Probabilistic Approach to Mean Field Games

Andrea Angiuli Jean-François Chassagneux François Delarue
Christy Graves Houzhi Li

CEMRACS August 2017

Outline

1. Mean Field Games
2. Algorithms for solving FBSDEs
   - Picard Iteration
   - Continuation in Time
   - Tree Algorithm
   - Grid Algorithm
3. Benchmark Examples
   - Toy Example
   - Trader Problem

Mean Field Games

- Search for analogue of Nash equilibrium for a continuum of players.

Mean Field Games

- Search for analogue of Nash equilibrium for a continuum of players.
- Strategy:

Mean Field Games

- Search for analogue of Nash equilibrium for a continuum of players.
- Strategy:
  1. For a fixed deterministic flow of probability measures $\mu = (\mu_t)_{0 \le t \le T}$ on the state space, solve the stochastic control problem:

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

subject to

$$\begin{aligned} dX_t^{\alpha} &= b(t, X_t^{\alpha}, \mu_t, \alpha_t)dt + \sigma(t, X_t^{\alpha}, \mu_t, \alpha_t)dW_t \\ X_0 &= \xi \end{aligned} \tag{2}$$

Mean Field Games

- Search for analogue of Nash equilibrium for a continuum of players.
- Strategy:
  1. For a fixed deterministic flow of probability measures $\mu = (\mu_t)_{0 \le t \le T}$ on the state space, solve the stochastic control problem:

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

     subject to

$$\begin{aligned} dX_t^{\alpha} &= b(t, X_t^{\alpha}, \mu_t, \alpha_t)dt + \sigma(t, X_t^{\alpha}, \mu_t, \alpha_t)dW_t \\ X_0 &= \xi \end{aligned} \tag{2}$$

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

Probabilistic Approaches

Weak Approach:

- Let $u(t, x)$ denote the value function.

Probabilistic Approaches

Weak Approach:

- Let $u(t, x)$ denote the value function.
- Let $Y_t = u(t, X_t)$.

Probabilistic Approaches

Weak Approach:

- Let $u(t,x)$ denote the value function.
- Let $Y_t = u(t, X_t)$.

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

Probabilistic Approaches

Pontryagin Stochastic Maximum Principle Approach:

- Let $u(t,x)$ denote the value function.

Probabilistic Approaches

Pontryagin Stochastic Maximum Principle Approach:

- Let $u(t,x)$ denote the value function.
- Let $Y_t = \partial_x u(t, X_t)$.

Probabilistic Approaches

Pontryagin Stochastic Maximum Principle Approach:

- Let $u(t,x)$ denote the value function.
- Let $Y_t = \partial_x u(t, X_t)$.

$$
\begin{aligned}
dX_t &= b\left(t, X_t, \mathcal{L}(X_t), \hat{\alpha}\left(t, X_t, \mathcal{L}(X_t), Y_t\right)\right) dt + \sigma dW_t \\
X_0 &= \xi \\
dY_t &= -\left[\partial_x b((t, X_t, \mathcal{L}(X_t), \hat{\alpha}\left(t, X_t, \mathcal{L}(X_t), Y_t\right)))\, Y_t \right. \\
&\qquad \left. + \partial_x f\left(t, X_t, \mathcal{L}(X_t), \hat{\alpha}\left(t, X_t, \mathcal{L}(X_t), Y_t\right)\right)\right] dt + Z_t dW_t \\
Y_T &= \partial_x g(X_T, \mathcal{L}(X_T))
\end{aligned}
\tag{4}
$$

General System

General form of coupled FBSDEs:

General System

General form of coupled FBSDEs:

$$\begin{aligned} dX_t &= B(t, X_t, Y_t, Z_t, \mathcal{L}(X_t, Y_t, Z_t))dt + \sigma dW_t \\ X_0 &= \xi \\ dY_t &= -F(t, X_t, Y_t, Z_t, \mathcal{L}(X_t, Y_t, Z_t))dt + Z_t dW_t, \\ Y_T &= G(X_T, \mathcal{L}(X_T)) \end{aligned} \tag{5}$$

Outline

Discretization and Picard iteration for FBSDE

- Time horizon $[0, T]$ with fixed time mesh $t_i = ih, i = 1, ..., N_t$.

