

CEMRACS 2017: NUMERICAL PROBABILISTIC APPROACH TO MFG

ANDREA ANGIULI¹, CHRISTY V. GRAVES², HOUZHI LI³, JEAN-FRANÇOIS
CHASSAGNEUX³, FRANÇOIS DELARUE⁴ AND RENÉ CARMONA⁵

Abstract. This project investigates numerical methods for solving fully coupled forward-backward stochastic differential equations (FBSDEs) of McKean-Vlasov type. Having numerical solvers for such mean field FBSDEs is of interest because of the potential application of these equations to optimization problems over a large population, say for instance mean field games (MFG) and optimal mean field control problems. Theory for this kind of problems has met with great success since the early works on mean field games by Lasry and Lions, see [29], and by Huang, Caines, and Malhamé, see [26]. Generally speaking, the purpose is to understand the continuum limit of optimizers or of equilibria (say in Nash sense) as the number of underlying players tends to infinity. When approached from the probabilistic viewpoint, solutions to these control problems (or games) can be described by coupled mean field FBSDEs, meaning that the coefficients depend upon the own marginal laws of the solution. In this note, we detail two methods for solving such FBSDEs which we implement and apply to five benchmark problems. The first method uses a tree structure to represent the pathwise laws of the solution, whereas the second method uses a grid discretization to represent the time marginal laws of the solutions. Both are based on a Picard scheme; importantly, we combine each of them with a generic continuation method that permits to extend the time horizon (or equivalently the coupling strength between the two equations) for which the Picard iteration converges.

1. INTRODUCTION

In this project, we examine numerical methods for solving forward backward stochastic differential equations (FBSDEs) of McKean-Vlasov type. We are particularly interested in equations of this type as they can be used to represent, from the probabilistic viewpoint, solutions to mean field games or, more generally, to mean field stochastic control problems.

Mean field games were developed independently and at about the same time by Lasry and Lions [29], and Huang, Caines, and Malhamé [26]. The goal of this theory is to understand the limit as $N \rightarrow \infty$ of the Nash equilibria for an N player stochastic dynamic game with mean field interaction, meaning that players interact with one another through their collective state. Equivalently, mean field games must be regarded as the continuum limit of games with a large number of symmetric players, each of them having a small effect on the dynamics of the whole group. The applications of mean field games are numerous, and spread across many disciplines, including social science (congestion [3, 21, 28]), cyber attacks [14]), biology (flocking [30, 31]),

¹ Department of Statistics and Applied Probability, University of California Santa Barbara

² Program in Applied and Computational Mathematics, Princeton University

³ Laboratoire de Probabilités, Modélisation et Statistiques, Université Paris Diderot

⁴ Laboratoire Jean-Alexandre Dieudonné, Université de Nice Sophia-Antipolis

⁵ Operations Research and Financial Engineering, Bendheim Center for Finance, Princeton University

jet lag [13]), and economics (systemic risk [12], production of exhaustible resources [15, 25]), just to name a few. As explained in [9, 10], solutions to mean field games can be characterized through a coupled system of two forward and backward stochastic differential equations of mean field type, like those we address in this note. The forward equation provides the dynamics of one typical player in the population at equilibrium. Generally speaking, the backward equation accounts for the optimality condition in the definition of an equilibrium and the McKean-Vlasov structure is precisely here to stress the fact that the player in hand is representative of the others. As exemplified in [11], other forms of equilibria can be addressed by means of this kind of equations. This includes optimal mean field control problems, which can be regarded as the continuum limit of a control problem involving a large number of symmetric players obeying a central planner. Below, we mostly focus on examples arising in the theory of mean field games.

Whilst deterministic numerical methods, based upon finite differences or variational approaches, are also conceivable for handling mean field games, see [1, 2, 4, 7] and [5, 24, 27], we here focus on the approach based on FBSDEs. In this regard, we implement (and compare) two different algorithms. The first algorithm, which is based on the paper of Chassagneux, Crisan, and Delarue [17], relies on a tree structure to represent the pathwise law of the solution. The second algorithm, and main contribution of this report, takes the algorithm presented in the paper of Delarue and Menozzi [20] for solving FBSDEs and extends it to the mean field framework in hand. In this algorithm, a grid structure is used to represent the marginal laws of the solution. The serious issue that we are facing in this note is that both methods are based upon a Picard scheme, the first method involving a global Picard scheme upon the whole process and the second one involving a Picard scheme on the sole marginal laws of the process. It is indeed a well-known fact that, because of the strong coupling between the forward and backward equations, Picard schemes for FBSDEs may just converge in small time, even in the classical case without mean field interaction. For sure, this limitation should persist in the mean field setting for the global Picard method; as exemplified below, it turns out that it persists as well when the Picard scheme is applied to the marginal laws. One of our main contribution in this report is to apply the time continuation approach presented in [17] to the grid algorithm and to compare the results with the tree algorithm for which the time continuation approach was originally designed in [17]. In brief, the time continuation permits to extend, by a continuation argument, the time interval on which the Picard scheme converges. We illustrate both algorithms on a handful of example problems.

Section 2 provides a review of Nash equilibria in N player stochastic differential games, and their continuum mean field game counterparts. We review two probabilistic approaches to formulate the solutions of mean field games and provide the general FBSDE system which we would like to solve. In Section 3, we describe the algorithms that we implement in the report. Some benchmark examples and the corresponding numerical results are presented in Section 4. We conclude in Section 5.

2. OVERVIEW OF MEAN FIELD GAMES AND FBSDEs

The purpose of this section is to introduce the theoretical material that is needed for our numerical analysis. The objective is purely pedagogical and the text does not contain any new result.

2.1. N Player Stochastic Differential Games

We start with the description of the prototype of a finite player game in the theory of mean field games. We consider $N \in \mathbb{Z}^+$ players indexed by $i \in \{1, \dots, N\}$. The dynamic game occurs over a fixed time horizon $[0, T]$ for some $T > 0$. We have N independent m -dimensional Brownian motions $(W_t^i)_{0 \leq t \leq T}$ which are supported by a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F} = (\mathcal{F}_t)_{0 \leq t \leq T}, \mathbb{P})$. Each player chooses its control $\alpha^i = (\alpha_t^i)_{0 \leq t \leq T}$ from the set \mathbb{A} defined as the set of square integrable \mathbb{F} adapted processes with values in a given set A (typically A is a closed convex subset of a Euclidean space). Each player i has a state X^i which evolves according to the stochastic differential equation:

$$dX_t^i = b^i(t, X_t^i, \bar{\mu}_t, \alpha_t)dt + \sigma^i(t, X_t^i, \bar{\mu}_t, \alpha_t)dW_t^i,$$

where $\bar{\mu}_t$ denotes the empirical distribution of the players' states: $\bar{\mu}_t = \frac{1}{N} \sum_{j=0}^N \delta_{X_t^j} \in \mathcal{P}_2(\mathbb{R}^d)$. Here $\mathcal{P}_2(\mathbb{R}^d)$ is the space of probability measures with a finite second moment, which we equip with the 2-Wasserstein distance, denoted by W_2 . For $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$, we call $\Gamma(\mu, \nu)$ the set of all the joint laws with marginals μ and ν . Then, the 2-Wasserstein distance is defined by:

$$W_2(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu, \nu)} \int |x - y|^2 d\gamma(x, y) \right)^{1/2}.$$

The drift and volatility functions, b^i and σ^i , respectively, are deterministic functions $(b^i, \sigma^i) : [0, T] \times \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times A \rightarrow \mathbb{R}^d \times \mathbb{R}^{d \times m}$. Most of the time, they are assumed to be bounded in time and to be Lipschitz continuous with respect to all the arguments, the Lipschitz property in the measure argument being understood with respect to W_2 . This ensures that, for a given $\alpha = (\alpha^1, \dots, \alpha^N)$, the state dynamics (X^1, \dots, X^N) are well defined.

Given a tuple of controls $\alpha = (\alpha^1, \dots, \alpha^N)$, we associate with player i a cost objective which we take to be of the form:

$$J^i(\alpha) = \mathbb{E} \left[\int_0^T f^i(t, X_t^i, \bar{\mu}_t, \alpha_t^i) dt + g^i(X_T^i, \bar{\mu}_T) \right].$$

Thus each player considers a deterministic running cost $f^i : [0, T] \times \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times A \rightarrow \mathbb{R}$, and deterministic terminal cost $g^i : \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \rightarrow \mathbb{R}$. Of course, each of them wishes to minimize its own cost by tuning its own control in the most relevant way. Note that we only allow the interaction of the players through their empirical measure, as this will be needed in our formulation of the continuum limit. Still, extensions exist, in which players also interact through the controls, see Subsection 2.2.3.

The players are in a *Nash equilibrium* if each player is no better off for switching their strategy when they consider the other players' strategies to be fixed. More precisely, the set of strategies α is a Nash equilibrium if

$$J^i(\alpha) \leq J^i(\alpha^1, \dots, \alpha^{i-1}, \alpha, \alpha^{i+1}, \dots, \alpha^N), \forall \alpha \in \mathbb{A}, \forall i \in \{1, \dots, N\}.$$

2.2. Mean Field Games

For games where N is large, the problem quickly becomes of an intractable complexity. Thus we turn to the continuum limit by considering the limit as N tends to infinity. In order for this limit to make sense, we require the players to be symmetric. Precisely, we require $b = b^i$, $\sigma = \sigma^i$, $f = f^i$, and $g = g^i \forall i \in \{1, \dots, N\}$. As the number of players increase, the impact of each player on the empirical distribution decreases, and we expect to have a propagation of chaos such that the players become asymptotically independent of each other. This is the rationale for passing to the limit: Asymptotically, the influence of one player on the group should be null and the statistical structure of the whole should be pretty simple.

We wish to formulate the analogue of a Nash equilibrium when there is a continuum of players. To this end, we consider the states and actions of the other players to be fixed, and consider the best response for a representative player (as we expect equilibria to inherit the symmetric structure of the game). Thus, the first step is to solve an optimization problem. The next step is to find a fixed point, providing an analogue of a Nash equilibrium for the mean field game.

We again have a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F} = (\mathcal{F}_t)_{0 \leq t \leq T}, \mathbb{P})$ where the filtration supports an m -dimensional Brownian motion $W = (W_t)_{0 \leq t \leq T}$ and an initial condition $\xi \in L^2(\Omega, \mathcal{F}_0, \mathbb{P}; \mathbb{R}^d)$.

The strategy for solving the asymptotic game is the following:

- (1) For a fixed deterministic flow of probability measures $\mu = (\mu_t)_{0 \leq t \leq T} \in \mathcal{C}([0, T], \mathcal{P}_2(\mathbb{R}^d))$, solve the standard stochastic control problem:

$$\inf_{\alpha \in \mathbb{A}} J^\mu(\alpha) = \mathbb{E} \left[\int_0^T f(t, X_t^\alpha, \mu_t, \alpha_t) dt + g(X_T^\alpha, \mu_T) \right], \quad (1)$$

subject to

$$\begin{aligned} dX_t^\alpha &= b(t, X_t^\alpha, \mu_t, \alpha_t)dt + \sigma(t, X_t^\alpha, \mu_t, \alpha_t)dW_t \\ X_0^\alpha &= \xi. \end{aligned}$$

(2) Find a fixed point, μ , such that $\mathcal{L}(X_t^\alpha) = \mu_t$ for all $0 \leq t \leq T$.

This strategy can be tackled from either the PDE viewpoint (leading to a coupled Hamilton-Jacobi-Bellman and Kolmogorov/Fokker-Plank equations, known as the MFG system in the literature) [26,29] or the probabilistic viewpoint [9,10], which is the focus of this project. Within the probabilistic viewpoint, there are two approaches, both of which are formulated with FBSDEs. See Chapters 3 and 4 of the manuscript of Carmona and Delarue [10] for reference on the two probabilistic approaches.

For simplicity, from now on we assume m , the dimension of the Brownian motion matches d , the dimension of the state variable. We also assume the diffusion coefficient, σ , is a constant matrix $\sigma \in \mathbb{R}^{d \times d}$. For both approaches, we will utilize the Hamiltonian deriving from the aforementioned stochastic control problem (1). In fact, since we assume that the drift is uncontrolled, we can just write the reduced Hamiltonian:

$$H(t, x, \mu, \alpha, y) = b(t, x, \mu, \alpha) \cdot y + f(t, x, \mu, \alpha),$$

for $t \in [0, T]$, $x \in \mathbb{R}^d$, $\mu \in \mathcal{P}_2(\mathbb{R}^d)$, $\alpha \in A$, and an adjoint variable $y \in \mathbb{R}^d$. Then, a key object in order to formulate the solution to (1) is (whenever it exists):

$$\hat{\alpha}(t, x, \mu, y) = \arg \inf_{\alpha \in A} H(t, x, \mu, \alpha, y).$$

We will provide below explicit examples for $\hat{\alpha}(t, x, \mu, y)$. We now give a brief introduction to the two probabilistic approaches.

