Mean Field Games: Analysis and Numerical Implementation

Joseph-Marie Ngueponwouo and Emmanuel Gardin

M2MO - Université Paris Cité

March 7, 2025

Outline

Introduction to Mean Field Games

The Hamilton-Jacobi-Bellman (HJB) Equation

Backward in time, describes the evolution of the value function:

$$\underbrace{-\partial_t u(t,x)}_{\text{time}} - \underbrace{\nu \Delta u(t,x)}_{\text{diffusion}} + \underbrace{H(x, m(t,x), \nabla u(t,x))}_{\text{optimal control}} = 0 \tag{1}$$

with terminal condition: $u(T,x) = g(x, m(T,\cdot))$

The Fokker-Planck-Kolmogorov (FPK) Equation

Forward in time, describes the evolution of the population density:

$$\underbrace{\partial_t m(t,x)}_{\text{time}} - \underbrace{\nu \Delta m(t,x)}_{\text{diffusion}} - \underbrace{\operatorname{div}(m(t,x)\nabla_p H(x, m(t,x), \nabla u))}_{\text{transport by optimal control}} = 0 \tag{2}$$

with initial condition: $m(0,x) = m_0(x)$

Challenge

- Coupling: agents optimize against m; m is determined by agents' optimal behaviors
- Nonlinearity in $H(x, \nabla u)$ prevents analytical solutions
- Forward-backward structure creates a two-point boundary value problem
- The coupling between u and m adds further complexity
- Numerical methods for PDEs are required to find approximate solutions

Congestion Model in Mean Field Games

Drift as Control and Running Cost

In our model:

- The drift $b(t, x, \gamma) = \gamma$ is directly controlled by the agent
- The running cost consists of: $f(x, m, \gamma) = L_0(\gamma, m(x)) + f_0(x, m(x))$
- $L_0(\gamma, m) = \frac{\beta-1}{\beta}(c_0 + c_1 m)^{\frac{\alpha}{\beta-1}} |\gamma|^{\frac{\beta}{\beta-1}}$ is the cost of control incorporating congestion effects
- $f_0(x, m)$ represents other costs related to state and distribution

Meaning of the Cost Components

- $L_0(\gamma, m(x))$ represents the energy or effort cost for control actions
- The factor $(c_0 + c_1 m)^{\frac{\alpha}{\beta-1}}$ increases the cost in congested areas
- When $\alpha > 0$, agents face higher costs when moving through crowded regions
- This naturally leads to congestion avoidance behavior

The Hamiltonian

The corresponding Hamiltonian is: $H(x, p, m) = H_0(p, m(x)) - f_0(x, m(x))$, where $H_0(p, \mu) = \frac{1}{\beta} \frac{|p|^\beta}{(c_0 + c_1 \mu)^\alpha}$

Mean Field Game PDEs with Congestion

Problem Setting Assumptions

- One-dimensional setting, i.e., $d = 1$
- State space is the domain $\Omega =]0, 1[$
- The stochastic process is reflected at the boundary $\partial\Omega$

Complete MFG System with Congestion

$$-\frac{\partial u}{\partial t}(t,x) - \nu \frac{\partial^2 u}{\partial x^2}(t,x) + \frac{1}{\beta} \frac{\left|\frac{\partial u}{\partial x}(t,\cdot)\right|^\beta}{(c_0 + c_1 m(t,x))^\alpha} = g(x) + \tilde{f}_0(m(t,x)), \tag{3}$$

$$\frac{\partial m}{\partial t}(t,x) - \nu \frac{\partial^2 m}{\partial x^2}(t,x) - \frac{\partial}{\partial x}\left(\frac{m(t,x)}{(c_0 + c_1 m(t,x))^\alpha} \left| \frac{\partial u}{\partial x} \right|^{\beta-2} \frac{\partial u}{\partial x} \right) = 0, \tag{4}$$

$$\frac{\partial u}{\partial x}(t,0) = \frac{\partial u}{\partial x}(t,1) = 0, \tag{5}$$

$$\frac{\partial m}{\partial x}(t,0) = \frac{\partial m}{\partial x}(t,1) = 0, \tag{6}$$

$$u(T,x) = \phi(x, m(T,x)), \quad m(0,x) = m_0(x). \tag{7}$$

All equations are defined in appropriate domains: $(0, T) \times \Omega$ or as boundary/initial conditions.

