Overcoming the curse of dimensionality in the approximation of semilinear Black-Scholes PDEs

Philippe von Wurstemberger
(The Chinese University of Hong Kong, Shenzhen & ETH Zurich, Switzerland)

Based on joint works with

Sebastian Becker (ETH Zurich, Switzerland),
Ramon Braunwarth (D ONE, Switzerland),
Tuan Anh Nguyen (University of Duisburg-Essen, Germany)
Martin Hutzenthaler (University of Duisburg-Essen, Germany), and
Arnulf Jentzen (The Chinese University of Hong Kong, Shenzhen & University of Münster, Germany).

BFS World Congress 2022, Hong Kong

Tuesday, June 14th 2022

Goal: Approximate solutions of semilinear Black-Scholes PDEs, e.g.,:

$$(\partial_t u)(t,x) + \left[\sum_{i=1}^{d} \frac{|\beta_i|^2 |x_i|^2}{2} (\partial_{x_i x_i} u)(t,x) + \alpha_i x_i (\partial_{x_i} u)(t,x)\right] + f(u(t,x)) = 0,$$
$$u(T,x) = g(x),$$

where $u\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$, $(\alpha_i)_{i\in\{1,\ldots,d\}}, (\beta_i)_{i\in\{1,\ldots,d\}} \subseteq \mathbb{R}$, $g\colon \mathbb{R}^d \to \mathbb{R}$, $f\colon \mathbb{R} \to \mathbb{R}$, $T > 0$, and $(t,x) \in (0,T) \times \mathbb{R}^d$. Dimension $d \in \mathbb{N}$ corresponds to # assets in model.

Example application: Pricing with default risk (e.g.: $f(u(t,x)) = a \max\{u(t,x), 0\}$, $a \in \mathbb{R}$)

Major issue for $d \gg 1$:

Classical approximations methods such as finite differences, finite element methods, sparse grids suffer under the **curse of dimensionality** (cod)

Linear case ($f = 0$):

- Combining Feynman-Kac formula with Monte Carlo algorithm overcomes cod

Nonlinear case:

- Nonlinear Feynman-Kac formula $\longrightarrow$ BSDE representation of solution $\longrightarrow$ cod
- Branching diffusion methods: Overcome cod if small T and/or small g and/or small f
- Deep learning methods: *Seem* to overcome cod. No proof of convergence.
- *Only one algorithm* has been proven to overcome cod:
  $\Rightarrow$ **Multilevel Picard algorithm**

Overview

Derivation of the Multilevel Picard (MLP) algorithm

Convergence result for the MLP algorithm

Generalizations of the MLP algorithm

Numerical results

Synergies between deep learning and MLP

Derivation of the Multilevel Picard (MLP) algorithm

Recall nonlin. Black-Scholes: Let $d \in \mathbb{N}$, $T > 0$, $(\alpha_i)_{i\in\{1,\dots,d\}}, (\beta_i)_{i\in\{1,\dots,d\}} \subseteq \mathbb{R}$, $g\colon \mathbb{R}^d \to \mathbb{R}$, $f\colon \mathbb{R} \to \mathbb{R}$, $u \in C^{1,2}([0,T]\times\mathbb{R}^d, \mathbb{R})$ satisfy $\forall\,(t,x) \in [0,T)\times\mathbb{R}^d$:

$$(\partial_t u)(t,x) + \left[\sum_{i=1}^{d} \tfrac{|\beta_i|^2|x_i|^2}{2}(\partial_{x_i x_i} u)(t,x) + \alpha_i x_i (\partial_{x_i} u)(t,x)\right] + f(u(t,x)) = 0,$$
$$u(T,x) = g(x).$$

Geometric brownian motion: Let $(\Omega, \mathcal{F}, \mathbb{P})$ prob. space, let $W\colon [0,T]\times\Omega \to \mathbb{R}^d$ BM, $\forall\, x \in \mathbb{R}^d$ let $X^x\colon [0,T]\times\Omega \to \mathbb{R}^d$ satisfy $\forall\, t \in [0,T]$, $i \in \{1,2,\dots,d\}$:

$$X_t^{x,(i)} = x_i \exp\Big(\big(\alpha_i - \tfrac{|\beta_i|^2}{2}\big)t + \beta_i W_t^{(i)}\Big).$$

Feynman-Kac formula $\Rightarrow$ **Stochastic fixed point equation**:

$$\begin{aligned}\forall\,(t,x) \in [0,T]\times\mathbb{R}^d\colon \quad u(t,x) &= \mathbb{E}\Big[g\big(X_{T-t}^x\big) + \textstyle\int_0^{T-t} f(u(t+s, X_s^x))\,\mathrm{d}s\Big] \\ &= \mathbb{E}\Big[g\big(X_{T-t}^x\big) + (T-t) f\big(u(t+R_t, X_{R_t}^x)\big)\Big],\end{aligned}$$

where $R_t\colon \Omega \to [0, T-t]$ is $\mathcal{U}_{[0,T-t]}$-distr. and independent of W.

