

This is an ACCEPTED VERSION of the following published document:

Gobet, E., López-Salas, J.G. & Vázquez, C. Quasi-Regression Monte-Carlo Scheme for Semi-Linear PDEs and BSDEs with Large Scale Parallelization on GPUs. Arch Computat Methods Eng 27, 889–921 (2020). <https://doi.org/10.1007/s11831-019-09335-x>

Link to published version: <https://doi.org/10.1007/s11831-019-09335-x>

General rights:

©2019 This version of the article has been accepted for publication, after peer review and is subject to [Springer Nature's AM terms of use](#), but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: <https://doi.org/10.1007/s11831-019-09335-x>

Quasi-Regression Monte-Carlo scheme for semi-linear PDEs and BSDEs with large scale parallelization on GPUs

E. Gobet · J. G. López-Salas · C. Vázquez

Received: date / Accepted: date

Abstract In this article we design a novel quasi-regression Monte Carlo algorithm in order to approximate the solution of discrete time backward stochastic differential equations (BSDEs), and we analyze the convergence of the proposed method. The algorithm also approximates the solution to the related semi-linear parabolic partial differential equation (PDE) obtained through the well known Feynman-Kac representation. For the sake of enriching the algorithm with high order convergence a weighted approximation of the solution is computed and appropriate conditions on the parameters of the method are inferred. With the challenge of tackling problems in high dimensions we propose suitable projections of the solution and efficient parallelizations of the algorithm taking advantage of powerful many core processors such as graphics processing units (GPUs).

Keywords BSDEs · semi-linear PDEs · dynamic programming equations · empirical regressions · parallel computing · GPUs · CUDA

Mathematics Subject Classification (2010) 49L20 · 62Jxx · 65C30 · 93E24 · 68W10

The first author research is part of the Finance for Energy Markets (FiME) lab, of the Chair Financial Risks of the Risk Foundation and of the ANR project CAESARS (ANR-15-CE05-0024). The second author has been financially supported by the Chair Financial Risks of the Risk Foundation, the Spanish grant MTM2016-76497-R and the *Xunta de Galicia* 2018 postdoctoral grant. The third author was partially supported by Spanish grant MTM2016-76497-R.

E. Gobet
Centre de Mathématiques Appliquées, École Polytechnique and CNRS, route de Saclay, 91128 Palaiseau cedex, France
E-mail: emmanuel.gobet@polytechnique.edu

J. G. López-Salas (✉ Corresponding author, E-mail: jose.lsalas@udc.es)
Centre de Mathématiques Appliquées, École Polytechnique and CNRS, route de Saclay, 91128 Palaiseau cedex, France

C. Vázquez
Department of Mathematics, Faculty of Informatics, Universidade da Coruña, Campus de Elviña s/n, 15071, A Coruña, Spain
E-mail: carlosv@udc.es

1 Introduction

Since the 1940s, Feynman-Kac formula has provided a probabilistic interpretation of the solutions of linear partial differential equations (PDEs). Thence, Monte Carlo method has become an alternative to deterministic numerical algorithms for solving linear PDEs. While traditional deterministic methods like finite differences, finite elements or finite volumes suffer from the so-called *curse of dimensionality* [4] (i.e. as the dimension grows, the complexity of the algorithms needed to achieve a given accuracy grows exponentially), the rate of convergence of Monte Carlo methods does not depend on the dimension of the state space. As a consequence, in order to solve equations of dimension greater than 4 or 5, the probabilistic approach often remains the only numerical method available.

In the 1990s, the theory of backward stochastic differential equations (BSDEs) provided a probabilistic interpretation for non-linear problems. BSDEs are stochastic differential equations (SDEs) for which a terminal condition is specified and their solutions are pairs of processes. Linear BSDEs were introduced by Bismut [8] in 1973 in order to study stochastic control problems and their connections with the Pontryagin maximum principle of optimality. A rigorous study of (non-linear) BSDEs was started by Pardoux and Peng in 1990, who proved in their seminal article [53] the first well-posedness result for non-linear BSDEs, see also [54]. These equations have the general form of

$$\begin{aligned} -dY_t &= f(t, Y_t, Z_t)dt - Z_t dW_t, \\ Y_T &= \xi, \end{aligned}$$

where $0 \leq t \leq T$, $(W_t)_{t \geq 0}$ is a Brownian motion, the terminal condition ξ is a random variable, and the solution is a pair of adapted processes (Y, Z) . The first applications of BSDEs in finance were conducted by El Karoui, Peng and Quenez [27] in 1997. They applied BSDEs in a European option pricing problem, where the randomness of (Y, Z) comes from a forward SDE (FSDE in short) in X as follows,

$$\begin{aligned} dX_t &= b(t, X_t)dt + \sigma(t, X_t)dW_t, \quad X_0 = x_0, \\ -dY_t &= f(t, X_t, Y_t, Z_t)dt - Z_t dW_t, \quad Y_T = g(X_T); \end{aligned}$$

the function f models possible imperfections and restrictions when trading in the markets. Now the terminal condition of the BSDE is determined by the stochastic terminal value of the FSDE through some deterministic function g . This system of SDEs, to which we will devote this article, is known as decoupled¹ forward-backward stochastic differential equation (FBSDE in short). It is also referred as Markovian BSDE, since the terminal condition of the BSDE and f are random only through a Markov process.

After the already mentioned celebrated work of El Karoui, Peng and Quenez, the interest in BSDEs has increased, mainly due to the connection of these tools

¹ Fully coupled FBSDEs have the general form

$$\begin{aligned} dX_t &= b(t, X_t, Y_t, Z_t)dt + \sigma(t, X_t, Y_t, Z_t)dW_t, \quad X_0 = x_0, \\ -dY_t &= f(t, X_t, Y_t, Z_t)dt - Z_t dW_t, \quad Y_T = g(X_T). \end{aligned}$$

with stochastic control and PDEs, connection that will be stated clearly soon. Let us start with a rigorous definition of the problem at hand.

The problem. In this work we are interested in numerically approximating the solution (X, Y, Z) of a decoupled forward-backward stochastic differential equation

$$Y_t = g(X_T) + \int_t^T f(s, X_s, Y_s, Z_s) ds - \int_t^T Z_s dW_s, \quad (1.1)$$

$$X_t = x + \int_0^t b(s, X_s) ds + \int_0^t \sigma(s, X_s) dW_s. \quad (1.2)$$

The terminal $T > 0$ is fixed. These equations are considered in a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{0 \leq t \leq T})$ supporting a $q \geq 1$ dimensional Brownian motion W . The filtration is assumed to be the natural filtration of W augmented with the \mathbb{P} -null sets. In this representation, X is a d -dimensional adapted continuous process (called the forward component), Y is a scalar adapted continuous process and Z is a q -dimensional progressively measurable process. Consideration of multi-dimensional Y is quite straightforward and is discarded for the sake of simplicity. Regarding terminology, $g(X_T)$ is called *terminal condition* and f the *driver*. Under standard assumptions (like L^2 -integrability and Lipschitz regularity), there is a unique solution (X, Y, Z) (see [55] for broad account on existence/uniqueness assumptions). Detailed hypotheses for the current study are exposed later.

Our aim in this article is to design a new numerical scheme for approximating the solution to the above BSDE. When f is linear in (Y, Z) , explicit representations are available [27, Theorem 1.1], which can be easily turned into simple and efficient Monte-Carlo algorithms. Our focus is thus more on the case of non-linear driver yielding non-linear equations.

Feynman-Kac representations. Let us clearly show that FBSDEs are important tools in semi-linear PDEs and stochastic control theory. Denote by \mathcal{A} the infinitesimal generator of X :

$$\mathcal{A} = \sum_i b_i(t, x) \partial_{x_i} + \frac{1}{2} \sum_{i,j} [\sigma \sigma^\top]_{i,j}(t, x) \partial_{x_i x_j}^2,$$

where $^\top$ denotes the transpose operator. Consider the parabolic partial differential equation (PDE) of the form

$$\begin{cases} \partial_t u(t, x) + \mathcal{A}u(t, x) + f(t, x, u(t, x), (\nabla_x u \sigma)(t, x)) = 0, & t < T, \\ u(T, \cdot) = g(\cdot). \end{cases} \quad (1.3)$$

Assuming that u is smooth enough, the Itô formula leads to

$$u(t, X_t) = u(0, x) + \int_0^t (\partial_t u(s, X_s) + \mathcal{A}u(s, X_s)) ds + \int_0^t (\nabla_x u \sigma)(s, X_s) dW_s.$$

Using the PDE (1.3) and noting $u(T, X_T) = g(X_T)$, we get

$$u(t, X_t) = g(X_T) + \int_t^T f(s, X_s, u(s, X_s), (\nabla_x u \sigma)(s, X_s)) ds - \int_t^T (\nabla_x u \sigma)(s, X_s) dW_s,$$

therefore proving that the solution (Y, Z) can be represented as

$$(Y_t, Z_t) = (u(t, X_t), (\nabla_x u \sigma)(t, X_t)).$$

This is the well-known Feynman-Kac representation, making connection between semi-linear PDEs and BSDEs. One may prefer one or the other representations depending on the application at hand, on the numerical scheme to use, and on the theoretical estimates available (of probabilistic or PDE nature). As a consequence, solving numerically (Y, Z) for any realization of X or computing $(u, \nabla u)$ in the full space \mathbb{R}^d gives, in both cases, essentially access to the same solution. The above reasoning holds for Cauchy boundary conditions and parabolic PDEs as presented above, however extensions in the directions of various boundary conditions (Dirichlet, Neumann, free-boundary) or stationary problems (elliptic PDEs) are also available, see [55, Sections 3.8, 5.4, 5.6, 5.7 and 5.8] and references therein.

In view of (1.3), the link with stochastic control appears clearly, by analogy with the Hamilton-Jacobi-Bellman equation. Since the non-linearity in f is related only to the first derivative of u (the Z component), the underlying control problem is associated to a control in the drift only (not in the diffusion coefficient). See [27, 25] for some examples in finance.

Applications. We now conclude the presentation of connection between BSDEs and PDEs by giving a broader overview of applications where these semi-linear PDEs come into play. We follow the presentation of [32, Section 7.1], some examples can be found in [38, Chapter 2] and [59, Chapter 14].

Ecology. In a region of the plane ($d = 2$), suppose the existence of 2 interacting species in the same environment and denote $u_k(t, x)$ ($k = 1, 2$) the population density of the individuals of each species at a point of the space x and at a time t . These population densities obey to the system

$$\begin{cases} \partial_t u_1(t, x) = \alpha_1 \Delta u_1(t, x) + f_1(t, x, u_1, u_2), \\ \partial_t u_2(t, x) = \alpha_2 \Delta u_2(t, x) + f_2(t, x, u_1, u_2), & t > 0, x \in \mathbb{R}^2, \\ u_i(0, \cdot) \text{ given,} \end{cases} \quad (1.4)$$

where the two Laplacian operators model the diffusive movement (as a Brownian motion) of each species. Note that the initial condition at $t = 0$ can be turned into a terminal condition at $t = T$ as in our previous setting, this is just a simple time reversal $t \leftrightarrow T - t$.

The function $f = (f_1, f_2)$ plays the role of a growth rate of the population, describing the available local resources and interactions between the species: the case $\partial_{u_2} f_1 < 0$ and $\partial_{u_1} f_2 > 0$ corresponds to the predator-prey model, where the growth rate of the species 1 (prey) decreases in the case of high density of the species 2 (predator) and conversely for the growth rate of predators; the case $\partial_{u_2} f_1 > 0$ and $\partial_{u_1} f_2 > 0$ models symbiosis, where each species benefits from the other; $\partial_{u_2} f_1 < 0$ and $\partial_{u_1} f_2 < 0$ describes competition between species. This kind of model, introduced in the 30s by Fisher, Kolmogorov, Petrovsky and Piskunov, is discussed with extra references in [58].

Chemistry. Suppose that a container contains N chemical compounds taking part in R independent reactions. Denote c_i the concentration of the i -th compound

and θ the temperature. Then their evolution follows the $N + 1$ equations (in $\mathbb{R}^+ \times \mathbb{R}^3$)

$$\begin{cases} \varepsilon_p \partial_t c_i = D_i \Delta c_i + \sum_{j=1}^R \nu_{ij} g_j(c_1, \dots, c_N, \theta), & i = 1, \dots, N, \\ \rho c_p \partial_t \theta = k \Delta \theta - \sum_{j=1}^R \sum_{i=1}^N \nu_{ij} H_i g_j(c_1, \dots, c_N, \theta), \\ c_i(0, \cdot) \text{ and } \theta(0, \cdot) \text{ given,} \end{cases} \quad (1.5)$$

where g_j is the speed of the j -th reaction and H_i is the partial molar enthalpy of the i -th compound. For more details, see [30]. For reaction-diffusion equations, see [59].

Neuroscience. The famous model of Hodgkin and Huxley is a set of equations describing the psychological phenomenon of signal transmission in the axon (nerve fiber), showing the dependences between electrical excitability and various chemical ion concentrations. The PDE system is of size 4, the first unknown u represents the electric potential and the three other unknowns (v_1, v_2, v_3) are chemical concentrations: the system writes for $(t, x) \in \mathbb{R}^+ \times \mathbb{R}$ (linear neuron)

$$\begin{cases} c_0 \partial_t u = \frac{1}{R} \partial_{xx}^2 u + \kappa_1 v_1^3 v_2 (c_1 - u) + \kappa_2 v_3^4 (c_2 - u) + \kappa_3 (c_3 - u), \\ \partial_t v_1 = \varepsilon_1 \partial_{xx}^2 v_1 + g_1(u) (h_1(u) - v_1), \\ \partial_t v_2 = \varepsilon_2 \partial_{xx}^2 v_2 + g_2(u) (h_2(u) - v_2), \\ \partial_t v_3 = \varepsilon_3 \partial_{xx}^2 v_3 + g_3(u) (h_3(u) - v_3), \\ u(0, \cdot) \text{ and } v_i(0, \cdot) \text{ given,} \end{cases} \quad (1.6)$$

for different positive constants $c_i, R, \kappa_i, \varepsilon_i$ and different functions g_i, h_i . There also exists a simplification of this model, known as the FitzHugh-Nagumo model.

Materials physics. The Allen-Cahn equation is a prototype model of phase transition with a diffusive interface, used to model, for example, a solid/liquid phase transition [62]. The system is one-dimensional, and takes the form (in $\mathbb{R}^+ \times \mathbb{R}^3$)

$$\begin{cases} \partial_t u = \varepsilon \Delta u + u(1 - u^2), \\ u(0, \cdot) \text{ given.} \end{cases} \quad (1.7)$$

The solution u thus represents an order parameter defining the arrangement of atoms in a crystal lattice.

Economy and finance. There are numerous applications of BSDEs (and semi-linear PDEs) in finance, we refer the reader to [18, 27, 16]. Let us mention: valuation/hedging of contingent claim in complete markets; some market imperfections, such as higher interest rate for borrowing [7]; incomplete markets [57] or super-replication [28]; connections with recursive utilities in [23]; non-linear pricing rule [56]; second order BSDEs (2BSDEs) [13, 60] which provide a stochastic representation for solutions of fully non-linear parabolic PDEs through a driver depending also on another process directly connected with the Hessian of the solution of the corresponding PDE; reflected BSDEs, where the solution Y is constrained to stay above a given process called obstacle, were introduced by El Karoui et al in [26], and combined with jumps are able to connect with partial-integro differential equations [3, 21], etc. See [25, 16] for extra references of applications in finance.

Numerical approximations. In the two past decades, there have been several attempts to design efficient approximation schemes for BSDEs. First, the four step algorithm was proposed by Ma, Proter and Yong [48] in 1994 to solve general FBSDEs. Their method is based on the finite difference approximation of the associated quasilinear parabolic PDE, and therefore becomes unaffordable in high dimension. In 1997, a discrete scheme based on the approximation of the Brownian motion by some discrete process was proposed by Chevance in [14], while Bally considered a random time scheme in [1], although this method requires the computation of conditional expectations to become a fully implementable solver. At that time, only time-discretization issue was considered. Since then, optimal discretization strategies have been derived: see [31] and references therein. All probabilistic approaches proposed later roughly follow three steps: time discretization, dynamic programming principle and computation of conditional expectations. The problem of discretization and simulation of the forward component X is well-understood, one could use standard Euler scheme or other choices like Milstein scheme, see [43]. The difficulty lies in the backward component Y , since a naive discretization of the backward SDE in Y is not guaranteed to be adapted. In order to obtain a computationally viable backward induction scheme one should take conditional expectations. If the time grid is $\{t_0 := 0, \dots, t_i, \dots, t_N =: T\}$, it leads to a nested expectation problem of the form (also called Dynamic Programming Equation, or DPE for short)

$$Z_{t_i} \approx \mathbb{E} \left[Y_{t_{i+1}} \frac{(W_{t_{i+1}} - W_{t_i})^\top}{t_{i+1} - t_i} \mid \mathcal{F}_{t_i} \right], \quad (1.8)$$

$$Y_T = g(X_T), \quad Y_{t_i} \approx \mathbb{E} \left[Y_{t_{i+1}} + f(t_i, X_{t_i}, Y_{t_{i+1}}, Z_{t_i})(t_{i+1} - t_i) \mid \mathcal{F}_{t_i} \right], \quad (1.9)$$

for $i = N - 1, \dots, 0$. We use the symbol \approx to emphasize the heuristics, a rigorous justification can be done using the aforementioned references. The scheme (1.8)-(1.9), which is known as one-step forward dynamic programming scheme, could be updated taking into account a resimulation of the forward process till terminal time T , thus resulting in the so-called multi-step forward dynamic programming scheme, which is written as follows

$$Z_{t_i} \approx \mathbb{E} \left[\left(g(X_T) + \sum_{j=i+1}^{N-1} f(t_j, X_{t_j}, Y_{t_{j+1}}, Z_{t_j})(t_{j+1} - t_j) \right) \frac{(W_{t_{i+1}} - W_{t_i})^\top}{t_{i+1} - t_i} \mid \mathcal{F}_{t_i} \right], \quad (1.10)$$

$$Y_{t_i} \approx \mathbb{E} \left[g(X_T) + \sum_{j=i}^{N-1} f(t_j, X_{t_j}, Y_{t_{j+1}}, Z_{t_j})(t_{j+1} - t_j) \mid \mathcal{F}_{t_i} \right], \quad (1.11)$$

for $i = N - 1, \dots, 0$.

Analytically, both schemes (1.8)-(1.9) and (1.10)-(1.11) are equivalent. However, differences arise when conditional expectations are numerically approximated. In fact, the multi-step scheme leads to better error estimates of the solutions computed by the developed algorithms, see [5, 36]. The above backward dynamic programming equations make clearly appear that at each of N grid times, one has to compute a conditional expectation function, that is a function from \mathbb{R}^d to \mathbb{R} , this is in a way equivalent to solving the PDE at each time point t_i . This is a

difficult problem, with increasing difficulty as the dimension gets large and/or as the unknown function gets less smooth (*curse of dimensionality*), like for finite differences schemes for PDEs. Among the numerous approaches, let us mention the quantization method of Bally and Pagès [2] (and generalized to coupled FB-SDEs by Delarue and Menozzi [19, 20]), which is an optimal space discretization of the underlying dynamic programming equation, suitable to compute easily the conditional expectations on the space grid; in [10] the authors compute the regression function with Malliavin calculus integration by parts combined with Monte Carlo simulations; the use of empirical regression methods and machine learning techniques is developed in [33, 47] (also called *regression Monte-Carlo method*). Optimal convergence rates with regression methods are derived in [36], variance reduction using importance sampling is designed and analysed in [6, 37].

Recently, Henry-Labordere et al. [39] and Bouchard et al. [9] proposed a branching method and a time step randomization applied to the Feynman-Kac representation of the PDE. Although branching techniques really overcome the curse of dimensionality, unfortunately they are only limited to small maturities, some small non-linearities and mainly to non-linearities that are polynomial in the solution and its gradient. Lately, in [24] the authors propose a deep learning technique, referred as deep BSDE, to solve semi-linear PDEs or the corresponding BSDEs. They interpret the BSDE as a stochastic control problem with the gradient of the solution being the policy function, which is approximated or learned by a deep neural network, as has been done in deep reinforcement learning. More precisely, after discretizing the forward SDE in X using Euler discretization, the algorithm reads the BSDE for Y as a forward SDE, and the method tries to learn the value of Y and Z at each time step of the Euler scheme by minimizing a global loss function between the forward simulation of Y until the maturity T and the goal $g(X_T)$. In the recent work [12], the authors give an improved version of the deep BSDE method, where a fixed point problem is solved using deep learning techniques. Numerical results show that these kind of algorithms are able to solve problems with some accuracy in dimension 50 and even above. However, these methods suffer so far from the lack of rigorous error estimates and the matter that convergence is not guaranteed: as in linear regression methods [36], there must be a tricky interplay between the number of time steps, the number of simulations, the number of neurons and layers, to ensure a good accuracy. Moreover, one has to account for the likely possibility that the stochastic gradient descend method used to minimize the loss function is trapped in a local minimum.

Parallel computations. Most of the above methods do not allow parallel computations because of the non-linearity of the equations. Recently in [34], the authors have proposed a novel approach inspired from regression Monte-Carlo schemes suitable for massive parallelization, with excellent results on GPUs. It takes advantage of local approximations of the regression function in different strata of the state space \mathbb{R}^d (partitioning estimates) and of a new stratified sampling scheme to guarantee that enough simulations are available in each stratum to compute accurately the regression functions. The parallelization step is made with respect to the strata.

Our current work lies in this vein and is aimed at overcoming some drawbacks of the stratified regression scheme of [34]. In the previous scheme, the solution u to the PDE is represented as a piecewise polynomial function (like in a discontinuous

Galerkin method [22]) and therefore, even if the exact solution is continuous, the approximation may be not, especially at the interface of the different strata. In the current work, we propose a different scheme, where the approximated solution is decomposed on basis functions that are smooth and that have natural orthonormal properties with respect to the sampling distribution, thus avoiding the possible discontinuity at the interfaces. Moreover, the sampling distribution is required to have some nice stability properties, useful for the propagation of error in the DPE, see Theorem 2; we prove that the Student's t-distribution meets this requirement, see Theorem 1. One possible set of smooth and orthonormal functions is given by the trigonometric basis composed with a space transformation related to the Cumulative Distribution Function (CDF for short) of the Student's t-distribution. We call such a basis the *Student-cosine basis*. Instead of a regression Monte Carlo scheme, we design a *quasi-regression Monte-Carlo scheme*: we take advantage of the fact that the basis functions enjoy orthonormal properties, in order to compute the basis coefficients as simple Monte-Carlo averages, instead of solutions of least-squares problems. On the one hand, it leads to a simplified and quicker procedure (because we avoid inverting linear systems). On the other hand, we obtain a globally smooth approximation which is mainly suitable for smooth solutions u [29, 11]: this is a significant difference with partitioning estimates which can approximate well locally (ir-)regular functions. In this new scheme, the notion of strata is lost and the parallelization is now made with respect to the numerous required simulations and basis coefficients. Moreover, we establish tight non-asymptotic error estimates (as a function of the size of the function basis and the number of Monte-Carlo simulations).

In recent years, GPUs have become increasingly attractive for general purpose parallel computations. GPUs are not only impressive graphics tools but also highly parallel programmable processors that can deliver order-of-magnitude performance gains over optimized CPU applications. CPU hardware is designed to handle complex tasks, instead, GPUs only do one thing well, they handle millions of repetitive low level tasks. These high performance computing devices offer an arithmetic capability and a memory bandwidth that substantially outpace their CPU counterpart. In fact, a single NVIDIA Titan V GPU can sustain over 13.8 tera-floating-point operations per second (Tflops) and a memory bandwidth of 652.8 GB/s. In the last years GPUs have been heavily used in scientific computation in order to achieve dramatic runtime reductions over traditional CPU codes. Particularly successful applications of GPUs include Monte Carlo methods like the one we target in the present work. Indeed, Monte Carlo algorithms follow all design principles of GPU computing, they give rise to embarrassingly parallel not divergent codes that can exploit the GPU arithmetic intensity and its streaming memory bandwidth. In the last part of this article our goal is to offer an efficient implementation of the proposed algorithm taking advantage of several GPUs, which will allow us to address the solution of semi-linear PDEs in high dimensions.

Organization of the paper. In Section 2, we set the framework of our study, define some notations used throughout the paper and describe our algorithm based on the approximation of conditional expectations by a projection on a finite orthonormal basis of functions. Compared to the usual DPE to solve (2.1), we propose an equivalent modification that includes a weighted factor: this weight is related to

the sampling distribution and its crucial purpose is to lead to higher order approximation of the unknown function under the condition it is smooth (see the Fourier-based analysis in Section 2.4). Proofs of the propagation of error are postponed to Section 3.2. This new numerical scheme has been tested on GPU, which is exposed in Section 4. Numerical results are collected in Section 5.