Discretization and Picard iteration for FBSDE

- Time horizon $[0, T]$ with fixed time mesh $t_i = ih, i = 1, ..., N_t$.
- The initial law ξ of the forward process X has a discretized distribution which has a finite support:

$$\mathbf{P}[\xi = x_m] = p_m, \quad \sum_{m=1}^{M} p_m = 1, \quad m = 1, ..., M$$

Discretization and Picard iteration for FBSDE

- Time horizon $[0, T]$ with fixed time mesh $t_i = ih, i = 1, ..., N_t$.
- The initial law ξ of the forward process X has a discretized distribution which has a finite support:

$$\mathbf{P}[\xi = x_m] = p_m, \quad \sum_{m=1}^{M} p_m = 1, \quad m = 1, ..., M$$

- The decoupled forward-backward scheme is:

$$\begin{aligned}
Y^j_{t_i} &= \mathbb{E}_{t_i}[Y^j_{t_{i+1}} + h \cdot F(t_i, X^{j-1}_{t_i}, Y^{j-1}_{t_i}, Z^{j-1}_{t_i}, \mathcal{L}(X^{j-1}_{t_i}, Y^{j-1}_{t_i}, Z^{j-1}_{t_i}))] \\
Y^j_T &= g(X^{j-1}_T, \mathcal{L}(X^{j-1}_T)) \\
X^j_{t_{i+1}} &= X^j_{t_i} + h \cdot B(t_i, X^j_{t_i}, Y^j_{t_i}, Z^j_{t_i}, \mathcal{L}(X^j_{t_i}, Y^j_{t_i}, Z^j_{t_i})) + \sigma \Delta W_i \\
X^j_0 &= \xi
\end{aligned}$$

- Initialize (X^0, Y^0, Z^0), then do Picard iteration $j = 1, ..., J$.

Mapping of Picard Iteration

- Picard iteration mapping Φ is defined by:

$$\begin{aligned}\Phi : &(X^j, Y^j, Z^j, \mathcal{L}(X^j, Y^j, Z^j)) \\ \mapsto &(X^{j+1}, Y^{j+1}, Z^{j+1}, \mathcal{L}(X^{j+1}, Y^{j+1}, Z^{j+1}))\end{aligned} \tag{6}$$

- Picard iteration mapping Φ is defined by:

$$\begin{aligned}\Phi : &(X^j, Y^j, Z^j, \mathcal{L}(X^j, Y^j, Z^j)) \\ &\mapsto (X^{j+1}, Y^{j+1}, Z^{j+1}, \mathcal{L}(X^{j+1}, Y^{j+1}, Z^{j+1}))\end{aligned} \tag{6}$$

- The Lipschitz coefficient K_Φ is an *increasing function* of ρ and T, where ρ is the coupling coefficient between the forward and backward processes and T is the time horizon.

Mapping of Picard Iteration

- Picard iteration mapping Φ is defined by:

$$\begin{aligned} \Phi : &(X^j, Y^j, Z^j, \mathcal{L}(X^j, Y^j, Z^j)) \\ &\mapsto (X^{j+1}, Y^{j+1}, Z^{j+1}, \mathcal{L}(X^{j+1}, Y^{j+1}, Z^{j+1})) \end{aligned} \tag{6}$$

- The Lipschitz coefficient K_Φ is an *increasing function* of ρ and T, where ρ is the coupling coefficient between the forward and backward processes and T is the time horizon.
- $K_\Phi < 1 \Rightarrow \Phi$ is a contraction $\Rightarrow$ Picard iteration converges.

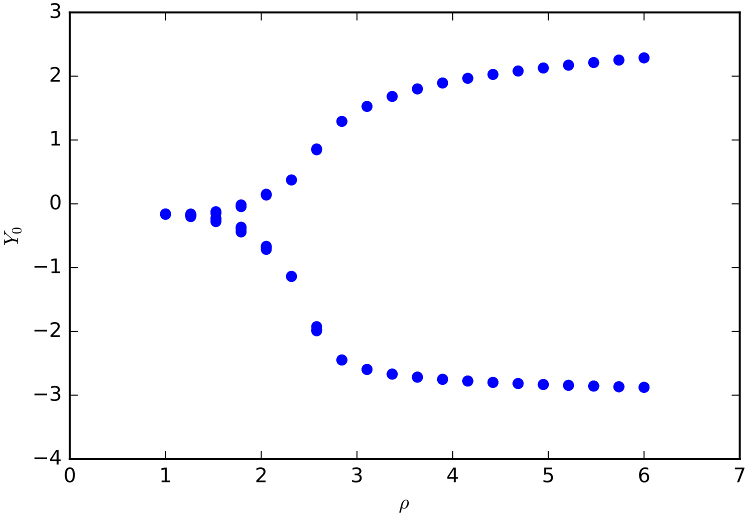
Mapping of Picard Iteration

- Picard iteration mapping Φ is defined by:

$$\begin{aligned} \Phi : &(X^j, Y^j, Z^j, \mathcal{L}(X^j, Y^j, Z^j)) \\ &\mapsto (X^{j+1}, Y^{j+1}, Z^{j+1}, \mathcal{L}(X^{j+1}, Y^{j+1}, Z^{j+1})) \end{aligned} \tag{6}$$

