Journal of Scientific Computing (2021) 88:10
https://doi.org/10.1007/s10915-021-01520-0

Artificial Viscosity Joint Spacetime Multigrid Method for Hamilton–Jacobi–Bellman and Kolmogorov–Fokker–Planck System Arising from Mean Field Games

Yangang Chen[1] · **Justin W. L. Wan**[2]

Received: 12 April 2019 / Revised: 5 April 2021 / Accepted: 7 May 2021 / Published online: 26 May 2021

Abstract
In this paper, we study numerical solutions for the Hamilton-Jacobi-Bellman (HJB) and Kolmogorov–Fokker–Planck (KFP) equations arising from mean field games. In order to solve the nonlinear discretized systems efficiently, we propose a multigrid method. Our proposed multigrid method is developed on the joint spacetime and is a full approximation scheme (FAS). We consider hybrid full-semi coarsening and kernel preserving biased restriction to address the anisotropy in time and convections in space. The main novelty of this paper is that we propose adding artificial viscosity to the direct discretization coarse grid operators, such that the coarse grid error estimations are more accurate. We use Fourier analysis to illustrate the efficiency of our proposed multigrid method. Numerical experiments show that the convergence rate of the proposed multigrid method is mesh-independent and faster than the existing methods in the literature.

Keywords Mean field games · Hamilton–Jacobi–Bellman equation · Kolmogorov–Fokker–Planck equation · Spacetime methods · Multigrid methods · Full approximation scheme · Artificial viscosity

1 Introduction

Mean field game theory, first proposed by [31–33], studies the competitions among infinite numbers of players, where individual players have negligible influence to the system, and respond to each other in a statistical sense. There are numerous applications of mean field games, including, but not limited to, micro or macro economics, sociology, engineering, urban planning, etc [15,17,24].

✉ Yangang Chen
y493chen@uwaterloo.ca

[1] Department of Applied Mathematics, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada

[2] David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada

In [24,33], mean field games are formulated into a system of nonlinear partial differential equations (PDEs) that contains two equations. One equation is a backward Hamilton–Jacobi–Bellman (HJB) equation for the optimal value function of the players. The other equation is a forward Kolmogorov–Fokker–Planck (KFP) equation for the distribution (or density function) of the players' states. Numerical methods for the HJB/KFP system have been studied extensively in [1–7,12,13,30]. The major challenge of numerical solution for the HJB/KFP system is the nonlinearity and the size of the discretized system, which requires effective and fast solvers.

However, very few literature, such as [1,5,6,12], has proposed effective and fast solvers for the nonlinear discretized HJB/KFP system. These solvers have two common features. One is that they are all spacetime methods. The reason to consider spacetime methods is that the HJB equation is backward and the KFP equation is forward, which makes it impossible to solve the system using the conventional forward timestepping. The other common feature is that all of these methods are outer-inner linearization-based multigrid methods. More specifically, in order to solve the nonlinear discretized problems, [1,5] propose spacetime Newton's iterations; [6] proposes ALG2 (i.e. Douglas-Rachford) iterations; [12] proposes primal-dual (i.e. Chambolle–Pock) iterations. Each nonlinear iteration requires solving a large spacetime linear system. In order to solve the linear system, they typically use BiCGStab iterations, and use multigrid cycles as preconditioners for each BiCGStab iteration. The common issue of these approaches is that they all involve many layers of iterations: outer nonlinear iterations, and inner multigrid iterations (or inner BiCGStab iterations with multigrid preconditioners) for each linearized system. As a result, the total number of iterations, which is approximately the product of the iteration counts of the outer and inner iterations, can be large. To be more concrete, the approach in [6] typically requires more than 1000 outer ALG2 iterations and 7 inner multigrid-preconditioned CG iterations per ALG2 iteration, i.e. more than 7000 iterations in total; the approach in [12] typically requires more than 20 outer Chambolle–Pock iterations and 4 inner multigrid-preconditioned BiCGStab iterations per outer iteration, i.e. more than 80 iterations in total.

To address the issues of these existing multigrid methods, in this paper, we propose another spacetime multigrid solver for the nonlinear discretized HJB/KFP system. In particular, our multigrid method is a full approximation scheme (FAS) [26,37]. Unlike the other multigrid methods that are applied iteratively on the inner linearized systems nested in outer nonlinear iterations, our FAS is directly applied on the nonlinear system itself and thus involves only one layer of iterations.

In our FAS multigrid method, we consider a hybrid of full and semi coarsenings to address the anisotropy arising from the time direction [5,27]. We note that convections in the HJB/KFP system pose a major challenge for multigrid methods, as standard multigrid methods are ineffective for convection-diffusion equations [37]. In order to address this difficulty, we consider adapting a stable and efficient multigrid method proposed in [9], which uses a type of biased restriction, called kernel preserving restriction, together with Petrov-Galerkin coarse grid operators. However, the multigrid method in [9] is designed for linear equations. It cannot be directly applied to the FAS framework, because the nonlinearity of FAS is incompatible with Petrov-Galerkin operators. Direct discretization, as the alternative of Petrov-Galerkin operators, is compatible with the FAS framework. However, it is well-known that when the convections are non-negligible, the convergence rate under direct discretization is no better than 0.5 [11].

Our approach, which is the main novel component of this paper, is to add artificial viscosity to the direct discretization coarse grid operators. Adding artificial viscosity allows us to design an effective FAS solver under the direct discretization operators when the convections are non-negligible. Our Local Fourier Analysis proves that adding artificial viscosity reduces the asymptotic convergence factor and the error reduction factor. We remark that although adding artificial viscosity is a well-known idea for stabilizing numerical solutions for convection-diffusion equations [19,29,34], it has not been used for the purpose of improving coarse grid correction for multigrid methods.

Significantly, our numerical simulations illustrate that our FAS multigrid method with artificial viscosity yields mesh-independent convergence rates for the HJB/KFP system. In particular, our approach typically converges in less than 10 iterations *in total*, which is faster than the outer-inner linearization-based multigrid methods in [1,5,6,12].

To the best of our knowledge, this paper is the first proposal of multigrid method with the following two features: it is an FAS directly designed for the nonlinear discretized HJB/KFP system (as opposed to the linearized version); and in particular, it adds artificial viscosity on the direct discretization coarse grid operator. These two features are critical for our proposed multigrid method to converge faster than the other methods.

To illustrate our proposed multigrid method, we first review the HJB/KFP PDE system arising from mean field games in Sect. 2. Then we describe finite difference discretization in Sect. 3. Section 4 introduces joint spacetime methods for solving the nonlinear discretized system. Section 5 describes our proposed spacetime FAS multigrid methods with artificial viscosity. Section 6 uses Local Fourier Analysis to demonstrate the efficiency of the proposed multigrid method. Section 7 presents numerical results. Section 8 is the conclusion.

2 Mean Field Games

In this section, we review the mathematical formulation for mean field games. Consider a differential game with N players, where each player sets its control to maximize its value function. For example, in a competitive smartphone market, each company (player) sets its smartphone price (control) to maximize its long-term profit (value function). Such games can be formulated by a PDE system with N HJB equations, where each equation solves for the optimal value function of each individual player. Since each player tries to win the competition by adjusting its own control in response to the other players' controls, the HJB equations are coupled. When N is large, the PDE system becomes extremely complicated. Fortunately, when $N \to \infty$, the model can be simplified. Each single player's impact to the entire system is negligible. Also, since it is difficult to keep track of every single opponent's control, players respond to each other's control in the statistical sense. Individual players are only distinguished and labeled by a "state variable". For example, in the smartphone market with many players, identifying every individual company becomes intractable. Instead, we only differentiate the companies by their "capacities" (state variable). Such kind of games can be reduced to a less complicated model, called "mean field games"; see [15,24,31–33]. These papers have shown that the model of mean field games yields a good approximation of the original N-HJB model when N is large. We refer readers to [24] for an introduction to the topic.

2.1 Mean Field Games with Local Coupling

We first review the mathematical formulation of mean field games with local coupling introduced in [24,33]. Let Ω be a space domain in $\mathbb{R}^d$. Let $\Omega \times [0, T]$ be a spacetime domain. Let $\mathbf{x} \in \Omega$ be the players' d-dimensional state variable, and let $t \in [0, T]$ be the time. Let

$u : \Omega \times [0, T] \to \mathbb{R}$ be the optimal value function of the players, and let $m : \Omega \times [0, T] \to \mathbb{R}$ be the distribution (or density function) of the players' state variable. Mean field games can be formulated into a system of PDEs that contains two equations. One equation is the Hamilton–Jacobi–Bellman (HJB) equation for the value function u:

$$\begin{aligned} -u_t - \sigma \Delta u + \max_{\mathbf{c}} [\mathbf{c} \cdot \nabla u - L(\mathbf{c})] + \rho u - g(m) = 0, &\ \text{ in } \Omega \times [0, T), \\ u(\mathbf{x}, T) = u_T(\mathbf{x}), &\ \text{ in } \Omega, \end{aligned} \tag{1}$$

where $\mathbf{c} : \Omega \times [0, T] \to \mathbb{R}^d$ is a d-dimensional control parameter, ρ is the discount factor, $g(m) = g(m(\mathbf{x}, t))$ is the local cost function, $L(\mathbf{c}) = L(\mathbf{c}(\mathbf{x}, t), \mathbf{x})$ is the Lagrangian, and $\mathbf{c} \cdot \nabla u - L(\mathbf{c})$ is the Hamiltonian. We assume that the Hamiltonian is concave in $\mathbf{c}$ and the maximum can be achieved at the corresponding stationary point[1]. Noticeably, the HJB equation is backward from the terminal time $t = T$ to the initial time $t = 0$. Hence, the HJB equation has a terminal condition $u(\mathbf{x}, T) = u_T(\mathbf{x})$ rather than an initial condition.

The other equation in the PDE system is the forward Kolmogorov–Fokker–Planck (KFP) equation for the distribution m:

$$\begin{aligned} m_t - \sigma \Delta m - \nabla \cdot (\mathbf{c}^* m) = 0, &\ \text{ in } \Omega \times (0, T], \\ m(\mathbf{x}, 0) = m_0(\mathbf{x}), &\ \text{ in } \Omega, \end{aligned} \tag{2}$$

where

$$\mathbf{c}^* \equiv \arg\max_{\mathbf{c}} [\mathbf{c} \cdot \nabla u - L(\mathbf{c})] \tag{3}$$

is the optimal control of the HJB equation (1). For simplicity, unless specified, we assume periodic boundary conditions for both (1) and (2). We refer readers to [1,33] for a discussion on the well posedness of the problem (1–3).

A unique feature of the HJB/KFP system is that, while the HJB equation (1) is backward from $t = T$ to $t = 0$, the KFP equation (2) is forward from $t = 0$ to $t = T$. In addition, these two equations are nonlinear and coupled with each other. More specifically, the HJB equation is nonlinear, since the optimal control $\mathbf{c}^*$ that maximizes $\mathbf{c} \cdot \nabla u - L(\mathbf{c})$ is a functional of u. The two equations are coupled, because in the HJB equation, the cost function $g(m)$ depends on the solution of the KFP equation m; and in the KFP equation, the optimal control $\mathbf{c}^*$ depends on the solution of the HJB equation u. As a result, the entire HJB/KFP system is nonlinear.

2.2 Bertrand Mean Field Games with Nonlocal Coupling

To illustrate a concrete application of mean field games, we review the Bertrand mean field games proposed in [15]. Consider again a competitive market with a large number of companies. Each company, distinguished by its capacity $x \in [0, \infty)$, evolves over time $t \in [0, T]$. Let $u(x, t)$ be the optimal expected profit (value function) of each company over the period $[t, T]$. Let $p(x, t)$ be the price of each company's product (control). Let $m(x, t)$ be the distribution of the companies. The HJB equation reads

$$\begin{aligned} -u_t - \sigma u_{xx} - \max_{p \geq 0} \{D(p, \bar{p}(m), \eta(m))(p - u_x)\} + \rho u = 0, \\ u(x, T) = 0, \quad u(0, t) = 0, \quad u_x(\infty, t) = 0, \end{aligned} \tag{4}$$

where ρ is the interest rate of the bank, and σ is the randomness of the demand in the market. In this HJB equation, the optimal control of each company $p^*(x, t)$, the total number of

[1] In some applications, the Hamiltonian is convex in $\mathbf{c}$. Then "max" in (1) is replaced by "min".

companies $\eta : t \to \eta(m)(t)$, the market average price $\bar{p} : t \to \bar{p}(m)(t)$ and the demand function $D : (x, t) \to D(p, \bar{p}(m), \eta(m))(x, t)$ are given by

$$
\begin{aligned}
p^* &= \underset{p \geq 0}{\arg\max} \left\{ D(p, \bar{p}, \eta)(p - u_x) \right\}, \\
\eta(t) &= \int_0^\infty m(x, t) dx, \quad \bar{p}(t) = \frac{1}{\eta(t)} \int_0^\infty p^*(x, t) m(x, t) dx, \\
D(x, t) &= s \left(\frac{1}{1 + \epsilon \eta(t)} - p(x, t) + \frac{\epsilon \eta(t)}{1 + \epsilon \eta(t)} \bar{p}(t) \right),
\end{aligned}
\tag{5}
$$

where ϵ and s are constants. The corresponding KFP equation reads

$$
\begin{aligned}
m_t - \sigma m_{xx} - \left[D(p^*, \bar{p}(m), \eta(m)) m \right]_x &= 0, \\
m(x, 0) = m_0(x), \quad m_x(0, t) = 0, \quad m(\infty, t) &= 0,
\end{aligned}
\tag{6}
$$

where $m_0(x)$ is the initial distribution (e.g. $m_0(x) = 1 - B(x; 2, 4)$, where B is the cumulative beta distribution function). We refer readers to [23] for the well posedness of the problem (4–6). We note that the coupling between the HJB equation (4) and the KFP equation (6) is more complicated than Equations (1–3). More specifically, (1–3) only involves local couplings, i.e. the convection coefficient is the local optimal control $\mathbf{c}^*$ and the cost function g is also local. However, in (4–6), the convection coefficient D is a functional of both p^* and m in the form of nonlocal integrals. To simplify the presentation, we will focus on Eqs. (1–3) in this paper and revisit Eqs. (4–6) in Sect. 7.