Discretization Grid

Grid Definition and Assumptions

- $N_T + 1$ points in time and N_h points in space
- Time step: $\Delta t = \frac{T}{N_T}$, Space step: $h = \frac{1}{N_h - 1}$
- Grid points: $t_n = n \times \Delta t$, $x_i = i \times h$ for $(n, i) \in \{0, \ldots, N_T\} \times \{0, \ldots, N_h - 1\}$
- Approximations: $u(t_n, x_i) \approx U_i^n$ and $m(t_n, x_i) \approx M_i^n$

Neumann Boundary Conditions

Ghost nodes are introduced at $x_{-1} = -h$ and $x_{N_h} = 1 + h$ with:

$$U_{-1}^n = U_0^n, \quad U_{N_h}^n = U_{N_h - 1}^n, \quad M_{-1}^n = M_0^n, \quad M_{N_h}^n = M_{N_h - 1}^n \tag{8}$$

These ensure proper implementation of the Neumann boundary conditions.

Finite Difference Operators

$$(D_t W)^n = \frac{1}{\Delta t}(W^{n+1} - W^n), \quad n \in \{0, \ldots, N_T - 1\} \tag{9}$$

$$(DW)_i = \frac{1}{h}(W_{i+1} - W_i), \quad i \in \{0, \ldots, N_h - 1\} \tag{10}$$

$$(\Delta_h W)_i = -\frac{1}{h^2}(2W_i - W_{i+1} - W_{i-1}), \quad i \in \{0, \ldots, N_h - 1\} \tag{11}$$

$$[\nabla_h W]_i = ((DW)_i, (DW)_{i-1}) \in \mathbb{R}^2, \quad i \in \{0, \ldots, N_h - 1\} \tag{12}$$

Discretization Operators

$$(DW)_i = \frac{1}{h} \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \tag{13}$$

$$(DW)_{i-1} = \frac{1}{h} \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & -1 & 1 \end{pmatrix} \tag{14}$$

$$(\Delta_h W)_i = \frac{1}{h^2} \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & -2 & 1 \\ 0 & 0 & \cdots & 0 & 1 & -1 \end{pmatrix} \tag{15}$$

Discrete Hamiltonian and HJB System

Discretized Hamiltonian

For the congestion model:

$$\tilde{H}([\nabla_h U^n]_i, M_i^{n+1}) = \frac{1}{\beta} \frac{|[\nabla_h U^n]_i|^\beta}{(c_0 + c_1 M_i^{n+1})^\alpha} \tag{16}$$

where $[\nabla_h U^n]_i = ((DU^n)_i, (DU^n)_{i-1})$ and we compute the norm using:

$$|[\nabla_h U^n]_i|^\beta = \left(((U_{i+1}^n - U_i^n)_-)^2 + ((U_i^n - U_{i-1}^n)_+)^2\right)^{\beta/2} \tag{17}$$

Discrete HJB System in Matrix Form

$$\mathbb{F}(U^n, U^{n+\mathbf{1}}, M^{n+\mathbf{1}}) := \begin{pmatrix} -(D_t U_{\mathbf{0}})^n - \nu(\Delta_h U^n)_{\mathbf{0}} + \tilde{H}([\nabla_h U^n]_{\mathbf{0}}, M_{\mathbf{0}}^{n+\mathbf{1}}) - g(x_{\mathbf{0}}) - \tilde{f}_{\mathbf{0}}(M_{\mathbf{0}}^{n+\mathbf{1}}) \\ \vdots \\ -(D_t U_i)^n - \nu(\Delta_h U^n)_i + \tilde{H}([\nabla_h U^n]_i, M_i^{n+\mathbf{1}}) - g(x_i) - \tilde{f}_{\mathbf{0}}(M_i^{n+\mathbf{1}}) \\ \vdots \\ -(D_t U_{N_h-\mathbf{1}})^n - \nu(\Delta_h U^n)_{N_h-\mathbf{1}} + \tilde{H}([\nabla_h U^n]_{N_h-\mathbf{1}}, M_{N_h-\mathbf{1}}^{n+\mathbf{1}}) - g(x_{N_h-\mathbf{1}}) - \tilde{f}_{\mathbf{0}}(M_{N_h-\mathbf{1}}^{n+\mathbf{1}}) \end{pmatrix} = \mathbf{0}$$

