FBSDE (forward-backward stochastic differential equations) and Hierarchical Tensors

Reinhold Schneider
TU Berlin
ongoing joint work with L. Sallandt, M. Oster and P. Trunschke (TU Berlin)
(Deep Neural Networks: Niklas Nüsken U Potsdam, Lorenz Richter ZIP Berlin - U Cottbus)

DFG Research Center MATHEON
Mathematics for key technologies

Motivation

(control affine) optimal control problem

$$\begin{aligned} dx(t) &= (f(t,x(t)) + g(t,x(t))u(t))dt + \Sigma(t,x(t))dW_t \\ x(t_0) &= x \in \mathbb{R}^d \ , \ t_0 \le t \le T \\ \mathcal{J}(x,u) &:= \mathbb{E}[\int_{t_0}^{T} (\|\ell(x(t)) + \lambda |u(t)|^2)dt + v_T(x(T))|x(t_0) = x] \end{aligned}$$

Value function - for policy $\alpha(t,x(t)) = u(t)|x(t_0) = x$

$$v_\alpha(t_0,x) := \mathcal{J}(x,\alpha(x)) \ , \ v(t_0,x) = \min_{\alpha \in \mathcal{A}} v_\alpha(t_0,x)$$

HJB (Hamilton Jacobi Bellmann equation)

$$\begin{aligned} \partial_t v + inf_{u \in \mathcal{A}} \ [\frac{1}{2}\Sigma\Sigma^T : \nabla^2 v + (f+gu)\nabla v + \ell + \lambda|u|^2)] &= 0 \\ \text{terminal condition} \ \ v(T,x) &= v_T(x) \end{aligned}$$

Motivation

Optimality condition

$$\frac{-1}{2\lambda} g(t,x(t))^T \nabla v(t,x(t)) \quad = \quad u(t) = \alpha(t,x(t))$$

$$\boxed{\partial_t v + [\frac{1}{2}\Sigma\Sigma^T : \mathcal{D}^2 v + f\mathcal{D}v + \ell - \frac{1}{2\lambda}|g^T\mathcal{D}v|^2] = 0}$$

- terminal condition $v(T,x) = v_T \Rightarrow$ Backward non-linear hyperbolic or parabolic PDE.
- Representing the differential operator in tensor product form can be extremely DIFFICULT
- We need other concepts.

Remark: Recent progress has been obtained using DNN (deep neural networks). A. Jentzen, Weinan E, P. Grohs, Warin, Pham et al.

Scalar Linear Hyperbolic PDE

Let us consider the homogenous linear PDE in $\mathbb{R}^d$

$$\partial_t v(t, x(t)) + f(t, x(t)) \cdot \nabla v = 0$$

$$\partial_t v + Lv = \partial_t v + \sum_{j=1}^{d} f_j \partial_{x_j} v = 0$$

with terminal condition

$$v(T, x) = v_T(x)$$

the *characteristic curves* are defined by the trajectories of the IVP

$$\begin{aligned} \dot{x}(t) &= f(t, x(t)) \,, \; t_0 \leq t \leq T \\ x(t_0) &= x \end{aligned}$$

Scalar Linear Hyperbolic PDE

Let us define

$$t \mapsto y(t) := v(t, x(t))$$

$$\begin{aligned}
\Rightarrow \quad \dot{y}(t) &= \partial_t v(t, x(t)) + \nabla v(t, x(t)) \cdot \dot{x}(t) \\
&= \partial_t v(t, x(t)) + v(t, x(t)) \cdot f(t, x(t)) \\
&= \partial_t v(t, x(t)) + Lv(t, x(t)) \\
&= 0 \\
y(T) &= v_T(x(T))
\end{aligned}$$

Flow of the dynamical system

$$\phi_t : \mathbb{R}^d \to \mathbb{R}^d \ , \ \phi_t(x) := x(t) \ , \quad \text{where } x(t_0) = x$$

This part is simplified for autonomous systems, i.e. $f(t, x) = f(x)$

Koopman operator $K_s := e^{-sL}$

$$K_{t-t_0} v(x) = v(\phi_t(x)) = v \circ \phi_t(x) = v(x(t))$$

Forward-Backward Systems - Deterministic

Deterministic: given forward-backward ODE system
$y(t) := v(t, x(t))$

$$\begin{aligned} \frac{dx}{dt}(t) &= f(t, x(t)) \ , \ x(t_0) = x, \ 0 \le t \le T \\ \frac{dy}{dt}(t) &= 0 \ , \ y(T) := v_T(x(T)), \end{aligned}$$