- The Lipschitz coefficient K_Φ is an *increasing function* of ρ and T, where ρ is the coupling coefficient between the forward and backward processes and T is the time horizon.
- $K_\Phi < 1 \Rightarrow \Phi$ is a contraction $\Rightarrow$ Picard iteration converges.
- Picard iteration convergence only guaranteed when ρ and T are not too large.

Bifurcation



Outline

Continuation in Time

- Divide the time horizon $[0, T]$ into N levels:
  $0 = T_1 \leq T_2, ..., T_{N-1} \leq T_N = T$.

Continuation in Time

- Divide the time horizon $[0, T]$ into N levels:
  $0 = T_1 \leq T_2, ..., T_{N-1} \leq T_N = T$.
- Let *picard* be the solver that applies the Picard iteration for level k:

$$(X_t, Y_t, Z_t)_{T_k \leq t \leq T_{k+1}} = picard(X, Y_{T_{k+1}}, Z_{T_{k+1}}) \tag{7}$$

Continuation in Time

- Divide the time horizon $[0, T]$ into N levels:
  $0 = T_1 \leq T_2, ..., T_{N-1} \leq T_N = T$.
- Let *picard* be the solver that applies the Picard iteration for level k:

$$(X_t, Y_t, Z_t)_{T_k \leq t \leq T_{k+1}} = picard(X, Y_{T_{k+1}}, Z_{T_{k+1}}) \tag{7}$$

- Define global *solver* recursively:
  $solver[k](X_{T_k})$:

Continuation in Time

- Divide the time horizon $[0, T]$ into N levels:
  $0 = T_1 \leq T_2, \ldots, T_{N-1} \leq T_N = T$.
- Let *picard* be the solver that applies the Picard iteration for level k:

$$(X_t, Y_t, Z_t)_{T_k \leq t \leq T_{k+1}} = picard(X, Y_{T_{k+1}}, Z_{T_{k+1}}) \tag{7}$$

- Define global *solver* recursively:
  $solver[k](X_{T_k})$:
  1. Initialize (X,Y,Z)
  2. For $1 \leq j \leq J$
     1. $Y_{T_{k+1}} = solver[k+1](X_{T_{k+1}})$
     2. $(X, Y, Z) = picard(X, Y_{T_{k+1}}, Z_{T_{k+1}})$
  3. return Y_{T_k}.

  where $solver[N] = g(X_T)$.

Continuation in Time

- Divide the time horizon $[0, T]$ into N levels:
  $0 = T_1 \leq T_2, ..., T_{N-1} \leq T_N = T$.
- Let *picard* be the solver that applies the Picard iteration for level k:

$$(X_t, Y_t, Z_t)_{T_k \leq t \leq T_{k+1}} = picard(X, Y_{T_{k+1}}, Z_{T_{k+1}}) \tag{7}$$

- Define global *solver* recursively:
  $solver[k](X_{T_k})$:
  1. Initialize (X,Y,Z)
  2. For $1 \leq j \leq J$
     1. $Y_{T_{k+1}} = solver[k+1](X_{T_{k+1}})$
     2. $(X, Y, Z) = picard(X, Y_{T_{k+1}}, Z_{T_{k+1}})$
  3. return Y_{T_k}.

  where $solver[N] = g(X_T)$.
- Then $Y_0 = solver[0](\xi)$.

Outline

1. Mean Field Games
2. Algorithms for solving FBSDEs
   - Picard Iteration
   - Continuation in Time
   - Tree Algorithm
   - Grid Algorithm
3. Benchmark Examples
   - Toy Example
   - Trader Problem

Binomial Tree

- Binomial approximation of Brownian increments:

$$ {}^{\uparrow\downarrow}X^j_{t_{i+1}} = X^j_{t_i} + h \cdot B(t_i, X^j_{t_i}, Y^j_{t_i}, Z^j_{t_i}, \mathcal{L}(X^j_{t_i}, Y^j_{t_i}, Z^j_{t_i}) \pm \sigma\sqrt{h} \tag{8} $$

Binomial Tree

- Binomial approximation of Brownian increments:

$$^{\uparrow\downarrow}X^j_{t_{i+1}} = X^j_{t_i} + h \cdot B(t_i, X^j_{t_i}, Y^j_{t_i}, Z^j_{t_i}, \mathcal{L}(X^j_{t_i}, Y^j_{t_i}, Z^j_{t_i}) \pm \sigma\sqrt{h} \qquad (8)$$

- For each starting value x_m of X_0 we create a binomial tree with depth equal to the number of time steps.