Jacobian Matrix of the Residual

$$\mathbb{J}(V, U^{n+1}, M^{n+1}) = \begin{pmatrix} \frac{\partial \mathbb{F}_0}{\partial V_0} & \frac{\partial \mathbb{F}_0}{\partial V_1} & \cdots & \frac{\partial \mathbb{F}_0}{\partial V_{N_h-1}} \\ \frac{\partial \mathbb{F}_1}{\partial V_0} & \frac{\partial \mathbb{F}_1}{\partial V_1} & \cdots & \frac{\partial \mathbb{F}_1}{\partial V_{N_h-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbb{F}_{N_h-1}}{\partial V_0} & \frac{\partial \mathbb{F}_{N_h-1}}{\partial V_1} & \cdots & \frac{\partial \mathbb{F}_{N_h-1}}{\partial V_{N_h-1}} \end{pmatrix}$$

This term only depends on $U^n_{i-1}, U^n_i, U^n_{i+1}$, so the Jacobian is a tridiagonal matrix:

$$\frac{\partial \mathbb{F}_i}{\partial U^n_{i-1}} = -\frac{\nu}{h^2} - \frac{1}{(c_0 + c_1 M_i^{n+1})^\alpha h^\beta}(U^n_i - U^n_{i-1})_+ \left(((U^n_{i+1} - U^n_i)_-)^2 + ((U^n_i - U^n_{i-1})_+)^2\right)^{\beta/2-1}$$

For interior points $1 \leq i \leq N_h - 2$:

$$\frac{\partial \mathbb{F}_i}{\partial U^n_i} = \frac{1}{\Delta t} + \frac{2\nu}{h^2} + \frac{1}{(c_0 + c_1 M_i^{n+1})^\alpha h^\beta} \left((U^n_{i+1} - U^n_i)_- - (U^n_i - U^n_{i-1})_+\right) \left(((U^n_{i+1} - U^n_i)_-)^2 + ((U^n_i - U^n_{i-1})_+)^2\right)^{\beta/2-1}$$

For boundary points $i = 0$ and $i = N_h - 1$ (due to Neumann conditions):

$$\frac{\partial \mathbb{F}_i}{\partial U^n_i} = \frac{1}{\Delta t} + \frac{\nu}{h^2} + \frac{1}{(c_0 + c_1 M_i^{n+1})^\alpha h^\beta} \left((U^n_{i+1} - U^n_i)_- - (U^n_i - U^n_{i-1})_+\right) \left(((U^n_{i+1} - U^n_i)_-)^2 + ((U^n_i - U^n_{i-1})_+)^2\right)^{\beta/2-1}$$

$$\frac{\partial \mathbb{F}_i}{\partial U^n_{i+1}} = -\frac{\nu}{h^2} - \frac{1}{(c_0 + c_1 M_i^{n+1})^\alpha h^\beta}(U^n_{i+1} - U^n_i)_- \left(((U^n_{i+1} - U^n_i)_-)^2 + ((U^n_i - U^n_{i-1})_+)^2\right)^{\beta/2-1}$$

Discrete KFP System (1/2)

Discretization of the Transport Term

For the term $\partial_x(m(t,x)|\partial_x u(t,x)|^{\beta-2}\partial_x u(t,x))$, after integration by parts and using Neumann boundary conditions: $-\int_\Omega \partial_x(H_p(\partial_x u, m)m)wdx = \int_\Omega mH_p(\partial_x u, m)\partial_x wdx$
The discretized right-hand side is:

$$h\sum_{i=0}^{N_h-1} M_i^{n+1}\left(\partial_{p_1}\tilde{H}([\nabla_h U^n]_i, M_i^{n+1})\frac{W_{i+1}^n - W_i^n}{h} + \partial_{p_2}\tilde{H}([\nabla_h U^n]_i, M_i^{n+1})\frac{W_i^n - W_{i-1}^n}{h}\right) \tag{18}$$