- with initial and terminal condition

$$x(t) =: x \ , \ y(T) = v_T(x(T)) \ , 0 \le t \le T.$$

corresponds to

$$\boxed{\partial_t v + Lv = 0 \ , \ v(T, x) = v_T(x)}$$

$$\text{where} \quad Lv = f \cdot \mathcal{D}v$$

Scalar Linear Parabolic PDE

Let us consider the homogenous linear PDE in $\mathbb{R}^d$

$$\partial_t v + Lv = \partial_t v + \sum_{j=1}^{d} f_j \partial_{x_j} v + \frac{1}{2}\Sigma^T \Sigma : \mathcal{D}^2 v = 0$$

with terminal condition

$$v(T, x) = v_T(x)$$

the characteristic curves correspond to solution of SDE's

$$\begin{aligned} dX_t &= f(t, X_t)dt + \Sigma(t, X_t)dW_t \ , \ t_0 \le t \le T \\ X_{t_0} &= x \end{aligned}$$

where X_t is $\mathcal{F}_t$ adapted process ($\mathcal{F}_t \subset \mathcal{F}_s$, $t \le s$)

Linear Parabolic PDE

Let us define

$$t \mapsto Y_t := v(t, X_t)\ ,\ x \mapsto v(t,x) : \mathbb{R}^d \to \mathbb{R}$$

then then Ito's formular yields

$$dY_t \ = \ \partial_t v + \mathcal{D}v \cdot f + \frac{1}{2}\Sigma^T\Sigma : \mathcal{D}^2 v + \mathcal{D}v \cdot \Sigma dW_t$$

There holds $v(t,x) = \mathbb{E}[Y_t|\mathcal{F}_t] \Rightarrow$ Feynman Kac theorem

$$\boxed{\partial_t v + Lv = \partial_t v + \mathcal{D}v \cdot f + \frac{1}{2}\Sigma^T\Sigma : \mathcal{D}^2 v = 0}$$

since $\mathbb{E}[\Sigma\mathcal{D}v \cdot dW_s|\mathcal{F}_t] = 0\ ,\ t \leq s$

Koopman operator $K_s := e^{-sL}$

$$K_{t-t_0} v(x) = \mathbb{E}[v(X_t)|X_{t_0} = x]$$

Forward-Backward Systems

Stochastic forward-backward SDE system $Y_t := v(t, X_t)$

Let us define

$$Z_t := \Sigma^T \mathcal{D} v(t, X_t)$$

$$\Rightarrow \; dY_t \;=\; Z_s \cdot dW_s$$

where X_t, Y_t, Z_t are $\mathcal{F}_t$ adapted

$$\begin{aligned} dX_t &= f(t, X_t)dt + \Sigma(t, X_t)dW_t \;,\; x \in \mathbb{R}^d \;,\; t_0 \leq t \leq T \\ dY_t &= \sum_{j=1} Z_{j,t} dW_{j,t} \;,\; Y_T := v_T(X_T), \end{aligned}$$

where X_s, Y_s, Z_s are $\mathcal{F}_s$ adapted

Corresponds to (Feynman-Kac) Backward Kolmogorov equation

$$\boxed{\partial_t v + Lv = 0 \;,\; v(T, x) = v_T(x)}$$

$$\text{where} \;\; Lv = f \cdot \mathcal{D} v + \frac{1}{2} \Sigma^T \Sigma : \mathcal{D}^2 v$$

Forward-Backward Systems - Inhomogenous case

Stochastic forward-backward SDE system $Y_t := v(t, X_t)$

Let us define

$$Z_t := \Sigma^T \mathcal{D} v(t, X_t)$$

$$\Rightarrow \; dY_t \;=\; Z_s \cdot dW_s$$

where X_t, Y_t, Z_t are $\mathcal{F}_t$ adapted

$$\begin{aligned} dX_t &= f(t, X_t)dt + \Sigma(t, X_t)dW_t \;,\; x \in \mathbb{R}^d \;,\; t_0 \leq t \leq T \\ dY_t &= -h(t, X_t)dt + \sum_{j=1} Z_{j,t} dW_{j,t} \;,\; Y_T := v_T(X_T), \end{aligned}$$

where X_s, Y_s, Z_s are $\mathcal{F}_s$ adapted

Corresponds to (Feynman-Kac)

$$\boxed{\partial_t v + Lv = h \;,\; v(T, x) = v_T(x)}$$

$$\text{where} \quad Lv = f \cdot \mathcal{D} v + \frac{1}{2} \Sigma^T \Sigma : \mathcal{D}^2 v$$

Forward-Backward Systems - Nonlinear

Let us define

$$Y_t := v(t, X_t) \ , \ Z_t := \Sigma^T \mathcal{D} v(t, X_t)$$

For regular Σ, and $z := \Sigma \mathcal{D} v$ we can write the non-linearity

$$\tilde{h}(t, x, v, \mathcal{D}v) =: h(t, x, v, z)$$

$\Rightarrow$ Stochastic forward-backward SDE (uncoupled) system

$$\begin{aligned} dX_t &= f(t, X_t)dt + \Sigma(t, X_t)dW_t \ , \ X_{t_0} = x \ , \ t_0 \leq t \leq T \\ dY_t &= -h(t, X_t, Y_t, Z_t)dt + Z_t \cdot dW_t \ , \ Y_T := v_T(X_T), \end{aligned}$$

where X_s, Y_s, Z_s are $\mathcal{F}_s$ adapted processes.
v solves the nonlinear PDE (non-linear Feynman Kac theorem),
e.g. Pardoux et al.

