Probabilistic numerical approximation schemes in finance : learning methods for high-dimensional BSDEs and unbiased Monte Carlo algorithms for stochastic volatility models

Junchao Chen

▶ **To cite this version:**

Junchao Chen. Probabilistic numerical approximation schemes in finance : learning methods for high-dimensional BSDEs and unbiased Monte Carlo algorithms for stochastic volatility models. Probability [math.PR]. Université Paris Cité, 2022. English. NNT : 2022UNIP7056 . tel-04206865

HAL Id: tel-04206865
https://theses.hal.science/tel-04206865

Submitted on 14 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Paris

École doctorale de Sciences Mathématiques de Paris Centre (386)

Laboratoire de Probabilités, Statistique et Modélisation (LPSM, UMR 8001)

Schémas d'approximation numérique probabiliste en finance: méthodes d'apprentissage pour les EDSRs de grande dimension et algorithmes de Monte Carlo sans biais pour des modèles à volatilité stochastique

Probabilistic numerical approximation schemes in finance: learning methods for high-dimensional BSDEs and unbiased Monte Carlo algorithms for stochastic volatility models

Par **Junchao CHEN**

Thèse de doctorat de **Mathématiques Appliquées**

Dirigée par **Jean-François Chassagneux**
Et par **Noufel Frikha**

Présentée et soutenue publiquement le **24 Mars 2022**

Devant un jury composé de:

Jean-François Chassagneux	Prof. à Université de Paris	Directeur
Noufel Frikha	MCF à Université de Paris	Directeur
Pierre Henry-Labordère	HDR, Natixis	Rapporteur
Céline Labart	MCF à Université Savoie Mont-Blanc	Examinatrice
Huyên Pham	Prof. à Université de Paris	Président
Christoph Reisinger	Prof. à Université d'Oxford	Examinateur
Xiaolu Tan	Prof. à CUHK	Rapporteur
Xavier Warin	Personnalité extérieure à EDF R&D	Membre invité

Acknowledgements

First of all, I would like to warmly thank my thesis supervisors, Jean-François Chassagneux and Noufel Frikha, for introducing me to the world of research in financial mathematics, without their help this work would not have seen the light of today. I am particularly grateful to them for their inspiring vision and helpful advices during the past three years. Their seriousness and rigor on mathematics research, and also the optimistic attitude to life have given me a lot, it will be an important part in my whole life.

All my gratitude also goes to Chao Zhou. Our collaboration contributed to part of this manuscript. I am also grateful to him for the rewarding stays in Suzhou Research Institute of NUS. And I am honored to join the Mathematics Discussion Group in NUS to communicate with his PhD students, such as Xizhi Su, Gang Guo, Weiwei Zhang.

I would like to express my thanks to M. Pierre Henry-Labordère and M. Xiaolu Tan who agreed to report this thesis. I am very honored by their valuable comments. I would also like to thank Mme Céline Labart, M. Christoph Reisinger, M. Huyên Pham, M. Xavier Warin, for having accepted to be part of the defense jury.

Then I would like to thank the whole Laboratory of LPSM, in particular the doctoral students and the Financial and Actuarial Mathematics, Numerical Probability team for welcoming me during these three years, always in a good mood. Especially thanks to Huyên Pham, Claudio Fontana, Zorana Grbac, Noufel Frikha for the nice courses when I studied in M2MO and some of my good friends in the Lab, such as Ashaaf, Aaroan, Azar, Barbara, Benjamin, Bohdan, Clément, Côme, Cyril, Enzo, Fabio, Guillaume C., Guillaume S., Hiroshi, Hoang-Dung, Houzhi, Ibrahim, Laure, Luca, Lucas, Marc, Maximilien, Mohan, Motto, Nathan, Nisrine, Simon, Sothea, Sylvain, William, Xiaoli, Xuanye, Yann, Yating, Yiyang and Ziad and so on. Also thanks to Nathalie and Valérie for accompanying us in the administrative problems that we encounter on a daily basis.

Thanks to Pr. Quanhua Xu who organised a program between Wuhan University and University of Besançon with the funding by China Scholarship Council, so that I have a chance to study in France. I would also like to thank other professors in Wuhan University and University of Besançon.

Thanks to Kuang Huang who is my classmate in Wuhan University and now a PhD student in Columbia University. We discussed a lot about academic questions and given me some useful ideas in the past years.

Special thanks to my parents in China, their trust and unconditional support during my many years of study allowed me to achieve what I am today.

Abstract

In this thesis, we propose some probabilistic numerical approximation in finance. Including a learning scheme by sparse grids and Picard approximations for semi-linear parabolic PDEs, high-order approximation for high-dimensional BSDEs by deep learning methods, and probabilistic representation of integration by parts formulae for some stochastic volatility models with unbounded drift.

In the first part of this thesis, we rely on the classical connection between Backward Stochastic Differential Equations (BSDEs) and non-linear parabolic partial differential equations (PDEs), to propose a new probabilistic learning scheme for solving high-dimensional semi-linear parabolic PDEs. This scheme is inspired by the approach coming from machine learning and developed using deep neural networks in Han et al. [56]. However, our algorithm is based on a Picard iteration scheme in which a sequence of linear-quadratic optimisation problem is solved by means of stochastic gradient descent (SGD) algorithm. In the framework of a linear specification of the approximation space, we manage to prove a convergence result for our scheme, under some smallness condition.

In practice, in order to be able to treat high-dimensional examples, we employ sparse grid approximation spaces. In the case of periodic coefficients and using pre-wavelet basis functions, we obtain an upper bound on the global complexity of our method. It shows in particular that the curse of dimensionality is tamed in the sense that in order to achieve a root mean squared error of order ε, for a prescribed precision ε, the complexity of the Picard algorithm grows polynomially in ε^{-1} up to some logarithmic factor $|\log(\varepsilon)|$ whose exponent grows linearly with respect to the PDE dimension. Various numerical results are presented to validate the performance of our method and to compare them with some recent machine learning schemes proposed in Han et al. [36] and Huré et al. [65].

Deep learning techniques are efficient techniques to overcome empirically the curse of dimensionality when solving high-dimensional backward stochastic differential equations (BSDEs). The current deep learning algorithms are based on an Euler type discretization. In the second work, we instead combine some high-order time discretization schemes such as Crank-Nicolson scheme or explicit multi stage Runge-Kutta scheme with non-linear regression. We prove theoretical convergence bounds for our algorithms. We then numerically compare the computational time cost of different methods and show that high order scheme for the discrete time error, if correctly implemented, are more efficient than classical Euler schemes.

In the second part, we establish a probabilistic representation as well as some integration by parts formulae for the marginal law at a given time maturity of some

stochastic volatility model with unbounded drift. Relying on a perturbation technique for Markov semigroups, our formulae are based on a simple Markov chain evolving on a random time grid for which we develop a tailor-made Malliavin calculus. Among other applications, an unbiased Monte Carlo path simulation method stems from our formulas so that it can be used in order to numerically compute with optimal complexity option prices as well as their sensitivities with respect to the initial values or *Greeks* in finance, namely the *Delta* and *Vega*, for a large class of non-smooth European payoff. Numerical results are proposed to illustrate the efficiency of the method.

Keywords: BSDEs, Semi-linear PDEs, Sparse grids, SGD algorithm, Deep learning, High-dimensional, High-order approximation, Probabilistic representation, Stochastic volatility model, Monte Carlo method.

Résumé

Dans la première partie, nous analysons en détail la convergence théorique de la solution des équations différentielles stochastiques rétrogrades (EDSRs) et des applications numériques dans le domaine de la finance avec à la fois l'algorithme SGD traditionnel et la méthode d'apprentissage en profondeur. Les méthodes sont basées sur la connexion classique entre les équations aux dérivées partielles (EDPs) paraboliques non linéaires et les EDSRs. Et de nombreux résultats numériques sur les EDSR de grande dimension sont présentés pour comparaison avec les articles [65, 36].

Dans le chapitre 2, nous introduisons ici un algorithme dont on montre qu'il converge vers un minimum global. Tout d'abord, nous passons de l'espace d'approximation des réseaux de neurones profonds à une spécification linéaire plus classique de l'espace d'approximation. Cependant, en raison de la non-linéarité du générateur f, le problème d'optimisation globale à résoudre est toujours non convexe. Pour contourner ce problème, nous utilisons une *procédure d'itération Picard*. La procédure globale devient alors une séquence de problèmes d'optimisation linéaire-quadratique qui sont résolus par un algorithme SGD. Notre premier résultat principal est un contrôle de l'erreur globale entre l'*algorithme implémenté* et la solution de la EDSR qui montre notamment la convergence de la méthode sous certaines conditions de petitesse, voir le Théorème 2.2.1. En particulier, contrairement à [56, 58] ou [65], notre résultat prend en compte l'erreur induite par l'algorithme SGD. Dans nos expériences numériques, nous nous appuyons sur des espaces d'approximation de sparse grid qui sont connus pour être bien adaptés pour traiter des problèmes de grande dimension. Dans le cadre des coefficients périodiques, nous établissons comme deuxième résultat principal, une borne supérieure sur la complexité globale pour notre *algorithme implémenté*, voir le Théorème 2.3.1. Nous montrons notamment que la malédiction de la dimensionnalité est apprivoisée dans le sens où la complexité est d'ordre $\varepsilon^{-p}|\log(\varepsilon)|^{q(d)}$, où p est un constante qui ne dépend pas de la dimension PDE et $d \mapsto q(d)$ est une fonction affine. Nous démontrons également numériquement l'efficacité de nos méthodes dans un cadre de grande dimension.

Dans le chapitre 3, nous rappelons d'abord la définition des schémas de Runge-Kutta pour les EDSR dans Section 3.2, puis nous étudions la stabilité des schémas de Runge-Kutta de deux manières différentes. Le Theorem 3.2.1 donne les erreurs en temps discret de 5 méthodes différentes qui seront étudiées dans ce chapitre. Dans Section 3.3, nous présentons une implémentation des schémas de Runge-Kutta pour résoudre les EDSR par réseaux de neurones, y compris le cas particulier des schémas d'Euler implicites [65], schéma d'Euler explicite, schéma de Crank-Nicolson, schéma de Runge-Kutta explicite en deux étapes. Nous fournissons le contrôle d'erreur de

la méthode d'apprentissage générale par le schéma de Runge-Kutta et le réseau de neurones à la fin de cette section, voir le Theorem 3.3.2. Dans Section 3.4, nous vérifions numériquement l'ordre de convergence de l'erreur en temps discret des 5 méthodes du Theorem 3.2.1, et nous comparons également le coût du temps de calcul de ces méthodes.

Dans la deuxième partie, nous présentons des formules de représentation probabilistes pour la loi marginale d'un modèles à volatilité stochastique à dérive non bornée. Nous établissons également des formules d'intégration par partie pour les *Delta* et *Vega*. Ces formules sont basées sur une chaîne de Markov évoluant le long d'une grille temporelle aléatoire donnée par les instants de saut d'un processus de renouvellement. Une méthode de Monte Carlo sans biais de complexité optimale découle de nos formules. La principale nouveauté de notre approche par rapport aux travaux est que nous permettons au coefficient de dérive d'être éventuellement non borné comme c'est le cas dans la plupart des modèles de volatilité stochastique (Stein-Stein, Heston, ...).

Mots clé: EDSRs, EDPs semi-linéaires, Sparse grids, Algorithme SGD, Deep learning, Grande dimension, Schémas de Runge-Kutta, Représentation probabiliste, Modèle de volatilité stochastique, Méthode de Monte Carlo.

Contents

List of Figures

List of Tables

Résumé détaillé

Ce manuscrit étudie les solutions des équations différentielles stochastiques rétrogrades (EDSRs) par différentes méthodes d'apprentissage et la représentation probabiliste des modèles de volatilité stochastique avec dérive non bornée.

Partie I: Méthodes d'apprentissage statistique pour EDSRs de grande dimension (Chapitres 2 et 3)

Soit $\mathcal{W}$ un mouvement brownien d-dimensionnel défini sur un espace de probabilité complet $(\mathfrak{O}, \mathcal{A}, \mathbb{P})$ et soit $\mathcal{X}_0$ un d-dimensionnel vecteur aléatoire $\mathcal{A}$-mesurable ayant support compact ou déterministe, indépendant de $\mathcal{W}$. Nous définissons $(\mathcal{F}_t)_{0\leqslant t\leqslant T}$ comme la filtration augmentée générée par $\mathcal{W}$ et $\mathcal{X}_0$. Pour $b : \mathbb{R}^d \to \mathbb{R}^d$ et $\sigma : \mathbb{R}^d \to \mathbb{M}_d$ (l'ensemble des matrices $d \times d$) deux fonctions mesurables, Nous définisons le processus de diffusion forward $\mathcal{X}$ comme la solution de l'équation différentielle stochastique suivante (EDS en abrégé)

$$\mathrm{d}\mathcal{X}_t = b(\mathcal{X}_t)\,\mathrm{d}t + \sigma(\mathcal{X}_t)\,\mathrm{d}\mathcal{W}_t\,, \tag{0.0.1}$$

et on définit son générateur infinitésimal $\mathcal{L}$, pour φ suffisamment reguliée, par

$$\mathcal{L}\varphi(t,x) := b(x)\cdot\nabla_x\varphi(t,x) + \frac{1}{2}\mathrm{Tr}[(\sigma\sigma^\top)(x)\nabla_x^2\varphi(t,x)]. \tag{0.0.2}$$

Dans ce manuscrit, nous nous intéressons dans la première partie à l'approximation numérique de la solution $(u(t,\mathcal{X}_t), \sigma^\top(\mathcal{X}_t)\nabla_x u(t,\mathcal{X}_t))$, où u est la solution de l'EDP parabolique semi-linéaire de grande dimension

$$\begin{cases} \partial_t u(t,x) + \mathcal{L}u(t,x) + f(u(t,x), \sigma^\top(x)\nabla_x u(t,x)) = 0, \quad (t,x)\in[0,T)\times\mathbb{R}^d, \\ u(T,x) = g(x), \quad x\in\mathbb{R}^d \end{cases} \tag{0.0.3}$$

et la solution $(\mathcal{F}_t)$-adaptée $(\mathcal{Y}_t, \mathcal{Z}_t) \in \mathbb{R}\times\mathbb{R}^d$ à l'EDSR

$$\mathcal{Y}_t = g(\mathcal{X}_T) + \int_t^T f(\mathcal{Y}_s, \mathcal{Z}_s)\,\mathrm{d}s - \int_t^T \mathcal{Z}_s\cdot\,\mathrm{d}\mathcal{W}_s\,,\ 0\leqslant t\leqslant T\,, \tag{0.0.4}$$

où $f\ : \mathbb{R}\times\mathbb{R}^d \to \mathbb{R}$, $g\ : \mathbb{R}^d \to \mathbb{R}$ sont des fonctions mesurables.

Il a été remarqué pour la première fois dans [81] qu'il existe une connexion entre les EDPs paraboliques semi-linéaires de la forme (0.0.3) et les EDSRs (0.0.4) comme suit:

$$(u(t,\mathcal{X}_t), \sigma^\top(\mathcal{X}_t)\nabla_x u(t,\mathcal{X}_t)) = (\mathcal{Y}_t, \mathcal{Z}_t), 0\leqslant t\leqslant T.$$

Cela montre que résoudre l'EDP (0.0.3) est équivalent à résoudre l'EDSR (0.0.4).

Depuis, l'approximation numérique de (0.0.4) a été largement étudiée à travers la recherche d'algorithmes numériques efficaces. En particulier, les méthodes de branchement [59], les méthodes de Picard multiniveaux récursive en histoire complète [66], les méthodes de cubature [25, 31, 32], les méthodes de quantification optimales [6, 5, 80, 77], les méthodes basées sur le calcul de Malliavin [33, 17, 64] et certaines méthodes de régression linéaire [49, 50, 51] ont été considérées. Il est reconnu que de telles approches seront réalisables pour des problèmes jusqu'à la dimension 10. Résoudre des EDSRs non linéaires de grande dimension est une tâche difficile en raison du "fléau de la dimension". Une avancée majeure est que la technique d'apprentissage automatique (en particulier à l'aide de réseaux de neurones profonds) a été appliquée à ce domaine par Weinan E, Jiequn Han, Arnulf Jentzen et Christian Beck en 2017 [36]. Dès lors, les EDSRs à 100 dimensions sont devenues résolubles. Ensuite, les performances des expériences numériques ont été améliorées par de nombreux nouveaux algorithmes basés sur le réseau de neurones, voir [56, 57, 58, 21, 65, 46, 85, 86, 47, 67, 68, 90, 28]. Et Teng Long a proposé un algorithme numérique basé sur l'amplification du gradient en 2021 [91] qui pourrait résoudre des EDSRs non linéaires à 10000 dimensions.

Dans la première partie de cette thèse, nous analysons en détail la convergence théorique de certaines approximations numériques, c'est-à-dire l'algorithme traditionnel de descente de gradient stochastique (SGD en anglais) et les méthodes d'apprentissage profond, de la solution à une équation différentielle stochastique rétrograde. Les algorithmes sont testés sur plusieurs exemples issus de la finance mathématique. Les approximations sont basées sur la connexion classique entre les EDPs paraboliques non linéaires et les EDSRs. Les résultats numériques sur les EDSRs de grande dimension sont comparés à ceux obtenus dans [65, 36].

Dans le Chapitre 2, nous introduisons un algorithme pour approximer les solutions en utilisant des fonctions de base sur certains espaces de grille clairsemée et il est montré qu'il converge vers un minimum global. Le problème d'optimisation global à résoudre est non convexe en raison du générateur non linéaire des EDSRs. Afin de contourner ce problème, nous utilisons une *procédure d'itération Picard*. Ensuite, la procédure globale devient une suite de problèmes d'optimisation linéaire-quadratique qui peuvent être résolus par l'algorithme SGD. Dans le Théorème 2.2.1, nous contrôlons l'erreur globale de l'*algorithme implémenté* qui montre la convergence de l'algorithme sous certaines conditions. En particulier, contrairement aux méthodes d'apprentissage profond [56, 58, 65], Nous obtenons une borne supérieure sur l'erreur induite par l'algorithme SGD. En pratique, nous nous appuyons sur des espaces d'approximation de grille clairsemée pour faire face au "fléau de la dimension". Notre deuxième résultat principal est le Théorème 2.3.1 qui montre, sous une hypothèse de coefficients périodiques, une borne supérieure pour la complexité globale de l'*algorithme implémenté*. En particulier, nous prouvons que le "fléau de la dimension" est apprivoisé, dans le sens où nous obtiensons une approximation avec une erreur inférieure à $\varepsilon > 0$ avec une complexité d'ordre $\varepsilon^{-p}|\ log(\varepsilon)|^{q(d)}$, où p est une constante et $q(d)$ est une fonction affine. Enfin, divers exemples en grande dimension montrent numériquement l'efficacité de nos méthodes, voir Section 2.3.

Soit une grille équidistante $\pi := \{t_n = nh, n = 0, \cdots N | h := T/N\}$ de $[0, T]$, nous

utilisons l'espace des grilles clairsemées afin d'approximer les solutions $(u(t_n,\cdot), \partial_x u(t_n,\cdot))$ de l'EDP parabolique semi-linéaire (0.0.3) au temps t_n. L'algorithme est appliqué comme algorithme d'apprentissage du gradient pour optimiser les coefficients des fonctions de base, et un schéma d'itération de Picard est introduit pour que l'algorithme converge vers la solution globale.

L'objectif principal est de calculer une approximation de $u(0,\mathcal{X}_0)$, où u est la solution de l'EDP (0.0.3) à l'instant initial sur un domaine donné ou en un point précis. Cela nous a conduit à introduire la configuration suivante pour la valeur initiale $\mathcal{X}_0$:

Assumption 0.0.1 *L'un des deux cas suivants est valable:*

(i) La loi de $\mathcal{X}_0$ est à support compact et absolument continue par rapport à la mesure de Lebesgue.

(ii) La loi de $\mathcal{X}_0$ est une masse de Dirac en un point $x_0 \in \mathbb{R}^d$.

Soit $W := (\mathcal{W}_{t_n})_{0\leqslant n\leqslant N}$ la version en temps discrets du mouvement brownien $\mathcal{W}$. On définit $\Delta W_n = W_{t_{n+1}} - W_{t_n}$, $0 \leqslant n \leqslant N-1$ et

$$X_0 = \mathcal{X}_0, \quad X_{t_{n+1}} = X_{t_n} + b(X_{t_n})h + \sigma(X_{t_n})\Delta W_n, \quad 0 \leqslant n \leqslant N-1\,. \tag{0.0.5}$$

Nous introduisons maintenant une approximation en temps discrets du processus $\mathcal{Z}$ et $\mathcal{Y}$. Pour un $t_n \in \pi\backslash\{T\}$ donné, supposons que $\mathscr{V}_n^z$ est un espace d'approximation fonctionnelle paramétrique généré par un ensemble de fonctions de base

$$(\psi_n^k(x))_{1\leqslant k\leqslant K_n^z}, \quad x \in \mathbb{R}^d \tag{0.0.6}$$

avec un entier positif K_n^z. Pour une utilisation ultérieure, nous définissons:

$$\bar{K}^z := \sum_{n=0}^{N-1} K_n^z\,. \tag{0.0.7}$$

Definition 0.0.1 (Class of discrete control process) *Soit $\mathcal{H}^{\pi,\psi}$ l'ensemble des processus de contrôle discrets Z définis par : pour $\mathfrak{z} \in \mathbb{R}^{d\bar{K}^z}$ et les fonctions de base (0.0.6),*

$$Z_{t_n} := \sum_{k=1}^{K_n^z} \psi_n^k(X_{t_n})\mathfrak{z}^{n,k}, \quad \textit{for } 0 \leqslant n \leqslant N-1, \tag{0.0.8}$$

et nous définissons $Z_t = Z_{t_n}$, $t_n \leqslant t < t_{n+1}, 0 \leqslant n \leqslant N-1$ avec la convention $Z_T = 0$.

Definition 0.0.2 *Étant donné $\mathfrak{u} = (\mathfrak{y},\mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, on note par $Z^{\mathfrak{u}} \in \mathcal{H}^{\pi,\psi}$ le processus de contrôle discret défini par (0.0.8). Ensuite, le processus contrôlé discret $Y^{\mathfrak{u}}$ est défini comme suit:*

1. *Initialisation: définir*

$$Y_0^{\mathfrak{u}} = \sum_{k=1}^{K} \psi_y^k(X_0)\mathfrak{y}^k \,. \tag{0.0.9}$$

2. *Version discrète: pour tout* $0 \leqslant n \leqslant N-1$*:*

$$Y_{t_{n+1}}^{\mathfrak{u}} = Y_{t_n}^{\mathfrak{u}} - hf(Y_{t_n}^{\mathfrak{u}}, Z_{t_n}^{\mathfrak{u}}) + Z_{t_n}^{\mathfrak{u}} \cdot \Delta W_n. \tag{0.0.10}$$

3. *Version continue: pour tout* $0 \leqslant n \leqslant N-1$ *et tout* $t_n \leqslant t < t_{n+1}$*,*

$$Y_t^{\mathfrak{u}} = Y_{t_n}^{\mathfrak{u}} - (t-t_n)f(Y_{t_n}^{\mathfrak{u}}, Z_{t_n}^{\mathfrak{u}}) + Z_{t_n}^{\mathfrak{u}} \cdot (\mathcal{W}_t - \mathcal{W}_{t_n}) \tag{0.0.11}$$

D'après la Définition 0.0.1 et la Définition 0.0.2, soit $\mathcal{B}^{\pi,\psi}$ l'ensemble des processus $(Y^{\mathfrak{u}}, Z^{\mathfrak{u}})$, avec $Z^{\mathfrak{u}} \in \mathcal{H}^{\pi,\psi}$, $Y^{\mathfrak{u}}$ défini comme ci-dessus pour un $\mathfrak{u} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$. Maintenant, l'idée principale d'*approximation par des méthodes d'apprentissage* est de minimiser une fonction de perte définie comme la différence entre la condition terminale approchée $g(X_T)$ et le processus contrôlé discret $Y_T^{\mathfrak{u}}$ au temps de maturité T. Ici, nous travaillons avec une fonction de perte quadratique, c'est-à-dire que nous devons besoin de résoudre le problème d'optimisation

$$\inf_{\mathfrak{u}=(\mathfrak{y},\mathfrak{z})\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}} \mathfrak{g}(\mathfrak{u}) := \mathbb{E}[\mathrm{G}(\mathcal{X}_0, W, \mathfrak{u})] \tag{0.0.12}$$

avec

$$\mathrm{G}(\mathcal{X}_0, W, \mathfrak{u}) = |g(X_T) - Y_T^{\mathfrak{u}}|^2 \,.$$

Cependant, le problème d'optimisation ci-dessus (0.0.12) n'est généralement pas convexe, nous ne pouvons donc pas garantir que l'algorithme converge vers des minima locaux ou globaux. Il est bien connu que la solution de l'EDSR (0.0.4) peut être obtenue par la limite d'une suite d'itérations de Picard, voir e.g. [37, 9] d'un point de vue numérique. Nous introduisons donc l'*algorithme de Picard* qui transforme le problème d'optimisation non convexe (0.0.12) à une suite des problèmes d'optimisation linéaire-quadratique. Notre *algorithme de Picard* est basé sur l'itération de l'opérateur Φ défini ci-dessous :

$$\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z} \ni \tilde{\mathfrak{u}} \mapsto \Phi(\tilde{\mathfrak{u}}) := \operatorname*{arg\,min}_{\mathfrak{u}\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}} \mathbb{E}\Big[|g(X_T) - U_T^{\tilde{\mathfrak{u}},\mathfrak{u}}|^2\Big], \tag{0.0.13}$$

où $U^{\tilde{\mathfrak{u}},\mathfrak{u}}$ est définie par le schéma d'approximation de découplage suivant:

1. Pour $\tilde{\mathfrak{u}} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, considérons $(Y^{\tilde{\mathfrak{u}}}, Z^{\tilde{\mathfrak{u}}}) \in \mathcal{B}^{\pi,\psi}$ introduit dans la Définition 0.0.2.

2. Alors, pour tout $\mathfrak{u} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, et $Z^{\mathfrak{u}} \in \mathcal{H}^{\pi,\psi}$ comme introduit dans la Définition 0.0.1, nous définissons le processus de contrôle $U^{\tilde{\mathfrak{u}},\mathfrak{u}}$ par

$$U_0^{\tilde{\mathfrak{u}},\mathfrak{u}} = Y_0^{\mathfrak{u}} \tag{0.0.14}$$

et pour tout $0 \leqslant n \leqslant N-1$,

$$U_{t_{n+1}}^{\tilde{\mathfrak{u}},\mathfrak{u}} = U_{t_n}^{\tilde{\mathfrak{u}},\mathfrak{u}} - hf(Y_{t_n}^{\tilde{\mathfrak{u}}}, Z_{t_n}^{\tilde{\mathfrak{u}}}) + Z_{t_n}^{\mathfrak{u}} \cdot (W_{t_{n+1}} - W_{t_n}) \,, \tag{0.0.15}$$

où le générateur $f(Y_{t_n}^{\tilde{\mathfrak{u}}}, Z_{t_n}^{\tilde{\mathfrak{u}}})$ ne dépend pas des processus $(Y^{\mathfrak{u}}, Z^{\mathfrak{u}})$ pour un $\tilde{\mathfrak{u}} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$ fixé à l'étape 1. Il existe donc une solution unique au problème d'optimisation (0.0.13). La définition suivante donne la procédure entière de l'*algorithme de Picard*.

Definition 0.0.3 (Theoretical Picard algorithm) *Pour un entier positif pré-spécifié P:*

1. *Initialisation: définir $\mathfrak{u}^0 \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$.*
2. *Itération: pour $1 \leqslant p \leqslant P$, calculer : $\mathfrak{u}^p = \Phi(\mathfrak{u}^{p-1})$.*

Ensuite, la sortie de l'algorithme est $\mathfrak{u}^P$.

Dans le théorème 2.2.1, sous certaines hypothèses théoriques telles que des conditions de petitesse, l'erreur quadratique moyenne de l'algorithme complet est contrôlée par la somme de l'erreur de discrétisation en temps, l'erreur introduite par l'algorithme SGD, l'erreur de discrétisation de l'espace de l'approximation de la grille clairsemée, et enfin l'erreur due à l'itération de Picard. La complexité numérique $\mathcal{C}_\varepsilon$ de l'algorithme complet est donnée par

$$\mathcal{C}_\varepsilon = O_d(PNKM)\,. \tag{0.0.16}$$

où K est le nombre de fonctions de base dans l'espace des grilles et M est les étapes d'itération de l'algorithme SGD.

Dans la cadre de la Section 2.3.1.2 , où les coefficients sont supposés périodiques, afin d'obtenir une erreur quadratique moyenne globale d'ordre ε^2 comme indiquée dans (0.0.16), la complexité complète de l'*algorithme de Picard* en utilisant des fonctions de base pré-ondelettes sur des grilles clairsemées est

$$\mathcal{C}_\varepsilon = O_d(\varepsilon^{-\frac{5}{2}(1+2\iota)} |\log_2(\varepsilon)|^{1+\frac{45+50\iota}{36}(d-1)})$$

pour tout $1 < \iota < \frac{9}{5}$.

En particulier, cela montre que le "fléau de la dimension" est apprivoisé en utilisant l'approximation de la grille clairsemée car on peut voir que $\mathcal{C}_\epsilon$ croît polynomialement sur ε^{-1} jusqu'à un certain facteur logarithmique $|\log_2(\varepsilon)|$ dont l'exposant est une fonction affine par rapport à d.

Dans le Chapitre 3, nous introduisons un schéma d'approximation d'ordre élevé (e.g. le schéma de Crank-Nicolson, le schéma de Runge-Kutta) pour réduire le temps de calcul du schéma d'apprentissage profond avec le schéma d'Euler introduit dans [65]. Nous montrons que la convergence en utilisant le théorème d'approximation universel des réseaux de neurones [62, 63]. De plus, certaines techniques de réduction de la variance sont utilisées.

Au cours des dix dernières années, les réseaux de neurones ont été appliqués dans nombreux domaines (e.g. le traitement d'images, le NLP, l'IA ...) qui ont des réalisations impressionnantes, notamment pour surmonter empiriquement le "fléau de la dimension" du problèmes de grande dimension. En effet, le temps du calcul de l'approximation par le réseau de neurones a une croissance au plus polynomial.

Le réseau de neurones a d'abord été appliqué pour résoudre des EDSRs de grande dimension par Weinan E, Jiequn Han, Arnulf Jentzen et Christian Beck en 2017 [36, 7]. Ils ont proposé dans leur travail un schéma forward (voir la Définition 0.0.4 une discretiozation par le schéma d'Euler) avec un grand schéma de réseau de neurones. De nombreux travaux basés sur les réseaux de neurones pour résoudre les EDSRs apparissent, voir par exemple [58, 21, 65, 46, 86, 47, 67, 68, 90, 28].

Definition 0.0.4 (Implemented deep forward scheme) *[36] Pour la fonction terminale fixée g, la solution numérique est calculée par les étapes suivantes:*

- *Pour $n = 0$, initialiser $X_0 = x_0, \hat{\mathcal{Y}}_0 = y_0$.*
- *Pour $n = 0, \cdots, N-1$, étant donné $\hat{\mathcal{Y}}_n$,*
  - *Calculer $X_{n+1} = X_n + b(X_n)h + \sigma(X_n)\Delta W_n$.*
  - *Calculer $\hat{\mathcal{V}}_n = \mathcal{N}(X_n; \theta_n)$, où $\mathcal{N}(x; \theta_n)$ est un réseau de neurones avec des paramètres réels θ_n et des variables d'entrée x.*
  - *Calculer $\hat{\mathcal{Y}}_{n+1} = \hat{\mathcal{Y}}_n - hf(\hat{\mathcal{Y}}_n, \hat{\mathcal{V}}_n) + \hat{\mathcal{V}}_n \Delta W_n$. A l'inverse, $\hat{\mathcal{Y}}_{n+1}$ dépend des paramètres y_0 et $(\theta_0, \cdots, \theta_n)$.*
- *Calculer un minimiseur de la fonction de perte :*

$$(y_0^\star, \theta^\star) \in \operatorname{argmin}_{y_0,\theta} \mathbb{E}\Big[|\hat{\mathcal{Y}}_N(y_0, \theta) - g(X_T)|^2\Big].$$

où $\theta = (\theta_0, \cdots, \theta_{N-1})$.

Par rapport au forward schéma profond défini ci-dessus, un schéma plus stable a été introduit pour la première fois par Côme Huré, Hûyen Pham et Xavier Warin [65], appelé backward schéma profond, voir Définition 0.0.5 ci-dessous. La convergence de leurs méthodes repose sur le fait que les réseaux de neurones sont des approximateurs universels. Théoriquement, les erreurs dues aux réseaux neuronaux pourraient être rendues arbitrairement petites en augmentant le nombre de neurones. Inspiré de la méthode **DBDP1** dans [65], notre objectif est de réduire le temps de calcul en utilisant des méthodes d'approximation d'ordre élevé (telles que le schéma de Crank-Nicolson, le schéma de Runge-Kutta...) pour contrôler l'erreur de discrétisation en temps.

Definition 0.0.5 (Implemented deep backward scheme) *Pour la fonction de perte fixée $\mathrm{L_n}[\varphi, \psi](\theta), (\varphi, \psi) \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}) \times \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$, la solution numérique est calculée par les étapes suivantes:*

- *Pour $n = N$, initialiser $\hat{\mathcal{U}}_N = g, \hat{\mathcal{V}}_N = \sigma^\top \nabla_x g$.*
- *Pour $n = N-1, \cdots, 1, 0$, étant donné les réseaux $\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}$,*
  - *Calculer un minimiseur de la fonction de perte :*

$$\theta_n^\star \in \operatorname{argmin}_\theta \mathrm{L_n}[\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}](\theta_n).$$

  où θ_n est les paramètres d'un réseau de neurones $\mathcal{N}(\cdot, \theta_n)$.

– *Définir* $(\hat{\mathcal{U}}_n, \hat{\mathcal{V}}_n) := \mathcal{N}(\cdot, \theta_n^\star)$ *comme les fonctions d'approximation de* $(u(t_n,\cdot), \sigma^\top \partial_x u(t_n,\cdot))$.

Le schéma d'Euler implicite pour les EDSRs [93, 16] est traditionnellement défini comme

$$Y_n = \mathbb{E}_{t_n}[Y_{n+1} + hf(Y_n, Z_n))] \text{ et } Z_n = \mathbb{E}_{t_n}\left[\frac{\Delta W_n}{h} Y_{n+1}\right], \quad 0 \leqslant n < N \tag{0.0.17}$$

avec $(Y_N, Z_N) := (g(X_N), \sigma^\top(X_N)\partial_x g(X_N))$.

Pour le schéma **DBDP1** dans [65], θ et ϑ sont à la fois optimisés: étant donné $y_{n+1}^\star(\cdot)$,

$$(\theta^\star, \vartheta^\star) = \text{argmin}_{\theta,\vartheta}\mathbb{E}\left[|y_{n+1}^\star(X_{n+1}) - \{y_n^\theta(X_n) - hf(y_n^\theta(X_n), z_n^\vartheta(X_n)) + z_n^\vartheta(X_n)\Delta W_n\}|^2\right]. \tag{0.0.18}$$

Comme nous le savons, le taux de convergence faible du schéma d'Euler est d'ordre 1 uniquement, nous introduisons maintenant le schéma de Crank-Nicolson qui est un schéma d'ordre 2 avec une structure simple, voir entre autres [32]. Bien qu'il soit implicite, il a presque la même complexité que les algorithmes obtenus par le schéma d'Euler. Nous allons étudier le schéma habituel de Crank-Nicolson, e.g. un schéma θ avec $\theta = \frac{1}{2}$, pour $0 \leqslant n < N$,

$$\begin{cases} Y_n &= \mathbb{E}_{t_n}\left[Y_{n+1} + \frac{h}{2}(f(Y_n, Z_n) + f(Y_{n+1}, Z_{n+1}))\right], \\ Z_n &= \mathbb{E}_{t_n}[H_n(Y_{n+1} + hf(Y_{n+1}, Z_{n+1}))] \end{cases} \tag{0.0.19}$$

avec $(Y_N, Z_N) := (g(X_N), \sigma^\top(X_N)\partial_x g(X_N))$, où $H_n \in \mathbb{R}^d$ est une variable aléatoire $\mathcal{F}_{t_n}$-mesurable vérifiant $\mathbb{E}_{t_n}[H_n] = 0$ et $h\mathbb{E}_{t_n}\left[|H_n|^2\right] < \infty$.

Dans le Lemme 3.3.3, un problème d'optimisation est introduit pour obtenir la solution du système (0.0.19) avec $A_n := -\frac{1}{2}\mathbb{E}_{t_n}[f(X_{n+1}, Y_{n+1}, Z_{n+1})hH_n]$:

$$\min_{y,z,a\in\mathcal{L}^2(\mathcal{F}_{t_n})} L^n(y,z,a) := C_0 h\mathbb{E}\left[\left|\frac{1}{2}hH_n f(X_{n+1}, Y_{n+1}, Z_{n+1}) + a\right|^2\right] +$$
$$\mathbb{E}\left[\left|Y_{n+1} - \left\{y - \frac{h}{2}\left(f(X_n, y, z) + f(X_{n+1}, Y_{n+1}, Z_{n+1})\right) + (z+a)\frac{H_n}{\upsilon_n}\right\}\right|^2\right] \tag{0.0.20}$$

où $\upsilon_n = \mathbb{E}_{t_n}\left[|H_n|^2\right]$, $C_0 > 0$ est une constante, et A_n est une variable intermédiaire afin d'implémenter l'algorithme en utilisant un seul réseau comme (0.0.18). Il ne fait aucun doute qu'une erreur apparaîtra également dans le terme A lorsque nous optimisons le réseau de neurones, mais elle a la même amplitude que celle dans le terme $\mathcal{Z}$ dans le réseau de neurones, donc l'impact introduit par le terme A sur le résultat final est négligeable.

Les méthodes de Runge-Kutta [24] sont une famille des méthodes de discrétisation implicites et explicites, qui incluent le schéma d'Euler implicite, le schéma d'Euler explicite, le schéma de Crank-Nicolson en particulier, voir le Théorème 3.2.1. D'après la Définition 0.0.6 ci-dessous, nous remarquons que le schéma de Runge-Kutta est un schéma à plusieurs étapes et peut atteindre un ordre supérieur

au schéma de Crank-Nicolson. Soit Q le nombre d'étapes, remarquons que le schéma de Runge-Kutta est toujours explicite pour le terme Z, il existe une barrière d'ordre pour que le schéma implicite obtienne un schéma d'ordre $Q+1$ avec un schéma de $Q-$étapes lorsque $Q>1$ tant que $\partial_z f \neq 0$. Par conséquent, nous ne considérons le schéma explicite que lorsque $Q=2,3$ car le schéma implicite n'a aucun avantage par rapport au schéma explicite pour les générateurs généraux. Cependant, l'algorithme converge trop vite lorsque $Q=3$, ce qui conduit à une erreur de discrétisation inférieure à la variance de l'algorithme. Et il existe également une barrière d'ordre pour le schéma explicite, ce qui signifie qu'il n'y a pas de méthodes explicites à quatre étapes dans la classe des méthodes pour la Définition 0.0.6. De sorte que nous ne considérons jamais le cas $Q \geqslant 4$.

Definition 0.0.6 *Pour $Q \in \mathbb{N}^+$, soit $c = (c_1, \dots, c_{Q+1}) \in [0,1]^{Q+1}$ satisfaisant $0 =: c_1 < c_2 \leqslant \dots \leqslant c_q \leqslant \dots \leqslant c_Q \leqslant c_{Q+1} := 1$, et $t_{n,q} := t_{n+1} - c_q h$. Alors $t_n = t_{n,Q+1} \leqslant \dots \leqslant t_{n,q} \leqslant \dots \leqslant t_{n,1} = t_{n+1}$. On note la "grille complète"*

$$\Pi := \{t_{n,q} \in [0,T] \,|\, 0 \leqslant n \leqslant N, 1 \leqslant q \leqslant Q\}.$$

Pour $t_{n,q} \in \Pi$, $\mathcal{X}_{t_{n,q}}$ est approximé par $X_{n,q} \in \mathcal{L}^2(\mathcal{F}_{t_{n,q}})$, $0 \leqslant n \leqslant N$ et $1 \leqslant q \leqslant Q$. Pour samplifier la notation, notons $(X_n)_{0\leqslant n\leqslant N}$ l'approximation de $\mathcal{X}$ sur la grille π. Observez que $X_{n,Q+1} = X_n$ et $X_{n,1} = X_{n+1}$. Supposons que X est un processus de Markov sur Π. Nous définissons maintenant (Y,Z) l'approximation de $(\mathcal{Y},\mathcal{Z})$, rappelons (0.0.4).

i) Définir la condition terminale par

$$(Y_N, Z_N) = (g(X_N), \sigma(X_N)^\top \nabla g(X_N)).$$

ii) Pour $0 \leqslant n \leqslant N-1$ et $Q \geqslant 1$, le passage de (Y_{n+1}, Z_{n+1}) à (Y_n, Z_n) implique Q étapes. Aux instants intermédiaires, pour $1 < q \leqslant Q+1$, soit

$$Y_{n,q} = \mathbb{E}_{t_{n,q}}\left[Y_{n+1} + h \sum_{k=1}^{q} a_{qk} f(X_{n,k}, Y_{n,k}, Z_{n,k}) \right], \tag{0.0.21}$$

$$Z_{n,q} = \mathbb{E}_{t_{n,q}}\left[H_q^n Y_{n+1} + h \sum_{k=1}^{q-1} \alpha_{qk} H_{q,k}^n f(X_{n,k}, Y_{n,k}, Z_{n,k}) \right], \tag{0.0.22}$$

où $(a_{qk})_{1\leqslant q,k\leqslant Q+1}, (\alpha_{qk})_{1\leqslant q,k\leqslant Q+1}$ prennent leurs valeurs dans $\mathbb{R}$ et avec $a_{1k} = \alpha_{1k} = 0, 1 \leqslant k \leqslant Q$, $a_{qk} = \alpha_{qk} = 0, 1 \leqslant q < k \leqslant Q+1$ et

$$\sum_{k=1}^{q} a_{qk} = \sum_{k=1}^{q-1} \alpha_{qk} \mathbb{1}_{\{c_k < c_q\}} = c_q, \qquad q \leqslant Q+1. \tag{0.0.23}$$

Nous définissons $(Y_n, Z_n) = (Y_{n,Q+1}, Y_{n,Q+1})$ au temps t_n.
Pour tout $1 \leqslant k < q \leqslant Q+1, n \leqslant N$, les variables aléatoires $H_q^n, H_{q,k}^n$ sont $\mathcal{F}_{t_{n+1}}-$mesurable, indépendants de $\mathcal{F}_{t_{n,q}}$ et $\mathcal{F}_{t_{n,k}}$ respectivement ayant la propriété

$$\mathbb{E}_{t_{n,q}}\big[H_q^n\big] = \mathbb{E}_{t_{n,k}}\big[H_{q,k}^n\big] = 0 \text{ et } \upsilon_q^n := \mathbb{E}_{t_{n,q}}\big[|H_q^n|^2\big], \upsilon_{q,k}^n := \mathbb{E}_{t_{n,k}}\big[|H_{q,k}^n|^2\big],$$
$$\frac{\lambda}{h} \leqslant \min(\upsilon_q^n, \upsilon_{q,k}^n) \text{ et } \max(\upsilon_q^n, \upsilon_{q,k}^n) \leqslant \frac{\Lambda}{h},$$

où λ, Λ sont des constantes positives qui ne dépendent pas de h.

L'approximation du schéma général de Runge-Kutta est essentiellement basée sur une itération de ce qui a été fait pour le schéma de Crank-Nicolson. Nous introduisons une nouvelle variable intermédiaire

$$A_{n,q} = \mathbb{E}_{t_{n,q}}\left[\sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right) hf(X_{n,k}, Y_{n,k}, Z_{n,k})\right], \tag{0.0.24}$$

et nous minimisons alors la fonction de perte

$$\begin{aligned} L_{n,q}(y,z,a) := \mathbb{E}\Bigg[& C_0 h\Big|a - \sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right) hf(X_{n,k}, Y_{n,k}, Z_{n,k})\Big|^2 + \\ & + \Big|Y_{n+1} + h\sum_{k=1}^{q-1} a_{qk} f(X_{n,k}, Y_{n,k}, Z_{n,k}) - \{y - ha_{qq}f(X_{n,q}, y, z) + (z+a)\frac{H_q^n}{\upsilon_q^n}\}\Big|^2\Bigg] \end{aligned}$$

Ensuite, nous pouvons implémenter l'ensemble du schéma Runge-Kutta comme nous l'avons décrit dans la Définition 0.0.7 ci-dessous.

Definition 0.0.7 (Implemented Runge-Kutta scheme) *La solution numérique est calculée en utilisant l'étape suivante:*

- *Pour $n = N$, initialiser $\hat{\mathcal{U}}_N = g, \hat{\mathcal{V}}_N = \sigma^\top \nabla_X g$, $\hat{\mathcal{A}}_N = 0$.*
- *Pour $n = N-1, \cdots, 0$, $1 < q \leqslant Q+1$ étant donné $(\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}) =: (\hat{\mathcal{U}}_{n,1}, \hat{\mathcal{V}}_{n,1})$ et $(\hat{\mathcal{U}}_{n,k}, \hat{\mathcal{V}}_{n,k})$, $1 < k < q$,*
  - *Définir $(\Phi_k, \Psi_k) := (\hat{\mathcal{U}}_{n,k}, \hat{\mathcal{V}}_{n,k})$, $1 \leqslant k < q$, $(\Phi_k, \Psi_k) := 0$, $k \geqslant q$*
  - *Calculer un minimiseur de la fonction de perte :*

$$\theta_{n,q}^\star \in \operatorname{argmin}_\theta \mathrm{L}_{\mathrm{n,q}}^{\mathrm{RK}}[\Phi, \Psi](\theta),$$

    où $\mathrm{L}_{\mathrm{n,q}}^{\mathrm{RK}}$ est la fonction de perte du schéma de Runge-Kutta à l'étape q et $\Phi = (\Phi_1, \cdots, \Phi_{q-1}) \in \mathcal{C}(\mathbb{R}^d, \mathbb{R})^{q-1}$ et $\Psi = (\Psi_1, \cdots, \Psi_{q-1}) \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)^{q-1}$
  - *Définir $(\hat{\mathcal{U}}_{n,q}, \hat{\mathcal{V}}_{n,q}, \hat{\mathcal{A}}_{n,q}) := \mathcal{N}_m(\ cdot; \theta_{n,q}^\star)$, où $\mathcal{N}_m(\cdot; \theta_{n,q}^\star)$ est un réseau de neurones.*

Définir $(\hat{\mathcal{U}}_n, \hat{\mathcal{V}}_n) := (\hat{\mathcal{U}}_{n,Q+1}, \hat{\mathcal{V}}_{n,Q+1})$

Partie I: Représentation probabiliste pour les modèles de volatilité stochastique (Chapitre 4)

Dans la deuxième partie de la thèse, nous établissons une formule de représentation probabiliste de deux formules d'intégration par parties (IPP) de la loi marginale du processus pour certains modèles de volatilité stochastique à un temps de maturité fixé T. Ensuite, une méthode de simulation de Monte Carlo non biaisée découle des

formules probabilistes basées sur une simple chaîne de Markov évoluant le long d'une grille de temps aléatoire donnée par les temps de saut d'un processus de renouvellement indépendant, de sorte qu'elle peut être utilisée pour calculer numériquement le prix et les grecques des options, en particulier delta et vega, pour une large classe de pay-off européen non reguliée. L'erreur obtenue est optimale puisque le calcul ne sera affecté que par l'erreur statistique. La principale nouveauté de notre approche par rapport aux travaux précédents [42, 3, 4] est que nous permettons au coefficient de dérive d'être éventuellement non borné comme c'est le cas dans la plupart des modèles de volatilité stochastique (modèles de Stein-Stein, modèles de Heston, ...).

En finance mathématique, un modèle à volatilité stochastique est un modèle dont la variance est donnée par un processus stochastique, au lieu d'un processus déterministe [44]. Ces modèles sont largement utilisés en finance mathématique pour le pricing des produits dérivés, tels que les options. Les modèles de volatilité stochastique sont des extensions du modèle Black-Scholes, pour lequel la volatilité est supposée constante au fil de temps. De nombreux modèles de volatilité stochastique ont été étudiés, tels que le modèle Heston [61], le modèle CEV [30], le modèle de volatilité SABR [54], le modèle GARCH [19] parmi les autres. Dans ce travail, nous considérons un modèle de volatilité stochastique bidimensionnel défini par la solution (S, Y) de l'EDS suivante

$$\begin{cases} S_t &= s_0 + \int_0^t r S_s \, ds + \int_0^t \sigma_S(Y_s) S_s \, dW_s, \\ Y_t &= y_0 + \int_0^t b_Y(Y_s) \, ds + \int_0^t \sigma_Y(Y_s) \, dB_s, \\ d\langle B, W\rangle_s &= \rho \, ds \end{cases} \tag{0.0.25}$$

où les coefficients b_Y, σ_S, σ_Y $: \mathbb{R} \to \mathbb{R}$ sont des fonctions reguliées, $r \in \mathbb{R}$, W et B sont des mouvements browniens standards unidimensionnels ayant un facteur de corrélation $\rho \in (-1, 1)$, définis sur un espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$.

Nous supposons que $a_S := \sigma_S^2$, $a_Y := \sigma_Y^2$ et la dérive b_Y sont infiniment différentiables, et supposons que a_S et a_Y sont bornés. Notre amélioration principale est que b_Y n'est pas borné, par rapport aux autres travaux sur la représentation probabiliste, voir e.g. [42] pour les processus tués et Agarwal et Gobet [2] pour les processus de diffusion multidimensionnels.

Nous nous intéressons à établir une formule de représentation probabiliste du prix d'une option européenne de maturité $T > 0$ et de pay-off $h(S_T, Y_T)$, donnée par

$$\mathbb{E}\left[h(S_T, Y_T)\right],$$

ainsi que les formules d'intégration par parties (IPP) des sensibilités (Grecques) de l'option, données par

$$\partial_{s_0} \mathbb{E}\left[h(S_T, Y_T)\right] \quad \text{et} \quad \partial_{y_0} \mathbb{E}\left[h(S_T, Y_T)\right],$$

d'où découle une méthode de simulation de Monte Carlo non biaisée.

Nous établissons une formule de représentation probabiliste de la loi marginale (S_T, Y_T), $T > 0$ basée sur une chaîne de Markov simple évoluant une grille en temps aléatoire donné par les temps de saut d'un processus de renouvellement indépendant.

Cette formule fournit une méthode de Monte Carlo non biaisée. Elle s'inspire de la formule de représentation probabiliste de Bally et Kohatsu-Higa [4] et les autres [3, 60, 42, 2] pour les processus de diffusion multidimensionnels, les processus unidimensionnels et de certains EDS de Lévy avec des coefficients bornés de dérive, de diffusion et de saut.

Le principal nouveau défi est de s'attaquer au cas où la dérive de la volatilité b_Y est non bornée. Afin de surmonter cette difficulté, nous figeons les coefficients b_Y , σ_S et σ_Y le flot d'équation différentielle ordinaire (ODE) $\frac{\mathrm{d}m_t}{\mathrm{d}t} = b_Y(m_t)$, $m_0 = y_0$ obtenue en supprimant le terme de diffusion dans la dynamique de Y. La chaîne de Markov sous-jacente $(\bar{X}, \bar{Y})$ sur laquelle la représentation probabiliste est basée, est alors obtenue à partir de

$$\begin{aligned}
\bar{X}_t^{x_0} &= x_0 + \int_0^t (r - \frac{1}{2} a_S(m_s))\, ds + \int_0^t \sigma_S(m_s)\ \mathrm{d}W_s, \\
\bar{Y}_t^{y_0} &= y_0 + \int_0^t b_Y(m_s)\, ds + \int_0^t \sigma_Y(m_s)\ \mathrm{d}B_s, \\
d\langle W, B\rangle_s &= \rho\ \mathrm{d}s.
\end{aligned}$$

Supposons que $\tau = (\tau_n)_{n\geqslant 0}, \tau_0 = 0$ est une suite non décroissante de $\mathbb{R}_+$ modélisant des temps de saut aléatoires et soit $N = (N_t)_{t\geqslant 0}$ le processus de renouvellement, défini par $N_t := \sum_{n\geqslant 1} \mathbf{1}_{\{\tau_n \leqslant t\}}$. N est indépendant des deux mouvements browniens W et B. On discrétise le processus $(\bar{X}, \bar{Y})$ en utilisant un schéma d'Euler sur la grille de temps aléatoire $(\zeta_i)_{i\geqslant 0}$ avec $\zeta_0 = 0$ et $\zeta_i = \tau_i \wedge T$, comme suit

$$\begin{aligned}
\bar{X}_{i+1} &= \bar{X}_i + \Big(r(\zeta_{i+1} - \zeta_i) - \frac{1}{2} a_{S,i}\Big) + \sigma_{S,i} Z_{i+1}^1, \\
\bar{Y}_{i+1} &= m_i + \sigma_{Y,i}\Big(\rho_i Z_{i+1}^1 + \sqrt{1-\rho_i^2}\, Z_{i+1}^2\Big).
\end{aligned}$$

où

$$\begin{aligned}
a_{S,i} := \sigma_{S,i}^2 &= \int_0^{\zeta_{i+1}-\zeta_i} a_S(m_s(\bar{Y}_i))\ \mathrm{d}s, \\
a_{Y,i} := \sigma_{Y,i}^2 &= \int_0^{\zeta_{i+1}-\zeta_i} a_Y(m_s(\bar{Y}_i))\ \mathrm{d}s, \\
\sigma_{S,Y,i} &:= \int_0^{\zeta_{i+1}-\zeta_i} (\sigma_S \sigma_Y)(m_s(\bar{Y}_i))\ \mathrm{d}s, \\
\rho_i &:= \rho \frac{\sigma_{S,Y,i}}{\sigma_{S,i}\sigma_{Y,i}}, \\
m_i &:= m_{\zeta_{i+1}-\zeta_i}(\bar{Y}_i),
\end{aligned}$$

où $Z = (Z_n^1, Z_n^2)_{n\geqslant 1}$ est une suite de i.i.d. variables aléatoires de loi $\mathcal{N}(0, I_2)$ indépendantes de (W, B).

Nous notons $\mathcal{B}_\gamma(\mathbb{R}^2)$ l'ensemble des applications Boréliennes $h : \mathbb{R}^2 \to \mathbb{R}$ satisfaisant l'hypothèse de croissance exponentielle à l'infini. Sous certaines hypothèses qui seront énoncées dans la partie II de cette thèse, la loi du couple (X_T, Y_T) satisfait

la représentation probabiliste suivante : pour tout $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$ et pour un certain $\gamma > 0$, le prix d'une option exercée à l'instant T avec pay-off $h(X_T, Y_T)$ satisfait :

$$\mathbb{E}[h(X_T, Y_T)] = \mathbb{E}\Big[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \prod_{i=1}^{N_T+1} \theta_i\Big],$$

où les variables aléatoires θ_i sont dans $\mathbb{S}_{i-1,n}(\bar{X}, \bar{Y})$ sur l'ensemble $\{N_T = n\}$. On a $\theta_{N_T+1} = (1 - F(T - \zeta_{N_T}))^{-1}$, et pour $i = 1, \cdots, N_T$,

$$\theta_i = (f(\zeta_i - \zeta_{i-1}))^{-1}\Big[\mathcal{I}_i^{(1,1)}(c_S^i) - \mathcal{I}_i^{(1)}(c_S^i) + \mathcal{I}_i^{(2,2)}(c_Y^i) + \mathcal{I}_i^{(2)}(b_Y^i) + \mathcal{I}_i^{(1,2)}(c_{Y,S}^i)\Big].$$

Enfin, si N est un processus de renouvellement avec des temps de saut qui suit une distribution $Beta(1/2, 1)$, alors pour tout $p \geqslant 1$ et $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$ pour un certain $\gamma p > 0$, la variable aléatoire $h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \prod_{i=1}^{N_T+1} \theta_i$ admet un moment $L^p(\mathbb{P})$ fini.

Comme d'habitude, nous définissons la dérivé par rapport au prix spot (resp. sa volatilité) de l'actif sous-jacent par *Delta* (resp. *Vega*). Sous les hypothèses appropriées **(AR)** et **(ND)** de la partie II, pour tout $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$ et pour un certain $\gamma > 0$ et tout $(s_0, y_0) = (exp(X_0), Y_0) \in \mathbb{R}^2$, la loi du couple (X_T, Y_T) vérifie les formules de type Bismut-Elworthy-Li suivantes :

$$s_0 T \partial_{s_0} \mathbb{E}\Big[h(X_T, Y_T)\Big] = \mathbb{E}\Big[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \sum_{k=1}^{N_T+1} (\zeta_k - \zeta_{k-1}) \overrightarrow{\theta}^{\mathcal{I}_k^{(1),N_T+1}}\Big]$$

avec

$$T \partial_{y_0} \mathbb{E}\Big[h(X_T, Y_T)\Big]$$
$$= \mathbb{E}\left[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \sum_{k=1}^{N_T+1} (\zeta_k - \zeta_{k-1})\Big(\overrightarrow{\theta}^{\mathcal{I}_k^{(2),N_T+1}} + \sum_{j=1}^{k} \overrightarrow{\theta}^{C_j^{N_T+1}} + \overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),N_T+1}}\Big)\right],$$

où $\overrightarrow{\theta}^{\mathcal{I}_k^{(1),n+1}}$, $\overrightarrow{\theta}^{C_j^{n+1}}$, $\overrightarrow{\theta}^{\mathcal{I}_k^{(2),n+1}}$ et $\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),n+1}}$ avec $n \geqslant 0$ sur $\{N_T = n\}$, $1 \leqslant j \leqslant k \leqslant n + 1$, sont des fonctions explicites des paramètres du modèle et des poids θ_i. Cela implique que les variables aléatoires qui apparaissent à l'intérieur des attentes sur le côté droit de la formule IPP peuvent être parfaitement simulées. Par conséquent, *Delta* et *Vega* peuvent être calculées par une méthode de simulation Monte-Carlo non biaisée avec une complexité optimale.

Chapter 1

Introduction

This manuscript investigates the solutions of BSDEs by different learning methods and the probabilistic representation for stochastic volatility models with unbounded drift. The aim of this chapter is to introduce and motivate the questions we studied and to summarize the main results obtained.

Contents

1.1 Machine learning methods for high-dimensional BSDEs

1.1.1 Connection between semilinear parabolic PDEs and BSDEs

Let $\mathcal{W}$ be a d-dimensional Brownian motion defined on a complete probability space $(\mathfrak{O}, \mathcal{A}, \mathbb{P})$ and let $\mathcal{X}_0$ be a $\mathcal{A}$-measurable d-dimensional random vector with compact support or deterministic, independent from $\mathcal{W}$. We define $(\mathcal{F}_t)_{0\leqslant t\leqslant T}$ as the augmented filtration generated by $\mathcal{W}$ and $\mathcal{X}_0$. For $b : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \to \mathbb{M}_d$ (the set of $d \times d$ matrices) two measurable functions, we define the forward diffusion process $\mathcal{X}$ as the solution to the following stochastic differential equation (SDE for short)

$$\mathrm{d}\mathcal{X}_t = b(\mathcal{X}_t)\,\mathrm{d}t + \sigma(\mathcal{X}_t)\,\mathrm{d}\mathcal{W}_t\,, \tag{1.1.1}$$

and we define its infinitesimal generator $\mathcal{L}$, for φ smooth enough, by

$$\mathcal{L}\varphi(t,x) := b(x)\cdot\nabla_x\varphi(t,x) + \frac{1}{2}\mathrm{Tr}[(\sigma\sigma^\top)(x)\nabla_x^2\varphi(t,x)]. \tag{1.1.2}$$

We are concerned in the first part of the thesis with the numerical approximation of the solution $(u(t,\mathcal{X}_t), \sigma^\top(\mathcal{X}_t)\nabla_x u(t,\mathcal{X}_t))$, where u is the solution to the high-dimensional semilinear parabolic PDE

$$\begin{cases} \partial_t u(t,x) + \mathcal{L}u(t,x) + f(u(t,x), \sigma^\top(x)\nabla_x u(t,x)) = 0, \quad (t,x) \in [0,T)\times\mathbb{R}^d, \\ u(T,x) = g(x), \quad x \in \mathbb{R}^d \end{cases} \tag{1.1.3}$$

and the $(\mathcal{F}_t)$-adapted solution $(\mathcal{Y}_t, \mathcal{Z}_t) \in \mathbb{R}\times\mathbb{R}^d$ to the BSDE

$$\mathcal{Y}_t = g(\mathcal{X}_T) + \int_t^T f(\mathcal{Y}_s, \mathcal{Z}_s)\,\mathrm{d}s - \int_t^T \mathcal{Z}_s\cdot\mathrm{d}\mathcal{W}_s\,,\ 0\leqslant t\leqslant T\,, \tag{1.1.4}$$

where $f : \mathbb{R}\times\mathbb{R}^d \to \mathbb{R}$, $g : \mathbb{R}^d \to \mathbb{R}$ are measurable functions.

It was first noticed in [81] that there exist a connection between semilinear parabolic PDEs of the form (1.1.3) and BSDEs (1.1.4) as follows:

$$(u(t,\mathcal{X}_t), \sigma^\top(\mathcal{X}_t)\nabla_x u(t,\mathcal{X}_t)) = (\mathcal{Y}_t, \mathcal{Z}_t), 0\leqslant t\leqslant T.$$

This shows that solving the PDE (1.1.3) is equivalent to solving the BSDE (1.1.4).

Since then, the numerical approximation of (1.1.4) was widely studied through the research of efficient numerical algorithms. In particular, branching methods [59], full history recursive multilevel Picard method (MLP for short) [66], cubature methods [25, 31, 32], optimal quantization methods [6, 5, 80, 77], Malliavin calculus based methods [33, 17, 64] and some linear regression methods [49, 50, 51] were considered. It is acknowledged that such approaches will be feasible for problems up to dimension 10. Solving high-dimensional non-linear BSDEs is a challenging task due to the "curse of dimensionality". A major breakthrough is that machine learning technique (especially using deep neural networks) was applied to this field by Weinan E, Jiequn Han, Arnulf Jentzen, and Christian Beck in 2017 [36]. From then on, 100-dimensional BSDEs became solvable. Then, the performance of numerical

experiments were improved by many new algorithms rely on neural network , see [56, 57, 58, 21, 65, 46, 85, 86, 47, 67, 68, 90, 28]. And Teng Long proposed a Gradient boosting-based numerical algorithm in 2021 [91] which could solve 10000 dimensional nonlinear BSDEs.

1.1.2 Our contributions

In the first part of this thesis, we analyse in detail the theoretical convergence of some numerical approximations, namely traditional Stochastic Gradient Descent (SGD for short) algorithm and deep learning methods, of the solution to a Backward Stochastic Differential Equation. The algorithms are tested on several examples coming from mathematical finance. The approximations are based on the classical connection between non-linear parabolic PDEs and BSDEs. The numerical results on high dimensional BSDEs are compared with the one obtained in [65, 36].

In Chapter 2, we introduce an algorithm to approximate the solutions by using some basis functions on some sparse grids spaces and it is shown to converge to a global minimum. The global optimisation problem to be solved is non-convex due to the non-linear driver of the BSDEs. In order to circumvent this issue, we employ a *Picard iteration procedure*. Then, the overall procedure becomes a sequence of linear-quadratic optimisation problems which can be solved by a SGD algorithm. In Theorem 2.2.1, we control the global error of the *implemented algorithm* which shows the convergence of the algorithm under some conditions. In particular, contrary to the deep learning methods [56, 58, 65], we obtain an upper bound on the error induced by the SGD algorithm. In practice, we rely on sparse grid approximation spaces to deal with the "curse of dimensionality". Our second main result is Theorem 2.3.1 which provides, under a periodic coefficients hypothesis, an upper bound for the global complexity of the *implemented algorithm*. Especially, we prove that the "curse of dimensionality" is tamed, in the sense that an approximation with error less than $\varepsilon > 0$ is obtained with complexity of order $\varepsilon^{-p}|\log(\varepsilon)|^{q(d)}$, where p is a constant and $q(d)$ is an affine function. Last, various examples in a high dimensional setting numerically show the efficiency of our methods, see Section 2.3.

In Chapter 3, we propose new algorithms by combining some high-order approximation schemes (Crank-Nicolson scheme, explicit multi-stage Runge-Kutta scheme) with neural networks to approximate the solutions of high-dimensional BSDEs. We study the stability of these algorithms. We obtain weak convergence of the algorithmms thanks to the Universal Approximation Theorem for neural networks [62, 63] and their discrete time errors, see Theorem 3.3.2. We implement these schemes to numerically compare the convergence rates and the computational time cost in Section 3.4.

1.1.3 SGD algorithms with sparse grids

In Chapter 2, given an equidistant grid $\pi := \{t_n = nh, n = 0, \cdots N | h := T/N\}$ of $[0, T]$, we use the sparse grids space to approach the solutions $(u(t_n, \cdot), \partial_x u(t_n, \cdot))$ of the semi-linear parabolic PDE (1.1.3) at time t_n. The algorithm is applied as the gradient learning algorithm to optimize the coefficients of the basis functions, and a

Picard iteration scheme is introduced so that the algorithm converges to the global solution.

The main goal is to compute an approximation of $u(0, \mathcal{X}_0)$, where u is the solution to the PDE (1.1.3) at the initial time on a given domain or at a specific point. This lead us to introduce the following setup for the initial value $\mathcal{X}_0$:

Assumption 1.1.1 *One of the two following cases holds:*

(i) The law of $\mathcal{X}_0$ has compact support and is absolutely continuous with respect to the Lebesgue measure.

(ii) The law of $\mathcal{X}_0$ is a Dirac mass at some point $x_0 \in \mathbb{R}^d$.

1.1.3.1 The *direct* and Picard algorithms

In Section 2.2, we first introduce the *direct algorithm*, which is a SGD algorithm with a linear specification of the approximation space. However, the non convexity of the optimisation (computed via SGD) may cause numerical difficulty. We then introduce a new numerical method to reach the global minimum, called the *Picard algorithm*.

Let $W := (\mathcal{W}_{t_n})_{0 \leqslant n \leqslant N}$ be the discrete-time version of the Brownian motion $\mathcal{W}$. We define $\Delta W_n = W_{t_{n+1}} - W_{t_n}$, $0 \leqslant n \leqslant N-1$ and

$$X_0 = \mathcal{X}_0, \quad X_{t_{n+1}} = X_{t_n} + b(X_{t_n})h + \sigma(X_{t_n})\Delta W_n, \quad 0 \leqslant n \leqslant N-1\,. \tag{1.1.5}$$

We now introduce a discrete-time approximation of the process $\mathcal{Z}$ and $\mathcal{Y}$. For a given $t_n \in \pi \backslash \{T\}$, assume $\mathcal{V}_n^z$ is a parametric functional approximation space generated by a set of basis functions

$$(\psi_n^k(x))_{1 \leqslant k \leqslant K_n^z}, \quad x \in \mathbb{R}^d \tag{1.1.6}$$

with some positive integer K_n^z. For later use, we set:

$$\bar{K}^z := \sum_{n=0}^{N-1} K_n^z\,. \tag{1.1.7}$$

Definition 1.1.1 (Class of discrete control process) *We let $\mathcal{H}^{\pi,\psi}$ be the set of discrete control process Z defined by: for $\mathfrak{z} \in \mathbb{R}^{d\bar{K}^z}$ and basis functions (1.1.6),*

$$Z_{t_n} := \sum_{k=1}^{K_n^z} \psi_n^k(X_{t_n})\mathfrak{z}^{n,k}, \quad \text{for } 0 \leqslant n \leqslant N-1, \tag{1.1.8}$$

and we set $Z_t = Z_{t_n}$, $t_n \leqslant t < t_{n+1}, 0 \leqslant n \leqslant N-1$ with the convention $Z_T = 0$.

Then, with the above approximation of the control process $\mathcal{Z}$ at hand, we naturally consider the following approximation scheme for the process $\mathcal{Y}$.

Definition 1.1.2 *Given $\mathfrak{u} = (\mathfrak{y}, \mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, we denote by $Z^{\mathfrak{u}} \in \mathcal{H}^{\pi,\psi}$ the discrete control process as given in (1.1.8). Then, the discrete controlled process $Y^{\mathfrak{u}}$ is defined as follows:*

1. *Initialization: Set*

$$Y_0^{\mathfrak{u}} = \sum_{k=1}^{K} \psi_y^k(X_0)\mathfrak{y}^k \,. \tag{1.1.9}$$

2. *Discrete version: for any* $0 \leqslant n \leqslant N-1$*:*

$$Y_{t_{n+1}}^{\mathfrak{u}} = Y_{t_n}^{\mathfrak{u}} - hf(Y_{t_n}^{\mathfrak{u}}, Z_{t_n}^{\mathfrak{u}}) + Z_{t_n}^{\mathfrak{u}} \cdot \Delta W_n. \tag{1.1.10}$$

3. *Continuous version: for any* $0 \leqslant n \leqslant N-1$ *and any* $t_n \leqslant t < t_{n+1}$*,*

$$Y_t^{\mathfrak{u}} = Y_{t_n}^{\mathfrak{u}} - (t-t_n)f(Y_{t_n}^{\mathfrak{u}}, Z_{t_n}^{\mathfrak{u}}) + Z_{t_n}^{\mathfrak{u}} \cdot (\mathcal{W}_t - \mathcal{W}_{t_n}) \tag{1.1.11}$$

Based on Definition 1.1.1 and Definition 1.1.2, let $\mathcal{B}^{\pi,\psi}$ be the set of processes $(Y^{\mathfrak{u}}, Z^{\mathfrak{u}})$, with $Z^{\mathfrak{u}} \in \mathcal{H}^{\pi,\psi}$, $Y^{\mathfrak{u}}$ defined as above for some $\mathfrak{u} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$. Now, the main idea of *approximation by learning methods* is to minimize a loss function defined as the difference of the approximated terminal condition $g(X_T)$ and the discrete controlled process $Y_T^{\mathfrak{u}}$ at maturity T. Here, we work with a quadratic loss function, that is we have to solve the optimization problem

$$\inf_{\mathfrak{u}=(\mathfrak{y},\mathfrak{z})\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}} \mathfrak{g}(\mathfrak{u}) := \mathbb{E}[\mathrm{G}(\mathcal{X}_0, W, \mathfrak{u})] \tag{1.1.12}$$

with

$$\mathrm{G}(\mathcal{X}_0, W, \mathfrak{u}) = |g(X_T) - Y_T^{\mathfrak{u}}|^2 \,.$$

However, the above optimization problem (1.1.12) is generally not convex, so that we cannot guarantee that the algorithm converges to local or global minima. It is well known that the solution of the BSDE (1.1.4) itself can be obtained by the limit of a sequence of Picard iterations, see e.g. [37] and [9] from a numerical perspective. Thus, we introduce the *Picard algorithm* which transforms the non-convex optimisation problem (1.1.12) into a sequence of linear-quadratic optimization problems. Our *Picard algorithm* is based on the iteration of the operator Φ defined below:

$$\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z} \ni \tilde{\mathfrak{u}} \mapsto \Phi(\tilde{\mathfrak{u}}) := \operatorname*{arg\,min}_{\mathfrak{u}\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}} \mathbb{E}\Big[|g(X_T) - U_T^{\tilde{\mathfrak{u}},\mathfrak{u}}|^2\Big] \,, \tag{1.1.13}$$

where $U^{\tilde{\mathfrak{u}},\mathfrak{u}}$ is given by the following decoupling approximation scheme:

1. For $\tilde{\mathfrak{u}} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, consider $(Y^{\tilde{\mathfrak{u}}}, Z^{\tilde{\mathfrak{u}}}) \in \mathcal{B}^{\pi,\psi}$ as introduced in Definition 1.1.2.

2. Then, for any $\mathfrak{u} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, and $Z^{\mathfrak{u}} \in \mathcal{H}^{\pi,\psi}$ as introduced in Definition 1.1.1, we define the control process $U^{\tilde{\mathfrak{u}},\mathfrak{u}}$ by

$$U_0^{\tilde{\mathfrak{u}},\mathfrak{u}} = Y_0^{\mathfrak{u}} \tag{1.1.14}$$

and for any $0 \leqslant n \leqslant N-1$,

$$U_{t_{n+1}}^{\tilde{\mathfrak{u}},\mathfrak{u}} = U_{t_n}^{\tilde{\mathfrak{u}},\mathfrak{u}} - hf(Y_{t_n}^{\tilde{\mathfrak{u}}}, Z_{t_n}^{\tilde{\mathfrak{u}}}) + Z_{t_n}^{\mathfrak{u}} \cdot (W_{t_{n+1}} - W_{t_n}) \,, \tag{1.1.15}$$

where the driver $f(Y_{t_n}^{\tilde{\mathfrak{u}}}, Z_{t_n}^{\tilde{\mathfrak{u}}})$ does not depend on the processes $(Y^{\mathfrak{u}}, Z^{\mathfrak{u}})$ for fixed $\tilde{\mathfrak{u}} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$ in step 1. There consequently exists an unique solution to the optimisation problem (1.1.13). The following definition gives the whole process of the *Picard algorithm*.

Definition 1.1.3 (Theoretical Picard algorithm) *For a prescribed positive integer P:*

1. *Initialization: set $\mathfrak{u}^0 \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$.*
2. *Iteration: for $1 \leqslant p \leqslant P$, compute: $\mathfrak{u}^p = \Phi(\mathfrak{u}^{p-1})$.*

Then, the output of the algorithm is $\mathfrak{u}^P$.

1.1.3.2 SGD algorithm

It is well known that SGD algorithms are efficient iterative methods for solving optimization problems under smooth conditions. The basic idea of SGD is traced back to the Robbins-Monro algorithm, which is introduced by Herbert Robbins and Sutton Monro in 1951 [88]. It is typically used for root-finding problems. In contrast with the Newton-Raphson algorithm, the Robbins–Monro algorithm does not require to compute the inverse of a matrix, which is costly in a high-dimensional setting.

We will implement the SGD algorithm to compute a solution $(\mathfrak{y}, \mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$ to the optimization problem (1.1.12). We first prescribe a positive integer M representing the number of steps that the stochastic algorithm will iterate, and then choose a deterministic non increasing sequence of positive real numbers $(\gamma_m)_{m\geqslant 1}$ representing the learning rates and satisfying the following conditions

$$\sum_{m\geqslant 1} \gamma_m = \infty \quad \text{and} \quad \sum_{m\geqslant 1} \gamma_m^2 < \infty. \tag{1.1.16}$$

A particular learning rates sequence which satisfy the above conditions, and was suggested by Robbins–Monro, have the form $\gamma_m = 1/m^\alpha$ for some $\alpha \in (0.5, 1]$. The algorithm consists in computing iteratively, for $0 \leqslant m \leqslant M-1$ and $\lambda_m = \mathfrak{y}, \mathfrak{z}^n, 0 \leqslant n \leqslant N-1$,

$$\lambda_{m+1} = \lambda_m - \gamma_{m+1} \nabla_\lambda \mathrm{G}\big(\mathcal{X}_0^{m+1}, W^{m+1}, \mathfrak{u}_m\big), \tag{1.1.17}$$

where $\nabla_\lambda \mathrm{G}$ is the gradient of $\mathrm{G}(\mathcal{X}_0, W, \mathfrak{u}) = |g(X_T) - Y_T^{\mathfrak{u}}|^2$ to λ. Under some classical hypothesis, the algorithm converges in L^2, see e.g. [35, 70, 10].

In Lemma 2.2.2, we provide the analytic expression of the local gradient functions $\nabla_\lambda \mathrm{G}(\mathcal{X}_0, W, \mathfrak{u})$, $\lambda \in \{\mathfrak{y}, \mathfrak{z}^n, 0 \leqslant n \leqslant N-1\}$ appearing in (1.1.17), allowing to easily compute $(\mathfrak{y}_{m+1}, \mathfrak{z}_{m+1})$ once $(Y^{\mathfrak{u}_m}, Z^{\mathfrak{u}_m})$, $0 \leqslant m \leqslant M-1$, have been simulated.

1.1.3.3 Sparse grids

For both the implemented *direct algorithm* and the implemented *Picard algorithm*, the choice of the approximation spaces $\mathscr{V}^y$ and $\mathscr{V}_n^z$, $0 \leqslant n \leqslant N-1$, and their related basis functions $(\psi_y^k)_{1\leqslant k\leqslant K^y}$ and $(\psi_n^k)_{0\leqslant n\leqslant N-1, 1\leqslant k\leqslant K_n^z}$, are of paramount importance.

In Section 2.3, we choose to use sparse grids approximation together with two types of basis functions: pre-wavelet [14] and "modified hat function" [43].

As usual, for $0 \leqslant n \leqslant N-1$, we build the basis functions on a compact domain

$$\mathcal{O}_n = \prod_{l=1}^{d} [\mathfrak{a}_l^n, \mathfrak{b}_l^n] \quad \text{where} \quad \mathfrak{a}_l^n < \mathfrak{b}_l^n \ \text{ for } l \in \{1, \dots, d\}. \tag{1.1.18}$$

The domain specification relies on the applications under study. We will consider two main cases in this work.

1. For all $1 \leqslant n \leqslant N-1$,

$$\mathcal{O}_n = \prod_{l=1}^{d} [\mathfrak{a}_l, \mathfrak{b}_l] =: \mathcal{O}. \tag{1.1.19}$$

   which does not depend on n. We will study this case in Section 2.3.1.2 where we consider coefficient functions that are $\mathcal{O}$-periodic.

2. The coefficient $\mathfrak{a}$ and $\mathfrak{b}$ are functions of the time-step and the diffusion coefficients (b, σ), recalling (1.1.1), and (1.1.3), meaning that $\mathfrak{a}^n$ and $\mathfrak{b}^n$ are defined as

$$\mathfrak{a}^n := \mathfrak{a}(t_n, b, \sigma, d) \quad \text{and} \quad \mathfrak{b}^n := \mathfrak{b}(t_n, b, \sigma, d), \tag{1.1.20}$$

   where $\mathfrak{a}$ and $\mathfrak{b}$ are given mappings.

In both cases, we can obtain the basis functions by a linear transformations from basis functions defined on standard sparse grids on the canonical domain $[0,1]^d$, see e.g. [20]. We choose the number of basis functions in sparse grids space to be of order

$$O(2^\ell \ell^{d-1}) \,, \tag{1.1.21}$$

where ℓ is the prescribed level of the sparse grids, so that "the curse of dimensionality" only depends on the level ℓ, see [38, 87]. [13, Theorem 3.25] shows that the approximation error of the sparse grids space is controlled as soon as the functions to be approximated are smooth enough.

1.1.3.4 Main results

In Theorem 2.2.1, under some theoretical assumptions such as a smallness conditions, the mean squared error of the complete algorithm is controlled by the sum of the time-discretisation error, the error induced by the SGD algorithm, the space-discretisation error from the sparse grid approximation, and finally the error due to the Picard iteration. The numerical complexity $\mathcal{C}_\varepsilon$ of the full algorithm is given by

$$\mathcal{C}_\varepsilon = O_d(PNKM) \,. \tag{1.1.22}$$

Under the setting of Section 2.3.1.2 , where coefficients are assumed periodic, in order to achieve a global mean squared error of order ε^2 as stated in (1.1.22),

the complexity of the full *Picard algorithm* by using pre-wavelet basis functions on sparse grids is

$$\mathcal{C}_\varepsilon = O_d(\varepsilon^{-\frac{5}{2}(1+2\iota)} |\log_2(\varepsilon)|^{1+\frac{45+50\iota}{36}(d-1)})$$

for any $1 < \iota < \frac{9}{5}$.

In particular, it shows that the "curse of dimensionality" is tamed by using the sparse grid approximation as we can see that $\mathcal{C}_\epsilon$ grows polynomially in ε^{-1} up to some logarithmic factor $|\log_2(\varepsilon)|$ whose exponent is an affine function with respect to d.

1.1.3.5 Numerical results

Periodic example We first work on a periodic example under the setting of Assumption 1.1.1 (i). We consider here 1-periodic coefficients of the forward SDE (1.1.1) on $\mathbb{R}^d$, see the model in detail in Section 2.3.1.2. We perform the test for $d = 3$ by *Picard Algorithm* with $P = 5$, then there are $K^y = K_n^z = K = 225$ basis functions. We obtain a mean square error $\mathcal{E}_{\text{MSE}} = 0.0201$ at the 5-th Picard iteration: See Figure 1.1 displaying the learning performance.

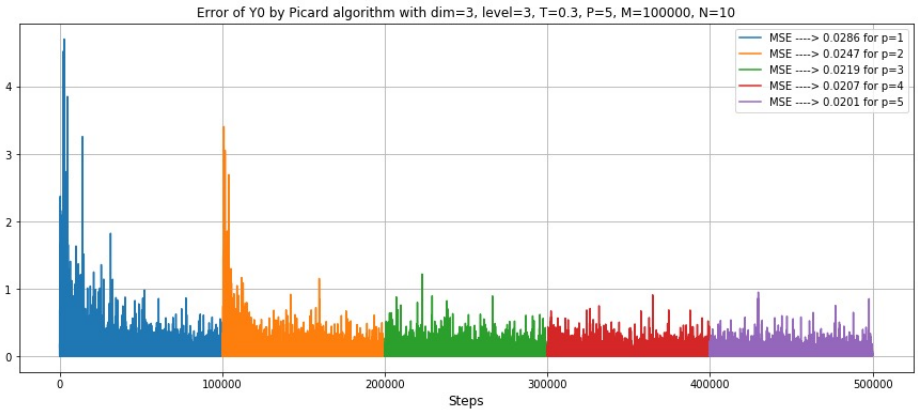

Figure 1.1 – $m \mapsto |\hat{Y}_0^m - u(0, \mathcal{X}_0^m)|^2$ for the *Picard algorithm*, $d = 3$. The MSE is computed by the mean of the last 10000 steps of each Picard iteration.

Numerical convergence of the *Picard* and *direct Algorithm* Under the setting of Assumption 1.1.1(ii), we also compare our methods to existing methods as the ones investigated in [36, 65]. We consider the quadratic example, whose driver is set to

$$f(y, z) = a|z|^2 = a(z_1^2 + z_2^2 + \cdots + z_d^2), \quad y \in \mathbb{R},\ z \in \mathbb{R}^d, \tag{1.1.23}$$

where $a \in \mathbb{R}$ is a constant, and the terminal condition to

$$g(x) = \log\left(\frac{1 + |x|^2}{2}\right), \quad x \in \mathbb{R}^d. \tag{1.1.24}$$

The explicit solution can be obtained through the Cole-Hopf transformation and then simulated by Monte Carlo method. For the 5-dimensional quadratic model,

Figure 1.2 shows the difference of $\hat{y}_0$ between our SGD algorithm and Monte Carlo method is less than 10^{-2}. It turns out that for this "low" dimensional example, it is more precise than the *deep learning algorithm* introduced in [36]. Figure 1.3 shows that $\hat{y}_0$ converges for each Picard iteration. We can observe that $\hat{y}_0$ is very close to the reference solution $\bar{y}_0$ when the number of iteration p is greater or equal to 4.

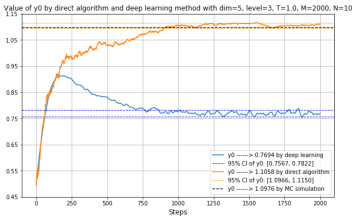

Figure 1.2 – $\hat{y}_0$ for the quadratic model with d=5 and T=1 by *direct algorithm* and deep learning algorithm.

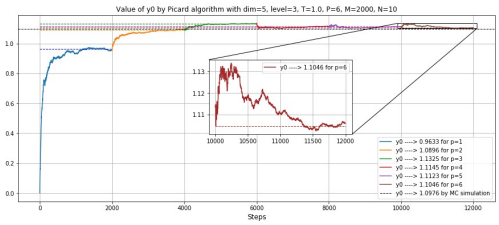

Figure 1.3 – The value of $\hat{y}_0$ by *Picard algorithm* with d=5, level=3, T=1, P=6, M=2000.

Numerical results with the modified hat functions basis We were able to establish a theoretical upper-bound on the global complexity for the *Picard algorithm* by using the pre-wavelet basis. However, the number of basis functions is still quite large which prevents us from dealing effectively with high-dimensional BSDEs. In fact, the number of basis functions used to capture what happens on the boundary of the domain is large. We could use the so-called "modified hat functions" [43] below that allows to get rid of the boundary basis. Table 1.1 shows the number of points in the sparse grids without boundary, it is much less than sparse grids with boundary for the same dimensions and levels, see Table 1.2.

$$\varphi_{l,i}(x) := \begin{cases} 1 & \text{if} \quad l = 1 \wedge i = 1 \\ \left\{ \begin{array}{ll} 1 - 2^{l-1} \cdot x & \text{if} \quad x \in [0, 2h_l] \\ 0 & \text{otherwise} \end{array} \right\} & \text{if} \quad l > 1 \wedge i = 1 \\ \left\{ \begin{array}{ll} 2^{l-1} \cdot x + (1-i)/2 & \text{if} \quad x \in [1 - 2h_l, 1] \\ 0 & \text{otherwise} \end{array} \right\} & \text{if} \quad l > 1 \wedge i = 2^l - 1 \\ \phi_{l,i}(x) & \text{otherwise}\,, \end{cases}$$

where $\phi^{l,i}(x)$ is the family of hat functions given by

$$\phi_{l,i}(x) = \phi(2^l x - i) \text{ with } \phi(x) = \begin{cases} 1 - |x| & if \ -1 < x < 1 \\ 0 & otherwise\,. \end{cases}$$

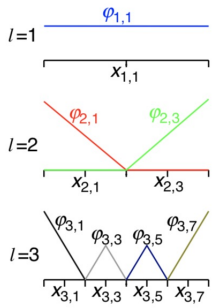

Figure 1.4 – 1-dimensional modified hat functions at level = 1,2,3

levels / dimensions	$l \leqslant 3$	$l \leqslant 4$	$l \leqslant 5$
d=2	17	49	129
d=4	49	209	769
d=5	71	351	1471
d=10	241	2001	13441
d=100	20401	1394001	\

Table 1.1 – The number of points in the sparse grid approximation without boundary functions for different dimensions and levels.

levels / dimensions	$\ell \leqslant 3$	$\ell \leqslant 4$	$\ell \leqslant 5$
d=2	49	113	257
d=4	945	2769	7681
d=5	3753	12033	36033

Table 1.2 – The number of functions in the sparse grid approximation with boundary for different dimensions and levels.