Transport operator

Define the transport operator:

$$\begin{aligned} T_i(U, M, \tilde{M}) = {} & \frac{1}{h}\left(M_i\partial_{p_1}\tilde{H}([\nabla_h U^n]_i, \tilde{M}_i) - M_{i-1}\partial_{p_1}\tilde{H}([\nabla_h U^n]_{i-1}, \tilde{M}_i)\right) && (19) \\ & + \frac{1}{h}\left(M_{i+1}\partial_{p_2}\tilde{H}([\nabla_h U^n]_{i+1}, \tilde{M}_{i+1}) - M_i\partial_{p_2}\tilde{H}([\nabla_h U^n]_i, \tilde{M}_i)\right) && (20) \end{aligned}$$

Discrete KFP System (2/2)

Discrete KFP System

The full discrete KFP system is:

$$\frac{M_i^{n+1} - M_i^n}{\Delta t} - \nu(\Delta_h M^{n+1})_i - T_i(U^n, M^{n+1}, M^{n+1}) = 0, \quad 0 \leq i < N_h, \, 0 < n < N_T \tag{21}$$

$$M_{-1}^n = M_0^n, \quad 0 < n \leq N_T \tag{22}$$

$$M_{N_h}^n = M_{N_h-1}^n, \quad 0 < n \leq N_T \tag{23}$$

$$M_i^0 = \tilde{m}_0(x_i), \quad 0 \leq i < N_h \tag{24}$$

Matrix Formulation and Solution

Since $M \mapsto T(U^n, M, \tilde{M})$ is linear with matrix $A = (-J_H)^T$, the system becomes:

$$\frac{M^{n+1} - M^n}{\Delta t} - \nu D_x^2 M^{n+1} + (J_H)^T M^{n+1} = 0 \tag{25}$$

$$\tag{26}$$

To solve for M^{n+1}:

$$(I_{N_h} - \nu \Delta t D_x^2 + \Delta t (J_H)^T) M^{n+1} = M^n \tag{27}$$

Forward-Backward Algorithm

Fixed Point Algorithm for Solving the MFG System

Let $\theta \in (0,1)$ be a relaxation parameter (e.g., $\theta = 0.01$). Given $(M^{(k)}, U^{(k)})$:

1. Solve discrete HJB equation using $(M^{(k)}, U^{(k)})$ to get $\hat{U}^{(k+1)}$
2. Solve discrete KFP equation using $(M^{(k)}, \hat{U}^{(k+1)})$ to get $\hat{M}^{(k+1)}$
3. Update with relaxation: $(M^{(k+1)}, U^{(k+1)}) = (1-\theta)(M^{(k)}, U^{(k)}) + \theta(\hat{M}^{(k+1)}, \hat{U}^{(k+1)})$
4. Stop when $\|(M^{(k+1)}, U^{(k+1)}) - (M^{(k)}, U^{(k)})\| < \varepsilon$ (e.g., $\varepsilon = 2.10^{-5}$)

Initialize with $M_i^{n,(0)} = \tilde{m}_0(x_i)$ for all i, n and $U_i^{n,(0)} = 0$ for all i, n.

Efficient Implementation (1/2)

Sparse Matrix Representation

Represent differential operators as sparse banded matrices to reduce complexity from $O(N_h^2)$ to $O(N_h)$

```
# First derivative operator (Dx) as sparse matrix
main_diag_Dx = np.full(N_h, -1)
main_diag_Dx[-1] = 0
super_diag_Dx = np.full(N_h - 1, 1)
Dx = sp.diags([main_diag_Dx, super_diag_Dx], [0, 1], shape=(N_h, N_h)) / h
```

Vectorized Forward-Backward Algorithm

- Naive approach: Loop over each spatial point $i \in [0, N_h - 1]$ to update U_i^n and M_i^n
- Our approach: Vectorized operations to update entire arrays at once
- For KFP: Construct banded matrix systems and solve efficiently with scipy
- For HJB: Use Newton-Raphson method with vectorized gradient computation