$$\boxed{\partial_t v(t, x) + Lv(t, x) + \tilde{h}(t, x, v(t, x), \mathcal{D}v(t, x)) = 0 \ , \ v(T, x) = v_T(x)}$$

FBSED - existence and uniqueness

Assumptions: (can be relaxed slightly)

- $f, \Sigma, h \in C^k$; (e.g. $k = 3$, for non-linear F-K thm.)
- Uniformly parabolic (for $\tilde{h} \sim h$)

$$w^T \Sigma(t,x) w \geq \gamma \|w\|^2, \forall w \in \mathbb{R}^d$$

- Lipschitz conditions for forward SDE

$$\|f(t,x_1) - f(t,x_2)\| + \|\Sigma(t,x_1) - \Sigma(t,x_2)\| \leq L\|x_1 - x_2\|$$

- growth conditions

$$\|f(t,x)\|^2 + \|\Sigma(t,x)\|^2 \leq c(1 + \|x\|^2)$$

- Lipschitz conditions for backward SDE

$$\|h(t,x_1,y_1,z_1) - h(t,x_2,y_2,z_2)\| \leq L(\|x_1 - x_2\| + \|y_1 - y_2\| + \|z_1 - z_2\|)$$

- growth condition

$$h(t,x,y,z)\| \leq C(1 + \|y\|^q + \|z\|^q) \ , \ q > 1?$$

$\Rightarrow$: existence and uniqueness of X_s, Y_s, Z_s

Time stepping for the forward dynamic

on a time grid

$$t_0 < t_1 < \dots t_n < \cdots < t_M = T \quad , \quad \widehat{X}_n \approx X_{t_n}$$

where $n \in \{0, \dots, M-1\}$ enumerates the steps,
by the Euler Mayurana scheme

For $n = 0, \dots, M$ do

$$\boxed{\widehat{X}_{n+1} = \widehat{X}_n + f(\widehat{X}_n, t_n)\Delta t + \Sigma(\widehat{X}_n, t_n)\xi_{n+1}\sqrt{\Delta t}}$$

where

- $\Delta t = t_{n+1} - t_n$ is the stepsize,
- $\xi_{n+1} \sim \mathcal{N}(0, I_{d\times d})$ are normally distributed random variables
- $\widehat{X}_0 = x_0$

Mild Solution - Numerical Approximation

Koopman operator $K_s := e^{-sL}$

$$K_{t-t_0}v(x) = [v(X_t)|X_{t_0} = x]$$

$$\begin{aligned} v(t,x) &= K_{T-t}v_T(x) + \int_t^T K_{T-s}\tilde{h}(s,x,v,\mathcal{D}v)ds \\ v(t,x) &= v_T(x(T)) + \int_t^T \tilde{h}(s,x(s),v(t,x(s)),\mathcal{D}v(t,x(s)))ds \end{aligned}$$

These formulars are exact , at least for $v \in C^2$.

- For numerical computation one need appropriate ODE or SDE solver's and time integration.
- In the linear case: Given fixed $x \in \Omega, t \in [t_0, T]$, the value $v(t,x)$ can be computed by Euler Mayurana scheme, without curse dimensionality.
- But in the nonlinear case: if we need $\mathcal{D}v(t,x)$, we ask for a function $x \in \Omega \mapsto v(t,x)$. Then we face the curse of dimensions.

Remark: the Euler Mayurana scheme can be approximately written be a DNN

(Jentzen er al.)

Numerical solution - Generalized Semi-Lagrangian Scheme

To get a global (local) function $x \in \Omega \mapsto v(t,x)$, $\Omega \subset \mathbb{R}^d$, we consider

$$x^i \in \Omega \ , \ \ i = 1, \dots, N \ \ \text{(sample points)}.$$

Standard Algorithm:

`For` $n = M-1, \dots, 0$ `do:` - backstapping

- Step 1: Transportation of states, compute $K_{\Delta t} v$ by ODE (SDE) -solve of forward system and RHS $h(t_n + \Delta t, ...)$ by quadrature $\Rightarrow v(t, x^i)$, $i = 1, \dots, N$

  (change between Eulerian and Lagrangian coordinates)
- Step 2: Recover the function $x \mapsto v(t,x)$ by "interpolation"

  $$v_n(x^i) \approx v(t_n, x^i) \ , \ \text{for all} \ \ x^i \in \Omega \subset \mathbb{R}^d$$

  using an appropriate model set, $\mathcal{M}$ e.g. DNN or HT with multi-polynomials etc.
- Step 3: update Z_{t_n} by

  $$Z_{t_n} \approx \Sigma^T \nabla v_n(X_{t_n})$$

Variational Monte Carlo - Surrogate Problem

Given a target functional $\mathcal{R} : \mathcal{V} \to \mathbb{R}$ find

$$\Psi = \operatorname{argmin} \left\{ \mathcal{R}(W) : W \in \mathcal{V} \right\} .$$

Given further a model class, here HT/TT tensor representations

$$\mathcal{M} := \mathcal{M}_{\leq \mathbf{r}} \subset \mathcal{V} ,$$

Constraint problem (Ritz - Galerkin) find

$$\Psi_h := \operatorname{argmin} \left\{ \mathcal{R}(W) : W \in \mathcal{V} \cap \mathcal{M} \right\} .$$

Suppose that the numerical solution is not feasible. But there exist computable surrogate functionals $\mathcal{R}_N : \mathcal{M} \to \mathbb{R}$ e.g. by numerical quadrature, defined at least on the model classes $\mathcal{M}$, find

$$\Psi_N := \operatorname{argmin} \left\{ \mathcal{R}_N(W) : W \in \mathcal{M} \right\} .$$

Using Monte Carlo (MC) quadrature in high-dimensions is a canonical choice, but QMC ....