Binomial Tree

- Binomial approximation of Brownian increments:

$$ {}^{\uparrow\downarrow}X^j_{t_{i+1}} = X^j_{t_i} + h \cdot B(t_i, X^j_{t_i}, Y^j_{t_i}, Z^j_{t_i}, \mathcal{L}(X^j_{t_i}, Y^j_{t_i}, Z^j_{t_i}) \pm \sigma\sqrt{h} \qquad (8) $$

- For each starting value x_m of X_0 we create a binomial tree with depth equal to the number of time steps.
- Each node has a value for (X, Y, Z). To get the distribution of $(X_{t_i}, Y_{t_i}, Z_{t_i})$, we look at the values on the nodes at depth i.

Binomial Tree

- Binomial approximation of Brownian increments:

$$ {}^{\uparrow\downarrow}X^j_{t_{i+1}} = X^j_{t_i} + h \cdot B(t_i, X^j_{t_i}, Y^j_{t_i}, Z^j_{t_i}, \mathcal{L}(X^j_{t_i}, Y^j_{t_i}, Z^j_{t_i}) \pm \sigma\sqrt{h} \tag{8} $$

- For each starting value x_m of X_0 we create a binomial tree with depth equal to the number of time steps.
- Each node has a value for (X, Y, Z). To get the distribution of $(X_{t_i}, Y_{t_i}, Z_{t_i})$, we look at the values on the nodes at depth i.
- The backward scheme is easily calculated on the tree: the expectation conditional to the filtration of time t_i is given by the average of the value of "up" and "down" branches at time t_{i+1} from the node Y_{t_i}.

Outline

Grid for Discretized Distribution

- The uniform spatial grid of parameter $(\Delta x, x_{\min}, x_{\max})$ for the forward process is defined by:

$$\begin{aligned} \chi &= \{x_{min} = x_1 < \ldots < x_{N_x} = x_{max}\} \\ &= \{x_{min} + k\Delta x, k = 0, \ldots N_x - 1\} \end{aligned} \tag{9}$$

Grid for Discretized Distribution

- The uniform spatial grid of parameter $(\Delta x, x_{\min}, x_{\max})$ for the forward process is defined by:

$$\begin{aligned}\chi &= \{x_{min} = x_1 < \ldots < x_{N_x} = x_{max}\} \\ &= \{x_{min} + k\Delta x, k = 0, \ldots N_x - 1\}\end{aligned} \tag{9}$$

- The distribution of X_{t_i} is defined by the distribution of its projection on the grid χ:

$$\begin{aligned}\mathcal{L}(X_{t_i}) \approx \mathcal{L}(\Pi(X_{t_i})) &= \{p_k | p_k = \mathbb{P}(\Pi(X_{t_i}) = x_k), k = 1, \ldots, N_x\} \\ &\approx \{p_k | p_k = \mathbb{P}(X_{t_i} \in B(x_k, \Delta x/2), k = 1, \ldots, N_x\}\end{aligned} \tag{10}$$

$B(x, \Delta x/2)$ the closed ball centered at x of radius $\Delta x/2$.

Scheme of Forward Process

- As before, the forward process is given by the Euler scheme with binomial approximation of Brownian increments:

$$ {}^{\uparrow\downarrow}\bar{X}_{t_{i+1}}^{t_i,x} = x + h \cdot b(x, \mu_i, Y_{t_i}, Z_{t_i}, \mathcal{L}(Y_{t_i}, Z_{t_i})) \pm \sigma\sqrt{h} \qquad (11) $$

Scheme of Forward Process

- As before, the forward process is given by the Euler scheme with binomial approximation of Brownian increments:

$$ {}^{\uparrow\downarrow}\bar{X}_{t_{i+1}}^{t_i,x} = x + h \cdot b(x, \mu_i, Y_{t_i}, Z_{t_i}, \mathcal{L}(Y_{t_i}, Z_{t_i})) \pm \sigma\sqrt{h} \qquad (11) $$

