Mean Field Games: Numerical Methods for finite horizon problems

Y. Achdou

May 13, 2011

Introduction

Numerical methods for the MFG system in the finite horizon setting

- Numerical schemes
- Methods for solving the finite-dimensional system of nonlinear equations which arises in the discrete MFG
  1. nonlinear strategies: here, Newton methods
  2. strategies for solving the linearized MFG systems

Outline of the present talk

- A brief review of the schemes (joint work with F. Camilli and I. Capuzzo-Dolcetta)
- Focus on the strategies for solving the linearized MFG systems : A good understanding of the continuous MFG system will be helpful.
- No proofs.

I Finite difference schemes

Goal: use a (semi-)implicit finite difference scheme, robust when $\nu \to 0$, which guarantees existence, and possibly uniform bounds and uniqueness.

Take $d = 2$:

$$\begin{cases} \dfrac{\partial u}{\partial t} - \nu \Delta u + H(x, \nabla u) = \Phi[m], & \text{in } (0,T) \times \mathbb{T}, \\ \dfrac{\partial m}{\partial t} + \nu \Delta m + \operatorname{div}\left(m \dfrac{\partial H}{\partial p}(x, \nabla u) \right) = 0, & \text{in } (0,T) \times \mathbb{T}, \\ \displaystyle\int_{\mathbb{T}} m dx = 1, \qquad m > 0 \quad \text{in } \mathbb{T}, \\ u(t=0) = \Phi_0[m(t=0)], \quad m(t=T) = m_\circ, \end{cases} \qquad (**)$$

- Let $\mathbb{T}_h$ be a uniform grid on the torus with mesh step h, and x_{ij} be a generic point in $\mathbb{T}_h$.
- Uniform time grid: $\Delta t = T/N_T$, $t_n = n\Delta t$.
- The values of u and m at $(x_{i,j}, t_n)$ are resp. approximated by $U^n_{i,j}$ and $M^n_{i,j}$.

Notation:

- The discrete Laplace operator:

$$(\Delta_h W)_{i,j} = -\frac{1}{h^2}(4W_{i,j} - W_{i+1,j} - W_{i-1,j} - W_{i,j+1} - W_{i,j-1}).$$

- Right-sided finite difference formulas for $\frac{\partial w}{\partial x_1}(x_{i,j})$ and $\frac{\partial w}{\partial x_2}(x_{i,j})$:

$$(D_1^+ W)_{i,j} = \frac{W_{i+1,j} - W_{i,j}}{h}, \quad \text{and} \quad (D_2^+ W)_{i,j} = \frac{W_{i,j+1} - W_{i,j}}{h}.$$

- The set of 4 finite difference formulas at $x_{i,j}$:

$$[D_h W]_{i,j} = \Big((D_1^+ W)_{i,j}, (D_1^+ W)_{i-1,j}, (D_2^+ W)_{i,j}, (D_2^+ W)_{i,j-1}\Big).$$

Discrete HJB equation

$$\frac{\partial u}{\partial t} - \nu \Delta u + H(x, \nabla u) = \Phi[m]$$

$$\downarrow$$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n}}{\Delta t} - \nu (\Delta_h U^{n+1})_{i,j} + g(x_{i,j}, [D_h U^{n+1}]_{i,j}) = (\Phi_h[M^n])_{i,j}$$

- $$\begin{aligned} & g(x_{i,j}, [D_h U^{n+1}]_{i,j}) \\ =& g\Big(x_{i,j}, (D_1^+ U^{n+1})_{i,j}, (D_1^+ U^{n+1})_{i-1,j}, (D_2^+ U^{n+1})_{i,j}, (D_2^+ U^{n+1})_{i,j-1}\Big), \end{aligned}$$

- for instance,

$$(\Phi_h[M])_{i,j} = \Phi[m_h](x_{i,j}),$$

calling m_h the piecewise constant function on $\mathbb{T}$ taking the value $M_{i,j}$ in the square $|x - x_{i,j}|_\infty \leq h/2$.

Classical assumptions on the discrete Hamiltonian g

$$(q_1, q_2, q_3, q_4) \to g\left(x, q_1, q_2, q_3, q_4\right).$$

- **Monotonicity:** g is nonincreasing with respect to q_1 and q_3 and nondecreasing with respect to to q_2 and q_4.
- **Consistency:**

$$g\left(x, q_1, q_1, q_3, q_3\right) = H(x, q), \quad \forall x \in \mathbb{T}, \forall q = (q_1, q_3) \in \mathbb{R}^2.$$

- **Differentiability:** g is of class $\mathcal{C}^1$, and

$$\left| \frac{\partial g}{\partial x} \Big(x, (q_1, q_2, q_3, q_4) \Big) \right| \le C(1 + |q_1| + |q_2| + |q_3| + |q_4|).$$

- **Convexity:** $(q_1, q_2, q_3, q_4) \to g\left(x, q_1, q_2, q_3, q_4\right)$ is convex.