Tensor Representation $\mathcal{M}_{n,\leq r}$

Let $\phi_\ell : [a, b] = [a, b]_n \subset \mathbb{R} \to \mathbb{R}$, $\ell = 0, \ldots, p$,
be orthogonal polynomials

$$\phi : [a, b] \subset \mathbb{R} \to \mathbb{R}^m, \quad \phi(x) = [\phi_1(x), \ldots, \phi_m(x)],$$

the TT-representation of $\widehat{v}_n$ is given as

$$\widehat{v}_n(x) = \sum_{\ell_1=0}^{p} \cdots \sum_{\ell_d}^{p} \sum_{j_1=1}^{r_1} \cdots \sum_{j_{d-1}=1}^{r_{d-1}} u_1[\ell_1, j_1] u_2[j_1, \ell_2, j_2] \cdots$$
$$\cdots u_d[j_{d-1}, \ell_d] \phi(x_1)[\ell_1] \cdots \phi(x_d)[\ell_d]$$

$$\widehat{v}_n(x) \quad =$$

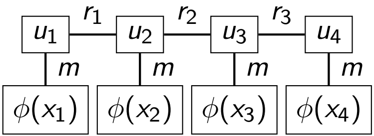


Figure: Graphical representation of $\widehat{v}_n : \Omega \subset \mathbb{R}^4 \to \mathbb{R}$.

Time stepping for the backward problem

times $t_0 < \cdots < t_n < t_{n+1} < \cdots < t_M$,
for simplicity: $\Delta t = t_{n+1} - t_n$ can be the same as before

$$Y_{t_{n+1}} = Y_{t_n} - \int_{t_n}^{t_{n+1}} h(X_s, s, Y_s, Z_s) ds + \int_{t_n}^{t_{n+1}} Z_s \cdot dW_s$$

shorthand notation $X_n :\approx X_{t_n}$, $Y_n :\approx Y_{t_n}$, $Z_n :\approx Z_{t_n}$

$$h_n := h(t_n, \widehat{X}_n, \widehat{Y}_n, \widehat{Z}_n) \quad \text{implicit}$$

$$h_{n+1} := h(t_{n+1}, \widehat{X}_{n+1}, \widehat{Y}_{n+1}, \widehat{Z}_{n+1}) \quad \text{explicit}$$

Backstapping:

For $n = M - 1$ until $n = 0$ do

$$\widehat{Y}_{n+1} = \widehat{Y}_n - h_{n+1} \Delta t + \widehat{Z}_n \cdot \xi_{n+1} \sqrt{\Delta t} \quad \text{explicit}$$

$$\widehat{Y}_{n+1} = \widehat{Y}_n - h_n \Delta t + \widehat{Z}_n \cdot \xi_{n+1} \sqrt{\Delta t} \quad \text{implicit}$$

e.g. explicit - formly

$$\boxed{\widehat{Y}_{n+1} = \widehat{Y}_n - h(t_{n+1}, \widehat{X}_{n+1}, t_{n+1}, \widehat{Y}_{n+1}, \widehat{Z}_{n+1}) \Delta t + \widehat{Z}_n \cdot \xi_{n+1} \sqrt{\Delta t}}$$

Time stepping for the backward problem

$\Rightarrow$ `For` $n = M - 1$ `until` $n = 0$ `do`

Step (2): Taking conditional expectations w.r.t. to the σ-algebra $\mathcal{F}_{t_n}$, yields

$$\widehat{Y}_n = \mathbb{E}\left[\widehat{Y}_{n+1} + h(\widehat{X}_{n+1}, t_{n+1}, \widehat{Y}_{n+1}, \widehat{Z}_{n+1})\Delta t \middle| \mathcal{F}_{t_n}\right].$$

The conditional expectation can be characterized as a best approximation in L^2, namely

$$\mathbb{E}[B|\mathcal{F}_{t_n}] = \underset{\substack{Y \in L^2 \\ \mathcal{F}_{t_n}-\text{measurable}}}{\operatorname{argmin}} \mathbb{E}\left[|Y - B|^2\right],$$

for any random variable $B \in L^2$, e.g. explicit method

$$\widehat{Y}_n = \underset{\substack{Y \in L^2 \\ \mathcal{F}_{t_n}-\text{measurable}}}{\operatorname{argmin}} \mathbb{E}\left[|Y - (\widehat{Y}_{n+1} + h_{n+1}\Delta t)|^2\right].$$