3 Finite Difference Discretization

Consider the numerical solution of (1–3). Without loss of generality, we assume that the space-time is (2+1)-dimensional. Define an $N_x \times N_y \times N_t$ spacetime mesh $\{\mathbf{x}_{i,j}^n = (x_i, y_j, t_n) \,|\, i = 1, ..., N_x;\ j = 1, ..., N_y;\ n = 0, ..., N_t\}$. Denote the corresponding mesh sizes as Δx, Δy, Δt. Our goal is to solve the set of the unknowns $\{u_{i,j}^n \equiv u(\mathbf{x}_{i,j}^n),\ m_{i,j}^n \equiv m(\mathbf{x}_{i,j}^n)\}$.

For the HJB equation (1), we use implicit timestepping for u_t, central differencing for Δu, and upwinding discretization for $\mathbf{c} \cdot \nabla u = c_1 u_x + c_2 u_y$. Define $c^+ = \max(c, 0)$, and $c^- = \min(c, 0)$. Then the discretization of HJB equation reads

$$
\begin{aligned}
&-\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \sigma \frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{\Delta x^2} - \sigma \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{\Delta y^2} \\
&+ \max_{\mathbf{c}_{i,j}^n} \Bigg[((c_1)_{i,j}^n)^+ \frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta x} + ((c_1)_{i,j}^n)^- \frac{u_{i+1,j}^n - u_{i,j}^n}{\Delta x} \\
&\qquad + ((c_2)_{i,j}^n)^+ \frac{u_{i,j}^n - u_{i,j-1}^n}{\Delta y} + ((c_2)_{i,j}^n)^- \frac{u_{i,j+1}^n - u_{i,j}^n}{\Delta y} - L(\mathbf{c}_{i,j}^n) \Bigg] \\
&\qquad\qquad + \rho u_{i,j}^n - g(m_{i,j}^n) = 0.
\end{aligned}
\tag{7}
$$

We note that since the HJB equation is backward in time, the implicit timestepping is given by $(u_{i,j}^{n+1} - u_{i,j}^n)/\Delta t$ rather than the conventional $(u_{i,j}^n - u_{i,j}^{n-1})/\Delta t$.

For the KFP equation (2), we also use implicit timestepping for m_t. Notice that the KFP equation is written into a conservation form [36]. A standard discretization for conservation laws is to use a numerical flux for $f = \mathbf{c}^* m$. For instance, we can choose the Engquist-Osher flux [36]:

$$\begin{aligned} \hat{f}^n_{i+1/2,j} &= ((c_1^*)^n_{i,j})^- m^n_{i,j} + ((c_1^*)^n_{i+1,j})^+ m^n_{i+1,j}, \\ \hat{f}^n_{i,j+1/2} &= ((c_2^*)^n_{i,j})^- m^n_{i,j} + ((c_2^*)^n_{i,j+1})^+ m^n_{i,j+1}. \end{aligned} \tag{8}$$

This is essentially an upwind flux with an additional consideration for rarefactions and shocks. As a result, the finite difference discretization for the KFP equation is given by

$$\begin{aligned} \frac{m^n_{i,j} - m^{n-1}_{i,j}}{\Delta t} - \sigma \frac{m^n_{i+1,j} - 2m^n_{i,j} + m^n_{i-1,j}}{\Delta x^2} - \sigma \frac{m^n_{i,j+1} - 2m^n_{i,j} + m^n_{i,j-1}}{\Delta y^2} \\ - \frac{\hat{f}^n_{i+1/2,j} - \hat{f}^n_{i-1/2,j}}{\Delta x} - \frac{\hat{f}^n_{i,j+1/2} - \hat{f}^n_{i,j-1/2}}{\Delta y} = 0. \end{aligned} \tag{9}$$

We note that the KFP equation (2) can also be discretized by other methods; see [8,14,16] for a few examples. However, unless specified, we focus our discussion on the discretization (9).

Some straightforward algebra can show that the finite difference discretization (7–9) is equivalent to the one proposed in [1,4], where the control parameter **c** is kept in our discretization but is eliminated in [1,4]. The reason we keep the control **c** is that once **c** is fixed, the differential operators of both HJB and KFP equations are linear. This motivates us to use policy iteration [18,28] for solving the nonlinear system, which is easier than Newton's iteration where Jacobians have to be constructed. We will explain policy iteration in Sect. 5. Due to the equivalence between our discretization and the one proposed in [1,4], we can follow [4] to obtain the convergence of the discrete solution to the continuous one.

For convenience, we rewrite (7) and (9) into matrix forms. Let $u^n_h \in \mathbb{R}^{N_x N_y \times 1}$ and $m^n_h \in \mathbb{R}^{N_x N_y \times 1}$ be the unknown vectors at the n-th time step. Let $(c_1)^n_h \in \mathbb{R}^{N_x N_y \times 1}$ and $(c_2)^n_h \in \mathbb{R}^{N_x N_y \times 1}$ be the control vectors in the x and y directions. Define $\mathbf{c}^n_h \equiv \left((c_1)^n_h, (c_2)^n_h\right)^T \in \mathbb{R}^{2N_x N_y \times 1}$. Then the HJB equation (7) can be rewritten as

$$\max_{\mathbf{c}^n_h} \left[A^n_{HJB}(\mathbf{c}^n_h) u^n_h - L(\mathbf{c}^n_h)\right] = 1/\Delta t \cdot u^{n+1}_h + g(m^n_h), \tag{10}$$

where $A^n_{HJB} \in \mathbb{R}^{N_x N_y \times N_x N_y}$ is a matrix that depends on the control vector $\mathbf{c}^n_h$. Similarly, the KFP equation (9) can be rewritten as

$$A^n_{KFP}\left((\mathbf{c}^*)^n_h\right) m^n_h = 1/\Delta t \cdot m^{n-1}_h, \tag{11}$$

where $(\mathbf{c}^*)^n_h \equiv \underset{\mathbf{c}^n_h}{\operatorname{argmax}} \left[A^n_{HJB}(\mathbf{c}^n_h) u^n_h - L(\mathbf{c}^n_h)\right]$ is the optimal control vector given by (10), and A^n_{KFP} is the corresponding matrix.

4 Joint Spacetime Formulation for Solving the Nonlinear Discretized System

Next we consider solving the nonlinear discretized HJB/KFP system (10) and (11). The standard approach for solving the time-dependent system is timestepping. For the HJB/KFP system, timestepping needs to be implemented as a forward/backward fixed point iteration. More specifically, one can start with an initial guess of m_h on the entire spacetime $\Omega \times [0, T]$, fix m_h and solve the HJB equation (10) for u_h and $\mathbf{c}^*_h$ by backward timestepping. Then one can fix $\mathbf{c}^*_h$ and solve the KFP equation (11) for m_h by forward timestepping. The pseudo-code can be found in Algorithm 3 in the appendix.

In this paper, we consider spacetime methods for solving the discretized HJB/KFP system (10–11). The idea of spacetime methods is to treat the unknowns of the discretized HJB/KFP system for *all the timesteps* as one entity, and solve them simultaneously. Our motivation for using spacetime methods is to develop fast solvers for the HJB/KFP system. The idea is that, in addition to the spatial dimensions, spacetime methods allow computational speed-up in the time dimension as well. Some literature, such as [21,22], has also considered using spacetime methods to speed up computation for time dependent PDEs. These methods would not be more advantageous than the conventional timestepping unless they are implemented in the parallel manner. However, we will show that our approach achieves faster convergence than timestepping even without parallelization. Spacetime methods have also been seen in numerical solution for time dependent PDEs on deforming domains [35]. However, this is beyond the scope of our paper.

Mathematically, introduce the spacetime unknown vectors

$$u_h \equiv (u_h^0, \ldots, u_h^{N_t-1})^T \in \mathbb{R}^{N_x N_y N_t \times 1}, \quad m_h \equiv (m_h^1, \ldots, m_h^{N_t})^T \in \mathbb{R}^{N_x N_y N_t \times 1},$$

and the corresponding spacetime control vector $\mathbf{c}_h = (\mathbf{c}_h^0, \ldots, \mathbf{c}_h^{N_t})^T \in \mathbb{R}^{2N_x N_y (N_t+1)\times 1}$. Then the HJB/KFP system (10–11) can be rewritten into the following spacetime matrix forms:

$$\begin{aligned} &A_{HJB}(\mathbf{c}_h^*)u_h = b_{HJB}(\mathbf{c}_h^*, m_h),\\ &\text{subject to} \quad \mathbf{c}_h^* = \underset{\mathbf{c}_h}{\operatorname{argmax}}\,[A_{HJB}(\mathbf{c}_h)u_h - L(\mathbf{c}_h)]\,, \end{aligned} \tag{12}$$

$$A_{KFP}(\mathbf{c}_h^*)m_h = b_{KFP}. \tag{13}$$

Here

$$A_{HJB} = \begin{pmatrix} A_{HJB}^0 & -\frac{1}{\Delta t}I & & \\ & A_{HJB}^1 & -\frac{1}{\Delta t}I & \\ & & \ddots & \ddots \\ & & & A_{HJB}^{N_t-1} \end{pmatrix}, A_{KFP} = \begin{pmatrix} A_{KFP}^1 & & & \\ -\frac{1}{\Delta t}I & A_{KFP}^2 & & \\ & \ddots & \ddots & \\ & & -\frac{1}{\Delta t}I & A_{KFP}^{N_t} \end{pmatrix}$$

are $N_x N_y N_t \times N_x N_y N_t$ matrices, and

$$b_{HJB} = \begin{pmatrix} L((\mathbf{c}^*)_h^0) + g(m_h^0) \\ L((\mathbf{c}^*)_h^1) + g(m_h^1) \\ \vdots \\ L((\mathbf{c}^*)_h^{N_t-1}) + g(m_h^{N_t-1}) + \frac{1}{\Delta t}u_h^{N_t} \end{pmatrix}, \quad b_{KFP} = \begin{pmatrix} \frac{1}{\Delta t}m_h^0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

are $N_x N_y N_t \times 1$ vectors.

We can further rewrite the system (12–13) into a nonlinear discretized system for the joint unknown variable (u_h, m_h) on the entire spacetime domain:

$$\begin{aligned} &\begin{pmatrix} A_{HJB}(\mathbf{c}_h^*) & \\ & A_{KFP}(\mathbf{c}_h^*) \end{pmatrix} \begin{pmatrix} u_h \\ m_h \end{pmatrix} = \begin{pmatrix} b_{HJB}(\mathbf{c}_h^*, m_h) \\ b_{KFP} \end{pmatrix},\\ &\text{subject to} \quad \mathbf{c}_h^* = \underset{\mathbf{c}_h}{\operatorname{argmax}}\,[A_{HJB}(\mathbf{c}_h)u_h - L(\mathbf{c}_h)]\,. \end{aligned} \tag{14}$$

To emphasize that (u_h, m_h) are solved together, we call (14) a “joint" spacetime formulation. We note that this is different from the previous forward/backward timestepping fixed point iteration (Algorithm 3), where timestepping is used for solving u_h and m_h separately.

One may argue that spacetime methods are usually more demanding than conventional timestepping methods in terms of memory, as spacetime methods require storing solutions on the entire spacetime and also the spacetime matrix (14). However, since the HJB equation is backward and the KFP equation is forward, both the forward/backward timestepping methods and the joint spacetime methods require storing both solutions u_h and m_h on the entire spacetime [1]. We can avoid storing the spacetime matrix by the full approximation scheme that will be proposed in Sect. 5. We remark that Sect. 5.3 of [37] explains that a full approximation scheme does not require explicitly constructing matrices. Hence, for the HJB/KFP system (14), the joint spacetime method does not increase memory requirement.

As a side remark, iterating between (10) and (11) with forward/backward timestepping is a block Gauss-Seidel iteration scheme for the joint spacetime system (14).

5 Multigrid Methods

The joint spacetime HJB/KFP system (14) is large, which requires fast and effective solvers. In this section, we will propose a multigrid method for solving (14) efficiently. For an introduction to multigrid methods, we refer readers to [37]. For simplicity, we assume that $\Delta x = \Delta y = h$.

5.1 A Review of Spacetime Multigrid Methods

There have been multigrid methods that are developed on the entire spacetime. For example, [22] considers spacetime multigrid methods for solving discretized linear parabolic PDEs. The authors propose using a block Jacobi smoother and a hybrid full & semi coarsening. We note that the block Jacobi smoother has two deficiencies. One is that it is a block smoother, which is more expensive than pointwise smoothers. The other is that it is a Jacobi smoother, which is weaker than the widely-used Gauss-Seidel smoothers. In addition, the spacetime multigrid method in [22] is only developed for linear systems and cannot be directly applied to the nonlinear system (14).

Outer-inner linearization-based multigrid methods, such as [1,5,6,12], have been proposed for (14). They have been discussed in Sect. 1. We remark that [1,5,6,12] are not the only outer-inner linearization-based multigrid methods. For instance, we can take advantage of the fact that our formulation (14) keeps the optimal control $\mathbf{c}_h^*$ explicitly, unlike [1,5] where $\mathbf{c}_h^*$ is eliminated. This allows us to use policy iteration [18,28]. More specifically, by fixing $\mathbf{c}_h^*$ and the right hand side's m_h, the HJB/KFP system (14) becomes linear. This motivates us to iterate between two tasks: solving the linearized HJB/KFP system under the fixed control $\mathbf{c}_h^*$, and solving the optimization problem for the control. We can use such policy iteration as the outer nonlinear iteration and solve each linearized system by inner multigrid cycles. The pseudo-code for this multigrid scheme is provided in Algorithm 4 in the appendix.

