COMMUN. MATH. SCI. © 2015 International Press
Vol. 13, No. 8, pp. 2177–2194

CENTRAL SCHEMES FOR MEAN FIELD GAMES*

BOJAN POPOV† AND VLADIMIR TOMOV‡

Abstract. Mean field type models have been recently introduced and analyzed by Lasry and Lions. They describe a limiting behavior of stochastic differential games as the number of players tends to infinity. Numerical methods for the approximation of such models have been developed by Achdou, Camilli, Capuzzo-Dolcetta, Gueant, and others. Efficient algorithms for such problems require special efforts and so far all methods introduced have been first order accurate. In this manuscript we design a second order accurate numerical method for time dependent Mean Field Games. The discretization is based on central schemes which are widely used in hyperbolic conservation laws.

Key words. mean field games, central schemes.

AMS subject classifications. 65M06, 65M12, 65H10.

1. Introduction

The Mean Field Games (MFG) equations describe situations that arise in economics, finance or other related subjects where a large number of individual players choose their optimal strategy by considering global (but limited) cost information that is available to everyone. As time evolves, each player's actions alters the cost information which leads to changes in the players' strategies. The mathematical model of such problems has first been introduced by Lions and Lasry in [13], and in one space dimension corresponds to the following system of equations:

$$\frac{\partial u}{\partial t} + H\left(\frac{\partial u}{\partial x}\right) = f(x,m) + \sigma\frac{\partial^2 u}{\partial^2 x}, \tag{1.1}$$

$$\frac{\partial m}{\partial t} + \frac{\partial}{\partial x}\left[H'\left(\frac{\partial u}{\partial x}\right)m\right] = -\sigma\frac{\partial^2 m}{\partial^2 x}, \tag{1.2}$$

$$u(x,0) = u_0(x), \quad m(x,T) = m_T(x), \ m > 0, \int_\Omega m \, dx = 1 \text{ for all } t \in [0,T],$$

where m is a distribution of players, u is a cost function, and σ is a volatility factor. Equation (1.1) is a forward Hamilton–Jacobi (FHJ) equation for u with a Hamiltonian H, a source f and a diffusion term. The source $f(x,m)$ describes how the players' actions affect the cost information. Equation (1.2) is a backward convection-diffusion (BCD) equation for m with a diffusion term. The advection term $H'\left(\frac{\partial u}{\partial x}\right)m$ describes how the cost function influences each player's actions.

There are already many numerical methods developed to solve the MFG system (1.1)–(1.2), see [3, 1, 7, 4, 12]. The main goal so far has been to find a stable and convergent discretization of the MFG model. This is usually done via monotone discretizations of the Hamiltonian in (1.1), a suitable weak formulation of (1.2) and some

*Received: April 2, 2014; accepted (in revised form): November 17, 2014. Communicated by Eitan Tadmor.

†Department of Mathematics, Texas A&M University, 3368 TAMU, College Station, TX 77843, USA (popov@math.tamu.edu).

This research was supported in part by the National Science Foundation grant DMS-1217262 and by the Air Force Office of Scientific Research, USAF, under grant/contract FA99550-12-0358.

‡Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551, USA (tomov2@llnl.gov).

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, LLNL-JRNL-674319.

implicit coupling to guarantee existence, uniqueness of the discrete solution, and convergence towards the exact solution of the system. However, it is well known that using monotone methods results in at most first order accurate approximate solutions. To the best of our knowledge, the construction and the convergence analysis of a second order accurate method for the MFG system is an open problem.

Central schemes are the standard tool for numerical approximation of hyperbolic conservation equations, the first such scheme was introduced by Lax in [14] and the first second order extension was given by Nessyahu and Tadmor in [19]. Their main feature is simplicity since they don't involve Riemann solvers, and their structure allows efficient parallelization. Utilizing these schemes for general convection-diffusion equations is straightforward, and by exploiting the general Hamilton–Jacobi equations' relation to conservation laws, see for example [5, 6], we can apply central schemes to those equations as well.

The main idea of this work is to modify and apply the existing explicit second order central schemes to each individual MFG equation and then to combine them into a fixed point iteration algorithm for the MFG system. The analysis of second order central schemes for transport and Hamilton–Jacobi equations is not a simple task, see for example [16, 18, 20]. At this point we do not have a convergence result for our new scheme for the MFG system. This paper is organized as follows:

- We derive a fully discrete explicit second order staggered finite difference scheme for the FHJ Equation (1.1) in Section 2. The algorithm we propose is a modification of the method derived by Lin and Tadmor in [17].
- We derive a fully discrete explicit second order staggered finite difference scheme for the BCD Equation (1.2) in Section 3. The scheme is based on the classical Nessyahu–Tadmor scheme from [19].
- Both of our schemes are combined into a fixed point iteration algorithm that solves the MFG equations in Section 4. We also describe how the two schemes interact in time, memory issues, and stopping criteria.
- Numerical results, convergence, and computational speed tests are presented in Section 5.
- We compare our approach to some already existing MFG numerical algorithms in Section 6.

2. Discretization of the forward Hamilton–Jacobi equation

Hamilton–Jacobi equations are closely related to conservation laws. If we consider the two equations:

$$\frac{\partial u}{\partial t}+H\left(\frac{\partial u}{\partial x}\right)=0, \qquad u(x,t)=u_0(x) \quad \text{and} \tag{2.1}$$

$$\frac{\partial \varphi}{\partial t}+\frac{\partial F(\varphi)}{\partial x}=0, \qquad \varphi(x,0)=\varphi_0(x), \tag{2.2}$$

then $u(x,t)$ is the unique physical solution (called viscosity solution) of (2.1) if and only if $\varphi(x,t)=\frac{\partial}{\partial x}u(x,t)$ is the unique physical solution (called entropy solution) of the conservation law (2.2) with flux $F(\varphi)=H\left(\frac{\partial u}{\partial x}\right)$ and initial condition $\varphi_0(x)=u_0(x)$. Details about this relation can be found in [5], and extension to multiple dimensions through numerical observations is given in [10]. Using this idea, schemes that are initially created for conservation laws can be applied to Hamilton–Jacobi equations, e.g., [11, 17, 21, 16, 9]. In this section we use the same approach and derive a modified version of the scheme presented in [17] which is suitable for the FHJ Equation (1.1).

We discretize our domain Ω by the grid points $x_j = j\Delta x$. The discrete points in time are $t_n = n\Delta t_{hj}$, where Δt_{hj} stands for the time step for the FHJ Equation (1.1). Note that here we march forward in time. Let u_j^n be the approximate value of $u(x_j,t_n)$. We think of our discrete approximation $u(\cdot,t_n)$ as a continuous, piecewise quadratic function with values u_j^n at the grid points x_j. Its first and second order spatial derivatives are defined as follows:

$$(\hat{u}_x)^n_{j+\frac{1}{2}} := \frac{u^n_{j+1} - u^n_j}{\Delta x}, \tag{2.3}$$

$$\begin{aligned}(\hat{u}_{xx})^n_{j+\frac{1}{2}} := \frac{1}{\Delta x}\text{minmod}\Bigg[&\theta\left((\hat{u}_x)^n_{j+\frac{3}{2}} - (\hat{u}_x)^n_{j+\frac{1}{2}}\right),\\ &\frac{1}{2}\left((\hat{u}_x)^n_{j+\frac{3}{2}} - (\hat{u}_x)^n_{j-\frac{1}{2}}\right),\\ &\theta\left((\hat{u}_x)^n_{j+\frac{1}{2}} - (\hat{u}_x)^n_{j-\frac{1}{2}}\right)\Bigg],\end{aligned} \tag{2.4}$$

where “minmod” is the well known nonlinear limiter:

$$\text{minmod}(a_1,a_2,\dots) := \begin{cases}\min_j(a_j), & \text{if } a_j > 0 \quad \forall j,\\ \max_j(a_j), & \text{if } a_j < 0 \quad \forall j,\\ 0 & \text{otherwise.}\end{cases} \tag{2.5}$$

The parameter θ in (2.4) must be in $[1,2]$ in order to prevent oscillations, larger values introduce less dissipation, i.e., $\theta = 2$ (which is our choice here) is the least dissipative pick, see [22, 15]. Then for $x \in [x_j, x_{j+1}]$ we define the discrete interpolant

$$\hat{u}(x,t_n) := u^n_j + (\hat{u}_x)^n_{j+\frac{1}{2}}(x - x_j) + \frac{1}{2}(\hat{u}_{xx})^n_{j+\frac{1}{2}}(x - x_j)(x - x_{j+1}). \tag{2.6}$$