Step (3): Updating $\widehat{v}_n(\widehat{X}_n) := \widehat{Y}_n$ and

$$\widehat{Z}_n := \Sigma^T(t_n, \widehat{X}_n)\mathcal{D}_x v_n(\widehat{X}_n)$$

Variational Monte Carlo or MC Least Squares -

Model set (nonlinear manifold)

$$(\mathcal{M} =)\mathcal{M}_n = \mathcal{M}_{n,\mathbf{r}}$$

Risk minimization

$$v(t_n, x) = \mathbb{E}[v(t_n, X_{t_n})|\mathcal{F}_{t_n}] = \text{argmin}\ _{u\in L_2(\Omega_n,\rho_n)}\{\mathbb{E}[\|u(x) - Y_{t_n}\|^2|\mathcal{F}_{t_n}]\}$$

Empirical Risk Minimization (surogate functional $\mathcal{R}_N$)

$$x \mapsto \widehat{v}_n(x) := \text{argmin}\ _{u\in\mathcal{M}}\{\frac{1}{N}\sum_{i=1}^{N}[\|u(x_n^i) - (\widehat{Y}_{n+1} + \Delta t h_{n+1})\|^2|\widehat{X}_n =: x_n^i]\}$$

(regression problem)
Remark:

- other choices of $\mathcal{M}$ are possible, e.g DNN (deep neural networks), sparse polynomials, sparse grid, kernel methods
- alternative for updating $\widehat{Z}_n$: Malliavin derivative

$$\widehat{Z}_n := \frac{1}{\sqrt{\Delta t}}\mathbb{E}_{emp}[\widehat{Y}_{n+1}\xi_n] \approx \widehat{Z}_n = \Sigma^T(t_n, \widehat{X}_n)\mathcal{D}\widehat{v}_n(\widehat{X}_n)$$

Variational Monte Carlo (VMC) - Improvements

- Improved model space

$$\mathcal{M}_n := \mathrm{span}\{v_T\} \oplus \mathcal{M}_{n,\leq \mathbf{r}}$$

- Regularization (Important)

$$\widehat{v}_{n,k} := \mathrm{argmin}_{w\in\mathcal{M}_n}\{\mathcal{R}_N(w) + \mu_k\|w\|^2_{H^2_{mix}}\}$$

  e.g. $\mu_k \sim \mathcal{R}_N(\widehat{v}_{n,k})$ (or $\mu_k \sim \|\widehat{v}_{n,k-1} - \widehat{v}_{n,k}\|$), where $\mu_k \to 0$ for $k \to \infty$
  The cross norm $\|w\|_{H^2_{mix}}$ can be easily computed.
  This provides accurate gradients $\mathcal{D}\widehat{v}_n$!
- Change of drift term by adding

$$f_1\mathcal{D}v - \Sigma^{-1}f_1 Z = 0$$

  $\Rightarrow$ "importance" sampling - policy iteration.
- implicit methods: e.g *Hure, & Warin (2019)*

Possible: (idea of PINN: add PDE loss)

$$\widehat{\mathcal{R}_N}(w) := \mathcal{R}_N(w) + \mathbb{E}_{emp}[\int_\Omega |\partial_t w + Lw + \tilde{h}(t,x,w(t,x),\mathcal{D}w(t,x))|^2 \rho dx]$$

Numerical example - - HJB in $\mathbb{R}^{32}$

- 1D Schlögel equation

$$\begin{aligned} \dot{y} &= \sigma \Delta y + y^3 + \chi_\omega u \quad , \; x \in L^2(-1,1) \\ y(0) &= x \end{aligned}$$

with Neumann boundary condition and χ_ω is the characteristic function w.r.t. $\omega = [-0.4, 0.4] \subset [-1, 1]$.

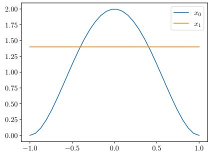
$$\begin{aligned} \dot{y} &= \sigma A y + y^3 + G u \\ y(0) &= x \end{aligned}$$

Using the step-size $h = \frac{1}{n+1}$ an $d = 32$ spatial dimensions. $\mathbf{r} =$

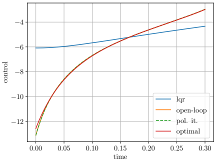
$[3, 4, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 6, 6, 6, 6, 6, 6, 6, 5, 5, 5, 4, 3]$

$\Omega = [-2, 2]^n, p = 4.$ full ansatz space has dimension $5^{32} \approx 10^{22}$, the TT has 5395 degrees of freedom. We use 32370 uniform Monte-Carlo samples.

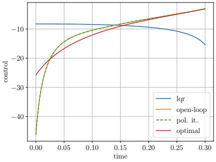
Numerical example - HJB in $\mathbb{R}^{32}$



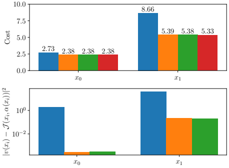
(a) Initial values x_0 and x_1.



(b) Generated controls, initial value x_0.



