 2, \dots, \mathtt{K} - 1$ **do**

(a) Compute

$$\phi^{(\mathtt{k}+1)} \in \operatorname*{argmin}_{\phi} \left\{ \mathcal{F}(\phi) - \langle \tilde{q}^{(\mathtt{k})}, \Lambda(\phi) \rangle + \frac{r}{2}\|\Lambda(\phi) - q^{(\mathtt{k})}\|^2 \right\}$$

(b) Compute

$$q^{(\mathtt{k}+1)} \in \operatorname*{argmin}_{q} \left\{ \mathcal{G}(q) + \langle \tilde{q}^{(\mathtt{k})}, q \rangle + \frac{r}{2}\|\Lambda(\phi^{(\mathtt{k}+1)}) - q\|^2 \right\}$$

References: ALG2 in the book of [Fortin and Glowinski, 1983]
$\rightarrow$ in MFG: [Benamou and Carlier, 2015a], [Andreev, 2017]
$\rightarrow$ in MFC:[Achdou and Laurière, 2016b], [Baudelet et al., 2023]

Alternating Direction Method of Multipliers (ADMM)

Reminder: $\mathcal{L}^r(\phi, q, \tilde{q}) = \mathcal{F}(\phi) + \mathcal{G}(q) - \langle \tilde{q}, \Lambda(\phi) - q \rangle + \frac{r}{2}\|\Lambda(\phi) - q\|^2$

Input: Initial guess $(\phi^{(0)}, q^{(0)}, \tilde{q}^{(0)})$; number of iterations $\mathtt{K}$
Output: Approximation of a saddle point $(\phi, q, \tilde{q})$ solving the finite difference system

Initialize $(\phi^{(0)}, q^{(0)}, \tilde{q}^{(0)})$
for $\mathtt{k} = 0, 1, 2, \ldots, \mathtt{K} - 1$ **do**

(a) Compute

$$\phi^{(\mathtt{k}+1)} \in \operatorname*{argmin}_{\phi} \left\{ \mathcal{F}(\phi) - \langle \tilde{q}^{(\mathtt{k})}, \Lambda(\phi) \rangle + \frac{r}{2}\|\Lambda(\phi) - q^{(\mathtt{k})}\|^2 \right\}$$

(b) Compute

$$q^{(\mathtt{k}+1)} \in \operatorname*{argmin}_{q} \left\{ \mathcal{G}(q) + \langle \tilde{q}^{(\mathtt{k})}, q \rangle + \frac{r}{2}\|\Lambda(\phi^{(\mathtt{k}+1)}) - q\|^2 \right\}$$

(c) Compute

$$\tilde{q}^{(\mathtt{k}+1)} = \tilde{q}^{(\mathtt{k})} - r \left(\Lambda(\phi^{(\mathtt{k}+1)}) - q^{(\mathtt{k}+1)} \right)$$

return $(\phi^{(\mathtt{K})}, q^{(\mathtt{K})}, \tilde{q}^{(\mathtt{K})})$

References: ALG2 in the book of [Fortin and Glowinski, 1983]
→ in MFG: [Benamou and Carlier, 2015a], [Andreev, 2017]
→ in MFC:[Achdou and Laurière, 2016b], [Baudelet et al., 2023]

ADMM: Discrete Primal Problem

Notation: N_h, N_T steps resp. in space and time, $N = (N_T + 1)N_h$, $N' = N_T N_h$.

Recall: $H_0(x,p) = \frac{1}{2}|p|^2$. We take $\tilde{H}_0(x,p_1,p_2) = \frac{1}{2}|(p_1^-, p_2^+)|^2$.

Discrete version of the **dual** convex problem:

$$(\mathbf{A_h}) = -\inf_{\phi \in \mathbb{R}^N} \left\{ \mathcal{F}_h(\phi) + \mathcal{G}_h(\Lambda_h(\phi)) \right\},$$

where $\Lambda_h : \mathbb{R}^N \to \mathbb{R}^{3N'}$ is defined by : $\forall\, n \in \{1, \ldots, N_T\}, \forall\, i \in \{0, \ldots, N_h - 1\}$,

$$(\Lambda_h(\phi))_i^n = \left((D_t \phi_i)^n + \nu \left(\Delta_h \phi^{n-1} \right)_i, [\nabla_h \phi^{n-1}]_i \right),$$

ADMM: Discrete Primal Problem

Notation: N_h, N_T steps resp. in space and time, $N = (N_T + 1)N_h$, $N' = N_T N_h$.

Recall: $H_0(x,p) = \frac{1}{2}|p|^2$. We take $\tilde{H}_0(x, p_1, p_2) = \frac{1}{2}|(p_1^-, p_2^+)|^2$.

Discrete version of the **dual** convex problem:

$$(\mathbf{A_h}) = -\inf_{\phi \in \mathbb{R}^N} \left\{ \mathcal{F}_h(\phi) + \mathcal{G}_h(\Lambda_h(\phi)) \right\},$$

where $\Lambda_h : \mathbb{R}^N \to \mathbb{R}^{3N'}$ is defined by : $\forall n \in \{1, \ldots, N_T\}, \forall i \in \{0, \ldots, N_h - 1\}$,

$$(\Lambda_h(\phi))_i^n = \left((D_t \phi_i)^n + \nu \left(\Delta_h \phi^{n-1} \right)_i, [\nabla_h \phi^{n-1}]_i \right),$$

where $\mathcal{F}_h, \mathcal{G}_h$ are the l.s.c. proper functions defined by:

$$\mathcal{F}_h : \mathbb{R}^N \ni \phi \mapsto \chi_T(\phi) - h \sum_{i=0}^{N_h - 1} \rho_i^0 \phi_i^0 \in \mathbb{R} \cup \{+\infty\},$$

$$\mathcal{G}_h : \mathbb{R}^{3N'} \ni (a, b, c) \mapsto -h \Delta t \sum_{n=1}^{N_T} \sum_{i=0}^{N_h - 1} \mathcal{K}_h(x_i, a_i^n, b_i^n, c_i^n) \in \mathbb{R} \cup \{+\infty\},$$

with

$$\mathcal{K}_h(x, a_0, p_1, p_2) = \min_{m \in \mathbb{R}_+} \left\{ m[a_0 + \tilde{H}_0(x, m, p_1, p_2)] \right\}, \quad \chi_T(\phi) = \begin{cases} 0 & \text{if } \phi_i^{N_T} \equiv \mathrm{g}_0(x_i) \\ +\infty & \text{otherwise.} \end{cases}$$

ADMM with Discretization

Discrete Aug. Lag.: $\mathcal{L}_h^r(\phi, q, \tilde{q}) = \mathcal{F}_h(\phi) + \mathcal{G}_h(q) - \langle \tilde{q}, \Lambda_h(\phi) - q \rangle + \frac{r}{2}\|\Lambda(\phi) - q\|^2$

Input: Initial guess $(\phi^{(0)}, q^{(0)}, \tilde{q}^{(0)})$; number of iterations $\mathtt{K}$

Output: Approximation of a saddle point $(\phi, q, \tilde{q})$

Initialize $(\phi^{(0)}, q^{(0)}, \tilde{q}^{(0)})$

for $\mathtt{k} = 0, 1, 2, \ldots, \mathtt{K} - 1$ **do**

- (a) Compute $\phi^{(\mathtt{k}+1)} \in \operatorname{argmin}_\phi \left\{ \mathcal{F}_h(\phi) - \langle \tilde{q}^{(\mathtt{k})}, \Lambda_h(\phi) \rangle + \frac{r}{2}\|\Lambda_h(\phi) - q^{(\mathtt{k})}\|^2 \right\}$
- (b) Compute $q^{(\mathtt{k}+1)} \in \operatorname{argmin}_q \left\{ \mathcal{G}_h(q) + \langle \tilde{q}^{(\mathtt{k})}, q \rangle + \frac{r}{2}\|\Lambda_h(\phi^{(\mathtt{k}+1)}) - q\|^2 \right\}$
- (c) Compute $\tilde{q}^{(\mathtt{k}+1)} = \tilde{q}^{(\mathtt{k})} - r\left(\Lambda_h(\phi^{(\mathtt{k}+1)}) - q^{(\mathtt{k}+1)}\right)$

return $(\phi^{(\mathtt{K})}, q^{(\mathtt{K})}, \tilde{q}^{(\mathtt{K})})$

ADMM with Discretization

Discrete Aug. Lag.: $\mathcal{L}_h^r(\phi, q, \tilde{q}) = \mathcal{F}_h(\phi) + \mathcal{G}_h(q) - \langle \tilde{q}, \Lambda_h(\phi) - q \rangle + \frac{r}{2}\|\Lambda(\phi) - q\|^2$

Input: Initial guess $(\phi^{(0)}, q^{(0)}, \tilde{q}^{(0)})$; number of iterations $\mathtt{K}$
Output: Approximation of a saddle point $(\phi, q, \tilde{q})$

Initialize $(\phi^{(0)}, q^{(0)}, \tilde{q}^{(0)})$
for $\mathtt{k} = 0, 1, 2, \ldots, \mathtt{K} - 1$ **do**
- (a) Compute $\phi^{(\mathtt{k}+1)} \in \mathrm{argmin}_\phi \left\{ \mathcal{F}_h(\phi) - \langle \tilde{q}^{(\mathtt{k})}, \Lambda_h(\phi) \rangle + \frac{r}{2}\|\Lambda_h(\phi) - q^{(\mathtt{k})}\|^2 \right\}$
- (b) Compute $q^{(\mathtt{k}+1)} \in \mathrm{argmin}_q \left\{ \mathcal{G}_h(q) + \langle \tilde{q}^{(\mathtt{k})}, q \rangle + \frac{r}{2}\|\Lambda_h(\phi^{(\mathtt{k}+1)}) - q\|^2 \right\}$
- (c) Compute $\tilde{q}^{(\mathtt{k}+1)} = \tilde{q}^{(\mathtt{k})} - r\left(\Lambda_h(\phi^{(\mathtt{k}+1)}) - q^{(\mathtt{k}+1)}\right)$

return $(\phi^{(\mathtt{K})}, q^{(\mathtt{K})}, \tilde{q}^{(\mathtt{K})})$

First-order Optimality Conditions:

Step (a): finite-difference equation

Step (b): minimization problem at each point of the grid

ADMM with Discretization

Discrete Aug. Lag.: $\mathcal{L}_h^r(\phi, q, \tilde{q}) = \mathcal{F}_h(\phi) + \mathcal{G}_h(q) - \langle \tilde{q}, \Lambda_h(\phi) - q \rangle + \frac{r}{2}\|\Lambda(\phi) - q\|^2$

Input: Initial guess $(\phi^{(0)}, q^{(0)}, \tilde{q}^{(0)})$; number of iterations $\mathtt{K}$
Output: Approximation of a saddle point $(\phi, q, \tilde{q})$

```
Initialize (φ^(0), q^(0), q̃^(0))
for k = 0, 1, 2, ..., K − 1 do
    (a) Compute φ^(k+1) ∈ argmin_φ { F_h(φ) − ⟨q̃^(k), Λ_h(φ)⟩ + (r/2)‖Λ_h(φ) − q^(k)‖² }
    (b) Compute q^(k+1) ∈ argmin_q { G_h(q) + ⟨q̃^(k), q⟩ + (r/2)‖Λ_h(φ^(k+1)) − q‖² }
    (c) Compute q̃^(k+1) = q̃^(k) − r (Λ_h(φ^(k+1)) − q^(k+1))
return (φ^(K), q^(K), q̃^(K))
```

Where:

(a) Compute $\phi^{(\mathtt{k}+1)} \in \mathrm{argmin}_\phi \left\{ \mathcal{F}_h(\phi) - \langle \tilde{q}^{(\mathtt{k})}, \Lambda_h(\phi) \rangle + \frac{r}{2}\|\Lambda_h(\phi) - q^{(\mathtt{k})}\|^2 \right\}$

(b) Compute $q^{(\mathtt{k}+1)} \in \mathrm{argmin}_q \left\{ \mathcal{G}_h(q) + \langle \tilde{q}^{(\mathtt{k})}, q \rangle + \frac{r}{2}\|\Lambda_h(\phi^{(\mathtt{k}+1)}) - q\|^2 \right\}$

(c) Compute $\tilde{q}^{(\mathtt{k}+1)} = \tilde{q}^{(\mathtt{k})} - r\left(\Lambda_h(\phi^{(\mathtt{k}+1)}) - q^{(\mathtt{k}+1)}\right)$

return $(\phi^{(\mathtt{K})}, q^{(\mathtt{K})}, \tilde{q}^{(\mathtt{K})})$

First-order Optimality Conditions:

Step (a): finite-difference equation

Step (b): minimization problem at each point of the grid

Rem.: For (a): discrete PDE

- if $\nu = 0$, a direct solver can be used
- if $\nu > 0$, PDE with 4^{th} order linear elliptic operator $\Rightarrow$ needs preconditioner

See e.g. [Achdou and Perez, 2012], [Andreev, 2017], [Briceño Arias et al., 2018]

Numerical Example: Congestion Without Viscosity

- Domain $\Omega = [0,1]^2 \backslash [0.4, 0.6]^2$ (obstacle at the center)
- Define the Hamiltonian by duality (on $\partial\Omega$ the vector speed is towards the interior)

$$H(x, m, p) = \begin{cases} \sup\limits_{\xi \in \mathbb{R}^2} \Big\{ -\xi \cdot p - L(x, m, \xi) \Big\} = m^{-\alpha} |p|^{\beta} - \ell(x, m), & \text{if } x \in \Omega, \\ \sup\limits_{\xi \in \mathbb{R}^2 \,:\, \xi \cdot n \leq 0} \Big\{ -\xi \cdot p - L(x, m, \xi) \Big\}, & \text{if } x \in \partial\Omega. \end{cases}$$