We emphasize, however, that all these schemes require multiple layers of iterations: outer nonlinear iterations (Newton, ALG2, primal-dual, policy); and inner multigrid cycles (or inner BiCGStab iterations with multigrid preconditioners). Due to the outer-inner iterative structure of these algorithms, the total number of iterations can be large. We also note that these multigrid schemes are not fully nonlinear, because the inner multigrid cycles are applied to the linearization of (14), rather than (14) itself.

5.2 Full Approximation Scheme

To address the issues of the outer-inner linearization-based multigrid methods, in this paper, we propose a full approximation scheme (FAS) [10,26,37]. FAS is a family of multigrid schemes that is directly developed upon nonlinear discretized systems, and involves only one layer of iterations. We refer readers to Sect. 5.3 of [37] for an introduction to the FAS. The key component of the FAS is the construction of the nonlinear coarse grid problem. Denote the discretized nonlinear operator for (14) as

$$\begin{aligned} &\mathcal{N}_h(u_h, m_h) \equiv \begin{pmatrix} A_{HJB}(\mathbf{c}_h^*(u_h)) & \\ & A_{KFP}(\mathbf{c}_h^*(u_h)) \end{pmatrix} \begin{pmatrix} u_h \\ m_h \end{pmatrix} - \begin{pmatrix} b_{HJB}(\mathbf{c}_h^*(u_h), m_h) \\ b_{KFP} \end{pmatrix}, \\ &\text{subject to} \quad \mathbf{c}_h^*(u_h) = \underset{\mathbf{c}_h}{\operatorname{argmax}} \left[A_{HJB}(\mathbf{c}_h) u_h - L(\mathbf{c}_h) \right]. \end{aligned} \tag{15}$$

Then we define the following coarse grid problem with respect to $(\hat{u}_{2h}, \hat{m}_{2h})$:

$$\mathcal{N}_{2h}(\hat{u}_{2h}, \hat{m}_{2h}) = \mathcal{N}_{2h}(u_{2h}, m_{2h}) + R_h r_h. \tag{16}$$

Here $\mathcal{N}_{2h}$ is called the direct discretization coarse grid operator, i.e. the left hand sides of (7) and (9) where the mesh spacing h is replaced by $2h$. $(\hat{u}_{2h}, \hat{m}_{2h})$ is the unknown on the coarse grid, (u_{2h}, m_{2h}) is the injection of the current approximated solution (u_h, m_h), r_h is the residual for the approximated (u_h, m_h), and R_h is a restriction operator. The coarse grid error is then given by $e_{2h} = (\hat{u}_{2h}, \hat{m}_{2h}) - (u_{2h}, m_{2h})$.

5.3 Failure of FAS with the Standard Multigrid Components

Naively, one may consider using FAS with the standard multigrid components (i.e. pointwise Gauss–Seidel smoother, full coarsening, bilinear interpolation, full-weighting restriction). However, it turns out that such multigrid methods do not converge in general. The failure is caused by the following issues:

(1) The standard pointwise Gauss–Seidel smoother does not smooth the error in the time direction, if $\sigma \frac{\Delta t}{\Delta x^2} \gg 1$. Consider the KFP equation with zero convection ($\mathbf{c}^* = 0$), namely $m_t - \sigma \Delta m = 0$. Figure 1 shows the errors after 10 steps of Gauss–Seidel iterations, where $\sigma \frac{\Delta t}{\Delta x^2} = 1$ and 32, respectively. We note that when $\sigma \frac{\Delta t}{\Delta x^2} = 32$, the error is not smooth in the time dimension; see Fig. 1(iv)-(v).

(2) The standard full-weighting restriction does not take into account of the one-sided nature of the information propagation in the time dimension.

(3) The standard full-weighting restriction does not take into account of the one-sided nature of the information propagation resulting from the convections in the spatial dimensions. It is well-known that convergence of multigrid deteriorates as the convections increase [37].

(4) Direct discretization coarse grid operator (16) results in a poor coarse grid estimated error. It is shown [11] that if direct discretization is used, when the convections are not aligned with the grid, the convergence factor is no better than 0.5.

In the next few subsections, we will discuss how to address these issues in details.

5.4 Joint Spacetime Pointwise Gauss–Seidel (GS) Smoother

We propose a smoother that is nonlinear based on the policy iteration discussed in Sect. 5.1. More specifically, in one step nonlinear smoothing, we first fix the control $\mathbf{c}_h^*$ in (14), obtain

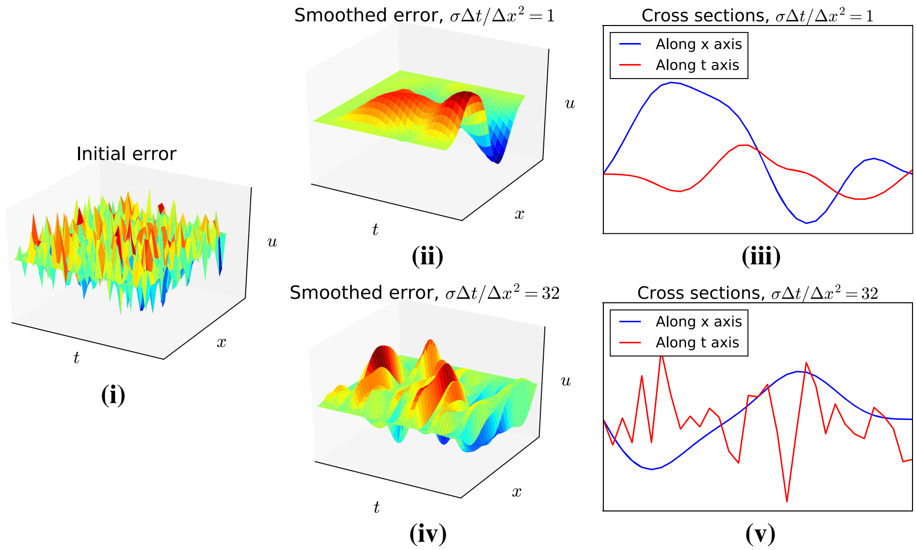


Fig. 1 The errors of the KFP equation with zero convection ($\mathbf{c}^* = 0$), namely $m_t - \sigma\Delta m = 0$. **(i)** Initial error on the 32×32 grid. **(ii)** Error after 10 steps of Gauss–Seidel iterations. $\sigma\Delta t/\Delta x^2 = 1$. **(iii)** Cross sections of the smoothed error in (ii). Blue and red lines are the cross sections of the error along x and t axes respectively. **(iv)** Error after 10 steps of Gauss–Seidel iterations. $\sigma\Delta t/\Delta x^2 = 32$. **(v)** Cross sections of the smoothed error in (iv)

the linearized HJB/KFP system and perform smoothing for the linearized problem; then we update the control by solving the nonlinear optimization problem in (14) under the latest u_h.

Regarding the smoother for the linearized problem of (14), we consider a spacetime pointwise GS smoother. That is, we perform forward timestepping for the KFP equation (9) and backward timestepping for the HJB equation (7), where pointwise GS smoother is applied at each timestep. We note that both (7) and (9) are convection-diffusion problems. In some mean field games where the convection is guaranteed to be non-negative (such as the demand function D in Sect. 2.2, production quantity in [15,24], etc), the smoother at each timestep is the downstream GS smoother; in other mean field games where the sign of the convection may change at different grid points, the smoother at each timestep is the four-direction GS smoother [37]. We note that if the four-direction smoother is applied, it is only applied in the spatial dimensions; in the time dimension, the smoother remains one-directional.

The pseudo-code for the proposed smoother is given as follows:

5.5 Hybrid Full-Semi Coarsening

Section 5.3 has shown that the pointwise GS smoother does not smooth the error in the time direction when $\sigma\frac{\Delta t}{\Delta x^2}$ is large. To explain this, consider again the KFP equation with zero convection, namely the heat equation $m_t - \sigma\Delta m = 0$. When $\sigma\frac{\Delta t}{\Delta x^2} \gg 1$, the discretized heat equation is highly anisotropic. That is, it is strongly connected in the spatial directions but weakly connected in the time direction. It is well-known that pointwise smoothers do not smooth errors in the weakly connected direction [37].

To address this issue, one may use block smoothers, where each block corresponds to the 2-dimensional sub-mesh at each time step [22]. However, each block is a linear sys-

Algorithm 1 Joint spacetime pointwise GS smoother

```
subroutine $(\bar{u}_h, \bar{m}_h) = \text{SMOOTH}\ (u_h, m_h)$
for $n = 1, \cdots, N_t$ do
    Update the control: $\bar{\mathbf{c}}_h^n = \underset{\mathbf{c}_h^n}{\text{argmax}} \left[ A_{HJB}^n(\mathbf{c}_h^n) u_h^n - L(\mathbf{c}_h^n) \right]$.
    Apply one step GS smoother on the linearized KFP equation $A_{KFP}^n(\bar{\mathbf{c}}_h^n) m_h^n = 1/\Delta t \cdot \bar{m}_h^{n-1}$, which updates the solution $m_h^n \to \bar{m}_h^n$.
end for
for $n = N_t - 1, \cdots, 0$ do
    Apply one step GS smoother on the linearized HJB equation $A_{HJB}^n(\bar{\mathbf{c}}_h^n) u_h^n = L(\bar{\mathbf{c}}_h^n) + g(\bar{m}_h^n) + 1/\Delta t \cdot \bar{u}_h^{n+1}$, which updates the solution $u_h^n \to \bar{u}_h^n$.
end for
```

tem of size $(N_x N_y) \times (N_x N_y)$, and the cost of solving the linear system is as high as $O(\min(N_x^2, N_y^2) N_x N_y)$. The problem is more severe if the dimension of the space is greater than 2.

Alternatively, following the idea in [5,12,27], we stick to a pointwise smoother, where the cost is only $O(N_x N_y)$. However, in order to use a pointwise smoother, the coarsening strategy is changed to semi-coarsening. More specifically, the strongly connected dimensions, namely the spatial dimensions, are fully coarsened; the weakly connected dimension, namely the time dimension, remains uncoarsened.

We note that if we perform semi-coarsening, then $\sigma \frac{\Delta t}{\Delta x^2}$ will decrease on the coarse grids. When $\sigma \frac{\Delta t}{\Delta x^2}$ is no longer large, the pointwise GS smoother can effectively smooth the error in the time direction as well; see Fig. 1(ii–iii). In this case, coarsening can also be applied in the time direction.

As a result, we can combine these two coarsening strategies together. More specifically, when $\sigma \frac{\Delta t}{\Delta x^2}$ is larger than a threshold value (e.g., 1), we use semi-coarsening in the spatial dimensions only; otherwise, we use full-coarsening on the entire spacetime grid. This gives rise to a hybrid full-semi coarsening scheme.

We remark that semi-coarsening has been proposed in [5,12] for the HJB/KFP system. However, our strategy is a hybrid coarsening rather than a pure semi-coarsening. In addition, we remark that although [22] also uses a hybrid coarsening (for linear PDEs), it is very different from our hybrid coarsening. More specifically, [22] uses full coarsening when $\sigma \frac{\Delta t}{\Delta x^2}$ is large; we use full coarsening when $\sigma \frac{\Delta t}{\Delta x^2}$ is small. In the event of semi-coarsening, [22] applies coarsening in the time dimension; we apply coarsening in the spatial dimensions. These differences are caused by difference choices of smoothers, i.e. the block smoother in [22] versus our pointwise smoother.

5.6 Interpolation

Under the proposed hybrid coarsening strategy, the errors are smooth along the coarsened directions. As a result, to transfer the errors from coarse grids to fine grids, we use the standard trilinear interpolation for full-coarsening and the standard bilinear interpolation for semi-coarsening.

5.7 Forward/Backward Restriction in Time and Kernel Preserving Restriction in Space

We note that the KFP equation is forward in time, while the HJB equation is backward in time. Hence, following [27], we use forward and backward restrictions in the time direction for the KFP and HJB equations, respectively. If we use the stencil notation introduced in Section 1.3.4 of [37], then the restriction operators read

$$\text{KFP: } \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}, \quad \text{HJB: } \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{17}$$

The restriction in the spatial dimensions must take into account of the one-sided convectional effect. Let us first consider the spatial restriction for the KFP equation. Following [9], we consider kernel preserving biased restriction. The idea is to capture the hyperbolic nature of the PDE. The restriction weights are biased towards the upwind side and matched with the flow direction of the error. Such biased restriction may not be unique. Kernel preserving scheme is one type of biased restriction that preserves the kernels of the differential operator of the KFP equation, $-\sigma\Delta m - \nabla\cdot(\mathbf{c}m)$. In this case, the kernels are the arbitrary constant and the exponential function $\exp(-\sigma^{-1}\mathbf{c}\cdot\mathbf{x})$. To preserve the kernels, [9] proposes the following restriction operator

$$R_h(\mathbf{c}) = \frac{1}{4}\begin{bmatrix} 0 & \frac{1}{1+\exp(-\sigma^{-1}\mathbf{c}\cdot(0,h))} & \frac{1}{1+\exp(-\sigma^{-1}\mathbf{c}\cdot(h,h))} \\ \frac{1}{1+\exp(-\sigma^{-1}\mathbf{c}\cdot(-h,0))} & 1 & \frac{1}{1+\exp(-\sigma^{-1}\mathbf{c}\cdot(h,0))} \\ \frac{1}{1+\exp(-\sigma^{-1}\mathbf{c}\cdot(-h,-h))} & \frac{1}{1+\exp(-\sigma^{-1}\mathbf{c}\cdot(0,-h))} & 0 \end{bmatrix}, \tag{18}$$

when $c_1c_2 > 0$. Similar operator can be derived for $c_1c_2 < 0$. We note that when $c_1 = c_2 = 0$, the restriction operator (18) is reduced to a 7-point constant restriction, which is an alternative of the full-weighting restriction. Conversely, when $|c_1|$ and $|c_2|$ are very large, it becomes a pure upwind biased constant restriction. Regarding the HJB equation, the convection coefficient has an opposite sign compared with the KFP equation. To obtain the restriction operator of the HJB equation, we simply replace $\mathbf{c}$ in (18) by $-\mathbf{c}$.