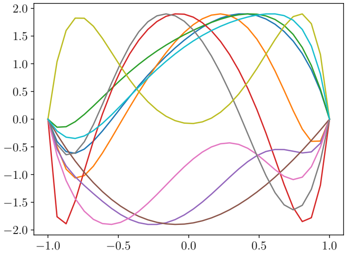
(c) Generated controls, initial value x_1.



(d) Generated cost and accuracy of the value function.

Figure: Blue ist the LQR controller, orange is the open-loop ansatz, green is the policy iteration ansatz, red is the optimal control.

Numerical example - HJB in $\mathbb{R}^{32}$



(a) Examples of samples drawn from the polynomial distribution.

controller	% cost < 100	avg. cost	avg. Bellman error
LQR	93.5	4.453	39.67
open-loop	100	2.61	0.048
pol. it.	100	2.61	0.047
optimal	100	2.60	

Numerical example - HJB in $\mathbb{R}^{32}$

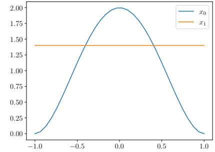
1D Allen Cahn

$$\begin{aligned} \dot{y} &= \sigma \Delta y + y - y^3 + \chi_\omega u \\ y(0) &= x \end{aligned}$$

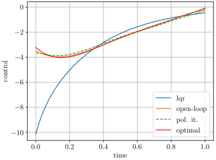
controller	% cost < 100	avg. cost	avg. Bellman error
LQR	100	2.434	24172
open-loop	100	1.985	0.03267
pol. it.	100	1.988	0.030593
optimal	100	1.981	

(b) Performance of the different controllers for 1000 samples drawn from the polynomial distribution. The averaged values are only taken from the subset of initial values that the LQR succeeded in stabilizing.

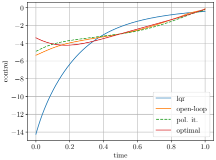
Numerical example - HJB in $\mathbb{R}^{32}$



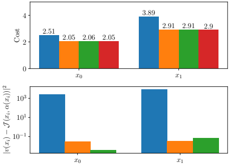
(c) Initial values x_0 and x_1.



(d) Generated controls, initial value x_0.



(e) Generated controls, initial value x_1.



(f) Generated cost and accuracy of the value function.

Numerical example - Allen-Cahn Equation

In courtesy of *N. Nüsken, L. Richter, L. Sallandt*: https://arxiv.org/abs/2102.11830

$$X_0 = x_0 = x_0^i \ , \ \mathcal{M} := \mathrm{span}\{v_T\} \oplus \mathcal{M}_n \text{ Trick !}$$

$$(\partial_t + \Delta) v(t,x) + v(t,x) - v^3(t,x) = 0 \ , \ x \in \mathbb{R}^d \ , \ d = 100$$
$$v(T,x) = v_T(x),$$

where

$$v_T(x) := \left(2 + \frac{2}{5}|x|^2\right)^{-1} \ , \ T = \frac{3}{10} \ , \ \Delta t = 0.01 \ , \ x_0 = (0,\dots,0)^\top$$

Corresponds to $f(t,x) = 0$, $h = \tilde{h} = -v + v^3$, $\Sigma = \sqrt{2} I_{d \times d}$

See *Weinan E et al. (2017)* for a reference solution $v(0,x_0) = 0.052802$

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{TT}_{\mathrm{expl}}$	$\mathrm{NN}_{\mathrm{impl}}$	$\mathrm{NN}^*_{\mathrm{impl}}$
$\widehat{v}_0(x_0)$	0.052800	0.05256	0.04678	0.05176
relative error	$4.75e^{-5}$	$4.65e^{-3}$	$1.14e^{-1}$	$1.97e^{-2}$
PDE loss	$2.40e^{-4}$	$2.57e^{-4}$	$9.08e^{-1}$	$6.92e^{-1}$
comp. time	24	10	23010	95278

with $N = 1000$ samples and $\mathrm{NN}^*_{\mathrm{impl}}$ uses $N^* = 8000$.

they used and implicit scheme of Hure , Pham & Warin

Numerical example - Hamilton Jacobi Bellman Equation

In courtesy of *N. Nüsken, L. Richter, L. Sallandt*: https://arxiv.org/abs/2102.11830

$$(\partial_t + \Delta)\, v(t,x) - |\nabla v(t,x)|^2 = 0,$$
$$v(T,x) = v_T(x),$$

with

$$v_T(x) = \log\left(\frac{1}{2} + \frac{1}{2}|x|^2\right)$$

corresponds to

$$f = \mathbf{0}, \quad \sigma = \sqrt{2}\, I_{d\times d}, \quad h(t,x,y,z) = -\frac{1}{2}|z|^2$$

$$\Rightarrow dX_s = \sqrt{2} dW_s \ , \ X_t = x \ , \ t \le s < T$$

Numerical example - Hamilton Jacobi Bellman Equation

In courtesy of *N. Nüsken, L. Richter, L. Sallandt*: https://arxiv.org/abs/2102.11830