The discrete version of

$$\frac{\partial m}{\partial t} + \nu \Delta m + \mathrm{div}\left(m \frac{\partial H}{\partial p}(x, \nabla v) \right) = 0. \qquad (\dagger)$$

It is chosen so that

- each time step leads to a linear system with a matrix
  - whose diagonal coefficients are negative,
  - whose off-diagonal coefficients are nonnegative,

  in order to hopefully use some **discrete maximum principle.**

- The argument for uniqueness should hold in the discrete case, so **the discrete Hamiltonian g should be used for $(\dagger)$ as well.**

Principle

Discretize

$$-\int_{\mathbb{T}} \operatorname{div}\left(m\frac{\partial H}{\partial p}(x,\nabla u)\right) w = \int_{\mathbb{T}} m\frac{\partial H}{\partial p}(x,\nabla u)\cdot\nabla w$$

by

$$-h^2\sum_{i,j}\mathcal{B}_{i,j}(U,M)W_{i,j} := h^2\sum_{i,j} M_{i,j}\nabla_q g(x_{i,j},[D_hU]_{i,j})\cdot[D_hW]_{i,j},$$

which leads to

$$\mathcal{B}_{i,j}(U,M) = \frac{1}{h}\left(\begin{array}{c}\left(\begin{array}{c} M_{i,j}\frac{\partial g}{\partial q_1}(x_{i,j},[D_hU]_{i,j}) - M_{i-1,j}\frac{\partial g}{\partial q_1}(x_{i-1,j},[D_hU]_{i-1,j}) \\ +M_{i+1,j}\frac{\partial g}{\partial q_2}(x_{i+1,j},[D_hU]_{i+1,j}) - M_{i,j}\frac{\partial g}{\partial q_2}(x_{i,j},[D_hU]_{i,j})\end{array}\right) \\ +\left(\begin{array}{c} M_{i,j}\frac{\partial g}{\partial q_3}(x_{i,j},[D_hU]_{i,j}) - M_{i,j-1}\frac{\partial g}{\partial q_3}(x_{i,j-1},[D_hU]_{i,j-1}) \\ +M_{i,j+1}\frac{\partial g}{\partial q_4}(x_{i,j+1},[D_hU]_{i,j+1}) - M_{i,j}\frac{\partial g}{\partial q_4}(x_{i,j},[D_hU]_{i,j})\end{array}\right)\end{array}\right)$$

This yields the semi-implicit scheme:

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n}}{\Delta t} - \nu(\Delta_h U^{n+1})_{i,j} + g(x_{i,j}, [D_h U^{n+1}]_{i,j}) = (V_h[M^n])_{i,j}$$

$$0 = \frac{M_{i,j}^{n+1} - M_{i,j}^{n}}{\Delta t} + \nu(\Delta_h M^n)_{i,j}$$

$$+\frac{1}{h}\left(\begin{array}{c} \left(\begin{array}{l} M_{i,j}^{n} \dfrac{\partial g}{\partial q_1}(x_{i,j}, [D_h U^{n+1}]_{i,j}) - M_{i-1,j}^{n} \dfrac{\partial g}{\partial q_1}(x_{i-1,j}, [D_h U^{n+1}]_{i-1,j}) \\ + M_{i+1,j}^{n} \dfrac{\partial g}{\partial q_2}(x_{i+1,j}, [D_h U^{n+1}]_{i+1,j}) - M_{i,j}^{n} \dfrac{\partial g}{\partial q_2}(x_{i,j}, [D_h U^{n+1}]_{i,j}) \end{array} \right) \\ \\ + \left(\begin{array}{l} M_{i,j}^{n} \dfrac{\partial g}{\partial q_3}(x_{i,j}, [D_h U^{n+1}]_{i,j}) - M_{i,j-1}^{n} \dfrac{\partial g}{\partial q_3}(x_{i,j-1}, [D_h U^{n+1}]_{i,j-1}) \\ + M_{i,j+1}^{n} \dfrac{\partial g}{\partial q_4}(x_{i,j+1}, [D_h U^{n+1}]_{i,j+1}) - M_{i,j}^{n} \dfrac{\partial g}{\partial q_4}(x_{i,j}, [D_h U^{n+1}]_{i,j}) \end{array} \right) \end{array} \right)$$