- The associated Lagrangian (corresponding to the running cost) is:

$$L(x, m, \xi) = (\beta - 1)\beta^{-\beta^*} m^{\frac{\alpha}{\beta - 1}} |\xi|^{\beta^*} + \ell(x, m), \qquad 1 < \beta \leq 2, 0 \leq \alpha < 1$$

- Domain $\Omega = [0,1]^2 \backslash [0.4, 0.6]^2$ (obstacle at the center)
- Define the Hamiltonian by duality (on $\partial\Omega$ the vector speed is towards the interior)

$$H(x,m,p) = \begin{cases} \sup_{\xi \in \mathbb{R}^2} \Big\{ -\xi \cdot p - L(x,m,\xi) \Big\} = m^{-\alpha}|p|^{\beta} - \ell(x,m), & \text{if } x \in \Omega, \\ \sup_{\xi \in \mathbb{R}^2 \,:\, \xi \cdot n \leq 0} \Big\{ -\xi \cdot p - L(x,m,\xi) \Big\}, & \text{if } x \in \partial\Omega. \end{cases}$$

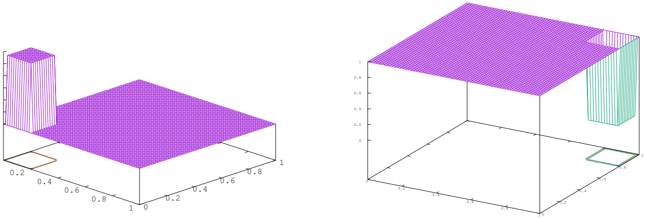
- The associated Lagrangian (corresponding to the running cost) is:

$$L(x,m,\xi) = (\beta - 1)\beta^{-\beta^*} m^{\frac{\alpha}{\beta-1}} |\xi|^{\beta^*} + \ell(x,m), \qquad 1 < \beta \leq 2, 0 \leq \alpha < 1$$

- Ex.: m_0 : & u_T : opposite corners; $\alpha = 0.01, \beta = 2, \ell(x,m) = 0.01m$.

Numerical Example: Congestion Without Viscosity

Results for the mean field control (MFC) problem, with $\nu = 0$

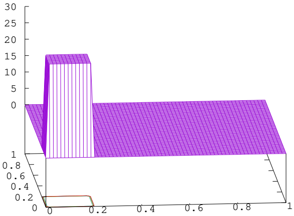


Initial distribution (left) and final cost (right)

For more details, see [Achdou and Laurière, 2016b]

Numerical Example: Congestion Without Viscosity

Results for the mean field control (MFC) problem, with $\nu = 0$

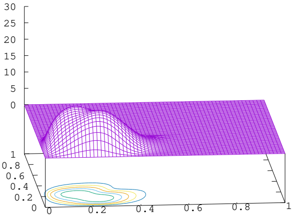


Density at time $t = 0$

For more details, see [Achdou and Laurière, 2016b]

Numerical Example: Congestion Without Viscosity

Results for the mean field control (MFC) problem, with $\nu = 0$

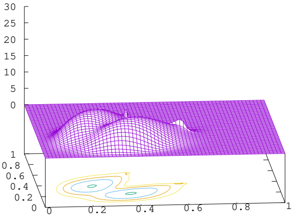


Density at time $t = T/8$

For more details, see [Achdou and Laurière, 2016b]

Numerical Example: Congestion Without Viscosity

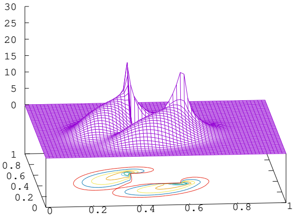
Results for the mean field control (MFC) problem, with $\nu = 0$



Density at time $t = T/4$

For more details, see [Achdou and Laurière, 2016b]

Results for the mean field control (MFC) problem, with $\nu = 0$

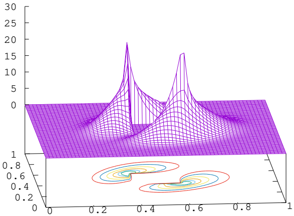


Density at time $t = 3T/8$

For more details, see [Achdou and Laurière, 2016b]

Numerical Example: Congestion Without Viscosity

Results for the mean field control (MFC) problem, with $\nu = 0$

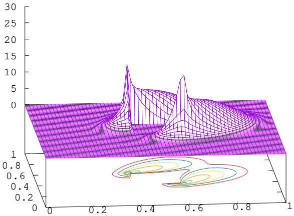


Density at time $t = T/2$

For more details, see [Achdou and Laurière, 2016b]

Numerical Example: Congestion Without Viscosity

Results for the mean field control (MFC) problem, with $\nu = 0$

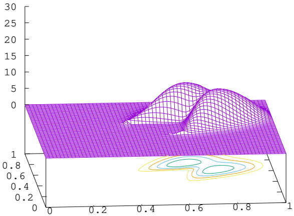


Density at time $t = 5T/8$

For more details, see [Achdou and Laurière, 2016b]

Numerical Example: Congestion Without Viscosity

Results for the mean field control (MFC) problem, with $\nu = 0$

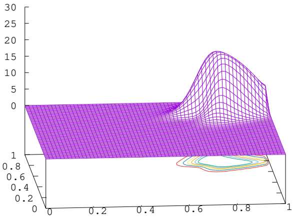


Density at time $t = 3T/4$

For more details, see [Achdou and Laurière, 2016b]

Numerical Example: Congestion Without Viscosity

Results for the mean field control (MFC) problem, with $\nu = 0$

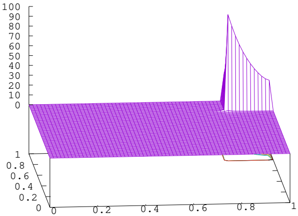


Density at time $t = 7T/8$

For more details, see [Achdou and Laurière, 2016b]

Numerical Example: Congestion Without Viscosity

Results for the mean field control (MFC) problem, with $\nu = 0$



Density at time $t = T$

For more details, see [Achdou and Laurière, 2016b]

Outline

1. Introduction

2. Methods for the PDE system

3. Optimization Methods for MFC and Variational MFG

- Variational MFGs and Duality
- Alternating Direction Method of Multipliers
- A Primal-Dual Method

4. Methods for MKV FBSDE

5. Conclusion

Optimality Conditions and Proximal Operator

- Let $\varphi, \psi \colon \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ be convex l.s.c. proper functions.
- Consider the optimization problem

$$\min_{y \in \mathbb{R}^N} \varphi(y) + \psi(y),$$

  and its dual

$$\min_{\sigma \in \mathbb{R}^N} \varphi^*(-\sigma) + \psi^*(\sigma).$$

Optimality Conditions and Proximal Operator

- Let $\varphi, \psi \colon \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ be convex l.s.c. proper functions.
- Consider the optimization problem

$$\min_{y \in \mathbb{R}^N} \varphi(y) + \psi(y),$$

and its dual

$$\min_{\sigma \in \mathbb{R}^N} \varphi^*(-\sigma) + \psi^*(\sigma).$$

- The 1^{st}-order opt. cond. satisfied by a solution $(\hat{y}, \hat{\sigma})$ are

$$\begin{cases} -\hat{\sigma} \in \partial\varphi(\hat{y}) \\ \hat{y} \in \partial\psi^*(\hat{\sigma}) \end{cases} \Leftrightarrow \begin{cases} \hat{y} - \tau\hat{\sigma} \in \tau\partial\varphi(\hat{y}) + \hat{y} \\ \hat{\sigma} + \gamma\hat{y} \in \gamma\partial\psi^*(\hat{\sigma}) + \hat{\sigma} \end{cases} \Leftrightarrow \begin{cases} \mathrm{prox}_{\tau\varphi}(\hat{y} - \tau\hat{\sigma}) = \hat{y} \\ \mathrm{prox}_{\gamma\psi^*}(\hat{\sigma} + \gamma\hat{y}) = \hat{\sigma}, \end{cases}$$

where $\gamma > 0$ and $\tau > 0$ are arbitrary and

- The **proximal operator** of a l.s.c. convex proper $\phi \colon \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ is:

$$\mathrm{prox}_{\gamma\phi}(x) := \operatorname*{argmin}_{y \in \mathbb{R}^N} \left\{ \phi(y) + \frac{|y-x|^2}{2\gamma} \right\} = (I + \partial(\gamma\phi))^{-1}(x), \qquad \forall\, x \in \mathbb{R}^N.$$

Chambolle-Pock's Primal-Dual Algorithm

The following algorithm has been proposed by [Chambolle and Pock, 2011]
It has been proved to converge when $\tau\gamma < 1$.

Input: Initial guess $(\sigma^{(0)}, y^{(0)}, \bar{y}^{(0)})$; $\theta \in [0, 1]$; $\gamma > 0, \tau > 0$; number of iterations $\mathtt{K}$
Output: Approximation of $(\hat{\sigma}, \hat{y})$ solving the optimality conditions

Initialize $(\sigma^{(0)}, y^{(0)}, \bar{y}^{(0)})$
for $\mathtt{k} = 0, 1, 2, \ldots, \mathtt{K} - 1$ **do**
  (a) Compute

$$\sigma^{(\mathtt{k}+1)} = \mathrm{prox}_{\gamma\psi^*}(\sigma^{(\mathtt{k})} + \gamma\bar{y}^{(\mathtt{k})}),$$

Chambolle-Pock's Primal-Dual Algorithm

The following algorithm has been proposed by [Chambolle and Pock, 2011]
It has been proved to converge when $\tau\gamma < 1$.

Input: Initial guess $(\sigma^{(0)}, y^{(0)}, \bar{y}^{(0)})$; $\theta \in [0,1]$; $\gamma > 0, \tau > 0$; number of iterations $\mathtt{K}$
Output: Approximation of $(\hat{\sigma}, \hat{y})$ solving the optimality conditions

Initialize $(\sigma^{(0)}, y^{(0)}, \bar{y}^{(0)})$
for $\mathtt{k} = 0, 1, 2, \ldots, \mathtt{K}-1$ **do**

(a) Compute

$$\sigma^{(\mathtt{k}+1)} = \mathrm{prox}_{\gamma\psi^*}(\sigma^{(\mathtt{k})} + \gamma\bar{y}^{(\mathtt{k})}),$$

(b) Compute

$$y^{(\mathtt{k}+1)} = \mathrm{prox}_{\tau\varphi}(y^{(\mathtt{k})} - \tau\sigma^{(\mathtt{k}+1)}),$$

Chambolle-Pock's Primal-Dual Algorithm

The following algorithm has been proposed by [Chambolle and Pock, 2011]
It has been proved to converge when $\tau\gamma < 1$.

Input: Initial guess $(\sigma^{(0)}, y^{(0)}, \bar{y}^{(0)})$; $\theta \in [0,1]$; $\gamma > 0, \tau > 0$; number of iterations $\mathtt{K}$
Output: Approximation of $(\hat{\sigma}, \hat{y})$ solving the optimality conditions

Initialize $(\sigma^{(0)}, y^{(0)}, \bar{y}^{(0)})$
for $\mathtt{k} = 0, 1, 2, \ldots, \mathtt{K} - 1$ **do**

- (a) Compute

$$\sigma^{(\mathtt{k}+1)} = \mathrm{prox}_{\gamma\psi^*}(\sigma^{(\mathtt{k})} + \gamma\bar{y}^{(\mathtt{k})}),$$

- (b) Compute

$$y^{(\mathtt{k}+1)} = \mathrm{prox}_{\tau\varphi}(y^{(\mathtt{k})} - \tau\sigma^{(\mathtt{k}+1)}),$$

- (c) Compute

$$\bar{y}^{(\mathtt{k}+1)} = y^{(\mathtt{k}+1)} + \theta(y^{(\mathtt{k}+1)} - y^{(\mathtt{k})}).$$

return $(\sigma^{(\mathtt{K})}, y^{(\mathtt{K})}, \bar{y}^{(\mathtt{K})})$

Dual of Discrete Problem $(\mathbf{A_h})$

By Fenchel-Rockafellar theorem, the dual problem of $(\mathbf{A_h})$ is:

$$(\mathbf{B_h}) = \min_{(m,w_1,w_2)=\sigma\in\mathbb{R}^{3N'}} \left\{ \mathcal{F}_h^*(\Lambda_h^*(\sigma)) + \mathcal{G}_h^*(-\sigma) \right\},$$

where $\mathcal{G}_h^*$ and $\mathcal{F}_h^*$ are respectively the Legendre-Fenchel conjugates of $\mathcal{G}_h$ and $\mathcal{F}_h$, defined by:

- $\mathcal{F}_h^*(\mu) = \sup_{\phi\in\mathbb{R}^N} \left\{ \langle \mu, \phi \rangle_{\ell^2(\mathbb{R}^N)} - \mathcal{F}_h(\phi) \right\}, \quad \forall\, \mu \in \mathbb{R}^N$
- $\mathcal{G}_h^*(-\sigma) = \max\limits_{q\in\mathbb{R}^{3N'}} \left\{ - \langle \sigma, q \rangle_{\ell^2(\mathbb{R}^{3N'})} - \mathcal{G}_h(q) \right\} = h\Delta t \sum_{n=1}^{N_T} \sum_{i=0}^{N_h-1} \tilde{L}_h(x_i, \sigma_i^n), \quad \forall\, \sigma \in \mathbb{R}^{3N'}$
- with $\tilde{L}_h(x, \sigma_0) = \max_{p_0\in\mathbb{R}^3} \left\{ - \sigma_0 \cdot p_0 + \mathcal{K}_h(x, q_0) \right\}, \quad \forall \sigma_0 \in \mathbb{R}^3.$