2.2.1. Weak approach

In the first probabilistic approach, which we will refer to as the weak approach, the optimization problem is solved using a backward SDE for the probabilistic representation of the value function. For a fixed flow of measures $\mu = (\mu_t)_{0 \leq t \leq T}$, let $V : [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ denote the value function:

$$V(t, x) := \inf_{(\alpha_s)_{t \leq s \leq T} \in \mathbb{A}} \mathbb{E} \left[\int_t^T f(s, X_s, \mu_s, \alpha_s) ds + g(X_T, \mu_T) \mid X_t = x \right].$$

The strategy is to evaluate the value function along the solution of the state process $(X_t)_{0 \leq t \leq T}$, namely we let $Y_t = V(t, X_t)$. The weak formulation of the stochastic control problem underpinning V says that, under suitable assumptions that are exemplified below, the pair $(X_t, Y_t)_{0 \leq t \leq T}$ has to solve the following FBSDE:

$$\begin{aligned} dX_t &= b(t, X_t, \mu_t, \hat{\alpha}(t, X_t, \mu_t, \sigma^{-1}Z_t)) dt + \sigma dW_t \\ X_0 &= \xi, \\ dY_t &= -f(t, X_t, \mu_t, \hat{\alpha}(t, X_t, \mu_t, \sigma^{-1}Z_t)) dt + Z_t dW_t \\ Y_T &= g(X_T, \mu_T), \end{aligned} \tag{2}$$

where we assume σ to be invertible. For instance, we take the following set of assumptions from Chapter 3 of the manuscript by Carmona and Delarue [10]: We may assume that the set A for the values of the controls is a bounded closed convex subset of a Euclidean space, the deterministic functions b , f , and g are defined on $[0, T] \times \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times A$, $[0, T] \times \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times A$, and $\mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d)$, respectively, and there exists a constant $C_0 > 1$ such that:

- For any $t \in [0, T]$, $x, x' \in \mathbb{R}^d$, $\alpha, \alpha' \in A$ and $\mu \in \mathcal{P}_2(\mathbb{R}^d)$:

$$\begin{aligned} |(b, f)(t, x', \mu, \alpha') - (b, f)(t, x, \mu, \alpha)| + |\sigma(t, x') - \sigma(t, x)| + \\ + |g(x', \mu) - g(x, \mu)| \leq C_0 |(x, \alpha) - (x', \alpha')|. \end{aligned}$$

- The functions b, f, σ and g are bounded by C_0 .
- There exists a function

$$\hat{\alpha} : [0, T] \times \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times \mathbb{R}^d \ni (t, x, \mu, y) \rightarrow \hat{\alpha}(t, x, \mu, y)$$

which is C_0 -Lipschitz continuous in (x, y) such that, for each $(t, x, \mu, y) \in [0, T] \times \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times \mathbb{R}^d$, $\hat{\alpha}(t, x, \mu, y)$ is the unique minimizer of the Hamiltonian $H(t, x, \mu, y, \alpha)$.

Under this set of assumptions, it is shown in Chapter 3 of the manuscript by Carmona and Delarue [10] that a flow of measures $\mu = (\mu_t)_{0 \leq t \leq T}$ is a mean field game equilibrium if and only if $\mu_t = \mathcal{L}(X_t), \forall t \in [0, T]$, where (X, Y, Z) is a solution of the weak approach FBSDE system in Equation (2), in which case (2) becomes

$$\begin{aligned} dX_t &= b(t, X_t, \mathcal{L}(X_t), \hat{\alpha}(t, X_t, \mathcal{L}(X_t), \sigma^{-1}Z_t)) dt + \sigma dW_t \\ X_0 &= \xi, \\ dY_t &= -f(t, X_t, \mathcal{L}(X_t), \hat{\alpha}(t, X_t, \mathcal{L}(X_t), \sigma^{-1}Z_t)) dt + Z_t dW_t \\ Y_T &= g(X_T, \mathcal{L}(X_T)), \end{aligned} \tag{3}$$

where we use the generic notation $\mathcal{L}(\cdot)$ for the law of a random variable. This approach is developed further in the papers and manuscript of Carmona and Delarue [9, 10].

2.2.2. Pontryagin approach

The second probabilistic approach, which we will refer to as the Pontryagin approach, is based on the Pontryagin stochastic maximum principle. In this formulation, the optimization problem is solved using a backward SDE for the probabilistic representation of the spatial derivative of the value function V . Formally, the strategy is thus to evaluate the process $(X_t)_{0 \leq t \leq T}$ along $\nabla_x V$. Hence we let $Y_t = \nabla_x V(t, X_t)$, which makes sense when $\nabla_x V$ is well-defined. In fact the Pontryagin system may be formulated without any further reference to the regularity of V , the Pontryagin formulation having the following general form:

$$\begin{aligned} dX_t &= b(t, X_t, \mu_t, \hat{\alpha}(t, X_t, \mu_t, Y_t)) dt + \sigma dW_t \\ X_0 &= \xi, \\ dY_t &= -[\nabla_x b(t, X_t, \mu_t, \hat{\alpha}(t, X_t, \mu_t, Y_t))] \cdot Y_t \\ &\quad + \nabla_x f(t, X_t, \mu_t, \hat{\alpha}(t, X_t, \mu_t, Y_t)) dt + Z_t dW_t \\ Y_T &= \nabla_x g(X_T, \mu_T), \end{aligned} \tag{4}$$

where we assume b, f and g to be differentiable with respect to x . We may use the following set of assumptions taken from Chapter 3 of the manuscript by Carmona and Delarue [10] to guarantee that the Pontryagin system is both a necessary and a sufficient condition of optimality: We assume the coefficients b, f , and g are defined on $[0, T] \times \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times A$, $[0, T] \times \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times A$, and $\mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d)$, respectively. We also assume that they satisfy:

- The drift b is an affine function of (x, α) of the form:

$$b(t, x, \mu, \alpha) = b_0(t, \mu) + b_1(t)x + b_2(t)\alpha,$$

where $b_0 : [0, T] \times \mathcal{P}_2(\mathbb{R}^d) \ni (t, \mu) \mapsto b_0(t, \mu)$, $b_1 : [0, T] \ni t \mapsto b_1(t)$ and $b_2 : [0, T] \ni t \mapsto b_2(t)$ are \mathbb{R}^d , $\mathbb{R}^{d \times d}$ and $\mathbb{R}^{d \times d}$ valued, respectively, and are measurable and bounded on bounded subsets of their respective domains.

- There exist two constants $C_1 > 0$ and $C_2 \geq 1$ such that the function $\mathbb{R}^d \times A \ni (x, \alpha) \mapsto f(t, x, \mu, \alpha) \in \mathbb{R}$ is once continuously differentiable with Lipschitz-continuous derivatives (so that $f(t, \cdot, \mu, \cdot)$ is $C^{1,1}$), the Lipschitz constant in x and α being bounded by C_2 (so that it is uniform in t and μ). Moreover, it satisfies the following strong form of convexity:

$$f(t, x', \mu, \alpha') - f(t, x, \mu, \alpha) - (x - x', \alpha - \alpha') \cdot \partial_{(x, \alpha)} f(t, x, \mu, \alpha) \geq C_1 |\alpha' - \alpha|^2.$$

The notation $\partial_{(x, \alpha)} f$ stands for the gradient in the joint variables (x, α) . Finally, f , $\partial_x f$ and $\partial_\alpha f$ are locally bounded over $[0, T] \times \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times A$.

- The function $\mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \ni (x, \mu) \mapsto g(x, \mu)$ is locally bounded. Moreover, for any $\mu \in \mathcal{P}_2(\mathbb{R}^d)$, the function $\mathbb{R}^d \ni x \mapsto g(x, \mu)$ is once continuously differentiable and convex, and has a C_2 -Lipschitz continuous first order derivative.

Under these assumptions, it is shown in Chapter 3 of the manuscript by Carmona and Delarue [10] that a flow of measures $\mu = (\mu_t)_{0 \leq t \leq T}$ is a mean field game equilibrium if and only if $\mu_t = \mathcal{L}(X_t), \forall t \in [0, T]$, where (X, Y, Z) is a solution of the Pontryagin approach FBSDE system in Equation (4), in which case (4) becomes

$$\begin{aligned} dX_t &= b(t, X_t, \mathcal{L}(X_t), \hat{\alpha}(t, X_t, \mathcal{L}(X_t), Y_t)) dt + \sigma dW_t \\ X_0 &= \xi, \\ dY_t &= -[\nabla_x b(t, X_t, \mathcal{L}(X_t), \hat{\alpha}(t, X_t, \mathcal{L}(X_t), Y_t))] \cdot Y_t \\ &\quad + \nabla_x f(t, X_t, \mathcal{L}(X_t), \hat{\alpha}(t, X_t, \mathcal{L}(X_t), Y_t)) dt + Z_t dW_t \\ Y_T &= \nabla_x g(X_T, \mathcal{L}(X_T)). \end{aligned} \tag{5}$$

This approach is also developed further in the papers and manuscript of Carmona and Delarue [9, 10].

2.2.3. Mean Field Games of Control

In many applications, individuals may interact through their controls, instead of their states. One example is an application of trade crowding which was tackled with a mean field game approach in the paper of Cardaliaguet and Lehalle [8]. There is also a model for price impact in the book of Carmona and Delarue which we take as one of our example problems in Section 4.4.2. Mean field games where players interact through the law of their controls is sometimes referred to as extended mean field games [23], see also Chapter 4 in [10]. We are interested in testing our numerical methods on a certain class of mean field games of control: those in which the interaction is through the marginal distributions, $\mathcal{L}(X_t)$ and $\mathcal{L}(\alpha_t)$. To design our algorithms to handle such a general framework of mean field interaction, we study numerical methods for solving a general FBSDE system which includes the two approaches detailed above, as well as this class of mean field games of control.

2.3. General System

We can address both probabilistic formulations for mean field games and a class of mean field games of control simultaneously by considering the following general FBSDE system. We now take the dimension of the state space to be $d = 1$ since, for simplicity, our algorithms will be just applied to this case. Let $[X] = \mathcal{L}(X)$ denote the law of a process X . With an abuse of notation, let $[X, Y, Z] := (\mathcal{L}(X), \mathcal{L}(Y), \mathcal{L}(Z))$ denote the laws

of the individual processes (and not their joint law). The general system is the following:

$$\begin{aligned}
dX_t &= B(t, X_t, Y_t, Z_t, [X_t, Y_t, Z_t])dt + \sigma dW_t \\
X_0 &= \xi \in L^2(\Omega, \mathcal{F}_0, \mathbb{P}; \mathbb{R}), \\
dY_t &= -F(t, X_t, Y_t, Z_t, [X_t, Y_t, Z_t])dt + Z_t dW_t \\
Y_T &= G(X_T, [X_T]).
\end{aligned} \tag{6}$$

The assumption that the coefficients depend, at most, upon the marginal laws $\mathcal{L}(X_t)$, $\mathcal{L}(Y_t)$ and $\mathcal{L}(Z_t)$ of the triple (X_t, Y_t, Z_t) (and not upon the full joint law) is tailor made to the applications we have in mind: As explained in the previous paragraph, we want to handle games in which players interact with one another through the law of the control. In order to guess the impact this may have on the FBSDE representation, it is worth recalling that, in the problem (1), the optimal control may be represented in a quite generic way through the function $\hat{\alpha}$ with the adjoint variable y therein taken as the gradient of the value function. Under the weak formulation approach, this turns the FBSDE system (3) into an FBSDE system depending on the marginal laws of Z , as Z_t is known to have the representation $Z_t = \nabla_x V(t, X_t)\sigma$. Under the Pontryagin approach, it turns the FBSDE system (5) into an FBSDE system depending upon the marginal laws of Y . Hence our choice to handle systems of the form (6). Still the reader should observe that, in order to handle the more general case when the players interact through the joint law of the state and of the control, it is necessary to address systems of the same type as (6) but with the convention that $[X_t, Y_t, Z_t]$ is understood as the joint law of the triple (X_t, Y_t, Z_t) ; as we just mentioned, we do not address this level of generality in the report.

In (6), the diffusion process X is coupled to the diffusion process (Y, Z) through the functions B and F , representing the drift of the forward process and the driver of the backward process, respectively. The functions B and F are assumed to be Lipschitz in each of their arguments on $([0, T], \mathbb{R}^3, (\mathcal{P}_2(\mathbb{R}))^3)$ and G Lipschitz on $(\mathbb{R}, \mathcal{P}_2(\mathbb{R}))$, namely, for $(x, y, z, x', y', z) \in \mathbb{R}^6$ and $(\mu, \nu, \lambda, \mu', \nu', \lambda') \in (\mathcal{P}_2(\mathbb{R}))^6$, we have:

$$\begin{aligned}
|B(t', x', y', z', \mu', \nu', \lambda') - B(t, x, y, z, \mu, \nu, \lambda)| &\leq K_B(|t' - t| + |x' - x| + |y' - y| + |z' - z| \\
&\quad + W_2(\mu', \mu) + W_2(\nu', \nu) + W_2(\lambda', \lambda)) \\
|F(t', x', y', z', \mu', \nu', \lambda') - F(t, x, y, z, \mu, \nu, \lambda)| &\leq K_F(|t' - t| + |x' - x| + |y' - y| + |z' - z| \\
&\quad + W_2(\mu', \mu) + W_2(\nu', \nu) + W_2(\lambda', \lambda)) \\
|G(x', \mu') - G(x, \mu)| &\leq K_G(|x' - x| + W_2(\mu', \mu)).
\end{aligned}$$

The goal of this project is to study numerical methods for solving this general FBSDE system. At some point in the report, we will relax the Lipschitz condition of F in the variables z and λ and address an FBSDE with a quadratic driver F , see our examples in Section 4.

3. ALGORITHMS

We implement two algorithms for numerically solving the FBSDE system in Equation (6). In the first algorithm, we represent paths of the stochastic processes (X, Y, Z) using a tree structure, where branches of the tree represent quantization of the Brownian motion. In the second algorithm, we no longer represent the paths of the process, but the marginal laws of the process. In this case, the law of the process is discretized on a fixed temporal and spatial grid.

In both cases, the representation serves as a basis for a Picard scheme for approaching the solution. For the first algorithm, the Picard scheme is implemented in the form of a global Picard scheme on the whole process; for the second one, iterations are just performed on the marginal laws of the process. Although Picard's method sounds very natural, this strategy suffers, whatever the algorithm, from a serious drawback as Picard iterations for forward backward systems are only expected to converge in small time. Basically, the value of T for which the algorithm will converge depends on the coupling strength of the system; we make this fact clear for the global

method by showing how T depends on the Lipschitz coefficients K_B , K_F and K_G . In any case, bifurcations can be observed when T is increased, or equivalently, when the coupling strength between the two equations is increased. For our convenience (since it can be costly to increase T), we will fix T and explore the convergence of the algorithms as we vary the coupling strength. This is one first step in our report: Compare how the two algorithms suffer from the coupling strength between the forward and backward equations.