We note that in [9], the kernel preserving biased restriction is combined with the Petrov-Galerkin coarse grid operator $A_{2h} \equiv R_hA_hP_h$, where R_h and P_h are the restriction (18) and the trilinear/bilinear interpolation.

As analyzed in [9], the kernel preserving biased restriction operator has several desirable properties. One is that it captures the one-sided nature of the convections and preserves the kernel of $-\sigma\Delta m - \nabla\cdot(\mathbf{c}m)$. Another advantage is that the phase error of the coarse grid correction is negligible. In addition, the resulting Petrov-Galerkin coarse grid operators are nearly M-matrices, which is crucial for the stability of multigrid. Conversely, Galerkin coarse grid operators under the standard full-weighting restriction are not M-matrices [9,37].

Eventually, the restriction operator on the entire spacetime is the tensor product of the restriction operators in the space dimensions and in the time dimension.

5.8 Direct Discretization and Artificial Viscosity

Despite the advantage of the kernel preserving restriction in [9], it combines with the Petrov-Galerkin coarse grid operator, which is incompatible with the nonlinearity of the FAS. More specifically, under the FAS framework, the Petrov-Galerkin coarse grid operator is given by $A_{2h}(\mathbf{c}) \equiv R_h(\mathbf{c})A_h(\mathbf{c})P_h$, where each matrix entry of $A_{2h}(\mathbf{c})$ becomes a nonlinear function that depends on the control $\mathbf{c}$. This is much more complicated than the Petrov-Galerkin coarse grid operator for the linear problems considered in [9], where each matrix entry of A_{2h} is a number. As a result, constructing the Petrov-Galerkin coarse grid operator for the nonlinear FAS is impractical.

In order to make FAS a practical approach, the FAS literature uses direct discretization as the coarse grid operator, as defined in (16). However, if direct discretization coarse grid operator is used, then the kernel preserving restriction does not yield precise coarse grid estimated errors any more.

In this paper, we propose to modify the direct discretization coarse grid operator (16), such that it becomes a good approximation to the Petrov-Galerkin coarse grid operator, and thus yields accurate coarse grid estimated errors.

In order to achieve this, we investigate the difference between Petrov-Galerkin and direct discretization coarse grid operators. For simplicity, we first consider the one dimensional steady-state linear convection-diffusion equation

$$-\sigma m_{xx} - c m_x = 0, \tag{19}$$

where c is a positive constant. If we again use the stencil notation introduced in Section 1.3.4 of [37], then the finite difference stencil is given by

$$A_h(c) = \left[-\frac{\sigma}{h^2} \quad \frac{2\sigma}{h^2} + \frac{c}{h} \quad -\frac{\sigma}{h^2} - \frac{c}{h} \right]. \tag{20}$$

If we replace h by $2h$, we obtain the direct discretization coarse grid operator:

$$A_{2h}^{DD}(c) = \left[-\frac{\sigma}{(2h)^2} \quad \frac{2\sigma}{(2h)^2} + \frac{c}{2h} \quad -\frac{\sigma}{(2h)^2} - \frac{c}{2h} \right]. \tag{21}$$

On the other hand, under the standard linear interpolation P_h and the kernel preserving restriction

$$R_h(c) = \frac{1}{2} \left[\frac{1}{1+\exp(\sigma^{-1}ch)} \quad 1 \quad \frac{1}{1+\exp(-\sigma^{-1}ch)} \right], \tag{22}$$

the Petrov-Galerkin coarse grid operator is given by

$$\begin{aligned} A_{2h}^{PG}(c) &= R_h(c) A_h(c) P_h \\ &= \left[-\frac{\sigma}{(2h)^2} + \frac{(1-\eta)c}{8h} \quad \frac{2\sigma}{(2h)^2} + \frac{(1+\eta)c}{4h} \quad -\frac{\sigma}{(2h)^2} - \frac{(3+\eta)c}{8h} \right], \end{aligned} \tag{23}$$

where $\eta \equiv \tanh(\frac{ch}{2\sigma})$. Then the difference between the two coarse grid operators (23) and (21) is

$$A_{2h}^{PG}(c) - A_{2h}^{DD}(c) = \frac{1}{2}(1-\eta)ch \cdot \left[\frac{1}{(2h)^2} \quad -\frac{2}{(2h)^2} \quad \frac{1}{(2h)^2} \right]. \tag{24}$$

Significantly, this turns out to be the stencil for an $O(h)$ viscosity $\frac{1}{2}(1-\eta)ch\, m_{xx}$. Motivated by this fact, we consider adding this $O(h)$ “artificial viscosity" to the direct discretization coarse grid operator $A_{2h}^{DD}(c)$. This yields the Petrov-Galerkin coarse grid operator $A_{2h}^{PG}(c)$ and thus a more precise coarse grid error estimation.

To summarize, our multigrid scheme for (19) is to construct the direct discretization coarse grid operator, but instead of using the original viscosity σ, we use the damped viscosity

$$\hat{\sigma} = \sigma - \frac{1}{2}(1-\eta)|c|h, \tag{25}$$

where

$$\eta \equiv \tanh\left(\frac{|c|h}{2\sigma}\right). \tag{26}$$

Here we put the absolute value on c to generalize the result from $c > 0$ to any c.

Next we consider the two dimensional linear convection-diffusion equation

$$-\sigma_1 m_{xx} - \sigma_2 m_{yy} - c_1 m_x - c_2 m_y = 0. \tag{27}$$

We obtain the difference between the two coarse grid operators:

$$\begin{aligned} A_{2h}^{PG}(c) - A_{2h}^{DD}(c) = \frac{1}{4}h\big[&(2-\eta_{12}-\eta_1)|c_1|m_{xx} + (2-\eta_{12}-\eta_2)|c_2|m_{yy} \\ &-\mathrm{sign}(c_1c_2)((\eta_{12}+\eta_2)|c_1| + (\eta_{12}+\eta_1)|c_2|)m_{xy} \\ &-\mathrm{sign}(c_2)2\sigma_1\eta_2 m_{xxy} - \mathrm{sign}(c_1)2\sigma_2\eta_1 m_{xyy} + O(h)\big], \end{aligned} \tag{28}$$

where

$$\eta_1 \equiv \tanh\left(\frac{|c_1|h}{2\sigma_1}\right),\ \eta_2 \equiv \tanh\left(\frac{|c_2|h}{2\sigma_2}\right),\ \eta_{12} \equiv \tanh\left(\frac{|c_1|h}{2\sigma_1} + \frac{|c_2|h}{2\sigma_2}\right). \tag{29}$$

Notice that when $h \to 0$, η_1, η_2 and η_{12} are also $O(h)$. If we assume that $\frac{|c_1|h}{2\sigma_1}$ and $\frac{|c_2|h}{2\sigma_2}$ are not much larger than 1, then (28) can be approximated by

$$A_{2h}^{PG}(c) - A_{2h}^{DD}(c) \approx \frac{1}{4}h\left[(2-\eta_{12}-\eta_1)|c_1|m_{xx} + (2-\eta_{12}-\eta_2)|c_2|m_{yy}\right], \tag{30}$$

where we only keep the viscosity terms in (28).

Similar to the one dimensional convection-diffusion equation, for the two dimensional case (27), our multigrid scheme is to construct the direct discretization coarse grid operator, where we add $O(h)$ artificial viscosity to the original σ_1 and σ_2 and damp them to

$$\hat{\sigma}_1 = \sigma_1 - \frac{1}{4}(2-\eta_{12}-\eta_1)|c_1|h, \quad \hat{\sigma}_2 = \sigma_2 - \frac{1}{4}(2-\eta_{12}-\eta_2)|c_2|h. \tag{31}$$

Figure 2 shows an example of the multigrid errors for the two-dimensional convection-diffusion equation (27). By comparing Figs. 2(iv) and 2(v), we observe that with artificial viscosity, the coarse grid estimated error (red) becomes closer to the pre-smoothed error (blue), and the post-smoothed error (black) becomes smaller. In other words, adding artificial viscosity yields a more precise coarse grid estimated error and a more efficient multigrid error reduction.

In this paper, we extend the proposed idea of artificial viscosity from the linear convection-diffusion equations to the nonlinear HJB/KFP system (14). We will demonstrate in Sect. 7 that by adding the artificial viscosity to the direct discretization coarse grid operator, we obtain a more efficient multigrid method for the nonlinear HJB/KFP system.

Eventually, we summarize the proposed artificial viscosity joint spacetime FAS multigrid method in Algorithm 2.

6 Local Fourier Analysis

In this section, we use Local Fourier Analysis (LFA) [37,38] to demonstrate the efficiency of the proposed multigrid method. Let us consider the linearized KFP equation

$$m_t - \sigma\Delta m + \nabla\cdot(\mathbf{c}m) = 0. \tag{32}$$

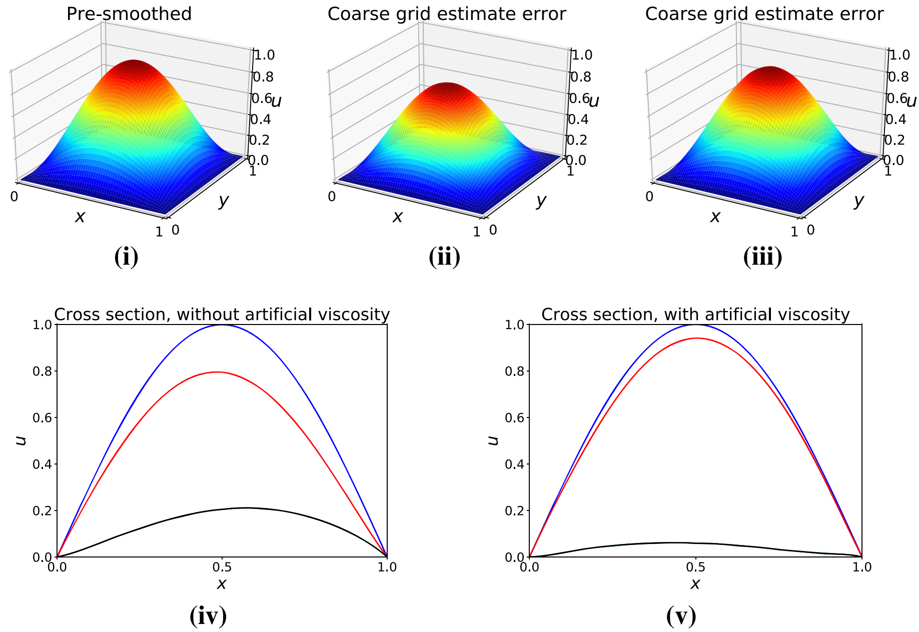


Fig. 2 The multigrid (6-grid) errors for the two-dimensional convection-diffusion equation $-\sigma\Delta m - \nabla\cdot(\mathbf{c}m) = 0$, where $\sigma = 1$ and $\mathbf{c} = (-7, -7)^T$. **(i)** Pre-smoothed error. **(ii)** Coarse grid estimated error *without* artificial viscosity. **(iii)** Coarse grid estimated error *with* artificial viscosity. **(iv)** Cross sections of the pre-smoothed error (blue), coarse grid estimated error (red) and post-smoothed error (black) along the x axis *without* artificial viscosity. **(v)** Cross sections of the pre-smoothed error (blue), coarse grid estimated error (red) and post-smoothed error (black) along the x axis *with* artificial viscosity

We assume that $\mathbf{c}$ is a positive constant. Also, we assume that $\Delta x = \Delta y = h$.

6.1 Smoothing Analysis

We first analyze the smoothing property of the joint spacetime GS smoother proposed in Sect. 5.4. Since we assume that $\mathbf{c} \geq 0$, in the spatial dimensions, it is sufficient to consider the downstream (rather than four-direction) GS smoother. Define the (2+1)-dimensional spacetime Fourier modes as $\varphi_{h,\Delta t}(\boldsymbol{\theta}, \mathbf{x}) \equiv \exp\left[i\left((\theta_1, \theta_2)\cdot(x, y)/h + \theta_0 t/\Delta t\right)\right]$, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_0) \in [-\pi, \pi)^3$, and $\mathbf{x} = (x, y, t)$. Following [37], we obtain the Fourier symbol of the smoother:

$$\tilde{S}_h(\boldsymbol{\theta}) = \frac{\frac{\sigma\Delta t}{h^2}(e^{i\theta_1}+e^{i\theta_2})}{1+\frac{\sigma\Delta t}{h^2}(4+\frac{c_1h}{\sigma}+\frac{c_2h}{\sigma}) - \frac{\sigma\Delta t}{h^2}(1+\frac{c_1h}{\sigma})e^{-i\theta_1} - \frac{\sigma\Delta t}{h^2}(1+\frac{c_2h}{\sigma})e^{-i\theta_2} - e^{-i\theta_0}}.$$

We define its smoothing factor as $\mu_{loc} \equiv \sup_\kappa \left\{|\tilde{S}_h(\boldsymbol{\theta})| : \boldsymbol{\theta} \in \text{high frequency mode}\right\}$. Smoothing factor reaches its best value at 0 and worst at 1. One can see that the smoothing factor is determined by three ratios: $\frac{\sigma\Delta t}{h^2}$, $\frac{c_1h}{\sigma}$ and $\frac{c_2h}{\sigma}$.