Dual of Discrete Problem $(\mathbf{A_h})$

By Fenchel-Rockafellar theorem, the dual problem of $(\mathbf{A_h})$ is:

$$(\mathbf{B_h}) = \min_{(m,w_1,w_2)=\sigma\in\mathbb{R}^{3N'}} \left\{\mathcal{F}_h^*(\Lambda_h^*(\sigma)) + \mathcal{G}_h^*(-\sigma)\right\},$$

where $\mathcal{G}_h^*$ and $\mathcal{F}_h^*$ are respectively the Legendre-Fenchel conjugates of $\mathcal{G}_h$ and $\mathcal{F}_h$, defined by:

- $\mathcal{F}_h^*(\mu) = \sup_{\phi\in\mathbb{R}^N} \left\{\langle \mu, \phi\rangle_{\ell^2(\mathbb{R}^N)} - \mathcal{F}_h(\phi)\right\}, \quad \forall\, \mu \in \mathbb{R}^N$
- $\mathcal{G}_h^*(-\sigma) = \max_{q\in\mathbb{R}^{3N'}} \left\{ - \langle \sigma, q\rangle_{\ell^2(\mathbb{R}^{3N'})} - \mathcal{G}_h(q)\right\} = h\Delta t \sum_{n=1}^{N_T} \sum_{i=0}^{N_h-1} \tilde{L}_h(x_i, \sigma_i^n), \quad \forall\, \sigma \in \mathbb{R}^{3N'}$
- with $\tilde{L}_h(x,\sigma_0) = \max_{p_0\in\mathbb{R}^3} \left\{ - \sigma_0 \cdot p_0 + \mathcal{K}_h(x, q_0)\right\}, \quad \forall \sigma_0 \in \mathbb{R}^3.$

Rem.: The max can be costly to compute but in some cases $\tilde{L}_h$ has a **closed-form** expression.

Finally $\Lambda_h^* : \mathbb{R}^{3N'} \to \mathbb{R}^N$ denotes the adjoint of Λ_h: for all $(m, y, z) \in \mathbb{R}^{3N'}, \phi \in \mathbb{R}^N$:

$$\langle \Lambda_h^*(m, y, z), \phi\rangle_{\ell^2(\mathbb{R}^N)} = \langle (m, y, z), \Lambda_h(\phi)\rangle_{\ell^2(\mathbb{R}^{3N'})}$$

Dual of Discrete Problem $(\mathbf{A_h})$

By Fenchel-Rockafellar theorem, the dual problem of $(\mathbf{A_h})$ is:

$$(\mathbf{B_h}) = \min_{(m,w_1,w_2)=\sigma\in\mathbb{R}^{3N'}} \left\{\mathcal{F}_h^*(\Lambda_h^*(\sigma)) + \mathcal{G}_h^*(-\sigma)\right\},$$

where $\mathcal{G}_h^*$ and $\mathcal{F}_h^*$ are respectively the Legendre-Fenchel conjugates of $\mathcal{G}_h$ and $\mathcal{F}_h$, defined by:

- $\mathcal{F}_h^*(\mu) = \sup_{\phi\in\mathbb{R}^N} \left\{\langle \mu, \phi\rangle_{\ell^2(\mathbb{R}^N)} - \mathcal{F}_h(\phi)\right\}, \quad \forall\, \mu \in \mathbb{R}^N$
- $\mathcal{G}_h^*(-\sigma) = \max_{q\in\mathbb{R}^{3N'}} \left\{ -\langle \sigma, q\rangle_{\ell^2(\mathbb{R}^{3N'})} - \mathcal{G}_h(q)\right\} = h\Delta t \sum_{n=1}^{N_T} \sum_{i=0}^{N_h-1} \tilde{L}_h(x_i, \sigma_i^n), \quad \forall\, \sigma \in \mathbb{R}^{3N'}$
- with $\tilde{L}_h(x,\sigma_0) = \max_{p_0\in\mathbb{R}^3} \left\{ -\sigma_0\cdot p_0 + \mathcal{K}_h(x,q_0)\right\}, \quad \forall \sigma_0 \in \mathbb{R}^3.$

Rem.: The max can be costly to compute but in some cases $\tilde{L}_h$ has a **closed-form** expression.

Finally $\Lambda_h^* : \mathbb{R}^{3N'} \to \mathbb{R}^N$ denotes the adjoint of Λ_h: for all $(m, y, z) \in \mathbb{R}^{3N'}, \phi \in \mathbb{R}^N$:

$$\langle \Lambda_h^*(m,y,z), \phi\rangle_{\ell^2(\mathbb{R}^N)} = \langle (m,y,z), \Lambda_h(\phi)\rangle_{\ell^2(\mathbb{R}^{3N'})}$$

Rem.: We have $\mathcal{F}_h^*(\Lambda_h^*(m,y,z)) = \begin{cases} h\sum_{i=0}^{N_h-1} m_i^{N_T}\, \mathfrak{g}_0(x_i), & \text{if } (m,y,z) \text{ satisfies } (\star) \text{ below,} \\ +\infty, & \text{otherwise,}\end{cases}$

with $\forall\, i \in \{0,\dots,N_h-1\}$, $m_i^0 = \rho_i^0$, and $\forall\, n \in \{0,\dots,N_T-1\}$:

$$(D_t m_i)^n - \nu\left(\Delta_h m^{n+1}\right)_i + \frac{y_i^{n+1} - y_{i-1}^{n+1}}{h} + \frac{z_{i+1}^{n+1} - z_i^{n+1}}{h} = 0. \qquad (\star)$$

Reformulation

The discrete dual problem can be recast as:

$$\inf_{(m,w)} \underbrace{\mathbb{B}_h(m,w) + \mathbb{F}_h(m)}_{\varphi(m,w)} + \underbrace{\iota_{\mathbb{G}^{-1}(\rho^0,0)}(m,w)}_{\psi(m,w)} \tag{P_h}$$

with the costs

$$\mathbb{F}_h(m) := \sum_{i,n} \widetilde{F}(x_i, m_i^n) + \frac{1}{\Delta t} \sum_i \widetilde{G}(x_i, m_i^{N_T}), \qquad \mathbb{B}_h(m,w) := \sum_{i,n} \hat{b}(m_i^n, w_i^{n-1}),$$

$$\hat{b}(m,w) := \begin{cases} mL\left(x, -\frac{w}{m}\right), & \text{if } m > 0, w \in K = \mathbb{R}_- \times \mathbb{R}_+, \\ 0, & \text{if } (m,w) = (0,0), \\ +\infty, & \text{otherwise,} \end{cases}$$

and $\mathbb{G}(m,w) := (m_0, (Am^{n+1} + Bw^n)_{0 \le n \le N_T - 1})$ with

$$(Am)_i^{n+1} := (D_t m)_i^n - \nu(\Delta_h m)_i^{n+1}, \qquad (Bw)_i^n := (D_h w^1)_{i-1}^n + (D_h w^2)_i^n.$$

Reformulation

The discrete dual problem can be recast as:

$$\inf_{(m,w)} \underbrace{\mathbb{B}_h(m,w) + \mathbb{F}_h(m)}_{\varphi(m,w)} + \underbrace{\iota_{\mathbb{G}^{-1}(\rho^0,0)}(m,w)}_{\psi(m,w)} \qquad (P_h)$$

with the costs

$$\mathbb{F}_h(m) := \sum_{i,n} \widetilde{F}(x_i, m_i^n) + \frac{1}{\Delta t} \sum_i \widetilde{G}(x_i, m_i^{N_T}), \qquad \mathbb{B}_h(m,w) := \sum_{i,n} \hat{b}(m_i^n, w_i^{n-1}),$$

$$\hat{b}(m,w) := \begin{cases} mL\left(x, -\frac{w}{m}\right), & \text{if } m > 0, w \in K = \mathbb{R}_- \times \mathbb{R}_+, \\ 0, & \text{if } (m,w) = (0,0), \\ +\infty, & \text{otherwise,} \end{cases}$$

and $\mathbb{G}(m,w) := (m_0, (Am^{n+1} + Bw^n)_{0 \le n \le N_T - 1})$ with

$$(Am)_i^{n+1} := (D_t m)_i^n - \nu(\Delta_h m)_i^{n+1}, \qquad (Bw)_i^n := (D_h w^1)_{i-1}^n + (D_h w^2)_i^n.$$

Rem.: The optimality conditions of this problem correspond to the **finite-difference system**

So we can apply **Chambolle-Pock**'s method for (P_h) with

$$y = (m,w), \qquad \varphi(m,w) = \mathbb{B}_h(m,w) + \mathbb{F}_h(m), \qquad \psi(m,w) = \iota_{\mathbb{G}^{-1}(\rho^0,0)}(m,w)$$

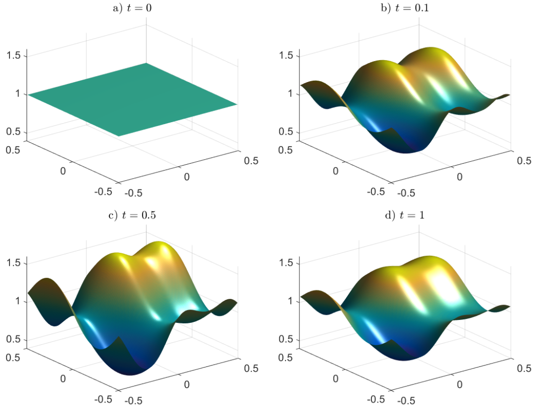
See [Briceño Arias et al., 2018] and [Briceño Arias et al., 2019] in stationary and dynamic cases.

Numerical Example

Setting: $g \equiv 0$ and $\mathbb{R}^2 \times \mathbb{R} \ni (x, m) \mapsto f(x, m) := m^2 - \overline{H}(x)$, with

$$\overline{H}(x) = \sin(2\pi x_2) + \sin(2\pi x_1) + \cos(2\pi x_1)$$

We solve the corresponding MFG and obtain the following evolution of the density:



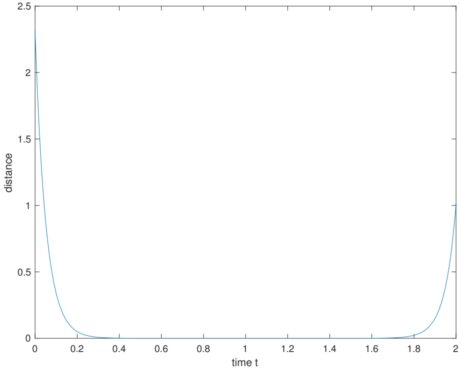
Evolution of the density

More details in [Briceño Arias et al., 2019]

Turnpike phenomenon

This example also illustrates the **turnpike phenomenon**, see e.g. [Porretta and Zuazua, 2013]

- the mass starts from an initial density;
- it **converges to a steady state**, influenced only by the running cost;
- as $t \to T$, the mass is influenced by the final cost and **converges to a final state**.



L^2 **distance between dynamic and stationary solutions**

More details in [Briceño Arias et al., 2019]

Outline

MKV FBSDE System

- Recall: generic form:

$$\begin{cases} dX_t = B(X_t, \mathcal{L}(X_t), Y_t, Z_t)dt + \sigma dW_t, & 0 \leq t \leq T \\ dY_t = -F(X_t, \mathcal{L}(X_t), Y_t, Z_t)dt + Z_t dW_t, & 0 \leq t \leq T \\ X_0 \sim m_0, \qquad Y_T = G(X_T, \mathcal{L}(X_T)) \end{cases}$$

- Decouple:
  - Given $(\mathcal{L}(X), Y, Z)$, solve for X
  - Given $(X, \mathcal{L}(X))$ solve for (Y, Z)
- Iterate
- Algorithm proposed by [Chassagneux et al., 2019, Angiuli et al., 2019]

Picard Scheme for MKV FBSDE System

Algorithm: Picard scheme for MKV FBSDE

Input: Initial guess (ξ, ζ); initial condition ξ; terminal condition ζ; time horizon T; number of iterations $\mathtt{K}$

Output: Approximation of (X, Y, Z) solving the MKV FBSDE system

Initialize $X_t^{(0)} = \xi, Y_t^{(0)} = 0, Z_t^{(0)} = 0, 0 \leq t \leq T$

for $\mathtt{k} = 0, 1, 2, \ldots, \mathtt{K} - 1$ **do**

  Let $X^{(\mathtt{k}+1)}$ be the solution to:

$$\begin{cases} dX_t = B(X_t^{(\mathtt{k})}, \mathcal{L}(X_t^{(\mathtt{k})}), Y_t^{(\mathtt{k})}, Z_t^{(\mathtt{k})})dt + \sigma dW_t, & 0 \leq t \leq T \\ X_0 = \xi \end{cases}$$

Picard Scheme for MKV FBSDE System

Algorithm: Picard scheme for MKV FBSDE

Input: Initial guess (ξ, ζ); initial condition ξ; terminal condition ζ; time horizon T; number of iterations $\mathtt{K}$