The second key feature of our report is to use, for both algorithms, a continuation in time, which allows us to extend the value of the coupling parameter for which the algorithms converge.

In the following sections, we detail our Picard approaches for the two algorithms and the continuation in time method.

3.1. Global Picard Iteration on a Small Time Interval

The main difficulty in numerically solving the FBSDE system is the fact that, not only the forward component $X = (X_t)_{0 \leq t \leq T}$ and backward component $(Y, Z) = (Y_t, Z_t)_{0 \leq t \leq T}$ are coupled, but also they run in opposite directions. Thus, neither equation can be solved independently of the other, seemingly requiring us to manage both time directions simultaneously. Several strategies are conceivable to sort out this issue. A first one is to make use of the *decoupling field* of the system in order to work with one time direction instead of two (roughly speaking, the decoupling field is the value function V in the weak approach and its derivative $\nabla_x V$ in the Pontryagin one). We will investigate this method for our second algorithm; its implementation is indeed pretty subtle in the mean field setting and it leads to the aforementioned *Picard method on the marginal laws*. For our first algorithm, however, we limit ourselves to a *brute force* approach. To decouple the equations, we propose a *global* Picard iteration scheme, whose definition is as follows. For the initial and terminal data of the problem (ξ and G), we want to define a mapping $\Phi_{\xi, G}$ that will take the $j - 1$ Picard iterate and produce the j Picard iterate:

$$\Phi_{\xi, G} : (X^{j-1}, Y^{j-1}, Z^{j-1}, [X^{j-1}, Y^{j-1}, Z^{j-1}]) \mapsto (X^j, Y^j, Z^j, [X^j, Y^j, Z^j]).$$

We define the decoupled Picard scheme $\Phi_{\xi, G}$ as the following:

(1) First, solve

$$\begin{aligned} dX_t^j &= B(t, X_t^{j-1}, Y_t^{j-1}, Z_t^{j-1}, [X_t^{j-1}, Y_t^{j-1}, Z_t^{j-1}])dt + \sigma dW_t \\ X_0^j &= \xi \in L^2(\Omega, \mathcal{F}_0, \mathbb{P}; \mathbb{R}), \end{aligned}$$

for X^j which gives us $[X^j]$.

(2) Next, solve

$$\begin{aligned} dY_t^j &= -F(t, X_t^j, Y_t^{j-1}, Z_t^{j-1}, [X_t^j, Y_t^{j-1}, Z_t^{j-1}])dt + Z_t^j dW_t \\ Y_T^j &= G(X_T^j, [X_T^j]), \end{aligned}$$

for Y^j and Z^j which gives us $[Y^j]$ and $[Z^j]$.

(3) Return $(X^j, Y^j, Z^j, [X^j, Y^j, Z^j])$.

After initializing $(X^0, Y^0, Z^0, [X^0, Y^0, Z^0])$, we can define a sequence by:

$$(X^j, Y^j, Z^j, [X^j, Y^j, Z^j]) = \Phi_{\xi, G}(X^{j-1}, Y^{j-1}, Z^{j-1}, [X^{j-1}, Y^{j-1}, Z^{j-1}]).$$

If the sequence $(X^j, Y^j, Z^j, [X^j, Y^j, Z^j])_{j=1, \dots}$ converges to some $(X, Y, Z, [X, Y, Z])$, then:

$$(X, Y, Z, [X, Y, Z]) = \Phi_{\xi, G}(X, Y, Z, [X, Y, Z]),$$

and thus, (X, Y, Z) is a solution to the original FBSDE system in Equation (6).

Picard schemes such as this one are only guaranteed to converge for a small time horizon, T , depending on the Lipschitz coefficients of the functions B , F and G (and in fact not only the convergence of the Picard sequence but also the solvability of the equation may fail on an arbitrary time interval). We illustrate this idea through the following example (the reader may find the general case in [19]): Let the driver function $F = 0$ and define the common Lipschitz coefficient $K = \max(K_B, K_G)$. The FBSDE system becomes:

$$\begin{aligned} dX_t &= B(Y_t)dt + \sigma dW_t, \quad X_0 = \xi \in \mathbb{R} \\ dY_t &= Z_t dW_t, \quad Y_T = G(X_T, [X_T]). \end{aligned}$$

We write the equation above in the integral form and take the expectation conditional to the filtration \mathcal{F}_t in the backward equation:

$$\begin{aligned} X_t &= \xi + \int_0^t B(Y_s)ds + \sigma W_t \\ Y_t &= G(X_T, [X_T]) - \int_t^T Z_s dW_s, \quad \text{i.e., } Y_t = \mathbb{E}(G(X_T, [X_T]) | \mathcal{F}_t). \end{aligned}$$

Instead of one single X , let us now consider two (initial) processes denoted by \hat{X} and \tilde{X} . After one Picard iteration, we arrive at \hat{X}' and \tilde{X}' . The Picard iteration is given by:

$$\begin{aligned} \hat{Y}_t &= \mathbb{E}(G(\hat{X}_T, [\hat{X}_T]) | \mathcal{F}_t) \\ \hat{X}'_t &= \xi + \int_0^t B(\hat{Y}_s)ds + \sigma W_t, \end{aligned}$$

and similarly for \tilde{X}' . From the forward component, we have the following estimate for the difference between \hat{X}' and \tilde{X}' :

$$\begin{aligned} |\hat{X}'_t - \tilde{X}'_t|^2 &\leq \left| \int_0^t (B(\hat{Y}_s) - B(\tilde{Y}_s))ds \right|^2 \leq \int_0^t |1|^2 ds \int_0^t |B(\hat{Y}_s) - B(\tilde{Y}_s)|^2 ds \\ &\leq tK^2 \cdot \int_0^t |\hat{Y}_s - \tilde{Y}_s|^2 ds \leq TK^2 \cdot T \sup_{0 \leq t \leq T} |\hat{Y}_t - \tilde{Y}_t|^2 \end{aligned}$$

The second inequality is Cauchy-Schwarz inequality and in the third inequality we used the fact that B is Lipschitz with coefficient K . By taking the supremum over $[0, T]$ and then the expectation, we get:

$$\mathbb{E} \left[\sup_{0 \leq t \leq T} |\hat{X}'_t - \tilde{X}'_t|^2 \right] \leq T^2 K^2 \mathbb{E} \left[\sup_{0 \leq t \leq T} |\hat{Y}_t - \tilde{Y}_t|^2 \right].$$

Then we write for the backward component:

$$\begin{aligned}
 \mathbb{E} \left[\sup_{0 \leq t \leq T} |\hat{Y}_t - \tilde{Y}_t|^2 \right] &= \mathbb{E} \left[\sup_{0 \leq t \leq T} |\mathbb{E}(G(\hat{X}_T, [\hat{X}_T]) - G(\tilde{X}_T, [\tilde{X}_T]) | \mathcal{F}_t)|^2 \right] \\
 &= \mathbb{E} \left[\left(\sup_{0 \leq t \leq T} |\mathbb{E}(G(\hat{X}_T, [\hat{X}_T]) - G(\tilde{X}_T, [\tilde{X}_T]) | \mathcal{F}_t)| \right)^2 \right] \\
 &\leq 4\mathbb{E} [|G(\hat{X}_T, [\hat{X}_T]) - G(\tilde{X}_T, [\tilde{X}_T])|^2] \\
 &\leq 4\mathbb{E} \left[\left| K \left(|\hat{X}_T - \tilde{X}_T| + W_2([\hat{X}_T], [\tilde{X}_T]) \right) \right|^2 \right] \\
 &\leq 8K^2\mathbb{E} \left[|\hat{X}_T - \tilde{X}_T|^2 + W_2([\hat{X}_T], [\tilde{X}_T])^2 \right] \\
 &\leq 16K^2\mathbb{E} [|\hat{X}_T - \tilde{X}_T|^2].
 \end{aligned}$$

In the first inequality, we used Doob's martingale inequality for the martingale term $\mathbb{E}(G(\hat{X}_T, [\hat{X}_T]) - G(\tilde{X}_T, [\tilde{X}_T]) | \mathcal{F}_t)_{0 \leq t \leq T}$, and in the last inequality, we used the fact that $W_2([\hat{X}_T], [\tilde{X}_T])^2 \leq \mathbb{E}[|\hat{X}_T - \tilde{X}_T|^2]$.

Combining the inequalities above for both the forward and backward components, we obtain the following estimate:

$$\mathbb{E} \left[\sup_{0 \leq t \leq T} |\hat{X}'_t - \tilde{X}'_t|^2 \right] \leq 16T^2K^4\mathbb{E} [|\hat{X}_T - \tilde{X}_T|^2] \leq 16T^2K^4\mathbb{E} \left[\sup_{0 \leq t \leq T} |\hat{X}_t - \tilde{X}_t|^2 \right].$$

Finally, when $16T^2K^4 < 1$, i.e. $T < 1/(4K^2)$, the mapping $\Phi_{\xi, G}$ realizes a contraction on the forward component and the Picard iteration defined above will converge to the fixed point, providing a solution to the original FBSDE. Thus, we have illustrated that Picard schemes are only expected to converge in small time or for small coupling (i.e. smaller Lipschitz coefficients).

Keeping in mind that we eventually want to describe numerical schemes, we define a solver *picard*, that will implement a finite number, $J_p \in \mathbb{Z}^+$, of Picard iterations. Thus, we define:

- *picard*(ξ, G):
 - (1) Initialize $X_t = \xi$, $Y_t = 0$, and $Z_t = 0, \forall t \in [0, T]$.
 - (2) For $1 \leq j \leq J_p$

$$(X, Y, Z, [X, Y, Z]) = \Phi_{\xi, G}(X, Y, Z, [X, Y, Z])$$
 - (3) Return $(X, Y, Z, [X, Y, Z])$.

3.2. Continuation in Time of the Global Method for Arbitrary Interval/Coupling

Of course, we would like the mapping $\Phi_{\xi, G}$ to realize a contraction to make sure that the Picard iteration converges. As we just explained, this is the case when the forward and backward processes have a small enough coupling strength or for small time horizon T . But in practice, this is not always the case: It may happen that the FBSDE system is uniquely solvable, but that we are unable to prove that the Picard sequence converges. In order to overcome this issue, we follow the approach introduced in Chassagneux, Crisan, Delarue [17]. Basically, the idea is to divide the time interval into smaller intervals, called *levels*, and to apply a Picard solver recursively between the levels.

To define the levels, we fix a time mesh: $\{0 = T_0 < T_1, \dots, T_k, \dots, T_{N_\ell-1} < T_{N_\ell} = T\}$. We would like to use the *picard* solver introduced in the previous section to apply the Picard iteration on a given level. Notice that for an arbitrary level $[T_k, T_{k+1}]$, we do not know the initial condition for X_{T_k} or the terminal condition for $Y_{T_{k+1}}$. Thus, our current approximation of these values will be inputs to the *picard* solver in place of ξ and G . We would also like to modify the solver *picard* to also take the current estimate of $(X, [X])$ so that we don't have to start from scratch every time we wish to use *picard*. Thus, we define for level k :

- $picard[k](X, Y_{T_{k+1}})$:
 - (1) Initialize $Y_t = 0$, and $Z_t = 0, \forall t \in [T_k, T_{k+1})$.
 - (2) For $1 \leq j \leq J_p$

$$(X, Y, Z, [X, Y, Z]) = \Phi_{X_{T_k}, Y_{T_{k+1}}}(X, Y, Z, [X, Y, Z])$$
 - (3) Return $(X, Y, Z, [X, Y, Z])$,

where $X = (X_t)_{T_k \leq t \leq T_{k+1}}$, and similarly for Y and Z and their laws. Note that $\Phi_{X_{T_k}, Y_{T_{k+1}}}$ is the same as $\Phi_{\xi, G}$ defined earlier except the time horizon is $[T_k, T_{k+1}]$ instead of $[0, T]$ and the initial and terminal conditions are given by X_{T_k} and $Y_{T_{k+1}}$ instead of ξ and G , respectively. In particular, notice that we no longer consider the terminal condition in the form of a mapping (like G), but in the form of a random variable (like $Y_{T_{k+1}}$). Implicitly, this requires to store, from one step to another, the full random variable $Y_{T_{k+1}}$; hence our choice below to use a tree.

Now that we have a solver $picard$ which will implement the Picard iteration for a given level, next, we want to define a global solver to apply a continuation in time. The global solver, called $solver$, is recursively defined as follows for some $J_s \in \mathbb{Z}^+$. For a given level k , define:

- $solver[k](X_{T_k}, [X_{T_k}])$:
 - (1) Initialize $X_t = X_{T_k}$, $Y_t = 0$, and $Z_t = 0, \forall t \in [T_k, T_{k+1})$.
 - (2) For $1 \leq j \leq J_s$
 - (a) $(Y_{T_{k+1}}, [Y_{T_{k+1}}]) = solver[k+1](X_{T_{k+1}}, [X_{T_{k+1}}])$
 - (b) $(X, Y, Z, [X, Y, Z]) = picard[k](X, Y_{T_{k+1}})$
 - (3) Return $(Y_{T_k}, [Y_{T_k}])$.

As before, it is important to notice that the entry X_{T_k} in $solver$ is a random variable; $[X_{T_k}]$ is its law. Having the two in our notations is a bit redundant, but we feel it is more transparent for the reader. The break condition of the recursion is given by the terminal condition:

$$solver[N_\ell](X_{T_{N_\ell}}, [X_{T_{N_\ell}}]) = (Y_{T_{N_\ell}}, [Y_{T_{N_\ell}}]) = (G(X_{T_{N_\ell}}, [X_{T_{N_\ell}}]), \mathcal{L}(G(X_{T_{N_\ell}}, [X_{T_{N_\ell}}])))$$

The goal of the continuation in time is for a Picard iteration scheme to converge even for large coupling parameters or large time horizon. We will see in Section 4 that the continuation in time successfully achieves this goal for our benchmark examples. We refer to [17] for its theoretical analysis.