- For $x \in \chi$, the distribution of $X_{t_{i+1}}$ conditional to $\{X_{t_i} = x\}$ is approximated by $\mathcal{L}(\Pi(\bar{X}_{t_{i+1}}^{t_i,x}))$. Starting from the initial distribution $\mu^0 = \xi$, the sequence $(\mu^i)_{i=1,\dots,N_t}$ of discretized probability distributions is defined recursively as:

$$ \mu^{i+1} = \mathcal{L}(\Pi(\bar{X}_{t_{i+1}}^{t_i,x}) | \mathcal{L}(x) = \mu^i) \qquad (12) $$

Forward Distribution on the Grid

- For $l = 1, ..., N_x$:

$$\begin{aligned}
&\mathbb{P}_{t_i}[X_{t_{i+1}}^{t_i,x_k} \in B(x_l, \Delta x/2)] \\
\approx&\mathbb{P}_{t_i}[\Pi(\bar{X}_{t_{i+1}}^{t_i,x_k}) = x_l] = \mathbb{P}_{t_i}[\bar{X}_{t_{i+1}}^{t_i,x_k} \in B(x_l, \Delta x/2)] \\
\approx&1/2 \cdot [\mathbf{1}\{^{\uparrow}\bar{X}_{t_{i+1}}^{t_i,x_k} \in B(x_l, \Delta x/2)\} + \mathbf{1}\{^{\downarrow}\bar{X}_{t_{i+1}}^{t_i,x_k} \in B(x_l, \Delta x/2)\}]
\end{aligned} \tag{13}$$

Forward Distribution on the Grid

- For $l = 1, \dots, N_x$:

$$\begin{aligned} &\mathbb{P}_{t_i}[X_{t_{i+1}}^{t_i,x_k} \in B(x_l, \Delta x/2)] \\ \approx &\mathbb{P}_{t_i}[\Pi(\bar{X}_{t_{i+1}}^{t_i,x_k}) = x_l] = \mathbb{P}_{t_i}[\bar{X}_{t_{i+1}}^{t_i,x_k} \in B(x_l, \Delta x/2)] \\ \approx &1/2 \cdot [\mathbf{1}\{{}^{\uparrow}\bar{X}_{t_{i+1}}^{t_i,x_k} \in B(x_l, \Delta x/2)\} + \mathbf{1}\{{}^{\downarrow}\bar{X}_{t_{i+1}}^{t_i,x_k} \in B(x_l, \Delta x/2)\}] \end{aligned} \tag{13}$$

- To get the probability distribution $\mu_{i+1} = \{p_1^{i+1}, \dots, p_{N_x}^{i+1}\}$ at time t_{i+1}, we sum over k with respect to p_k^i:

$$p_l^{i+1} = \sum_{k=1}^{N_x} \frac{p_k^i}{2} [\mathbf{1}\{{}^{\uparrow}\bar{X}_{t_{i+1}}^{t_i,x_k} \in B(x_l, \Delta x/2)\} + \mathbf{1}\{{}^{\downarrow}\bar{X}_{t_{i+1}}^{t_i,x_k} \in B(x_l, \Delta x/2)\}] \tag{14}$$

Scheme of Backward Process

- We suppose there exist functions u and v such that:

$$(Y_t, Z_t) = (u, v)(t, X_t, \mathcal{L}(X_t))$$

Scheme of Backward Process

- We suppose there exist functions u and v such that:

$$(Y_t, Z_t) = (u, v)(t, X_t, \mathcal{L}(X_t))$$

- For each time step t_i, we compute values of $(u, v)(t_i, x, \mu^i)$ for $x \in \chi$ and $supp(\mu^i) = \chi$. The values of (Y_{t_i}, Z_{t_i}) are approximated as following:

$$(Y_{t_i}, Z_{t_i}) \approx (u, v)(t_i, x, \mu^i) \text{ for } \Pi(X_{t_i}) = x \text{ and } \mathcal{L}(\Pi(X_{t_i})) = \mu^i \quad (15)$$

Scheme of Backward Process

- We suppose there exist functions u and v such that:

$$(Y_t, Z_t) = (u, v)(t, X_t, \mathcal{L}(X_t))$$

- For each time step t_i, we compute values of $(u, v)(t_i, x, \mu^i)$ for $x \in \chi$ and $supp(\mu^i) = \chi$. The values of (Y_{t_i}, Z_{t_i}) are approximated as following:

$$(Y_{t_i}, Z_{t_i}) \approx (u, v)(t_i, x, \mu^i) \text{ for } \Pi(X_{t_i}) = x \text{ and } \mathcal{L}(\Pi(X_{t_i})) = \mu^i \quad (15)$$