Notations of the paper.

- (i) $\log(x)$ stands for the natural logarithm of $x \in \mathbb{R}_+$.
- (ii) $|\mathbf{x}|$ stands for the Euclidean norm of the vector $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$.
- (iii) $\bar{\mathbf{x}}$ stands for the l_1 -norm of the vector $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, i.e. $\bar{\mathbf{x}} = \|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$. It will be mainly used when x is a multi-index of integers used for differentiation.
- (iv) $(x)_+$ means the positive part of $x \in \mathbb{R}$ defined by $(x)_+ = \max(x, 0)$.
- (v) $\operatorname{Re}(z)$ denotes the real part of the complex $z \in \mathbb{C}$; we set $\mathbf{i} := \sqrt{-1}$.
- (vi) For any finite $L > 0$ and $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, define the soft truncation function

$$\mathcal{T}_L(\mathbf{x}) := (-L \vee x_1 \wedge L, \dots, -L \vee x_n \wedge L).$$

Moreover, if $L : \mathbb{R}^n \rightarrow \mathbb{R}^+$ is a measurable function, then the above definition of \mathcal{T}_L is extended by replacing L by $L(\mathbf{x})$:

$$\mathcal{T}_L(\mathbf{x}) := (-L(\mathbf{x}) \vee x_1 \wedge L(\mathbf{x}), \dots, -L(\mathbf{x}) \vee x_n \wedge L(\mathbf{x})). \quad (1.12)$$

- (vii) Let $\mathbf{j}, \mathbf{k} \in \mathbb{N}^d$ be two multi-indices. $\mathbf{j} \leq \mathbf{k}$ means that $j_l \leq k_l$ for all $l = 1, \dots, d$.
- (viii) The finite multi-indices set $\Lambda \subset \mathbb{N}^d$ is downward closed if

$$(\mathbf{k} \in \Lambda \text{ and } \mathbf{j} \leq \mathbf{k}) \Rightarrow \mathbf{j} \in \Lambda.$$

- (ix) Let $\Lambda \subset \mathbb{N}^d$ be an arbitrary downward closed multi-index set and $f : \mathbb{R}^d \rightarrow \mathbb{R}$. f is Λ -smooth if $\partial_{\mathbf{x}}^{\mathbf{n}} f$ exist and are continuous for all $\mathbf{n} \in \Lambda$.

2 Mathematical framework and basic properties

2.1 Model

2.1.1 Dynamic Programming Equation

From now on, we aim at solving the dynamic programming equation of the form (1.10)-(1.11). First, for the sake of simplicity, we intentionally consider the case where the non-linearity does not depend on the gradient (the Z component). Second, we consider the case of N equidistant grid times of $[0, T]$, i.e. we set $t_k = k\Delta$ with $\Delta = T/N$. Third, the forward component X is approximated by an Euler scheme with time-step Δ , see (2.6) below. These two last approximations are known to be accurate enough when data (f, g, b, σ) are sufficiently regular (see [31] for a precise statement). To simplify notation, we write X_i and Y_i for an approximation of X_{t_i} and Y_{t_i} .

All in all, our aim now is to solve

$$Y_i = \mathbb{E} \left[g(X_N) + \sum_{j=i}^{N-1} f_j(X_j, Y_{j+1}) \Delta \mid \mathcal{F}_{t_i} \right] \quad \text{for } i \in \{N-1, \dots, 0\}, \quad (2.1)$$

where $f_j(x, y) := f(t_j, x, y)$, f being the driver in (1.1). In other words, our subsequent scheme will approximate the solutions to

$$\begin{aligned} X_t &= x + \int_0^t b(s, X_s) ds + \int_0^t \sigma(s, X_s) dW_s, \\ Y_t &= \mathbb{E} \left[g(X_T) + \int_t^T f(s, X_s, Y_s) ds \mid \mathcal{F}_t \right], \end{aligned} \quad (2.2)$$

and

$$\partial_t u(t, x) + \mathcal{A}u(t, x) + f(t, x, u(t, x)) = 0 \quad \text{for } t < T \text{ and } u(T, \cdot) = g(\cdot). \quad (2.3)$$

Our assumptions on the functions b, σ, f, g are as follows.

(A_{f,g}) g is a polynomially bounded measurable function from \mathbb{R}^d to \mathbb{R} : there exist two finite non-negative constants C_g and η_g such that

$$|g(\mathbf{x})| \leq C_g(1 + |\mathbf{x}|^2)^{\eta_g/2}, \quad \forall \mathbf{x} \in \mathbb{R}^d.$$

For every $i < N$, $f_i(\mathbf{x}, y)$ is a measurable function $\mathbb{R}^d \times \mathbb{R}$ to \mathbb{R} , and there exist three finite non-negative constants L_f, C_f, η_f such that, for every $i < N$,

$$\begin{aligned} |f_i(\mathbf{x}, y) - f_i(\mathbf{x}, y')| &\leq L_f |y - y'|, \quad \forall (\mathbf{x}, y, y') \in \mathbb{R}^d \times (\mathbb{R})^2, \\ |f_i(\mathbf{x}, 0)| &\leq C_f(1 + |\mathbf{x}|^2)^{\eta_f/2}, \quad \forall \mathbf{x} \in \mathbb{R}^d. \end{aligned}$$

(A_{b,σ}) The drift coefficient $b : [0, T] \times \mathbb{R}^d \mapsto \mathbb{R}^d$ and the diffusion coefficient $\sigma : [0, T] \times \mathbb{R}^d \mapsto \mathbb{R}^d \otimes \mathbb{R}^q$ are bounded measurable functions.

One important observation is that, due to the Markov property of the Euler scheme, for every i , there exist measurable deterministic functions $y_i : \mathbb{R}^d \rightarrow \mathbb{R}$, such that

$$Y_i = y_i(X_i),$$

almost surely. See the justification in [36, Section 4.1]. Moreover, we have an uniform absolute bound for the function $y_i(\cdot)$, the proof is postponed in Section 3.3.

Proposition 1 *Assume (A_{f,g}) and (A_{b,σ}). Then, for any $\mathbf{x} \in \mathbb{R}^d$ and any $0 \leq i \leq N$, we have*

$$|y_i(\mathbf{x})| \leq C_\eta(C_g + TC_f)e^{C_\eta L_f T} (1 + |\mathbf{x}|^2)^{\frac{\eta_g \vee \eta_f}{2}} =: L^*(\mathbf{x}) \quad (2.4)$$

with

$$C_\eta := \sup_{0 \leq i \leq j \leq N, \mathbf{x} \in \mathbb{R}^d} \frac{\mathbb{E} \left[(1 + |X_j|^2)^{\frac{\eta_g \vee \eta_f}{2}} \mid X_i = \mathbf{x} \right]}{(1 + |\mathbf{x}|^2)^{\frac{\eta_g \vee \eta_f}{2}}} < +\infty. \quad (2.5)$$

A second crucial observation is that the value functions $y_i(\cdot)$ are independent of how we initialize the forward component. Our subsequent algorithm takes advantage of this observation. For instance, let X_i^i be a random variable in \mathbb{R}^d with some distribution ν (more details on this to follow) and let X_j^i be the Euler scheme evolution of X_j starting from X_i ; it writes

$$X_{j+1}^i = X_j^i + b(t_j, X_j^i)\Delta + \sigma(t_j, X_j^i)(W_{t_{j+1}} - W_{t_j}), \quad j \geq i. \quad (2.6)$$

This flexibility property w.r.t. the initialization then writes

$$y_i(X_i^i) := \mathbb{E} \left[g(X_N^i) + \sum_{j=i}^{N-1} f_j \left(X_j^i, y_{j+1}(X_{j+1}^i) \right) \Delta \mid X_i^i \right]. \quad (2.7)$$

Approximating the solution to (2.1) is actually achieved by approximating the functions $y_i(\cdot)$. In this way, we are directly approximating the solution to the semi-linear PDE (2.3). For some reasons that will appear clear later, we define a weighted modification of y_i by

$$y_i^{(q)}(\mathbf{x}) = \frac{y_i(\mathbf{x})}{(1 + |\mathbf{x}|^2)^{q/2}} \quad (2.8)$$

for a damping exponent $q \geq 0$. For $q = 0$, $y_i^{(q)}$ and y_i coincide. The previous DPE (2.7) becomes

$$y_i^{(q)}(X_i^i) := \mathbb{E} \left[\frac{g(X_N^i)}{(1 + |X_i^i|^2)^{q/2}} + \sum_{j=i}^{N-1} \frac{f_j \left(X_j^i, y_{j+1}^{(q)}(X_{j+1}^i) (1 + |X_{j+1}^i|^2)^{q/2} \right)}{(1 + |X_i^i|^2)^{q/2}} \Delta \mid X_i^i \right]. \quad (2.9)$$

The introduction of the polynomial factor $(1 + |X_i^i|^2)^{q/2}$ gives higher flexibility in the error analysis, it ensures that $y_i^{(q)}$ decreases faster at infinity, which will provide nicer estimates on the approximation error when dealing with Fourier-basis, see Section 2.4.

2.1.2 Randomizing the initial value X_i^i

For the subsequent algorithm, X_i will be sampled randomly according to the distribution ν , which we list the properties below.

(\mathbf{A}_ν) The distribution $\nu(d\mathbf{x})$ of X_i^i has a density with respect to the Lebesgue measure on \mathbb{R}^d , of the form

$$\nu(\mathbf{x}) = \prod_{l=1}^d \nu_l(x_l), \quad \mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d, \quad (2.10)$$

for some probability density functions ν_l , which are assumed to be strictly positive.

The function $\mathbf{x} \in \mathbb{R}^d \mapsto \frac{L^*(\mathbf{x})}{(1 + |\mathbf{x}|^2)^{q/2}}$ (where L^* is defined in (2.4)) is square-integrable on \mathbb{R}^d with respect to $\nu(\mathbf{x})d\mathbf{x}$.

We denote by F_{ν_l} the cumulative distribution function (CDF) of the one-dimensional marginal of ν_l ,

$$F_{\nu_l}(x) := \int_{-\infty}^x \nu_l(x')dx',$$

and by $F_{\nu_l}^{-1}$ its inverse (which is standardly defined since F_{ν_l} is continuous and strictly increasing).

Note that the above integrability condition ensures that $y_i^{(q)}$ (for any i) is square integrable on \mathbb{R}^d with respect to $\nu(\mathbf{x})d\mathbf{x}$.

In this work, we deal essentially with one type of distribution.

Example 1 (Student's t-distribution) We deal with the non-standardized Student's t-distribution on \mathbb{R}^d with parameter $\mu > 0$ defined by its probability density function of the form (2.10) with

$$\nu_l(x_l) = \frac{c_\mu}{(1 + x_l^2)^{(\mu+1)/2}}, \quad (2.11)$$

where $c_\mu \in \mathbb{R}^+$. This is the density of $X/\sqrt{\mu}$ where X is a standard Student's t-distribution with μ degrees of freedom. It leads to numerous ways to simulate ν_l , as acceptance rejection or ratio-of-uniforms methods, for example. From [40, p. 507] we know that

$$c_\mu = \frac{\Gamma((\mu+1)/2)\sqrt{\mu}}{\Gamma(\mu/2)\sqrt{\pi\mu}} = \frac{\Gamma((\mu+1)/2)}{\Gamma(\mu/2)\sqrt{\pi}},$$

where Γ is the Gamma function defined in [40, p. 501]. Note that the marginal CDF and its inverse are not explicit in general for the Student's t-distribution, however their values (for usual parameter μ) are tabulated in most of books in statistics and computable using standard software libraries. Moreover, a priori estimates on F_{ν_l} , $F_{\nu_l}^{-1}$ and $d_u^n F_{\nu_l}^{-1}(u)$, which will be used in the error analysis of the algorithm, are computed in the Appendix A.

Note that the square integrability of $\mathbf{x} \in \mathbb{R}^d \mapsto \frac{L^*(\mathbf{x})}{(1 + |\mathbf{x}|^2)^{q/2}}$ (see Assumption (\mathbf{A}_ν)) is guaranteed by the condition $\mu > \eta_g \vee \eta_f - q$.

The distribution we have considered so far has mean zero, this is suitable for problems where 0 has a natural interpretation as the center in the application at hand. In order to solve the problem far away from the neighborhood of zero we should allow non zero means (say $\bar{x} \in \mathbb{R}^d$): this is easy to handle by shifting the above distribution by \bar{x} . For instance, in the case of the Student's t-distribution (Example 1), we would take

$$\nu_l(x_l) = c_\mu(1 + (x_l - \bar{x}_l)^2)^{-(\mu+1)/2}. \quad (2.12)$$

Since this centering adjustment is quite obvious, we mainly discuss the case of Example 1 in what follows.

More importantly, the probability measure $\nu(d\mathbf{x})$ is expected to satisfy another assumption stated below.

$\mathbf{H}_\rho(\mathbf{q})$ $\exists c_\nu > 0$ (depending on q, b, σ, ν, T) such that for any measurable function $\gamma : \mathbb{R}^d \rightarrow \mathbb{R}$ with $\gamma \in L_\nu^2(\mathbb{R}^d)$, any $N \geq j \geq i \geq 0$, we have

$$\mathbb{E} \left[\frac{|\gamma(X_j^i)|^2 (1 + |X_j^i|^2)^q}{(1 + |X_i^i|^2)^q} \right] \leq c_\nu \mathbb{E} \left[|\gamma(X_i^i)|^2 \right]. \quad (2.13)$$

This is a stability property about expectation of weighted functionals of the Markov chain $(X_j^i)_{i \leq j \leq N}$. This property will be crucial for analyzing the propagation of error in the quasi-regression scheme (see the proof in Section 3.2). We now illustrate this property in relation with previous Example 1, the proof is postponed to Section 3.1.

Theorem 1 (Norm-stability) *Assume that X is the Euler scheme (2.6) with time step Δ of a SDE (2.2) which drift coefficient $b : [0, T] \times \mathbb{R}^d \mapsto \mathbb{R}^d$ is bounded measurable, and which diffusion coefficient $\sigma : [0, T] \times \mathbb{R}^d \mapsto \mathbb{R}^d \otimes \mathbb{R}^q$ is bounded measurable and η -Hölder continuous in space uniformly in time (for some $\eta \in (0, 1]$),*

$$\sup_{0 \leq t \leq T} \sup_{\mathbf{x} \neq \mathbf{x}'} \frac{|\sigma(t, \mathbf{x}) - \sigma(t, \mathbf{x}')|}{|\mathbf{x} - \mathbf{x}'|^\eta} < +\infty.$$

In addition, we assume that σ fulfills a uniform ellipticity condition:

$$\inf_{0 \leq t \leq T, \mathbf{x} \in \mathbb{R}^d} \inf_{\zeta \in \mathbb{R}^d \text{ s.t. } |\zeta|=1} \zeta \cdot \sigma \sigma^\top(t, \mathbf{x}) \zeta > 0.$$

Let X_i^i be sampled from distribution ν in the case of Student's t -distribution (Example 1): then $\mathbf{H}_\rho(\mathbf{q})$ holds for any $q \geq 0$.

2.2 Approximation spaces and basis functions

The aim of this section is to define some proper basis functions which satisfy orthogonality properties in \mathbb{R}^d and which span some L^2 space, see Proposition 2. It turns out that the choice of measure for defining the L^2 space has to coincide with the sampling measure of X_i^i defined in (\mathbf{A}_ν) : this plays an important role for designing Algorithm 1. Our strategy for defining such basis functions is to start from trigonometric basis on $[0, 1]^d$ and then to apply appropriate changes of variable: later, this transform will allow to easily quantify the approximation error when truncating the basis, see Propositions 3 and 4.

Notation. We start by setting notations to deal with appropriate L^2 space on \mathbb{R}^d or $[0, 1]^d$.

- (i) Let $\nu : \mathbb{R}^d \rightarrow \mathbb{R}$ be a probability density satisfying (\mathbf{A}_ν) . We will consider the space of measurable functions

$$L_\nu^2(\mathbb{R}^d) := \left\{ v : \mathbb{R}^d \rightarrow \mathbb{R} \text{ such that } \int_{\mathbb{R}^d} v^2 \nu d\mathbf{x} < \infty \right\},$$

$$L_U^2([0, 1]^d) := \left\{ w : [0, 1]^d \rightarrow \mathbb{R} \text{ such that } \int_{[0, 1]^d} w^2 d\mathbf{u} < \infty \right\}.$$

In these spaces we define the inner products,

$$(v_1, v_2)_{L^2_\nu(\mathbb{R}^d)} = \int_{\mathbb{R}^d} v_1 v_2 \nu d\mathbf{x}, \quad \forall v_1, v_2 \in L^2_\nu(\mathbb{R}^d),$$

$$(w_1, w_2)_{L^2_U([0,1]^d)} = \int_{[0,1]^d} w_1 w_2 d\mathbf{u}, \quad \forall w_1, w_2 \in L^2_U([0,1]^d),$$

and the norms,

$$\|v\|_{L^2_\nu(\mathbb{R}^d)} = \sqrt{(v, v)_{L^2_\nu(\mathbb{R}^d)}} = \left(\int_{\mathbb{R}^d} v^2 \nu d\mathbf{x} \right)^{\frac{1}{2}}, \quad \forall v \in L^2_\nu(\mathbb{R}^d),$$

$$\|w\|_{L^2_U([0,1]^d)} = \sqrt{(w, w)_{L^2_U([0,1]^d)}} = \left(\int_{[0,1]^d} w^2 d\mathbf{u} \right)^{\frac{1}{2}}, \quad \forall w \in L^2_U([0,1]^d).$$

Before the general multidimensional case, we start with an easy computation in dimension 1.

Lemma 1 (Orthonormal basis functions in dimension 1) *Assume that ν satisfies (\mathbf{A}_ν) in dimension $d = 1$. Let $x \in \mathbb{R}$ and $k \in \mathbb{N}$. The basis functions*

$$\phi_k(x) := \begin{cases} 1, & \text{if } k = 0, \\ \sqrt{2} \cos(k\pi F_{\nu_1}(x)), & \text{if } k > 0, \end{cases} \quad (2.14)$$

are orthonormal in \mathbb{R} with respect to the probability density weight $\nu = \nu_1$.

Proof Let $i, j \in \mathbb{N}$. It is sufficient to see that

$$\int_{-\infty}^{\infty} \cos(i\pi F_{\nu_1}(x)) \cos(j\pi F_{\nu_1}(x)) \nu_1(x) dx = \int_0^1 \cos(i\pi u) \cos(j\pi u) du,$$

where we have used the change of variable $u = F_{\nu_1}(x)$. Clearly the cosine basis functions

$$C_k(u) = \begin{cases} 1, & \text{if } k = 0, \\ \sqrt{2} \cos(k\pi u), & \text{if } k > 0, \end{cases} \quad (2.15)$$

are orthonormal in $[0, 1]$ with respect to the Lebesgue measure, therefore $(\phi_i, \phi_j)_{L^2_\nu(\mathbb{R})} = \mathbf{1}_{i=j}$. \square

Note that $(C_k \circ F_{\nu_1})(x) = \phi_k(x)$ and $(\phi_k \circ F_{\nu_1}^{-1})(u) = C_k(u)$. This change of variable will be useful in order to analyze the truncation error of ν_1 series in terms of the truncation error of Fourier cosine series (see Section 2.4).

Now that the reader gets acquainted with the one-dimensional case, we move to the more general result for any dimension $d \geq 1$. First, the natural way to build an orthonormal multidimensional set of basis functions is to take tensor products of orthonormal one-dimensional basis functions. To this end we need to add the superscript (l) to the notations of the basis functions in dimension 1. Therefore, from now on, $\phi_{k_l}^{(l)}$ and $C_{k_l}^{(l)}$ stand for the one-dimensional basis functions (2.14) and (2.15) in the l -th coordinate, respectively. For any multiindex $\mathbf{k} = (k_1, \dots, k_d)$, set

$$\phi_{\mathbf{k}}(\mathbf{x}) := \prod_{l=1}^d \phi_{k_l}^{(l)}(x_l), \quad \mathbf{x} := (x_1, \dots, x_d) \in \mathbb{R}^d, \quad (2.16)$$

$$C_{\mathbf{k}}(\mathbf{u}) := \prod_{l=1}^d C_{k_l}^{(l)}(u_l), \quad \mathbf{u} := (u_1, \dots, u_d) \in [0, 1]^d. \quad (2.17)$$

The transformation function $F_\nu : \mathbb{R}^d \rightarrow [0, 1]^d$ and its inverse are defined by

$$F_\nu(\mathbf{x}) := (F_{\nu_1}(x_1), \dots, F_{\nu_1}(x_d)), \quad F_\nu^{-1}(\mathbf{u}) := (F_{\nu_1}^{-1}(u_1), \dots, F_{\nu_d}^{-1}(u_d)).$$

Therefore, from the above and from (2.16)-(2.17), we readily check the relations

$$(C_{\mathbf{k}} \circ F_\nu)(\mathbf{x}) := \prod_{l=1}^d (C_{k_l}^{(l)} \circ F_{\nu_l})(x_l) = \phi_{\mathbf{k}}(\mathbf{x}), \quad (2.18)$$

$$(\phi_{\mathbf{k}} \circ F_\nu^{-1})(\mathbf{u}) := \prod_{l=1}^d (\phi_{k_l}^{(l)} \circ F_{\nu_l}^{-1})(u_l) = C_{\mathbf{k}}(\mathbf{u}). \quad (2.19)$$

To analyze orthogonality properties, we repeatedly use the following lemma, which proof is quite straightforward and left to the reader (use the same change-of-variable arguments as in the proof of Lemma 1).

Lemma 2 *Assume ν satisfies (\mathbf{A}_ν) . Then, for all functions $v_1, v_2 \in L_\nu^2(\mathbb{R}^d)$, we have the following identities*

$$(v_1, v_2)_{L_\nu^2(\mathbb{R}^d)} = (v_1 \circ F_\nu^{-1}, v_2 \circ F_\nu^{-1})_{L_U^2([0, 1]^d)}, \quad \|v_1\|_{L_\nu^2(\mathbb{R}^d)} = \left\| v_1 \circ F_\nu^{-1} \right\|_{L_U^2([0, 1]^d)}.$$

As a consequence of Lemma 2 and Equality (2.19), since the tensorized cosine functions $(C_{\mathbf{k}}(\cdot) : \mathbf{k} \in \mathbb{N}^d)$ are orthonormal in $[0, 1]^d$ with respect to the Lebesgue measure, it implies that $(\phi_{\mathbf{k}}(\cdot) : \mathbf{k} \in \mathbb{N}^d)$ are orthonormal in \mathbb{R}^d with respect to the probability measure $\nu(d\mathbf{x})$. Moreover, the cosine functions $(C_{\mathbf{k}}(\cdot) : \mathbf{k} \in \mathbb{N}^d)$ form a complete family in $L_U^2([0, 1]^d)$: this is a standard result in Fourier analysis (observe that here, the sine functions are not needed to get a complete basis since the scalar product $(\cdot, \cdot)_{L_U^2([0, 1]^d)}$ is defined on $[0, 1]^d$ and not $[-1, 1]^d$). We summarize our findings in a proposition ready to be used.

Proposition 2 *Assume ν satisfies (\mathbf{A}_ν) . The family of functions $(\phi_{\mathbf{k}}(\cdot) : \mathbf{k} \in \mathbb{N}^d)$ forms a complete orthonormal basis of \mathbb{R}^d with respect to the measure ν .*

2.3 Global Quasi-Regression algorithm and convergence results

We now describe our main algorithm which is able to compute accurately the function $y_i^{(q)}$ defined in (2.9). For this, define

$$S_i^{(q)}(x_{i:N}^i) := \frac{g(x_N^i)}{(1 + |x_i^i|^2)^{q/2}} + \sum_{j=i}^{N-1} \frac{f_j(x_j^i, y_{j+1}^{(q)}(x_{j+1}^i)(1 + |x_{j+1}^i|^2)^{q/2})}{(1 + |x_i^i|^2)^{q/2}} \Delta, \quad (2.20)$$

$$x_{i:N}^i := (x_i^i, \dots, x_N^i) \in (\mathbb{R}^d)^{N-i+1},$$

which allows us to rewrite the exact solution as

$$y_i^{(q)}(\mathbf{x}) = \mathbb{E} \left[S_i^{(q)}(X_{i:N}^i) \mid X_i^i = \mathbf{x} \right], \quad \mathbf{x} \in \mathbb{R}^d.$$

Note that it is easy to justify that $S_i^{(q)}(X_{i:N}^i)$ is square-integrable under $(\mathbf{A}_{\mathbf{f},\mathbf{g}})$, (\mathbf{A}_ν) and $\mathbf{H}_\rho(\mathbf{q})$ (remind of the bound (2.4) and the integrability condition in (\mathbf{A}_ν)). Therefore, the exact solution $y_i^{(q)}$ is in $L_\nu^2(\mathbb{R}^d)$, and consequently (Proposition 2) it can be decomposed in its ϕ series:

$$y_i^{(q)}(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}^d} \alpha_{i,\mathbf{k}}^{(q)} \phi_{\mathbf{k}}(\mathbf{x}) \quad (2.21)$$

for some coefficients $(\alpha_{i,\mathbf{k}}^{(q)} : \mathbf{k} \in \mathbb{N}^d)$. In practice, the coefficients $\alpha_{i,\mathbf{k}}^{(q)}$ are computed only in a finite set of multi-indices $\Gamma \subset \mathbb{N}^d$. Different choices of Γ are discussed later in Section 2.4.2.