- **The linear operator in the discrete Fokker-Planck equation is the adjoint of the linearized discrete HJB operator.**

- **The discrete system has the same structure as the continuous MFG system**

The discrete MFG system: known facts

- **Existence for finite and infinite horizon** under rather general assumptions: does not need monotonicity of Φ and Φ_0 (Y.A. - I. Capuzzo Dolcetta)

- **Uniqueness** if Φ and Φ_0 are strictly monotone operators

- **Under suitable assumptions, uniform Lipschitz bounds on u_h w.r.t. h and Δt**

- **Optimization** If Φ and Φ_0 are local operators and furthermore increasing functions, the discrete MFG system can be seen as the optimality conditions of a saddle point problem.

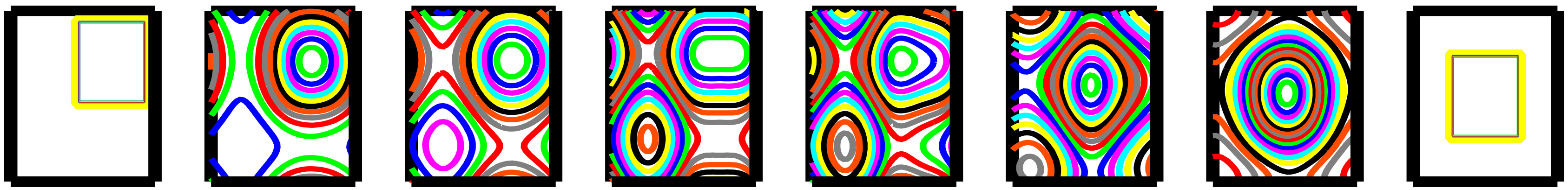
- **Discrete planning problems** (Y.A. - F. Camilli - I. Capuzzo Dolcetta)

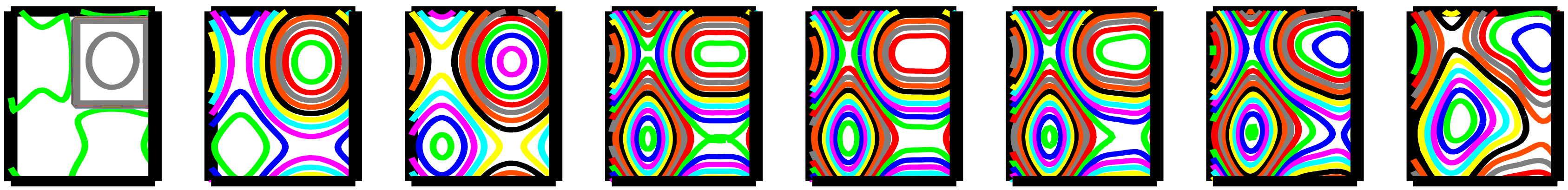
Other numerical works

- Lachapelle-Salomon-Turinici, Lachapelle-Wolfram (congestion)
- Guéant (2009) (2011)