- The backward scheme then becomes:

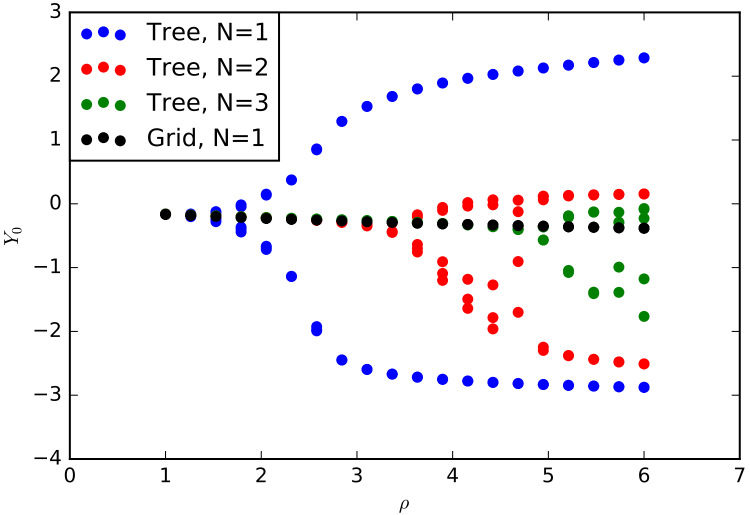
$$\begin{aligned} u(t_i, x, \mu_i) &\approx \frac{1}{2}[u(t_{i+1}, \Pi(^{\uparrow}\bar{X}^{t_i,x}_{t_{i+1}}), \mu_{i+1}) + u(t_{i+1}, \Pi(^{\downarrow}\bar{X}^{t_i,x}_{t_{i+1}}), \mu_{i+1})] \\ &\quad + h \cdot f(X_{t_i}, Y_{t_i}, Z_{t_i}, \mathcal{L}(X_{t_i}, Y_{t_i}, Z_{t_i})) \\ v(t_i, x, \mu_i) &\approx \frac{1}{2}h^{-1/2}[u(t_{i+1}, \Pi(^{\uparrow}\bar{X}^{t_i,x}_{t_{i+1}}), \mu_{i+1}) - u(t_{i+1}, \Pi(^{\downarrow}\bar{X}^{t_i,x}_{t_{i+1}}), \mu_{i+1})] \end{aligned} \quad (16)$$

Outline

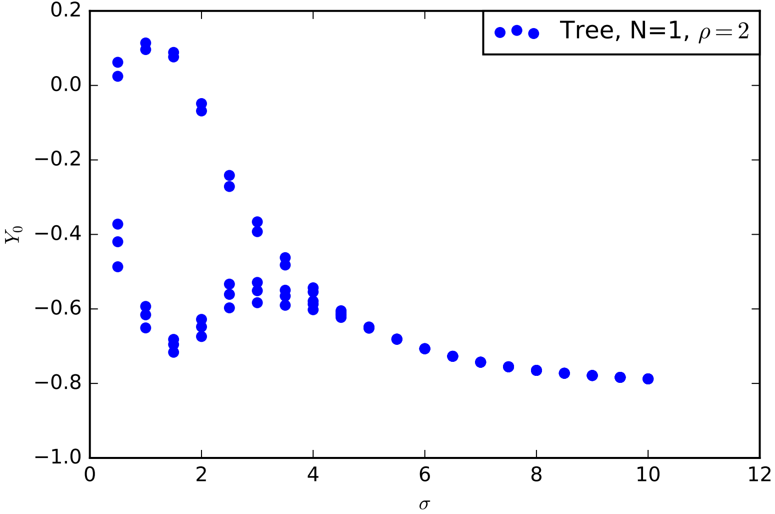
Toy Example

$$
\begin{aligned}
dX_t &= -\rho Y_t dt + \sigma dW_t \\
X_0 &= x_0 = 2 \\
dY_t &= \arctan\left(\mathbb{E}(X_t)\right) dt + Z_t dW_t \\
Y_T &= \arctan\left(X_T\right)
\end{aligned}
\tag{17}
$$