The orthonormality property of the $\phi_{\mathbf{k}}$'s (Proposition 2) combined with (2.21) yields

$$\alpha_{i,\mathbf{k}}^{(q)} = (y_i^{(q)}, \phi_{\mathbf{k}})_{L_\nu^2(\mathbb{R}^d)}. \quad (2.22)$$

Thus, using the tower property of conditional expectation, we obtain, $\forall \mathbf{k} \in \mathbb{N}^d$,

$$\begin{aligned} \alpha_{i,\mathbf{k}}^{(q)} &= \mathbb{E} \left[y_i^{(q)}(X_i^i) \phi_{\mathbf{k}}(X_i^i) \right] = \mathbb{E} \left[\mathbb{E} \left[S_i^{(q)}(X_{i:N}^i) \mid X_i^i \right] \phi_{\mathbf{k}}(X_i^i) \right] \\ &= \mathbb{E} \left[S_i^{(q)}(X_{i:N}^i) \phi_{\mathbf{k}}(X_i^i) \right]. \end{aligned} \quad (2.23)$$

The above expectations in (2.23) can be computed using Monte-Carlo simulations and evaluations of $S_i^{(q)}$ along different sampled paths $x_{i:N}^i$. Since $S_i^{(q)}$ depends on the unknown solution, we have to replace it by approximations, using previous steps in the dynamic programming equation (2.9). It gives rise to the following Algorithm 1, which we will call *Global Quasi-Regression Multistep-forward Dynamical Programming* (GQRMDP) algorithm. To maintain good integrability properties along the iteration, we use the truncation function \mathcal{T}_{L^*} (1.12) with the upper bound (2.4).

Algorithm 1 (GQRMDP algorithm)

Initialization. Set $\bar{y}_N^{(q,M)}(x_N) := \frac{g(x_N)}{(1+|x_N|^2)^{q/2}}$.

Backward iteration for $i = N - 1$ **to** $i = 0$,

$$\bar{y}_i^{(q,M)}(\cdot) := \sum_{\mathbf{k} \in \Gamma} \bar{\alpha}_{i,\mathbf{k}}^{(q,M)} \phi_{\mathbf{k}}(\cdot), \quad (2.24)$$

where for all $\mathbf{k} \in \Gamma$,

$$\bar{\alpha}_{i,\mathbf{k}}^{(q,M)} := \frac{1}{M} \sum_{m=1}^M S_i^{(q,M)}(X_{i:N}^{i,m}) \phi_{\mathbf{k}}(X_i^{i,m}), \quad (2.25)$$

and

$$\begin{aligned} S_i^{(q,M)}(x_{i:N}^i) &:= \frac{g(x_N^i)}{(1+|x_N^i|^2)^{q/2}} \\ &+ \sum_{j=i}^{N-1} \frac{f_j \left(x_j^i, \mathcal{T}_{L^*} \left(\bar{y}_{j+1}^{(q,M)}(x_{j+1}^i) (1+|x_{j+1}^i|^2)^{q/2} \right) \right)}{(1+|x_j^i|^2)^{q/2}} \Delta. \end{aligned} \quad (2.26)$$

In the algorithm, in order to compute $\bar{y}_i^{(q,M)}(\cdot)$ we have considered M independent copies of

$$X_i^i \xrightarrow{\text{Euler scheme}} X_{i+1}^i \xrightarrow{\text{Euler scheme}} \dots \xrightarrow{\text{Euler scheme}} X_N^i,$$

which form a cloud of simulations $\mathcal{C}_i = \{X_{i:N}^{i,m} : m = 1, \dots, M\}$. We assume that the clouds of simulations $(\mathcal{C}_i : 0 \leq i < N)$ are independently generated.

Our main result is the following error propagation theorem, which proof is postponed to Section 3.2.

Theorem 2 *Assume $(\mathbf{A}_{\mathbf{f},\mathbf{g}})$, $(\mathbf{A}_{\mathbf{b},\sigma})$, (\mathbf{A}_ν) and $\mathbf{H}_\rho(\mathbf{q})$. Let $\mathbf{k} \in \mathbb{N}^d$ be a multi-index and let $\Gamma \subset \mathbb{N}^d$ be a finite set of multi-indices. Let $(\phi_{\mathbf{k}}(\cdot) : \mathbf{k} \in \mathbb{N}^d)$ be the basis functions defined in (2.18), as an orthonormal basis in $L_\nu^2(\mathbb{R}^d)$, and let \mathcal{L}_Γ be the related Christoffel number [50] restricted to Γ :*

$$\mathcal{L}_\Gamma = \sum_{\mathbf{k} \in \Gamma} \|\phi_{\mathbf{k}}(\cdot)\|_\infty^2 \leq 2^d \#\Gamma. \quad (2.27)$$

For each $i \in \{0, \dots, N-1\}$, define the following local error term

$$\mathcal{E}_i := \sum_{\mathbf{k} \in (\mathbb{N}^d - \Gamma)} (\alpha_{i,\mathbf{k}}^{(q)})^2 + \frac{\mathcal{L}_\Gamma}{M} \mathbb{E} \left[\left(S_i^{(q)}(X_{i:N}^i) \right)^2 \right]. \quad (2.28)$$

There is a finite positive constant C (depending on model parameters but not on N, M, Γ) such that

$$\mathbb{E} \left[\left\| \bar{y}_i^{(q,M)}(\cdot) - y_i^{(q)}(\cdot) \right\|_\nu^2 \right] \leq C \left(\mathcal{E}_i + \sum_{j=i}^{N-1} \mathcal{E}_{j+1} \Delta \right).$$

Observe that provided that $\frac{\mathcal{L}_\Gamma}{M}$ remains bounded, the constant C can be taken as uniform w.r.t. the algorithm parameters. This is the situation in practice since for getting accurate approximations, we need $\frac{\mathcal{L}_\Gamma}{M}$ to be small.

The convergence rate of the term $\sum_{\mathbf{k} \in (\mathbb{N}^d - \Gamma)} (\alpha_{i,\mathbf{k}}^{(q)})^2$ is discussed in Section 2.4 considering ν as the Student's t-distribution (Example 1). This analysis will be undertaken assuming that $y_i(\mathbf{x})$ has a polynomial growth in \mathbf{x} at infinity (coherently with the a priori estimate of Proposition 1) as well as its derivatives.

2.4 Truncation error for the Student-cosine basis

From Equations (2.21)-(2.22) and Lemma 2 we recall that $y_i^{(q)} = \sum_{\mathbf{k} \in \mathbb{N}^d} \alpha_{i,\mathbf{k}}^{(q)} \phi_{\mathbf{k}}$, where

$$\alpha_{i,\mathbf{k}}^{(q)} = (y_i^{(q)}, \phi_{\mathbf{k}})_{L_\nu^2(\mathbb{R}^d)} = (y_i^{(q)} \circ F_\nu^{-1}, C_{\mathbf{k}})_{L_\nu^2([0,1]^d)}. \quad (2.29)$$

Define the truncation of the ν -series of $y_i^{(q)}$ by

$$P_\Gamma y_i^{(q)} := \sum_{\mathbf{k} \in \Gamma} \alpha_{i,\mathbf{k}}^{(q)} \phi_{\mathbf{k}}.$$

Therefore, from orthonormal properties of the basis functions, we have

$$\left\| y_i^{(q)} - P_\Gamma y_i^{(q)} \right\|_{L_\nu^2(\mathbb{R}^d)}^2 = \left\| y_i^{(q)} \circ F_\nu^{-1} - \sum_{\mathbf{k} \in \Gamma} \alpha_{i,\mathbf{k}}^{(q)} C_{\mathbf{k}} \right\|_{L_\nu^2([0,1]^d)}^2 = \sum_{\mathbf{k} \in (\mathbb{N}^d - \Gamma)} |\alpha_{i,\mathbf{k}}^{(q)}|^2.$$

In other words, $\alpha_{i,\mathbf{k}}^{(q)}$ are the Fourier coefficients of the projection of

$$h_i^{(q)} := y_i^{(q)} \circ F_\nu^{-1} \in L_U^2([0,1]^d) \quad (2.30)$$

over cosine basis functions. In the following section we will take advantage of this property to derive a tight bound for the truncation error $\sum_{\mathbf{k} \in (\mathbb{N}^d - \Gamma)} |\alpha_{i,\mathbf{k}}^{(q)}|^2$, using assumptions on $y_i^{(q)}$ and its derivatives only (and not on $y_i^{(q)} \circ F_\nu^{-1}$). The specific choice of the Student's t -distribution measure ν of Example 1 plays an important role to get tractable conditions.

Now our aim is twofold. Firstly we prove that if y_i is r times ($r \geq 1$) continuously differentiable on \mathbb{R}^d , with (for some $p \in \mathbb{N}$),

$$\limsup_{|\mathbf{x}| \rightarrow +\infty} \frac{|\partial_{x_i}^n y_i(\mathbf{x})|}{|\mathbf{x}|^{p-n}} < +\infty, \quad \text{for } n = 0, \dots, r, \quad l = 1, \dots, d,$$

and if the parameters are such that

$$p + \left(r - \frac{1}{2} \right) \mu < q,$$

then the truncation error is controlled as

$$\sum_{\mathbf{k} \in \mathbb{N}^d: \exists k_l > K_l} |\alpha_{i,\mathbf{k}}^{(q)}|^2 \leq \sum_{l=1}^d \frac{C}{(K_l + 1)^{2r}}$$

for some constant $C > 0$. In the following, we first establish the above result in dimension 1 (Proposition 3) and then in dimension d (Proposition 4).

Our second goal is improving this result when mixed derivatives of y_i are also supposed smooth. Thus, we prove that if y_i is Λ -smooth, where $\Lambda(r) = \{0, \dots, r\}^d$, with (for some $p \in \mathbb{N}$),

$$\limsup_{|\mathbf{x}| \rightarrow +\infty} \frac{|\partial_{\mathbf{x}}^{\mathbf{n}} y_i(\mathbf{x})|}{|\mathbf{x}|^{p-\bar{\mathbf{n}}}} < +\infty, \quad \text{for } \mathbf{n} \in \Lambda,$$

and if the parameters are such that

$$p + \left(d r - \frac{1}{2} \right) \mu < q,$$

then the truncation error satisfies the tail inequality

$$\sum_{\mathbf{k} \in \bar{F}_H(\text{DEG})} |\alpha_{i,\mathbf{k}}^{(q)}|^2 \leq \frac{C}{(\text{DEG} + 1)^{2r}},$$

for some constant $C > 0$ (Proposition 5).

2.4.1 One-dimensional case

In this section, we let $\nu = \nu_1$ be the Student density (2.11).

Lemma 3 *Assume that y_i is r times ($r \in \mathbb{N}$) continuously differentiable on \mathbb{R} , with the existence of a polynomial growth degree $p \in \mathbb{N}$ such that*

$$\limsup_{x \rightarrow \pm\infty} \frac{|d_x^n y_i(x)|}{|x|^{p-n}} < +\infty, \quad \text{for } n = 0, \dots, r. \quad (2.31)$$

Assume the relation

$$p + r\mu < q \quad (2.32)$$

between r , p , the Student parameter $\mu > 0$ and the damping exponent $q \geq 0$. Then $h_i^{(q)}$ is r times continuously differentiable on $(0, 1)$, and its first r derivatives vanish at $u = 0$ and $u = 1$:

$$\lim_{u \rightarrow 0^+, 1^-} d_u^n h_i^{(q)}(u) = 0, \quad n = 0, \dots, r.$$

Proof The regularities of $y_i^{(q)}$ and $h_i^{(q)}$ are straightforward in view of (2.8)-(2.30) and Lemma 15. The estimate $\lim_{u \rightarrow 0^+, 1^-} h_i^{(q)}(u) = 0$ is clearly true since it is equivalent to $0 = \lim_{x \rightarrow \pm\infty} y_i^{(q)}(x)$, which holds owing to the condition $q > p$.

From now on, we may consider the case $n \geq 1$ and $r \geq 1$. We start showing that for $1 \leq n \leq r$,

$$d_x^n y_i^{(q)}(x) = O(x^{p-n-q}) \text{ as } x \rightarrow \pm\infty. \quad (2.33)$$

In fact, using Leibniz rule,

$$d_x^n y_i^{(q)}(x) = \sum_{m=0}^n \binom{n}{m} d_x^{n-m} y_i(x) d_x^m \frac{1}{(1+x^2)^{q/2}}.$$

Besides,

$$d_x^{n-m} y_i(x) = O(x^{p-n+m}), \quad d_x^m \frac{1}{(1+x^2)^{q/2}} = O(x^{-q-m}), \quad \text{as } x \rightarrow \pm\infty,$$

which leads directly to the desired result (2.33).

We now estimate $d_u^n h_i^{(q)}(u)$. In order to compute the n -th derivative of $h_i^{(q)}$ we take advantage of the following Faà di Bruno's formula [42, pp. 224–226],

$$\begin{aligned} d_u^n h_i^{(q)}(u) &= d_u^n y_i^{(q)}(F_{\nu_1}^{-1}(u)) \\ &= \sum_{m=1}^n \frac{1}{m!} d_x^m y_i^{(q)}(x)|_{x=F_{\nu_1}^{-1}(u)} \sum_{\mathbf{j} \in J_{m,n}} \frac{n!}{j_1! j_2! \dots j_m!} \prod_{i=1}^m d_u^{j_i} F_{\nu_1}^{-1}(u), \\ J_{m,n} &= \{\mathbf{j} = (j_1, \dots, j_m) \in \mathbb{N}_+^m : j_1 + \dots + j_m = n\}. \end{aligned} \quad (2.34)$$

As $u \rightarrow 0^+$, using also (A.7),

$$d_u^n h_i^{(q)}(u) = O\left(\max_{1 \leq m \leq n, \mathbf{j} \in J_{m,n}} \left| \left(F_{\nu_1}^{-1}(u) \right)^{p-m-q} d_u^{j_1} F_{\nu_1}^{-1}(u) \dots d_u^{j_m} F_{\nu_1}^{-1}(u) \right|\right)$$

$$\begin{aligned}
&= O\left(\max_{1 \leq m \leq n, \mathbf{j} \in J_{m,n}} u^{-\frac{p-m-q}{\mu}} u^{-\frac{1+j_1\mu}{\mu}} \cdots u^{-\frac{1+j_m\mu}{\mu}}\right) \\
&= O\left(u^{-\frac{p+n\mu-q}{\mu}}\right).
\end{aligned} \tag{2.35}$$

The above bound goes to 0 as $u \rightarrow 0^+$ since $p + n\mu - q \leq p + r\mu - q < 0$. Analogously, as $u \rightarrow 1^-$, we have

$$d_u^n h_i^{(q)}(u) = O\left((1-u)^{-\frac{p+n\mu-q}{\mu}}\right) \tag{2.36}$$

which converges to 0 as $u \rightarrow 1^-$ since $p + n\mu < q$. \square

Lemma 4 *Assume the hypotheses of Lemma 3 and replace the condition (2.32) by the weaker condition*

$$p + \left(r - \frac{1}{2}\right)\mu < q. \tag{2.37}$$

Then the r first derivatives of $h_i^{(q)}$ are square integrable on $(0, 1)$:

$$\left\|d_u^n h_i^{(q)}\right\|_{L_V^2([0,1])}^2 < +\infty, \quad n = 0, \dots, r.$$

Proof The integrability of $d_u^n h_i^{(q)}$ on any compact set of the form $[\varepsilon, 1 - \varepsilon]$ ($\varepsilon > 0$) is obvious owing to the continuity of the function to integrate. As a consequence one has only to consider what happens around 0 and 1. As we have just seen in the proof of Lemma 3 (see (2.35)-(2.36)),

$$d_u^n h_i^{(q)}(u) = O_{u \rightarrow 0^+}\left(u^{-\frac{p+n\mu-q}{\mu}}\right), \quad d_u^n h_i^{(q)}(u) = O_{u \rightarrow 1^-}\left((1-u)^{-\frac{p+n\mu-q}{\mu}}\right);$$

therefore it is square-integrable around $u = 0$ and $u = 1$ provided that $-\frac{p+n\mu-q}{\mu} > -\frac{1}{2}$. Since $n \leq r$ the condition (2.37) is sufficient. \square

Proposition 3 *Assume that y_i is r times ($r \geq 1$) continuously differentiable on \mathbb{R} , with the existence of a polynomial growth degree $p \in \mathbb{N}$ such that (2.31) holds. Assume the relation (2.37) holds. Then, the coefficients $(\alpha_{i,k}^{(q)}, k \in \mathbb{N})$ defined in (2.29) satisfy the tail inequality (for any $K \geq 0$)*

$$\sum_{k=K+1}^{\infty} |\alpha_{i,k}^{(q)}|^2 \leq \frac{C_{(2.38)}^{(r)}}{(K+1)^{2r}}, \quad \text{where } C_{(2.38)}^{(r)} := \frac{1}{\pi^{2r}} \left\|d_u^r h_i^{(q)}\right\|_{L_V^2([0,1])}^2 < +\infty. \tag{2.38}$$

Proof Under (2.37), we can invoke Lemma 3 (with $r - 1$ derivatives) and Lemma 4 (with r derivatives), to get

$$\begin{aligned}
\lim_{u \rightarrow 0^+, 1^-} d_u^n h_i^{(q)}(u) &= 0, \quad \text{for } n = 0, \dots, r-1, \\
\left\|d_u^n h_i^{(q)}\right\|_{L_V^2([0,1])}^2 &< +\infty, \quad \text{for } n = 0, \dots, r.
\end{aligned} \tag{2.39}$$

The coefficients $(\alpha_{i,k}^{(q)}, k \in \mathbb{N})$ coincide with the Fourier coefficients of the decomposition of $h_i^{(q)}$ with respect to the orthonormal basis functions $\{C_k, k \in \mathbb{N}\}$ in $L_U^2([0, 1])$ (see the relations (2.29)-(2.30)). The C_k -cosine series of $h_i^{(q)}$ writes

$$h_i^{(q)}(u) = \sum_{k=0}^{\infty} \alpha_{i,k}^{(q)} C_k(u) = \alpha_{i,0}^{(q)} + \sum_{k=1}^{\infty} \alpha_{i,k}^{(q)} C_k(u),$$

where

$$\alpha_{i,0}^{(q)} = \int_0^1 h_i^{(q)}(u) du \quad \text{and} \quad \alpha_{i,k}^{(q)} = \sqrt{2} \int_0^1 h_i^{(q)}(u) \cos(k\pi u) du \quad \text{for } k > 0.$$

Applying integration by parts ($h_i^{(q)}$ is continuously differentiable) and using that $\lim_{u \rightarrow 0^+, 1^-} h_i^{(q)}(u) = 0$ we get (for $k \neq 0$)

$$\begin{aligned} \alpha_{i,k}^{(q)} &= \operatorname{Re} \left(\sqrt{2} \lim_{\varepsilon \rightarrow 0} \int_{\varepsilon}^{1-\varepsilon} h_i^{(q)}(u) e^{ik\pi u} du \right) \\ &= \operatorname{Re} \left(\frac{\sqrt{2}}{ik\pi} \left[h_i^{(q)}(u) e^{ik\pi u} \right]_{0^+}^{1^-} - \frac{\sqrt{2}}{ik\pi} \int_0^1 d_u h_i^{(q)}(u) e^{ik\pi u} du \right) \\ &= \operatorname{Re} \left(\frac{\sqrt{2}}{-ik\pi} \int_0^1 d_u h_i^{(q)}(u) e^{ik\pi u} du \right). \end{aligned} \quad (2.40)$$

Thanks to (2.62), we can repeat $r-1$ times the integration by parts as above, and we obtain

$$\begin{aligned} \alpha_{i,k}^{(q)} &= \operatorname{Re} \left(\frac{\sqrt{2}}{(-ik\pi)^r} \int_0^1 d_u^r h_i^{(q)}(u) e^{ik\pi u} du \right) \\ &= \frac{1}{(k\pi)^r} \int_0^1 d_u^r h_i^{(q)}(u) \operatorname{Re} \left(\sqrt{2} i^r e^{ik\pi u} \right) du. \end{aligned} \quad (2.41)$$

Observe that the sets

$$\{u \mapsto \sqrt{2} \cos(\pi k u), k \geq 1\},$$

and

$$\{u \mapsto \sqrt{2} \sin(\pi k u), k \geq 1\},$$

are made of orthonormal functions in $L_U^2([0, 1])$. Consequently, for any value of r , $\{\operatorname{Re}(\sqrt{2} i^r e^{ik\pi u}), k \geq 1\}$ is a set of orthonormal functions and the equality (2.41) means that $\{(k\pi)^r \alpha_{i,k}^{(q)} : k \geq 1\}$ are the projection coefficients of $\pm d_u^r h_i^{(q)}$ on these orthonormal functions. As such, the Bessel inequality gives

$$\sum_{k=1}^{\infty} (k\pi)^{2r} |\alpha_{i,k}^{(q)}|^2 \leq \left\| d_u^r h_i^{(q)} \right\|_{L_U^2([0,1])}^2 = \pi^{2r} C_{(2.38)}^{(r)}.$$

It readily follows

$$\sum_{k=K+1}^{\infty} |\alpha_{i,k}^{(q)}|^2 \leq \frac{1}{\pi^{2r}} \sum_{k=K+1}^{\infty} \frac{(k\pi)^{2r}}{(K+1)^{2r}} |\alpha_{i,k}^{(q)}|^2 \leq \frac{C_{(2.38)}^{(r)}}{(K+1)^{2r}}.$$

□

2.4.2 Multi-indices sets

In this work only arbitrary downward closed multi-indices sets are considered. As already stated in the introduction, the finite multi-indices set $\Gamma \subset \mathbb{N}^d$ is downward closed if

$$(\mathbf{k} \in \Gamma \text{ and } \mathbf{j} \leq \mathbf{k}) \Rightarrow \mathbf{j} \in \Gamma,$$

where $\mathbf{j} \leq \mathbf{k}$ means that $j_l \leq k_l$ for all $l = 1, \dots, d$. The most well-known of such multi-indices sets is the full or tensor product multi-indices set. For any $\mathbf{K} = (K_1, \dots, K_d)$ where $K_l \in \mathbb{N}$ for all $l = 1, \dots, d$, the full anisotropic multi-indices set and its complement are defined by

$$\begin{aligned} \Gamma_F(K_1, \dots, K_d) &:= \{\mathbf{k} \in \mathbb{N}^d : k_l \leq K_l, \quad \forall l = 1 \dots d\}, \\ \bar{\Gamma}_F(K_1, \dots, K_d) &:= \{\mathbf{k} \in \mathbb{N}^d : \mathbf{k} \notin \Gamma_F(K_1, \dots, K_d)\}. \end{aligned}$$

In high dimensions the full multi-indices set suffers from the curse of dimensionality. In order to effectively compute the projection of $y_i^{(g)}$ over a multidimensional set of basis functions we propose to consider the following two sparse sets of downward closed multi-indices sets, where DEG denotes a nonnegative integer:

- Total degree set defined by

$$\Gamma_T(\text{DEG}) := \left\{ \mathbf{k} \in \mathbb{N}^d : \sum_{l=1}^d k_l \leq \text{DEG} \right\},$$

where $\#\Gamma_T(\text{DEG}) = \binom{\text{DEG}+d}{d}$.