Example of results for the planning problem

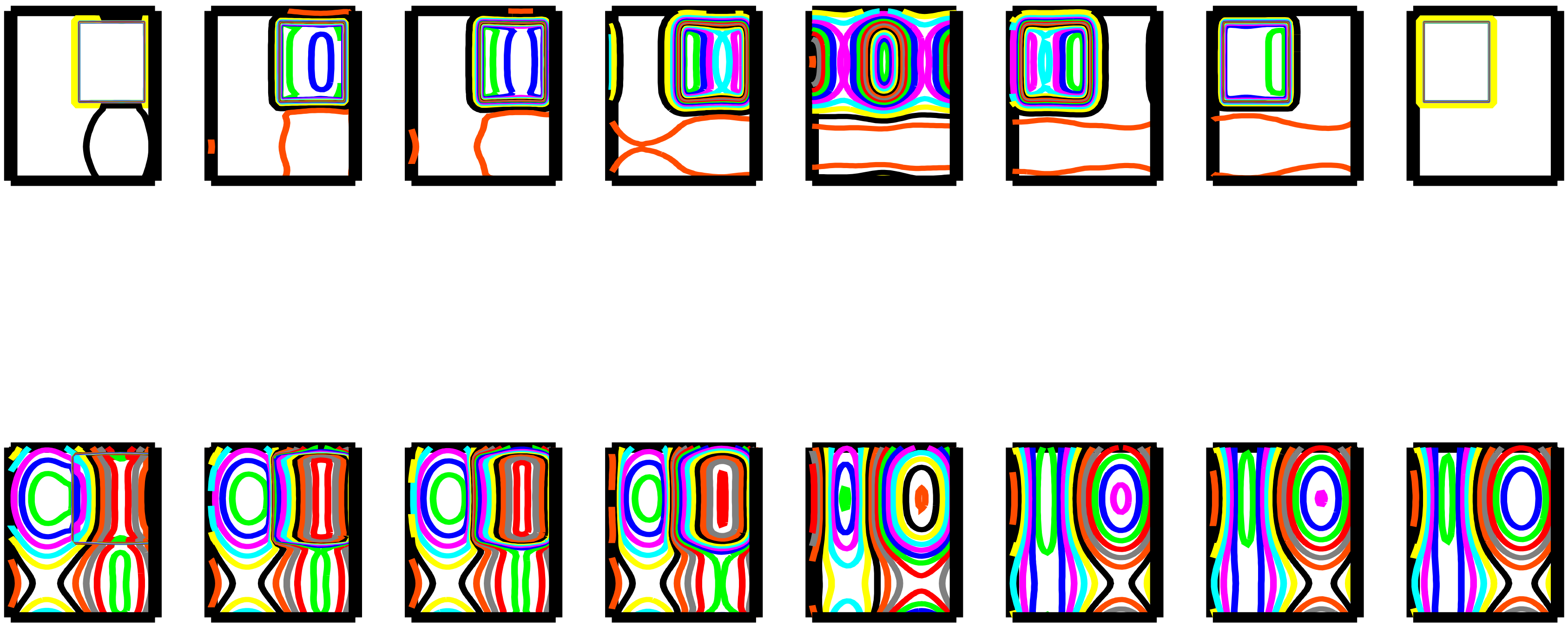
$T = 1, \nu = 1, \Phi(m) = m^2, H(p) = \sin(2\pi x_2) + \sin(2\pi x_1) + \cos(4\pi x_1) + |p|^2$





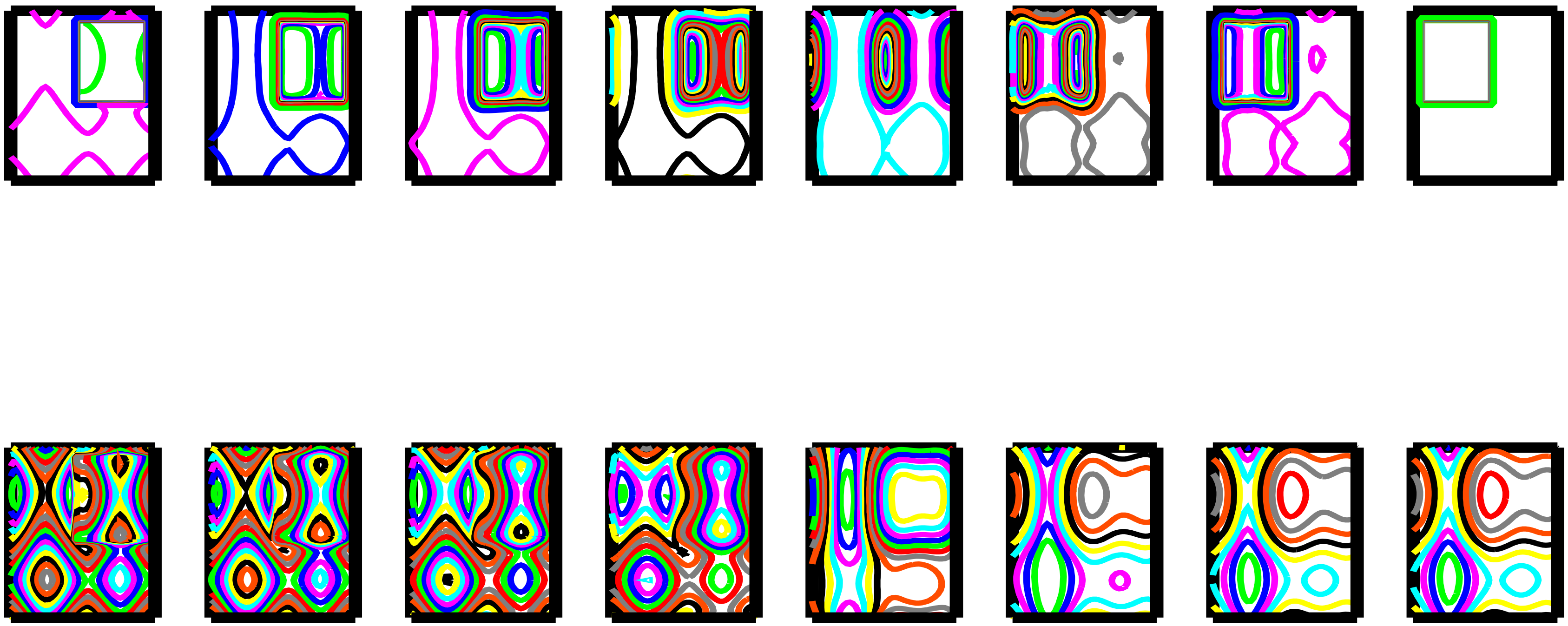
Snapshots at $t = (0, 4, 8, 100, 180, 190, 196, 200)/200$