Example: Vectorized computation of Hamiltonian Jacobian

```
def compute_JH(U, M):
  # Compute upwind derivatives
  backward_spatial_diff = np.maximum(Dx_shift @ U, 0)
  forward_spatial_diff = -np.minimum(Dx @ U, 0)

  # Coefficient for nonlinear term
  coef = backward_spatial_diff**2 + forward_spatial_diff**2
  coef = (1 / (c0 + c1 * M) ** alpha) * np.power(coef, beta/2-1,
       out=np.zeros_like(coef), where=(coef!=0))

  # Construct diagonal elements
  main_diag = (1/h) * (forward_spatial_diff + backward_spatial_diff) * coef
  sub_diag = (-1/h) * backward_spatial_diff * coef
  sup_diag = (-1/h) * forward_spatial_diff * coef

  # Assemble banded matrix
  banded_matrix = np.zeros((3, U.size))
  banded_matrix[0, 1:] = sup_diag[:-1]
  banded_matrix[1] = main_diag
  banded_matrix[2, :-1] = sub_diag[1:]
  return banded_matrix
```

Rescaling HJB for Numerical Precision

- HJB equation can be rescaled by Δt for better numerical stability
- Original: $\frac{u_i^{n-1} - u_i^n}{\Delta t} + \mathcal{H} + ... = F$
- Rescaled: $u_i^{n-1} - u_i^n + \Delta t \cdot \mathcal{H} + ... = \Delta t \cdot F$
- Reduces roundoff errors when Δt is small

Conservation Check

- Monitor $\sum_i m_i^n$ at each time step
- Deviations indicate numerical errors

Simulation Parameters

Domain and Functions

- Domain: $\Omega =]0,1[$, Time horizon: $T = 1$
- Terminal cost: $g(x) = 0$
- Running cost: $\tilde{f}(m(x)) = m(x)$
- Terminal cost: $\phi(x,m) = -\exp(-40(x-0.7)^2)$
- Initial distribution: $m_0(x) = \sqrt{\frac{300}{\pi}}\exp(-300(x-0.2)^2)$

Parameter Sets

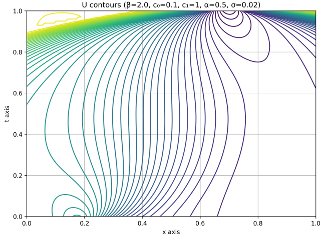
Case	β	c_0	c_1	α	σ	θ
(a)	2	0.1	1	0.5	0.02	0.05
(b)	2	0.1	5	1	0.02	0.05
(c)	2	0.01	2	1.2	0.1	0.05
(d)	2	0.01	2	1.5	0.2	0.05
(e)	2	1	3	2	0.002	0.005

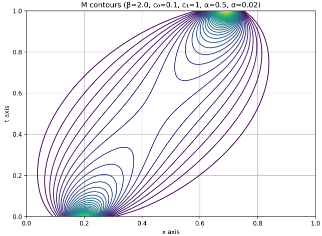
Discretization and Algorithm Settings

- Grid: $N_h = 201$ spatial points, $N_T = 100$ time steps
- Stopping criteria: Newton method 10^{-12}, Fixed point method 2×10^{-5}

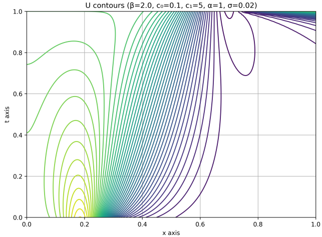
Experimental Results: Parameter Sets (a) and (b)

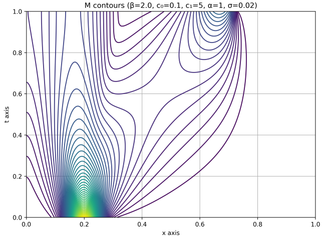
Parameter set (a): $\alpha = 0.5,\ c_1 = 1,\ c_0 = 0.1,\ \sigma = 0.02$





Parameter set (b): $\alpha = 1,\ c_1 = 5,\ c_0 = 0.1,\ \sigma = 0.02$



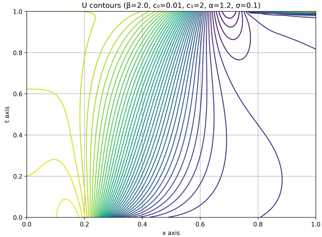


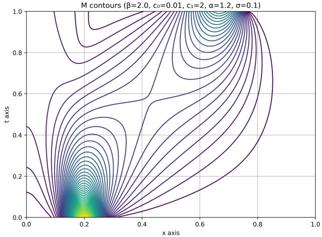
Observations

- Parameter set (a): Moderate congestion effects with smooth density evolution
- Parameter set (b): Stronger congestion avoidance with increased spreading
- Higher congestion sensitivity ($c_1 = 5$) leads to more uniform density distribution
- Agents clearly deviate from direct routes to avoid high-density regions