As further explained in Section 3, let $\hat{m}(x,t)$ be the approximation of $m(x,t)$ and $m^n_{j+\frac{1}{2}}, (\hat{m}_x)^n_{j+\frac{1}{2}}$ be the value and first spatial derivative of $\hat{m}(x_{j+\frac{1}{2}}, t_n)$. Suppose we already have the values u^n_j, then the next staggered values in time are derived by integrating (1.1) over $[t_n, t_{n+1}]$ and evaluating at $x_{j+\frac{1}{2}}$:

$$\begin{aligned}u^{n+1}_{j+\frac{1}{2}} =& \hat{u}(x_{j+\frac{1}{2}}, t_n)\\ &+ \int_{t_n}^{t_{n+1}} \left(-H(\hat{u}_x(x_{j+\frac{1}{2}},t)) + f(x_{j+\frac{1}{2}}, \hat{m}(x_{j+\frac{1}{2}},t)) + \sigma\hat{u}_{xx}(x_{j+\frac{1}{2}},t)\right) dt.\end{aligned} \tag{2.7}$$

At this point, in order to get values of $\hat{u}_x$ in the time interval (t_n, t_{n+1}), we use the relation of our FHJ problem to conservation laws (2.2), namely in our case $\hat{u}_x$ satisfies the conservation law

$$\frac{\partial}{\partial t}(\hat{u}_x) + \frac{\partial}{\partial x}H(\hat{u}_x) = \frac{\partial}{\partial x}f(x,\hat{m}) + O(\Delta x^2). \tag{2.8}$$

Here we have ignored the diffusion term, because the error from doing that does not affect the second order accuracy of the scheme, see Subsection A.1. Equation (2.8) has finite propagation speed, which means that under a standard hyperbolic CFL condition on the time step

$$\frac{\Delta t_{hj}}{\Delta x}\max_x |H'(\hat{u}_x)| \leq \frac{1}{2}, \tag{2.9}$$

our interpolant's spatial derivatives $\hat{u}_x, \hat{u}_{xx}$ are expected to remain well-defined around $x_{j+\frac{1}{2}}$ for $t \in [t_n, t_{n+1}]$. Then we can use a quadrature rule for the integral in (2.7). We use the midpoint rule where the midpoint values in time are computed by Taylor expansion of $\hat{u}_x$ that uses the time derivative from Equation (2.8). Namely, we define

$$\begin{aligned}(\hat{u}_x)_{j+\frac{1}{2}}^{n+\frac{1}{2}} = (\hat{u}_x)_{j+\frac{1}{2}}^{n} + \frac{\Delta t_{hj}}{2}\Bigg[& -H'\left((\hat{u}_x)_{j+\frac{1}{2}}^{n}\right)(\hat{u}_{xx})_{j+\frac{1}{2}}^{n} \\ & + f_x(x_{j+\frac{1}{2}}, m_{j+\frac{1}{2}}^{n}) + f_m(x_{j+\frac{1}{2}}, m_{j+\frac{1}{2}}^{n})(\hat{m}_x)_{j+\frac{1}{2}}^{n}\Bigg].\end{aligned} \tag{2.10}$$

The above expansion is only done up to first derivative because this is sufficient to provide the desired accuracy, see Subsection A.2. After we apply the midpoint rule in (2.7) and substitute (2.10), (2.6) into (2.7), we get the following forward staggered scheme for the FHJ Equation (1.1):

$$\begin{aligned}u_{j+\frac{1}{2}}^{n+1} = & \frac{1}{2}\left(u_j^n + u_{j+1}^n\right) - \frac{(\Delta x)^2}{8}(\hat{u}_{xx})_{j+\frac{1}{2}}^{n} \\ & + \Delta t_{hj}\left[-H\left((\hat{u}_x)_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right) + f(x_{j+\frac{1}{2}}, m_{j+\frac{1}{2}}^{n+\frac{1}{2}}) + \sigma\frac{(\hat{u}_x)_{j+\frac{3}{2}}^{n} - (\hat{u}_x)_{j-\frac{1}{2}}^{n}}{2\Delta x}\right]\end{aligned} \tag{2.11}$$

where for the σ term instead of using $(\hat{u}_{xx})_{j+\frac{1}{2}}^{n+\frac{1}{2}}$ computed by (2.4) at $t_{n+\frac{1}{2}}$ (which requires a lot of operations), we apply a simple central difference at time t_n for the second derivative. This approach provides reduction of the computational cost and is sufficient to achieve second order accurate discretization (see Subsection A.1).

The time step $\Delta t_{hj} = t_{n+1} - t_n$ for this scheme must take into account not only the hyperbolic CFL condition (2.9), but also the presence of the Laplace term, namely

$$\Delta t_{hj} := \min\left(\frac{c\Delta x}{\max_j \left|H'\left((\hat{u}_x)_{j+\frac{1}{2}}^{n}\right)\right|}, \frac{c(\Delta x)^2}{\sigma}\right), \tag{2.12}$$

where c is a CFL constant, we usually use 0.4. The term involving σ is derived from positivity preservation: if we suppose $f = H = \hat{u}_{xx} = 0$ and $u_j^n \geq 0\ \forall j$ in (2.11), then we enforce $u_{j+\frac{1}{2}}^{n+1} \geq 0$ by

$$\frac{1}{2}\left(u_j^n + u_{j+1}^n\right) - \sigma\frac{\Delta t}{2\Delta x^2}\left(u_j^n + u_{j+1}^n\right) \geq 0,\ \forall j \Rightarrow \Delta t \leq \frac{\Delta x^2}{\sigma}.$$

The expression (2.12) is recomputed before each time step, because the dependence of (2.8) on $\hat{m}$ causes changes in the local maximum of H'.

REMARK 2.1. Depending on σ, in (2.12) we may have $\Delta t = O(\Delta x)$ or $\Delta t = O(\Delta x^2)$. We say that our simulation is in 'hyperbolic regime" when $\Delta t = O(\Delta x)$, and we say that our simulation is in "parabolic regime" when $\Delta t = O(\Delta x^2)$. In a hyperbolic regime we have $\sigma = O(\Delta x)$, but in a parabolic regime we have $\sigma = O(1)$.

Remark 2.2. We can define a simpler version of (2.11) by choosing continuous, piecewise linear approximation with values u_j^n at the grid points x_j. This corresponds to the choice $(\hat{u}_{xx})_{j+\frac{1}{2}}^n = 0$, instead of the definition (2.4). Then we can apply the above derivation by using a left-point rule for the integral in Equation (2.7) and obtain the scheme

$$\begin{aligned} u_{j+\frac{1}{2}}^{n+1} =& \frac{1}{2}\left(u_j^n + u_{j+1}^n\right) \\ &+ \Delta t_{hj}\left[-H\left((\hat{u}_x)_{j+\frac{1}{2}}^n\right) + f(x_{j+\frac{1}{2}}, m_{j+\frac{1}{2}}^n) + \sigma \frac{(\hat{u}_x)_{j+\frac{3}{2}}^n - (\hat{u}_x)_{j-\frac{1}{2}}^n}{2\Delta x}\right]. \end{aligned} \tag{2.13}$$

This scheme uses the same time step computation as in (2.12), and requires much less operations than (2.11). However, as derived later and verified numerically, the scheme (2.13) does not produce second order convergent method.

Remark 2.3. To guarantee second order accuracy in the parabolic regime, it is necessary to use the minmod limiter (2.4) with $\theta = 2$, or the UNO limiter in (2.11) (see Equation (3.1) for definition). The derivation of the truncation error is given in Subsection A.2.