$T = 0.01$



Snapshots at $t = (0, 4, 8, 100, 180, 190, 196, 200)/20000$

$T = 0.1, \nu = 0.125, \Phi(m) = -\log(m)$



Snapshots at $t = (0, 4, 8, 100, 180, 190, 196, 200)/2000$

II. Strategies for solving the discrete problem

Difficulty: time dependent problem with conditions at both initial and final times.

$$\begin{cases} \mathcal{F}_U(\mathcal{U},\mathcal{M}) & = & 0, & \text{(discrete HJB)} \\ \mathcal{F}_M(\mathcal{U},\mathcal{M}) & = & 0 & \text{(discrete Fokker-Planck)}, \end{cases}$$

Strategy: Newton method

$$\begin{pmatrix} \mathcal{U} \\ \mathcal{M} \end{pmatrix} \longleftarrow \begin{pmatrix} \mathcal{U} \\ \mathcal{M} \end{pmatrix} - \begin{pmatrix} A_{U,U}(\mathcal{U},\mathcal{M}) & A_{U,M}(\mathcal{U},\mathcal{M}) \\ A_{M,U}(\mathcal{U},\mathcal{M}) & A_{M,M}(\mathcal{U},\mathcal{M}) \end{pmatrix}^{-1} \begin{pmatrix} \mathcal{F}_U(\mathcal{U},\mathcal{M}) \\ \mathcal{F}_M(\mathcal{U},\mathcal{M}) \end{pmatrix}$$

where

$$\begin{aligned} A_{U,U}(\mathcal{U},\mathcal{M}) &= D_{\mathcal{U}}\mathcal{F}_{\mathcal{U}}(\mathcal{U},\mathcal{M}), & A_{U,M}(\mathcal{U},\mathcal{M}) &= D_{\mathcal{M}}\mathcal{F}_{\mathcal{U}}(\mathcal{U},\mathcal{M}), \\ A_{M,U}(\mathcal{U},\mathcal{M}) &= D_{\mathcal{U}}\mathcal{F}_{\mathcal{M}}(\mathcal{U},\mathcal{M}), & A_{M,M}(\mathcal{U},\mathcal{M}) &= D_{\mathcal{M}}\mathcal{F}_{\mathcal{M}}(\mathcal{U},\mathcal{M}). \end{aligned}$$

The linear systems

For simplicity, we assume that $\Phi_0(m)$ does not depend of m, so the initial condition is

$$u_{|t=0} = u_0.$$

We are led to study the linearized discrete MFG system

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

where $\mathcal{U} = (U^1, \dots, U^{N_T})^T$ and $\mathcal{M} = (M^0, \dots, M^{N_T-1})^T$.

The matrices A_{UU} and A_{UM} have the form:

$$A_{UU} = \begin{pmatrix} D_1 & & & \\ -\frac{1}{\Delta t} I & D_2 & & \\ & \ddots & \ddots & \\ & & -\frac{1}{\Delta t} I & D_{N_T} \end{pmatrix} \text{ and } A_{UM} = \text{Block-Diag}(E_1, \dots, E_{N_T}).$$

- A_{UU} corresponds to a linearized HJB equation and the block D_n correponds to the finite difference operator

$$(Z_{i,j}) \mapsto (Z_{i,j}/\Delta t - \nu(\Delta_h Z)_{i,j} + [D_h Z]_{i,j} \cdot \nabla g(x_{i,j}, [D_h U^n]_{i,j}))\,.$$

Monotonicity $\Rightarrow$ D_n is a M-matrix, thus A_{UU} is invertible.

- **The blocks E_n are diagonal matrices, with negative diagonal entries if $m \to \Phi(m)$ is strictly increasing. E_n^{-1} is available.**

The matrices A_{MM} and A_{MU} have the form

$$A_{MM} = A_{UU}^T, \quad \text{and} \quad A_{MU} = \text{Block-Diag}(\widetilde{E}_1, \ldots, \widetilde{E}_{N_T}).$$

- A_{MM} **corresponds to a linear transport equation.**
- Note that

$$\mathcal{V}^T \widetilde{E}_n \mathcal{W} = \sum_{i,j} M_{i,j}^{n-1} [D_h V]_{i,j} \cdot D_{q,q}^2 g(x_{i,j}, [D_h U^n]_{i,j}) [D_h W]_{i,j}.$$

From the convexity of g, $\widetilde{E}_n$ is positive if $M^{n-1} \geq 0$.