Toy Example: Continuation in Time



Toy Example: Effect of σ



Outline

Trader Example

- X_t represents the number of shares owned at time t:

$$dX_t = \alpha_t dt + \sigma dW_t, \quad t \in [0, T]. \tag{18}$$

Trader Example

- X_t represents the number of shares owned at time t:

$$dX_t = \alpha_t dt + \sigma dW_t, \quad t \in [0, T]. \tag{18}$$

- The trader controls his/her trading velocity, α_t, to minimize the cost:

$$J(\alpha) = \mathbb{E}\left[\int_0^T \left(\frac{c_\alpha}{2}\alpha_t^2 + \frac{c_X}{2}X_t^2 - \bar{h}\mathbb{E}(\alpha_t)X_t\right) dt + \frac{c_g}{2}X_T^2\right] \tag{19}$$

Trader Example

- X_t represents the number of shares owned at time t:

$$dX_t = \alpha_t dt + \sigma dW_t, \quad t \in [0, T]. \tag{18}$$

- The trader controls his/her trading velocity, α_t, to minimize the cost:

$$J(\alpha) = \mathbb{E}\left[\int_0^T \left(\frac{c_\alpha}{2}\alpha_t^2 + \frac{c_X}{2}X_t^2 - \bar{h}\mathbb{E}(\alpha_t)X_t\right) dt + \frac{c_g}{2}X_T^2\right] \tag{19}$$

- The stock price is influenced by the average trading velocity of all of the players. Thus, the players interact through the empirical distribution of their controls.

Trader Example

- X_t represents the number of shares owned at time t:

$$dX_t = \alpha_t dt + \sigma dW_t, \quad t \in [0, T]. \tag{18}$$

- The trader controls his/her trading velocity, α_t, to minimize the cost:

$$J(\alpha) = \mathbb{E}\left[\int_0^T \left(\frac{c_\alpha}{2}\alpha_t^2 + \frac{c_X}{2}X_t^2 - \bar{h}\mathbb{E}(\alpha_t)X_t\right) dt + \frac{c_g}{2}X_T^2\right] \tag{19}$$

- The stock price is influenced by the average trading velocity of all of the players. Thus, the players interact through the empirical distribution of their controls.
- This is an example of an *Extended Mean Field Game*.

Trader Example: Two Formulations

- Weak Formulation:

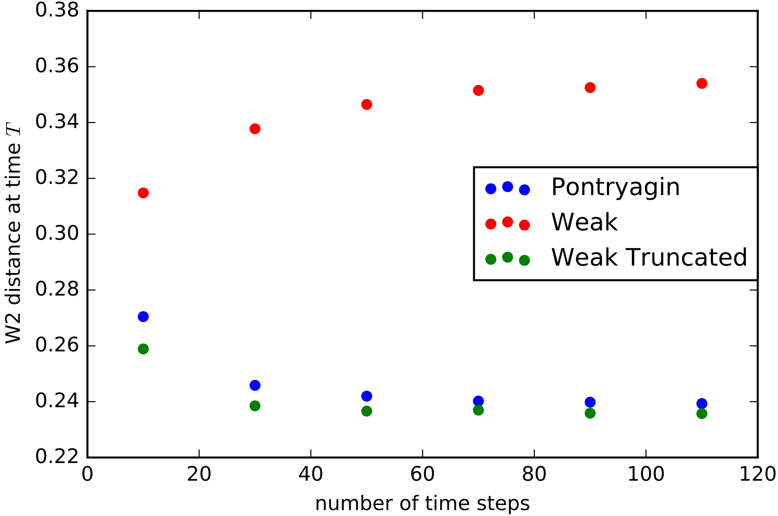
$$dX_t = -\frac{1}{c_\alpha}\frac{Z_t}{\sigma}dt + \sigma dW_t, X_0 = x_0$$

$$dY_t = -\left[\frac{c_X}{2}X_t^2 + \frac{\bar{h}}{c_\alpha}\frac{\mathbb{E}[Z_t]}{\sigma}X_t + \frac{1}{2c_\alpha}\left(\frac{Z_t}{\sigma}\right)^2\right]dt + Z_t dW_t, Y_T = c_g\frac{X_T^2}{2} \tag{20}$$