Output: Approximation of (X, Y, Z) solving the MKV FBSDE system

Initialize $X_t^{(0)} = \xi, Y_t^{(0)} = 0, Z_t^{(0)} = 0, 0 \leq t \leq T$

for $\mathtt{k} = 0, 1, 2, \ldots, \mathtt{K} - 1$ **do**

Let $X^{(\mathtt{k}+1)}$ be the solution to:

$$\begin{cases} dX_t = B(X_t^{(\mathtt{k})}, \mathcal{L}(X_t^{(\mathtt{k})}), Y_t^{(\mathtt{k})}, Z_t^{(\mathtt{k})})dt + \sigma dW_t, & 0 \leq t \leq T \\ X_0 = \xi \end{cases}$$

Let $(Y^{(\mathtt{k}+1)}, Z^{(\mathtt{k}+1)})$ be the solution to:

$$\begin{cases} dY_t = -F(X_t^{(\mathtt{k}+1)}, \mathcal{L}(X_t^{(\mathtt{k}+1)}), Y_t^{(\mathtt{k})}, Z_t^{(\mathtt{k})})dt + Z_t^{(\mathtt{k})} dW_t, & 0 \leq t \leq T \\ Y_T = \zeta \end{cases}$$

return $\mathtt{Picard}[T](\xi, \zeta) = (X^{(\mathtt{K})}, Y^{(\mathtt{K})}, Z^{(\mathtt{K})})$

Algorithm: Picard scheme for MKV FBSDE

Input: Initial guess (ξ, ζ); initial condition ξ; terminal condition ζ; time horizon T; number of iterations $\mathtt{K}$

Output: Approximation of (X, Y, Z) solving the MKV FBSDE system

Initialize $X_t^{(0)} = \xi, Y_t^{(0)} = 0, Z_t^{(0)} = 0, 0 \leq t \leq T$

for $\mathtt{k} = 0, 1, 2, \ldots, \mathtt{K} - 1$ **do**

Let $X^{(\mathtt{k}+1)}$ be the solution to:

$$\begin{cases} dX_t = B(X_t^{(\mathtt{k})}, \mathcal{L}(X_t^{(\mathtt{k})}), Y_t^{(\mathtt{k})}, Z_t^{(\mathtt{k})})dt + \sigma dW_t, & 0 \leq t \leq T \\ X_0 = \xi \end{cases}$$

Let $(Y^{(\mathtt{k}+1)}, Z^{(\mathtt{k}+1)})$ be the solution to:

$$\begin{cases} dY_t = -F(X_t^{(\mathtt{k}+1)}, \mathcal{L}(X_t^{(\mathtt{k}+1)}), Y_t^{(\mathtt{k})}, Z_t^{(\mathtt{k})})dt + Z_t^{(\mathtt{k})} dW_t, & 0 \leq t \leq T \\ Y_T = \zeta \end{cases}$$

return $\mathtt{Picard}[T](\xi, \zeta) = (X^{(\mathtt{K})}, Y^{(\mathtt{K})}, Z^{(\mathtt{K})})$

Notation: $\Phi_{\xi,\zeta} : (X^{(\mathtt{k})}, \mathcal{L}(X^{(\mathtt{k})}), Y^{(\mathtt{k})}, Z^{(\mathtt{k})}) \mapsto (X^{(\mathtt{k}+1)}, \mathcal{L}(X^{(\mathtt{k}+1)}), Y^{(\mathtt{k}+1)}, Z^{(\mathtt{k}+1)})$

Algorithm: Picard scheme for MKV FBSDE

Input: Initial guess (ξ, ζ); initial condition ξ; terminal condition ζ; time horizon T; number of iterations $\mathtt{K}$

Output: Approximation of (X, Y, Z) solving the MKV FBSDE system

Initialize $X_t^{(0)} = \xi, Y_t^{(0)} = 0, Z_t^{(0)} = 0, 0 \leq t \leq T$

for $\mathtt{k} = 0, 1, 2, \ldots, \mathtt{K} - 1$ **do**

Let $X^{(\mathtt{k}+1)}$ be the solution to:

$$\begin{cases} dX_t = B(X_t^{(\mathtt{k})}, \mathcal{L}(X_t^{(\mathtt{k})}), Y_t^{(\mathtt{k})}, Z_t^{(\mathtt{k})})dt + \sigma dW_t, & 0 \leq t \leq T \\ X_0 = \xi \end{cases}$$

Let $(Y^{(\mathtt{k}+1)}, Z^{(\mathtt{k}+1)})$ be the solution to:

$$\begin{cases} dY_t = -F(X_t^{(\mathtt{k}+1)}, \mathcal{L}(X_t^{(\mathtt{k}+1)}), Y_t^{(\mathtt{k})}, Z_t^{(\mathtt{k})})dt + Z_t^{(\mathtt{k})} dW_t, & 0 \leq t \leq T \\ Y_T = \zeta \end{cases}$$

return $\mathtt{Picard}[T](\xi, \zeta) = (X^{(\mathtt{K})}, Y^{(\mathtt{K})}, Z^{(\mathtt{K})})$

Notation: $\Phi_{\xi,\zeta} : (X^{(\mathtt{k})}, \mathcal{L}(X^{(\mathtt{k})}), Y^{(\mathtt{k})}, Z^{(\mathtt{k})}) \mapsto (X^{(\mathtt{k}+1)}, \mathcal{L}(X^{(\mathtt{k}+1)}), Y^{(\mathtt{k}+1)}, Z^{(\mathtt{k}+1)})$

Contraction? Small T or small Lipschitz constants for B, F, G

Continuation Method

- If T is big: Solve FBSDE on small intervals & "patch" the solutions together

Continuation Method

- If T is big: Solve FBSDE on small intervals & “patch” the solutions together
- Grid: $0 = T_0 < T_1 < \cdots < T_{M-1} < T_M = T$
- Subproblem: Given $(\xi_{T_m}, \mathcal{L}(\xi_{T_m}))$ and $\zeta_{T_{m+1}}$, solve:

$$\begin{cases} dX_t = B(X_t, \mathcal{L}(X_t), Y_t, Z_t)dt + \sigma dW_t, & T_m \le t \le T_{m+1} \\ dY_t = -F(X_t, \mathcal{L}(X_t), Y_t, Z_t)dt + Z_t dW_t, & T_m \le t \le T_{m+1} \\ X_{T_m} = \xi_{T_m}, \qquad Y_{T_{m+1}} = \zeta_{T_{m+1}} \end{cases}$$

Continuation Method

- If T is big: Solve FBSDE on small intervals & "patch" the solutions together
- Grid: $0 = T_0 < T_1 < \cdots < T_{M-1} < T_M = T$
- Subproblem: Given $(\xi_{T_m}, \mathcal{L}(\xi_{T_m}))$ and $\zeta_{T_{m+1}}$, solve:

$$\begin{cases} dX_t = B(X_t, \mathcal{L}(X_t), Y_t, Z_t)dt + \sigma dW_t, & T_m \leq t \leq T_{m+1} \\ dY_t = -F(X_t, \mathcal{L}(X_t), Y_t, Z_t)dt + Z_t dW_t, & T_m \leq t \leq T_{m+1} \\ X_{T_m} = \xi_{T_m}, \qquad Y_{T_{m+1}} = \zeta_{T_{m+1}} \end{cases}$$

- How to find ξ_{T_m} and $\zeta_{T_{m+1}}$?
  - $\to$ ξ_{T_m} from previous problem's solution (or initial condition)
  - $\to$ $\zeta_{T_{m+1}}$ from next problem's solution (or terminal condition)

Global Solver for MKV FBSDE System

Following [Chassagneux et al., 2019], define a global solver recursively, and then call:

$$\texttt{Solver}[m](\xi_0, \mu_0)$$

with ξ_0 a random variable with distribution μ_0

Input: Initial guess $(\xi, \mathcal{L}(\xi))$; time step index m; number of iterations $\texttt{K}$

Output: Approximation of Y_{T_m} where (X, Y, Z) solves the MKV FBSDE system on $[T_m, T]$ starting with $(\xi, \mathcal{L}(\xi))$ at time T_m

Initialize $X_t^{(0)} = \xi, \mathcal{L}(X_t^{(0)}) = \mathcal{L}(\xi)$ for all $T_m \leq t \leq T_{m+1}$

for $\texttt{k} = 0, 1, 2, \ldots, \texttt{K} - 1$ **do**

  If $T_{m+1} = T$, $Y_{T_{m+1}}^{(\texttt{k}+1)} = G(X_{T_{m+1}}^{(\texttt{k})}, \mathcal{L}(X_{T_{m+1}}^{(\texttt{k})}))$

Global Solver for MKV FBSDE System

Following [Chassagneux et al., 2019], define a global solver recursively, and then call:

$$\texttt{Solver}[m](\xi_0, \mu_0)$$

with ξ_0 a random variable with distribution μ_0

Input: Initial guess $(\xi, \mathcal{L}(\xi))$; time step index m; number of iterations $\texttt{K}$
Output: Approximation of Y_{T_m} where (X, Y, Z) solves the MKV FBSDE system on $[T_m, T]$ starting with $(\xi, \mathcal{L}(\xi))$ at time T_m

1 Initialize $X_t^{(0)} = \xi, \mathcal{L}(X_t^{(0)}) = \mathcal{L}(\xi)$ for all $T_m \leq t \leq T_{m+1}$
2 **for** $\texttt{k} = 0, 1, 2, \ldots, \texttt{K} - 1$ **do**
3 If $T_{m+1} = T$, $Y_{T_{m+1}}^{(\texttt{k}+1)} = G(X_{T_{m+1}}^{(\texttt{k})}, \mathcal{L}(X_{T_{m+1}}^{(\texttt{k})}))$
4 Else: compute recursively:

$$Y_{T_{m+1}}^{(\texttt{k}+1)} = \texttt{Solver}[m+1](X_{T_{m+1}}^{(\texttt{k})}, \mathcal{L}(X_{T_{m+1}}^{(\texttt{k})}))$$

Global Solver for MKV FBSDE System

Following [Chassagneux et al., 2019], define a global solver recursively, and then call:

$$\texttt{Solver}[m](\xi_0, \mu_0)$$

with ξ_0 a random variable with distribution μ_0

Input: Initial guess $(\xi, \mathcal{L}(\xi))$; time step index m; number of iterations $\texttt{K}$
Output: Approximation of Y_{T_m} where (X, Y, Z) solves the MKV FBSDE system on $[T_m, T]$ starting with $(\xi, \mathcal{L}(\xi))$ at time T_m

Initialize $X_t^{(0)} = \xi, \mathcal{L}(X_t^{(0)}) = \mathcal{L}(\xi)$ for all $T_m \le t \le T_{m+1}$
for $\texttt{k} = 0, 1, 2, \dots, \texttt{K} - 1$ **do**
  If $T_{m+1} = T$, $Y_{T_{m+1}}^{(\texttt{k}+1)} = G(X_{T_{m+1}}^{(\texttt{k})}, \mathcal{L}(X_{T_{m+1}}^{(\texttt{k})}))$
  Else: compute recursively:

$$Y_{T_{m+1}}^{(\texttt{k}+1)} = \texttt{Solver}[m+1](X_{T_{m+1}}^{(\texttt{k})}, \mathcal{L}(X_{T_{m+1}}^{(\texttt{k})}))$$

  Compute:

$$(X_t^{(\texttt{k}+1)}, \mathcal{L}(X_t^{(\texttt{k}+1)}), Y_t^{(\texttt{k}+1)}, Z_t^{(\texttt{k}+1)})_{T_m \le t \le T_{m+1}} = \texttt{Picard}[T_{m+1} - T_m](X_{T_m}^{(\texttt{k})}, Y_{T_{m+1}}^{(\texttt{k}+1)})$$

return $\texttt{Solver}[m](\xi, \mathcal{L}(\xi)) := Y_{T_m}^{(\texttt{K})}$

Implementation: Discretizations

In the sequel, we present two algorithms, following [Angiuli et al., 2019]

- Tree algorithm:
  - Time discretization
  - Space discretization: binomial tree structure
  - Look at trajectories

- Grid algorithm:
  - Time and space discretization on a grid
  - Look at time marginals

Tree-Based Algorithm: Time Discretization

- Focus on an interval $[0, T]$ with small enough T (otherwise: call recursive solver)

Tree-Based Algorithm: Time Discretization

- Focus on an interval $[0, T]$ with small enough T (otherwise: call recursive solver)
- Time discretization: $0 = t_0 < t_1 < \cdots < t_{N_t} = T$, $t_{i+1} - t_i = \Delta t$
- Euler Scheme: $0 \leq i \leq N_t - 1$

$$
\begin{cases}
X_{t_{i+1}}^{(\mathtt{k}+1)} = X_{t_i}^{(\mathtt{k}+1)} + B(X_{t_i}^{(\mathtt{k}+1)}, \mathcal{L}(X_{t_i}^{(\mathtt{k}+1)}), Y_{t_i}^{(\mathtt{k})}, Z_{t_i}^{(\mathtt{k})})\Delta t + \sigma \Delta W_{t_{i+1}} \\
X_0^{(\mathtt{k}+1)} = \xi \\
Y_{t_i}^{(\mathtt{k}+1)} = \mathbb{E}_{t_i}[Y_{t_{i+1}}^{(\mathtt{k}+1)}] + F(X_{t_i}^{(\mathtt{k}+1)}, \mathcal{L}(X_{t_i}^{(\mathtt{k}+1)}), Y_{t_i}^{(\mathtt{k})}, Z_{t_i}^{(\mathtt{k})})\Delta t \\
\qquad \approx Y_{t_{i+1}}^{(\mathtt{k}+1)} + F(X_{t_i}^{(\mathtt{k}+1)}, \mathcal{L}(X_{t_i}^{(\mathtt{k}+1)}), Y_{t_i}^{(\mathtt{k})}, Z_{t_i}^{(\mathtt{k})})\Delta t - Z_{t_i}^{(\mathtt{k}+1)} \Delta W_{t_{i+1}} \\
Y_T^{(\mathtt{k}+1)} = G(X_T^{(\mathtt{k}+1)}, \mathcal{L}(X_T^{(\mathtt{k}+1)})) \\
Z_{t_i}^{(\mathtt{k}+1)} = \frac{1}{\Delta t} \mathbb{E}_{t_i}[Y_{t_{i+1}}^{(\mathtt{k}+1)} \Delta W_{t_{i+1}}] \\
Z_T^{(\mathtt{k}+1)} = 0
\end{cases}
$$