3. Discretization of the backward convection-diffusion equation

In this section we derive a modification of the central scheme presented in [19] to discretize the BCD Equation (1.2). We use the same spatial grid points $x_j = j\Delta x$ as in Section 2. However, the discrete points in time are different. We consider $t_k = k\Delta t_{cd}$ where Δt_{cd} stands for the time step for the BCD Equation (1.2). Note that in this algorithm we march backwards in time. We think of our discrete approximation as a piecewise linear function $\hat{m}$ where $m_{j+\frac{1}{2}}^n$ is its average value for the cell $[x_j, x_{j+1}]$ (or the value at $x_{j+\frac{1}{2}}$). The spatial derivative $(\hat{m}_X)_{j+\frac{1}{2}}$ at $x_{j+\frac{1}{2}}$ is constructed using the uniformly non-oscillatory (UNO) flux limiter introduced in [8]:

$$\begin{aligned} (\hat{m}_X)_{j+\frac{1}{2}}^k := \frac{1}{\Delta x}\text{minmod}\Bigg(& m_{j+\frac{1}{2}}^k - m_{j-\frac{1}{2}}^k + \frac{1}{2}\text{minmod}(\Delta^2 m_{j-\frac{1}{2}}^k, \Delta^2 m_{j+\frac{1}{2}}^k), \\ & m_{j+\frac{3}{2}}^k - m_{j-\frac{1}{2}}^k - \frac{1}{2}\text{minmod}(\Delta^2 m_{j+\frac{1}{2}}^k, \Delta^2 m_{j+\frac{3}{2}}^k)\Bigg), \end{aligned} \tag{3.1}$$

$$\Delta^2 m_{j+\frac{1}{2}}^k := m_{j+\frac{3}{2}}^k - 2m_{j+\frac{1}{2}}^k + m_{j-\frac{1}{2}}^k,$$

and sometimes we use the minmod limiter that doesn't need as many values, namely

$$(\hat{m}_x)_{j+\frac{1}{2}}^k := \frac{1}{\Delta x}\text{minmod}\left(m_{j+\frac{1}{2}}^k - m_{j-\frac{1}{2}}^k, m_{j+\frac{3}{2}}^k - m_{j+\frac{1}{2}}^k\right). \tag{3.2}$$

Then for $x \in [x_j, x_{j+1}]$ the approximation function $\hat{m}$ has the form

$$\hat{m}(x, t_k) = m_{j+\frac{1}{2}}^k + \frac{1}{\Delta x}(x - x_{j+\frac{1}{2}})(\hat{m}_X)_{j+\frac{1}{2}}^k. \tag{3.3}$$

Suppose we already have the values $m^k_{j+\frac{1}{2}}$, then the next staggered values in time, going backwards, are obtained by integrating (1.2) over $[t_k,t_{k-1}]$ and $[x_{j+\frac{1}{2}},x_{j+\frac{3}{2}}]$:

$$\begin{aligned}\int_{x_{j+\frac{1}{2}}}^{x_{j+\frac{3}{2}}} &\hat{m}(x,t_{k-1})-\hat{m}(x,t_k)\ dx\\ &+\int_{t_k}^{t_{k-1}}\left[H'(\hat{u}_x(x_{j+\frac{3}{2}},t))\hat{m}(x_{j+\frac{3}{2}},t)-H'(\hat{u}_x(x_{j+\frac{1}{2}},t))\hat{m}(x_{j+\frac{1}{2}},t)\right]\ dt\\ =&-\sigma\int_{t_k}^{t_{k-1}}\hat{m}_X(x_{j+\frac{3}{2}},t)-\hat{m}_X(x_{j+\frac{1}{2}},t)\ dt. \end{aligned}\tag{3.4}$$

Similar to Section 2, the BCD Equation (1.2) has a finite speed of propagation, hence with the standard hyperbolic CFL condition on the time step

$$\frac{\Delta t_{cd}}{\Delta x}\max_x|H'(\hat{u}_x)|\leq\frac{1}{2},\tag{3.5}$$

the value of $\hat{m}$ and its spatial derivative $\hat{m}_x$ remain well-defined around $x_{j+\frac{1}{2}}$ for $t\in[t_{n-1},t_n]$. Then we can safely use a quadrature rule for the time integrals in (3.4). We use the midpoint rule where the midpoint values in time are computed by Taylor expansion of $\hat{m}$ that uses the time derivative from Equation (1.2), namely we define

$$m^{k-\frac{1}{2}}_{j+\frac{1}{2}}=m^k_{j+\frac{1}{2}}+\frac{\Delta t_{cd}}{2}\left[H''\left((\hat{u}_x)^k_{j+\frac{1}{2}}\right)(\hat{u}_{xx})^k_{j+\frac{1}{2}}m^k_{j+\frac{1}{2}}+H'\left((\hat{u}_x)^k_{j+\frac{1}{2}}\right)(\hat{m}_x)^k_{j+\frac{1}{2}}\right],\tag{3.6}$$

where we ignore the diffusion term and use the less accurate approximation $\hat{m}_x$ instead of $\hat{m}_X$ without affecting the method's second order accuracy, see Subsection A.3. After we apply the midpoint rule and substitute (3.6) for the time integrals of (3.4), and use (3.3) for the space integral of (3.4), we get the following backward staggered scheme for the BCD Equation (1.2):

$$\begin{aligned}m^{k-1}_{j+1}=&\frac{1}{2}\left(m^k_{j+\frac{1}{2}}+m^k_{j+\frac{3}{2}}\right)-\frac{\Delta x}{8}\left((\hat{m}_X)^k_{j+\frac{3}{2}}-(\hat{m}_X)^k_{j+\frac{1}{2}}\right)\\ &+\Delta t_{cd}\left(\frac{H'\left((\hat{u}_x)^{k-\frac{1}{2}}_{j+\frac{3}{2}}\right)m^{k-\frac{1}{2}}_{j+\frac{3}{2}}-H'\left((\hat{u}_x)^{k-\frac{1}{2}}_{j+\frac{1}{2}}\right)m^{k-\frac{1}{2}}_{j+\frac{1}{2}}}{\Delta x}\right.\\ &\qquad\left.+\sigma\frac{m^k_{j+\frac{5}{2}}-m^k_{j+\frac{3}{2}}-m^k_{j+\frac{1}{2}}+m^k_{j-\frac{1}{2}}}{2\Delta x^2}\right),\end{aligned}\tag{3.7}$$

where m^{k-1}_{j+1} is the average for the staggered cell $[x_{j+\frac{1}{2}},x_{j+\frac{3}{2}}]$, and the exact computation of $(\hat{u}_x)^{k-\frac{1}{2}}_{j+\frac{3}{2}},(\hat{u}_x)^{k-\frac{1}{2}}_{j+\frac{1}{2}}$ is given in Subsection 4.1. Similar to the approach in (2.11), for the σ term instead of using the difference between the midpoint values $(\hat{m}_X)^{k-\frac{1}{2}}_{j+\frac{3}{2}}$ and $(\hat{m}_X)^{k-\frac{1}{2}}_{j+\frac{1}{2}}$, we apply a standard central difference at time t_k. Doing this allows us to reduce computational cost while maintaining second order accuracy, see Subsection A.3.

The time step $\Delta t_{cd}=t_k-t_{k-1}$ for this scheme must take into account not only the hyperbolic CFL condition (3.5), but also the presence of the Laplace term:

$$\Delta t_{cd}:=\min\left(\frac{c\Delta x}{\max_j\left|H'\left((\hat{u}_x)^k_{j+\frac{1}{2}}\right)\right|},\frac{c(\Delta x)^2}{\sigma}\right),\tag{3.8}$$

where the derivation of the term involving σ and the CFL constant c are the same as in (2.12). This expression is recomputed before each time step, because the maximum of H' changes.

Remark 3.1. The scheme (3.7) preserves initial mass (up to contributions from the boundary). If the mass at time t_k is

$$M_k = \sum_{j=0}^{n-1} m^k_{j+\frac{1}{2}},$$

then the mass at time t_{k-1} is obtained by summing (3.7) over all $j = 0...n-1$:

$$\begin{aligned} M_{k-1} = & M_k - \frac{1}{2} m^k_{\frac{1}{2}} + \frac{1}{2} m^k_{n+\frac{1}{2}} - \frac{\Delta x}{8}\left((\hat{m}_X)^k_{n+\frac{1}{2}} - (\hat{m}_X)^k_{\frac{1}{2}} \right) \\ & + \Delta t_{cd} \left(\frac{H'\left((\hat{u}_x)^{k-\frac{1}{2}}_{n+\frac{1}{2}} \right) m^{k-\frac{1}{2}}_{n+\frac{1}{2}} - H'\left((\hat{u}_x)^{k-\frac{1}{2}}_{\frac{1}{2}} \right) m^{k-\frac{1}{2}}_{\frac{1}{2}}}{\Delta x} \right. \\ & \left. \qquad + \sigma \frac{m^k_{n+\frac{3}{2}} - m^k_{n-\frac{1}{2}} - m^k_{\frac{3}{2}} + m^k_{-\frac{1}{2}}}{2\Delta x^2} \right). \end{aligned}$$

We see that the mass may change due to a limited number of boundary terms. These terms cancel each other for the case of periodic boundary conditions (i.e., $m_{n+p+\frac{1}{2}} = m_{p+\frac{1}{2}}, u_{n+p} = u_p$ for any integer p). However for any other type of boundary conditions (Dirichlet, constant extensions, etc.) the preservation of total mass is true only on continuous level. For such cases the mass error decreases under mesh refinement with linear rate.