Table 1 reports the smoothing factor μ_{loc} under different combinations of $\frac{\sigma\Delta t}{h^2}$ and $(\frac{c_1h}{\sigma}, \frac{c_2h}{\sigma})$. We can see that if full-coarsening is used, then the smoothing factor depends on the ratio $\frac{\sigma\Delta t}{h^2}$. When $\frac{\sigma\Delta t}{h^2}$ is large, the smoothing factor is close to 1. When $\frac{\sigma\Delta t}{h^2}$ is small,

Algorithm 2 Artificial viscosity joint spacetime FAS multigrid method

```
Start with an initial guess $(u_h^{(0)}, m_h^{(0)})$.
for $k = 1, 2, \ldots$ until convergence do
    $(u_h^{(k+1)}, m_h^{(k+1)})$ = FASCYC $(u_h^{(k)}, m_h^{(k)}, 0, \sigma, \sigma, \gamma, \nu_1, \nu_2)$. See below for the subroutine "FASCYC".
end for

subroutine $(u_h^{(k+1)}, m_h^{(k+1)})$ = FASCYC $(u_h^{(k)}, m_h^{(k)}, q_h, (\sigma_1)_h, (\sigma_2)_h, \gamma, \nu_1, \nu_2)$
Construct the nonlinear operator (15), $\mathcal{N}_h$, using the viscosity $(\sigma_1)_h$ and $(\sigma_2)_h$.
Perform $\nu_1$ smoothing steps (Algorithm 1) on $\mathcal{N}_h(u_h, m_h) = q_h$, which updates the solution $(u_h^{(k)}, m_h^{(k)}) \to (\bar{u}_h^{(k)}, \bar{m}_h^{(k)})$.
Compute the residual: $r_h = q_h - \mathcal{N}_h(\bar{u}_h^{(k)}, \bar{m}_h^{(k)})$.
Determine whether to use full-coarsening or semi-coarsening according to Section 5.5.
Inject the solution: $(\bar{u}_h^{(k)}, \bar{m}_h^{(k)}) \to (\bar{u}_{2h}^{(k)}, \bar{m}_{2h}^{(k)})$.
Restrict the residual: $r_{2h} = R_h r_h$, where $R_h$ is the restriction in Section 5.7.
Inject the viscosity: $(\sigma_1)_h \to (\sigma_1)_{2h}$, $(\sigma_2)_h \to (\sigma_2)_{2h}$.
Add artificial viscosity: $(\sigma_1)_{2h} \leftarrow (\sigma_1)_{2h} - \frac{1}{4}(2 - \eta_{12} - \eta_1)|c_1|h$, $(\sigma_2)_{2h} \leftarrow (\sigma_2)_{2h} - \frac{1}{4}(2 - \eta_{12} - \eta_2)|c_2|h$, where $\eta_1, \eta_2$ and $\eta_{12}$ are defined in (29).
Construct the coarse grid nonlinear operator $\mathcal{N}_{2h}$, using the viscosity $(\sigma_1)_{2h}$ and $(\sigma_2)_{2h}$.
Compute the right hand side: $q_{2h} = \mathcal{N}_{2h}(\bar{u}_{2h}^{(k)}, \bar{m}_{2h}^{(k)}) + R_h r_h$.
if on the coarsest grid then
    Solve $\mathcal{N}_{2h}(\hat{u}_{2h}^{(k)}, \hat{m}_{2h}^{(k)}) = q_{2h}$ for $(\hat{u}_{2h}^{(k)}, \hat{m}_{2h}^{(k)})$, using Algorithm 1 repeatedly.
else
    Solve $\mathcal{N}_{2h}(\hat{u}_{2h}^{(k)}, \hat{m}_{2h}^{(k)}) = q_{2h}$ for $(\hat{u}_{2h}^{(k)}, \hat{m}_{2h}^{(k)})$ approximately by $\gamma$-time recursions of
            $(\hat{u}_{2h}^{(k)}, \hat{m}_{2h}^{(k)})$ = FASCYC $(\bar{u}_{2h}^{(k)}, \bar{m}_{2h}^{(k)}, q_{2h}, (\sigma_1)_{2h}, (\sigma_2)_{2h}, \gamma, \nu_1, \nu_2)$.
end if
Compute the coarse grid estimated error: $e_{2h} = (\hat{u}_{2h}^{(k)}, \hat{m}_{2h}^{(k)}) - (\bar{u}_{2h}^{(k)}, \bar{m}_{2h}^{(k)})$.
Interpolate the estimated error: $e_h = P_h e_{2h}$, where $P_h$ is the trilinear or bilinear interpolation.
Correct the fine grid solution: $(\tilde{u}_h^{(k)}, \tilde{m}_h^{(k)}) = (\hat{u}_h^{(k)}, \hat{m}_h^{(k)}) + e_h$.
Perform $\nu_2$ smoothing steps (Algorithm 1) on $\mathcal{N}_h(u_h, m_h) = q_h$, which updates the solution $(\tilde{u}_h^{(k)}, \tilde{m}_h^{(k)}) \to (u_h^{(k+1)}, m_h^{(k+1)})$.
```

the smoothing factor is much smaller than 1. This means that it is desirable to use full-coarsening if and only if $\frac{\sigma \Delta t}{h^2}$ is small. On the other hand, if semi-coarsening is applied, then the smoothing factor is basically determined by the ratios between convection and diffusion, $(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma})$. The smoothing factor decreases as the ratios increase. Indeed, when the ratios are infinity, or, when the linear problem (32) becomes purely hyperbolic, the smoothing factor becomes 0, which seems to suggest that purely hyperbolic problems can be solved by one single Gauss–Seidel iteration. However, this is only true for linear problems. The original KFP equation (2) would not be solved by one single Gauss–Seidel iteration due to the nonlinearity and the coupling with the HJB equation (1), even when the problem becomes hyperbolic.

6.2 Two-grid Analysis

In this subsection, we follow [38] and consider two-grid analysis for the full and semi coarsenings. The Fourier symbols of the differential operator of (32), the interpolation operators

Table 1 The smoothing factor μ_{loc} for different combinations of $\frac{\sigma \Delta t}{h^2}$ and $(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma})$ and for **(i)** full coarsening, and **(ii)** semi coarsening

	$(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma}) = (0.1, 0.1)$	$(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma}) = (1, 1)$	$(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma}) = (10, 10)$
(i) Full coarsening			
$\frac{\sigma \Delta t}{h^2} = 10$	0.951	0.951	0.951
$\frac{\sigma \Delta t}{h^2} = 2$	0.792	0.793	0.784
$\frac{\sigma \Delta t}{h^2} = 0.4$	0.499	0.411	0.433
(ii) Semi coarsening			
$\frac{\sigma \Delta t}{h^2} = 10$	0.478	0.334	0.083
$\frac{\sigma \Delta t}{h^2} = 2$	0.483	0.339	0.084
$\frac{\sigma \Delta t}{h^2} = 0.4$	0.499	0.354	0.086

and the restriction operators are

$$\begin{aligned}\tilde{L}_h(\boldsymbol{\theta}) = {} & \left(1 + \frac{\sigma \Delta t}{h^2}\left(4 + \frac{c_1 h}{\sigma} + \frac{c_2 h}{\sigma}\right)\right) - \frac{\sigma \Delta t}{h^2}\left(1 + \frac{c_1 h}{\sigma}\right) e^{-i\theta_1} - \frac{\sigma \Delta t}{h^2}\left(1 + \frac{c_2 h}{\sigma}\right) e^{-i\theta_2} \\ & - \frac{\sigma \Delta t}{h^2} e^{i\theta_1} - \frac{\sigma \Delta t}{h^2} e^{i\theta_2} - e^{-i\theta_0}.\end{aligned}$$

$$\tilde{I}_{2h}^{h}(\boldsymbol{\theta}) = \begin{cases} \frac{1}{8}\left(1 + \cos\theta_1\right)\left(1 + \cos\theta_2\right)\left(1 + \cos\theta_0\right), & \text{full coarsening;} \\ \frac{1}{4}\left(1 + \cos\theta_1\right)\left(1 + \cos\theta_2\right), & \text{semi coarsening.} \end{cases}$$

$$\tilde{I}_{h}^{2h}(\boldsymbol{\theta}) = \begin{cases} \frac{1}{8}\left(1 + e^{-i\theta_0}\right)\Bigg(1 + \frac{e^{i\theta_1}}{1+\exp(\sigma_1^{-1} c_1 h)} + \frac{e^{-i\theta_1}}{1+\exp(-\sigma_1^{-1} c_1 h)} + \frac{e^{i\theta_2}}{1+\exp(\sigma_2^{-1} c_2 h)} \\ \quad + \frac{e^{-i\theta_2}}{1+\exp(-\sigma_2^{-1} c_2 h)} + \frac{e^{i(\theta_1+\theta_2)}}{1+\exp(\sigma_1^{-1} c_1 h + \sigma_2^{-1} c_2 h)} + \frac{e^{-i(\theta_1+\theta_2)}}{1+\exp(-\sigma_1^{-1} c_1 h - \sigma_2^{-1} c_2 h)}\Bigg), \\ \hfill \text{full coarsening;} \\ \frac{1}{4}\Bigg(1 + \frac{e^{i\theta_1}}{1+\exp(\sigma_1^{-1} c_1 h)} + \frac{e^{-i\theta_1}}{1+\exp(-\sigma_1^{-1} c_1 h)} + \frac{e^{i\theta_2}}{1+\exp(\sigma_2^{-1} c_2 h)} \\ \quad + \frac{e^{-i\theta_2}}{1+\exp(-\sigma_2^{-1} c_2 h)} + \frac{e^{i(\theta_1+\theta_2)}}{1+\exp(\sigma_1^{-1} c_1 h + \sigma_2^{-1} c_2 h)} + \frac{e^{-i(\theta_1+\theta_2)}}{1+\exp(-\sigma_1^{-1} c_1 h - \sigma_2^{-1} c_2 h)}\Bigg), \\ \hfill \text{semi coarsening.} \end{cases}$$

Now we are ready to construct the Fourier symbol of the two-grid operator. Given a low frequency mode $\boldsymbol{\theta}^{000} \equiv \boldsymbol{\theta} \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^3$, we define an 8-dimensional space: $\mathrm{span}\{\varphi_{h,\Delta t}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}, \cdot) : \boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_0), \alpha_1, \alpha_2, \alpha_0 \in \{0, 1\}\}$, where $\boldsymbol{\theta}^{\boldsymbol{\alpha}} \equiv \boldsymbol{\theta}^{000} - (\alpha_1 \mathrm{sign}(\theta_1), \alpha_2 \mathrm{sign}(\theta_2), \alpha_0 \mathrm{sign}(\theta_0)) \cdot \pi$. This 8-dimensional space is called $2h$-harmonics. The significance of the $2h$-harmonics is that it is invariant under the two-grid operator M_h^{2h}. In the $2h$-harmonics, the two-grid operator is given by an 8×8 matrix

$$M_h^{2h}(\boldsymbol{\theta}) \equiv (S_h(\boldsymbol{\theta}))^{\nu_2}\left[I_h - I_{2h}^h(\boldsymbol{\theta})(L_{2h}(2\boldsymbol{\theta}))^{-1} I_h^{2h}(\boldsymbol{\theta}) L_h(\boldsymbol{\theta})\right](S_h(\boldsymbol{\theta}))^{\nu_1},$$

where $I_h \in \mathbb{R}^{8\times 8}$ is an identity matrix, $L_h(\boldsymbol{\theta}) \in \mathbb{C}^{8\times 8}$ is a diagonal matrix consisting of $\{\tilde{L}_h(\boldsymbol{\theta}^{\boldsymbol{\alpha}})\}$, $S_h(\boldsymbol{\theta}) \in \mathbb{C}^{8\times 8}$ is a diagonal matrix consisting of $\{\tilde{S}_h(\boldsymbol{\theta}^{\boldsymbol{\alpha}})\}$. For full-coarsening, the 8-dimensional $2h$-harmonics is mapped to a single Fourier mode $\varphi_{2h,2\Delta t}(2\boldsymbol{\theta}^{000}, \cdot)$. Hence, $I_{2h}^h(\boldsymbol{\theta}) \in \mathbb{C}^{8\times 1}$ is a column matrix consisting of $\{\tilde{I}_{2h}^h(\boldsymbol{\theta}^{\boldsymbol{\alpha}})\}$, $I_h^{2h}(\boldsymbol{\theta}) \in \mathbb{C}^{1\times 8}$ is a row matrix consisting of $\{\tilde{I}_h^{2h}(\boldsymbol{\theta}^{\boldsymbol{\alpha}})\}$, and $L_{2h}(2\boldsymbol{\theta}) \equiv \tilde{L}_{2h}(2\boldsymbol{\theta}^{000})$ is a 1×1 matrix.

Table 2 The two-grid convergence factor $\rho_{loc}(M_h^{2h})$ and error reduction factor $\sigma_{loc}(M_h^{2h})$ for different combinations of $\frac{\sigma\Delta t}{h^2}$ and $(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma})$ and for **(i)** full coarsening, and **(ii)** semi coarsening

	$(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma}) = (0.1, 0.1)$		$(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma}) = (1, 1)$		$(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma}) = (10, 10)$	
	$\rho_{loc}(M_h^{2h})$	$\sigma_{loc}(M_h^{2h})$	$\rho_{loc}(M_h^{2h})$	$\sigma_{loc}(M_h^{2h})$	$\rho_{loc}(M_h^{2h})$	$\sigma_{loc}(M_h^{2h})$
(i) Full coarsening						
$\frac{\sigma\Delta t}{h^2} = 10$	0.89	0.97	0.89	0.97	0.89	0.97
$\frac{\sigma\Delta t}{h^2} = 2$	0.54	0.67	0.59	0.67	0.51	0.67
$\frac{\sigma\Delta t}{h^2} = 0.4$	0.50	0.52	0.56	0.60	0.50	0.52
(ii) Semi coarsening						
$\frac{\sigma\Delta t}{h^2} = 10$	0.22	0.29	0.22	0.28	0.18	0.19
$\frac{\sigma\Delta t}{h^2} = 2$	0.23	0.29	0.22	0.28	0.18	0.19
$\frac{\sigma\Delta t}{h^2} = 0.4$	0.26	0.30	0.22	0.34	0.18	0.19

For semi-coarsening, the 8-dimensional $2h$-harmonics is mapped to two Fourier modes $\{\varphi_{2h,\Delta t}((2\theta_1, 2\theta_2, \theta_0), \cdot), \varphi_{2h,\Delta t}((2\theta_1, 2\theta_2, \theta_0 - \text{sign}(\theta_0)\pi), \cdot)\}$. Hence, $I_{2h}^h(\boldsymbol{\theta})$, $I_h^{2h}(\boldsymbol{\theta})$ and $L_{2h}(2\boldsymbol{\theta})$ are changed accordingly into 8×2, 2×8 and 2×2 matrices. We refer readers to [38] for technical details.