Tree-Based Algorithm: Time Discretization

- Focus on an interval $[0, T]$ with small enough T (otherwise: call recursive solver)
- Time discretization: $0 = t_0 < t_1 < \cdots < t_{N_t} = T$, $t_{i+1} - t_i = \Delta t$
- Euler Scheme: $0 \le i \le N_t - 1$

$$
\begin{cases}
X_{t_{i+1}}^{(k+1)} = X_{t_i}^{(k+1)} + B(X_{t_i}^{(k+1)}, \mathcal{L}(X_{t_i}^{(k+1)}), Y_{t_i}^{(k)}, Z_{t_i}^{(k)})\Delta t + \sigma \Delta W_{t_{i+1}} \\
X_0^{(k+1)} = \xi \\
Y_{t_i}^{(k+1)} = \mathbb{E}_{t_i}[Y_{t_{i+1}}^{(k+1)}] + F(X_{t_i}^{(k+1)}, \mathcal{L}(X_{t_i}^{(k+1)}), Y_{t_i}^{(k)}, Z_{t_i}^{(k)})\Delta t \\
\qquad \approx Y_{t_{i+1}}^{(k+1)} + F(X_{t_i}^{(k+1)}, \mathcal{L}(X_{t_i}^{(k+1)}), Y_{t_i}^{(k)}, Z_{t_i}^{(k)})\Delta t - Z_{t_i}^{(k+1)} \Delta W_{t_{i+1}} \\
Y_T^{(k+1)} = G(X_T^{(k+1)}, \mathcal{L}(X_T^{(k+1)})) \\
Z_{t_i}^{(k+1)} = \frac{1}{\Delta t} \mathbb{E}_{t_i}[Y_{t_{i+1}}^{(k+1)} \Delta W_{t_{i+1}}] \\
Z_T^{(k+1)} = 0
\end{cases}
$$

- Questions:
  - How to represent $\mathcal{L}(X_{t_i}^{(k+1)})$?
  - How to compute the conditional expectation $\mathbb{E}_{t_i}[Y_{t_{i+1}}^{(k+1)}]$?

Tree-Based Algorithm: Remarks

- At each t_i, replace $\Delta W_{t_{i+1}}$ by a branch with 2 values: $\pm\sqrt{\Delta t}$ w.p. $1/2$
- Answers:
  - $\mathcal{L}(X_{t_i}^{(\mathtt{k}+1)}) \approx$ weighted empirical distribution:

$$\mathcal{L}(X_{t_0}^{(\mathtt{k}+1)}) \approx \sum_{n=1}^{N_{x_0}} p_0^k \delta_{x_0^k},$$

    and at time $t_i, i \geq 1$: look at values on the nodes at depth i
  - $\mathbb{E}_{t_i}[Y_{t_{i+1}}^{(\mathtt{k}+1)}] \approx$ weighted average of values on the two next branches

Tree-Based Algorithm: Remarks

- At each t_i, replace $\Delta W_{t_{i+1}}$ by a branch with 2 values: $\pm\sqrt{\Delta t}$ w.p. $1/2$
- Answers:
  - $\mathcal{L}(X_{t_i}^{(\mathtt{k}+1)}) \approx$ weighted empirical distribution:

$$\mathcal{L}(X_{t_0}^{(\mathtt{k}+1)}) \approx \sum_{n=1}^{N_{x_0}} p_0^k \delta_{x_0^k},$$

    and at time $t_i, i \geq 1$: look at values on the nodes at depth i
  - $\mathbb{E}_{t_i}[Y_{t_{i+1}}^{(\mathtt{k}+1)}] \approx$ weighted average of values on the two next branches
- Starting from some x_0, doing N_t steps: 2^{N_t} paths
- N_{x_0} starting points i.i.d. $\sim \mu_0$: $N_{x_0} \times 2^{N_t}$ paths !

Tree-Based Algorithm: Remarks

- At each t_i, replace $\Delta W_{t_{i+1}}$ by a branch with 2 values: $\pm\sqrt{\Delta t}$ w.p. $1/2$
- Answers:
  - $\mathcal{L}(X_{t_i}^{(\mathtt{k}+1)}) \approx$ weighted empirical distribution:

$$\mathcal{L}(X_{t_0}^{(\mathtt{k}+1)}) \approx \sum_{n=1}^{N_{x_0}} p_0^k \delta_{x_0^k},$$

    and at time $t_i, i \geq 1$: look at values on the nodes at depth i
  - $\mathbb{E}_{t_i}[Y_{t_{i+1}}^{(\mathtt{k}+1)}] \approx$ weighted average of values on the two next branches
- Starting from some x_0, doing N_t steps: 2^{N_t} paths
- N_{x_0} starting points i.i.d. $\sim \mu_0$: $N_{x_0} \times 2^{N_t}$ paths !
- Save space thanks to recombinations?

Tree-Based Algorithm: Remarks

- At each t_i, replace $\Delta W_{t_{i+1}}$ by a branch with 2 values: $\pm\sqrt{\Delta t}$ w.p. $1/2$
- Answers:
  - $\mathcal{L}(X_{t_i}^{(\mathtt{k}+1)}) \approx$ weighted empirical distribution:

$$\mathcal{L}(X_{t_0}^{(\mathtt{k}+1)}) \approx \sum_{n=1}^{N_{x_0}} p_0^k \delta_{x_0^k},$$

    and at time $t_i, i \geq 1$: look at values on the nodes at depth i
  - $\mathbb{E}_{t_i}[Y_{t_{i+1}}^{(\mathtt{k}+1)}] \approx$ weighted average of values on the two next branches
- Starting from some x_0, doing N_t steps: 2^{N_t} paths
- N_{x_0} starting points i.i.d. $\sim \mu_0$: $N_{x_0} \times 2^{N_t}$ paths !
- Save space thanks to recombinations? *Not really but ...*

Grid-Based Algorithm: Time & Space Discretization

- Decoupling functions (see e.g., Section 6.4 in [Carmona and Delarue, 2018]):

$$Y_t = u(t, X_t, \mathcal{L}(X_t)), \qquad Z_t = v(t, X_t, \mathcal{L}(X_t))$$

$\rightarrow$ Approximate $u(\cdot,\cdot,\cdot), v(\cdot,\cdot,\cdot)$ instead of $(Y_t, Z_t)_{t\in[0,T]}$

Grid-Based Algorithm: Time & Space Discretization

- Decoupling functions (see e.g., Section 6.4 in [Carmona and Delarue, 2018]):

$$Y_t = u(t, X_t, \mathcal{L}(X_t)), \qquad Z_t = v(t, X_t, \mathcal{L}(X_t))$$

$\rightarrow$ Approximate $u(\cdot,\cdot,\cdot), v(\cdot,\cdot,\cdot)$ instead of $(Y_t, Z_t)_{t\in[0,T]}$

- Difficulty: space of $\mathcal{L}(X_t)$ is infinite dimensional
$\rightarrow$ Freeze it during each Picard iteration:

$$Y_t^{(\mathtt{k}+1)} = u^{(\mathtt{k}+1)}(t, X_t^{(\mathtt{k}+1)}), \qquad Z_t^{(\mathtt{k}+1)} = v^{(\mathtt{k}+1)}(t, X_t^{(\mathtt{k}+1)}) \qquad (\star)$$

Grid-Based Algorithm: Time & Space Discretization

- Decoupling functions (see e.g., Section 6.4 in [Carmona and Delarue, 2018]):

$$Y_t = u(t, X_t, \mathcal{L}(X_t)), \qquad Z_t = v(t, X_t, \mathcal{L}(X_t))$$

$\rightarrow$ Approximate $u(\cdot,\cdot,\cdot), v(\cdot,\cdot,\cdot)$ instead of $(Y_t, Z_t)_{t\in[0,T]}$

- Difficulty: space of $\mathcal{L}(X_t)$ is infinite dimensional
  $\rightarrow$ Freeze it during each Picard iteration:

$$Y_t^{(k+1)} = u^{(k+1)}(t, X_t^{(k+1)}), \qquad Z_t^{(k+1)} = v^{(k+1)}(t, X_t^{(k+1)}) \qquad (\star)$$

- Picard iterations for distribution & decoupling functions:
  - **Step 1:** Given $(\mu^{(k)}, u^{(k)}, v^{(k)})$, compute $\mu_t^{(k+1)} = \mathcal{L}(X_t^{(k+1)}), 0 \le t \le T$, where

$$dX_t^{(k+1)} = B\Big(X_t^{(k+1)}, \mu_t^{(k)}, u^{(k)}(t, X_t^{(k+1)}), v^{(k)}(t, X_t^{(k+1)})\Big)dt + \sigma dW_t$$

  - **Step 2:** Given $(X^{(k)}, \mu^{(k+1)})$, compute $(u^{(k+1)}, v^{(k+1)})$ such that $(\star)$ holds, where

$$dY_t^{(k+1)} = -F\Big(X_t^{(k+1)}, \mu_t^{(k+1)}, Y_t^{(k+1)}, Z_t^{(k+1)}\Big)dt + Z_t^{(k+1)}dW_t$$

  - Return $(\mu^{(k+1)}, u^{(k+1)}, v^{(k+1)})$

Grid-Based Algorithm: Forward Equation

- Focus on an interval $[0, T]$ with small enough T (otherwise: call recursive solver)
- Time discretization: $0 = t_0 < t_1 < \cdots < t_{N_t} = T$, $t_{i+1} - t_i = \Delta t$
- Space discretization ($d = 1$): Grid Γ: $x_0 < x_1 < \cdots < x_{N_x}$, $x_{j+1} - x_j = \Delta x$

Grid-Based Algorithm: Forward Equation

- Focus on an interval $[0, T]$ with small enough T (otherwise: call recursive solver)
- Time discretization: $0 = t_0 < t_1 < \cdots < t_{N_t} = T$, $t_{i+1} - t_i = \Delta t$
- Space discretization ($d = 1$): Grid Γ: $x_0 < x_1 < \cdots < x_{N_x}$, $x_{j+1} - x_j = \Delta x$
- Use projection Π to stay on Γ at every t_i: $\mathcal{L}(X_{t_i}^{(\mathtt{k}+1)}) \approx$ vector of weights

Grid-Based Algorithm: Forward Equation

- Focus on an interval $[0, T]$ with small enough T (otherwise: call recursive solver)
- Time discretization: $0 = t_0 < t_1 < \cdots < t_{N_t} = T$, $t_{i+1} - t_i = \Delta t$
- Space discretization ($d = 1$): Grid Γ: $x_0 < x_1 < \cdots < x_{N_x}$, $x_{j+1} - x_j = \Delta x$
- Use projection Π to stay on Γ at every t_i: $\mathcal{L}(X_{t_i}^{(\mathtt{k}+1)}) \approx$ vector of weights
- Picard iterations for distribution & decoupling functions:
  - **Step 1:** Given $(\mu^{(\mathtt{k})}, u^{(\mathtt{k})}, v^{(\mathtt{k})})$, compute $\mu_{t_i}^{(\mathtt{k}+1)} = \mathcal{L}(X_{t_i}^{(\mathtt{k}+1)})$, $i = 0, \ldots, N_t$, where

$$X_{t_{i+1}}^{(\mathtt{k}+1)} = \Pi\left[X_{t_i}^{(\mathtt{k}+1)} + B\left(X_{t_i}^{(\mathtt{k}+1)}, \mu_{t_i}^{(\mathtt{k})}, u_{t_i}^{(\mathtt{k})}(X_{t_i}^{(\mathtt{k}+1)}), v_{t_i}^{(\mathtt{k})}(X_{t_i}^{(\mathtt{k}+1)})\right)dt + \sigma \Delta W_{t_{i+1}}\right]$$

Grid-Based Algorithm: Forward Equation

- Focus on an interval $[0, T]$ with small enough T (otherwise: call recursive solver)
- Time discretization: $0 = t_0 < t_1 < \cdots < t_{N_t} = T$, $t_{i+1} - t_i = \Delta t$
- Space discretization ($d = 1$): Grid Γ: $x_0 < x_1 < \cdots < x_{N_x}$, $x_{j+1} - x_j = \Delta x$
- Use projection Π to stay on Γ at every t_i: $\mathcal{L}(X_{t_i}^{(\mathtt{k}+1)}) \approx$ vector of weights
- Picard iterations for distribution & decoupling functions:
  - **Step 1:** Given $(\mu^{(\mathtt{k})}, u^{(\mathtt{k})}, v^{(\mathtt{k})})$, compute $\mu_{t_i}^{(\mathtt{k}+1)} = \mathcal{L}(X_{t_i}^{(\mathtt{k}+1)})$, $i = 0, \ldots, N_t$, where

$$X_{t_{i+1}}^{(\mathtt{k}+1)} = \Pi\left[X_{t_i}^{(\mathtt{k}+1)} + B\left(X_{t_i}^{(\mathtt{k}+1)}, \mu_{t_i}^{(\mathtt{k})}, u_{t_i}^{(\mathtt{k})}(X_{t_i}^{(\mathtt{k}+1)}), v_{t_i}^{(\mathtt{k})}(X_{t_i}^{(\mathtt{k}+1)})\right)dt + \sigma\Delta W_{t_{i+1}}\right]$$