Picard iterations:

Consider $v_n\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$, $n\in\mathbb{N}_0$, such that $v_0\equiv 0$ and $\forall\, n\in\mathbb{N},(t,x)\in[0,T]\times\mathbb{R}^d$:

$$\begin{aligned} v_n(t,x) &= \mathbb{E}\Big[g\big(X^x_{T-t}\big)+(T-t)f\big(v_{n-1}\big(t+R_t,X^x_{R_t}\big)\big)\Big]\\ &= \mathbb{E}\Big[g\big(X^x_{T-t}\big)+(T-t)f\big(v_0\big(t+R_t,X^x_{R_t}\big)\big)\Big]\\ &\quad+\textstyle\sum_{k=1}^{n-1}(T-t)\mathbb{E}\Big[f\big(v_k\big(t+R_t,X^x_{R_t}\big)\big)-f\big(v_{k-1}\big(t+R_t,X^x_{R_t}\big)\big)\Big]. \end{aligned}$$

Multilevel Monte Carlo approximation:

Consider $V_{M,n}\colon [0,T]\times\mathbb{R}^d\times\Omega\to\mathbb{R}$, $M\in\mathbb{N}$, $n\in\mathbb{N}_0$, such that $\forall\, M\in\mathbb{N}\colon\ V_{M,0}\equiv 0$ and $\forall\, n,M\in\mathbb{N},(t,x)\in[0,T]\times\mathbb{R}^d$:

$$\begin{aligned} &V_{M,n}(t,x)\\ &=\frac{1}{M^n}\sum_{m=1}^{M^n} g\big(X^{x,(n,0,m)}_{T-t}\big)+(T-t)f\Big(V_{M,0}\big(t+R_t^{(n,0,m)},X^{x,(n,0,m)}_{R_t^{(n,0,m)}}\big)\Big)\\ &\quad+\sum_{k=1}^{n-1}\frac{(T-t)}{M^{n-k}}\sum_{m=1}^{M^{n-k}} f\Big(V_{M,k}\big(t+R_t^{(n,k,m)},X^{x,(n,k,m)}_{R_t^{(n,k,m)}}\big)\Big)-f\Big(V_{M,k-1}\big(t+R_t^{(n,k,m)},X^{x,(n,k,m)}_{R_t^{(n,k,m)}}\big)\Big), \end{aligned}$$

where $\big(R^{(n,k,m)},X^{x,(n,k,m)}\big)_{(n,k,m)\in\mathbb{N}_0}$ are independent realizations of $\big(R,X^x\big)$.

Convergence result for the MLP algorithm

Theorem (Hutzenthaler, Jentzen, PvW 19)

Let $T, C > 0$, *let* $(\alpha_{d,i})_{d,i\in\mathbb{N}}, (\beta_{d,i})_{d,i\in\mathbb{N}} \subseteq (-C, C)$, $\forall\, d \in \mathbb{N}$ *let* $g_d \in C^2(\mathbb{R}^d, \mathbb{R})$ *satisfy* $\forall\, x \in \mathbb{R}^d\colon |g_d(x)| \leq Cd^C(1 + \|x\|^C)$, *let* $f\colon \mathbb{R} \to \mathbb{R}$ *be Lipschitz,* $\forall\, d \in \mathbb{N}$ *let* $\xi_d \in \mathbb{R}^d$ *satisfy* $\|\xi_d\| \leq Cd^C$, $\forall\, d \in \mathbb{N}$ *let* $u_d \in C^{1,2}([0, T] \times \mathbb{R}^d, \mathbb{R})$ *be at most pol. grow. and satisfy* $\forall\, (t, x) \in [0, T) \times \mathbb{R}^d\colon u_d(T, x) = g_d(x)$ *and*

$$(\partial_t u_d)(t,x) + \left[\sum_{i=1}^{d} \frac{|\beta_{d,i}|^2 |x_i|^2}{2} (\partial_{x_i x_i} u_d)(t,x) + \alpha_{d,i} x_i (\partial_{x_i} u_d)(t,x)\right] + f(u_d(t,x)) = 0, \quad (1)$$

let $(\Omega, \mathcal{F}, \mathbb{P})$ *be a prob. space,* $\forall\, d, M, n \in \mathbb{N}$ *let* $V^d_{M,n}\colon [0, T] \times \mathbb{R}^d \times \Omega \to \mathbb{R}$ *be the MLP algorithm for* (1)*, and* $\forall\, d, M, n \in \mathbb{N}$ *let* $\mathrm{Cost}_{d,M,n} \in \mathbb{N}_0$ *be the cost to compute* $V^d_{M,n}(0, \xi_d)$. *Then* $\exists\, c > 0\colon\ \forall\, d \in \mathbb{N}, \varepsilon \in (0, 1]\colon\ \exists\, N_{d,\varepsilon} \in \mathbb{N}\colon$