- Hyperbolic cross index set defined by

$$\Gamma_H(\text{DEG}) := \left\{ \mathbf{k} \in \mathbb{N}^d : \prod_{l=1}^d \max(k_l, 1) \leq \text{DEG} \right\}.$$

In order to compute the number of elements in $\Gamma_H(\text{DEG})$ we split this set in the following two disjoint sets:

- $\Gamma_1^d(\text{DEG}) := \left\{ \mathbf{k} \in \mathbb{N}^d, \forall l, k_l \neq 0 : \prod_{l=1}^d k_l \leq \text{DEG} \right\},$
- $\Gamma_2(\text{DEG}) := \left\{ \mathbf{k} \in \mathbb{N}^d, \exists l, k_l = 0 : \prod_{l=1}^d \max(k_l, 1) \leq \text{DEG} \right\}.$

On the one hand, let us compute $\#\Gamma_1^d(\text{DEG})$. Firstly we define

$$\Upsilon^d(\text{DEG}) := \left\{ \mathbf{k} \in \mathbb{N}^d, \forall l, k_l \neq 0 : \prod_{l=1}^d k_l = \text{DEG} \right\}.$$

Secondly we divide $\Gamma_1^d(\text{DEG})$ in the following disjoint sets,

$$\Gamma_1^d(\text{DEG}) = \bigcup_{g=2}^{\text{DEG}} \Upsilon^d(g) \cup \{\mathbf{1}\}.$$

For any integer $g \in \{2, \dots, \text{DEG}\}$ having the prime factorization

$$g = \prod_i p_i^{v_i},$$

$\#\Upsilon^d(g)$ is equal to the number of possible ways to write g as a product of exactly d of its factors,

$$\#\Upsilon^d(g) = \prod_i \binom{v_i + d - 1}{d - 1}.$$

Therefore,

$$\#\Gamma_1^d(\text{DEG}) = \sum_{g=2}^{\text{DEG}} \#\Upsilon^d(g) + 1.$$

On the other hand, let us compute $\#\Gamma_2(\text{DEG})$. Firstly we define $\Upsilon_c(\text{DEG})$ as the subset of $\Gamma_2(\text{DEG})$ containing the multi-indices of $\Gamma_2(\text{DEG})$ with exactly c of the d indexes equal to zero, $0 < c < d$. Note that $\#\Upsilon_c(\text{DEG}) = \binom{d}{c} \#\Gamma_1^{d-c}(\text{DEG})$. Therefore, since

$$\Gamma_2(\text{DEG}) = \bigcup_{c=1}^{d-1} \Upsilon_c(\text{DEG}) \cup \{\mathbf{0}\},$$

$$\#\Gamma_2(\text{DEG}) = \sum_{c=1}^{d-1} \binom{d}{c} \#\Gamma_1^{d-c}(\text{DEG}) + 1.$$

Finally, we conclude using that $\#\Gamma_H(\text{DEG}) = \#\Gamma_1^d(\text{DEG}) + \#\Gamma_2(\text{DEG})$.

Note that $\forall g \in \mathbb{N}, g \leq (\text{DEG} - d + 1)_+, \Gamma_H(g) \subset \Gamma_T(\text{DEG})$. See Figure 1 and Table 1.

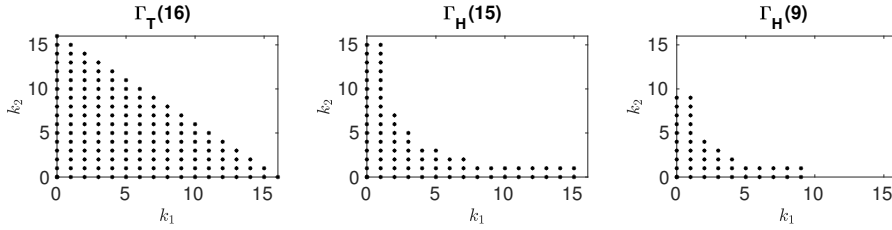


Fig. 1 For $d = 2$ examples of total and hyperbolic sets. Note that $\Gamma_H(9) \subset \Gamma_H(15) \subset \Gamma_T(16)$.

Table 1 Cardinalities of Γ_T and Γ_H .

$d = 3$			$d = 4$		
DEG	$\#\Gamma_T(\text{DEG})$	$\#\Gamma_H(\text{DEG} - 2)$	DEG	$\#\Gamma_T(\text{DEG})$	$\#\Gamma_H(\text{DEG} - 3)$
6	84	50	5	126	48
8	165	86	7	330	136
10	286	123	9	715	248
12	455	165	11	1365	368

2.4.3 Multidimensional case

In this section, we let ν be the Student density (2.10)-(2.11) in dimension d . In the first part of the section only y_i and its coordinate-wise derivatives are assumed to be smooth. In the second part, on top of that mixed derivatives of y_i will be supposed smooth. Let us start with the first case.

Lemma 5 *Assume that y_i is r times ($r \in \mathbb{N}$) coordinate-wise continuously differentiable on \mathbb{R}^d , with the existence of a polynomial growth degree $p \in \mathbb{N}$ such that*

$$\limsup_{|\mathbf{x}| \rightarrow +\infty} \frac{|\partial_{x_l}^n y_i(\mathbf{x})|}{|\mathbf{x}|^{p-n}} < +\infty, \quad \text{for } n = 0, \dots, r, \quad l = 1, \dots, d. \quad (2.42)$$

Assume the relation

$$p + r\mu < q \quad (2.43)$$

between r , p , the Student parameter $\mu > 0$ and the damping exponent $q \geq 0$. Then $h_i^{(q)}$ is r times continuously differentiable on $(0, 1)^d$, and its first r coordinate-wise derivatives vanish at $\partial[0, 1]^d$:

$$\lim_{\mathbf{u} \rightarrow \partial[0, 1]^d} \partial_{u_l}^n h_i^{(q)}(\mathbf{u}) = 0, \quad n = 0, \dots, r, \quad l = 1, \dots, d.$$

Proof As in dimension 1, the regularities of $y_i^{(q)}$ and $h_i^{(q)}$ are straightforward to obtain. The statement $\lim_{\mathbf{u} \rightarrow \partial[0, 1]^d} h_i^{(q)}(\mathbf{u}) = 0$ is clear since the limit coincides with that of $y_i^{(q)}(\mathbf{x})$ as $|\mathbf{x}| \rightarrow +\infty$, which equals to 0 in view of $q > p$. Therefore, from now on, we may consider $n \geq 1$ and let $l = 1, \dots, d$ be fixed.

In the following, to deal with comparison of functions as $|\mathbf{x}|$ tends to $+\infty$, we use the same notation C for any generic constants, that do not depend on \mathbf{x} , but that may depend on the order n, r, \dots of derivatives under consideration. We will keep the same C from line to line, in order to alleviate the computations, although its value changes.

We start by showing that for $n = 1, \dots, r$,

$$|\partial_{x_l}^n y_i^{(q)}(\mathbf{x})| \leq C(1 + |\mathbf{x}|^2)^{p/2 - n/2 - q/2}. \quad (2.44)$$

By applying the Leibniz rule to the product (2.8) (as in dimension 1, since we differentiate n times with respect to the same variable x_l) we get

$$\partial_{x_l}^n y_i^{(q)}(\mathbf{x}) = \sum_{m=0}^n \binom{n}{m} \partial_{x_l}^{n-m} y_i(\mathbf{x}) \partial_{x_l}^m \frac{1}{(1 + |\mathbf{x}|^2)^{q/2}}. \quad (2.45)$$

Our standing assumptions write

$$|\partial_{x_l}^{n-m} y_i(\mathbf{x})| \leq C(1 + |\mathbf{x}|)^{p - (n-m)}. \quad (2.46)$$

From (A.2) we know that

$$\left| \partial_{x_l}^m \frac{1}{(1 + |\mathbf{x}|^2)^{q/2}} \right| \leq C(1 + |\mathbf{x}|^2)^{-q/2 - m/2}.$$

By plugging the previous upper bound and (2.46) into the Leibniz formula (2.45), we get

$$\begin{aligned} \left| \partial_{x_l}^n y_i^{(q)}(\mathbf{x}) \right| &\leq C \max_{0 \leq m \leq n} (1 + |\mathbf{x}|)^{p-(n-m)} (1 + |\mathbf{x}|^2)^{-q/2-m/2} \\ &\leq C(1 + |\mathbf{x}|^2)^{p/2-q/2-n/2}, \end{aligned}$$

we are done with (2.44).

In order to compute the n -th derivative of $h_i^{(q)}(\mathbf{u}) = y_i^{(q)}(F_{\nu_1}^{-1}(u_1), \dots, F_{\nu_d}^{-1}(u_d))$ we again take advantage of the Faà di Bruno's formula (as in (2.34))

$$\partial_{u_l}^n h_i^{(q)}(\mathbf{u}) = \sum_{m=1}^n \frac{1}{m!} \partial_{x_l}^m y_i^{(q)}(\mathbf{x}) \Big|_{\mathbf{x}=(F_{\nu_1}^{-1}(u_1), \dots, F_{\nu_d}^{-1}(u_d))} \sum_{\mathbf{j} \in J_{m,n}} \frac{n!}{j_1! j_2! \dots j_m!} \prod_{i=1}^m d_{u_l}^{j_i} F_{\nu_i}^{-1}(u_i).$$

Combining the above with Lemma 11 and the estimates (A.7), we obtain

$$\begin{aligned} \left| \partial_{u_l}^n h_i^{(q)}(\mathbf{u}) \right| &\leq C \max_{1 \leq m \leq n, \mathbf{j} \in J_{m,n}} \left(1 + \sum_{i=1}^d |F_{\nu_i}^{-1}(u_i)|^2 \right)^{p/2-q/2-m/2} \prod_{i=1}^m \left| d_{u_l}^{j_i} F_{\nu_i}^{-1}(u_i) \right| \\ &\leq C \max_{1 \leq m \leq n, \mathbf{j} \in J_{m,n}} \left(1 + \sum_{i=1}^d \frac{1}{u_i^{1/\mu} (1-u_i)^{1/\mu}} \right)^{p-q-m} \left(\frac{1}{u_l^{1/\mu} (1-u_l)^{1/\mu}} \right)^{m+\mu \sum_{i=1}^m j_i} \\ &\leq C \max_{1 \leq m \leq n} \left(1 + \sum_{i=1}^d \frac{1}{u_i^{1/\mu} (1-u_i)^{1/\mu}} \right)^{p-q-m} \left(1 + \sum_{i=1}^d \frac{1}{u_i^{1/\mu} (1-u_i)^{1/\mu}} \right)^{m+n\mu} \\ &\leq C \left(1 + \sum_{i=1}^d \frac{1}{u_i^{1/\mu} (1-u_i)^{1/\mu}} \right)^{p-q+n\mu}. \end{aligned} \quad (2.47)$$

Since $p + n\mu < q$, we get that $\lim_{\mathbf{u} \rightarrow \partial[0,1]^d} \partial_{u_l}^n h_i^{(q)}(\mathbf{u}) = 0$. \square

Lemma 6 Assume the hypotheses of Lemma 5 and replace the condition (2.43) by the weaker condition

$$p + \left(r - \frac{1}{2} \right) \mu < q. \quad (2.48)$$

Then the r first coordinate-wise derivatives of $h_i^{(q)}$ are square integrable on $(0,1)^d$:

$$\left\| \partial_{u_l}^n h_i^{(q)}(\mathbf{u}) \right\|_{L^2_{\nu}([0,1]^d)}^2 < +\infty, \quad n = 0, \dots, r, \quad l = 1, \dots, d.$$

Proof In view of the bound (2.47), it is sufficient to check the integrability on $(0,1)^d$ of

$$l(\mathbf{u}) := \left(1 + \sum_{i=1}^d \frac{1}{u_i^{1/\mu} (1-u_i)^{1/\mu}} \right)^{2(p-q+r\mu)}.$$

Observe that the function is symmetric with respect to each hyperplan $\{\mathbf{u} \in (0,1)^d : u_l = \frac{1}{2}\}$, $l = 1, \dots, d$. Therefore, only the integrability on $(0,1/2)^d$ has to be established, which is equivalent to prove

$$\int_{(0,1/2)^d} \left(1 + \sum_{i=1}^d \frac{1}{u_i^{1/\mu}} \right)^{2(p-q+r\mu)} d\mathbf{u} < +\infty$$

Use the change of variables $z_i = u_i^{-1/2\mu}$, then the above integral writes (up to a non-zero factor)

$$\mathcal{I} := \int_{(2^{1/(2\mu)}, +\infty)^d} \left(1 + \sum_{i=1}^d z_i^2\right)^{\mu-\varepsilon} \left(\prod_{i=1}^d z_i^{-2\mu-1}\right) d\mathbf{z},$$

where $\varepsilon > 0$ is given by $2(p - q + r\mu) = \mu - \varepsilon$. To show that $\mathcal{I} < +\infty$, we may decrease $\varepsilon \downarrow 0$ as much as needed to get $\mu - \varepsilon > 0$. Once done, using the inequality

$$(x_1 + x_2)^\alpha \leq 2^{(\alpha-1)_+} (x_1^\alpha + x_2^\alpha),$$

available for any $x_1 \geq 0, x_2 \geq 0, \alpha > 0$, it yields (by direct induction) the existence of a constant $C_{\alpha,d}$ such that

$$\left(1 + \sum_{i=1}^d x_i\right)^\alpha \leq C_{\alpha,d} \left(1 + \sum_{i=1}^d x_i^\alpha\right)$$

for any $x_1 \geq 0, \dots, x_d \geq 0$. Consequently,

$$\mathcal{I} \leq C_{\mu-\varepsilon,d} \int_{(2^{1/(2\mu)}, +\infty)^d} \left(1 + \sum_{i=1}^d z_i^{2(\mu-\varepsilon)}\right) \left(\prod_{i=1}^d z_i^{-2\mu-1}\right) d\mathbf{z} < +\infty.$$

We are done. \square

Proposition 4 *Assume that y_i is r times ($r \geq 1$) coordinate-wise continuously differentiable on \mathbb{R}^d with the existence of a polynomial growth degree $p \in \mathbb{N}$ such that (2.42) holds. Assume the relation (2.48) holds. Then, the coefficients $(\alpha_{i,\mathbf{k}}^{(q)}, \mathbf{k} \in \mathbb{N}^d)$ defined in (2.29) satisfy the tail inequality*

$$\sum_{\mathbf{k} \in \bar{\Gamma}_F(K_1, \dots, K_d)} |\alpha_{i,\mathbf{k}}^{(q)}|^2 \leq \sum_{l=1}^d \frac{C_{(2.49)}^{(l,r)}}{(K_l + 1)^{2r}} \quad (2.49)$$

where $C_{(2.49)}^{(l,r)} := \frac{1}{\pi^{2r}} \left\| \partial_{u_l}^r h_i^{(q)} \right\|_{L_U^2([0,1]^d)}^2 < +\infty$.

Proof By definition of $\Gamma_F(K_1, \dots, K_d)$, we have the union bound

$$\{\mathbf{k} \in \bar{\Gamma}_F(K_1, \dots, K_d)\} = \bigcup_{l=1}^d \{\mathbf{k} \in \mathbb{N}^d : k_l > K_l\},$$

therefore the advertised inequality (2.49) directly follows from

$$\sum_{\mathbf{k} \in \mathbb{N}^d : k_l > K_l} |\alpha_{i,\mathbf{k}}^{(q)}|^2 \leq \frac{C_{(2.49)}^{(l,r)}}{(K_l + 1)^{2r}}, \quad l = 1, \dots, d. \quad (2.50)$$

We only prove the case $l = 1$, the arguments apply similarly for other l .

Under (2.48), we can invoke Lemma 5 (with $r - 1$ derivatives) and Lemma 6 (with r derivatives), to get

$$\begin{aligned} \lim_{\mathbf{u} \rightarrow \partial[0,1]^d} \partial_{u_1}^n h_i^{(q)}(\mathbf{u}) &= 0, \quad \text{for } n = 0, \dots, r-1, \\ \left\| \partial_{u_1}^n h_i^{(q)} \right\|_{L_U^2([0,1]^d)}^2 &< +\infty, \quad \text{for } n = 0, \dots, r. \end{aligned} \quad (2.51)$$

In view of (2.29)-(2.30), the Fourier coefficients of the decomposition of $h_i^{(q)}$ with respect to the orthonormal basis functions $\{C_{\mathbf{k}}, \mathbf{k} \in \mathbb{N}^d\}$ in $L_U^2([0,1]^d)$ are given by $(\alpha_{i,\mathbf{k}}^{(q)}, \mathbf{k} \in \mathbb{N}^d)$ and the $C_{\mathbf{k}}$ -cosine series of $h_i^{(q)}$ writes

$$h_i^{(q)}(u) = \sum_{\mathbf{k} \in \mathbb{N}^d} \alpha_{i,\mathbf{k}}^{(q)} C_{\mathbf{k}}(u),$$

where

$$\begin{aligned} \alpha_{i,\mathbf{k}}^{(q)} &= \beta_{\mathbf{k}} \int_{[0,1]^d} h_i^{(q)}(\mathbf{u}) \cos(k_1 \pi u_1) \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d) d\mathbf{u} \\ &= \beta_{\mathbf{k}} \operatorname{Re} \left(\int_{[0,1]^d} h_i^{(q)}(\mathbf{u}) e^{i k_1 \pi u_1} \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d) d\mathbf{u} \right), \\ \beta_{\mathbf{k}} &:= \prod_{l=1}^d (\mathbf{1}_{k_l=0} + \mathbf{1}_{k_l>0} \sqrt{2}). \end{aligned}$$

Applying integration by parts in u_1 ($h_i^{(q)}$ is uniformly bounded on $(0,1)^d$, continuously differentiable and the derivative with respect to u_1 is integrable) and using that $\lim_{u_1 \rightarrow 0^+, 1^-} h_i^{(q)}(\mathbf{u}) = 0$, we get for $k_1 > 0$

$$\begin{aligned} \alpha_{i,\mathbf{k}}^{(q)} &= \operatorname{Re} \left(\frac{\beta_{\mathbf{k}}}{i k_1 \pi} \left[\int_{[0,1]^{d-1}} h_i^{(q)}(\mathbf{u}) e^{i k_1 \pi u_1} \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d) du_2 \dots du_d \right]_{u_1=0^+}^{u_1=1^-} \right. \\ &\quad \left. - \frac{\beta_{\mathbf{k}}}{i k_1 \pi} \int_{[0,1]^d} \partial_{u_1} h_i^{(q)}(\mathbf{u}) e^{i k_1 \pi u_1} \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d) d\mathbf{u} \right) \\ &= \operatorname{Re} \left(\frac{\beta_{\mathbf{k}}}{-i k_1 \pi} \int_{[0,1]^d} \partial_{u_1} h_i^{(q)}(\mathbf{u}) e^{i k_1 \pi u_1} \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d) d\mathbf{u} \right). \end{aligned}$$

Thanks to (2.51), we can repeat $r - 1$ times the integration by parts as above, and we obtain

$$\begin{aligned} \alpha_{i,\mathbf{k}}^{(q)} &= \operatorname{Re} \left(\frac{\beta_{\mathbf{k}}}{(-i k_1 \pi)^r} \int_{[0,1]^d} \partial_{u_1}^r h_i^{(q)}(\mathbf{u}) e^{i k_1 \pi u_1} \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d) d\mathbf{u} \right) \\ &= \frac{1}{(k_1 \pi)^r} \int_{[0,1]^d} \partial_{u_1}^r h_i^{(q)}(\mathbf{u}) \operatorname{Re} \left(\beta_{\mathbf{k}} i^r e^{i k_1 \pi u_1} \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d) \right) d\mathbf{u}. \end{aligned} \quad (2.52)$$

Observe that the sets

$$\{\mathbf{u} \mapsto \beta_{\mathbf{k}} \cos(k_1 \pi u_1) \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d), \mathbf{k} \in \mathbb{N}^d\},$$

$$\{\mathbf{u} \mapsto \beta_{\mathbf{k}} \sin(k_1 \pi u_1) \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d), \mathbf{k} \in \mathbb{N}^d\},$$

are made of orthonormal functions in $L^2_U([0, 1]^d)$. Consequently,

$$\left\{ \operatorname{Re} \left(\beta_{\mathbf{k}} e^{i k_1 \pi u_1} \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d) \right), \mathbf{k} \in \mathbb{N}^d \right\}$$

is (for any value of r) a set of orthonormal functions and the equality (2.52) means that $\{(k_1 \pi)^r \alpha_{i, \mathbf{k}}^{(q)} : \mathbf{k} \in \mathbb{N}^d\}$ are the projection coefficients of $\pm \partial_{u_1}^r h_i^{(q)}$ on these orthonormal functions. As such, the Bessel inequality gives

$$\sum_{\mathbf{k} \in \mathbb{N}^d} (k_1 \pi)^{2r} |\alpha_{i, \mathbf{k}}^{(q)}|^2 \leq \left\| \partial_{u_1}^r h_i^{(q)} \right\|_{L^2_U([0, 1]^d)}^2 = \pi^{2r} C_{(2.49)}^{(1, r)}.$$

It readily follows

$$\sum_{\mathbf{k} \in \mathbb{N}^d: k_1 > K_1} |\alpha_{i, \mathbf{k}}^{(q)}|^2 \leq \frac{1}{\pi^{2r}} \sum_{\mathbf{k} \in \mathbb{N}^d: k_1 > K_1} \frac{(k_1 \pi)^{2r}}{(K_1 + 1)^{2r}} |\alpha_{i, \mathbf{k}}^{(q)}|^2 \leq \frac{C_{(2.49)}^{(1, r)}}{(K_1 + 1)^{2r}}.$$

The proof of (2.50) is now finished and Proposition 4 is proved. \square

Now we consider that not only y_i and its coordinate-wise derivatives are smooth but also its mixed derivatives.

Lemma 7 *Let $\Lambda \subset \mathbb{N}^d$ be an arbitrary downward closed multi-index set and set $r = \max_{\mathbf{r} \in \Lambda} (\bar{\mathbf{r}})$. Assume that y_i is Λ -smooth with the existence of a polynomial growth degree $p \in \mathbb{N}$ such that*

$$\limsup_{|\mathbf{x}| \rightarrow +\infty} \frac{|\partial_{\mathbf{x}}^{\mathbf{n}} y_i(\mathbf{x})|}{|\mathbf{x}|^{p - \bar{\mathbf{n}}}} < +\infty, \quad \text{for } \mathbf{n} \in \Lambda. \quad (2.53)$$

Assume the relation

$$p + r\mu < q \quad (2.54)$$

between r , p , the Student parameter $\mu > 0$ and the damping exponent $q \geq 0$. Then $h_i^{(q)}$ is Λ -smooth on $(0, 1)^d$ and its derivatives vanish at $\partial[0, 1]^d$:

$$\lim_{\mathbf{u} \rightarrow \partial[0, 1]^d} \partial_{\mathbf{u}}^{\mathbf{n}} h_i^{(q)}(\mathbf{u}) = 0, \quad \mathbf{n} \in \Lambda.$$

Proof We start by showing that for $\bar{\mathbf{n}} > 0$, $\mathbf{n} \in \Lambda$,

$$|\partial_{\mathbf{x}}^{\mathbf{n}} y_i^{(q)}(\mathbf{x})| \leq C(1 + |\mathbf{x}|^2)^{p/2 - \bar{\mathbf{n}}/2 - q/2}. \quad (2.55)$$

By applying the multivariate Leibniz rule to the product (2.8) we get

$$\partial_{\mathbf{x}}^{\mathbf{n}} y_i^{(q)}(\mathbf{x}) = \sum_{\{\mathbf{m} \in \mathbb{N}^d: \mathbf{m} \leq \mathbf{n}\}} C_{\mathbf{n}, \mathbf{m}} \partial_{\mathbf{x}}^{\mathbf{n} - \mathbf{m}} y_i(\mathbf{x}) \partial_{\mathbf{x}}^{\mathbf{m}} \frac{1}{(1 + |\mathbf{x}|^2)^{q/2}}. \quad (2.56)$$

Our standing assumptions write

$$|\partial_{\mathbf{x}}^{\mathbf{n} - \mathbf{m}} y_i(\mathbf{x})| \leq C(1 + |\mathbf{x}|)^{p - \overline{(\mathbf{n} - \mathbf{m})}} \leq C(1 + |\mathbf{x}|^2)^{p/2 - \bar{\mathbf{n}}/2 + \bar{\mathbf{m}}/2}. \quad (2.57)$$

From (A.4) we know that

$$\left| \partial_x^{\mathbf{m}} \frac{1}{(1 + |\mathbf{x}|^2)^{q/2}} \right| \leq C(1 + |\mathbf{x}|^2)^{-q/2 - \bar{\mathbf{m}}/2}.$$

By plugging the previous upper bound and (2.57) into the Leibniz formula (2.56), we get

$$\begin{aligned} \left| \partial_x^{\mathbf{n}} y_i^{(q)}(\mathbf{x}) \right| &\leq C \max_{\mathbf{0} \leq \mathbf{m} \leq \mathbf{n}} (1 + |\mathbf{x}|^2)^{p/2 - \bar{\mathbf{n}}/2 + \bar{\mathbf{m}}/2} (1 + |\mathbf{x}|^2)^{-q/2 - \bar{\mathbf{m}}/2} \\ &\leq C(1 + |\mathbf{x}|^2)^{p/2 - q/2 - \bar{\mathbf{n}}/2}, \end{aligned}$$

we are done with (2.55).