  - In fact $\mu_{t_{i+1}}^{(\mathtt{k}+1)}$ can be expressed in terms of $\mu_{t_i}^{(\mathtt{k}+1)}$ and a transition kernel
  - Ex: binomial approx. of $W \to$ efficient computation using quantization

Grid-Based Algorithm: Backward Equation

- Picard iterations for distribution & decoupling functions (continued):
  - **Step 2:** Update u, v: for all $0 \leq i \leq N_t$, $x \in \Gamma$,

$$\begin{cases} u_{t_i}^{(k+1)}(x) = \mathbb{E}\Big[u_{t_{i+1}}^{(k+1)}(X_{t_i}^{(k+1)}) \\ \qquad\qquad + F\Big(X_{t_i}^{(k+1)}, \mu_{t_i}^{(k+1)}, u_{t_i}^{(k)}(X_{t_i}^{(k+1)}), v_{t_i}^{(k)}(X_{t_i}^{(k+1)})\Big)\Delta t \,\Big|\, X_{t_i}^{(k+1)} = x \Big] \\ u_T^{(k+1)}(x) = G(x, \mu_{t_i}^{(k+1)}) \\ v_{t_i}^{(k+1)}(x) = \mathbb{E}\Big[\frac{1}{\Delta t} u_{t_{i+1}}^{(k+1)}(X_{t_i}^{(k+1)}) \,\Big|\, X_{t_i}^{(k+1)} = x \Big] \\ v_T^{(k+1)}(x) = 0 \end{cases}$$

  - Ex.: binomial approximation of $W \to$ more explicit formulas

Grid-Based Algorithm: Backward Equation

- Picard iterations for distribution & decoupling functions (continued):
  - **Step 2:** Update u, v: for all $0 \leq i \leq N_t$, $x \in \Gamma$,

$$\begin{cases} u_{t_i}^{(k+1)}(x) = \mathbb{E}\Big[u_{t_{i+1}}^{(k+1)}(X_{t_i}^{(k+1)}) \\ \qquad\qquad +F\Big(X_{t_i}^{(k+1)}, \mu_{t_i}^{(k+1)}, u_{t_i}^{(k)}(X_{t_i}^{(k+1)}), v_{t_i}^{(k)}(X_{t_i}^{(k+1)})\Big)\Delta t \,\Big|\, X_{t_i}^{(k+1)} = x\Big] \\ u_T^{(k+1)}(x) = G(x, \mu_{t_i}^{(k+1)}) \\ v_{t_i}^{(k+1)}(x) = \mathbb{E}\Big[\frac{1}{\Delta t} u_{t_{i+1}}^{(k+1)}(X_{t_i}^{(k+1)}) \,\Big|\, X_{t_i}^{(k+1)} = x\Big] \\ v_T^{(k+1)}(x) = 0 \end{cases}$$

  - Ex.: binomial approximation of $W \to$ more explicit formulas
- Summary:
  - Forward: $(\mu^{(k)}, u^{(k)}, v^{(k)}) \mapsto \mu^{(k+1)} = \mathcal{L}(X^{(k+1)})$
  - Backward: $(\mu^{(k+1)}, u^{(k)}, v^{(k)}) \mapsto (u^{(k+1)}, v^{(k+1)})$

Grid-Based Algorithm: Backward Equation

- Picard iterations for distribution & decoupling functions (continued):
  - **Step 2:** Update u, v: for all $0 \leq i \leq N_t$, $x \in \Gamma$,

$$\begin{cases} u_{t_i}^{(\mathtt{k}+1)}(x) = \mathbb{E}\Big[u_{t_{i+1}}^{(\mathtt{k}+1)}(X_{t_i}^{(\mathtt{k}+1)}) \\ \qquad\qquad +F\big(X_{t_i}^{(\mathtt{k}+1)}, \mu_{t_i}^{(\mathtt{k}+1)}, u_{t_i}^{(\mathtt{k})}(X_{t_i}^{(\mathtt{k}+1)}), v_{t_i}^{(\mathtt{k})}(X_{t_i}^{(\mathtt{k}+1)})\big)\Delta t \,\Big|\, X_{t_i}^{(\mathtt{k}+1)} = x\Big] \\ u_T^{(\mathtt{k}+1)}(x) = G(x, \mu_{t_i}^{(\mathtt{k}+1)}) \\ v_{t_i}^{(\mathtt{k}+1)}(x) = \mathbb{E}\Big[\frac{1}{\Delta t} u_{t_{i+1}}^{(\mathtt{k}+1)}(X_{t_i}^{(\mathtt{k}+1)}) \,\Big|\, X_{t_i}^{(\mathtt{k}+1)} = x\Big] \\ v_T^{(\mathtt{k}+1)}(x) = 0 \end{cases}$$

  - Ex.: binomial approximation of $W \to$ more explicit formulas
- Summary:
  - Forward: $(\mu^{(\mathtt{k})}, u^{(\mathtt{k})}, v^{(\mathtt{k})}) \mapsto \mu^{(\mathtt{k}+1)} = \mathcal{L}(X^{(\mathtt{k}+1)})$
  - Backward: $(\mu^{(\mathtt{k}+1)}, u^{(\mathtt{k})}, v^{(\mathtt{k})}) \mapsto (u^{(\mathtt{k}+1)}, v^{(\mathtt{k}+1)})$

Details and numerical examples in [Chassagneux et al., 2019, Angiuli et al., 2019]

Dependence on the Moments

- In general: b, f, g involve the whole distribution $\mu_t = \mathcal{L}(X_t)$ (infinite dim.)
- What if they involve only the first moment $\overline{\mu}_t = \mathbb{E}[X_t]$?

Dependence on the Moments

- In general: b, f, g involve the whole distribution $\mu_t = \mathcal{L}(X_t)$ (infinite dim.)
- What if they involve only the first moment $\overline{\mu}_t = \mathbb{E}[X_t]$?
- Ex. 1: LQ (see lecture 2)
  - optimal control is a function of X_t and $\overline{\mu}_t = \mathbb{E}[X_t]$
  - ODE for $\overline{\mu}_t$ of the form $\frac{d}{dt}\overline{\mu}_t = \varphi(t, \overline{\mu}_t)$

Dependence on the Moments

- In general: b, f, g involve the whole distribution $\mu_t = \mathcal{L}(X_t)$ (infinite dim.)
- What if they involve only the first moment $\overline{\mu}_t = \mathbb{E}[X_t]$?
- Ex. 1: LQ (see lecture 2)
  - optimal control is a function of X_t and $\overline{\mu}_t = \mathbb{E}[X_t]$
  - ODE for $\overline{\mu}_t$ of the form $\frac{d}{dt}\overline{\mu}_t = \varphi(t, \overline{\mu}_t)$
- Ex. 2:

$$\begin{cases} b(x, \mu, \alpha) = b(x, \overline{\mu}, \alpha) = (\cos(x) + \cos(\overline{\mu}))\alpha \\ f(x, \mu, \alpha) = |\alpha|^2, \qquad g(x, \mu) = 0 \end{cases}$$

  - Can the optimal control be expressed as a function of $X_t, \mathbb{E}[X_t]$ only?
  - ODE for $\overline{\mu}_t$?

Dependence on the Moments

- In general: b, f, g involve the whole distribution $\mu_t = \mathcal{L}(X_t)$ (infinite dim.)
- What if they involve only the first moment $\overline{\mu}_t = \mathbb{E}[X_t]$?
- Ex. 1: LQ (see lecture 2)
  - optimal control is a function of X_t and $\overline{\mu}_t = \mathbb{E}[X_t]$
  - ODE for $\overline{\mu}_t$ of the form $\frac{d}{dt}\overline{\mu}_t = \varphi(t, \overline{\mu}_t)$
- Ex. 2:

$$\begin{cases} b(x, \mu, \alpha) = b(x, \overline{\mu}, \alpha) = (\cos(x) + \cos(\overline{\mu}))\alpha \\ f(x, \mu, \alpha) = |\alpha|^2, \qquad g(x, \mu) = 0 \end{cases}$$

  - Can the optimal control be expressed as a function of $X_t, \mathbb{E}[X_t]$ only?
  - ODE for $\overline{\mu}_t$?

$$\frac{d}{dt}\overline{\mu}_t = \mathbb{E}\left[(\cos(X_t) + \cos(\overline{\mu}_t))\alpha(t, X_t)\right]$$

It involves not only $\mathbb{E}[X_t] = \overline{\mu}_t$ but also $\mathbb{E}[\cos(X_t)]$

Dependence on the Moments

- In general: b, f, g involve the whole distribution $\mu_t = \mathcal{L}(X_t)$ (infinite dim.)

- What if they involve only the first moment $\overline{\mu}_t = \mathbb{E}[X_t]$?

- Ex. 1: LQ (see lecture 2)
  - optimal control is a function of X_t and $\overline{\mu}_t = \mathbb{E}[X_t]$
  - ODE for $\overline{\mu}_t$ of the form $\frac{d}{dt}\overline{\mu}_t = \varphi(t, \overline{\mu}_t)$

- Ex. 2:

$$\begin{cases} b(x, \mu, \alpha) = b(x, \overline{\mu}, \alpha) = (\cos(x) + \cos(\overline{\mu}))\alpha \\ f(x, \mu, \alpha) = |\alpha|^2, \qquad g(x, \mu) = 0 \end{cases}$$

  - Can the optimal control be expressed as a function of $X_t, \mathbb{E}[X_t]$ only?
  - ODE for $\overline{\mu}_t$?

$$\frac{d}{dt}\overline{\mu}_t = \mathbb{E}\left[(\cos(X_t) + \cos(\overline{\mu}_t))\alpha(t, X_t)\right]$$

  It involves not only $\mathbb{E}[X_t] = \overline{\mu}_t$ but also $\mathbb{E}[\cos(X_t)]$

- Class of MFC s.t. the problem can be solved with a finite number of moments?

Finite-Dimensional Reformulation

Following [Balata et al., 2019]

- In some cases, MFC problems can be written as:

$$J(\alpha) = \mathbb{E}\left[\int_0^T \mathcal{F}(\underline{X}_t, \alpha_t)dt + \mathcal{G}(\underline{X}_T)\right]$$

subject to:

$$d\underline{X}_t = \mathcal{B}(\underline{X}_t, \alpha_t)dt + \Sigma d\mathbb{W}_t$$

where the state is: $\underline{X}_t = (\mathbb{E}[X_t], \mathbb{E}[|X_t|^2], \ldots, \mathbb{E}[|X_t|^p]) \in (\mathbb{R}^d)^p$

Finite-Dimensional Reformulation

Following [Balata et al., 2019]

- In some cases, MFC problems can be written as:

$$J(\alpha) = \mathbb{E}\left[\int_0^T \mathcal{F}(\underline{X}_t, \alpha_t)dt + \mathcal{G}(\underline{X}_T)\right]$$

  subject to:

$$d\underline{X}_t = \mathcal{B}(\underline{X}_t, \alpha_t)dt + \Sigma d\mathbb{W}_t$$

  where the state is: $\underline{X}_t = (\mathbb{E}[X_t], \mathbb{E}[|X_t|^2], \ldots, \mathbb{E}[|X_t|^p]) \in (\mathbb{R}^d)^p$

- Time discretization: $0 = t_0 < t_1 < \cdots < t_{N_t} = T$, $t_{i+1} - t_i = \Delta t$

Finite-Dimensional Reformulation

Following [Balata et al., 2019]

- In some cases, MFC problems can be written as:

$$J(\alpha) = \mathbb{E}\left[\int_0^T \mathcal{F}(\underline{X}_t, \alpha_t)dt + \mathcal{G}(\underline{X}_T)\right]$$

  subject to:

$$d\underline{X}_t = \mathcal{B}(\underline{X}_t, \alpha_t)dt + \Sigma d\mathbb{W}_t$$

  where the state is: $\underline{X}_t = (\mathbb{E}[X_t], \mathbb{E}[|X_t|^2], \dots, \mathbb{E}[|X_t|^p]) \in (\mathbb{R}^d)^p$
- Time discretization: $0 = t_0 < t_1 < \dots < t_{N_t} = T$, $t_{i+1} - t_i = \Delta t$
- DPP for $V : [0,T] \times (\mathbb{R}^d)^p \to \mathbb{R}$ or rather $V_{\Delta t} : \{t_0, \dots, t_{N_t}\} \times (\mathbb{R}^d)^p \to \mathbb{R}$:

$$\begin{cases} V_{\Delta t}(T, \underline{x}) = \mathcal{G}(\underline{x}) \\ V_{\Delta t}(t_n, \underline{x}) = \sup_{\alpha} \left\{\mathcal{F}(\underline{x}, \alpha)\Delta t + \mathbb{E}^{t_n, \underline{x}, \alpha}\left[V_{\Delta t}(t_{n+1}, \underline{X}_{t_{n+1}})\right]\right\}, n = N_t - 1, \dots, 1, 0 \end{cases}$$

where $\mathbb{E}^{t_n, \underline{x}, \alpha}\left[V_{\Delta t}(t_{n+1}, \underline{X}_{t_{n+1}})\right] = \mathbb{E}\left[V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) \,|\, \underline{X}^{\alpha}_{t_n} = \underline{x}\right]$

Finite-Dimensional Reformulation

Following [Balata et al., 2019]