$$\left(\mathbb{E}\left[|u_d(0,\xi_d) - V^d_{N_{d,\varepsilon},N_{d,\varepsilon}}(0,\xi_d)|^2\right]\right)^{1/2} \leq \varepsilon \qquad \text{and} \qquad \mathrm{Cost}_{d,N_{d,\varepsilon},N_{d,\varepsilon}} \leq cd^c\varepsilon^{-c}.$$

Remarks:

- First result proving that approximating (1) is polynomially tractable.
- Complexity of MLP is essentially the same as for standard Monte Carlo for linear PDEs.

Generalizations of the MLP algorithm

Semilinear Kolmogorov PDEs:

$$\partial_t u(t,x) + \langle \mu(t,x), \nabla_x u(t,x) \rangle + \tfrac{1}{2} \operatorname{Tr}\big([\sigma\sigma^*](t,x)\operatorname{Hess}_x u(t,x)\big) = f(t,x,u(t,x)),$$

$$u(T,x) = g(x),$$

where $\mu\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$, $\sigma\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}^{d\times m}$, $f\colon [0,T]\times\mathbb{R}^d\times\mathbb{R}\to\mathbb{R}$.

- Analogous Algorithm: Replace geom. BM with $\mathrm{d}X_t = \mu(t,X_t)\,\mathrm{d}t + \sigma(t,X_t)\,\mathrm{d}W_t$
- E.g.: heat equation, Correlated Black-Scholes, Allen Cahn, Sine Gordon

Semilinear Kolmogorov PDEs with gradient-dependent nonlinearities:

$$\partial_t u(t,x) + \langle \mu(t,x), \nabla_x u(t,x) \rangle + \tfrac{1}{2} \operatorname{Tr}\big([\sigma\sigma^*](t,x)\operatorname{Hess}_x u(t,x)\big) = f(t,x,u(t,x),\nabla_x u(t,x)),$$

$$u(T,x) = g(x),$$

where $\mu\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$, $\sigma\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}^{d\times m}$, $f\colon [0,T]\times\mathbb{R}^d\times\mathbb{R}\times\mathbb{R}^d\to\mathbb{R}$.

- Non-trivial extension: Combine Feynman-Kac with Bismut-Elworthy-Lee formula
- E.g.: Pricing with different interest rates for borrowing and lending, HJB equation

Any other problem with stochastic fixed point equations:

- E.g., McKean-Vlasov SDEs, semilinear PIDEs

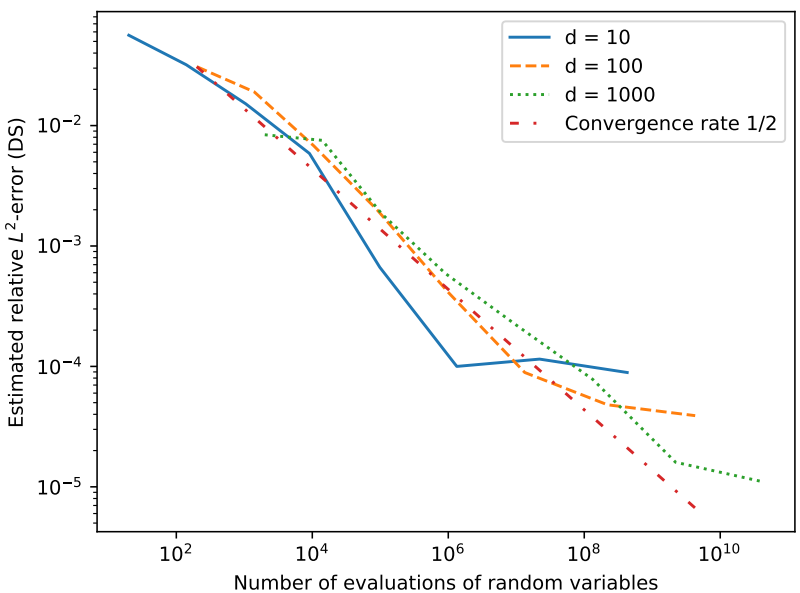
Numerical results

Semilinear Black-Scholes PDE

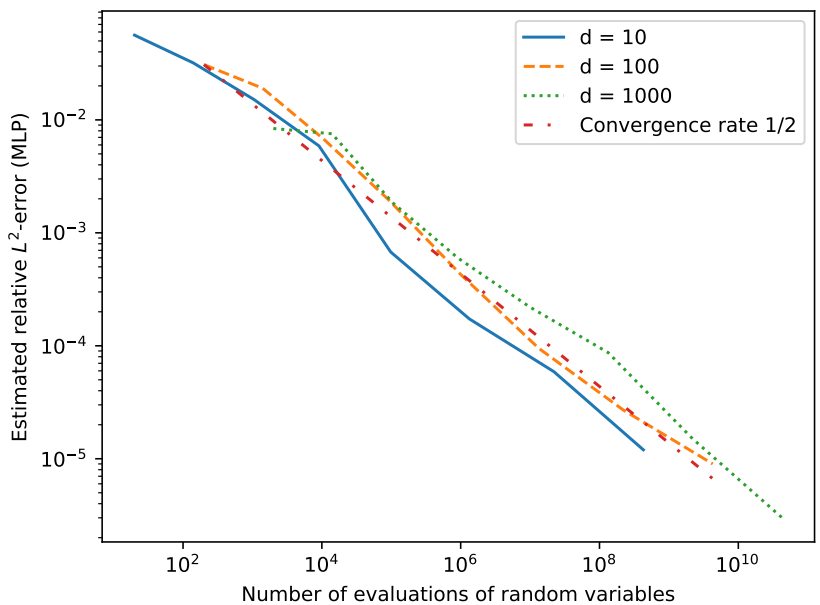
$$(\partial_t u)(t,x) + \tfrac{1}{2}\sum_{i=1}^{d}\left[|x_i|^2(\partial_{x_i x_i} u)(t,x)\right] + \sum_{i=1}^{d}\left[x_i(\partial_{x_i} u)(t,x)\right] + \frac{u(t,x)}{1+(u(t,x))^2} = 0,$$

$$u(T,x) = \log\left(\tfrac{1}{2}[1+\|x\|^2]\right), \qquad \text{for} \qquad (t,x) \in (0,1)\times\mathbb{R}^d.$$

Reference solutions computed by deep splitting



Reference solutions computed by MLP

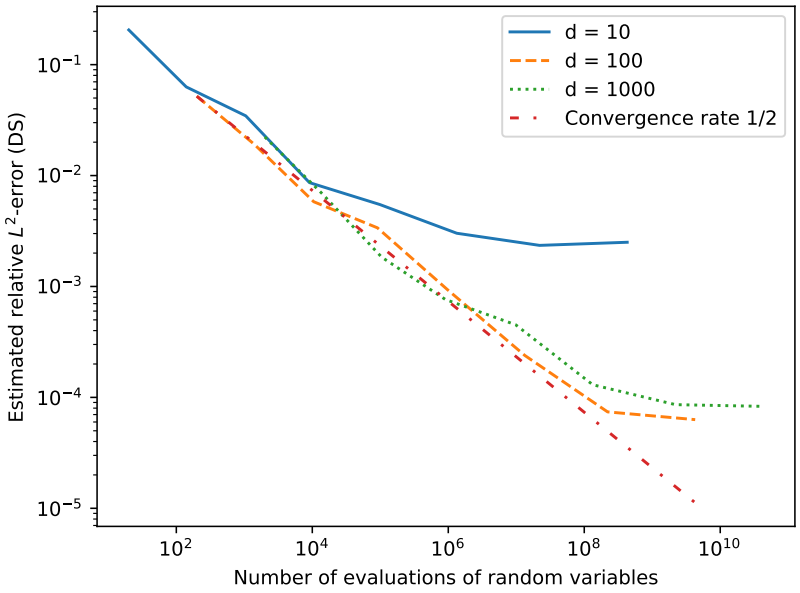


System of semilinear heat equations

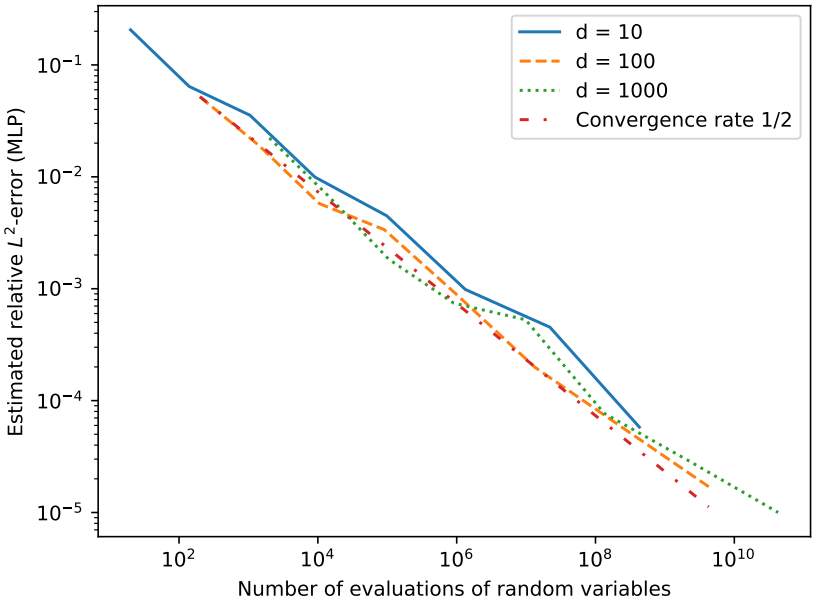
$$\begin{pmatrix}(\partial_t u_1)(t,x)\\(\partial_t u_2)(t,x)\end{pmatrix} + \begin{pmatrix}(\Delta_x u_1)(t,x)\\(\Delta_x u_2)(t,x)\end{pmatrix} + \begin{pmatrix}\frac{u_2(t,x)}{1+|u_2(t,x)|^2}\\ \frac{2u_1(t,x)}{3}\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix},$$