Based on $M_h^{2h}(\boldsymbol{\theta})$, we define the asymptotic convergence factor and the error reduction factor as

$$\begin{aligned} \rho_{loc}(M_h^{2h}) &\equiv \sup\left\{\rho(M_h^{2h}(\boldsymbol{\theta})) : \boldsymbol{\theta} \in \text{low frequency mode}\right\}, \\ \sigma_{loc}(M_h^{2h}) &\equiv \sup\left\{\|M_h^{2h}(\boldsymbol{\theta})\|_2 : \boldsymbol{\theta} \in \text{low frequency mode}\right\}. \end{aligned}$$

Similar to the smoothing factor, the asymptotic convergence factor and the error reduction factor reach their best values at 0 and worst at 1, and are determined by the three ratios $\frac{\sigma\Delta t}{h^2}$, $\frac{c_1 h}{\sigma}$ and $\frac{c_2 h}{\sigma}$.

Table 2 reports the two-grid convergence factor $\rho_{loc}(M_h^{2h})$ and error reduction factor $\sigma_{loc}(M_h^{2h})$ under different combinations of $\frac{\sigma\Delta t}{h^2}$ and $(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma})$. Table 2(i) shows that if full-coarsening is used, then the factors depend on the ratio $\frac{\sigma\Delta t}{h^2}$. When $\frac{\sigma\Delta t}{h^2}$ is large, the factors are close to 1. When $\frac{\sigma\Delta t}{h^2}$ is small, the factors are much smaller than 1. Again, this suggests an efficient error reduction under full-coarsening if and only if $\frac{\sigma\Delta t}{h^2}$ is small. Table 2(ii) shows that if semi-coarsening is applied, then the factors are mainly determined by the ratios $(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma})$. The factors decrease as $(\frac{c_1 h}{\sigma}, \frac{c_2 h}{\sigma})$ increase. However, similar to the discussion at the end of Section 6.1, this does not imply a more efficient error reduction in convection-dominant regime than in diffusion-dominant regime, due to the nonlinearity of the HJB/KFP system. Interested readers are referred to [20] for more discussions on the relation between spacetime multigrid convergence results and the corresponding LFA estimate.

6.3 The Effect of Adding Artificial Viscosity

Our proposed multigrid method considers adding artificial viscosity. For simplicity, here we consider the steady-state version of (32), or equivalently, $-\sigma\Delta m + \nabla \cdot (\mathbf{c}m) = 0$. In this subsection, we will use multigrid (i.e. three-grid, four-grid, etc) analysis to illustrate the improvement of adding artificial viscosity. An introduction to multigrid analysis can be found in [38].

Table 3 reports the multigrid asymptotic convergence factors and error reduction factors for the linearized KFP equation. Figure 3 illustrates the multigrid asymptotic convergence factors $\rho_{loc}(M)$ versus the convection coefficient $\mathbf{c}$. We compare the results with and without adding artificial viscosity. To summarize, the improvements of adding artificial viscosity include the following:

(1) The significant improvement of the multigrid convergence factors occurs approximately between $\mathbf{c} = (10, 10)^T$ and $(30, 30)^T$. By adding artificial viscosity, the multigrid factors are significantly reduced by 20%-40%.

(2) When $\mathbf{c}$ is close to $(0, 0)^T$, or when the problem is diffusion dominant, adding artificial viscosity does not have a significant impact on the multigrid factors, because the original direct discretization already yields good error estimations.

(3) When $\mathbf{c}$ is greater than $(40, 40)^T$, or when the problem is convection dominant, the multigrid factors have modest improvements if artificial viscosity is added. We note that as the convection increases, the ratios $(\frac{|c_1|h}{\sigma}, \frac{|c_2|h}{\sigma})$ also increase. The terms dropped out of (28) are no longer negligible.

(4) For $h = \frac{1}{64}$ and $h = \frac{1}{128}$, adding artificial viscosity yields approximately the same amount of improvement on the convergence factors.

The LFA provides an estimate on how many iterations are saved by adding artificial viscosity. Denote the convergence (or error reduction) factors with and without artificial viscosity as ρ_{yes} and ρ_{no} (or σ_{yes} and σ_{no}), respectively. Using the reported numbers in Table 3, the ratio of the numbers of iterations with and without artificial viscosity is given by $\log \rho_{\text{no}} / \log \rho_{\text{yes}} \approx 70\%$ (or $\log \sigma_{\text{no}} / \log \sigma_{\text{yes}} \approx 80\%$). That is, adding artificial viscosity may save 2-3 iterations per 10 iterations, or 20%-30% of computational cost. We will show in Sect. 7 that the numerical simulation agrees with this LFA estimate.

The Local Fourier Analysis only studies the convergence behavior of the multigrid algorithm on the linearized system. It would be desirable to provide a further convergence proof for the fully nonlinear FAS multigrid method. Some literature, such as [25], provides a theoretical framework for a convergence proof for FAS schemes. Unfortunately, devising a convergence proof for our FAS scheme for the mean field games is challenging. One reason is that the existing theoretical framework is based on a scalar PDE, whereas our FAS is developed for a system of PDEs; another reason is that the existing framework studies elliptic PDE, whereas our FAS is a fully spacetime scheme and applied on the entire parabolic system.

7 Numerical Results

In this section, we apply our proposed spacetime multigrid method to the joint HJB/KFP system (1)-(3), or equivalently, (14). We illustrate the fast and mesh-independent convergence rates.

7.1 Implementation Descriptions

Unless specified, we use the V(1,1)-cycle [37]. That is, we choose $\nu_1 = \nu_2 = \gamma = 1$ in Algorithm 2, or equivalently, we perform one pre and post smoothings respectively, and perform multigrid recursion only once on each coarse grid. We terminate the multigrid iterations at

Table 3 Multigrid asymptotic convergence factors and error reduction factors for $-\sigma\Delta m + \nabla\cdot(\mathbf{c}m) = 0$, where $\sigma = 1$. The results without artificial viscosity (the "No" rows) and with artificial viscosity (the "Yes" rows) are compared

(i) $h = \frac{1}{64}$, four-grid Artificial viscosity	$\mathbf{c} = (20, 20)^T$ $\rho_{loc}(M_h^{8h})$	 $\sigma_{loc}(M_h^{8h})$	$\mathbf{c} = (30, 30)^T$ $\rho_{loc}(M_h^{8h})$	 $\sigma_{loc}(M_h^{8h})$
No	0.34	0.40	0.43	0.47
Yes	**0.20**	**0.32**	**0.33**	**0.39**
(ii) $h = \frac{1}{128}$, five-grid				
Artificial viscosity	$\mathbf{c} = (20, 20)^T$		$\mathbf{c} = (30, 30)^T$	
	$\rho_{loc}(M_h^{16h})$	$\sigma_{loc}(M_h^{16h})$	$\rho_{loc}(M_h^{16h})$	$\sigma_{loc}(M_h^{16h})$
No	0.39	0.44	0.48	0.51
Yes	**0.23**	**0.35**	**0.38**	**0.43**

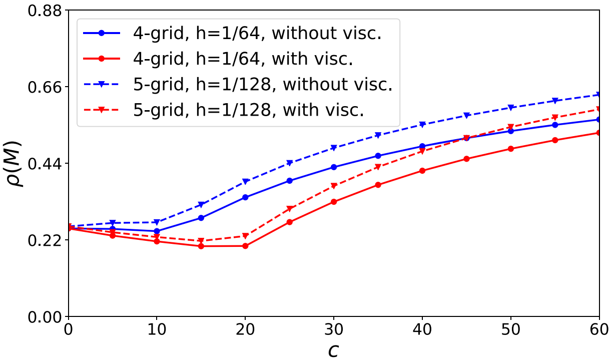


Fig. 3 The multigrid asymptotic convergence factors $\rho_{loc}(M)$ versus the convection coefficient $\mathbf{c}$ for the two-dimensional convection-diffusion equation $-\sigma\Delta m + \nabla\cdot(\mathbf{c}m) = 0$, where $\sigma = 1$ and $\mathbf{c} = (0, 0), (5, 5), (10, 10), \cdots$. The blue lines are the convergence factors *without* artificial viscosity, while the red lines are the corresponding convergence factors *with* artificial viscosity

the residual norm $\|r_h\| \leq 10^{-6}$. The initial guesses for the grid size (N_x, N_y, N_t) are the trilinear interpolation of the solutions from the grid size $(N_x/2, N_y/2, N_t/2)$.

In Examples 1 and 2, we compare the following multigrid schemes:

Scheme I (our proposed scheme) is the spacetime FAS scheme for the HJB/KFP system (14), where artificial viscosity is added to the direct discretization coarse grid operator. The number of iterations is counted.

Scheme II is the same as Scheme I, except that *no* artificial viscosity is added to the direct discretization coarse grid operator.

Scheme III is the spacetime FAS scheme, where we use the multigrid components proposed in [22]. More specifically, we apply the block Jacobi smoother, the coarsening strategy, the standard full-weighting restriction and trilinear interpolation proposed in [22]. We note that the scheme in [22] is developed for linear problems and cannot be directly applied to nonlinear problems. We use the FAS with direct discretization operators to adapt the scheme in [22] for the nonlinear HJB/KFP system. The number of iterations is counted.

Scheme IV is the outer-inner linearization-based spacetime multigrid method described in Algorithm 4. For each linearized problem, we use multigrid V(1,1)-cycle (inner loop). We use the proposed multigrid components described in Sect. 5. The only exception is that we use Petrov-Galerkin coarse grid operator, since it is accurate and not difficult to construct for the linearized system. The number of iterations is defined as the sum of the numbers of inner loops.

Scheme V is the forward/backward timestepping fixed point iteration described in Algorithm 3. Each timestep (Lines 4 and 7) is solved by the FAS scheme, using our proposed multigrid components described in Sect. 5. We note that the FAS here is applied on each timestep rather than on the entire spacetime. In order to make a fair comparison between Scheme V and the spacetime Schemes I-IV, we define the number of iterations for Scheme V as the average number of iterations per timestep, namely, $1/N_t \times$ sum of the FAS V-cycle counts over all the N_t timesteps.

For all these five multigrid schemes, we note that the computational costs per iteration are dominated by the costs of pre and post smoothings. In other words, the computational costs of restrictions and interpolations are negligible. Hence, the complexities per iteration are the same for Schemes I, II, IV and V. As a result, the number of iterations is a good measure for the complexity of each multigrid scheme. We note that for Scheme III, since block smoother is used, the complexity per iteration is more expensive than the other schemes.

In Example 7.4, we compare the numerical results by our proposed multigrid method with the results given by the numerical scheme in [1].

7.2 Example 1

Consider the (2+1)-dimensional mean field games in Sect. 5 of [24]. This is the same as solving (1–3), where the spacetime domain is $T = 1$, $\Omega = [0, 1]^2$, the cost function is $g(m) = \ln(m)$, and the Lagrangian is $L(\mathbf{c}) = -\frac{\|\mathbf{c}\|^2}{2}$. Section 5 of [24] derives the following exact solution:

$$u = -a(x^2 + y^2) + b, \quad m = \frac{a}{\pi\sigma} \exp\left(-\frac{a(x^2 + y^2)}{\sigma}\right), \tag{33}$$

where $a = \frac{1}{2\sigma} - \frac{\rho}{2}$, $b = \frac{1}{\rho}\left(\ln\frac{a}{\pi\sigma} - 4a\sigma\right)$. Here we impose Dirichlet boundary conditions for both u and m, and let the terminal, initial and boundary conditions be the same as (33). We set the discount factor as $\rho = 0.1$. We note that the convection, given by $\mathbf{c}^* = -\nabla u = 2a(x, y)^T$, is proportional to a. We test the following two cases.

Case 1: $(\sigma, a) = (1, 0.45)$, which is diffusion dominant. Table 4(i) reports the convergence rates of the numerical solutions towards the exact solution (33). The convergence rates of $\|u - u_h\|$ and $\|m - m_h\|$ are first order, namely, $O(h)$.