Come back to the quadratic model introduced in (1.1.23)-(1.1.24). In this setting, we can test the 100-dimensional version of this model. the convergence of $\hat{y}_0$ by using the *direct algorithm* is shown in Figure 1.5: 3819 seconds were spent on this test. The error for $\hat{y}_0$ appears to be less than 0.01.

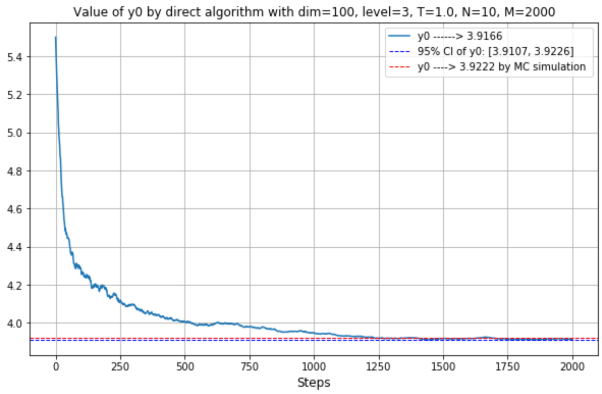

Figure 1.5 – $\hat{z}_0^i$ for the quadratic model with d=100 and T=1

There is a challenging example with an unbounded and complex unbounded structure solution below that was analyzed in [65].

$$u(t,x) = \frac{T-t}{d}\sum_{i=1}^{d}(\sin(x_i)\mathbb{1}_{\{x_i<0\}} + x_i\mathbb{1}_{\{x_i\geqslant 0\}}) + \cos\left(\sum_{i=1}^{d} ix_i\right), \quad x \in \mathbb{R}^d.$$

We compare the approximation of y_0 by using five different algorithms to the theoretical solution in Table 1.3. The *deep learning algorithm* [55] fails when $d \geqslant 3$. The two deep learning schemes of [65] and our algorithms with sparse grids still works well when $d \leqslant 8$. When the dimension $d = 10$, all the algorithms failed at providing correct estimates of the solution as shown in the table, but the errors of our algorithms appear to be smaller than the errors of deep learning methods.

dimensions	Theoretical solution	SGD algo with L^2 sparse grids and hat functions		DL scheme of HPW [65]		DL scheme of HJE [55]
		direct algo	*Picard algo*	DBDP1	DBDP2	
d=1	1.3776	1.3790	1.3825	1.3720	1.3736	1.3724
d=2	0.5707	0.5795	0.5794	0.5715	0.5708	0.5715
d=5	0.8466	0.8734	0.8606	0.8666	0.8365	NC
d=8	1.1603	1.1745	1.1801	1.1694	1.0758	NC
d=10	-0.2149	-0.2439	-0.2594	-0.3105	-0.3961	NC

Table 1.3 – Comparison of the value of $\hat{y}_0$ by different methods when $T = 1$.

1.1.4 Deep learning methods

In Chapter 3, we introduce some high order approximation scheme (such as Crank-Nicolson scheme, Runge-Kutta scheme) to reduce the computational time cost of the backward deep learning scheme with Euler scheme introduced in [65]. The convergence is proved using the universal approximation theorem of neural networks [62, 63]. In addition, some variance reduction techniques are used.

1.1.4.1 Neural networks

In the past ten years, neural networks have been used in many fields (such as image Processing, NLP, AI ...) and this resulted in impressive achievements, especially to overcome empirically the curse of dimensionality when solving high-dimensional problems. Indeed, using neural networks results in approximations computed in an at-most polynomially growing time. It was first applied to solve high-dimensional BSDEs by Weinan E, Jiequn Han, Arnulf Jentzen, and Christian Beck in 2017 [36, 7]. They proposed in their work a forward scheme (see Definition 1.1.4 which is discretized by Euler scheme) with a large neural network scheme. Many works based on neural networks to solve BSDEs have then appeared, see for example [58, 21, 65, 46, 86, 47, 67, 68, 90, 28].

Definition 1.1.4 (Implemented deep forward scheme) *[36] For the given terminal function g, the numerical solution is computed using the following step:*

- *For $n = 0$, initialize $X_0 = x_0, \hat{\mathcal{Y}}_0 = y_0$.*
- *For $n = 0, \cdots, N-1$, given $\hat{\mathcal{Y}}_n$,*
  - *Compute $X_{n+1} = X_n + b(X_n)h + \sigma(X_n)\Delta W_n$.*
  - *Compute $\hat{\mathcal{V}}_n = \mathcal{N}(X_n; \theta_n)$, where $\mathcal{N}(x; \theta_n)$ is a neural network with real parameters θ_n and input variables x.*
  - *Compute $\hat{\mathcal{Y}}_{n+1} = \hat{\mathcal{Y}}_n - hf(\hat{\mathcal{Y}}_n, \hat{\mathcal{V}}_n) + \hat{\mathcal{V}}_n \Delta W_n$. Obversely, $\hat{\mathcal{Y}}_{n+1}$ depends on the parameters y_0 and $(\theta_0, \cdots, \theta_n)$.*
- *Compute a minimizer of the loss function:*

$$(y_0^\star, \theta^\star) \in \operatorname{argmin}_{y_0,\theta} \mathbb{E}\Big[|\hat{\mathcal{Y}}_N(y_0, \theta) - g(X_T)|^2\Big].$$

where $\theta = (\theta_0, \cdots, \theta_{N-1})$.

Compared to the deep forward scheme defined above, a more stable scheme was first introduced by Côme Huré, Hûyen Pham, and Xavier Warin [65], called deep backward scheme, see Definition 1.1.5 below. The convergence of their methods is based on the fact that the neural networks are universal approximators, see Theorem 3.1.1. Theoretically, the errors due to neural networks could be made arbitrarily small by increasing the number of neurons. Inspired by the **DBDP1** method in [65], our goal is to reduce the computational time cost by using high-order approximation methods (such as Crank-Nicolson scheme, Runge-Kutta scheme...) to control the time-discretisation error.

Definition 1.1.5 (Implemented deep backward scheme) *For the given loss function $\mathrm{L_n}[\varphi, \psi](\theta), (\varphi, \psi) \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}) \times \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$, the numerical solution is computed using the following step:*

- *For $n = N$, initialize $\hat{\mathcal{U}}_N = g, \hat{\mathcal{V}}_N = \sigma^\top \nabla_x g$.*
- *For $n = N-1, \cdots, 1, 0$, given the networks $\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}$,*

– *Compute a minimizer of the loss function:*

$$\theta_n^\star \in \text{argmin}_\theta \, \mathrm{L_n}[\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}](\theta_n).$$

where θ_n is the parameters of a neural network $\mathcal{N}(\cdot, \theta_n)$.

– *Set $(\hat{\mathcal{U}}_n, \hat{\mathcal{V}}_n) := \mathcal{N}(\cdot, \theta_n^\star)$ as the approximation functions of $(u(t_n, \cdot), \sigma^\top \partial_x u(t_n, \cdot))$.*

1.1.4.2 Euler scheme

Implicit Euler scheme: The implicit Euler scheme for BSDEs [93, 16] is traditionally defined as

$$Y_n = \mathbb{E}_{t_n}[Y_{n+1} + hf(Y_n, Z_n))] \text{ and } Z_n = \mathbb{E}_{t_n}\left[\frac{\Delta W_n}{h} Y_{n+1}\right], \quad 0 \leqslant n < N \tag{1.1.25}$$

with $(Y_N, Z_N) := (g(X_N), \sigma^\top(X_N)\partial_x g(X_N))$. The conditional expectations could be computed directly by Monte Carlo simulation. For two given sets of parameters θ and ϑ, $x \mapsto y_n^\theta(x)$, $x \mapsto z_n^\vartheta(x)$ are functions (represented as neural network with the given parameters) and, our goal is to find optimal parameters $\theta^\star$, $\vartheta^\star$ so that the associated functions $y_n^\star(\cdot)$, $z_n^\star(\cdot)$ should approximate $u(t_n, \cdot)$ and $\sigma^\top \partial_x u(t_n, \cdot)$ (recalling $Y_t = u(t, X_t)$, $Z_t = \sigma^\top(X_t)\partial_x u(t, X_t)$). The optimal set of parameters is computed recursively. Given $y_{n+1}^\star$, one needs to solve the two following optimisation problems at time t_n:

$$\vartheta^\star = \text{argmin}_\vartheta \mathbb{E}\left[|\frac{\Delta W_i}{h} y_{n+1}^\star(X_{n+1}) - z_n^\vartheta(X_n)|^2\right], \tag{1.1.26}$$

and

$$\theta^\star = \text{argmin}_\theta \mathbb{E}\left[|y_{n+1}^\star(X_{n+1}) - \{y_n^\theta(X_n) - hf(y_n^\theta(X_n), z_n^\star(X_n))\}|^2\right]. \tag{1.1.27}$$

One could also add a term $z_n^\star(W_{t_n})\Delta W_n$ to reduce the variance.

For the scheme called **DBDP1** in [65], the authors optimised both θ and ϑ at the same time: given $y_{n+1}^\star(\cdot)$,

$$(\theta^\star, \vartheta^\star) = \text{argmin}_{\theta,\vartheta} \mathbb{E}\left[|y_{n+1}^\star(X_{n+1}) - \{y_n^\theta(X_n) - hf(y_n^\theta(X_n), z_n^\vartheta(X_n)) + z_n^\vartheta(X_n)\Delta W_n\}|^2\right], \tag{1.1.28}$$

so that only one network is needed to solve this optimization problem which could save half of the computational time cost. In fact, the optimization problem (1.1.28) can be rewritten as

$$\begin{aligned}(\theta^\star, \vartheta^\star) = \text{argmin}_{\theta,\vartheta} \Bigg\{ & \mathbb{E}\left[|y_{n+1}^\star(X_{n+1}) - \{y_n^\theta(X_n) - hf(y_n^\theta(X_n), z_n^\vartheta(X_n))\}|^2\right] \\ & + h\mathbb{E}\left[|\frac{\Delta W_n}{h} y_{n+1}^\star(W_{t_{n+1}}) - z_n^\vartheta(W_{t_n})|^2\right] - \frac{1}{h}\mathbb{E}\left[|y_{n+1}^\star(W_{t_{n+1}})\Delta W_n|^2\right] \Bigg\}.\end{aligned} \tag{1.1.29}$$

We recognize that the second term is the term to minimize in (1.1.26) and can be used to find $\vartheta^\star$, while the first term is the term to minimize in (1.1.27) and can be used to find $\theta^\star$. The last term is a negative constant which could reduce the loss of (1.1.28) compared to (1.1.27)-(1.1.26).

Explicit Euler scheme: In the implicit Euler scheme defined in (1.1.25), we observe that only the Y-part is implicit. The explicit Euler scheme considers an explicit conditional expectation for the Y-part also:

$$Y_n = \mathbb{E}_{t_n}[Y_{n+1} + hf(Y_{n+1}, Z_{n+1}))] \text{ and } Z_n = \mathbb{E}_{t_n}\left[\frac{\Delta W_n}{h} Y_{n+1}\right], \quad 0 \leqslant n < N. \tag{1.1.30}$$

To implement this scheme in practice, one only needs to replace the optimization problem (1.1.28) by

$$\begin{aligned}(\theta^\star, \vartheta^\star) = \text{argmin}_{\theta,\vartheta}\mathbb{E}\Big[|y^\star_{n+1}(X_{n+1}) - \\ \{y^\theta_n(X_n) - hf(y^\star_{n+1}(X_{n+1}), z^\star_{n+1}(X_{n+1})) + z^\vartheta_n(X_n)\Delta W_n\}|^2\Big].\end{aligned} \tag{1.1.31}$$

given $y^\star_{n+1}(\cdot), z^\star_{n+1}(\cdot)$. In practice, using implicit or explicit Euler scheme together with neural networks approximation should induce similar time costs and variance for Y_0.

1.1.4.3 Crank-Nicolson scheme

As we know, the weak convergence rate of Euler scheme is of order 1 only, we now introduce the Crank-Nicolson scheme which is a second-order scheme with a simple structure, see among others [32]. Though it is implicit, it has almost the same complexity as the algorithms obtained by using Euler scheme. For the $Y-$part, we will study the usual Crank-Nicolson scheme, namely a θ-scheme with $\theta = \frac{1}{2}$,

$$\begin{cases} Y_N &= g(X_N), \\ Y_n &= \mathbb{E}_{t_n}\big[Y_{n+1} + \frac{h}{2}(f(Y_n, Z_n) + f(Y_{n+1}, Z_{n+1}))\big], \quad 1 \leqslant n \leqslant N-1. \end{cases} \tag{1.1.32}$$

For the general expression of $Z-$part, we define $\{Z_i\}_{0\leqslant i\leqslant N}$ as follows:

$$\begin{cases} Z_N = \partial_x g(X_N), \\ Z_n = \mathbb{E}_{t_n}[H_n(Y_{n+1} + hf(Y_{n+1}, Z_{n+1}))], \end{cases} \tag{1.1.33}$$

where $H_n \in \mathbb{R}^d$ is a $\mathcal{F}_{t_n}$-mesureable random variables satisfying $\mathbb{E}_{t_n}[H_n] = 0$ and $h\mathbb{E}_{t_n}\big[|H_n|^2\big] < \infty$. When the underlying is Brownian motion, one can choose (among others)

$$H_n = \frac{\Delta W_n}{h}.$$

The numerical analysis for discretization error has been done e.g. in [32, 24]. The local discretization error will be $O(h^3)$, so that the global error is $O(h^2)$. For the general diffusion case, one can choose $H_n = \frac{c-2}{c-1}\frac{\Delta W_n - \Delta W_{n,2}}{h} + \frac{c-1}{c}\frac{\Delta W_{n,2}}{h} \in \mathbb{R}^d$, where $c \in (0,1)$ and $\Delta W_{n,2} = W_{t_{n+1}} - W_{t_{n+1}-ch}$. This directly comes from the Runge-Kutta scheme for BSDEs [24].

Compare to the two steps scheme with loss functions

$$\begin{aligned}\min_{z\in\mathcal{L}^2(\mathcal{F}_{t_n})} \hat{L}^n(z) &:= h\mathbb{E}\Big[\Big|\left(Y_{n+1} + hf(Y_{n+1}, Z_{n+1})\right) H_n - z\Big|^2\Big], \\ \min_{y\in\mathcal{L}^2(\mathcal{F}_{t_n})} \hat{L}^n(y) &:= \mathbb{E}\Big[\Big|Y_{n+1} - \Big\{y - \frac{h}{2}f(Y_{n+1}, Z_{n+1}) - \frac{h}{2}f(y, Z_n)\Big\}\Big|^2\Big],\end{aligned}$$

one can optimize two loss functions together by only one neural network below to save some computational time:

$$\min_{y,z\in\mathcal{L}^2(\mathcal{F}_{t_n})} \hat{L}^n(y,z) := \mathbb{E}\Big[\Big|Y_{n+1} - \Big\{y - \frac{h}{2}f(Y_{n+1},Z_{n+1}) - \frac{h}{2}f(y,Z_n)\Big\}\Big|^2\Big] + h\mathbb{E}\Big[\Big|\left(Y_{n+1} + hf(Y_{n+1},Z_{n+1})\right)H_n - z\Big|^2\Big]. \tag{1.1.34}$$

In Lemma 3.3.3, an equivalent optimisation problem is introduced to get the solution of system (1.1.32)-(1.1.33) with $A_n := -\frac{1}{2}\mathbb{E}_{t_n}[f(X_{n+1},Y_{n+1},Z_{n+1})hH_n]$:

$$\min_{y,z,a\in\mathcal{L}^2(\mathcal{F}_{t_n})} L^n(y,z,a) := C_0h\mathbb{E}\left[\left|\frac{1}{2}hH_nf(X_{n+1},Y_{n+1},Z_{n+1}) + a\right|^2\right] + \mathbb{E}\left[\left|Y_{n+1} - \left\{y - \frac{h}{2}\left(f(X_n,y,z) + f(X_{n+1},Y_{n+1},Z_{n+1})\right) + (z+a)\frac{H_n}{\upsilon_n}\right\}\right|^2\right] \tag{1.1.35}$$

where $\upsilon_n = \mathbb{E}_{t_n}\big[|H_n|^2\big]$, $C_0 > 0$ is a constant, and A_n is an intermediate variable that we introduced in order to implement the algorithm by using only one network as (1.1.28) or (1.1.31). The loss function (1.1.35) can achieve smaller variance for the approximation of Y_0 compares to the loss function (1.1.34). It does not increase the computational time compares to the implicit Euler scheme. There is no doubt that an error will also appear on the $A-$part when we optimise the neural network, but it has the same amplitude as the associated to the $\mathcal{Z}-$part in the neural network, the influence on the final is negligible.

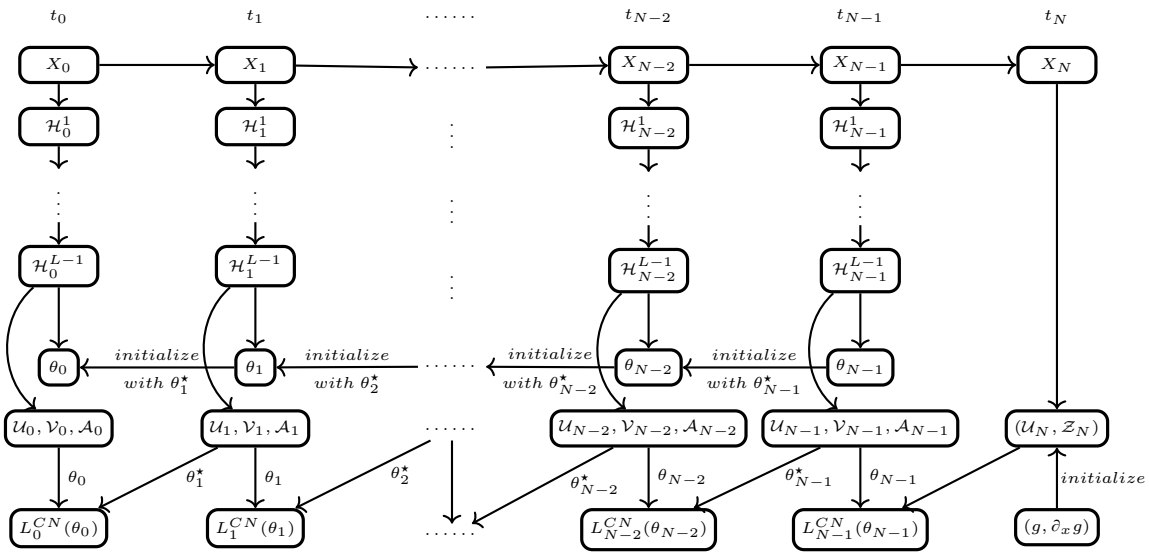

Figure 1.6 – Graph with common parameters and variables of Crank-Nicolson scheme in the deep learning algorithm.

1.1.4.4 Runge-Kutta scheme

Runge-Kutta methods [24] are a family of implicit and explicit discretization methods, which include implicit Euler scheme, explicit Euler scheme, Crank-Nicolson

scheme in particular, see Theorem 3.2.1. From Definition 1.1.6 below, we notice that the Runge-Kutta scheme is a muti-stage scheme and can achieve a higher order than Crank-Nicolson scheme. We denote by Q the number of stages. Note that the Runge-Kutta scheme is always explicit for Z-part, there exists an order barrier for implicit scheme to get an order $Q+1$ scheme with a $Q-$stage scheme when $Q>1$ as long as $\partial_z f \neq 0$. Hence, we only consider the explicit scheme when $Q=2,3$ as the implicit scheme has no advantage compared to the explicit scheme for general drivers. However, the algorithm converges too fast when $Q=3$, which leads to a discretization error smaller than the variance of the algorithm. And there also exists an order barrier for explicit scheme, which means that there is no explicit four stage methods in the class of methods for the definition 1.1.6. So that we never consider the case $Q \geqslant 4$.

Definition 1.1.6 *For $Q \in \mathbb{N}^+$, let $c=(c_1,\dots,c_{Q+1}) \in [0,1]^{Q+1}$ satisfying $0=: c_1 < c_2 \leqslant \dots \leqslant c_q \leqslant \dots \leqslant c_Q \leqslant c_{Q+1} := 1$, and $t_{n,q} := t_{n+1} - c_q h$. Then $t_n = t_{n,Q+1} \leqslant \dots \leqslant t_{n,q} \leqslant \dots \leqslant t_{n,1} = t_{n+1}$. We denote the "full grid"*

$$\Pi := \{t_{n,q} \in [0,T] \,|\, 0 \leqslant n \leqslant N, 1 \leqslant q \leqslant Q\}.$$

For $t_{n,q} \in \Pi$, $\mathcal{X}_{t_{n,q}}$ is approximated by $X_{n,q} \in \mathcal{L}^2(\mathcal{F}_{t_{n,q}})$, $0 \leqslant n \leqslant N$ and $1 \leqslant q \leqslant Q$. For ease of notation, denote by $(X_n)_{0\leqslant n\leqslant N}$ the approximation of $\mathcal{X}$ on the grid π. Observe that $X_{n,Q+1} = X_n$ and $X_{n,1} = X_{n+1}$. Assume that X is a Markov process on Π. We now define (Y,Z) the approximation of $(\mathcal{Y},\mathcal{Z})$, recall (1.1.4).

i) Set the terminal condition as

$$(Y_N, Z_N) = (g(X_N), \sigma(X_N)^\top \nabla g(X_N)).$$

ii) For $0 \leqslant n \leqslant N-1$ and $Q \geqslant 1$, the transition from (Y_{n+1}, Z_{n+1}) to (Y_n, Z_n) involves Q stages. At the intermediate instances, for $1 < q \leqslant Q+1$, let

$$Y_{n,q} = \mathbb{E}_{t_{n,q}}\left[Y_{n+1} + h \sum_{k=1}^{q} a_{qk} f(X_{n,k}, Y_{n,k}, Z_{n,k}) \right], \tag{1.1.36}$$

$$Z_{n,q} = \mathbb{E}_{t_{n,q}}\left[H_q^n Y_{n+1} + h \sum_{k=1}^{q-1} \alpha_{qk} H_{q,k}^n f(X_{n,k}, Y_{n,k}, Z_{n,k}) \right], \tag{1.1.37}$$

where $(a_{qk})_{1\leqslant q,k\leqslant Q+1}, (\alpha_{qk})_{1\leqslant q,k\leqslant Q+1}$ take their values in $\mathbb{R}$ and with $a_{1k} = \alpha_{1k} = 0, 1 \leqslant k \leqslant Q$, $a_{qk} = \alpha_{qk} = 0, 1 \leqslant q < k \leqslant Q+1$ and

$$\sum_{k=1}^{q} a_{qk} = \sum_{k=1}^{q-1} \alpha_{qk} \mathbb{1}_{\{c_k < c_q\}} = c_q, \qquad q \leqslant Q+1. \tag{1.1.38}$$

We set $(Y_n, Z_n) = (Y_{n,Q+1}, Y_{n,Q+1})$ at the dates on π.
For all $1 \leqslant k < q \leqslant Q+1, n \leqslant N$, the random variables $H_q^n, H_{q,k}^n$ are $\mathcal{F}_{t_{n+1}}-$measurable, independent of $\mathcal{F}_{t_{n,q}}$ and $\mathcal{F}_{t_{n,k}}$ respectively with the property

$$\mathbb{E}_{t_{n,q}}\big[H_q^n\big] = \mathbb{E}_{t_{n,k}}\big[H_{q,k}^n\big] = 0 \text{ and } \upsilon_q^n := \mathbb{E}_{t_{n,q}}\big[|H_q^n|^2\big], \upsilon_{q,k}^n := \mathbb{E}_{t_{n,k}}\big[|H_{q,k}^n|^2\big],$$
$$\frac{\lambda}{h} \leqslant \min(\upsilon_q^n, \upsilon_{q,k}^n) \text{ and } \max(\upsilon_q^n, \upsilon_{q,k}^n) \leqslant \frac{\Lambda}{h},$$

where λ, Λ are positive constants which do not depend on h.

The approximation of general Runge-Kutta scheme is essentially based on an iteration of what has been done for the Crank-Nicolson scheme. We introduce a new intermediate variable

$$A_{n,q} = \mathbb{E}_{t_{n,q}}\left[\sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right) hf(X_{n,k}, Y_{n,k}, Z_{n,k})\right], \tag{1.1.39}$$

and we then minimise the loss function

$$L_{n,q}(y,z,a) := \mathbb{E}\Bigg[C_0 h\Big|a - \sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right) hf(X_{n,k}, Y_{n,k}, Z_{n,k})\Big|^2 +$$
$$+\Big|Y_{n+1} + h\sum_{k=1}^{q-1} a_{qk} f(X_{n,k}, Y_{n,k}, Z_{n,k}) - \{y - ha_{qq}f(X_{n,q}, y, z) + (z+a)\frac{H_q^n}{\upsilon_q^n}\}\Big|^2\Bigg]$$

Then, one can implement the whole Runge-Kutta scheme as we described in Definition 1.1.7 below.

Definition 1.1.7 (Implemented Runge-Kutta scheme) *The numerical solution is computed using the following step:*

- *For $n = N$, initialize $\hat{\mathcal{U}}_N = g, \hat{\mathcal{V}}_N = \sigma^\top \nabla_X g$, $\hat{\mathcal{A}}_N = 0$.*
- *For $n = N-1, \cdots, 0$, for $1 < q \leqslant Q+1$ given $(\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}) =: (\hat{\mathcal{U}}_{n,1}, \hat{\mathcal{V}}_{n,1})$ and $(\hat{\mathcal{U}}_{n,k}, \hat{\mathcal{V}}_{n,k})$, $1 < k < q$,*
  - *set $(\Phi_k, \Psi_k) := (\hat{\mathcal{U}}_{n,k}, \hat{\mathcal{V}}_{n,k})$, $1 \leqslant k < q$, $(\Phi_k, \Psi_k) := 0$, $k \geqslant q$*
  - *Compute a minimizer of the loss function:*

$$\theta_{n,q}^\star \in \operatorname{argmin}_\theta \mathrm{L}_{\mathrm{n,q}}^{\mathrm{RK}}[\Phi, \Psi](\theta),$$

    where $\mathrm{L}_{\mathrm{n,q}}^{\mathrm{RK}}$ is the loss function of Runge-Kutta scheme at stage q and $\Phi = (\Phi_1, \cdots, \Phi_{q-1}) \in \mathcal{C}(\mathbb{R}^d, \mathbb{R})^{q-1}$ and $\Psi = (\Psi_1, \cdots, \Psi_{q-1}) \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)^{q-1}$
  - *set $(\hat{\mathcal{U}}_{n,q}, \hat{\mathcal{V}}_{n,q}, \hat{\mathcal{A}}_{n,q}) := \mathcal{N}_m(\cdot; \theta_{n,q}^\star)$, where $\mathcal{N}_m(\cdot; \theta_{n,q}^\star)$ is a neural network.*

Set $(\hat{\mathcal{U}}_n, \hat{\mathcal{V}}_n) := (\hat{\mathcal{U}}_{n,Q+1}, \hat{\mathcal{V}}_{n,Q+1})$

1.1.4.5 Main results

For $n < N$ and $1 < q \leqslant Q+1$, based on the following perturbed scheme,

$$\tilde{Y}_{n,q} = \mathbb{E}_{t_{n,q}}\left[\tilde{Y}_{n+1} + h\sum_{k=1}^{q} a_{qk} f(X_{n,k}, \tilde{Y}_{n,k}, \tilde{Z}_{n,k})\right] + \zeta_{n,q}^y, \tag{1.1.40}$$

$$\tilde{Z}_{n,q} = \mathbb{E}_{t_{n,q}}\left[H_q^n \tilde{Y}_{n+1} + h\sum_{k=1}^{q-1} \alpha_{qk} H_{q,k}^n f(X_{n,k}, \tilde{Y}_{n,k}, \tilde{Z}_{n,k})\right] + \zeta_{n,q}^z, \tag{1.1.41}$$

with $(\zeta_{n,q}^y, \zeta_{n,q}^z) \in \mathcal{L}^2(\mathcal{F}_{t_{n,q}})$, we obtain a new stability result to control the error linked to the estimation of the conditional expectations at each stage of the scheme.

Proposition 1.1.1 *Assume that f is Lipschitz continuous. Then, setting $\delta Y_n := \tilde{Y}_n - Y_n$ and $\delta Z_n := \tilde{Z}_n - Z_n$, the following holds*

$$\begin{aligned}&\max_{n<N} \mathbb{E}\big[|\delta Y_n|^2\big] + \sum_{n=0}^{N-1} h\mathbb{E}\big[|\delta Z_n|^2\big] \\ &\leqslant C\mathbb{E}\left[|\delta Y_N|^2 + h|\delta Z_N|^2 + \sum_{n=0}^{N-1}\sum_{q=2}^{Q+1}\left(\frac{|\zeta^y_{n,q}|^2}{h} + h|\zeta^z_{n,q}|^2\right)\right]. \end{aligned} \tag{1.1.42}$$

Under some regularity assumptions on the coefficients, and noticing that $\delta Y_N = \delta Z_N = 0$ in our setting, from the above proposition and the order of discretisation errors, we obtain a global control on the error between the solution of the BSDEs and the deep backward approximations, solutions to our different algorithms.

Theorem 1.1.1 *Let $(\bar{Y}_n, \bar{Z}_n) := (u(t_n, X_n), \sigma^\top \nabla_x(t_n, X_n))$, for $n \leqslant N$. We define*

$$Y^{(\Phi,\Psi)}_{n,q} := \mathbb{E}_{t_{n,q}}\left[Y^{(\Phi,\Psi)}_{n,1} + h\sum_{k=1}^{q} a_{qk} f(X_{n,k}, Y^{(\Phi,\Psi)}_{n,k}, Z^{(\Phi,\Psi)}_{n,k})\right] \tag{1.1.43}$$

$$Z^{(\Phi,\Psi)}_{n,q} := \mathbb{E}_{t_n}\left[H^n_q Y^{(\Phi,\Psi)}_{n,1} + h\sum_{k=1}^{q-1} \alpha_{qk} H^n_{q,k} f(X_{n,k}, Y^{(\Phi,\Psi)}_{n,k}, Z^{(\Phi,\Psi)}_{n,k})\right] \tag{1.1.44}$$

$$A^{(\Phi,\Psi)}_{n,q} = \mathbb{E}_{t_{n,q}}\left[\sum_{k=1}^{q-1}\left(a_{qk}H^n_q - \alpha_{qk}H^n_{q,k}\right) h f(X_{n,k}, Y^{(\Phi,\Psi)}_{n,k}, Z^{(\Phi,\Psi)}_{n,k})\right] \tag{1.1.45}$$

with $(Y^{(\Phi,\Psi)}_{n,k}, Z^{(\Phi,\Psi)}_{n,k}) := (\Phi_k(X_{n,k}), \Psi_k(X_{n,k}))$, for $1 \leqslant k < q$, and

$$\mathcal{E}_{n,q}(\Phi,\Psi) = \epsilon^{\mathcal{N},y}_{n,q}(\Phi,\Psi) + h\epsilon^{\mathcal{N},z}_{n,q}(\Phi,\Psi) + h\epsilon^{\mathcal{N},a}_{n,q}(\Phi,\Psi) \tag{1.1.46}$$

where

$$\epsilon^{\mathcal{N},y}_{n,q}(\Phi,\Psi) := \inf_{\theta^y} \mathbb{E}\Big[|Y^{(\Phi,\Psi)}_{n,q} - \mathcal{U}_{n,q}(X_{n,q};\theta^y)|^2\Big], \tag{1.1.47}$$

$$\epsilon^{\mathcal{N},a}_{n,q}(\Phi,\Psi) := \inf_{\theta^a} \mathbb{E}\Big[|A^{(\Phi,\Psi)}_{n,q} - \mathcal{A}_{n,q}(X_{n,q};\theta^a)|^2\Big], \tag{1.1.48}$$

$$\epsilon^{\mathcal{N},z}_{n,q}(\Phi,\Psi) := \inf_{\theta^z} \mathbb{E}\Big[|Z^{(\Phi,\Psi)}_{n,q} - \mathcal{V}_{n,q}(X_{n,q};\theta^z)|^2\Big]. \tag{1.1.49}$$

Then, under some regularity assumptions on the solution of (1.1.3) and process X, the following holds

$$\max_n \mathbb{E}\Big[|\bar{Y}_n - \hat{\mathcal{U}}_n(X_n)|^2\Big] + \sum_{n=0}^{N-1} h\mathbb{E}\Big[|\bar{Z}_n - \hat{\mathcal{V}}_n(X_n)|^2\Big] \leqslant C(h^\alpha + N\sum_{n=0}^{N-1}\bar{\mathcal{E}}_n). \tag{1.1.50}$$

where h^α is the discrete time errors of the scheme, and $\bar{\mathcal{E}}_n$ equals to $\mathcal{E}_{n,1}(\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1})$, $\mathcal{E}_{n,2}\left((\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{U}}_{n,2}), (\hat{\mathcal{V}}_{n+1}, \hat{\mathcal{V}}_{n,2})\right), \mathcal{E}_{n,3}\left((\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{U}}_{n,2}, \hat{\mathcal{U}}_{n,3}), (\hat{\mathcal{V}}_{n+1}, \hat{\mathcal{V}}_{n,2}, \hat{\mathcal{V}}_{n,3})\right)$ respectively for Crank-Nicolson scheme, two stage explicit Runge-Kutta scheme, three stage explicit Runge-Kutta scheme. $N\sum_{n=0}^{N-1}\bar{\mathcal{E}}_n$ represents the global approximation errors due to the neural networks.

Note carefully that the second term appearing in the right side of (1.1.50) depends on the considered scheme that we apply and the total number of time steps N, so that it may become large when N increases and has to be balance with the total number of neurons in practical implementaton.

1.1.4.6 Numerical results

In Section 3.4, we analyse the performance of Euler scheme, Crank-Nicolson scheme, two stage explicit Runge-Kutta scheme and three stage explicit Runge-Kutta scheme from a numerical perspective with a special case that the underlying $\{\mathcal{X}_t\}_{0\leqslant t\leqslant T}$ is a 10-dimensional drifted Brownian motion. We plotted the error of Y_0 w.r.t. the time steps in Figure 1.7 and the error of Y_0 w.r.t. the time cost in Figure 1.8 for the 5 schemes mentioned above. As we expected, the order of both explicit Euler scheme and implicit Euler scheme are 1, the Crank-Nicolson scheme and two stage explicit Runge-Kutta scheme are almost order 2 scheme, and the three stage explicit Runge-Kutta scheme converges too fast leads to that we can not observe the order clearly. Finally, We conclude that the Crank-Nicolson scheme is the most efficient one if we want an error smaller than $0.01 \approx 2^{-6.64}$.

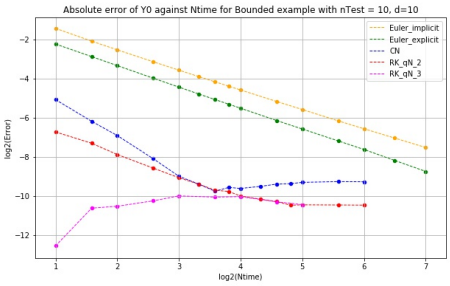

Figure 1.7 – Error against time steps for different schemes

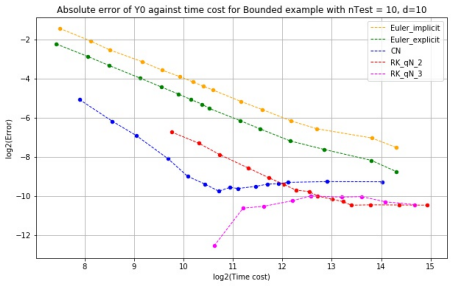

Figure 1.8 – Error against time cost for different schemes

Next, we will pay attention to the numerical results of the order 2 schemes with the underlying is general diffusion process. We have to discretize the forward diffusion process with a second order weak approximation scheme in practice, such as Ninomiya-Victoire scheme [76], see the detail results in Section 3.4.1.2 and Section 3.4.2.2.

1.2 Probabilistic representation for stochastic volatility models

In the second part of the thesis, we establish a probabilistic representation formula for two integration by parts (IBP) formulae for the marginal law of the process for some stochastic volatility models at a given time maturity T. Then an unbiased Monte Carlo path simulation method stems from the probabilistic formulae based on a simple Markov chain evolving along a random time grid given by the jump times

of an independent renewal process, so that it can be used in order to numerically compute options' prices and greeks, in particular delta and vega, for a large class of non-smooth European payoffs. The achieved error is optimal since the computation will be only affected by the statistical error. The main novelty of our approach in comparison to the previous works [42, 3, 4] is that we allow the drift coefficient to be possibly unbounded as it is the case in most stochastic volatility models (Stein-Stein, Heston, ...).

1.2.1 Stochastic volatility model

In mathematical finance, a stochastic volatility model is a model for which the variance given by a stochastic process, instead of being a deterministic process [44]. These models are widely used in mathematical finance to evaluate derivative securities, such as options. Stochastic volatility models are extensions of the Black–Scholes model, for which the volatility is assumed constant over time. This model can not explain long-observed features of the implied volatility surface such as volatility smile and skew. It becomes possible to price derivatives more accurately if we assume that the volatility of the underlying price is a stochastic process.

Many stochastic volatility models have been studied, such as the Heston model [61], the CEV model [30], the SABR volatility model [54], the GARCH model [19] among others. In this work, we consider a two-dimensional stochastic volatility model given by the solution (S, Y) of the following SDE

$$\begin{cases} S_t &= s_0 + \displaystyle\int_0^t r S_s \, ds + \int_0^t \sigma_S(Y_s) S_s \, dW_s, \\ Y_t &= y_0 + \displaystyle\int_0^t b_Y(Y_s) \, ds + \int_0^t \sigma_Y(Y_s) \, dB_s, \\ d\langle B, W \rangle_s &= \rho \, ds \end{cases} \tag{1.2.1}$$

where the coefficients b_Y, σ_S, $\sigma_Y : \mathbb{R} \to \mathbb{R}$ are smooth functions, $r \in \mathbb{R}$, W and B are one-dimensional standard Brownian motions with correlation factor $\rho \in (-1, 1)$ both being defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

We assume that $a_S := \sigma_S^2$, $a_Y := \sigma_Y^2$ and drift b_Y coefficients are infinitely differentiable, and assume a_S and a_Y are bounded. One main improvement is that b_Y is not bounded, in comparison to other works on probabilistic representation, see [42] for killed processes and Agarwal and Gobet [2] for multi-dimensional diffusion processes, for example. Typically, the drift corresponds to a mean reversion term, *i.e.* $b_Y(y) = \lambda(\mu - y)$ for some $\lambda > 0$, as in Stein-Stein model [89]. In addition the volatilities $a_S(x)$ and $a_Y(x)$ should also satisfy some uniform ellipticity conditions. We would like to consider the log-price process $X_t = \ln(S_t)$ instead of the spot price, so that

$$dX_t = \Big(r - \frac{1}{2} a_S(Y_t)\Big) \, dt + \sigma_S(Y_t) \, dW_t,$$

and the couple $(X_t, Y_t)_{t \in [0,T]}$ has initial conditions $(X_0, Y_0) = (\ln(s_0), y_0)$.

1.2.2 Probabilistic representation

1.2.2.1 Background

The Probabilistic representation method originally developed by Bally, Kohatsu-Higa, Anderson in [4, 3]. For sake of simplicity, we will consider the one dimensional case and $\sigma(t,x) = \sigma, b(x) \in \mathcal{C}_b^1(\mathbb{R})$. Thus, we introduce the one-step Euler scheme

$$\bar{X}_t^{s,x} = x + \sigma(W_t - W_s).$$

Assume that $[0,T] \times \mathbb{R} \ni (s,x) \mapsto u(s,x) = \mathbb{E}[f(X_s^x)]$ is the unique solution of the PDE:

$$\begin{cases} (\partial_s - \mathcal{L})u(s,x) &= 0, \\ u(0,x) &= f(x). \end{cases}$$

Apply Itô's rule to $u(T-t, \bar{X}_t^{0,x})_{t\in[0,T]}$, we have

$$\begin{aligned}
\mathbb{E}\Big[f(\bar{X}_T^{0,x})\Big] &= \mathbb{E}\Big[u(0,\bar{X}_T^{0,x})\Big] \\
&= u(T,x) + \int_0^T \mathbb{E}\Big[-\partial_s u(T-r,\bar{X}_r^{0,x}) + \frac{1}{2}\sigma^2\partial_x^2 u(T-r,\bar{X}_r^{0,x})\Big]\Big] \,\mathrm{d}r \\
&= u(T,x) + \int_0^T \mathbb{E}\Big[(-\mathcal{L} + \frac{1}{2}\sigma^2\partial_x^2)u(T-r,\bar{X}_r^{0,x})\Big] \,\mathrm{d}r \\
&= u(T,x) - \int_0^T \mathbb{E}\big[b(\bar{X}_r^{0,x})\partial_x u(T-r,\bar{X}_r^{0,x})\big] \,\mathrm{d}r \\
&= u(T,x) - \int_0^T \int_{\mathbb{R}} b(y)\partial_x u(T-r,y)\frac{e^{-\frac{(y-x)^2}{2\sigma^2 r}}}{\sqrt{2\pi\sigma^2 r}} \,\mathrm{d}y\,\mathrm{d}r \\
&= u(T,x) + \int_0^T \int_{\mathbb{R}} u(T-r,y)(b(y) - b'(y)\frac{y-x}{\sigma^2 r})\frac{e^{-\frac{(y-x)^2}{2\sigma^2 r}}}{\sqrt{2\pi\sigma^2 r}} \,\mathrm{d}y\,\mathrm{d}r \\
&=: u(T,x) - \int_0^T \mathbb{E}\big[\bar{\theta}_r(x,\bar{X}_r^{0,x})u(T-r,\bar{X}_r^{0,x})\big] \,\mathrm{d}r.
\end{aligned}$$

with the notation $\bar{\theta}_r(x,y) = -(b(y) - b'(y)\frac{y-x}{\sigma^2 r})$. Hence we get

$$u(T,x) = \mathbb{E}\Big[f(\bar{X}_T^{0,x})\Big] + \int_0^T \mathbb{E}\big[\bar{\theta}_r(x,\bar{X}_r^{0,x})u(T-r,\bar{X}_r^{0,x})\big] \,\mathrm{d}r. \tag{1.2.2}$$

It could be shown that for $r \in [0,T]$,

$$|\mathbb{E}\big[\bar{\theta}_r(x,\bar{X}_r^{0,x})u(T-r,\bar{X}_r^{0,x})\big]\,| \in L^1([0,T]).$$

Then, by the same argument,

$$\begin{aligned}
u(T-r,x) &= \mathbb{E}\Big[f(\bar{X}_{T-r}^{0,x})\Big] + \int_0^{T-r} \mathbb{E}\big[\bar{\theta}_{r_1}(x,\bar{X}_{r_1}^{0,x})u(T-r-r_1,\bar{X}_{r_1}^{0,x})\big] \,\mathrm{d}r_1 \\
&= \mathbb{E}\Big[f(\bar{X}_{T-r}^{0,x})\Big] + \int_r^T \mathbb{E}\Big[\bar{\theta}_{r_1-r}(x,\bar{X}_{r_1-r}^{0,x})u(T-r_1,\bar{X}_{r_1-r}^{0,x})\Big] \,\mathrm{d}r_1 \\
&= \mathbb{E}\Big[f(\bar{X}_{T-r}^{0,x})\Big] + \int_r^T \mathbb{E}\big[\bar{\theta}_{r_1-r}(x,\bar{X}_{r_1}^{r,x})u(T-r_1,\bar{X}_{r_1}^{r,x})\big] \,\mathrm{d}r_1,
\end{aligned}$$

so that

$$u(T,x) = \mathbb{E}\Big[f(\bar{X}_T^{0,x})\Big] + \int_0^T \mathbb{E}\Big[\bar{\theta}_{r_1}(x,\bar{X}_{r_1}^{0,x})f(\bar{X}_T^{0,x})\Big]\,\mathrm{d}r_1 + \int_0^T\int_{r_1}^T \mathbb{E}\big[\bar{\theta}_{r_1}(x,\bar{X}_{r_1}^{0,x})\bar{\theta}_{r_2-r_1}(\bar{X}_{r_1}^{0,x},\bar{X}_{r_2}^{0,x})u(T-r_2,\bar{X}_{r_2}^{0,x})\big]\,\mathrm{d}r_1\,\mathrm{d}r_2.$$

Repeating N time the same arguments, we can prove by induction

$$u(T,x) = \sum_{n=0}^{N}\int_{\Delta_n(T)} \mathbb{E}\left[f(\bar{X}_T^{0,x})\prod_{k=1}^{n}\bar{\theta}_{r_k-r_{k-1}}(\bar{X}_{r_{k-1}}^{0,x},\bar{X}_{r_k}^{0,x})\right]\mathrm{d}r_1\cdots\mathrm{d}r_n \tag{1.2.3}$$
$$+\int_{\Delta_{N+1}(T)} \mathbb{E}\left[u(T-r_{N+1},\bar{X}_T^{0,x})\prod_{k=1}^{n}\bar{\theta}_{r_k-r_{k-1}}(\bar{X}_{r_{k-1}}^{0,x},\bar{X}_{r_k}^{0,x})\right]\mathrm{d}r_1\cdots\mathrm{d}r_{N+1},$$

where $\Delta_n(T) := \{(r1,\cdots,r_n)\in[0,T]^n | 0\leqslant r_1\leqslant\cdots\leqslant r_n\leqslant T\}$ and by convention $r_0=0, \prod_{\varnothing}=1$. Under some good controls, let $N\to\infty$ on the previous identity

$$u(T,x) = \sum_{n\geqslant 0}\int_{\Delta_n(T)} \mathbb{E}\left[f(\bar{X}_T^{0,x})\prod_{k=1}^{n}\bar{\theta}_{r_k-r_{k-1}}(\bar{X}_{r_{k-1}}^{0,x},\bar{X}_{r_k}^{0,x})\right]\mathrm{d}r_1\cdots\mathrm{d}r_n.$$

In order to get a probabilistic representation of the series, one remarks that for a Poisson process N with intensity λ, independent of W, when $N_T=n$, its jump times $\zeta_1,\cdots,\zeta_n$ are distributed as the order statistics of n i.i.d. uniform random variable on $[0,T]$, that is

$$\mathbb{P}(N_T=n,\zeta_1\in\mathrm{d}r_1,\cdots,\zeta_n\in\mathrm{d}r_n) = \lambda^n e^{-\lambda T}\,\mathrm{d}r_1\cdots\mathrm{d}r_n$$

on $\Delta_b(T)$. As a consequence,

$$\begin{aligned}\mathbb{E}\Big[f(X_T^{0,x})\Big] &= u(T,x)\\ &= \sum_{n\geqslant 0} e^{\lambda T}\mathbb{E}\left[f(\bar{X}_T^{0,x})\prod_{k=1}^{n}\lambda^{-1}\bar{\theta}_{r_k-r_{k-1}}(\bar{X}_{r_{k-1}}^{0,x},\bar{X}_{r_k}^{0,x})\mathbf{1}_{N_T=n}\right]\\ &= e^{\lambda T}\mathbb{E}\left[f(\bar{X}_T^{0,x})\prod_{k=1}^{n}\lambda^{-1}\bar{\theta}_{r_k-r_{k-1}}(\bar{X}_{r_{k-1}}^{0,x},\bar{X}_{r_k}^{0,x})\right].\end{aligned}$$

The probabilistic representation allows to compute $u(T,x)=\mathbb{E}\Big[f(X_T^{0,x})\Big]$ without any discretization error but only a statistical error.

1.2.2.2 Our contributions

We are interested in establishing a probabilistic representation formula for the price of a European option with maturity $T>0$ and payoff $h(S_T,Y_T)$, given by

$$\mathbb{E}\left[h(S_T,Y_T)\right],$$

as well as integration by parts (IBP) formulae for the sensitivities (Greeks) of the option, given by

$$\partial_{s_0}\mathbb{E}\left[h(S_T, Y_T)\right] \quad \text{and} \quad \partial_{y_0}\mathbb{E}\left[h(S_T, Y_T)\right],$$

from which stem an unbiased Monte Carlo simulation method.

We establish a probabilistic representation formula for the marginal law (S_T, Y_T), $T > 0$ based on a simple Markov chain evolving along a random time grid given by the jump times of an independent renewal process. This formula provides an unbiased Monte Carlo method. It is inspired by the probabilistic representation formula derived in Bally and Kohatsu-Higa [4] and others [3, 60, 42, 2] for multi-dimensional diffusion process, one-dimensional killed processes and of some Lévy driven SDEs with bounded drift, diffusion and jump coefficients. The main novel challenge is to tackle the case where the volatility drift b_Y is unbounded. In order to overcome this difficulty, we freeze the coefficients b_Y , σ_S and σ_Y along the flow of the ordinary differential equation (ODE) $\frac{\mathrm{d}m_t}{\mathrm{d}t} = b_Y(m_t)$, $m_0 = y_0$ obtained by removing the diffusion term in the dynamics of Y. The underlying Markov chain $(\bar{X}, \bar{Y})$ on which the probabilistic representation is based, is then obtained from

$$\begin{aligned}
\bar{X}_t^{x_0} &= x_0 + \int_0^t (r - \frac{1}{2}a_S(m_s))\, ds + \int_0^t \sigma_S(m_s)\ \mathrm{d}W_s, \\
\bar{Y}_t^{y_0} &= y_0 + \int_0^t b_Y(m_s)\, ds + \int_0^t \sigma_Y(m_s)\ \mathrm{d}B_s, \\
d\langle W, B\rangle_s &= \rho\ \mathrm{d}s.
\end{aligned}$$

Assume $\tau = (\tau_n)_{n\geqslant 0}, \tau_0 = 0$ is a non-decreasing sequence of $\mathbb{R}_+$ modelling random jump times and let $N = (N_t)_{t\geqslant 0}$ be the renewal proces, defined by $N_t := \sum_{n\geqslant 1} \mathbf{1}_{\{\tau_n \leqslant t\}}$. N is independent of the two Brownian motions W and B. We discretise the process $(\bar{X}, \bar{Y})$ using a Euler scheme on the random time grid $(\zeta_i)_{i\geqslant 0}$ with $\zeta_0 = 0$ and $\zeta_i = \tau_i \wedge T$, namely

$$\begin{aligned}
\bar{X}_{i+1} &= \bar{X}_i + \Big(r(\zeta_{i+1} - \zeta_i) - \frac{1}{2}a_{S,i}\Big) + \sigma_{S,i} Z_{i+1}^1, \\
\bar{Y}_{i+1} &= m_i + \sigma_{Y,i}\Big(\rho_i Z_{i+1}^1 + \sqrt{1-\rho_i^2}\, Z_{i+1}^2\Big).
\end{aligned}$$

where

$$\begin{aligned}
a_{S,i} := \sigma_{S,i}^2 &= \int_0^{\zeta_{i+1}-\zeta_i} a_S(m_s(\bar{Y}_i))\ \mathrm{d}s, \\
a_{Y,i} := \sigma_{Y,i}^2 &= \int_0^{\zeta_{i+1}-\zeta_i} a_Y(m_s(\bar{Y}_i))\ \mathrm{d}s, \\
\sigma_{S,Y,i} &:= \int_0^{\zeta_{i+1}-\zeta_i} (\sigma_S\sigma_Y)(m_s(\bar{Y}_i))\ \mathrm{d}s, \\
\rho_i &:= \rho\frac{\sigma_{S,Y,i}}{\sigma_{S,i}\sigma_{Y,i}}, \\
m_i &:= m_{\zeta_{i+1}-\zeta_i}(\bar{Y}_i),
\end{aligned}$$

where $Z = (Z_n^1, Z_n^2)_{n\geqslant 1}$ is a sequence of i.i.d. random variables of law $\mathcal{N}(0, I_2)$ which is independent of (W, B) and $\sigma'_{S,i},\ \sigma'_{Y,i},\ \sigma'_{S,Y,i},\ \rho'_i$ and m'_i are the partial derivatives of $\sigma_{S,i},\ \sigma_{Y,i},\ \sigma_{S,Y,i},\ \rho_i,\ m_i$ with respect to $\bar{Y}_i$.

The approximation process $(\bar{X}, \bar{Y})$ is a Markov chain with respect to the filtration defined by $\mathcal{G} = (\mathcal{G}_i)_{i\geqslant 0}$ where $\mathcal{G}_i = \sigma(Z_j,\ 0 \leqslant j \leqslant i)$ for $i \geqslant 1$ and $\mathcal{G}_0$ is the trivial $\sigma-$field. We define $\zeta^n = (\zeta_0, \cdots, \zeta_n), \tau^n = (\tau_0, \cdots, \tau_n)$ for $n \in \mathbb{N}^+$. We first define the set $\mathbb{S}_{i,n}(\bar{X}, \bar{Y}),\ n \in \mathbb{N},\ i \in \{0, \cdots, n\}$ as the space of random variables H satisfying:

- $H = h(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1})$ on $\{N_T = n\}$, where $\zeta^{n+1} := (0 = \zeta_0, \zeta_1, \cdots, \zeta_n, \zeta_{n+1} = T)$
- For all $\mathbf{s}_{n+1} \in \Delta_{n+1}(T) := \{\mathbf{s}_{n+1} \in [0,T]^n : 0 \leqslant \mathbf{s}_1 < \cdots \mathbf{s}_{n+1} \leqslant T\}$, the function $h(., \mathbf{s}_{n+1})$ is in $\mathcal{C}_p^\infty(\mathbb{R}^4)$.

We also define the derivative operator $\mathcal{D}_{i+1}^{(\alpha)} H,\ \alpha \in \{1, 2\}$ for $H \in \mathbb{S}_{i,n}(\bar{X}, \bar{Y})$:

$$\mathcal{D}_{i+1}^{(1)} H = \partial_{\bar{X}_{i+1}} H \ \text{ and } \ \mathcal{D}_{i+1}^{(2)} H = \partial_{\bar{Y}_{i+1}} H.$$

We then develop a tailor-made Malliavin calculus for the Euler scheme $(\bar{X}_i, \bar{Y}_i)_{0\leqslant i\leqslant N_T+1}$.

$$\mathcal{I}_{i+1}^{(1)}(H) = H\Big[\frac{Z_{i+1}^1}{\sigma_{S,i}(1-\rho_i^2)} - \frac{\rho_i}{1-\rho_i^2}\frac{\rho_i Z_{i+1}^1 + \sqrt{1-\rho_i^2} Z_{i+1}^2}{\sigma_{S,i}}\Big] - \mathcal{D}_{i+1}^{(1)} H,$$
$$\mathcal{I}_{i+1}^{(2)}(H) = H\Big[\frac{\rho_i Z_{i+1}^1 + \sqrt{1-\rho_i^2} Z_{i+1}^2}{\sigma_{Y,i}(1-\rho_i^2)} - \frac{\rho_i}{1-\rho_i^2}\frac{Z_{i+1}^1}{\sigma_{Y,i}}\Big] - \mathcal{D}_{i+1}^{(2)} H.$$

Then, for a multi-index $\alpha = (\alpha_1, ..., \alpha_p),\ \alpha_i \in \{1, 2\}$ with length p and an index $\alpha_{p+1} \in \{1, 2\}$, we have the following relations:

$$\mathcal{I}_{i+1}^{(\alpha,\alpha_{p+1})}(H) = \mathcal{I}_{i+1}^{(\alpha_{p+1})}(\mathcal{I}_{i+1}^{(\alpha)}(H)), \quad \mathcal{D}_{i+1}^{(\alpha,\alpha_{p+1})} H = \mathcal{D}_{i+1}^{(\alpha_{p+1})}(\mathcal{D}_{i+1}^{(\alpha)} H).$$

Setting $\mathbb{E}_{i,n}[X] = \mathbb{E}[X|\mathcal{G}_i, \tau^{n+1}, N_T = n]$ for $X \in L^1(\mathbb{P})$ and $i \in \{0, \cdots, n\}$, we obtain the following duality identity

$$\mathbb{E}_{i,n}\Big[\mathcal{D}_{i+1}^{(\alpha)} f(\bar{X}_{i+1}, \bar{Y}_{i+1}) H\Big] = \mathbb{E}_{i,n}\Big[f(\bar{X}_{i+1}, \bar{Y}_{i+1}) \mathcal{I}_{i+1}^{(\alpha)}(H)\Big].$$

We denote $\mathcal{B}_\gamma(\mathbb{R}^2)$ the set of Borel measurable map $h : \mathbb{R}^2 \to \mathbb{R}$ satisfying the exponential growth assumption at infinity. Under some assumptions that will be stated in Part II of this thesis, the law of the couple (X_T, Y_T) satisfies the following probabilistic representation: for all $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$ for some $\gamma > 0$, the price of an option exercised at time T with payoff $h(X_T, Y_T)$ satisfies:

$$\mathbb{E}[h(X_T, Y_T)] = \mathbb{E}\Big[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \prod_{i=1}^{N_T+1} \theta_i\Big],$$

where the random variables θ_i are in $\mathbb{S}_{i-1,n}(\bar{X},\bar{Y})$ on the set $\{N_T = n\}$. We have $\theta_{N_T+1} = (1 - F(T - \zeta_{N_T}))^{-1}$, and for $i = 1, \cdots, N_T$,

$$\theta_i = (f(\zeta_i - \zeta_{i-1}))^{-1}\Big[\mathcal{I}_i^{(1,1)}(c_S^i) - \mathcal{I}_i^{(1)}(c_S^i) + \mathcal{I}_i^{(2,2)}(c_Y^i) + \mathcal{I}_i^{(2)}(b_Y^i) + \mathcal{I}_i^{(1,2)}(c_{Y,S}^i)\Big].$$

Last, if N is a renewal process with jump times of distribution $Beta(1/2, 1)$, then for all $p \geqslant 1$ and $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$ for some $\gamma p > 0$, the random variable $h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \prod_{i=1}^{N_T+1} \theta_i$ admits a finite $L^p(\mathbb{P})$ moment.

1.2.3 Integration by parts formulae

As usual, we define the derivative with respect to the spot price of the underlying asset (resp. its volatility) by *Delta* (resp. *Vega*). We are interested in establishing a Bismut-Elworthy-Li-type formulae for the two quantities:

$$\partial_{s_0}\mathbb{E}[h(X_T, Y_T)] \text{ and } \partial_{y_0}\mathbb{E}[h(X_T, Y_T)],$$

where $s_0 = \exp(X_0),\ y_0 = Y_0$.

The central idea is to exchange the order of the derivative and the expectation, so that the two Greeks can be computed by a Monte Carlo simulation. Applying the probabilistic representation, the Greeks write

$$\partial_{s_0}\mathbb{E}\Big[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \prod_{i=1}^{N_T+1} \theta_i\Big] \text{ and } \partial_{y_0}\mathbb{E}\Big[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \prod_{i=1}^{N_T+1} \theta_i\Big].$$

The second step is to apply an appropriate integration by parts formula in order to differentiate before averaging. However, the usual Malliavin's IBP formula cannot be applied here due to integrability issues. Following the ideas developed in [42] for killed diffusion processes with bounded drift coefficient, we first transfer the derivatives forward in time on each random time intervals $[\zeta_i, \zeta_{i+1}]$, $i = 0, \cdots, N_T$, as explained in the following graph.

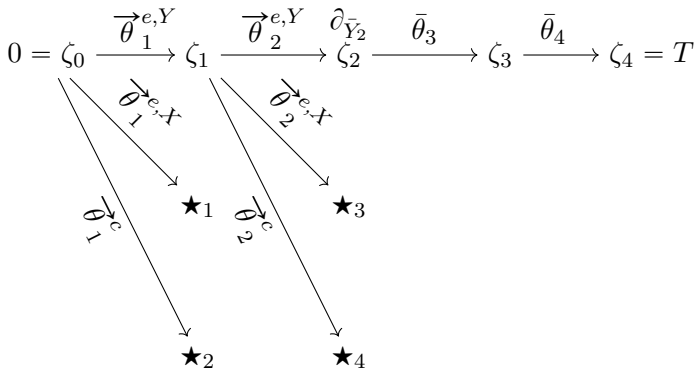

Figure 1.9 – Transfer the derivatives forward in time on each random intervals with $N_T = 3$

Then, we perform a local IBP formula on each random time interval $[\zeta_i, \zeta_{i+1}]$, $i = 0, \cdots, N_T$. For instance, on the last time interval $[\zeta_{N_T}, \zeta_{N_T+1}] = [\zeta_{N_T}, T]$, on the set $\{N_T = n\}$, to obtain the IBP of *Vega* one has

$$\begin{aligned}
&\partial_{\bar{Y}_n} \mathbb{E}_{n,n}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1})\theta_{n+1}\Big] \\
&= \mathbb{E}_{n,n}\Big[\partial_{\bar{Y}_{n+1}} h(\bar{X}_{n+1}, \bar{Y}_{n+1})\, \overrightarrow{\theta}^{e,Y}_{n+1}\Big] + \mathbb{E}_{n,n}\Big[\partial_{\bar{X}_{n+1}} h(\bar{X}_{n+1}, \bar{Y}_{n+1})\, \overrightarrow{\theta}^{e,X}_{n+1}\Big] \\
&\quad + \mathbb{E}_{n,n}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1})\, \overrightarrow{\theta}^{c}_{n+1}\Big] \\
&= \mathbb{E}_{n,n}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1})(\mathcal{I}^{(2)}(\overrightarrow{\theta}^{e,Y}_{n+1}) + \mathcal{I}^{(1)}(\overrightarrow{\theta}^{e,X}_{n+1}) + \overrightarrow{\theta}^{c}_{n+1})\Big]
\end{aligned}$$

for some new weights $\overrightarrow{\theta}^{e,Y}_{n+1}$, $\overrightarrow{\theta}^{e,X}_{n+1}$, $\overrightarrow{\theta}^{c}_{n+1}$.

Finally, we combine each local IBP formula in an adequate manner to establish the global IBP formula, using the fact that

$$\mathbb{E}[h(X_T, Y_T)] = \sum_{n\geqslant 0} \mathbb{E}\left[\mathbb{E}\Big[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \prod_{i=1}^{N_T+1} \bar{\theta}_i | \tau_{n+1}\Big]\mathbf{1}_{\{N_T=n\}}\right]$$

Under appropriate assumptions **(AR)** and **(ND)** in Part II, for all $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$ for some $\gamma > 0$ and all $(s_0, y_0) \in \mathbb{R}^2$, the law of the couple (X_T, Y_T) satisfies the following Bismut-Elworthy-Li type formulae:

$$s_0 T \partial_{s_0} \mathbb{E}\Big[h(X_T, Y_T)\Big] = \mathbb{E}\Big[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \sum_{k=1}^{N_T+1} (\zeta_k - \zeta_{k-1})\, \overrightarrow{\theta}^{\mathcal{I}^{(1),N_T+1}_k}_k\Big]$$

and

$$\begin{aligned}
&T\partial_{y_0} \mathbb{E}\Big[h(X_T, Y_T)\Big] \\
&= \mathbb{E}\left[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \sum_{k=1}^{N_T+1} (\zeta_k - \zeta_{k-1})\Big(\overrightarrow{\theta}^{\mathcal{I}^{(2),N_T+1}_k}_k + \sum_{j=1}^{k} \overrightarrow{\theta}^{C_j^{N_T+1}}_j + \overrightarrow{\theta}^{\mathcal{I}^{(1),N_T+1}_k}_j\Big)\right],
\end{aligned}$$

where $\overrightarrow{\theta}^{\mathcal{I}^{(1),n+1}_k}_k$, $\overrightarrow{\theta}^{C_j^{n+1}}_j$, $\overrightarrow{\theta}^{\mathcal{I}^{(2),n+1}_k}_k$ and $\overrightarrow{\theta}^{\mathcal{I}^{(1),n+1}_k}_j$ with $n \geqslant 0$ on $\{N_T = n\}$, $1 \leqslant j \leqslant k \leqslant n+1$, are explicit functions of the parameters of the model and the weights θ_i. This implies that the random variables that appear inside the expectations on the right-hand side of the IBP formula can be perfectly simulated. Therefore, *Delta* and *Vega* can be computed by an unbiased Monte-Carlo simulation method with optimal complexity.

1.2.4 Numerical results

Finally, we numerically compute the *prices* and *Greeks* of the following model

$$\begin{aligned}
dS_t &= rS_t\, dt + \sigma_S(Y_t) S_t\, dW_t, \\
dY_t &= b_Y(Y_t)\, dt + \sigma_Y(Y_t) dB_t, \\
d\langle B, W\rangle_t &= \rho dt, \qquad \rho \in (-1, 1),
\end{aligned}$$

with unbounded mean-reversion drift $b_Y(x) = \lambda_Y(\mu - x)$, for both European call options (payoffs: $h(x, y) = (\exp(x) - K)_+$) and digital call options which has non-continuous payoff functions $h(x, y) = \mathbf{1}_{\{\exp(x) \geqslant K\}}$. We compare the results of the unbiased Monte Carlo method by Exponential sampling (a Poisson process with intensity parameter $\lambda = 0.5$, then $\mathbb{E}[N_T] = 1.25$) and Beta sampling (a renewal process with $[0, 2]-$valued $Beta(0.5, 1)$ jump times, then $\mathbb{E}[N_T] = 1.79$) to Euler scheme of three different models:

- Black-Scholes model where $\sigma_S(x) = \sigma_S$ is a constant.
- A Stein-Stein type model where $\sigma_S(x) = \sigma_1 x + \sigma_2$, we choose σ_1, σ_2 such that $\sigma_S(x)$ is positive in the domain of x.
- A model with a periodic diffusion coefficient function where $\sigma_S(x) = \sigma_1 \cos(x) + \sigma_2$, $\sigma_2 - \sigma_1 > 0$.

We fix the parameters as follows: $T = 0.5$, $r = 0.03$, $K = 1.5$, $x_0 = \ln(s_0) = 0.4$, $Y_0 = 0.2$, $\sigma_Y(.) \equiv \sigma_Y = 0.2$, $\lambda_Y = 0.5$, $\mu = 0.3$ and $\rho = 0.6$. We perform M_1 paths such that $\mathbb{E}[N_T] \times M_1 = 3.2 \times 10^7$ for the unbiased Monte Carlo method in both Exponential sampling and Beta sampling case. And for the Euler-Maruyama approximation scheme, we simulate $M_2 = 160000$ Monte Carlo simulations paths and set mesh size $\delta = T/n$ where $n = 200$, we also have $M_2 n = 3.2 \times 10^7$.

For the Black-Scholes model, we have explicit formulas for the price and the *Greeks*. The numerical results for the price, *Greeks*, together with their variance, 95% half width of the confidence interval of the unbiased Monte Carlo estimation and the Monte Carlo Euler-Maruyama scheme of a Call option in the Black-Scholes model for different values of σ_S are provided in the three tables below. Additional numerical results are presented in Section 4.5.

σ_S	B-S	Euler Scheme			Exponential sampling			Beta sampling		
	formula	Price	Half-width	Variance	Price	Half-width	Variance	Price	Half-width	Variance
0.25	0.111804	0.111853	0.000860286	0.0308244	0.112196	0.000124112	0.102648	0.112199	0.000154064	0.110598
0.3	0.132621	0.132808	0.0010515	0.0460493	0.133193	0.000152038	0.15404	0.133036	0.000187336	0.163524
0.4	0.174152	0.173559	0.00144315	0.0867423	0.174754	0.000208983	0.291037	0.174711	0.000257441	0.308813
0.6	0.256572	0.255388	0.00235625	0.231233	0.257287	0.000334903	0.747423	0.256978	0.0004127	0.793617

Table 1.4 – Comparison between the unbiased Monte Carlo estimation and the Monte Carlo Euler-Maruyama scheme for the price of a Call option in the Black-Scholes model for different values of σ_S.

σ_S	B-S	Euler Scheme			Exponential sampling			Beta sampling		
	formula	Delta	Half-width	Variance	Delta	Half-width	Variance	Delta	Half-width	Variance
0.25	0.556589	0.55675	0.00280539	0.327789	0.554992	0.000895101	5.33915	0.555192	0.00114054	6.0612
0.3	0.560018	0.560534	0.00290622	0.351775	0.558285	0.000923515	5.6835	0.557974	0.00116621	6.33719
0.4	0.569512	0.570228	0.00311011	0.402864	0.567568	0.000978965	6.38649	0.567091	0.00123	7.04938
0.6	0.592743	0.590041	0.00358714	0.535925	0.589	0.0010899	7.91588	0.587681	0.00137469	8.80548

Table 1.5 – Comparison between the unbiased Monte Carlo estimation and the Monte Carlo Euler-Maruyama scheme for the Delta of a Call option in the Black-Scholes model for different values of σ_S.

σ_S	B-S	Exponential sampling			Beta sampling		
	formula	Vega	Half-width	Variance	Vega	Half-width	Variance
0.25	0	0.000690222	0.00115103	8.82877	-0.000559242	0.00128448	7.68766
0.3	0	0.00182175	0.00137953	12.6821	0.000500579	0.00156401	11.3978
0.4	0	-0.00163321	0.00189888	24.0283	-0.000817515	0.00215655	21.6701
0.6	0	-0.000830748	0.00300346	60.1136	-0.001055	0.00340386	53.9862

Table 1.6 – Comparison between the unbiased Monte Carlo estimation for the Vega of a Call option in the Black-Scholes model for different values of σ_S.

Part I

Schemes for solving BSDEs

Chapter 2

A learning scheme by sparse grids and Picard approximations for semilinear parabolic PDEs

The content of this chapter is from an article in collaboration with Jean-François Chassagneux, Noufel Frikha, Chao Zhou [23]. Submitted to IMA Journal of Numerical Analysis.

Contents

2.1 Introduction

In the present work, we are interested in the numerical approximation in high dimension d of the solution to the semilinear parabolic PDE

$$\begin{cases} \partial_t u(t,x) + \mathcal{L}u(t,x) + f(u(t,x), \sigma^\top(x)\nabla_x u(t,x)) = 0, \quad (t,x) \in [0,T) \times \mathbb{R}^d, \\ u(T,x) = g(x), \quad x \in \mathbb{R}^d \end{cases} \tag{2.1.1}$$

where $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$, $g : \mathbb{R}^d \to \mathbb{R}$ are measurable functions and $\mathcal{L}$ is the infinitesimal generator of the forward diffusion process with dynamics

$$\mathrm{d}\mathcal{X}_t = b(\mathcal{X}_t)\,\mathrm{d}t + \sigma(\mathcal{X}_t)\,\mathrm{d}\mathcal{W}_t \tag{2.1.2}$$

and defined, for a smooth function φ, by

$$\mathcal{L}\varphi(t,x) := b(x) \cdot \nabla_x \varphi(t,x) + \frac{1}{2}\mathrm{Tr}[(\sigma\sigma^\top)(x)\nabla_x^2 \varphi(t,x)]. \tag{2.1.3}$$

Here, $\mathcal{W}$ is a d-dimensional Brownian motion defined on a complete probability space $(\mathfrak{O}, \mathcal{A}, \mathbb{P})$, $b : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \to \mathcal{M}_d$ are measurable functions, $\mathcal{M}_d$ being the set of $d \times d$ matrix. The initial condition $\mathcal{X}_0$ is a square integrable random variable independent from the Brownian Motion $\mathcal{W}$. We denote by $(\mathcal{F}_t)_{0 \leqslant t \leqslant T}$ the filtration generated by $\mathcal{W}$ and $\mathcal{X}_0$, augmented with $\mathbb{P}$ null sets.

Developing efficient algorithms for the numerical approximation of high-dimensional non-linear PDEs is a challenging task that has attracted considerable attention from the research community in the last two decades. We can quote various approaches (limiting to the "stochastic" ones) that have proven to be efficient in a high dimensional setting: branching methods, see e.g. [59], machine learning methods (especially using deep neural networks), see e.g. [56], and full history recursive multilevel Picard method (abbreviated MLP in the literature) see e.g. [66]. This is a very active field of research, we refer to the recent survey papers [57, 8] for more references and an overview of the numerical and theoretical results available. We focus now more on one stream of research which uses the celebrated link between semilinear parabolic PDEs of the form (2.1.1) and BSDEs. This connection, initiated in [81], is as follows: denoting by u a classical solution to (2.1.1), we have that

$(u(t,\mathcal{X}_t), \sigma^\top(\mathcal{X}_t)\nabla_x u(t,\mathcal{X}_t)) = (\mathcal{Y}_t, \mathcal{Z}_t)$ where the pair $(\mathcal{Y},\mathcal{Z})$ is the $\mathbb{R}\times\mathbb{R}^d$-valued and $(\mathcal{F}_t)$-adapted process solution to the BSDE with dynamics

$$\mathcal{Y}_t = g(\mathcal{X}_T) + \int_t^T f(\mathcal{Y}_s,\mathcal{Z}_s)\,\mathrm{d}s - \int_t^T \mathcal{Z}_s\cdot\,\mathrm{d}\mathcal{W}_s\,,\ 0\leqslant t\leqslant T\,, \tag{2.1.4}$$

so that, the original problem boils down to the numerical approximation of the above stochastic system. Various strategies have been used to numerically approximate the stochastic system $(\mathcal{X}_t,\mathcal{Y}_t,\mathcal{Z}_t)_{t\in[0,T]}$. The most studied one is based on a time discretization of (2.1.4) leading to a backward programming algorithm to approximate (Y,Z), as exposed in e.g [17, 93] (see the references therein for early works). This involves computing a sequence of conditional expectations and various methods have been developed: Malliavin calculus based methods [17, 33, 64], optimal quantization methods [5, 6, 80], cubature methods [31, 32, 25] and (linear) regression methods, see among others [49, 51, 50]. It is acknowledged that such approaches will be feasible for problems up to dimension 10. This limitation is a manifestation of the so-called "curse of dimensionality". Recently, non-linear regression methods using deep neural networks were succesfuly combined with this approach and proved to be capable of tackling problems in high dimension [65]. However, other strategies have been introduced in the last five years or so to approximate (2.1.4) trying to adopt a "forward point of view". Relying on Wiener chaos expansion and Picard iteration, [18, 45] introduced a method that notably works in non-Markovian setting but is still impacted by the curse of dimension. A key step forward has been realized by the so called *deep BSDEs solver* introduced in [56]. Interpreting the resolution of a BSDEs as an optimisation problem, it relies on the expressivity of deep neural network and well established SGD algorithms to show great performance in practice. More precisely, in this approach, the $\mathcal{Y}$-process is now interpreted as a forward SDE controlled by the $\mathcal{Z}$-process. Then, an Euler-Maruyama approximation scheme is derived in which the derivative of the solution u appearing in the non-linear function f (through the $\mathcal{Z}$-process) is approximated by a multi-layer neural network. The optimal weights are then computed by minimizing the mean-squared error between the value of the approximation scheme at time T and a good approximation of the target $g(\mathcal{X}_T)$ using stochastic gradient descent algorithms. Again, this kind of deep learning technique seems to be very efficient to numerically approximate the solution to semi-linear parabolic PDEs in practice. However a complete theory concerning its theoretical performance is still not achieved [8]. One important observation is that, due to highly non-linear specification, the optimisation problem that has to be solved in practice, has no convexity property. The numerical procedure designed can only converge to local minima, whose properties (with respect to the approximation question) are still not completely understood.

Inspired by this new forward approach, we introduce here an algorithm which is shown to converge to a global minimum. This, of course, comes with a price. First, we move from the deep neural networks approximation space to a more classical linear specification of the approximation space. However, due the non-linearity in the BSDE driver, the global optimisation problem to be solved is still non-convex. To circumvent this issue, we employ a *Picard iteration procedure*. The overall procedure becomes then a sequence of linear-quadratic optimisation problems which

are solved by a SGD algorithm. Our first main result is a control of the global error between the *implemented algorithm* and the solution to the BSDE which notably shows the convergence of the method under some smallness conditions, see Theorem 2.2.1. In particular, contrary to [56, 58] or [65], our result takes into account the error induced by the SGD algorithm. In our numerical experiments, we rely on sparse grid approximation spaces which are known to be well-suited to deal with high-dimensional problems. Under the framework of periodic coefficients, we establish as our second main result, an upper bound on the global complexity for our *implemented algorithm*, see Theorem 2.3.1. We notably prove that the curse of dimensionality is tamed in the sense that the complexity is of order $\varepsilon^{-p}|\log(\varepsilon)|^{q(d)}$, where p is a constant which does not depend on the PDE dimension and $d \mapsto q(d)$ is an affine function. We also demonstrate numerically the efficiency of our methods in high dimensional setting.

The rest of the Chapter is organized as follows. In Section 2.2, we first recall the *deep BSDEs solver* of [56] but adapted to our framework. Namely, we use a linear specification of the approximation space together with SGD algorithms. For sake of clarity, we denote this method: the *direct algorithm*. Then, we introduce our new numerical method: the *Picard algorithm*. We present our main assumptions on the coefficients and state our main convergence results. In Section 2.3, we use sparse grid approximation with the *direct and Picard algorithms*, using two types of basis functions: pre-wavelet [14] and modified hat function [43]. We discuss their numerical performances in practice through various test examples. We also compare them with some deep learning techniques [56, 65]. We also state our main theoretical complexity result. Section 2.4 is devoted to the theoretical analysis required to establish our main theorems: all the proofs are contained in this section. Finally, we give a complete list of the algorithm parameters that have been used to obtain the numerical results in Appendix 2.5.1.

Notation: Elements of $\mathbb{R}^q$ are seen as column vectors. For $x \in \mathbb{R}^q$, x_i is the ith component and $|x|$ corresponds to its Euclidian norm, $x \cdot y$ denotes the scalar product of x and $y \in \mathbb{R}^q$. $\mathcal{M}_q$ is the set of $q \times q$ real matrices. We denote by $\mathbf{e}^\ell$ the ℓth vector of the standard basis of $\mathbb{R}^q$. The vector $(1, \dots, 1)^\top$ is denoted $\mathbf{1}$, I_d is the $d \times d$ identity matrix. We use the bold face notations $\mathbf{l} \in \mathbb{N}^d$ for multidimensional indices $\mathbf{l} := (l_1, \cdots, l_d)$ with (index) norms denoted by $|\mathbf{l}|_p := (\sum_{i=1}^{d} l_i^p)^{1/p}$ and $|\mathbf{l}|_\infty := \max_{1 \leqslant i \leqslant d} |l_i|$. For later use, for a positive integer k, we introduce the set $\mathbf{J}_{d,k}^{\infty}$ of multidimensional indices $\mathbf{l} \in \mathbb{N}^d$ satisfying $|\mathbf{l}|_\infty \leqslant k$. For a finite set A, we denote by $|A|$ its cardinality.

For a function $f : \mathbb{R}^d \to \mathbb{R}$, we denote by $\partial_{x_l} f$ the partial derivative function with respect to x_l, ∇f denotes the gradient function of f, valued in $\mathbb{R}^d$. We also use $\nabla^2 f = (\partial^2_{x_i,x_j} f)_{1 \leqslant i,j \leqslant d}$ to denote the Hessian matrix of f, valued in $\mathcal{M}_d$. For a sufficiently smooth real-valued function f defined in $\mathbb{R}^d$, we let $D^{\mathbf{l}} f = \partial^{l_1}_{x_1} \cdots \partial^{l_d}_{x_d} f$ denote the differentiation operator with respect to the multi-index $\mathbf{l} \in \mathbb{N}^d$. For a fixed positive integer k and a function f defined on an open domain $\mathcal{U} \subset \mathbb{R}^d$, we

define its Sobolev norm of mixed smoothness

$$\|f\|_{H^k_{mix}(\mathcal{U})} := \Big(\sum_{\mathbf{l}\in\mathbf{J}^\infty_{d,k}} \|D^{\mathbf{l}} f\|^2_{L_2(\mathcal{U})} \Big)^{\frac{1}{2}} \tag{2.1.5}$$

where the derivative $D^{\mathbf{l}} f$ in the above formula has to be understood in the weak sense and for a map $g : \mathcal{U} \mapsto \mathbb{R}$, $\|g\|^2_{L^2(\mathcal{U})} := \int_{\mathcal{U}} |g(x)|^2 dx$. The Sobolev space of mixed smoothness $H^k_{mix}(\mathcal{U})$ is then defined by

$$H^k_{mix}(\mathcal{U}) := \Big\{ f \in L_2(\mathcal{U}) : \|f\|_{H^k_{mix}(\mathcal{U})} < \infty \Big\}. \tag{2.1.6}$$

For a positive integer q, the set $\mathcal{H}^2_q$ is the set of progressively measurable processes V defined on the probability space $(\mathfrak{O}, \mathcal{A}, \mathbb{P})$ with values in $\mathbb{R}^q$ and satisfying $\mathbb{E}\Big[\int_0^T |V_t|^2 \, \mathrm{d}t\Big] < +\infty$. The set $\mathcal{S}^2_q$ is the set of adapted càdlàg processes U defined on the probability space $(\mathfrak{O}, \mathcal{A}, \mathbb{P})$ with values in $\mathbb{R}^q$ and satisfying $\mathbb{E}\Big[\sup_{t\in[0,T]} |U_t|^2\Big] < +\infty$. We also define $\mathcal{B}^2 := \mathcal{S}^2_1 \times \mathcal{H}^2_d$.

2.2 The *direct* and *Picard algorithms*

We describe here the numerical methods studied in this work. The first one, the *direct algorithm* is an adaptation of the *Deep BSDEs solver* introduced in [36] to the linear specification of the parametric space that we use here. The second one, the *Picard algorithm*, is new and is the main contribution of our work. We also give here the main general convergence results related to the *Picard algorithm*. The complexity analysis is postponed to the next section.

The methods we introduce below have for goal to compute an approximation of the value function u, satisfying the PDE (2.1.1), at the initial time on a given domain or at a specific point. This lead us to introduce the following setup for the initial value $\mathcal{X}_0$:

Assumption 2.2.1 *One of the two following cases holds:*

(i) The law of $\mathcal{X}_0$ has compact support and is absolutely continuous with respect to the Lebesgue measure.

(ii) The law of $\mathcal{X}_0$ is a Dirac mass at some point $x_0 \in \mathbb{R}^d$.

Most of our numerical applications are done in the setting of Assumption 2.2.1(ii), see next section. Then, obviously, the approximation of the value function is known only at the point x_0 at the initial time. However, one should note that it could also be interesting to work in the setting of Assumption 2.2.1(i) if one seeks to obtain an approximation of the *whole* value function (on the support of $\mathcal{X}_0$) at the initial time.

2.2.1 Assumptions on the coefficients and connection with the semi-linear PDE

In this subsection, we first give the assumptions on the BSDE coefficients that will be required for our approach and then recall the connection with semilinear PDEs. In particular, under these assumptions, the underlying PDE admits a unique classical solution. Under an additional regularity assumption on the coefficients, the unique solution to the PDE admits smooth derivatives of enough order which are controlled on the whole domain by known parameters. This additional regularity, together with a periodicity assumption, will be used to obtain our theoretical complexity result, see Section 2.3.1.2. For sake of simplicity, it is also assumed that the coefficients b, σ and f do not depend on time and that f does not depend on the space variable.

Assumption 2.2.2 *(i) The coefficients b, σ, f and g are bounded, Lipschitz-continuous with respect to all variables and $g \in \mathcal{C}^{2+\alpha}(\mathbb{R}^d)$, for some $\alpha \in (0,1]$. We will denote by L the Lipschitz-constant of the map f.*

(ii) The coefficient $a = \sigma\sigma^\top$ is uniformly elliptic, that is, there exists $\lambda_0 \geqslant 1$ such that for any $(x,\zeta) \in (\mathbb{R}^d)^2$ it holds

$$\lambda_0^{-1}|\zeta|^2 \leqslant a(x)\zeta\cdot\zeta \leqslant \lambda_0|\zeta|^2. \tag{2.2.1}$$

(iii) For any $(i,j) \in \{1,\cdots,d\}^2$, the coefficients b_i, $\sigma_{i,j}$, g belong to $\mathcal{C}^{2d+1}(\mathbb{R}^d,\mathbb{R})$ and f belongs $\mathcal{C}^{2d+1}(\mathbb{R}\times\mathbb{R}^d,\mathbb{R})$. Moreover, their derivatives of any order up and equal to $2d+1$ are bounded and Lipschitz continuous.

(iv) The coefficients b, σ, f and g are periodic functions.

From now on, we will say that Assumption 2.2.2 holds if and only if Assumption 2.2.2 (i), (ii), (iii) and (iv) are satisfied.

Under Assumption 2.2.2 (i) and (ii), it is known (see e.g. [81]) that for any square integrable initial condition $\mathcal{X}_0$ there exists a unique couple $(\mathcal{Y},\mathcal{Z}) \in \mathcal{B}^2$ satisfying equation (2.1.4) $\mathbb{P}$-a.s. Moreover, from [72] Chapter VI and [41] Chapter 7, the PDE (2.1.1) admits a unique solution $u \in \mathcal{C}^{1,2}([0,T]\times\mathbb{R}^d,\mathbb{R})$ satisfying: there exists a positive constant C, depending on T and the parameters appearing in Assumption 2.2.2 (i) and (ii), such that for all $(t,x) \in [0,T]\times\mathbb{R}^d$

$$|u(t,x)| + |\partial_t u(t,x)| + |\nabla_x u(t,x)| + |\nabla_x^2 u(t,x)| \leqslant C.$$

From [84, 82, 83], the semilinear PDE (2.1.1) and the BSDE (2.1.2)-(2.1.4) are connected, namely, for all $(t,x) \in [0,T)\times\mathbb{R}^d$, it holds

$$\mathcal{Y}_t = u(t,\mathcal{X}_t), \quad \mathcal{Z}_t = \sigma^\top(\mathcal{X}_t)\nabla_x u(t,\mathcal{X}_t).$$

Finally, under Assumption 2.2.2, still from [72] Chapter IV and [41] Chapter III, the unique solution u to the PDE (2.1.1) is smooth, namely, setting $v_i = (\sigma^\top\nabla_x u)_i$, $1 \leqslant i \leqslant d$, for any $\mathbf{l} \in \mathbf{J}^\infty_{d,2}$, $D^{\mathbf{l}}v_i(t,.)$ exists and is bounded. In particular, there exists a positive constant, depending on T and the known parameters appearing in Assumption 2.2.2 (i), (ii) and (iii) such that for all $\mathbf{l} \in \mathbf{J}^\infty_{d,2}$ and all $(t,x) \in [0,T]\times\mathbb{R}^d$,

$$\max_{1\leqslant i\leqslant d} |D^{\mathbf{l}}v_i(t,x)| \leqslant C. \tag{2.2.2}$$

2.2.2 *Direct algorithm*

We first consider the approximation of the forward component (2.1.2). Given an equidistant grid $\pi := \{t_0 = 0 < \cdots < t_n < \cdots < t_N = T\}$ of the time interval $[0, T]$, $t_n = nh$, $n = 0, \cdots N$, with time-step $h := T/N$, we denote by $W := (\mathcal{W}_{t_n})_{0 \leqslant n \leqslant N}$ the discrete-time version of the Brownian motion $\mathcal{W}$ and define $\Delta W_n = W_{t_{n+1}} - W_{t_n}$, $0 \leqslant n \leqslant N-1$.