$$\begin{pmatrix}u_1(T,x)\\u_2(T,x)\end{pmatrix} = \begin{pmatrix}(2+\frac{2}{5}\|x\|^2)^{-1}\\ \log(\frac{1}{2}[1+\|x\|^2])\end{pmatrix}, \qquad \text{for} \qquad (t,x)\in(0,1)\times\mathbb{R}^d.$$

Reference solutions computed by deep splitting



Reference solutions computed by MLP



Synergies between deep learning and MLP

Deep learning for (nonlinear) PDEs:

- DeepBSDE method (E, Han, & Jentzen 17)
- many more: E, Han, & Jentzen, Beck 2017; Becker, Grohs, Jaafari, Jentzen 2018; Fujii, Takahashi, Takahashi 2017; E, Yu 2017; Sirignano, Spiliopoulos 2017; Berg, Nyström 2017; Henry-Labordere 2017; Raissi 2018; Chan-Wai-Nam, Mikael, Warin 2018; Farahmand, Nabi, Nikovski 2017; Goudenege, Molent, Zanette 2019; Huré, Pham, Warin 2019; . . .

$\longrightarrow$ Can deep learning overcome cod?

- No (complete) proof yet
- Partial proof: Artificial neural networks have the capacity to approximate semilinear PDEs without cod (Hutzenthaler, Jentzen, Kruse, Nguyen 19)

$\longrightarrow$ Proof based on MLP

Algorithms combining deep learning with MLP:

- Learning the random variables (Becker, Jentzen, Müller, PvW 22)

$\longrightarrow$ Start with precision of MLP

$\longrightarrow$ Improve with deep learning methods

Recap

- Approximation of semilinear parabolic PDEs without curse of dimensionality
- MLP algorithm: First and so far only algorithm which provably overcomes cod
  - Feynman-Kac formula $\longrightarrow$ Stochastic fixed point equation
  - Approximate Picard iterations with multilevel Monte Carlo $\longrightarrow$ **"multilevel Picard"**
- Outlook: Connections between deep learning and MLP
  - Theory: ANNs have capacity to overcome cod
  - Practice: **Learning the random variables**

Thank you!

Appendix

- Algorithmic description of MLP
- Pricing with default risks
- Other numerical results
- Literature

Appendix: Algorithmic description of MLP

Setting: $T > 0$, $d, m \in \mathbb{N}$, $\mu\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$, $\sigma\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}^{d\times m}$, $f\colon [0,T] \times \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$, $g\colon \mathbb{R}^d \to \mathbb{R}$.
Input: $n \in \mathbb{N}_0$, $M \in \mathbb{N}$, $t \in [0,T]$, $x \in \mathbb{R}^d$.
Output: MLP approximation of $u(t,x)$, where $u\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$ is a solution of

$$\partial_s u(s,y) + \langle \mu(s,y), \nabla_y u(s,y)\rangle + \tfrac{1}{2}\operatorname{Tr}\big([\sigma\sigma^*](s,y)\operatorname{Hess}_y u(s,y)\big) + f(s,y,u(s,y)) = 0, \quad u(T,y) = g(y).$$