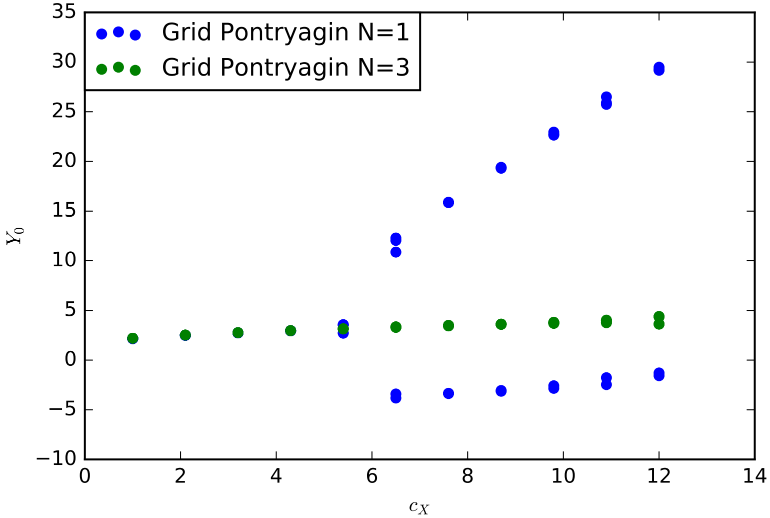
- Pontryagin Formulation:

$$\begin{aligned} dX_t &= -\frac{1}{c_\alpha}Y_t dt + \sigma dW_t, X_0 = x_0 \\ dY_t &= -(c_X X_t + \frac{\bar{h}}{c_\alpha}\mathbb{E}[Y_t])dt + Z_t dW_t, Y_T = c_g X_T \end{aligned} \tag{21}$$

Trader Example: W_2 Distance



Trader Example: Continuation in Time



Comparison of Tree and Grid Algorithms

- Advantages of tree algorithm:

Comparison of Tree and Grid Algorithms

- Advantages of tree algorithm:
    1. No error of discretization or truncation. No need to choose grid parameters, as the stability condition that links Δx and h is unknown and the a priori bounds $x_{\min}$ and $x_{\max}$ are difficult to estimate in practice.

Comparison of Tree and Grid Algorithms

- Advantages of tree algorithm:
  1. No error of discretization or truncation. No need to choose grid parameters, as the stability condition that links Δx and h is unknown and the a priori bounds $x_{\min}$ and $x_{\max}$ are difficult to estimate in practice.
  2. The process defined on the tree is pathwise, extendable to mean field games with common noise.

Comparison of Tree and Grid Algorithms

- Advantages of tree algorithm:
  1. No error of discretization or truncation. No need to choose grid parameters, as the stability condition that links Δx and h is unknown and the a priori bounds $x_{\min}$ and $x_{\max}$ are difficult to estimate in practice.
  2. The process defined on the tree is pathwise, extendable to mean field games with common noise.
- Advantages of grid algorithm:

Comparison of Tree and Grid Algorithms

- Advantages of tree algorithm:
  1. No error of discretization or truncation. No need to choose grid parameters, as the stability condition that links Δx and h is unknown and the a priori bounds $x_{\min}$ and $x_{\max}$ are difficult to estimate in practice.
  2. The process defined on the tree is pathwise, extendable to mean field games with common noise.
- Advantages of grid algorithm:
  1. For a given computational capacity, it is much easier to have a large number of time steps in grid algorithm than in tree algorithm where the complexity grows exponentially with number of time steps.

Conclusion

- Implemented two algorithms (*tree* and *grid*) to solve FBSDEs using a Picard iteration.

Conclusion

- Implemented two algorithms (*tree* and *grid*) to solve FBSDEs using a Picard iteration.
- Extended these algorithms via continuation in time.

Conclusion

- Implemented two algorithms (*tree* and *grid*) to solve FBSDEs using a Picard iteration.
- Extended these algorithms via continuation in time.
- Tested the algorithms on a variety of benchmark examples.

Conclusion

- Implemented two algorithms (*tree* and *grid*) to solve FBSDEs using a Picard iteration.
- Extended these algorithms via continuation in time.
- Tested the algorithms on a variety of benchmark examples.
- Future Work:
  1. Error analysis of two algorithms
  2. Estimation of grid parameters $(\Delta x, x_{\min}, x_{\max})$
  3. Investigate the effect of σ in the convergence of Picard iteration
  4. Application of tree algorithm to Mean Field Games with common noise

References

R. Carmona, and F. Delarue. Probabilistic approach to mean field games, Volume 1. *Springer*, to appear in 2017.

J.F. Chassagneux, D. Crisan, and F. Delarue Numerical Method for FBSDEs of McKean-Vlasov Type. Arxiv.

F. Delarue and S. Menozzi (2006) A forward-backward stochastic algorithm for quasi-linear PDEs *Ann. Appl. Probab.*, 16 (1), pp 140-184.

Questions?