We then introduce a standard Euler-Maruyama approximation scheme of $\mathcal{X}$ on π defined by $X_0 = \mathcal{X}_0$ and for $0 \leqslant n \leqslant N-1$,

$$X_{t_{n+1}} = X_{t_n} + b(X_{t_n})h + \sigma(X_{t_n})\Delta W_n \,. \tag{2.2.3}$$

Before discussing the approximation of the backward component, we here state an important lemma concerning the existence of two-sided Gaussian estimates for the transition density of the above Euler-Maruyama approximation scheme. These estimates will prove very useful in the sequel, when studying the theoretical complexity of the *Picard algorithm*. We denote by $p^\pi(t_i, t_j, x, \cdot)$ the transition density function of the Euler-Maruyama scheme starting from the point x at time t_i and taken at time t_j, with $0 \leqslant t_i < t_j \leqslant T$. We refer e.g. to [75] for a proof of the following result.

Lemma 2.2.1 *Assume that the coefficients b and σ satisfies Assumption 2.2.2 (i) and (ii). There exist constants $\mathfrak{c} := \mathfrak{c}(\lambda_0, b, \sigma, d) \in (0, 1]$ and $\mathfrak{C} := \mathfrak{C}(T, \lambda, b, \sigma, d) \geqslant 1$ such that for any $(x, x') \in (\mathbb{R}^d)^2$ and for any $0 \leqslant i < j \leqslant N$*

$$\mathfrak{C}^{-1} p(\mathfrak{c}(t_j - t_i), x - x') \leqslant p^\pi(t_i, t_j, x, x') \leqslant \mathfrak{C} p(\mathfrak{c}^{-1}(t_j - t_i), x' - x) \tag{2.2.4}$$

where for any $(t, x) \in (0, \infty) \times \mathbb{R}^d$, $p(t, x) := (1/(2\pi t))^{d/2} \exp(-|x|^2/(2t))$.

We now turn to the approximation of the backward component (2.1.4). We first introduce a linear parametrization of the process $\mathcal{Z}$. For each discrete date $t_n \in \pi \backslash \{T\}$, we consider a parametric functional approximation space $\mathscr{Y}_n^z$ generated by a set of basis functions $(\psi_n^k)_{1 \leqslant k \leqslant K_n^z}$, for $0 \leqslant n \leqslant N-1$ and some positive integer K_n^z. The measurable functions $\psi_n^k : \mathbb{R}^d \mapsto \mathbb{R}$ have at most polynomial growth. Note that, for $n \geqslant 1$, the specification of the basis function could depend on the time t_n, but in order to simplify the discussion, we let the number of basis functions be the same and set to K. Namely, $K_n^z = K$, for all $n \geqslant 1$. For $n = 0$, the specification will depend on the nature of $\mathcal{X}_0$: if Assumption 2.2.1(i) holds, then we will set $K_0^z = K$; if Assumption 2.2.1(ii) holds, then we simply set $K_0^z = 1$ and ψ_0^1 is a function satisfying $\psi_0^1(x_0) = 1$. For latter use, we set:

$$\bar{K}^z := \sum_{n=0}^{N-1} K_n^z = K_0^z + (N-1)K \,. \tag{2.2.5}$$

Remark that there is no need to introduce an approximation space at T since the function g is explicitly known.
For $0 \leqslant n \leqslant N-1$, each component of $(\sigma^\top \nabla_x u)(t_n, \cdot)$ should be approximated in an optimal way by a function in $\mathscr{Y}_n^z$. The process $\mathcal{Z}$ appearing in the dynamics of

the controlled process $\mathcal{Y}$, that has to be optimized, is parametrized using the spaces $(\mathscr{V}_n^z)_{0\leqslant n\leqslant N-1}$. Namely, the $\mathbb{R}^d$-valued random variable $\mathcal{Z}_{t_n}$ will be approximated by

$$\sum_{k=1}^{K_n^z} \psi_n^k(X_{t_n})\mathfrak{z}^{n,k}\,, \tag{2.2.6}$$

where $\mathfrak{z}^{n,k} \in \mathbb{R}^d$ for any $1 \leqslant k \leqslant K_n^z$ and $0 \leqslant n \leqslant N-1$. Importantly, we denote, for later use, $\mathfrak{z}^\top := ((\mathfrak{z}^{n,k})^\top)_{0\leqslant n\leqslant N-1, 1\leqslant k\leqslant K_n^z}$ so that $\mathfrak{z} \in \mathbb{R}^{d\bar{K}^z}$.

Definition 2.2.1 (Class of discrete control process) *We let $\mathcal{H}^{\pi,\psi}$ be the set of discrete control process Z defined by: for $\mathfrak{z} \in \mathbb{R}^{d\bar{K}^z}$,*

$$Z_{t_n} := \sum_{k=1}^{K_n^z} \psi_n^k(X_{t_n})\mathfrak{z}^{n,k}, \quad \textit{for } 0 \leqslant n \leqslant N-1, \tag{2.2.7}$$

and where we set $Z_t = Z_{t_n}$, $t_n \leqslant t < t_{n+1}, 0 \leqslant n \leqslant N-1$ with the convention $Z_T = 0$.

Remark 2.2.1 *We insist on the fact that for a given $Z \in \mathcal{H}^{\pi,\psi}$, the $\mathbb{R}^d$-valued random variable Z_{t_n} depends only on $\mathfrak{z}^n$, for any $0 \leqslant n \leqslant N-1$. The approximation space we consider is a finite dimensional vector space. This notably differs from the recent works [7, 55, 65] where a non-linear approximation using neural network is used.*

The dynamics of $\mathcal{Y}$ given by (2.1.4), in turn, has to be approximated. As previously mentioned in the introduction, we first rewrite it in forward form as follows

$$\mathcal{Y}_t = \mathcal{Y}_0 - \int_0^t f(\mathcal{Y}_s, \mathcal{Z}_s)\,ds + \int_0^t \mathcal{Z}_s \cdot d\mathcal{W}_s,\ t \in [0,T], \ \text{ with } \mathcal{Y}_0 = u(0, \mathcal{X}_0).$$

The main goal of the algorithm is to obtain a good estimate of $u(0,.)$ on the support of $\mathcal{X}_0$. In order to do so, we define the starting point of Y, standing for the approximation of $\mathcal{Y}$, by using a linear functional approximation space denoted $\mathscr{V}^y$, namely

$$Y_0 := \sum_{k=1}^{K^y} \psi_y^k(X_0)\mathfrak{y}^k \quad \text{with} \quad \mathfrak{y} \in \mathbb{R}^{K^y}. \tag{2.2.8}$$

The specification of $\mathscr{V}^y$ will depend also on the nature of $\mathcal{X}_0$. Namely, if Assumption 2.2.1(i) holds, then we set $K^y = K$, while if Assumption 2.2.1(ii) holds, then we simply set $K^y = 1$ and ψ_y^1 is a function satisfying $\psi_y^1(x_0) = 1$.

Then, employing a standard Euler scheme on π together with the above approximation $Z \in \mathcal{H}^{\pi,\psi}$ of the control process $\mathcal{Z}$, we are naturally led to consider the following approximation scheme for $\mathcal{Y}$.

Definition 2.2.2 *i) Given* $\mathfrak{u} = (\mathfrak{y}, \mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$*, we denote by* $Z^{\mathfrak{u}} \in \mathcal{H}^{\pi,\psi}$ *the discrete control process as given in* (2.2.7)*. Then, the discrete controlled process* $Y^{\mathfrak{u}}$ *is defined as follows.*

(a) Initialization: Set

$$Y_0^{\mathfrak{u}} = \sum_{k=1}^{K} \psi_y^k(X_0)\mathfrak{y}^k \,. \tag{2.2.9}$$

(b) Discrete version: for any $0 \leqslant n \leqslant N-1$*:*

$$Y_{t_{n+1}}^{\mathfrak{u}} = Y_{t_n}^{\mathfrak{u}} - hf(Y_{t_n}^{\mathfrak{u}}, Z_{t_n}^{\mathfrak{u}}) + Z_{t_n}^{\mathfrak{u}} \cdot \Delta W_n \tag{2.2.10}$$

where we recall that $\Delta W_n = W_{t_{n+1}} - W_{t_n}$*.*

(c) Continuous version: for any $0 \leqslant n \leqslant N-1$ *and any* $t_n \leqslant t < t_{n+1}$*,*

$$Y_t^{\mathfrak{u}} = Y_{t_n}^{\mathfrak{u}} - (t-t_n)f(Y_{t_n}^{\mathfrak{u}}, Z_{t_n}^{\mathfrak{u}}) + Z_{t_n}^{\mathfrak{u}} \cdot (\mathcal{W}_t - \mathcal{W}_{t_n}) \tag{2.2.11}$$

ii) Based on the previous step, we define $\mathcal{B}^{\pi,\psi} \subset \mathcal{B}^2$ *as the set of processes* $(Y^{\mathfrak{u}}, Z^{\mathfrak{u}})$*, with* $Z^{\mathfrak{u}} \in \mathcal{H}^{\pi,\psi}$*,* $Y^{\mathfrak{u}}$ *defined as above for some* $\mathfrak{u} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$*.*

Remark 2.2.2 *Let us note that the discrete process* $(X_t, Y_t^{\mathfrak{u}}, Z_t^{\mathfrak{u}})_{t\in\pi}$ *depends on* $\mathcal{X}_0$ *and* $(W_t)_{t\in\pi}$ *but we omit these dependences in the notation.*

The main idea of *approximation by learning* methods is to force the discrete controlled process $Y_T^{\mathfrak{u}}$ at maturity T to match the approximated terminal condition $g(X_T)$, by minimizing a loss function. Here, we work with the quadratic loss function, so that one faces the optimization problem

$$\inf_{\mathfrak{u}=(\mathfrak{y},\mathfrak{z})\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}} \mathfrak{g}(\mathfrak{u}) := \mathbb{E}[\mathrm{G}(\mathcal{X}_0, W, \mathfrak{u})] \quad \text{with} \quad \mathrm{G}(\mathcal{X}_0, W, \mathfrak{u}) = |g(X_T) - Y_T^{\mathfrak{u}}|^2 \,. \tag{2.2.12}$$

However, one has to come up with a numerical procedure to compute the solution in practice.

In order to numerically compute a solution to the optimization problem (2.2.12) (if any exists), one generally employs a stochastic approximation scheme such as a SGD algorithm. For an overview of the theory of stochastic approximation, the reader may refer to [35], [70] and [10] and to [7, 36, 55, 65, 53] for applications to deep learning approximation of PDEs.

We now describe the SGD algorithm that we implement in order to compute a solution $(\mathfrak{y}, \mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$ to the optimization problem (2.2.12).

For a prescribed positive integer M representing the number of steps in the stochastic algorithm and two deterministic non increasing sequences of positive real number $(\gamma_m^y)_{m\geqslant 1}$ and $(\gamma_m^z)_{m\geqslant 1}$ representing the learning rates, we design the following *direct algorithm*.

Definition 2.2.3 *(Implemented direct algorithm)*

1. *Simulate M independent discrete paths of the Brownian motion $\mathfrak{W} = (W^m)_{1\leqslant m\leqslant M}$ and M independent samples of the initial condition $(\mathcal{X}_0^m)_{1\leqslant m\leqslant M}$.*

2. *Initialization: select a random vector $\mathfrak{u}_0 = (\mathfrak{y}_0, \mathfrak{z}_0)$ with values in $\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, independent of $\mathfrak{W}$ and $(\mathcal{X}_0^m)_{1\leqslant m\leqslant M}$, and such that $\mathbb{E}[|\mathfrak{u}_0|^2] < \infty$.*

3. *Iteration: For $0 \leqslant m \leqslant M-1$, compute*

$$\mathfrak{y}_{m+1} = \mathfrak{y}_m - \gamma_{m+1}^y \nabla_{\mathfrak{y}} \mathrm{G}\big(\mathcal{X}_0^{m+1}, W^{m+1}, \mathfrak{u}_m\big)\,, \tag{2.2.13}$$

$$\mathfrak{z}_{m+1}^n = \mathfrak{z}_m^n - \gamma_{m+1}^z \nabla_{\mathfrak{z}^n} \mathrm{G}\big(\mathcal{X}_0^{m+1}, W^{m+1}, \mathfrak{u}_m\big)\,, \tag{2.2.14}$$

   for $0 \leqslant n \leqslant N-1$.

The output of the algorithm is then $\mathfrak{u}_M = (\mathfrak{y}_M, \mathfrak{z}_M)$.

Remark 2.2.3 *In order to analyse the asymptotic properties of stochastic approximation schemes, one usually chooses the learning sequences $(\gamma_m)_{m\geqslant 1} = (\gamma_m^y)_{m\geqslant 1}$ or $(\gamma_m)_{m\geqslant 1} = (\gamma_m^z)_{m\geqslant 1}$ such that*

$$\sum_{m\geqslant 1} \gamma_m = \infty \quad \textit{and} \quad \sum_{m\geqslant 1} \gamma_m^2 < \infty, \tag{2.2.15}$$

see e.g. [35, 70, 10].

The following lemma, whose proof is postponed to Section 2.4.2, provides the analytic expression of the local gradient functions $\nabla_\lambda \mathrm{G}(\mathcal{X}_0, W, \mathfrak{u})$, $\lambda \in \{\mathfrak{y}, \mathfrak{z}^n, 0 \leqslant n \leqslant N-1\}$ appearing in the above SGD algorithm. It shows that $(\mathfrak{y}_{m+1}, \mathfrak{z}_{m+1})$ can be easily computed once $(Y^{\mathfrak{u}_m}, Z^{\mathfrak{u}_m})$ have been simulated for any $0 \leqslant m \leqslant M-1$.

Lemma 2.2.2 *For $\lambda \in \big\{\mathfrak{y}^k, 1 \leqslant k \leqslant K^y\big\} \bigcup \big\{\mathfrak{z}^{n,k}, 1 \leqslant k \leqslant K_n^z, 0 \leqslant n \leqslant N-1\big\}$ and $\mathfrak{u} = (\mathfrak{y}, \mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, it holds*

$$\nabla_\lambda \mathrm{G}(\mathcal{X}_0, W, \mathfrak{u}) = -2(g(X_T) - Y_T^{\mathfrak{u}}) \nabla_\lambda Y_T^{\mathfrak{u}} \tag{2.2.16}$$

with

$$\nabla_{\mathfrak{y}^k} Y_T^{\mathfrak{u}} = \psi_y^k(X_0) \prod_{l=0}^{N-1} \big(1 - h\nabla_y f(Y_{t_l}^{\mathfrak{u}}, Z_{t_l}^{\mathfrak{u}})\big)\,,\ 1 \leqslant k \leqslant K^y,$$

and for any $0 \leqslant n \leqslant N-1$ and any $1 \leqslant k \leqslant K_n^z$,

$$\nabla_{\mathfrak{z}^{n,k}} Y_T^{\mathfrak{u}} = \psi_n^k(X_{t_n}) \big(\Delta W_n^\top - h\nabla_z f(Y_{t_n}^{\mathfrak{u}}, Z_{t_n}^{\mathfrak{u}})\big) \prod_{l=n+1}^{N-1} \big(1 - h\nabla_y f(Y_{t_l}^{\mathfrak{u}}, Z_{t_l}^{\mathfrak{u}})\big)$$

with the convention $\prod_{\varnothing} = 1$.

Under Assumption 2.2.1, Assumption 2.2.2 (i) and Assumption 2.2.3, the well-posedness of Algorithm 2.2.3, that is, the fact that it holds

$$\operatorname*{arg\,min}_{\mathfrak{u}\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}} \mathfrak{g}(\mathfrak{u}) \neq \varnothing\,,$$

is proved in Lemma 2.4.1.
Additionally, for any $\mathfrak{u}^\star \in \arg\min_{\mathfrak{u}\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}} \mathfrak{g}(\mathfrak{u})$, we show in Proposition 2.4.2 that

$$\mathbb{E}\left[|u(0,\mathcal{X}_0) - Y_0^{\mathfrak{u}^\star}|^2 + h\sum_{n=0}^{N-1}|\mathcal{Z}_{t_n} - Z_{t_n}^{\mathfrak{u}^\star}|^2\right] \leqslant C\left(\mathcal{E}_\pi + \mathcal{E}_\psi\right) ,$$

for some positive constant C. The quantities $\mathcal{E}_\pi$ and $\mathcal{E}_\psi$ represents the discrete-time error and the error due to the approximation in the functional spaces $(\mathscr{Y}^y, \mathscr{Y}_n^z)$, respectively. They are defined by

$$\mathcal{E}_\pi := \mathbb{E}\left[\sum_{n=0}^{N-1}\int_{t_n}^{t_{n+1}}\left(|\mathcal{Y}_s - \mathcal{Y}_{t_n}|^2 + |\mathcal{Z}_s - \mathcal{Z}_{t_n}|^2 + |\mathcal{X}_{t_n} - X_{t_n}|^2\right)ds\right] \tag{2.2.17}$$

and

$$\mathcal{E}_\psi := \inf_{\mathfrak{u}\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}} \mathbb{E}\left[|u(0,\mathcal{X}_0) - Y_0^{\mathfrak{u}}|^2 + \sum_{n=0}^{N-1} h|(\sigma^\top\nabla_x u)(t_n, X_{t_n}) - Z_{t_n}^{\mathfrak{u}}|^2\right] . \tag{2.2.18}$$

Let us mention, for later use, that, in the setting of Assumptions 2.2.1 and 2.2.2(i),

$$\mathcal{E}_\pi \leqslant Ch , \tag{2.2.19}$$

for some positive constant C, see e.g. Ma and Zhang [73] and Pagès [79].

We shall not seek to obtain theoretical convergence results for the *direct algorithm* itself. However, we illustrate its performance numerically in Section 2.3 when using sparse grids approximations [20].

Remark 2.2.4 *The numerical complexity $\mathcal{C}$ will be measured by the number of coefficients update realized to obtain the approximation. From the previous description, we obtain straightforwardly that the complexity at worst satisfies*

$$\mathcal{C} = O_d(NKM) . \tag{2.2.20}$$

2.2.3 A *Picard algorithm*

An issue with the above algorithm comes from the fact that the optimization problem (2.2.12) is generally not convex. Even though $\mathfrak{u} \mapsto (Y_0^{\mathfrak{u}}, Z^{\mathfrak{u}})$ is linear for our choice of parametrisation, in general the mapping $\mathfrak{u} \to Y^{\mathfrak{u}}$ is non-linear since f itself is non-linear. As a consequence, in practical implementation, we have no guarantee that the algorithm converges to local or global minima. On top of practical problems, this renders the theoretical analysis of the implemented *direct algorithm* difficult, in particular if one wants to obtain rates of convergence to assess precisely the numerical complexity of the method.

In this section, we introduce a *Picard algorithm* which transforms this non-convex optimisation problem into a sequence of linear-quadratic optimization problems. This is done by using the special structure of the original problem. Indeed, it is well known that the solution of the BSDE (2.1.4) itself is obtained as the limit of a sequence of Picard iterations, see e.g. [37] and [9] from a numerical perspective.

2.2.3.1 Theoretical Picard algorithm

Our *Picard algorithm* is based on the iteration of the following operator:

$$\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z} \ni \tilde{\mathfrak{u}} \mapsto \Phi(\tilde{\mathfrak{u}}) := \check{\mathfrak{u}} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z} \tag{2.2.21}$$

where,

$$\check{\mathfrak{u}} = \underset{\mathfrak{u}\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}}{\arg\min}\ \mathbb{E}\Big[|g(X_T) - U_T^{\tilde{\mathfrak{u}},\mathfrak{u}}|^2\Big]\,. \tag{2.2.22}$$

In the above expectation, the process X is the Euler-Maruyama approximation scheme on the time grid π with dynamics (2.2.3) and $U^{\tilde{\mathfrak{u}},\mathfrak{u}}$ (simply denoted as U below) is given by the following decoupling approximation scheme:

1. For $\tilde{\mathfrak{u}} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, we first consider $(Y^{\tilde{\mathfrak{u}}}, Z^{\tilde{\mathfrak{u}}}) \in \mathcal{B}^{\pi,\psi}$ as introduced in Definition 2.2.2.

2. Then, for any $\mathfrak{u} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, consider the discrete control process $Z^{\mathfrak{u}} \in \mathcal{H}^{\pi,\psi}$ as introduced in (2.2.7) of Definition 2.2.1 and define the control process $U^{\tilde{\mathfrak{u}},\mathfrak{u}}$ by

$$U_0^{\tilde{\mathfrak{u}},\mathfrak{u}} = Y_0^{\mathfrak{u}} \tag{2.2.23}$$

recall (2.2.9) and for any $0 \leqslant n \leqslant N-1$,

$$U_{t_{n+1}}^{\tilde{\mathfrak{u}},\mathfrak{u}} = U_{t_n}^{\tilde{\mathfrak{u}},\mathfrak{u}} - hf(Y_{t_n}^{\tilde{\mathfrak{u}}}, Z_{t_n}^{\tilde{\mathfrak{u}}}) + Z_{t_n}^{\mathfrak{u}} \cdot (W_{t_{n+1}} - W_{t_n})\,, \tag{2.2.24}$$

and its continuous version, for any $t_n \leqslant t \leqslant t_{n+1}$,

$$U_t^{\tilde{\mathfrak{u}},\mathfrak{u}} = U_{t_n}^{\tilde{\mathfrak{u}},\mathfrak{u}} - (t-t_n)f(Y_{t_n}^{\tilde{\mathfrak{u}}}, Z_{t_n}^{\tilde{\mathfrak{u}}}) + Z_{t_n}^{\mathfrak{u}} \cdot (\mathcal{W}_t - W_{t_n})\,.$$

Note that under Assumption 2.2.2(i) it holds $\mathbb{E}\Big[\sup_{t\in[0,T]} |U_t^{\tilde{\mathfrak{u}},\mathfrak{u}}|^2\Big] < +\infty$.

Definition 2.2.4 (Theoretical Picard algorithm) *For a prescribed positive integer P:*

1. *Initialization: set $\mathfrak{u}^0 \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$.*

2. *Iteration: for $1 \leqslant p \leqslant P$, compute: $\mathfrak{u}^p = \Phi(\mathfrak{u}^{p-1})$.*

The output of the algorithm is then $\mathfrak{u}^P$.

2.2.3.2 Well-posedness of the theoretical algorithm

The main novelty compared to the optimization problem (2.2.12) comes from the fact that the map $\mathfrak{u} \mapsto U^{\tilde{\mathfrak{u}},\mathfrak{u}}$ is now linear. This linearity is achieved by freezing the driver f in the dynamics of the control process along the process $(Y^{\tilde{\mathfrak{u}}}, Z^{\tilde{\mathfrak{u}}}) \in \mathcal{B}^{\pi,\psi}$. The parameter $\tilde{\mathfrak{u}} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$ is then updated through the Picard iteration procedure. This is of course the main purpose of this *Picard algorithm*, compared to the *direct algorithm* of Section 2.2.2. At this stage, the above algorithm is theoretical and

its solution (if any exists) still needs to be numerically approximated. This will be discussed in full details in the next section.

We here discuss the well-posedness of the optimization problem (2.2.22). We first introduce some notations that will be useful in the sequel to study the *Picard algorithm* as iterated least-square optimization problems.
First, to clarify the linear structure, we introduce the following notations

i) For $1 \leqslant k \leqslant K^y$, $\theta^k := \psi_y^k(X_0)$.

ii) For $0 \leqslant n \leqslant N-1$, $1 \leqslant k \leqslant K$, the $\mathbb{R}^d$-valued random vectors $\omega^{n,k}$ is defined by

$$\omega^{n,k} = \Psi^{n,k}\Delta W_n \text{ with } \Psi^{n,k} = \psi_n^k(X_{t_n}) \,, \tag{2.2.25}$$

and we set $\omega^\top := ((\omega^{n,k})^\top)_{0\leqslant n\leqslant N-1, 1\leqslant k\leqslant K_n^z}$ (so that ω is an $\mathbb{R}^{d\bar{K}^z}$-valued random vector).

iii) the random vector $\Omega = (\theta^\top, \omega^\top)^\top$ which takes values in $\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$.

Note that both ω and Ω depends on W and $\mathcal{X}_0$, but we will omit this in the notation. Then, we rewrite

$$g(X_T) - U_T^{\tilde{\mathfrak{u}},\mathfrak{u}} = \mathfrak{G}^{\tilde{\mathfrak{u}}} - \mathfrak{u}\cdot\Omega \tag{2.2.26}$$

where

$$\mathfrak{G}^{\tilde{\mathfrak{u}}} = \mathfrak{G}^{\tilde{\mathfrak{u}}}(\mathcal{X}_0, W) := g(X_T) + \sum_{n=0}^{N-1} hf(Y_{t_n}^{\tilde{\mathfrak{u}}}, Z_{t_n}^{\tilde{\mathfrak{u}}}) \,. \tag{2.2.27}$$

Thus, the optimization problem (2.2.22) is given by

$$\check{\mathfrak{u}} = \text{argmin}_{\mathfrak{u}\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}} \mathfrak{H}(\tilde{\mathfrak{u}},\mathfrak{u}) \quad \text{with} \quad \mathfrak{H}(\tilde{\mathfrak{u}},\mathfrak{u}) := \mathbb{E}\Big[|\mathfrak{G}^{\tilde{\mathfrak{u}}} - \mathfrak{u}\cdot\Omega|^2\Big] \tag{2.2.28}$$

and simply reads as a Linear-Quadratic optimization problem. Classically, we introduce semi-norms on the parameter spaces.

Definition 2.2.5 *For $\mathfrak{u} = (\mathfrak{y},\mathfrak{z}) \in \mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}$, we define*

$$\|\mathfrak{y}\|_y^2 := \mathbb{E}\big[|\mathfrak{y}\cdot\theta|^2\big] \,,\ \|\mathfrak{z}\|_z^2 := \mathbb{E}\big[|\mathfrak{z}\cdot\omega|^2\big] \text{ and } |||\mathfrak{u}|||^2 := \mathbb{E}\big[|\mathfrak{u}\cdot\Omega|^2\big] \,.$$

Let us insist on the fact that these quantities depend on the choices of π,ψ though this is not reflected in the notation.

Remark 2.2.5 *i) Observe that from the very definition of the random vector Ω, for any $\mathfrak{u} = (\mathfrak{y},\mathfrak{z}) \in \mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}$, it holds*

$$|||\mathfrak{u}|||^2 = \|\mathfrak{y}\|_y^2 + \|\mathfrak{z}\|_z^2. \tag{2.2.29}$$

ii) With the notations of Section 2.2.2, the following relations hold

$$|||\mathfrak{u}|||^2 = \mathbb{E}\left[\left|Y_0^{\mathfrak{u}} + \int_0^T Z_t^{\mathfrak{u}}\,\mathrm{d}\mathcal{W}_t\right|^2\right],\ \|\mathfrak{y}\|_y^2 = \mathbb{E}\big[|Y_0^{\mathfrak{u}}|^2\big] \text{ and } \|\mathfrak{z}\|_z^2 = \sum_{n=0}^{N-1} h\mathbb{E}\big[|Z_{t_n}^{\mathfrak{u}}|^2\big]\ , \tag{2.2.30}$$

for any $\mathfrak{u} = (\mathfrak{y}, \mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$.

iii) For later use, see Section 2.2.3.3, we also note that $\|\cdot\|_z$ *develops as*

$$\|\mathfrak{z}\|_z^2 = \sum_{n=0}^{N-1} h \sum_{l=1}^{d} (\mathfrak{z}_l^{n,\cdot})^\top \mathbb{E}\big[\Psi^{n,\cdot}(\Psi^{n,\cdot})^\top\big]\,\mathfrak{z}_l^{n,\cdot} \tag{2.2.31}$$

by using the independence of the increments $(\Delta W_n)_l$, *for* $0 \leqslant n \leqslant N-1$ *and* $l \in \{1, \dots, d\}$ *and where we used the notations* $\mathfrak{z}_l^{n,\cdot} = (\mathfrak{z}_l^{n,1}, \dots, \mathfrak{z}_l^{n,K_n^z})$ *and* $\Psi^{n,\cdot} = (\Psi^{n,1}, \dots, \Psi^{n,K_n^z})$.

A key assumption to ensure the well-posedness of our approach is the following.

Assumption 2.2.3 *There exist two positive constants* $\kappa_K \geqslant 1 \geqslant \alpha_K$ *such that for any* $(\mathfrak{y}, \mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$

$$\alpha_K|\mathfrak{y}|^2 \leqslant \|\mathfrak{y}\|_y^2 \leqslant \kappa_K|\mathfrak{y}|^2 \quad \text{and} \quad h\alpha_K|\mathfrak{z}|^2 \leqslant \|\mathfrak{z}\|_z^2 \leqslant h\kappa_K|\mathfrak{z}|^2\ .$$

Lemma 2.2.3 *For all* $(\tilde{\mathfrak{u}}, \mathfrak{u}) \in \left(\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}\right)^2$, *it holds*

$$\mathfrak{u}^\top \nabla_{\mathfrak{u}}^2 \mathfrak{H}(\tilde{\mathfrak{u}}, \mathfrak{u})\mathfrak{u} = 2|||\mathfrak{u}|||^2\,, \tag{2.2.32}$$

where $\nabla_{\mathfrak{u}}^2\mathfrak{H}$ *denotes the Hessian of the function* $\mathfrak{u} \mapsto \mathfrak{H}(\tilde{\mathfrak{u}}, \mathfrak{u})$.
Moreover, under Assumption 2.2.3, the optimization problem (2.2.22) admits a unique solution and for any $\tilde{\mathfrak{u}} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$ *and* $(\mathfrak{y}, \mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, *it holds*

$$(\mathfrak{y} - \check{\mathfrak{y}}) \cdot \nabla_{\mathfrak{y}}\mathfrak{H}(\tilde{\mathfrak{u}}, \mathfrak{u}) = 2\|\mathfrak{y} - \check{\mathfrak{y}}\|_y^2 \geqslant 2\alpha_K|\mathfrak{y} - \check{\mathfrak{y}}|^2 \tag{2.2.33}$$

$$(\mathfrak{z} - \check{\mathfrak{z}}) \cdot \nabla_{\mathfrak{z}}\mathfrak{H}(\tilde{\mathfrak{u}}, \mathfrak{u}) = 2\|\mathfrak{z} - \check{\mathfrak{z}}\|_z^2 \geqslant 2h\alpha_K|\mathfrak{z} - \check{\mathfrak{z}}|^2 \tag{2.2.34}$$

where $\check{\mathfrak{u}} = (\check{\mathfrak{y}}, \check{\mathfrak{z}}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$ *is the unique solution to (2.2.22).*

Proof. From (2.2.28), we straightforwardly compute

$$\nabla_{\mathfrak{u}}\mathfrak{H}(\tilde{\mathfrak{u}}, \mathfrak{u}) = -2\mathbb{E}\Big[(\mathfrak{G}^{\tilde{\mathfrak{u}}} - \mathfrak{u} \cdot \Omega)\Omega\Big] \text{ and } \nabla_{\mathfrak{u}}^2\mathfrak{H}(\tilde{\mathfrak{u}}, \mathfrak{u}) = 2\mathbb{E}\big[\Omega\Omega^\top\big] \tag{2.2.35}$$

which, recalling Definition 2.2.5, directly yields (2.2.32). In particular, we have

$$\nabla_{\mathfrak{y}}\mathfrak{H}(\tilde{\mathfrak{u}}, \mathfrak{u}) = -2\mathbb{E}\Big[(\mathfrak{G}^{\tilde{\mathfrak{u}}} - \mathfrak{u} \cdot \Omega)\theta\Big] = -2\mathbb{E}\Big[(\mathfrak{G}^{\tilde{\mathfrak{u}}} - \mathfrak{y} \cdot \theta)\theta\Big] \tag{2.2.36}$$

and, for any $0 \leqslant n \leqslant N-1$ and any $1 \leqslant l \leqslant d$,

$$\nabla_{\mathfrak{z}_l^{n,\cdot}}\mathfrak{H}(\tilde{\mathfrak{u}}, \mathfrak{u}) = -2\mathbb{E}\Big[(\mathfrak{G}^{\tilde{\mathfrak{u}}} - \Omega \cdot \mathfrak{u})\omega_l^{n,\cdot}\Big] = -2\mathbb{E}\Big[(\mathfrak{G}^{\tilde{\mathfrak{u}}} - \omega_l^{n,\cdot} \cdot \mathfrak{z}_l^{n,\cdot})\omega_l^{n,\cdot}\Big]\ . \tag{2.2.37}$$

Under Assumption 2.2.3, we deduce from (2.2.32) that the problem is strictly convex and has a unique (global) minimum $\check{\mathfrak{u}} = (\check{\mathfrak{y}}, \check{\mathfrak{z}})$. The inequalities (2.2.33) and (2.2.34) then follow from (2.2.36) and (2.2.37) combined with the fact that $\nabla_{\mathfrak{z}}\mathfrak{H}(\tilde{\mathfrak{u}}, \check{\mathfrak{u}}) = 0$.

2.2.3.3 Algorithm implementation

From a practical point of view, the sequence of theoretical Linear-Quadratic optimization problem described in the previous section has to be approximated. Due to the possibly high dimension of the matrix $\mathbb{E}[\Omega\Omega^\top]$, we will rely on a SGD algorithm[1] to compute the unique solution to (2.2.22). Indeed, for a fixed vector $\tilde{\mathfrak{u}}$ in $\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, the key point is to observe that the unique minimizer $\check{\mathfrak{u}}$ is the unique solution to the equation

$$\nabla_{\mathfrak{u}}\mathfrak{H}(\tilde{\mathfrak{u}}, \mathfrak{u}) = 0. \tag{2.2.38}$$

We importantly remark, using (2.2.36) and (2.2.37) that the above relation (2.2.38) holds true if and only if

$$\mathbb{E}[H^y(\mathcal{X}_0, W, \tilde{\mathfrak{u}}, \mathfrak{y})] = 0, \quad \text{and} \quad \mathbb{E}\Big[H^{n,l}(\mathcal{X}_0, W, \tilde{\mathfrak{u}}, \mathfrak{z}_l^{n,\cdot})\Big] = 0, \tag{2.2.39}$$

where H^y is a map from $\mathbb{R}^d \times (\mathbb{R}^d)^N \times (\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}) \times \mathbb{R}^{K^y}$ to $\mathbb{R}^{K^y}$ and defined by

$$H^y(\mathcal{X}_0, W, \tilde{\mathfrak{u}}, \mathfrak{y}) := -\frac{2}{\beta_K}(\mathfrak{G}^{\tilde{\mathfrak{u}}} - \mathfrak{y} \cdot \theta)\theta, \tag{2.2.40}$$

and $H^{n,l}$ are maps defined on $\mathbb{R}^d \times (\mathbb{R}^d)^N \times (\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}) \times \mathbb{R}^{K_n^z}$ taking values in $\mathbb{R}^{K_n^z}$ and given by

$$H^{n,l}(\mathcal{X}_0, W, \tilde{\mathfrak{u}}, \mathfrak{z}_l^{n,\cdot}) := -\frac{2}{\beta_K\sqrt{h}}(\mathfrak{G}^{\tilde{\mathfrak{u}}} - \omega_l^{n,\cdot} \cdot \mathfrak{z}_l^{n,\cdot})\omega_l^{n,\cdot}, \quad 0 \leqslant n \leqslant N-1, \quad 1 \leqslant l \leqslant d. \tag{2.2.41}$$

We importantly point out that we abuse the notation in (2.2.40) and (2.2.41) since the variable $(\mathcal{X}_0, W)$ stands for a vector of $\mathbb{R}^d \times (\mathbb{R}^d)^N$ and $(\mathcal{X}_0, W) \mapsto \mathfrak{G}^{\tilde{\mathfrak{u}}} = \mathfrak{G}^{\tilde{\mathfrak{u}}}(\mathcal{X}_0, W)$ is also defined by (2.2.27) while in (2.2.39) the random vector $W = (\mathcal{W}_{t_n})_{1\leqslant n\leqslant N}$ stands for the discrete path of the Brownian motion $\mathcal{W}$ and $\mathcal{X}_0$ for the starting value of X.
In (2.2.40) and (2.2.41), the deterministic constant β_K corresponding to a normalizing factor is introduced in order to control the $L^2(\mathbb{P})$-moment of the random vectors $(\mathfrak{G}^{\tilde{\mathfrak{u}}} - \mathfrak{y} \cdot \theta)\theta$ and $(\mathfrak{G}^{\tilde{\mathfrak{u}}} - \omega_l^{n,\cdot} \cdot \mathfrak{z}_l^{n,\cdot})\omega_l^{n,\cdot}$. Namely, we select β_K large enough so that

$$(\beta_K)^2 \geqslant (1 + \mathbb{E}\big[|\theta|^4\big]) \vee \max_{0\leqslant n\leqslant N-1, 1\leqslant l\leqslant d}(1 + \mathbb{E}\big[|\tilde{\omega}_l^{n,\cdot}|^4\big]) \tag{2.2.42}$$

with $\tilde{\omega}_\ell^{n,\cdot} = \frac{\omega_\ell^{n,\cdot}}{\sqrt{h}}$. Let us insist on the fact that the chosen β_K above should be uniform for all time grid π. It depends only on the level of approximation coming from the definition of the approximation spaces. This qualitative level of approximation is controlled by the number of basis function per time step, namely K.
For latter use, comparing (2.2.40) to (2.2.36) and (2.2.41) to (2.2.37), we remark that

$$\mathbb{E}[H^y(\mathcal{X}_0, W, \tilde{\mathfrak{u}}, \mathfrak{y})] = \frac{1}{\beta_K}\nabla_{\mathfrak{y}}\mathfrak{H}(\tilde{\mathfrak{u}}, \mathfrak{u}) \text{ and } \mathbb{E}\Big[H^{n,l}(\mathcal{X}_0, W, \tilde{\mathfrak{u}}, \mathfrak{z}_l^{n,\cdot})\Big] = \frac{1}{\beta_K\sqrt{h}}\nabla_{\mathfrak{z}_l^{n,\cdot}}\mathfrak{H}(\tilde{\mathfrak{u}}, \mathfrak{u}), \tag{2.2.43}$$

[1] Since it is also the procedure used for the *direct algorithm*, the numerical comparison between the two will be more relevant.

for $0 \leqslant n \leqslant N-1$, $l \in \{1, \dots, d\}$ and $(\tilde{\mathfrak{u}}, \mathfrak{u}) \in (\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z})^2$.

The implemented *Picard algorithm* is obtained by iterating a stochastic gradient operator which is the counterpart of Φ defined by (2.2.21) obtained by the numerical approximation that we now introduce.

Definition 2.2.6 *Let M be a positive integer. Let $\mathfrak{W} := (W^m)_{1 \leqslant m \leqslant M}$, be M discrete paths along the time grid π of the Brownian motion $\mathcal{W}$, $\mathfrak{X}_0 := (\mathcal{X}_0^m)_{1 \leqslant m \leqslant M}$, be M independent samples of the initial condition (and independent from $\mathfrak{W}$) and $(\gamma_m)_{m \geqslant 1}$ a deterministic sequence of positive real numbers satisfying:*

$$\sum_{m \geqslant 1} \gamma_m = \infty \quad \text{and} \quad \sum_{m \geqslant 1} \gamma_m^2 < \infty. \tag{2.2.44}$$

We set, for all $m \geqslant 1$,

$$\gamma_m^y = \gamma_m \quad \text{and} \quad \gamma_m^z = \frac{\gamma_m}{\sqrt{h}}\,. \tag{2.2.45}$$

Let $\mathfrak{u}_0 = (\mathfrak{y}_0, \mathfrak{z}_0)$ be a random vector taking values in $\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, independent of $(\mathfrak{X}_0, \mathfrak{W})$ and such that $\mathbb{E}[|\mathfrak{u}_0|^2] < \infty$.

The operator Φ_M, parametrized by $(\mathfrak{u}_0, \mathfrak{X}_0, \mathfrak{W})$, is given by

$$\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z} \ni \tilde{\mathfrak{u}} \mapsto \Phi_M(\mathfrak{u}_0, \mathfrak{X}_0, \mathfrak{W}, \tilde{\mathfrak{u}}) = \mathfrak{u}_M \tag{2.2.46}$$

where $\mathfrak{u}_M$ is the output of the SGD algorithm after M steps and is obtained as follows:

1. *The initial value is set to $\mathfrak{u}_0$.*
2. *Iteration: For $0 \leqslant m \leqslant M-1$, compute*

$$\mathfrak{y}_{m+1} = \mathfrak{y}_m - \gamma_{m+1}^y H^y(\mathcal{X}_0^{m+1}, W^{(m+1)}, \tilde{\mathfrak{u}}, \mathfrak{y}_m)\,, \tag{2.2.47}$$

and

$$\mathfrak{z}_{l,m+1}^{n,\cdot} = \mathfrak{z}_{l,m}^{n,\cdot} - \gamma_{m+1}^z H^{n,l}(\mathcal{X}_0^{m+1}, W^{(m+1)}, \tilde{\mathfrak{u}}, \mathfrak{z}_{l,m}^{n,\cdot})\,, \tag{2.2.48}$$

for any $0 \leqslant n \leqslant N-1$ and any $1 \leqslant l \leqslant d$.

Definition 2.2.7 (Implemented *Picard algorithm*) *For a prescribed positive integer P:*

1. *Initialization: Select a random vector $\mathfrak{u}_0^0$ taking values in $\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$ such that $\mathbb{E}[|\mathfrak{u}_0^0|^2] < \infty$. Set $\mathfrak{u}_M^0 := \mathfrak{u}_0^0$.*
2. *Iteration: for $1 \leqslant p \leqslant P$, simulate independently a set of M independent discrete paths $\mathfrak{W}^p$ of the Brownian motion $\mathcal{W}$, independent initial condition $\mathfrak{X}_0^p$ and an initial starting point $\mathfrak{u}_0^p$ (independently also of $\mathfrak{u}_0^0$, $\mathfrak{u}_0^j$ and of the previous $(\mathfrak{X}_0^j, \mathfrak{W}^j)$, $1 \leqslant j \leqslant p-1$), and compute $\mathfrak{u}_M^p := \Phi_M(\mathfrak{u}_0^p, \mathfrak{X}_0^p, \mathfrak{W}^p, \mathfrak{u}_M^{p-1})$ as in Definition 2.2.6.*

The output of the algorithm is then $\mathfrak{u}_M^P$.

Remark 2.2.6 *i) The choice of the learning sequence* γ *for the SGD algorithm* (2.2.47), (2.2.48) *might be delicate in practice, see e.g. Section 2.3.1.3.*

ii) The initialisation is random in the above algorithm. We do not always follow this procedure in our numerical experiments, see Section 2.3.

iii) The numerical complexity $\mathcal{C}$ *of the full algorithm is the sum of the local complexity of each SGD algorithm so that*

$$\mathcal{C} = O_d(PNKM)\,. \tag{2.2.49}$$

Using the output $\mathfrak{u}_M^P$ of the *Picard algorithm*, we set the approximating function at time 0 to be:

$$\mathcal{U}_M^P(x) := \sum_{k=1}^{K^y} (\mathfrak{y}_M^P)^k \psi_y^k(x)\ , \tag{2.2.50}$$

recalling (2.2.8).

We then aim to control the following mean squared error:

$$\mathcal{E}_{\text{MSE}} := \mathbb{E}\Big[\big|\mathcal{U}_M^P(\mathcal{X}_0) - u(0,\mathcal{X}_0)\big|^2\Big]\ . \tag{2.2.51}$$

We obtain an explicit upper bound on the mean-squared error when specifying the parameters of the algorithm as follows. For $\gamma > 0$, $\rho \in (\frac{1}{2}, 1)$, we set $\gamma_m := \gamma m^{-\rho}$, $m \geqslant 1$ and we assume that the number of steps M in the SGD algorithm satisfies, for some $\eta \geqslant 0$,

$$\gamma \frac{\alpha_K}{\beta_K} M^{1-\rho} \geqslant \frac{\sqrt{2}}{2} \quad \text{and} \quad \frac{\kappa_K}{h \wedge \alpha_K}\left(e^{-2\sqrt{2}\ln(2)\gamma\frac{\alpha_K}{\beta_K}M^{1-\rho}} + \frac{\beta_K}{\alpha_K M^\rho}\right) \leqslant \eta\ . \tag{2.2.52}$$

Theorem 2.2.1 *Let Assumption 2.2.1, Assumption 2.2.2 (i), (ii), Assumption 2.2.3 and* (2.2.52) *hold. If* LT^2 *and* η *are small enough, then there exists* $\delta < 1$ *such that*

$$\mathcal{E}_{\text{MSE}} \leqslant C_{\rho,\gamma}\left(\delta^P + \frac{\kappa_K}{h \wedge \alpha_K}\left(e^{-2\sqrt{2}\ln(2)\gamma\frac{\alpha_K}{\beta_K}M^{1-\rho}} + \frac{\beta_K}{\alpha_K M^\rho}\right) + h + \mathcal{E}_\psi\right). \tag{2.2.53}$$

for some positive constant $C_{\rho,\gamma}$, *where we recall that* $\mathcal{E}_\psi$ *is given by* (2.2.18).

Remark 2.2.7 *1. As expected, the above upper bound is the sum of the error due to the Picard iteration, the error induced by the SGD algorithm, the discrete-time approximation error, recall* (2.2.19), *and the error* $\mathcal{E}_\psi$ *generated by the approximation in the functional spaces* $(\mathscr{V}^y, \mathscr{V}_n^z)$.

2. The smallness condition on LT^2 *is precisely given in the statement of Proposition 2.4.4. This condition should not come as a surprise since we use Picard iteration. The smallness condition on* η *is not restrictive in practice as the quantity it controls should go to zero to obtain the convergence of the numerical procedure.*

To deduce a rate of convergence from (2.2.53), one has to chose the approximation basis functions $(\psi_y^k)_{1\leqslant k\leqslant K^y}$ and $(\psi_n^k)_{0\leqslant n\leqslant N-1,1\leqslant k\leqslant K_n^z}$ and to set optimally the algorithm's parameters. The choice of the basis function has a dramatic impact on the complexity of the algorithm. In the next section, we work with sparse grid approximation and we are able to show that the complexity is controlled both theoretically and in practice under Assumption 2.2.2.

2.3 Convergence results for sparse grid approximation

Both the implemented *direct algorithm*, see Definition 2.2.3, and the implemented *Picard algorithm*, see Definition 2.2.7, rely on the choice of the approximation spaces $\mathscr{V}^y$ and $\mathscr{V}_n^z$, $0\leqslant n\leqslant N-1$ and the choice of the related basis functions $(\psi_y^k)_{1\leqslant k\leqslant K^y}$ and $(\psi_n^k)_{0\leqslant n\leqslant N-1,1\leqslant k\leqslant K_n^z}$. The impact is both theoretical, in terms of convergence rate and numerical complexity, and practical in terms of computational time. We choose here to use sparse grid approximations. This will allow us to obtain interesting numerical complexity results in the setting of Assumption 2.2.2, see Theorem 2.3.1. We carefully investigate the convergence of the implemented *Picard Algorithm*. It is not the first time that sparse grid approximations are investigated in the context of linear regression. We will use the framework introduced in [14]. Note however that some restriction in the choice of sparse grid approximations are introduced by Assumption 2.2.3.

The basis functions are built using elementary bricks that have a compact support included in the bounded domain

$$\mathcal{O}_n=\prod_{l=1}^{d}[\mathfrak{a}_l^n,\mathfrak{b}_l^n]\quad\text{where}\quad\mathfrak{a}_l^n<\mathfrak{b}_l^n\ \text{ for } l\in\{1,\dots,d\}. \tag{2.3.1}$$

The domain specification strongly depends on the applications under study. We will consider two main cases in this work.

1. For all $1\leqslant n\leqslant N-1$,

$$\mathcal{O}_n=\prod_{l=1}^{d}[\mathfrak{a}_l,\mathfrak{b}_l]=:\mathcal{O}. \tag{2.3.2}$$

   Namely, the coefficients $\mathfrak{a}$ and $\mathfrak{b}$ do not depend on n. This will be the case in Section 2.3.1.2 where we consider coefficient functions that are $\mathcal{O}$-periodic.

2. Alternatively, the coefficient $\mathfrak{a}$ and $\mathfrak{b}$ are functions of the time-step but also of the diffusion coefficients (b,σ), recall (2.1.2), and the PDE dimension d. Namely

$$\mathfrak{a}^n:=\mathfrak{a}(t_n,b,\sigma,d)\quad\text{and}\quad\mathfrak{b}^n:=\mathfrak{b}(t_n,b,\sigma,d). \tag{2.3.3}$$

However, In both cases the basis functions are obtained by a transformation of the domain $[0,1]^d$ on which we define the primary basis using *sparse grids*. The

transformation is defined as follows:

$$\tau_n : \mathbb{R}^d \ni x \mapsto \tau_n(x) = \left(\frac{x_1 - \mathfrak{a}_1^n}{\mathfrak{b}_1^n - \mathfrak{a}_1^n}, \dots, \frac{x_d - \mathfrak{a}_d^n}{\mathfrak{b}_d^n - \mathfrak{a}_d^n}\right)^\top \in \mathbb{R}^d . \tag{2.3.4}$$

We will introduce two types of basis functions: the first one, based on pre-wavelet basis, follows from [14] and the second one, based on hat functions modified at the boundary of the domain, follows from [43].

2.3.1 Convergence results for the pre-wavelet basis

2.3.1.1 Definition of the pre-wavelet basis

We describe here the elementary bricks that are used to build the basis functions of the approximation spaces.

For a level $l \in \mathbb{N}$ and an index $i \in \{0, \dots, 2^l\}$, we first consider the family of hat functions given by

$$\phi^{l,i}(x) = \phi(2^l x - i) \text{ with } \phi(x) = \begin{cases} 1 - |x| & if \ -1 < x < 1 \\ 0 & otherwise . \end{cases} \tag{2.3.5}$$

The univariate pre-wavelet basis functions $\chi^{l,i} : \mathbb{R} \to \mathbb{R}$ are defined by

$$\chi^{0,0} = 1_{[0,1]} , \chi^{0,1} = \phi^{0,1} , \chi^{1,1} = 2\phi^{1,1} - 1 \tag{2.3.6}$$

and for $l \geqslant 2$, $i \in I_l \backslash \{1, 2^l - 1\}$ with $I_l := \{1 \leqslant i \leqslant 2^l - 1 \,|\, i \text{ odd}\}$

$$\chi^{l,i} = 2^{\frac{l}{2}} \left(\frac{1}{10}\phi^{l,i-2} - \frac{6}{10}\phi^{l,i-1} + \phi^{l,i} - \frac{6}{10}\phi^{l,i+1} + \frac{1}{10}\phi^{l,i+2}\right) . \tag{2.3.7}$$

For the boundary points $i \in \{1, 2^l - 1\}$, we set

$$\chi^{l,1} = 2^{\frac{l}{2}} \left(-\frac{6}{5}\phi^{l,0} + \frac{11}{10}\phi^{l,1} - \frac{3}{5}\phi^{l,2} + \frac{1}{10}\phi^{l,3}\right) \text{ and } \chi^{l,2^l-1}(x) = \chi^{l,1}(1 - x) , \, x \in \mathbb{R} . \tag{2.3.8}$$

The multivariate pre-wavelet function on $\mathbb{R}^d$ are obtained by a classical tensor-product approach. For a multi-index level $\mathbf{l} = (l_1, \dots, l_d)$ and a multi-index position $\mathbf{i} = (i_1, \dots, i_d)$,

$$\chi^{\mathbf{l},\mathbf{i}}(x) = \prod_{j=1}^{d} \chi^{l_j,i_j}(x^j). \tag{2.3.9}$$

In this multivariate case, the index sets are given by

$$\mathbf{I}_\mathbf{l} = \left\{ \mathbf{i} \in \mathbb{N}^d \,\middle|\, \begin{array}{rl} 0 \leqslant i_j \leqslant 1 & if \ l_j = 0, \\ i_j \in I_{l_j} & if \ l_j > 0, \end{array} \text{ for all } 1 \leqslant j \leqslant d \right\} . \tag{2.3.10}$$

The hierarchical increment spaces are then defined for $\mathbf{l} \in \mathbb{N}^d$ by

$$\mathscr{W}_\mathbf{l} := \text{span}\{\chi^{\mathbf{l},\mathbf{i}} \,|\, \mathbf{i} \in \mathbf{I}_\mathbf{l}\} .$$

The sparse grid space approximation at level ℓ is given by

$$\mathscr{S}_\ell := \bigoplus_{\mathbf{l}\in\mathcal{L}_\ell} \mathscr{W}_\mathbf{l}, \quad \mathcal{L}_\ell := \{\mathbf{l}\in\mathbb{N}^d\,, \zeta_d(\mathbf{l}) \leqslant \ell\} \tag{2.3.11}$$

with $\zeta_d(\mathbf{0}) := 0$ and for $\mathbf{l} \neq \mathbf{0}$

$$\zeta_d(\mathbf{l}) = |\mathbf{l}|_1 - d + |\{j|l_j = 0\}| + 1\,, \tag{2.3.12}$$

where for a multi-index $\mathbf{l}\in\mathbb{N}^d$ we recall that $|\mathbf{l}|_1 = \sum_{\ell=1}^d l_\ell$ and that $|A|$ is the cardinality of A.
The key point here is that the dimension of $\mathscr{S}_\ell$ satisfies

$$\dim(\mathscr{S}_\ell) = O(2^\ell \ell^{d-1})\,, \tag{2.3.13}$$

so that the curse of dimensionality only appears with respect to the level ℓ, see [38] (and also in the constant related to the notation $O(.)$). The key point now is that the approximation error when using the sparse space is also controlled if the function to be approximated is smooth enough. To this end, for the fixed open domain $(0,1)^d$, we consider the space of function with mixed derivatives $H^2_{mix}((0,1)^d)$ (see the section Notation for a precise definition). Then, for any $v\in H^2_{mix}((0,1)^d)$, it holds

$$\inf_{\xi\in\mathscr{S}_\ell} \|\xi - v\|^2_{L^2((0,1)^d)} \leqslant C 2^{-4\ell}\ell^{d-1}\|v\|^2_{H^2_{mix}((0,1)^d)} \tag{2.3.14}$$

for some positive constant $C := C(d)$. We refer e.g. Theorem 3.25 in [13] for a proof of this result. Again, we importantly emphasize that in the above control of the error the curse of dimensionality only appears with respect to the level ℓ.

Remark 2.3.1 *The number of basis functions is thus $K = \dim(\mathscr{S}_\ell)$. We denote by $k : \mathcal{C} \mapsto \{1,\dots,K\}$ any bijection enumerating $\mathcal{C}$. We will often slightly abuse the notation and write directly $(\psi_n^k)_{1\leqslant k\leqslant K}$ instead of $(\psi_n^{(\mathbf{l},\mathbf{i})})_{(\mathbf{l},\mathbf{i})\in\mathcal{C}}$ to be consistent with the notation introduced in the previous section.*

2.3.1.2 The *Picard Algorithm* in the case of periodic coefficients

In this section, we work under the setting of Assumption 2.2.2 (iv). To alleviate the notation – but without loss of generality – we assume that the coefficients are 1-*periodic* in the following sense: for $\lambda = b, \sigma$ or g

$$\lambda(x+q) = \lambda(x)\,, \text{ for all } (x,q)\in\mathbb{R}^d\times\mathbb{Z}^d\,, \tag{2.3.15}$$

which implies the same property for the value function u and its derivatives.
We thus consider here that $\mathcal{O} = [0,1]^d$, recall (2.3.2) and $\tau = I_d$, recall (2.3.4). Here, we are looking for an approximation $\mathcal{U}_M^P(\cdot)$ of $u(0,\cdot)$ on the whole domain $\mathcal{O}$, recall (2.2.50). We thus set $\mathcal{X}_0$ to be uniformly distributed on $(0,1)^d$, which means that Assumption 2.2.1(i) holds true.
For sake of clarity, we summarize the current setting in the following assumption:

Assumption 2.3.1 *Let Assumption 2.2.1(i) and Assumption 2.2.2 hold true. Moreover, set* $\mathcal{O} = [0,1]^d$ *and* $\mathcal{X}_0 \sim \mathcal{U}((0,1)^d)$.

To take into account the periodic setting in our approximation, let us first define the 1-*periodisation* of a compactly supported function φ by

$$\breve{\varphi}(x) := \sum_{q \in \mathbb{Z}^d} \varphi(x+q) \ , \ \text{for all } x \in \mathbb{R}^d \ . \tag{2.3.16}$$

The basis functions ψ are then given by $\psi = \breve{\chi}$. Namely, for any $0 \leqslant n \leqslant N-1$, for an approximation date t_n, we introduce the set of functions

$$\mathscr{V}_n^z := \{\xi : \mathbb{R}^d \mapsto \mathbb{R} \,|\, \xi(x) = \breve{v}(x), \ \text{ for some } \ v \in \mathscr{S}_\ell\} \ . \tag{2.3.17}$$

Moreover, at the initial time, the approximation of $u(0,\cdot)$ will also be computed in

$$\mathscr{V}^y := \{\xi : \mathbb{R}^d \mapsto \mathbb{R} \,|\, \xi(x) = \breve{v}(x), \ \text{ for some } \ v \in \mathscr{S}_\ell\} \ . \tag{2.3.18}$$

Remark 2.3.2 *We could have set an approximation level different for each time step, however we shall not use this possibility in our theoretical or numerical convergence results. We thus simply consider a fixed positive level* ℓ *of approximation, that, obviously, will be chosen later in an optimal way.*

Let also introduce the function

$$\mathbb{R}^d \ni x \mapsto \widehat{x} \in [0,1)^d \tag{2.3.19}$$

such that $\widehat{x}$ and x belong to the same equivalence class in $\mathbb{R}^d/\mathbb{Z}^d$. Denoting by $\mathbb{P}_{X_{t_n}}$ the probability measure on $\mathbb{R}^d$ associated to the random vector X_{t_n} given by the Euler-Maruyama scheme (2.2.3) taken at time t_n and starting from $\mathcal{X}_0$ at time 0 and using Lemma 2.2.1, we remark that the boundary of the domain $\mathcal{O}$ has null $\mathbb{P}_{X_{t_n}}$-measure. We thus deduce

$$\breve{\psi}(X_{t_n}) = \psi(\widehat{X}_{t_n}) \quad \mathbb{P}-a.s. \ , \tag{2.3.20}$$

and in practice we should work with the latter quantity. Namely, we construct our approximation scheme using:

$$Y_0^{\mathfrak{u}} := \sum_{k=1}^{K^y} \psi_y^k(\widehat{X}_0)\mathfrak{y}^k \quad \text{and} \quad Z_{t_n}^{\mathfrak{u}} := \sum_{k=1}^{K_n^z} \psi_n^k(\widehat{X}_{t_n})\mathfrak{z}^{n,k}, \ \text{ for } 0 \leqslant n \leqslant N-1 \, , \tag{2.3.21}$$

with $\mathfrak{u} = (\mathfrak{y}, \mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$.

Under the current setting of periodic coefficients and sparse grid approximation, we take benefit of the convergence results given in Theorem 2.2.1 to obtain our main theoretical result on the complexity of the *Picard algorithm*. Indeed, the next theorem shows that the curse of dimensionality is tamed by using the sparse grid approximation.

Theorem 2.3.1 *Let Assumption 2.3.1 hold and assume that LT^2 is small enough. For a prescribed $\varepsilon > 0$, the complexity $\mathcal{C}_\varepsilon$, defined in Remark 2.2.6, of the full* Picard algorithm *in order to achieve a global error $\mathcal{E}_{\text{MSE}}$ of order ε^2, recall (2.2.51), satisfies*

$$\mathcal{C}_\varepsilon = O_d(\varepsilon^{-\frac{5}{2}(1+2\iota)} |\log_2(\varepsilon)|^{1+\frac{45+50\iota}{36}(d-1)})$$

for any $1 < \iota < \frac{9}{5}$.

The proof of this theorem is given in Section 2.4.4 where the algorithm's parameters are optimally set with respect to ε.

2.3.1.2.1 Periodic example We consider here 1-periodic coefficients on $\mathbb{R}^d$. The coefficients of the forward SDE (2.1.2) are given by, for $x \in \mathbb{R}^d$,

$$b_i(x) = 0.2\sin(2\pi x_i)\,,\ \sigma_{i,j}(x) = \frac{1}{\sqrt{d\pi}}(0.25 + 0.1\cos(2\pi x_i))\mathbf{1}_{\{i=j\}}\,,\ 1 \leqslant i,j \leqslant d\,.$$

The coefficients of the BSDE reads

$$g(x) = \frac{1}{\pi}\left(\sin\left(2\pi\sum_{i=1}^d x_i\right) + \cos\left(2\pi\sum_{i=1}^d x_i\right)\right), x \in \mathbb{R}^d,$$

$$f(t,x,y,z) = 2\pi^2 y\sum_{i=1}^d (\sigma_{i,i}(x))^2 - \sum_{i=1}^d b_i(x)\frac{z_i}{\sigma_{i,i}(x)} + h(t,x), t \in [0,T], x \in \mathbb{R}^d, y \in \mathbb{R}, z \in \mathbb{R}^d,$$

where $h(t,x) = 2\left(\cos(2\pi\sum_{i=1}^d x_i + 2\pi(T-t)) - \sin(2\pi\sum_{i=1}^d x_i + 2\pi(T-t))\right)$. The explicit solution is given by

$$u(t,x) = \frac{1}{\pi}\left(\sin\left(2\pi\sum_{i=1}^d x_i + 2\pi(T-t)\right) + \cos\left(2\pi\sum_{i=1}^d x_i + 2\pi(T-t)\right)\right), x \in \mathbb{R}^d.$$

We perform the test for $d = 3$ and $M = 100000, N = 10, T = 0.3, level = 3$ by *Picard Algorithm* with $P = 5$, then there are $K^y = K_n^z = K = 225$ basis functions. We obtain a mean square error $\mathcal{E}_{\text{MSE}} = 0.0201$ at the 5-th Picard iteration: See Figure 2.1 displaying the learning performance. The parameters of the test are shown in Table 2.6 in the appendix 2.5.1.

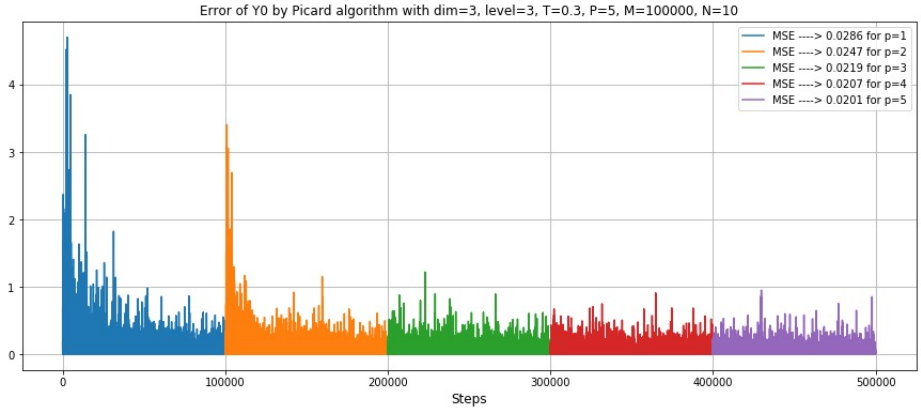

Figure 2.1 – $m \mapsto |\hat{Y}_0^m - u(0, \mathcal{X}_0^m)|^2$ for the *Picard algorithm*, $d = 3$. The MSE is computed by the mean of the last 10000 steps of each Picard iteration.

2.3.1.3 Numerical convergence of the *Picard* and *direct Algorithm*

We will now investigate numerically the behavior of the *Picard algorithm* and *direct algorithm* on "test" examples that have been already considered in the literature. In particular, this will allow us also to compare our methods to existing methods as the ones investigated in [36, 65].

For this section, we work in the setting of Assumption 2.2.1(ii). This means that at the initial time, the output $(\hat{y}_0, \hat{z}_0)$ of the algorithms: $(\hat{y}_0, \hat{z}_0) = (\mathfrak{y}_M, \mathfrak{z}_M^0)$, for the *direct algorithm*, recall Definition 2.2.3, or $(\hat{y}_0, \hat{z}_0) = (\mathfrak{y}_M^P, (\mathfrak{z}_M^P)^0)$, for the *Picard algorithm*[2], recall Definition 2.2.7, are approximating $(u(0, x_0), \sigma^\top \nabla_x u(0, x_0)) \in \mathbb{R} \times \mathbb{R}^d$. Since, these are one point values, there is no need to introduce basis functions at the initial time and the approximating spaces are just $\mathscr{V}_0^y = \mathbb{R}$ and $\mathscr{V}_0^z = \mathbb{R}^d$. Then, for any $1 \leqslant n \leqslant N-1$, for the discrete time t_n, we set the approximating space as follows:

$$\mathscr{V}_n^z := \{\xi : \mathbb{R}^d \mapsto \mathbb{R} \,|\, \xi(x) = v(\tau_n(x)), \text{ for some } v \in \mathscr{S}_\ell\}\,, \tag{2.3.22}$$

recall (2.3.4). In particular, the basis functions are given by $\psi_n^k(x) = \chi^k(\tau_n(x))$, recall (2.3.9) and Remark 2.3.1.

We now report more specifically the various algorithms parameters that have been used in practice. The first thing to note is that we are able to obtain good results with a low level of approximation. Indeed, in all our numerical tests, we set the level $\ell = 3$. The Table 2.1 below indicates the number of basis functions that have theoretically to be considered when including boundary function.

[2]Deviating slightly from Definition 2.2.7, we will use for the initialization of the current SGD step, the last value computed at the previous step instead of a random value.

dimensions \ levels	$\ell \leqslant 3$	$\ell \leqslant 4$	$\ell \leqslant 5$
d=2	49	113	257
d=3	225	593	1505
d=4	945	2769	7681
d=5	3753	12033	36033

Table 2.1 – The number of functions in the sparse grid approximation with boundary for different dimensions and levels.

Next, we need to define the domain $\mathcal{O}_n, 1 \leqslant n \leqslant N-1$, where the approximation will be computed, which depends on the underlying process, recall (2.3.1)-(2.3.3). We will consider two cases in our simulations, each component of the forward SDE is given by a Brownian motion with drift μ and volatility σ: $t \mapsto x_0 + \mu t + \sigma W_t$ or a geometric Brownian motion: $t \mapsto x_0 \exp((\mu - \sigma^2/2)t + \sigma W_t)$.

1. For the Brownian motion with drift, we set

$$\mathcal{O}_n = x_0 + [\mu t_n - r\sigma\sqrt{t_n}, \mu t_n + r\sigma\sqrt{t_n}]^d, \quad \text{for some } r \in \mathbb{R}^+. \tag{2.3.23}$$

2. For the geometric Brownian motion, we set

$$\mathcal{O}_n = [x_0 e^{R - r\sigma\sqrt{t_n}}, x_0 e^{R + r\sigma\sqrt{t_n}}], \quad R = (\mu - \frac{1}{2}\sigma^2)t_n, \quad \text{for some } r \in \mathbb{R}^+. \tag{2.3.24}$$

Finally, a delicate parameter to chose is the the learning rate. Empirically, it was set to: for $\lambda \in \{\mathfrak{y}, \mathfrak{z}^{0,\cdot}\}_{n=0} \cup \{\mathfrak{z}^{n,\cdot}\}_{1 \leqslant n \leqslant N-1}$,

$$\gamma_m(\lambda, t_n, \alpha, \beta_0, \beta_1, m_0) = \frac{\beta_1(\lambda)n + \beta_0(\lambda)}{1 + (m + m_0(\lambda))^{\alpha(\lambda)}}, \quad 1 \leqslant m \leqslant M, \tag{2.3.25}$$

where $\beta_0, \beta_1 \in \mathbb{R}^+, m_0 \in \mathbb{N}^+, \alpha \in \left(\frac{1}{2}, 1\right]$.

Remark 2.3.3 *i) m_0 is a suitable positive number to decrease the learning rates for avoiding a big jump of the estimated λ in the beginning steps of the algorithm.*
ii) The parameter $r \in \mathbb{R}^+$ is a suitable number to balance the running time and the errors of the algorithms.
iii) Both β_0 and α can be used to adjust the converge speed and the variance of the estimated λ. Suitable parameters make the algorithm more stable, converge faster and reduce the variance of the estimated λ.
iv) Usually, we increase the value of α or decrease the value of (β_0, β_1) gradually to decrease the convergence rate with the increase of step $p, 1 \leqslant p \leqslant P$ for Picard algorithm.

Concerning the number of steps in the SGD algorithm, we make the following remark.

Remark 2.3.4 *We used two techniques to control M in order to reduce the computational cost:*
i) We use $\alpha \in \left(0, \frac{1}{2}\right]$ which still works well as the SGD algorithm can converge faster.
ii) If $\beta_1(\lambda) \equiv 0$, for M large enough, the algorithm eventually converge, but $\{\mathfrak{z}^{n,k}\}_{1\leqslant n\leqslant N-1}^{1\leqslant k\leqslant K}$ convergence becomes slower and slower with the increase of n (the time step). We thus choose $\beta_1(\lambda) > 0$ in practice to make all $\{\mathfrak{z}^{n,k}\}_{1\leqslant n\leqslant N-1}^{1\leqslant k\leqslant K}$ converge altogether with a smaller M (thanks to the learning rates increase with n).

The remaining parameters are precised in the examples below. We refer also Section to the Appendix 2.5.1 for the collection of all algorithm parameters values used in the numerical simulation.

2.3.1.3.1 Quadratic model First, we consider the quadratic example, whose driver is set to

$$f(y,z) = a|z|^2 = a(z_1^2 + z_2^2 + \cdots + z_d^2), \quad y \in \mathbb{R},\ z \in \mathbb{R}^d, \tag{2.3.26}$$

where $a \in \mathbb{R}$ is a constant, and the terminal condition to

$$g(x) = \log\left(\frac{1+|x|^2}{2}\right), \quad x \in \mathbb{R}^d. \tag{2.3.27}$$

The explicit solution can be obtained through the Cole-Hopf transformation(see e.g. [27, 36]):

$$y_t = u(t,x) = \begin{cases} \frac{1}{a}\log\mathbb{E}\left[e^{a\cdot g(x+\mathcal{W}_{T-t})}\right], & a \neq 0 \\ \mathbb{E}[g(x+\mathcal{W}_{T-t})], & a = 0 \end{cases}$$

and

$$z_t^i = \partial_{x_i} u(t,x) = \frac{\mathbb{E}\left[\partial_{x_i} g(x+\mathcal{W}_{T-t}) e^{a\cdot g(x+\mathcal{W}_{T-t})}\right]}{\mathbb{E}\left[e^{a\cdot g(x+\mathcal{W}_{T-t})}\right]}, \quad i = 1,2,\cdots,d.$$

Thus, to obtain a numerical reference solution and 95% confidence interval for y_0 and $z_0^i, i = 1,2,\cdots,d$, we use classical Monte Carlo estimation of the expectations.

The underlying diffusion $\mathcal{X}$ is given by the Brownian motion $\mathcal{W}$, and the parameters are selected as follows: $a = 1, M = 2000, N = 10, T = 1, x_0 = (0,\cdots,0)$. We compute a reference solution $\bar{y}_0 = 1.0976$ with 95% confidence interval $(1.0943, 1.1009)$ when $d = 5$ by Monte Carlo method using 10^5 simulation paths. Figure 2.2 shows the numerical approximation of y_0 and its 95% confidence interval by the same color line of the 5-dimensional quadratic model by *direct algorithm* and the *deep learning algorithm* introduced in [36] which used a large neural network contains $N-1$ fully-connected sub-neural network to represent $Z_{t_i}, i = 1,\cdot,N-1$ and minimized the loss function at the maturity T. The difference of $\hat{y}_0$ between our SGD algorithm and Monte Carlo simulation is less than 10^{-2}. It turns out that for this "low" dimensional example and with this set of parameter, it is more precise than the *deep BSDEs solver*.

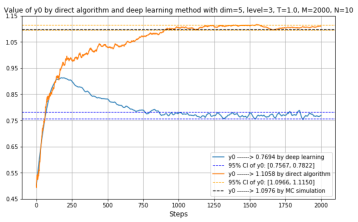

Figure 2.2 – $\hat{y}_0$ for the quadratic model with d=5 and T=1 by *direct algorithm* and deep learning algorithm.

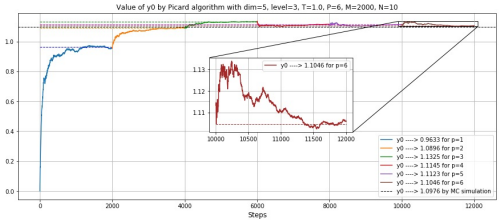

Figure 2.3 – The value of $\hat{y}_0$ by *Picard algorithm* with d=5, level=3, T=1, P=6, M=2000.

For $d = 5, M = 2000, N = 10, P = 6, T = 1, a = 1$, Figure 2.3 shows that $\hat{y}_0$ converges for each Picard iteration, and overall $\hat{y}_0 \to 1.1046$. We can observe that $\hat{y}_0$ is very close to the reference solution $\bar{y}_0$ when the number of iteration p is greater or equal to 4.

2.3.1.3.2 A financial model We now report our numerical results for a model with a financial flavour. The underlying process $\mathcal{X}$ follows a d-dimensional geometric Brownian motion, for $\mu \in \mathbb{R}$, $\sigma > 0$, namely

$$d\mathcal{X}_t^i = \mathcal{X}_t^i(\mu dt + \sigma dW_t^i), \quad i = 1, 2, \cdots d, \quad \mathcal{X}_0 = x_0 \in (\mathbb{R}_+)^d .$$

The driver of the BSDE is given by, for $(y, z) \in \mathbb{R} \times \mathbb{R}^d$,

$$f(y,z) = -R^l y - \frac{\mu - R^l}{\sigma} \sum_{i=1}^{d} z_i + (R^b - R^l) \max\left\{0, \frac{\sum_{i=1}^{d} z_i}{\sigma} - y\right\},$$

and the terminal condition

$$g(x) = \max\left\{\left[\max_{1 \leqslant i \leqslant d} x_i\right] - K_1, 0\right\} - 2\max\left\{\left[\max_{1 \leqslant i \leqslant d} x_i\right] - K_2, 0\right\}.$$

Hence, for all $t \in [0, T), x \in \mathbb{R}^d$, it holds that $u(T, x) = g(x)$ and

$$\frac{\partial u}{\partial t} + \frac{\sigma^2}{2} \sum_{i=1}^{d} x_i^2 \frac{\partial^2 u}{\partial x_i^2} - \min\left\{R^b(u - \sum_{i=1}^{d} x_i \frac{\partial u}{\partial x_i}), R^l(u - \sum_{i=1}^{d} x_i \frac{\partial u}{\partial x_i})\right\} = 0 . \quad (2.3.28)$$

This is a typical example of "non-linear market" specification, where there are two different interest rates for borrowing and lending money, see Bergman [11] and e.g. [36, 49, 18, 31, 9], where this example has been used as a test example for numerical methods for BSDEs.

In our numerical test below, we set the parameters as follows: $N = 10, M = 6000$, $\mu = 0.06, \sigma = 0.2$, $R^l = 0.04, R^b = 0.06$, $K_1 = 110$, $K_2 = 130$, $T = 0.5$ and $x_0 = (100, \cdots, 100)$.
Table 2.2 compares the results of the *direct algorithm* and the *deep learning algorithm* [36] when $d = 2, 4$. The approximated value of y_0 obtained by the two methods are very close. Figure 2.4 and Figure 2.5 show the performance of the *Picard algorithm* with parameters $d = 4, P = 12$, and $\hat{y}_0$ converges to 7.1352 at the last step.

dimensions	SGD algo with Sparse grids		Deep learning scheme [36]	
	y_0	95% CI of y_0	y_0	95% CI of y_0
d=2	4.3332	[4.2921, 4.3743]	4.3516	[4.3420, 4.3612]
d=4	7.0960	[7.0432, 7.1487]	7.1130	[7.0649, 7.1611]

Table 2.2 – Comparison of the *direct algorithm* and the *deep learning algorithm* for the financial model.

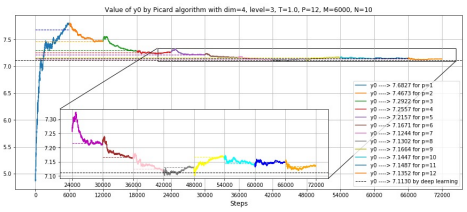

Figure 2.4 – Approximation $\hat{y}_0$ by *Picard algorithm* when d=4 and T=0.5.

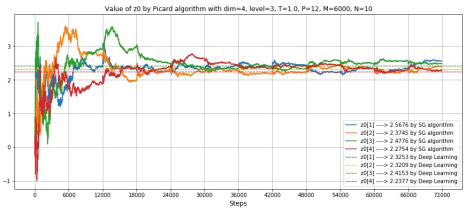

Figure 2.5 – Approximation $\hat{z}_0$ by *Picard algorithm* when d=4 and T=0.5.

2.3.1.4 Limits of the *Picard Algorithm*

We now illustrate on a numerical example that the smallness assumption may be necessary to obtain the convergence of the *Picard Algorithm*. To this end, we consider the following model. For a given $a \in \mathbb{R}$, the BSDE driver is given by

$$f(y,z) := \arctan(ay) + \sum_{j=1}^{d} z_j, \quad (y,z) \in \mathbb{R} \times \mathbb{R}^d,$$

and the terminal condition

$$g(x) := \frac{e^{1+\mathbf{1}\cdot x}}{1+e^{1+\mathbf{1}\cdot x}}, \quad x \in \mathbb{R}^d.$$

The underlying process $\mathcal{X}$ is simply equal to the Brownian motion $\mathcal{W}$, namely $b = 0$, $\sigma = I_d$. We set the terminal time $T = 1$ and the dimension $d = 2$.

We study numerically the above model for different value of a, which controls the Lipschitz constant of f, in the case of the *Picard Algorithm*. The value obtained are compared to the ones obtained by two other methods: a multistep scheme in [22] and the *deep BSDEs solver* of [36]. The values obtained by these two methods are considered to be close to the true solution.

When $a = -0.4$, Figure 2.6 shows that $\hat{y}_0$ converges. However, this is not the case anymore when $a = -1.5$, see Figure 2.7, as $\hat{y}_0$ oscillates between two values. Actually, we see on Figure 2.8 that a bifurcation occurs for the *Picard Algorithm* around $a = -0.8$.

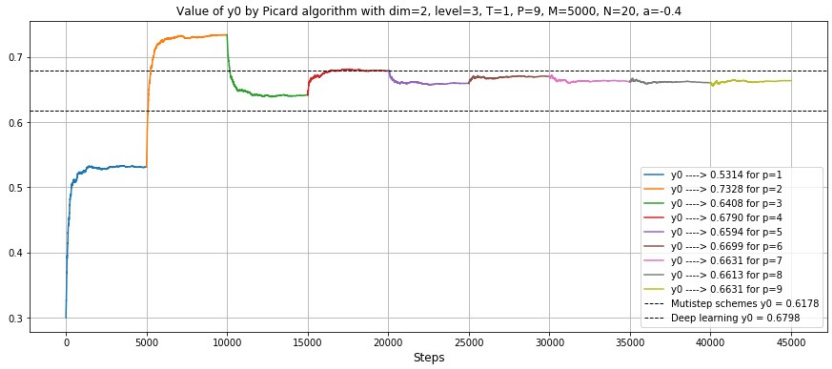

Figure 2.6 – The value of $\hat{y}_0$ by *Picard algorithm* with d=2, level=3, T=1, P=9, M=5000, a=-0.4.

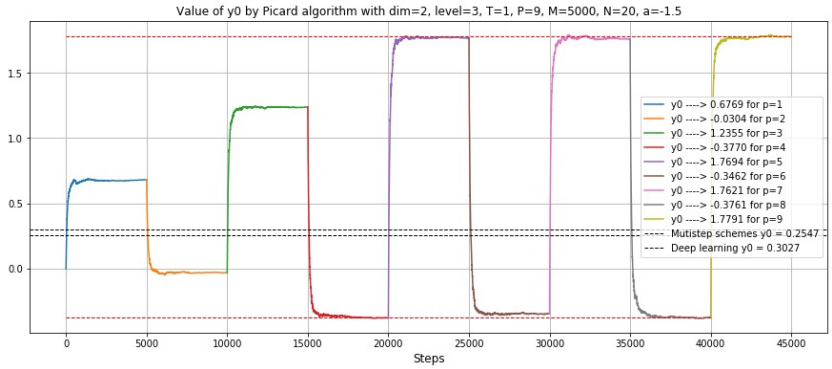

Figure 2.7 – The value of $\hat{y}_0$ by *Picard algorithm* with d=2, level=3, T=1, P=9, M=5000, a=-1.5

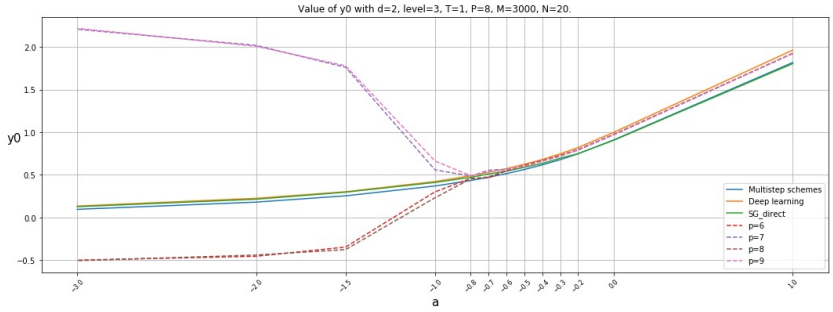

Figure 2.8 – The value of $\hat{y}_0$ by *Picard algorithm*, *direct algorithm* and deep learning method with d=2, level=3, T=1, P=9, M=5000. The last four steps are shown for the *Picard Algorithm* illustrating the bifurcation phenomenon. Note that the *direct algorithm* does not exhibit such behaviour.

2.3.2 Numerical results with the modified hat functions basis

In the previous section, using the pre-wavelet basis, we were able to establish a theoretical upper-bound on the global complexity for the *Picard algorithm* and to show

that both the *Picard algorithm* and *direct algorithm* converge in practice too. However, the number of basis functions, even though we use a sparse approximation, is still quite important which prevents us from dealing effectively with high-dimensional PDE. In particular, the number of basis functions used to capture what happens on the boundary of the domain is large. In this section, we use, the so-called "modified hat functions" that allows to get rid of the boundary basis.

2.3.2.1 Definition of the basis functions

The modified hat functions are defined by the following method (which corresponds to equation (2.16) in [43]),

$$\tilde{\phi}^{l,i}(x) := \begin{cases} 1 & \text{if} \quad l = 1 \wedge i = 1 \\ \left\{ \begin{array}{ll} 1 - 2^{l-1} \cdot x & \text{if} \quad x \in [0, 2h_l] \\ 0 & \text{otherwise} \end{array} \right\} & \text{if} \quad l > 1 \wedge i = 1 \\ \left\{ \begin{array}{ll} 2^{l-1} \cdot x + (1-i)/2 & \text{if} \quad x \in [1 - 2h_l, 1] \\ 0 & \text{otherwise} \end{array} \right\} & \text{if} \quad l > 1 \wedge i = 2^l - 1 \\ \phi^{l,i}(x) & \text{otherwise}\,, \end{cases} \tag{2.3.29}$$

The multivariate hat function on $\mathbb{R}^d$ are obtained by a classical tensor-product approach. For these basis functions, we can remove the points on the boundary of the space so that all the components $l_j, j = 1, \cdots, d$, are positive for a multi-index level $\mathbf{l} = (l_1, \ldots, l_d)$ and a multi-index position $\mathbf{i} = (i_1, \ldots, i_d)$,

$$\tilde{\phi}^{\mathbf{l},\mathbf{i}}(x) = \prod_{j=1}^{d} \tilde{\phi}^{l_j,i_j}(x^j). \tag{2.3.30}$$

In this multivariate case, the index set are given by

$$\mathbf{I}_{\mathbf{l}} = \left\{ i \in \mathbb{N}^d \,|\, i_j \in I_{l_j} \text{ for all } 1 \leqslant j \leqslant d \right\} . \tag{2.3.31}$$

Table 2.3 shows the number of points in the sparse grids without boundary. In particular, we observe that it is much less than sparse grids with boundary for the same dimensions and levels, recall Table 2.1.

2.3.2.1.1 The quadratic model We come back to the quadratic model introduced in (2.3.26)-(2.3.27). In this setting, we can test the 100-dimensional version of this model. Let $M = 2000, N = 10, T = 1, a = 1$, the convergence of $\hat{y}_0$ and $\hat{z}_0$, when using the *direct algorithm*, is shown in Figure 2.9 and Figure 2.10: 3819 seconds were spent on this test. The error for $\hat{y}_0$ appears to be less than 0.01. For Z, the true solution is $\bar{z}_0^i = \frac{2\mathcal{W}_0^i}{\mathbb{E}[1+|\mathcal{W}_T|^2]} = 0, \forall i = 1, ..., d$. The gain in computational time is important in comparison with the pre-wavelet specification of the last section. Not only less basis functions are used, but one should also note that the computational cost of a hat function is less than a pre-wavelet function up to a factor 5. We do also test the *Picard algorithm* in 25-dimensional setting. We set $M = 1500, N = 10, T = 1, a = 1, P = 3$ and get $\hat{y}_0 \approx 2.5481$ quite quickly, see

dimensions \ levels	$l \leqslant 3$	$l \leqslant 4$	$l \leqslant 5$
d=2	17	49	129
d=4	49	209	769
d=5	71	351	1471
d=10	241	2001	13441
d=20	881	13201	154881
d=25	1351	24751	352351
d=50	5201	182001	4867201
d=100	20401	1394001	\

Table 2.3 – The number of points in the sparse grid approximation without boundary functions for different dimensions and levels.

Figure 2.11 (all the initial value of $\mathfrak{z}^{n,k}, 1 \leqslant n \leqslant N-1, 1 \leqslant k \leqslant K_n^z$ are set to 0 in this test).