Th. If Φ is strictly increasing and if $\mathcal{M} \geq 0$, then the Jacobian matrix

$$\begin{pmatrix} A_{U,U}(\mathcal{U},\mathcal{M}) & A_{U,M}(\mathcal{U},\mathcal{M}) \\ A_{M,U}(\mathcal{U},\mathcal{M}) & A_{M,M}(\mathcal{U},\mathcal{M}) \end{pmatrix} \text{ is invertible.}$$

Proof: similar to the proof of uniqueness of the MFG system of PDE.

Two iterative strategies for solving the linearized discrete MFG system

$$\left(\begin{array}{cc} A_{U,U} & A_{U,M} \\ A_{M,U} & A_{M,M} \end{array} \right) \left(\begin{array}{c} \mathcal{U} \\ \mathcal{M} \end{array} \right) = \left(\begin{array}{c} G_U \\ G_M \end{array} \right).$$

First strategy

1. Solve first $A_{U,U}\widetilde{\mathcal{U}} = G_U$. This is done by sequentially solving

$$D_k \widetilde{U}^k = -\Delta t^{-1} \widetilde{U}^{k-1} + G_U^k, \tag{1}$$

i.e. marching in time in the forward direction. Systems (1) are solved with efficient direct solvers.

2. Introducing $\overline{\mathcal{U}} = \mathcal{U} - \widetilde{\mathcal{U}}$,

$$\begin{pmatrix} A_{U,U} & A_{U,M} \\ A_{M,U} & A_{M,M} \end{pmatrix} \begin{pmatrix} \overline{\mathcal{U}} \\ \mathcal{M} \end{pmatrix} = \begin{pmatrix} 0 \\ G_M - A_{M,U}\widetilde{\mathcal{U}} \end{pmatrix}$$

$$\Rightarrow \qquad \left(A_{M,M} - A_{M,U} A_{U,U}^{-1} A_{U,M}\right) \mathcal{M} = G_M - A_{M,U}\widetilde{\mathcal{U}}. \tag{2}$$

(2) is solved by an iterative method **which does not require assembling** $A_{M,M} - A_{M,U} A_{U,U}^{-1} A_{U,M}$, e.g. BiCGStab.

Speeding the method

Instead of

$$\left(A_{M,M} - A_{M,U}A_{U,U}^{-1}A_{U,M}\right)\mathcal{M} = G_M - A_{M,U}\widetilde{\mathcal{U}},$$

we rather solve

$$\left(I - A_{M,M}^{-1}A_{M,U}A_{U,U}^{-1}A_{U,M}\right)\mathcal{M} = A_{M,M}^{-1}(G_M - A_{M,U}\widetilde{\mathcal{U}}). \qquad (3)$$

Left multiplication by $A_{M,M}^{-1}$ $\Leftrightarrow$ solving a backward in time discrete transport problem.

This is done by marching backward in time, and solving at each time step a system of the form

$$D_k^T M^k = \Delta t^{-1} M^{k+1} + F^k, \qquad (4)$$

(note that D_k^T is invertible from the monotonicity of the scheme). Systems (4) are solved with efficient direct solvers.

Note that the matrix $I - A_{M,M}^{-1} A_{M,U} A_{U,U}^{-1} A_{U,M}$ is not assembled.

- **PDE interpretation** $A_{M,M}^{-1}A_{M,U}A_{U,U}^{-1}A_{U,M}$ is the discrete version of

$$(\text{linear-FP})^{-1} \circ \text{div}\,(mH_{pp}(Du)D\cdot) \circ (\text{linear-HJB})^{-1} \circ (\Phi'(m)\cdot).$$

If $\nu > 0$ and if m and u are smooth, this is a **compact operator in L^2.**

Thus, the matrix $I - A_{M,M}^{-1}A_{M,U}A_{U,U}^{-1}A_{U,M}$ is expected to have **a nice condition number**, which should not depend on h and Δt. As we shall see, the iterative method has a fast convergence.

- **Complexity** The complexity of the method is mainly that of solving the systems

$$D_k\widetilde{U}^k = -\Delta t^{-1}\widetilde{U}^{k-1} + G_U^k,$$

and

$$D_k^T M^k = \Delta t^{-1}M^{k+1} + F^k,$$

for $k = 1, \ldots, N_T$.