We then investigate the convergence rates of the five multigrid schemes; see Table 4(i). Scheme I takes only 4 iterations to converge. In addition, the CPU time for Scheme I is approximately linear to the grid size, namely $O(N_x N_y N_t)$. The convergence rates of Scheme II are basically the same as Scheme I. The reason is that the problem is diffusion dominant, and thus direct discretization without artificial viscosity yields sufficiently good coarse grid error estimations. Scheme I converges faster than Scheme III. Figure 5 explains the reason. More specifically, Scheme I's pre-smoothed error (blue lines) is smoother than Scheme III's, which leads to a more precise coarse grid estimate (red lines), a smaller post-smoothed error (purple lines), a more efficient error reduction, and eventually a faster convergence. In addition, the cost of Scheme I's pointwise smoother is only $O(N_x N_y N_t)$, while the cost of

Table 4 Example 7.2: Convergence of the five multigrid schemes. **(i)** $(\sigma, a) = (1, 0.45)$. **(ii)** $(\sigma, a) = (0.2, 2.45)$

N_xxN_yxN_t	Error of HJB/KFP $\|u - u_h\|$	$\|m - m_h\|$	Number of iterations (CPU time) **Scheme I**	Scheme II	Scheme III	Scheme IV	Scheme V
(i) $(\sigma, a) = (1, 0.45)$							
16x16x16	1.11×10^{-3}	2.88×10^{-4}	**4 (3.0s)**	4 (2.9s)	12 (18s)	28 (8.3s)	21 (8.4s)
32x32x32	5.73×10^{-4}	1.49×10^{-4}	**4 (23s)**	4 (24s)	11 (219s)	28 (74s)	21 (42s)
64x64x64	2.91×10^{-4}	7.61×10^{-5}	**4 (193s)**	4 (188s)	11 (3404s)	21 (491s)	19 (245s)
128x128x128	1.47×10^{-4}	3.84×10^{-5}	**3 (1104s)**	4 (1686s)	11 (39616s)	15 (3507s)	19 (2125s)
(ii) $(\sigma, a) = (0.2, 2.45)$							
16x16x16	3.68×10^{-2}	4.17×10^{-2}	**6 (5.0s)**	7 (5.9s)	18 (29s)	64 (32s)	69 (24s)
32x32x32	2.65×10^{-2}	2.46×10^{-2}	**7 (45s)**	10 (66s)	23 (474s)	75 (340s)	78 (137s)
64x64x64	1.70×10^{-2}	1.35×10^{-2}	**7 (349s)**	∞	∞	86 (3145s)	80 (1032s)
128x128x128	9.73×10^{-3}	7.04×10^{-3}	**7 (2554s)**	∞	∞	96 (29381s)	77 (9400s)

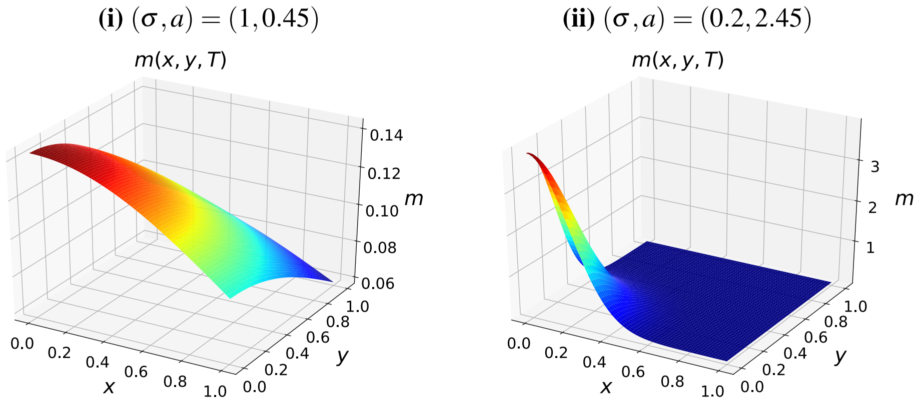
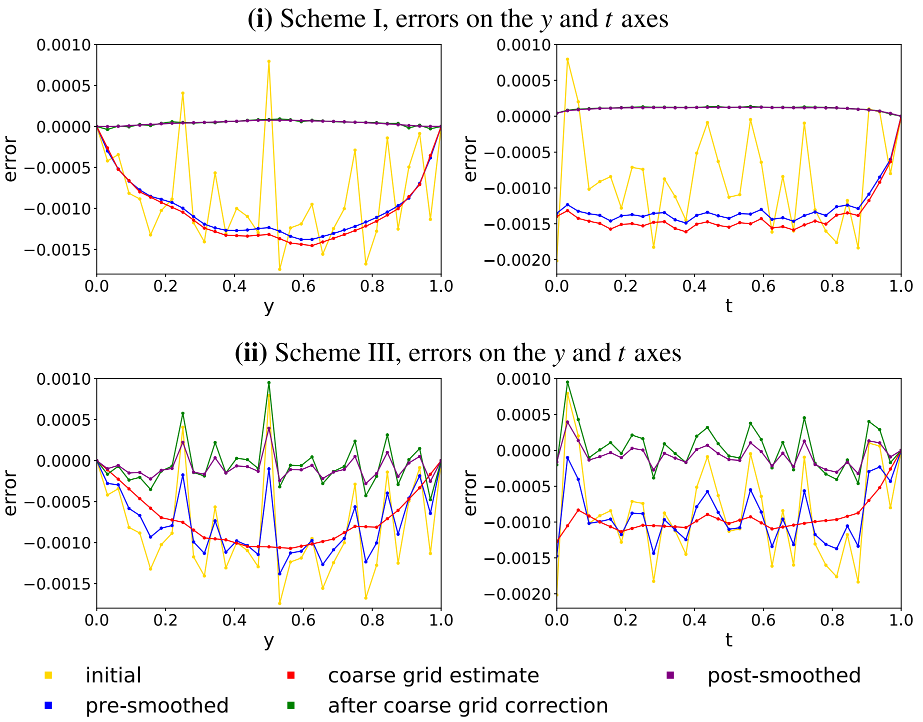


Fig. 4 Example 7.2: Numerical solutions $m_h(x, y, T)$. **(i)** $(\sigma, a) = (1, 0.45)$. **(ii)** $(\sigma, a) = (0.2, 2.45)$

Scheme III's block smoother is $O(\min(N_x^2, N_y^2)N_x N_y N_t)$. We note that Schemes IV and V are both outer-inner iterations, where the total iteration count is the number of outer iterations times the average number of inner iterations; conversely, Scheme I is a single-layer iteration, where the total iteration count is the FAS iteration count itself. The inner iteration count of Schemes IV and V is approximately equal to the iteration count of Scheme I. As a result, the total iteration count of Schemes IV and V is much higher than Scheme I. Comparing with Schemes III-V, we conclude that the proposed Scheme I has the fastest convergence rates.

Case 2: $(\sigma, a) = (0.2, 2.45)$, which is convection dominant. Table 4(ii) shows that the numerical solutions converge to the exact solution as $h \to 0$. We note that $m(1, 1, t) \approx 0$; see Fig. 4(ii). Hence in the HJB equation (1), $\ln(m(1, 1, t)) \approx -\infty$. Due to this singularity, the convergence rates of $\|u - u_h\|$ and $\|m - m_h\|$ are slightly slower than $O(h)$.

Table 4(ii) reports the convergence rates of the five multigrid schemes. The numbers of iterations for $(\sigma, a) = (0.2, 2.45)$ are larger than those for $(\sigma, a) = (1, 0.45)$. The reason is that multigrid is usually less efficient when the problem becomes more hyperbolic. Significantly, Scheme I converges in 6-7 iterations, independent of the mesh size. The total complexity (reflected by the total CPU time) is linear to the grid size, namely $O(N_x N_y N_t)$. By comparing Schemes I and II, we see that adding artificial viscosity saves 14% and 30% of iterations when the mesh size is 16x16x16 and 32x32x32, respectively. This agrees with the LFA estimate, which is around 20%-30% (see the end of Sect. 6.3). In particular, Scheme II fails to converge when the mesh size is larger than 64x64x64. However, Scheme I successfully converges in only 7 iterations, and the convergence rate is mesh-independent. Scheme III fails to converge. The reason, which has been explained in Fig. 5, is that Scheme III's errors are oscillatory. The oscillation grows quickly as the iteration proceeds. For Scheme IV, the number of iterations grows as the grid becomes larger. The reason is that the number of outer iterations (policy iterations) grows, and is not mesh-independent. In addition, the number of iterations for Scheme IV is above 64, much larger than Scheme I. Similarly, Scheme V is also slower than Scheme I, since it is also an outer-inner iteration.



(i) Scheme I, errors on the y and t axes

(ii) Scheme III, errors on the y and t axes

Fig. 5 Comparison of the error reductions between Scheme I and Scheme III. Here we consider the errors of the HJB equation in Example 7.2 under one multigrid V-cycle. Cross sections of the errors along the y and t axes are plotted. The initial errors are the same. Scheme I's error reduction is faster than Scheme III's

7.3 Example 2

We consider solving (4–6) in Section 2.2. We first verify the correctness of our nonlinear solver by a simulation under the parameters in [15], i.e. $\rho = 0.2$, $s = 1$, $\sigma = 0$ or 0.005[2], $\epsilon = 0$ or 0.3, $T = 5$. Figure 6 shows the plots of the initial value function $u(x, 0)$ and the number of remaining players $\eta(t)$. The plots given by our numerical results are the same as those in [15]. We note that unlike Example 7.2, here analytical solution is unavailable; numerical solution is only provided as figures in [15], and thus it is impossible to compare our solution with the solution in [15] using some norms. Interested readers are referred to [15] for an explanation on the economic meaning of the solution.

Reference [15] does not discuss fast and efficient solvers for (4–6). Here we test the five multigrid schemes. We restrict the choice of parameters to $\sigma = 0.005$, $\epsilon = 0.3$, $T = 5$, and test two cases. One is $(\rho, s) = (0.2, 1)$, which is relatively diffusive, as $\frac{D}{\sigma} \sim \frac{s}{\sigma}$ is small; the other is $(\rho, s) = (0.02, 10)$, which is relatively convective, as $\frac{D}{\sigma} \sim \frac{s}{\sigma}$ is large. We note that since the demand function D is always non-negative, we simply use the low-cost one-direction downstream GS smoother rather than multi-direction smoother for our proposed Scheme I. The convergence rates are reported in Table 5. All the methods have mesh-independent convergence rates. However, our proposed Scheme I has the fastest convergence rates, which is around 7 iterations for the diffusive case and 9 iterations for the convective case. The comparison between Schemes I and II shows that adding artificial viscosity saves 2-4

[2] $\sigma = 0.005$ in Equation (4)-(6) corresponds to $\sigma = 0.1$ (namely $\sigma^2/2 = 0.005$) in [15].

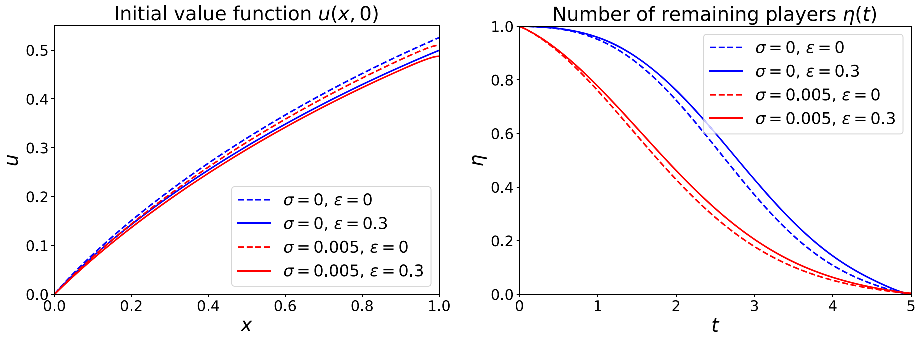


Fig. 6 Example 7.3: Initial value function $u(x, 0)$ and number of remaining players $\eta(t)$ for different (σ, ϵ). The plots given by our numerical solutions are the same as those in [15]

Table 5 Example 7.3: Convergence of the five multigrid schemes. $\sigma = 0.005$, $\epsilon = 0.3$, $T = 5$. **(i)** $(\rho, s) = (0.2, 1)$. **(ii)** $(\rho, s) = (0.02, 10)$. Note that for Scheme III, the numbers in the parentheses are the iteration counts if the numbers of pre and post smoothings are changed from (1, 1) to (2, 2)

N_xxN_t	Number of iterations				
	Scheme I	Scheme II	Scheme III	Scheme IV	Scheme V
(i) $(\rho, s) = (0.2, 1)$					
64x64	**6**	7	12	29	17
128x128	**7**	7	12	35	19
256x256	**7**	8	13	40	20
512x512	**7**	8	14	46	20
(ii) $(\rho, s) = (0.02, 10)$					
64x64	**9**	11	∞ (11)	38	19
128x128	**9**	11	∞ (11)	44	21
256x256	**9**	12	∞ (11)	50	21
512x512	**9**	13	∞ (11)	54	22

iterations, which is consistent with the LFA estimate in Sect. 6.3. We note that to make Scheme III converge for the convective case, we change the numbers of pre and post smoothings from (1, 1) to (2, 2), as shown in Table 5(ii). However, this doubles the computational cost per iteration and yet the convergence rate is still slower than Scheme I.

7.4 Example 3

Consider the example in Sect. 4.2 of [1]. This is the same as solving (1)-(3), where the spacetime domain is $T = 1$, $\Omega = [-0.5, 0.5]^2$, the cost function is $g(m) = m$, the terminal condition is $u(x, y, T) = 0$, the initial condition is $m(x, y, 0) = 1$, the boundary conditions for both u and m are periodic, and the Lagrangian is $L(\mathbf{c}) = \frac{2\sqrt{3}}{9}\|\mathbf{c}\|^{3/2} - \sin(2\pi x) - \sin(2\pi y) - \cos(4\pi x)$. Straightforward algebra can show that under this Lagrangian, the optimal control is $\mathbf{c}^* = 3\|\nabla u\|\nabla u$. This yields the same Hamiltonian $H(\mathbf{x}, \nabla u) = \|\nabla u\|^3 + \sin(2\pi x) + \sin(2\pi y) + \cos(4\pi x)$ as the one in [1].