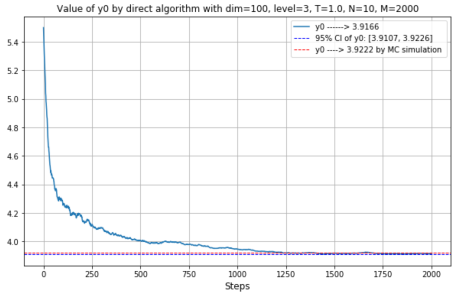

Figure 2.9 – $\hat{y}_0$ for the quadratic model with d=100 and T=1

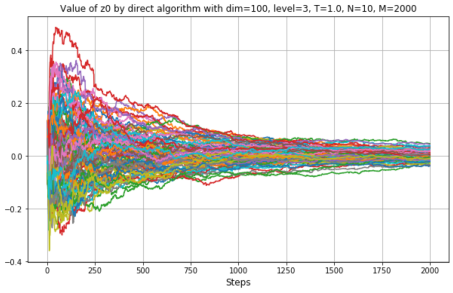

Figure 2.10 – $\hat{z}_0^i$ for the quadratic model with d=100 and T=1

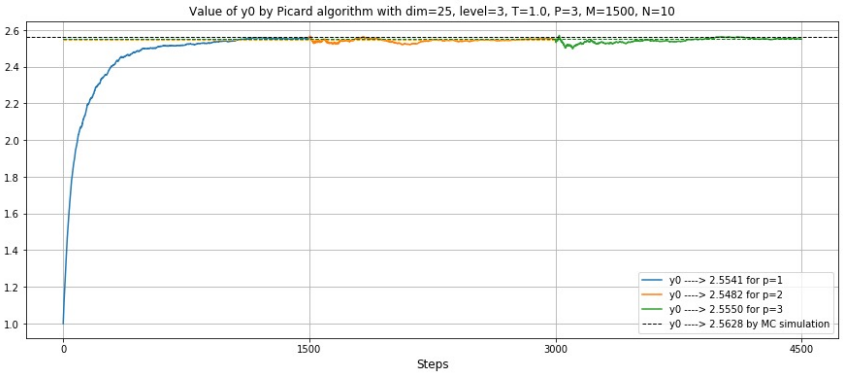

Figure 2.11 – $\hat{y}_0$ for the quadratic model by *Picard algorithm* with d=25, T=1, P=3, N=10, M=1500

2.3.2.1.2 The financial model of (2.3.28) For the *direct algorithm* in this example, we set the parameters $N = 10, M = 5000, \mu = 0.06, \sigma = 0.2, R^l =$

$0.04, R^b = 0.06$, $K_1 = 110$, $K_2 = 130$, $T = 0.5$. Table 2.4 compares the results for the *direct algorithm* and *deep BSDEs solver*: the approximated value for y_0 obtained by the two methods are very close. The running time for the *deep BSDEs solver* shows almost no increase up to $d \leqslant 25$. For the *direct algorithm*, it does increase with the dimensions but it stays reasonable. Actually, it is even competitive when $d \leqslant 25$[3].

dimensions	direct SGD algo with Sparse grids			Deep BSDEs solver[36]		
	y_0	95% CI of y_0	time	y_0	95% CI of y_0	time
d=5	8.0966	[8.0226, 8.1705]	3 s	8.1010	[8.0747, 8.1273]	115 s
d=10	10.9865	[10.9224, 11.0506]	12 s	10.9216	[10.8944, 10.9489]	120 s
d=15	11.848	[11.7853, 11.9107]	33 s	11.8226	[11.7750, 11.8702]	122 s
d=20	11.8674	[11.7962, 11.9387]	61 s	11.9508	[11.8965, 12.0051]	127 s
d=25	11.7801	[11.6467, 11.9135]	130 s	11.6416	[11.5316, 11.7517]	132 s

Table 2.4 – Comparison of the *direct algorithm* and the *deep learning algorithm*.

2.3.2.1.3 A challenging example We now consider a model with an unbounded and complex structure solution, which has been analyzed in [65]. The value function in this case is given by:

$$u(t,x) = \frac{T-t}{d}\sum_{i=1}^{d}(\sin(x_i)\mathbb{1}_{\{x_i<0\}} + x_i\mathbb{1}_{\{x_i\geqslant 0\}}) + \cos\left(\sum_{i=1}^{d} i x_i\right), \quad x \in \mathbb{R}^d. \quad (2.3.32)$$

It corresponds to a BSDE, with underlying process given by $\mathcal{X}_t = x + \frac{1}{\sqrt{d}}\mathrm{I}_d\mathcal{W}_t$, and $x_0 = 0.5\mathbb{1}_d$ and driver and terminal condition given respectively by

$$\begin{aligned} f(t,x,y,z) &= \left(1 + (T-t)(\frac{1}{2d} - C)\right)A(x) + (1-(T-t)C)B(x) + Cy, \\ &= \left(1 + \frac{T-t}{2d}\right)A(x) + B(x) + C\cos\left(\sum_{i=1}^{d} i x_i\right), \quad x \in \mathbb{R}^d,\ y \in \mathbb{R},\ z \in \mathbb{R}^d, \\ g(x) &= u(T,x) = \cos\left(\sum_{i=1}^{d} i x_i\right), \quad x \in \mathbb{R}^d, \end{aligned}$$

where

$$A(x) = \frac{1}{d}\sum_{i=1}^{d}\sin(x_i)\mathbb{1}_{\{x_i<0\}}, B(x) = \frac{1}{d}\sum_{i=1}^{d} x_i\mathbb{1}_{\{x_i\geqslant 0\}}, C = \frac{(d+1)(2d+1)}{12}.$$

[3] The numerical experiments were realised by C++ 17 on a MacBook Pro 6-core Intel Core i7, using only one core and compiling with optimisation flag '-O3' in gcc. The *deep BSDEs solver* [36], using *Tensorflow*, spends most of the time to build the graph for the NN and initialize the variables when the dimension is small then the learning phase is quick. On the contrary, our algorithm builds the approximation grid space quite efficiently (less than 1 second when $d \leqslant 100, level \leqslant 3$) and then the runtime is spent on the the SG algorithm.

In Table 2.5, we compare the approximation of y_0 by using five different algorithms to the theoretical solution. When the dimension $d \leqslant 2$, all the algorithms perform well. However, as already mentioned in [65] the *deep learning algorithm* [55] fails when $d \geqslant 3$ (no matter the chosen initial learning rate and the activation function for the hidden layers, among the tanh, ELU, ReLu and sigmoid ones; besides, taking 3 or 4 hidden layers does not improve the results.) The two deep learning schemes of [65] and our algorithms with sparse grids still works well when $d \leqslant 8$. Figure 2.12 shows the performance of the *direct algorithm* , $\hat{y}_0$ converges to 1.1745 when $d = 8$, it is close to the theoretical solution 1.1603, and the 95% confidence interval of $\hat{y}_0$ is $[1.1611, 1.1881]$. When the dimension $d = 10$, all the algorithms failed at providing correct estimates of the solution as shown in the table, but the errors of our algorithms appear to be smaller than the errors of deep learning methods. Figure 2.13 and Figure 2.14 show the performance of the *direct algorithm*, *Picard algorithm* respectively.

dimensions	Theoretical solution	SGD algo with L^2 sparse grids and hat functions		DL scheme of HPW [65]		DL scheme of HJE [55]
		direct algo	*Picard algo*	DBDP1	DBDP2	
d=1	1.3776	1.3790	1.3825	1.3720	1.3736	1.3724
d=2	0.5707	0.5795	0.5794	0.5715	0.5708	0.5715
d=5	0.8466	0.8734	0.8606	0.8666	0.8365	NC
d=8	1.1603	1.1745	1.1801	1.1694	1.0758	NC
d=10	-0.2149	-0.2439	-0.2594	-0.3105	-0.3961	NC

Table 2.5 – Comparison of the value of $\hat{y}_0$ by different methods when $T = 1$.

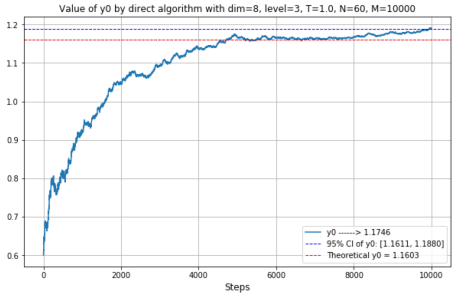

Figure 2.12 – $\hat{y}_0 \rightarrow 1.1745$ by direct SGD algorithm when d=8, N=60, M=10000.

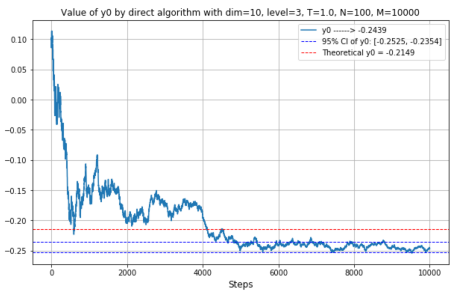

Figure 2.13 – $\hat{y}_0 \rightarrow -0.2439$ by direct algorithm when d=10, N=100, M=10000.

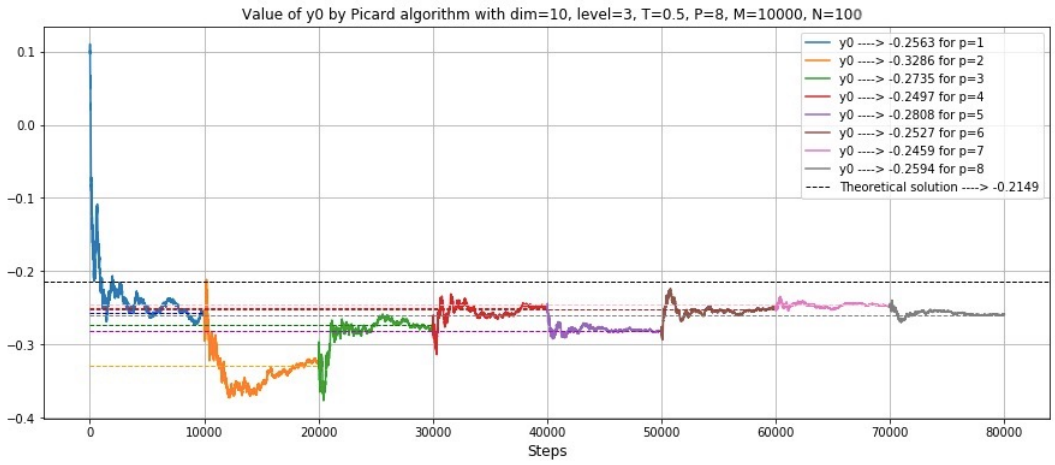

Figure 2.14 – $\hat{y}_0 \to -0.2594$ by picard SGD algorithm when d=10, P=8, N=100, M=10000.

2.4 Study of the discrete optimization problems

In this section, we study theoretically the *direct algorithm* and the *Picard algorithm* in order to prove the results stated in Section 2.2 and 2.3.1.2. We first obtain forward and backward estimates on perturbed BSDEs. This allows in particular to derive the wellposedness of the *direct algorithm*. However, most of the work concentrates on the *Picard algorithm* in Section 2.4.3. A careful study of the iterated SGD algorithms allows to prove the convergence announced in Theorem 2.2.1. Finally, we prove Theorem 2.3.1 concerning the complexity of the method in the case of periodic coefficients and using pre-wavelet basis.

2.4.1 Preliminary estimates

In this subsection, we prove general technical estimates for the backward component of a BSDE that will be used in the proof of the convergence of the numerical methods under study. We will essentially compare two processes with dynamics given by (2.2.11) but taken at two different starting points and controlled processes.
The first process, denoted by V, is a scheme built with a random driver F satisfying:

Assumption 2.4.1

1. *For all $(y,z) \in \mathbb{R} \times \mathbb{R}^d$, $F(\cdot, y, z)$ is progressively measurable.*

2. *There exists some deterministic constant $C \geqslant 0$ such that for any $t \in [0,T]$ and any $(y, y', z, z') \in \mathbb{R}^2 \times \mathbb{R}^{2d}$*

$$|F(t,y,z) - F(t,y',z')| \leqslant C\left(|y-y'| + |z-z'|\right).$$

For $Z \in \mathcal{S}_d^2$ and $\zeta \in \mathcal{L}^2(\mathcal{F}_0)$, we thus define

$$V_t^{\zeta,Z} = \zeta - \int_0^t F(\bar{s}, V_{\bar{s}}^{\zeta,Z}, Z_{\bar{s}})\,\mathrm{d}s + \int_0^t Z_{\bar{s}}\,\mathrm{d}W_s, \tag{2.4.1}$$

where we introduced the notation $\bar{s} := t_n$ for $t_n \leqslant s < t_{n+1}$. The second one, denoted $\tilde{V}$, corresponds to the the true solution to the BSDE

$$\tilde{V}_t^{\zeta,Z} = \zeta - \int_0^t \tilde{F}(s, \tilde{V}_s^{\zeta,Z}, Z_s)\,\mathrm{d}s + \int_0^t Z_s\,\mathrm{d}W_s\,, \tag{2.4.2}$$

where $\tilde{F}$ satisfies the same assumptions as F above.

Proposition 2.4.1 *Let Assumption 2.4.1 hold for F and $\tilde{F}$. For $(\zeta,\zeta') \in \mathcal{L}^2(\mathcal{F}_0) \times \mathcal{L}^2(\mathcal{F}_0)$ and $(Z,Z') \in \mathcal{S}_d^2 \times \mathcal{S}_d^2$, we consider $V^{\zeta,Z}$ and $\tilde{V}^{\zeta',Z'}$ as defined in (2.4.1)-(2.4.2) and we set $\delta F := \tilde{F}(\cdot, V^{\zeta,Z}, Z) - F(\cdot, V^{\zeta,Z}, Z)$, $\eta_s^f = \tilde{F}(s, \tilde{V}_s^{\zeta',Z'}, Z'_s) - \tilde{F}(s, \tilde{V}_{\bar{s}}^{\zeta',Z'}, Z'_{\bar{s}})$ and $\eta_s^z = Z'_s - Z'_{\bar{s}}$. Then, under the above assumptions on F and $\tilde{F}$, it holds*

1. *Forward estimate:*

$$\mathbb{E}\left[\sup_{t\in[0,T]} |\tilde{V}_t^{\zeta',Z'} - V_t^{\zeta,Z}|^2\right] \leqslant C\left(\mathbb{E}\left[|\zeta-\zeta'|^2 + h\sum_{i=0}^{N-1} |Z_{t_n} - Z'_{t_n}|^2\right] + \mathbb{E}\left[\int_0^T (|\delta F_{\bar{s}}|^2 + |\eta_s^f|^2 + |\eta_s^z|^2)\,\mathrm{d}s\right]\right).$$

2. *Backward estimate:*

$$\mathbb{E}\left[\sup_{t\in[0,T]} |\tilde{V}_t^{\zeta',Z'} - V_t^{\zeta,Z}|^2 + h\sum_{i=0}^{N-1} |Z_{t_n} - Z'_{t_n}|^2\right] \leqslant C\mathbb{E}\Big[|\tilde{V}_T^{\zeta',Z'} - V_T^{\zeta,Z}|^2 + \int_0^T (|\delta F_{\bar{s}}|^2 + |\eta_s^f|^2 + |\eta_s^z|^2)\,\mathrm{d}s\Big].$$

Proof.
1. Denote $\Delta V := \tilde{V}^{\zeta',Z'} - V^{\zeta,Z}$, $\Delta Z = Z' - Z$, $\Delta F = \tilde{F}(\cdot, \tilde{V}^{\zeta',Z'}, Z') - \tilde{F}(\cdot, V^{\zeta,Z}, Z)$ and $\Delta\Gamma_s = \Delta Z_{\bar{s}} + \eta_s^z$. Applying Itô's formula, we compute

$$|\Delta V_t|^2 = |\Delta V_0|^2 + \int_0^t \{-2\Delta V_s(\Delta F_{\bar{s}} + \delta F_{\bar{s}} + \eta_s^f) + |\Delta\Gamma_s|^2\}\,\mathrm{d}s + 2\int_0^t \Delta V_s \Delta\Gamma_s\,\mathrm{d}W_s\,. \tag{2.4.3}$$

Since $\tilde{F}$ is Lipschitz-continuous, we have

$$2|\Delta V_s(\Delta F_{\bar{s}} + \delta F_{\bar{s}} + \eta_s^f)| \leqslant (4+L)\sup_{0\leqslant r\leqslant s} |\Delta V_r|^2 + L|\Delta Z_{\bar{s}}|^2 + |\delta F_{\bar{s}}|^2 + |\eta_s^f|^2$$

which combined with (2.4.3) leads to

$$\mathbb{E}\left[\sup_{0\leqslant s\leqslant t} |\Delta V_s|^2\right] \leqslant \mathbb{E}\left[|\Delta V_0|^2 + C\int_0^t \{\sup_{0\leqslant r\leqslant s} |\Delta V_r|^2 + |\Delta Z_{\bar{s}}|^2 + |\delta F_{\bar{s}}|^2 + |\eta_s^f|^2 + |\eta_s^z|^2\}\,\mathrm{d}s\right] + 2\,\mathbb{E}\left[\sup_{0\leqslant r\leqslant t} |\int_0^r \Delta V_s(\Delta Z_{\bar{s}} + \eta_s^z)\,\mathrm{d}W_s|\right]. \tag{2.4.4}$$

Applying the Burkholder-Davis-Gundy inequality, we obtain

$$\mathbb{E}\left[\sup_{r\in[0,t]}\left|\int_0^r \Delta V_s(\Delta Z_{\bar{s}}+\eta_s^z)\,\mathrm{d}W_s\right|\right] \leqslant C\mathbb{E}\left[\left|\int_0^t |\Delta V_s(\Delta Z_{\bar{s}}+\eta_s^z)|^2\,\mathrm{d}s\right|^{\frac{1}{2}}\right]$$
$$\leqslant C\left(\mathbb{E}\left[\sup_{0\leqslant s\leqslant t}|\Delta V_s|^2+\int_0^t(|\Delta Z_{\bar{s}}|^2+|\eta_s^z|^2)\,\mathrm{d}s\right]\right)$$

where we used Young's inequality for the last inequality. Inserting the previous inequality into (2.4.4), we get

$$\mathbb{E}\left[\sup_{0\leqslant r\leqslant t}|\Delta V_t|^2\right] \leqslant |\Delta V_0|^2 + C\int_0^t \mathbb{E}\left[\sup_{0\leqslant r\leqslant s}|\Delta V_r|^2+|\Delta Z_{\bar{s}}|^2+|\eta_s^f|^2+|\delta F_{\bar{s}}|^2+|\eta_s^z|^2\right]\mathrm{d}s\,.$$

The proof for this step is concluded by applying Grönwall's Lemma.

2. From (2.4.3), we compute

$$\mathbb{E}\left[|\Delta V_t|^2+\int_t^T|\Delta\Gamma_s|^2\,\mathrm{d}s\right] \leqslant \mathbb{E}\left[|\Delta V_T|^2+2\int_t^T\Delta V_s(\Delta F_{\bar{s}}+\delta F_{\bar{s}}+\eta_s^f)\,\mathrm{d}s\right].$$

We observe that, since $\tilde{F}$ is Lipschitz continuous,

$$\Delta V_s\Delta F_{\bar{s}} \leqslant C(|\Delta V_s|^2+|\Delta V_{\bar{s}}|^2+|\Delta V_s\Delta Z_{\bar{s}}|).$$

For $\alpha>0$, to be fixed later on, we get using Young's inequality,

$$\Delta V_s\Delta F_{\bar{s}} \leqslant C\left((1+\frac{1}{\alpha})|\Delta V_s|^2+|\Delta V_{\bar{s}}|^2+\alpha|\Delta Z_{\bar{s}}|^2\right),$$
$$\leqslant C\left((1+\frac{1}{\alpha})|\Delta V_s|^2+|\Delta V_{\bar{s}}|^2+\alpha|\Delta\Gamma_s|^2+|\eta_s^z|^2\right).$$

For α small enough, we thus obtain

$$\mathbb{E}\left[|\Delta V_t|^2+\frac{1}{2}\int_t^T|\Delta\Gamma_s|^2\,\mathrm{d}s\right] \tag{2.4.5}$$
$$\leqslant \mathbb{E}\left[|\Delta V_T|^2+C\int_t^T\left(|\Delta V_s|^2+|\Delta V_{\bar{s}}|^2+|\delta F_{\bar{s}}|^2+|\eta_s^f|^2+|\eta_s^z|^2\right)\mathrm{d}s\right]. \tag{2.4.6}$$

Applying Grönwall's Lemma leads to, for all $t\leqslant T$,

$$\mathbb{E}\big[|\Delta V_t|^2\big] \leqslant C\mathbb{E}\left[\mathscr{B}_T+\int_t^T|\Delta V_{\bar{s}}|^2\,\mathrm{d}s\right] \tag{2.4.7}$$

with

$$\mathscr{B}_T := |\Delta V_T|^2+\int_0^T(|\eta_s^f|^2+|\eta_s^z|^2+|\delta F_{\bar{s}}|^2)\,\mathrm{d}s\,.$$

In particular, for $n\leqslant N$ and $t_n\in\pi$, we have

$$\mathbb{E}\big[|\Delta V_{t_n}|^2\big] \leqslant C\mathbb{E}\left[\mathscr{B}_T+h\sum_{j=n}^{N-1}|\Delta V_{t_j}|^2\right]$$

which in turn, using the discrete-time Grönwall Lemma, leads to $\max_{n\leqslant N}\mathbb{E}\big[|\Delta V_{t_n}|^2\big] \leqslant C\mathbb{E}[\mathscr{B}_T]$. Combining this inequality with (2.4.5) and (2.4.7), we obtain

$$\mathbb{E}\left[|\Delta V_t|^2 + \frac{1}{2}\int_t^T |\Delta\Gamma_s|^2\,\mathrm{d}s\right] \leqslant C\mathbb{E}[\mathscr{B}_T]\,,\ t\leqslant T.$$

To conclude the proof one applies the Burkholder-Davis-Gundy inequality as in step 1. □

2.4.2 Application to the *direct algorithm*

We here prove the results announced in Section 2.2.2. We start by proving the analytic expression of the main quantities appearing in the *direct algorithm*, recall Definition 2.2.3.

Proof of Lemma 2.2.2 By standard computations, recall (2.2.6), for any $0\leqslant n\leqslant N-1$,

$$\nabla_{\mathfrak{y}^k} Z^{\mathfrak{u}}_{t_n} = 0, \quad \text{and} \quad \nabla_{\mathfrak{z}^{n,k}}(Z^{\mathfrak{u}}_t)^l = \psi^k_n(X_{t_n})\mathbf{1}_{\{t=t_n\}}\mathbf{e}^l\,.$$

This leads to, for $0\leqslant q\leqslant N-1$,

$$\nabla_{\mathfrak{z}^{n,k}}(Z^{\mathfrak{u}}_{t_q}\cdot\Delta W_q) = \psi^k_n(X_{t_n})\Delta W_n\mathbf{1}_{\{q=n\}} \tag{2.4.8}$$

and

$$\nabla_{\mathfrak{z}^{n,k}} f(Y^{\mathfrak{u}}_{t_q}, Z^{\mathfrak{z}}_{t_q}) = \nabla_y f(Y^{\mathfrak{u}}_{t_q}, Z^{\mathfrak{u}}_{t_q})\nabla_{\mathfrak{z}^{n,k}} Y^{\mathfrak{u}}_{t_q} + \psi^k_n(X_{t_n})\nabla_z f(Y^{\mathfrak{u}}_{t_q}, Z^{\mathfrak{u}}_{t_q})\mathbf{1}_{\{q=n\}}. \tag{2.4.9}$$

Now, differentiating both side of (2.2.10) with respect to the variable $\mathfrak{y}^k, 1\leqslant k\leqslant K^y$ yields

$$\nabla_{\mathfrak{y}^k} Y^{\mathfrak{u}}_{t_n} = \psi^k_y(X_0)\prod_{j=0}^{n-1}\Big(1-h\nabla_y f(Y^{\mathfrak{u}}_{t_j}, Z^{\mathfrak{u}}_{t_j})\Big) \text{ for } n\geqslant 0.$$

From (2.2.10), by differentiation, we obtain $\nabla_{\mathfrak{z}^{n,k}} Y^{\mathfrak{u}}_0 = 0$ and using (2.4.8) and (2.4.9), for $q\geqslant 1$,

$$\begin{aligned}\nabla_{\mathfrak{z}^{n,k}} Y^{\mathfrak{u}}_{t_q} = \nabla_{\mathfrak{z}^{n,k}} Y^{\mathfrak{u}}_{t_{q-1}}&\Big(1-h\nabla_y f(Y^{\mathfrak{u}}_{t_{q-1}}, Z^{\mathfrak{z}}_{t_{q-1}})\Big)\\ &+\psi^k_n(X_{t_n})\mathbf{1}_{\{n=q-1\}}\Big(\Delta W^\top_{q-1} - h\nabla_z f(Y^{\mathfrak{u}}_{t_{q-1}}, Z^{\mathfrak{u}}_{t_{q-1}})\Big)\,,\end{aligned}$$

which in turn yields

$$\nabla_{\mathfrak{z}^{n,k}} Y^{\mathfrak{u}}_{t_q} = 0 \text{ for } q\leqslant n\,,\quad \nabla_{\mathfrak{z}^{n,k}} Y^{\mathfrak{u}}_{t_{n+1}} = \psi^k_n(X_{t_n})\left(\Delta W^\top_n - h\nabla_z f(Y^{\mathfrak{u}}_{t_n}, Z^{\mathfrak{u}}_{t_n})\right)$$

and for $q\geqslant n+2$,

$$\nabla_{\mathfrak{z}^{n,k}} Y^{\mathfrak{u}}_{t_q} = \psi^k_n(X_{t_n})\left(\Delta W_n - h\nabla_z f(Y^{\mathfrak{u}}_{t_n}, Z^{\mathfrak{u}}_{t_n})\right)\prod_{j=n+1}^{q-1}\Big(1-h\nabla_y f(Y^{\mathfrak{u}}_{t_j}, Z^{\mathfrak{u}}_{t_j})\Big)\,.$$

This concludes the proof. □

Recall that, the time discretization error that will appear in our estimates is classically given by

$$\mathcal{E}_\pi = \mathbb{E}\left[\sum_{n=0}^{N-1}\int_{t_n}^{t_{n+1}} \left(|\mathcal{Y}_s - \mathcal{Y}_{t_n}|^2 + |\mathcal{Z}_s - \mathcal{Z}_{t_n}|^2 + |\mathcal{X}_{t_n} - X_{t_n}|^2\right) ds\right]. \tag{2.4.10}$$

The approximation error due to the restriction to the functional space, expressed in (2.2.18), is also given by

$$\mathcal{E}_\psi = \mathbb{E}\left[|u(0,\mathcal{X}_0) - \bar{u}_0(\mathcal{X}_0)|^2 + \sum_{n=1}^{N-1} h|(\sigma^\top \nabla_x u)(t_n, X_{t_n}) - \bar{v}_n(X_{t_n})|^2\right], \tag{2.4.11}$$

where, $\bar{v}_n$ is the $L^2(\mathbb{R}^d, \mathbb{P}_{X_{t_n}})$-projection of the map $(\sigma^\top \nabla_x u)(t_n, \cdot)$ onto $\mathscr{V}_n^z$, $0 \leqslant n \leqslant N-1$ and $\bar{u}_0$ is the $L^2(\mathbb{R}^d, \mathbb{P}_{\mathcal{X}_0})$-projection of the map $u(0,\cdot)$ onto $\mathscr{V}^y$. We denote $\bar{\mathfrak{y}}$ the coefficient associated to the decomposition of $\bar{u}$, namely

$$\mathbb{R}^d \ni x \mapsto \bar{u}_0(x) = \sum_{k=1}^{K^y} \bar{\mathfrak{y}}^k \psi_0^k(x) \in \mathbb{R}, \tag{2.4.12}$$

and also $\bar{\mathfrak{z}}^n$ the coefficient associated to the decomposition of $\bar{v}_n$, namely

$$\mathbb{R}^d \ni x \mapsto \bar{v}_n(x) = \sum_{k=1}^{K_n^z} \bar{\mathfrak{z}}^{n,k} \psi_n^k(x) \in \mathbb{R}^d,\ 0 \leqslant n \leqslant N-1. \tag{2.4.13}$$

For later use, we introduce a *reference solution*

$$\bar{\mathfrak{u}} = (\bar{\mathfrak{y}}, \bar{\mathfrak{z}}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}. \tag{2.4.14}$$

We first discuss the well-posedness of the optimization problem (2.2.12).

Lemma 2.4.1 *Under Assumption 2.2.1, Assumption 2.2.2 (i) and Assumption 2.2.3, it holds*

$$\operatorname*{arg\,min}_{\mathfrak{u}\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}} \mathfrak{g}(\mathfrak{u}) \neq \varnothing.$$

Proof. Let $\mathfrak{u} = (\mathfrak{y}, \mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$. Using the backward estimate of Proposition 2.4.1 with $V^{\zeta,Z} := Y^{\mathfrak{u}}$ and $\tilde{V}^{\zeta',Z'} := Y^0$ ($\zeta' = 0$, $Z' \equiv 0$) yields

$$|||\mathfrak{u}|||^2 = \mathbb{E}\left[|Y_0^{\mathfrak{u}}|^2 + \sum_{n=0}^{N-1} h|Z_{t_n}^{\mathfrak{u}}|^2\right] \leqslant C\mathbb{E}\big[|Y_T^{\mathfrak{u}} - Y_T^0|^2\big] \leqslant C(1 + \mathbb{E}\big[|g(X_T) - Y_T^{\mathfrak{u}}|^2\big]).$$

Under Assumption 2.2.3, we thus deduce that the continuous function $\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z} \ni \mathfrak{u} \mapsto \mathfrak{g}(\mathfrak{u})$ is coercive. As a consequence, it admits a global minimizer so that the optimization problem (2.2.12) is well-posed. □

The following proposition can be seen as a version of the results in [58] (see Theorem 1 & 2) adapted to our context. Let us note that our setting is simpler as we do not deal with fully coupled Forward Backward SDEs.

Proposition 2.4.2 *Under Assumption 2.2.1, Assumption 2.2.2 (i), (ii) and Assumption 2.2.3, there exists a positive constant C such that for any*

$$\mathfrak{u}^\star := (\mathfrak{y}^\star, \mathfrak{z}^\star) \in \underset{\mathfrak{u}\in\mathbb{R}^{K^y}\times\mathbb{R}^{d\bar{K}^z}}{\arg\min}\ \mathfrak{g}(\mathfrak{u}),$$

it holds

$$\mathbb{E}\left[|u(0,\mathcal{X}_0) - Y_0^{\mathfrak{u}^\star}|^2 + h\sum_{n=0}^{N-1}|\mathcal{Z}_{t_n} - Z_{t_n}^{\mathfrak{u}^\star}|^2\right] \leqslant C\left(\mathcal{E}_\pi + \mathcal{E}_\psi\right) \tag{2.4.15}$$

and

$$\mathfrak{g}(\mathfrak{u}^\star) \leqslant \mathfrak{g}(\bar{\mathfrak{u}}) \leqslant C\left(\mathcal{E}_\pi + \mathcal{E}_\psi\right)\ . \tag{2.4.16}$$

Proof. We use the backward estimate of Proposition 2.4.1 with $V^{\zeta,Z} := Y^{\mathfrak{u}^\star}$ and $\tilde{V}^{\zeta',Z} := Y^{\bar{\mathfrak{u}}}$ where $\bar{\mathfrak{u}}$ is given in (2.4.14), to obtain

$$\begin{aligned}\mathbb{E}\left[\sup_{t\in[0,T]}|Y_t^{\mathfrak{u}^\star} - Y_t^{\bar{\mathfrak{u}}}|^2 + h\sum_{n=0}^{N-1}\mathbb{E}\Big[|Z_{t_n}^{\bar{\mathfrak{u}}} - Z_{t_n}^{\mathfrak{u}^\star}|^2\Big]\right] &\leqslant C\mathbb{E}\Big[|Y_T^{\mathfrak{u}^\star} - Y_T^{\bar{\mathfrak{u}}}|^2\Big]\\ &\leqslant C\mathbb{E}\Big[|Y_T^{\mathfrak{u}^\star} - g(X_T^x)|^2 + |g(X_T^x) - Y_T^{\bar{\mathfrak{u}}}|^2\Big].\end{aligned}$$

By optimality of $\mathfrak{u}^\star$, we get

$$\mathbb{E}\left[\sup_{t\in[0,T]}|Y_t^{\mathfrak{u}^\star} - Y_t^{\bar{\mathfrak{u}}}|^2 + h\sum_{n=0}^{N-1}\mathbb{E}\Big[|Z_{t_n}^{\bar{\mathfrak{u}}} - Z_{t_n}^{\mathfrak{u}^\star}|^2\Big]\right] \leqslant C\mathbb{E}\big[|g(X_T^x) - Y_T^{\bar{\mathfrak{u}}}|^2\big]\ . \tag{2.4.17}$$

We now use the forward estimate of Proposition 2.4.1 with $V^{\zeta,Z} := Y^{\bar{\mathfrak{u}}}$ and $\tilde{V}^{\zeta',Z} := \mathcal{Y}$. We obtain

$$\begin{aligned}\mathfrak{g}(\bar{\mathfrak{u}}) &= \mathbb{E}\big[|g(X_T^x) - Y_T^{\bar{\mathfrak{u}}}|^2\big]\\ &\leqslant C\mathbb{E}\left[|\mathcal{Y}_0 - Y_0^{\bar{\mathfrak{u}}}|^2 + \int_0^T\left(|\mathcal{Y}_s - \mathcal{Y}_{\bar{s}}|^2 + |\mathcal{Z}_s - \mathcal{Z}_{\bar{s}}|^2\right)\mathrm{d}s + \sum_{n=0}^{N-1}h|\mathcal{Z}_{t_n} - Z_{t_n}^{\bar{\mathfrak{u}}}|^2\right].\end{aligned} \tag{2.4.18}$$

Observe now that in our current smooth coefficients framework $\mathcal{Z}_{t_n} = (\sigma^\top\nabla_x u)(t_n,\mathcal{X}_{t_n})$ so that one has

$$\begin{aligned}&\mathbb{E}\big[|\mathcal{Z}_{t_n} - Z_{t_n}^{\bar{\mathfrak{u}}}|^2\big]\\ &= \mathbb{E}\Big[|(\sigma^\top\nabla_x u)(t_n,\mathcal{X}_{t_n}) - Z_{t_n}^{\bar{\mathfrak{u}}}|^2\Big]\\ &\leqslant 2\Big(\mathbb{E}\Big[|(\sigma^\top\nabla_x u)(t_n,\mathcal{X}_{t_n}) - (\sigma^\top\nabla_x u)(t_n,X_{t_n})|^2\Big] + \mathbb{E}\Big[|(\sigma^\top\nabla_x u)(t_n,X_{t_n}) - Z_{t_n}^{\bar{\mathfrak{u}}}|^2\Big]\Big)\\ &\leqslant C\Big(\mathbb{E}\Big[|\mathcal{X}_{t_n} - X_{t_n}|^2\Big] + \mathbb{E}\Big[|(\sigma^\top\nabla_x u)(t_n,X_{t_n}^{x_0}) - Z_{t_n}^{\bar{\mathfrak{u}}}|^2\Big]\Big)\end{aligned} \tag{2.4.19}$$

where we used the Lipschitz regularity of $x\mapsto(\sigma^\top\nabla_x u)(t_n,x)$ uniformly with respect to the variable t_n. Combining (2.4.18) and (2.4.19) leads to

$$\mathfrak{g}(\bar{\mathfrak{u}}) \leqslant C\left(\mathcal{E}_\pi + \mathcal{E}_\psi\right)$$

which, since $\mathfrak{g}(\mathfrak{u}^\star) \leqslant \mathfrak{g}(\bar{\mathfrak{u}})$, proves (2.4.16).

The above estimate together with (2.4.17) leads to

$$\mathbb{E}\left[\sup_{t\in[0,T]}|Y_t^{\mathfrak{u}^\star}-Y_t^{\bar{\mathfrak{u}}}|^2+h\sum_{n=0}^{N-1}\mathbb{E}\Big[|Z_{t_n}^{\bar{\mathfrak{u}}}-Z_{t_n}^{\mathfrak{u}^\star}|^2\Big]\right]\leqslant C\left(\mathcal{E}_\pi+\mathcal{E}_\psi\right).$$

Then, using the inequalities $|\mathcal{Z}_{t_n}-Z_{t_n}^{\mathfrak{u}^\star}|^2\leqslant 2|Z_{t_n}^{\mathfrak{u}^\star}-Z_{t_n}^{\bar{\mathfrak{u}}}|^2+2|\mathcal{Z}_{t_n}-Z_{t_n}^{\bar{\mathfrak{u}}}|^2$, $0\leqslant n\leqslant N-1$, and $|Y_t^{\mathfrak{u}^\star}-\mathcal{Y}_t|^2\leqslant 2|Y_t^{\mathfrak{u}^\star}-Y_t^{\bar{\mathfrak{u}}}|^2+2|\mathcal{Y}_t-Y_t^{\bar{\mathfrak{u}}}|^2$ yields (2.4.15) and concludes the proof. □

2.4.3 Study of the *Picard algorithm*

We introduce the following mean squared error:

$$\mathcal{E}_p:=\mathbb{E}\big[|||\mathfrak{u}_M^p-\bar{\mathfrak{u}}|||^2\big]\,,0\leqslant p\leqslant P\,, \tag{2.4.20}$$

where the sequence $(\mathfrak{u}_M^p)_{0\leqslant p\leqslant P}$ is given by Definition 2.2.7, $|||\cdot|||$ is given by Definition 2.2.5 and $\bar{\mathfrak{u}}$ is the reference solution introduced in (2.4.14). In this subsection, our aim is to establish an upper bound for the quantity $\mathcal{E}_P$ that will allow us to prove Theorem 2.2.1.

2.4.3.1 Preliminary estimates

Proposition 2.4.3 *Suppose that Assumption 2.2.1, Assumption 2.2.2 (i), (ii) and Assumption 2.2.3 hold. If $T(1+2L^2(1+h))<1$ and $\delta_h:=\frac{8L^2T}{1-T(1+2L^2(1+h))}<1$, then for any $\varepsilon>0$ such that $\delta_{h,\varepsilon}:=\delta_h(1+\varepsilon)<1$ there exists a positive constant C_ε such that for any positive integer P*

$$\mathcal{E}_P\leqslant\delta_{h,\varepsilon}^P\mathcal{E}_0+C_\varepsilon\big(\mathcal{E}_{\mathrm{RM}}+\mathcal{E}_\psi+\mathcal{E}_\pi\big) \tag{2.4.21}$$

with the notation

$$\mathcal{E}_{\mathrm{RM}}:=\max_{1\leqslant p\leqslant P}\mathbb{E}\Big[|||\mathfrak{u}_M^p-\Phi(\mathfrak{u}_M^{p-1})|||^2\Big]. \tag{2.4.22}$$

Proof. From the decomposition,

$$\mathfrak{u}_M^p-\bar{\mathfrak{u}}=\Phi_M(\mathfrak{u}_M^{p-1})-\Phi(\mathfrak{u}_M^{p-1})+\Phi(\mathfrak{u}_M^{p-1})-\bar{\mathfrak{u}}$$

we obtain, for any $\varepsilon>0$,

$$\mathcal{E}_p\leqslant(1+\frac{1}{\varepsilon})\mathcal{E}_{\mathrm{RM}}+(1+\varepsilon)\mathbb{E}\Big[|||\Phi(\mathfrak{u}_M^{p-1})-\bar{\mathfrak{u}}|||^2\Big].$$

Then, using Lemma 2.4.2 below, we get

$$\mathcal{E}_p\leqslant\delta_{h,\varepsilon}\mathcal{E}_{p-1}+C_\varepsilon\left(\mathcal{E}_\psi+\mathcal{E}_\pi+\mathcal{E}_{\mathrm{RM}}\right)$$

up to a modification of ε. By an induction argument, we derive

$$\mathcal{E}_P\leqslant\delta_{h,\varepsilon}^P\mathcal{E}_0+C_\varepsilon\big(\mathcal{E}_{\mathrm{RM}}+\mathcal{E}_\psi+\mathcal{E}_\pi\big)$$

which concludes the proof. □

Lemma 2.4.2 *Suppose that Assumption 2.2.1, Assumption 2.2.2 (i), (ii) and Assumption 2.2.3 hold. If* $T(1+2L^2(1+h)) < 1$ *and* $\delta_h := \frac{8L^2T}{1-T(1+2L^2(1+h))} < 1$*, then, for any* $\varepsilon > 0$ *there exists a positive constant* C_ε *(*$\varepsilon \mapsto C_\varepsilon$ *being non-increasing) such that for any* $\tilde{\mathfrak{u}} \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$ *it holds*

$$|||\Phi(\tilde{\mathfrak{u}}) - \bar{\mathfrak{u}}|||^2 \leqslant \delta_h(1+\varepsilon)|||\tilde{\mathfrak{u}} - \bar{\mathfrak{u}}|||^2 + C_\varepsilon\left(\mathcal{E}_\psi + \mathcal{E}_\pi\right).$$

Proof.

Step 1: We denote $\check{\mathfrak{u}} = \Phi(\tilde{\mathfrak{u}})$ where $\check{\mathfrak{u}} = (\check{\mathfrak{y}}, \check{\mathfrak{z}})$ and $\tilde{\mathfrak{u}} = (\tilde{\mathfrak{y}}, \tilde{\mathfrak{z}})$ belongs to $\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$. We first observe that, recalling (2.2.21), (2.2.30) and (2.2.23),

$$\begin{aligned}|||\Phi(\tilde{\mathfrak{u}}) - \bar{\mathfrak{u}}|||^2 &= \mathbb{E}\Big[|Y_0^{\bar{\mathfrak{u}}} - Y_0^{\check{\mathfrak{u}}}|^2 + \int_0^T |Z_t^{\check{\mathfrak{u}}} - Z_t^{\bar{\mathfrak{u}}}|^2\,\mathrm{d}t\Big]\\ &= \mathbb{E}\Big[|U_T^{\tilde{\mathfrak{u}},\check{\mathfrak{u}}} - U_T^{\tilde{\mathfrak{u}},\bar{\mathfrak{u}}}|^2\Big].\end{aligned}$$

Moreover, by optimality of $\check{\mathfrak{u}}$

$$\begin{aligned}\mathbb{E}\Big[|U_T^{\tilde{\mathfrak{u}},\check{\mathfrak{u}}} - U_T^{\tilde{\mathfrak{u}},\bar{\mathfrak{u}}}|^2\Big] &\leqslant 2\left(\mathbb{E}\Big[|U_T^{\tilde{\mathfrak{u}},\check{\mathfrak{u}}} - g(X_T)|^2\Big] + \mathbb{E}\Big[|g(X_T) - U_T^{\tilde{\mathfrak{u}},\bar{\mathfrak{u}}}|^2\Big]\right)\\ &\leqslant 4\mathbb{E}\Big[|g(X_T) - U_T^{\tilde{\mathfrak{u}},\bar{\mathfrak{u}}}|^2\Big].\end{aligned}$$

We now compute, for any $\varepsilon > 0$,

$$\mathbb{E}\Big[|g(X_T) - U_T^{\tilde{\mathfrak{u}},\bar{\mathfrak{u}}}|^2\Big] \leqslant (1+\frac{1}{\varepsilon})\mathbb{E}\Big[|g(X_T) - U_T^{\bar{\mathfrak{u}},\bar{\mathfrak{u}}}|^2\Big] + (1+\varepsilon)\mathbb{E}\Big[|U_T^{\bar{\mathfrak{u}},\bar{\mathfrak{u}}} - U_T^{\tilde{\mathfrak{u}},\bar{\mathfrak{u}}}|^2\Big],$$

which, combined with the previous inequality, yields

$$|||\Phi(\tilde{\mathfrak{u}}) - \bar{\mathfrak{u}}|||^2 \leqslant 4(1+\frac{1}{\varepsilon})\mathbb{E}\Big[|g(X_T) - U_T^{\bar{\mathfrak{u}},\bar{\mathfrak{u}}}|^2\Big] + 4(1+\varepsilon)\mathbb{E}\Big[|U_T^{\bar{\mathfrak{u}},\bar{\mathfrak{u}}} - U_T^{\tilde{\mathfrak{u}},\bar{\mathfrak{u}}}|^2\Big]. \quad (2.4.23)$$

Since $U^{\bar{\mathfrak{u}},\bar{\mathfrak{u}}} = Y^{\bar{\mathfrak{u}}}$, we can give an upper bound for the first term appearing on the right-hand side of the above inequality by using (2.4.16),

$$\mathbb{E}\Big[|g(X_T) - U_T^{\bar{\mathfrak{u}},\bar{\mathfrak{u}}}|^2\Big] = \mathfrak{g}(\bar{\mathfrak{u}}) \leqslant C\left(\mathcal{E}_\psi + \mathcal{E}_\pi\right). \quad (2.4.24)$$

Step 2: We now turn to the study of the second term appearing on the right hand side of (2.4.23). Recalling the dynamics (2.2.24), denoting $\delta U := U^{\tilde{\mathfrak{u}},\bar{\mathfrak{u}}} - Y^{\bar{\mathfrak{u}}}$, $\delta Z = Z^{\tilde{\mathfrak{u}}} - Z^{\bar{\mathfrak{u}}}$, $\delta Y = Y^{\tilde{\mathfrak{u}}} - Y^{\bar{\mathfrak{u}}}$ and $\delta f_{t_n} = f(Y_{t_n}^{\tilde{\mathfrak{u}}}, Z_{t_n}^{\tilde{\mathfrak{u}}}) - f(Y_{t_n}^{\bar{\mathfrak{u}}}, Z_{t_n}^{\bar{\mathfrak{u}}})$, $0 \leqslant n \leqslant N-1$, using the Cauchy-Schwarz inequality and the Lipschitz-regularity of the map f, we get

$$\mathbb{E}\big[|\delta U_T|^2\big] \leqslant T\int_0^T \mathbb{E}\big[|\delta f_{\bar{s}}|^2\big]\,\mathrm{d}s \leqslant 2L^2T\sum_{n=0}^{N-1} h\mathbb{E}\big[|\delta Y_{t_n}|^2 + |\delta Z_{t_n}|^2\big]\,\mathrm{d}s. \quad (2.4.25)$$

For all $0 \leqslant n \leqslant N-1$, one has

$$\delta Y_{t_{n+1}} = \delta Y_{t_n} - h\delta f_{t_n} + \delta Z_{t_n}\Delta W_n$$

which in turn, setting $\Delta M_n := 2(\delta Y_{t_n} - h\delta f_{t_n})\delta Z_{t_n}\Delta W_n$, yields

$$|\delta Y_{t_{n+1}}|^2 = |\delta Y_{t_n}|^2 - 2h\delta Y_{t_n}\delta f_{t_n} + h^2|\delta f_{t_n}|^2 + |\delta Z_{t_n}\Delta W_n|^2 + \Delta M_n.$$

Using the fact that $\mathbb{E}[\Delta M_n] = 0$ and the Lipschitz regularity of the map f, we deduce

$$\begin{aligned}\mathbb{E}\big[|\delta Y_{t_{n+1}}|^2\big] &\leqslant \mathbb{E}\big[|\delta Y_{t_n}|^2 + h|\delta Y_{t_n}|^2 + (h+h^2)|\delta f_{t_n}|^2 + h|\delta Z_{t_n}|^2\big] \\ &\leqslant \mathbb{E}\big[|\delta Y_{t_n}|^2 + (h + 2L^2(h+h^2))(|\delta Y_{t_n}|^2 + |\delta Z_{t_n}|^2)\big].\end{aligned}$$

Summing the previous inequality, we obtain

$$\mathbb{E}\big[|\delta Y_{t_n}|^2\big] \leqslant |\delta Y_0|^2 + \mathbb{E}\left[\sum_{j=0}^{N-1}(h + 2L^2(h+h^2))(|\delta Y_{t_j}|^2 + |\delta Z_{t_j}|^2)\right]$$

so that, multiplying both side of the previous inequality by h and summing again, we get

$$\begin{aligned}\sum_{n=0}^{N-1} h\mathbb{E}\big[|\delta Y_{t_n}|^2\big] &\leqslant T|\delta Y_0|^2 + T(1+2L^2(1+h))\left(\mathbb{E}\Big[\sum_{j=0}^{N-1} h|\delta Y_{t_j}|^2\Big] + \mathbb{E}\Big[\sum_{j=0}^{N-1} h|\delta Z_{t_j}|^2\Big]\right) \\ &\leqslant \frac{T}{1-T(1+2L^2(1+h))}\left(|\delta Y_0|^2 + (1+2L^2(1+h))\mathbb{E}\Big[\sum_{j=0}^{N-1} h|\delta Z_{t_j}|^2\Big]\right)\end{aligned}$$

where, for the last inequality, we used the fact that $T(1+2L^2(1+h)) < 1$. Combining the previous inequality with (2.4.25), we obtain

$$\mathbb{E}\big[|\delta U_T|^2\big] \leqslant \frac{2L^2T}{1-T(1+2L^2(1+h))}\left(T|\delta Y_0|^2 + \sum_{n=0}^{N-1} h\mathbb{E}\big[|\delta Z_{t_n}|^2\big]\right). \tag{2.4.26}$$

We finally complete the proof by combining (2.4.23), (2.4.24) and (2.4.26). □

2.4.3.2 Study of the approximation error of the stochastic gradient descent algorithm

In this subsection, our aim is to study the approximation error of Φ by Φ_M where Φ_M has been introduced in Definition 2.2.6.

Lemma 2.4.3 *Suppose that Assumption 2.2.1, Assumption 2.2.2 (i) and Assumption 2.2.3 hold. Let $\tilde{\mathfrak{u}}$ be a fixed $\mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$-valued random vector and set $\check{\mathfrak{u}} = (\check{\mathfrak{y}}, \check{\mathfrak{z}}) := \Phi(\tilde{\mathfrak{u}})$ and $\mathfrak{u}_M = (\mathfrak{y}_M, \mathfrak{z}_M) := \Phi_M(\mathfrak{u}_0, \mathfrak{X}_0, \mathfrak{W}, \tilde{\mathfrak{u}})$, M being a positive integer and where $(\mathfrak{u}_0, \mathfrak{X}_0, \mathfrak{W})$ (recall Definition 2.2.6) is independent of $\tilde{\mathfrak{u}}$. We denote by $\mathbb{E}_{\tilde{\mathfrak{u}}}[\cdot]$, the conditional expectation with respect to the sigma-field $\sigma(\tilde{\mathfrak{u}})$ generated by $\tilde{\mathfrak{u}}$. Then, for any positive integer M, the random vector $(\mathfrak{y}_M, \mathfrak{z}_M)$ satisfies:*

$$\mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|\mathfrak{y}_M - \check{\mathfrak{y}}|^2\Big] \leqslant L_{K,M}(1+|\check{\mathfrak{y}}|^2) \quad \text{and} \quad \mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|\mathfrak{z}_{M,l}^{n,\cdot} - \check{\mathfrak{z}}_l^{n,\cdot}|^2\Big] \leqslant L_{K,M}\left(\frac{1}{h} + |\check{\mathfrak{z}}_l^{n,\cdot}|^2\right) \tag{2.4.27}$$

for any $l \in \{1, \cdots, d\}$, *with*

$$L_{K,M} := \varrho_0 \varrho_1 \Big(1 + \mathbb{E}\Big[|\mathfrak{u}_0|^2\Big]\Big) \sum_{m=1}^{M} \exp\left(-4\frac{\alpha_K}{\beta_K}(\Gamma_M - \Gamma_m)\right)\gamma_m^2 \tag{2.4.28}$$

$$\Gamma_m := \sum_{k=1}^{m} \gamma_k \ , \ m \geqslant 1 \, , \tag{2.4.29}$$

where the constants ϱ_0, ϱ_1 *are defined respectively in equation* (2.4.32) *and* (2.4.36) *below.*
Moreover, it holds

$$\mathbb{E}_{\tilde{\mathfrak{u}}}\Big[\|\!|\Phi_M(\tilde{\mathfrak{u}}) - \Phi(\tilde{\mathfrak{u}})|\!\|^2\Big] \leqslant \kappa_K L_{K,M}\,(1 + dN) + \frac{\kappa_K}{\alpha_K} L_{K,M} \|\!|\Phi(\tilde{\mathfrak{u}})|\!\|^2 \, . \tag{2.4.30}$$

Proof.
Step 1: We prove the estimate for the difference $\mathfrak{z}_{M,l}^{n,\cdot} - \check{\mathfrak{z}}_l^{n,\cdot}$. The proof for $\mathfrak{y}_M - \check{\mathfrak{y}}$ follows from similar arguments and we omit some technical details. From (2.2.41), one gets

$$\begin{aligned} |H^{n,l}(\mathcal{X}_0, W, \tilde{\mathfrak{u}}, \mathfrak{z}_l^{n,\cdot})|^2 &= \frac{4}{(\beta_K\sqrt{h})^2}|\mathfrak{G}^{\tilde{\mathfrak{u}}} - \omega_l^{n,\cdot} \cdot \mathfrak{z}_l^{n,\cdot}|^2 |\omega_l^{n,\cdot}|^2 \\ &\leqslant \frac{8}{\beta_K^2}(|\mathfrak{G}^{\tilde{\mathfrak{u}}}|^2 + h|\tilde{\omega}_l^{n,\cdot}|^2|\mathfrak{z}_l^{n,\cdot}|^2)|\tilde{\omega}_l^{n,\cdot}|^2 \, . \end{aligned}$$

Under the boundedness Assumption 2.2.2 (i), recalling (2.2.27), we obtain

$$|\mathfrak{G}^{\tilde{\mathfrak{u}}}|^2 \leqslant 2\left(|g|_\infty^2 + T^2|f|_\infty^2\right) \tag{2.4.31}$$

so that setting

$$\varrho_0 := 8\max\{(|g|_\infty + T|f|_\infty)^2, 2\} \tag{2.4.32}$$

and from the lower bound (2.2.42), we obtain

$$\mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|H^{n,l}(\mathcal{X}_0, W, \tilde{\mathfrak{u}}, \mathfrak{z}_l^{n,\cdot})|^2\Big] \leqslant \varrho_0(1 + h|\mathfrak{z}_l^{n,\cdot} - \check{\mathfrak{z}}_l^{n,\cdot}|^2 + h|\check{\mathfrak{z}}_l^{n,\cdot}|^2) \, . \tag{2.4.33}$$

Step 2: We now introduce the natural filtration of the algorithm namely $\mathcal{F} = (\mathcal{F}_m)_{0\leqslant m\leqslant M}$, defined by $\mathcal{F}_m = \sigma(\mathfrak{u}_0, \mathcal{X}_0^{(k)}, W^{(k)}, 1 \leqslant k \leqslant m)$, $m \geqslant 1$, and $\mathcal{F}_0 = \sigma(\mathfrak{u}_0)$. From the dynamics (2.2.48), we directly get

$$\begin{aligned} |\mathfrak{z}_{l,m+1}^{n,\cdot} - \check{\mathfrak{z}}_l^{n,\cdot}|^2 = |\mathfrak{z}_{l,m}^{n,\cdot} - \check{\mathfrak{z}}_l^{n,\cdot}|^2 &- 2\gamma_{m+1}^z H^{n,l}(\mathcal{X}_0^{(m+1)}, W^{(m+1)}, \tilde{\mathfrak{u}}, \mathfrak{z}_{l,m}^{n,\cdot}) \cdot (\mathfrak{z}_{l,m}^{n,\cdot} - \check{\mathfrak{z}}_l^{n,\cdot}) \\ &+ (\gamma_{m+1}^z)^2 |H^{n,l}(\mathcal{X}_0^{(m+1)}, W^{(m+1)}, \tilde{\mathfrak{u}}, \mathfrak{z}_{l,m}^{n,\cdot})|^2 \end{aligned}$$

so that introducing the sequence of $\mathcal{F}$-martingale increments, for $m \geqslant 0$,

$$\Delta M_{m+1} := \Big(\frac{1}{\beta_K\sqrt{h}} \nabla_{\mathfrak{z}_l^{n,\cdot}} \mathfrak{H}^{n,l}(\tilde{\mathfrak{u}}, \mathfrak{u}_m) - H^{n,l}(\mathcal{X}_0^{(m+1)}, W^{(m+1)}, \tilde{\mathfrak{u}}, \mathfrak{z}_{l,m}^{n,\cdot})\Big) \cdot (\mathfrak{z}_{l,m}^{n,\cdot} - \check{\mathfrak{z}}_l^{n,\cdot}),$$

we have

$$|\mathfrak{z}^{n,\cdot}_{l,m+1} - \check{\mathfrak{z}}^{n,\cdot}_{l}|^2 = |\mathfrak{z}^{n,\cdot}_{l,m} - \check{\mathfrak{z}}^{n,\cdot}_{l}|^2 - \frac{2}{\beta_K\sqrt{h}}\gamma^z_{m+1}\nabla_{\mathfrak{z}^{n,\cdot}_l}\mathfrak{H}^{n,l}(\tilde{\mathfrak{u}},\mathfrak{u}_m)\cdot(\mathfrak{z}^{n,\cdot}_{l,m} - \check{\mathfrak{z}}^{n,\cdot}_{l}) + 2\gamma^z_{m+1}\Delta M_{m+1} + (\gamma^z_{m+1})^2|H^{n,l}(\mathcal{X}_0^{(m+1)}, W^{(m+1)}, \tilde{\mathfrak{u}}, \mathfrak{z}^{n,\cdot}_{l,m})|^2 .$$

Now, from the previous equality, (2.2.34) and the fact that $\mathbb{E}_{\tilde{\mathfrak{u}}}[\Delta M_{m+1}|\mathcal{F}_m] = 0$, recall (2.2.43), we obtain

$$\begin{aligned}
&\mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|\mathfrak{z}^{n,\cdot}_{l,m+1} - \check{\mathfrak{z}}^{n,\cdot}_{l}|^2\Big] \\
&\leqslant \mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|\mathfrak{z}^{n,\cdot}_{l,m} - \check{\mathfrak{z}}^{n,\cdot}_{l}|^2\Big]\Big(1 - 4\frac{h\alpha_K}{\beta_K\sqrt{h}}\gamma^z_{m+1}\Big) + (\gamma^z_{m+1})^2\mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|H^{n,l}(\mathcal{X}_0^{(m+1)}, W^{(m+1)}, \tilde{\mathfrak{u}}, \mathfrak{z}^{n,\cdot}_{l,m})|^2\Big] \\
&\leqslant \mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|\mathfrak{z}^{n,\cdot}_{l,m} - \check{\mathfrak{z}}^{n,\cdot}_{l}|^2\Big]\Big(1 - 4\frac{h\alpha_K}{\beta_K\sqrt{h}}\gamma^z_{m+1} + \varrho_0 h(\gamma^z_{m+1})^2\Big) + \varrho_0(\gamma^z_{m+1})^2(1 + h|\check{\mathfrak{z}}^{n,\cdot}_{l}|^2)
\end{aligned} \tag{2.4.34}$$

where, for the last inequality, we used (2.4.33) together with the fact that, since $(\mathcal{X}_0^{(m+1)}, W^{(m+1)})$ is independent of $\mathcal{F}_m$ and $\mathfrak{z}_m$ is $\mathcal{F}_m$-measurable, one has

$$\mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|H^{n,l}(\mathcal{X}_0^{(m+1)}, W^{(m+1)}, \tilde{\mathfrak{u}}, \mathfrak{z}^{n,\cdot}_{l,m})|^2|\mathcal{F}_m\Big] = \mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|H^{n,l}(\mathcal{X}_0, W, \tilde{\mathfrak{u}}, \mathfrak{z}^{n,\cdot}_{l})|^2\Big]_{|\mathfrak{z}^{n,\cdot}_l = \mathfrak{z}^{n,\cdot}_{l,m}} .$$

Observe now that by the very definition (2.2.45) of the sequence $(\gamma^z_m)_{m\geqslant 1}$ and using the fact that $\alpha_K/\beta_K \leqslant 1$ and $\varrho_0 \geqslant 16$, one gets $1 - 4\frac{h\alpha_K}{\beta_K\sqrt{h}}\gamma^z_{m+1} + \varrho_0 h(\gamma^z_{m+1})^2 = 1 - (4\frac{\alpha_K}{\beta_K}\gamma_{m+1} - \varrho_0\gamma^2_{m+1}) \geqslant 1 - (\sqrt{\varrho_0}\gamma_{m+1} - \varrho_0\gamma^2_{m+1}) \geqslant 1 - 1/4 = 3/4$. As a consequence, $\Pi_m := \prod_{k=1}^m(1 - 4\frac{h\alpha_K}{\beta_K\sqrt{h}}\gamma^z_k + \varrho_0 h(\gamma^z_k)^2) = \prod_{k=1}^m(1 - 4\frac{\alpha_K}{\beta_K}\gamma_k + \varrho_0\gamma^2_k)$ is a product of positive terms. From (2.4.34), we thus deduce

$$\begin{aligned}
&\mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|\mathfrak{z}^{n,\cdot}_{l,m+1} - \check{\mathfrak{z}}^{n,\cdot}_{l}|^2\Big] \\
&\leqslant \Pi_{m+1}\mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|\mathfrak{z}^{n,\cdot}_{l,0} - \check{\mathfrak{z}}^{n,\cdot}_{l}|^2\Big] + \varrho_0(\frac{1}{h} + |\check{\mathfrak{z}}^{n,\cdot}_{l}|^2)\sum_{q=1}^{m+1}\frac{\Pi_{m+1}}{\Pi_q}\gamma^2_q \\
&\leqslant 2\varrho_1\exp\Big(-4\frac{\alpha_K}{\beta_K}\Gamma_{m+1}\Big)\Big(\mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|\mathfrak{z}^{n,\cdot}_{l,0}|^2\Big] + |\check{\mathfrak{z}}^{n,\cdot}_{l}|^2\Big) \\
&\quad + \varrho_1\varrho_0(\frac{1}{h} + |\check{\mathfrak{z}}^{n,\cdot}_{l}|^2)\sum_{q=1}^{m+1}\exp\Big(-4\frac{\alpha_K}{\beta_K}(\Gamma_{m+1} - \Gamma_q)\Big)\gamma^2_q \\
&\leqslant \varrho_0\varrho_1\Big(1 + \mathbb{E}_{\tilde{\mathfrak{u}}}\Big[|\mathfrak{z}^{n,\cdot}_{l,0}|^2\Big]\Big)\Big(\frac{1}{h} + |\check{\mathfrak{z}}^{n,\cdot}_{l}|^2\Big)\sum_{q=1}^{m+1}\exp\Big(-4\frac{\alpha_K}{\beta_K}(\Gamma_{m+1} - \Gamma_q)\Big)\gamma^2_q
\end{aligned} \tag{2.4.35}$$

where we used the standard inequality $1+x \leqslant e^x$, the fact that $\rho_0 \geqslant 2$ and introduced the quantity

$$\varrho_1 := \exp(\varrho_0\sum_{m\geqslant 1}\gamma^2_m). \tag{2.4.36}$$

This concludes the proof for (2.4.27).

Step 3: Recalling Definition 2.2.5 and using (2.2.29) as well as Assumption 2.2.3, we directly deduce

$$\mathbb{E}_{\tilde{\mathfrak{u}}}\Big[\|\!|\Phi_M(\tilde{\mathfrak{u}}) - \Phi(\tilde{\mathfrak{u}})|\!\|^2\Big] \leqslant \mathbb{E}_{\tilde{\mathfrak{u}}}\Big[\kappa_K|\mathfrak{y}_M - \check{\mathfrak{y}}|^2 + h\kappa_K \sum_{n=1}^{N-1}\sum_{l=1}^{d} |\mathfrak{z}_{l,M}^{n,\cdot} - \check{\mathfrak{z}}_l^{n,\cdot}|^2\Big]$$

so that, using (2.4.27)

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathfrak{u}}}\Big[\|\!|\Phi_M(\tilde{\mathfrak{u}}) - \Phi(\tilde{\mathfrak{u}})|\!\|^2\Big] &\leqslant \kappa_K L_{K,M}(1 + |\check{\mathfrak{y}}|^2) + h\kappa_K L_{K,M} \sum_{n=1}^{N-1}\sum_{l=1}^{d}(\frac{1}{h} + |\check{\mathfrak{z}}_l^{n,\cdot}|^2)\,, \\
&\leqslant \kappa_K L_{K,M} + \frac{\kappa_K}{\alpha_K} L_{K,M}\|\check{\mathfrak{y}}\|_y^2 + h\kappa_K L_{K,M} N d\frac{1}{h} + \frac{\kappa_K}{\alpha_K} L_{K,M}\|\check{\mathfrak{z}}\|_z^2\,, \\
&\leqslant \kappa_K L_{K,M}\,(1 + dN) + \frac{\kappa_K}{\alpha_K} L_{K,M}|\!|\!|\check{\mathfrak{u}}|\!|\!|^2\,,
\end{aligned}$$

which concludes the proof. □

The following result provides an upper-bound for the quantity $L_{K,M}$ for a given specification of the learning step that is useful to study the complexity of the global algorithm.

Lemma 2.4.4 *Let Assumption 2.2.3 hold. For $\gamma > 0$, $\rho \in (\frac{1}{2}, 1)$, set $\gamma_m := \gamma m^{-\rho}$, $m \geqslant 1$. If the number of steps M in the stochastic gradient descent algorithm satisfies*

$$\gamma\frac{\alpha_K}{\beta_K} M^{1-\rho} \geqslant \frac{\sqrt{2}}{2}\,, \tag{2.4.37}$$

then, there exists some positive constant $C := C(\rho, \gamma)$ such that

$$L_{K,M} \leqslant C\left(e^{-2\sqrt{2}\ln(2)\gamma\frac{\alpha_K}{\beta_K}M^{1-\rho}} + \frac{\beta_K}{\alpha_K M^\rho}\right). \tag{2.4.38}$$

Remark 2.4.1

1. *In practice, $L_{K,M}$ should go to zero with respect to the optimal parameters. Thus, we must have that $\frac{\beta_K}{\alpha_K M^\rho}$ goes to zero and $\frac{\alpha_K}{\beta_K}M^{1-\rho}$ goes to infinity at the same time as M goes to infinity. This will be carefully discussed in Section 2.4.4.3. With these constraints, we will naturally have that (2.4.37) is satisfied.*
2. *A careful analysis of the proof below shows that $\lim_{\rho\to 0.5^+} C(\rho,\gamma) = +\infty$, which comes from the dependence of $C(\rho,\gamma)$ with respect to $\sum_1^{+\infty}\frac{1}{m^{2\rho}}$. However, one must bear in mind that ρ is a fixed (but optimised) parameter.*

Proof of Lemma 2.4.4. We have, since $\Gamma_m = \gamma\sum_{q=1}^m \frac{1}{q^\rho}$, for $m \geqslant 1$,

$$\frac{\gamma}{1-\rho}\left(m^{1-\rho} - 1\right) \leqslant \Gamma_m \leqslant \frac{\gamma}{1-\rho}\left(m^{1-\rho} - 1\right) + \gamma \tag{2.4.39}$$

by standard computations based on comparison between series and integral, leading to

$$\Gamma_m - \Gamma_M \leqslant \frac{\gamma}{1-\rho}\left(m^{1-\rho} - M^{1-\rho}\right) + \gamma. \tag{2.4.40}$$

Recalling (2.4.28), we employ the following decomposition

$$L_{K,M} = \varrho_0\varrho_1\left(1+\mathbb{E}\left[|\mathfrak{u}_0|^2\right]\right)\sum_{m=1}^{M}\exp\left(-4\frac{\alpha_K}{\beta_K}(\Gamma_M-\Gamma_m)\right)\gamma_m^2 \tag{2.4.41}$$

$$\leqslant \varrho_0\varrho_1\left(1+\mathbb{E}\left[|\mathfrak{u}_0|^2\right]\right)\exp\left(4\gamma\frac{\alpha_K}{\beta_K}\right)\left(A_M + B_M + \gamma_M^2\right) \tag{2.4.42}$$

with

$$A_M := \sum_{m=1}^{\lfloor M/2\rfloor}\exp\left(4\frac{\alpha_K}{\beta_K}\frac{\gamma}{1-\rho}\{m^{1-\rho}-M^{1-\rho}\}\right)\frac{\gamma^2}{m^{2\rho}},$$

$$B_M := \sum_{m=\lfloor M/2\rfloor+1}^{M-1}\exp\left(4\frac{\alpha_K}{\beta_K}\frac{\gamma}{1-\rho}\{m^{1-\rho}-M^{1-\rho}\}\right)\frac{\gamma^2}{m^{2\rho}}.$$

For the first term A_M, we observe that, for $m \leqslant \lfloor M/2\rfloor \leqslant M/2$ and $\frac{1}{2} < \rho < 1$,

$$m^{1-\rho} - M^{1-\rho} \leqslant -\frac{\sqrt{2}}{2}\ln(2)(1-\rho)M^{1-\rho}\,. \tag{2.4.43}$$

We then compute

$$A_M \leqslant \exp\left(-2\sqrt{2}\ln(2)\gamma\frac{\alpha_K}{\beta_K}M^{1-\rho}\right)\sum_{m=1}^{\lfloor M/2\rfloor}\frac{\gamma^2}{m^{2\rho}}$$

$$\leqslant C_{\rho,\gamma}\exp\left(-2\sqrt{2}\ln(2)\gamma\frac{\alpha_K}{\beta_K}M^{1-\rho}\right). \tag{2.4.44}$$

We now study the term B_M which reads

$$B_M = \gamma^2\exp\left(-4\frac{\alpha_K}{\beta_K}\frac{\gamma}{1-\rho}M^{1-\rho}\right)\sum_{m=\lfloor M/2\rfloor+1}^{M-1}\lambda(m)$$

where, the map λ is defined for $x \geqslant 1$ by

$$\lambda(x) := \exp\left(4\frac{\alpha_K}{\beta_K}\frac{\gamma}{1-\rho}x^{1-\rho}\right)\frac{1}{x^{2\rho}}. \tag{2.4.45}$$

We observe that λ is increasing on $[\lfloor M/2\rfloor+1, +\infty)$ when (2.4.37) holds. This leads to

$$\sum_{m=\lfloor M/2\rfloor+1}^{M-1}\lambda(m) \leqslant \int_{M/2}^{M}\lambda(x)\,\mathrm{d}x$$

$$\leqslant \frac{2^\rho}{M^\rho}\int_{M/2}^{M}\exp\left(4\frac{\alpha_K}{\beta_K}\frac{\gamma}{1-\rho}x^{1-\rho}\right)\frac{1}{x^\rho}\,\mathrm{d}x$$

$$\leqslant \frac{2^{\rho-2}\beta_K}{\gamma M^\rho\alpha_K}\exp\left(4\frac{\alpha_K}{\beta_K}\frac{\gamma}{1-\rho}M^{1-\rho}\right)$$

which in turn yields $B_M \leqslant \frac{\gamma\beta_K}{M^\rho\alpha_K}$.

Inserting the previous inequality and estimate (2.4.44) into (2.4.42) concludes the proof, since $\mathbb{E}\Big[|\mathfrak{u}_0|^2\Big] < \infty$, recall Definition 2.2.7, step 1. and $\frac{\alpha_K}{\beta_K} \leqslant 1$ recall Assumption 2.2.3 and (2.2.42). □

Remark 2.4.2 *Let us importantly point out that if one choses $\gamma_m = \gamma/m$, with $\gamma > 0$, then from standard comparison between series and integral $\Gamma_m - \Gamma_M \leqslant \gamma(\ln(m/M) + 1)$ so that repeating the computations of the proof of Lemma 2.4.4, one has to consider the two disjoint cases $\gamma < \frac{\beta_K}{4\alpha_K}$ and $\gamma > \frac{\beta_K}{4\alpha_K}$ in order to provide an upper bound for the quantity of interest $L_{K,M}$. Only the latter allows to obtain the best convergence rate of order $1/M$. However, in practice, the user does not know the exact value of $\frac{\beta_K}{4\alpha_K}$ so that one will often consider higher values of γ than requested which will have the undesirable effect to deteriorate the upper-bound as suggested by the value of ϱ_1 in (2.4.36). Moreover, as shown in Section 2.4.4, the value $\frac{\beta_K}{4\alpha_K}$ actually goes to infinity when the prescribed approximation error ε goes to zero so that the latter condition becomes more and more stringent.*

We now give an upper bound for the error $\mathcal{E}_P$ defined by (2.4.20), with respect to all the algorithm's parameters. These parameters will be chosen in the next section taking into account the precise specification of the functional approximation space.

Proposition 2.4.4 *Suppose that Assumption 2.2.1, Assumption 2.2.2 (i), (ii) and Assumption 2.2.3 hold. Assume that there exists a positive constant η (independent of N, M and the basis functions (ψ)) such that*

$$\frac{\kappa_K}{h \wedge \alpha_K} L_{K,M} \leqslant \eta \,. \tag{2.4.46}$$

If $T(1 + 2L^2(1 + h)) < 1$ and $\delta_{h,\eta} := \frac{16L^2T(1+\eta)}{1-T(1+2L^2(1+h))} < 1$, then for any $\varepsilon > 0$ there exists a positive constant C_ε such that

$$\mathcal{E}_{\mathrm{RM}} \leqslant C_\varepsilon \frac{\kappa_K}{h \wedge \alpha_K} L_{K,M}$$

so that, with the notations of Proposition 2.4.3, it holds

$$\mathcal{E}_P \leqslant \delta_{h,\varepsilon}^P \mathcal{E}_0 + C_\varepsilon \left(\frac{\kappa_K}{h \wedge \alpha_K} L_{K,M} + \mathcal{E}_\psi + \mathcal{E}_\pi \right). \tag{2.4.47}$$

Remark 2.4.3 *In practice, η will be fixed to be a small constant as the term in the left hand side of (2.4.46) should be asymptotically zero.*

Proof of Proposition 2.4.4.
Step 1: From Proposition 2.4.3, we see that to obtain (2.4.47), it remains to control

$$\mathcal{E}_{\mathrm{RM}} = \max_{1 \leqslant p \leqslant P} \mathbb{E}\Big[\|\!| \Phi_M(\mathfrak{u}_M^{p-1}) - \Phi(\mathfrak{u}_M^{p-1}) |\!\|^2 \Big] .$$

Using Lemma 2.4.3, we have

$$\mathbb{E}\Big[\|\!|\Phi_M(\mathfrak{u}_M^{p-1}) - \Phi(\mathfrak{u}_M^{p-1})|\!|\!|^2\Big] \leqslant \kappa_K L_{K,M}(1+dN) + \frac{\kappa_K}{\alpha_K} L_{K,M}\mathbb{E}\Big[|\!|\!|\Phi(\mathfrak{u}_M^{p-1})|\!|\!|^2\Big]. \tag{2.4.48}$$

To conclude the proof, we will in the next step, provide an upper bound for the term $\mathbb{E}\Big[|\!|\!|\Phi(\mathfrak{u}_M^{p-1})|\!|\!|^2\Big]$, uniformly with respect to p.

Step 2: We denote by C_ε a constant that may change from line to line along with ε. From Young's inequality and Lemma 2.4.2, we obtain

$$\begin{aligned}\mathbb{E}\big[|\!|\!|\Phi(\mathfrak{u}_M^p)|\!|\!|^2\big] &\leqslant (1+\varepsilon)\mathbb{E}\big[|\!|\!|\Phi(\mathfrak{u}_M^p) - \bar{\mathfrak{u}}|\!|\!|^2\big] + (1+\frac{1}{\varepsilon})|\!|\!|\bar{\mathfrak{u}}|\!|\!|^2 \\ &\leqslant \delta_h(1+\varepsilon)\mathbb{E}\big[|\!|\!|\mathfrak{u}_M^p - \bar{\mathfrak{u}}|\!|\!|^2\big] + C_\varepsilon(\mathcal{E}_\pi + \mathcal{E}_\psi + |\!|\!|\bar{\mathfrak{u}}|\!|\!|^2) \\ &\leqslant \delta_h(1+\varepsilon)\mathbb{E}\big[|\!|\!|\mathfrak{u}_M^p|\!|\!|^2\big] + C_\varepsilon(\mathcal{E}_\pi + \mathcal{E}_\psi + |\!|\!|\bar{\mathfrak{u}}|\!|\!|^2)\end{aligned}$$

up to a modification of ε. From the previous inequality, we readily obtain

$$\mathbb{E}\big[|\!|\!|\Phi(\mathfrak{u}_M^0)|\!|\!|^2\big] \leqslant C_\varepsilon.$$

Now, if p is a positive integer, noting again that

$$\mathbb{E}\big[|\!|\!|\mathfrak{u}_M^p|\!|\!|^2\big] \leqslant 2\mathbb{E}\Big[|\!|\!|\Phi_M(\mathfrak{u}_M^{p-1}) - \Phi(\mathfrak{u}_M^{p-1})|\!|\!|^2\Big] + 2\mathbb{E}\Big[|\!|\!|\Phi(\mathfrak{u}_M^{p-1})|\!|\!|^2\Big]$$

we obtain

$$\begin{aligned}\mathbb{E}\big[|\!|\!|\Phi(\mathfrak{u}_M^p)|\!|\!|^2\big] \leqslant 2\delta_h(1+\varepsilon)\mathbb{E}\Big[|\!|\!|\Phi(\mathfrak{u}_M^{p-1})|\!|\!|^2\Big] + 2\delta_h(1+\varepsilon)\mathbb{E}\Big[|\!|\!|\Phi_M(\mathfrak{u}_M^{p-1}) - \Phi(\mathfrak{u}_M^{p-1})|\!|\!|^2\Big] \\ + C_\varepsilon\Big(\mathcal{E}_\pi + \mathcal{E}_\psi + |\!|\!|\bar{\mathfrak{u}}|\!|\!|^2\Big)\end{aligned}$$

so that using (2.4.48), we get

$$\begin{aligned}\mathbb{E}\big[|\!|\!|\Phi(\mathfrak{u}_M^p)|\!|\!|^2\big] \leqslant& 2\delta_h(1+\varepsilon)\left(1+\frac{\kappa_K}{\alpha_K}L_{K,M}\right)\mathbb{E}\Big[|\!|\!|\Phi(\mathfrak{u}_M^{p-1})|\!|\!|^2\Big] \\ &+ 2\delta_h(1+\varepsilon)\kappa_K L_{K,M}\,(1+dN) + C_\varepsilon\Big(\mathcal{E}_\pi + \mathcal{E}_\psi + |\!|\!|\bar{\mathfrak{u}}|\!|\!|^2\Big)\,.\end{aligned}$$

From (2.4.46) and the fact that $\delta_{h,\eta}$, we can set ε such that $2\delta_h(1+\varepsilon)(1+\frac{\kappa_K}{\alpha_K}L_{K,M}) \leqslant \delta_{h,\eta}(1+\varepsilon) < 1$ so that from the above inequality, by induction, for any positive integer p, we get

$$\mathbb{E}\big[|\!|\!|\Phi(\mathfrak{u}_M^p)|\!|\!|^2\big] \leqslant (\delta_{h,\eta}(1+\varepsilon))^p\mathbb{E}\big[|\!|\!|\Phi(\mathfrak{u}_M^0)|\!|\!|^2\big] + C_\varepsilon(\mathcal{E}_\pi + \mathcal{E}_\psi + |\!|\!|\bar{\mathfrak{u}}|\!|\!|^2 + \delta_h) \leqslant C_\varepsilon\,,$$

which concludes the proof. □

We now have all the ingredients to give the proof of the main result announced in Section 2.2.3 on the upper bound for the global convergence error at the initial time.

2.4.3.2.1 Proof of Theorem 2.2.1 From the very definition (2.2.51) of the global error, we deduce

$$\mathcal{E}_{\mathrm{MSE}} \leqslant 2\mathbb{E}\Big[|u(0,\mathcal{X}_0) - \bar{u}(\mathcal{X}_0)|^2 + |Y_0^{\bar{\mathfrak{u}}} - Y_0^{\mathfrak{u}_M^P}|^2\Big] \tag{2.4.49}$$

where we used the notations introduced in (2.2.9), (2.4.12) and (2.4.14). A fortiori, we have

$$\mathbb{E}\big[|u(0,\mathcal{X}_0) - \bar{u}_0(\mathcal{X}_0)|^2\big] \leqslant \mathcal{E}_\psi \quad \text{and} \quad \mathbb{E}\Big[|Y_0^{\bar{\mathfrak{u}}} - Y_0^{\mathfrak{u}_M^P}|^2\Big] \leqslant \mathcal{E}_P \tag{2.4.50}$$

recalling (2.4.11), (2.2.30) and (2.4.20). Combining (2.4.50) and (2.4.49) yields

$$\mathcal{E}_{\mathrm{MSE}} \leqslant 2\left(\mathcal{E}_\psi + \mathcal{E}_P\right) . \tag{2.4.51}$$

We eventually conclude the proof by invoking Proposition 2.4.4 together with Lemma 2.4.4. □

2.4.4 Convergence and complexity analysis for sparse grid approximations

For this part, we work in the setting of Section 2.3.1.2. Our goal is to prove the theoretical upper-bound on the algorithm's complexity stated in Theorem 2.3.1.

We first state the following useful estimate.

Lemma 2.4.5 *Suppose that Assumption 2.3.1 is satisfied. Let $\phi : \mathbb{R}^d \to \mathbb{R}$ be a non-negative measurable function whose support is included in $\mathcal{O}$ and $\check{\phi}$ be its* 1-periodisation *defined by (2.3.16). Then, it holds*

$$\mathfrak{C}^{-1}\mathbb{E}[\phi(U)] \leqslant \mathbb{E}\Big[\check{\phi}(X_{t_n})\Big] = \mathbb{E}\Big[\phi(\widehat{X}_{t_n})\Big] \leqslant \mathfrak{C}\mathbb{E}[\phi(U)]$$

where U has law $\mathcal{U}((0,1)^d)$ and $\mathfrak{C}$ is given in Lemma 2.2.1.

Proof. We denote by $x' \mapsto p_X(t_n, x')$ the density function of X_{t_n} given by the Euler-Maruyama scheme taken at time t_n and starting from $\mathcal{X}_0$ with law $\mathcal{U}((0,1)^d)$ at time 0. Note that we have $p_X(t_n, x') = \int p^\pi(0, t_n, x, x')\mathbf{1}_{(0,1)^d}(x)\,\mathrm{d}x$ and using (2.2.4),

$$\mathfrak{C}^{-1}\int p(\mathfrak{c}t_n, x - x')\mathbf{1}_{(0,1)^d}(x)\,\mathrm{d}x \leqslant p_X(t_n, x') \leqslant \mathfrak{C}\int p(\mathfrak{c}^{-1}t_n, x' - x)\mathbf{1}_{(0,1)^d}(x)\,\mathrm{d}x\,.$$

Then,

$$\mathbb{E}\Big[\phi(\widehat{X}_{t_n})\Big] = \mathbb{E}\Big[\check{\phi}(X_{t_n})\Big] \geqslant \mathfrak{C}^{-1}\int \check{\phi}(x')\int p(\mathfrak{c}t_n, x - x')\mathbf{1}_{(0,1)^d}(x)\,\mathrm{d}x\,\mathrm{d}x'$$

so that introducing the notation $\Xi = \xi + W_{\mathfrak{c}t_n}$,

$$\int \check{\phi}(x')\int p(\mathfrak{c}t_n, x - x')\mathbf{1}_{(0,1)^d}(x)\,\mathrm{d}x\,\mathrm{d}x' = \mathbb{E}\Big[\check{\phi}(\Xi)\Big] = \mathbb{E}\Big[\phi(\widehat{\Xi})\Big] \tag{2.4.52}$$

The proof is then concluded by observing that $\mathcal{L}(\widehat{\Xi}) = \mathcal{L}(U)$. The proof of the upper-bound follows from similar arguments. □

2.4.4.1 Sparse grid approximation error

We now provide some upper-bound estimates for the sparse grid approximation error.

Theorem 2.4.1 *Under Assumption 2.3.1, there exists a positive constant* $C := C(T, b, \sigma, d, \lambda_0)$ *such that*

$$\mathcal{E}_\psi \leqslant C 2^{-4\ell} \ell^{d-1}. \tag{2.4.53}$$

To obtain an error of order ε^2 *for the quantity* $\mathcal{E}_\psi$ *one may thus set*

$$\ell_\varepsilon = \log_2(\varepsilon^{-\frac{1}{2}} |\log_2(\varepsilon)|^{\frac{d-1}{4}}),$$

so that, for each $n = 1, \cdots, N-1$, *the number of basis functions required satisfies*

$$K_\varepsilon = \varepsilon^{-\frac{1}{2}} |\log_2(\varepsilon)|^{\frac{5(d-1)}{4}}.$$

Proof. For $1 \leqslant i \leqslant d$, $n = 0, \cdots, N-1$, setting $v_i(t_n, x) = (\sigma^\top \nabla_x u)_i(t_n, x) \mathbf{1}_{\{x \in \mathcal{O}\}}$, $x \in \mathbb{R}^d$, from (2.2.2) we have

$$\|u\|_{H^k_{mix}(\mathcal{O})} + \max_{1 \leqslant i \leqslant d} \|v_i\|_{H^k_{mix}(\mathcal{O})} \leqslant C. \tag{2.4.54}$$

Moreover, from (2.3.20), we obtain

$$(\sigma^\top \nabla_x u)_i(t_n, X_{t_n}) = v_i(t_n, \widehat{X}_{t_n}),$$

thus

$$\begin{aligned}
\mathbb{E}\big[|(\sigma^\top \nabla_x u)(t_n, X_{t_n}) - Z^{\mathfrak{u}}_{t_n}|^2\big] &= \sum_{i=1}^d \mathbb{E}\left[\left|v_i(t_n, \widehat{X}_{t_n}) - \sum_{k=1}^K \mathfrak{z}_i^{n,k} \psi_n^k(\widehat{X}_{t_n})\right|^2\right] \\
&\leqslant \mathfrak{C} \sum_{i=1}^d \mathbb{E}\left[\left|v_i(t_n, U) - \sum_{k=1}^K \mathfrak{z}_i^{n,k} \psi_n^k(U)\right|^2\right]
\end{aligned}$$

with $U \sim \mathcal{U}((0,1)^d)$ and where we use the upper-estimate given in Lemma 2.4.5 to obtain the last inequality. We also recall that

$$\mathbb{E}\big[|u(0, \mathcal{X}_0) - Y_0^{\mathfrak{u}}|^2\big] = \mathbb{E}\left[|u(0, U) - \sum_{k=1}^K \mathfrak{y}^k \psi_y^k(U)|^2\right]. \tag{2.4.55}$$

From (2.2.18) and the previous estimates, we thus deduce

$$\begin{aligned}
\mathcal{E}_\psi &\leqslant C \left(\inf_{\xi \in \mathscr{V}_y} \|\xi - u(0,.)\|^2_{L^2(\mathcal{O})} + \max_{0 \leqslant n \leqslant N-1} \sum_{i=1}^d \inf_{\xi \in \mathscr{V}_n^z} \|\xi - v_i(t_n,.)\|^2_{L^2(\mathcal{O})} \right), \\
&\leqslant C 2^{-4\ell} \ell^{d-1},
\end{aligned} \tag{2.4.56}$$

where for the last inequality we used (2.3.14) and (2.4.54).
Finally, setting $\ell_\varepsilon = \log_2(\varepsilon^{-\frac{1}{2}} |\log_2(\varepsilon)|^{\frac{d-1}{4}})$ yields $\mathcal{E}_\psi = O(\varepsilon^2)$ as $\varepsilon \downarrow 0$ and from (2.3.13) (see also Remark 2.3.1) we deduce that in this case $K_\varepsilon = \varepsilon^{-\frac{1}{2}} |\log_2(\varepsilon)|^{\frac{5(d-1)}{4}}$. □

2.4.4.2 Norm equivalence constants

We now provide some estimates for the value of α_K and κ_K appearing in Assumption 2.2.3.