Table 1: First iterative strategy for solving the linearized MFG system: average number of iterations to decrease the residual by a factor 10^{-7}

ν	$32 \times 32 \times 32$	$64 \times 64 \times 64$	$128 \times 128 \times 64$
0.6	2	2	2
0.36	2	2	2
0.2	3.5	3.5	4
0.12	6	6	6.1

Second strategy for solving the linear systems when Φ is strictly monotone.

- This strategy is inspired by the proof of uniqueness for the MFG system.
- The idea is to eliminate m from the linearized HJB equation: this is possible since Φ is strictly monotone.

The system reads

$$\begin{pmatrix} A_{U,U} & A_{U,M} \\ A_{M,U} & A_{M,M} \end{pmatrix} \begin{pmatrix} \mathcal{U} \\ \mathcal{M} \end{pmatrix} = \begin{pmatrix} G_U \\ G_M \end{pmatrix}.$$

Eliminating $\mathcal{M}$ from the first block of equations, we get a system of the form

$$\left(A_{M,U} - A_{M,M}(A_{U,M})^{-1}A_{U,U}\right)\,\overline{\mathcal{U}} = \mathcal{F}$$

Note that $A_{U,M}$ is diagonal, with negative diagonal entries, so the above matrix can be assembled.

PDE interpretation The partial differential operator in the continuous version of $\left(A_{M,U} - A_{M,M}(A_{U,M})^{-1}A_{U,U}\right)$ is

$$\text{div}\left(m\frac{\partial^2 H(Du)}{\partial p^2}D\cdot\right) - (\text{linear- FP}) \circ ((\Phi'(m))^{-1}\cdot) \circ (\text{linear- HJB}).$$

This is a fourth order differential operator w.r.t. x and second order w.r.t. t. Its principal part is

$$(\Phi'(m))^{-1}\left(-\frac{\partial^2}{\partial t^2} + \nu^2\Delta^2\right)$$

for which a weak elliptic theory may be used (quasi-elliptic).

The boundary conditions at $t = T$ is of the type

$$(\text{linear- HJB})u = g.$$

Consequences

- **Bad news :** The matrix $A_{M,U} - A_{M,M}(A_{U,M})^{-1}A_{U,U}$ **is very ill-conditioned**: the condition number grows like $\nu^2 h^{-4}$. Indeed, we can observe that standard iterative methods like BICGstab do not yield convergence even for $h \sim 1/10$. (BICGstab cannot even reduce the residual by a factor 0.1 for $h = 1/64$)

- **Good news :** degenerate elliptic operator, so we can try solving

$$\left(A_{M,U} - A_{M,M}(A_{U,M})^{-1}A_{U,U}\right)\,\overline{\mathcal{U}} = \mathcal{F}$$

with an iterative method using a **multigrid** **preconditioner**.

Multigrid methods: main ingredients

- **A family of nested grids** : $(\mathcal{G}_\ell)_{\ell=0,\dots,L}$ of step sizes $\sim 2^{-\ell}$: the system to be solved is

$$B_L u_L = f_L.$$

- **Intergrid communications:**
  - **Prolongation operators**, in order to represent a grid function on the next finer grid: $I_\ell^{\ell+1} \ : \ \mathcal{G}_\ell \to \mathcal{G}_{\ell+1}$.
  - **Restriction operators**, in order to interpolate a grid function on the next coarser grid: $I_\ell^{\ell-1} \ : \ \mathcal{G}_\ell \to \mathcal{G}_{\ell-1}$.
- With each grid, we associate a matrix for an approximate system, e.g. $B_\ell = I_{\ell+1}^{\ell} B_{\ell+1} I_\ell^{\ell+1}$.
- **Elementary stationary iterative methods** in order to solve $B_\ell u_\ell = f_\ell$, for example Gauss-Seidel method.

Principle of multigrid methods The basic principle of multigrid method is as follows:

- For an elliptic operator, one can find simple iterative methods (Gauss-Seidel or close to it) such that a few iterations of these methods are enough to damp the higher frequency components of the error, i.e. to make the error smooth.
- These iterative methods have bad convergence properties, but they have good smoothing properties: they are called smoothers.
- For such methods, the produced residual is well represented on the next coarser grid. So the residual is transfered to the next coarser grid.
- This is the basis for a recursive algorithm.