function MLP(n, M, t, x) **:**
  if $n = 0$ **then**
    return 0
  Generate M^n realizations $\chi^{(m)}$, $m \in \{1,\dots,M^n\}$, of X_T, where $X_t = x$ and $\mathrm{d}X_s = \mu(s,X_s)\,\mathrm{d}s + \sigma(s,X_s)\,\mathrm{d}W_s$, $s \in [t,T]$
  $v = \sum_{m=1}^{M^n} \frac{g(\chi^{(m)})}{M^n}$
  for *all* $k \in \{0,1,\dots,n-1\}$ **do**
    Generate M^{n-k} realizations $\tau^{(m)}$, $m \in \{1,\dots,M^{n-k}\}$, of a $\mathcal{U}_{[t,T]}$ random variable
    For all $m \in \{1,\dots,M^{n-k}\}$ generate a realization $\chi^{(m)}$ of $X_{\tau^{(m)}}$, where $X_t = x$ and $\mathrm{d}X_s = \mu(s,X_s)\,\mathrm{d}s + \sigma(s,X_s)\,\mathrm{d}W_s$, $s \in [t,\tau^{(m)}]$
    if $k > 0$ **then**
      $v = v + \frac{T-t}{M^{n-k}} \sum_{m=1}^{M^{n-k}} \big(f(\tau^{(m)},\chi^{(m)},\mathrm{MLP}(k,M,\tau^{(m)},\chi^{(m)})) - f(\tau^{(m)},\chi^{(m)},\mathrm{MLP}(k-1,M,\tau^{(m)},\chi^{(m)}))\big)$
    else
      $v = v + \frac{T-t}{M^n} \sum_{m=1}^{M^n} f(\tau^{(m)},\chi^{(m)},\mathrm{MLP}(0,M,\tau^{(m)},\chi^{(m)}))$
  return v

Appendix: Example equations for pricing with default risks

We denote

$$(\mathcal{L}u)(t,x) = \sum_{i=1}^{d} \frac{|\beta_i|^2 |x_i|^2}{2} (\partial_{x_i x_i} u)(t,x) + \alpha_i x_i (\partial_{x_i} u)(t,x).$$

Burgard-Kjaer (2011), Henry-Labordere (2012):

$$(\partial_t u)(t,x) + (\mathcal{L}u)(t,x) - ru(t,x) + a \max\{u(t,x), 0\} + b \min\{u(t,x), 0\} = 0, \quad (2)$$

where $r, a, b \in \mathbb{R}$.
Duffie et al. (1996):

$$(\partial_t u)(t,x) + (\mathcal{L}u)(t,x) - ru(t,x) - (1-\delta) Q(u(t,x)) u(t,x) = 0, \quad (3)$$

where $r, v^h, \gamma^h \in \mathbb{R}$, $v^l \in (v^h, \infty)$, $\gamma^l \in (-\infty, \gamma^h)$ and $\forall\, y \in \mathbb{R}$:
$Q(y) = \mathbb{1}_{[-\infty, v^h)}(y)\gamma^h + \mathbb{1}_{[v^h, v^l)}(y) \left[\gamma^h + \left(\frac{\gamma^h - \gamma^l}{v^h - v^l}\right)(y - \gamma^h)\right] + \mathbb{1}_{[v^l, \infty)}(y)\gamma^l.$

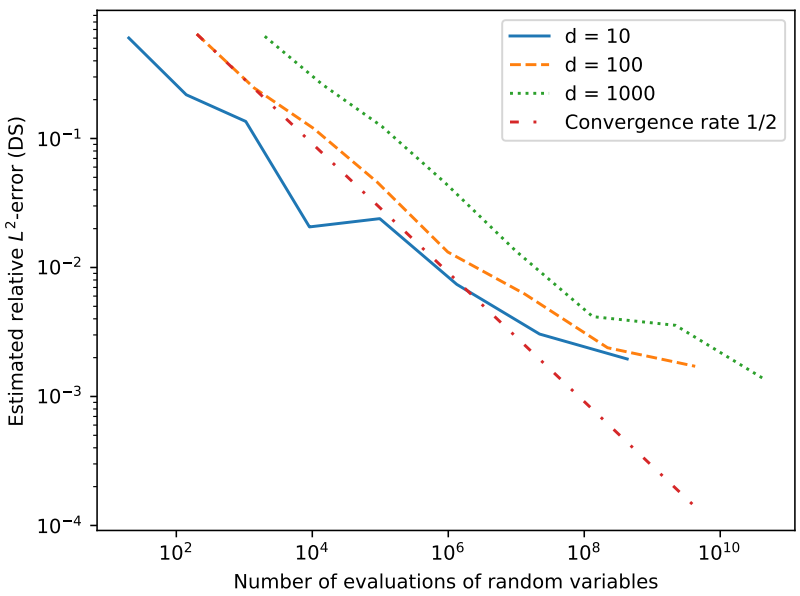
Appendix: Other numerical results

Allen-Cahn PDE:

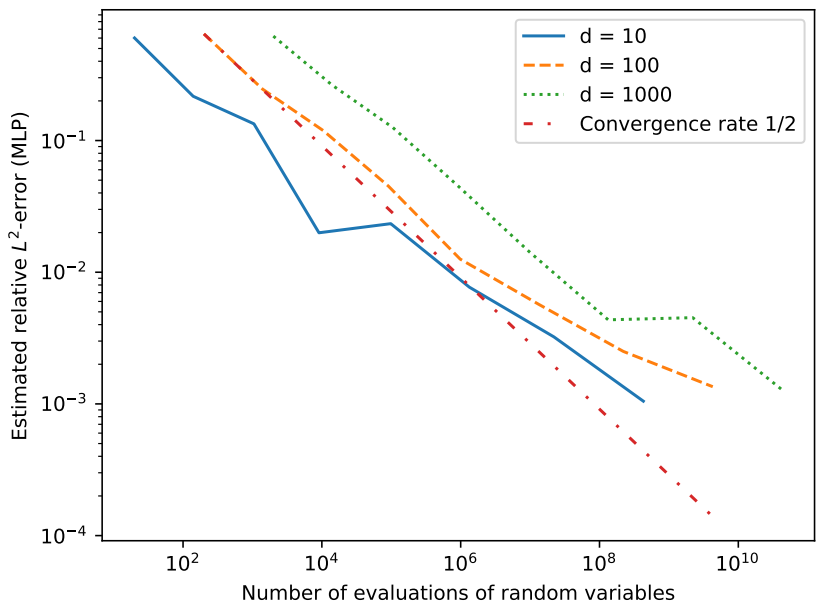
$$(\partial_t u)(t,x) + (\Delta_x u)(t,x) + u(t,x) - (u(t,x))^3 = 0,$$

$$u(1,x) = \left(2 + \tfrac{2}{5}\|x\|^2\right)^{-1}, \qquad \text{for} \qquad (t,x) \in (0,1) \times \mathbb{R}^d.$$

Reference solutions computed by deep splitting



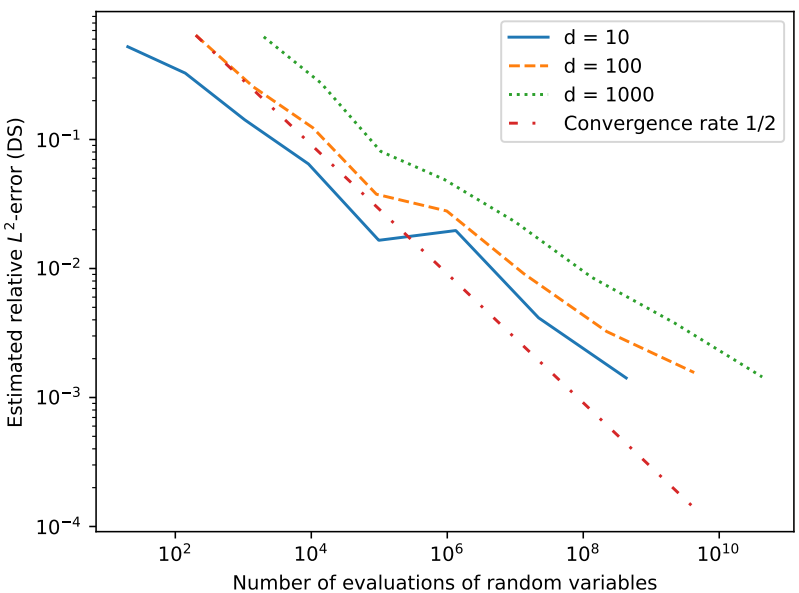
Reference solutions computed by MLP



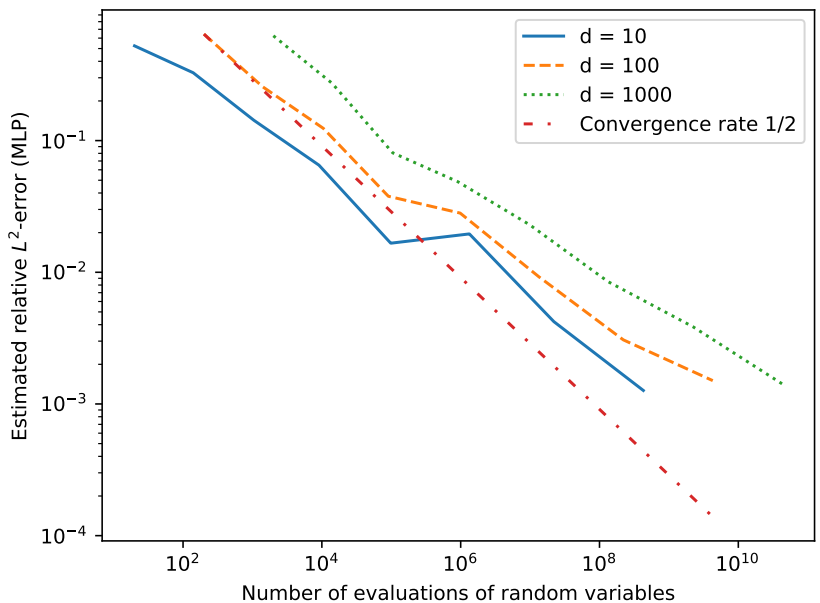
Sine-Gordon type PDE:

$$(\partial_t u)(t,x) + (\Delta_x u)(t,x) + \sin(u(t,x)) = 0,$$
$$u(1,x) = \left(2 + \tfrac{2}{5}\|x\|^2\right)^{-1}, \qquad \text{for} \qquad (t,x) \in (0,1) \times \mathbb{R}^d.$$