Proposition 2.4.5 *Suppose that Assumption 2.3.1 holds. In the setting of Section 2.3.1.2, there exists a constant $\mathfrak{k} \leqslant 1$ such that*

$$\mathfrak{k} = \alpha_K = \frac{1}{\kappa_K} .$$

Proof. For any $\mathfrak{u} = (\mathfrak{y}, \mathfrak{z}) \in \mathbb{R}^{K^y} \times \mathbb{R}^{d\bar{K}^z}$, any $0 \leqslant n \leqslant N-1$ and any $l \in \{1, \cdots, d\}$, from (2.2.6) we have

$$\mathbb{E}\Big[|(Z^{\mathfrak{u}}_{t_n})^l|^2\Big] = \mathbb{E}\Bigg[\Big|\sum_{k=1}^{K} \mathfrak{z}_l^{n,k} \psi_n^k(\widehat{X}_{t_n})\Big|^2\Bigg] ,$$

Using Lemma 2.4.5, we obtain

$$\mathfrak{C}^{-1}\mathbb{E}\Bigg[\Big|\sum_{k=1}^{K} \mathfrak{z}_l^{n,k} \psi_n^k(U)\Big|^2\Bigg] \leqslant \mathbb{E}\Big[|(Z^{\mathfrak{u}}_{t_n})^l|^2\Big] \leqslant \mathfrak{C}\mathbb{E}\Bigg[\Big|\sum_{k=1}^{K} \mathfrak{z}_l^{n,k} \psi_n^k(U)\Big|^2\Bigg] . \tag{2.4.57}$$

Note that in our setting, $\psi_n^k = \chi^k$, so it holds

$$\int_{\mathcal{O}} \Big|\sum_{k=1}^{K} \mathfrak{z}_l^{n,k} \psi_n^k(z)\Big|^2 \mathrm{d}z = \int_{\mathcal{O}} \Big|\sum_{k=1}^{K} \mathfrak{z}_l^{n,k} \chi^k(x)\Big|^2 \mathrm{d}x . \tag{2.4.58}$$

Since the basis functions $(\chi^k)_{1\leqslant k\leqslant K}$ forms a Riesz basis [52], there exists a constant $\underline{c} \geqslant 1$ such that it holds

$$\underline{c}^{-1} \sum_{k=1}^{K} |\mathfrak{z}_l^{n,k}|^2 \leqslant \int_{[0,1]^d} \Big|\sum_{k=1}^{K} \mathfrak{z}_l^{n,k} \chi^k(x)\Big|^2 \mathrm{d}x \leqslant \underline{c} \sum_{k=1}^{K} |\mathfrak{z}_l^{n,k}|^2 \tag{2.4.59}$$

Combining (2.4.57)-(2.4.58)-(2.4.59) and taking into account all the component of $Z^{\mathfrak{u}}_{t_n}$, we compute

$$\underline{c}^{-1}\mathfrak{C}^{-1} d \sum_{k=1}^{K} |\mathfrak{z}_l^{n,k}|^2 \leqslant \mathbb{E}\big[|(Z^{\mathfrak{u}}_{t_n})|^2\big] \leqslant \underline{c}\mathfrak{C} d \sum_{k=1}^{K} |\mathfrak{z}_l^{n,k}|^2 . \tag{2.4.60}$$

We also observe that

$$\mathbb{E}\big[|Y^{\mathfrak{u}}_0|^2\big] = \mathbb{E}\Bigg[\Big|\sum_{k=1}^{K} \mathfrak{y}^k \chi^k(\mathcal{X}_0)\Big|^2\Bigg] . \tag{2.4.61}$$

Since $\mathcal{X}_0 \sim \mathcal{U}((0,1)^d)$, we similarly deduce that

$$\underline{c}^{-1} \sum_{k=1}^{K} |\mathfrak{y}_l^{n,k}|^2 \leqslant \mathbb{E}\big[|Y^{\mathfrak{u}}_0|^2\big] \leqslant \underline{c} \sum_{k=1}^{K} |\mathfrak{y}_l^{n,k}|^2. \tag{2.4.62}$$

The proof is concluded by combining (2.4.60) and (2.4.62) with (2.2.30) and setting $\mathfrak{k} = \underline{c}^{-1}\mathfrak{C}^{-1}d^{-1}$. □

2.4.4.3 Complexity analysis

Lemma 2.4.6 *Under Assumption 2.3.1, one can set*

$$\beta_K = C_d(1 + 2^\ell \ell^{d-1}) \tag{2.4.63}$$

for some positive constant C_d which depends on the PDE dimension d.

Proof.
Step 1: For any $n \in \{0, \dots, N-1\}$, any $\ell \in \{1, \dots, d\}$, one has

$$\mathbb{E}\big[|\tilde{\omega}_\ell^{n,\cdot}|^4\big] = \mathbb{E}\left[\left(\sum_{k=1}^K |\tilde{\omega}_\ell^{n,k}|^2\right)^2\right] . \tag{2.4.64}$$

Using Jensen's inequality, we obtain

$$\mathbb{E}\big[|\tilde{\omega}_\ell^{n,\cdot}|^4\big] \leqslant K\mathbb{E}\left[\sum_{k=1}^K |\tilde{\omega}_\ell^{n,k}|^4\right] \leqslant 3d^2K \sum_{k=1}^K \mathbb{E}\Big[|\psi_n^k(\widehat{X}_{t_n})|^4\Big] . \tag{2.4.65}$$

From now on, we use the indexation related to the sparse grid description introduced in Remark 2.3.1, namely, we write

$$\sum_{k=1}^K \mathbb{E}\Big[|\psi_n^k(\widehat{X}_{t_n})|^4\Big] = \sum_{(\mathbf{l},\mathbf{i})\in\mathcal{C}} \mathbb{E}\Big[|\psi_n^{(\mathbf{l},\mathbf{i})}(\widehat{X}_{t_n})|^4\Big] \tag{2.4.66}$$

$$= \sum_{(\mathbf{l},\mathbf{i})\in\mathcal{C}} \mathbb{E}\Big[|\chi^{(\mathbf{l},\mathbf{i})}(\widehat{X}_{t_n})|^4\Big] \tag{2.4.67}$$

in our setting.
Step 2: Using Lemma 2.4.5, we observe

$$\mathbb{E}\Big[|\chi^{(\mathbf{l},\mathbf{i})}(\widehat{X}_{t_n})|^4\Big] \leqslant \mathfrak{C}\mathbb{E}\Big[|\chi^{(\mathbf{l},\mathbf{i})}(U)|^4\Big] . \tag{2.4.68}$$

Moreover, we compute

$$\int |\phi^{(l,i)}(x)|^4 \,\mathrm{d}x = \int |\phi(2^l x - i)|^4 \,\mathrm{d}x \leqslant C2^{-l}$$

and from the definition of $\chi^{(l_j,i_j)}$, we deduce

$$\int |\chi^{(l_j,i_j)}(x)|^4 \,\mathrm{d}x \leqslant C2^l .$$

Combining the previous inequality with (2.4.68) leads to

$$\mathbb{E}\Big[|\chi^{(\mathbf{l},\mathbf{i})}(\widehat{X}_{t_n})|^4\Big] \leqslant C2^{|\mathbf{l}|_1}$$

and inserting the previous estimate in (2.4.66) yields

$$\sum_{k=1}^K \mathbb{E}\Big[|\psi_n^k(\widehat{X}_{t_n})|^4\Big] \leqslant C \sum_{(\mathbf{l},\mathbf{i})\in\mathcal{C}} 2^{|\mathbf{l}|_1}. \tag{2.4.69}$$

Step 3: We now quantify the term appearing in the right-hand side of (2.4.69), namely

$$Q := \sum_{(\mathbf{l},\mathbf{i})\in\mathcal{C}} 2^{|\mathbf{l}|_1} = 1 + \sum_{k=1}^{\ell} \sum_{\mathbf{l}\in\mathbb{N}^d} 2^{|\mathbf{l}|_1} \mathbf{1}_{\{\zeta_d(\mathbf{l})=k\}} \,. \tag{2.4.70}$$

We denote by $\|\mathbf{l}\|_0 = |\{j|l_j = 0\}|$. For $\mathbf{l} \neq \mathbf{0}$, recall that $\zeta_d(\mathbf{l}) = |\mathbf{l}|_1 + \|\mathbf{l}\|_0 - (d-1)$ (from the definition of ζ_d). Thus,

$$\begin{aligned} Q &= 1 + \sum_{k=1}^{\ell} \sum_{q=0}^{d-1} \sum_{\mathbf{l}\in(\mathbb{N}_{>0})^{d-q}} 2^{|\mathbf{l}|_1} \mathbf{1}_{\{|\mathbf{l}|_1=k+d-1-q \text{ and } \|\mathbf{l}\|_0=q\}} \\ &= 1 + \sum_{k=1}^{\ell} \sum_{q=0}^{d-1} 2^{k+d-1-q} \mathrm{C}_{k+d-q-2}^{d-q-1} \mathrm{C}_d^q \end{aligned}$$

recall that $|\{\mathbf{l} \in (\mathbb{N}_{>0})^{d-q} \,|\, |\mathbf{l}|_1 = k+d-1-q\}| = \mathrm{C}_{k+d-q-2}^{d-q-1}$. Introducing $\theta = d-1-q$, we get

$$Q = 1 + \sum_{k=1}^{\ell} d2^k + \sum_{\theta=1}^{d-1} 2^{1+\theta} \mathrm{C}_d^{d-1-\theta} \sum_{k=1}^{\ell} 2^{k-1} \mathrm{C}_{k-1+\theta}^{\theta} \tag{2.4.71}$$

From Lemma 3.6 in [20], we know that $\sum_{k=1}^{\ell} 2^{k-1} \mathrm{C}_{k-1+\theta}^{\theta} = 2^{\ell}(\frac{\ell^\theta}{\theta!} + O_d(\ell^{\theta-1}))$. We thus obtain

$$Q = 2^{\ell} \left(\frac{2^d}{(d-1)!} \ell^{d-1} + O_d(\ell^{d-2}) \right)$$

which combined with (2.4.69) yields

$$\sum_{k=1}^{K} \mathbb{E}\Big[|\psi_n^k(\widehat{X}_{t_n})|^4 \Big] \leqslant C_d 2^{\ell} \ell^{d-1}. \tag{2.4.72}$$

Combining the previous inequality with (2.4.65), we obtain

$$\mathbb{E}\big[|\tilde{\omega}_\ell^{n,\cdot}|^4\big] \leqslant C_d 2^{2\ell} \ell^{2d-2} \,. \tag{2.4.73}$$

Using similar arguments, as the basis function are chosen to be the same in our setting, we also have

$$\mathbb{E}\big[|\theta|^4\big] \leqslant C_d 2^{2\ell} \ell^{2d-2} \,.$$

The proof is then concluded recalling the definition of β_K in (2.2.42). □

We now turn to the analysis of the convergence and complexity of the full Picard algorithm. The following corollary is a preparatory result and expresses the main convergence results in terms of the parameters P, M, ℓ and h.

Corollary 2.4.1 *Suppose that Assumption 2.3.1 as well as (2.4.37) and (2.4.46) hold. Set $\gamma_m = \gamma/m^\rho$, for some $\rho \in (1/2, 1)$ and $\gamma > 0$. If $T(1+2L^2(1+h)) < 1$ and $\delta_{h,\eta} = \frac{16L^2T(1+2\eta)}{1-T(1+2L^2(1+h))} < 1$, then, with the notations of Proposition 2.4.3, for any $\varepsilon > 0$ such that $\delta_{h,\varepsilon} := \delta_{h,\eta}(1+\varepsilon) < 1$ there exist constants $C_\varepsilon := C(\varepsilon, T, b, \sigma, d, \gamma, \rho) \geqslant 1$, $c := c(T, b, \sigma, d, \gamma) > 0$ such that it holds*

$$\mathcal{E}_P \leqslant \delta_{h,\varepsilon}^P \mathcal{E}_0 + C_\varepsilon \left(N e^{-c\frac{M^{1-\rho}}{1+2^\ell \ell^{d-1}}} + \frac{1+2^\ell \ell^{d-1}}{hM^\rho} + 2^{-4\ell}\ell^{d-1} + h \right). \tag{2.4.74}$$

Proof. Combining Proposition 2.4.4 with Theorem 2.4.1 and (2.2.19), we obtain

$$\mathcal{E}_P \leqslant \delta_{h,\varepsilon}^P \mathcal{E}_0 + C_\varepsilon \left(\frac{\kappa_K}{h \wedge \alpha_K} L_{K,M} + 2^{-4\ell}\ell^{d-1} + h \right). \tag{2.4.75}$$

From Proposition 2.4.5, we have that $\frac{\kappa_K}{h\wedge\alpha_K} \leqslant \frac{C}{h}$, which combined with Lemma 2.4.4 gives

$$\frac{\kappa_K}{h \wedge \alpha_K} L_{K,M} \leqslant \frac{C_{\rho,\gamma}}{h} \left(e^{-\gamma 2\sqrt{2}\ln(2)\frac{\mathfrak{k}}{\beta_K} M^{1-\rho}} + \frac{\beta_K}{\mathfrak{k} M^\rho} \right) \tag{2.4.76}$$

$$\leqslant \frac{C_{\rho,\gamma}}{h} \left(e^{-c\frac{M^{1-\rho}}{1+2^\ell \ell^{d-1}}} + \frac{1+2^\ell \ell^{d-1}}{M^\rho} \right) \tag{2.4.77}$$

for some positive constant c, where we used Lemma 2.4.6 for the last inequality. □

We are now ready to establish the complexity of the full Picard algorithm.

2.4.4.3.1 Proof of Theorem 2.3.1 *Step 1: Setting the parameters P, N, M, ℓ and ρ.*
We will chose the parameters P, N, M, ℓ and ρ in order to achieve a global error $\mathcal{E}_P$ of order ε^2, as this error controls $\mathcal{E}_{\text{MSE}}$. We first set $P = 2|\log_\delta(\varepsilon)|$ and $N_\varepsilon = \lceil T\varepsilon^{-2} \rceil$ so that $h_\varepsilon = T/N_\varepsilon \leqslant \varepsilon^2$. From Theorem 2.4.1, we also know that setting $\ell_\varepsilon = \log_2(\varepsilon^{-\frac{1}{2}}|\log_2(\varepsilon)|^{\frac{d-1}{4}})$, we obtain $\mathcal{E}_\psi = O_d(\varepsilon^2)$ and $K_\varepsilon = O_d(\varepsilon^{-\frac{1}{2}}|\log_2(\varepsilon)|^{\frac{5(d-1)}{4}})$.

We now set M such that the term $\frac{K_\varepsilon}{M^\rho h_\varepsilon}$ is of order ε^2, which leads to

$$M_\varepsilon = O_d(\varepsilon^{-\frac{9}{2\rho}}|\log_2(\varepsilon)|^{\frac{5(d-1)}{4\rho}}).$$

For $\iota > 1$, we set $\bar\rho = \frac{9}{10\iota}$ with the constraint $\bar\rho > \frac{1}{2}$ and we verify that

$$\frac{M_\epsilon^{1-\bar\rho}}{K_\varepsilon} \geqslant c\varepsilon^{5(1-\iota)}|\log_2(\varepsilon)|^{\frac{5}{4}(d-1)(\frac{1}{\bar\rho}-2)}$$

for some constant $c > 0$. This leads to

$$e^{-c\frac{M_\varepsilon^{1-\bar\rho}}{1+2^{\ell_\varepsilon}\ell_\varepsilon^{d-1}}} = o(\varepsilon^4)$$

and we also have that (2.4.37) is satisfied.
Step 2: Computing the complexity $\mathcal{C}_\varepsilon$. Recalling Remark 2.2.4, we see that the

overall complexity $\mathcal{C}_\varepsilon$ to reach the prescribed approximation accuracy ε^2 satisfies

$$\begin{aligned}\mathcal{C}_\varepsilon &= P_\varepsilon N_\varepsilon K_\varepsilon M_\varepsilon \\ &= O_d\left(|\log_2(\varepsilon)|\varepsilon^{-2}\varepsilon^{-\frac{1}{2}}|\log_2(\varepsilon)|^{\frac{5(d-1)}{4}}\varepsilon^{-5\iota}|\log_2(\varepsilon)|^{\frac{25(d-1)\iota}{18}}\right) \\ &= O_d(\varepsilon^{-\frac{5}{2}(1+2\iota)}|\log_2(\varepsilon)|^{1+\frac{45+50\iota}{36}(d-1)}) ,\end{aligned}$$

which concludes the proof.

□

2.5 Appendix

2.5.1 Algorithms parameters

We gather below all the parameters values used in the various algorithms, examples and basis functions settings. In particular, we recall that the domain specification is given in (2.3.23) and (2.3.24) The learning rates are given by (2.3.25). Denoting

$$\Gamma(\lambda) := (\alpha(\lambda), \beta_1(\lambda), \beta_0(\lambda), m_0(\lambda)) \in \mathbb{R}^4,$$

we set the parameters in all the approximating space $\mathscr{V}_n^z, 1 \leqslant n \leqslant N-1$ to be the same: namely $\Gamma(\mathfrak{z}^{n,\cdot}) = \Gamma(\mathfrak{z}^{1,\cdot})$ for all $n \geqslant 2$. Thus, in the table below, the parameters of the learning rates are simply denoted by:

$$\Gamma := \{\Gamma(\mathfrak{y}), \Gamma(\mathfrak{z}^{0,\cdot}), \Gamma(\mathfrak{z}^{1,\cdot})\} \in \mathbb{R}^{3\times 4}.$$

Algorithms	Basis functions	dim	N	M	T	Initial $\mathfrak{z}^{n,k}$	p	Γ	$\mathcal{E}_{\text{MSE}}$
Picard Algorithm	Pre-wavelets	3	10	100000	0.3	0	1	(0.6, 0, 3, 15000), (0.6, 0, 20, 10000)	0.0286
							2	(0.6, 0, 2, 8000), (0.6, 0, 10, 8000	0.0247
							3	(0.6, 0, 1, 8000), (0.6, 0, 5, 8000	0.0219
							4	(0.6, 0, 0.5, 8000), (0.6, 0, 3, 8000)	0.0207
							5	(0.6, 0, 0.5, 8000), (0.6, 0, 3, 8000)	0.0201

Table 2.6 – Parameters for the periodic example

Examples	a	dim	M	N	Initial y_0	Learning rate
Quadratic	1	5	2000	10	0.5	0.01
Limits to Picard algorithm	-0.4	2	5000	20	0.3	0.002
	-1.5	2	5000	20	0	0.001
Financial example	N.A.	2	6000	20	2	0.005
		4	6000	20	5	0.005
		5	5000	20	5	0.005
		10	5000	20	8	0.005
		15	5000	20	8	0.005
		20	5000	20	8	0.005
		25	5000	20	8	0.005

Table 2.7 – Parameters by model for the *deep learning method* with $layers = 4, batchsize = 64$

Examples	Basis functions	dim	N	r	Initial value			Γ
					$\mathfrak{y}$	$\mathfrak{z}^{0,\cdot}$	$\mathfrak{z}^{n,k}$	
Quadratic model	Pre-wavelets	5	10	2	0.5	-0.2	0	(1, 0, 1, 100), (0.8, 0, 1, 100), (0.86, 0.02, 0.05, 100)
	Hat	100	10	3.2	5.5	0	0	(0.9, 0, 1, 100), (0.7, 0, 1, 100), (1, 0.003, 0.01, 1000)
Financial example	Pre-wavelets	2	10	2	2	0	0	(1, 0, 1.5, 1000), (1, 0, 20, 1000), (1, 0.003, 0.01, 1000)
		4	10	2	5	0	0	(1, 0, 1, 1000), (1, 0, 20, 1000), (1, 0.001, 0.01, 1000)
	Hat	5	10	2	5	0	0	(0.95, 0, 0.3, 100), (1, 0, 5, 100), (1, 0.001, 0.01, 100)
		10	10	2.5	8	0	0	(0.9, 0, 0.35, 300), (1, 0, 5, 300), (1, 0.001, 0.01, 300)
		15	10	2.5	8	0	0	(0.85, 0, 0.3, 500), (1, 0, 5, 500), (1, 0.001, 0.01, 500)
		20	10	2.5	8	0	0	(0.8, 0, 0.2, 1000), (1, 0, 5, 1000), (1, 0.001, 0.01, 1000)
		25	10	2.8	8	0	0	(0.7, 0, 0.2, 1500), (1, 0, 5, 1500), (1, 0.001, 0.01, 1500)
The challenging example	Hat	1	10	2	0.5	0	0	(1, 0, 0.5, 100), (1, 0, 3, 100), (1, 0.1, 0.1, 100)
		2	20	2	0.1	0	0	(1, 0, 0.5, 100), (1, 0, 5, 100), (1, 0.2, 0.5, 100)
		5	40	2	0.4	-1	0	(1, 0, 0.5, 300), (0.95, 0, 5, 500), (1, 0.2, 0.1, 500)
		8	60	2.2	0.6	1	0.08	(1, 0, 0.35, 500), (1, 0, 5, 500), (1, 0.2, 1, 500)
		10	100	2.5	0.1	-1	-0.1	(1, 0, 0.35, 500), (0.95, 0, 5, 500), (1, 0.2, 1, 500)

Table 2.8 – Parameters by model for the *direct algorithm*

Examples	Basis functions	dim	r	Initial value			a	p	Γ
				$\mathfrak{y}$	$\mathfrak{z}^{0,\cdot}$	$\mathfrak{z}^{n,k}$			
Quadratic model	Pre-wavelets	5	2	0.5	-0.2	0	1	1	(1, 0, 0.8, 100), (0.9, 0, 1, 100), (0.84, 0.02, 0.05, 100)
								2	(1, 0, 0.3, 100), (0.9, 0, 0.4, 100), (0.84, 0.01, 0.02, 100)
								$p \geqslant 3$	(1, 0, 0.2, 100), (0.9, 0, 0.2, 100), (0.84, 0.005, 0.01, 100)
	Hat	25	2.8	2	0.1	0	1	$1 \leqslant p \leqslant 3$	(0.9, 0, 0.5, 100), (0.8, 0, 0.8, 100), (1, 0.003, 0.01, 1000)
Financial example	Pre-wavelets	4	2	5	0.1	-0.01	N.A.	1	(1, 0, 1, 1000), (1, 0, 20, 1000), (1, 0.001, 0.01, 1000)
								2	(1, 0, 0.3, 1000), (1, 0, 5, 1000), (1, 0.0005, 0.005, 1000)
								3	(1, 0, 0.2, 1000), (1, 0, 5, 1000), (1, 0.0003, 0.003, 1000)
								4, 5	(1, 0, 0.15, 1000), (1, 0, 3, 1000), (1, 0.0002, 0.002, 1000)
								$p \geqslant 6$	(1, 0, 0.1, 1000), (1, 0, 2, 1000), (1, 0.0001, 0.001, 1000)
	Hat	20	2.5	8	0	0	N.A.	1	(0.9, 0, 0.6, 1000), (1, 0, 5, 1000), (1, 0.001, 0.01, 1000)
								2	(0.9, 0, 0.2, 1000), (1, 0, 4, 1000), (1, 0.001, 0.01, 1000)
								3	(0.9, 0, 0.15, 1000), (1, 0, 3, 1000), (1, 0.0005, 0.005, 1000)
								$p \geqslant 4$	(0.9, 0, 0.1, 1000), (1, 0, 2, 1000), (1, 0.0005, 0.005, 1000)
Limits to Picard algorithm	Pre-wavelets	2	2	0.3	0	0	-0.4	$1 \leqslant p \leqslant 9$	(1, 0, 1, 300), (1, 0, 1, 300), (1, 0.6, 0.1, 300)
				0	0	0	-1.5	$1 \leqslant p \leqslant 9$	(1, 0, 2, 300), (1, 0, 1, 300), (1, 0.6, 0.1, 300)

Table 2.9 – Parameters by model for the *Picard algorithm*

Examples	Basis functions	dim	r	Initial value			N	p	Γ
				$\mathfrak{y}$	$\mathfrak{z}^{0,\cdot}$	$\mathfrak{z}^{n,k}$			
The challenging example	Hat	1	2	0.5	0	0	10	$1 \leqslant p \leqslant 5$	$(1, 0, 0.5 * (0.8)^{p-1}, 100)$, $(1, 0, 3*(0.8)^{p-1}, 100)$, $(1, 0.1*(0.8)^{p-1}, 0.1*(0.8)^{p-1}, 100)$
		2	2	0.1	0	0	20	$1 \leqslant p \leqslant 5$	$(1, 0, 0.5*(0.8)^{p-1}, 100)$, $(1, 0, 5*(0.8)^{p-1}, 100)$, $(1, 0.2*(0.8)^{p-1}, 0.5*(0.8)^{p-1}, 100)$
		5	2	0.4	-2	0	40	$1 \leqslant p \leqslant 5$	$(1, 0, 0.5*(0.8)^{p-1}, 300)$, $(0.95, 0, 5*(0.8)^{p-1}, 500)$, $(1, 0.2*(0.8)^{p-1}, 0.1*(0.8)^{p-1}, 500)$
		8	2.2	0.6	1	0.08	60	$1 \leqslant p \leqslant 5$	$(1, 0, 0.35*(0.8)^{p-1}, 500)$, $(1, 0, 5*(0.8)^{p-1}, 500)$, $(1, 0.2*(0.8)^{p-1}, (0.8)^{p-1}, 500)$
		10	2.5	0.1	-1	-0.1	100	1	$(1, 0, 0.25, 500)$, $(0.95, 0, 4, 500)$, $(1, 0.15, 0.5, 500)$
								2	$(1, 0, 0.2, 500)$, $(0.95, 0, 3, 500)$, $(1, 0.12, 0.4, 500)$
								3	$(1, 0, 0.15, 500)$, $(0.95, 0, 2, 500)$, $(1, 0.1, 0.3, 500)$
								$p \geqslant 4$	$(1, 0, 0.1, 500)$, $(0.95, 0, 1, 500)$, $(1, 0.08, 0.25, 500)$
								$p \geqslant 6$	$(1, 0, 0.05, 500)$, $(0.95, 0, 0.5, 500)$, $(1, 0.05, 0.2, 500)$

Table 2.10 – Parameters by model for the *Picard algorithm*

Chapter 3

Deep Runge-Kutta schemes for BSDEs

Contents

3.1 Introduction

In this chapter, we consider the forward diffusion process with dynamics

$$\mathcal{X}_t = \mathcal{X}_0 + \int_0^t \mu(\mathcal{X}_s)\,\mathrm{d}s + \int_0^t \sigma(\mathcal{X}_s)\,\mathrm{d}W_s, \quad 0 \leqslant t \leqslant T, \tag{3.1.1}$$

and we would like to approximate the solution of the BSDEs

$$\mathcal{Y}_t = g(\mathcal{X}_T) + \int_t^T f(\mathcal{X}_s, \mathcal{Y}_s, \mathcal{Z}_s)\,\mathrm{d}s - \int_t^T \mathcal{Z}_s \cdot \mathrm{d}W_s\,, \; 0 \leqslant t \leqslant T\,, \tag{3.1.2}$$

where W is a d-dimensional Brownian motion defined on a complete probability space $(\Omega, \mathcal{A}, \mathbb{P})$, $\mu : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \to \mathbb{M}_d$ (the set of $d \times d$ matrices) are measurable functions, the initial condition $\mathcal{X}_0 \in \mathbb{R}^d$. We denote the filtration generated by W and $\mathcal{X}_0$ as $(\mathcal{F}_t)_{0\leqslant t\leqslant T}$, augmented with $\mathbb{P}$ null sets.

Relying on the classical connection between Backward Stochastic Differential Equations (BSDEs) and non-linear parabolic partial differential equations (PDEs) initiated in [81], we have, under some regularity assumptions on μ, σ:

$$Y_t = u(t, \mathcal{X}_t), \quad Z_t = \sigma^\top(\mathcal{X}_t)\nabla_x u(t, \mathcal{X}_t), \quad 0 \leqslant t \leqslant T, \tag{3.1.3}$$

where $u : [0, T] \times \mathbb{R}^d \longmapsto \mathbb{R}$ is the solution to a semi-linear PDE:

$$\begin{cases} \partial_t u(t,x) + \mathcal{L}u(t,x) + f(u(t,x), \sigma^\top(x)\nabla_x u(t,x)) = 0, & (t,x) \in [0,T) \times \mathbb{R}^d, \\ u(T,x) = g(x), & x \in \mathbb{R}^d \end{cases} \tag{3.1.4}$$

and $\mathcal{L}$ is the infinitesimal generator defined by

$$\mathcal{L}u(t,x) := \mu(x) \cdot \nabla_x u(t,x) + \frac{1}{2}\mathrm{Tr}[(\sigma\sigma^\top)(x)\nabla_x^2 u(t,x)]. \tag{3.1.5}$$

Since BSDEs have been introduced by Pardoux and Peng [81, 82] in 1990s, designing efficient numerical algorithms to solve BSDEs has attracted considerable attention. However, solving high-dimensional BSDEs is a challenging task due to the "curse of dimensionality". Many traditional methods to solve BSDEs have been proposed in the last two decades, such as the cubature methods [25, 31, 32], optimal quantization methods [6, 5, 80, 77], Malliavin calculus based methods [33, 17, 64] and some linear regression methods [49, 50, 51]. However, these methods are bounded by a low dimensional setting $d \leqslant 10$. [23] proposed a learning scheme based on sparse grids and Picard approximations proved that the"curse of dimensionality" is tamed in the sense that the complexity is of order $\varepsilon^{-p}|\log(\varepsilon)|^{q(d)}$, where p is a constant which does not depend on d and $d \mapsto q(d)$ is an affine function. In some case, 100 dimensional BSDEs can be solved.

In the past five years, numerous numerical methods based on deep learning method (non-linear regression) to solve BSDEs have been proposed, including the forward scheme [36, 7, 58] and the backward scheme [65, 46, 86, 47], see more related research results from the papers [21, 67, 68, 90, 28]. These algorithms are based on the use of Euler schemes for the time-discretization. It is well known that the weak convergence rate of Euler scheme is of order 1, so that the computational time cost

for these algorithms is still large for high-dimensional BSDEs as many time steps might be required to achieve good accuracy. In this chapter, we combine some high-order time discretization numerical schemes with non-linear regression based on deep neural network to solve high-dimensional BSDEs.

High-order discrete-time approximation schemes have been introduced in [22, 24], see also the references therein. These high-order schemes are based on a backward algorithm and, as usual, they require a good estimation of conditional espectation in practice. In particular, the Crank-Nicolson scheme is a second-order scheme with a simple structure, see among others [32]. Though it is implicit, it requires no extra computation of conditional expectation compared to Euler scheme. More generally, Runge-Kutta methods [24] are a family of implicit and explicit discretization methods, which include implicit Euler scheme, explicit Euler scheme, Crank-Nicolson scheme and some other iterative methods that can achieve higher order convergence rates. To the best of our knowledge, these high order schemes have not been tested with (non-linear) regression techniques.

In this chapter, we establish the convergence of these algorithms with the help of universal approximation theorem of neural network, see Theorem 3.3.2. We also implement these schemes to compare the convergence rates and the computational time cost. We conclude that Crank-Nicolson scheme seems to be the best scheme to use if we want to achieve an error smaller than 0.01.

The rest of the chapter is organized as follows. We first recall the definition of Runge-Kutta schemes for BSDEs in Section 3.2, then we study the stability of Runge-Kutta schemes in two different ways. Theorem 3.2.1 gives the discrete time errors for the main methods that will be studied in this chapter. In Section 3.3, we present an implementation of the Runge-Kutta schemes to solve BSDEs by neural networks, including the special case of implicit Euler schemes [65], explicit Euler scheme, Crank-Nicolson scheme, two stage explicit Runge-Kutta scheme. We provide the error control of the general learning method by Runge-Kutta scheme and neural network in the end of this section, see Theorem 3.3.2. In Section 3.4, we numerically verify the convergence order of the discrete time error of the methods given in Theorem 3.2.1. We also compare the computational time cost of these methods.

In the whole chapter, we assume that the driver f and terminal function g satisfy the Lipschitz condition:

Assumption 3.1.1 *There exists constants $[f]_L > 0$ and $[g]_L > 0$ such that*

$$|f(x_2, y_2, z_2) - f(x_1, y_1, z_1)| \leqslant [f]_L \left(|y_2 - y_1| + |z_2 - z_1|\right), \tag{3.1.6}$$

$$|g(x_2) - g(x_1)| \leqslant [g]_L |x_2 - x_1|. \tag{3.1.7}$$

Assumption 3.1.2 *There exists a constant $C > 0$ such that the following two conditions hold for all $x, y \in \mathbb{R}^d$,*

$$|\mu(x) - \mu(y)| + |\sigma(x) - \sigma(y)| \leqslant C|x - y|. \tag{3.1.8}$$

Some notations and basic definitions about the neural networks that will be used

in this chapter, state as follows:

$$\begin{cases} \mathbb{M}_{m\times n}(\mathbb{R}): & \text{the matrix space for all } m\times n \text{ matrices with elements in } \mathbb{R},\\ d_0 = d: & \text{input dimension},\\ d_1: & \text{output dimension},\\ L+1\in\mathbb{N}\backslash\{0,1,2\}: & \text{number of layers of the network},\\ m_\ell, \ell = 0,1,\cdots,L: & \text{number of neurons on each layer, note } m_0 = d_0, m_L = d_1,\\ \mathcal{H}_i^\ell, \ell = 1,\cdots,L-1: & \text{output of the hidden layers at time } t_n, 0\leqslant n\leqslant N-1. \end{cases}$$

For Euler scheme, the output dimension $d_1 := 1+d$ which includes 1 component for $Y-$part and d components for $Z-$part. However, we choose $d_1 := 1+2d$ for the networks of more general schemes as Crank-Nicolson scheme: The components consist $1, d, d$ dimensions for Y, Z, A, respectively, where A is a $d-$dimensional variable that will be introduced later, see Section 3.3.2. For the $L-1$ hidden layers in this neural network, we choose for simplicity the same number of neurons $m_\ell = m, \ell = 1,\cdots,L-1$.

For $\ell = 1,\cdots,L$, we define the maps $\mathcal{M}_\ell : \mathbb{R}^{m_{\ell-1}} \longmapsto \mathbb{R}^{m_\ell}$ as:

$$\mathcal{M}_\ell(x) = \mathcal{W}_\ell x + \beta_\ell, \tag{3.1.9}$$

where $\mathcal{W}_\ell \in \mathbb{M}_{m_\ell\times m_{\ell-1}}(\mathbb{R})$ is a matrix called weight, and $\beta_\ell \in \mathbb{R}^{m_\ell}$ is a vector called bias. Then $\mathcal{M}_\ell$ is an affine transformation that can map the features of the $(\ell-1)$-th layer to the ℓ-th layer. A feedforward neural network is a function from $\mathbb{R}^{d_0}$ to $\mathbb{R}^{d_1}$ defined as the composition

$$x\in\mathbb{R}^{d_0} \longmapsto \mathcal{M}_L\circ\rho_{L-1}\circ\mathcal{M}_{L-1}\circ\cdots\circ\rho_1\circ\mathcal{M}_1(x)\in\mathbb{R}^{d_1}, \tag{3.1.10}$$

where $\rho_\ell(x) = (\rho(x_1),\cdots,\rho(x_{m_\ell})), x\in\mathbb{R}^{m_\ell}, \ell = 1,\cdots,L-1$, here $\rho:\mathbb{R}\longmapsto\mathbb{R}$ is an activation function which is also a nonlinear function, such as ReLu, Elu, tanh, sigmoid. Then the parameters of the neural network consist of the weight matrices $(\mathcal{W}_\ell)_{1\leqslant\ell\leqslant L}$, the bias vector $(\beta_\ell)_{1\leqslant\ell\leqslant L}$. For fixed d_0, d_1 and L, the total number of parameters is

$$N_m := \sum_{\ell=1}^{L} m_\ell(m_{\ell-1}+1) = d_0(1+m) + m(m+1)(L-2) + m(1+d_1),$$

so that the parameters can be identified with an element $\theta\in\mathbb{R}^{N_m}$. Defining

$$\mathbb{R}^{d_0}\ni x\longmapsto \mathcal{N}_m(x;\theta) = \mathcal{M}_L\circ\rho_{L-1}\circ\mathcal{M}_{L-1}\circ\cdots\circ\rho_1\circ\mathcal{M}_1(x)\in\mathbb{R}^{d_1}, \tag{3.1.11}$$

we introduce

$$\mathcal{S}^\rho_{d_0,d_1,L,m}(\mathbb{R}^{N_m}) := \left\{\mathcal{N}_m(\cdot;\theta)\in\mathbb{R}^{d_1} \quad | \quad \theta\in\mathbb{R}^{N_m}\right\} \tag{3.1.12}$$

and

$$\mathcal{S}^\rho_{d_0,d_1,L} := \bigcup_{m\in\mathbb{N}^+} \mathcal{S}^\rho_{d_0,d_1,L,m}(\mathbb{R}^{N_m}). \tag{3.1.13}$$

The fundamental result of Hornik et al. [62] states the following universal approximation theorem to justify that the neural networks can be applied as function approximators:

Theorem 3.1.1 (Universal approximation theorem) *$\mathcal{S}^{\rho}_{d_0,d_1,L}$ is dense in $L^2(\upsilon)$ for any finite measure υ on $\mathbb{R}^d$, whenever ρ is continuous and non-constant.*

3.2 Runge-Kutta schemes for BSDEs

We consider in our work a class of Runge-Kutta schemes, that have been introduced in [24]. The main difference with the previous work is that we also consider an approximation of the forward process.

3.2.1 Definitions

We consider an equidistant grid

$$\pi := \{t_0 = 0 < \cdots < t_n < \cdots < t_N = T\}$$

of the time interval $[0,T]$ with time step $h := \frac{T}{N}, t_n = nh, n = 0, \cdots, N$. And we denote $\Delta W_n = W_{t_{n+1}} - \Delta W_{t_n}, 0 \leqslant n \leqslant N-1$.

The Runge-Kutta schemes involve in full generality intermediate steps of computation between two dates of the main grid π. Thus, for a positive integer Q let $c = (c_1, \ldots, c_{Q+1}) \in [0,1]^{Q+1}$ satisfying $0 =: c_1 < c_2 \leqslant \ldots \leqslant c_q \leqslant \cdots \leqslant c_Q \leqslant c_{Q+1} := 1$. We introduce the intermediate "instances" $t_{n,q} := t_{n+1} - c_q h$. With these notations, we observe that $t_n = t_{n,Q+1} \leqslant \ldots \leqslant t_{n,q} \leqslant \ldots \leqslant t_{n,1} = t_{n+1}$. We denote the "full grid"

$$\Pi := \{t_{n,q} \in [0,T] \,|\, 0 \leqslant n \leqslant N, 1 \leqslant q \leqslant Q\}.$$

First, we are given an approximation of the forward component (3.1.1) on the grid Π. Namely, for $t_{n,q} \in \Pi$, $\mathcal{X}_{t_{n,q}}$ is approximated by $X_{n,q} \in \mathcal{L}^2(\mathcal{F}_{t_{n,q}})$, $0 \leqslant n \leqslant N$ and $1 \leqslant q \leqslant Q$. For ease of notation, we will simply denote by $(X_n)_{0 \leqslant n \leqslant N}$ the approximation of $\mathcal{X}$ on the grid π. Observe that $X_{n,Q+1} = X_n$ and $X_{n,1} = X_{n+1}$. In the following, we assume that X is a Markov process on Π. In this chapter, $R_{n,1} \equiv R_{n+1} \equiv R_{n+1,Q+1}, 0 \leqslant n \leqslant N-1$ represent the same random variables.

We now define (Y,Z) the approximation of $(\mathcal{Y}, \mathcal{Z})$, recall (3.1.2).

Definition 3.2.1 *i) Set the terminal condition as*

$$(Y_N, Z_N) = (g(X_N), \sigma(X_N)^\top \nabla g(X_N)).$$

ii) For $0 \leqslant n \leqslant N-1$ and $Q \geqslant 1$, the transition from (Y_{n+1}, Z_{n+1}) to (Y_n, Z_n) involves Q stages. At the intermediate instances, for $1 < q \leqslant Q+1$, let

$$Y_{n,q} = \mathbb{E}_{t_{n,q}}\left[Y_{n+1} + h \sum_{k=1}^{q} a_{qk} f(X_{n,k}, Y_{n,k}, Z_{n,k}) \right], \tag{3.2.1}$$

$$Z_{n,q} = \mathbb{E}_{t_{n,q}}\left[H^n_q Y_{n+1} + h \sum_{k=1}^{q-1} \alpha_{qk} H^n_{q,k} f(X_{n,k}, Y_{n,k}, Z_{n,k}) \right], \tag{3.2.2}$$

where $(a_{qk})_{1\leqslant q,k\leqslant Q+1}, (\alpha_{qk})_{1\leqslant q,k\leqslant Q+1}$ *take their values in* $\mathbb{R}$ *and with* $a_{1k} = \alpha_{1k} = 0, 1 \leqslant k \leqslant Q$, $a_{qk} = \alpha_{qk} = 0, 1 \leqslant q < k \leqslant Q+1$ *and*

$$\sum_{k=1}^{q} a_{qk} = \sum_{k=1}^{q-1} \alpha_{qk} \mathbb{1}_{\{c_k<c_q\}} = c_q, \qquad q \leqslant Q+1. \tag{3.2.3}$$

We set $(Y_n, Z_n) = (Y_{n,Q+1}, Y_{n,Q+1})$ *at the dates on* π.
For all $1 \leqslant k < q \leqslant Q+1, n \leqslant N$, *the random variables* $H_q^n, H_{q,k}^n$ *are* $\mathcal{F}_{t_{n+1}}-$*measurable, independent of* $\mathcal{F}_{t_{n,q}}$ *and* $\mathcal{F}_{t_{n,k}}$ *respectively with the property*

$$\mathbb{E}_{t_{n,q}}\big[H_q^n\big] = \mathbb{E}_{t_{n,k}}\big[H_{q,k}^n\big] = 0 \text{ and } \upsilon_q^n := \mathbb{E}_{t_{n,q}}\big[|H_q^n|^2\big], \upsilon_{q,k}^n := \mathbb{E}_{t_{n,k}}\big[|H_{q,k}^n|^2\big], \tag{3.2.4}$$

$$\frac{\lambda}{h} \leqslant \min(\upsilon_q^n, \upsilon_{q,k}^n) \text{ and } \max(\upsilon_q^n, \upsilon_{q,k}^n) \leqslant \frac{\Lambda}{h}, \tag{3.2.5}$$

where λ, Λ *are positive constants which do not depend on* h.

We note that (3.2.1) may define $Y_{n,q}$ implicitly but this definition is well-posed for h small enough (e.g. as soon as $\max_{1<q<Q+1} a_{qq} hL < 1$). Iteratively, one also obtains that

$$\max_{n,q} \mathbb{E}\big[|Y_{n,q}|^2 + |Z_{n,q}|^2\big] < +\infty\,. \tag{3.2.6}$$

3.2.2 Stability of Runge-Kutta scheme

A key property to obtain the convergence results stated in Theorem 3.2.1 is – classically – the L^2-stability of the schemes of Definition 3.2.1. This has already been observed in [24]. We shall review here this property as it will be useful in the sequel.

The first observation is the fact that the schemes given in Definition 3.2.1 can be written in the following implicit form, for $n < N$:

$$Y_n = \mathbb{E}_{t_n}\big[Y_{n+1} + \Phi_n^Y(Y_{n+1}, Z_{n+1}, h)\big] \tag{3.2.7}$$

$$Z_n = \mathbb{E}_{t_n}\big[H_{Q+1}^n Y_{n+1} + \Phi_n^Z(Y_{n+1}, Z_{n+1}, h)\big] \tag{3.2.8}$$

where $(\Phi_n^Y, \Phi_n^Z) : \Omega \times \mathcal{L}^2(\mathcal{F}_{t_{n+1}}) \times \mathcal{L}^2(\mathcal{F}_{t_{n+1}}) \times \mathbb{R}_+ \to \mathcal{L}^2(\mathcal{F}_{t_{n+1}})$. This writing really stresses the fact that the schemes are one-step scheme. One introduces a perturbed version of the scheme, namely,

$$\check{Y}_n = \mathbb{E}_{t_n}\Big[\check{Y}_{n+1} + \Phi_n^Y(\check{Y}_{n+1}, \check{Z}_{n+1}, h)\Big] + \check{\zeta}_n^Y \tag{3.2.9}$$

$$\check{Z}_n = \mathbb{E}_{t_n}\Big[H_{Q+1}^n \check{Z}_{n+1} + \Phi_n^Z(\check{Y}_{n+1}, \check{Z}_{n+1}, h)\Big] + \check{\zeta}_n^Z \tag{3.2.10}$$

for $(\check{\zeta}_n^Y, \check{\zeta}_n^Z) \in \mathcal{L}^2(\mathcal{F}_{t_n}) \times \mathcal{L}^2(\mathcal{F}_{t_n})$, and obtains, see Theorem 1.2(i) in [24], the following stability result, setting $\delta Y_n := \check{Y}_n - Y_n$, $\delta Z_n := \check{Z}_n - Z_n$,

$$\max_{n<N} \mathbb{E}\big[|\delta Y_n|^2\big] + \sum_{n=0}^{N-1} h\mathbb{E}\big[|\delta Z_n|^2\big] \leqslant C\mathbb{E}\left[|\delta Y_N|^2 + h|\delta Z_N|^2 + \sum_{n=0}^{N-1} \frac{|\check{\zeta}_n^Y|^2}{h} + h|\check{\zeta}_n^Z|^2\right]. \tag{3.2.11}$$

This approach is particularly well-suited for the study of the discrete time error, see the proof of Theorem 3.2.1 in Section 3.2.3 below. However, we need also a stability result to control the error linked to the estimation of the conditional expectations at each stage of the schemes. To this end, we now introduce another perturbed scheme, for $n < N$, at the intermediate instances, for $1 < q \leqslant Q + 1$, let

$$\tilde{Y}_{n,q} = \mathbb{E}_{t_{n,q}}\left[\tilde{Y}_{n+1} + h\sum_{k=1}^{q} a_{qk} f(X_{n,k}, \tilde{Y}_{n,k}, \tilde{Z}_{n,k})\right] + \zeta^y_{n,q}, \tag{3.2.12}$$

$$\tilde{Z}_{n,q} = \mathbb{E}_{t_{n,q}}\left[H^n_q \tilde{Y}_{n+1} + h\sum_{k=1}^{q-1} \alpha_{qk} H^n_{q,k} f(X_{n,k}, \tilde{Y}_{n,k}, \tilde{Z}_{n,k})\right] + \zeta^z_{n,q}, \tag{3.2.13}$$

with $(\zeta^y_{n,q}, \zeta^z_{n,q}) \in \mathcal{L}^2(\mathcal{F}_{t_{n,q}})$.

Associated to the above perturbed version, we can state the following stability result.

Proposition 3.2.1 *Assume that f is Lipschitz continuous. Then, setting $\delta Y_n := \tilde{Y}_n - Y_n$, $\delta Z_n := \tilde{Z}_n - Z_n$, the following holds*

$$\begin{aligned}&\max_{n<N} \mathbb{E}\big[|\delta Y_n|^2\big] + \sum_{n=0}^{N-1} h\mathbb{E}\big[|\delta Z_n|^2\big] \\ &\leqslant C\mathbb{E}\left[|\delta Y_N|^2 + h|\delta Z_N|^2 + \sum_{n=0}^{N-1}\sum_{q=2}^{Q+1}\left(\frac{|\zeta^y_{n,q}|^2}{h} + h|\zeta^z_{n,q}|^2\right)\right].\end{aligned} \tag{3.2.14}$$

Proof. See the proof in Section 3.5.1. □

3.2.3 Discrete time error

In [24], the discrete-time error has been studied when $X = \mathcal{X}$, namely there is no error in the approximation of the underlying process. Building on the results in [24], we will give an upper bound of the discrete-time error when the forward process is indeed approximated. We will focus here on one stage schemes both implicit and explicit and two and three stage explicit scheme: See Remark 3.2.1 for an explanation of this limitation. The control of the discrete-time error is based on smoothness assumptions satisfied by the value function u solution to (3.1.4). We now introduce the necessary notations to formalise this statement.

Let

$$\mathcal{M} := \{\varnothing\} \cup \bigcup_{m=1}^{\infty} \{0, \dots, d\}^m,$$

the set of multi-indices with entry $0, \dots, d$. We define the differential operators as

$$L^{(0)} = \partial_t + \sum_{i=1}^{d} \mu_i \partial_{x_i} + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} (\sigma\sigma^\top)_{ij} \partial^2_{x_i,x_j}, \tag{3.2.15}$$

$$L^{(\ell)} = \sum_{k=1}^{d} \sigma_{k\ell} \partial_{x_k}, \qquad \ell \in \{1, \dots, d\}, \tag{3.2.16}$$

and their iteration, namely, for $\alpha \in \mathcal{M}$,

$$L^{\alpha} := L^{(\alpha_1)} \circ \cdots \circ L^{(\alpha_p)},$$

for a multi-index α with length $\mathfrak{l}(\alpha) := p$. By convention, $L^{\varnothing}$ is the identity operator, and denote $\alpha = (0, \cdots, 0) = (0)_p$ with $\mathfrak{l}(\alpha) = p$. We denote by $*$ the concatenation of two multi-indices namely $\alpha * \beta = (\alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q)$ with $p = \mathfrak{l}(\alpha)$ and $q = \mathfrak{l}(\beta)$. We denote by $\mathcal{G}_b^l$ the set of all functions $v : [0,T] \times \mathbb{R}^d \longmapsto \mathbb{R}$ for which $L^{\alpha} v$ is well defined, continuous and bounded for all muti-index $\alpha \in \{(\alpha_1, \cdots, \alpha_p) | 1 \leqslant p \leqslant l\}$.

In particular, we shall use the following assumption, for $p = 1, 2, 3$:
$(\mathrm{H}r)_p$: the value function $u \in \mathcal{G}_b^{p+1}$ and $f \in \mathcal{C}_b^p$.

The key to control the discrete-time error by a judicious choice of the scheme coefficients is to be able to expand the value function u along the approximation scheme X. To this end, we introduce the following assumption for $M \geqslant 1$:
$(\mathrm{H}X)_M$: the process X satisfies, for all $v \in \mathcal{G}_b^{M+1}$, $0 \leqslant n \leqslant N$, $1 \leqslant q \leqslant Q$, $k \leqslant q$, $1 \leqslant \ell \leqslant d$, denote $v^{\alpha} = L^{\alpha} v$,

$$\mathbb{E}_{t_{n,q}}[v(t_{n,k}, X_{n,k})] = \sum_{m=0}^{M} v^{(0)_m}(t_{n,q}, X_{n,q}) \frac{(\{c_q - c_k\}h)^m}{m!} + O_{t_{n,q}}(h^{M+1}), \tag{3.2.17}$$

$$\mathbb{E}_{t_{n,q}}\Big[(H_q^n)^{\ell} v(t_{n+1}, X_{n+1})\Big] = \sum_{m=0}^{M-1} v^{(\ell)*(0)_m}(t_{n,q}, X_{n,q}) \frac{(c_q h)^m}{m!} + O_{t_{n,q}}(h^{M+1}), \tag{3.2.18}$$

$$\mathbb{E}_{t_{n,q}}\Big[(H_{q,k}^n)^{\ell} v(t_{n+1}, X_{n+1})\Big] = \sum_{m=0}^{M-1} v^{(\ell)*(0)_m}(t_{n,q}, X_{n,q}) \frac{(\{c_q - c_k\}h)^m}{m!} + O_{t_{n,q}}(h^{M+1}), \tag{3.2.19}$$

where the Landau notation $O_t(r)$ means that for a random variable R such that $|R| \leqslant \lambda_t^h r$ with λ_t^h is a positive random variable satisfying

$$\mathbb{E}\Big[|\lambda_t^h|^p\Big] \leqslant C_p, \quad \forall p > 0, h > 0.$$

We note that (3.2.17) indicates that X is a weak approximation scheme of order M approximation [69, 24]. Conditions (3.2.18)-(3.2.19) are required to manage the error coming from the approximation of the Z-component.

Regarding the discrete-time error, our main result reads as follows. Define

$$(\bar{Y}_n, \bar{Z}_n) = (u(t_n, X_n), \sigma^{\top} \nabla_x u(t_n, X_n)), \tag{3.2.20}$$

and the global discrete time error as

$$\mathcal{T}_N = \max_{n<N} \mathbb{E}\big[|\bar{Y}_n - Y_n|^2\big] + h \sum_{n=0}^{N-1} \mathbb{E}\big[|\bar{Z}_n - Z_n|^2\big]. \tag{3.2.21}$$

As usual, we say that the scheme is of order $\mathfrak{a} \in [0, \infty)$ if $\mathcal{T}_N = O(h^{2\mathfrak{a}})$.

Theorem 3.2.1 *i) Euler scheme (one stage scheme): Assume* $(\mathrm{Hr})_1$ *and* $(\mathrm{H}X)_1$*, the global truncation error of Runge-Kutta scheme is at least order 1 if*

$$a_{21} + a_{22} = 1.$$

This condition leads to the explicit Euler scheme when $a_{21} = 1$ *and implicit Euler scheme when* $a_{22} = 1$*, respectively.*

ii) Crank-Nicolson scheme (one stage scheme): Assume $(\mathrm{Hr})_2$ *and* $(\mathrm{H}X)_2$*, the global truncation error of Runge-Kutta scheme is at least order 2 if*

$$a_{21} = a_{22} = \frac{1}{2} \qquad \text{and} \qquad \alpha_{21} = 1.$$

iii) Two stage explicit scheme: Assume $(\mathrm{Hr})_2$ *and* $(\mathrm{H}X)_2$*, the global truncation error of Runge-Kutta scheme is at least order 2 if* $a_{22} = a_{33} = 0$ *and*

$$a_{21} = c_2, \quad a_{31} = 1 - \frac{1}{2c_2}, \quad a_{32} = \frac{1}{2c_2}, \quad \text{and} \quad \alpha_{31} + \alpha_{32}\mathbb{1}_{\{c_2<1\}} = 1.$$

iv) Three stage explicit scheme: Assume $(\mathrm{Hr})_3$ *and* $(\mathrm{H}X)_3$*, the global truncation error of Runge-Kutta scheme is at least order 3 if* $0 < c_2 < c_3 \leqslant 1 (c_2 \neq \frac{2}{3}$ *when* $c_3 = 1), a_{22} = a_{33} = a_{44} = 0$ *and the following conditions holds true*

$$\begin{aligned}
&a_{41} + a_{42} + a_{43} = 1, \quad a_{42}c_2 + a_{43}c_3 = \frac{1}{2},\\
&a_{42}c_2^2 + a_{43}c_3^2 = \frac{1}{3}, \quad a_{43}a_{32}c_2 = a_{43}\alpha_{32}c_2 = \frac{1}{6},\\
&\alpha_{41} + \alpha_{42} + \alpha_{43}\mathbb{1}_{\{c_3<1\}} = 1, \quad \alpha_{42}c_2 + \alpha_{43}c_3\mathbb{1}_{\{c_3<1\}} = \frac{1}{2}.
\end{aligned}$$

Proof. 1. We give a short proof of the discrete time error upper bound for these most interesting schemes, as it is closed to the one given in [24].
Recall the definition of $(\bar{Y}_n, \bar{Z}_n)_{0\leqslant n\leqslant N}$ in (3.2.20). We observe that it can be written also as

$$\begin{aligned}
\bar{Y}_n &= \mathbb{E}_{t_n}\big[\bar{Y}_{n+1} + \Phi_n^Y(\bar{Y}_{n+1}, \bar{Z}_{n+1}, h)\big] + \check{\zeta}_n^Y,\\
\bar{Z}_n &= \mathbb{E}_{t_n}\big[H_{Q+1}^n\bar{Y}_{n+1} + \Phi_n^Z(\bar{Y}_{n+1}, \bar{Z}_{n+1}, h)\big] + \check{\zeta}_n^Z.
\end{aligned}$$

Now, thanks to assumption $(\mathrm{H}X)_M$ with $M = 1, 2, 3$, one can follow the computations made in Theorem 1.3 for statement (i)-(ii), Theorem 1.5 for statement (iii) or Theorem 1.6 for statement (iv) in [24] to obtain the corresponding upper bounds for the error linked to the perturbation, namely

$$\mathbb{E}\left[\frac{|\check{\zeta}_n^Y|^2}{h} + h|\check{\zeta}_n^Z|^2\right] = O(h^{2\mathfrak{a}+1}),$$

for $\mathfrak{a} = 1, 2$ or 3. The proof is then concluded using (3.2.11).

2. We give the proof with full computation for the Crank-Nicolson scheme only, see Section 3.5.2. □

Remark 3.2.1 *(i) See [24], contrary to the ODE case, since the scheme we considered are always explicit for Z-part, there exists an order barrier for implicit scheme to get an order $Q+1$ scheme with a $Q-$stage scheme when $Q > 1$ as long as $\partial_z f \neq 0$. Hence, we only consider the explicit scheme when $Q > 1$ as the implicit scheme has no advantage compared to the explicit scheme for general drivers f.*
(ii) For the explicit Runge-Kutta scheme, we can choose the coefficients such that $\alpha_{qk} = a_{qk}, \forall 1 \leqslant k < q \leqslant Q+1$.
(iii) When $Q = 3$, the algorithm converges too fast to lead to the discretization error smaller than the variance even the time steps N is very small, we can not observe the order of the algorithm. Hence, we don't consider the case with $Q > 3$. In fact, there is an order barrier when $Q > 3$ that is also why we do not study $Q \geqslant 4$, see also [24].

3.3 A learning method for Runge-Kutta schemes

In this section, we present an implementation of the Runge-Kutta schemes given in Definition 3.2.1, which is particularly well suited for non-linear regression. It extends the method proposed in [65] to more efficient schemes, in term of discrete-time error. We start then by recalling one approach developed in [65] for the implicit Euler scheme (the **DBDP1** scheme). We then present the case of the Crank-Nicolson scheme which reveals the necessary extension to consider. We present then the method for two stage and three stage explicit scheme that will be illustrated numerically in Section 3.4. We conclude this section by discussing general Runge-Kutta schemes and proving some error control.

3.3.1 Euler scheme

We first recall the implementation of the implicit Euler scheme in [65] using deep neural networks. Our aim is to obtain a similar implementation for general Runge-Kutta scheme. In this section only, we assume that $(X_n)_{n\leqslant N}$ is given by the classical Euler scheme on π:

$$X_{n+1} = X_n + \mu(X_n)h + \sigma(X_n)\Delta W_n \,, n \leqslant N-1 \,, \tag{3.3.1}$$

with $\Delta W_n := W_{t_{n+1}} - W_{t_n}$ and $X_0 = \mathcal{X}_0$.

The implicit Euler scheme of BSDEs [93, 16] reads classically as follows, for $0 \leqslant n < N$,

$$Y_n = \mathbb{E}_{t_n}[Y_{n+1} + hf(X_n, Y_n, Z_n))] \text{ and } Z_n = \mathbb{E}_{t_n}\left[\frac{\Delta W_n}{h} Y_{n+1}\right], \tag{3.3.2}$$

where we set $H_n = \frac{W_{t_{n+1}} - W_{t_n}}{h}$. Though implicit in the Y-component, the scheme is well-posed for h small enough as f is Lipschitz continuous. Numerous methods have been considered to compute the scheme in practice [26, 25, 15, 65, 22, 34]. However, in the numerical section of this work, we rely on the high representative power of Deep Neural Network (recall the Universal Representation Theorem 3.1.1). Using this class of functions, a first approach could be, for parameters (θ, ϑ),

associated mappings $x \mapsto \mathcal{U}(x,\theta)$, $x \mapsto \mathcal{V}(x,\vartheta)$, given $(\mathcal{U}(X_N,\theta_N^\star),\mathcal{V}(X_N,\vartheta_N^\star)) = (g(X_N),\sigma^\top(X_N)\partial_x g(X_N))$, to find the parameters by solving the following optimisation problems: for $0 \leqslant n \leqslant N-1$,

$$\vartheta_n^\star = \text{argmin}_\vartheta \mathbb{E}\left[\left|\frac{\Delta W_n}{h}\mathcal{U}(X_{n+1},\theta_{n+1}^\star) - \mathcal{V}(X_n,\vartheta)\right|^2\right], \tag{3.3.3}$$

and then

$$\theta_n^\star = \text{argmin}_\theta \mathbb{E}\Big[\big|\mathcal{U}(X_{n+1},\theta_{n+1}^\star) - \{\mathcal{U}(X_n,\theta) - hf(\mathcal{U}(X_n,\theta),\mathcal{V}(X_n,\vartheta_n^\star))\}\big|^2\Big]. \tag{3.3.4}$$

However, a better empirical approach, introduced in [65] is to optimise in "one go" θ and ϑ at each step, namely

$$\begin{aligned}(\theta_n^\star,\vartheta_n^\star) = \text{argmin}_{\theta,\vartheta}\mathbb{E}\Big[\Big|&\mathcal{U}(X_{n+1},\theta_{n+1}^\star) \\ &-\{\mathcal{U}(X_n,\theta) - hf(\mathcal{U}(X_n,\theta),\mathcal{V}(X_n,\vartheta_n^\star)) + \mathcal{V}(X_n,\vartheta_n^\star)\Delta W_n\}\Big|^2\Big],\end{aligned} \tag{3.3.5}$$

The previous procedure is named **DBDP1** in [65]. One observes that it is possible to use only one network in the above optimization. This will also save half of the computational time cost.

The above scheme is jutified by the following observation.

Lemma 3.3.1 *Under our standing assumptions,*

$$(Y_n,Z_n) = \text{argmin}_{y,z\in\mathcal{L}^2(\mathcal{F}_{t_n})}\mathbb{E}\big[|Y_{n+1} - (y - hf(X_n,y,z) + z\Delta W_n)|^2\big]. \tag{3.3.6}$$

Proof. We first observe that

$$Y_{n+1} = Y_n - hf(X_n,Y_n,Z_n) + Z_n\Delta W_n - \Delta M_n, \tag{3.3.7}$$

where $\mathbb{E}_{t_n}[\Delta M_n] = \mathbb{E}_{t_n}[\Delta M_n \Delta W_n] = 0, \mathbb{E}_{t_n}\big[|\Delta M_n|^2\big] < \infty.$, so that

$$\begin{aligned}&\mathbb{E}\big[|Y_{n+1} - (y - hf(X_n,y,z) + z\Delta W_n)|^2\big] \\ &= \mathbb{E}\big[|Y_n - hf(X_n,Y_n,Z_n) - \{y - hf(X_n,y,z)\} + (Z_n - z)\Delta W_n - \Delta M_n|^2\big] \\ &= \mathbb{E}\big[|Y_n - hf(X_n,Y_n,Z_n) - \{y - hf(X_n,y,z)\}|^2\big] + h\mathbb{E}\big[|Z_n - z|^2\big] + \mathbb{E}\big[|\Delta M_n|^2\big].\end{aligned}$$

Obviously, (Y_n,Z_n) does achieve the minimum of the right side of the above equation. Reciprocally, any optimal solution $(y^\star,z^\star)$ must satisfy $z^\star = Z_n$ from the second term of the right side in the above equality. Moreover, necessarily one has

$$\begin{aligned}y^\star &= \mathbb{E}_{t_n}[Y_n - hf(X_n,Y_n,Z_n) + hf(X_n,y^\star,Z_n)] \\ &= \mathbb{E}_{t_n}[Y_{n+1} + hf(X_n,y^\star,Z_n)]\ .\end{aligned}$$

By uniqueness of the scheme definition, we conclude $y^\star = Y_n$. □

Let us define rigorously the scheme. First, we introduce the loss function, for $\varphi \in \mathcal{C}(\mathbb{R}^d, \mathbb{R})$:

$$\begin{aligned} &\mathrm{L}_{\mathrm{n}}^{\mathrm{EUi}}[\varphi](\theta) \qquad (3.3.8)\\ &:= \mathbb{E}\Big[\Big|\varphi(X_{n+1}) - \big\{\mathcal{U}(X_n;\theta) - hf(X_n, \mathcal{U}(X_n;\theta), \mathcal{V}(X_n;\theta)) + \mathcal{V}(X_n;\theta)\Delta W_n\big\}\Big|^2\Big], \end{aligned}$$

with $(\mathcal{U}, \mathcal{V}) := \mathcal{N}_m \in \mathcal{S}^{\rho}_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12).

The implemented implicit Euler scheme is then given by

Definition 3.3.1 (Implemented implicit Euler scheme) *The numerical solution is computed using the following step:*

- *For $n = N$, initialize $\hat{\mathcal{U}}_N = g, \hat{\mathcal{V}}_N = \sigma^\top \nabla_x g$.*
- *For $n = N-1, \cdots, 1, 0$, given $\hat{\mathcal{U}}_{n+1}$,*
  - *Compute a minimizer of the loss function:*
  $$\theta_n^\star \in \mathrm{argmin}_\theta \, \mathrm{L}_{\mathrm{n}}^{\mathrm{EUi}}[\hat{\mathcal{U}}_{n+1}](\theta), \qquad (3.3.9)$$
  recall (3.3.8).
  - *Set $(\hat{\mathcal{U}}_n, \hat{\mathcal{V}}_n) := \mathcal{N}_m(\cdot;\theta_n^\star) \in \mathcal{S}^{\rho}_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12).*

In the following, we build on this approach to obtain implementations of the theoretical Runge-Kutta schemes given in Definition 3.2.1. For sake of completeness, let us also mention that the explicit Euler scheme could be considered instead, with essentially the same empirical result, see Section 3.4. Namely, for $(\varphi, \psi) \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}) \times \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$, we introduce the loss function at step $n < N$:

$$\begin{aligned} \mathrm{L}_{\mathrm{n}}^{\mathrm{EUe}}[\varphi, \psi](\theta) := \mathbb{E}\Big[\Big|&\varphi(X_{n+1}) + hf(X_{n+1}, \varphi(X_{n+1}), \psi(X_{n+1}))\\ &- \big\{\mathcal{U}(X_n;\theta) + \mathcal{V}(X_n;\theta)\Delta W_n\big\}\Big|^2\Big] \qquad (3.3.10) \end{aligned}$$

with $(\mathcal{U}, \mathcal{V}) := \mathcal{N}_m \in \mathcal{S}^{\rho}_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12).

Similarly, we have the following lemma and the definition of implemented explicit Euler scheme.

Lemma 3.3.2 *Under our standing assumptions,*

$$(Y_n, Z_n) = \mathrm{argmin}_{y,z \in \mathcal{L}^2(\mathcal{F}_{t_n})} \mathbb{E}\big[|Y_{n+1} - (y - hf(X_{n+1}, Y_{n+1}, Z_{n+1}) + z\Delta W_n)|^2\big]. \qquad (3.3.11)$$

Definition 3.3.2 (Implemented explicit Euler scheme) *The numerical solution is computed using the following step:*

- *For $n = N$, initialize $\hat{\mathcal{U}}_N = g, \hat{\mathcal{V}}_N = \sigma^\top \nabla_x g$, $\hat{\mathcal{A}}_N = 0$.*
- *For $n = N-1, \cdots, 1, 0$, given $\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}$,*
  - *Compute a minimizer of the loss function:*
  $$\theta_n^\star \in \mathrm{argmin}_\theta \, \mathrm{L}_{\mathrm{n}}^{\mathrm{EUe}}[\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}](\theta), \qquad (3.3.12)$$
  recall (3.3.10).
  - *Set $(\hat{\mathcal{U}}_n, \hat{\mathcal{V}}_n) := \mathcal{N}_m(\cdot;\theta_n^\star) \in \mathcal{S}^{\rho}_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12).*

3.3.2 Crank-Nicolson scheme

We now turn to the study of the Crank-Nicolson scheme for BSDEs. It is a one stage scheme which belongs to the class given in Definition 3.2.1. It has been introduced in [32], where it is implemented using cubature methods and tree based branching algorithm(TBBA). Other θ-scheme entering into this class will not be considered here as there are suboptimal in terms of discrete-time error bound.

The scheme reads as follows. For the $Y-$part, it is the usual Crank-Nicolson scheme, namely

$$\begin{cases} Y_N &= g(X_N), \\ Y_n &= \mathbb{E}_{t_n}\big[Y_{n+1} + \frac{h}{2}(f(X_n, Y_n, Z_n) + f(X_{n+1}, Y_{n+1}, Z_{n+1}))\big], \quad 0 \leqslant n \leqslant N-1, \end{cases} \tag{3.3.13}$$

and for $Z-$part,

$$\begin{cases} Z_N = \sigma^\top \nabla_x g(X_N), \\ Z_n = \mathbb{E}_{t_n}[H_n(Y_{n+1} + hf(X_{n+1}, Y_{n+1}, Z_{n+1}))], \end{cases} \tag{3.3.14}$$

where $H_n \in \mathbb{R}^d$ is a $\mathcal{F}_{t_{n+1}}$-mesurable random variable, satisfying (3.2.4)-(3.2.5).

In order to suggest an implementation using non-linear regression mimicking (3.3.6) or (3.3.11) for the Euler scheme, we first make the following observation.

Lemma 3.3.3 *For $0 \leqslant n \leqslant N-1$, set $A_n := -\frac{1}{2}\mathbb{E}_{t_n}[f(X_{n+1}, Y_{n+1}, Z_{n+1})hH_n]$. Then, (Y_n, Z_n, A_n) is the unique solution to the following optimisation problem*

$$\begin{aligned} \min_{y,z,a \in \mathcal{L}^2(\mathcal{F}_{t_n})} L^n(y,z,a) := \mathbb{E}\Bigg[\Bigg|Y_{n+1} - \Bigg\{y - \frac{h}{2}\left(f(X_n, y, z) + f(X_{n+1}, Y_{n+1}, Z_{n+1})\right) \\ + (z+a)\frac{H_n}{\upsilon_n}\Bigg\}\Bigg|^2\Bigg] + C_0 h \mathbb{E}\left[\left|\frac{1}{2}hH_n f(X_{n+1}, Y_{n+1}, Z_{n+1}) + a\right|^2\right], \end{aligned} \tag{3.3.15}$$

where $C_0 > 0$ is a constant and $\upsilon_n = \mathbb{E}_{t_n}\big[|H_n|^2\big]$.

Proof. We first observe that

$$Y_{n+1} = Y_n - \frac{h}{2}\{f(X_n, Y_n, Z_n) + f(X_{n+1}, Y_{n+1}, Z_{n+1})\} + (Z_n + A_n)\frac{H_n}{\upsilon_n} - \Delta M_n, \tag{3.3.16}$$

where $\mathbb{E}_{t_n}[\Delta M_n] = \mathbb{E}_{t_n}[\Delta M_n H_n] = 0, \mathbb{E}_{t_n}\big[|\Delta M_n|^2\big] < \infty$. From the very definition of A_n,

$$\begin{aligned} &\mathbb{E}\left[\left|\frac{1}{2}hf(X_{n+1}, Y_{n+1}, Z_{n+1})H_n + a\right|^2\right] \\ &= \mathbb{E}\left[\left|\frac{1}{2}hf(X_{n+1}, Y_{n+1}, Z_{n+1})H_n + A_n\right|^2 + |A_n - a|^2\right]. \end{aligned}$$

Inserting (3.3.16) into the definition of $L^n(y,z,a)$ and using the previous equality, we compute

$$\begin{aligned}
&L^n(y,z,a)\\
&= \mathbb{E}\Big[\big|Y_n - \frac{h}{2}f(X_n,Y_n,Z_n) - \{y - \frac{h}{2}f(X_n,y,z)\} + (Z_n - z + (A_n - a))\frac{H_n}{\upsilon_n} - \Delta M_n\big|^2\Big]\\
&+ \mathbb{E}\Big[\Big|\frac{1}{2}hf(X_{n+1},Y_{n+1},Z_{n+1})H_n + A_n\Big|^2 + |A_n - a|^2\Big]\\
&= \tilde{L}_1^n(y,z) + \tilde{L}_2^n(a,z) + \tilde{L}_3^n(a) + \ell_n,
\end{aligned}$$

where

$$\tilde{L}_1^n(y,z) = \mathbb{E}\Big[\big|Y_n - \frac{h}{2}f(X_n,Y_n,Z_n) - \{y - \frac{h}{2}f(X_n,y,z)\}\big|^2\Big], \tag{3.3.17}$$

$$\tilde{L}_2^n(a,z) = \frac{1}{\upsilon_n}\mathbb{E}\big[|Z_n - z + A_n - a|^2\big], \tag{3.3.18}$$

$$\tilde{L}_3^n(a) = C_0h\mathbb{E}\big[|A_n - a|^2\big], \tag{3.3.19}$$

$$\ell_n = \mathbb{E}\Big[|\Delta M_n|^2 + C_0h\Big|\frac{1}{2}hf(X_{n+1},Y_{n+1},Z_{n+1})H_n + A_n\Big|^2\Big]. \tag{3.3.20}$$

We then observe that $\tilde{L}_3^n(A_n) = 0$, $\tilde{L}_2^n(A_n,Z_n) = 0$ and $\tilde{L}_1^n(Y_n,Z_n) = 0$, so that (Y_n,Z_n,A_n) does achieve the minimum of L^n.
Reciprocally, any optimal solution $(y^\star,z^\star,a^\star)$ must satisfy $\tilde{L}_3^n(a^\star) = 0, \tilde{L}_2^n(a^\star,z^\star) = 0$, which implies $a^\star = A_n, z^\star = Z_n$. Moreover, necessarily one has $\tilde{L}_1^n(y^\star,Z_n) = 0$, then using (3.3.13), we find

$$y^\star = \mathbb{E}_{t_n}\Big[Y_{n+1} + \frac{h}{2}(f(X_n,y^\star,Z_n) + f(X_{n+1},Y_{n+1},Z_{n+1}))\Big].$$

By uniqueness of the scheme definition, we conclude $y^\star = Y_n$. □

The previous result indicates how to adapt (3.3.5) for the Crank-Nicolson scheme. The implemented scheme will be given by iterations of the following optimisation problems.

Let $(\varphi,\psi) \in \mathcal{C}(\mathbb{R}^d,\mathbb{R}) \times \mathcal{C}(\mathbb{R}^d,\mathbb{R}^d)$, we introduce the loss function at step $n < N$:

$$\begin{aligned}
\mathrm{L}_{\mathrm{n}}^{\mathrm{CN}}[\varphi,\psi](\theta) := \mathbb{E}\Big[\Big|&\varphi(X_{n+1}) - \big\{\mathcal{U}(X_n;\theta) - \frac{h}{2}f(X_{n+1},\varphi(X_{n+1}),\psi(X_{n+1}))\\
&- \frac{h}{2}f(X_n,\mathcal{U}(X_n;\theta),\mathcal{V}(X_n;\theta)) + (\mathcal{V}(X_n;\theta) + \mathcal{A}(X_n;\theta))\frac{H_n}{\upsilon_n}\big\}\Big|^2\\
&+ C_0h\Big|\frac{h}{2}f(X_{n+1},\varphi(X_{n+1}),\psi(X_{n+1}))H_n + \mathcal{A}(X_n;\theta)\Big|^2\Big],
\end{aligned} \tag{3.3.21}$$

with $(\mathcal{U},\mathcal{V},\mathcal{A}) := \mathcal{N}_m \in \mathcal{S}^\rho_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12).

We now define the scheme which is implemented in practice. Each optimisation is done using SGD algorithm, see Section 3.4 for details.

Definition 3.3.3 (Implemented Crank-Nicolson scheme) *The numerical solution is computed using the following step:*

- *For $n = N$, initialize $\hat{\mathcal{U}}_N = g, \hat{\mathcal{V}}_N = \sigma^\top \nabla_X g$, $\hat{\mathcal{A}}_N = 0$.*
- *For $n = N-1, \cdots, 1, 0$, given $\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}$,*
  - *Compute a minimizer of the loss function:*

$$\theta_n^\star \in \operatorname{argmin}_\theta \mathrm{L}_{\mathrm{n}}^{\mathrm{CN}}[\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}](\theta), \tag{3.3.22}$$

  recall (3.3.21).
  - *Set $(\hat{\mathcal{U}}_n, \hat{\mathcal{V}}_n, \hat{\mathcal{A}}_n) := \mathcal{N}_m(\cdot; \theta_n^\star) \in \mathcal{S}^\rho_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12).*

Remark 3.3.1 *i) Note carefully that we don't have any theoretical guarantees that the minimization problems (3.3.9), (3.3.12), (3.3.22) are well-posed on the whole space $\mathbb{R}^{N_m}$. A sufficient (but far from necessary) condition is the convexity and coercivity ($\lim_{|\theta|\to\infty} L_n[\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}](\theta) = +\infty$) of the objective function. However, in practical implementation, we restrict the domain to a compact subset of $\mathbb{R}^{N_m}$, so that the minimization problems are always well-posed. This remark also applies to the explicit Runge-Kutta schemes in the following section.*

ii) Lemma 3.3.5 below shows that the minimal possible loss $\mathrm{L}_{\mathrm{n}}^{\mathrm{CN}}[\varphi, \psi]$ is controlled in $O(h^2)$ under some good conditions. This minimum loss has no influence on the study of the weak error we perform here. However, it has an impact when doing the Monte Carlo simulation.

iii) The value of the C_0, that we called the balance number *C_0, recall (3.3.21), impacts the numerical results, see Figure 3.1 in Section 3.4.*

iv) For the numerical part, we can use the following loss function instead of (3.3.21) in order to reduce the variance of $A-$part by the control variate technique, it has no influence on the theoretical part.

$$\begin{aligned}\mathrm{L}_{\mathrm{n,R}}^{\mathrm{CN}}[\varphi, \psi](\theta) := \mathbb{E}\Big[\Big|\varphi(X_{n+1}) - \big\{\mathcal{U}(X_n;\theta) - \frac{h}{2} f(X_{n+1}, \varphi(X_{n+1}), \psi(X_{n+1}))\\ - \frac{h}{2} f(X_n, \mathcal{U}(X_n;\theta), \mathcal{V}(X_n;\theta)) + (\mathcal{V}(X_n;\theta) + \mathcal{A}(X_n;\theta))\frac{H_n}{\upsilon_n}\big\}\Big|^2\\ + C_0 h \Big|\frac{h}{2}\left(f(X_{n+1}, \varphi(X_{n+1}), \psi(X_{n+1})) - f(X_n, \varphi(X_n), \psi(X_n))\right) H_n\\ + \quad \mathcal{A}(X_n;\theta)\Big|^2\Big].\end{aligned} \tag{3.3.23}$$

We now give an estimation of the (theoretical) error implied by the non-linear regression procedure for one time step.

Lemma 3.3.4 *Assume that*

$$\theta^\star \in \operatorname{argmin}_\theta \mathrm{L}_{\mathrm{n}}^{\mathrm{CN}}[\varphi, \psi](\theta),$$

and define

$$Y_n^{(\varphi,\psi)} := \mathbb{E}_{t_n}\Big[Y_{n+1}^{(\varphi,\psi)} + \frac{h}{2}\Big(f(X_{n+1}, Y_{n+1}^{(\varphi,\psi)}, Z_{n+1}^{(\varphi,\psi)}) + f(X_n, Y_n^{(\varphi,\psi)}, Z_n^{(\varphi,\psi)})\Big)\Big] \tag{3.3.24}$$

$$Z_n^{(\varphi,\psi)} := \mathbb{E}_{t_n}\Big[H_n\Big(Y_{n+1}^{(\varphi,\psi)} + hf(X_{n+1}, Y_{n+1}^{(\varphi,\psi)}, Z_{n+1}^{(\varphi,\psi)})\Big)\Big] \tag{3.3.25}$$

$$A_n^{(\varphi,\psi)} := -\frac{h}{2}\mathbb{E}_{t_n}\Big[H_n\Big(f(X_{n+1}, Y_{n+1}^{(\varphi,\psi)}, Z_{n+1}^{(\varphi,\psi)})\Big)\Big] \tag{3.3.26}$$

with $(Y_{n+1}^{(\varphi,\psi)}, Z_{n+1}^{(\varphi,\psi)}) := (\varphi(X_{n+1}), \psi(X_{n+1}))$. *Then, the following holds*

$$\begin{aligned}&\mathbb{E}\Big[|Y_n^{(\varphi,\psi)} - \mathcal{U}_n(X_n;\theta^\star)|^2 + h|Z_n^{(\varphi,\psi)} - \mathcal{V}_n(X_n;\theta^\star)|^2 + h|A_n^{(\varphi,\psi)} - \mathcal{A}_n(X_n;\theta^\star)|^2\Big]\\ &\leqslant C\mathcal{E}_n(\varphi,\psi)\end{aligned} \tag{3.3.27}$$

where

$$\mathcal{E}_n(\varphi,\psi) = \epsilon_n^{\mathcal{N},y}(\varphi,\psi) + h\epsilon_n^{\mathcal{N},z}(\varphi,\psi) + h\epsilon_n^{\mathcal{N},a}(\varphi,\psi) \tag{3.3.28}$$

and

$$\epsilon_n^{\mathcal{N},y}(\varphi,\psi) := \inf_{\theta^y}\mathbb{E}\Big[|Y_n^{(\varphi,\psi)} - \mathcal{U}_n(X_n;\theta^y)|^2\Big], \tag{3.3.29}$$

$$\epsilon_n^{\mathcal{N},a}(\varphi,\psi) := \inf_{\theta^a}\mathbb{E}\Big[|A_n^{(\varphi,\psi)} - \mathcal{A}_n(X_n;\theta^a)|^2\Big], \tag{3.3.30}$$

$$\epsilon_n^{\mathcal{N},z}(\varphi,\psi) := \inf_{\theta^z}\mathbb{E}\Big[|Z_n^{(\varphi,\psi)} - \mathcal{V}_n(X_n;\theta^z)|^2\Big]. \tag{3.3.31}$$

Proof. 1. We first observe that (3.3.24) rewrites

$$\begin{aligned}Y_{n+1}^{(\varphi,\psi)} = Y_n^{(\varphi,\psi)} &- \frac{h}{2}\{f(X_n, Y_n^{(\varphi,\psi)}, Z_n^{(\varphi,\psi)}) + f(X_{n+1}, Y_{n+1}^{(\varphi,\psi)}, Z_{n+1}^{(\varphi,\psi)})\}\\ &+ \Big(Z_n^{(\varphi,\psi)} + A_n^{(\varphi,\psi)}\Big)\frac{H_n}{\upsilon_n} + \Delta M_n^{(\varphi,\psi)},\end{aligned} \tag{3.3.32}$$

where $\mathbb{E}_{t_n}\Big[\Delta M_n^{(\varphi,\psi)}\Big] = \mathbb{E}_{t_n}\Big[\Delta M_n^{(\varphi,\psi)} H_n\Big] = 0, \mathbb{E}_{t_n}\Big[|\Delta M_n^{(\varphi,\psi)}|^2\Big] < \infty$. Following the same computations as in the proof of Lemma 3.3.3, we obtain that

$$\mathrm{L}_{\mathrm{n}}^{\mathrm{CN}}[\varphi,\psi](\theta) = \tilde{L}_1^n[\varphi,\psi](\theta) + \tilde{L}_2^n[\varphi,\psi](\theta) + \tilde{L}_3^n[\varphi,\psi](\theta) + \ell_n[\varphi,\psi] \tag{3.3.33}$$

with

$$\begin{aligned}\tilde{L}_1^n[\varphi,\psi](\theta) = \mathbb{E}\Bigg[&|Y_n^{(\varphi,\psi)} - \frac{h}{2}f(X_n, Y_n^{(\varphi,\psi)}, Z_n^{(\varphi,\psi)})\\ &- \{\mathcal{U}_n(X_n;\theta) - \frac{h}{2}f(X_n, \mathcal{U}_n(X_n;\theta), \mathcal{V}_n(X_n;\theta))\}|^2\Bigg],\end{aligned} \tag{3.3.34}$$

$$\tilde{L}_2^n[\varphi,\psi](\theta) = \frac{1}{\upsilon_n}\mathbb{E}\Big[|Z_n^{(\varphi,\psi)} - \mathcal{V}_n(X_n;\theta) + A_n^{(\varphi,\psi)} - \mathcal{A}_n(X_n;\theta)|^2\Big], \tag{3.3.35}$$

$$\tilde{L}_3^n[\varphi,\psi](\theta) = C_0 h\mathbb{E}\Big[|A_n^{(\varphi,\psi)} - \mathcal{A}_n(X_n;\theta)|^2\Big], \tag{3.3.36}$$

$$\ell_n[\varphi,\psi] = \mathbb{E}\Bigg[|\Delta M_n^{(\varphi,\psi)}|^2 + C_0 h\Big|\frac{1}{2}hf(X_{n+1}, Y_{n+1}^{(\varphi,\psi)}, Z_{n+1}^{(\varphi,\psi)})H_n + A_n^{(\varphi,\psi)}\Big|^2\Bigg]. \tag{3.3.37}$$

Setting $\tilde{L}_n[\varphi,\psi](\theta) := \tilde{L}_1^n[\varphi,\psi](\theta) + \tilde{L}_2^n[\varphi,\psi](\theta) + \tilde{L}_3^n[\varphi,\psi](\theta)$, we then deduce that

$$\operatorname{argmin}_\theta \tilde{L}_n[\varphi,\psi](\theta) = \operatorname{argmin}_\theta \mathrm{L}_\mathrm{n}^{\mathrm{CN}}[\varphi,\psi](\theta) \tag{3.3.38}$$

2. Using the Lipschitz continuity of f and recalling (3.2.4)-(3.2.5), we obtain the following upper bound:

$$\begin{aligned}\tilde{L}_n[\varphi,\psi](\theta) \leqslant C\mathbb{E}\Big[&|Y_n^{(\varphi,\psi)} - \mathcal{U}_n(X_n;\theta)|^2 + h|Z_n^{(\varphi,\psi)} - \mathcal{V}_n(X_n;\theta)|^2 \\ &+h|A_n^{(\varphi,\psi)} - \mathcal{A}_n(X_n;\theta)|^2\Big] .\end{aligned} \tag{3.3.39}$$

We now prove a lower bound for the previous quantity. First, we observe that, for any $0 < \alpha < 1$,

$$(x+y)^2 \geqslant x^2(1-\alpha) + y^2(1-\frac{1}{\alpha}) \tag{3.3.40}$$

Thus, for any α such that $1 > \alpha > \frac{2}{2+\Lambda C_0}$, we obtain

$$\begin{aligned}\tilde{L}_n^2[\varphi,\psi](\theta) + \tilde{L}_n^3[\varphi,\psi](\theta) \geqslant\ & h\frac{1-\alpha}{\Lambda}\mathbb{E}\Big[|Z_n^{(\varphi,\psi)} - \mathcal{V}_n(X_n;\theta)|^2\Big] \\ &+ h\frac{C_0}{2}\mathbb{E}\Big[|A_n^{(\varphi,\psi)} - \mathcal{A}_n(X_n;\theta)|^2\Big] .\end{aligned} \tag{3.3.41}$$

Using again (3.3.40), we get

$$\begin{aligned}\tilde{L}_n^1[\varphi,\psi](\theta) \geqslant\ & \frac{1}{2}\mathbb{E}\Big[|Y_n^{(\varphi,\psi)} - \mathcal{U}_n(X_n,\theta)|^2\Big] \\ &- \frac{h^2}{4}\mathbb{E}\Big[|f(X_n, Y_n^{(\varphi,\psi)}, Z_n^{(\varphi,\psi)}) - f(X_n, \mathcal{U}_n(X_n;\theta), \mathcal{V}_n(X_n;\theta))|^2\Big]\end{aligned}$$

Since f is Lipschitz continuous, we compute

$$\tilde{L}_n^1[\varphi,\psi](\theta) \geqslant (\frac{1}{2} - \frac{L^2h^2}{2})\mathbb{E}\Big[|Y_n^{(\varphi,\psi)} - \mathcal{U}_n(X_n,\theta)|^2\Big] - \frac{L^2h^2}{2}\mathbb{E}\Big[|Z_n^{(\varphi,\psi)} - \mathcal{V}_n(X_n,\theta)|^2\Big].$$

This leads, combined with (3.3.41), for h small enough, to

$$\begin{aligned}&\mathbb{E}\Big[|Y_n^{(\varphi,\psi)} - \mathcal{U}_n(X_n;\theta)|^2 + h|Z_n^{(\varphi,\psi)} - \mathcal{V}_n(X_n;\theta)|^2 + h|A_n^{(\varphi,\psi)} - \mathcal{A}_n(X_n;\theta)|^2\Big] \\ &\leqslant C\tilde{L}_n[\varphi,\psi](\theta).\end{aligned} \tag{3.3.42}$$

3. The above inequality is *a fortiori* true at the optimum $\theta^\star$. Moreover, optimizing on separated networks is always more costly than optimizing on a fully connected network thus leading to (3.3.27). □

Lemma 3.3.5 *Assume* $(\mathrm{H}X)_1$ *and* $\varphi,\psi \in \mathcal{G}_b^2$, *then the following holds*

$$\ell_n[\varphi,\psi] \leqslant C_\varphi h^2,$$

recall (3.3.37).