Thus far, we have described our Picard approach and a continuation in time method. Now we need to provide a scheme for discretizing the Picard iteration mapping $\Phi_{\xi, G}$. In this report, we implement a tree algorithm; we also give a variant of it, in the form of a grid algorithm. For both algorithms, we consider the uniform time mesh with time step $h = T/N_t > 0$ with $N_\ell < N_t \in \mathbb{Z}^+$ and $t_i = ih, i = 0, \dots, N_t$. For convenience, we will assume the coarse time mesh used to define the levels, $\{0 = T_0 < T_1, \dots, T_k, \dots, T_{N_\ell-1} < T_{N_\ell} = T\}$, is a subset of the fine time mesh $\{0 = t_0 < t_1, \dots, t_{N_t-1} < t_{N_t} = T\}$.

3.3. Tree Algorithm for the Global Method

The first implementation of the Picard iteration, $\Phi_{\xi, G}$, is the tree algorithm presented in Chassagneux, Crisan, Delarue [17]. We now provide a brief presentation of this algorithm.

3.3.1. Time Discretization

Our first step in developing a discretization for the Picard iteration $\Phi_{\xi, G}$ is to discretize the problem in the time domain. We use the decoupled scheme derived in Chassagneux, Crisan, Delarue [17], and repeated below for convenience. As noted above, we consider the uniform time mesh with time step $h = T/N_t > 0$ with $N_\ell < N_t \in \mathbb{Z}^+$ and $t_i = ih, i = 0, \dots, N_t$.

Step 1) in defining $\Phi_{\xi, G}$ requires solving the forward equation for X^j . We use the classical Euler scheme:

$$X_{t_{i+1}}^j = X_{t_i}^j + h B(t_i, X_{t_i}^{j-1}, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}, [X_{t_i}^{j-1}, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}]) + \sigma \Delta W_i, \quad X_0^j = \xi.$$

Note that when we calculate $X_{t_{i+1}}^j$, the value of $X_{t_i}^j$ and its law are known and could be substituted for $X_{t_i}^{j-1}$ and its law in the drift function, B and the scheme becomes:

$$X_{t_{i+1}}^j = X_{t_i}^j + h B(t_i, X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}, [X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}]) + \sigma \Delta W_i, \quad X_0^j = \xi.$$

Step 2) in defining $\Phi_{\xi, G}$ requires solving the backward equation for Y^j and Z^j . We derive the discrete-time scheme to approximate the backward component, see e.g. [6]. The Y^j component at time t_i in the backward scheme is obtained by taking the expectation conditional to \mathcal{F}_{t_i} , denoted as \mathbb{E}_{t_i} , of the backward equation between t_i and t_{i+1} . Let $\Delta W_i = W_{t_{i+1}} - W_{t_i}$ denote the forward Brownian increment between t_i and t_{i+1} . The driver function F is approximated by its value at time t_i and we use the fact that F at t_i and Z_{t_i} are \mathcal{F}_{t_i} -measurable, and $\mathbb{E}_{t_i}(\Delta W_i) = \mathbb{E}_{t_i}(W_{t_{i+1}} - W_{t_i}) = 0$:

$$\begin{aligned} Y_{t_i}^j &= Y_{t_{i+1}}^j + \int_{t_i}^{t_{i+1}} F(t, X_t^j, Y_t^{j-1}, Z_t^{j-1}, [X_t^j, Y_t^{j-1}, Z_t^{j-1}]) dt - \int_{t_i}^{t_{i+1}} Z_t^j dW_t \\ &\approx Y_{t_{i+1}}^j + h F(t_i, X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}, [X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}]) - Z_{t_i}^j \Delta W_i \\ Y_{t_i}^j &= \mathbb{E}_{t_i}(Y_{t_{i+1}}^j) + h F(t_i, X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}, [X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}]) \\ Y_T^j &= G(X_T^j, [X_T^j]). \end{aligned}$$

As for the Z^j component, we multiply the approximation of $Y_{t_i}^j$ by the Brownian increment ΔW_i , taking the conditional expectation \mathbb{E}_{t_i} , and using $\mathbb{E}((\Delta W_i)^2) = h$. By noticing that our scheme will never make use of Z_T^j , we can simply set the terminal condition for the Z^j component to 0.

$$\begin{aligned} Z_{t_i}^j (\Delta W_i)^2 &\approx Y_{t_{i+1}}^j \Delta W_i + (-Y_{t_i}^j + h F(t_i, X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}, [X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}])) \Delta W_i \\ Z_{t_i}^j &= h^{-1} \mathbb{E}_{t_i}(Y_{t_{i+1}}^j \Delta W_i), \quad Z_T^j = 0. \end{aligned}$$

Putting this together, the time-discretized decoupled forward-backward scheme for Picard iteration of our general FBSDE system (6) is the following:

$$\begin{cases} X_{t_{i+1}}^j = X_{t_i}^j + h B(t_i, X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}, [X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}]) + \sigma \Delta W_i \\ X_0^j = \xi, \\ Y_{t_i}^j = \mathbb{E}_{t_i}(Y_{t_{i+1}}^j) + h F(t_i, X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}, [X_{t_i}^j, Y_{t_i}^{j-1}, Z_{t_i}^{j-1}]) \\ Y_T^j = G(X_T^j, [X_T^j]) \\ Z_{t_i}^j = h^{-1} \mathbb{E}_{t_i}(Y_{t_{i+1}}^j \Delta W_i) \\ Z_T^j = 0. \end{cases} \quad (7)$$

Other variants of this forward-backward scheme are possible. For example, in the forward scheme we could change j to $j-1$ and t_i to t_{i+1} on the right hand side. It is easy to observe that the forward-backward system is decoupled because of lagged Picard indices $j-1$ and j . Thus, given $(X^{j-1}, Y^{j-1}, Z^{j-1}, [X^{j-1}, Y^{j-1}, Z^{j-1}])$ where $X^{j-1} = (X_{t_i}^{j-1})_{0 \leq i \leq N_t}$, and similarly for Y and Z , we can solve the backward scheme autonomously and then the forward scheme to obtain $(X^j, Y^j, Z^j, [X^j, Y^j, Z^j])$. We have not fully provided a discrete scheme for $\Phi_{\xi, G}$ yet, however, because for a given t_i , we have not discretized $(X_{t_i}, Y_{t_i}, Z_{t_i})$. This is the goal of the next section.

3.3.2. Spatial Discretization via Tree Structure

The forward-backward decoupled scheme (7) above looks quite simple and explicit. However, it still presents some difficulties for the numerical computation. Firstly, it is difficult to compute the conditional expectation in the backward scheme. Secondly, it is non trivial to compute the law of $X_{t_{i+1}}$ forward in time. Even if there was no drift, the computation would involve the convolution of the law of X_{t_i} and a Gaussian law of the Brownian increment. Ultimately, we will need a spatial discretization.

The approach of this algorithm is to approximate the Brownian increments using a simple binomial approximation: $\Delta W_i = \pm\sqrt{h}$ with probability $1/2$. This gives rise to a binomial tree for the forward scheme. Each node on the tree at depth i represents a value of X_{t_i} , and has two children nodes representing the two possible values of $X_{t_{i+1}}$ (the “up \uparrow ” and the “down \downarrow ” value), conditioned on the value of X_{t_i} . The two values are computed as follows:

$$X_{t_{i+1}}^j(\uparrow\downarrow) = X_{t_i}^j + h B(t_i, X_{t_i}^j, Y_{t_i}^j, Z_{t_i}^j, [X_{t_i}^j, Y_{t_i}^j, Z_{t_i}^j]) \pm \sigma\sqrt{h}. \quad (8)$$

Suppose that we use M points x_1, \dots, x_M for the approximation of the law ξ of the forward process at the initial time, i.e. $[X_0] = \xi \approx \sum_{k=1}^M p_k^0 \delta_{x_k}(\cdot)$. Then we have M parallel binomial trees at each Picard iteration. For Picard iterate j and time t_i , the number of nodes at depth i is $M \times 2^i$ with values of $(X_{t_i}^j, Y_{t_i}^j, Z_{t_i}^j)$ saved on the nodes of the tree at depth i . The marginal law of each process at time t_i can be determined by looking at all the values on the nodes at depth i . The backward scheme can be easily computed on the binomial tree. At the last time step, $T = t_{N_t}$, we have $Y_T^j = G(X_T^j, [X_T^j])$ for each of the $M \times 2^{N_t}$ nodes. The conditional expectation in the backward scheme at t_i is simply the average of the “up” and “down” branches at t_{i+1} .

To initialize the $j = 0$ Picard iterate as in the definition of the solver *picard*, we want to set $X_{t_i} = \xi, \forall i \in \{0, \dots, N_t\}$. This amounts to taking each initial value x_k and initializing its entire tree to this value, meaning that $X_{t_i}^0 = x_k$ for all nodes at depth i and for all $i \in \{1, \dots, N_t\}$ of the k -th binomial tree. We then begin the Picard iteration by applying the mapping $\Phi_{\xi, G}$ as detailed above. Using the binomial tree and approximation of Brownian increments, the forward-backward decoupled scheme becomes fully implementable.

3.4. Picard Iteration on the Marginal Laws: a Grid Algorithm

The complexity of the tree algorithm is exponential with respect to the number of time steps N_t , since the size of the binomial tree is of order 2^{N_t} . The exponential complexity becomes problematic and makes continuation in time much slower when we deal with large time horizons. In order to reduce the size of the tree, a natural idea is to make some “recombination” of the binomial tree. But since the drift function B depends on the value of the process, the two branches “up-down” and “down-up” from the same node at time t_i will not coincide at time t_{i+2} , in general. Instead of recombination, we may fix a spatial grid of a controllable size that the binomial tree can be projected onto, in order to avoid exponential complexity. This will be the first ingredient of our grid algorithm, which is the main novelty in our report.

Inspired by the paper of Delarue and Menozzi [20], where the authors used a spatial grid for the approximation of FBSDEs without mean-field interaction, we make an intensive use of the notion of *decoupling field*, which is the second key ingredient of our strategy. Indeed, using the representation result in Proposition 2.2 in [16], we know that there exist deterministic feedback functions $(u, v) : [0, T] \times \mathbb{R} \times \mathcal{P}_2(\mathbb{R}) \rightarrow \mathbb{R}$ with u a solution to a nonlinear PDE (on the space of measures) such that for a solution (X, Y, Z) to the general FBSDE system in Equation (6):

$$Y_t = u(t, X_t, [X_t]) \quad \text{and} \quad Z_t = v(t, X_t, [X_t]).$$

Generally speaking, u is called the decoupling field of (6). Here comes the main observation: the time marginals of the solution to the general FBSDE system in Equation (6), $([X_t, Y_t, Z_t])_{0 \leq t \leq T}$, can be equivalently characterized by $(\mu_t, u(t, \cdot), v(t, \cdot))_{0 \leq t \leq T}$ where $[X_t] = \mu_t$. As in [20], our strategy is to approximate (u, v) instead of (X, Y, Z) , but this is not so easy as u and v are defined on a space of infinite dimension (because the mean field component lives in $\mathcal{P}(\mathbb{R})$). In order to overcome this difficulty, we propose to freeze the mean field argument in (u, v) . This permits to regard u and v as functions of a finite-dimensional variable and is to thus approximate both along the underlying spatial grid. Once an approximation of u and v has been computed for the given

proxy of the marginal laws, we can update the value of this proxy by using a Picard method. Hence the name of this subsection. So, as opposed to the tree algorithm, we will no longer keep track of the pathwise laws of the processes. Instead, we will only compute the marginal laws at each time step of the time mesh by means of a Picard iteration.

3.4.1. Picard Iteration without Grid

We first give the inputs and outputs of our new Picard mapping without any grid approximation:

$$\Psi_{[\xi],G} : (\mu_t^{j-1}, u^{j-1}(t, \cdot), v^{j-1}(t, \cdot))_{0 \leq t \leq T} \mapsto (\mu_t^j, u^j(t, \cdot), v^j(t, \cdot))_{0 \leq t \leq T},$$

with $\mu_0^{j-1} = \mu_0^j = [\xi]$. Denoting below $\varphi \# \nu$ as the push-forward measure of the measure ν by the function φ , $\Psi_{[\xi],G}((\mu_t^{j-1}, u^{j-1}(t, \cdot), v^{j-1}(t, \cdot))_{0 \leq t \leq T})$ is defined by:

- (1) Solve the following SDE for X^j :

$$\begin{aligned} dX_t^j &= B(t, X_t^j, u^{j-1}(t, X_t^j), v^{j-1}(t, X_t^j), \mu_t^{j-1}, (u^{j-1}(t, \cdot), v^{j-1}(t, \cdot)) \# \mu_t^{j-1}) dt + \sigma dW_t \\ X_0^j &= \xi \in L^2(\Omega, \mathcal{F}_0, \mathbb{P}; \mathbb{R}). \end{aligned}$$

Set $\mu_t^j := [X_t^j]$.

- (2) Next, let $\mathcal{D}^j = \{(t, x) \in [0, T] \times \mathbb{R} \mid x \in \text{supp}(\mu_t^j)\} \subset [0, T] \times \mathbb{R}$ denote the support of $(\mu_t^j)_{0 \leq t \leq T}$. Find $(u^j, v^j)(\cdot, \cdot) : \mathcal{D}^j \rightarrow \mathbb{R}^2$ through $(u^j, v^j)(t, X_t^j) := (Y_t^j, Z_t^j)$ by solving:

$$\begin{aligned} dY_t^j &= -F(t, X_t^j, Y_t^j, Z_t^j, \mu_t^j, (u^{j-1}(t, \cdot), v^{j-1}(t, \cdot)) \# \mu_t^j) dt + Z_t^j dW_t, \\ Y_T^j &= G(X_T^j, \mu_T^j). \end{aligned}$$

- (3) Return $((\mu_t^j, u^j(t, \cdot), v^j(t, \cdot))_{0 \leq t \leq T})$.