Table 6 Example 7.4: Convergence of our proposed multigrid schemes. Total number of iterations is counted

N_xxN_yxN_t	Total number of iterations				
	$\sigma = 0.6$	$\sigma = 0.36$	$\sigma = 0.2$	$\sigma = 0.12$	$\sigma = 0.046$
32x32x32	**4**	**4**	**5**	**6**	**6**
64x64x64	**4**	**5**	**5**	**7**	**8**
128x128x128	**4**	**5**	**5**	**7**	**8**

Table 7 Average (on the Newton loop) numbers of BiCGstab iterations given by the algorithm in [1]. We note that the total numbers of iterations of the algorithm = the numbers of Newton loops × the numbers of BiCGstab iterations, which can be much higher than the numbers listed in the table

N_xxN_yxN_t	Average numbers of BiCGstab iterations per Newton loop				
	$\sigma = 0.6$	$\sigma = 0.36$	$\sigma = 0.2$	$\sigma = 0.12$	$\sigma = 0.046$
32x32x32	2	2	3.5	6	10
64x64x64	2	2	3.5	6	10
128x128x64	2	2	4	6.1	10

We test our proposed multigrid method under different diffusion parameters; see Table 6. When $\sigma = 0.12, 0.046$, the problem is convection dominant and poses a challenge to multigrid methods. When the spatial grid is coarser than 8×8, or $h \geq \frac{1}{8}$, the convection-diffusion ratios $(\frac{|c_1|h}{\sigma}, \frac{|c_2|h}{\sigma})$ turn out to be much larger than 1. As discussed in Sect. 5.8, artificial viscosity is efficient under the assumption that $(\frac{|c_1|h}{\sigma}, \frac{|c_2|h}{\sigma})$ are not much larger than 1. To further improve the error estimations, we consider using W-cycle on the coarsest grids where $N_x \times N_y$ is coarser than 8×8. W-cycle is more expensive than V-cycle. However, we emphasize that for the grids where $N_x \times N_y$ is finer than 8×8, we still apply V-cycle. Since W-cycle is only applied on the very coarsest grids, the extra computational cost is negligible. In addition, for the KFP equation, we use the finite volume Scharfetter-Gummel method [8].

Table 6 shows that our proposed multigrid method converges in around 4-8 iterations; the convergence rates are approximately mesh-independent.

Table 7 shows the convergence rates of the algorithm in [1]. We note that the total number of iterations can be much higher than the numbers listed in the table. The reason is that the total number of iterations is the product of the number of the outer Newton's iterations and the number of the inner BiCGstab iterations. The number of Newton's iterations is not reported in [1]. If the number of Newton's iterations is greater than 2, then the total number of iterations can be higher than our proposed method. In addition, each Newton-BiCGstab iteration requires solving linear systems of size $N_t N_x N_y \times N_t N_x N_y$, which is expensive. Another observation of Table 7 is that as the diffusion decreases, the convergence rates of the algorithm in [1] deteriorate quickly from 2 to 10. However, using our proposed multigrid method, the increase of iteration counts is modest, from 4 to 8; see Table 6.

8 Conclusion

We propose a joint spacetime multigrid method for the HJB and KFP system arising from mean field games. We propose a nonlinear FAS scheme, which requires only one layer of iteration (as opposed to outer-inner iterations). We use a hybrid full-semi coarsening and kernel preserving biased restriction operator to treat the anisotropy in time and the convection in space properly. We propose adding artificial viscosity to improve the precision of the coarse grid error estimation using direct discretization. These properties are supported by our Fourier analysis. The resulting multigrid method converges at the mesh-independent rate and at a faster rate than the other approaches considered in Sect. 7.

We note that artificial viscosity is efficient under the assumption that $(\frac{|c_1|h}{\sigma}, \frac{|c_2|h}{\sigma})$ are not much larger than 1. This may not be true if the convection is extremely large or if the diffusion is extremely small. In Sect. 7.4, we have seen that applying W-cycle on the coarsest grids is a good remedy in practice. However, it is desirable to find a more elegant solution, where V-cycle can be applied on the coarsest grids and still yields an effective multigrid scheme. Extending our proposed multigrid method to extremely large convection or extremely small diffusion cases, and providing a complete convergence proof for our FAS algorithm, would be an interesting future work.

Declarations

Researh The research presented in this paper was funded by Natural Sciences and Engineering Research Council of Canada (NSERC).

A Pseudo-code for the Associated Algorithms

Algorithm 3 Forward/backward timestepping fixed point iteration

```
Start with an initial guess $(u_h^n)^{(0)}$ and $(m_h^n)^{(0)}$ for $n = 0, 1, \cdots, N_t$.
for $k = 1, 2, ...$ until convergence do
    for $n = N_t - 1, \cdots, 1, 0$ do
        Solve $\max_{\mathbf{c}_h^n}\left[A^n_{HJB}(\mathbf{c}^n_h)(u^n_h)^{(k)} - L(\mathbf{c}^n_h)\right] = 1/\Delta t \cdot (u^{n+1}_h)^{(k)} + g((m^n_h)^{(k-1)})$ for $(u^n_h)^{(k)}$ and $(\mathbf{c}^n_h)^{(k)} = \operatorname*{argmax}_{\mathbf{c}^n_h}\left[A^n_{HJB}(\mathbf{c}^n_h)(u^n_h)^{(k)} - L(\mathbf{c}^n_h)\right]$.
    end for
    for $n = 1, 2, \cdots, N_t$ do
        Solve $A^n_{KFP}((\mathbf{c}^n_h)^{(k)})(m^n_h)^{(k)} = 1/\Delta t \cdot (m^{n-1}_h)^{(k)}$ for $(m^n_h)^{(k)}$.
    end for
end for
```

Algorithm 4 Outer-inner linearization-based spacetime multigrid methods under policy iterations

```
Start with an initial guess $(u_h^{(0)}, m_h^{(0)})$.
for $k = 1, 2, ...$ until convergence (outer loop) do
    $\mathbf{c}_h^{(k)} = \operatorname*{argmax}_{\mathbf{c}_h}\left[A_{HJB}(\mathbf{c}_h)u_h^{(k-1)} - L(\mathbf{c}_h)\right]$.
    $u_h^{(k;0)} = u_h^{(k-1)}, m_h^{(k;0)} = m_h^{(k-1)}$.
    for $l = 1, 2, ...$ until convergence (inner loop) do
        Compute one multigrid V-cycle for the linearized KFP equation $A_{KFP}(\mathbf{c}_h^{(k)})m_h = b_{KFP}$, which updates the solution $m_h^{(k;l-1)} \to m_h^{(k;l)}$.
        Compute one multigrid V-cycle for the linearized HJB equation $A_{HJB}(\mathbf{c}_h^{(k)})u_h = b_{HJB}(\mathbf{c}_h^{(k)}, m_h^{(k;l)})$, which updates the solution $u_h^{(k;l-1)} \to u_h^{(k;l)}$.
    end for
    $u_h^{(k)} = u_h^{(k;l)}, m_h^{(k)} = m_h^{(k;l)}$.
end for
```

References

1. Achdou, Y.: Finite difference methods for mean field games. In: Hamilton-Jacobi equations: approximations, numerical analysis and applications, *Lecture Notes in Math.*, vol. 2074, pp. 1–47. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36433-4_1
2. Achdou, Y., Camilli, F., Capuzzo-Dolcetta, I.: Mean field games: numerical methods for the planning problem. SIAM J. Control Optim. **50**(1), 77–109 (2012). https://doi.org/10.1137/100790069
3. Achdou, Y., Camilli, F., Capuzzo-Dolcetta, I.: Mean field games: convergence of a finite difference method. SIAM J. Numer. Anal. **51**(5), 2585–2612 (2013). https://doi.org/10.1137/120882421
4. Achdou, Y., Capuzzo-Dolcetta, I.: Mean field games: numerical methods. SIAM J. Numer. Anal. **48**(3), 1136–1162 (2010). https://doi.org/10.1137/090758477
5. Achdou, Y., Perez, V.: Iterative strategies for solving linearized discrete mean field games systems. Netw. Heterog. Media **7**(2), 197–217 (2012). https://doi.org/10.3934/nhm.2012.7.197
6. Andreev, R.: Preconditioning the augmented Lagrangian method for instationary mean field games with diffusion. SIAM J. Sci. Comput. **39**(6), A2763–A2783 (2017). https://doi.org/10.1137/16M1072346
7. Briceño Arias, L.M., Kalise, D., Silva, F.J.: Proximal methods for stationary mean field games with local couplings. SIAM J. Control Optim. **56**(2), 801–836 (2018). https://doi.org/10.1137/16M1095615
8. Bank, R.E., Coughran Jr., W., Cowsar, L.C.: The finite volume scharfetter-gummel method for steady convection diffusion equations. Comput. Vis. Sci. **1**(3), 123–136 (1998)
9. Bank, R.E., Wan, J.W.L., Qu, Z.: Kernel preserving multigrid methods for convection-diffusion equations. SIAM J. Matrix Anal. Appl. **27**(4), 1150–1171 (2006). https://doi.org/10.1137/040619533
10. Brandt, A.: Multi-level adaptive solutions to boundary-value problems. Math. Comp. **31**(138), 333–390 (1977)
11. Brandt, A., Yavneh, I.: On multigrid solution of high-Reynolds incompressible entering flows. J. Comput. Phys. **101**(1), 151–164 (1992). https://doi.org/10.1016/0021-9991(92)90049-5
12. Briceño-Arias, L., Kalise, D., Kobeissi, Z., Laurière, M., González, Á.M., Silva, F.J.: On the implementation of a primal-dual algorithm for second order time-dependent mean field games with local couplings. arXiv preprint arXiv:1802.07902 (2018)
13. Carlini, E., Silva, F.J.: A semi-Lagrangian scheme for a degenerate second order mean field game system. Dis. Contin. Dyn. Syst. **35**(9), 4269–4292 (2015). https://doi.org/10.3934/dcds.2015.35.4269
14. Carlini, E., Silva, F.J.: On the discretization of some nonlinear Fokker-Planck-Kolmogorov equations and applications. SIAM J. Numer. Anal. **56**(4), 2148–2177 (2018). https://doi.org/10.1137/17M1143022
15. Chan, P., Sircar, R.: Bertrand and cournot mean field games. Appl. Math. Optim. **71**(3), 533–569 (2015)
16. Chen, L., Jakobsen, E.R., Naess, A.: On numerical density approximations of solutions of SDEs with unbounded coefficients. Adv. Comput. Math. **44**(3), 693–721 (2018). https://doi.org/10.1007/s10444-017-9558-4
17. Djehiche, B., Tcheukam, A., Tembine, H.: Mean-field-type games in engineering. arXiv preprint arXiv:1605.03281 (2016)
18. Forsyth, P.A., Labahn, G.: Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. J. Comput. Fin. **11**(2), 1 (2007)
19. Franca, L.P., Frey, S.L., Hughes, T.J.R.: Stabilized finite element methods. I. Application to the advective-diffusive model. Comput. Methods Appl. Mech. Engrg. **95**(2), 253–276 (1992). https://doi.org/10.1016/0045-7825(92)90143-8
20. Friedhoff, S., MacLachlan, S.: A generalized predictive analysis tool for multigrid methods. Num. Linear Algebra Appl. **22**(4), 618–647 (2015)
21. Gander, M.J.: 50 years of time parallel time integration. In: Multiple Shooting and Time Domain Decomposition Methods, pp. 69–113. Springer (2015)
22. Gander, M.J., Neumüller, M.: Analysis of a new space-time parallel multigrid algorithm for parabolic problems. SIAM J. Sci. Comput. **38**(4), A2173–A2208 (2016). https://doi.org/10.1137/15M1046605
23. Graber, P.J., Bensoussan, A.: Existence and uniqueness of solutions for Bertrand and Cournot mean field games. Appl. Math. Optim. **77**(1), 47–71 (2018). https://doi.org/10.1007/s00245-016-9366-0
24. Guéant, O., Lasry, J.M., Lions, P.L.: Mean field games and applications. In: Paris-Princeton lectures on mathematical finance 2010, pp. 205–266. Springer (2011)
25. Hackbusch, W.: Multi-grid methods and applications, vol. 4. Springer Science & Business Media (2013)
26. Han, D., Wan, J.W.L.: Multigrid methods for second order Hamilton-Jacobi-Bellman and Hamilton-Jacob-Bellman-Isaacs equations. SIAM J. Sci. Comput. **35**(5), S323–S344 (2013). https://doi.org/10.1137/120881476
27. Horton, G., Vandewalle, S.: A space-time multigrid method for parabolic partial differential equations. SIAM J. Sci. Comput. **16**(4), 848–864 (1995). https://doi.org/10.1137/0916050

28. Howard, R.A.: Dynamic programming and Markov processes. The Technology Press of M.I.T., Cambridge, Mass.; John Wiley & Sons, Inc., New York-London (1960)
29. Hughes, T.J., Franca, L.P., Hulbert, G.M.: A new finite element formulation for computational fluid dynamics: Viii. the galerkin/least-squares method for advective-diffusive equations. Comput. Methods Appl. Mech. Eng. **73**(2), 173–189 (1989)
30. Lachapelle, A., Salomon, J., Turinici, G.: Computation of mean field equilibria in economics. Math. Models Methods Appl. Sci. **20**(4), 567–588 (2010). https://doi.org/10.1142/S0218202510004349
31. Lasry, J.M., Lions, P.L.: Jeux à champ moyen. I-le cas stationnaire. Comptes Rendus Mathématique **343**(9), 619–625 (2006)
32. Lasry, J.M., Lions, P.L.: Jeux à champ moyen. II–horizon fini et contrôle optimal. Comptes Rendus Mathématique **343**(10), 679–684 (2006)
33. Lasry, J.M., Lions, P.L.: Mean field games. Jpn. J. Math. **2**(1), 229–260 (2007). https://doi.org/10.1007/s11537-007-0657-8
34. Lax, P., Wendroff, B.: Systems of conservation laws. Comm. Pure Appl. Math. **13**, 217–237 (1960). https://doi.org/10.1002/cpa.3160130205
35. Rhebergen, S., Cockburn, B., van der Vegt, J.J.W.: A space-time discontinuous Galerkin method for the incompressible Navier-Stokes equations. J. Comput. Phys. **233**, 339–358 (2013). https://doi.org/10.1016/j.jcp.2012.08.052
36. Toro, E.F.: Riemann solvers and numerical methods for fluid dynamics: a practical introduction. Springer Science & Business Media (2013)
37. Trottenberg, U., Oosterlee, C.W., Schüller, A.: Multigrid. Academic Press, Inc., San Diego, CA (2001). With contributions by A. Brandt, P. Oswald and K. Stüben
38. Wienands, R., Oosterlee, C.W.: On three-grid Fourier analysis for multigrid. SIAM J. Sci. Comput. **23**(2), 651–671 (2001)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.