Remark 3.2. We can define a simpler version of (3.7) by choosing a piecewise constant function $\hat{m}$ where $m^k_{j+\frac{1}{2}}$ is its value at $x_{j+\frac{1}{2}}$. This corresponds to the choice $(\hat{m}_X)^k_{j+\frac{1}{2}} = (\hat{m}_x)^k_{j+\frac{1}{2}} = 0$, instead of the definitions (3.1) and (3.2). Then we can apply the above derivation by using a left-point rule for the integral in Equation (3.4) and obtain the scheme

$$\begin{aligned} m^{k-1}_{j+1} = & \frac{1}{2}\left(m^k_{j+\frac{1}{2}} + m^k_{j+\frac{3}{2}} \right) \\ & + \Delta t_{cd} \left(\frac{H'\left((\hat{u}_x)^k_{j+\frac{3}{2}} \right) m^k_{j+\frac{3}{2}} - H'\left((\hat{u}_x)^k_{j+\frac{1}{2}} \right) m^k_{j+\frac{1}{2}}}{\Delta x} \right. \\ & \left. \qquad + \sigma \frac{m^k_{j+\frac{5}{2}} - m^k_{j+\frac{3}{2}} - m^k_{j+\frac{1}{2}} + m^k_{j-\frac{1}{2}}}{2\Delta x^2} \right). \end{aligned} \tag{3.9}$$

This scheme uses the same time step computation as in (3.8), and requires much less operations than (3.7). However, as derived later and verified numerically, the scheme (3.9) does not produce second order convergent method.

Remark 3.3. To guarantee second order accuracy in the parabolic regime, it is necessary to use the UNO limiter (3.1) or the minmod limiter (2.4) with $\theta = 2$ in (3.7). The derivation of the truncation error is given in Subsection A.3. Using a clipping type limiter such as standard minmod ($\theta = 1$ in (2.4)) will result in loss a of accuracy in the regions of local extrema and in the parabolic case this will deteriorate the performance of the method from second to first order.

4. Fixed point iteration

In this section we combine the two presented algorithms into a fixed point iteration.

4.1. Interaction between the equations. First we explain how the schemes (2.11), (3.7) obtain values in time from each other. Let us suppose that we know the values $m^k_{j+\frac{1}{2}}, m^{k-2}_{j+\frac{1}{2}}$ for all j where $t_k \geq t_n \geq t_{k-2}$. Looking at the forward scheme (2.10), (2.11), we use a second order interpolation in time:

$$m^n_{j+\frac{1}{2}} := m^{k-2}_{j+\frac{1}{2}} + \frac{m^k_{j+\frac{1}{2}} - m^{k-2}_{j+\frac{1}{2}}}{t_k - t_{k-2}}(t_n - t_{k-2}). \tag{4.1}$$

It's important to note that the values used in (4.1) have the same cell staggering. Namely, the values $m^k_{j+\frac{1}{2}}, m^{k-2}_{j+\frac{1}{2}}$ are defined at all points $x_{j+\frac{1}{2}}$, while the values $m^{k-1}_{j+\frac{1}{2}}$ are undefined, because evolution from $\hat{m}(x,t_k)$ to $\hat{m}(x,t_{k-1})$ by (3.7) would define the values of $\hat{m}(x,t_{k-1})$ only at the grid points x_j. The derivative $(\hat{m}_x)^n_j$ in (2.10) is computed by combining (4.1) and (3.1), and the value $m^{n+\frac{1}{2}}_{j+\frac{1}{2}}$ in (2.11) is computed by applying (4.1) at time $t_{n+\frac{1}{2}}$.

The same approach is used when we consider the backward scheme (3.6), (3.7): suppose we know the values $u^n_{j+\frac{1}{2}}, u^{n+2}_{j+\frac{1}{2}}$ for all j where $t_{n+2} \geq t_k \geq t_n$. Then $u^k_{j+\frac{1}{2}}$ is defined by

$$u^k_{j+\frac{1}{2}} := u^n_{j+\frac{1}{2}} + \frac{u^{n+2}_{j+\frac{1}{2}} - u^n_{j+\frac{1}{2}}}{t_{n+2} - t_n}(t_k - t_n). \tag{4.2}$$

Again, note that the values used in (4.2) have the same cell staggering: the values $u^{n+2}_{j+\frac{1}{2}}, u^n_{j+\frac{1}{2}}$ are defined at all points $x_{j+\frac{1}{2}}$, while the values $u^{n+1}_{j+\frac{1}{2}}$ are undefined, because evolution from $\hat{u}(x,t_n)$ to $\hat{u}(x,t_{n+1})$ by (2.11) would define the values of $\hat{u}(x,t_{n+1})$ only at the grid points x_j. The derivatives $(\hat{u}_x)^k_{j+\frac{1}{2}}, (\hat{u}_{xx})^k_{j+\frac{1}{2}}$ in (3.6), (3.8) are computed by combining (4.2), (2.3) and (2.4), and the ones in (3.7) are obtained by applying (4.2) at $t_{k-\frac{1}{2}}$ and (2.3).

4.2. Difference norms. In order to use a fixed point iteration, we need to define suitable norms for measuring difference between consecutive solutions. We motivate our choice by some theoretical results from [13]. The solution of (1.1), (1.2) is unique, if f is monotone in L^2 and H is strictly convex i.e.,

$$\begin{aligned}&\int_\Omega (f(x,m_1) - f(x,m_2))(m_1 - m_2)\, dx \geq 0, \qquad \forall m_1, \forall m_2,\\ &H(p+q) - H(p) - H'(p)q \geq 0, \qquad \forall p,q \in \mathbb{R}, \text{ equality implies } q = 0.\end{aligned}$$

and under additional assumptions on H, f and u_0, there exist smooth or weak solutions. Then for $\sigma \to 0$ there exists a unique solution s.t. u is Lipschitz and m is a probability measure. Therefore, we use the L^∞ norm for $\hat{u}$ and the following *negative* norm for $\hat{m}$:

$$||\hat{m}^{i+1}(x,t) - \hat{m}^i(x,t)||_* = \int_\Omega \left| \int_0^x (\hat{m}^{i+1}(s,t) - \hat{m}^i(s,t)) ds \right| dx, \tag{4.3}$$

where $\hat{m}^i(x,t), \hat{u}^i(x,t)$ are the solutions obtained after the i-th iteration.

Remark 4.1. The proof for uniqueness of (1.1), (1.2) from [13] can be modified for the case where $-f$ is monotone in L^2 and $-H$ is strictly convex. This is the setting for all numerical tests we present in Section 5.

4.3. Final algorithm. We are ready to state the complete algorithm:

1. $\hat{m}^0(x,t)$ is initialized by the values of $m_T(x)$ at every point $x_{j+\frac{1}{2}}$, let $i=0$.
2. $\hat{u}^{i+1}(x,t)$ is computed by the algorithm from Section 2 using $\hat{m}^i(x,t)$.
3. $\hat{m}^{i+1}(x,t)$ is computed by the algorithm from Section 3 using $\hat{u}^{i+1}(x,t)$.
4. if convergence is achieved, namely

$$||\hat{m}^{i+1}(x,0)-\hat{m}^i(x,0)||_* < \varepsilon \quad \text{and} \quad ||\hat{u}^{i+1}(x,T)-\hat{u}^i(x,T)||_\infty < \varepsilon,$$

then we stop, the solution is $\hat{m}^{i+1},\hat{u}^{i+1}$. Otherwise $i=i+1$, go to 2.

The tolerance we usually use is $\varepsilon = 10^{-6}$. Notice that the algorithm is fully explicit and it doesn't involve any matrix computations. In all numerical tests presented in Section 5 the number of iterations used remains bounded by a constant.