Reference solutions computed by deep splitting



Reference solutions computed by MLP



Appendix: Literature

Papers this talk is based on:

- Hutzenthaler, Jentzen, Kruse, Nguyen, & von Wurstemberger, *Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations. Proc. A., (2020)*
- Hutzenthaler, Jentzen, von Wurstemberger, *Overcoming the curse of dimensionality in the approximative pricing of financial derivatives with default risks. Electron. J. Probab. (2020)*
- Becker, Braunwarth, Hutzenthaler, Jentzen, & von Wurstemberger, *Numerical simulations for full history recursive multilevel Picard approximations for systems of high-dimensional partial differential equations. Commun. Comput. Phys. (2020)*

Original MLP papers:

- E, Hutzenthaler, Jentzen, & Kruse, *Multilevel Picard iterations for solving smooth semilinear parabolic heat equations. Partial Differ. Equ. Appl. (2021)*
- E, Hutzenthaler, Jentzen, & Kruse, *On multilevel Picard numerical approximations for high-dimensional nonlinear parabolic partial differential equations and high-dimensional nonlinear backward stochastic differential equations. J. Sci. Comput., (2019)*

MLP Algorithms for different problems:

- **Semilinear PDEs with gradient-independent nonlinearities (Euler discr. of SDE)**: Hutzenthaler, Jentzen, Kruse, & Nguyen, *Multilevel Picard approximations for high-dimensional semilinear second-order PDEs with Lipschitz nonlinearities. arXiv (2020)*
- **Allen-Cahn PDEs**: Beck, Hornung, Hutzenthaler, Jentzen, & Kruse, *Overcoming the curse of dimensionality in the numerical approximation of Allen–Cahn partial differential equations via truncated full-history recursive multilevel Picard approximations. arXiv (2019)*
- **Elliptic PDEs**: Beck, Gonon, & Jentzen, *Overcoming the curse of dimensionality in the numerical approximation of high-dimensional semilinear elliptic partial differential equations. arXiv (2020)*
- **Semilinear PDEs with gradient-dependent nonlinearities**: Hutzenthaler, Jentzen, & Kruse, *Overcoming the curse of dimensionality in the numerical approximation of parabolic partial differential equations with gradient-dependent nonlinearities. Found. Comput. Math. (2021)*
- **McKean-Vlasov SDEs**: Hutzenthaler, Kruse, Nguyen, Multilevel Picard approximations for McKean-Vlasov stochastic differential equations. arXiv (2021)
- **Semilinear PIDEs**: Neufeld, Wu, *Multilevel Picard approximation algorithm for semilinear partial integro-differential equations and its complexity analysis. arXiv (2022)*
- **Abstract formulation of MLP**: Giles, Jentzen, & Welti, *Generalised multilevel Picard approximations. arXiv (2019)*

Other approaches for semilinear PDEs (very rough!):

- **Nested conditional expectations for nonlinear Feynman-Kac formula**: Bouchard & Touzi, *Discrete Time Approximation and Monte-Carlo Simulation of Backward Stochastic Differential Equations. Stochastic Process. Appl. (2004)*
- **Branching diffusions for semilinear PDEs**: Henry-Labordere, Oudjane, Tan, Touzi, & Warin, *Branching diffusion representation of semilinear PDEs and Monte Carlo approximation. arXiv (2016)*
- **Deep learning (deepBSDE method)**: E, Han, Jentzen *Solving high-dimensional partial differential equations using deep learning. Proc. Natl. Acad. Sci. (2017)*

Pricing with default risks:

- Burgard & Kjaer, *Partial differential equation representations of derivatives with bilateral counterparty risk and funding costs. The Journal of Credit Risk (2011)*
- Crépey, Gerboud, Grbac, & Ngor, *Counterparty risk and funding: The four wings of the TVA. International Journal of Theoretical and Applied Finance (2013)*
- Duffie, Schroder, Skiadas, *Recursive valuation of defaultable securities and the timing of resolution of uncertainty. Ann. Appl. Probab. (1996)*
- Henry-Labordere, *Counterparty Risk Valuation: A Marked Branching Diffusion Approach. arXiv (2012)*

Pricing with different interest rates for borrowing and lending:

- Bergman, *Option pricing with differential interest rates. Rev. Financ. Stud. (1995)*