Let $[\ast, \dots, \ast]$ denotes a list. In order to compute the \mathbf{n} ($\bar{\mathbf{n}} > 0$, $\mathbf{n} \in \Lambda$) derivative of $h_i^{(q)}(\mathbf{u}) = y_i^{(q)}(F_{\nu_1}^{-1}(u_1), \dots, F_{\nu_d}^{-1}(u_d))$ we take advantage of the multivariate Faà di Bruno's formula derived from [15, Theorem 2.1]:

$$\begin{aligned} \partial_{\mathbf{u}}^{\mathbf{n}} h_i^{(q)}(\mathbf{u}) &= \sum_{\mathbf{m} \in M_{\mathbf{n}}} \partial_x^{\mathbf{m}} y_i^{(q)}(\mathbf{x}) \Big|_{\mathbf{x}=(F_{\nu_1}^{-1}(u_1), \dots, F_{\nu_d}^{-1}(u_d))} \sum_{J \in A_{\mathbf{m}, \mathbf{n}}} C_{\mathbf{m}, J} \prod_{i=1}^d \prod_{k=1}^{m_i} d_{u_i}^{j_{ik}} F_{\nu_i}^{-1}(u_i), \\ M_{\mathbf{n}} &= \{\mathbf{m} = (m_1, \dots, m_d) \in \mathbb{N}_+^d : \forall i = 1, \dots, d, 0 \leq m_i \leq n_i, \mathbf{m} \neq \mathbf{0}\}, \\ A_{\mathbf{m}, \mathbf{n}} &= \{J = [\mathbf{j}_1 \in \mathbb{N}_+^{m_1}, \dots, \mathbf{j}_d \in \mathbb{N}_+^{m_d}] : \bar{\mathbf{j}}_i = n_i \forall i = 1, \dots, d\}. \end{aligned}$$

Combining the above with Lemma 11 and the estimates (A.7), we obtain

$$\begin{aligned} \left| \partial_{\mathbf{u}}^{\mathbf{n}} h_i^{(q)}(\mathbf{u}) \right| &\leq C \max_{\mathbf{m} \in M_{\mathbf{n}}, J \in A_{\mathbf{m}, \mathbf{n}}} \left(1 + \sum_{i=1}^d |F_{\nu_i}^{-1}(u_i)|^2 \right)^{p/2 - q/2 - \bar{\mathbf{m}}/2} \prod_{i=1}^d \prod_{k=1}^{m_i} \left| d_{u_i}^{j_{ik}} F_{\nu_i}^{-1}(u_i) \right| \\ &\leq C \max_{\mathbf{m} \in M_{\mathbf{n}}, J \in A_{\mathbf{m}, \mathbf{n}}} \left(1 + \sum_{i=1}^d \frac{1}{u_i^{1/\mu} (1 - u_i)^{1/\mu}} \right)^{p - q - \bar{\mathbf{m}}} \prod_{i=1}^d \left(\frac{1}{u_i^{1/\mu} (1 - u_i)^{1/\mu}} \right)^{m_i + \mu \sum_{i=1}^{m_i} j_{i1}} \\ &\leq C \max_{\mathbf{m} \neq \mathbf{0}, \mathbf{m} \leq \mathbf{n}} \left(1 + \sum_{i=1}^d \frac{1}{u_i^{1/\mu} (1 - u_i)^{1/\mu}} \right)^{p - q - \bar{\mathbf{m}}} \prod_{i=1}^d \left(\frac{1}{u_i^{1/\mu} (1 - u_i)^{1/\mu}} \right)^{m_i + \mu n_i} \\ &\leq C \max_{\mathbf{m} \neq \mathbf{0}, \mathbf{m} \leq \mathbf{n}} \left(1 + \sum_{i=1}^d \frac{1}{u_i^{1/\mu} (1 - u_i)^{1/\mu}} \right)^{p - q - \bar{\mathbf{m}}} \prod_{i=1}^d \left(1 + \sum_{j=1}^d \frac{1}{u_j^{1/\mu} (1 - u_j)^{1/\mu}} \right)^{m_i + \mu n_i} \\ &\leq C \max_{\mathbf{m} \neq \mathbf{0}, \mathbf{m} \leq \mathbf{n}} \left(1 + \sum_{i=1}^d \frac{1}{u_i^{1/\mu} (1 - u_i)^{1/\mu}} \right)^{p - q - \bar{\mathbf{m}} + \bar{\mathbf{m}} + \mu \bar{\mathbf{n}}} \tag{2.58} \end{aligned}$$

$$\leq C \left(1 + \sum_{i=1}^d \frac{1}{u_i^{1/\mu} (1 - u_i)^{1/\mu}} \right)^{p - q + \bar{\mathbf{n}} \mu}. \tag{2.59}$$

Since $p + \bar{\mathbf{n}} \mu < q$, we get that $\lim_{\mathbf{u} \rightarrow \partial[0,1]^d} \partial_{\mathbf{u}}^{\mathbf{n}} h_i^{(q)}(\mathbf{u}) = 0$. \square

Lemma 8 Assume the hypotheses of Lemma 7 and replace the condition (2.54) by the weaker condition

$$p + \left(r - \frac{1}{2} \right) \mu < q. \tag{2.60}$$

Then the Λ derivatives of $h_i^{(q)}$ are square integrable on $(0, 1)^d$:

$$\left\| \partial_{\mathbf{u}}^{\mathbf{n}} h_i^{(q)}(\mathbf{u}) \right\|_{L_V^2([0,1]^d)}^2 < +\infty, \quad \mathbf{n} \in \Lambda.$$

Proof The proof follows readily from the proof of Lemma 6 by replacing r by $r = \max_{\mathbf{r} \in \Lambda}(\bar{\mathbf{r}})$.

Proposition 5 Let $\Lambda(r) = \{0, \dots, r\}^d$. Assume that y_i is Λ -smooth with the existence of a polynomial growth degree $p \in \mathbb{N}$ such that (2.53) holds. Assume the relation (2.60) holds, i.e.

$$p + (d r - 1/2)\mu < q.$$

Then, the coefficients $(\alpha_{i,\mathbf{k}}^{(q)}, \mathbf{k} \in \mathbb{N}^d)$ defined in (2.29) satisfy the tail inequality

$$\sum_{\mathbf{k} \in \bar{\Gamma}_H(\text{DEG})} |\alpha_{i,\mathbf{k}}^{(q)}|^2 \leq \frac{C_{(2.61)}}{(\text{DEG} + 1)^{2r}}, \quad (2.61)$$

where $C_{(2.61)} := \frac{1}{\pi^{2dr}} \left\| \partial_{\mathbf{u}}^{(r, \dots, r)} h_i^{(q)} \right\|_{L_V^2([0,1]^d)}^2 < +\infty$.

Proof Under (2.60), we can invoke Lemma 7 (with $\#\Lambda(r) - 1$ derivatives) (condition $p + (dr - 1)\mu < q$) and Lemma 8 (with $\#\Lambda(r)$ derivatives) (condition $p + (dr - 1/2)\mu < q$), to get

$$\begin{aligned} \lim_{\mathbf{u} \rightarrow \partial[0,1]^d} \partial_{\mathbf{u}}^{\mathbf{n}} h_i^{(q)}(\mathbf{u}) &= 0, \quad \mathbf{n} \in \Lambda(r), \mathbf{n} \neq (r, \dots, r), \\ \left\| \partial_{\mathbf{u}}^{\mathbf{n}} h_i^{(q)}(\mathbf{u}) \right\|_{L_V^2([0,1]^d)}^2 &< +\infty, \quad \mathbf{n} \in \Lambda(r). \end{aligned} \quad (2.62)$$

By doing integration by parts on the Fourier coefficients of $h_i^{(q)}$ as we have already done in Proposition 4 we readily obtain

$$\begin{aligned} \alpha_{i,\mathbf{k}}^{(q)} &= \int_{[0,1]^d} h_i^{(q)}(\mathbf{u}) \beta_{\mathbf{k}} \cos(k_1 \pi u_1) \cdots \cos(k_d \pi u_d) d\mathbf{u}, \\ &= \frac{1}{\pi^{dr} (k_1 \cdots k_d)^r} \int_{[0,1]^d} \partial_{\mathbf{u}}^{(r, \dots, r)} h_i^{(q)}(\mathbf{u}) \beta_{\mathbf{k}} \cos^{f,r}(k_1 \pi u_1) \cdots \cos^{f,r}(k_d \pi u_d) d\mathbf{u}, \end{aligned} \quad (2.63)$$

where $\cos^{f,r}(\cdot)$ is the r -th antiderivative of $\cos(\cdot)$ such that

$$\begin{aligned} \cos^{f,4m}(\cdot) &= \cos(\cdot), & \cos^{f,4m+1}(\cdot) &= \sin(\cdot), \\ \cos^{f,4m+2}(\cdot) &= -\cos(\cdot), & \cos^{f,4m+3}(\cdot) &= -\sin(\cdot), \quad m \in \mathbb{N}. \end{aligned}$$

Equation (2.63) tells us that $h_i^{(q)}$ is a function in a Korobov space [44, Definition 6.5.6, p. 225]. Observe that the sets

$$\begin{aligned} \{\mathbf{u} \mapsto \beta_{\mathbf{k}} \cos(k_1 \pi u_1) \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d), \mathbf{k} \in \mathbb{N}^d\}, \\ \{\mathbf{u} \mapsto \beta_{\mathbf{k}} \sin(k_1 \pi u_1) \cos(k_2 \pi u_2) \cdots \cos(k_d \pi u_d), \mathbf{k} \in \mathbb{N}^d\}, \end{aligned}$$

$$\begin{aligned} & \vdots \\ & \{\mathbf{u} \mapsto \beta_{\mathbf{k}} \sin(k_1 \pi u_1) \sin(k_2 \pi u_2) \cdots \sin(k_d \pi u_d), \mathbf{k} \in \mathbb{N}^d\}, \end{aligned}$$

are made of orthonormal functions in $L^2_U([0, 1]^d)$. Consequently,

$$\left\{ \beta_{\mathbf{k}} \cos^{f,r}(k_1 \pi u_1) \cdots \cos^{f,r}(k_d \pi u_d), \mathbf{k} \in \mathbb{N}^d \right\}$$

is (for any value of r) a set of orthonormal functions and the equality (2.63) means that $\{(k_1 \pi)^r \cdots (k_d \pi)^r \alpha_{i,\mathbf{k}}^{(q)} : \mathbf{k} \in \mathbb{N}^d\}$ are the projection coefficients of $\partial_u^{(r,\dots,r)} h_i^{(q)}$ on these orthonormal functions. As such, the Bessel inequality gives

$$\sum_{\mathbf{k} \in \mathbb{N}^d} \pi^{2dr} (k_1 \cdots k_d)^{2r} |\alpha_{i,\mathbf{k}}^{(q)}|^2 \leq \left\| \partial_u^{(r,\dots,r)} h_i^{(q)} \right\|_{L^2_U([0,1]^d)}^2 = \pi^{2dr} C_{(2.61)}.$$

It readily follows

$$\sum_{\substack{\mathbf{k} \in \mathbb{N}^d, \\ \mathbf{k} \notin \Gamma_H(\text{DEG})}} |\alpha_{i,\mathbf{k}}^{(q)}|^2 \leq \frac{1}{\pi^{2dr}} \sum_{\substack{\mathbf{k} \in \mathbb{N}^d, \\ \mathbf{k} \notin \Gamma_H(\text{DEG})}} \frac{\pi^{2dr} (k_1 \cdots k_d)^{2r}}{(\text{DEG} + 1)^{2r}} |\alpha_{i,\mathbf{k}}^{(q)}|^2 \leq \frac{C_{(2.61)}}{(\text{DEG} + 1)^{2r}}.$$

□

Equation (2.63) means that the most significant coefficients $\alpha_{i,\mathbf{k}}^{(q)}$ are the ones for which the product $k_1 \cdots k_d$ is as small as possible. Under this setting we prefer to consider the hyperbolic cross index set (Proposition 5) rather than the total degree set (Proposition 4). In fact, although the hyperbolic truncation error with $\Gamma_H((\text{DEG} - d + 1)_+)$ will be greater than the total truncation error with $\Gamma_T(\text{DEG})$, if $h_i^{(q)}$ belongs to a Korobov space, the hyperbolic truncation error will be near to the total truncation error (here we are assuming that DEG is big enough to take into account all the significant Fourier coefficients). Besides, having in mind that $\#\Gamma_H(\text{DEG} - d + 1) \ll \#\Gamma_T(\text{DEG})$ (specially in high dimensions), the statistical error, driven by $\frac{2^d \#\Gamma}{M}$, will be smaller for the hyperbolic multi-indices set, considering the same number of simulations M . Last but not least, the algorithm executed with the hyperbolic multi-indices set will be less time consuming than with the total degree set. This behavior is shown in the numerical experiments of Section 5.

3 Proofs related to error analysis

3.1 Proof of Theorem 1

First observe that we can restrict in the proof of (2.13) to the case of bounded measurable functions γ : indeed, for $\gamma \in L^2_\nu(\mathbb{R}^d)$ (possibly unbounded), apply the inequality (2.13) to the bounded function $\mathbf{x} \mapsto \mathcal{T}_L(\gamma(\mathbf{x}))$ where $L > 0$ to get (for any $0 \leq i \leq j \leq N$)

$$\mathbb{E} \left[\frac{|\mathcal{T}_L(\gamma(X_j^i))|^2 (1 + |X_j^i|^2)^q}{(1 + |X_i^i|^2)^q} \right] \leq c_\nu \mathbb{E} \left[|\mathcal{T}_L(\gamma(X_i^i))|^2 \right], \quad (3.1)$$

and take the limit $L \rightarrow +\infty$. On the one hand, the dominated convergence theorem (because $\gamma \in L_\nu^2$) ensures that the r.h.s. of (3.1) converges to $c_\nu \mathbb{E} [|\gamma(X_i^i)|^2]$. On the other hand, by Fatou's lemma, the liminf of the l.h.s. of (3.1) is bounded from

below by $\mathbb{E} \left[\frac{|\gamma(X_j^i)|^2 (1 + |X_j^i|^2)^q}{(1 + |X_i^i|^2)^q} \right]$, which all in all, gives the result (2.13) for the square integrable function γ .

Now we establish (2.13) for bounded functions γ . We follow and adapt the arguments of [37, Proposition 3.1 and Proposition 3.3 item (b)] which paves the way for $q = 0$ (although the distribution ν is different). Under the current assumptions on b and σ , the Euler scheme X is a non-homogeneous Markov chain which transition kernel has a probability density w.r.t. the Lebesgue measure on \mathbb{R}^d . This density is denoted by $p_{i,j}(\mathbf{x}, \mathbf{y})$ where $i < j$ stands for the time indices and \mathbf{x} (resp. \mathbf{y}) stands for the initial (resp. final) space point. This probability density satisfies to a Gaussian upper bound of Aronson type (see [49, Theorem 2.1]): there is a finite constant $C_{(3.2)} > 0$ (depending on d, q, T, b, σ but not on N) such that

$$p_{i,j}(\mathbf{x}, \mathbf{y}) \leq C_{(3.2)} \frac{\exp\left(-\frac{|\mathbf{y}-\mathbf{x}|^2}{2C_{(3.2)}(t_j-t_i)}\right)}{(2\pi(t_j-t_i))^{d/2}}, \quad (3.2)$$

for any $0 \leq i < j \leq N$ and any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. Assume that ν is such that, for any $\Lambda \geq 0$, there is a finite constant $C_\nu(\Lambda) > 0$ (depending on μ and q) such that, for any $\lambda \in [0, \Lambda]$ and any $\mathbf{y} \in \mathbb{R}^d$, we have

$$\int_{\mathbb{R}^d} \nu(\mathbf{y} + \mathbf{z}\sqrt{\lambda}) \frac{(1 + |\mathbf{y}|^2)^q}{(1 + |\mathbf{y} + \mathbf{z}\sqrt{\lambda}|^2)^q} \frac{e^{-|\mathbf{z}|^2/2}}{(2\pi)^{d/2}} d\mathbf{z} \leq C_\nu(\Lambda)\nu(\mathbf{y}). \quad (3.3)$$

Then we claim that the inequality (2.13) holds, still considering bounded γ (which clearly ensures that both sides of (2.13) are finite). Indeed, combining (3.2) and (3.3), we get

$$\begin{aligned} & \mathbb{E} \left[\frac{|\gamma(X_j^i)|^2 (1 + |X_j^i|^2)^q}{(1 + |X_i^i|^2)^q} \right] \\ &= \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \nu(\mathbf{x}) p_{i,j}(\mathbf{x}, \mathbf{y}) \frac{(1 + |\mathbf{y}|^2)^q}{(1 + |\mathbf{x}|^2)^q} \gamma^2(\mathbf{y}) d\mathbf{x} d\mathbf{y} \end{aligned}$$

(use (3.2) and the change of variable $(\mathbf{x}, \mathbf{y}) \rightarrow (\mathbf{y} + \mathbf{z}\sqrt{C_{(3.2)}(t_j-t_i)}, \mathbf{y})$)

$$\begin{aligned} & \leq C_{(3.2)} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \nu(\mathbf{x}) \frac{e^{-\frac{|\mathbf{y}-\mathbf{x}|^2}{2C_{(3.2)}(t_j-t_i)}}}{(2\pi(t_j-t_i))^{d/2}} \frac{(1 + |\mathbf{y}|^2)^q}{(1 + |\mathbf{x}|^2)^q} \gamma^2(\mathbf{y}) d\mathbf{x} d\mathbf{y} \\ &= C_{(3.2)}^{1+d/2} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \nu\left(\mathbf{y} + \mathbf{z}\sqrt{C_{(3.2)}(t_j-t_i)}\right) \\ & \quad \times \frac{(1 + |\mathbf{y}|^2)^q}{(1 + |\mathbf{y} + \mathbf{z}\sqrt{C_{(3.2)}(t_j-t_i)}|^2)^q} \frac{e^{-|\mathbf{z}|^2/2}}{(2\pi)^{d/2}} \gamma^2(\mathbf{y}) d\mathbf{z} d\mathbf{y} \\ & \leq C_{(3.2)}^{1+d/2} C_\nu(C_{(3.2)}T) \int_{\mathbb{R}^d} \gamma^2(\mathbf{y}) \nu(\mathbf{y}) d\mathbf{y} \end{aligned}$$

which gives the announced result (2.13) with $c_\nu := C_{(3.2)}^{1+d/2} C_\nu(C_{(3.2)}T)$.

Therefore, it remains to establish (3.3); since

$$(1 + z^2)^{1/2} \leq (1 + z) \leq \sqrt{2}(1 + z^2)^{1/2}, \quad \forall z \geq 0,$$

for getting (3.3) it is enough to prove (for any $\mathbf{y} = (y_1, \dots, y_d)$)

$$\begin{aligned} & \int_{\mathbb{R}^d} \left(\prod_{l=1}^d \frac{1}{(1 + |y_l + z_l \sqrt{\lambda}|)^{\mu+1}} \right) \frac{(1 + |\mathbf{y}|^2)^q}{(1 + |\mathbf{y} + \mathbf{z} \sqrt{\lambda}|^2)^q} \frac{e^{-|\mathbf{z}|^2/2}}{(2\pi)^{d/2}} d\mathbf{z} \\ & \leq \frac{C_\nu(\Lambda)}{2^{(\mu+1)d/2}} \prod_{l=1}^d \frac{1}{(1 + |y_l|)^{\mu+1}}. \end{aligned} \quad (3.4)$$

Set $\mathcal{A}_\lambda := \{\mathbf{z} \in \mathbb{R}^d : |\mathbf{z}| \sqrt{\lambda} \leq |\mathbf{y}|/2\}$. Denote by \mathcal{I}_1 (resp. \mathcal{I}_2) the above integral restricted to \mathcal{A}_λ (resp. \mathcal{A}_λ^c). On \mathcal{A}_λ , use that $1 + |\mathbf{y} + \mathbf{z} \sqrt{\lambda}|^2 \geq 1 + \frac{1}{2}|\mathbf{y}|^2 - |\mathbf{z} \sqrt{\lambda}|^2 \geq \frac{1}{4} + \frac{1}{4}|\mathbf{y}|^2$ to obtain

$$\begin{aligned} \mathcal{I}_1 & \leq 4^q \int_{\mathcal{A}_\lambda} \left(\prod_{l=1}^d \frac{1}{(1 + |y_l + z_l \sqrt{\lambda}|)^{\mu+1}} \right) \frac{e^{-|\mathbf{z}|^2/2}}{(2\pi)^{d/2}} d\mathbf{z} \\ & \leq 4^q \prod_{l=1}^d \int_{\mathbb{R}} \frac{1}{(1 + |y_l + z_l \sqrt{\lambda}|)^{\mu+1}} \frac{e^{-|z_l|^2/2}}{\sqrt{2\pi}} dz_l. \end{aligned}$$

Split each integral according to the set $\mathcal{A}_{\lambda,l} := \{z_l \in \mathbb{R} : |z_l| \sqrt{\lambda} \leq |y_l|/2\}$ and use the bound $1 + |y_l + z_l \sqrt{\lambda}| \geq \mathbf{1}_{\mathcal{A}_{\lambda,l}^c} + \frac{1}{2}(1 + |y_l|)\mathbf{1}_{\mathcal{A}_{\lambda,l}}$: it gives

$$\begin{aligned} & \int_{\mathbb{R}} \frac{1}{(1 + |y_l + z_l \sqrt{\lambda}|)^{\mu+1}} \frac{e^{-|z_l|^2/2}}{\sqrt{2\pi}} dz_l \\ & \leq \int_{\mathcal{A}_{\lambda,l}^c} \frac{e^{-|z_l|^2/2}}{\sqrt{2\pi}} dz_l + 2^{\mu+1} \int_{\mathcal{A}_{\lambda,l}} \frac{1}{(1 + |y_l|)^{\mu+1}} \frac{e^{-|z_l|^2/2}}{\sqrt{2\pi}} dz_l \\ & \leq 2\mathcal{N}\left(-\frac{|y_l|}{2\sqrt{\lambda}}\right) + 2^{\mu+1} \frac{1}{(1 + |y_l|)^{\mu+1}} \\ & \leq \frac{C_{\mu,\Lambda}}{(1 + |y_l|)^{\mu+1}} \end{aligned}$$

where $\mathcal{N}(\cdot)$ is the cumulative density function of the standard normal distribution, where $C_{\mu,\Lambda}$ is a finite positive constant depending only on μ and Λ , and where at the last line, we have used that $\mathcal{N}(-x)$ is decreasing when $x \rightarrow +\infty$ faster than any polynomial. Therefore, we have proved

$$\mathcal{I}_1 \leq 4^q \prod_{l=1}^d \frac{C_{\mu,\Lambda}}{(1 + |y_l|)^{\mu+1}}. \quad (3.5)$$

Now we deal with \mathcal{I}_2 , i.e. the part of the integral of (3.4) restricted to \mathcal{A}_λ^c . On this set, $|\mathbf{z}|^2/2 \geq |\mathbf{z}|^2/4 + |\mathbf{y}|^2/(16\Lambda)$, therefore we obtain

$$\mathcal{I}_2 \leq (1 + |\mathbf{y}|^2)^q \int_{\mathcal{A}_\lambda^c} \frac{e^{-|\mathbf{z}|^2/2}}{(2\pi)^{d/2}} d\mathbf{z} \leq (1 + |\mathbf{y}|^2)^q e^{-|\mathbf{y}|^2/(16\Lambda)} \int_{\mathcal{A}_\lambda^c} \frac{e^{-|\mathbf{z}|^2/4}}{(2\pi)^{d/2}} d\mathbf{z}$$

$$\begin{aligned}
&\leq 2^{d/2}(1+|\mathbf{y}|^2)^q e^{-|\mathbf{y}|^2/(16\Lambda)} \\
&\leq C_{\mu,q,d,\Lambda} \prod_{l=1}^d \frac{1}{(1+|y_l|)^{\mu+1}}
\end{aligned} \tag{3.6}$$

where the last inequality follows from usual comparisons between exponential and polynomial growth. Gathering (3.5) and (3.6) gives the announced estimate (3.4) for some constant $C_\nu(\Lambda)$. Theorem 1 is proved. \square

3.2 Proof of Theorem 2

In all this section, we assume the assumptions of Theorem 2 are in force, without further reference.