4.4. Memory usage. The memory problem is the following: values computed from steps 2 and 3 must be kept in memory in order to be used for the next iteration of the other equation (the values obtained in step 2 are used in step 3 and vice versa). If our time steps are in the parabolic regime, meaning $\Delta t_{hj},\Delta t_{cd} = O(\Delta x^2)$, and we store all values in time, then the space-time memory consumption would be $O(\Delta x^{-3})$. If the time steps are in hyperbolic regime, meaning $\Delta t_{hj},\Delta t_{cd} = O(\Delta x)$, the problem doesn't exist, because the space-time consumption is the standard $O(\Delta x^{-2})$.

We notice that values of $\hat{m}$ used in the FHJ scheme (2.11) and values of $\hat{u}$ used in the BCD scheme (3.7) are already scaled in a sense by Δt. Since our goal is to achieve $O(\Delta x^2)$ convergence rates, then in parabolic regime it is sufficient to provide accuracy of order $O(\Delta t)$ for these interpolated values, because this would give LTE of order $O(\Delta t^2)$ or GTE of order $O(\Delta t) = O(\Delta x^2)$. This means that storing only $O(\Delta x^{-1})$ instead of $O(\Delta t^{-1})$ values in time, and interpolating for the intermediate times when needed, will preserve the second order accuracy of the method and keep the space-time memory consumption to $O(\Delta x^{-2})$. Our implementation always stores $O(\Delta x^{-1})$ values in time for any $O(\Delta x) \leq \sigma \leq O(1)$ and the resulting method remains second order accurate.

5. Numerical tests

In this section we first show results that are in agreement with the 1D results obtained in [7]. Then we test the convergence properties of the algorithm on a manufactured smooth test case. In the last subsection we demonstrate some computational features of our algorithm. For all tests our CFL constant is $c = 0.4$.

5.1. Test Problem 1. We first examine a test case presented in [7]: it models a maximization problem, i.e., the players are trying to maximize the utility function u. The players see increasing utility in the middle of the domain, but at the same time they prefer to be away from other players:

$$f(x,m) = -16\left(x-\frac{1}{2}\right)^2 - 0.1\max(0,\min(5,m)), \qquad H\left(\frac{\partial u}{\partial x}\right) = -\frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2,$$
$$m_0(x) = \frac{1.0}{1.1}\left[1.0 + 0.2\cos\left(\pi\left(2x-\frac{3}{2}\right)\right)^2\right], \qquad u_T(x) = 0.0.$$

Notice that the system discussed in [7] is forward in time with respect to m and backward with respect to u. In order to simulate the same test case, but with reversed time, we solve the form (1.1), (1.2) by taking the same expressions for f and H, but we switch the initial and final conditions:

$$m_T(x) = \frac{1.0}{1.1}\left[1.0 + 0.2\cos\left(\pi\left(2x - \frac{3}{2}\right)\right)^2\right], \qquad u_0(x) = 0.0.$$

The domain is $[0,1]$, the volatility is $\sigma = 0.5$, the final time is $T = 0.5$ and the boundary conditions are $\frac{\partial u}{\partial x} = \frac{\partial m}{\partial x} = 0$ on both ends. Since in this example σ is big compared to Δx, we optimize memory usage by saving the solutions of $\hat{m}, \hat{u}$ for only 800 time steps (out of 100 000 steps). In Figure 5.1 we show the distribution of players m at final and initial times. In Figure 5.2 we show the cost function u and its gradient $\frac{\partial u}{\partial x}$ at the final time. The result is computed on 400 cells, the fixed point iteration converges on the fifth loop. We observe that our results are in agreement with the ones in [7].

For this problem's boundary conditions our algorithm preserves mass only on a continuous level. The difference between initial and final mass converges to zero linearly under refinement. For the presented simulation on 400 cells the difference is $9.85E{-}3$.

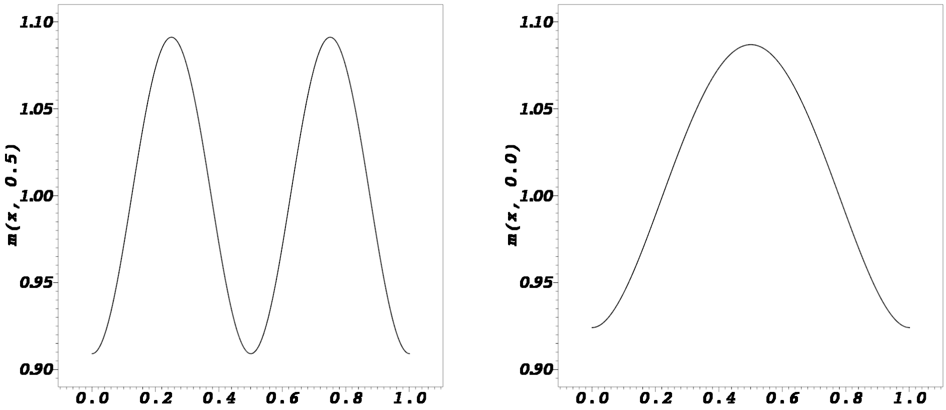

FIG. 5.1. *Plot of $m_T(x)$ (on the left side) and the solution $m(x, 0.0)$ (on the right side) computed on 400 cells for Test Problem 1.*

5.2. Test Problem 2.

The purpose of this example is to verify the method's ability to obtain second order convergence rate for a smooth problem. We use a similar setup as in Test Problem 1, but we initialize $m_T(x)$ by a C^1 function with compact support:

$$m_T(x) = \begin{cases} 4.0\sin^2\left(2\pi\left(x - \frac{1}{4}\right)\right) & x \in [\frac{1}{4}, \frac{3}{4}], \\ 0 & \text{otherwise.} \end{cases}$$

and we keep u smooth by using a similar source:

$$u_0(x) = 0.0, \qquad f(x,m) = 3.0 m_T(x) - \min(4.0, m), \qquad H\left(\frac{\partial u}{\partial x}\right) = -\frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2.$$

The domain is $(0,1)$, the volatility is $\sigma = 0.05$ and in order to keep the solution smooth enough we use a final time $T = 0.05$. We compute convergence speed by considering a reference solution calculated using 3000 cells in space. Each simulation optimizes

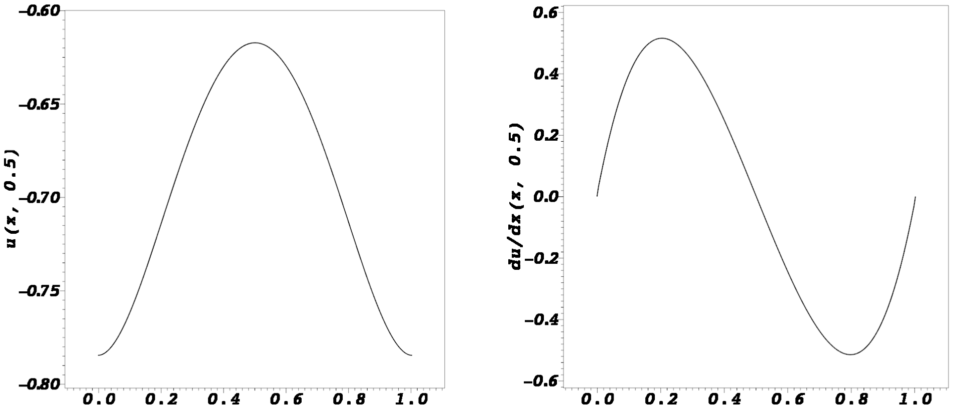

FIG. 5.2. *Solution for u (on the left side) and $\frac{\partial u}{\partial x}$ (on the right side) computed on 400 cells for Test Problem 1.*

memory usage by storing only Δx^{-1} solutions in time. In Figure 5.3 we show the distributions of players m at final and initial times. In Figure 5.4 we show the cost functions u and their gradients u_x at the final time. In Table 5.1 we show convergence speeds for the L^∞ and L^1 norms, and mass preservation. The presented norms are computed by dividing the domain in 10 000 cells, comparing the end points of each cell to obtain the L^∞ norm, and applying a 3-point Gauss quadrature rule in each cell to obtain the L^1 norm and the mass. We observe the expected second order in L^∞ and L^1, and the linear dissipation of the mass error. The mass error for the reference solution is 3.28E-9.