- In some cases, MFC problems can be written as:

$$J(\alpha) = \mathbb{E}\left[\int_0^T \mathcal{F}(\underline{X}_t, \alpha_t)dt + \mathcal{G}(\underline{X}_T)\right]$$

subject to:

$$d\underline{X}_t = \mathcal{B}(\underline{X}_t, \alpha_t)dt + \Sigma d\mathbb{W}_t$$

where the state is: $\underline{X}_t = (\mathbb{E}[X_t], \mathbb{E}[|X_t|^2], \ldots, \mathbb{E}[|X_t|^p]) \in (\mathbb{R}^d)^p$

- Time discretization: $0 = t_0 < t_1 < \cdots < t_{N_t} = T$, $t_{i+1} - t_i = \Delta t$
- DPP for $V : [0,T] \times (\mathbb{R}^d)^p \to \mathbb{R}$ or rather $V_{\Delta t} : \{t_0, \ldots, t_{N_t}\} \times (\mathbb{R}^d)^p \to \mathbb{R}$:

$$\begin{cases} V_{\Delta t}(T, \underline{x}) = \mathcal{G}(\underline{x}) \\ V_{\Delta t}(t_n, \underline{x}) = \sup_{\alpha} \left\{\mathcal{F}(\underline{x}, \alpha)\Delta t + \mathbb{E}^{t_n, \underline{x}, \alpha}\left[V_{\Delta t}(t_{n+1}, \underline{X}_{t_{n+1}})\right]\right\}, n = N_t - 1, \ldots, 1, 0 \end{cases}$$

where $\mathbb{E}^{t_n, \underline{x}, \alpha}\left[V_{\Delta t}(t_{n+1}, \underline{X}_{t_{n+1}})\right] = \mathbb{E}\left[V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) \,|\, \underline{X}^{\alpha}_{t_n} = \underline{x}\right]$

$\to$ **Key difficulty:** estimation of the conditional expectation

Estimation Method 1: Regression Monte Carlo

- Family of basis functions $\phi = (\phi^m)_{m=1,\ldots,M}$
- Projection:

$$\mathbb{E}\left[V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) \,|\, \underline{X}^{\alpha}_{t_n}\right] \approx \sum_{m=1}^{M} \beta^m_{t_n} \phi^m(\underline{X}^{\alpha}_{t_n})$$

where

$$\beta^m_{t_n} = \underset{\beta \in \mathbb{R}^M}{\operatorname{argmin}} \, \mathbb{E}\left[\left| V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) - \sum_{m=1}^{M} \beta^m \phi^m(\underline{X}^{\alpha}_{t_n}) \right|^2\right]$$

Estimation Method 1: Regression Monte Carlo

- Family of basis functions $\phi = (\phi^m)_{m=1,\ldots,M}$
- Projection:

$$\mathbb{E}\left[V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) \,|\, \underline{X}^{\alpha}_{t_n}\right] \approx \sum_{m=1}^{M} \beta^m_{t_n} \phi^m(\underline{X}^{\alpha}_{t_n})$$

where

$$\beta^m_{t_n} = \operatorname*{argmin}_{\beta \in \mathbb{R}^M} \mathbb{E}\left[\left|V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) - \sum_{m=1}^{M} \beta^m \phi^m(\underline{X}^{\alpha}_{t_n})\right|^2\right]$$

- Explicit expression:

$$\beta^m_{t_n} = \mathbb{E}[\phi(\underline{X}^{\alpha}_{t_n})\phi(\underline{X}^{\alpha}_{t_n})^\top]^{-1}\,\mathbb{E}[V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}})\phi(\underline{X}^{\alpha}_{t_n})]$$

Estimation Method 1: Regression Monte Carlo

- Family of basis functions $\phi = (\phi^m)_{m=1,\ldots,M}$
- Projection:

$$\mathbb{E}\left[V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) \,|\, \underline{X}^{\alpha}_{t_n}\right] \approx \sum_{m=1}^{M} \beta^m_{t_n} \phi^m(\underline{X}^{\alpha}_{t_n})$$

where

$$\beta^m_{t_n} = \underset{\beta \in \mathbb{R}^M}{\operatorname{argmin}} \, \mathbb{E}\left[\left|V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) - \sum_{m=1}^{M} \beta^m \phi^m(\underline{X}^{\alpha}_{t_n})\right|^2\right]$$

- Explicit expression:

$$\beta^m_{t_n} = \mathbb{E}[\phi(\underline{X}^{\alpha}_{t_n})\phi(\underline{X}^{\alpha}_{t_n})^\top]^{-1}\,\mathbb{E}[V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}})\phi(\underline{X}^{\alpha}_{t_n})]$$

- Estimation with N_{MC} Monte Carlo samples:

$$\mathbb{E}[\phi(\underline{X}^{\ell,\alpha}_{t_n})\phi(\underline{X}^{\ell,\alpha}_{t_n})^\top] \approx \frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} \phi(\underline{X}^{\ell,\alpha}_{t_n})\phi(\underline{X}^{\ell,\alpha}_{t_n})^\top$$

and

$$\mathbb{E}[V_{\Delta t}(t_{n+1}, \underline{X}^{\ell,\alpha}_{t_{n+1}})\phi(\underline{X}^{\ell,\alpha}_{t_n})] \approx \frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} V_{\Delta t}(t_{n+1}, \underline{X}^{\ell,\alpha}_{t_{n+1}})\phi(\underline{X}^{\ell,\alpha}_{t_n})$$

with training set $\left\{\left(\underline{X}^{\ell,\alpha}_{t_n}, \underline{X}^{\ell,\alpha}_{t_{n+1}}\right); \ell = 1, \ldots, N_{MC}\right\}$

Estimation Method 1: Regression Monte Carlo

- Family of basis functions $\phi = (\phi^m)_{m=1,\dots,M}$ *Not always easy to choose !*
- Projection:

$$\mathbb{E}\left[V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) \,|\, \underline{X}^{\alpha}_{t_n}\right] \approx \sum_{m=1}^{M} \beta^m_{t_n} \phi^m(\underline{X}^{\alpha}_{t_n})$$

where

$$\beta^m_{t_n} = \underset{\beta \in \mathbb{R}^M}{\operatorname{argmin}} \, \mathbb{E}\left[\left| V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) - \sum_{m=1}^{M} \beta^m \phi^m(\underline{X}^{\alpha}_{t_n}) \right|^2\right]$$

- Explicit expression:

$$\beta^m_{t_n} = \mathbb{E}[\phi(\underline{X}^{\alpha}_{t_n})\phi(\underline{X}^{\alpha}_{t_n})^\top]^{-1} \, \mathbb{E}[V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}})\phi(\underline{X}^{\alpha}_{t_n})]$$

- Estimation with N_{MC} Monte Carlo samples:

$$\mathbb{E}[\phi(\underline{X}^{\ell,\alpha}_{t_n})\phi(\underline{X}^{\ell,\alpha}_{t_n})^\top] \approx \frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} \phi(\underline{X}^{\ell,\alpha}_{t_n})\phi(\underline{X}^{\ell,\alpha}_{t_n})^\top$$

and

$$\mathbb{E}[V_{\Delta t}(t_{n+1}, \underline{X}^{\ell,\alpha}_{t_{n+1}})\phi(\underline{X}^{\ell,\alpha}_{t_n})] \approx \frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} V_{\Delta t}(t_{n+1}, \underline{X}^{\ell,\alpha}_{t_{n+1}})\phi(\underline{X}^{\ell,\alpha}_{t_n})$$

with training set $\left\{\left(\underline{X}^{\ell,\alpha}_{t_n}, \underline{X}^{\ell,\alpha}_{t_{n+1}}\right); \ell = 1, \dots, N_{MC}\right\}$

Estimation Method 2: Quantization

- Two space discretizations:
  - Set of points Γ on which we want to approximate $V_{\Delta t}$; projection Π_Γ
  - Quantization of noise (see e.g. [Pagès, 2018]):
    - ★ Set of cells $\mathcal{C}_Q = \{C_j; j = 1, \dots, J_Q\}$
    - ★ Associated grid points $\mathcal{G}_Q = \{\zeta_j; j = 1, \dots, J_Q\}$
    - ★ Weights for Gaussian r.v. $\Delta\mathbb{W} \sim \mathcal{N}(0, \Delta t)$: $p_j = \mathbb{P}(\Delta\mathbb{W} \in C_j)$
    - ★ Discrete version: $\Delta\hat{\mathbb{W}} \in \mathcal{G}_Q$: $\mathbb{P}(\Delta\hat{\mathbb{W}} = \zeta_j) = p_j$
    - ★ Can be optimized[1]; particularly helpful when $d > 1$

[1] Optimal grids/weights available here: `http://www.quantize.maths-fi.com`

Estimation Method 2: Quantization

- Two space discretizations:
  - Set of points Γ on which we want to approximate $V_{\Delta t}$; projection Π_Γ
  - Quantization of noise (see e.g. [Pagès, 2018]):
    - ★ Set of cells $\mathcal{C}_Q = \{C_j; j = 1, \dots, J_Q\}$
    - ★ Associated grid points $\mathcal{G}_Q = \{\zeta_j; j = 1, \dots, J_Q\}$
    - ★ Weights for Gaussian r.v. $\Delta\mathbb{W} \sim \mathcal{N}(0, \Delta t)$: $p_j = \mathbb{P}(\Delta\mathbb{W} \in C_j)$
    - ★ Discrete version: $\Delta\hat{\mathbb{W}} \in \mathcal{G}_Q$: $\mathbb{P}(\Delta\hat{\mathbb{W}} = \zeta_j) = p_j$
    - ★ Can be optimized[1]; particularly helpful when $d > 1$
- Estimation with piecewise constant interpolation: $\bar{V}_{\Delta t} : \{t_0, \dots, t_{N_t}\} \times \Gamma \to \mathbb{R}$

$$\mathbb{E}\left[V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) \,|\, \underline{X}^{\alpha}_{t_n} = \underline{x}\right] \approx \sum_{j=1}^{J_Q} p_j \bar{V}_{\Delta t}\Big(t_{n+1}, \Pi_\Gamma\Big(\mathcal{B}(\underline{x}, \alpha_{t_n})\Delta t + \Sigma\zeta_j\Big)\Big)$$

for all $\underline{x} \in \Gamma$

[1] Optimal grids/weights available here: `http://www.quantize.maths-fi.com`

Estimation Method 2: Quantization

- Two space discretizations:
  - Set of points Γ on which we want to approximate $V_{\Delta t}$; projection Π_Γ
  - Quantization of noise (see e.g. [Pagès, 2018]):
    - ★ Set of cells $\mathcal{C}_Q = \{C_j; j = 1, \dots, J_Q\}$
    - ★ Associated grid points $\mathcal{G}_Q = \{\zeta_j; j = 1, \dots, J_Q\}$
    - ★ Weights for Gaussian r.v. $\Delta\mathbb{W} \sim \mathcal{N}(0, \Delta t)$: $p_j = \mathbb{P}(\Delta\mathbb{W} \in C_j)$
    - ★ Discrete version: $\Delta\hat{\mathbb{W}} \in \mathcal{G}_Q$: $\mathbb{P}(\Delta\hat{\mathbb{W}} = \zeta_j) = p_j$
    - ★ Can be optimized[1]; particularly helpful when $d > 1$
- Estimation with piecewise constant interpolation: $\bar{V}_{\Delta t} : \{t_0, \dots, t_{N_t}\} \times \Gamma \to \mathbb{R}$

$$\mathbb{E}\left[V_{\Delta t}(t_{n+1}, \underline{X}^{\alpha}_{t_{n+1}}) \,|\, \underline{X}^{\alpha}_{t_n} = \underline{x}\right] \approx \sum_{j=1}^{J_Q} p_j \bar{V}_{\Delta t}\Big(t_{n+1}, \Pi_\Gamma\Big(\mathcal{B}(\underline{x}, \alpha_{t_n})\Delta t + \Sigma\zeta_j\Big)\Big)$$

for all $\underline{x} \in \Gamma$

- Other interpolations are possible

[1] Optimal grids/weights available here: `http://www.quantize.maths-fi.com`

Estimation Method 2: Quantization

- Two space discretizations:
  - Set of points Γ on which we want to approximate $V_{\Delta t}$; projection Π_Γ
  - Quantization of noise (see e.g. [Pagès, 2018]):
    - ★ Set of cells $\mathcal{C}_Q = \{C_j; j = 1, \ldots, J_Q\}$
    - ★ Associated grid points $\mathcal{G}_Q = \{\zeta_j; j = 1, \ldots, J_Q\}$
    - ★ Weights for Gaussian r.v. $\Delta\mathbb{W} \sim \mathcal{N}(0, \Delta t)$: $p_j = \mathbb{P}(\Delta\mathbb{W} \in C_j)$
    - ★ Discrete version: $\Delta\hat{\mathbb{W}} \in \mathcal{G}_Q$: $\mathbb{P}(\Delta\hat{\mathbb{W}} = \zeta_j) = p_j$
    - ★ Can be optimized[1]; particularly helpful when $d > 1$
- Estimation with piecewise constant interpolation: $\bar{V}_{\Delta t} : \{t_0, \ldots, t_{N_t}\} \times \Gamma \to \mathbb{R}$

$$\mathbb{E}\left[V_{\Delta t}(t_{n+1}, \underline{X}^\alpha_{t_{n+1}}) \,|\, \underline{X}^\alpha_{t_n} = \underline{x}\right] \approx \sum_{j=1}^{J_Q} p_j \bar{V}_{\Delta t}\Big(t_{n+1}, \Pi_\Gamma\Big(\mathcal{B}(\underline{x}, \alpha_{t_n})\Delta t + \Sigma\zeta_j\Big)\Big)$$

  for all $\underline{x} \in \Gamma$

- Other interpolations are possible

For more details and numerical examples, see [Balata et al., 2019]

[1] Optimal grids/weights available here: `http://www.quantize.maths-fi.com`

Outline

Summary

- Two schemes for FB PDEs of MFG
- Optimization methods for MFC and variational MFGs
- Two methods based on the probabilistic approach

Other numerical methods

The previous presentation is not exhaustive!