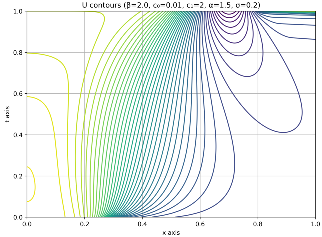
Experimental Results: Parameter Sets (c) and (d)

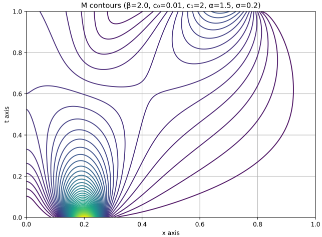
Parameter set (c): $\alpha = 1.2,\ c_1 = 2,\ c_0 = 0.01,\ \sigma = 0.1$





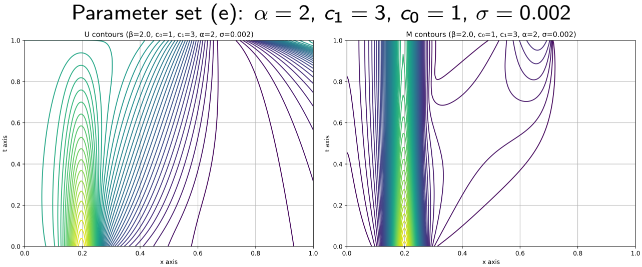
Parameter set (d): $\alpha = 1.5,\ c_1 = 2,\ c_0 = 0.01,\ \sigma = 0.2$





Observations

- Parameter set (c) : Increased diffusion and congestion sensitivity
- Parameter set (d) : Higher noise (σ) results in more spread-out distributions
- Stronger congestion effects (α) cause earlier splitting of density profiles
- Multiple pathways emerge when agents strongly avoid congestion



Observations

- Parameter set (e): Extreme congestion with minimal diffusion causes "gridlock"
- Very low noise ($\sigma = 0.002$) prevents density diffusion, creating rigid barriers
- High congestion penalty ($\alpha = 2$) makes moving through dense regions prohibitively costly
- Vertical "tunnels" form as agents become trapped, unable to reach terminal condition
- This represents a Nash equilibrium: no single agent can improve by deviating

Visualization of Agent Movement

Video Simulation Results

- Time evolution shows agents moving from initial position ($x = 0.2$) to target ($x = 0.7$)
- With congestion: agents spread out more to avoid high-density regions
- Without congestion: more concentrated movement toward target

Video: MFG Simulation with Parameter Set (a)

Play Video

(Click to open in external player)

Mean Field Control (MFC): Social Optimization (1/2)

Mean Field Control Problem

- MFG: Each agent optimizes their own individual cost
- MFC: A central planner optimizes social welfare (aggregate cost)
- Social cost: $J_{soc}(m, \gamma) = \mathbb{E}\left[\int_0^T \left(L_0(X_t, \gamma_t, m_t) + \tilde{f}_0(X_t, m_t)\right) dt + \phi(X_T, m_T)\right]$
- MFC internalizes congestion externalities that individual agents ignore

Complete MFC System

$$-\frac{\partial u}{\partial t}(t,x) - \nu \frac{\partial^2 u}{\partial x^2}(t,x) + \frac{1}{\beta} \frac{|\frac{\partial u}{\partial x}(t,x)|^\beta}{(c_0 + c_1 m(t,x))^\alpha} - \frac{c_1 \alpha}{\beta} \frac{m(t,x)|\frac{\partial u}{\partial x}(t,x)|^\beta}{(c_0 + c_1 m(t,x))^{\alpha+1}} = g(x) + \tilde{f}_0(m(t,x)) + m(t,x)\tilde{f}_0'(m(t,x)) \tag{28}$$

$$\frac{\partial m}{\partial t}(t,x) - \nu \frac{\partial^2 m}{\partial x^2}(t,x) - \frac{\partial}{\partial x}\left(\frac{m(t,\cdot)}{(c_0 + c_1 m(t,\cdot))^\alpha}\left|\frac{\partial u}{\partial x}(t,\cdot)\right|^{\beta-2} \frac{\partial u}{\partial x}(t,\cdot)\right)(x) = 0 \tag{29}$$

$$\frac{\partial u}{\partial x}(t,0) = \frac{\partial u}{\partial x}(t,1) = 0, \quad \frac{\partial m}{\partial x}(t,0) = \frac{\partial m}{\partial x}(t,1) = 0 \tag{30}$$

$$u(T,x) = \phi(x, m(T,x)), \quad m(0,x) = m_0(x) \tag{31}$$

Mean Field Control (MFC): Social Optimization (2/2)