Given $\Psi_{[\xi],G}$, we can go through the construction of the global Picard method and define similar routines *picard* and *solver* for our new Picard approach by replacing formally $\Phi_{\xi,G}$ by $\Psi_{[\xi],G}$. The various inputs and outputs in the new routines should be clear. The next step to make the whole algorithm entirely tractable is to show how to compute μ_t^j , u^j , and v^j explicitly in the mapping $\Psi_{[\xi],G}$. Similar to $\Phi_{\xi,G}$, this mapping will be defined explicitly for a discretized scheme on a temporal and spatial grid in the next two sections. As in the tree algorithm, we consider the uniform time mesh with time step $h = T/N_t > 0$ with $N_\ell < N_t \in \mathbb{Z}^+$ and $t_i = ih, i = 0, \dots, N_t$.

3.4.2. Grid Approximation of Forward Component and its Law

We begin by fixing a spatial discretization grid. This grid could in principle be defined differently for each time step t_i , but for simplicity, we consider a homogeneous grid Γ fixed for all time steps t_i , $i \in \{0, \dots, N_t\}$ with constant spatial step size Δx . Let Π be the projection function on the grid

$$\Gamma = \{x_k = x_1 + (k-1)\Delta x, k = 1, \dots, N_x\}.$$

Precisely, Π is given by

$$\Pi(x) = x_k \text{ if } x \in [x_k - \Delta x/2, x_k + \Delta x/2) \quad \text{and} \quad \Pi(x) = x_1 \text{ if } x \leq x_1, \quad \Pi(x) = x_{N_x} \text{ if } x \geq x_{N_x}.$$

The initial law ξ of the forward process is approximated as $\xi \approx \mu_0(\cdot)$ on the grid Γ with N_x points. Recall that in the tree algorithm, we cannot choose M , the number of points for the approximation of the initial law, to be too large as we will need M parallel binomial trees. Because the tree algorithm has exponential complexity, we are able to choose N_x to be much larger than M , and in turn, the approximation of the initial law is more

accurate in the grid algorithm than in the tree algorithm. We can initialize the Picard iterate $(\mu_i^0, u_i^0, v_i^0)_{0 \leq i \leq N_t}$ similarly as in the tree algorithm, by letting $\mu_i^0 = \mu_0$ and $(u_i^0, v_i^0) = (0, 0)$, for all $i \leq N_t$.

We follow the definition of Step 1) for $\Psi_{[\xi], G}$, but we use the Euler scheme for the forward process between t_i and t_{i+1} :

$$X_{t_{i+1}}^j = X_{t_i}^j + h B(X_{t_i}^j, (u_i^{j-1}, v_i^{j-1})(X_{t_i}^j), \mu_i^{j-1}, (u_i^{j-1}, v_i^{j-1})\#\mu_i^{j-1}) + \sigma \Delta W_i. \quad (9)$$

Suppose the j Picard iterate at time t_i is given by μ_i^j with $\mu_0^j = \mu_0$:

$$\mu_i^j(\cdot) = \sum_{k=1}^{N_x} p_{i,k}^j \delta_{x_k}(\cdot), \quad p_{i,k}^j \geq 0 \quad \forall k \in \{1, \dots, N_x\} \quad \text{and} \quad \sum_{k=1}^{N_x} p_{i,k}^j = 1.$$

The law of $X_{t_i}^j$ in the Euler scheme of the forward process is μ_i^j . Then we would like to define μ_{i+1}^j as the law of $X_{t_{i+1}}^j$ in the Euler scheme above, but the quantity $X_{t_{i+1}}^j$ may not belong to the grid. Instead of using formula (9), the natural idea is then to replace it by its projection on the grid Γ :

$$\begin{aligned} X_{t_{i+1}}^j &= \Pi \left(X_{t_i}^j + h B(X_{t_i}^j, (u_i^{j-1}, v_i^{j-1})(X_{t_i}^j), \mu_i^{j-1}, (u_i^{j-1}, v_i^{j-1})\#\mu_i^{j-1}) + \sigma \Delta W_i \right) \\ [X_{t_i}^j] &= \mu_i^j(\cdot) \quad \text{and} \quad \mu_{i+1}^j(\cdot) = [X_{t_{i+1}}^j]. \end{aligned} \quad (10)$$

The law of $X_{t_{i+1}}^j$ is the convolution of μ_i^j with transition density $q^j(t_i, t_{i+1}; x_k, x_n)$ with x_k, x_n two points of the grid Γ and $k, n \in \{1, \dots, N_x\}$. The transition probability is equal to the conditional probability that the process X^j starting at time t_i from point x_k arrives at time t_{i+1} at the point x_n , i.e.

$$q^j(t_i, t_{i+1}; x_k, x_n) = \mathbb{P}(X_{t_{i+1}}^j = x_n | X_{t_i}^j = x_k).$$

The discretized law μ_{i+1}^j is then written as:

$$\begin{aligned} \mu_{i+1}^j(\cdot) &= (\mu_i * q^j(t_i, t_{i+1}))(\cdot) = \sum_{n=1}^{N_x} p_{i+1,n}^j \delta_{x_n}(\cdot) \\ p_{i+1,n}^j &= \sum_{k=1}^{N_x} \left(p_{i,k}^j \times q^j(t_i, t_{i+1}; x_k, x_n) \right) = \sum_{k=1}^{N_x} \left(p_{i,k}^j \times \mathbb{P}(X_{t_{i+1}}^j = x_n | X_{t_i}^j = x_k) \right). \end{aligned}$$

It is worth noticing that if we did not take the projection when computing $X_{t_{i+1}}^j$ in the scheme (10), then its law would be given by convolution of the law μ_i^j with the Gaussian transition density $\bar{q}^j(t_i, t_{i+1}; x_k, y)$ associated to the Euler scheme. The transition densities q^j and \bar{q}^j have the following relation:

$$\mathbb{P}(X_{t_{i+1}}^j = x_n | X_{t_i}^j = x_k) = q^j(t_i, t_{i+1}; x_k, x_n) = \int_{\beta(x_n, \Delta x/2)} \bar{q}^j(t_i, t_{i+1}; x_k, y) dy.$$

In fact, for a more tractable implementation of the forward scheme, we use the binomial approximation for the Brownian increments (8) introduced in the previous section. Note that quantization with more points can be easily applied. In this binomial case, with $(\uparrow)/(\downarrow)$ representing the ‘‘up’’ and ‘‘down’’ branches, respectively, the transition probabilities on the grid can be easily computed:

$$\begin{aligned} &\mathbb{P}(X_{t_{i+1}}^j = x_n | X_{t_i}^j = x_k) \\ &= \frac{1}{2} \left(\mathbf{1}(X_{t_{i+1}}^j(\uparrow) = x_n | X_{t_i}^j = x_k) + \mathbf{1}(X_{t_{i+1}}^j(\downarrow) = x_n | X_{t_i}^j = x_k) \right). \end{aligned}$$

Then we can write μ_{i+1}^j by computing the probabilities:

$$p_{i+1,n}^j = \sum_{k=1}^{N_x} \frac{p_{i,k}^j}{2} \cdot \left(\mathbf{1}(X_{t_{i+1}}^j(\uparrow) = x_n | X_{t_i}^j = x_k) + \mathbf{1}(X_{t_{i+1}}^j(\downarrow) = x_n | X_{t_i}^j = x_k) \right)$$

At Picard iteration $j \geq 1$, the forward scheme finally gives the flow of measures $(\mu_i^j)_{i=0}^{N_t}$ at discrete time steps $(t_i)_{i=0}^{N_t}$ of the discretized law defined on the grid Γ . Thus, we have described an implementation of Step 1) in the definition of the Picard mapping $\Psi_{[\xi],G}$. In the next section, we detail the implementation of the backward components in Step 2).

3.4.3. Grid for the approximation of the backward component

Given the current Picard iterates $(\mu_i^j, u_i^{j-1}(\cdot), v_i^{j-1}(\cdot))$, $i = 0, \dots, N_t$, we would like to detail Step 2) in the definition of $\Psi_{[\xi],G}$. Since we have a discrete spatial grid Γ , we wish to compute the values of $u_i^j(x)$ and $v_i^j(x)$ for $x \in \Gamma$. By replacing $Y_{t_i}^j$ and $Z_{t_i}^j$ with their respective feedback functions in the backward component in Equation (7), for $x \in \Gamma$ we have the following backward scheme starting, for $i \leq N_t - 1$, with terminal condition at time $T = t_{N_t}$, $(u_{N_t}^j, v_{N_t}^j) = (G, 0)$:

$$\begin{aligned} u_i^j(x) &= \mathbb{E} \left(u_{i+1}^j(X_{t_{i+1}}^j, \mu_{i+1}^j) + h \cdot F(X_{t_i}^j, u_i^{j-1}(X_{t_i}^j), v_i^{j-1}(X_{t_i}^j), \mu_i^j, (u_i^{j-1}, v_i^{j-1}) \# \mu_i^j) \mid X_{t_i}^j = x \right), \\ v_i^j(x) &= \mathbb{E} \left(u_{i+1}^j(X_{t_{i+1}}^j) \cdot \Delta W_i / h \mid X_{t_i}^j = x \right). \end{aligned}$$

The variable $X_{t_{i+1}}^j$ with law $[X_{t_{i+1}}^j] = \mu_{i+1}^j$ is given by the forward scheme presented in the previous section with starting point $X_{t_i}^j = x$. Notice that by construction of the forward scheme, $X_{t_{i+1}}^j \in \Gamma$ and $\text{supp}(\mu_{i+1}^j) = \Gamma$ so $u_{i+1}^j(\cdot)$ has been calculated and saved at time t_{i+1} .

We have a more explicit scheme in the case of binomial approximation of the Brownian increments, which is used for the numerical results for our examples, with $X_{t_{i+1}}^j(\uparrow)$ and $X_{t_{i+1}}^j(\downarrow)$ as defined in the forward scheme, always conditional to $X_{t_i}^j = x$ and given μ_i^j at time t_i :

$$\begin{aligned} u_i^j(x) &= \frac{1}{2} \left(u_{i+1}^j(X_{t_{i+1}}^j(\uparrow)) + u_{i+1}^j(X_{t_{i+1}}^j(\downarrow)) \right) + h \cdot F(x, u_i^{j-1}(x), v_i^{j-1}(x), \mu_i^j, (u_i^{j-1}, v_i^{j-1}) \# \mu_i^j), \\ v_i^j(x) &= \frac{h^{-1/2}}{2} \left(u_{i+1}^j(X_{t_{i+1}}^j(\uparrow)) - u_{i+1}^j(X_{t_{i+1}}^j(\downarrow)) \right). \end{aligned}$$

We have now described a scheme for Step 2) in the definition of $\Psi_{[\xi],G}$. Putting this together with the previous section, we have described a fully implementable scheme for $\Psi_{[\xi],G}$. Note that we can use this Picard iteration mapping to define the analogue of the solvers *picard* and *solver*. Importantly, we can also apply the continuation in time method to the grid algorithm as well. In the next section, we apply these two methods, the tree and grid algorithms, to five example problems.

4. EXAMPLES

We have collected five example problems to test the algorithms presented in Section 3.

4.1. Linear Example

The first example is a linear model which comes directly from Chassagneux, Crisan, Delarue [17], in which they implemented the tree algorithm. The system of interest is the following:

$$\begin{aligned}
dX_t &= -\rho \mathbb{E}(Y_t) dt + \sigma dW_t \\
X_0 &= x_0 \\
dY_t &= -aY_t dt + Z_t dW_t \\
Y_T &= X_T.
\end{aligned}$$

For this problem, the solution is known explicitly:

$$Y_0 = \frac{x_0 e^{aT}}{1 + \frac{\rho}{a}(e^{aT} - 1)}$$

For the numerical results, we let $\rho = 0.1$, $a = 0.25$, $\sigma = 1$, $T = 1$, and $x_0 = 2$. We vary h , the time step size, and for the grid algorithm, we use $\Delta x = h^2$. Figure 1 shows the log error between the numerical and true solution values of Y_0 as a function of log number of time steps for the tree and grid algorithm with one level (i.e. without using continuation in time). The left figure (tree algorithm) repeats the results in [17] and as expected, it decreases linearly. On the other hand, we also observe a negative trend in the rate of convergence for the grid algorithm. Thus, both algorithms appear to converge to the true solution.

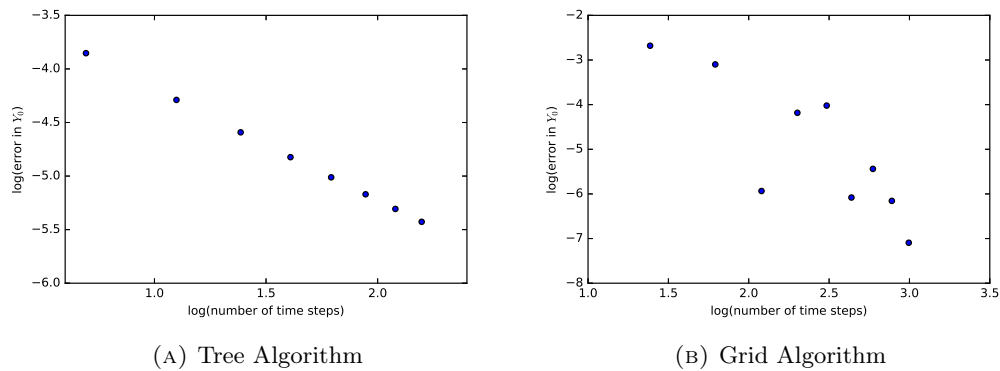


FIGURE 1. Linear Example: Convergence of the algorithms with one level as the number of time steps increases.