In addition to the coefficients $\bar{\alpha}_{i,\mathbf{k}}^{(q,M)}$ calculated with the responses computed using the erroneous functions $\left(\bar{y}_j^{(q,M)}\right)_{i < j < N}$ (see the definitions (2.24) and (2.25)), we define the following coefficients $\bar{\alpha}_{i,\mathbf{k}}^{(q)}$ calculated with the responses computed with the exact functions $\left(y_j^{(q)}\right)_{i < j < N}$,

$$\bar{\alpha}_{i,\mathbf{k}}^{(q)} := \frac{1}{M} \sum_{m=1}^M S_i^{(q)} \left(X_{i:N}^{i,m} \right) \phi_{\mathbf{k}} \left(X_i^{i,m} \right), \tag{3.7}$$

and thus

$$\bar{y}_i^{(q)}(\cdot) := \sum_{\mathbf{k} \in \Gamma} \bar{\alpha}_{i,\mathbf{k}}^{(q)} \phi_{\mathbf{k}}(\cdot). \tag{3.8}$$

3.2.1 Preliminary results

The next result states a decomposition of the local error as a superposition of an approximation error (truncation over the functions basis) and a statistical error.

Lemma 9 (Bias/variance decomposition) *For each $i \in \{0, \dots, N-1\}$ we have*

$$\mathbb{E} \left[\left\| \bar{y}_i^{(q)}(\cdot) - y_i^{(q)}(\cdot) \right\|_\nu^2 \right] \leq \sum_{\mathbf{k} \in (\mathbb{N}^d - \Gamma)} (\alpha_{i,\mathbf{k}}^{(q)})^2 + \frac{\mathcal{L}_\Gamma}{M} \mathbb{E} \left[\left(S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \right] =: \mathcal{E}_i$$

where \mathcal{L}_Γ is the Christoffel number of the functions basis defined in (2.27) and the definition of \mathcal{E}_i is stated in (2.28).

Proof By definition of $\bar{y}_i^{(q)}$ and $y_i^{(q)}$,

$$\mathbb{E} \left[\left\| \bar{y}_i^{(q)}(\cdot) - y_i^{(q)}(\cdot) \right\|_\nu^2 \right] = \mathbb{E} \left[\left\| - \sum_{\mathbf{k} \in (\mathbb{N}^d - \Gamma)} \alpha_{i,\mathbf{k}}^{(q)} \phi_{\mathbf{k}}(\cdot) + \sum_{\mathbf{k} \in \Gamma} \left(\bar{\alpha}_{i,\mathbf{k}}^{(q)} - \alpha_{i,\mathbf{k}}^{(q)} \right) \phi_{\mathbf{k}}(\cdot) \right\|_\nu^2 \right].$$

By orthonormality of the $\{\phi_{\mathbf{k}}(\cdot)\}$ we obtain

$$\mathbb{E} \left[\left\| \bar{y}_i^{(q)}(\cdot) - y_i^{(q)}(\cdot) \right\|_\nu^2 \right] = \sum_{\mathbf{k} \in (\mathbb{N}^d - \Gamma)} (\alpha_{i,\mathbf{k}}^{(q)})^2 + \sum_{\mathbf{k} \in \Gamma} \mathbb{E} \left[\left| \bar{\alpha}_{i,\mathbf{k}}^{(q)} - \alpha_{i,\mathbf{k}}^{(q)} \right|^2 \right].$$

For $\mathbf{k} \in \Gamma$, observe that $\mathbb{E} \left[\bar{\alpha}_{i,\mathbf{k}}^{(q)} \right] = \alpha_{i,\mathbf{k}}^{(q)}$, therefore

$$\begin{aligned} \mathbb{E} \left[\left| \bar{\alpha}_{i,\mathbf{k}}^{(q)} - \alpha_{i,\mathbf{k}}^{(q)} \right|^2 \right] &= \mathbb{V}\text{ar} \left[\bar{\alpha}_{i,\mathbf{k}}^{(q)} \right] = \frac{1}{M} \mathbb{V}\text{ar} \left[S_i^{(q)} \left(X_{i:N}^i \right) \phi_{\mathbf{k}} \left(X_i^i \right) \right] \\ &\leq \frac{1}{M} \|\phi_{\mathbf{k}}(\cdot)\|_{\infty}^2 \mathbb{E} \left[\left(S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \right], \end{aligned}$$

which leads to the desired result. \square

Before stating the next preliminary result we define the following σ -algebras generated by the cloud of simulations,

$$\mathcal{G}_i := \sigma(\mathcal{C}_i) \quad \text{and} \quad \mathcal{G}_{i:N} := \sigma(\mathcal{C}_i, \dots, \mathcal{C}_N).$$

Lemma 10 *We have*

$$\begin{aligned} \sum_{\mathbf{k} \in \Gamma} \left(\mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right) \phi_{\mathbf{k}} \left(X_i^i \right) \right] \right)^2 \\ \leq \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \right], \end{aligned} \quad (3.9)$$

$$\begin{aligned} \sum_{\mathbf{k} \in \Gamma} \left(\mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right) \phi_{\mathbf{k}} \left(X_i^i \right) \mid \mathcal{G}_{i+1:N} \right] \right)^2 \\ \leq \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \mid \mathcal{G}_{i+1:N} \right]. \end{aligned} \quad (3.10)$$

Proof We start with the proof of (3.9). Let $\beta_{i,\mathbf{k}}$ be the projection coefficients in $L_{\nu}^2(\mathbb{R}^d)$ of the function

$$h_i(\mathbf{x}) := \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right) \mid X_i^i = \mathbf{x} \right]$$

over the functions $\{\phi_{\mathbf{k}}(\cdot)\}$. Since the latter form an orthonormal basis of $L_{\nu}^2(\mathbb{R}^d)$ (Proposition 2) and owing to the tower property of conditional expectation, we have

$$\beta_{i,\mathbf{k}} = \mathbb{E} \left[h_i \left(X_{i:N}^i \right) \phi_{\mathbf{k}} \left(X_i^i \right) \right] = \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right) \phi_{\mathbf{k}} \left(X_i^i \right) \right]. \quad (3.11)$$

By the Parseval identity and the Jensen inequality, we get

$$\begin{aligned} \sum_{\mathbf{k} \in \mathbb{N}^d} \beta_{i,\mathbf{k}}^2 &= \mathbb{E} \left[h_i^2 \left(X_i^i \right) \right] \leq \mathbb{E} \left[\mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \mid X_i^i \right] \right] \\ &= \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \right]. \end{aligned}$$

By recalling the representation (3.11) for $\beta_{i,\mathbf{k}}$, the above inequality readily leads to (3.9).

We now prove (3.10). Similarly, denote by $\beta_{i,\mathbf{k}}(\mathcal{G}_{i+1:N})$ the $L_{\nu}^2(\mathbb{R}^d)$ -projection coefficients of the (random) function

$$h_i(\mathbf{x}, \mathcal{G}_{i+1:N}) = \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right) \mid X_i^i = \mathbf{x}, \mathcal{G}_{i+1:N} \right].$$

Remind of the independence of $\mathcal{G}_{i+1:N}$ and $X_{i:N}^i$. Therefore, using similar arguments as before, we obtain

$$\beta_{i,\mathbf{k}}(\mathcal{G}_{i+1:N}) = \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right) \phi_{\mathbf{k}}(X_i^i) \mid \mathcal{G}_{i+1:N} \right]$$

and

$$\sum_{\mathbf{k} \in \mathbb{N}^d} |\beta_{i,\mathbf{k}}(\mathcal{G}_{i+1:N})|^2 \leq \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \mid \mathcal{G}_{i+1:N} \right],$$

from which (3.10) readily follows. We are done. \square

3.2.2 Proof of Theorem 2

Let $\varepsilon_i := \mathbb{E} \left[\left\| \bar{y}_i^{(q,M)}(\cdot) - y_i^{(q)}(\cdot) \right\|_{\nu}^2 \right]$. By definition of $\bar{y}_i^{(q,M)}$, $y_i^{(q)}(\cdot)$ and $\bar{y}_i^{(q)}(\cdot)$,

$$\begin{aligned} \varepsilon_i &= \mathbb{E} \left[\left\| \sum_{\mathbf{k} \in \Gamma} \bar{\alpha}_{i,\mathbf{k}}^{(q,M)} \phi_{\mathbf{k}}(\cdot) - y_i^{(q)}(\cdot) \right\|_{\nu}^2 \right] \\ &= \mathbb{E} \left[\left\| \bar{y}_i^{(q)}(\cdot) - y_i^{(q)}(\cdot) + \sum_{\mathbf{k} \in \Gamma} \left(\bar{\alpha}_{i,\mathbf{k}}^{(q,M)} - \bar{\alpha}_{i,\mathbf{k}}^{(q)} \right) \phi_{\mathbf{k}}(\cdot) \right\|_{\nu}^2 \right]. \end{aligned}$$

Then, using the triangular inequality for $\|\cdot\|_{\nu}$ norm and the inequality $(a+b)^2 \leq 2a^2 + 2b^2$, we obtain

$$\varepsilon_i \leq 2\mathbb{E} \left[\left\| \bar{y}_i^{(q)}(\cdot) - y_i^{(q)}(\cdot) \right\|_{\nu}^2 \right] + 2\mathbb{E} \left[\left\| \sum_{\mathbf{k} \in \Gamma} \left(\bar{\alpha}_{i,\mathbf{k}}^{(q,M)} - \bar{\alpha}_{i,\mathbf{k}}^{(q)} \right) \phi_{\mathbf{k}}(\cdot) \right\|_{\nu}^2 \right] := \varepsilon_i^{(a)} + \varepsilon_i^{(b)}. \quad (3.12)$$

The first term $\varepsilon_i^{(a)}$ is bounded taking into account Lemma 9:

$$\varepsilon_i^{(a)} \leq 2\mathcal{E}_i, \quad (3.13)$$

Regarding the second term $\varepsilon_i^{(b)}$, the orthonormality of $\{\phi_{\mathbf{k}}(\cdot)\}$ gives

$$\begin{aligned} \varepsilon_i^{(b)} &= 2 \sum_{\mathbf{k} \in \Gamma} \mathbb{E} \left[\left(\bar{\alpha}_{i,\mathbf{k}}^{(q,M)} - \bar{\alpha}_{i,\mathbf{k}}^{(q)} \right)^2 \right] \\ &= 2 \sum_{\mathbf{k} \in \Gamma} \text{Var} \left[\bar{\alpha}_{i,\mathbf{k}}^{(q,M)} - \bar{\alpha}_{i,\mathbf{k}}^{(q)} \right] + 2 \sum_{\mathbf{k} \in \Gamma} \left(\mathbb{E} \left[\bar{\alpha}_{i,\mathbf{k}}^{(q,M)} - \bar{\alpha}_{i,\mathbf{k}}^{(q)} \right] \right)^2 \\ &:= \varepsilon_i^{(b,c)} + \varepsilon_i^{(b,d)}. \end{aligned} \quad (3.14)$$

To handle $\varepsilon_i^{(b,d)}$ write

$$\bar{\alpha}_{i,\mathbf{k}}^{(q,M)} - \bar{\alpha}_{i,\mathbf{k}}^{(q)} = \frac{1}{M} \sum_{m=1}^M \left(S_i^{(q,M)} \left(X_{i:N}^{i,m} \right) - S_i^{(q)} \left(X_{i:N}^{i,m} \right) \right) \phi_{\mathbf{k}}(X_i^{i,m}),$$

and apply Lemma 10, thus obtaining

$$\begin{aligned}\varepsilon_i^{(b,d)} &= 2 \sum_{\mathbf{k} \in \Gamma} \left(\mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right) \phi_{\mathbf{k}} \left(X_i^i \right) \right] \right)^2 \\ &\leq 2 \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \right].\end{aligned}\quad (3.15)$$

Now we focus on $\varepsilon_i^{(b,c)}$. We recall the conditional variance formula $\mathbb{V}\text{ar}[X] = \mathbb{E}[\mathbb{V}\text{ar}[X | \mathcal{G}]] + \mathbb{V}\text{ar}[\mathbb{E}[X | \mathcal{G}]]$, available for any scalar random variable X and any sigma-algebra \mathcal{G} : then,

$$\begin{aligned}\varepsilon_i^{(b,c)} &= 2 \sum_{\mathbf{k} \in \Gamma} \mathbb{E} \left[\mathbb{V}\text{ar} \left[\bar{\alpha}_{i,\mathbf{k}}^{(q,M)} - \bar{\alpha}_{i,\mathbf{k}}^{(q)} \mid \mathcal{G}_{i+1:N} \right] \right] + 2 \sum_{\mathbf{k} \in \Gamma} \mathbb{V}\text{ar} \left[\mathbb{E} \left[\bar{\alpha}_{i,\mathbf{k}}^{(q,M)} - \bar{\alpha}_{i,\mathbf{k}}^{(q)} \mid \mathcal{G}_{i+1:N} \right] \right] \\ &:= \varepsilon_i^{(b,c,e)} + \varepsilon_i^{(b,c,f)}.\end{aligned}$$

We start with $\varepsilon_i^{(b,c,e)}$:

$$\begin{aligned}\varepsilon_i^{(b,c,e)} &= 2 \sum_{\mathbf{k} \in \Gamma} \mathbb{E} \left[\mathbb{V}\text{ar} \left[\frac{1}{M} \sum_{m=1}^M \left(S_i^{(q,M)} \left(X_{i:N}^{i,m} \right) - S_i^{(q)} \left(X_{i:N}^{i,m} \right) \right) \phi_{\mathbf{k}} \left(X_i^{i,m} \right) \mid \mathcal{G}_{i+1:N} \right] \right] \\ &= 2 \sum_{\mathbf{k} \in \Gamma} \mathbb{E} \left[\frac{1}{M^2} \sum_{m=1}^M \mathbb{V}\text{ar} \left[\left(S_i^{(q,M)} \left(X_{i:N}^{i,m} \right) - S_i^{(q)} \left(X_{i:N}^{i,m} \right) \right) \phi_{\mathbf{k}} \left(X_i^{i,m} \right) \mid \mathcal{G}_{i+1:N} \right] \right] \\ &\leq 2 \sum_{\mathbf{k} \in \Gamma} \mathbb{E} \left[\frac{1}{M^2} \sum_{m=1}^M \|\phi_{\mathbf{k}}(\cdot)\|_{\infty}^2 \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^{i,m} \right) - S_i^{(q)} \left(X_{i:N}^{i,m} \right) \right)^2 \mid \mathcal{G}_{i+1:N} \right] \right] \\ &\leq \frac{2\mathcal{L}_{\Gamma}}{M} \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \right].\end{aligned}$$

We continue with $\varepsilon_i^{(b,c,f)}$: use Lemma 10 to write

$$\begin{aligned}\varepsilon_i^{(b,c,f)} &= 2 \sum_{\mathbf{k} \in \Gamma} \mathbb{V}\text{ar} \left[\mathbb{E} \left[\bar{\alpha}_{i,\mathbf{k}}^{(q,M)} - \bar{\alpha}_{i,\mathbf{k}}^{(q)} \mid \mathcal{G}_{i+1:N} \right] \right] \\ &\leq 2 \sum_{\mathbf{k} \in \Gamma} \mathbb{E} \left[\left(\mathbb{E} \left[\bar{\alpha}_{i,\mathbf{k}}^{(q,M)} - \bar{\alpha}_{i,\mathbf{k}}^{(q)} \mid \mathcal{G}_{i+1:N} \right] \right)^2 \right] \\ &= 2 \mathbb{E} \left[\sum_{\mathbf{k} \in \Gamma} \left(\mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right) \phi_{\mathbf{k}} \left(X_i^i \right) \mid \mathcal{G}_{i+1:N} \right] \right)^2 \right] \\ &\leq 2 \mathbb{E} \left[\mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \mid \mathcal{G}_{i+1:N} \right] \right] \\ &= 2 \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \right].\end{aligned}$$

Summarizing the above computations, we get a bound for $\varepsilon_i^{(b,c)}$:

$$\varepsilon_i^{(b,c)} \leq 2 \left(\frac{\mathcal{L}_{\Gamma}}{M} + 1 \right) \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \right].$$

All in all, in view of (3.14)-(3.15) and the above, we have derived a bound on $\varepsilon_i^{(b)}$:

$$\varepsilon_i^{(b)} \leq 2 \left(\frac{\mathcal{L}_\Gamma}{M} + 2 \right) \mathbb{E} \left[\left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \right]. \quad (3.16)$$

It remains to estimate the above term with the difference of the responses. Using the Cauchy-Schwarz inequality for the summation in j , and taking into account that f is Lipschitz in y , we obtain

$$\begin{aligned} & \left(S_i^{(q,M)} \left(X_{i:N}^i \right) - S_i^{(q)} \left(X_{i:N}^i \right) \right)^2 \\ &= \frac{1}{(1 + |X_i^i|^2)^q} \left(\sum_{j=i}^{N-1} \left(f_j \left(X_j^i, \mathcal{T}_{L^*}(\bar{y}_{j+1}^{(q,M)}(X_{j+1}^i))(1 + |X_{j+1}^i|^2)^{q/2} \right) \right. \right. \\ & \quad \left. \left. - f_j \left(X_j^i, y_{j+1}^{(q)}(X_{j+1}^i)(1 + |X_{j+1}^i|^2)^{q/2} \right) \right) \Delta \right)^2 \\ &\leq \frac{(T - t_i)}{(1 + |X_i^i|^2)^q} \sum_{j=i}^{N-1} \left(f_j \left(X_j^i, \mathcal{T}_{L^*}(\bar{y}_{j+1}^{(q,M)}(X_{j+1}^i))(1 + |X_{j+1}^i|^2)^{q/2} \right) \right. \\ & \quad \left. - f_j \left(X_j^i, y_{j+1}^{(q)}(X_{j+1}^i)(1 + |X_{j+1}^i|^2)^{q/2} \right) \right)^2 \Delta \end{aligned}$$

using that $\mathcal{T}_{L^*}(y_{j+1}^{(q)}(\mathbf{x})(1 + |\mathbf{x}|^2)^{q/2}) = y_{j+1}^{(q)}(\mathbf{x})(1 + |\mathbf{x}|^2)^{q/2}$ and that \mathcal{T}_{L^*} is 1-Lipschitz (see Proposition 1 and Definition (1.12))

$$\leq \frac{(T - t_i)}{(1 + |X_i^i|^2)^q} \sum_{j=i}^{N-1} L_f^2 \left| \bar{y}_{j+1}^{(q,M)}(X_{j+1}^i) - y_{j+1}^{(q)}(X_{j+1}^i) \right|^2 (1 + |X_{j+1}^i|^2)^q \Delta.$$

Combine this with (3.16) to obtain,

$$\varepsilon_i^{(b)} \leq 2 \left(\frac{\mathcal{L}_\Gamma}{M} + 2 \right) T L_f^2 \sum_{j=i}^{N-1} \mathbb{E} \left[\frac{(1 + |X_{j+1}^i|^2)^q}{(1 + |X_i^i|^2)^q} \left| \bar{y}_{j+1}^{(q,M)}(X_{j+1}^i) - y_{j+1}^{(q)}(X_{j+1}^i) \right|^2 \right] \Delta.$$

Finally, taking advantage of the norm-stability property of the distribution ν (see (2.13)) we get,

$$\varepsilon_i^{(b)} \leq 2 \left(\frac{\mathcal{L}_\Gamma}{M} + 2 \right) T L_f^2 c_\nu \sum_{j=i}^{N-1} \varepsilon_{j+1} \Delta. \quad (3.17)$$

Let us set $C_\varepsilon := 2 \left(\frac{\mathcal{L}_\Gamma}{M} + 2 \right) T L_f^2 c_\nu$. At this point, in view of (3.12), (3.13) and (3.17) we have obtained

$$\varepsilon_i \leq 2\mathcal{E}_i + C_\varepsilon \sum_{j=i}^{N-1} \varepsilon_{j+1} \Delta.$$

Apply now [35, Proposition 2.4] with $\alpha = \beta = 1/2$ to get the existence of a constant $\mathcal{C}(C_\varepsilon, T)$ (depending only on C_ε, T) such that

$$\varepsilon_i \leq 2\mathcal{E}_i + \mathcal{C}(C_\varepsilon, T)C_\varepsilon \sum_{j=i}^{N-1} \mathcal{E}_{j+1}\Delta.$$

We are done. \square

3.3 Proof of Proposition 1

Without loss of generality, we can assume from now that $\eta_f = \eta_g$ because assumptions on f and g are still valid by choosing the maximum parameter between η_f and η_g . Set $\eta := \eta_f = \eta_g$. First note that the constant C_η in (2.5) is finite since b and σ are bounded, this follows from standard manipulations which are left to the reader.

Second we prove (2.4) by induction on $i = N, \dots, 0$. It is obvious for $i = N$ since $C_y \geq C_g$ (note that $C_\eta \geq 1$). Now assume that

$$\sup_{\mathbf{x} \in \mathbb{R}^d} \frac{|y_j(\mathbf{x})|}{(1 + |\mathbf{x}|^2)^{\frac{\eta}{2}}} \leq C_j \quad (3.18)$$

for $j = N, \dots, i+1$ for some $i \in \{N, \dots, 0\}$ and for some finite C_j 's, and let us prove that the inequality holds also for $j = i$. In the proof below, we will get a relation between the C_j 's leading to the statement of Proposition 1. We start from (2.7) and make use of the growth assumptions on f, g and the Lipschitz property of f : we have

$$\begin{aligned} |y_i(\mathbf{x})| &\leq \mathbb{E} \left[C_g(1 + |X_N|^2)^{\frac{\eta}{2}} \right. \\ &\quad \left. + \sum_{j=i}^{N-1} \left(C_f(1 + |X_j|^2)^{\frac{\eta}{2}} + L_f C_{j+1}(1 + |X_{j+1}|^2)^{\frac{\eta}{2}} \right) \Delta \mid X_i = \mathbf{x} \right] \\ &\leq (1 + |\mathbf{x}|^2)^{\frac{\eta}{2}} \left(C_g C_\eta + T C_f C_\eta + C_\eta L_f \sum_{j=i+1}^N C_j \Delta \right). \end{aligned}$$

Therefore, (3.18) holds for $j = i$ too, and thus owing to the induction principle, it holds for any $j \in \{0, \dots, N\}$. Moreover, we can take

$$C_i := C_g C_\eta + T C_f C_\eta + C_\eta L_f \sum_{j=i+1}^N C_j \Delta.$$

The Gronwall lemma implies that $C_i \leq C_\eta(C_g + T C_f)e^{C_\eta L_f T}$. \square

4 GPU implementations

Two implementations on GPUs of Algorithm 1 are proposed in this section. The first one is a parallelization of the algorithm on a single GPU, we will refer to it as the (mono-)GPU version. This version includes two kernels, one simulates the paths of the forward process and computes the associated responses, the other one computes the regression coefficients $(\alpha_{i,\mathbf{k}}^{(q)}, \mathbf{k} \in \Gamma)$. In the first kernel the initial value of each simulated path of the forward process is stored in a device vector in global memory, it will be read later in the second kernel. In order to minimize the number of memory transactions and therefore maximize performance, all accesses to global memory have been implemented in a coalesced way. The random numbers needed for the path generation of the forward process were generated on the fly (inline generation) taking advantage of the NVIDIA cuRAND library [17] and the generator MRG32k3a proposed by L'Ecuyer in [45]. Therefore, inside this kernel the random number generator is called as needed. Another approach would be the pre-generation of the random numbers in a separate previous kernel, storing them in GPU global memory and reading them back from this device memory in the next kernel. Both alternatives have advantages and drawbacks. In this work we have chosen inline generation having in mind that this option is faster and saves global memory. Besides, register swapping was not observed on the implementation and the quality of the obtained solutions is similar to the accuracy of pure sequential traditional CPU solutions achieved employing more complex random number generators. Additionally, the pre-generation of random numbers is not feasible when dealing with a huge number of simulations M , as required for high dimensional problems, since it will require more memory than the available in the GPU. In the second kernel, in order to compute the regression coefficients, a parallelization not only over the multi-indices $\mathbf{k} \in \Gamma$ but also over the simulations $1 \leq m \leq M$ was proposed. Thus, blocks of threads parallelize the outer for loop $\forall \mathbf{k} \in \Gamma$, whilst the threads inside each block carry out in parallel the inner loop traversing the vectors of the responses and the simulations. We stress that the parallel sum reduction (2.25) has been implemented in shared memory avoiding bank conflicts in pursuance of getting the most of the GPU.