Modified Hamiltonian for MFC

The only difference between MFC and MFG lies in the Hamiltonian term in the HJB equation. For MFC:

$$H_0(t,x) = \frac{1}{\beta}\frac{|\frac{\partial u}{\partial x}(t,x)|^{\beta}}{(c_0 + c_1 m(t,x))^{\alpha}}\left(1 - \frac{c_1 \alpha m(t,x)}{c_0 + c_1 m(t,x)}\right) \tag{32}$$

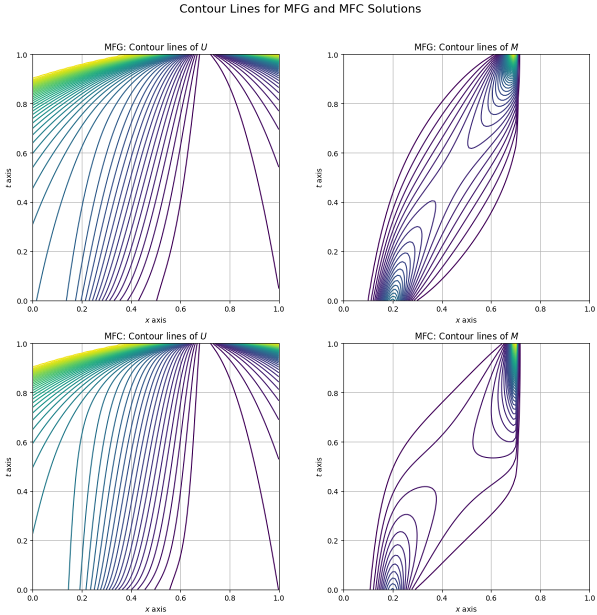
The discrete Hamiltonian is given by:

$$\tilde{H}(p_1, p_2, \mu) = \frac{1}{\beta}\frac{((p_1)_-^2 + (p_2)_+^2)^{\frac{\beta}{2}}}{(c_0 + c_1 \mu)^{\alpha}}\left(1 - \frac{c_1 \alpha \mu}{c_0 + c_1 \mu}\right) \tag{33}$$

Implementation Changes for MFC

- Replace the Hamiltonian in the HJB equation
- Set f to 0
- Keep the same numerical scheme structure but set Nh = 101 and Nt = 100 for stability purposes

MFC Implementation and Results



Key Observations

- MFC solutions tend to be more spread out than MFG solutions
- MFC accounts for the externality impact of congestion
- MFC agents converge faster towards the terminal condition
- Lower peak densities in MFC (reduces social congestion cost)
- Smoother transitions between high and low density regions

Social Cost Comparison

Social Cost Implementation

The social cost calculation involves:

- Computing optimal control γ using upwind derivatives
- Calculating Lagrangian $L_0(\gamma, m) = \frac{\beta-1}{\beta}(c_0 + c_1 m)^{\frac{\alpha}{\beta-1}} |\gamma|^{\frac{\beta}{\beta-1}}$
- Running cost: $\int_0^T \int_\Omega L_0(\gamma, m) \cdot m \, dx \, dt$ computed with time and space integration
- Terminal cost: $\int_\Omega \phi(x) \cdot m(T, x) \, dx$ computed with spatial integration

	Running Cost	Terminal Cost	Total Cost
MFG	0.24427	-0.96882	-0.72455
MFC	0.24479	-0.98913	-0.74434

Table 1: Cost comparison between MFG and MFC solutions (less is better)

Results

- MFC achieves 2.73% experimental improvement in total social cost (a thinner grid would be needed to confirm this result)
- Terminal cost optimization drives the overall improvement
- Centralized coordination slightly increases the running cost in order to achieve a better total cost

Thank you for your attention!

Questions?