Proof. Observe that

$$\begin{aligned}\varphi(X_{n+1}) &= \varphi(X_n + \mu(X_n)h + \sigma(X_n)\Delta W_n)\\ &= \varphi(X_n) + \partial_x\varphi(X_n)(\mu(X_n)h + \sigma(X_n)\Delta W_n) + O(|\mu(X_n)h + \sigma(X_n)\Delta W_n|^2).\end{aligned}$$

Recalling (3.3.32), we deduce

$$\varphi(X_n) + \partial_x\varphi(X_n)(\mu(X_n)h + \sigma(X_n)\Delta W_n) + O(|\mu(X_n)h + \sigma(X_n)\Delta W_n|^2) = Y_n^{(\varphi,\psi)} + \frac{h}{2}f(X_{n+1}, Y_{n+1}^{(\varphi,\psi)}, Z_{n+1}^{(\varphi,\psi)}) - \frac{h}{2}f(X_n, Y_n^{(\varphi,\psi)}, Z_n^{(\varphi,\psi)}) + \left(Z_n^{(\varphi,\psi)} + A_n^{(\varphi,\psi)}\right)\Delta W_n + \Delta M_n^{(\varphi,\psi)}\,.$$

Reorganizing the above equality, we get

$$\begin{aligned}&\left(\partial_x\varphi(X_n)\mu(X_n) + \frac{1}{2}f(X_{n+1}, Y_{n+1}^{(\varphi,\psi)}, Z_{n+1}^{(\varphi,\psi)})\right)h + O(|\mu(X_n)h + \sigma(X_n)\Delta W_n|^2)\\ &= Y_n^{(\varphi,\psi)} - \varphi(X_n) - \frac{h}{2}f(X_n, Y_n^{(\varphi,\psi)}, Z_n^{(\varphi,\psi)})\\ &+ \left(Z_n^{(\varphi,\psi)} + A_n^{(\varphi,\psi)} - \partial_x\varphi(X_n)\sigma(X_n)\right)\Delta W_n + \Delta M_n^{(\varphi,\psi)}\,.\end{aligned}$$

Squaring the previous estimate, taking expectation and recalling the orthogonality property of $(1, \Delta W_n, \Delta M_n^{(\varphi,\psi)})$, one gets

$$\mathbb{E}\Big[|\Delta M_n^{(\varphi,\psi)}|^2\Big] \leqslant C_\varphi(\mathbb{E}\big[|\Delta W_n|^4\big] + h^2) \leqslant C_\varphi h^2.$$

One also computes that $\mathbb{E}\Big[|\frac{1}{2}f(X_{n+1}, Y_{n+1}^{(\varphi,\psi)}, Z_{n+1}^{(\varphi,\psi)})\Delta W_n + A_n^{(\varphi,\psi)}|^2\Big] \leqslant Ch$, which combined with the previous upper bound, concludes the proof. □

3.3.2.1 Pseudo-consistency of the implemented scheme

Proposition 3.3.1 *Assume that the scheme given in Definition 3.3.3 is well-posed, then*

$$\max_{n\leqslant N}\mathbb{E}\Big[|Y_n - \hat{\mathcal{U}}_n(X_n)|^2\Big] + h\sum_{n=0}^{N}\mathbb{E}\Big[|Z_n - \hat{\mathcal{V}}_n(X_n)|^2\Big] \leqslant CN\sum_{n=0}^{N-1}\bar{\mathcal{E}}_n\,, \tag{3.3.43}$$

where $\bar{\mathcal{E}}_n := \mathcal{E}_n(\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1})$, *recall* (3.3.28).

Proof. Let us define, for $n < N$, recalling (3.3.24)-(3.3.25)-(3.3.26) and Definition 3.3.3

$$\bar{Y}_n = Y_n^{(\hat{\mathcal{U}}_{n+1},\hat{\mathcal{V}}_{n+1})}\,,\ \bar{Z}_n = Z_n^{(\hat{\mathcal{U}}_{n+1},\hat{\mathcal{V}}_{n+1})}\,,\ \bar{A}_n = A_n^{(\hat{\mathcal{U}}_{n+1},\hat{\mathcal{V}}_{n+1})}\,. \tag{3.3.44}$$

We first observe that $(\hat{U}_n, \hat{V}_n) := (\mathcal{U}_n(X_n), \mathcal{V}_n(X_n))$ can be rewritten as a perturbed scheme, namely

$$\hat{U}_n = \mathbb{E}_{t_n}\left[\hat{U}_{n+1} + \frac{h}{2}\left(f(X_{n+1}, \hat{U}_{n+1}, \hat{V}_{n+1}) + f(X_n, \hat{U}_n, \hat{V}_n)\right)\right] + \zeta_n^y, \tag{3.3.45}$$

$$\hat{V}_n = \mathbb{E}_{t_n}\left[H_n\left(\hat{U}_{n+1} + hf(X_{n+1}, \hat{U}_{n+1}, \hat{V}_{n+1})\right)\right] + \zeta_n^z, \tag{3.3.46}$$

with

$$\zeta_n^y := \hat{U}_n - \bar{Y}_n + \frac{h}{2}\Big(f(X_n, \bar{Y}_n, \bar{Z}_n) - f(X_n, \hat{U}_n, \hat{V}_n)\Big) \quad \text{and} \quad \zeta_n^z := \hat{V}_n - \bar{Z}_n\,. \tag{3.3.47}$$

Indeed, with our notations, we have $(\hat{U}_{n+1}, \hat{V}_{n+1}) = (Y_{n+1}^{\hat{\mathcal{U}}_{n+1},\hat{\mathcal{V}}_{n+1}}, Z_{n+1}^{\hat{\mathcal{U}}_{n+1},\hat{\mathcal{V}}_{n+1}})$. Moreover, since $\hat{U}_n = \hat{\mathcal{U}}_n(X_n) = \mathcal{U}_n(X_n, \theta_n^\star)$, recall Definition 3.3.3, we have

$$\begin{aligned}\mathbb{E}\Big[\frac{1}{h}|\zeta_n^y|^2 + h|\zeta_n^z|^2\Big] &\leqslant C\mathbb{E}\Big[\frac{1}{h}|\mathcal{U}_n(X_n,\theta_n^\star) - \bar{Y}_n|^2 + h|\mathcal{V}_n(X_n,\theta_n^\star) - \bar{Z}_n|^2\Big]\\ &\leqslant CN\mathcal{E}_n(\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}),\end{aligned}$$

where for the last inequality we applied Lemma 3.3.4. Now the proof is concluded using the stability result given in Proposition 3.2.1. □

We conclude this section with a global control on the error between the true solution of the BSDEs and the scheme introduced in Definition 3.3.3.

Theorem 3.3.1 *Let $(\bar{Y}_n, \bar{Z}_n) := (u(t_n, X_n), \sigma^\top \nabla_x u(t_n, X_n))$, for $n \leqslant N$. Then, the following holds*

$$\max_n \mathbb{E}\Big[|\bar{Y}_n - \hat{\mathcal{U}}_n(X_n)|^2\Big] + \sum_{n=0}^{N-1} h\mathbb{E}\Big[|\bar{Z}_n - \hat{\mathcal{V}}_n(X_n)|^2\Big] \leqslant C(h^4 + N\sum_{n=0}^{N-1}\bar{\mathcal{E}}_n)\,.$$

Proof. First, one observes that

$$\begin{aligned}&\max_n \mathbb{E}\Big[|\bar{Y}_n - \hat{\mathcal{U}}_n(X_n)|^2\Big] + \sum_{n=0}^{N-1} h\mathbb{E}\Big[|\bar{Z}_n - \hat{\mathcal{V}}_n(X_n)|^2\Big]\\ &\leqslant 2\left(\max_n \mathbb{E}\big[|\bar{Y}_n - Y_n|^2\big] + \sum_{n=0}^{N-1} h\mathbb{E}\big[|\bar{Z}_n - Z_n|^2\big]\right)\\ &+ 2\left(\max_n \mathbb{E}\Big[|Y_n - \hat{\mathcal{U}}_n(X_n)|^2\Big] + \sum_{n=0}^{N-1} h\mathbb{E}\Big[|Z_n - \hat{\mathcal{V}}_n(X_n)|^2\Big]\right)\end{aligned}$$

The first two terms in the right hand side of the previous inequality is the discrete-time error, whose upper bound follows from Theorem 3.2.1(ii). The second term is upper bounded using Proposition 3.3.1. □

3.3.3 Two stage explicit Runge-Kutta scheme

We now present the numerical procedure to compute two stage Runge-Kutta scheme. It is essentially based on an iteration of what has been done for the Crank-Nicolson scheme. Some simplifications occur for the first stage, as this is an explicit Euler step: indeed, there is no need to introduce the correction term A. Recalling Theorem 3.2.1(iii), one can choose the coefficients such that

$$a_{21} = \alpha_{21} = c_2, \quad a_{31} = \alpha_{31} = 1 - \frac{1}{2c_2}, \quad a_{32} = \alpha_{32} = \frac{1}{2c_2},$$

to obtain the optimal bound on the discrete time error.
The scheme reads thus as follows

$$Y_{n,2} = \mathbb{E}_{t_{n,2}}\big[Y_{n+1} + c_2 h f(X_{n+1}, Y_{n+1}, Z_{n+1})\big], \tag{3.3.48}$$
$$Z_{n,2} = \mathbb{E}_{t_{n,2}}\big[H_2^n Y_{n+1} + H_{2,1}^n c_2 h f(X_{n+1}, Y_{n+1}, Z_{n+1}))\big], \tag{3.3.49}$$

and

$$Y_n = \mathbb{E}_{t_i}\Big[Y_{n+1} + (1 - \frac{1}{2c_2})hf(X_{n+1}, Y_{n+1}, Z_{n+1}) + \frac{1}{2c_2} hf(X_{n,2}, Y_{n,2}, Z_{n,2})\Big], \tag{3.3.50}$$

$$Z_n = \mathbb{E}_{t_i}\Big[H_3^n Y_{n+1} + (1 - \frac{1}{2c_2})hH_3^n f(X_{n+1}, Y_{n+1}, Z_{n+1}) + \frac{1}{2c_2} hH_{3,2}^n f(X_{n,2}, Y_{n,2}, Z_{n,2})\Big]. \tag{3.3.51}$$

Note that we have used $H_{3,1}^n = H_3^n$, which simplifies slightly the term $A_{n,3}$ below.
We must consider loss functions for each stage of computations, namely:
- First stage: For $(\varphi, \psi) \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}) \times \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$,

$$\mathrm{L}_{\mathrm{n},2}^{\mathrm{RK2}}[\varphi, \psi](\theta) := \mathbb{E}\Big[\Big|\varphi(X_{n+1}) + hc_2 f(X_{n+1}, \varphi(X_{n+1}), \psi(X_{n+1})) - \Big\{\mathcal{U}(X_{n,2}; \theta) + \mathcal{V}(X_{n,2}; \theta)\frac{H_2^n}{\upsilon_2^n}\Big\}\Big|^2\Big]$$

with $(\mathcal{U}, \mathcal{V}) \in \mathcal{S}^{\rho}_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12);
- last stage: For $(\Phi, \Psi) \in \mathcal{C}(\mathbb{R}^d, \mathbb{R})^2 \times \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)^2$,

$$\begin{aligned}
&\mathrm{L}_{\mathrm{n},3}^{\mathrm{RK2}}[\Phi, \Psi](\theta) := \\
&\mathbb{E}\Big[\Big|\Phi_1(X_{n+1}) + h(1 - \frac{1}{2c_2})f(X_{n+1}, \Phi_1(X_{n+1}), \Psi_1(X_{n+1})) \\
&\quad + \frac{h}{2c_2} f(X_{n,2}, \Phi_2(X_{n,2}), \Psi_2(X_{n,2})) - \Big\{\mathcal{U}(X_n; \theta) + (\mathcal{V}(X_n; \theta) + \mathcal{A}(X_n; \theta))\frac{H_3^n}{\upsilon_3^n}\Big\}\Big|^2 \\
&\quad + C_0 h\Big|\mathcal{A}(X_n; \theta) - \frac{1}{2c_2}\left(H_3^n - H_{3,2}^n\right) hf(X_{n,2}, \Phi_2(X_{n,2}), \Psi_2(X_{n,2}))\Big|^2\Big]
\end{aligned} \tag{3.3.52}$$

with $(\mathcal{U}, \mathcal{V}, \mathcal{A}) := \mathcal{N}_m \in \mathcal{S}^{\rho}_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12).

The implemented scheme in then given by

Definition 3.3.4 *For a given fixed balance number $C_0 > 0$, the algorithm is designed as follows:*

- *For $i = N$, initialize $\hat{\mathcal{U}}_N = g, \hat{\mathcal{V}}_N = \sigma^\top \nabla_x g$.*
- *For $i = N-1, \cdots, 0$, given $\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}$,*
  - *Compute a minimizer of the loss function at step $(n, 2)$:*

$$\theta_{n,2}^\star \in \mathrm{argmin}_\theta\ \mathrm{L}_{\mathrm{n},2}^{\mathrm{RK2}}[\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}](\theta)\ .$$

  Set $(\hat{\mathcal{U}}_{n,2}, \hat{\mathcal{V}}_{n,2}) = (\mathcal{U}(\cdot, \theta_{n,2}^\star), \mathcal{V}(\cdot, \theta_{n,2}^\star))$.

– *Compute a minimizer of the loss at step* $(n, 3)$*:*

$$\theta^{\star}_{n,3} \in \operatorname{argmin}_{\theta} \mathrm{L}^{\mathrm{RK2}}_{\mathrm{n},3}[(\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{U}}_{n,2}), (\hat{\mathcal{V}}_{n+1}, \hat{\mathcal{V}}_{n,2})](\theta) \, .$$

Set $(\hat{\mathcal{U}}_n, \hat{\mathcal{V}}_n) = (\mathcal{U}(\cdot, \theta^{\star}_{n,3}), \mathcal{V}(\cdot, \theta^{\star}_{n,3}))$.

The convergence result for the above scheme is stated below, see Theorem 3.3.2.

3.3.4 Three stage explicit Runge-Kutta scheme

The numerical procedure of three stage Runge-Kutta scheme consists in one more iteration than the two stage Runge-Kutta scheme. Recalling Theorem 3.2.1(iv), one can choose the coefficients such that

$$a_{21} = \alpha_{21} = c_2, \quad a_{31} = \alpha_{31} = \frac{c_3(3c_2 - 3c_2^2 - c_3)}{c_2(2 - 3c_2)}, \quad a_{32} = \alpha_{32} = \frac{c_3(c_3 - c_2)}{c_2(2 - 3c_2)},$$
$$a_{41} = \alpha_{41} = \frac{-3c_3 + 6c_2c_3 - 3c_2}{6c_2c_3}, \; a_{42} = \alpha_{42} = \frac{3c_3 - 2}{6c_2(c_3 - c_2)}, \; a_{43} = \alpha_{43} = \frac{2 - 3c_2}{6c_3(c_3 - c_2)}.$$

to obtain the optimal bound on the discrete time error. The scheme reads thus as follows

$$Y_{n,2} = \mathbb{E}_{t_{n,2}}[Y_{n+1} + c_2 h f(X_{n+1}, Y_{n+1}, Z_{n+1})] \, , \tag{3.3.53}$$
$$Z_{n,2} = \mathbb{E}_{t_{n,2}}\big[H^n_2 Y_{n+1} + H^n_{2,1} c_2 h f(X_{n+1}, Y_{n+1}, Z_{n+1}))\big] \, , \tag{3.3.54}$$

$$Y_{n,3} = \mathbb{E}_{t_{n,3}}[Y_{n+1} + a_{31} h f(X_{n+1}, Y_{n+1}, Z_{n+1}) + a_{32} h f(X_{n,2}, Y_{n,2}, Z_{n,2})] \, , \tag{3.3.55}$$
$$Z_{n,3} = \mathbb{E}_{t_{n,3}}\big[H^n_3 Y_{n+1} + H^n_{3,1} \alpha_{31} h f(X_{n+1}, Y_{n+1}, Z_{n+1}) + H^n_{3,2} \alpha_{32} h f(X_{n,2}, Y_{n,2}, Z_{n,2}))\big] \, , \tag{3.3.56}$$

and

$$\begin{aligned} Y_n = \mathbb{E}_{t_{n,4}}[&Y_{n+1} + a_{41} h f(X_{n+1}, Y_{n+1}, Z_{n+1}) + a_{42} h f(X_{n,2}, Y_{n,2}, Z_{n,2}) \\ &+ a_{43} h f(X_{n,3}, Y_{n,3}, Z_{n,3})] \, , \end{aligned} \tag{3.3.57}$$
$$\begin{aligned} Z_n = \mathbb{E}_{t_{n,4}}[&H^n_4 Y_{n+1} + H^n_{4,1} \alpha_{41} h f(X_{n+1}, Y_{n+1}, Z_{n+1}) \\ &+ H^n_{4,2} \alpha_{42} h f(X_{n,2}, Y_{n,2}, Z_{n,2})) + H^n_{4,3} \alpha_{43} h f(X_{n,3}, Y_{n,3}, Z_{n,3}))], \end{aligned} \tag{3.3.58}$$

Note that we have used $H^n_{q,1} = H^n_q$ for $q = 3, 4$, which simplifies slightly the term $A_{n,3}, A_{n,4}$ below.
We must consider loss functions for each stage of computations, namely:
- First stage: For $(\Phi, \Psi) \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}) \times \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$,

$$\begin{aligned} \mathrm{L}^{\mathrm{RK3}}_{\mathrm{n},2}[\Phi, \Psi](\theta) := \mathbb{E}\Big[\Big| &\varphi(X_{n+1}) + h c_2 f(X_{n+1}, \Phi(X_{n+1}), \Psi(X_{n+1})) \\ &- \Big\{ \mathcal{U}(X_{n,2}; \theta) + \mathcal{V}(X_{n,2}; \theta) \frac{H^n_2}{\upsilon^n_2} \Big\} \Big|^2 \Big] \end{aligned}$$

with $(\mathcal{U},\mathcal{V}) \in \mathcal{S}^{\rho}_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12);
- Second stage: For $(\Phi,\Psi) \in \mathcal{C}(\mathbb{R}^d,\mathbb{R})^2 \times \mathcal{C}(\mathbb{R}^d,\mathbb{R}^d)^2$,

$$\begin{aligned}\mathrm{L}^{\mathrm{RK3}}_{\mathrm{n},3}[\Phi,\Psi](\theta) := \mathbb{E}\Big[\Big|&\Phi_1(X_{n+1}) + ha_{31}f(X_{n+1},\Phi_1(X_{n+1}),\Psi_1(X_{n+1})) \\ &+a_{32}hf(X_{n,2},\Phi_2(X_{n,2}),\Psi_2(X_{n,2})) - \Big\{\mathcal{U}(X_{n,3};\theta) + (\mathcal{V}(X_{n,3};\theta)+\mathcal{A}(X_{n,3};\theta))\frac{H^n_3}{\upsilon^n_3}\Big\}\Big|^2 \\ &+C_0h\Big|\mathcal{A}(X_{n,3};\theta) - \big(a_{32}H^n_3 - \alpha_{32}H^n_{3,2}\big)\,hf(X_{n,2},\Phi_2(X_{n,2}),\Psi_2(X_{n,2}))\Big|^2\Big]\end{aligned} \tag{3.3.59}$$

with $(\mathcal{U},\mathcal{V},\mathcal{A}) := \mathcal{N}_m \in \mathcal{S}^{\rho}_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12).
- Third stage: For $(\Phi,\Psi) \in \mathcal{C}(\mathbb{R}^d,\mathbb{R})^3 \times \mathcal{C}(\mathbb{R}^d,\mathbb{R}^d)^3$,

$$\begin{aligned}\mathrm{L}^{\mathrm{RK3}}_{\mathrm{n},3}[\Phi,\Psi](\theta) := \mathbb{E}\Big[\Big|&\Phi_1(X_{n+1}) + ha_{41}f(X_{n+1},\Phi_1(X_{n+1}),\Psi_1(X_{n+1})) \\ &+a_{42}hf(X_{n,2},\Phi_2(X_{n,2}),\Psi_2(X_{n,2})) + a_{43}hf(X_{n,3},\Phi_3(X_{n,3}),\Psi_3(X_{n,3})) \\ &- \Big\{\mathcal{U}(X_n;\theta) + (\mathcal{V}(X_n;\theta)+\mathcal{A}(X_n;\theta))\frac{H^n_4}{\upsilon^n_4}\Big\}\Big|^2 \\ &+C_0h\Big|\mathcal{A}(X_n;\theta) - \big(a_{42}H^n_4 - \alpha_{42}H^n_{4,2}\big)\,hf(X_{n,2},\Phi_2(X_{n,2}),\Psi_2(X_{n,2})) \\ &- \big(a_{43}H^n_4 - \alpha_{43}H^n_{4,3}\big)\,hf(X_{n,3},\Phi_3(X_{n,3}),\Psi_3(X_{n,3}))\Big|^2\Big]\end{aligned} \tag{3.3.60}$$

with $(\mathcal{U},\mathcal{V},\mathcal{A}) := \mathcal{N}_m \in \mathcal{S}^{\rho}_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12).

The implemented scheme is then:

Definition 3.3.5 *For a given fixed balance number $C_0 > 0$, the algorithm is designed as follows:*

- *For $i = N$, initialize $\hat{\mathcal{U}}_N = g, \hat{\mathcal{V}}_N = \sigma^\top \nabla_x g$.*
- *For $i = N-1,\cdots,1,0$, given $\hat{\mathcal{U}}_{n+1},\hat{\mathcal{V}}_{n+1}$,*
  - *Compute a minimizer of the loss function at step $(n,2)$:*

$$\theta^\star_{n,2} \in \mathrm{argmin}_\theta\ \mathrm{L}^{\mathrm{RK3}}_{\mathrm{n},2}[\hat{\mathcal{U}}_{n+1},\hat{\mathcal{V}}_{n+1}](\theta)\ .$$

  Set $(\hat{\mathcal{U}}_{n,2},\hat{\mathcal{V}}_{n,2}) = (\mathcal{U}(\cdot,\theta^\star_{n,2}),\mathcal{V}(\cdot,\theta^\star_{n,2}))$.
  - *Compute a minimizer of the loss at step $(n,3)$:*

$$\theta^\star_{n,3} \in \mathrm{argmin}_\theta\ \mathrm{L}^{\mathrm{RK3}}_{\mathrm{n},3}[(\hat{\mathcal{U}}_{n+1},\hat{\mathcal{U}}_{n,2}),(\hat{\mathcal{V}}_{n+1},\hat{\mathcal{V}}_{n,2})](\theta)\ .$$

  Set $(\hat{\mathcal{U}}_n,\hat{\mathcal{V}}_n) = (\mathcal{U}(\cdot,\theta^\star_{n,3}),\mathcal{V}(\cdot,\theta^\star_{n,3}))$.
  - *Compute a minimizer of the loss at step $(n,4)$:*

$$\theta^\star_{n,4} \in \mathrm{argmin}_\theta\ \mathrm{L}^{\mathrm{RK3}}_{\mathrm{n},4}[(\hat{\mathcal{U}}_{n+1},\hat{\mathcal{U}}_{n,2},\hat{\mathcal{U}}_{n,3}),(\hat{\mathcal{V}}_{n+1},\hat{\mathcal{V}}_{n,2},\hat{\mathcal{V}}_{n,3})](\theta)\ .$$

  Set $(\hat{\mathcal{U}}_n,\hat{\mathcal{V}}_n) = (\mathcal{U}(\cdot,\theta^\star_{n,4}),\mathcal{V}(\cdot,\theta^\star_{n,4}))$.

The convergence result for the above scheme is stated below in Theorem 3.3.2.

3.3.5 General case

The general case is built using the approach developed for the Crank-Nicolson scheme with the necessary introduction of the $\mathcal{A}$-terms. Each stage will be computed recursively. Our first observation is the following.

Lemma 3.3.6 *The transition from step n to $n-1$ in the scheme given in Definition 3.2.1 is solution of the following sequence of optimisation problems. For $1 < q \leqslant Q+1$, define*

$$A_{n,q} = \mathbb{E}_{t_{n,q}}\left[\sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right) hf(X_{n,k},Y_{n,k},Z_{n,k})\right], \tag{3.3.61}$$

then, recall (3.2.1)-(3.2.2), we have

$$(Y_{n,q},Z_{n,q},A_{n,q}) = \operatorname{argmin}_{(y,z,a)\in\mathcal{L}^2(\mathcal{F}_{t_{n,q}})} L_{n,q}(y,z,a), \tag{3.3.62}$$

with

$$\begin{aligned} &L_{n,q}(y,z,a) \qquad\qquad (3.3.63)\\ &:= \mathbb{E}\Bigg[\Big|Y_{n+1} + h\sum_{k=1}^{q-1} a_{qk}f(X_{n,k},Y_{n,k},Z_{n,k}) - \{y - ha_{qq}f(X_{n,q},y,z) + (z+a)\frac{H_q^n}{\upsilon_q^n}\}\Big|^2\\ &+C_0h\Big|a - \sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right) hf(X_{n,k},Y_{n,k},Z_{n,k})\Big|^2\Bigg]. \end{aligned}$$

Proof. We first observe that

$$Y_{n+1} = Y_{n,q} - h\sum_{k=1}^{q} a_{qk}f(X_{n,k},Y_{n,k},Z_{n,k}) + (Z_{n,q}+A_{n,q})\frac{H_q^n}{\upsilon_q^n} - \Delta M_{n,q}, \tag{3.3.64}$$

where $\mathbb{E}_{t_n}[\Delta M_{n,q}] = \mathbb{E}_{t_n}\big[\Delta M_{n,q}H_q^n\big] = 0, \mathbb{E}_{t_n}\big[|\Delta M_{n,q}|^2\big] < \infty$. We also have that

$$\begin{aligned} &\mathbb{E}\left[\left|a - \sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right) hf(X_{n,k},Y_{n,k},Z_{n,k})\right|^2\right]\\ &= \mathbb{E}\left[|a-A_{n,q}|^2 + \left|A_{n,q} - \sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right) hf(X_{n,k},Y_{n,k},Z_{n,k})\right|^2\right]. \end{aligned}$$

Inserting (3.3.64) into the definition of $L_{n,q}(y,z,a)$ and using the previous equality, we compute

$$\begin{aligned} &L_{n,q}(y,z,a)\\ &= \mathbb{E}\left[\Big|Y_{n+1} + h\sum_{k=1}^{q-1} a_{qk}f(X_{n,k},Y_{n,k},Z_{n,k}) - \{y - ha_{qq}f(X_{n,q},y,z) + (z+a)\frac{H_q^n}{\upsilon_q^n}\}\Big|^2\right]\\ &+ C_0h\mathbb{E}\left[|a-A_{n,q}|^2 + \left|A_{n,q} - \sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right) hf(X_{n,k},Y_{n,k},Z_{n,k})\right|^2\right]\\ &= \tilde{L}_1^{n,q}(y,z) + \tilde{L}_2^{n,q}(a,z) + \tilde{L}_3^{n,q}(a) + \ell_{n,q}, \end{aligned}$$

where

$$\tilde{L}_1^{n,q}(y,z) = \mathbb{E}\big[|Y_{n,q} - a_{qq}hf(X_{n,q},Y_{n,q},Z_{n,q}) - \{y - a_{qq}hf(X_{n,q},y,z)\}|^2\big], \quad (3.3.65)$$

$$\tilde{L}_2^{n,q}(a,z) = \frac{1}{\upsilon_q^n}\mathbb{E}\big[|Z_{n,q} - z + A_{n,q} - a|^2\big], \quad (3.3.66)$$

$$\tilde{L}_3^{n,q}(a) = C_0 h\mathbb{E}\big[|A_{n,q} - a|^2\big], \quad (3.3.67)$$

$$\ell_{n,q} = \mathbb{E}\left[|\Delta M_{n,q}|^2 + C_0 h\left|A_{n,q} - \sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right)hf(X_{n,k},Y_{n,k},Z_{n,k})\right|^2\right]. \quad (3.3.68)$$

We then observe that $\tilde{L}_3^{n,q}(A_n) = 0$, $\tilde{L}_2^{n,q}(A_n,Z_n) = 0$ and $\tilde{L}_1^{n,q}(Y_n,Z_n) = 0$, so that $(Y_{n,q},Z_{n,q},A_{n,q})$ does achieve the minimum of $L_{n,q}$.
Reciprocally, any optimal solution $(y^\star,z^\star,a^\star)$ must satisfy $\tilde{L}_3^{n,q}(a^\star) = 0, \tilde{L}_2^{n,q}(a^\star,z^\star) = 0$, which implies $a^\star = A_{n,q}, z^\star = Z_{n,q}$. Moreover, necessarily one has $\tilde{L}_1^{n,q}(y^\star,Z_{n,q}) = 0$, then using (3.2.1), we find

$$y^\star = \mathbb{E}_{t_n}\left[Y_{n+1} + h\sum_{k=1}^{q-1}a_{qk}f(X_{n,k},Y_{n,k},Z_{n,k}) + a_{qq}f(X_{n,q},y^\star,Z_{n,q})\right].$$

By uniqueness of the scheme definition, we get $y^\star = Y_{n,q}$, which concludes the proof. □

3.3.5.1 Implementation

Let $\Phi = (\Phi_1,\cdots,\Phi_{Q+1}) \in \mathcal{C}(\mathbb{R}^d,\mathbb{R})^{Q+1}$ and $\Psi = (\Psi_1,\cdots,\Psi_{Q+1}) \in \mathcal{C}(\mathbb{R}^d,\mathbb{R}^d)^{Q+1}$, we introduce a generic loss function at each stage of computation (n,q), $1 \leqslant n < N$, $1 < q \leqslant Q+1$:

$$\begin{aligned}
&\mathrm{L}_{\mathrm{n,q}}^{\mathrm{RK}}[\Phi,\Psi](\theta) := \\
&\mathbb{E}\Bigg[\Big|\Phi_1(X_{n+1}) + h\sum_{k=1}^{q-1}a_{qk}f(X_{n,k},\Phi_k(X_{n,k}),\Psi_k(X_{n,k})) \\
&\quad -\{\mathcal{U}(X_{n,q};\theta) - ha_{qq}f(X_{n,q},\mathcal{U}(X_{n,q};\theta),\mathcal{V}(X_{n,q};\theta)) + (\mathcal{V}(X_{n,q};\theta) + \mathcal{A}(X_{n,q};\theta))\frac{H_q^n}{\upsilon_q^n}\}\Big|^2 \\
&\quad +C_0 h\Big|\mathcal{A}(X_{n,q};\theta) - \sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right)hf(X_{n,k},\Phi_k(X_{n,k}),\Psi_k(X_{n,k}))\Big|^2\Bigg]
\end{aligned} \quad (3.3.69)$$

with $(\mathcal{U},\mathcal{V},\mathcal{A}) := \mathcal{N}_m \in \mathcal{S}^\rho_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, recall (3.1.11)-(3.1.12).

Definition 3.3.6 (Implemented Runge-Kutta scheme) *The numerical solution is computed using the following step:*

- *For $n = N$, initialize $\hat{\mathcal{U}}_N = g, \hat{\mathcal{V}}_N = \sigma^\top\nabla_X g$, $\hat{\mathcal{A}}_N = 0$.*

- *For* $n = N-1, \cdots, 0$, *for* $1 < q \leqslant Q+1$ *given* $(\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}) =: (\hat{\mathcal{U}}_{n,1}, \hat{\mathcal{V}}_{n,1})$ *and* $(\hat{\mathcal{U}}_{n,k}, \hat{\mathcal{V}}_{n,k})$, $1 < k < q$,
  - *set* $(\Phi_k, \Psi_k) := (\hat{\mathcal{U}}_{n,k}, \hat{\mathcal{V}}_{n,k})$, $1 \leqslant k < q$, $(\Phi_k, \Psi_k) := 0$, $k \geqslant q$
  - *Compute a minimizer of the loss function:*

$$\theta^{\star}_{n,q} \in \operatorname{argmin}_{\theta} \mathrm{L}^{\mathrm{RK}}_{\mathrm{n,q}}[\Phi, \Psi](\theta),$$

    where $\mathrm{L}^{\mathrm{RK}}_{\mathrm{n,q}}$ *defined by* (3.3.69).
  - *set* $(\hat{\mathcal{U}}_{n,q}, \hat{\mathcal{V}}_{n,q}, \hat{\mathcal{A}}_{n,q}) := \mathcal{N}_m(\cdot; \theta^{\star}_{n,q}) \in \mathcal{S}^{\rho}_{d_0,d_1,L,m}(\mathbb{R}^{N_m})$, *recall* (3.1.11)-(3.1.12).

  Set $(\hat{\mathcal{U}}_n, \hat{\mathcal{V}}_n) := (\hat{\mathcal{U}}_{n,Q+1}, \hat{\mathcal{V}}_{n,Q+1})$

3.3.5.2 Pseudo-consistency

We study here a kind of minimal consistency of the implemented scheme in terms of approximation error made at each step.

Lemma 3.3.7 *Let* $n < N$ *and* $1 < q \leqslant Q+1$. *Assume that*

$$\theta^{\star} \in \operatorname{argmin}_{\theta} \mathrm{L}^{\mathrm{RK}}_{\mathrm{n,q}}[\Phi, \Psi](\theta),$$

and define

$$Y^{(\Phi,\Psi)}_{n,q} := \mathbb{E}_{t_{n,q}}\left[Y^{(\Phi,\Psi)}_{n,1} + h \sum_{k=1}^{q} a_{qk} f(X_{n,k}, Y^{(\Phi,\Psi)}_{n,k}, Z^{(\Phi,\Psi)}_{n,k}) \right], \tag{3.3.70}$$

$$Z^{(\Phi,\Psi)}_{n,q} := \mathbb{E}_{t_n}\left[H^n_q Y^{(\Phi,\Psi)}_{n,1} + h \sum_{k=1}^{q-1} \alpha_{qk} H^n_{q,k} f(X_{n,k}, Y^{(\Phi,\Psi)}_{n,k}, Z^{(\Phi,\Psi)}_{n,k}) \right], \tag{3.3.71}$$

$$A^{(\Phi,\Psi)}_{n,q} = \mathbb{E}_{t_{n,q}}\left[\sum_{k=1}^{q-1} \left(a_{qk} H^n_q - \alpha_{qk} H^n_{q,k} \right) h f(X_{n,k}, Y^{(\Phi,\Psi)}_{n,k}, Z^{(\Phi,\Psi)}_{n,k}) \right], \tag{3.3.72}$$

with $(Y^{(\Phi,\Psi)}_{n,k}, Z^{(\Phi,\Psi)}_{n,k}) := (\Phi_k(X_{n,k}), \Psi_k(X_{n,k}))$, *for* $1 \leqslant k < q$. *Then, the following holds*

$$\mathbb{E}\Big[|Y^{(\Phi,\Psi)}_{n,q} - \mathcal{U}_{n,q}(X_{n,q}; \theta^{\star})|^2 + h|Z^{(\Phi,\Psi)}_{n,q} - \mathcal{V}_{n,q}(X_{n,q}; \theta^{\star})|^2 + h|A^{(\Phi,\Psi)}_{n,q} - \mathcal{A}_{n,q}(X_{n,q}; \theta^{\star})|^2 \Big]$$
$$\leqslant C\mathcal{E}_{n,q}(\Phi, \Psi), \tag{3.3.73}$$

where

$$\mathcal{E}_{n,q}(\Phi, \Psi) = \epsilon^{\mathcal{N},y}_{n,q}(\Phi, \Psi) + h\epsilon^{\mathcal{N},z}_{n,q}(\Phi, \Psi) + h\epsilon^{\mathcal{N},a}_{n,q}(\Phi, \Psi) \tag{3.3.74}$$

and

$$\epsilon^{\mathcal{N},y}_{n,q}(\Phi, \Psi) := \inf_{\theta^y} \mathbb{E}\Big[|Y^{(\Phi,\Psi)}_{n,q} - \mathcal{U}_{n,q}(X_{n,q}; \theta^y)|^2 \Big], \tag{3.3.75}$$

$$\epsilon^{\mathcal{N},a}_{n,q}(\Phi, \Psi) := \inf_{\theta^a} \mathbb{E}\Big[|A^{(\Phi,\Psi)}_{n,q} - \mathcal{A}_{n,q}(X_{n,q}; \theta^a)|^2 \Big], \tag{3.3.76}$$

$$\epsilon^{\mathcal{N},z}_{n,q}(\Phi, \Psi) := \inf_{\theta^z} \mathbb{E}\Big[|Z^{(\Phi,\Psi)}_{n,q} - \mathcal{V}_{n,q}(X_{n,q}; \theta^z)|^2 \Big]. \tag{3.3.77}$$

Proof. 1. We first observe that (3.3.70) rewrites

$$Y_{n+1}^{(\Phi,\Psi)} = Y_{n,q}^{(\Phi,\Psi)} - h\sum_{k=1}^{q} a_{qk} f(X_{n,k}, Y_{n,k}^{(\Phi,\Psi)}, Z_{n,k}^{(\Phi,\Psi)}) + \left(Z_n^{(\Phi,\Psi)} + A_n^{(\Phi,\Psi)}\right)\frac{H_q^n}{\upsilon_q^n} + \Delta M_{n,q}^{(\Phi,\Psi)}, \tag{3.3.78}$$

where $\mathbb{E}_{t_{n,q}}\left[\Delta M_{n,q}^{(\Phi,\Psi)}\right] = \mathbb{E}_{t_{n,q}}\left[\Delta M_{n,q}^{(\Phi,\Psi)} H_q^n\right] = 0, \mathbb{E}_{t_{n,q}}\left[|\Delta M_{n,q}^{(\Phi,\Psi)}|^2\right] < \infty$. Following the same computations as in the proof of Lemma 3.3.6, we obtain that

$$\mathrm{L}_{\mathrm{n,q}}^{\mathrm{RK}}[\Phi,\Psi](\theta) = \tilde{L}_1^{n,q}[\Phi,\Psi](\theta) + \tilde{L}_2^{n,q}[\Phi,\Psi](\theta) + \tilde{L}_3^{n,q}[\Phi,\Psi](\theta) + \ell_{n,q}[\Phi,\Psi] \tag{3.3.79}$$

with

$$\begin{aligned}
\tilde{L}_1^{n,q}[\Phi,\Psi](\theta) = \mathbb{E}\Big[&|Y_{n,q}^{(\Phi,\Psi)} - a_{qq} h f(X_{n,q}, Y_{n,q}^{(\Phi,\Psi)}, Z_{n,q}^{(\Phi,\Psi)}) \\
&- \{\mathcal{U}_{n,q}(X_{n,q};\theta) - a_{qq} h f(X_{n,q}, \mathcal{U}_{n,q}(X_{n,q};\theta), \mathcal{V}_{n,q}(X_{n,q};\theta))\}|^2\Big],
\end{aligned} \tag{3.3.80}$$

$$\tilde{L}_2^{n,q}[\Phi,\Psi](\theta) = \frac{1}{\upsilon_q^n}\mathbb{E}\left[|Z_{n,q}^{(\Phi,\Psi)} - \mathcal{V}_{n,q}(X_{n,q};\theta) + A_{n,q}^{(\Phi,\Psi)} - \mathcal{A}_{n,q}(X_{n,q};\theta)|^2\right], \tag{3.3.81}$$

$$\tilde{L}_3^{n,q}[\Phi,\Psi](\theta) = C_0 h \mathbb{E}\left[|A_{n,q}^{(\Phi,\Psi)} - \mathcal{A}_{n,q}(X_{n,q};\theta)|^2\right], \tag{3.3.82}$$

$$\begin{aligned}
\ell_{n,q}[\Phi,\Psi] = \mathbb{E}\Big[&|\Delta M_{n,q}^{(\Phi,\Psi)}|^2 + \\
&C_0 h \left|A_{n,q}^{(\Phi,\Psi)} - \sum_{k=1}^{q-1}\left(a_{qk} H_q^n - \alpha_{qk} H_{q,k}^n\right) h f(X_{n,k}, Y_{n,k}^{(\Phi,\Psi)}, Z_{n,k}^{(\Phi,\Psi)})\right|^2\Big].
\end{aligned} \tag{3.3.83}$$

Setting $\tilde{L}_{n,q}[\Phi,\Psi](\theta) := \tilde{L}_1^{n,q}[\Phi,\Psi](\theta) + \tilde{L}_2^{n,q}[\Phi,\Psi](\theta) + \tilde{L}_3^{n,q}[\Phi,\Psi](\theta)$, we then deduce that

$$\mathrm{argmin}_\theta\, \tilde{L}_{n,q}[\Phi,\Psi](\theta) = \mathrm{argmin}_\theta\, \mathrm{L}_{\mathrm{n,q}}^{\mathrm{RK}}[\Phi,\Psi](\theta). \tag{3.3.84}$$

2. Using the Lipschitz continuity of f and recalling (3.2.4)-(3.2.5), we easily compute the following upper bound:

$$\begin{aligned}
\tilde{L}_{n,q}[\Phi,\Psi](\theta) \leqslant C\mathbb{E}\Big[&|Y_{n,q}^{(\Phi,\Psi)} - \mathcal{U}_{n,q}(X_{n,q};\theta)|^2 + h|Z_{n,q}^{(\Phi,\Psi)} - \mathcal{V}_{n,q}(X_{n,q};\theta)|^2 \\
&+ h|A_{n,q}^{(\Phi,\Psi)} - \mathcal{A}_{n,q}(X_{n,q};\theta)|^2\Big].
\end{aligned} \tag{3.3.85}$$

We now prove a lower bound for the previous quantity. First, we observe that, for any $0 < \alpha < 1$,

$$(x+y)^2 \geqslant x^2(1-\alpha) + y^2(1 - \frac{1}{\alpha}). \tag{3.3.86}$$

Thus, for any α such that $1 > \alpha > \frac{2}{2+\Lambda C_0}$, we obtain

$$\begin{aligned}
\tilde{L}_{n,q}^2[\Phi,\Psi](\theta) + \tilde{L}_{n,q}^3[\Phi,\Psi](\theta) \geqslant &h\frac{1-\alpha}{\Lambda}\mathbb{E}\left[|Z_{n,q}^{(\Phi,\Psi)} - \mathcal{V}_{n,q}(X_{n,q};\theta)|^2\right] \\
&+ h\frac{C_0}{2}\mathbb{E}\left[|A_{n,q}^{(\Phi,\Psi)} - \mathcal{A}_{n,q}(X_{n,q};\theta)|^2\right].
\end{aligned} \tag{3.3.87}$$

Using again (3.3.86), we get

$$\tilde{L}^1_{n,q}[\Phi,\Psi](\theta) \geqslant \frac{1}{2}\mathbb{E}\Big[|Y^{(\Phi,\Psi)}_{n,q} - \mathcal{U}_{n,q}(X_{n,q},\theta)|^2\Big] \\ - a^2_{qq}h^2\mathbb{E}\Big[|f(X_{n,q},Y^{(\Phi,\Psi)}_{n,q},Z^{(\Phi,\Psi)}_{n,q}) - f(X_{n,q},\mathcal{U}_{n,q}(X_{n,q};\theta),\mathcal{V}_{n,q}(X_{n,q};\theta))|^2\Big].$$

Since f is Lipschitz continuous, we obtain

$$\tilde{L}^1_{n,q}[\Phi,\Psi](\theta) \geqslant (\frac{1}{2} - 2a^2_{qq}L^2h^2)\mathbb{E}\Big[|Y^{(\Phi,\Psi)}_{n,q} - \mathcal{U}_{n,q}(X_{n,q},\theta)|^2\Big] \\ - 2a^2_{qq}L^2h^2\mathbb{E}\Big[|Z^{(\Phi,\Psi)}_{n,q} - \mathcal{V}_{n,q}(X_{n,q},\theta)|^2\Big].$$

Combining the previous inquality with (3.3.87), we deduce that for h small enough,

$$\mathbb{E}\Big[|Y^{(\Phi,\Psi)}_{n,q} - \mathcal{U}_{n,q}(X_{n,q};\theta)|^2 + h|Z^{(\Phi,\Psi)}_{n,q} - \mathcal{V}_{n,q}(X_{n,q};\theta)|^2 + \\ h|A^{(\Phi,\Psi)}_{n,q} - \mathcal{A}_{n,q}(X_{n,q};\theta)|^2\Big] \leqslant C\tilde{L}_{n,q}[\Phi,\Psi](\theta) \quad (3.3.88)$$

3. The above inequality is *a fortiori* true at the optimum $\theta^\star$. Moreover, optimizing on separated networks is always more costly than optimizing on a fully connected network thus leading to (3.3.73). □

Proposition 3.3.2 *Assume that the scheme given in Definition 3.3.6 is well-posed then*

$$\max_{n\leqslant N}\mathbb{E}\Big[|Y_n - \hat{\mathcal{U}}_n(X_n)|^2\Big] + h\sum_{n=0}^{N}\mathbb{E}\Big[|Z_n - \hat{\mathcal{V}}_n(X_n)|^2\Big] \leqslant CN\sum_{n=0}^{N-1}\bar{\mathcal{E}}_n\,, \quad (3.3.89)$$

where $\bar{\mathcal{E}}_n := \sum_{k=1}^{Q}\mathcal{E}_{n,k}(\hat{\Phi}_n,\hat{\Psi}_n)$ *and with* $\hat{\Phi} = (\hat{\mathcal{U}}_{n,k})_{1\leqslant k\leqslant Q}$ *and* $\hat{\Psi} = (\hat{\mathcal{V}}_{n,k})_{1\leqslant k\leqslant Q}$*, recall* (3.3.74).

Proof. Let us define, for $n < N, 1 < q \leqslant Q+1$, recalling (3.3.70)-(3.3.71)-(3.3.72),

$$\bar{Y}_{n,q} = Y^{(\hat{\Phi},\hat{\Psi})}_{n,q}\,,\ \bar{Z}_{n,q} = Z^{(\hat{\Phi},\hat{\Psi})}_{n,q}\,,\ \bar{A}_{n,q} = A^{(\hat{\Phi},\hat{\Psi})}_{n,q}\,. \quad (3.3.90)$$

We first observe that $(\hat{U}_{n,q},\hat{V}_{n,q}) := (\mathcal{U}_{n,q}(X_{n,q}),\mathcal{V}_{n,q}(X_{n,q}))$ can be rewritten as a perturbed scheme, namely

$$\hat{U}_{n,q} = \mathbb{E}_{t_{n,q}}\left[\hat{U}_{n+1} + h\sum_{k=1}^{q}a_{qk}f(X_{n,k},\hat{U}_{n,k},\hat{V}_{n,k})\right] + \zeta^y_{n,q}, \quad (3.3.91)$$

$$\hat{V}_{n,q} = \mathbb{E}_{t_{n,q}}\left[H^n_q\hat{U}_{n+1} + h\sum_{k=1}^{q-1}\alpha_{qk}H^n_{q,k}f(X_{n,k},\hat{U}_{n,k},\hat{V}_{n,k})\right] + \zeta^z_{n,q}, \quad (3.3.92)$$

with

$$\zeta^y_{n,q} := \hat{U}_{n,q} - \bar{Y}_{n,q} + a_{qq}h\left(f(X_{n,q},\bar{Y}_{n,q},\bar{Z}_{n,q}) - f(X_{n,q},\hat{U}_{n,q},\hat{V}_{n,q})\right), \quad (3.3.93)$$

$$\zeta^z_{n,q} := \hat{V}_{n,q} - \bar{Z}_{n,q}\,. \quad (3.3.94)$$

Indeed, with our notations, we have $(\hat{U}_{n,k}, \hat{V}_{n,k}) = (Y_{n,k}^{(\hat{\Phi},\hat{\Psi})}, Z_{n,k}^{(\hat{\Phi},\hat{\Psi})}), 1 \leqslant k < q$. Moreover, since $\hat{U}_{n,q} = \hat{\mathcal{U}}_{n,q}(X_{n,q}) = \mathcal{U}_{n,q}(X_{n,q}, \theta^{\star}_{n,q})$, recall Definition 3.3.6, it holds

$$\begin{aligned}\mathbb{E}\Big[\frac{1}{h}|\zeta^y_{n,q}|^2 + h|\zeta^z_{n,q}|^2\Big] &\leqslant C\mathbb{E}\Big[\frac{1}{h}|\mathcal{U}_{n,q}(X_{n,q},\theta^{\star}_{n,q}) - \bar{Y}_{n,q}|^2 + h|\mathcal{V}_{n,q}(X_{n,q},\theta^{\star}_{n,q}) - \bar{Z}_{n,q}|^2\Big] \\ &\leqslant CN\mathcal{E}_{n,q}(\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{V}}_{n+1}),\end{aligned}$$

where for the last inequality we applied Lemma 3.3.7. Now the proof is concluded using the stability result given in Proposition 3.2.1. □

We can then state the following convergence result for two stage explicit Runge-Kutta scheme and three stage explicit Runge-Kutta scheme.

Theorem 3.3.2 *Let* $(\bar{Y}_n, \bar{Z}_n) := (u(t_n, X_n), \sigma^\top \nabla_x u(t_n, X_n))$, *for* $n \leqslant N$.

1. *Assume* $(\mathrm{Hr})_2$ *and* $(\mathrm{HX})_2$. *Then, the following holds*

$$\max_n \mathbb{E}\Big[|\bar{Y}_n - \hat{\mathcal{U}}_n(X_n)|^2\Big] + \sum_{n=0}^{N-1} h\mathbb{E}\Big[|\bar{Z}_n - \hat{\mathcal{V}}_n(X_n)|^2\Big] \leqslant C(h^4 + \sum_{n=0}^{N-1} \bar{\mathcal{E}}_n^{\mathrm{RK2}}) .$$

with $\bar{\mathcal{E}}_n^{\mathrm{RK2}} = \mathcal{E}_{n,2}\Big((\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{U}}_{n,2}), (\hat{\mathcal{V}}_{n+1}, \hat{\mathcal{V}}_{n,2})\Big)$, *where* $\mathcal{E}_{n,2}$ *is defined by* (3.3.74).

2. *Assume* $(\mathrm{Hr})_3$ *and* $(\mathrm{HX})_3$. *Then, the following holds*

$$\max_n \mathbb{E}\Big[|\bar{Y}_n - \hat{\mathcal{U}}_n(X_n)|^2\Big] + \sum_{n=0}^{N-1} h\mathbb{E}\Big[|\bar{Z}_n - \hat{\mathcal{V}}_n(X_n)|^2\Big] \leqslant C(h^6 + \sum_{n=0}^{N-1} \bar{\mathcal{E}}_n^{\mathrm{RK3}}) .$$

with $\bar{\mathcal{E}}_n^{\mathrm{RK3}} = \mathcal{E}_{n,3}\Big((\hat{\mathcal{U}}_{n+1}, \hat{\mathcal{U}}_{n,2}, \hat{\mathcal{U}}_{n,3}), (\hat{\mathcal{V}}_{n+1}, \hat{\mathcal{V}}_{n,2}, \hat{\mathcal{V}}_{n,3})\Big)$, *where* $\mathcal{E}_{n,3}$ *is defined by* (3.3.74).

Proof. First, one observes that

$$\begin{aligned}&\max_n \mathbb{E}\Big[|\bar{Y}_n - \hat{\mathcal{U}}_n(X_n)|^2\Big] + \sum_{n=0}^{N-1} h\mathbb{E}\Big[|\bar{Z}_n - \hat{\mathcal{V}}_n(X_n)|^2\Big] \\ &\leqslant 2\left(\max_n \mathbb{E}\big[|\bar{Y}_n - Y_n|^2\big] + \sum_{n=0}^{N-1} h\mathbb{E}\big[|\bar{Z}_n - Z_n|^2\big]\right) \\ &+ 2\left(\max_n \mathbb{E}\Big[|Y_n - \hat{\mathcal{U}}_n(X_n)|^2\Big] + \sum_{n=0}^{N-1} h\mathbb{E}\Big[|Z_n - \hat{\mathcal{V}}_n(X_n)|^2\Big]\right).\end{aligned}$$

The first term in the right hand side of the previous inequality is the discrete-time error, whose upper bound follows from Theorem 3.2.1(iii), (iv), the second term is upper bounded using Proposition 3.3.2. □

3.4 Numerical results

3.4.1 Approximation of the forward process

3.4.1.1 Brownian motion case

We study the special case where $\{\mathcal{X}_t\}_{0\leqslant t\leqslant T}$ is a drifted Brownian motion, that is

$$\mathcal{X}_t = \mathcal{X}_0 + \mu t + \sigma W_t, \quad 0 \leqslant t \leqslant T. \tag{3.4.1}$$

We use directly the forward diffusion process $\mathcal{X}$ on the the grid Π. There is no discretization error for this special case and it corresponds directly to the Euler scheme on Π. In the case, where the underlying is the Brownian motion, note that $L^{(0)} \circ L^{(\ell)} = L^{(\ell)} \circ L^{(0)}$ for $\ell \in \{1, \cdots, d\}$ and see Remark 2.1(ii) of [24], one can choose to compute the Z-part, the H random weight with this simple form: for $n < N$,

$$\begin{aligned} H_q^n &= \frac{W_{t_{n+1}} - W_{t_{n,q}}}{c_q h}, \quad 1 < q \leqslant Q+1, \\ H_{q,k}^n &= \frac{W_{t_{n,k}} - W_{t_{n,q}}}{t_{n,k} - t_{n,q}} = \frac{W_{t_{n,q}} - W_{t_{n,k}}}{(c_k - c_q)h}, \quad 1 < q < k \leqslant Q+1. \end{aligned}$$

It can be verified that $H_q^n, H_{q,k}^n$ satisfy the assumptions $(\mathrm{H}X)_2$, see Proposition 2.3 in [24].

3.4.1.2 General diffusion case of Crank-Nicolson scheme

For the general diffusion case, we also have to discretize the forward diffusion process with some second order weak approximation schemes, , see [1, 92], for all $\varphi \in \mathcal{C}_b^4(\mathbb{R}^d)$,

$$\sup_{t\in[0,T]} |\mathbb{E}[\varphi(X_t)] - \mathbb{E}[\varphi(\mathcal{X}_t)]| \leqslant Ch^2. \tag{3.4.2}$$

We implemented it by Ninomiya-Victoire scheme [76] in practice, the convergence of this algorithm only requires φ to be continuous, under a condition on the vector fields weaker than Hörmander condition. For example, we apply the Ninomiya-Victoire scheme to the d-dimensional independent Cox–Ingersoll–Ross(CIR for short) process:

$$\mathrm{d}\mathcal{X}_t = a(b - \mathcal{X}_t)\,\mathrm{d}t + \sigma\Sigma_d\,\mathrm{d}W_t, \quad 0 \leqslant t \leqslant T.$$

where $\Sigma_d = \mathrm{diag}(\sqrt{\mathcal{X}_t^1}, \cdots, \sqrt{\mathcal{X}_t^d})$. Under the Ninomiya-Victoire scheme, assume Λ_n is a Bernoulli random variable independent of $(W_n)_{n=0,1,\cdots,N-1}$, then

$$\begin{aligned} X_0 &= \mathcal{X}_0 \\ X_{n+1} &= \begin{cases} \exp(\frac{hV_0}{2})\exp(\Delta W_n^d V_d)\ldots\exp(\Delta W_n^1 V_1)\exp(\frac{hV_0}{2})X_n, & \Lambda_n = 1 \\ \exp(\frac{hV_0}{2})\exp(\Delta W_n^1 V_1)\ldots\exp(\Delta W_n^d V_d)\exp(\frac{hV_0}{2})X_n, & \Lambda_n = -1 \end{cases} \end{aligned}$$

where for $x \in \mathbb{R}^d$,

$$\begin{cases} \exp(sV_0)(x) = xe^{-as} + (b - \frac{\sigma^2}{4\mu})(1 - e^{-as})\mathbf{1}_d, \\ \exp(sV_i)(x) = (x_1, \cdots, x_{i-1}, (\frac{\sigma s}{2} + \sqrt{x_i})^2, x_{i+1}, \cdots, x_d), \quad i \neq 0 \end{cases}$$

Hence, if all the components of $(W_n)_{n=0,1,\cdots,N-1}$ are independent,

$$\begin{aligned} \exp(\Delta W_n V) :&= \exp(\Delta W_n^1 V_1) \cdots \exp(\Delta W_n^d V_d)(x) = \exp(\Delta W_n^d V_d) ... \exp(\Delta W_n^1 V_1)(x) \\ &= \left((\frac{\sigma \Delta W_n^1}{2} + \sqrt{x_1})^2, \cdots, (\frac{\sigma \Delta W_n^d}{2} + \sqrt{x_d})^2 \right). \end{aligned}$$

Thus,

$$X_{n+1} = \exp(\frac{hV_0}{2}) \exp(\Delta W_n V) \exp(\frac{hV_0}{2}) X_n.$$

And in this case, for $0 \leqslant n \leqslant N-1$, one can choose

$$H_n = \frac{c-2}{c-1} \frac{W_{t_{n+1}-ch} - W_{t_n}}{h} + \frac{c-1}{c} \frac{W_{t_{n+1}} - W_{t_{n+1}-ch}}{h} \in \mathbb{R}^d, \quad c \in (0,1)$$

which directly comes from Example 2.1(ii.b) of [24]. Then, for each component ${}^l H_i, 1 \leqslant l \leqslant d$ of H_i, we have $\mathbb{E}_{t_i}\left[|{}^l H_i|^2\right] = \left(1 + \frac{1}{c(1-c)}\right) \frac{1}{h}$. We use $c = 0.5$ for the implementation of this Chapter.

3.4.2 Empirical convergence results

For the neural networks in numerical experiments, we used a fully connected feedforward network with 2 hidden layers and the number of neurons of each hidden layer is $d + 10$. And for the networks of all schemes, a tanh activation function is used after each hidden layer. We choose the $batchsize = b$ (set $b = 1000$ for Brownian motion case) to train the network and check the convergence of loss function with a test dataset of $batchsize = 2b$ after every 50 training epochs, we decrease the learning rate with a discount factor $\gamma = 0.5$ if the loss decay is less than a given threshold, and we stop training after the learning rate less than 10^{-9}. In fact, it is good enough to choose 10^{-6} for Euler scheme. We choose a smaller stopping learning rate in order to reduce the impact of the variance of Y_0 for high order schemes. And small batchsize $b_1 = 1000, b_2 = 10000$ is enough for the Brownian motion case. However, for the general diffusion process, small batchsize usually leads to a bias of Y_0 due to the cumulative error from the accuracy of gradient by Monte Carlo simulation.

We take the absolute of the difference of the theoretical solution and average result of $nTest = n$ runs as the error:

$$\epsilon := \left| \frac{1}{n} \sum_{i=1}^{n} \hat{Y}_0^i - Y_0 \right|,$$

we set $n = 10$ for the numerical results below.

We implemented the code in Python3 with the multi-process technique to run n tests at the same time. Due to the long running time, especially when the discretization time steps N is large, we run the code on a server [1].

3.4.2.1 Brownian motion case

We consider the following example borrowed from [65]. For $d = 10, T = 1, t \in [0, T]$, let

$$\begin{aligned} \mathrm{d}\mathcal{X}_t &= \frac{0.2}{d}\mathbf{1}_d\,\mathrm{d}t + \frac{1}{\sqrt{d}}\mathbf{I}_d\,\mathrm{d}W_t, \quad \mathcal{X}_0 = \mathbf{1}_d, \\ f(t,x,y,z) &= (\cos(\bar{x}) + 0.2\sin(\bar{x}))\, e^{\frac{T-t}{2}} - \frac{1}{2}(\sin(\bar{x})\cos(\bar{x})e^{T-t})^2 + \frac{1}{2d}\left(y\,(z\cdot\mathbf{1}_d)\right)^2, \\ g(x) &= \cos(\bar{x}), \end{aligned}$$

where $\bar{x} = \sum_{i=1}^{d} x_i$. The theoretical solution of this BSDE is $Y_t = \cos(\bar{X}_t)e^{\frac{T-t}{2}}$ and $Z_t^i = -\frac{1}{\sqrt{d}}\sin(\bar{X}_t)e^{\frac{T-t}{2}}, i = 1, \cdots, d.$.

We will compare the numerical results for the 5 cases stated in Theorem 3.2.1 and called them explicit Euler scheme, implicit Euler scheme, CN scheme, RK-2 scheme, RK-3 scheme, respectively. We set $c_2 = 0.5$ and we recall Section 3.3.3

$$a_{21} = \alpha_{21} = c_2, \quad a_{31} = \alpha_{31} = 1 - \frac{1}{2c_2}, \quad a_{32} = \alpha_{32} = \frac{1}{2c_2},$$

for RK-2 scheme. For RK-3 scheme, we set $c_2 = 0.3, c_3 = 0.7$ and recall Section 3.3.4

$$\begin{aligned} &a_{21} = \alpha_{21} = c_2, \quad a_{31} = \alpha_{31} = \frac{c_3(3c_2 - 3c_2^2 - c_3)}{c_2(2 - 3c_2)}, \quad a_{32} = \alpha_{32} = \frac{c_3(c_3 - c_2)}{c_2(2 - 3c_2)}, \\ &a_{41} = \alpha_{41} = \frac{-3c_3 + 6c_2c_3 - 3c_2}{6c_2c_3}, \; a_{42} = \alpha_{42} = \frac{3c_3 - 2}{6c_2(c_3 - c_2)}, \; a_{43} = \alpha_{43} = \frac{2 - 3c_2}{6c_3(c_3 - c_2)}. \end{aligned}$$

We know discuss the balance number C_0 as it has some influence on the numerical results. On Figure 3.1, we can see that the error of CN scheme is stable when $\log_2(C_0) > 3$. However, we can get smaller error when $-1 < \log_2(C_0) < 3$. In the implementation, we choose $C_0 = \frac{4}{3}$ for CN scheme, the convergence rate is almost order 2 when the time steps $N \leqslant 12$, and then if $N > 12$, the error is almost a constant, this might be caused by the variance of $\hat{Y}_0$. Similarly, we choose $C_0 = 25c_q$ for Runge-Kutta scheme at stage $1 < q < Q + 1$.

Figure 3.2 shows that the computational time is almost linear to the time steps for each scheme which is reasonable. The computational time of implicit Euler scheme and explicit scheme are almost the same. The computational time of CN scheme is only slightly larger than Euler scheme since this is still a one-stage scheme though the driver f is computed on both t_n and t_{n+1} at each step $0 \leqslant n \leqslant N - 1$.

[1] The server of the Laboratory LPSM, which has 32 cores CPU and 2 GPUs

For the Runge-Kutta scheme in the general case: the computational time is more than Q times the computational time of CN scheme, as expected.

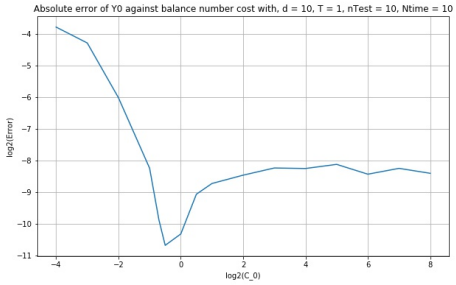

Figure 3.1 – The error of Y_0 for CN scheme based on the balance number

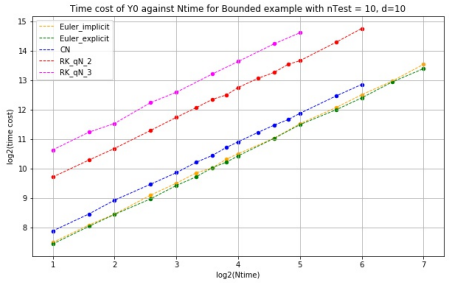

Figure 3.2 – Time cost against time steps for different schemes

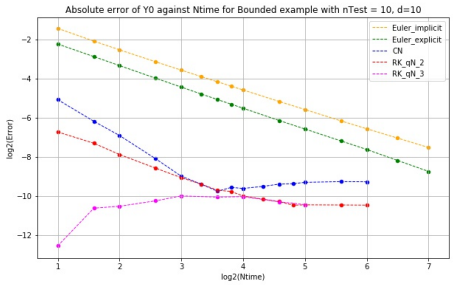

Figure 3.3 – Error against time steps for different schemes

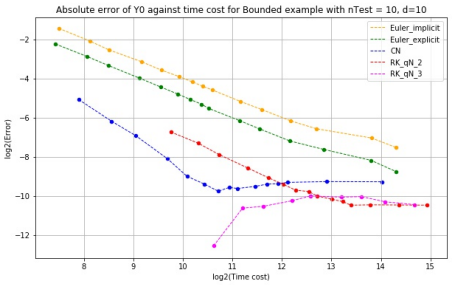

Figure 3.4 – Error against time cost for different schemes

In Figure 3.3, we compare the convergence rate of the 5 schemes mentioned above. We verify that the implicit Euler scheme and explicit Euler scheme are almost order 1. The CN scheme is almost order 2. The convergence rate of RK-2 scheme is slightly less than CN scheme, but the error is smaller. The RK-3 scheme converges so fast that we are not able to observe any convergence order.

In Figure 3.4, we plotted the error w.r.t. the time cost for the 5 schemes mentioned above. We see that the Euler schemes are too slow reach a small error. The RK-3 scheme is very fast but it spend too much time even the number of time steps N is small. As we expected, CN scheme is faster than RK-2 scheme. In conclusion, if we want an error smaller than $0.01 \approx 2^{-6.64}$, CN scheme seems to be the best scheme to use.

3.4.2.2 Cox–Ingersoll–Ross process

In this section, we test the CIR process as we mentioned before,

$$d\mathcal{X}_t = a(b - \mathcal{X}_t)dt + \sigma\sqrt{\mathcal{X}_t}dW_t,$$

which is a mean reversion process, the following conditions ensure $(\mathcal{X}_t)_{t>0}$ is always positive:

$$a > 0,\ b > 0,\ 2ab \geqslant \sigma^2,\ \mathcal{X}_0 \geqslant 0.$$

And the distribution of future values of a CIR process can be computed in closed form: for $t > 0$,

$$\mathcal{X}_t \sim \frac{\chi_k'^2(p)}{2c} \quad \text{with} \quad c = \frac{2a}{(1-e^{-at})\sigma^2},\ k = \frac{4ab}{\sigma^2},\ p = 2c\mathcal{X}_0 e^{-at},$$

where $\chi_k'^2(p)$ is a non-central chi-squared distribution with k degrees of freedom and non-centrality parameter p. One can also compute the expectation and Variance of $\mathcal{X}_t$:

$$\mathbb{E}[\mathcal{X}_t] = \mathcal{X}_0 e^{-at} + b(1-e^{-at}), \quad Var(\mathcal{X}_t) = \mathcal{X}_0 \frac{\sigma^2}{a}(e^{-at} - e^{-2at}) + \frac{b\sigma^2}{2a}(1-e^{-at})^2.$$

We tested the BSDE with the same solution of the previous subsection $Y_t = u(t, X_t) = \cos(\bar{X}_t)e^{\frac{T-t}{2}}$, $Z_t^i = -\sqrt{\frac{X_t^i}{d}}\sin(\bar{X}_t)e^{\frac{T-t}{2}}, i = 1, \cdots, d$, recalling $\bar{x} := \sum_{i=1}^{d} x_i$, and keep the terminal function g of Brownian Motion case, but with the forward diffusion process is CIR process with $a = \frac{1}{5d}, b = 3, \sigma = \frac{1}{\sqrt{d}}$ and recalling $\Sigma_d = \text{diag}(\sqrt{\mathcal{X}_t^1}, \cdots, \sqrt{\mathcal{X}_t^d})$:

$$\mathrm{d}\mathcal{X}_t = \frac{1}{5d}(3 - \mathcal{X}_t)\,\mathrm{d}t + \frac{1}{\sqrt{d}}\Sigma_d\,\mathrm{d}W_t, \quad \mathcal{X}_0 = 10\mathbf{1}_d.$$

Then, setting the driver f:

$$\begin{aligned}
\tilde{f}(x) &= \left(\frac{1}{2}\cos(\bar{x})(1+\frac{\bar{x}}{d}) + \sin(\bar{x})(\frac{3}{5} - \frac{\bar{x}}{5d})\right)e^{\frac{T-t}{2}}, \\
f(t,x,y,z) &= \tilde{f}(x) - \frac{1}{5}(\sin(\bar{x})\cos(\bar{x})e^{T-t})^2 + \frac{1}{5d}\left(y\sum_{i=1}^{d}\frac{z_i}{\sqrt{x_i}}\right)^2, \\
&= \tilde{f}(x) - \frac{1}{5}(\sin(\bar{x})\cos(\bar{x})e^{T-t})^2 + \frac{1}{5d^2}\left(u(t,x)\sum_{i=1}^{d}\partial_{x_i}u(t,x)\right)^2,
\end{aligned}$$

where $\bar{x} = \sum_{i=1}^{d} x_i$ and recalling Section 3.4.1.2, for $x \in \mathbb{R}^d$,

$$\begin{cases}
\exp(sV_0)(x) = xe^{-\frac{s}{5d}} + \frac{7}{4}(1-e^{-\frac{s}{5d}})\mathbf{1}_d, \\
\exp(sV_i)(x) = (x_1, \cdots, x_{i-1}, (\frac{s}{2\sqrt{d}} + \sqrt{x_i})^2, x_{i+1}, \cdots, x_d), \quad i \neq 0
\end{cases}$$

We only compare Crank-Nicolson scheme with implicit Euler scheme in this subsection since Crank-Nicolson scheme is the best one as we discussed above. Setting $d = 10, T = 1, C_0 = 1$. We only test the implicit Euler scheme with $b = 5000$, and we test the Crank-Nicolson scheme for both $b = 5000$ and $b = 50000$.

In Figure 3.5, the implicit Euler scheme is almost order 1, however it can only achieve an error around 2^{-5} even $N = 128$. And then see Figure 3.6, the approximation solution $\hat{Y}_0$ cross the real solution Y_0 around $N = 256$ and a bias appeared when $N = 512$. For the Crank-Nicolson scheme with batchsize $b_1 = 5000$ (red line), it converger faster than order 2 and achieve the minimal error at $N = 12$, then the error almost converges to 2^{-5} (the same with Euler scheme with $N = 512$). When we increase the batchsize 10 times, the Crank-Nicolson scheme (blue line) is almost order 2 when $N \leqslant 16$, and then achieve an error smaller than 2^{-6} at last, it seems there still exist a bias, but it smaller than the red line.

Figure 3.7 shows the relation between discreate time steps and time cost, except the impact of the machine, the lines are almost order 1. In Figure 3.8, see the orange line and the red line, CN scheme can achieve the same error with Euler scheme but less time cost when the batchsize for both scheme are identical. However, we have to increase the value of batchsize if we want to improve the accuracy which will greatly increase the computational time cost.

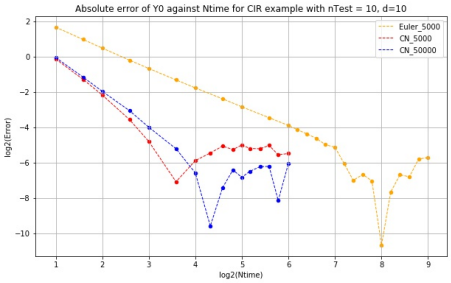

Figure 3.5 – Error against time steps for different schemes

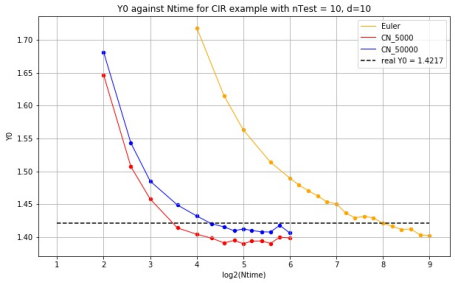

Figure 3.6 – Y_0 against time steps for different schemes

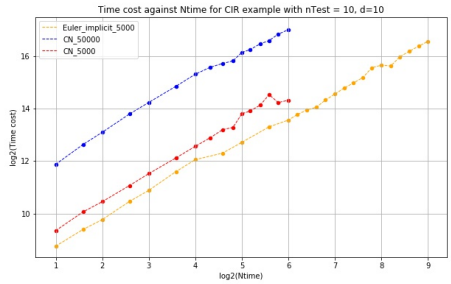

Figure 3.7 – Time cost against time steps for different schemes

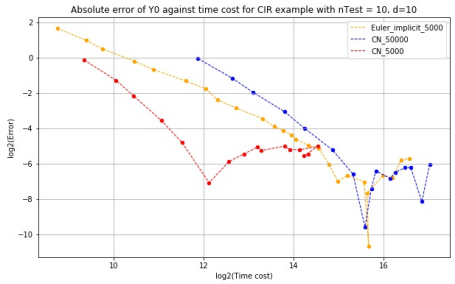

Figure 3.8 – Error against time cost for different schemes

3.5 Appendix

3.5.1 Proof of Proposition 3.2.1

Recalling (3.2.1)-(3.2.2) and $\upsilon_q^n = \mathbb{E}_{t_{n,q}}\big[|H_q^n|^2\big]$, observe that for $1 < q \leqslant Q+1$, it holds

$$Y_{n,q} = Y_{n+1} + h\sum_{k=1}^{q} a_{qk} f(X_{n,k}, Y_{n,k}, Z_{n,k}) - (Z_{n,q} + A_{n,q})\frac{H_q^n}{\upsilon_q^n} + \Delta M_{n,q}, \tag{3.5.1}$$

where $\mathbb{E}_{t_{n,q}}[\Delta M_{n,q}] = \mathbb{E}_{t_{n,q}}\big[H_q^n \Delta M_{n,q}\big] = 0, \mathbb{E}_{t_{n,q}}\big[|\Delta M_{n,q}|^2\big] < \infty$, and

$$\begin{aligned} A_{n,q} &= \mathbb{E}_{t_{n,q}}\left[H_q^n\left(Y_{n+1} + h\sum_{k=1}^{q-1} a_{qk} f(X_{n,k}, Y_{n,k}, Z_{n,k})\right)\right] - Z_{n,q} \\ &= \mathbb{E}_{t_{n,q}}\left[\sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right) h f(X_{n,k}, Y_{n,k}, Z_{n,k})\right]. \end{aligned} \tag{3.5.2}$$

And for the perturbed scheme defined in (3.2.12)-(3.2.13), observe that

$$\tilde{Y}_{n,q} = \tilde{Y}_{n+1} + h\sum_{k=1}^{q} a_{qk} f(X_{n,k}, \tilde{Y}_{n,k}, \tilde{Z}_{n,k}) + \zeta_{n,q}^y - (\tilde{Z}_{n,q} + \tilde{A}_{n,q} - \zeta_{n,q}^z)\frac{H_q^n}{\upsilon_q^n} + \Delta\tilde{M}_{n,q}, \tag{3.5.3}$$

where $\mathbb{E}_{t_{n,q}}\big[\Delta\tilde{M}_{n,q}\big] = \mathbb{E}_{t_{n,q}}\big[H_q^n \Delta\tilde{M}_{n,q}\big] = 0, \mathbb{E}_{t_{n,q}}\big[|\Delta\tilde{M}_{n,q}|^2\big] < \infty$, and

$$\tilde{A}_{n,q} = \mathbb{E}_{t_{n,q}}\left[\sum_{k=1}^{q-1}\left(a_{qk}H_q^n - \alpha_{qk}H_{q,k}^n\right) h f(X_{n,k}, \tilde{Y}_{n,k}, \tilde{Z}_{n,k})\right]. \tag{3.5.4}$$

Set $\delta Y_{n,q} = Y_{n,q} - \tilde{Y}_{n,q}, \delta Z_{n,q} = Z_{n,q} - \tilde{Z}_{n,q}, \delta A_{n,q} = A_{n,q} - \tilde{A}_{n,q}, \delta f_{n,q} = f(X_{n,q}, Y_{n,q}, Z_{n,q}) - f(X_{n,q}, \tilde{Y}_{n,q}, \tilde{Z}_{n,q})$, $\delta\Delta M_{n,q} = \Delta M_{n,q} - \Delta\tilde{M}_{n,q}$ for all $0 \leqslant n \leqslant N-1, 1 < q \leqslant Q+1$. From eq. (3.5.1) and eq. (3.5.3), we get

$$\delta Y_{n,q} + (\delta Z_{n,q} + \zeta_{n,q}^z)\frac{H_q^n}{\upsilon_q^n} + \delta\Delta M_{n,q} = \delta Y_{n+1} + h\sum_{k=1}^{q} a_{qk}\delta f_{n,k} - \zeta_{n,q}^y - \delta A_{n,q}\frac{H_q^n}{\upsilon_q^n}. \tag{3.5.5}$$

Step 1: For $0 \leqslant n \leqslant N-1$, control on $\mathbb{E}\big[|\delta Y_n|^2\big]$ by the term $h\sum_{k=n}^{N-1}\sum_{q=2}^{Q+1}\mathbb{E}\big[|\delta Z_{k,q}|^2\big]$.
Squaring both sides of (3.5.5), taking conditional expectation and using Young's inequality, we obtain

$$\begin{aligned} |\delta Y_{n,q}|^2 + \frac{1}{\upsilon_q^n}|\delta Z_{n,q} + \zeta_{n,q}^z|^2 \leqslant \left(1 + \frac{h}{C}\right)\mathbb{E}_{t_{n,q}}\left[\left|\delta Y_{n+1} + h\sum_{k=1}^{j} a_{qk}\delta f_{n,k}\right|^2\right] \\ + 2\left(1 + \frac{C}{h}\right)\left(|\zeta_{n,q}^y|^2 + \frac{1}{\upsilon_q^n}|\delta A_{n,q}|^2\right). \end{aligned} \tag{3.5.6}$$

Recalling (3.2.4)-(3.2.5), and denote $\bar{a} = \max_{q,k}\{|a_{qk}|, |\alpha_{qk}|\}$, thus have

$$\begin{aligned}
|\delta A_{n,q}|^2 &= h^2 \left| \mathbb{E}_{t_{n,q}} \left[\sum_{k=1}^{q-1} (a_{qk} H_q^n - \alpha_{qk} H_{q,k}^n) \delta f_{n,k} \right] \right|^2 \\
&\leqslant (q-1) h^2 \sum_{k=1}^{q-1} \left| \mathbb{E}_{t_{n,q}} \big[(a_{qk} H_q^n - \alpha_{qk} H_{q,k}^n) \delta f_{n,k} \big] \right|^2 \\
&\leqslant q h^2 \sum_{k=1}^{q-1} \mathbb{E}_{t_{n,q}} \big[|a_{qk} H_q^n - \alpha_{qk} H_{q,k}^n|^2 \big] \mathbb{E}_{t_{n,q}} \big[|\delta f_{n,k}|^2 \big] \\
&\leqslant q h^2 \sum_{k=1}^{q-1} \mathbb{E}_{t_{n,q}} \big[2a_{qk}^2 |H_q^n|^2 + 2\alpha_{qk}^2 |H_{q,k}^n|^2 \big] \mathbb{E}_{t_{n,q}} \big[2[f]_L^2 (|\delta Y_{n,k}|^2 + |\delta Z_{n,k}|^2) \big] \\
&\leqslant 8 q^2 \bar{a}^2 [f]_L^2 \Lambda h \sum_{k=1}^{q-1} \mathbb{E}_{t_{n,q}} \big[|\delta Y_{n,k}|^2 + |\delta Z_{n,k}|^2 \big] \\
&\leqslant C h \sum_{k=1}^{q-1} \mathbb{E}_{t_{n,q}} \big[|\delta Y_{n,k}|^2 + |\delta Z_{n,k}|^2 \big], \qquad (3.5.7)
\end{aligned}$$

and

$$|\delta Z_{n,q} + \zeta_{n,q}^z|^2 \geqslant (1-\eta) |\delta Z_{n,q}|^2 + (1 - \frac{1}{\eta}) |\zeta_{n,q}^z|^2 \geqslant (1-\eta) |\delta Z_{n,q}|^2 - \frac{1}{\eta} |\zeta_{n,q}^z|^2, \qquad (3.5.8)$$

for $\eta > 0$. Using the Lipschitz regularity of f, Young's inequality and Jensen's inequality, for any $\epsilon > 0$, we obtain

$$\begin{aligned}
&\left| \delta Y_{n+1} + h \sum_{k=1}^{q} a_{qk} \delta f_{n,k} \right|^2 \\
\leqslant &\left(|\delta Y_{n+1}| + h [f]_L \sum_{k=1}^{q} |a_{qk}| (|\delta Y_{n,k}| + |\delta Z_{n,k}|) \right)^2 \\
\leqslant &\left((1 + Ch) |\delta Y_{n+1}| + Ch |\delta Z_{n+1}| + Ch \sum_{k=2}^{q} (|\delta Y_{n,k}| + |\delta Z_{n,k}|) \right)^2 \\
\leqslant &\left(1 + \frac{h}{\epsilon} \right) \left((1 + Ch) |\delta Y_{n+1}| + Ch \sum_{k=2}^{q} |\delta Y_{n,k}| \right)^2 + C (h + \epsilon) h \left(\sum_{k=1}^{q} |\delta Z_{n,k}| \right)^2 \\
\leqslant &\left(1 + \frac{h}{\epsilon} \right) \left((1 + Ch) |\delta Y_{n+1}|^2 + Ch \sum_{k=2}^{q} |\delta Y_{n+1}| |\delta Y_{n,k}| + Ch^2 \left(\sum_{k=2}^{q} |\delta Y_{n,k}| \right)^2 \right) \\
&+ C (h + \epsilon) h \left(\sum_{k=1}^{q} |\delta Z_{n,k}|^2 \right) \\
\leqslant &\left(1 + \frac{h}{\epsilon} \right) \left((1 + Ch) |\delta Y_{n+1}|^2 + Ch \sum_{k=2}^{q} |\delta Y_{n,k}|^2 \right) + C (h + \epsilon) h \sum_{k=1}^{q} |\delta Z_{n,k}|^2.
\end{aligned}$$

Choosing h small enough and ϵ such that $C(h+\epsilon) \leqslant \frac{1}{2\Lambda}$, we obtain

$$\left|\delta Y_{n+1} + h\sum_{k=1}^{q} a_{qk}\delta f_{n,k}\right|^2 \leqslant (1+Ch)\,|\delta Y_{n+1}|^2 + Ch\sum_{k=2}^{q}|\delta Y_{n,k}|^2 + \frac{h}{2\Lambda}\sum_{k=1}^{q}|\delta Z_{n,k}|^2. \tag{3.5.9}$$

Thus choosing $\eta = \frac{1}{4}$ in (3.5.8), and using (3.5.9), (3.5.7) into (3.5.6), observing that $\frac{h}{\Lambda} \leqslant \frac{1}{v_q^n} \leqslant \frac{h}{\lambda}$, then for h small enough, we get

$$\begin{aligned}
&|\delta Y_{n,q}|^2 \leqslant |\delta Y_{n,q}|^2 + \frac{h}{4\Lambda}|\delta Z_{n,q}|^2 \\
&\leqslant (1+Ch)\,\mathbb{E}_{t_{n,q}}\big[|\delta Y_{n+1}|^2\big] + Ch\sum_{k=1}^{q-1}\mathbb{E}_{t_{n,q}}\big[|\delta Y_{n,k}|^2 + |\delta Z_{n,k}|^2\big] \\
&\quad + \mathbb{E}_{t_{n,q}}\left[\frac{C}{h}|\zeta^y_{n,q}|^2 + Ch\,|\zeta^z_{n,q}|^2\right] \\
&\leqslant (1+Ch)\,\mathbb{E}_{t_{n,q}}\big[|\delta Y_{n+1}|^2\big] + Ch\sum_{k=1}^{q-1}\mathbb{E}_{t_{n,q}}\big[|\delta Z_{n,k}|^2\big] + C\sum_{k=2}^{q}\mathbb{E}_{t_{n,q}}\left[\frac{1}{h}|\zeta^y_{n,k}|^2 + h\,|\zeta^z_{n,k}|^2\right].
\end{aligned} \tag{3.5.10}$$

Using the discrete version of Grönwall's lemma, we even eventually conclude:

$$\mathbb{E}\big[|\delta Y_n|^2\big] \leqslant C\left(\mathbb{E}\big[|\delta Y_N|^2 + |\delta Z_N|^2\big] + h\sum_{k=n}^{N-1}\sum_{q=2}^{Q+1}\mathbb{E}\left[|\delta Z_{k,q}|^2 + \frac{1}{h^2}|\zeta^y_{k,q}|^2 + |\zeta^z_{k,q}|^2\right]\right). \tag{3.5.11}$$

Step 2: Control of $h\sum_{k=n}^{N-1}\sum_{q=2}^{Q+1}\mathbb{E}\big[|\delta Z_{k,q}|^2\big]$.
Using the Cauchy-Schwarz inequality and the Lipschitz regularity of f, we get

$$\begin{aligned}
h\left|\mathbb{E}_{t_{n,q}}\big[hH^n_{q,k}\delta f_{n,k}\big]\right|^2 &\leqslant h^3\mathbb{E}_{t_{n,q}}\big[|H^n_{q,k}|^2\big]\,\mathbb{E}_{t_{n,q}}\big[|\delta f_{n,k}|^2\big] \\
&\leqslant 2d[f]_L^2\Lambda h^2\mathbb{E}_{t_{n,q}}\big[|\delta Y_{n,k}|^2 + |\delta Z_{n,k}|^2\big]
\end{aligned} \tag{3.5.12}$$

and

$$\begin{aligned}
&\mathbb{E}_{t_n}\Big[h\left|\mathbb{E}_{t_{n,q}}\big[H^n_q(\delta Y_{n+1} - \mathbb{E}_{t_n}[\delta Y_{n+1}])\big]\right|^2\Big] \\
&\leqslant \mathbb{E}_{t_n}\big[h\mathbb{E}_{t_{n,q}}\big[|H^n_q|^2\big]\,\mathbb{E}_{t_{n,q}}\big[|\delta Y_{n+1} - \mathbb{E}_{t_n}[\delta Y_{n+1}]\,|^2\big]\big] \\
&\leqslant \Lambda\mathbb{E}_{t_n}\big[|\delta Y_{n+1} - \mathbb{E}_{t_n}[\delta Y_{n+1}]\,|^2\big] \\
&\leqslant \Lambda\left(\mathbb{E}_{t_n}\big[|\delta Y_{n+1}|^2\big] - \mathbb{E}_{t_n}[\delta Y_{n+1}]^2\right).
\end{aligned} \tag{3.5.13}$$

Taking $q = Q + 1$ in (3.5.5), using Young's inequality, and note the subscript $\{n\} \equiv \{n, Q+1\}$, we get

$$
\begin{aligned}
&\mathbb{E}_{t_n}\big[|\delta Y_{n+1}|^2\big] - \mathbb{E}_{t_n}[\delta Y_{n+1}]^2 \\
&= \mathbb{E}_{t_n}\big[|\delta Y_{n+1}|^2\big] - \mathbb{E}_{t_n}\left[\delta Y_n + \zeta_n^y - h\sum_{k=1}^{Q+1} a_{qk}\mathbb{E}_{t_{n,q}}[\delta f_{n,k}]\right]^2 \\
&\leqslant \mathbb{E}_{t_n}\big[|\delta Y_{n+1}|^2\big] - |\delta Y_n|^2 + C\left(|\delta Y_n||\zeta_n^y| + h(|\delta Y_n| + |\zeta_n^y|)\sum_{k=1}^{Q+1}\mathbb{E}_{t_i}[|\delta f_{n,k}|]\right) \\
&\leqslant \mathbb{E}_{t_n}\big[|\delta Y_{n+1}|^2\big] - |\delta Y_n|^2 + C\left(|\delta Y_n||\zeta_n^y| + h(|\delta Y_n| + |\zeta_n^y|)\sum_{k=1}^{Q+1}\mathbb{E}_{t_{n,q}}[|\delta Y_{n,k}| + |\delta Z_{n,k}|]\right) \\
&\leqslant \mathbb{E}_{t_n}\big[|\delta Y_{n+1}|^2\big] - |\delta Y_n|^2 + Ch\sum_{k=1}^{Q+1}\mathbb{E}_{t_n}\left[\frac{1}{\epsilon}|\delta Y_{n,k}|^2 + \epsilon|\delta Z_{n,k}|^2\right] + \frac{C}{h}\mathbb{E}_{t_n}\big[|\zeta_n^y|^2\big],
\end{aligned}
\tag{3.5.14}
$$

for any $\epsilon > 0$. Using Jensen's inequality, and the inequalities (3.5.12), (3.5.13), (3.5.14), we obtain

$$
\begin{aligned}
&h\mathbb{E}_{t_n}\big[|\delta Z_{n,q}|^2\big] = h\mathbb{E}_{t_n}\left[\left|\mathbb{E}_{t_{n,q}}\left[H_q^n\delta Y_{n+1} + h\sum_{k=1}^{q-1}\alpha_{qk}H_{q,k}^n\delta f_{n,k} - \zeta_{n,q}^z\right]\right|^2\right] \\
&\leqslant C\mathbb{E}_{t_n}\left[h\left|\mathbb{E}_{t_{n,q}}\big[H_q^n(\delta Y_{n+1} - \mathbb{E}_{t_n}[\delta Y_{n+1}])\big]\right|^2 + \sum_{k=1}^{q-1}\alpha_{qk}^2 h\left|\mathbb{E}_{t_{n,q}}\big[hH_{q,k}^n\delta f_{n,k}\big]\right|^2 + h|\zeta_{n,q}^z|^2\right] \\
&\leqslant C\left(\mathbb{E}_{t_n}\big[|\delta Y_{n+1}|^2\big] - |\mathbb{E}_{t_n}[\delta Y_{n+1}]\,|^2 + h^2\sum_{k=1}^{q-1}\mathbb{E}_{t_n}\big[|\delta Y_{n,k}|^2 + |\delta Z_{n,k}|^2\big] + h\mathbb{E}_{t_n}\big[|\zeta_{n,q}^z|^2\big]\right) \\
&\leqslant C\Bigg(\mathbb{E}_{t_n}\big[|\delta Y_{n+1}|^2\big] - |\delta Y_n|^2 + Ch\sum_{k=1}^{Q+1}\mathbb{E}_{t_n}\left[\frac{1}{\epsilon}|\delta Y_{n,k}|^2 + \epsilon|\delta Z_{n,k}|^2\right] \\
&\qquad + \mathbb{E}_{t_n}\left[\frac{C}{h}|\zeta_n^y|^2 + h|\zeta_{n,q}^z|^2\right]\Bigg)
\end{aligned}
\tag{3.5.15}
$$

Summing over q and n, setting $\epsilon = \frac{1}{2(Q+2)C}$, and note that the subscript $\{n,1\} \equiv \{n+1\} \equiv \{n+1, Q+1\}$, then for h small enough,

$$\begin{aligned}
&h\sum_{k=n}^{N-1}\sum_{q=2}^{Q+1}\mathbb{E}\big[|\delta Z_{k,q}|^2\big] \\
&\leqslant C\sum_{k=n}^{N-1}\sum_{q=2}^{Q+1}\mathbb{E}\bigg[|\delta Y_{k+1}|^2 - |\delta Y_k|^2 + h\sum_{l=1}^{Q+1}(\frac{1}{\epsilon}|\delta Y_{k,l}|^2 + \epsilon|\delta Z_{k,l}|^2) + \frac{1}{h}|\zeta_k^y|^2 + h|\zeta_{k,q}^z|^2\bigg] \\
&= C\sum_{k=n}^{N-1}\mathbb{E}\bigg[Q|\delta Y_{k+1}|^2 - Q|\delta Y_k|^2 + \frac{Qh}{\epsilon}\sum_{l=1}^{Q+1}|\delta Y_{k,l}|^2 + \frac{Q}{h}|\zeta_k^y|^2 + h\sum_{q=2}^{Q+1}|\zeta_{k,q}^z|^2\bigg] \\
&\quad + \frac{Qh}{Q+2}\sum_{k=n}^{N-1}\sum_{l=2}^{Q+1}\mathbb{E}\big[|\delta Z_{k,l}|^2\big]
\end{aligned}$$

so that

$$\begin{aligned}
&h\sum_{k=n}^{N-1}\sum_{q=2}^{Q+1}\mathbb{E}\big[|\delta Z_{k,q}|^2\big] \\
&\leqslant C\sum_{k=n}^{N-1}\mathbb{E}\bigg[|\delta Y_{k+1}|^2 - |\delta Y_k|^2 + h\sum_{q=1}^{Q+1}|\delta Y_{k,q}|^2 + \frac{1}{h}|\zeta_k^y|^2 + h\sum_{q=2}^{Q+1}|\zeta_{k,q}^z|^2\bigg] \\
&\leqslant C\mathbb{E}\bigg[|\delta Y_N|^2 + h|\delta Z_N|^2 + h\sum_{k=n}^{N-1}|\delta Y_k|^2 + \sum_{k=n}^{N-1}\sum_{q=2}^{Q+1}\bigg(\frac{1}{h}|\zeta_{k,q}^y|^2 + h|\zeta_{k,q}^z|^2\bigg)\bigg] \qquad (3.5.16)
\end{aligned}$$

where we used (3.5.10) and the subscript $\{n,1\} \equiv \{n+1\} \equiv \{n+1, Q+1\}, 0 \leqslant n \leqslant N-1$ again in the last line.