The algorithm (V-cycle)

function MGS (ℓ, u_ℓ, f_ℓ)

  if $\ell = 0$ **then**

    $u_0 \leftarrow B_0^{-1} f_0$ // **level 0 : solve exactly**

  else

    $u_\ell \leftarrow S_\ell(u_\ell, f_\ell, \nu_1)$ // **presmoothing**

    $u_{\ell-1} \leftarrow 0$

    MGS $\left(\ell - 1, u_{\ell-1}, I_\ell^{\ell-1}(f_\ell - B_\ell u_\ell)\right)$ // **coarse gr. correct.**

    $u_\ell \leftarrow S_\ell(u_\ell + I_{\ell-1}^\ell u_{\ell-1}, f_\ell, \nu_2)$ // **postsmoothing**

  endif

endfunction

The multigrid operator can also be used as a preconditioner for the matrix B_L in an iterative solver like BICGstab.

Complexity of a multigrid step: linear

In our case, standard multigrid methods do not behave well !

Table 2: Full coarsening multigrid with 4 levels: average number of iterations to decrease the residual by a factor 0.01

ν	$32 \times 32 \times 32$	$64 \times 64 \times 64$	$128 \times 128 \times 64$
0.6	40	92	-
0.36	24	61	-
0.2	21	45	-

Why? Because the usual smoothers actually make the error smooth in the planes $t = cst$, but not w.r.t. the variable t.

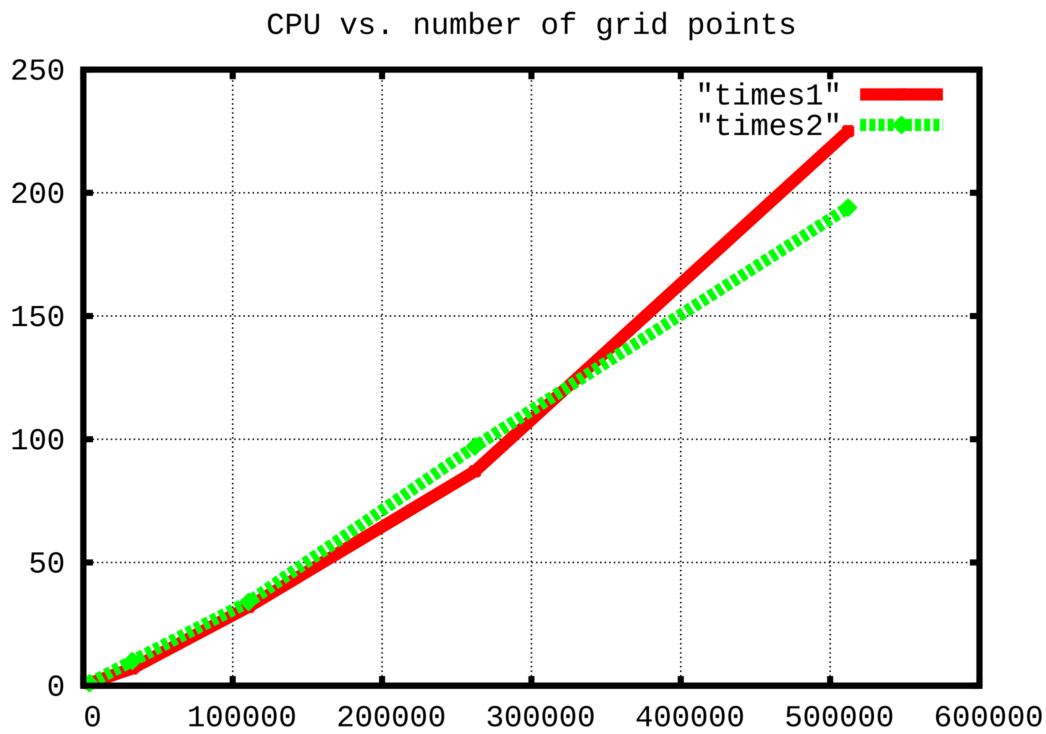
Reason: The unknowns are strongly coupled in the planes $t = cst$, (4-th order operator), stronglier than on the lines $x = cst$, (2nd order operator).

Fix :The hierarchy of nested grids should be obtained by coarsening the grids in the x directions only, but not on the t direction.

Results with the semi-coarsening multigrid methods

Table 3: Semi-coarsening multigrid with 5 levels: average number of iterations to decrease the residual by a factor 0.001

ν	$32 \times 32 \times 32$	$64 \times 64 \times 64$	$128 \times 128 \times 64$
0.6	4	5	7
0.36	4	5	7
0.2	4	5.5	7
0.12	6	9	12



CPU times for a Newton iterate vs. number of grid points for the two strategies

Conclusion

- Two iterative strategies that work well in a rather broad setting.
- The first one looks more robust if Φ has the bad monotonicity.
- Also the nonlinear part of the solver needs improvements.