A reference solution is given via Cole Hopf transformation

$$v(t,x) = -\log \mathbb{E}\left[e^{-v_T(x+\sqrt{T-t}\sigma\xi)}\right],$$

where $\xi \sim \mathcal{N}(\mathbf{0},\ Id_{d\times d})$, $x_0 = (0,\dots,0)^T$

$$\Rightarrow \quad v_{ref}(0,x_0) = 4.589992$$

cf. *Weinan E et al.(2017)*

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{TT}_{\mathrm{expl}}$	$\mathrm{NN}_{\mathrm{impl}}$	$\mathrm{NN}_{\mathrm{expl}}$
$\widehat{v}_0(x_0)$	4.5903	4.5909	4.5822	4.4961
relative error	$5.90\mathrm{e}^{-5}$	$3.17\mathrm{e}^{-4}$	$1.71\mathrm{e}^{-3}$	$2.05\mathrm{e}^{-2}$
reference loss	$3.55\mathrm{e}^{-4}$	$5.74\mathrm{e}^{-4}$	$4.23\mathrm{e}^{-3}$	$1.91\mathrm{e}^{-2}$
PDE loss	$1.99\mathrm{e}^{-3}$	$3.61\mathrm{e}^{-3}$	90.89	91.12
comp. time	41	25	44712	25178

Table: HJB equation in $d = 100$, $N = 2000$

Conclusions - Non-linear Feynman-Kac Method

- Key: Non-linear Feynman-Kac theorem -
- uses the semi-group (Koopman operator) instead of PDE
- Backward stepping- Forward transport of states
- Recovery by Least Squares Variational) Monte Carlo - regression - in a model set M
- explicit or implicit

Already known as or predecessors

- Dynamical programming approach
- generalized SEMI-Lagrangian
- Variational Monte Carlo (Bender et al.)
- Longstaff Schwartz Algorithm

Conclusions - Non-linear Feynman-Kac Method

Model classes

- polynomials - curse of dimensions
- sparse polynomials - sparse grid - not tried yet
- deep neural networks - gained recent interest breakthrough
  W. E , A. Jentzen at al. , Warin , Pham
- tensor methods *Horowitz , Damle , -. (2014) , Dolgov, Kalise, Kunisch (HJB) (2019) Oster, Sallandt, S. et al.* (2020-)
- HT/TT seems to much more accurate ($\times \geq 10$) and faster ($\times \geq 10 - 1000$) than DNN for all problems we found in the literature
- this seems to be due to controlled regression and more efficient optimization strategy

Work in progress

- determinstic control - finite horizon - infinite horizon (available)
- HJB with boundary conditions - shortest exit time problems (available)
- control constraints (available) - state constraints

Future developments

- theory
- detreministic and stochastic control of large dynamical system
- controll of (2D) Navier stokes equations, by solving an HJB in $\mathbb{R}^d$, $d \sim 2000$
- polynomial chaos representation for FBSDEs

References

HT/TT for non-linear PDE

- *N. Nüsken, L. Richter, L. Sallandt: Solving high-dimensional parabolic PDEs using the tensor train format, (2021) https://arxiv.org/abs/2102.11830*
- C. Bayer, M. Eigel, L. Sallandt and P. Trunschke, Pricing high-dimensional Bermudan options with hierarchical tensor formats (2021), arXiv:2103.01934
- K Fackeldey, M. Oster, L. Sallandt and R. S., Approximative Policy Iteration for Exit Time Feedback Control Problems driven by Stochastic Differential Equations using Tensor Train format (2020) arXiv:2010.04465
- M. Oster, L. Sallandt and R. S. Approximating the Stationary Hamilton-Jacobi-Bellman Equation by Hierarchical Tensor Products (2019) arXiv preprint arXiv:1911.00279
- S. Dolgov, D. Kalise and K. Kunisch A Tensor Decomposition Approach for High-Dimensional Hamilton-Jacobi-Bellman Equations (2019), arXiv:1908.01533

DNN for FBSDE

- J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. In: Proc. Natl. Acad. Sci. USA 115 (2017).
- C. Hure, H. Pham and X. Warin Deep backward schemes for high-dimensional nonlinear PDEs (2019) arXiv:1902.01599
- .....

FBSDE

- H. Pham. Continuous-time Stochastic Control and Optimization with Financial Applications. Vol. 61. SMAP. Springer, 2009.
- Bender

Numerical example - another Hamilton Jacobi Bellman Equation

In courtesy of *N. Nüsken, L. Richter, L. Sallandt*: https://arxiv.org/abs/2102.11830

$$\partial_t v(t,x) - \nabla\Psi)(t,x) \cdot \nabla v(t,x) - \frac{1}{2}|(\Sigma^\top \nabla) v(t,x)|^2 = 0,$$
$$v(T,x) = v_T(x),$$

where $f(t,x) = -\nabla\Psi(x)$ and Ψ is a double-well potential

$$\Psi(x) = \sum_{i,j=1}^{d} C_{ij}(x_i^2 - 1)(x_j^2 - 1)$$

The terminal condition is $v_T(x) = \sum_{i=1}^{d} \nu_i (x_i - 1)^2$, $\nu_i > 0$.
Reference solution by Cole Hopf Transform

$$v(t,x) = -\log \mathbb{E}\left[e^{-v_T(X_T)} \middle| X_t = x \right],$$

where $dX_s = -\nabla\Psi(X_s)ds + \sqrt{2}dW_s$, $X_t = x$, $t \leq s < T$

Numerical example - Hamilton Jacobi Bellman Equation

In courtesy of *N. Nüsken, L. Richter, L. Sallandt*: https://arxiv.org/abs/2102.11830