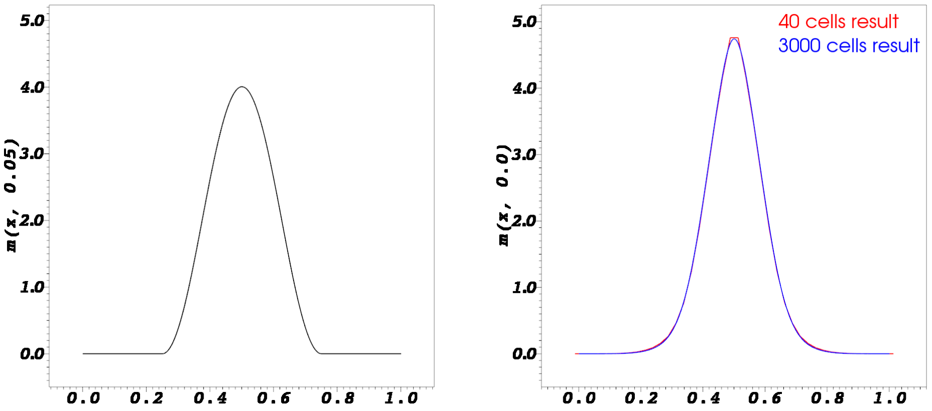

FIG. 5.3. *Plot of $m_T(x)$ (on the left side) and the solution $m(x,0.0)$ (on the right side) computed on 40 and 3000 cells for Test Problem 2.*

5.3. Strong scaling test. Both schemes (2.11), (3.7) admit easy parallelization. Our algorithm is developed on C++ with OpenMP threads. In this section we report execution times and perform a strong scaling test.

The problem we consider is Test Problem 2 on 6000 cells with all other parameters as in Subsection 5.2. We make one iteration of both schemes (2.11), (3.7) that consists

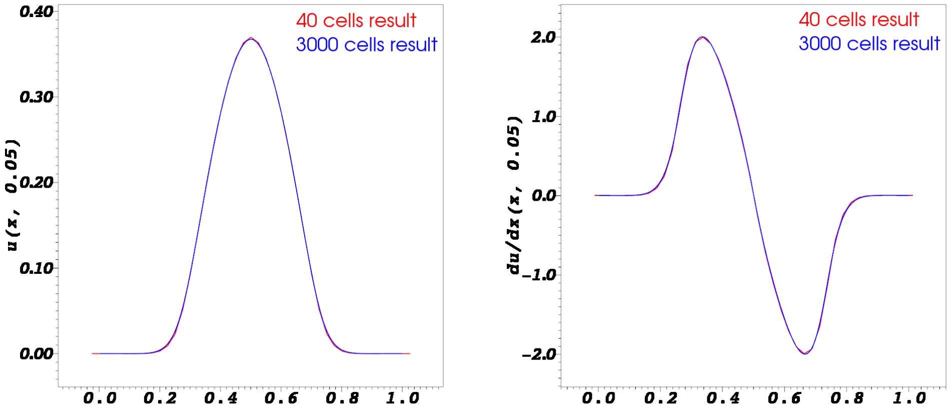

FIG. 5.4. *Solution for u (on the left side) and $\frac{\partial u}{\partial x}$ (on the right side) computed on 40 and 3000 cells for Test Problem 2.*

	m errors				u errors					
cells	L^∞	rate	L^1	rate	L^∞	rate	L^1	rate	mass error	rate
40	2.75E-1		1.74E-2		3.43E-3		9.40E-4		3.74E-7	
80	6.66E-2	2.05	4.28E-3	2.02	7.93E-4	2.11	2.51E-4	1.90	1.47E-7	1.34
160	1.64E-2	2.01	1.04E-3	2.04	1.98E-4	1.99	6.42E-5	1.97	6.29E-8	1.22
320	4.13E-3	1.99	2.57E-4	2.01	4.96E-5	2.00	1.60E-5	2.00	3.03E-8	1.05
640	1.01E-3	2.03	6.34E-5	2.02	1.25E-5	1.97	3.90E-6	2.04	1.52E-8	0.99
1280	2.41E-4	2.06	1.47E-5	2.10	3.43E-6	1.87	8.94E-7	2.12	7.66E-9	0.98

TABLE 5.1. *L^∞ and L^1 errors, differences between initial and final mass, and convergence rates with respect to a reference solution computed on 3000 cells for Test Problem 2.*

of 112 500 time steps for each equation. The machine we use is an AMD Opteron 6174, 2.2 Ghz with 48 total cores. The execution times and the scaling result are displayed on Figure 5.5. We observe that linear scaling is achieved when we have at least 500 cells per processor. Since the parallelism is in space and not in time, our code is faster for cases when the ratio between cells in space versus steps in time is bigger i.e., for smaller values of σ.

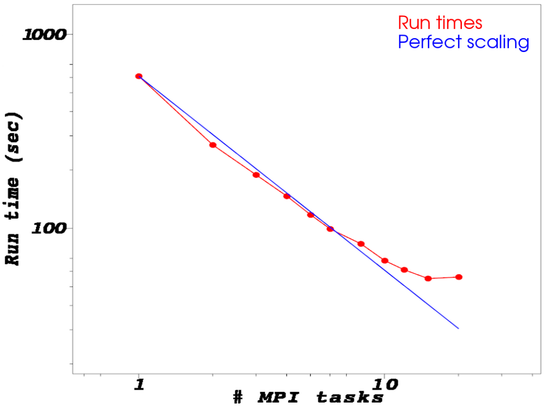

FIG. 5.5. *Strong scaling test on 6000 cells for Test Problem 2.*

6. Related work

In this section we describe some already existing algorithms related to the MFG equations (1.1), (1.2).

In [3], Achdou and Capuzzo-Dolcetta propose implicit finite difference methods for the stationary case, the time-dependent case where both MFG equations progress forward in time, and the case of (1.1), (1.2). The authors present detailed proofs of existence and uniqueness for the discrete problems, and provide bounds on the solutions. The paper contains results of numerical simulations for 2-dimensional test cases where both equations go forward in time. The simulations make use of a long time approximation strategy of the stationary problem. The tests confirm that the used approach is robust when $\sigma \to 0$, and the results suggest linear convergence, i.e., the scheme is only first order accurate.

In [1], Achdou, Camilli, and Capuzzo-Dolcetta study the mean field planning problem (MFGP), which puts an initial condition on $m(x,0)$ instead of the one on $u(x,0)$, and the penalized mean field planning problem (MFGPP), which is in the same form as (1.1), (1.2). The authors present semi-implicit finite difference schemes and prove existence and uniqueness of the solution by exploiting a connection between the discrete formulations and a minimization problem. Results for the MFGP discrete equations are obtained by solving the MFGPP discrete equations and passing to the limit of a penalization parameter. The forward-backward MFGPP finite difference scheme is solved by a Newton method. The presented numerical results show correct behavior for small σ and first order convergence. The Newton method convergence is slower for smaller values of σ.

In [7], Gueant examines the MFG equations (1.1), (1.2) for the special case of $H(u_x) = u_x^2/2$. The author uses a change of variables which produces two coupled heat equations with source terms. Under some assumptions on $f(x,m)$, existence and uniqueness of weak solutions for the new system are proved. Each equation is approximated in space-time, so that $m_T(x), u_0(x)$ appear as boundary conditions, by implicit finite difference schemes. The author proves existence and uniqueness for both schemes. The discrete equations are solved recursively until fixed point is reached, a Newton method is applied inside each step. The presented numerical results show first order convergence and increasing number of Newton iterations for smaller values of σ.

Alternative to these methods and to our approach here can be found in [4, 12] but these methods also seem to be first order accurate. Convergence results for the schemes in [3, 1] are provided by their authors in [2].

7. Conclusion

We have presented a parallel fixed point iteration algorithm that combines a second order scheme for the forward Hamilton–Jacobi Equation (1.1), and a second order scheme for the backward convection-diffusion Equation (1.2). The second order accuracy of the method is confirmed numerically, and our numerical results agree with the already existing data in the field. Both schemes are explicit, which means that in a parabolic regime we have $\Delta t = O(\Delta x^2)$. This results in a high number of time steps, however the schemes' simplicity and the method's parallel ability allow us to use highly refined meshes. We have also eliminated the memory problems arising from the combination of explicit time stepping and forward-backward coupling of the equations. The main drawback of our method is that the structure of the original system is somewhat lost in the discrete method, making it hard to prove uniqueness, stability and convergence of the fixed point iteration. However, we achieve (numerically) second order accuracy which makes the method more cost efficient.