4.2. Trigonometric Drivers Example

The second example also comes directly from Chassagneux, Crisan, Delarue [17]. The system of interest is the following:

$$\begin{aligned}
dX_t &= \rho \cos(Y_t) dt + \sigma dW_t \\
X_0 &= x_0 \\
Y_t &= \mathbb{E}_t(\sin(X_T))
\end{aligned}$$

For the numerical results, we let $\sigma = 1$, $T = 1$, $x_0 = 0$, $h = 1/6$ for the tree algorithm and $h = 1/12$ and $\Delta x = h^2$ for the grid algorithm. For this problem, we observe a bifurcation when using the tree algorithm as we increase the coupling parameter, ρ . Figure 2 shows the values of Y_0 from the last 5 Picard iterations. Starting at about $\rho = 3.5$, the tree algorithm without continuation in time bifurcates. If the continuation in time method is used for the tree algorithm with two levels, there is no bifurcation for the range of values of ρ showed in the

plot. Note that the results from the tree algorithm repeat those in [17]. The grid algorithm performs quite well in the sense that even with only one level, the algorithm converges for all of the values of ρ in the plot. In particular, it avoids the exponential growth of the data structure characterizing the tree algorithm. Note that even though both the tree method with two levels and the grid method with one level converge, they produce different values for Y_0 for larger values of ρ . We believe this is because the tree algorithm is less accurate since the time step is larger.

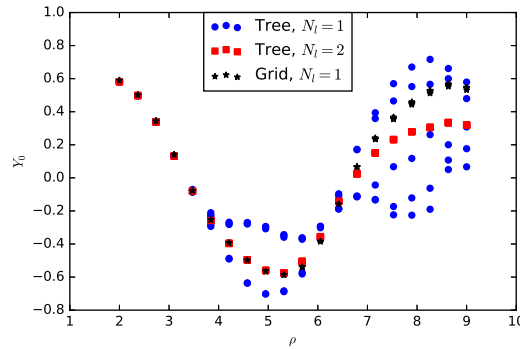


FIGURE 2. Trigonometric Drivers Example: Bifurcations in the values of Y_0 appear as the coupling parameter, ρ , increases. The tree algorithm with one level is shown in blue circles and with two levels is shown in red squares. The grid algorithm with one level is shown in black asterisks.

For values of ρ for which the tree algorithm bifurcates, we were interested in the effect of changing σ on the convergence. Figure 3 shows the last 5 values of Y_0 from the Picard iteration as σ varies for $\rho = 5$. Surprisingly, the value of σ plays no role in the bifurcation for this value of ρ . To see if σ would play a role when ρ is closer to the bifurcation point, we have analogous plots when $\rho = 3.5$ and $\rho = 4$ (see Figure 4). In these plots, however, it is clear that σ does affect the bifurcation. Understanding the role of σ on the bifurcation is an open question. It is interesting to note that even though the tree algorithm does not bifurcate for $\rho = 3.5$ when $\sigma = 1$, we still observe a bifurcation when we fix $\rho = 3.5$ and vary σ . This suggests that we check if the grid algorithm bifurcates as we vary σ for a fixed value of ρ where there is no bifurcation when $\sigma = 1$, such as $\rho = 5$. Figure 5 shows that the grid algorithm does not bifurcate as we change σ for fixed $\rho = 5$.

4.3. Mixed Model

The third example also comes directly from Chassagneux, Crisan, Delarue [17]. The system of interest is the following:

$$\begin{aligned} dX_t &= -\rho Y_t dt + \sigma dW_t \\ X_0 &= x_0 \\ dY_t &= \arctan(\mathbb{E}(X_t)) dt + Z_t dW_t \\ Y_T &= \arctan(X_T). \end{aligned}$$

For the numerical results, we let $\sigma = 1$, $T = 1$, $x_0 = 2$, $h = 1/6$ for the tree algorithm and $h = 1/12$ and $\Delta x = h^2$ for the grid algorithm. We also observe a bifurcation for this problem as we increase the coupling parameter, ρ . Figure 6 shows the values of Y_0 from the last 5 Picard iterations. Starting at about $\rho = 1.5$, the tree algorithm without continuation in time bifurcates. If the continuation in time method is used for the tree

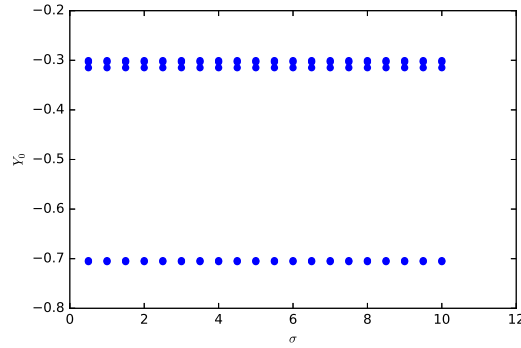
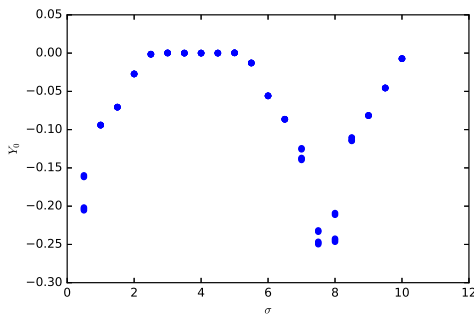
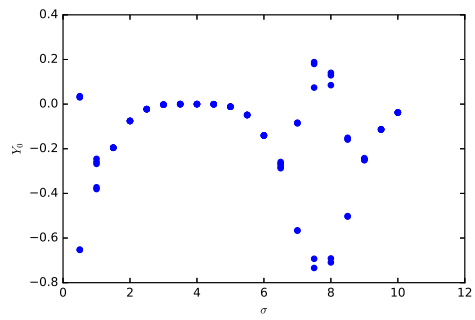


FIGURE 3. Trigonometric Drivers Example: Tree algorithm with one level for $\rho = 5$. Changing σ has no effect on the bifurcation.



(A) $\rho = 3.5$



(B) $\rho = 4$

FIGURE 4. Trigonometric Drivers Example: Tree algorithm with one level for $\rho = 3.5$ and $\rho = 4$. Changing σ produces unexplained results.

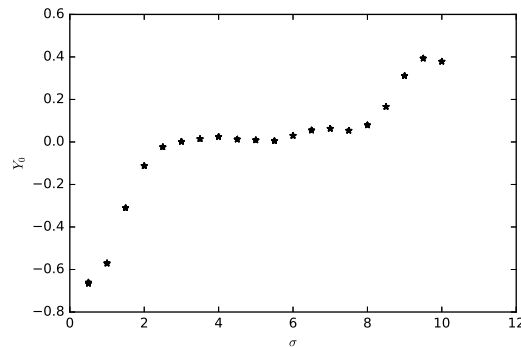


FIGURE 5. Trigonometric Drivers Example: Grid algorithm with one level for $\rho = 5$. There is no bifurcation as we vary σ .

algorithm with two levels, the bifurcation point is pushed back to about $\rho = 3$, and pushed back further when using three levels to about $\rho = 5$. Note that these results for the tree algorithm repeat those in [17]. The grid algorithm performs quite well again in the sense that it converges for all of the values of ρ shown, even when using only one level. Further, its major attractiveness is the lower complexity compared to the tree algorithm.

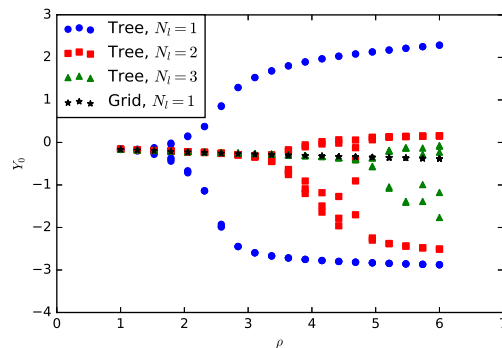


FIGURE 6. Mixed Model: Bifurcations in the values of Y_0 appear as the coupling parameter, ρ , increases. The tree algorithm with one, two, and three levels is shown in blue circles, red squares, and green triangles, respectively. The grid algorithm with one level is shown in black asterisks.

As in the trigonometric drivers example, we can also investigate the effect of changing σ for a value of ρ where the tree algorithm without continuation in time bifurcates. For $\rho = 2$, Figure 7 shows that the tree algorithm converges for large enough values of σ .

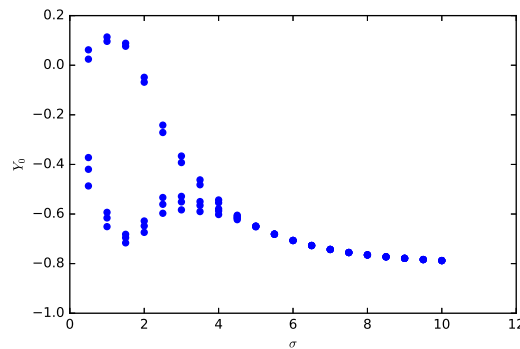


FIGURE 7. Mixed Model: Tree algorithm with one level for $\rho = 2$. The algorithm converges for large enough values of σ .

Since we have observed that the tree algorithm converges for small value of ρ and large values of σ , this suggests trying a continuation in ρ and/or σ , instead of the continuation in time. Instead of implementing a full continuation method, we used a simpler method of incrementing the parameter of interest.

The incrementation in ρ is performed by starting the algorithm with a small value of ρ and letting it converge. Then ρ is increased by some fixed $\Delta\rho$, and the algorithm is initialized with the solution from the previous value of ρ . Figure 8 shows the results from the incrementation in ρ . The incrementation in ρ only increases the bifurcation point by a small amount.

The incrementation in σ is similar, except for each value of ρ , we start the algorithm with a sufficiently large value of σ such that it will converge. Then σ is decreased by a fixed $\Delta\sigma$. Figure 9 shows the results from the incrementation in σ . The incrementation in σ also only increases the bifurcation point by a small amount.

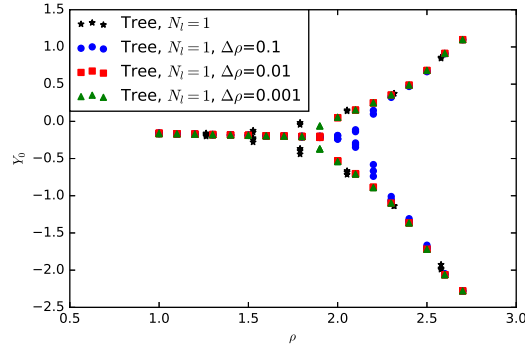


FIGURE 8. Mixed Model: Tree algorithm with one level. Without continuation or incrementation is shown in black. Incrementation in ρ with $\Delta\rho = 0.1, 0.01,$ and 0.001 are shown in blue circles, red squares, and green triangles, respectively.

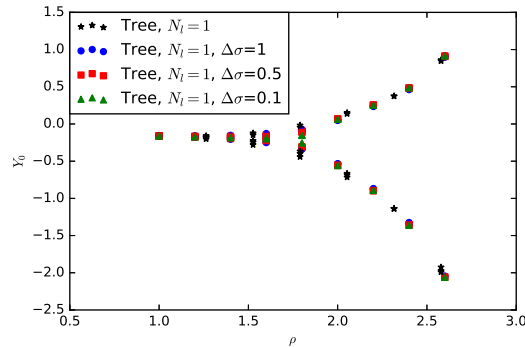


FIGURE 9. Mixed Model: Tree algorithm with one level. Without continuation or incrementation is shown in black asterisks. Incrementation in σ with $\Delta\sigma = 1, 0.5,$ and 0.1 are shown in blue circles, red squares, and green triangles, respectively.

If we take the previous example but change the drift and driver functions to also be in terms of the mean of the processes, then we have the following system, which we will refer to as the mixed model of means:

$$\begin{aligned} dX_t &= -\rho\mathbb{E}(Y_t)dt + \sigma dW_t \\ X_0 &= x_0 = 2 \\ dY_t &= \arctan(\mathbb{E}(X_t))dt + Z_t dW_t \\ Y_T &= \arctan(\mathbb{E}(X_T)). \end{aligned}$$

For the last example, we noticed that increasing σ allows the tree algorithm to converge. We are thus interested to see if replacing the dynamics with the mean of the process will remove the effect of changing σ

on the convergence. We use the same values of the parameters as before. Figure 10 shows the bifurcation with one level of the tree algorithm (i.e. without using continuation in time). The effect of changing σ is shown in Figure 11. Our prediction is confirmed: σ no longer affects the convergence when the dynamics are replaced with the mean of the process.

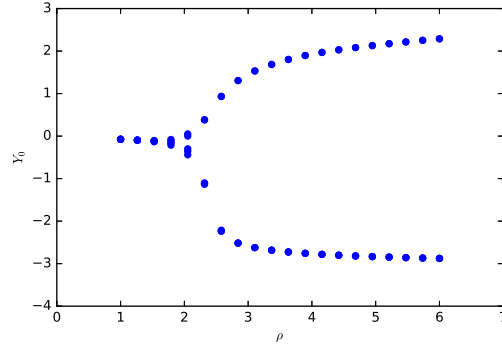


FIGURE 10. Mixed Model of Means: Bifurcation for the tree algorithm with one level.

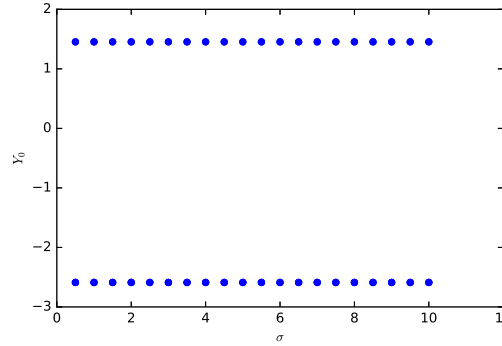


FIGURE 11. Mixed Model of Means: Tree algorithm with one level with $\rho = 3$. Changing σ has no affect on the bifurcation.