Step 3: Control of the term $\mathbb{E}\big[|\delta Y_n|^2\big]$, $0 \leqslant n \leqslant N-1$.

Combining the inequality (3.5.16) with (3.5.11), we get

$$\mathbb{E}\big[|\delta Y_n|^2\big] \leqslant C\mathbb{E}\bigg[|\delta Y_N|^2 + h|\delta Z_N|^2 + h\sum_{k=n}^{N-1}|\delta Y_k|^2 + h\sum_{k=n}^{N-1}\sum_{q=2}^{Q+1}\bigg(\frac{1}{h^2}|\zeta_{k,q}^y|^2 + |\zeta_{k,q}^z|^2\bigg)\bigg]. \qquad (3.5.17)$$

Step 4: Control on $\max_{n\leqslant k\leqslant N-1}\mathbb{E}\big[|\delta Y_k|^2\big] + h\sum_{k=n}^{N-1}\mathbb{E}\big[|\delta Z_k|^2\big]$.

Set $\delta_n := \sum_{k=n}^{N-1}\mathbb{E}\big[|\delta Y_k|^2\big]$, and

$$\theta_n := \mathbb{E}\big[|\delta Y_N|^2 + h|\delta Z_N|^2\big] + h\sum_{k=n}^{N-1}\sum_{q=2}^{Q+1}\mathbb{E}\bigg[\frac{1}{h^2}|\zeta_{k,q}^y|^2 + |\zeta_{k,q}^z|^2\bigg]. \qquad (3.5.18)$$

It holds

$$\delta_n - \delta_{n+1} \leqslant Ch\delta_n + C\theta_n. \qquad (3.5.19)$$

Using the discrete version of Grönwall's lemma, and noting that $\delta_{N-1} \leqslant \theta_n$, $\theta_k \leqslant \theta_n$ for $k \geqslant n$, we obtain

$$\delta_n \leqslant C\left(\delta_{N-1} + \sum_{k=n}^{N-1} \theta_k e^{C(N-k-1)h}\right) \leqslant C\theta_n \frac{1}{e^{Ch}-1}. \tag{3.5.20}$$

This last inequality combined with (3.5.19) leads to

$$\mathbb{E}\big[|\delta Y_n|^2\big] = \delta_n - \delta_{n+1} \leqslant C\theta_n.$$

And for $Z-$part, the proof is concluded plugging (3.5.20) into (3.5.16) with $n = 0$ in this equation. Then, we conclude the proof

$$\max_{n\leqslant k\leqslant N-1} \mathbb{E}\big[|\delta Y_k|^2\big] + h\sum_{k=n}^{N-1} \mathbb{E}\big[|\delta Z_k|^2\big] \leqslant C\theta_n. \tag{3.5.21}$$

3.5.2 Proof of step 2 of Theorem 3.2.1

Denote $H_n = H_1^n = H_{2,1}^n$, that is

$$\begin{aligned} Y_n &= \mathbb{E}_{t_n}\Big[Y_{n+1} + \frac{1}{2}(f(Y_n, Z_n) + f(Y_{n+1}, Z_{n+1}))\Big], \\ Z_n &= \mathbb{E}_{t_n}\Big[H_n\left(Y_{n+1} + hf(Y_{n+1}, Z_{n+1})\right)\Big]. \end{aligned}$$

Then recalling (3.1.4) and (3.2.18), set $Y_{n+1} = u(t_{n+1}, \mathcal{X}_{t_{n+1}}) =: u_{t_{n+1}}$ we have for $1 \leqslant \ell \leqslant d$,

$$\begin{aligned} \bar{Z}_n^\ell &= \mathbb{E}_{t_n}\Big[(H_n)^\ell\left(Y_{n+1} + hf(Y_{n+1}, Z_{n+1})\right)\Big] \\ &= \mathbb{E}_{t_n}\Big[(H_n)^\ell\left(u_{t_{n+1}} - hu_{t_{n+1}}^{(0)}\right)\Big] \\ &= (u_{t_n}^{(\ell)} + hu_{t_n}^{(\ell,0)} + O(h^2)) - h(u_{t_n}^{(\ell,0)} + O(h)) \\ &= u_{t_n}^{(\ell)} + O(h^2) = Z_{t_n}^\ell + O(h^2), \end{aligned} \tag{3.5.22}$$

which yields,

$$h\sum_{n=0}^{N-1} \mathbb{E}\big[|\bar{Z}_n - Z_n|^2\big] = O(h^4). \tag{3.5.23}$$

Using a first-order Taylor expansion, this leads to

$$\begin{aligned} f(Y_n, \bar{Z}_n) &= f(Y_n, Z_n) + \sum_{\ell=1}^{d} (\bar{Z}_n - Z_n)\partial_{z^\ell} f(Y_n, Z_n) + O(h^2) \\ &= -u_{t_n}^{(0)} - O(h^2)\sum_{\ell=1}^{d} u_{t_n}^{(\ell,0)} + O(h^2) \\ &= -u_{t_n}^{(0)} + O(h^2). \end{aligned} \tag{3.5.24}$$

For the Y-part, we have

$$\begin{aligned}\bar{Y}_n &= \mathbb{E}_{t_n}\Big[Y_{n+1} + \frac{h}{2}f(Y_{n+1}, Z_{n+1})\Big] + \frac{h}{2}f(\bar{Y}_n, \bar{Z}_n) \\ &= \mathbb{E}_{t_n}\Big[Y_{n+1} + \frac{h}{2}f(Y_{n+1}, Z_{n+1}) + \frac{h}{2}f(Y_n, \bar{Z}_n)\Big] + \frac{h}{2}(f(\bar{Y}_n, \bar{Z}_n) - f(Y_n, \bar{Z}_n)) \\ &= \mathbb{E}_{t_n}\Big[u_{t_{n+1}} - \frac{h}{2}u^{(0)}_{t_{n+1}} - \frac{h}{2}u^{(0)}_{t_n}\Big] + \frac{h}{2}(f(\bar{Y}_n, \bar{Z}_n) - f(Y_n, \bar{Z}_n)) + O(h^3). \quad (3.5.25)\end{aligned}$$

On the other hand,

$$u_{t_{n+1}} = u_{t_n} + hu^{(0)}_{t_n} + \frac{h^2}{2}u^{(0,0)}_{t_n} + O(h^3), \quad (3.5.26)$$

$$u^{(0)}_{t_{n+1}} = u^{(0)}_{t_n} + hu^{(0,0)}_{t_n} + O(h^2). \quad (3.5.27)$$

Hence,

$$\bar{Y}_n = Y_n + \frac{h}{2}(f(\bar{Y}_n, \bar{Z}_n) - f(Y_n, \bar{Z}_n)) + O(h^3). \quad (3.5.28)$$

Observe that $\bar{Y}_{t_n} = Y_{t_n} + \mathcal{O}(h)$ which leads to

$$f(\bar{Y}_n, \bar{Z}_n) - f(Y_n, \bar{Z}_n) = O(h). \quad (3.5.29)$$

Therefore,

$$\bar{Y}_n = Y_n + O(h^2), \quad (3.5.30)$$

$$f(\bar{Y}_n, \bar{Z}_n) - f(Y_n, \bar{Z}_n) = O(h^2). \quad (3.5.31)$$

Combine with (3.5.28) again, we have

$$\bar{Y}_n = Y_n + O(h^3), \quad (3.5.32)$$

$$\frac{1}{h}\sum_{i=0}^{N-1}\mathbb{E}\big[|\bar{Y}_n - Y_n|^2\big] = O(h^4), \quad (3.5.33)$$

which combine with (3.5.23) yields

$$\mathcal{T}_N = O(h^4). \quad (3.5.34)$$

Part II

Probabilistic representation of integration by parts formulae for stochastic volatility models with unbounded drift

Chapter 4

Probabilistic representation of IBP formulae for stochastic volatility models with unbounded drift

The content of this chapter is from an article in collaboration with Noufel Frikha, Houzhi Li [29]. Submitted to ESAIM: Probability and Statistics.

Contents

4.1 Introduction

In this work, we consider a two dimensional stochastic volatility model given by the solution of the following stochastic differential equation (SDE for short) with dynamics

$$\begin{cases} S_t & = s_0 + \int_0^t rS_s\,ds + \int_0^t \sigma_S(Y_s)S_s\,dW_s, \\ Y_t & = y_0 + \int_0^t b_Y(Y_s)\,ds + \int_0^t \sigma_Y(Y_s)\,dB_s, \\ d\langle B, W\rangle_s & = \rho\,ds \end{cases} \tag{4.1.1}$$

where the coefficients b_Y, σ_S, $\sigma_Y : \mathbb{R} \to \mathbb{R}$ are smooth functions, $r \in \mathbb{R}$, W and B are one-dimensional standard Brownian motions with correlation factor $\rho \in (-1, 1)$ both being defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

The aim of this part is to prove a probabilistic representation formula for two integration by parts (IBP) formulae for the marginal law of the process (S, Y) at a given time maturity T. To be more specific, for a given starting point $(s_0, y_0) \in (0, \infty) \times \mathbb{R}$ and a given finite time horizon $T > 0$, we establish two Bismut-Elworthy-Li (BEL) type formulae for the two following quantities

$$\partial_{s_0}\mathbb{E}\left[h(S_T, Y_T)\right] \quad \text{and} \quad \partial_{y_0}\mathbb{E}\left[h(S_T, Y_T)\right] \tag{4.1.2}$$

where h is a real-valued possibly non-smooth payoff function defined on $[0, \infty) \times \mathbb{R}$.

Such IBP formulae have attracted a lot of interest during the last decades both from a theoretical and a practical point of views as they can be further analyzed to derive properties related to the transition density of the underlying process or to develop Monte Carlo simulation algorithm among other practical applications, see e.g. Nualart [78], Malliavin and Thalmaier [74] and the references therein. They are also of major interest for computing sensitivities, also referred as to *Greeks* in finance, of arbitrage price of financial derivatives which is the keystone for hedging purpose, i.e. for protecting the value of a portfolio against some possible changes in sources of risk. The two quantities appearing in (4.1.2) corresponds respectively to the Delta and Vega of the European option with payoff $h(S_T, Y_T)$. For a more detailed discussion on this topic, we refer the interested reader to Fournié and al. [40],[39] for IBP formulae related to European, Asian options and conditional expectations, Gobet and al.[48], [12] for IBP formulae related to some barrier or lookback options. Let us importantly point out that, from a numerical point of view, the aforementioned IBP formulae will inevitably involve a time discretization procedure of the underlying process and Malliavin weights, thus introducing two sources of error given by a bias and a statistical error, as it is already the case for the computation of the price $\mathbb{E}[h(S_T, Y_T)]$.

Relying on a perturbation argument for the Markov semigroup generated by the couple (X, Y), we first establish a probabilistic representation formula for the marginal law (S_T, Y_T) for a fixed prescribed maturity $T > 0$ based on a simple Markov chain evolving along a random time grid given by the jump times of an independent renewal process. Such type of probabilistic representation formula was first derived in Bally and Kohatsu-Higa [4] for the marginal law of a multi-dimensional diffusion process and of some Lévy driven SDEs with bounded drift, diffusion and

jump coefficients. Still in the case of bounded coefficients, it was then further investigated in Labordère and al. [60], Agarwal and Gobet [2] for multi-dimensional diffusion processes and in Frikha and al. [42] for one-dimensional killed processes. The major advantage of the aforementioned probabilistic formulae lies in the fact that an unbiased Monte Carlo simulation method directly stems from it. Thus, it may be used to numerically compute an option price with optimal complexity since its computation will be only affected by the statistical error. However, let us emphasize that in general the variance of the Monte Carlo estimator tends to be large or even infinite. In order to circumvent this issue, an importance sampling scheme based on the law of the jump times of the underlying renewal process has been proposed in Anderson and Kohatsu-Higa [3] in the multi-dimensional diffusion framework and in [42] for one-dimensional killed processes.

The main novelty of our approach in comparison to the aforementioned works is that we allow the drift coefficient b_Y to be possibly unbounded as it is the case in most stochastic volatility models (Stein-Stein, Heston, ...). Such boundedness condition on the drift coefficient has appeared persistently in the previous contributions and is actually essential since basically it allows to remove the drift in the choice of the approximation process in order to derive the probabilistic representation formula. Importantly, a direct application of the methodology developed in [4, 60, 42] does not work when the drift is unbounded. The key ingredient that we here develop in order to remove this restriction consists in choosing adequatly the approximation process around which the original perturbation argument of the Markov semigroup (X, Y) is done by taking into account the transport of the initial condition by the deterministic ordinary differential equation (ODE) having unbounded coefficient[1]. The approximation process, or equivalently the underlying Markov chain on which the probabilistic representation is based, is then obtained from the original dynamics (4.1.1) by *freezing the coefficients* b_Y, σ_S *and* σ_Y *along the flow of this ODE*. We stress that the previous choice is here crucial since it provides the adequate approximation process on which some good controls on the weights involved in the probabilistic representation formulae can be established. Roughly speaking, it allows to cancel the time singularity generated by the Malliavin IBP operators appearing in the weights. To the best of our knowledge, this feature appears to be new in this context.

Having this probabilistic representation formula at hand together with the tailor-made Malliavin calculus machinery for this well-chosen underlying Markov chain, in the spirit of the BEL formula established in [42] for killed diffusion processes with bounded drift coefficient, we rely on a propagation of the spatial derivatives forward in time then perform local IBP formulas on each time interval of the random time grid and eventually merge them in a suitable manner in order to establish the two BEL formulae for the two quantities (4.1.2). Following the ideas developed in [3], we achieve finite variance for the Monte Carlo estimators obtained from the probabilistic representation formulas of the couple (S_T, Y_T) and of both IBP formulae by selecting adequatly the law of the jump times of the renewal process. We finally provide some numerical tests illustrating our previous analysis. Let us eventually mention that

[1] This dynamical system is obtained by removing the noise, that is, by setting $\sigma_Y \equiv 0$, from the dynamics of Y in (4.1.1).

for sake of simplicity in the present chapter we have decided to consider only one-dimensional processes S and Y but that some multi-dimensional generalizations of the above formulae could be achieved at the price of additional technicalities which we believe would be prejudicial to the understanding of the main idea.

This chapter is organized as follows. In Section 4.2, we introduce our assumptions on the coefficients, present the approximation process that will be the main building block for our perturbation argument as well as the Markov chain that will play a central role in our probabilistic representation for the marginal law of the process (X, Y) and for our IBP formulae. In addition, we construct the taillor-made Malliavin calculus machinery related to the underlying Markov chain upon which both IBP formulae are made. In Section 4.3, relying on the Markov chain introduced in Section 4.2, we establish in Theorem 4.3.1 the probabilistic representation formula for the coupled (S_T, Y_T). In Section 4.4, we establish the BEL formulae for the two quantities appearing in (4.1.2). The main result of this section is Theorem 4.4.1. As a proof of concept, some numerical results are presented in Section 4.5. Clearly, we believe that one needs to study numerical issues in more details and these are left for later studies. The proofs of Theorem 4.3.1 and of some other technical but important results are postponed to the appendix of Section 4.6.

Notations:

For a fixed time T and positive integer n, we will use the following notation for time and space variables $\mathbf{s}_n = (s_1, \cdots, s_n)$, $\mathbf{x}_n = (x_1, \cdots, x_n)$, the differentials $d\mathbf{s}_n = ds_1 \cdots ds_n$, $d\mathbf{x}_n = dx_1 \cdots dx_n$ and also introduce the simplex $\Delta_n(T) := \{\mathbf{s}_n \in [0,T]^n : 0 \leqslant s_1 < \cdots s_n \leqslant T\}$.

In order to deal with time-degeneracy estimates, we will often use the following space-time inequality:

$$\forall p \geqslant 0, q > 0, \forall x \in \mathbb{R}, \quad |x|^p e^{-q|x|^2} \leqslant (p/(2qe))^{p/2}. \tag{4.1.3}$$

For two positive real numbers α and β, we define the Mittag-Leffler function $z \mapsto E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} z^k/\Gamma(\alpha k + \beta)$. For a positive integer d, we denote by $\mathcal{C}_p^\infty(\mathbb{R}^d)$ the space of real-valued functions which are infinitely differentiable on $\mathbb{R}^d$ with derivatives of any order having polynomial growth.

4.2 Preliminaries: assumptions, definition of the underlying Markov chain and related Malliavin calculus

4.2.1 Assumptions

Throughout the chapter, we work on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ which is assumed to be rich enough to support all random variables that we will consider in what follows. We will work under the following assumptions on the coefficients:

(AR) The coefficients σ_S and σ_Y are bounded and smooth, in particular σ_S and σ_Y belong to $\mathcal{C}_b^\infty(\mathbb{R})$. The drift coefficient b_Y belongs to $\mathcal{C}^\infty(\mathbb{R})$ and admits bounded

derivatives of any order greater than or equal to one. In particular, the drift coefficient b_Y may be unbounded.

(ND) There exists $\kappa \geqslant 1$ such that for all $x \in \mathbb{R}$,

$$\kappa^{-1} \leqslant a_S(x) \leqslant \kappa, \quad \kappa^{-1} \leqslant a_Y(x) \leqslant \kappa$$

where $a_S = \sigma_S^2$ and $a_Y = \sigma_Y^2$. Therefore, without loss of generality, we will assume that both σ_S and σ_Y are positive function.

Apply Itô's lemma to $X_t = \ln(S_t)$. We get

$$\begin{cases} X_t & = x_0 + \displaystyle\int_0^t \Big(r - \frac{1}{2} a_S(Y_s)\Big)\, ds + \int_0^t \sigma_S(Y_s)\, dW_s, \\ Y_t & = y_0 + \displaystyle\int_0^t b_Y(Y_s)\, ds + \int_0^t \sigma_Y(Y_s)\, dB_s, \\ d\langle B, W\rangle_s & = \rho\, ds, \end{cases} \tag{4.2.1}$$

with $x_0 = \ln(s_0)$. Without loss of generality, we will thus work with the Markov semigroup associated to the process (X, Y), namely $P_t h(x_0, y_0) = \mathbb{E}[h(X_t, Y_t)]$.

4.2.2 Choice of the approximation process

As already mentioned in the introduction, our strategy here is based on a probabilistic representation of the marginal law, in the spirit of the unbiased simulation method introduced for multi-dimensional diffusion processes by Bally and Kohatsu-Higa [4], see also Labordère and al. [60], and investigated from a numerical perspective by Andersson and Kohatsu-Higa [3]. We also mention the recent contribution of one of the author with Kohatsu-Higa and Li [42] for IBP formulae for the marginal law of one-dimensional killed diffusion processes.

However, at this stage, it is important to point out that our choice of approximation process significantly differs from the four aforementioned references. Indeed, in the previous contributions, the drift is assumed to be bounded and basically plays no role so that one usually removes it in the dynamics of the approximation process. In order to handle the unbounded drift term b_Y appearing in the dynamics of the volatility process, one has to take into account the transport of the initial condition by the ODE obtained by removing the noise in the dynamics of Y. To be more specific, we denote by $(m_t(s, y))_{t\in[s,T]}$, $0 \leqslant s \leqslant T$, the unique solution to the ODE $\dot{m}_t = b_Y(m_t)$ with initial condition $m_s = y$. Observe that by time-homogeneity of the coefficient b_Y, one has $m_t(s, y) = m_{t-s}(0, y)$. We will simplify the notation when $s = 0$ and write $m_t(y_0)$ for $m_t(0, y_0)$. When there is no ambiguity, we will often omit the dependence with respect to the initial point y_0 and we only write m_t for $m_t(y_0)$. We now introduce the approximation process $(\bar{X}, \bar{Y})$ defined by

$$\begin{cases} \bar{X}_t^{x_0} & = x_0 + \displaystyle\int_0^t (r - \frac{1}{2} a_S(m_s))\, ds + \int_0^t \sigma_S(m_s)\, dW_s, \\ \bar{Y}_t^{y_0} & = y_0 + \displaystyle\int_0^t b_Y(m_s)\, ds + \int_0^t \sigma_Y(m_s)\, dB_s, \\ d\langle B, W\rangle_s & = \rho\, ds. \end{cases} \tag{4.2.2}$$

Observe that the couple $(\bar{X}_t^{x_0}, \bar{Y}_t^{y_0})_{t\geqslant 0}$ is a Gaussian process. We will make intensive use of the explicit form of the Markov semigroup $(\bar{P}_t)_{t\in[0,T]}$ defined for any bounded measurable map $h : \mathbb{R}^2 \to \mathbb{R}$ by $\bar{P}_t h(x_0, y_0) = \mathbb{E}[h(\bar{X}_t^{x_0}, \bar{Y}_t^{y_0})]$.

Lemma 4.2.1 *Let $(x_0, y_0) \in \mathbb{R}^2$, $\rho \in (-1, 1)$ and $t \in (0, \infty)$. Then, for any bounded and measurable map $h : \mathbb{R}^2 \to \mathbb{R}$, it holds*

$$\bar{P}_t h(x_0, y_0) = \int_{\mathbb{R}^2} h(x, y)\, \bar{p}(t, x_0, y_0, x, y)\, dxdy \tag{4.2.3}$$

with

$$\begin{aligned}\bar{p}(t, x_0, y_0, x, y) = {} & \frac{1}{2\pi\sigma_{S,t}\sigma_{Y,t}\sqrt{1-\rho_t^2}} \exp\Big(-\frac{1}{2}\frac{(x - x_0 - (rt - \frac{1}{2}a_{S,t}))^2}{a_{S,t}(1-\rho_t^2)} - \frac{1}{2}\frac{(y-m_t)^2}{a_{Y,t}(1-\rho_t^2)}\Big) \\ & \times \exp\Big(\frac{\rho_t}{(1-\rho_t^2)}\frac{(x - x_0 - (rt - \frac{1}{2}a_{S,t}))(y-m_t)}{\sigma_{S,t}\sigma_{Y,t}}\Big)\end{aligned}$$

where we introduced the notations

$$\begin{aligned} a_{S,t} = a_{S,t}(y_0) := \sigma_{S,t}^2 &:= \int_0^t a_S(m_s(y_0))\, ds, \\ a_{Y,t} = a_{Y,t}(y_0) := \sigma_{Y,t}^2 &:= \int_0^t a_Y(m_s(y_0))\, ds, \\ \sigma_{S,Y,t} = \sigma_{S,Y,t}(y_0) &:= \int_0^t (\sigma_S\sigma_Y)(m_s(y_0))\, ds, \\ \rho_t &:= \rho\sigma_{S,Y,t}/(\sigma_{S,t}\sigma_{Y,t}). \end{aligned}$$

Moreover, there exists some positive constant $C := C(T, \rho, a, r, \kappa)$ such that for any $t \in (0, T]$

$$\bar{p}(t, x_0, y_0, x, y) \leqslant C\bar{q}_{4\kappa}(t, x_0, y_0, x, y) \tag{4.2.4}$$

where, for a positive parameter c, we introduced the density function

$$(x, y) \mapsto \bar{q}_c(t, x_0, y_0, x, y) := \frac{1}{2\pi ct}\exp\Big(-\frac{(x-x_0)^2}{2ct} - \frac{(y-m_t)^2}{2ct}\Big). \tag{4.2.5}$$

Proof. We write

$$(\bar{X}_t^{x_0}, \bar{Y}_t^{y_0}) = \Big(x_0 + rt - \frac{1}{2}a_{S,t} + \int_0^t \sigma_S(m_s)\, dW_s, m_t + \int_0^t \sigma_Y(m_s)\, \Big(\rho dW_s + \sqrt{1-\rho^2}d\widetilde{W}_s\Big)\Big)$$

where $\widetilde{W}$ is a one-dimensional standard Brownian motion independent of W. We thus deduce that $(\bar{X}_t^{x_0}, \bar{Y}_t^{y_0}) \sim \mathcal{N}(\mu(t, x_0, y_0), \Sigma_t)$ with

$$\mu(t, x_0, y_0) = \Big(x_0 + rt - \frac{1}{2}a_{S,t}, m_t\Big) \quad \text{and} \quad \Sigma_t = \begin{pmatrix} a_{S,t} & \rho\sigma_{S,Y,t} \\ \rho\sigma_{S,Y,t} & a_{Y,t}\end{pmatrix}.$$

The expression of the transition density then readily follows. Now, from **(ND)**, it is readily seen that $a_{S,t}, a_{Y,t} \leqslant \kappa t$ so that using the inequalities $ab \leqslant \frac{1}{2}a^2 + \frac{1}{2}b^2$,

$(a-b)^2 \geqslant \frac{1}{2}a^2 - b^2$ and $\rho_t^2 \leqslant \rho^2 \leqslant 1$, it holds

$$\begin{aligned}
&\bar{p}(t, x_0, y_0, x, y) \\
&= \frac{1}{2\pi\sigma_{S,t}\sigma_{Y,t}\sqrt{1-\rho_t^2}} \exp\Big(-\frac{1}{2}\frac{(x - x_0 - (rt - \frac{1}{2}a_{S,t}))^2}{a_{S,t}(1-\rho_t^2)} - \frac{1}{2}\frac{(y-m_t)^2}{a_{Y,t}(1-\rho_t^2)}\Big) \\
&\quad \times \exp\Big(\frac{\rho_t}{1-\rho_t^2}\frac{(x - x_0 - (rt - \frac{1}{2}a_{S,t}))(y-m_t)}{\sigma_{S,t}\sigma_{Y,t}}\Big) \\
&\leqslant C\frac{1}{2\pi\kappa t} \exp\Big(-\frac{1}{2}\frac{(x - x_0 - (rt - \frac{1}{2}a_{S,t}))^2}{a_{S,t}(1-\rho_t^2)}(1-|\rho_t|) - \frac{1}{2}\frac{(y-m_t)^2}{a_{Y,t}(1-\rho_t^2)}(1-|\rho_t|)\Big) \\
&\leqslant C\frac{1}{2\pi(4\kappa)t} \exp\Big(-(4\kappa)^{-1}\frac{(x-x_0)^2}{2t} - (4\kappa)^{-1}\frac{(y-m_t)^2}{2t}\Big) \\
&=: C\bar{q}_{4\kappa}(t, x_0, y_0, x, y)
\end{aligned}$$

for some positive constants $C := C(T, \lambda, \rho, a, r, \kappa)$.

We will also use the notation $(\bar{X}_t^{s,x}, \bar{Y}_t^{s,y})_{t\geqslant s}$ for the approximation process starting from (x, y) at time s and with coefficients frozen along the deterministic flow $\{m_t(s, y) = m_{t-s}(y), t \geqslant s\}$. Note that the corresponding Markov semigroup satisfies $\bar{P}_{s,t}h(x, y) := \mathbb{E}\left[h(\bar{X}_t^{s,x}, \bar{Y}_t^{s,y})\right] = \mathbb{E}\left[h(\bar{X}_{t-s}^{x}, \bar{Y}_{t-s}^{y})\right] = \bar{P}_{t-s}h(x, y)$.

4.2.3 Markov chain on random time grid

The first tool that we will employ is a renewal process N that we now introduce.

Definition 4.2.1 *Let $\tau := (\tau_n)_{n\geqslant 0}$ be a sequence of random variables such that $(\tau_n - \tau_{n-1})_{n\geqslant 1}$, with the convention $\tau_0 = 0$, are i.i.d. with positive density function f and cumulant distribution function $t \mapsto F(t) = \int_{-\infty}^{t} f(s)\,ds$ and τ is independent of $(W_s, B_s)_{0\leqslant s\leqslant T}$. Then, the renewal process $N := (N_t)_{t\geqslant 0}$ with jump times τ is defined by $N_t := \sum_{n\geqslant 1} \mathbf{1}_{\{\tau_n \leqslant t\}}$.*

It is readily seen that, for any $t > 0$, $\{N_t = n\} = \{\tau_n \leqslant t < \tau_{n+1}\}$ and by an induction argument that we omit, one may prove that the joint distribution of $(\tau_1, \cdots, \tau_n)$ is given by

$$\mathbb{P}(\tau_1 \in ds_1, \cdots, \tau_n \in ds_n) = \prod_{j=0}^{n-1} f(s_{j+1} - s_j)\mathbf{1}_{\{0<s_1<\cdots<s_n\}}$$

which in turn implies

$$\begin{aligned}
\mathbb{E}[\mathbf{1}_{\{N_t=n\}}\Phi(\tau_1, \cdots, \tau_n)] &= \mathbb{E}[\mathbf{1}_{\{\tau_n\leqslant t<\tau_{n+1}\}}\Phi(\tau_1, \cdots, \tau_n)] \\
&= \int_t^\infty \int_{\Delta_n(t)} \Phi(s_1, \cdots, s_n) \prod_{j=0}^{n} f(s_{j+1} - s_j)\, d\mathbf{s}_{n+1}
\end{aligned}$$

with the convention $s_0 = 0$. Hence, by Fubini's theorem, it holds

$$\mathbb{E}[\mathbf{1}_{\{N_t=n\}}\Phi(\tau_1, \cdots, \tau_n)] = \int_{\Delta_n(t)} \Phi(s_1, \cdots, s_n)(1 - F(t - s_n)) \prod_{j=0}^{n-1} f(s_{j+1} - s_j)\, d\mathbf{s}_n \tag{4.2.6}$$

for any measurable map $\Phi : \Delta_n(t) \to \mathbb{R}$ satisfying $\mathbb{E}[\mathbf{1}_{\{N_t=n\}}|\Phi(\tau_1, \cdots, \tau_n)|] < \infty$.

Usual choices that we will consider are the followings.

Example 4.2.2 *1. If the density function f is given by $f(t) = \lambda e^{-\lambda t}\mathbf{1}_{[0,\infty)}(t)$ for some positive parameter λ, then N is a Poisson process with intensity λ.*

2. If the density function f is given by $f(t) = \frac{1-\alpha}{\bar{\tau}^{1-\alpha}}\frac{1}{t^\alpha}\mathbf{1}_{[0,\bar{\tau}]}(t)$ for some parameters $(\alpha, \bar{\tau}) \in (0,1)\times(T,\infty)$, then N is a renewal process with $[0,\bar{\tau}]$-valued Beta$(1-\alpha, 1)$ jump times.

3. More generally, if the density function f is given by $f(t) = \frac{\bar{\tau}^{1-\alpha-\beta}}{B(\alpha,\beta)}\frac{1}{t^{1-\alpha}(\bar{\tau}-t)^{1-\beta}}\mathbf{1}_{[0,\bar{\tau}]}(t)$ for some parameters $(\alpha, \beta, \bar{\tau}) \in (0,1)^2 \times (T,\infty)$, then N is a renewal process with $[0,\bar{\tau}]$-valued Beta(α, β) jump times.

Given a sequence $Z = (Z_n^1, Z_n^2)_{n\geqslant 1}$ of i.i.d. random vector with law $\mathcal{N}(0, I_2)$ which is independent of (W, B) and a renewal process N independent of Z with jump times $(\tau_i)_{i\geqslant 0}$, we set $\zeta_i = \tau_i \wedge T$, with the convention $\zeta_0 = 0$, and we consider the two-dimensional Markov chain $(\bar{X}, \bar{Y})$ with $(\bar{X}_0, \bar{Y}_0) = (x_0, y_0)$ at time 0 (evolving on the random time grid $(\zeta_i)_{i\geqslant 0}$) and with dynamics for any $0 \leqslant i \leqslant N_T$

$$\left\{\begin{array}{ll} \bar{X}_{i+1} & = \bar{X}_i + \left(r(\zeta_{i+1}-\zeta_i) - \frac{1}{2}a_{S,i}\right) + \sigma_{S,i}Z_{i+1}^1, \\ \bar{Y}_{i+1} & = m_i + \sigma_{Y,i}\left(\rho_i Z_{i+1}^1 + \sqrt{1-\rho_i^2}Z_{i+1}^2\right), \end{array}\right. \tag{4.2.7}$$

where we introduced the notations

$$\begin{aligned} a_{S,i} := \sigma_{S,i}^2 &:= a_{S,\zeta_{i+1}-\zeta_i}(\bar{Y}_i) = \int_0^{\zeta_{i+1}-\zeta_i} a_S(m_s(\bar{Y}_i))\,ds, \\ a_{Y,i} := \sigma_{Y,i}^2 &:= a_{Y,\zeta_{i+1}-\zeta_i}(\bar{Y}_i) = \int_0^{\zeta_{i+1}-\zeta_i} a_Y(m_s(\bar{Y}_i))\,ds, \\ \sigma_{S,Y,i} &:= \int_0^{\zeta_{i+1}-\zeta_i} (\sigma_S\sigma_Y)(m_s(\bar{Y}_i))\,ds, \\ \rho_i &:= \rho_{\zeta_{i+1}-\zeta_i}(\bar{Y}_i) = \rho\frac{\sigma_{S,Y,i}}{\sigma_{S,i}\sigma_{Y,i}}, \\ m_i &:= m_{\zeta_{i+1}-\zeta_i}(\bar{Y}_i). \end{aligned}$$

We will denote by $\sigma'_{S,i}$ the first derivative of $y \mapsto \sigma_{S,i}(y)$ taken at $\bar{Y}_i$ and proceed similarly for the quantities $\sigma'_{Y,i}$, $\sigma'_{S,Y,i}$, ρ'_i and m'_i. We define the filtration $\mathcal{G} = (\mathcal{G}_i)_{i\geqslant 0}$ where $\mathcal{G}_i = \sigma(Z_j^1, Z_j^2,\ 1 \leqslant j \leqslant i)$, for $i \geqslant 1$ and $\mathcal{G}_0$ stands for the trivial σ-field. We assume that the filtration $\mathcal{G}$ satisfies the usual conditions. For an integer n, we will use the notations $\zeta^n = (\zeta_0, \cdots, \zeta_n)$ and $\tau^n = (\tau_0, \cdots, \tau_n)$.

4.2.4 Tailor-made Malliavin calculus for the Markov chain $(\bar{X}, \bar{Y})$.

In this section we introduce a tailor-made Malliavin calculus for the underlying Markov chain $(\bar{X}, \bar{Y})$ defined by (4.2.7) which will be employed in order to establish our IBP formulae. Instead of using an infinite dimensional calculus as it is usually done in the literature, see e.g. Nualart [78], the approach developed below is based on a finite dimensional calculus for which the dimension is given by the number of jumps of the underlying renewal process involved in the Markov chain $(\bar{X}, \bar{Y})$.

Definition 4.2.3 *Let $n \in \mathbb{N}$. For any $i \in \{0, \cdots, n\}$, we define the set $\mathbb{S}_{i,n}(\bar{X}, \bar{Y})$, as the space of random variables H such that*

- *$H = h(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1})$, on the set $\{N_T = n\}$, where we recall $\zeta^{n+1} := (\zeta_0, \cdots, \zeta_{n+1}) = (0, \zeta_1, \cdots, \zeta_n, T)$.*
- *For any $\mathbf{s}_{n+1} \in \Delta_{n+1}(T)$, the map $h(.,.,.,.,\mathbf{s}_{n+1}) \in \mathcal{C}_p^\infty(\mathbb{R}^4)$.*

For a r.v. $H \in \mathbb{S}_{i,n}(\bar{X}, \bar{Y})$, we will often abuse the notations and write

$$H \equiv H(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1})$$

that is the same symbol H may denote the r.v. or the function in the set $\mathbb{S}_{i,n}(\bar{X}, \bar{Y})$. One can easily define the flow derivatives for $H \in \mathbb{S}_{i,n}(\bar{X}, \bar{Y})$ as follows

$$\begin{aligned}
\partial_{\bar{X}_{i+1}} H &= \partial_3 h(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1}),\\
\partial_{\bar{Y}_{i+1}} H &= \partial_4 h(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1}),\\
\partial_{\bar{X}_i} H &= \partial_1 h(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1}) + \partial_3 h(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1}) \partial_{\bar{X}_i} \bar{X}_{i+1},\\
\partial_{\bar{Y}_i} H &= \partial_2 h(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1}) + \partial_3 h(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1}) \partial_{\bar{Y}_i} \bar{X}_{i+1}\\
&\quad + \partial_4 h(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1}) \partial_{\bar{Y}_i} \bar{Y}_{i+1},
\end{aligned}$$

and from the dynamics (4.2.7)

$$\partial_{\bar{X}_i} \bar{X}_{i+1} = 1,$$

$$\partial_{\bar{Y}_i} \bar{Y}_{i+1} = m_i' + \sigma_{Y,i}' \Big(\rho_i Z_{i+1}^1 + \sqrt{1-\rho_i^2} Z_{i+1}^2 \Big) + \sigma_{Y,i} \frac{\rho_i'}{\sqrt{1-\rho_i^2}} \left(\sqrt{1-\rho_i^2} Z_{i+1}^1 - \rho_i Z_{i+1}^2 \right), \tag{4.2.8}$$

$$\partial_{\bar{Y}_i} \bar{X}_{i+1} = -\frac{1}{2} a_{S,i}' + \sigma_{S,i}' Z_{i+1}^1 = -\frac{1}{2} a_{S,i}' + \frac{\sigma_{S,i}'}{\sigma_{S,i}} \Big(\bar{X}_{i+1} - \bar{X}_i - \Big(r(\zeta_{i+1} - \zeta_i) - \frac{1}{2} a_{S,i} \Big) \Big). \tag{4.2.9}$$

We now define the integral and derivative operators for $H \in \mathbb{S}_{i,n}(\bar{X}, \bar{Y})$, as

$$\mathcal{I}_{i+1}^{(1)}(H) = H \Big[\frac{Z_{i+1}^1}{\sigma_{S,i}(1-\rho_i^2)} - \frac{\rho_i}{1-\rho_i^2} \frac{\rho_i Z_{i+1}^1 + \sqrt{1-\rho_i^2} Z_{i+1}^2}{\sigma_{S,i}} \Big] - \mathcal{D}_{i+1}^{(1)} H, \tag{4.2.10}$$

$$\mathcal{I}_{i+1}^{(2)}(H) = H \Big[\frac{\rho_i Z_{i+1}^1 + \sqrt{1-\rho_i^2} Z_{i+1}^2}{\sigma_{Y,i}(1-\rho_i^2)} - \frac{\rho_i}{1-\rho_i^2} \frac{Z_{i+1}^1}{\sigma_{Y,i}} \Big] - \mathcal{D}_{i+1}^{(2)} H, \tag{4.2.11}$$

$$\mathcal{D}_{i+1}^{(1)} H = \partial_{\bar{X}_{i+1}} H, \tag{4.2.12}$$

$$\mathcal{D}_{i+1}^{(2)} H = \partial_{\bar{Y}_{i+1}} H. \tag{4.2.13}$$

Note that due to the above definitions and assumptions **(AR)** and **(ND)**, it is readily checked that $\mathcal{I}_{i+1}^{(1)}(H)$, $\mathcal{I}_{i+1}^{(2)}(H)$, $\mathcal{D}_{i+1}^{(1)} H$ and $\mathcal{D}_{i+1}^{(2)} H$ are elements of $\mathbb{S}_{i,n}(\bar{X}, \bar{Y})$ so that we can define iterations of the above operators. Namely, by induction, for

a multi-index $\alpha = (\alpha_1, \cdots, \alpha_p)$ of length p with $\alpha_i \in \{1,2\}$ and $\alpha_{p+1} \in \{1,2\}$, we define

$$\mathcal{I}_{i+1}^{(\alpha,\alpha_{p+1})}(H) = \mathcal{I}_{i+1}^{(\alpha_{p+1})}(\mathcal{I}_{i+1}^{(\alpha)}(H)), \quad \mathcal{D}_{i+1}^{(\alpha,\alpha_{p+1})}H = \mathcal{D}_{i+1}^{(\alpha_{p+1})}(\mathcal{D}_{i+1}^{(\alpha)}H)$$

with the intuitive notation $(\alpha, \alpha_{p+1}) = (\alpha_1, \cdots, \alpha_{p+1})$.

Throughout the chapter, we will use the following notation for a certain type of conditional expectation that will be frequently employed. For any $X \in L^1(\mathbb{P})$ and any $i \in \{0, \cdots, n\}$,

$$\mathbb{E}_{i,n}[X] = \mathbb{E}[X|\mathcal{G}_i, \tau^{n+1}, N_T = n]$$

where we recall that we employ the notation $\tau^{n+1} = (\tau_0, \cdots, \tau_{n+1})$. Having the above definitions and notations at hand, the following duality formula is satisfied: for any non-empty multi-index α of length p, with $\alpha_i \in \{1,2\}$, for any $i \in \{1, \cdots, p\}$, p being a positive integer, it holds

$$\mathbb{E}_{i,n}\Big[\mathcal{D}_{i+1}^{(\alpha)} f(\bar{X}_{i+1}, \bar{Y}_{i+1})H\Big] = \mathbb{E}_{i,n}\Big[f(\bar{X}_{i+1}, \bar{Y}_{i+1})\mathcal{I}_{i+1}^{(\alpha)}(H)\Big]. \tag{4.2.14}$$

In order to obtain explicit norm estimates for random variables in $\mathbb{S}_{i,n}(\bar{X}, \bar{Y})$, it is useful to define for $H \in \mathbb{S}_{i,n}(\bar{X}, \bar{Y})$, $i \in \{0, \cdots, n\}$ and $p \geqslant 1$

$$\|H\|_{p,i,n}^p = \mathbb{E}_{i,n}[|H|^p].$$

We will also employ a chain rule formula for the integral operators defined above.

Lemma 4.2.2 *Let $H \equiv H(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1}) \in \mathbb{S}_{i,n}(\bar{X}, \bar{Y})$, for some $i \in \{0, \cdots, n\}$. The following chain rule formulae hold for any $(\alpha_1, \alpha_2) \in \{1,2\}^2$*

$$\partial_{\bar{X}_i}\mathcal{I}_{i+1}^{(\alpha_1)}(H) = \mathcal{I}_{i+1}^{(\alpha_1)}(\partial_{\bar{X}_i}H), \quad \partial_{\bar{X}_i}\mathcal{I}_{i+1}^{(\alpha_1,\alpha_2)}(H) = \mathcal{I}_{i+1}^{(\alpha_1,\alpha_2)}(\partial_{\bar{X}_i}H). \tag{4.2.15}$$

Moreover, one has

$$\partial_{\bar{Y}_i}\mathcal{I}_{i+1}^{(1)}(H) = \mathcal{I}_{i+1}^{(1)}(\partial_{\bar{Y}_i}H) - \frac{\sigma'_{S,i}}{\sigma_{S,i}}\mathcal{I}_{i+1}^{(1)}(H) - \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}\mathcal{I}_{i+1}^{(2)}(H), \tag{4.2.16}$$

$$\partial_{\bar{Y}_i}\mathcal{I}_{i+1}^{(2)}(H) = \mathcal{I}_{i+1}^{(2)}(\partial_{\bar{Y}_i}H) - \left(\frac{\sigma'_{Y,i}}{\sigma_{Y,i}} - \frac{\rho'_i\rho_i}{1-\rho_i^2}\right)\mathcal{I}_{i+1}^{(2)}(H), \tag{4.2.17}$$

$$\partial_{\bar{Y}_i}\mathcal{I}_{i+1}^{(1,1)}(H) = \mathcal{I}_{i+1}^{(1,1)}(\partial_{\bar{Y}_i}H) - 2\frac{\sigma'_{S,i}}{\sigma_{S,i}}\mathcal{I}_{i+1}^{(1,1)}(H) - \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}\left(\mathcal{I}_{i+1}^{(1,2)}(H) + \mathcal{I}_{i+1}^{(2,1)}(H)\right), \tag{4.2.18}$$

$$\partial_{\bar{Y}_i}\mathcal{I}_{i+1}^{(2,2)}(H) = \mathcal{I}_{i+1}^{(2,2)}(\partial_{\bar{Y}_i}H) - 2\left(\frac{\sigma'_{Y,i}}{\sigma_{Y,i}} - \frac{\rho'_i\rho_i}{1-\rho_i^2}\right)\mathcal{I}_{i+1}^{(2,2)}(H), \tag{4.2.19}$$

$$\partial_{\bar{Y}_i}\mathcal{I}_{i+1}^{(1,2)}(H) = \mathcal{I}_{i+1}^{(1,2)}(\partial_{\bar{Y}_i}H) - \left(\frac{\sigma'_{S,i}}{\sigma_{S,i}} + \frac{\sigma'_{Y,i}}{\sigma_{Y,i}} - \frac{\rho'_i\rho_i}{1-\rho_i^2}\right)\mathcal{I}_{i+1}^{(1,2)}(H) - \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}\mathcal{I}_{i+1}^{(2,2)}(H). \tag{4.2.20}$$

Proof. Observe that from the very definitions (4.2.10) and (4.2.11), one directly gets

$$\partial_{\bar{X}_i}\mathcal{I}^{(1)}_{i+1}(1) = \partial_{\bar{X}_i}\mathcal{I}^{(2)}_{i+1}(1) = 0$$

while, also by direct computation, we obtain

$$\begin{aligned}
\partial_{\bar{Y}_i}\mathcal{I}^{(1)}_{i+1}(1) &= -\frac{\sigma'_{S,i}}{\sigma_{S,i}}\mathcal{I}^{(1)}_{i+1}(1) - \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}\mathcal{I}^{(2)}_{i+1}(1),\\
\partial_{\bar{Y}_i}\mathcal{I}^{(2)}_{i+1}(1) &= -\left(\frac{\sigma'_{Y,i}}{\sigma_{Y,i}} - \frac{\rho'_i\rho_i}{1-\rho_i^2}\right)\mathcal{I}^{(2)}_{i+1}(1).
\end{aligned}$$

We thus deduce

$$\begin{aligned}
\partial_{\bar{X}_i}\mathcal{I}^{(\alpha_1)}_{i+1}(H) &= \partial_{\bar{X}_i}H\mathcal{I}^{(\alpha_1)}_{i+1}(1) + H\partial_{\bar{X}_i}\mathcal{I}^{(\alpha_1)}_{i+1}(1) - \partial_{\bar{X}_i}\mathcal{D}^{(\alpha_1)}_{i+1}H\\
&= \partial_{\bar{X}_i}H\mathcal{I}^{(\alpha_1)}_{i+1}(1) - \mathcal{D}^{(\alpha_1)}_{i+1}(\partial_{\bar{X}_i}H)\\
&= \mathcal{I}^{(\alpha_1)}_{i+1}(\partial_{\bar{X}_i}H)
\end{aligned}$$

where we used the fact $\mathcal{D}^{(\alpha_1)}_{i+1}\partial_{\bar{X}_i}H = \partial_{\bar{X}_i}\mathcal{D}^{(\alpha_1)}_{i+1}H$ which easily follows by direct computation. As a consequence, it is readily seen that

$$\begin{aligned}
\partial_{\bar{X}_i}\mathcal{I}^{(\alpha_1,\alpha_2)}_{i+1}(H) &= \partial_{\bar{X}_i}\mathcal{I}^{(\alpha_2)}_{i+1}(\mathcal{I}^{(\alpha_1)}_{i+1}(H)) = \mathcal{I}^{(\alpha_2)}_{i+1}(\partial_{\bar{X}_i}\mathcal{I}^{(\alpha_1)}_{i+1}(H)) = \mathcal{I}^{(\alpha_2)}_{i+1}(\mathcal{I}^{(\alpha_1)}_{i+1}(\partial_{\bar{X}_i}H))\\
&= \mathcal{I}^{(\alpha_1,\alpha_2)}_{i+1}(\partial_{\bar{X}_i}H).
\end{aligned}$$

This concludes the proof of (4.2.15). The chain rule formulae (4.2.16), (4.2.17), (4.2.18), (4.2.19) and (4.2.20) follow from similar arguments. Let us prove (4.2.16) and (4.2.17). The proofs of (4.2.18), (4.2.19) and (4.2.20) are omitted. Observe first that in general $\mathcal{D}^{(\alpha_1)}_{i+1}\partial_{\bar{Y}_i}H \neq \partial_{\bar{Y}_i}\mathcal{D}^{(\alpha_1)}_{i+1}H$. Indeed, by standard computations, it holds

$$\begin{aligned}
\partial_{\bar{Y}_i}\mathcal{D}^{(1)}_{i+1}H &= \partial_{\bar{Y}_i}\partial_{\bar{X}_{i+1}}h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\\
&= \partial^2_{2,3}h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1}) + \partial^2_3h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\partial_{\bar{Y}_i}\bar{X}_{i+1}\\
&\quad + \partial^2_{4,3}h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\partial_{\bar{Y}_i}\bar{Y}_{i+1},\\
\mathcal{D}^{(1)}_{i+1}\partial_{\bar{Y}_i}H &= \partial_{\bar{X}_{i+1}}\partial_{\bar{Y}_i}h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\\
&= \partial_{\bar{X}_{i+1}}(\partial_2h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1}) + \partial_3h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\partial_{\bar{Y}_i}\bar{X}_{i+1}\\
&\quad + \partial_4h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\partial_{\bar{Y}_i}\bar{Y}_{i+1})\\
&= \partial^2_{3,2}h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1}) + \partial^2_3h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\partial_{\bar{Y}_i}\bar{X}_{i+1}\\
&\quad + \partial_3h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\partial_{\bar{X}_{i+1}}\partial_{\bar{Y}_i}\bar{X}_{i+1}\\
&\quad + \partial^2_{3,4}h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\partial_{\bar{Y}_i}\bar{Y}_{i+1}\\
&\quad + \partial_4h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\partial_{\bar{X}_{i+1}}\partial_{\bar{Y}_i}\bar{Y}_{i+1}\\
&= \partial_{\bar{Y}_i}\mathcal{D}^{(1)}_{i+1}H + \partial_3h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\partial_{\bar{X}_{i+1}}\partial_{\bar{Y}_i}\bar{X}_{i+1}\\
&\quad + \partial_4h(\bar{X}_i,\bar{Y}_i,\bar{X}_{i+1},\bar{Y}_{i+1},\zeta^{n+1})\partial_{\bar{X}_{i+1}}\partial_{\bar{Y}_i}\bar{Y}_{i+1}\\
&= \partial_{\bar{Y}_i}\mathcal{D}^{(1)}_{i+1}H + \mathcal{D}^{(1)}_{i+1}H\partial_{\bar{X}_{i+1}}\partial_{\bar{Y}_i}\bar{X}_{i+1} + \mathcal{D}^{(2)}_{i+1}H\partial_{\bar{X}_{i+1}}\partial_{\bar{Y}_i}\bar{Y}_{i+1}\\
&= \partial_{\bar{Y}_i}\mathcal{D}^{(1)}_{i+1}H + \frac{\sigma'_{S,i}}{\sigma_{S,i}}\mathcal{D}^{(1)}_{i+1}H + \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}\mathcal{D}^{(2)}_{i+1}H
\end{aligned}$$

where we used the two identities $\partial_{\bar{X}_{i+1}}\partial_{\bar{Y}_i}\bar{X}_{i+1} = \frac{\sigma'_{S,i}}{\sigma_{S,i}}$ and $\partial_{\bar{X}_{i+1}}\partial_{\bar{Y}_i}\bar{Y}_{i+1} = \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}$ which readily stems from (4.2.8), (4.2.9) and the dynamics (4.2.7).

From (4.2.10) and the previous identity, we thus obtain

$$\begin{aligned}
\partial_{\bar{Y}_i}\mathcal{I}^{(1)}_{i+1}(H) &= \partial_{\bar{Y}_i}\mathcal{I}^{(1)}_{i+1}(1)H + \mathcal{I}^{(1)}_{i+1}(1)\partial_{\bar{Y}_i}H - \partial_{\bar{Y}_i}\mathcal{D}^{(1)}_{i+1}H \\
&= -\frac{\sigma'_{S,i}}{\sigma_{S,i}}\mathcal{I}^{(1)}_{i+1}(1)H - \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}\mathcal{I}^{(2)}_{i+1}(1)H + \mathcal{I}^{(1)}_{i+1}(1)\partial_{\bar{Y}_i}H - \mathcal{D}^{(1)}_{i+1}\partial_{\bar{Y}_i}H \\
&\quad + \frac{\sigma'_{S,i}}{\sigma_{S,i}}\mathcal{D}^{(1)}_{i+1}H + \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}\mathcal{D}^{(2)}_{i+1}H \\
&= -\frac{\sigma'_{S,i}}{\sigma_{S,i}}\left(\mathcal{I}^{(1)}_{i+1}(1)H - \mathcal{D}^{(1)}_{i+1}H\right) + \mathcal{I}^{(1)}_{i+1}(1)\partial_{\bar{Y}_i}H - \mathcal{D}^{(1)}_{i+1}\partial_{\bar{Y}_i}H \\
&\quad - \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}\left(\mathcal{I}^{(2)}_{i+1}(1)H - \mathcal{D}^{(2)}_{i+1}H\right) \\
&= \mathcal{I}^{(1)}_{i+1}(\partial_{\bar{Y}_i}H) - \frac{\sigma'_{S,i}}{\sigma_{S,i}}\mathcal{I}^{(1)}_{i+1}(H) - \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}\mathcal{I}^{(2)}_{i+1}(H).
\end{aligned}$$

Similarly, after some algebraic manipulations using (4.2.7) and (4.2.8), we get $\partial_{\bar{Y}_{i+1}}\partial_{\bar{Y}_i}\bar{Y}_{i+1} = \frac{\sigma'_{Y,i}}{\sigma_{Y,i}} - \frac{\rho'_i\rho_i}{1-\rho_i^2}$ so that

$$\mathcal{D}^{(2)}_{i+1}\partial_{\bar{Y}_i}H = \partial_{\bar{Y}_i}\mathcal{D}^{(2)}_{i+1}H + \mathcal{D}^{(2)}_{i+1}H\partial_{\bar{Y}_{i+1}}\partial_{\bar{Y}_i}\bar{Y}_{i+1} = \partial_{\bar{Y}_i}\mathcal{D}^{(2)}_{i+1}H + \left(\frac{\sigma'_{Y,i}}{\sigma_{Y,i}} - \frac{\rho'_i\rho_i}{1-\rho_i^2}\right)\mathcal{D}^{(2)}_{i+1}H$$

so that, omitting some technical details, we get

$$\begin{aligned}
\partial_{\bar{Y}_i}\mathcal{I}^{(2)}_{i+1}(H) &= \mathcal{I}^{(2)}_{i+1}(\partial_{\bar{Y}_i}H) - \left(\frac{\sigma'_{Y,i}}{\sigma_{Y,i}} - \frac{\rho'_i\rho_i}{1-\rho_i^2}\right)\mathcal{I}^{(2)}_{i+1}(1)H + \left(\frac{\sigma'_{Y,i}}{\sigma_{Y,i}} - \frac{\rho'_i\rho_i}{1-\rho_i^2}\right)\mathcal{D}^{(2)}_{i+1}H \\
&= \mathcal{I}^{(2)}_{i+1}(\partial_{\bar{Y}_i}H) - \left(\frac{\sigma'_{Y,i}}{\sigma_{Y,i}} - \frac{\rho'_i\rho_i}{1-\rho_i^2}\right)\mathcal{I}^{(2)}_{i+1}(H).
\end{aligned}$$

The identities (4.2.18), (4.2.19) and (4.2.20) eventually follows from (4.2.16) and (4.2.17) using some simple algebraic computations.

We conclude this section by introducing the following space of random variables which satisfy some time regularity estimates.

Definition 4.2.4 *Let $\ell \in \mathbb{Z}$ and $n \in \mathbb{N}$. For any $i \in \{0, \cdots, n\}$, we define the space $\mathbb{M}_{i,n}(\bar{X}, \bar{Y}, \ell/2)$ as the set of finite random variables $H \in \mathbb{S}_{i,n}(\bar{X}, \bar{Y})$ satisfying the following time regularity estimate: for any $p \geqslant 1$, for any $c > 0$, there exists some positive constants $C := C(T, c)$, $c' := c'(c)$, $T \mapsto C(T, c)$ being non-decreasing and c' being independent of T, such that for any $(x_i, y_i, x_{i+1}, y_{i+1}, \mathbf{s}_{n+1}) \in \mathbb{R}^4 \times \Delta_{n+1}(T)$,*

$$\begin{aligned}
|H(x_i, y_i, x_{i+1}, y_{i+1}, \mathbf{s}_{n+1})|^p \bar{q}_c(s_{i+1} - s_i, x_i, y_i, x_{i+1}, y_{i+1}) \\
\leqslant C(s_{i+1} - s_i)^{\frac{p\ell}{2}} \bar{q}_{c'}(s_{i+1} - s_i, x_i, y_i, x_{i+1}, y_{i+1})
\end{aligned} \tag{4.2.21}$$

where the density function $\mathbb{R}^2 \ni (x_{i+1}, y_{i+1}) \mapsto \bar{q}_c(s_{i+1} - s_i, x_i, y_i, x_{i+1}, y_{i+1})$ is defined in Lemma 4.2.1.

We again remark that since the space $\mathbb{M}_{i,n}(\bar{X},\bar{Y},\ell/2)$ is a subset of $\mathbb{S}_{i,n}(\bar{X},\bar{Y})$, when we say that a random variable $\mathbb{M}_{i,n}(\bar{X},\bar{Y},\ell/2)$ this statement is always understood on the set $\{N_T = n\}$.

Before proceeding, let us provide a simple example of some random variables that belong to the aforementioned space. From (4.2.10) and the dynamics (4.2.7) of the Markov chain $(\bar{X},\bar{Y})$, it holds

$$\mathcal{I}^{(1)}_{i+1}(1) = \frac{\bar{X}_{i+1}-\bar{X}_i-(r(\zeta_{i+1}-\zeta_i)-\frac{1}{2}a_{S,i})}{a_{S,i}(1-\rho_i^2)} - \frac{\rho_i}{1-\rho_i^2}\frac{\bar{Y}_{i+1}-m_i}{\sigma_{S,i}\sigma_{Y,i}},$$
$$\mathcal{I}^{(1,1)}_{i+1}(1) = (\mathcal{I}^{(1)}_{i+1}(1))^2 - \mathcal{D}^{(1)}_{i+1}(\mathcal{I}^{(1)}_{i+1}(1)) = (\mathcal{I}^{(1)}_{i+1}(1))^2 - \frac{1}{a_{S,i}(1-\rho_i^2)},$$

so that, $\mathcal{I}^{(1)}_{i+1}(1)$ and $\mathcal{I}^{(1,1)}_{i+1}(1)$ belong to $\mathbb{S}_{i,n}(\bar{X},\bar{Y})$. Moreover, under **(ND)**, for any $p \geqslant 1$, it holds

$$\left|\mathcal{I}^{(1)}_{i+1}(1)(x_i,y_i,x_{i+1},y_{i+1},\mathbf{s}_{n+1})\right|^p \leqslant C\Big(1+\frac{|x_{i+1}-x_i|^p}{(s_{i+1}-s_i)^p}+\frac{|y_{i+1}-m_i(y_i)|^p}{(s_{i+1}-s_i)^p}\Big)$$

and similarly,

$$\left|\mathcal{I}^{(1,1)}_{i+1}(1)(x_i,y_i,x_{i+1},y_{i+1},\mathbf{s}_{n+1})\right|^p \leqslant C\Big(1+\frac{1}{(s_{i+1}-s_i)^p}+\frac{|x_{i+1}-x_i|^{2p}}{(s_{i+1}-s_i)^{2p}}+\frac{|y_{i+1}-m_i(y_i)|^{2p}}{(s_{i+1}-s_i)^{2p}}\Big).$$

Hence, from the space-time inequality (4.1.3), for any $c>0$ and any $c'>c$, it holds

$$\begin{aligned}&\left|\mathcal{I}^{(1)}_{i+1}(1)(x_i,y_i,x_{i+1},y_{i+1},\mathbf{s}_{n+1})\right|^p \bar{q}_c(s_{i+1}-s_i,x_i,y_i,x_{i+1},y_{i+1})\\ &\leqslant C(s_{i+1}-s_i)^{-\frac{p}{2}}\bar{q}_{c'}(s_{i+1}-s_i,x_i,y_i,x_{i+1},y_{i+1})\end{aligned}$$

and

$$\begin{aligned}&\left|\mathcal{I}^{(1,1)}_{i+1}(1)(x_i,y_i,x_{i+1},y_{i+1},\mathbf{s}_{n+1})\right|^p \bar{q}_c(s_{i+1}-s_i,x_i,y_i,x_{i+1},y_{i+1})\\ &\leqslant C(s_{i+1}-s_i)^{-p}\bar{q}_{c'}(s_{i+1}-s_i,x_i,y_i,x_{i+1},y_{i+1})\end{aligned}$$

for some positive constant $C := C(T)$, $T \mapsto C(T)$ being non-decreasing. We thus conclude that $\mathcal{I}^{(1)}_{i+1}(1) \in \mathbb{M}_{i,n}(\bar{X},\bar{Y},-1/2)$ and $\mathcal{I}^{(1,1)}_{i+1}(1) \in \mathbb{M}_{i,n}(\bar{X},\bar{Y},-1)$ for any $i \in \{0,\cdots,n\}$.

A straightforward generalization of the above example is the following property that will be frequently used in the sequel. We omit its proof.

Lemma 4.2.3 *Fix $n \in \mathbb{N}$ and $i \in \{0,\cdots,n\}$.*

- *Let $\ell_1,\ell_2 \in \mathbb{Z}$ and $H_1 \in \mathbb{M}_{i,n}(\bar{X},\bar{Y},\ell_1/2), H_2 \in \mathbb{M}_{i,n}(\bar{X},\bar{Y},\ell_2/2)$. Then, one has $H_1H_2 \in \mathbb{M}_{i,n}(\bar{X},\bar{Y},(\ell_1+\ell_2)/2)$.*

- *Let $\ell \in \mathbb{Z}$ and $H \in \mathbb{M}_{i,n}(\bar{X},\bar{Y},\ell/2)$ such that $\mathcal{D}^{(\alpha_1)}_{i+1}H \in \mathbb{M}_{i,n}(\bar{X},\bar{Y},\ell'/2)$ for some $\alpha_1 \in \{1,2\}$ and $\ell' \in \mathbb{Z}$.*
  - It holds that $\mathcal{I}^{(\alpha_1)}_{i+1}(H) \in \mathbb{M}_{i,n}(\bar{X},\bar{Y},((\ell-1)\wedge\ell')/2)$ and $(\zeta_{i+1}-\zeta_i)\mathcal{I}^{(\alpha_1)}_{i+1}(H_1) \in$

$\mathbb{M}_{i,n}(\bar{X}, \bar{Y}, ((\ell+1) \wedge (\ell'+2))/2)$.
- Assume additionally that $\mathcal{D}_{i+1}^{(\alpha_1,\alpha_2)} H \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, \ell''/2)$ *for some* $\ell'' \in \mathbb{Z}$ *and* $\alpha_2 \in \{1,2\}$. *Then it holds that* $\mathcal{I}_{i+1}^{(\alpha_1,\alpha_2)}(H) \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, ((\ell-2) \wedge (\ell'-1) \wedge \ell'')/2)$.

Finally, we importantly emphasize that if $H \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, \ell/2)$ for some $n \in \mathbb{N}$, $i \in \{0, \cdots, n\}$ and $\ell \in \mathbb{Z}$, then, its conditional $L^p(\mathbb{P})$-moment is finite and also satisfies a time regularity estimate. More precisely, for any $p \geqslant 1$, it holds

$$\|H\|_{p,i,n} \leqslant C(\zeta_{i+1} - \zeta_i)^{\ell/2} \tag{4.2.22}$$

for some positive constant $C := C(T)$, $T \mapsto C(T)$ being non-decreasing. Indeed, using the fact that the sequence Z is independent of N as well as the upper-estimate (4.2.4) of Lemma 4.2.1 and finally (4.2.21), one directly gets

$$\begin{aligned}
\|H\|_{p,i,n}^p &= \mathbb{E}\Big[|H(\bar{X}_i, \bar{Y}_i, \bar{X}_{i+1}, \bar{Y}_{i+1}, \zeta^{n+1})|^p \Big| \bar{X}_i, \bar{Y}_i, \tau^{n+1}, N_T = n\Big] \\
&= \int_{\mathbb{R}^2} |H(\bar{X}_i, \bar{Y}_i, x_{i+1}, y_{i+1}, \zeta^{n+1})|^p \bar{p}(\zeta_{i+1} - \zeta_i, \bar{X}_i, \bar{Y}_i, x_{i+1}, y_{i+1})\, dx_{i+1} dy_{i+1} \\
&\leqslant C \int_{\mathbb{R}^2} |H(\bar{X}_i, \bar{Y}_i, x_{i+1}, y_{i+1}, \zeta^{n+1})|^p \bar{q}_{4\kappa}(\zeta_{i+1} - \zeta_i, \bar{X}_i, \bar{Y}_i, x_{i+1}, y_{i+1})\, dx_{i+1} dy_{i+1} \\
&\leqslant C(\zeta_{i+1} - \zeta_i)^{p\ell/2}
\end{aligned}$$

so that (4.2.22) directly follows. The previous conditional $L^p(\mathbb{P})$-moment estimate will be used at several places in the sequel.

4.3 Probabilistic representation for the couple (S_T, Y_T).

In this section, we establish a probabilistic representation for the marginal law (S_T, Y_T), or equivalently, for the law of (X_T, Y_T) which is based on the Markov chain $(\bar{X}, \bar{Y})$ introduced in the previous section. For $\gamma > 0$, we denote by $\mathcal{B}_\gamma(\mathbb{R}^2)$ the set of Borel measurable map $h : \mathbb{R}^2 \to \mathbb{R}$ satisfying the following exponential growth assumption at infinity, namely, for some positive constant C, for any $(x, y) \in \mathbb{R}^2$,

$$|h(x, y)| \leqslant C \exp(\gamma(|x|^2 + |y|^2)). \tag{4.3.1}$$

Theorem 4.3.1 *Let $T > 0$. Under assumptions* **(AR)** *and* **(ND)**, *the law of the couple (X_T, Y_T) given by the unique solution to the SDE (4.2.1) at time T starting from $(x_0 = \ln(s_0), y_0)$ at time 0 satisfies the following probabilistic representation: there exists a positive constant $c := c(T, b_Y, \kappa)$ such that for any $0 \leqslant \gamma < c^{-1}$ and any $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$, it holds*

$$\mathbb{E}[h(X_T, Y_T)] = \mathbb{E}\Big[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \prod_{i=1}^{N_T+1} \theta_i\Big] \tag{4.3.2}$$

where the random variables $\theta_i \in \mathbb{S}_{i-1,n}(\bar{X}, \bar{Y})$, $1 \leqslant i \leqslant N_T$ are defined by

$$\theta_i = (f(\zeta_i - \zeta_{i-1}))^{-1}\Big[\mathcal{I}_i^{(1,1)}(c_S^i) - \mathcal{I}_i^{(1)}(c_S^i) + \mathcal{I}_i^{(2,2)}(c_Y^i) + \mathcal{I}_i^{(2)}(b_Y^i) + \mathcal{I}_i^{(1,2)}(c_{Y,S}^i)\Big], \tag{4.3.3}$$

and

$$\theta_{N_T+1} = (1 - F(T - \zeta_{N_T}))^{-1}, \tag{4.3.4}$$

with

$$\begin{aligned} c_S^i &:= \frac{1}{2}\Big(a_S(\bar{Y}_i) - a_S(m_{i-1})\Big), \\ c_Y^i &:= \frac{1}{2}\Big(a_Y(\bar{Y}_i) - a_Y(m_{i-1})\Big), \\ b_Y^i &:= b_Y(\bar{Y}_i) - b_Y(m_{i-1}), \\ c_{Y,S}^i &:= \rho((\sigma_S\sigma_Y)(\bar{Y}_i) - (\sigma_S\sigma_Y)(m_{i-1})). \end{aligned}$$

Assume furthermore that N is a renewal process with $Beta(\alpha, 1)$ jump times. For any $p \geqslant 1$ satisfying $p(\frac{1}{2} - \alpha) \leqslant 1 - \alpha$, for any γ such that $0 \leqslant p\gamma < c^{-1}$ and any $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$, the random variable appearing inside the expectation in the right-hand side of (4.3.2) admits a finite $L^p(\mathbb{P})$-moment. In particular, if $\alpha = 1/2$ then for any $p \geqslant 1$, for any $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$ with $0 \leqslant p\gamma < c^{-1}$, the $L^p(\mathbb{P})$-moment is finite.

The proof of Theorem 4.3.1 is postponed to Appendix 4.6.1.

Remark 4.3.2 *The strategy to establish a probabilistic representation formula for the couple (X_T, Y_T) follows similar lines to the one implemented in [4, 3, 42]. The central argument consists in a perturbation argument of the Markov semigroup associated to the original process $(X_t, Y_t)_{t\geqslant 0}$ around the one generated by an approximation process $(\bar{X}_t, \bar{Y}_t)_{t\geqslant 0}$. As previously mentioned, the main difference here with respect to the aforementioned references lies exactly in the choice of this approximation process around which this perturbation argument is performed, see Section 4.2.2 for its definition. This choice is crucial and leads to a specification of the weights $(\theta_i)_{1\leqslant i\leqslant N_T+1}$, see (4.3.3)-(4.3.4), that is different from the previous works. The main difficulty then consists in proving that the conditional $L^1(\mathbb{P})$-moment of the weights θ_i are of the correct order, that is, they do not lead to a non-integrable time singularity as hinted in the estimate (4.2.21) (with $p = 1$) of Definition 4.2.4. Roughly speaking, these weights are given by Malliavin IBP operators of order 1 or 2 applied to the difference of the coefficients appearing in the infinitesimal generators associated to $(X_t, Y_t)_{t\geqslant 0}$ and $(\bar{X}_t, \bar{Y}_t)_{t\geqslant 0}$. As discussed right after the Definition 4.2.4, the Malliavin IBP operator $\mathcal{I}_{i+1}^{(1,1)}(1) \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, -1)$ so that it generates a non-integrable time singularity of order one and the same conclusion holds true for $\mathcal{I}_{i+1}^{(2,2)}(1)$ and $\mathcal{I}_{i+1}^{(1,2)}(1)$. However, the coefficients c_S^i, c_Y^i, b_Y^i, $c_{Y,S}^i$ appearing inside these Malliavin IBP operators, which write as the difference of the coefficients evaluated along the dynamics (4.2.7) between two consecutive times, allow to remove this time singularity. We refer the reader to the technical Lemma 4.6.2 for a rigorous proof of this claim.*

4.4 Integration by parts formulae

In this section, we establish two IBP formulae for the law of the couple (S_T, Y_T). More precisely, we are interested in providing a Bismut-Elworthy-Li formula for the

two quantities

$$\partial_{s_0}\mathbb{E}[h(S_T, Y_T)], \quad \partial_{y_0}\mathbb{E}[h(S_T, Y_T)].$$

Our strategy is divided into two steps as follows:

Step 1: The first step was performed with the probabilistic representation established in Theorem 4.3.1 for the couple (X_T, Y_T) involving the two-dimensional Markov chain $(\bar{X}, \bar{Y})$ evolving on a time grid governed by the jump times of the renewal process N. Introducing $h(x, y) = f(e^x, y)$ and assuming that f is of polynomial growth at infinity, it is sufficient to consider the two quantities

$$\partial_{s_0}\mathbb{E}\Big[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \prod_{i=1}^{N_T+1} \theta_i\Big], \quad \partial_{y_0}\mathbb{E}\Big[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \prod_{i=1}^{N_T+1} \theta_i\Big]$$

recalling that $x_0 = \ln(s_0)$.

Step 2: At this stage, one might be tempted to perform a standard IBP formula as presented in Nualart [78] on the whole time interval $[0, T]$. However, such a strategy is likely to fail. The main reason is that the Skorokhod integral of the product of weights $\prod_{i=1}^{N_T+1} \theta_i$ will inevitably involve the Malliavin derivative of θ_i which will in turn raise some integrability issues of the resulting Malliavin weight. The key idea that we use in order to circumvent this issue consists in performing local IBP formulae on each of the random intervals $[\zeta_i, \zeta_{i+1}]$, $i = 0, \cdots, N_T$, that is, by using the noise of the Markov chain on this specific time interval and then by combining all these local IBP formulae in a suitable way.

To implement successfully our strategy, two main ingredients are needed. Our first ingredient consists in transferring the partial derivatives ∂_{s_0} and ∂_{y_0} on the expectation forward in time from the first time interval $[0, \zeta_1]$ to the interval on which we perform the local IBP formula, say $[\zeta_i, \zeta_{i+1}]$. Our second ingredient consists in combining these various local IBP formulae in an adequate manner. Roughly speaking, we will consider a weighted sum of each IBP formula, the weight being precisely the length of the corresponding time interval.

4.4.1 The transfer of derivative formula

Lemma 4.4.1 *Let $h \in \mathcal{C}_p^1(\mathbb{R}^2)$ and $n \in \mathbb{N}$. The maps $\mathbb{R}^2 \ni (x, y) \mapsto \mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1}, \bar{Y}_{i+1})\theta_{i+1} | (\bar{X}_i, \bar{Y}_i) = (x, y)\Big]$, $i \in \{0, \cdots, n\}$, belong to $\mathcal{C}_p^1(\mathbb{R}^2)$ a.s. Moreover, the following transfer of derivative formulae hold*

$$\partial_{s_0}\mathbb{E}_{0,n}\Big[h(\bar{X}_1, \bar{Y}_1)\theta_1\Big] = \mathbb{E}_{0,n}\Big[\partial_{\bar{X}_1} h(\bar{X}_1, \bar{Y}_1)\frac{\theta_1}{s_0}\Big] \tag{4.4.1}$$

while for $1 \leqslant i \leqslant n$,

$$\partial_{\bar{X}_i}\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1}, \bar{Y}_{i+1})\theta_{i+1}\Big] = \mathbb{E}_{i,n}\Big[\partial_{\bar{X}_{i+1}} h(\bar{X}_{i+1}, \bar{Y}_{i+1})\theta_{i+1}\Big]. \tag{4.4.2}$$

Similarly, the following transfer of derivative formulae hold: for any $0 \leqslant i \leqslant n-1$

$$\begin{aligned}\partial_{\bar{Y}_i}\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\theta_{i+1}\Big] &= \mathbb{E}_{i,n}\Big[\partial_{\bar{Y}_{i+1}}h(\bar{X}_{i+1},\bar{Y}_{i+1})\,\overrightarrow{\theta}^{e,Y}_{i+1}\Big]\\ &\quad+\mathbb{E}_{i,n}\Big[\partial_{\bar{X}_{i+1}}h(\bar{X}_{i+1},\bar{Y}_{i+1})\,\overrightarrow{\theta}^{e,X}_{i+1}\Big]\\ &\quad+\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\,\overrightarrow{\theta}^{c}_{i+1}\Big] \qquad (4.4.3)\end{aligned}$$

with

$$\begin{aligned}\overrightarrow{\theta}^{e,Y}_{i+1} &= (f(\zeta_{i+1}-\zeta_i))^{-1}\Big[\mathcal{I}^{(1,1)}_{i+1}(d^{i+1}_S)+\mathcal{I}^{(2,2)}_{i+1}(d^{i+1}_Y)+\mathcal{I}^{(1)}_{i+1}(e^{Y,i+1}_S)+\mathcal{I}^{(2)}_{i+1}(e^{Y,i+1}_Y)\\ &\quad+\mathcal{I}^{(1,2)}_{i+1}(d^{i+1}_{Y,S})\Big],\\ \overrightarrow{\theta}^{e,X}_{i+1} &= (f(\zeta_{i+1}-\zeta_i))^{-1}\mathcal{I}^{(1)}_{i+1}(e^{X,i+1}_S),\\ \overrightarrow{\theta}^{c}_{i+1} &= \mathcal{I}^{(1)}_{i+1}\Big(\partial_{\bar{Y}_i}\bar{X}_{i+1}\theta_{i+1}-\overrightarrow{\theta}^{e,X}_{i+1}\Big)+\partial_{\bar{Y}_i}\theta_{i+1}+\mathcal{I}^{(2)}_{i+1}\Big[m_i'\theta_{i+1}-\overrightarrow{\theta}^{e,Y}_{i+1}\\ &\quad+\Big(\sigma'_{Y,i}\Big(\rho_iZ^1_{i+1}+\sqrt{1-\rho_i^2}Z^2_{i+1}\Big)+\sigma_{Y,i}\frac{\rho_i'}{\sqrt{1-\rho_i^2}}\Big(\sqrt{1-\rho_i^2}Z^1_{i+1}-\rho_iZ^2_{i+1}\Big)\Big)\theta_{i+1}\Big],\\ d^{i+1}_S &= m_i'c^{i+1}_S,\\ d^{i+1}_Y &= m_i'c^{i+1}_Y,\\ d^{i+1}_{Y,S} &= m_i'c^{i+1}_{Y,S},\\ e^{Y,i+1}_S &= -m_i'c^{i+1}_S+\partial_{\bar{Y}_i}c^{i+1}_{Y,S},\\ e^{Y,i+1}_Y &= m_i'b^{i+1}_Y+\partial_{\bar{Y}_i}c^{i+1}_Y,\\ e^{X,i+1}_S &= \partial_{\bar{Y}_i}c^{i+1}_S.\end{aligned}$$

For $i=n$*, one also has*

$$\begin{aligned}\partial_{\bar{Y}_n}\mathbb{E}_{n,n}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\theta_{n+1}\Big] &= \mathbb{E}_{n,n}\Big[\partial_{\bar{Y}_{n+1}}h(\bar{X}_{n+1},\bar{Y}_{n+1})\,\overrightarrow{\theta}^{e,Y}_{n+1}\Big]\\ &\quad+\mathbb{E}_{n,n}\Big[\partial_{\bar{X}_{n+1}}h(\bar{X}_{n+1},\bar{Y}_{n+1})\,\overrightarrow{\theta}^{e,X}_{n+1}\Big]\\ &\quad+\mathbb{E}_{n,n}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\,\overrightarrow{\theta}^{c}_{n+1}\Big] \qquad (4.4.4)\end{aligned}$$

with

$$\begin{aligned}\overrightarrow{\theta}^{e,Y}_{n+1} &= (1-F(T-\zeta_n))^{-1}\Big(m_n'+\sigma'_{Y,n}\Big(\rho_nZ^1_{n+1}+\sqrt{1-\rho_n^2}Z^2_{n+1}\Big)\\ &\quad+\sigma_{Y,n}\frac{\rho_n'}{\sqrt{1-\rho_n^2}}\Big(\sqrt{1-\rho_n^2}Z^1_{n+1}-\rho_nZ^2_{n+1}\Big)\Big),\\ \overrightarrow{\theta}^{e,X}_{n+1} &= (1-F(T-\zeta_n))^{-1}\Big(-\frac{1}{2}a'_{S,n}+\sigma'_{S,n}Z^1_{n+1}\Big),\end{aligned}$$

and we set $\overrightarrow{\theta}^{c}_{n+1}=0$ *for notational convenience.*

Finally, the weight sequences $(\overrightarrow{\theta}_i^{e,Y})_{1\leqslant i\leqslant n+1}$, $(\overrightarrow{\theta}_i^{e,X})_{1\leqslant i\leqslant n+1}$ *and* $(\overrightarrow{\theta}_i^{c})_{1\leqslant i\leqslant n+1}$ *defined above satisfy*

$$f(\zeta_i-\zeta_{i-1})\overrightarrow{\theta}_i^{e,Y},\ f(\zeta_i-\zeta_{i-1})\overrightarrow{\theta}_i^{c}\in\mathbb{M}_{i-1,n}(\bar{X},\bar{Y},-1/2),\qquad i\in\{1,\cdots,n\},$$

$$f(\zeta_i-\zeta_{i-1})\overrightarrow{\theta}_i^{e,X}\in\mathbb{M}_{i-1,n}(\bar{X},\bar{Y},0),\qquad i\in\{1,\cdots,n\},$$

and $(1-F(T-\zeta_n))\overrightarrow{\theta}_{n+1}^{e,Y}\in\mathbb{M}_{n,n}(\bar{X},\bar{Y},0)$, $(1-F(T-\zeta_n))\overrightarrow{\theta}_{n+1}^{e,X},\in\mathbb{M}_{n,n}(\bar{X},\bar{Y},1/2)$.