$C = 0.1\, Id_{d\times d}, \Sigma = \sqrt{2}\, I_{d\times d},\ T = 0.5, d = 50,\ \Delta t = 0.01,$
$N = 2000,\ \nu_i = 0.05,\ x_0 = (-1, \dots, -1)^\top$

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{NN}_{\mathrm{impl}}$
$\widehat{V}_0(x_0)$	9.3949	9.6942
relative error	$3.15e^{-2}$	$7.27e^{-4}$
reference loss	$2.15e^{-2}$	$4.25e^{-3}$
PDE loss	$2.43e^{-2}$	$2.66e^{-1}$
computation time	96	1987

$C = Id_{d\times d} + (\xi_{ij}),\ \xi_{ij} \sim \mathcal{N}(0, 0.01),\ \nu_i = 0.5,\ T = 0.3.$

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{NN}_{\mathrm{impl}}$
$\widehat{V}_0(x_0)$	34.278	34.228
relative error	$2.95e^{-4}$	$1.20e^{-3}$
reference loss	$3.26e^{-2}$	$4.00e^{-2}$
PDE loss	6.60	14.86
computation time	21	1693

Numerical example - Difficult Equation

In courtesy of *N. Nüsken, L. Richter, L. Sallandt*: https://arxiv.org/abs/2102.11830

$$f(t,x) = 0,\ \sigma(x,t) = \frac{I_{d\times d}}{\sqrt{d}},\ v_T(x) = \cos\left(\sum_{i=1}^{d} i x_i\right),$$

$$h(t,x,y,z) = k(x) + \frac{y}{2\sqrt{d}}\sum_{i=1}^{d} z_i + \frac{y^2}{2},$$

where, with an appropriately chosen k, a solution can shown to be

$$v(t,x) = \frac{T-t}{d}\sum_{i=1}^{d}\left(\sin(x_i)1_{x_i<0} + x_i 1_{x_i\geq 0}\right) + \cos\left(\sum_{i=1}^{d} i x_i\right).$$

Numerical example - Difficult Equation

In courtesy of *N. Nüsken, L. Richter, L. Sallandt*: https://arxiv.org/abs/2102.11830

$d = 10, N = 1000, T = 1, \Delta t = 0.001, x_0 = (0.5, \ldots, 0.5)^\top.$

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{TT}^*_{\mathrm{impl}}$	$\mathrm{NN}_{\mathrm{impl}}$
$\widehat{V}_0(x_0)$	-0.1887	-0.2136	-0.2137
relative error	$1.22\mathrm{e}^{-1}$	$6.11\mathrm{e}^{-3}$	$5.50\mathrm{e}^{-3}$
ref loss	$2.47\mathrm{e}^{-1}$	$7.57\mathrm{e}^{-2}$	$3.05\mathrm{e}^{-1}$
abs. ref loss	$2.52\mathrm{e}^{-2}$	$9.29\mathrm{e}^{-3}$	$1.69\mathrm{e}^{-2}$
PDE loss	2.42	0.60	1.38
computation time	360	1778	4520

Table: For $\mathrm{TT}^*_{\mathrm{impl}}$ they used $N^* = 20000$, and $N = 1000$ samples elsewhere

TT ranks $= 1$ seems to be sufifcient. It is mproved if one increases the sample size by a factor 20, but computational time remains lower than the NN implementation.

Numerical example - Cox-Ingersoll-Ross model

In courtesy of *N. Nüsken, L. Richter, L. Sallandt*: https://arxiv.org/abs/2102.11830

$$\partial_t v(t,x) + \frac{1}{2}\sum_{i,j=1}^{d} \sqrt{x_i x_j}\gamma_i\gamma_j \partial_{x_i}\partial_{x_j} v(t,x) + \sum_{i=1}^{d} a_i(b_i - x_i)\partial_{x_i} v(t,x) - \left(\max_{1\le i\le d} x_i\right) v(t,x) = 0.$$

Here, $a_i, b_i, \gamma_i \in [0,1]$ are uniformly sampled $v(T,x) = 1$.
$x_0 = (1,\dots,1)^\top$.

	$\mathrm{TT}_{\mathrm{impl}}$	$\mathrm{TT}_{\mathrm{expl}}$	$\mathrm{NN}_{\mathrm{impl}}$
$\widehat{v}_0(x_0)$	0.312	0.306	0.31087
PDE loss	$5.06\mathrm{e}^{-4}$	$5.04\mathrm{e}^{-4}$	$7.57\mathrm{e}^{-3}$
computation time	5281	197	9573

Table: $d = 100$, $N = 1000$