Some other references:

- Gradient descent based methods [Laurière and Pironneau, 2016], [Pfeiffer, 2016], [Lavigne and Pfeiffer, 2022]
- Monotone operators [Almulla et al., 2017], [Gomes and Saúde, 2018], [Gomes and Yang, 2020]
- Policy iteration [Cacace et al., 2021], [Cui and Koeppl, 2021], [Camilli and Tang, 2022], [Tang and Song, 2022], [Laurière et al., 2023]
- Finite elements [Benamou and Carlier, 2015b], [Andreev, 2017]
- Cubature [de Raynal and Trillos, 2015]
- Gaussian processes [Mou et al., 2022]
- Kernel-based representation [Liu et al., 2021]
- Fourier approximation [Nurbekyan et al., 2019]
- ...

Other numerical methods

The previous presentation is not exhaustive!

Some other references:

- Gradient descent based methods [Laurière and Pironneau, 2016], [Pfeiffer, 2016], [Lavigne and Pfeiffer, 2022]
- Monotone operators [Almulla et al., 2017], [Gomes and Saúde, 2018], [Gomes and Yang, 2020]
- Policy iteration [Cacace et al., 2021], [Cui and Koeppl, 2021], [Camilli and Tang, 2022], [Tang and Song, 2022], [Laurière et al., 2023]
- Finite elements [Benamou and Carlier, 2015b], [Andreev, 2017]
- Cubature [de Raynal and Trillos, 2015]
- Gaussian processes [Mou et al., 2022]
- Kernel-based representation [Liu et al., 2021]
- Fourier approximation [Nurbekyan et al., 2019]
- ...

However efficient, these methods are usually limited to problems with:

- (relatively) small dimension
- (relatively) simple structure

$\Rightarrow$ motivations to develop machine learning methods (see next lectures)

Thank you for your attention

Questions?

Feel free to reach out: mathieu.lauriere@nyu.edu

References I

[Achdou, 2013] Achdou, Y. (2013).
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.

[Achdou et al., 2012] Achdou, Y., Camilli, F., and Capuzzo-Dolcetta, I. (2012).
Mean field games: numerical methods for the planning problem.
SIAM J. Control Optim., 50(1):77–109.

[Achdou and Capuzzo-Dolcetta, 2010] Achdou, Y. and Capuzzo-Dolcetta, I. (2010).
Mean field games: numerical methods.
SIAM J. Numer. Anal., 48(3):1136–1162.

[Achdou and Lasry, 2019] Achdou, Y. and Lasry, J.-M. (2019).
Mean field games for modeling crowd motion.
In Chetverushkin, B. N., Fitzgibbon, W., Kuznetsov, Y. A., Neittaanmäki, P., Periaux, J., and Pironneau, O., editors, *Contributions to Partial Differential Equations and Applications*, chapter 4, pages 17–42. Springer International Publishing.

[Achdou and Laurière, 2015] Achdou, Y. and Laurière, M. (2015).
On the system of partial differential equations arising in mean field type control.
Discrete Contin. Dyn. Syst., 35(9):3879–3900.

References II

[Achdou and Laurière, 2016a] Achdou, Y. and Laurière, M. (2016a).
Mean Field Type Control with Congestion.
Appl. Math. Optim., 73(3):393–418.

[Achdou and Laurière, 2016b] Achdou, Y. and Laurière, M. (2016b).
Mean Field Type Control with Congestion (II): An augmented Lagrangian method.
Appl. Math. Optim., 74(3):535–578.

[Achdou and Laurière, 2020] Achdou, Y. and Laurière, M. (2020).
Mean field games and applications: Numerical aspects.
Mean Field Games: Cetraro, Italy 2019, 2281:249–307.

[Achdou et al., 2021] Achdou, Y., Lauriere, M., and Lions, P.-L. (2021).
Optimal control of conditioned processes with feedback controls.
Journal de Mathématiques Pures et Appliquées, 148:308–341.

[Achdou and Perez, 2012] Achdou, Y. and Perez, V. (2012).
Iterative strategies for solving linearized discrete mean field games systems.
Netw. Heterog. Media, 7(2):197–217.

[Achdou and Porretta, 2016] Achdou, Y. and Porretta, A. (2016).
Convergence of a finite difference scheme to weak solutions of the system of partial differential equations arising in mean field games.
SIAM J. Numer. Anal., 54(1):161–186.

References III

[Achdou and Porretta, 2018] Achdou, Y. and Porretta, A. (2018).
Mean field games with congestion.
Ann. Inst. H. Poincaré Anal. Non Linéaire, 35(2):443–480.

[Almulla et al., 2017] Almulla, N., Ferreira, R., and Gomes, D. (2017).
Two numerical approaches to stationary mean-field games.
Dyn. Games Appl., 7(4):657–682.

[Andreev, 2017] Andreev, R. (2017).
Preconditioning the augmented lagrangian method for instationary mean field games with diffusion.
SIAM Journal on Scientific Computing, 39(6):A2763–A2783.

[Angiuli et al., 2019] Angiuli, A., Graves, C. V., Li, H., Chassagneux, J.-F., Delarue, F., and Carmona, R. (2019).
Cemracs 2017: numerical probabilistic approach to MFG.
ESAIM: ProcS, 65:84–113.

[Aurell and Djehiche, 2018] Aurell, A. and Djehiche, B. (2018).
Mean-field type modeling of nonlocal crowd aversion in pedestrian crowd dynamics.
SIAM Journal on Control and Optimization, 56(1):434–455.

[Balata et al., 2019] Balata, A., Huré, C., Laurière, M., Pham, H., and Pimentel, I. (2019).
A class of finite-dimensional numerically solvable mckean-vlasov control problems.
ESAIM: Proceedings and Surveys, 65:114–144.

References IV

[Baudelet et al., 2023] Baudelet, S., Frénais, B., Laurière, M., Machtalay, A., and Zhu, Y. (2023).
Deep learning for mean field optimal transport.
arXiv preprint arXiv:2302.14739.

[Benamou and Carlier, 2015a] Benamou, J.-D. and Carlier, G. (2015a).
Augmented Lagrangian methods for transport optimization, mean field games and degenerate elliptic equations.
J. Optim. Theory Appl., 167(1):1–26.

[Benamou and Carlier, 2015b] Benamou, J.-D. and Carlier, G. (2015b).
Augmented lagrangian methods for transport optimization, mean field games and degenerate elliptic equations.
Journal of Optimization Theory and Applications, 167(1):1–26.

[Benamou et al., 2017] Benamou, J.-D., Carlier, G., and Santambrogio, F. (2017).
Variational mean field games.
In *Active Particles, Volume 1*, pages 141–171. Springer.

[Briceño Arias et al., 2019] Briceño Arias, L. M., Kalise, D., Kobeissi, Z., Laurière, M., Mateos González, A., and Silva, F. J. (2019).
On the implementation of a primal-dual algorithm for second order time-dependent mean field games with local couplings.
ESAIM: ProcS, 65:330–348.

References V

[Briceño Arias et al., 2018] Briceño Arias, L. M., Kalise, D., and Silva, F. J. (2018).
Proximal methods for stationary mean field games with local couplings.
SIAM J. Control Optim., 56(2):801–836.

[Cacace et al., 2021] Cacace, S., Camilli, F., and Goffi, A. (2021).
A policy iteration method for mean field games.
ESAIM: Control, Optimisation and Calculus of Variations, 27:85.

[Camilli and Tang, 2022] Camilli, F. and Tang, Q. (2022).
Rates of convergence for the policy iteration method for mean field games systems.
Journal of Mathematical Analysis and Applications, 512(1):126138.

[Cardaliaguet, 2015] Cardaliaguet, P. (2015).
Weak solutions for first order mean field games with local coupling.
In *Analysis and geometry in control theory and its applications*, pages 111–158. Springer.

[Cardaliaguet and Graber, 2015] Cardaliaguet, P. and Graber, P. J. (2015).
Mean field games systems of first order.
ESAIM Control Optim. Calc. Var., 21(3):690–722.

[Cardaliaguet et al., 2015] Cardaliaguet, P., Graber, P. J., Porretta, A., and Tonon, D. (2015).
Second order mean field games with degenerate diffusion and local coupling.
NoDEA Nonlinear Differential Equations Appl., 22(5):1287–1317.

References VI

[Carlini and Silva, 2014] Carlini, E. and Silva, F. J. (2014).
A fully discrete semi-Lagrangian scheme for a first order mean field game problem.
SIAM J. Numer. Anal., 52(1):45–67.

[Carlini and Silva, 2015] Carlini, E. and Silva, F. J. (2015).
A semi-Lagrangian scheme for a degenerate second order mean field game system.
Discrete Contin. Dyn. Syst., 35(9):4269–4292.

[Carmona and Delarue, 2018] Carmona, R. and Delarue, F. (2018).
Probabilistic theory of mean field games with applications. I, volume 83 of *Probability Theory and Stochastic Modelling*.
Springer, Cham.
Mean field FBSDEs, control, and games.

[Chambolle and Pock, 2011] Chambolle, A. and Pock, T. (2011).
A first-order primal-dual algorithm for convex problems with applications to imaging.
J. Math. Imaging Vision, 40(1):120–145.

[Chassagneux et al., 2019] Chassagneux, J.-F., Crisan, D., and Delarue, F. (2019).
Numerical method for FBSDEs of McKean-Vlasov type.
Ann. Appl. Probab., 29(3):1640–1684.

[Cui and Koeppl, 2021] Cui, K. and Koeppl, H. (2021).
Approximately solving mean field games via entropy-regularized deep reinforcement learning.
In *International Conference on Artificial Intelligence and Statistics*, pages 1909–1917. PMLR.

References VII

[de Raynal and Trillos, 2015] de Raynal, P. C. and Trillos, C. G. (2015).
A cubature based algorithm to solve decoupled mckean–vlasov forward–backward stochastic differential equations.
Stochastic Processes and their Applications, 125(6):2206–2255.

[Fortin and Glowinski, 1983] Fortin, M. and Glowinski, R. (1983).
Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems.
North-Holland.

[Gomes and Saúde, 2018] Gomes, D. A. and Saúde, J. (2018).
Numerical methods for finite-state mean-field games satisfying a monotonicity condition.
Applied Mathematics & Optimization.

[Gomes and Yang, 2020] Gomes, D. A. and Yang, X. (2020).
The hessian riemannian flow and newton's method for effective hamiltonians and mather measures.
ESAIM: Mathematical Modelling and Numerical Analysis, 54(6):1883–1915.

[Lachapelle and Wolfram, 2011] Lachapelle, A. and Wolfram, M.-T. (2011).
On a mean field game approach modeling congestion and aversion in pedestrian crowds.
Transportation research part B: methodological, 45(10):1572–1589.

[Lasry and Lions, 2007] Lasry, J.-M. and Lions, P.-L. (2007).
Mean field games.
Jpn. J. Math., 2(1):229–260.

[Laurière, 2021] Laurière, M. (2021).
Numerical methods for mean field games and mean field type control.
arXiv preprint arXiv:2106.06231.

[Laurière and Pironneau, 2016] Laurière, M. and Pironneau, O. (2016).
Dynamic programming for mean-field type control.
J. Optim. Theory Appl., 169(3):902–924.

[Laurière et al., 2023] Laurière, M., Song, J., and Tang, Q. (2023).
Policy iteration method for time-dependent mean field games systems with non-separable hamiltonians.
Applied Mathematics & Optimization, 87(2):17.

[Lavigne and Pfeiffer, 2022] Lavigne, P. and Pfeiffer, L. (2022).
Generalized conditional gradient and learning in potential mean field games.
arXiv preprint arXiv:2209.12772.

[Liu et al., 2021] Liu, S., Jacobs, M., Li, W., Nurbekyan, L., and Osher, S. J. (2021).
Computational methods for first-order nonlocal mean field games with applications.
SIAM Journal on Numerical Analysis, 59(5):2639–2668.

References IX

[Mou et al., 2022] Mou, C., Yang, X., and Zhou, C. (2022).
Numerical methods for mean field games based on gaussian processes and fourier features.
Journal of Computational Physics, 460:111188.

[Nurbekyan et al., 2019] Nurbekyan, L. et al. (2019).
Fourier approximation methods for first-order nonlocal mean-field games.
Portugaliae Mathematica, 75(3):367–396.

[Pagès, 2018] Pagès, G. (2018).
Numerical probability.
In *Universitext*. Springer.

[Pfeiffer, 2016] Pfeiffer, L. (2016).
Numerical methods for mean-field type optimal control problems.
Pure Appl. Funct. Anal., 1(4):629–655.

[Porretta and Zuazua, 2013] Porretta, A. and Zuazua, E. (2013).
Long time versus steady state optimal control.
SIAM J. Control Optim., 51(6):4242–4273.

[Tang and Song, 2022] Tang, Q. and Song, J. (2022).
Learning optimal policies in potential mean field games: Smoothed policy iteration algorithms.
arXiv preprint arXiv:2212.04791.