This work can be extended by introducing 2D algorithms that use the same central schemes approach. This will result in more computations inside a single time step, hence it will exploit better the parallel abilities of our numerical method. We expect to achieve similar run times as in 1D, since the 2D methods will do the same number of time steps while performing more computations per cell.

Appendix A.

A.1. Convergence properties. Here we justify our expectations of second order accuracy in L^∞ and the choices of specific limiters (2.4), (3.1), (3.2). For the time being we refer to $\Delta t_{hj}, \Delta t_{cd}$ just as Δt since this argument doesn't focus on the differences between the two. In order to produce global truncation errors (GTE) of at most $O(\Delta x^2)$ for both hyperbolic and a parabolic regimes, we need local truncation errors (LTE) of sizes at most $O(\Delta x^4), O(\Delta x^2 \Delta t), O(\Delta x \Delta t^2)$, or $O(\Delta t^3)$.

First we discuss why the diffusion terms are ignored in the half-time equations (2.10), (3.6) (skipping the diffusion term in (2.10) follows from skipping it in (2.8)). Since the half-time values are already scaled in a sense by $O(\Delta t)$ in (2.11), (3.7), and the diffusion terms are scaled by an additional $O(\Delta t)$ in (2.10), (3.6), then these terms' influence in the final LTE is at most $O(\sigma \Delta t^2)$ which results in GTE of at most $O(\sigma \Delta t)$. If we are in a parabolic regime, then $O(\Delta t) = O(\Delta x^2), \sigma = O(1)$ and ignoring the diffusion terms doesn't affect the desired accuracy. If we are in a hyperbolic regime, then $\sigma = O(\Delta x)$ and the diffusion terms affect the LTE as $O(\Delta t^2 \Delta x)$, hence they can be ignored again.

A.2. Second order accuracy of the Hamilton–Jacobi scheme. Now we consider FHJ Equation (1.1). A centered difference for $u_t(x, t_{n+\frac{1}{2}})$ gives us the midpoint method:

$$u(x,t_{n+1}) = u(x,t_n) + \Delta t \frac{\partial}{\partial t} u(x, t_{n+\frac{1}{2}}) + O(\Delta t^3), \Rightarrow$$

$$u(x,t_{n+1}) = u(x,t_n) + \Delta t \Bigg[-H\left(\frac{\partial}{\partial x} u(x,t_{n+\frac{1}{2}})\right) + f\left(x, m(x,t_{n+\frac{1}{2}})\right) + \sigma \frac{\partial^2}{\partial^2 x} u(x, t_{n+\frac{1}{2}}) \Bigg] + O(\Delta t^3). \tag{A.1}$$

Suppose all values u_j^n are exact for every x_j at a fixed time t_n. We can also interpolate $\hat{m}$ values at times $t_n, t_{n+\frac{1}{2}}$ up to at least $O(\Delta x^2)$ accuracy with Equation (4.1) as explained in Subsection 4.1. Comparing (2.11) and (A.1), we see that an acceptable LTE are achieved if

$$(\hat{u}_{xx})^n_{j+\frac{1}{2}} = \frac{\partial^2}{\partial^2 x} u(x_{j+\frac{1}{2}}, t_n) + O(\Delta x^2), \tag{A.2}$$

$$\frac{1}{2}\left(u_j^n + u_{j+1}^n\right) - \frac{(\Delta x)^2}{8} (\hat{u}_{xx})^n_{j+\frac{1}{2}} = u(x_{j+\frac{1}{2}}, t_n) + O(\Delta x^4), \tag{A.3}$$

$$(\hat{u}_x)^{n+\frac{1}{2}}_{j+\frac{1}{2}} = \frac{\partial}{\partial x} u(x_{j+\frac{1}{2}}, t_{n+\frac{1}{2}}) + O(\Delta x^2) \quad [\text{or} \quad O(\Delta t \Delta x)], \tag{A.4}$$

$$m^{n+\frac{1}{2}}_{j+\frac{1}{2}} = m(x_{j+\frac{1}{2}}, t_{n+\frac{1}{2}}) + O(\Delta x^2), \tag{A.5}$$

$$\sigma\frac{(\hat{u}_x)^n_{j+\frac{3}{2}}-(\hat{u}_x)^n_{j-\frac{1}{2}}}{2\Delta x}=\sigma\frac{\partial^2}{\partial^2 x}u(x_{j+\frac{1}{2}},t_{n+\frac{1}{2}})+O(\Delta x^2). \tag{A.6}$$

Condition (A.2) is satisfied by the limiter (2.4) with $\theta=2$, or the UNO limiter. Note that condition (A.2) is not satisfied for the usual minmod limiter ((2.4) with $\theta=1$) at local extrema. Then (A.3) comes from taking Taylor expansions of $u(x_j,t_n),u(x_{j+1},t_n)$ at $x_{j+\frac{1}{2}}$. Condition (A.5) is guaranteed by our time interpolation. In order to verify (A.4), we need to look at the half-time step Equation (2.10). Then we see that condition (A.4) holds if:

$$(\hat{u}_x)^n_{j+\frac{1}{2}}=\frac{\partial}{\partial x}u(x_{j+\frac{1}{2}},t_n)+O(\Delta x^2), \tag{A.7}$$

$$m^n_{j+\frac{1}{2}}=m(x_{j+\frac{1}{2}},t_n)+O(\Delta x), \tag{A.8}$$

$$(\hat{m}_x)^n_{j+\frac{1}{2}}=\frac{\partial}{\partial x}m(x_{j+\frac{1}{2}},t_n)+O(\Delta x). \tag{A.9}$$

(A.7) is clearly satisfied by (2.3), (A.8) is true by the time interpolation properties, and (A.9) follows from (A.8) and the properties of the minmod limiter (3.2). Finally, for the left side of (A.6) we have:

$$\begin{aligned}\sigma\frac{(\hat{u}_x)^n_{j+\frac{3}{2}}-(\hat{u}_x)^n_{j-\frac{1}{2}}}{2\Delta x}&=\sigma\left(\frac{\partial^2}{\partial^2 x}u(x_{j+\frac{1}{2}},t_n)+O(\Delta x^2)\right)\\&=\sigma\left(\frac{\partial^2}{\partial^2 x}u(x_{j+\frac{1}{2}},t_{n+\frac{1}{2}})+O(\Delta t)+O(\Delta x^2)\right)\\&=\sigma\frac{\partial^2}{\partial^2 x}u(x_{j+\frac{1}{2}},t_{n+\frac{1}{2}})+O(\Delta x^2),\end{aligned}$$

where in the last equality we used that $\sigma O(\Delta t)=O(\Delta x^2)$ by the CFL condition (2.12). Thus we have verified all conditions (A.2)–(A.9), which means that the central difference scheme (2.11) provides second order accuracy in L^∞.

Remark A.1. If we consider the simplified scheme (2.13), we immediately see, in the spirit of Equation (A.3), that not considering second derivatives of $\hat{u}$ results in approximating $u(x_{j+\frac{1}{2}},t_n)$ by the term $\frac{1}{2}\left(u^n_j+u^n_{j+1}\right)$. By standard Taylor expansion at $x_{j+\frac{1}{2}}$ we have

$$\frac{1}{2}\left(u^n_j+u^n_{j+1}\right)=u(x_{j+\frac{1}{2}},t_n)+O(\Delta x^2),$$

hence this term gives us LTE of size $O(\Delta x^2)$. This means that in a hyperbolic regime we obtain GTE of size $O(\Delta x)$, and in a parabolic regime the scheme doesn't converge since the GTE is $O(1)$.