4.4. Examples: Linear Quadratic Mean Field Games

The last two examples belong to the family of linear quadratic (LQ) games. In these models, the dynamics of the state are linear in the sense that the drift is defined by an affine function:

$$b(t, x, \alpha) = A_t x + B_t \alpha + \beta_t.$$

Furthermore, the running and the terminal cost are quadratic in the state and control variables. For the sake of simplicity, we choose not to include the cross terms, so that we define f and g as follows:

$$f(t, x, \alpha) = \frac{1}{2} P_t x^2 + \frac{1}{2} Q_t \alpha^2$$

$$g(x) = \frac{1}{2} S x^2.$$

The FBSDE system derived by approaching the LQ mean field games via the weak approach becomes:

$$\begin{aligned} dX_t &= [A_t x + B_t \alpha + \beta_t] dt + \sigma dW_t, & X_0 &= \xi \\ dY_t &= \left[-\frac{1}{2} P_t X_t^2 + (A_t X_t + \beta_t) \frac{Z_t}{\sigma} + \frac{1}{2} \frac{B_t^2}{Q_t} \frac{Z_t^2}{\sigma^2} \right] dt + dW_t, & Y_T &= \frac{1}{2} S X_T^2. \end{aligned} \quad (11)$$

We present two examples of linear quadratic games: a problem of flocking, and a price impact model of a trader.

Remark 1. *We observe that the driver of the BSDE is quadratic in Z_t , which makes it seldom solvable. On the other hand, it is possible to obtain explicit solutions for LQ models. If we were to numerically solve equation (11), we might observe blow up from the quadratic terms in the backward equation and consequently the algorithm to not converge. Luckily, we do not observe such blowup in the examples we consider. But if blowup does occur, one could consider adapting the method presented by Chassagneux and Richou in [18] to numerically approximate a quadratic BSDE.*

4.4.1. Flocking Problem

The next example problem models flocking. As in the paper of Nourian, Caines, and Malhamé [31], we consider the spatially homogeneous case where the state, X_t , represents the velocity of a representative player, or bird. Each bird controls its velocity through the process:

$$dX_t = \alpha_t dt + \sigma dW_t,$$

where the control is chosen to minimize:

$$J(\alpha) = \mathbb{E} \left[\int_0^T \left(\frac{1}{2} \alpha_t^2 + \frac{\rho}{2} (X_t - \bar{\mu}_t)^2 \right) dt \right].$$

over $\alpha \in \mathbb{A}$. Above, we let $\bar{\mu}_t$ denote the mean of the distribution μ_t (of the velocities of the birds) at time t . The running cost consists of two components. The first term encourages the birds to minimize their kinetic energy by not choosing a large control. The second term encourages the birds to align their velocities with the mean velocity of the group.

This model falls into the class of linear quadratic games. Assume that the initial condition is given by a constant, $X_0 = x_0$. It can be shown that the solution is Gaussian with mean and variance:

$$\begin{aligned} \mathbb{E}(X_t) &= x_0 \\ \text{Var}(X_t) &= \sigma^2 \int_0^t \exp \left(-2 \int_s^t \eta_u du \right) ds, \end{aligned}$$

where:

$$\eta_t = \sqrt{\rho} \frac{e^{2\sqrt{\rho}(T-t)} - 1}{e^{2\sqrt{\rho}(T-t)} + 1}.$$

Using the weak formulation, the FBSDE system of interest is the following:

$$\begin{aligned}
dX_t &= -\frac{Z_t}{\sigma} dt + \sigma dW_t \\
X_0 &= x_0 \\
dY_t &= -\left(\frac{Z_t^2}{2\sigma^2} + \frac{\rho}{2}(X_t - \mathbb{E}X_t)^2\right) dt + Z_t dW_t \\
Y_T &= 0.
\end{aligned}$$

If we use the Pontryagin formulation, the FBSDE system becomes:

$$\begin{aligned}
dX_t &= -Y_t dt + \sigma dW_t \\
X_0 &= x_0 \\
dY_t &= -\rho(X_t - \mathbb{E}(X_t)) dt + Z_t dW_t \\
Y_T &= 0.
\end{aligned}$$

The numerical results are presented for $\rho = 1$, $\sigma = 1$, $T = 1$, $x_0 = 0$, $h = 1/20$ for the tree algorithm and $h = 1/130$ and $\Delta x = h^2$ for the grid algorithm. Figure 12(a) shows the results for the grid algorithm for both the weak and Pontryagin approaches. The plot shows the weights of the distribution $\mathcal{L}(X_T)$. The results are similar between both approaches and coincide with the true solution.

We can look at the convergence rate by calculating the 2-Wasserstein distance between the numerical results and the true solution as we change the number of time steps. Since our state space is in one dimension, we can calculate the Wasserstein distance explicitly using the representation provided by Prokhorov [32]:

$$W_p(\mu, \nu) = \left(\int_0^1 |F_\mu^{-1}(u) - F_\nu^{-1}(u)|^p du \right)^{1/p},$$

where $F_\mu(x) = \mu([0, x])$, denotes the cumulative distribution function. Figure 12(b) presents the convergence rate of the grid algorithm in terms of the 2-Wasserstein distance calculated between the true solution and numerical results with respect to the number of time steps. As expected, the 2-Wasserstein distance decreases towards 0 as we increase the number of time steps, for both the Pontryagin and weak approaches.

The results for the tree algorithm are shown in Figure 13 for the weak and Pontryagin approaches. As with the grid algorithm, the weak and Pontryagin solutions are similar to each other and coincide with the true solution.

4.4.2. Trader Problem

The last example shows an application of mean field games to finance, such as the trader congestion model in the paper of Cardaliaguet and Lehalle, [8]. We focus on the *Price Impact Model* presented in the book of Carmona and Delarue in [10]. The interest for this kind of model is motivated by their use in optimal execution problems for high frequency trading. Furthermore, it represents an instance of extended mean field game, also known as mean field game of controls, in which the representative agent interacts with the law of the control instead of the law of their state. The problem consists in a group of traders who have to buy or sell a large amount of shares in a given interval of time $[0, T]$. If they trade too fast, they will suffer from market impact. On the other hand, if they trade too slow, they will be affected by a large risk penalization. Approaching this problem as a mean field game, the inventory of the representative trader is modeled by a stochastic process $(X_t)_{0 \leq t \leq T}$ such that

$$dX_t = \alpha_t dt + \sigma dW_t, \quad t \in [0, T],$$

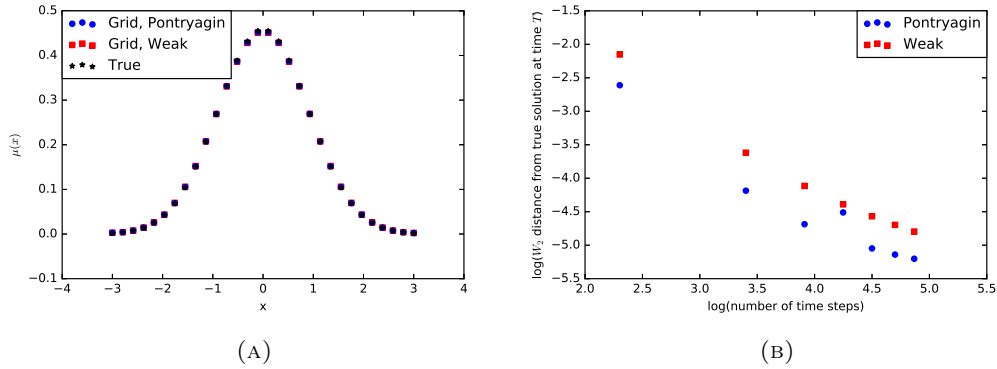


FIGURE 12. Flocking Problem: (a) Distribution μ_T of the players' states at time T for the grid algorithm with one level. Pontryagin is in blue circles, weak is in red squares, true solution is shown in black asterisks. (b) 2-Wasserstein distance between true solution and numerical solution for grid algorithm with one level as we increase the number of time steps, plotted as a log-log plot. Pontryagin approach is in blue circles and weak approach is in red squares.

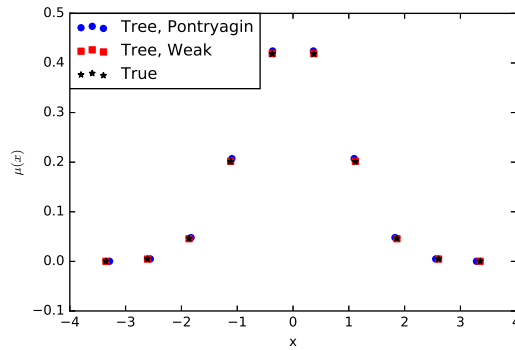


FIGURE 13. Flocking Problem: Distribution μ_T of the players' states at time T for the tree algorithm with one level. Pontryagin is shown in blue circles, weak is shown in red squares, and true solution is shown in black asterisks.

where α_t corresponds to the trading rate. The price of the asset $(S_t)_{0 \leq t \leq T}$ is influenced by the trading strategies of all the traders through the law of the controls $(\theta_t = \mathcal{L}(\alpha_t))_{0 \leq t \leq T}$ as follows:

$$dS_t = \gamma \left(\int_{\mathbb{R}} a d\theta_t(a) \right) dt + \sigma_0 dW_t^0, \quad t \in [0, T],$$

where γ and σ_0 are constants and the Brownian motion W^0 is independent from W . The amount of cash held by the trader at time t is denoted by the process $(K_t)_{0 \leq t \leq T}$. The dynamic of K is modeled by

$$dK_t = -[\alpha_t S_t + c_\alpha(\alpha_t)] dt,$$

where the function $\alpha \mapsto c_\alpha(\alpha)$ is a non-negative convex function satisfying $c_\alpha(0) = 0$, representing the cost for trading at rate α . The wealth V_t of the trader at time t is defined as the sum of the cash held by the trader

and the value of the inventory with respect to the price S_t :

$$V_t = K_t + X_t S_t.$$

Applying the self-financing condition of Black-Scholes' theory, the changes over time of the wealth V are given by the equation:

$$\begin{aligned} dV_t &= dK_t + X_t dS_t + S_t dX_t \\ &= \left[-c_\alpha(\alpha_t) + \gamma X_t \int_{\mathbb{R}} a d\theta_t(a) \right] dt + \sigma S_t dW_t + \sigma_0 X_t dW_t^0. \end{aligned} \quad (12)$$

We assume that the trader is subject to a running liquidation constraint modeled by a function c_X of the shares they hold, and to a terminal liquidation constraint at maturity T represented by a scalar function g . Thus, the cost function is defined by:

$$J(\alpha) = \mathbb{E} \left[\int_0^T c_X(X_t) dt + g(X_T) - V_T \right].$$

Applying Equation (12), it follows that

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

where the running cost is defined by

$$f(t, x, \theta, \alpha) = c_\alpha(\alpha) + c_X(x) - \gamma x \int_{\mathbb{R}} a d\theta(a),$$

for $0 \leq t \leq T$, $x \in \mathbb{R}^d$, $\theta \in \mathcal{P}(A)$ and $\alpha \in A = \mathbb{R}$. We assume that the functions c_X and g are quadratic and that the function c_α is strongly convex in the sense that its second derivative is bounded away from 0. Such a particular case is known as the Almgren-Chriss linear price impact model. Thus, the control is chosen to minimize:

$$J(\alpha) = \mathbb{E} \left[\int_0^T \left(\frac{c_\alpha}{2} \alpha_t^2 + \frac{c_X}{2} X_t^2 - \gamma X_t \int_{\mathbb{R}} a d\theta_t(a) \right) dt + \frac{c_g}{2} X_T^2 \right],$$

over $\alpha \in \mathbb{A}$. To summarize, the running cost consists of three components. The first term represents the cost for trading at rate α . The second term takes into consideration the running liquidation constraint in order to penalize unwanted inventories. The third term defines the actual price impact. Finally, the terminal cost represents the terminal liquidation constraint.

As for the flocking example, this model falls in the class of linear quadratic games. Assume that the initial condition is given by a constant, $X_0 = x_0$. The solution is Gaussian with mean and variance defined as:

$$\begin{aligned} \mathbb{E}(X_t) &= x_0 e^{-\int_0^t \frac{\bar{\eta}_s}{c_\alpha} ds} \\ \text{Var}(X_t) &= \sigma^2 \int_0^t e^{-\frac{2}{c_\alpha} \int_s^t \eta_r dr} ds \end{aligned}$$

where:

$$\begin{aligned} \bar{\eta}_t &= \frac{-C(e^{(\delta^+ - \delta^-)(T-t)} - 1) - c_g(\delta^+ e^{(\delta^+ - \delta^-)(T-t)} - \delta^-)}{(\delta^- e^{(\delta^+ - \delta^-)(T-t)} - \delta^+) - c_g B(e^{(\delta^+ - \delta^-)(T-t)} - 1)} \\ \eta_t &= \frac{-c_\alpha \sqrt{c_X/c_\alpha} c_\alpha \sqrt{c_X/c_\alpha} - c_g - (c_\alpha \sqrt{c_X/c_\alpha} + c_g) e^{2\sqrt{c_X/c_\alpha}(T-t)}}{c_\alpha \sqrt{c_X/c_\alpha} - c_g + (c_\alpha \sqrt{c_X/c_\alpha} + c_g) e^{2\sqrt{c_X/c_\alpha}(T-t)}}, \end{aligned}$$

for $t \in [0, T]$, where $B = 1/c_\alpha$, $C = c_X$, $\delta^\pm = -D \pm \sqrt{R}$, with $D = -\gamma/(2c_\alpha)$ and $R = D^2 + BC$. Using the weak approach yields the following FBSDEs system:

$$\begin{aligned} dX_t &= -\frac{1}{c_\alpha} \frac{Z_t}{\sigma} dt + \sigma dW_t, & X_0 &= x_0 \\ dY_t &= -\left[\frac{c_X}{2} X_t^2 + \frac{\gamma}{c_\alpha} \frac{\mathbb{E}[Z_t]}{\sigma} X_t + \frac{1}{2c_\alpha} \left(\frac{Z_t}{\sigma} \right)^2 \right] dt + Z_t dW_t, & Y_T &= c_g \frac{X_T^2}{2}. \end{aligned}$$

Alternatively, the FBSDE system obtained via the Pontryagin approach is:

$$\begin{aligned} dX_t &= -\frac{1}{c_\alpha} Y_t dt + \sigma dW_t, & X_0 &= x_0 \\ dY_t &= -\left(c_X X_t + \frac{\gamma}{c_\alpha} \mathbb{E}[Y_t] \right) dt + Z_t dW_t, & Y_T &= c_g X_T. \end{aligned}$$

The numerical results focus on the effect of the continuation method for the grid algorithm. In contrast with the previous examples, we show that the grid algorithm is also affected by bifurcation. Figure 14 shows the last five Picard iterations of Y_0 for the Pontryagin approach when the number of levels ranges from 1 to 3. Fixing the parameters $x_0 = 1$, $\sigma = 0.7$, $1/c_\alpha = 1.5$, $c_g = 0.3$, $\gamma = 2$, $T = 1$, $h = 1/12$, $\Delta x = h^2$ and increasing the coupling parameter, c_X , we observe that the bifurcation effect can be corrected by increasing the number of levels. In fact, Figure 14 shows that the true value of Y_0 matches the value computed numerically when using three levels.