A host can have several GPUs attached (in our machine, 4 GPUs TITAN BLACK). When dealing with high dimensional problems demanding a huge number of simulations M it is compelling to harness the computing power of these extra GPUs to further reduce execution times. The approach we have followed is to use OpenMP [52] to control all GPUs inside the host node. This strategy involves launching as many CPU threads as GPUs available on the node, so that each CPU thread handles one GPU. We will allude this version as M(ulti)GPU. In order to simulate the forward process and compute the responses, each GPU runs in parallel the first previous mono-GPU kernel, storing in its global memory the initial Student random variable of each path and the computed responses of the paths. Having done that, in order to compute the regression coefficients, the outer loop $\forall \mathbf{k} \in \Gamma$ is executed sequentially in CPU, whereas the inner loop on the simulations $1 \leq m \leq M$ is performed in parallel across all GPUs. More precisely, the parallel reduction operation is executed in two levels. The first reduction level is carried out inside each one of the GPUs, it has been performed using the *reduce* primitive of the NVIDIA Thrust library [61]. The second level of reduction was made in the plane of CPU threads. This approach of parallelizing only over the

simulations of the forward process behaves optimally when the number of simulations is large enough to fully occupy the GPU. In fact, this is the scenario we will be dealing with when tackling high dimensional problems. Actually, as sketched in Theorem 2, in order to get accurate approximations we need $\frac{2^d \#I}{M}$ to be small, therefore one needs to choose M much bigger than $\#I$. This fact motivates a small variant of this multi-GPU version, specifically designed for problems where M is so large that we cannot store the initial values of the simulated paths in the set of global memories of the GPUs. In the first kernel in charge of simulating the paths of the forward process and computing the associated responses, now the initial values of the forward process at each time step are not stored, thus saving $d \times M \times 4$ bytes of global memory². Instead, we just store the initial random seeds of the launched threads, which will be used in the next kernel in order to resimulate the same initial values of the forward process and allow the computation of the regression coefficients. This strategy is even as efficient as the previous one, on the grounds that GPUs are (most often) faster recomputing data than getting this data back from global memory. This is due to the fact that the version recomputing data is less memory bounded than the first one.

5 Numerical experiments

We introduce the following functions

$$g(\mathbf{x}) := 1 + \kappa + \sin \left(\lambda \sum_{l=1}^d x_l \right),$$

$$f(t, \mathbf{x}, y) := \min \left(1, \left[y - \kappa - 1 - \sin \left(\lambda \sum_{l=1}^d x_l \right) \exp \left(\frac{\lambda^2 d(t-T)}{2} \right) \right]^2 \right).$$

We aim to solve the PDE:

$$\begin{aligned} \partial_t u(t, \mathbf{x}) + \frac{1}{2}(\Delta_x u)(t, \mathbf{x}) + f(t, \mathbf{x}, u(t, \mathbf{x})) &= 0, \quad t \leq T, \\ u(T, \mathbf{x}) &= g(\mathbf{x}). \end{aligned} \tag{5.1}$$

Note that

$$u(t, \mathbf{x}) = 1 + \kappa + \sin \left(\lambda \sum_{l=1}^d x_l \right) \exp \left(\frac{\lambda^2 d(t-T)}{2} \right),$$

is solution to (5.1). The probabilistic formulation goes as follows:

$$\begin{aligned} dX_t &= dW_t, \\ -dY_t &= f(t, X_t, Y_t)dt - Z_t dW_t, \\ Y_T &= g(X_T). \end{aligned}$$

Then, the processes (Y, Z) satisfy $Y_t = u(t, X_t)$ and $Z_t = \nabla_x u(t, X_t)$. For the numerical results the final time T is fixed to 1, $\kappa = \frac{6}{10}$ and $\lambda = \frac{1}{\sqrt{d}}$.

² Single precision data type is assumed.

The μ parameter of the ν_l Student's t-distribution is chosen as $\mu = 2$. Under this choice $\nu_l(x_l) = 0.5(1 + x_l^2)^{-3/2}$. The marginal CDF and its inverse are given by

$$F_{\nu_l}(x) = \frac{1}{2} \left(\frac{x}{\sqrt{x^2 + 1}} + 1 \right),$$

$$F_{\nu_l}^{-1}(u) = \begin{cases} -\sqrt{\frac{-u^2 + u - 0.25}{u(u-1)}} & \text{if } 0 < u \leq 0.5, \\ \sqrt{\frac{-u^2 + u - 0.25}{u(u-1)}} & \text{if } 0.5 < u < 1. \end{cases} \quad (5.2)$$

In order to sample according to ν , we draw d independent random variables (U_1, \dots, U_d) with uniform distribution on $(0, 1)$, and we compute

$$X_i^i := \left(F_{\nu_1}^{-1}(U_1), \dots, F_{\nu_d}^{-1}(U_d) \right) \stackrel{d}{\sim} \nu.$$

First we present some numerical results in dimension 1. Taking into account that

$$d_x^r y_i(x) = \pm \lambda \exp\left(\frac{t-1}{2}\right) [\mathbf{1}_{r \text{ even}} \sin(x) + \mathbf{1}_{r \text{ odd}} \cos(x)],$$

in order to fit the assumptions of Lemma 3 we select $p = r$. The values of q are selected using relation (2.37) under $p = r$ for $r = 0, 1, 2$ and 3. Therefore, for $q = 0$ only the function $h_i^{(q)}$ is square integrable; for $q = 2.1$ the function $h_i^{(q)}$ and its first spatial derivative are square integrable and the function $h_i^{(q)}$ vanish for $u \rightarrow 0^+, 1^-$; finally, for $q = 8.1$ the function $h_i^{(q)}$ and its first two spatial derivatives are square integrable and vanish for $u \rightarrow 0^+, 1^-$, its third spatial derivative is square integrable, as well.

In Figure 2, the analytical solution $y_i^{(q)}$ versus the computed approximation $\bar{y}_i^{(q,M)}$ are plotted for $t = 0.5$ and the values of $q = 0, 2.1, 5.1, 8.1$. The function was approximated on 1000 independent and identically ν distributed simulations. As long as q increases the convergence of the algorithm is improved (the truncation error shrinks to 0, coherently with Proposition 3). In Figure 3 the corresponding results are presented for $h_i^{(q)}$ and $\bar{h}_i^{(q,M)}$; for q large the functions are very flat at the boundaries 0 and 1, as expected. The parameters of the GQRMDP algorithm were chosen $\Delta = 0.01$, $K = 200$ meaning that $y_i^{(q)}$ was projected over 201 basis functions, and $M = 2 \times 10^6$. This selection makes $\frac{2^d \# \Gamma}{M}$ small, in fact it is 2.01×10^{-4} , thus we control well the statistical error of the algorithm.

In order to assess the performance of the algorithm, we compute the average mean squared error (MSE) over 10^3 independent runs of the algorithm for two error indicators:

$$MSE_{\max} := \ln \left\{ 10^{-3} \max_{0 \leq i \leq N-1} \sum_{m=1}^{10^3} \left| y_i^{(q)}(R_{i,m}) - \bar{y}_i^{(q,M)}(R_{i,m}) \right|^2 \right\},$$

$$MSE_{\text{av}} := \ln \left\{ 10^{-3} N^{-1} \sum_{m=1}^{10^3} \sum_{i=0}^{N-1} \left| y_i^{(q)}(R_{i,m}) - \bar{y}_i^{(q,M)}(R_{i,m}) \right|^2 \right\}, \quad (5.3)$$

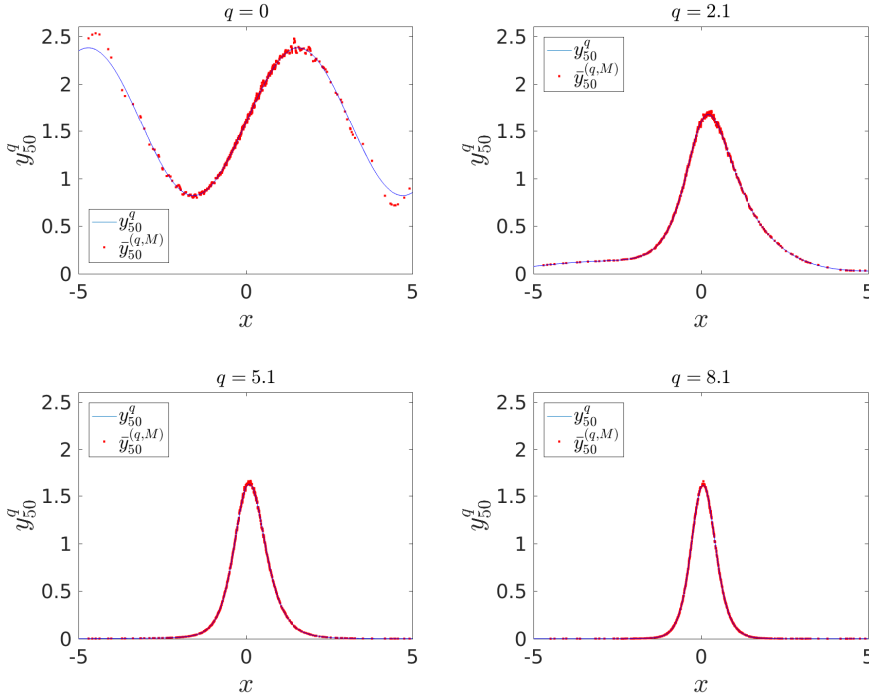


Fig. 2 For $t = 0.5$, $y_i^{(q)}$ versus $\bar{y}_i^{(q,M)}$ considering $q = 0, 2.1, 5.1, 8.1$. $\bar{y}_i^{(q,M)}$ were computed using $\Delta = 0.01$, $K = 200$, $M = 2 \times 10^6$.

where the simulations $\{R_{i,m}; i = 0, \dots, N-1, m = 1, \dots, 10^3\}$ are independent and identically ν -distributed, and independently drawn from the simulations used for the GQRMDP scheme.

In order to test the theoretical results we compare the performance of the algorithm according to the choice of the damping exponent q and the multi-indices set. We will show the impact of these choices on the convergence of the approximation of the semi-linear PDE and on the computational performance in terms of computational time and memory consumption.

The numerical experiments have been performed with the following hardware and software configurations: four GPUs GeForce GTX TITAN Black with 6 GBytes of global memory per device (see [51] for details in the architecture), two multicore Intel Xeon CPUs E5-2620 v2 clocked at 2.10 GHz (6 cores per socket) with 62 GBytes of RAM, CentOS Linux, NVIDIA CUDA SDK 7.5 and INTEL C compiler 15.0.6. The CPU programs were optimized and parallelized using OpenMP [52]. The multicore CPUs time (CPU) and the GPU time (GPU) will all be measured in seconds in the forthcoming tables. CPU times correspond to executions using 24 threads so as to take advantage of Intel Hyperthreading. Unless otherwise indicated, the results are obtained using single precision, both in CPU and GPU.

All examples will be run using 64 thread blocks, each with 256 threads. In the following Table 2 we show results for $d = 1$ and the choices of the damping

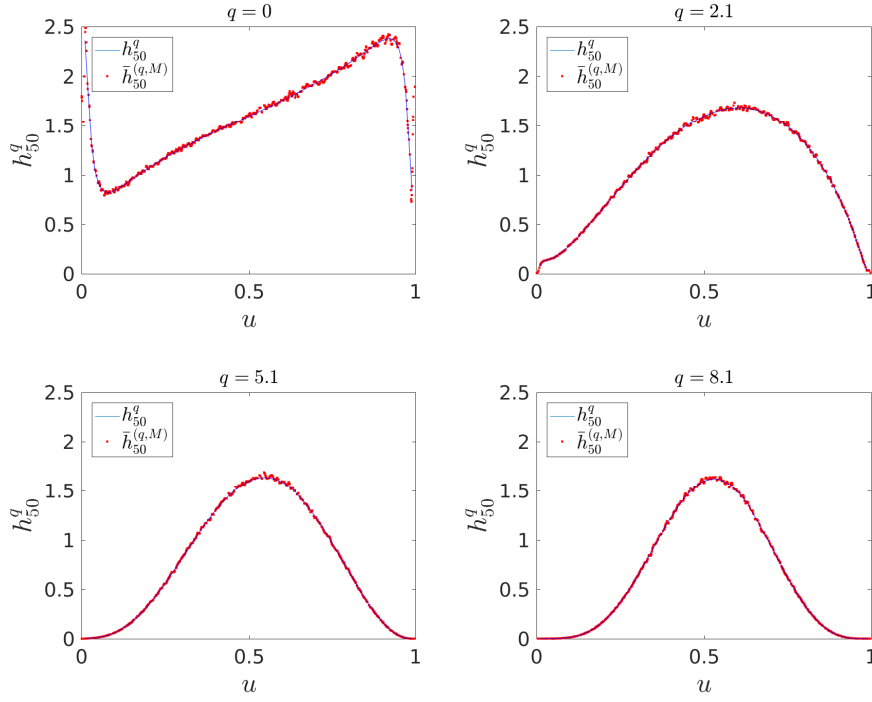


Fig. 3 For $t = 0.5$, $h_i^{(q)}$ versus $\bar{h}_i^{(q,M)}$ considering $q = 0, 2.1, 5.1, 8.1$. $\bar{h}_i^{(q,M)}$ were computed using $\Delta = 0.01$, $K = 200$, $M = 2 \times 10^6$.

exponent $q = 0, 2.1, 5.1, 8.1$. As we have just seen in Figure 2, we expect (see Proposition 3) that the convergence of the algorithm is improved for large values of the damping exponent q . Besides, as long as the time step decreases, our parameter choice is such that K increases and $\frac{2K}{M}$ decreases so that the algorithm is more and more accurate. The GPU implementation provides a significant reduction in the computational time: the GPU speed-up reaches the value 12.2. One usually expects a GPU speed-up factor between 3x-10x vs two sockets of CPU [46], therefore both CPU and GPU codes were thoroughly optimized.

Table 2 $d = 1$.

Δ	K	M	$q = 0$		$q = 2.1$		$q = 5.1$	
			MSE_{\max}	MSE_{av}	MSE_{\max}	MSE_{av}	MSE_{\max}	MSE_{av}
0.05	100	2×10^4	-3.658	-3.868	-4.615	-4.874	-4.517	-5.130
0.02	150	2×10^5	-4.427	-4.900	-6.450	-6.740	-6.685	-7.094
0.01	200	2×10^6	-4.704	-5.524	-8.465	-8.729	-8.717	-9.096
Δ	K	M	$q = 8.1$		CPU	GPU		
			MSE_{\max}	MSE_{av}				
0.05	100	2×10^4	-4.781	-5.319	0.94	0.09		
0.02	150	2×10^5	-6.740	-7.221	89.31	7.37		
0.01	200	2×10^6	-8.830	-9.301	5056.16	414.44		

In order to assess the accuracy of the proposed algorithm, in the following Table 3, for $t = 0$, $q = 0$ and different values of N , K and M , 99% confidence intervals for $\bar{y}_i^{(q,M)}(0)$ are shown in the knowledge that $u(0,0) = 1.6$. These confidence intervals were computed using 50 independent runs of the algorithm. Since we are computing the solution at the origin, similar results are obtained for values of $q > 0$.

Table 3 $d = 1$, $i = 0$, 99% confidence intervals for $\bar{y}_i^{(q,M)}(0)$. $y_i^{(q)}(0) = 1.6$.

Δ	K	M	$q = 0$
0.05	100	2×10^4	[1.578, 1.667]
0.02	150	2×10^5	[1.595, 1.624]
0.01	300	2×10^7	[1.598, 1.612]

Now we present results in dimensions greater than one, for which the total degree and the hyperbolic cross index sets are different. Coherently with Section 2.4.3, we start showing the behaviour of the algorithm when the discussed conditions are enforced only on coordinate-wise derivatives of $h_i^{(q)}$. Table 4 shows results for $d = 2$. Convergence is clearly improved by increasing q , considering $q = 5.1$ ($p = r = 2$), not only $h_i^{(q)}$ and its first two coordinate-wise derivatives are square integrable but also $h_i^{(q)}$ and its first coordinate-wise derivatives vanish at the boundaries of $[0, 1]^2$. Additionally, convergence is enhanced as long as Δ decreases: DEG and the number of simulations M increase in a way such that $\frac{2^2 \# \Gamma}{M}$ decreases. Finally, the hyperbolic cross index set offers better performance than the total degree set, not only an improved rate of convergence but also much faster computational times. In fact, for $\Delta = 0.01$, the hyperbolic cross index set is not only almost 3.4 times faster than the total degree set (on GPU), but also more accurate achieving $MSE_{av} = -10.137$ instead of $MSE_{av} = -9.107$. The GPU speed-up with respect to a fully optimized CPU parallel version reaches the value 11.21.

Table 4 $d = 2$.

					$q = 0$		$q = 5.1$			
		DEG	$\# \Gamma(\cdot)$	M	MSE_{max}	MSE_{av}	MSE_{max}	MSE_{av}	CPU	GPU
Total.	0.05	20	231	2×10^4	-2.322	-2.773	-4.646	-5.239	3.24	0.33
	0.02	25	351	2×10^5	-2.869	-3.688	-6.761	-7.186	362.55	35.51
	0.01	31	528	2×10^6	-2.978	-3.966	-8.793	-9.107	25144.70	2243.06
Hyper.	0.05	19	99	2×10^4	-2.526	-3.062	-5.588	-6.196	1.43	0.16
	0.02	24	133	2×10^5	-2.745	-3.550	-7.487	-8.191	129.99	12.38
	0.01	30	172	2×10^6	-2.898	-3.770	-9.389	-10.137	7221.68	662.54

Tables 5 and 6 show results for $d = 3$ and 5, respectively. As before, the convergence of the algorithm improves for large values of the damping exponent, so that we benefit from small approximation error (see Proposition 4 with the conditions (2.43)-(2.48)). Furthermore, the hyperbolic cross index set outperforms the total degree set (which can be expected from the equation (2.63)). Except for the cases with $\Delta = 0.05$, where there are not enough simulations to fully take

advantage of four GPUs, the multi-GPU implementation provides a significant reduction in the computational time: the speed-up with respect to the mono-GPU version reaches the value 3.75. The ideal speed-up would be 4, although the multi-GPU parallelization introduces some overhead which makes this value unattainable.

Table 5 $d = 3$.

	Δ	DEG	$\#\Gamma(\cdot)$	M	$q = 0$		$q = 5.1$		GPU	MGPU
					MSE_{\max}	MSE_{av}	MSE_{\max}	MSE_{av}		
Total	0.05	22	2300	2×10^5	-2.164	-2.785	-5.699	-6.141	53.89	121.75
	0.02	24	2925	2×10^6	-2.423	-3.580	-7.693	-8.329	4361.22	1351.36
	0.01	26	3654	2×10^7	-2.484	-3.763	-9.836	-10.487	220687.47	58783.53
Hyper.	0.05	20	411	2×10^5	-2.255	-3.191	-6.883	-7.662	8.37	22.99
	0.02	22	459	2×10^6	-2.249	-3.356	-7.778	-8.875	592.19	191.59
	0.01	24	528	2×10^7	-2.248	-3.380	-8.387	-9.576	27317.65	7306.78

Table 6 $d = 5$.

	Δ	DEG	$\#\Gamma(\cdot)$	M	$q = 0$		$q = 5.1$		MGPU
					MSE_{\max}	MSE_{av}	MSE_{\max}	MSE_{av}	
Total	0.2	20	53130	2×10^6	-1.906	-2.049	-6.537	-6.994	874.36
	0.1	21	65780	2×10^7	-2.221	-3.356	-8.482	-9.103	19163.72
Hyper.	0.2	16	3042	2×10^6	-2.353	-3.259	-7.312	-7.835	49.27
	0.1	17	3122	2×10^7	-1.955	-3.278	-7.438	-8.125	773.70
	0.1	36	10503	2×10^7	-2.216	-3.494	-8.609	-9.407	2718.07
	0.1	66	24893	2×10^7	-2.381	-3.603	-8.819	-9.703	6563.77

Finally, results for $d = 7, 9$ and 11 are shown in Table 7 considering only the scenario performing better, i.e. the multi-GPU version considering a large value of the damping exponent along with the hyperbolic cross index set for the selection of the basis functions on which to project $y_i^{(q)}$.

Table 7 $d = 7, 9, 11$, hyperbolic cross index set.

d	Δ	DEG	$\#\Gamma_H(\text{DEG})$	M	$q = 5.1$		MGPU
					MSE_{\max}	MSE_{av}	
7	0.1	36	94460	2×10^7	-8.879	-9.574	33284.41
9	0.1	12	102656	2×10^7	-8.101	-8.359	44437.40
11	0.1	5	75264	2×10^8	-7.828	-8.785	373816.97

Before the end of this section, the behavior of the algorithm is exhibited using larger values of the damping exponent q that account for the case where not only coordinate-wise derivatives of $h_i^{(q)}$ but also its mixed derivatives are required to vanish at the boundaries of $[0, 1]^d$ and to be square integrable. Taking into account that

$$\partial_x^{(r, \dots, r)} y_i(\mathbf{x}) = \pm \lambda \exp\left(\frac{t-1}{2}\right) [\mathbf{1}_{dr \text{ even}} \sin(\mathbf{x}) + \mathbf{1}_{dr \text{ odd}} \cos(\mathbf{x})],$$

and under $\Lambda = \{0, \dots, r\}^d$, in order to fit the assumptions of Lemma 7 we select $p = dr$. In the following numerical experiments $r = 2$ or 1 will be selected. In order to avoid too large values of q , which will hinder the computation of the damping factor $(1 + |\mathbf{x}|^2)^{q/2}$ when single and even double precision are considered, the μ parameter of the ν_l Student's t-distribution is chosen as $\mu = 1$ from now on. Under this choice we deal with the Cauchy distribution $\nu_l(x_l) = \pi^{-1}(1 + x^2)^{-1}$, whose CDF and its inverse are given by $F_{\nu_l}(x) = 0.5 + \pi^{-1} \arctan(x)$ and $F_{\nu_l}^{-1}(u) = \tan(\pi x - 0.5\pi)$, respectively.

Tables 8 and 9 show the behavior of the algorithm using a value of q satisfying the condition (2.60) for $r = 2$, which boils down to $q > 2dr - 0.5$ in our benchmark scenario. As expected from equation (2.63) for the Fourier coefficients of $h_i^{(q)}$, the hyperbolic cross index set clearly outperforms the total degree set. In order to take the most of our runtime environment, the codes were executed taking advantage of the whole available GPUs and the computations were performed using single precision.

Table 8 $d = 2, \Lambda = \{0, 1, 2\}^2$.

						$q = 7.6$		
		Δ	DEG	$\#\Gamma(\text{DEG})$	M	MSE_{\max}	MSE_{av}	MGPU
Total	0.05	31	528	2×10^5		-7.453	-7.871	27.98
	0.02	41	903	2×10^6		-8.916	-9.598	297.99
	0.01	51	1378	2×10^7		-10.528	-11.416	13220.48
Hyper.	0.05	30	172	2×10^5		-7.710	-8.494	9.94
	0.02	40	239	2×10^6		-9.019	-10.152	73.66
	0.01	50	308	2×10^7		-10.425	-11.964	2646.07

Table 9 $d = 3, \Lambda = \{0, 1, 2\}^3$.

						$q = 11.6$		
		Δ	DEG	$\#\Gamma(\text{DEG})$	M	MSE_{\max}	MSE_{av}	MGPU
Tot.	0.2	142	497640	2×10^6		-5.009	-5.423	6950.81
	0.2	152	608685	2×10^7		-6.848	-7.085	27347.05
Hyper.	0.2	140	4881	2×10^6		-7.955	-8.693	68.73
	0.2	150	5322	2×10^7		-8.173	-9.039	205.92
	0.1	160	5763	2×10^8		-8.319	-9.494	8030.68

Ultimately, the semi-linear PDE (5.1) is solved for dimension 5 and 7 in Table 10, and for dimension 9 and 11 in Table 11. The GQRMDP algorithm was executed using double precision and considering only the hyperbolic cross index set for the projection of $y_i^{(q)}$. For $d = 5, 7$, the so-discussed conditions are imposed for $\Lambda = \{0, 1, 2\}^d$, while for $d = 9, 11$, in order to avoid larger values of q , these smoothness conditions are only forced for $\Lambda = \{0, 1\}^d$. The error indicators and the computational time for $d = 7$ inside Table 10 are to be compared with those of Table 7: as it was anticipated, for $q = 27.6$ the algorithm obtains much better convergence, whereas the execution time is larger due to more accurate computations on double precision.