A.3. Second order accuracy of the convection-diffusion scheme. Now we consider BCD Equation (1.2), a centered difference for $m_t(x,t_{k-\frac{1}{2}})$ gives us the midpoint method:

$$m(x,t_{k-1})=m(x,t_k)-\Delta t\frac{\partial}{\partial t}m(x,t_{k-\frac{1}{2}})+O(\Delta t^3),\Rightarrow$$

$$m(x,t_{k-1})=m(x,t_k)+\Delta t\left[\frac{\partial}{\partial x}\left(H'\left(\frac{\partial}{\partial x}u(x,t_{k-\frac{1}{2}})\right)m(x,t_{k-\frac{1}{2}})\right)\right.$$
$$\left.+\sigma\frac{\partial^2}{\partial^2 x}m(x,t_{k-\frac{1}{2}})\right]+O(\Delta t^3). \qquad \text{(A.10)}$$

Suppose all values $m^k_{j+\frac{1}{2}}$ are exact for every $x_{j+\frac{1}{2}}$ at a fixed time t_k. We can also interpolate $\hat{u}$ values at times $t_k,t_{k-\frac{1}{2}}$ up to at least $O(\Delta x^2)$ accuracy with Equation (4.2) as explained in Subsection 4.1. Comparing (3.7) and (A.10), we see that acceptable LTE are achieved if

$$\frac{1}{\Delta x}\left((\hat{m}_X)^k_{j+\frac{3}{2}}-(\hat{m}_X)^k_{j+\frac{1}{2}}\right)=\frac{\partial^2}{\partial^2 x}m(x_{j+1},t_k)+O(\Delta x^2), \qquad \text{(A.11)}$$

$$\frac{1}{2}\left(m^k_{j+\frac{1}{2}}+m^k_{j+\frac{3}{2}}\right)-\frac{\Delta x}{8}\left((\hat{m}_X)^k_{j+\frac{3}{2}}-(\hat{m}_X)^k_{j+\frac{1}{2}}\right)=m(x_{j+1},t_k)+O(\Delta x^4), \qquad \text{(A.12)}$$

$$(\hat{u}_x)^{k-\frac{1}{2}}_{j+\frac{1}{2}}=\frac{\partial}{\partial x}u(x_{j+\frac{1}{2}},t_{k-\frac{1}{2}})+O(\Delta x^2), \qquad \text{(A.13)}$$

$$m^{k-\frac{1}{2}}_{j+\frac{1}{2}}=m(x_{j+\frac{1}{2}},t_{k-\frac{1}{2}})+O(\Delta x^2) \quad [\text{or} \quad O(\Delta t\Delta x)], \qquad \text{(A.14)}$$

$$\sigma\frac{m^k_{j+\frac{5}{2}}-m^k_{j+\frac{3}{2}}-m^k_{j+\frac{1}{2}}+m^k_{j-\frac{1}{2}}}{2\Delta x^2}=$$
$$\sigma\frac{\partial^2}{\partial^2 x}m(x_{j+1},t_{k-\frac{1}{2}})+O(\Delta x^2) \quad [\text{or} \quad O(\Delta t\Delta x)]. \qquad \text{(A.15)}$$

Condition (A.11) is satisfied by the UNO limiter (3.1). The choice of UNO is important, since condition (A.11) is not true for the minmod limiter (3.2). Then (A.12) comes from taking Taylor expansions of $m(x_{j+\frac{1}{2}},t_k),m(x_{j+\frac{3}{2}},t_k)$ at x_{j+1}. The left side of condition (A.13) is computed by doing time interpolations, followed by applying (2.3) on them, hence the condition is true by the time interpolation properties and (A.7). In order to verify (A.14), we need to look at the half-time step Equation (3.6). Then we see that condition (A.14) holds if:

$$(\hat{u}_x)^k_{j+\frac{1}{2}}=\frac{\partial}{\partial x}u(x_{j+\frac{1}{2}},t_k)+O(\Delta x), \qquad \text{(A.16)}$$

$$(\hat{u}_{xx})^k_{j+\frac{1}{2}}=\frac{\partial^2}{\partial^2 x}u(x_{j+\frac{1}{2}},t_k)+O(\Delta x), \qquad \text{(A.17)}$$

$$(\hat{m}_x)^k_{j+\frac{1}{2}}=\frac{\partial}{\partial x}m(x_{j+\frac{1}{2}},t_k)+O(\Delta x). \qquad \text{(A.18)}$$

Condition (A.16) follows immediately from (A.13), and (A.17) follows from time interpolation properties and (A.2). Condition (A.18) is guaranteed by the minmod limiter (3.2). Finally, the argument about condition (A.15) is the same as the one for condition (A.6). We have verified all conditions (A.11)–(A.18) hence the central difference scheme

(3.7) provides second GTE. The second order accuracy of our scheme was observed in all test examples with smooth solutions.

Thus we have verified all conditions (A.11)–(A.18), hence the central difference scheme (3.7) provides second order accuracy in L^∞.

Remark A.2. If we consider the simplified scheme (3.9), we immediately see, in the spirit of Equation (A.12), that not considering first derivatives of $\hat{m}$ results in approximating $m(x_{j+1},t_k)$ by the term $\frac{1}{2}\left(m^k_{j+\frac{1}{2}}+m^k_{j+\frac{3}{2}}\right)$. By standard Taylor expansion at x_{j+1} we have

$$\frac{1}{2}\left(m^k_{j+\frac{1}{2}}+m^k_{j+\frac{3}{2}}\right)=m(x_{j+1},t_k)+O(\Delta x^2),$$

hence this term gives us LTE of size $O(\Delta x^2)$. This means that in a hyperbolic regime we obtain GTE of size $O(\Delta x)$, and in a parabolic regime the GTE is $O(1)$ and we verified numerically that the scheme doesn't converge in this case.

REFERENCES

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

[2] Y. Achdou, F. Camilli, and I. Capuzzo-Dolcetta, *Mean field games: Convergence of a Finite Difference Method*, SIAM J. Numer. Anal., 51(5), 2585–2612, 2013.

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

[4] Y. Achdou, J.-M. Lasry, P.-L. Lions, and B. Moll, *Heterogeneous agent models in continuous time*, Draft.

[5] L. Corrias, M. Falcone, and R. Natalini, *Numerical schemes for conservation laws via Hamilton–Jacobi equations*, Math. Comp., 64, 555–580, 1995.

[6] M.G. Crandall and P.-L. Lions, *Two approximations of solutions of Hamilton-Jacobi equations*, Math. Comp., 43, 1–19, 1984.

[7] O. Gueant, *Mean field games equations with quadratic Hamiltonian: a specific approach*, Math. Models Meth. Appl. Sci., 22(9), 2012.

[8] A. Harten and S. Osher, *Uniformly high-order accurate nonoscillatory schemes I*, SIAM J. Numer. Anal., 24(2), 279–309, 1987.

[9] C. Hu and C.-W. Shu, *A discontinuous Galerkin finite element method for Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 21(2), 669–690, 1999.

[10] S. Jin and Z. Xin, *Numerical passage from systems of conservation laws to Hamilton–Jacobi equations, relaxation schemes*, SIAM J. Numer. Anal., 35(6), 2385–2404, 1998.

[11] A. Kurganov and E. Tadmor, *New high-resolution semi-discrete central schemes for Hamilton–Jacobi equations*, J. Comput. Phys., 160(2), 720–742, 2000.

[12] A. Lachapelle, J. Salomon, and G. Turinici, *A monotonic algorithm for mean field games model in economics*, Technical report, CEREMADE, U. Paris Dauphine, 2009.

[13] J.-M. Lasry and P.-L. Lions, *Mean Field Games*, Jpn. J. Math., 2(1), 229–260, 2007.

[14] P. Lax, *Weak solutions of nonlinear hyperbolic equations and their numerical computation*, Commun. Pure Appl. Math., 7, 159–193, 1954.

[15] B. van Leer, *Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method*, J. Comput. Phys., 32(1), 101–136, 1979.

[16] P.-L. Lions and P. Souganidis, *Convergence of MUSCL and filtered schemes for scalar conservation laws and Hamilton–Jacobi equations*, Numer. Math., 69(4), 441–470, 1995.

[17] C.-T. Lin and E. Tadmor, *High-Resolution non-oscillatory central schemes for Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 21(6), 2163–2186, 1999.

[18] C.-T. Lin and E. Tadmor, *L^1-stability and error estimates for approximate Hamilton–Jacobi solutions*, Numer. Math., 87(4), 701–735, 2001.

[19] H. Nessyahu and E. Tadmor, *Nonoscillatory central differencing for hyperbolic conservation laws*, J. Comput. Phys., 87(2), 408–463, 1990.

[20] Bojan Popov and Ognian Trifonov, *One-sided stability and convergence of the Nessyahu–Tadmor scheme*, Numer. Math., 104(4), 539–559, 2006.

[21] S. Osher and C.-W. Shu, *High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 28(4), 907–922, 1991.

[22] S. Osher and E. Tadmor, *On the convergence of difference approximations to scalar conservation laws*, Math. Comp., 50(181), 19–51, 1988.