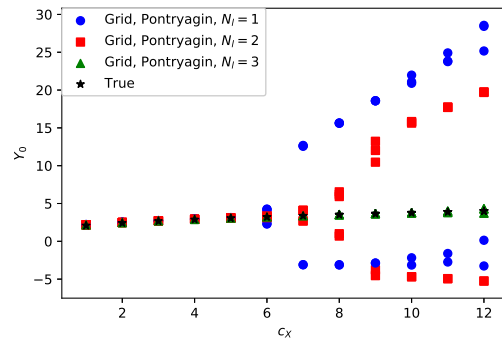


FIGURE 14. Trader Problem: Bifurcations in the values of Y_0 depending on the coupling parameter c_X for different number of levels in the grid algorithm. One, two, and three levels are shown in blue circles, red squares, and green triangles, respectively. The true value of Y_0 is shown in black asterisks.

Furthermore, Figure 15(a) compares the distribution $\mathcal{L}(X_T)$ obtained by the Pontryagin and the weak approaches using the grid algorithm with parameters $x_0 = 1$, $\sigma = 0.7$, $1/c_\alpha = 0.3$, $c_g = 0.3$, $\gamma = 2$, $c_X = 2$, $T = 1$, $h = 1/130$ and $\Delta x = h^2$. The two approaches produce similar results that coincide with the true solution.

Figure 15(b) presents the convergence rate in terms of the 2-Wasserstein distance calculated between the true solution and numerical results with respect to the number of time steps. We again make use of the explicit representation of the Wasserstein distance [32]. The numerical solution is obtained using the grid algorithm with parameters $x_0 = 1$, $\sigma = 0.7$, $1/c_\alpha = 0.3$, $c_g = 0.3$, $\gamma = 2$, $c_X = 2$, $T = 1$, and $\Delta x = h^2$. As expected, the 2-Wasserstein distance decreases towards 0 as we increase the number of time steps.

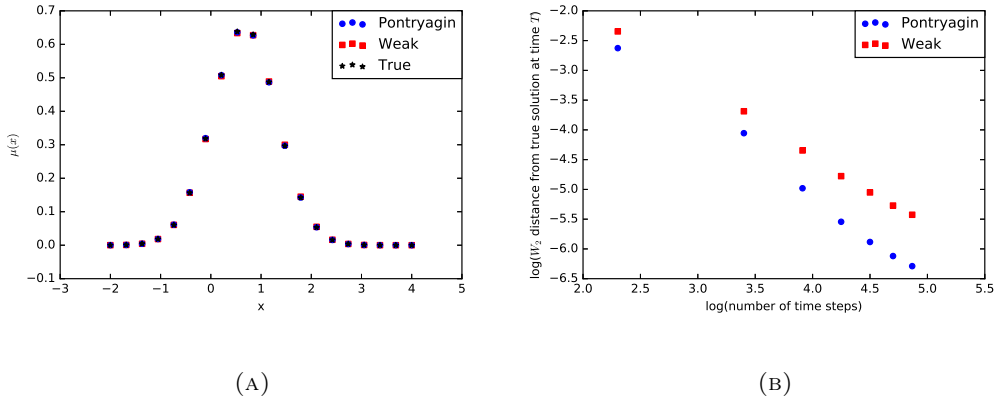


FIGURE 15. Trader Problem: (a) Distribution μ_T of the players' states at time T for the grid algorithm with one level. Pontryagin is shown in blue circles, weak is shown in red squares, and the true solution is shown in black asterisks. (b) 2-Wasserstein distance between true solution and numerical solution for grid algorithm with one level as we increase the number of time steps, plotted as a log-log plot. Pontryagin approach is shown in blue circles and weak approach is shown in red squares.

The last plot, Figure 16, shows the error from the true solution of the control at time 0, α_0 , as we increase the number of time steps. This value is given by $\alpha_0 = -Y_0/c_X$ for the Pontryagin approach and $\alpha_0 = -Z_0/(c_X\sigma)$ for the weak approach. The true value is given by $\alpha_0 = -\bar{\eta}_0x_0/c_X$. As for the 2-Wasserstein distance, the error in the control decreases towards 0.

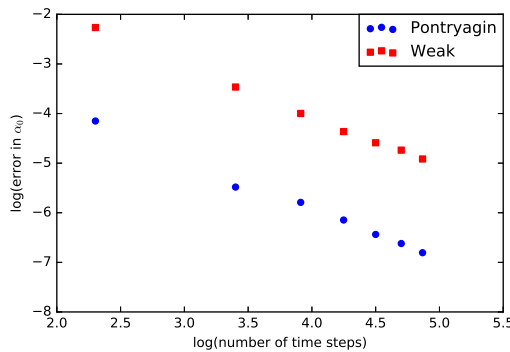


FIGURE 16. Trader Problem: Error in the control at time 0, α_0 , as we increase the number of time steps, plotted as a log-log plot.

5. CONCLUSION

In conclusion, we have provided two algorithms for numerically solving FBSDEs of McKean-Vlasov type, which can be used to formulate the solutions to mean field game problems. The first algorithm is based on a pathwise tree structure. The second algorithm is based on a marginal grid structure. We have also proposed various refinements to the algorithms, including a continuation in time, and incrementation of a coupling

parameter or the diffusion coefficient. The different numerical methods were illustrated on five benchmark examples.

The tree algorithm's main advantage is that we do not need to project the values of X_t onto a discretized spatial grid, which potentially makes the algorithm more accurate. However, a significant disadvantage of the tree algorithm is the exponential growth of the data structure as the number of time steps is increased. This exponential growth is made worse yet if a higher order of quantization were to be used for approximating the Brownian increments.

The grid algorithm's main advantage is it avoids the exponential growth of the data structure. A higher order of quantization may be used without drastically changing the algorithm's complexity. A disadvantage of the grid algorithm is its sensitivity to the spatial step size with respect to the time step size. For the algorithm to be stable, the two step sizes need to be well adjusted to each other.

For both the tree and grid algorithms, we have observed that the continuation in time is able to extend the range of values of the coupling parameters for which the algorithms will converge. The incrementation methods proposed in Subsection 4.3, however, were not very successful at avoiding the bifurcations.

This report has touched on many things that could be explored deeper. First of all, the extension of the grid algorithm in [20] to the mean field setting has not been studied from a theoretical standpoint. It is an open question to determine if this algorithm converges (meaning that the error decreases as the grid size decreases). The effect of the continuation in time or incrementation of the coupling parameter and/or diffusion coefficient has also yet to be studied. The numerical results also raised questions on the influence of the diffusion coefficient in the bifurcations.

6. ACKNOWLEDGMENTS

The authors would like to thank the organizers of the 2017 CEMRACS for the opportunity to study at CIRM. The student authors would also like to thank Professors Chassagneux and Delarue for mentoring us during the summer school. Finally, we would like to thank the funding sources that supported us during CEMRACS, including our respective universities, the NSF, and SMAI.

REFERENCES

- [1] Y. Achdou, F. Camilli and I. Capuzzo-Dolcetta. Mean field games: convergence of a finite difference method, *SIAM J. Numer. Anal.*, 51, pp. 2585-2612 (2013).
- [2] Y. Achdou and I. Capuzzo-Dolcetta. Mean field games: numerical methods *SIAM J. Numer. Anal.*, 48, pp. 1136-1162 (2010).
- [3] Y. Achdou and M. Lauri re. Mean field type control with congestion. *Applied Mathematics & Optimization* 73.3 (2016): 393-418.
- [4] Y. Achdou and A. Porretta. Convergence of a finite difference scheme to weak solutions of the system of partial differential equations arising in mean field games, *SIAM J. Numer. Anal.*, 54, pp. 161-186. (2016).
- [5] J.D. Benamou and G. Carlier, Augmented Lagrangian methods for transport optimization, mean field games and degenerate elliptic equations, *Journal of Optimization Theory and Applications*, 167, pp. 1-26 (2015).
- [6] B. Bouchard and N. Touzi, Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations, *Stochastic Processes and their applications*, 111, (2), pp 175-206, (2004).
- [7] L. Briceno-Arias, D. Kalise and F. Silva, Proximal methods for stationary mean field games with local couplings, *SIAM J. Control Optim.*, 56 (2), 801-836 (2013).
- [8] P. Cardaliaguet and C.A. Lehalle. Mean field game of controls and an application to trade crowding. arXiv:1610.09904[q-fin.TR] (2016).
- [9] R. Carmona and F. Delarue. Probabilistic analysis of mean-field games. *SIAM J. Control Optim.*, 51(4), 2705-2734 (2013).
- [10] R. Carmona, and F. Delarue. *Probabilistic Theory of Mean Field Games with Applications I*. Springer, 2018.
- [11] R. Carmona, F. Delarue, and A. Lachapelle. Control of McKean-Vlasov versus mean field games. *Mathematics and Financial Economics* (2013): Vol. 7, 131-166.
- [12] R. Carmona, J.P. Fouque, and L.H. Sun. Mean field games and systemic risk. arXiv:1308.2172 (2013).
- [13] R. Carmona, C. Graves. Jet lag recovery: synchronization of circadian oscillators as a mean field game. arXiv:1804.04729. (2018)
- [14] R. Carmona, and P. Wang. Finite state mean field games with major and minor players. arXiv:1610.05408. (2016).
- [15] P. Chan, and R. Sircar. Fracking, renewables, and mean field games. *SIAM Review* 59.3 (2017): 588-615.

- [16] J.F. Chassagneux, D. Crisan, and F. Delarue. A probabilistic approach to classical solutions of the master equation for large population equilibria. arXiv:1411.3009. (2014).
- [17] J.F. Chassagneux, D. Crisan, and F. Delarue. Numerical method for FBSDEs of McKean-Vlasov type. To appear in *The Annals of Applied Probability*.
- [18] J.F. Chassagneux, A. Richou. Numerical simulations of quadratic BSDEs. *The Annals of Applied Probability* (2016): Vol. 26, No. 1, 262-304
- [19] F. Delarue. On the existence and uniqueness of solutions to FBSDEs in a non-degenerate case. *Stochastic Processes and Applications*, 99, 209-286 (2002).
- [20] F. Delarue and S. Menozzi. A forward-backward stochastic algorithm for quasi-linear PDEs *Ann. Appl. Probab.*, 16 (1), pp 140- 184. (2006).
- [21] D. A. Gomes and H. Mitake. Existence for stationary mean-field games with congestion and quadratic Hamiltonians. *Nonlinear Differential Equations and Applications NoDEA 22.6* (2015): 1897-1910.
- [22] D. A. Gomes and J. Saúde. Mean field games models- a brief survey. *Dynamic Games and Applications*, 4(2), 110-154. (2014).
- [23] D. A. Gomes and V. K. Voskanyan. Extended mean field games. *Izv. Nats. Akad. Nauk Armenii Mat.* 48 63-76. MR3112690 (2013).
- [24] O. Guéant. New numerical methods for mean field games with quadratic costs. *Networks and Heterogeneous Media*, 2, pp. 315-336 (2012).
- [25] C. Harris, S. Howison, and R. Sircar. Games with exhaustible resources. *SIAM Journal on Applied Mathematics* 70.7 (2010): 2556-2581.
- [26] M. Huang, P. E. Caines, and R. P. Malhamé. Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Communications in Information and Systems*, 6:221-252, (2006).
- [27] A. Lachapelle, J. Salomon and G. Turinici. Computation of mean field equilibria in economics. *Mathematical Models and Methods in Applied Sciences*, 20, pp. 567-588 (2010).
- [28] A. Lachapelle, and M.T. Wolfram. On a mean field game approach modeling congestion and aversion in pedestrian crowds. *Transportation research part B: methodological* 45.10 (2011): 1572-1589.
- [29] J. M. Lasry, and P. L. Lions. Mean field games. *Japanese Journal of Mathematics* 2.1 (2007): 229-260.
- [30] M. Nourian, P. E. Caines, and R. P. Malhamé. Mean field analysis of controlled Cucker-Smale type flocking: linear analysis and perturbation equations. *IFAC Proceedings Volumes* 44.1 (2011): 4471-4476.
- [31] M. Nourian, P. E. Caines, and R. P. Malhamé. Synthesis of Cucker-Smale type flocking via mean field stochastic control theory: Nash equilibria. *IEEE*, (2010): 814-819
- [32] Y. V. Prokhorov. Convergence of random processes and limit theorems in probability theory. *Theory of Probability & Its Applications* 1.2 (1956): 157-214.