Table 10 $d = 5, 7$, hyperbolic cross index set, $\Lambda = \{0, 1, 2\}^d$, double precision.

d	Δ	DEG	$\# \Gamma_H(\text{DEG})$	M	q	MSE_{\max}	MSE_{av}	MGPU
5	0.1	100	45443	2×10^7	19.6	-9.065	-10.198	16889.41
7	0.1	36	94460	2×10^7	27.6	-15.054	-16.381	48021.18

Table 11 $d = 9, 11$, hyperbolic cross index set, $\Lambda = \{0, 1\}^d$, double precision.

d	Δ	DEG	$\# \Gamma_H(\text{DEG})$	M	q	MSE_{\max}	MSE_{av}	MGPU
9	0.2	5	14336	2×10^7	17.6	-21.016	-21.852	2466.21
11	0.2	2	13312	2×10^7	21.6	-25.559	-27.161	2648.61

A Appendix. Technical estimates on Student's t-distribution

In this appendix, estimates on the CDF of the Student's t-distribution, its inverse and the derivatives of the inverse of the CDF are computed. As we have been doing in the article, we use the same notation C for any generic constants, we will keep the same C from line to line, in order to alleviate the computations, although its value changes. In this appendix ν_l stands for Student's t-distribution density (2.11) with parameter μ , F_{ν_l} its marginal CDF and $F_{\nu_l}^{-1}$ its inverse.

Lemma 11 (Estimates on F_{ν_l} and $F_{\nu_l}^{-1}$) *Let $x \in \mathbb{R}$, $u \in (0, 1)$. The following estimates hold*

$$\begin{aligned}
 F_{\nu_l}(x) &\sim_{x \rightarrow -\infty} \frac{c_\mu}{\mu|x|^\mu}, \\
 1 - F_{\nu_l}(x) &\sim_{x \rightarrow +\infty} \frac{c_\mu}{\mu|x|^\mu}, \\
 F_{\nu_l}^{-1}(u) &\sim_{u \rightarrow 0^+} -\tilde{c}_\mu u^{-1/\mu}, \\
 F_{\nu_l}^{-1}(u) &\sim_{u \rightarrow 1^-} \tilde{c}_\mu (1-u)^{-1/\mu},
 \end{aligned} \tag{A.1}$$

with $\tilde{c}_\mu = \left(\frac{c_\mu}{\mu}\right)^{1/\mu}$.

Proof In view of (2.11) we know that $\nu_l(x_l) \sim_{x_l \rightarrow \pm\infty} \frac{c_\mu}{|x_l|^{\mu+1}}$. Besides,

$$F_{\nu_l}(x) = \int_{-\infty}^x \nu_l(x_l) dx_l \sim_{x \rightarrow -\infty} \int_{-\infty}^x \frac{c_\mu}{|x_l|^{\mu+1}} dx_l = \frac{c_\mu}{\mu|x|^\mu}.$$

The distribution ν_l being symmetric, we have $F_{\nu_l}(x) + F_{\nu_l}(-x) = 1$ and $F_{\nu_l}^{-1}(u) = -F_{\nu_l}^{-1}(1-u)$. Therefore, the estimate $1 - F_{\nu_l}(x)$ for $x \rightarrow +\infty$ follows from the case $x \rightarrow -\infty$.

In order to compute $F_{\nu_l}^{-1}$ we use that $F_{\nu_l}(F_{\nu_l}^{-1}(u)) = u$: If $u \rightarrow 0^+$, $F_{\nu_l}^{-1}(u) \rightarrow -\infty$, $u = F_{\nu_l}(F_{\nu_l}^{-1}(u)) \sim_{u \rightarrow 0^+} \frac{c_\mu}{\mu|F_{\nu_l}^{-1}(u)|^\mu}$. Therefore,

$$|F_{\nu_l}^{-1}(u)|^\mu \sim_{u \rightarrow 0^+} \frac{c_\mu}{\mu u}, \quad |F_{\nu_l}^{-1}(u)| \underset{u \rightarrow 0^+}{=} -F_{\nu_l}^{-1}(u) \sim_{u \rightarrow 0^+} \left(\frac{c_\mu}{\mu u}\right)^{1/\mu},$$

which is the advertised result. The case $u \rightarrow 1^-$ follows from the case for $u \rightarrow 0^+$ using again the symmetry of the distribution ν_l . \square

Lemma 12 (Estimates on derivatives of $\frac{1}{(1+|\mathbf{x}|^2)^{q/2}}$) *Let $\mathbf{x} \in \mathbb{R}^d$, $q \in \mathbb{R}^+$ and $n \in \mathbb{N}$. The following upper bound holds*

$$\left| \partial_{x_l}^n \frac{1}{(1+|\mathbf{x}|^2)^{q/2}} \right| \leq C(1+|\mathbf{x}|^2)^{-q/2-n/2}. \tag{A.2}$$

Proof We only have to consider the case $n \geq 1$. In order to compute $\partial_{x_l}^n \frac{1}{(1+|\mathbf{x}|^2)^{q/2}}$ we use the following Faà di Bruno's formula (as long as the partial derivative is only with respect to one variable we can extend in the following way the one-dimensional Faà di Bruno's formula (2.34)):

$$\partial_{x_l}^n f(g(\mathbf{x})) = \sum_{m=1}^n \frac{1}{m!} d_u^m f(u)|_{u=g(\mathbf{x})} \sum_{\mathbf{j} \in J_{m,n}} \frac{n!}{j_1! j_2! \dots j_m!} \prod_{i=1}^m \partial_{x_l}^{j_i} g(\mathbf{x}), \quad (\text{A.3})$$

$$J_{m,n} = \{\mathbf{j} = (j_1, \dots, j_m) \in \mathbb{N}_+^m : j_1 + \dots + j_m = n\}.$$

Here, consider $f(u) = u^{-q/2}$ and $g(\mathbf{x}) = 1 + |\mathbf{x}|^2$: $d_u^m f(u) = C_{m,q} u^{-q/2-m}$ and $\partial_{x_l}^j g(\mathbf{x}) = 2x_l \mathbf{1}_{j=1} + 2\mathbf{1}_{j=2}$. Observe that in the Faà di Bruno's formula (A.3), the sum over $\mathbf{j} \in J_{m,n}$ will be made only for $j_i = 1$ with $m_1 := \#\{j_i : \mathbf{j} \in J_{m,n}, j_i = 1\}$ terms, and for $j_i = 2$ with $m_2 := \#\{j_i : \mathbf{j} \in J_{m,n}, j_i = 2\}$ terms. Owing to this property and by definition of $J_{m,n}$, we have $m_1 + 2m_2 = n$ and $m_1 + m_2 = m$, which results in $m_1 = 2m - n$ and $m_2 = n - m$. Invoking the form of the derivatives $\partial_{x_l}^{j_i} g(\mathbf{x})$, it readily follows that

$$\prod_{i=1}^m \partial_{x_l}^{j_i} g(\mathbf{x}) = 2^m x_l^{m_1}.$$

All in all, we deduce

$$\begin{aligned} \left| \partial_{x_l}^n \frac{1}{(1+|\mathbf{x}|^2)^{q/2}} \right| &\leq C \sum_{m=\lceil n/2 \rceil}^n (1+|\mathbf{x}|^2)^{-q/2-m} (1+|x_l|)^{2m-n} \\ &\leq C \sum_{m=\lceil n/2 \rceil}^n (1+|\mathbf{x}|^2)^{-q/2-m} (1+|\mathbf{x}|^2)^{m-n/2} \leq C(1+|\mathbf{x}|^2)^{-q/2-n/2}. \end{aligned}$$

□

Lemma 13 (Estimates on derivatives of $\frac{1}{(1+|\mathbf{x}|^2)^{q/2}}$) Let $\mathbf{x} \in \mathbb{R}^d$, $q \in \mathbb{R}^+$, $\mathbf{n} = (n_1, \dots, n_d) \in \mathbb{N}_+^d$. The following upper bound holds

$$\left| \partial_{\mathbf{x}}^{\mathbf{n}} \frac{1}{(1+|\mathbf{x}|^2)^{q/2}} \right| \leq C(1+|\mathbf{x}|^2)^{-q/2-\bar{\mathbf{n}}/2}. \quad (\text{A.4})$$

Proof We only have to consider the case $\bar{\mathbf{n}} \geq 1$. For such \mathbf{n} , we use the following multivariate Faà di Bruno's formula derived from [15, Corollary 2.10]

$$\partial_{\mathbf{x}}^{\mathbf{n}} f(g(\mathbf{x})) = \sum_{m=1}^{\bar{\mathbf{n}}} d_u^m f(u)|_{u=g(\mathbf{x})} \sum_{J \in \mathbf{J}_{m,\mathbf{n}}} C_{m,J} \prod_{i=1}^m \partial_{\mathbf{x}}^{\mathbf{j}_i} g(\mathbf{x}), \quad (\text{A.5})$$

$$\mathbf{J}_{m,\mathbf{n}} = \left\{ J = \begin{pmatrix} \mathbf{j}_1 \\ \vdots \\ \mathbf{j}_m \end{pmatrix} \in \mathbb{N}^m \times \mathbb{N}^d : \mathbf{j}_1 + \dots + \mathbf{j}_m = \mathbf{n}, \mathbf{j}_1, \dots, \mathbf{j}_m \neq \mathbf{0} \right\},$$

where the positive constants $C_{m,J}$ depend on m and the matrix J under consideration.

Note that non coordinate-wise derivatives of $g(\mathbf{x}) = 1 + |\mathbf{x}|^2$ are all zero. Besides, coordinate-wise derivatives of g are $\partial_{x_l}^j g(\mathbf{x}) = 2x_l \mathbf{1}_{j=1} + 2\mathbf{1}_{j=2}$. For a given J , let $a_l \in \mathbb{N} \forall l = 1, \dots, d$ denote the number of vectors in J with 1 in the l -th coordinate and zero anywhere else. It holds that

$$\begin{aligned} \prod_{i=1}^m \partial_{\mathbf{x}}^{\mathbf{j}_i} g(\mathbf{x}) &= \mathbf{1}_{2m \geq \bar{\mathbf{n}}} C x_1^{a_1} \dots x_d^{a_d} \text{ with } a_1 + \dots + a_d = 2m - \bar{\mathbf{n}}, \\ \left| \prod_{i=1}^m \partial_{\mathbf{x}}^{\mathbf{j}_i} g(\mathbf{x}) \right| &\leq C(1+|\mathbf{x}|^2)^{a_1/2} \dots (1+|\mathbf{x}|^2)^{a_d/2} \mathbf{1}_{\sum_{l=1}^d a_l = 2m - \bar{\mathbf{n}}} \leq C(1+|\mathbf{x}|^2)^{m - \bar{\mathbf{n}}/2} \end{aligned}$$

for $2m \geq \bar{n}$. All in all, we deduce

$$\left| \partial_x^n \frac{1}{(1+|\mathbf{x}|^2)^{q/2}} \right| \leq C \sum_{m=\lceil \bar{n}/2 \rceil}^{\bar{n}} (1+|\mathbf{x}|^2)^{-q/2-m} (1+|\mathbf{x}|^2)^{m-\bar{n}/2} \leq C(1+|\mathbf{x}|^2)^{-q/2-\bar{n}/2}.$$

□

Lemma 14 (Estimates on derivatives of Student's t-distribution) *Let $n \in \mathbb{N}$. The following upper bound holds*

$$|d_x^n \nu_l(x)| \leq C(1+x^2)^{-\frac{\mu+1}{2}-\frac{n}{2}}. \quad (\text{A.6})$$

Proof This result follows readily using the previous estimate (A.2). □

Lemma 15 (Estimates on derivatives of $F_{\nu_l}^{-1}$) *Let $n \in \mathbb{N}$. The following upper bounds hold*

$$\left| d_u^n F_{\nu_l}^{-1}(u) \right| \leq_{u \rightarrow 0^+} C u^{-\frac{\mu n+1}{\mu}}, \quad \left| d_u^n F_{\nu_l}^{-1}(u) \right| \leq_{u \rightarrow 1^-} C(1-u)^{-\frac{\mu n+1}{\mu}}, \quad (\text{A.7})$$

where C is a constant depending on μ and n .

Proof Using the combinatorial formula for higher derivatives of inverses [41] and the fact that for $b \in \mathbb{N}$, $b \geq 1$, $d_x^b F_{\nu_l}(x) = d_x^{b-1} \nu_l(x)$,

$$\begin{aligned} d_u^{n+1} F_{\nu_l}^{-1}(u) &= \sum_{k=0}^n \frac{(-1)^k}{k!} [\nu_l(x)|_{x=F_{\nu_l}^{-1}(u)}]^{-n-k-1} \\ &\quad \times \sum_{\substack{b_1+\dots+b_k=n+k \\ b_i \geq 2}} \frac{(n+k)!}{b_1! b_2! \dots b_k!} \prod_{i=1}^k d_x^{b_i-1} \nu_l(x)|_{x=F_{\nu_l}^{-1}(u)}. \end{aligned}$$

Therefore, using (A.6),

$$\begin{aligned} |d_u^{n+1} F_{\nu_l}^{-1}(u)| &\leq \sum_{k=0}^n C \left[\left(1 + \left(F_{\nu_l}^{-1}(u) \right)^2 \right)^{-\frac{\mu+1}{2}} \right]^{-n-k-1} \\ &\quad \times \sum_{\substack{b_1+\dots+b_k=n+k \\ b_i \geq 2}} \prod_{i=1}^k \left(1 + \left(F_{\nu_l}^{-1}(u) \right)^2 \right)^{-\frac{\mu+1}{2}-\frac{b_i-1}{2}} \\ &\leq \sum_{k=0}^n C \left(1 + \left(F_{\nu_l}^{-1}(u) \right)^2 \right)^{\frac{\mu+1}{2}(n+k+1) - \frac{k(\mu+1)+n}{2}} \\ &\leq C \left(1 + \left(F_{\nu_l}^{-1}(u) \right)^2 \right)^{\frac{\mu(n+1)+1}{2}}. \end{aligned}$$

Finally, combining the previous upper bound with (A.1) yields

$$\begin{aligned} \left| d_u^n F_{\nu_l}^{-1}(u) \right| \leq_{u \rightarrow 0^+} C \left| F_{\nu_l}^{-1}(u) \right|^{\mu n+1} \leq_{u \rightarrow 0^+} C u^{-\frac{\mu n+1}{\mu}}, \\ \left| d_u^n F_{\nu_l}^{-1}(u) \right| \leq_{u \rightarrow 1^-} C(1-u)^{-\frac{\mu n+1}{\mu}}, \end{aligned}$$

where the constant C may change value along computations. □

Compliance with Ethical Standards

- Conflict of Interest: The authors declare that they have no conflict of interest.

References

1. Bally, V.: Approximation scheme for solutions of BSDE. In: Backward stochastic differential equations (Paris, 1995–1996), *Pitman Res. Notes Math. Ser.*, vol. 364, pp. 177–191. Longman, Harlow (1997)
2. Bally, V., Pagès, G.: A quantization algorithm for solving multi-dimensional optimal stopping problems. *Bernoulli* **9**(6), 1003–1049 (2003)
3. Barles, G., Buckdahn, R., Pardoux, E.: Backward stochastic differential equations and integral-partial differential equations. *Stochastics and Stochastics Reports* **60**(1-2), 57–83 (1997)
4. Bellmann, R.: Adaptive control processes: A guided tour. Princeton University Press (1961)
5. Bender, C., Denk, R.: A forward scheme for backward SDEs. *Stochastic Processes and their Applications* **117**(12), 1793–1823 (2007)
6. Bender, C., Moseler, T.: Importance sampling for backward SDEs. *Stochastic Analysis and Applications* **28**(2), 226–253 (2010)
7. Bergman, Y.: Option pricing with differential interest rates. *Rev. of Financial Studies* **8**(2), 475–500 (1995)
8. Bismut, J.M.: Conjugate convex functions in optimal stochastic control. *Journal of Mathematical Analysis and Applications* **44**(2), 384–404 (1973)
9. Bouchard, B., Tan, X., Warin, X., Zou, Y.: Numerical approximation of BSDEs using local polynomial drivers and branching processes. *Monte Carlo Methods and Applications* **23**(4), 241–263 (2017)
10. Bouchard, B., Touzi, N.: Discrete time approximation and Monte-Carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Applications* **111**, 175–206 (2004)
11. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: Spectral Methods, Fundamentals in Single Domains. Springer, Berlin (2006)
12. Chan-Wai-Nam, Q., Mikael, J., Warin, X.: Machine learning for semi linear PDEs. *Journal of Scientific Computing* pp. 1–46 (2019)
13. Cheridito, P., Soner, H.M., Touzi, N., Victoir, N.: Second-order backward stochastic differential equations and fully nonlinear parabolic PDEs. *Communications on Pure and Applied Mathematics* **60**(7), 1081–1110 (2007)
14. Chevance, D.: Numerical methods for backward stochastic differential equations. In: Numerical methods in finance, Publ. Newton Inst., pp. 232–244. Cambridge Univ. Press, Cambridge (1997)
15. Constantine, G.M., Savits, T.H.: A multivariate Faà di Bruno formula with applications. *Transactions of the American Mathematical Society* **348**(2), 503–520 (1996)
16. Crepey, S.: Financial modeling: a Backward Stochastic Differential Equations perspective. Springer. Springer-Verlag, Berlin (2013)
17. NVIDIA cuRAND web page: <https://developer.nvidia.com/curand>
18. Cvitanic, J., Ma, J.: Hedging options for a large investor and Forward-Backward SDEs. *The Annals of Applied Probability* **6**(2), 370–398 (1996)
19. Delarue, F., Menozzi, S.: A forward backward algorithm for quasi-linear PDEs. *Annals of Applied Probability* **16**(1), 140–184 (2006)
20. Delarue, F., Menozzi, S.: An interpolated stochastic algorithm for quasi-linear PDEs. *Math. Comput.* **77**(261), 125–158 (2008)
21. Delong, L.: Backward Stochastic Differential Equations with Jumps and Their Actuarial and Financial Applications. BSDEs with Jumps. Springer (2013)
22. Di Pietro, D., Ern, A.: Mathematical aspects of discontinuous Galerkin methods, vol. 69. Springer Science & Business Media (2011)
23. Duffie, D., Epstein, L.: Stochastic differential utility. *Econometrica* **60**(2), 353–394 (1992)
24. E, W., Han, J., Jentzen, A.: Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics* **5**(4), 349–380 (2017)
25. El Karoui, N., Hamadène, S., Matoussi, A.: Backward stochastic differential equations and applications. In: R. Carmona (ed.) Indifference pricing: theory and applications, chap. 8, pp. 267–320. Springer-Verlag (2008)
26. El Karoui, N., Kapoudjian, C., Pardoux, E., Peng, S., Quenez, M.: Reflected solutions of BSDEs and related obstacle problems for PDE’s. *Annals of Probability* **25**(2), 702–737 (1997)

27. El Karoui, N., Peng, S., Quenez, M.: Backward stochastic differential equations in finance. *Math. Finance* **7**(1), 1–71 (1997)
28. El Karoui, N., Quenez, M.: Dynamic programming and pricing of contingent claims in an incomplete market. *SIAM J. Control Optim.* **33**(1), 29–66 (1995)
29. Funaro, D.: *Polynomial Approximation of Differential Equations*. Lecture Notes in Physics. Monographs 8. Springer, Berlin (1992)
30. Gavalas, G.: *Nonlinear differential equations of chemically reacting systems*. Springer Tracts in Natural Philosophy, Vol. 17. Springer-Verlag New York Inc., New York (1968)
31. Geiss, C., Geiss, S., Gobet, E.: Generalized fractional smoothness and L_p -variation of BSDEs with non-Lipschitz terminal condition. *Stochastic Processes and their Applications* **122**(5), 2078–2116 (2012)
32. Gobet, E.: *Monte-Carlo methods and stochastic processes: from linear to non-linear*. CRC Press (2016)
33. Gobet, E., Lemor, J.P., Warin, X.: A regression-based Monte-Carlo method to solve backward stochastic differential equations. *Annals of Applied Probability* **15**(3), 2172–2202 (2005)
34. Gobet, E., López-Salas, J.G., Turkedjiev, P., Vázquez, C.: Stratified Regression Monte-Carlo Scheme for Semilinear PDEs and BSDEs with Large Scale Parallelization on GPUs. *SIAM Journal on Scientific Computing* **38**(6), C652–C677 (2016)
35. Gobet, E., Turkedjiev, P.: Approximation of backward stochastic differential equations using Malliavin weights and least-squares regression. *Bernoulli* **22**(1), 530–562 (2016)
36. Gobet, E., Turkedjiev, P.: Linear regression MDP scheme for discrete backward stochastic differential equations under general conditions. *Mathematics of Computation* **85**(299), 1359–1391 (2016)
37. Gobet, E., Turkedjiev, P.: Adaptive importance sampling in least-squares Monte-Carlo algorithms for backward stochastic differential equations. *Stochastic Processes and their Applications* **127**(4), 1171–1203 (2017)
38. Henry, D.: *Geometric theory of semilinear parabolic equations*, *Lecture Notes in Mathematics*, vol. 840. Springer-Verlag, Berlin (1981)
39. Henry-Labordere, P., Tan, X., Touzi, N.: A numerical algorithm for a class of BSDEs via the branching process. *Stochastic Processes and their Applications* **124**(2), 1112–1140 (2014)
40. Jackman, S.: *Bayesian Analysis for the Social Sciences*. Wiley Series In Probability and Statistics. John Wiley & Sons, Sussex, United Kingdom (2009)
41. Johnson, W.: Combinatorics of Higher Derivatives of Inverses. *The American Mathematical Monthly* **109**(3), 273–277 (2002)
42. Johnson, W.: The curious history of Faà di Bruno’s formula. *The American Mathematical Monthly* **109**(3), 217–234 (2002)
43. Kloeden, P.E., Platen, E.: *Numerical Solution of Stochastic Differential Equations*. Springer (1992)
44. Krommer, A.R., Ueberhuber, C.W.: *Computational Integration*. SIAM, Philadelphia (1998)
45. L’Ecuyer, P.: Good parameters and implementations for combined multiple recursive random number generators. *Operations Research* **47**(1), 159–164 (1999)
46. Lee, V.W., Kim, C., Chhugani, J., Deisher, M., Kim, D., Nguyen, A.D., Satish, N., Smelyanskiy, M., Chennupaty, S., Hammarlund, P., Singhal, R., Dubey, P.: Debunking the 100X GPU vs. CPU Myth: An Evaluation of Throughput Computing on CPU and GPU. *Newsletter ACM SIGARCH Computer Architecture News - ISCA ’10* **38**, 451–460 (2010)
47. Lemor, J.P., Gobet, E., Warin, X.: Rate of convergence of an empirical regression method for solving generalized backward stochastic differential equations. *Bernoulli* **12**(5), 889–916 (2006)
48. Ma, J., Protter, P., Yong, J.: Solving forward-backward stochastic differential equations explicitly - a four step scheme. *Probability Theory and Related Fields* **98**(3), 339–359 (1994)
49. Menozzi, S., Lemaire, V.: On some non asymptotic bounds for the Euler scheme. *Electron. J. Probab.* **15**, no. 53, 1645–1681 (2010)
50. Nevai, P.: Géza Freud, orthogonal polynomials and Christoffel functions. a case study. *Journal of Approximation Theory* **48**(1), 3–167 (1986)
51. Nvidia Corporation: *NVIDIA’s Next Generation CUDA Compute Architecture: Kepler GK110/210*

52. OpenMP web page: <http://www.openmp.org>
53. Pardoux, E., Peng, S.: Adapted solution of a backward stochastic differential equation. *Systems Control Lett.* **14**(1), 55–61 (1990)
54. Pardoux, E., Peng, S.: Backward stochastic differential equations and quasilinear parabolic partial differential equations. In: *Stochastic Partial Differential Equations and Their Applications, Lecture Notes in Control and Information Sciences*, vol. 176, pp. 200–217. Springer (1992)
55. Pardoux, E., Rascanu, A.: *Stochastic Differential Equations, Backward SDEs, Partial Differential Equations, Stochastic Modelling and Applied Probability*, vol. 69. Springer-Verlag (2014)
56. Peng, S.: Nonlinear expectations, nonlinear evaluations and risk measures. In: *Lectures presented in CIME EMS Bressanone, Italy July 6 13, 2003* (2003)
57. Pham, H.: Continuous-time stochastic control and optimization with financial applications, *Series Stochastic Modeling and Applied Probability*, vol. 61. Springer (2009)
58. Shigesada, N., Kawasaki, K. (eds.): *Biological invasions : theory and practice*. Oxford Series in Ecology and Evolution, Oxford : Oxford University Press (1997)
59. Smoller, J.: Shock waves and reaction-diffusion equations, *Fundamental Principles of Mathematical Sciences*, vol. 258, second edn. Springer-Verlag, New York (1994)
60. Soner, H.M., Touzi, N., Zhang, J.: Well-posedness of second order backward SDEs. *Probability Theory and Related Fields* **153**(1-2), 149–190 (2012)
61. NVIDIA Thrust web page: <https://developer.nvidia.com/thrust>
62. Visintin, A.: *Models of phase transitions. Progress in Nonlinear Differential Equations and their Applications*, 28. Birkhäuser Boston Inc., Boston, MA (1996)