The proof of Lemma 4.4.1 is postponed to Appendix 4.6.2. The transfer of derivative procedure starts on the first time interval $[0,\zeta_1]$ according to formulae (4.4.1) and (4.4.3) (for $i=0$). It expresses the fact that the flow derivatives ∂_{s_0} and ∂_{y_0} of the conditional expectations on the left-hand side of the equations are transferred to derivative operators $\partial_{\bar{X}_1}$ and $\partial_{\bar{Y}_1}$ on the test function h appearing on the right-hand side. Remark that the first derivatives of h have been written ubiquitously as $\partial_{\bar{X}_{i+1}}h(\bar{X}_{i+1},\bar{Y}_{i+1})$ and $\partial_{\bar{Y}_{i+1}}h(\bar{X}_{i+1},\bar{Y}_{i+1})$.

Then, by the Markov property satisfied by the process $(\bar{X},\bar{Y})$, the function h appearing inside the (conditional) expectations on the right-hand side of (4.4.1) and (4.4.3) (for $i=0$) will be given by the conditional expectation appearing on the left-hand side of the same equations but for $i=1$. The transfer of derivative formulae for the following time intervals are obtained by induction using (4.4.2) and (4.4.3) up to the last time interval. Doing so, we obtain various transfer of derivative formulae by transferring successively the derivative operators through all intervals forward in time.

4.4.2 The integration by parts formulae

We first define the weights that will be used in our IBP formulae. For an integer n, on the set $\{N_T=n\}$, for any $k\in\{1,\cdots,n+1\}$ and any $j\in\{1,\cdots,k\}$, we define

$$\begin{aligned}
\overrightarrow{\theta}^{\mathcal{I}_k^{(1),n+1}} &:= \prod_{i=k+1}^{n+1}\theta_i\times\mathcal{I}_k^{(1)}(\theta_k)\times\prod_{i=1}^{k-1}\theta_i,\\
\overrightarrow{\theta}^{C_j^{n+1}} &:= \prod_{i=j+1}^{n+1}\theta_i\times\overrightarrow{\theta}_j^{c}\times\prod_{i=1}^{j-1}\overrightarrow{\theta}_i^{e,Y},\\
\overrightarrow{\theta}^{\mathcal{I}_k^{(2),n+1}} &:= \prod_{i=k+1}^{n+1}\theta_i\times\mathcal{I}_k^{(2)}(\overrightarrow{\theta}_k^{e,Y})\times\prod_{i=1}^{k-1}\overrightarrow{\theta}_i^{e,Y},\\
\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),n+1}} &:= \prod_{i=k+1}^{n+1}\theta_i\times\mathcal{I}_k^{(1)}(\theta_k)\times\prod_{i=j+1}^{k-1}\theta_i\times\overrightarrow{\theta}_j^{e,X}\times\prod_{i=1}^{j-1}\overrightarrow{\theta}_i^{e,Y},\quad j=1,\cdots,k-1,\\
\overrightarrow{\theta}_k^{\mathcal{I}_k^{(1),n+1}} &:= \prod_{i=k+1}^{n+1}\theta_i\times\mathcal{I}_k^{(1)}(\overrightarrow{\theta}_k^{e,X})\times\prod_{i=1}^{k-1}\overrightarrow{\theta}_i^{e,Y}
\end{aligned}$$

with the convention $\prod_{\varnothing}\cdots=1$. Having the above definitions at hand, we are now able to state our IBP formulae.

Theorem 4.4.1 *Let $T > 0$. Under assumptions* **(AR)** *and* **(ND)**, *the law of the couple (X_T, Y_T), given by the unique solution to the SDE (4.2.1) at time T starting from $(x_0 = \ln(s_0), y_0)$ at time 0, satisfies the following Bismut-Elworthy-Li IBP formulae: there exists some positive constant $c := c(T, b_Y, \kappa)$ such that for any $0 \leqslant \gamma < c^{-1}$, any $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$ and any $(s_0, y_0) \in (0,\infty) \times \mathbb{R}$, it holds*

$$s_0 T \partial_{s_0} \mathbb{E}\Big[h(X_T, Y_T)\Big] = \mathbb{E}\Big[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \sum_{k=1}^{N_T+1} (\zeta_k - \zeta_{k-1}) \overrightarrow{\theta}^{\mathcal{I}_k^{(1),N_T+1}}\Big] \tag{4.4.5}$$

and

$$T\partial_{y_0} \mathbb{E}\Big[h(X_T, Y_T)\Big] = \mathbb{E}\Bigg[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1}) \sum_{k=1}^{N_T+1} (\zeta_k - \zeta_{k-1}) \Big(\overrightarrow{\theta}^{\mathcal{I}_k^{(2),N_T+1}} + \sum_{j=1}^{k} \Big(\overrightarrow{\theta}^{C_j^{N_T+1}} + \overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),N_T+1}}\Big)\Big)\Bigg]. \tag{4.4.6}$$

Moreover, if N is a renewal process with $Beta(\alpha, 1)$ jump times, then, for any $p \geqslant 1$ satisfying $p(\frac{1}{2} - \alpha) \leqslant 1 - \alpha$, for any γ such that $0 \leqslant p\gamma < c^{-1}$ and any $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$, the random variables appearing inside the expectation in the right-hand side of (4.4.5) and (4.4.6) admit a finite $L^p(\mathbb{P})$-moment. In particular, if $\alpha = 1/2$ then for any $p \geqslant 1$, for any $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$ with $0 \leqslant p\gamma < c^{-1}$, the $L^p(\mathbb{P})$-moment is finite.

Proof. We only prove the IBP formula (4.4.6). The proof of (4.4.5) follows by completely analogous (and actually more simple) arguments and is thus omitted.

Step 1: proof of the IBP formula (4.4.6) for $h \in \mathcal{C}_b^1(\mathbb{R}^2)$.
Let $h \in \mathcal{C}_b^1(\mathbb{R}^2)$. From Theorem 4.3.1 and Fubini's theorem, we write

$$\mathbb{E}[h(X_T, Y_T)] = \sum_{n \geqslant 0} \mathbb{E}\Big[\mathbb{E}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1}) \prod_{i=1}^{n+1} \theta_i | \tau^{n+1}\Big] \mathbf{1}_{\{N_T = n\}}\Big] \tag{4.4.7}$$

where we used the fact that $\{N_T = n\} = \{\tau_{n+1} > T\} \cap \{\tau_n \leqslant T\}$. In most of the arguments below, we will work on the set $\{N_T = n\}$. In order to perform our induction argument forward in time through the Markov chain structure, we define for $k \in \{0, \cdots, n\}$ the functions

$$\begin{aligned} H_k(\bar{X}_k, \bar{Y}_k) &:= \mathbb{E}_{k,n}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1}) \prod_{i=k+1}^{n+1} \theta_i\Big] \\ &= \mathbb{E}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1}) \prod_{i=k+1}^{n+1} \theta_i | \bar{X}_k, \bar{Y}_k, \tau^{n+1}, N_T = n\Big]. \end{aligned}$$

We also let $H_{n+1}(\bar{X}_{n+1}, \bar{Y}_{n+1}) := h(\bar{X}_{n+1}, \bar{Y}_{n+1})$. Note that we omit the dependence with respect to the sequence τ^{n+1} in the definition of the (random) maps $(H_k)_{0 \leqslant k \leqslant n+1}$. From the above definition and using **(ND)**, **(AR)**, it follows that

the map H_k belongs to $\mathcal{C}_p^1(\mathbb{R}^2)$ *a.s.* for any $0 \leqslant k \leqslant n+1$. Moreover, from the tower property of conditional expectation the following relation is satisfied for any $k \in \{0, \cdots, n\}$

$$H_k(\bar{X}_k, \bar{Y}_k) = \mathbb{E}_{k,n}[H_{k+1}(\bar{X}_{k+1}, \bar{Y}_{k+1})\theta_{k+1}]. \tag{4.4.8}$$

Now, iterating the transfer of derivative formula (4.4.3) in Lemma 4.4.1, for any $k \in \{1, \cdots, n\}$, we obtain[2]

$$\begin{aligned}
&\partial_{y_0} H_0(\bar{X}_0, \bar{Y}_0) \\
&= \partial_{y_0} \mathbb{E}_{0,n}[H_1(\bar{X}_1, \bar{Y}_1)\theta_1] \\
&= \mathbb{E}_{0,n}[\partial_{\bar{Y}_1} H_1(\bar{X}_1, \bar{Y}_1)\, \overrightarrow{\theta}_1^{e,Y}] + \mathbb{E}_{0,n}[\partial_{\bar{X}_1} H_1(\bar{X}_1, \bar{Y}_1)\, \overrightarrow{\theta}_1^{e,X}] + \mathbb{E}_{0,n}[H_1(\bar{X}_1, \bar{Y}_1)\, \overrightarrow{\theta}_1^{c}] \\
&= \cdots \\
&= \mathbb{E}_{0,n}[\mathcal{D}_k^{(2)} H_k(\bar{X}_k, \bar{Y}_k) \prod_{i=1}^{k} \overrightarrow{\theta}_i^{e,Y}] + \sum_{j=1}^{k} \mathbb{E}_{0,n}[H_j(\bar{X}_j, \bar{Y}_j)\, \overrightarrow{\theta}_j^{c} \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y}] \\
&\quad + \sum_{j=1}^{k} \mathbb{E}_{0,n}[\mathcal{D}_j^{(1)} H_j(\bar{X}_j, \bar{Y}_j)\, \overrightarrow{\theta}_j^{e,X} \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y}].
\end{aligned} \tag{4.4.9}$$

Hence, by the Lebesgue differentiation theorem, we deduce

$$\begin{aligned}
&\partial_{y_0} \mathbb{E}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1}) \prod_{i=1}^{n+1} \theta_i \Big| \tau^{n+1}\Big] \\
&= \partial_{y_0} \mathbb{E}\Big[H_0(\bar{X}_0, \bar{Y}_0) \Big| \tau^{n+1}\Big] \\
&= \mathbb{E}\Big[\partial_{y_0} H_0(\bar{X}_0, \bar{Y}_0) \Big| \tau^{n+1}\Big] \\
&= \mathbb{E}\Big[\mathcal{D}_k^{(2)} H_k(\bar{X}_k, \bar{Y}_k) \prod_{i=1}^{k} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] + \sum_{j=1}^{k} \mathbb{E}\Big[H_j(\bar{X}_j, \bar{Y}_j)\, \overrightarrow{\theta}_j^{c} \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] \\
&\quad + \sum_{j=1}^{k} \mathbb{E}\Big[\mathcal{D}_j^{(1)} H_j(\bar{X}_j, \bar{Y}_j)\, \overrightarrow{\theta}_j^{e,X} \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big].
\end{aligned} \tag{4.4.10}$$

To further simplify the first term appearing on the right-hand side of (4.4.10), we use the tower property of conditional expectation (w.r.t $\mathbb{E}_{k-1,n}[.]$) and the integration by parts formula (4.2.14). For any $k \in \{1, \cdots, n\}$, we obtain

$$\mathbb{E}\Big[\mathcal{D}_k^{(2)} H_k(\bar{X}_k, \bar{Y}_k)\, \overrightarrow{\theta}_k^{e,Y} \Big| \mathcal{G}_{k-1}, \tau^{n+1}\Big] = \mathbb{E}\Big[H_k(\bar{X}_k, \bar{Y}_k) \mathcal{I}_k^{(2)}(\overrightarrow{\theta}_k^{e,Y}) \Big| \mathcal{G}_{k-1}, \tau^{n+1}\Big]. \tag{4.4.11}$$

We also simplify the third term appearing on the right-hand side of (4.4.10), by using the transfer of derivatives formula (4.4.2) up to the time interval $[\zeta_{k-1}, \zeta_k]$. For any $j \in \{1, \cdots, k\}$, it holds

$$\mathbb{E}\Big[\mathcal{D}_j^{(1)} H_j(\bar{X}_j, \bar{Y}_j)\, \overrightarrow{\theta}_j^{e,X} \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] = \mathbb{E}\Big[\mathcal{D}_k^{(1)} H_k(\bar{X}_k, \bar{Y}_k) \prod_{i=j+1}^{k} \theta_i\, \overrightarrow{\theta}_j^{e,X} \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big],$$

[2] As before, we use the convention $\sum_{\varnothing} \cdots = 0$, $\prod_{\varnothing} \cdots = 1$.

so that, if $j \in \{1, \cdots, k-1\}$, taking conditional expectation (using again $\mathbb{E}_{k-1,n}[.]$) and then performing an IBP formula on the last time interval $[\zeta_{k-1}, \zeta_k]$ yield

$$\begin{aligned}
&\mathbb{E}\Big[\mathcal{D}_k^{(1)} H_k(\bar{X}_k, \bar{Y}_k) \prod_{i=j+1}^{k} \theta_i \overrightarrow{\theta}_j^{e,X} \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] \\
&= \mathbb{E}\Big[H_k(\bar{X}_k, \bar{Y}_k) \mathcal{I}_k^{(1)}(\theta_k) \prod_{i=j+1}^{k-1} \theta_i \overrightarrow{\theta}_j^{e,X} \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big], \qquad (4.4.12)
\end{aligned}$$

while if $j = k$, we obtain

$$\begin{aligned}
&\mathbb{E}\Big[\mathcal{D}_k^{(1)} H_k(\bar{X}_k, \bar{Y}_k) \prod_{i=j+1}^{k} \theta_i \overrightarrow{\theta}_j^{e,X} \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] \\
&= \mathbb{E}\Big[H_k(\bar{X}_k, \bar{Y}_k) \mathcal{I}_k^{(1)}(\overrightarrow{\theta}_k^{e,X}) \prod_{i=1}^{k-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big]. \qquad (4.4.13)
\end{aligned}$$

Coming back to (4.4.10), gathering (4.4.11), (4.4.12), (4.4.13) and using the definition of the maps $(H_k)_{0\leqslant k\leqslant n+1}$, we thus deduce

$$\begin{aligned}
&\partial_{y_0}\mathbb{E}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1}) \prod_{i=1}^{n+1} \theta_i \Big| \tau^{n+1}\Big] \\
&= \mathbb{E}\Big[H_k(\bar{X}_k, \bar{Y}_k) \mathcal{I}_k^{(2)}(\overrightarrow{\theta}_k^{e,Y}) \times \prod_{i=1}^{k-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] + \sum_{j=1}^{k} \mathbb{E}\Big[H_j(\bar{X}_j, \bar{Y}_j) \overrightarrow{\theta}_j^{c} \times \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] \\
&\quad + \sum_{j=1}^{k-1} \mathbb{E}\Big[H_k(\bar{X}_k, \bar{Y}_k) \mathcal{I}_k^{(1)}(\theta_k) \times \prod_{i=j+1}^{k-1} \theta_i \times \overrightarrow{\theta}_j^{e,X} \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] \\
&\quad + \mathbb{E}\Big[H_k(\bar{X}_k, \bar{Y}_k) \mathcal{I}_k^{(1)}(\overrightarrow{\theta}_k^{e,X}) \prod_{i=1}^{k-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] \\
&= \mathbb{E}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1}) \prod_{i=k+1}^{n+1} \theta_i \times \mathcal{I}_k^{(2)}(\overrightarrow{\theta}_k^{e,Y}) \times \prod_{i=1}^{k-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] \\
&\quad + \sum_{j=1}^{k} \mathbb{E}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1}) \prod_{i=j+1}^{n+1} \theta_i \times \overrightarrow{\theta}_j^{c} \times \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] \\
&\quad + \sum_{j=1}^{k-1} \mathbb{E}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1}) \prod_{i=k+1}^{n+1} \theta_i \times \mathcal{I}_k^{(1)}(\theta_k) \times \prod_{i=j+1}^{k-1} \theta_i \times \overrightarrow{\theta}_j^{e,X} \times \prod_{i=1}^{j-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big] \\
&\quad + \mathbb{E}\Big[h(\bar{X}_{n+1}, \bar{Y}_{n+1}) \prod_{i=k+1}^{n+1} \theta_i \times \mathcal{I}_k^{(1)}(\overrightarrow{\theta}_k^{e,X}) \prod_{i=1}^{k-1} \overrightarrow{\theta}_i^{e,Y} \Big| \tau^{n+1}\Big]. \qquad (4.4.14)
\end{aligned}$$

In the case $k = n + 1$, using the transfer of derivative formulae (4.4.4), (4.4.2) of Lemma 4.4.1 on the last time interval and then performing the IBP formula (4.2.14),

we obtain the representation

$$
\begin{aligned}
&\partial_{y_0}\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=1}^{n+1}\theta_i\Big|\tau^{n+1}\Big]\\
&=\mathbb{E}\Big[\mathcal{D}^{(2)}_{n+1}h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=1}^{n+1}\overrightarrow{\theta}^{e,Y}_i\Big|\tau^{n+1}\Big]+\sum_{j=1}^{n+1}\mathbb{E}\Big[H_j(\bar{X}_j,\bar{Y}_j)\overrightarrow{\theta}^{c}_j\prod_{i=1}^{j-1}\overrightarrow{\theta}^{e,Y}_i\Big|\tau^{n+1}\Big]\\
&\quad+\sum_{j=1}^{n+1}\mathbb{E}\Big[\mathcal{D}^{(1)}_j H_j(\bar{X}_j,\bar{Y}_j)\overrightarrow{\theta}^{e,X}_j\prod_{i=1}^{j-1}\overrightarrow{\theta}^{e,Y}_i\Big|\tau^{n+1}\Big]\\
&=\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\mathcal{I}^{(2)}_{n+1}(\overrightarrow{\theta}^{e,Y}_{n+1})\prod_{i=1}^{n}\overrightarrow{\theta}^{e,Y}_i\Big|\tau^{n+1}\Big]+\sum_{j=1}^{n+1}\mathbb{E}\Big[H_j(\bar{X}_j,\bar{Y}_j)\overrightarrow{\theta}^{c}_j\prod_{i=1}^{j-1}\overrightarrow{\theta}^{e,Y}_i\Big|\tau^{n+1}\Big]\\
&\quad+\sum_{j=1}^{n+1}\mathbb{E}\Big[\mathcal{D}^{(1)}_{n+1}h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=j+1}^{n+1}\theta_i\times\overrightarrow{\theta}^{e,X}_j\times\prod_{i=1}^{j-1}\overrightarrow{\theta}^{e,Y}_i\Big|\tau^{n+1}\Big]\\
&=\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\mathcal{I}^{(2)}_{n+1}(\overrightarrow{\theta}^{e,Y}_{n+1})\times\prod_{i=1}^{n}\overrightarrow{\theta}^{e,Y}_i\Big|\tau^{n+1}\Big]\\
&\quad+\sum_{j=1}^{n+1}\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=j+1}^{n+1}\theta_i\times\overrightarrow{\theta}^{c}_j\times\prod_{i=1}^{j-1}\overrightarrow{\theta}^{e,Y}_i\Big|\tau^{n+1}\Big]\\
&\quad+\sum_{j=1}^{n}\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\mathcal{I}^{(1)}_{n+1}(\theta_{n+1})\times\prod_{i=j+1}^{n}\theta_i\times\overrightarrow{\theta}^{e,X}_j\times\prod_{i=1}^{j-1}\overrightarrow{\theta}^{e,Y}_i\Big|\tau^{n+1}\Big]\\
&\quad+\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\mathcal{I}^{(1)}_{n+1}(\overrightarrow{\theta}^{e,X}_{n+1})\times\prod_{i=1}^{n}\overrightarrow{\theta}^{e,Y}_i\Big|\tau^{n+1}\Big]
\end{aligned}
\tag{4.4.15}
$$

where, for the last term appearing in the right-hand side of the above identities, we employed the transfer of derivative formula (4.4.2) up to the last time interval and then performed an IBP formula.

Now, the key point in order to establish the IBP formula (4.4.6) is to combine in a suitable way the identities (4.4.14) and (4.4.15). For each $k\in\{0,\cdots,n\}$, we multiply the above formulae by the length of the interval on which the local IBP formula is performed, namely we multiply by $\zeta_k-\zeta_{k-1}$ both sides of (4.4.14), $k=1,\cdots,n-1$, and we multiply by $T-\zeta_n$ both sides of (4.4.15). We then sum them over all k.

Recalling that $\sum_{k=1}^{n+1} \zeta_k - \zeta_{k-1} = T - \zeta_0 = T$, we deduce

$$
\begin{aligned}
&T\partial_{y_0}\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=1}^{n+1}\theta_i\Big|\tau^{n+1}\Big]\\
&=\sum_{k=1}^{n+1}(\zeta_k-\zeta_{k-1})\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=k+1}^{n+1}\theta_i\times\mathcal{I}_k^{(2)}(\overrightarrow{\theta}_k^{e,Y})\times\prod_{i=1}^{k-1}\overrightarrow{\theta}_i^{e,Y}\Big|\tau^{n+1}\Big]\\
&\quad+\sum_{k=1}^{n+1}(\zeta_k-\zeta_{k-1})\sum_{j=1}^{k}\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=j+1}^{n+1}\theta_i\times\overrightarrow{\theta}_j^{c}\times\prod_{i=1}^{j-1}\overrightarrow{\theta}_i^{e,Y}\Big|\tau^{n+1}\Big]\\
&\quad+\sum_{k=1}^{n+1}(\zeta_k-\zeta_{k-1})\Big(\sum_{j=1}^{k-1}\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=k+1}^{n+1}\theta_i\times\mathcal{I}_k^{(1)}(\theta_k)\times\prod_{i=j+1}^{k-1}\theta_i\times\overrightarrow{\theta}_j^{e,X}\\
&\quad\times\prod_{i=1}^{j-1}\overrightarrow{\theta}_i^{e,Y}\Big|\tau^{n+1}\Big]+\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=k+1}^{n+1}\theta_i\times\mathcal{I}_k^{(1)}(\overrightarrow{\theta}_k^{e,X})\times\prod_{i=1}^{n}\overrightarrow{\theta}_i^{e,Y}\Big|\tau^{n+1}\Big]\Big)\\
&=\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\sum_{k=1}^{n+1}(\zeta_k-\zeta_{k-1})\Big(\overrightarrow{\theta}^{\mathcal{I}_k^{(2),n+1}}+\sum_{j=1}^{k}\Big(\overrightarrow{\theta}^{C_j^{n+1}}+\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),n+1}}\Big)\Big)\Big|\tau^{n+1}\Big].
\end{aligned}
$$

We now provide a sharp upper-estimate for the above quantity. From Lemma 4.6.2 and Lemma 4.4.1, it follows that $f(\zeta_i-\zeta_{i-1})\theta_i$, $f(\zeta_i-\zeta_{i-1})\overrightarrow{\theta}_i^{e,Y}$, $f(\zeta_i-\zeta_{i-1})\overrightarrow{\theta}_i^{c}\in\mathbb{M}_{i-1,n}(\bar{X},\bar{Y},-1/2)$ and $f(\zeta_i-\zeta_{i-1})\overrightarrow{\theta}_i^{e,X}\in\mathbb{M}_{i-1,n}(\bar{X},\bar{Y},0)$ for any $i\in\{1,\cdots,n\}$. Moreover, from the very definition of the weights θ_i, $\overrightarrow{\theta}_i^{e,X}$ and $\overrightarrow{\theta}_i^{e,Y}$, after some simple but cumbersome computations that we omit (we also refer the reader to Appendix 4.6.4 which contains some expansion formulae), one has $f(\zeta_i-\zeta_{i-1})\mathcal{D}_i^{(1)}(\theta_i)$, $f(\zeta_i-\zeta_{i-1})\mathcal{D}_i^{(2)}(\overrightarrow{\theta}_i^{e,Y})\in\mathbb{M}_{i-1,n}(\bar{X},\bar{Y},-1)$ and $f(\zeta_i-\zeta_{i-1})\mathcal{D}_i^{(1)}(\overrightarrow{\theta}_i^{e,X})\in\mathbb{M}_{i-1,n}(\bar{X},\bar{Y},-1/2)$ so that from Lemma 4.2.3 we conclude $f(\zeta_i-\zeta_{i-1})(\zeta_i-\zeta_{i-1})\mathcal{I}_i^{(1)}(\theta_i)\in\mathbb{M}_{i-1,n}(\bar{X},\bar{Y},0)$, $f(\zeta_i-\zeta_{i-1})(\zeta_i-\zeta_{i-1})\mathcal{I}_i^{(2)}(\overrightarrow{\theta}_i^{e,Y})\in\mathbb{M}_{i-1,n}(\bar{X},\bar{Y},0)$ and $f(\zeta_i-\zeta_{i-1})(\zeta_i-\zeta_{i-1})\mathcal{I}_i^{(1)}(\overrightarrow{\theta}_i^{e,X})\in\mathbb{M}_{i-1,n}(\bar{X},\bar{Y},1/2)$. Hence, from the boundedness of h, the tower property of conditional expectation and (4.2.22), it holds

$$
\begin{aligned}
&\Big|(\zeta_k-\zeta_{k-1})\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=k+1}^{n+1}\theta_i\times\mathcal{I}_k^{(2)}(\overrightarrow{\theta}_k^{e,Y})\times\prod_{i=1}^{k-1}\overrightarrow{\theta}_i^{e,Y}\Big|\tau^{n+1}\Big]\Big|\\
&\leqslant C^{n+1}(1-F(T-\zeta_n))^{-1}\prod_{i=k+1}^{n}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-\frac{1}{2}}(f(\zeta_k-\zeta_{k-1}))^{-1}\\
&\quad\prod_{i=1}^{k-1}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-\frac{1}{2}}
\end{aligned}
$$

so that using the identity (4.2.6)

$$\begin{aligned}
&\sum_{n\geqslant 0}\mathbb{E}\Big[\sum_{k=1}^{n+1}\Big|(\zeta_k-\zeta_{k-1})\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=k+1}^{n+1}\theta_i\times\mathcal{I}_k^{(2)}(\overrightarrow{\theta}_k^{e,Y})\times\prod_{i=1}^{k-1}\overrightarrow{\theta}_i^{e,Y}\,\Big|\,\tau^{n+1}\Big]\Big|\mathbf{1}_{\{N_T=n\}}\Big]\\
&\quad\leqslant\sum_{n\geqslant 0}C^{n+1}\sum_{k=1}^{n+1}\mathbb{E}\Big[(1-F(T-\zeta_n))^{-1}(f(\zeta_k-\zeta_{k-1}))^{-1}\\
&\qquad\times\prod_{i=1,i\neq k}^{n}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-\frac{1}{2}}\mathbf{1}_{\{N_T=n\}}\Big]\\
&\quad=\sum_{n\geqslant 0}C^{n+1}\sum_{k=1}^{n+1}\int_{\Delta_n(T)}\prod_{i=1,i\neq k}^{n}(s_i-s_{i-1})^{-1/2}\,d\mathbf{s}_n\\
&\quad\leqslant\sum_{n\geqslant 0}(n+1)C^{n+1}T^{(n+1)/2}\frac{\Gamma^n(1/2)}{\Gamma(1+n/2)}<\infty.
\end{aligned}$$

From similar arguments that we omit, it follows

$$\begin{aligned}
&\Big|(\zeta_k-\zeta_{k-1})\sum_{j=1}^{k}\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\Big(\overrightarrow{\theta}^{C_j^{n+1}}+\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),n+1}}\Big)\,\Big|\,\tau^{n+1}\Big]\Big|\\
&\quad\leqslant C^{n+1}(\zeta_k-\zeta_{k-1})\sum_{j=1}^{k}(1-F(T-\zeta_n))^{-1}\\
&\qquad\times\prod_{i=1}^{n}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-1/2}[1+\mathbf{1}_{\{i=k\}}(\zeta_i-\zeta_{i-1})^{-1/2}].
\end{aligned}$$

so that using again the identity (4.2.6)

$$\begin{aligned}
&\sum_{n\geqslant 0}\mathbb{E}\Big[\sum_{k=1}^{n+1}\Big|(\zeta_k-\zeta_{k-1})\sum_{j=1}^{k}\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\Big(\overrightarrow{\theta}^{C_j^{n+1}}+\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),n+1}}\Big)\,\Big|\,\tau^{n+1}\Big]\Big|\mathbf{1}_{\{N_T=n\}}\Big]\\
&\quad\leqslant\sum_{n\geqslant 0}C^{n+1}\sum_{k=1}^{n+1}\mathbb{E}\big[(\zeta_k-\zeta_{k-1})\sum_{j=1}^{k}(1-F(T-\zeta_n))^{-1}\\
&\qquad\times\prod_{i=1}^{n}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-1/2}[1+\mathbf{1}_{\{i=k\}}(\zeta_i-\zeta_{i-1})^{-1/2}]\mathbf{1}_{\{N_T=n\}}\big]\\
&\quad\leqslant\sum_{n\geqslant 0}C^{n+1}(n+1)(n+2)T^{(n+1)/2}\frac{\Gamma^n(1/2)}{\Gamma(1+n/2)}<\infty.
\end{aligned}$$

The preceding estimates combined with (4.4.7) and the Lebesgue dominated convergence theorem allows to conclude that $y_0\mapsto\mathbb{E}[h(X_T,Y_T)]$ is continuously

differentiable with

$$
\begin{aligned}
&T\partial_{y_0}\mathbb{E}[h(X_T,Y_T)]\\
&=T\partial_{y_0}\mathbb{E}\Big[h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})\prod_{i=1}^{N_T+1}\theta_i\Big]\\
&=\sum_{n\geqslant 0}\mathbb{E}\Big[T\partial_{y_0}\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=1}^{n+1}\theta_i\Big|\tau^{n+1}\Big]\mathbf{1}_{\{N_T=n\}}\Big]\\
&=\sum_{n\geqslant 0}\mathbb{E}\Big[\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\sum_{k=1}^{n+1}(\zeta_k-\zeta_{k-1})\\
&\quad\times\Big(\overrightarrow{\theta}^{\mathcal{I}_k^{(2),n+1}}+\sum_{j=1}^{k}\Big(\overrightarrow{\theta}^{C_j^{n+1}}+\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),n+1}}\Big)\Big)\Big|\tau^{n+1}\Big]\mathbf{1}_{\{N_T=n\}}\Big]\\
&=\mathbb{E}\Big[h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})\sum_{k=1}^{N_T+1}(\zeta_k-\zeta_{k-1})\Big(\overrightarrow{\theta}^{\mathcal{I}_k^{(2),N_T+1}}+\sum_{j=1}^{k}\Big(\overrightarrow{\theta}^{C_j^{n+1}}+\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),N_T+1}}\Big)\Big)\Big]
\end{aligned}
$$

where we used Fubini's theorem for the last equality. This completes the proof of the IBP formula (4.4.6) for $h\in\mathcal{C}_b^1(\mathbb{R}^2)$.

Step 2: Extension to $h\in\mathcal{B}_\gamma(\mathbb{R}^2)$ for some positive γ.

We now extend the two IBP formulae that we have established in the previous step to the case of a test function $h\in\mathcal{B}_\gamma(\mathbb{R}^2)$ for some sufficiently small $\gamma>0$. Let us note that under assumption (15) **(AR)** and **(ND)**, from Kusuoka and Stroock [71], Corollary (3.25) and the upper-estimate (3.27) therein, the process $(X_t,Y_t)_{t\geqslant 0}$ admits a smooth transition density $(t,x_0,y_0,x,y)\mapsto p(t,x_0,y_0,x,y)\in\mathcal{C}^\infty((0,\infty)\times\mathbb{R}^2\times\mathbb{R}^2)$ and for any $h\in\mathcal{C}_b^1(\mathbb{R}^2)$, it holds

$$
\partial_{s_0}^\alpha\partial_{y_0}^\beta\mathbb{E}[h(X_T,Y_T)]=\int_{\mathbb{R}^2}h(x,y)\,\partial_{s_0}^\alpha\partial_{y_0}^\beta p(T,x_0,y_0,x,y)\,dxdy \tag{4.4.16}
$$

for any $T>0$ and any integers α and β.

We then proceed as in step 2 of the proof of Theorem 4.3.1. Namely, we prove that

$$
\begin{aligned}
&T\partial_{y_0}\mathbb{E}[h(X_T,Y_T)]\\
&=\mathbb{E}\Big[h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})\sum_{k=1}^{N_T+1}(\zeta_k-\zeta_{k-1})\Big(\overrightarrow{\theta}^{\mathcal{I}_k^{(2),N_T+1}}+\sum_{j=1}^{k}\Big(\overrightarrow{\theta}^{C_j^{n+1}}+\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),N_T+1}}\Big)\Big)\Big]\\
&=\int_{\mathbb{R}^2}h(x,y)\,\mathbb{E}\Big[\bar{p}(T-\zeta_{N_T},\bar{X}_{N_T},\bar{Y}_{N_T},x,y)\sum_{k=1}^{N_T+1}(\zeta_k-\zeta_{k-1})\Big(\overrightarrow{\theta}^{\mathcal{I}_k^{(2),N_T+1}}+\sum_{j=1}^{k}\\
&\quad\times\Big(\overrightarrow{\theta}^{C_j^{n+1}}+\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),N_T+1}}\Big)\Big)\Big]\,dxdy
\end{aligned}
\tag{4.4.17}
$$

for any $h\in\mathcal{C}_b^1(\mathbb{R}^2)$.

Indeed, since $f(\zeta_i-\zeta_{i-1})\theta_i, f(\zeta_i-\zeta_{i-1})\overrightarrow{\theta}_i^{e,Y} \in \mathbb{M}_{i-1,n}(\bar{X},\bar{Y},-1/2)$ and $f(\zeta_k-\zeta_{k-1})(\zeta_k-\zeta_{k-1})\mathcal{I}_k^{(2)}(\overrightarrow{\theta}_k^{e,Y}) \in \mathbb{M}_{k-1,n}(\bar{X},\bar{Y},0)$, for some $c := c(T,b_Y,\kappa) > 4\kappa$, it holds

$$
\begin{aligned}
&\mathbb{E}\Big[\bar{p}(T-\zeta_n,\bar{X}_n,\bar{Y}_n,x,y)\sum_{k=1}^{n+1}(\zeta_k-\zeta_{k-1})\Big|\overrightarrow{\theta}^{\mathcal{I}_k^{(2),n+1}}\Big|\Big|\tau^{n+1}\Big]\\
&\leqslant C^{n+1}\int_{(\mathbb{R}^2)^n}\bar{q}_{4\kappa}(T-\zeta_n,x_n,y_n,x,y)\sum_{k=1}^{n+1}(1-F(T-\zeta_n))^{-1}\\
&\quad\times\prod_{i=k+1}^{n}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-1/2}(f(\zeta_k-\zeta_{k-1}))^{-1}\\
&\quad\times\prod_{i=1}^{k-1}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-1/2}\prod_{i=1}^{n}\bar{q}_{4\kappa}(\zeta_i-\zeta_{i-1},x_{i-1},y_{i-1},x_i,y_i)\,d\mathbf{x}_n d\mathbf{y}_n\\
&\leqslant C^{n+1}\bar{q}_c(T,x_0,y_0,x,y)\sum_{k=1}^{n+1}(1-F(T-\zeta_n))^{-1}\prod_{i=1}^{n}(f(\zeta_i-\zeta_{i-1}))^{-1}\prod_{i=1,i\neq k}^{n}(\zeta_i-\zeta_{i-1})^{-1/2}
\end{aligned}
\tag{4.4.18}
$$

where, for the first inequality we used the upper-estimate (4.2.4) and for the last inequality we used Lemma 4.6.3. From similar arguments, one gets

$$
\begin{aligned}
&\mathbb{E}\Big[\bar{p}(T-\zeta_n,\bar{X}_n,\bar{Y}_n,x,y)\sum_{k=1}^{n+1}(\zeta_k-\zeta_{k-1})\sum_{j=1}^{k}\Big(\Big|\overrightarrow{\theta}^{C_j^{n+1}}\Big|+\Big|\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),N_T+1}}\Big|\Big)\Big|\tau^{n+1}\Big]\\
&\leqslant C^{n+1}\bar{q}_c(T,x_0,y_0,x,y)\sum_{k=1}^{n+1}(\zeta_k-\zeta_{k-1})\sum_{j=1}^{k}(1-F(T-\zeta_n))^{-1}\\
&\quad\times\prod_{i=1}^{n}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-1/2}[1+\mathbf{1}_{\{i=k\}}(\zeta_i-\zeta_{i-1})^{-1/2}].
\end{aligned}
\tag{4.4.19}
$$

Now, from the upper-bounds (4.4.18) and (4.4.19) as well as the identity (4.2.6),

we conclude

$$\begin{aligned}
&\sum_{n\geqslant 0}\mathbb{E}\Big[\mathbb{E}\Big[\bar{p}(T-\zeta_n,\bar{X}_n,\bar{Y}_n,x,y)\sum_{k=1}^{n+1}(\zeta_k-\zeta_{k-1})\\
&\times\left(\Big|\overrightarrow{\theta}^{\mathcal{I}_k^{(2),n+1}}\Big|+\sum_{j=1}^{k}\left(\Big|\overrightarrow{\theta}^{C_j^{n+1}}\Big|+\Big|\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),n+1}}\Big|\right)\right)\Big|\tau^{n+1}\Big]\mathbf{1}_{\{N_T=n\}}\Big]\\
&\leqslant \bar{q}_c(T,x_0,y_0,x,y)\sum_{n\geqslant 0}C^{n+1}\mathbb{E}\Big[\Big(\sum_{k=1}^{n+1}(1-F(T-\zeta_n))^{-1}\prod_{i=1}^{n}(f(\zeta_i-\zeta_{i-1}))^{-1}\\
&\times\prod_{i=1,i\neq k}^{n}(\zeta_i-\zeta_{i-1})^{-1/2}+\sum_{k=1}^{n+1}(\zeta_k-\zeta_{k-1})\sum_{j=1}^{k}(1-F(T-\zeta_n))^{-1}\\
&\times\prod_{i=1}^{n}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-1/2}[1+\mathbf{1}_{\{i=k\}}(\zeta_i-\zeta_{i-1})^{-1/2}]\Big)\mathbf{1}_{\{N_T=n\}}\Big]\\
&\leqslant \bar{q}_c(T,x_0,y_0,x,y)\sum_{n\geqslant 0}C^{n+1}[(n+1)+(n+1)(n+2)/2]T^{(n+1)/2}\frac{\Gamma^n(1/2)}{\Gamma(1+n/2)}\\
&= CT^{1/2}\bar{q}_c(T,x_0,y_0,x,y). \qquad (4.4.20)
\end{aligned}$$

From the preceding inequality and Fubini's theorem, we thus get

$$\begin{aligned}
&\Big|\mathbb{E}\Big[\bar{p}(T-\zeta_{N_T},\bar{X}_{N_T},\bar{Y}_{N_T},x,y)\sum_{k=1}^{N_T+1}(\zeta_k-\zeta_{k-1})\\
&\qquad\left(\overrightarrow{\theta}^{\mathcal{I}_k^{(2),N_T+1}}+\sum_{j=1}^{k}\left(\overrightarrow{\theta}^{C_j^{N_T+1}}+\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),N_T+1}}\right)\right)\Big]\Big|\\
&\leqslant CT^{1/2}\bar{q}_c(T,x_0,y_0,x,y) \qquad (4.4.21)
\end{aligned}$$

for some positive constant $C := C(T)$ such that $T \mapsto C(T)$ is non-decreasing. Applying again Fubini's theorem allows to complete the proof of (4.4.17). Hence, combining (4.4.16) with (4.4.17)

$$\begin{aligned}
&T\partial_{y_0}\mathbb{E}[h(X_T,Y_T)]\\
&=\int_{\mathbb{R}^2}h(x,y)\,T\partial_{y_0}p(T,x_0,y_0,x,y)\,dxdy\\
&=\int_{\mathbb{R}^2}h(x,y)\,\mathbb{E}\Big[\bar{p}(T-\zeta_{N_T},\bar{X}_{N_T},\bar{Y}_{N_T},x,y)\sum_{k=1}^{N_T+1}(\zeta_k-\zeta_{k-1})\\
&\quad\times\left(\overrightarrow{\theta}^{\mathcal{I}_k^{(2),N_T+1}}+\sum_{j=1}^{k}\left(\overrightarrow{\theta}^{C_j^{N_T+1}}+\overrightarrow{\theta}_j^{\mathcal{I}_k^{(1),N_T+1}}\right)\right)\Big]\,dxdy
\end{aligned}$$

for any $h\in\mathcal{C}_b^1(\mathbb{R}^2)$. A monotone class argument allows to conclude that the preceding identity is still valid for any bounded and measurable map h defined over $\mathbb{R}^2$ and a standard approximation argument allows to extend it to $h\in\mathcal{B}_\gamma(\mathbb{R}^2)$ for any $0\leqslant\gamma<(2cT)^{-1}$, c being the positive constant appearing in the right-hand side

of (4.4.21). We eventually conclude from the preceding identity, (4.4.21) combined with Fubini's theorem that

$$T\partial_{y_0}\mathbb{E}[h(X_T,Y_T)]$$
$$=\mathbb{E}\Big[h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})\sum_{k=1}^{N_T+1}(\zeta_k-\zeta_{k-1})\Big(\overrightarrow{\theta}^{\mathcal{I}_k^{(2),N_T+1}}+\sum_{j=1}^{k}\Big(\overrightarrow{\theta}^{C_j^{N_T+1}}+\overrightarrow{\theta}^{\mathcal{I}_k^{(1),N_T+1}}_{j}\Big)\Big)\Big]$$

for any $h\in\mathcal{B}_\gamma(\mathbb{R}^2)$ such that $0\leqslant\gamma<(2cT)^{-1}$.

Step 3: $L^p(\mathbb{P})$-moments for a renewal process with Beta jump times.

From the above formula, the proof of the $L^p(\mathbb{P})$-moment estimate when N is a renewal process with Beta jump times follows by similar arguments as those employed at step 3 of the proof of Theorem 4.3.1. We omit the remaining technical details.

4.5 Numerical Results

In this section, as a proof of concept, we provide some simple numerical results for the unbiased Monte Carlo algorithm that stems from the probabilistic representation formula established in Theorem 4.3.1 and the Bismut-Elworthy-Li formulae of Theorem 4.4.1 for the couple (S_T,Y_T) that allows to compute the Delta and the Vega related to the option price of the vanilla option with payoff $h(S_T)$. As already mentioned in the introduction, we believe that one needs to study numerical issues and to compare our algorithm with other existing methods to compute Greeks in more details. However, this is beyond the scope of the current thesis and is left to future research.

We here consider the unique strong solution associated to the SDE (4.1.1) for three different models corresponding to three different diffusion coefficient function σ_S and two different options, namely Call and digital Call options with maturity T and strike K, with payoff functions $h(x,y)=(\exp(x)-K)_+$ and $h(x,y)=\mathbf{1}_{\{\exp(x)\geqslant K\}}$ respectively. For these three models, the drift function of the volatility process is defined by $b_Y(x)=\lambda_Y(\mu-x)$ and we fix the parameters as follows: $T=0.5$, $r=0.03$, $K=1.5$, $x_0=\ln(s_0)=0.4$, $Y_0=0.2$, $\sigma_Y(.)\equiv\sigma_Y=0.2$, $\lambda_Y=0.5$, $\mu=0.3$ and $\rho=0.6$. We also consider two type of renewal process N: a Poisson process with intensity parameter $\lambda=0.5$ and a renewal process with $Beta(1-\alpha,1)$ jump times with parameters $\alpha=0.5$ and $\bar{\tau}=2$. A crude Monte Carlo estimator gives that $\mathbb{E}[N_T]=1.25$ for Exponential sampling (which is inline with the theoretical value $1+\lambda T$) and $\mathbb{E}[N_T]=1.79$ for Beta sampling.

The total time for the computation of the price, Delta and Vega are about 8 seconds for the Monte Carlo method with Euler scheme and about 10 seconds for the unbiased Monte Carlo method with Exponential and Beta sampling. Generally speaking, we observe that the variance of the unbiased Monte Carlo estimators is larger than the variance of the Monte Carlo estimator with Euler-Maruyama discretization scheme. This should not come as a big surprise since this fact is reminiscent of unbiased Monte Carlo methods. However, the Monte Carlo method with Euler scheme is also affected by its inherent bias.

4.5.1 Black-Scholes Model

We first consider the simple (toy) example corresponding to the Black-Scholes dynamics

$$dS_t = rS_t\,dt + \sigma_S S_t\,dW_t, \quad dY_t = b_Y(Y_t)\,dt + \sigma_Y(Y_t)dB_t, \quad d\langle B, W\rangle_t = \rho dt,\ \rho \in (-1,1).$$

with constant diffusion coefficient function $\sigma_S(.) \equiv \sigma_S > 0$. The law of (S_T, Y_T) can be computed explicitly so that analytical formulas are available for the price, Delta and Vega. Note that the discount factor e^{-rT} has been added in our probabilistic representation formula for comparison purposes. In this example, we importantly remark that the dynamics of the Euler scheme writes

$$\begin{cases} \bar{X}_{i+1} &= \bar{X}_i + \left(r - \frac{1}{2}a_{S,i}\right)(\zeta_{i+1} - \zeta_i) + \sigma_{S,i}\sqrt{\zeta_{i+1} - \zeta_i}\,Z^1_{i+1}, \\ \bar{Y}_{i+1} &= m_i + \sigma_{Y,i}\sqrt{\zeta_{i+1} - \zeta_i}\left(\rho_i Z^1_{i+1} + \sqrt{1-\rho_i^2}\,Z^2_{i+1}\right), \end{cases} \tag{4.5.1}$$

with $m_i = m_{\zeta_{i+1}-\zeta_i}(\bar{Y}_i) = \mu + (\bar{Y}_i - \mu)e^{-\lambda(\zeta_{i+1}-\zeta_i)}$. Also, the weights $(\theta_i)_{1\leqslant i\leqslant N_T+1}$ in the probabilistic representation (4.3.2) of Theorem 4.3.1 greatly simplifies, namely

$$\theta_i = (f(\zeta_i - \zeta_{i-1}))^{-1}\mathcal{I}_i^{(2)}(b_Y^i),\ 1 \leqslant i \leqslant N_T, \quad \text{and} \quad \theta_{N_T+1} = (1 - F(T - \zeta_{N_T}))^{-1}.$$

We perform $M_1 = 2.56 \times 10^7$ for the unbiased Monte Carlo method with Exponential sampling and $M_1 = 1.79 \times 10^7$ in the case of Beta sampling to approximate the price as well as the two Greeks so that the (average) computational cost (up to a constant multiplicative factor) is given by $\mathbb{E}[N_T] \times M_1 = 3.2 \times 10^7$ in both cases. We compare them with the corresponding values obtained using the standard Monte Carlo method combined with an Euler-Maruyama approximation scheme for the dynamics (4.1.1) with $M_2 = 160000$ Monte Carlo simulations paths and mesh size $\delta = T/n$ where $n = 200$. Its computational complexity (up to a constant multiplicative factor) is given by $n \times M_2 = 3.2 \times 10^7$. Hence, both Monte Carlo estimators have comparable computational complexity though their computational time are slightly different in practical implementation. The Delta and Vega are obtained using the Monte Carlo finite difference approach combined with the Euler-Maruyama discretization scheme, that is, denoting by $E^n_{M_2}(s_0, y_0)$ the Monte Carlo estimator associated to the Euler-Maruyama scheme, we compute $(E^n_{M_2}(s_0 + \varepsilon, y_0) - E^n_{M_2}(s_0, y_0))/\varepsilon$ and $(E^n_{M_2}(s_0, y_0 + \varepsilon) - E^n_{M_2}(s_0, y_0))/\varepsilon$ respectively with $\varepsilon = 10^{-2}$. The numerical results for the three different quantities are summarized in Table 4.1, Table 4.2, Table 4.3 respectively. The first column provides the value of the parameter σ_S. The second column stands for the value of the price, Delta or Vega obtained by the corresponding Black-Scholes formula. The third, fourth and fifth columns correspond to the value obtained by the Monte Carlo estimator using Euler-Maruyama discretization scheme together with its half-width 95% confidence interval and its empirical variance. The sixth, seventh and eighth (resp. the ninth, tenth and eleventh) columns provide the estimated value with its halfwidth 95% confidence interval and empirical variance by our method in the case of Exponential sampling (resp. Beta sampling). Note that though the variance of the Monte Carlo estimator in the case of Exponential sampling may explode, we compute it

for sake of completeness. Indeed, in our numerical experiences, we observed that the variance of the Monte Carlo estimator in the Exponential sampling case slightly increases with respect to M_1. Nevertheless, we observe a good behaviour of the unbiased estimators for all three quantities and for all the values of the parameter σ_S.

σ_S	B-S	Euler Scheme			Exponential sampling			Beta sampling		
	formula	Price	Half-width	Variance	Price	Half-width	Variance	Price	Half-width	Variance
0.25	0.111804	0.111853	0.000860286	0.0308244	0.112196	0.000124112	0.102648	0.112199	0.000154064	0.110598
0.3	0.132621	0.132808	0.0010515	0.0460493	0.133193	0.000152038	0.15404	0.133036	0.000187336	0.163524
0.4	0.174152	0.173559	0.00144315	0.0867423	0.174754	0.000208983	0.291037	0.174711	0.000257441	0.308813
0.6	0.256572	0.255388	0.00235625	0.231233	0.257287	0.000334903	0.747423	0.256978	0.0004127	0.793617

Table 4.1 – Comparison between the unbiased Monte Carlo estimation and the Monte Carlo Euler-Maruyama scheme for the price of a Call option in the Black-Scholes model for different values of σ_S.

σ_S	B-S	Euler Scheme			Exponential sampling			Beta sampling		
	formula	Delta	Half-width	Variance	Delta	Half-width	Variance	Delta	Half-width	Variance
0.25	0.556589	0.55675	0.00280539	0.327789	0.554992	0.000895101	5.33915	0.555192	0.00114054	6.0612
0.3	0.560018	0.560534	0.00290622	0.351775	0.558285	0.000923515	5.6835	0.557974	0.00116621	6.33719
0.4	0.569512	0.570228	0.00311011	0.402864	0.567568	0.000978965	6.38649	0.567091	0.00123	7.04938
0.6	0.592743	0.590041	0.00358714	0.535925	0.589	0.0010899	7.91588	0.587681	0.00137469	8.80548

Table 4.2 – Comparison between the unbiased Monte Carlo estimation and the Monte Carlo Euler-Maruyama scheme for the Delta of a Call option in the Black-Scholes model for different values of σ_S.

σ_S	B-S	Exponential sampling			Beta sampling		
	formula	Vega	Half-width	Variance	Vega	Half-width	Variance
0.25	0	0.000690222	0.00115103	8.82877	-0.000559242	0.00128448	7.68766
0.3	0	0.00182175	0.00137953	12.6821	0.000500579	0.00156401	11.3978
0.4	0	-0.00163321	0.00189888	24.0283	-0.000817515	0.00215655	21.6701
0.6	0	-0.000830748	0.00300346	60.1136	-0.001055	0.00340386	53.9862

Table 4.3 – Comparison between the unbiased Monte Carlo estimation for the Vega of a Call option in the Black-Scholes model for different values of σ_S.

4.5.2 A Stein-Stein type model

In this second example, we consider a Stein-Stein type model where the diffusion coefficient function for the spot price is an affine function, namely $\sigma_S(x) = \sigma_1 x + \sigma_2$ where σ_1 and σ_2 are two positive constants. Note carefully that σ_S is not uniformly elliptic and bounded so that **(AR)** and **(ND)** are not satisfied. However, we heuristically choose σ_1 and σ_2 so that $\sigma_S(Y_t)$ is bounded and strictly positive with high probability. Also, analytical expressions for the coefficients are available, namely

$$
\begin{aligned}
a_{S,i} &= \int_0^{\zeta_{i+1}-\zeta_i} \Big[\sigma_1\big(\mu + (\bar{Y}_i - \mu)e^{-\lambda_Y s}\big) + \sigma_2\Big]^2 ds, \\
&= (\sigma_1\mu + \sigma_2)^2(\zeta_{i+1} - \zeta_i) + \sigma_1^2(\bar{Y}_i - \mu)^2 \frac{1 - e^{-2\lambda_Y(\zeta_{i+1}-\zeta_i)}}{2\lambda_Y} \\
&\quad + 2\sigma_1(\sigma_1\mu + \sigma_2)(\bar{Y}_i - \mu)\frac{1 - e^{-\lambda_Y(\zeta_{i+1}-\zeta_i)}}{\lambda_Y}, \\
a'_{S,i} &= \sigma_1^2(\bar{Y}_i - \mu)\frac{1 - e^{-2\lambda_Y(\zeta_{i+1}-\zeta_i)}}{\lambda_Y} + 2\sigma_1(\sigma_1\mu + \sigma_2)\frac{1 - e^{-\lambda_Y(\zeta_{i+1}-\zeta_i)}}{\lambda_Y}, \\
\rho_i &= \rho\frac{\int_0^{\zeta_{i+1}-\zeta_i}\Big[\alpha\big(\mu + (\bar{Y}_i - \mu)e^{-\lambda s}\big) + C\Big] ds}{\sigma_{S,i}\sqrt{\zeta_{i+1} - \zeta_i}} \\
&= \rho\frac{\alpha(\bar{Y}_i - \mu)(1 - e^{-\lambda(\zeta_{i+1}-\zeta_i)})/\lambda + (\sigma_1\mu + \sigma_2)(\zeta_{i+1} - \zeta_i)}{\sigma_{S,i}\sqrt{\zeta_{i+1} - \zeta_i}}, \\
\rho'_i &= \rho\frac{\sigma_{S,i}\big(\sigma_1(1 - e^{-\lambda_Y(\zeta_{i+1}-\zeta_i)})/\lambda_Y\big)}{a_{S,i}\sqrt{\zeta_{i+1} - \zeta_i}} \\
&\quad - \rho\frac{\sigma'_{S,i}\big(\sigma_1(\bar{Y}_i - \mu)(1 - e^{-\lambda_Y(\zeta_{i+1}-\zeta_i)})/\lambda_Y + (\sigma_1\mu + \sigma_2)(\zeta_{i+1} - \zeta_i)\big)}{a_{S,i}\sqrt{\zeta_{i+1} - \zeta_i}}
\end{aligned}
$$

The parameters for the unbiased Monte Carlo method and the Monte Carlo method combined with an Euler-Maruyama approximation scheme are chosen as in the first example. The numerical results related to the price, Delta and Vega are provided in Table 4.4, Table 4.5, Table 4.6 respectively for the Call option and in Table 4.7, Table 4.8, Table 4.9 for the digital Call option. In spite of the fact that the main assumptions are not satisfied, we again observe a good performance of the unbiased estimators for all three quantities and for all the values of the parameters σ_1, σ_2, except for the computation of the Vega of a Call option for large values of σ_1 and σ_2.

4.5.3 A model with a periodic diffusion coefficient function

In our last example, the volatility of spot price takes the following form $\sigma_S(x) = \sigma_1\cos(x) + \sigma_2$ where σ_1 and σ_2 are two positive constants such that $\sigma_2 - \sigma_1 > 0$

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Price	Half-width	Variance	Price	Half-width	Variance	Price	Half-width	Variance
0.1	0.15	0.0788438	0.000591655	0.0145796	0.0785159	0.000184533	0.226921	0.0787702	0.000127826	0.076134
0.2	0.25	0.129	0.0010556	0.0464096	0.129024	0.000391611	1.02197	0.128967	0.000238248	0.264483
0.3	0.4	0.200983	0.00179584	0.13432	0.200121	0.000570515	2.16901	0.200039	0.000388453	0.703103
0.4	0.5	0.250972	0.00240492	0.240884	0.249897	0.000762565	3.87509	0.249507	0.000539937	1.3584

Table 4.4 – Comparison between the unbiased Monte Carlo estimation for the price of a Call option in the Stein-Stein type model for different values of the parameters σ_1 and σ_2.

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Delta	Half-width	Variance	Delta	Half-width	Variance	Delta	Half-width	Variance
0.1	0.15	0.547988	0.00265342	0.293238	0.538724	0.00165166	18.179	0.539677	0.00137889	8.85936
0.2	0.25	0.54942	0.00289865	0.349943	0.539137	0.00211916	29.9265	0.538131	0.00152902	10.8935
0.3	0.4	0.566344	0.00328048	0.448211	0.556553	0.00254168	43.0495	0.556605	0.00162791	12.3481
0.4	0.5	0.580157	0.00359471	0.53819	0.567956	0.0026141	45.5377	0.567737	0.00184814	15.9151

Table 4.5 – Comparison between the unbiased Monte Carlo estimation for the Delta of a Call option in the Stein-Stein type model for different values of the parameters σ_1 and σ_2.

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Vega	Half-width	Variance	Vega	Half-width	Variance	Vega	Half-width	Variance
0.1	0.15	0.0369128	0.000309236	0.0039828	0.0325922	0.00150912	15.1766	0.0349527	0.00104028	5.04243
0.2	0.25	0.0733673	0.000689904	0.0198237	0.0671736	0.00333565	74.1462	0.0680856	0.00190913	16.9829
0.3	0.4	0.109991	0.00121245	0.0612259	0.0990781	0.004919	161.243	0.0954942	0.00311339	45.1655
0.4	0.5	0.145413	0.0018251	0.138734	0.129808	0.00689955	317.226	0.122975	0.00443729	91.7436

Table 4.6 – Comparison between the unbiased Monte Carlo estimation for the Vega of a Call option in the Stein-Stein type model for different values of the parameters σ_1 and σ_2.

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Price	Half-width	Variance	Price	Half-width	Variance	Price	Half-width	Variance
0	0.3	0.468101	0.00241055	0.242013	0.468695	0.000363387	0.879968	0.46889	0.000466078	1.01218
0.1	0.15	0.490844	0.00241351	0.242608	0.48959	0.000706916	3.33015	0.489924	0.000534225	1.32981
0.2	0.25	0.458089	0.00240761	0.241423	0.458472	0.000780893	4.0636	0.458395	0.000536405	1.34069
0.3	0.4	0.430371	0.00239421	0.238744	0.428881	0.000840943	4.71261	0.429222	0.000513132	1.22687
0.4	0.5	0.410102	0.00237947	0.235813	0.409966	0.000737924	3.6287	0.409407	0.000510215	1.21296

Table 4.7 – Comparison between the unbiased Monte Carlo estimation for the price of a digital Call option in the Stein-Stein type model for different values of the parameters σ_1 and σ_2.

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Delta	Half-width	Variance	Delta	Half-width	Variance	Delta	Half-width	Variance
0	0.3	1.23092	0.0306499	39.126	1.2445	0.00157472	16.5247	1.24456	0.00199696	18.5815
0.1	0.15	2.20347	0.0403758	67.8968	2.17998	0.0049492	163.229	2.18349	0.00400828	74.8611
0.2	0.25	1.27673	0.0311925	40.5237	1.27004	0.00456344	138.776	1.27093	0.00239161	26.6516
0.3	0.4	0.79344	0.0247765	25.5675	0.793619	0.0023143	35.6918	0.792876	0.00144989	9.79521
0.4	0.5	0.617625	0.0219193	20.0107	0.623268	0.00168461	18.9116	0.622453	0.00115744	6.24217

Table 4.8 – Comparison between the unbiased Monte Carlo estimation for the Delta of a digital Call option in the Stein-Stein type model for different values of the parameters σ_1 and σ_2.

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Vega	Half-width	Variance	Vega	Half-width	Variance	Vega	Half-width	Variance
0	0.3	0	0	0	-0.000131092	0.00343628	78.6873	0.000348826	0.00401465	75.0995
0.1	0.15	-0.0369417	0.0147786	9.09656	-0.0279754	0.00583723	227.061	-0.0278411	0.00446809	93.0219
0.2	0.25	-0.0292455	0.0138245	7.95983	-0.0324527	0.00764528	389.506	-0.0368005	0.00445648	92.539
0.3	0.4	-0.0415594	0.015675	10.2334	-0.0437127	0.00764653	389.633	-0.0405366	0.0042496	84.1467
0.4	0.5	-0.0538733	0.0178463	13.2649	-0.0526566	0.00643334	275.8048	-0.0546736	0.00423566	83.5954

Table 4.9 – Comparison between the unbiased Monte Carlo estimation for the Vega of a digital Call option in the Stein-Stein type model for different values of the parameters σ_1 and σ_2.

in order to ensure that **(ND)** is satisfied. Here, the coefficients appearing in the dynamics (4.2.7) write

$$
\begin{aligned}
a_{S,i} &= \int_0^{\zeta_{i+1}-\zeta_i} \left[\sigma_1 \cos\left(\mu + (\bar{Y}_i - \mu)e^{-\lambda_Y s}\right) + \sigma_2\right]^2 ds, \\
a'_{S,i} &= -2\alpha \int_0^{\zeta_{i+1}-\zeta_i} e^{-\lambda_Y s} \sin\left(\mu + (\bar{Y}_i - \mu)e^{-\lambda_Y s}\right)\left[\sigma_1 \cos\left(\mu + (\bar{Y}_i - \mu)e^{-\lambda_Y s}\right) + \sigma_2\right] ds, \\
\rho_i &= \rho \frac{\int_0^{\zeta_{i+1}-\zeta_i} \left[\sigma_1 \cos\left(\mu + (\bar{Y}_i - \mu)e^{-\lambda_Y s}\right) + \sigma_2\right] ds}{\sigma_{S,i}\sqrt{\zeta_{i+1}-\zeta_i}}, \\
\rho'_i &= -\rho \frac{\sigma_1 \sigma_{S,i} \int_0^{\zeta_{i+1}-\zeta_i} e^{-\lambda_Y s} \sin\left(\mu + (\bar{Y}_i - \mu)e^{-\lambda_Y s}\right) ds}{a_{S,i}\sqrt{\zeta_{i+1}-\zeta_i}} \\
&\quad - \rho \frac{\sigma'_{S,i} \int_0^{\zeta_{i+1}-\zeta_i} \left[\sigma_1 \cos\left(\mu + (\bar{Y}_i - \mu)e^{-\lambda_Y s}\right) + \sigma_2\right] ds}{a_{S,i}\sqrt{\zeta_{i+1}-\zeta_i}}
\end{aligned}
$$

and no analytical expressions are available. However, a simple numerical integration method can be employed for the computation of the above integrals. We here use Simpson's 3/8 rule which for a real-valued $\mathcal{C}^4([0,T])$ function g writes as follows

$$
\forall t \in [0,T], \quad \int_0^t g(s)ds \approx \frac{t}{8}\left(g(0) + 3g\left(\frac{t}{3}\right) + 3g\left(\frac{2t}{3}\right) + g(t)\right)
$$

with an error given by $g^{(4)}(t')T^5/6480$ for some $t' \in [0,T]$.

The parameters of the unbiased Monte Carlo method and the Monte Carlo Euler-Maruyama scheme remain unchanged. The numerical results related to the price, Delta and Vega are provided in Table 4.10, Table 4.11, Table 4.12 respectively for the Call option and in Table 4.13, Table 4.14, Table 4.15 for the digital Call option. Here again, the unbiased estimators perform very well for all range of values of the parameters.

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Price	Half-width	Variance	Price	Half-width	Variance	Price	Half-width	Variance
0.1	0.15	0.110563	0.000847678	0.0299274	0.111364	0.000124637	0.10352	0.111372	0.000153086	0.109198
0.2	0.25	0.193444	0.00164016	0.112042	0.193513	0.000243912	0.396457	0.193538	0.000291832	0.396832
0.3	0.4	0.294835	0.00281222	0.329386	0.295101	0.000416276	1.15476	0.295277	0.000496958	1.15075
0.4	0.5	0.372503	0.0039339	0.644546	0.373974	0.000648198	2.79991	0.374822	0.000693144	2.23866

Table 4.10 – Comparison between the unbiased Monte Carlo estimation for the price of a Call option in the model with $\sigma_S(x) = \sigma_1 \cos(x) + \sigma_2$ for different values of the parameters σ_1 and σ_2.

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Delta	Half-width	Variance	Delta	Half-width	Variance	Delta	Half-width	Variance
0.1	0.15	0.556212	0.00279964	0.326447	0.558216	0.000913553	5.56155	0.558105	0.00114048	6.06062
0.2	0.25	0.576577	0.00321369	0.430147	0.573738	0.00101499	6.86522	0.574758	0.00125848	7.37962
0.3	0.4	0.608348	0.0038355	0.612705	0.60132	0.00118321	9.32936	0.601943	0.00145509	9.86555
0.4	0.5	0.629084	0.00444972	0.824655	0.623976	0.00135463	12.2284	0.625204	0.001653	12.7317

Table 4.11 – Comparison between the unbiased Monte Carlo estimation for the Delta of a Call option in the model with $\sigma_S(x) = \sigma_1 \cos(x) + \sigma_2$ for different values of the parameters σ_1 and σ_2.

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Vega	Half-width	Variance	Vega	Half-width	Variance	Vega	Half-width	Variance
0.1	0.15	-0.00775169	9.12481e-05	0.000346781	-0.00651665	0.00112781	8.47614	-0.00739643	0.00128018	7.63629
0.2	0.25	-0.0156966	0.000218523	0.00198885	-0.0142195	0.00220569	32.4202	-0.015778	0.00245716	28.1324
0.3	0.4	-0.0233796	0.000412417	0.00708403	-0.0174822	0.00373758	93.0911	-0.0179794	0.00413541	79.6854
0.4	0.5	-0.0307742	0.000670215	0.0187084	-0.0311582	0.00565561	213.151	-0.0304437	0.0042962	153.747

Table 4.12 – Comparison between the unbiased Monte Carlo estimation for the Vega of a Call option in the model with $\sigma_S(x) = \sigma_1 \cos(x) + \sigma_2$ for different values of the parameters σ_1 and σ_2.

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Price	Half-width	Variance	Price	Half-width	Variance	Price	Half-width	Variance
0	0.3	0.466531	0.00241015	0.241934	0.468532	0.000363404	0.880049	0.468702	0.000466623	1.01455
0.1	0.15	0.481467	0.00241291	0.242488	0.481189	0.000371395	0.91918	0.481203	0.000469696	1.02795
0.2	0.25	0.442993	0.00240127	0.240155	0.445142	0.000368266	0.903758	0.445054	0.000456271	0.970033
0.3	0.4	0.406075	0.00237603	0.235133	0.407653	0.000357256	0.850523	0.407567	0.000441207	0.907039
0.4	0.5	0.377704	0.00234699	0.22942	0.380003	0.000346223	0.798802	0.380009	0.000429336	0.858886

Table 4.13 – Comparison between the unbiased Monte Carlo estimation for the price of a digital Call option in the model with $\sigma_S(x) = \sigma_1 \cos(x) + \sigma_2$ for different values of the parameters σ_1 and σ_2.

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Delta	Half-width	Variance	Delta	Half-width	Variance	Delta	Half-width	Variance
0	0.3	1.23339	0.0306795	39.2017	1.24309	0.00156929	16.411	1.24507	0.0019971	18.5841
0.1	0.15	1.51796	0.0338824	47.8142	1.51053	0.00193524	24.9572	1.51051	0.00242614	27.4265
0.2	0.25	0.816965	0.0251319	26.3063	0.834766	0.00107561	7.70968	0.834635	0.00132928	8.23333
0.3	0.4	0.52951	0.0203232	17.2025	0.527783	0.000676488	3.04964	0.527829	0.000838507	3.27608
0.4	0.5	0.389601	0.0174702	12.7117	0.403047	0.000518279	1.79001	0.403017	0.0006438	1.93127

Table 4.14 – Comparison between the unbiased Monte Carlo estimation for the Delta of a digital Call option in the model with $\sigma_S(x) = \sigma_1 \cos(x) + \sigma_2$ for different values of the parameters σ_1 and σ_2.

σ_1	σ_2	Euler Scheme			Exponential sampling			Beta sampling		
		Vega	Half-width	Variance	Vega	Half-width	Variance	Vega	Half-width	Variance
0	0.3	0	0	0	0.000525633	0.00341519	77.7246	-0.00121724	0.00401747	75.2048
0.1	0.15	0.0107747	0.00798181	2.65345	0.00924064	0.00351717	82.4356	0.00920282	0.00407047	77.2024
0.2	0.25	0.0138531	0.00905046	3.41153	0.0129711	0.00349226	81.2719	0.0125995	0.00391887	71.5586
0.3	0.4	0.0123139	0.00853288	3.03249	0.0117377	0.00336349	75.389	0.0119138	0.00375579	65.7269
0.4	0.5	0.0153924	0.00953999	3.79057	0.0170944	0.00319472	68.0131	0.0168796	0.00363443	61.548

Table 4.15 – Comparison between the unbiased Monte Carlo estimation for the Vega of a digital Call option in the model with $\sigma_S(x) = \sigma_1 \cos(x) + \sigma_2$ for different values of the parameters σ_1 and σ_2.

4.6 Appendix

4.6.1 Proof of Theorem 4.3.1

The proof is divided into three steps. In the first part, we establish the probabilistic representation for a bounded and continuous function h. We then provide the extension to measurable maps satisfying the growth condition 4.3.1. We eventually conclude by establishing the L^p-moments when the jump times are distributed according to the Beta law.

Denote by $\mathcal{L}$ and $(\bar{\mathcal{L}}_t)_{t\geqslant 0}$ the infinitesimal generators of $(P_t)_{t\geqslant 0}$ and $(\bar{P}_t)_{t\geqslant 0}$ respectively given by

$$\begin{aligned}
\mathcal{L}f(x,y) &= (r-\frac{1}{2}a_S(y))\partial_x f(x,y) + \frac{1}{2}a_S(y)\partial_x^2 f(x,y) + b_Y(y)\partial_y f(x,y) \\
&\quad + \frac{1}{2}a_Y(y)\partial_y^2 f(x,y) + \rho(\sigma_S\sigma_Y)(y)\partial_{x,y}^2 f(x,y), \\
\bar{\mathcal{L}}_t f(x,y) &= (r-\frac{1}{2}a_S(m_t(y_0)))\partial_x f(x,y) + \frac{1}{2}a_S(m_t(y_0))\partial_x^2 f(x,y) + b_Y(m_t(y_0))\partial_y f(x,y) \\
&\quad + \frac{1}{2}a_Y(m_t(y_0))\partial_y^2 f(x,y) + \rho(\sigma_S\sigma_Y)(m_t(y_0))\partial_{x,y}^2 f(x,y)
\end{aligned}$$

for any $f \in \mathcal{C}_b^2(\mathbb{R}^2)$.

Step 1: Probabilistic representation for a bounded and continuous map h

We establish a first order expansion of the Markov semigroup $(P_t)_{t\geqslant 0}$ around $(\bar{P}_t)_{t\geqslant 0}$. We apply Itô's rule to the map $[0,t]\times\mathbb{R}^2 \ni (s,x,y) \mapsto P_{t-s}h(x,y) \in \mathcal{C}_b^{1,2}([0,t]\times\mathbb{R}^2)$ for $h \in \mathcal{C}_b^\infty(\mathbb{R}^2)$, observing that $\partial_s P_{t-s}h(x,y) = -\mathcal{L}P_{t-s}h(x,y)$. We obtain

$$\begin{aligned}
h(\bar{X}_T,\bar{Y}_T) &= P_T h(x_0,y_0) + \int_0^T \Big(\partial_s P_{T-s}h(\bar{X}_s,\bar{Y}_s) + \bar{\mathcal{L}}_s P_{T-s}h(\bar{X}_s,\bar{Y}_s)\Big)\,ds + M_T \\
&= P_T h(x_0,y_0) + \int_0^T (\bar{\mathcal{L}}_s - \mathcal{L})P_{T-s}h(\bar{X}_s,\bar{Y}_s)\,ds + M_T
\end{aligned}$$

where $M := (M_t)_{t\geqslant 0}$ is a square integrable martingale. We then take expectation in the previous expression and make use of Fubini's theorem so that

$$\begin{aligned}
&P_T h(x_0,y_0) \\
&= \bar{P}_T h(x_0,y_0) + \int_0^T \mathbb{E}[(\mathcal{L}-\bar{\mathcal{L}}_s)P_{T-s}h(\bar{X}_s,\bar{Y}_s)]\,ds \\
&= \mathbb{E}[h(\bar{X}_T^{x_0},\bar{Y}_T^{y_0})] + \int_0^T \mathbb{E}\Big[\frac{1}{2}(a_Y(\bar{Y}_s^{y_0}) - a_Y(m_s(y_0)))\partial_y^2 P_{T-s}h(\bar{X}_s^{x_0},\bar{Y}_s^{y_0})\Big]\,ds \\
&\quad + \int_0^T \mathbb{E}\Big[\frac{1}{2}(a_S(\bar{Y}_s^{y_0}) - a_S(m_s(y_0)))[\partial_x^2 P_{T-s}h(\bar{X}_s^{x_0},\bar{Y}_s^{y_0}) - \partial_x P_{T-s}h(\bar{X}_s^{x_0},\bar{Y}_s^{y_0})]\Big]\,ds \\
&\quad + \int_0^T \mathbb{E}\Big[(b_Y(\bar{Y}_s^{y_0}) - b_Y(m_s(y_0)))\partial_y P_{T-s}h(\bar{X}_s^{x_0},\bar{Y}_s^{y_0})\Big]\,ds \\
&\quad + \int_0^T \mathbb{E}\Big[\rho((\sigma_S\sigma_Y)(\bar{Y}_s^{x_0}) - (\sigma_S\sigma_Y)(m_s(y_0)))\partial_{x,y}^2 P_{T-s}h(\bar{X}_s^{x_0},\bar{Y}_s^{y_0})\Big]\,ds. \qquad (4.6.1)
\end{aligned}$$

We now rewrite the previous first order expansion using the Markov chain $(\bar{X}_i, \bar{Y}_i)_{0\leqslant i\leqslant N_T+1}$ and the renewal process N. From the previous identity, the definition of θ_{N_T+1} in (4.3.4) and the identity (4.2.6), we directly obtain

$$
\begin{aligned}
&P_T h(x_0, y_0)\\
&= \mathbb{E}[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1})\theta_{N_T+1}\mathbf{1}_{\{N_T=0\}}]\\
&\quad + \mathbb{E}\Big[((1-F(T-\zeta_1))f(\zeta_1))^{-1}\mathbf{1}_{\{N_T=1\}}\Big[\frac{1}{2}(a_S(\bar{Y}_1)-a_S(m_0))\mathcal{D}_1^{(1,1)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\\
&\quad -\frac{1}{2}(a_S(\bar{Y}_1)-a_S(m_0))\mathcal{D}_1^{(1)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)+\frac{1}{2}(a_Y(\bar{Y}_1)-a_Y(m_0))\mathcal{D}_1^{(2,2)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\\
&\quad +(b_Y(\bar{Y}_1)-b_Y(m_0))\mathcal{D}_1^{(2)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\\
&\quad +\rho((\sigma_S\sigma_Y)(\bar{Y}_1)-(\sigma_S\sigma_Y)(m_0))\mathcal{D}_1^{(1,2)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\Big]\Big]\\
&= \mathbb{E}[h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1})\theta_{N_T+1}\mathbf{1}_{\{N_T=0\}}] + \mathbb{E}\Big[((1-F(T-\zeta_1))f(\zeta_1))^{-1}\mathbf{1}_{\{N_T=1\}}\\
&\quad \times\Big[c_S^1\mathcal{D}_1^{(1,1)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)-c_S^1\mathcal{D}_1^{(1)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)+c_Y^1\mathcal{D}_1^{(2,2)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\\
&\quad + b_Y^1\mathcal{D}_1^{(2)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)+c_{S,Y}^1\mathcal{D}_1^{(1,2)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\Big]\Big]. \qquad (4.6.2)
\end{aligned}
$$

Next, we apply the IBP formula (4.2.14) with respect to the random vector $(\bar{X}_1, \bar{Y}_1)$ in the above expression. In order to do that rigorously, one first has to take the conditional expectation $\mathbb{E}_{0,1}[.]$ in the second term of the above equality. We thus obtain

$$
\begin{aligned}
&\mathbb{E}_{0,1}\Big[c_S^1\mathcal{D}_1^{(1,1)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)-c_S^1\mathcal{D}_1^{(1)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)+c_Y^1\mathcal{D}_1^{(2,2)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\\
&\qquad + b_Y^1\mathcal{D}_1^{(2)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)+c_{S,Y}^1\mathcal{D}_1^{(1,2)}P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\Big]\\
&= \mathbb{E}_{0,1}\Big[\Big[\mathcal{I}_1^{(1,1)}(c_S^1)-\mathcal{I}_1^{(1)}(c_S^1)+\mathcal{I}_1^{(2,2)}(c_Y^1)+\mathcal{I}_1^{(2)}(b_Y^1)+\mathcal{I}_1^{(1,2)}(c_{S,Y}^1)\Big]P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\Big]. \qquad (4.6.3)
\end{aligned}
$$

From Lemma 4.6.2 and the estimate (4.2.22), we get

$$
\begin{aligned}
&\mathbb{E}_{0,1}\Big[\Big|\Big[\mathcal{I}_1^{(1,1)}(c_S^1)-\mathcal{I}_1^{(1)}(c_S^1)+\mathcal{I}_1^{(2,2)}(c_Y^1)+\mathcal{I}_1^{(2)}(b_Y^1)+\mathcal{I}_1^{(1,2)}(c_{S,Y}^1)\Big]\Big|P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\Big|\Big]\\
&\leqslant C_T|h|_\infty\zeta_1^{-1/2} \qquad (4.6.4)
\end{aligned}
$$

for some positive constant C_T such that $T\mapsto C_T$ is non-decreasing. The previous estimate yields an integrable time singularity. Indeed, from the previous estimate and (4.2.6), one directly gets

$$
\begin{aligned}
&\mathbb{E}\Big[((1-F(T-\zeta_1))f(\zeta_1))^{-1}\mathbf{1}_{\{N_T=1\}}\Big|\mathbb{E}_{0,1}\Big[\Big[\mathcal{I}_1^{(1,1)}(c_S^1)-\mathcal{I}_1^{(1)}(c_S^1)+\mathcal{I}_1^{(2,2)}(c_Y^1)\\
&\qquad +\mathcal{I}_1^{(2)}(b_Y^1)+\mathcal{I}_1^{(1,2)}(c_{S,Y}^1)\Big]P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\Big]\Big]\\
&\leqslant C\mathbb{E}[((1-F(T-\zeta_1))f(\zeta_1))^{-1}\zeta_1^{-1/2}\mathbf{1}_{\{N_T=1\}}]\\
&= C\int_0^T s_1^{-1/2}\,ds_1<\infty.
\end{aligned}
$$

Coming back to (4.6.2) and using (4.6.3), we thus derive

$$\begin{aligned}
&P_T h(x_0,y_0)\\
&= \mathbb{E}[h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})\theta_{N_T+1}\mathbf{1}_{\{N_T=0\}}]\\
&\quad + \mathbb{E}\Big[((1-F(T-\zeta_1))f(\zeta_1))^{-1}\mathbf{1}_{\{N_T=1\}}\\
&\qquad \times \Big[\mathcal{I}_1^{(1,1)}(c_S^1) - \mathcal{I}_1^{(1)}(c_S^1) + \mathcal{I}_1^{(2,2)}(c_Y^1) + \mathcal{I}_1^{(2)}(b_Y^1) + \mathcal{I}_1^{(1,2)}(c_{S,Y}^1)\Big]P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\Big]\\
&= \mathbb{E}[h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})\theta_{N_T+1}\mathbf{1}_{\{N_T=0\}}] + \mathbb{E}\Big[P_{T-\zeta_1}h(\bar{X}_1,\bar{Y}_1)\theta_2\theta_1\mathbf{1}_{\{N_T=1\}}\Big]. \qquad (4.6.5)
\end{aligned}$$

Our aim now is to iterate the above first order expansion. We prove by induction the following formula: for any positive integer n, one has

$$\begin{aligned}
P_T h(x_0,y_0) = &\sum_{j=0}^{n-1}\mathbb{E}\left[h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})\prod_{i=1}^{N_T+1}\theta_i\mathbf{1}_{\{N_T=j\}}\right]\\
&+ \mathbb{E}\left[P_{T-\zeta_n}h(\bar{X}_n,\bar{Y}_n)\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}\right]. \qquad (4.6.6)
\end{aligned}$$

The case $n=1$ corresponds to (4.6.5). We thus assume that (4.6.6) holds at step n. We expand the last term appearing in the right-hand side of the previous equality using again (4.6.1), by then applying Lemma 4.6.1 and by finally performing IBPs as before.

To be more specific, using the notations introduced in Subsection 4.2.2, from (4.6.1) and a change of variable, for any (deterministic) $\zeta\in[0,T]$, one has

$$\begin{aligned}
&P_{T-\zeta}h(x,y)\\
&= \mathbb{E}[h(\bar{X}_T^{\zeta,x},\bar{Y}_T^{\zeta,y})]\\
&\quad + \int_\zeta^T \mathbb{E}\Big[\frac{1}{2}(a_S(\bar{Y}_s^{\zeta,y}) - a_S(m_{s-\zeta}(y)))[\partial_x^2 P_{T-s}h(\bar{X}_s^{\zeta,x},\bar{Y}_s^{\zeta,y}) - \partial_x P_{T-s}h(\bar{X}_s^{\zeta,x},\bar{Y}_s^{\zeta,y})]\Big]\,ds\\
&\quad + \int_\zeta^T \mathbb{E}\Big[\frac{1}{2}(a_Y(\bar{Y}_s^{\zeta,y}) - a_Y(m_{s-\zeta}(y)))\partial_y^2 P_{T-s}h(\bar{X}_s^{\zeta,x},\bar{Y}_s^{\zeta,y})\Big]\,ds\\
&\quad + \int_\zeta^T \mathbb{E}\Big[(b_Y(\bar{Y}_s^{\zeta,y}) - b_Y(m_{s-\zeta}(y)))\partial_y P_{T-s}h(\bar{X}_s^{\zeta,x},\bar{Y}_s^{\zeta,y})\Big]\,ds\\
&\quad + \int_\zeta^T \mathbb{E}\Big[\rho((\sigma_S\sigma_Y)(\bar{Y}_s^{\zeta,y}) - (\sigma_S\sigma_Y)(m_{s-\zeta}(y)))\partial_{x,y}^2 P_{T-s}h(\bar{X}_s^{\zeta,x},\bar{Y}_s^{\zeta,y})\Big]\,ds.
\end{aligned}$$

We take $\zeta=\zeta_n$, $(x,y)=(\bar{X}_{N_T},\bar{Y}_{N_T})$ in the previous equality, then multiply it

by $\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}$ and finally take expectation. We obtain

$$
\begin{aligned}
&\mathbb{E}\Big[P_{T-\zeta_n}h(\bar{X}_n,\bar{Y}_n)\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}\Big]\\
&=\mathbb{E}\Big[h(\bar{X}_T^{\zeta_n,\bar{X}_n},\bar{Y}_T^{\zeta_n,\bar{Y}_n})\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}\Big]\\
&+\mathbb{E}\Big[\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}\int_{\zeta_n}^{T}\frac{1}{2}(a_S(\bar{Y}_s^{\zeta_n,\bar{Y}_n})-a_S(m_{s-\zeta_n}(\bar{Y}_n)))\\
&\quad\times[\partial_x^2P_{T-s}h(\bar{X}_s^{\zeta_n,\bar{X}_n},\bar{Y}_s^{\zeta_n,\bar{Y}_n})-\partial_xP_{T-s}h(\bar{X}_s^{\zeta_n,\bar{X}_n},\bar{Y}_s^{\zeta_n,\bar{Y}_n})]ds\Big]\\
&+\mathbb{E}\Big[\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}\int_{\zeta_n}^{T}\frac{1}{2}(a_Y(\bar{Y}_s^{\zeta_n,\bar{Y}_n})-a_Y(m_{s-\zeta_n}(\bar{Y}_n)))\partial_y^2P_{T-s}h(\bar{X}_s^{\zeta_n,\bar{X}_n},\bar{Y}_s^{\zeta_n,\bar{Y}_n})ds\Big]\\
&+\mathbb{E}\Big[\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}\int_{\zeta_n}^{T}(b_Y(\bar{Y}_s^{\zeta_n,\bar{Y}_n})-b_Y(m_{s-\zeta_n}(\bar{Y}_n)))\partial_yP_{T-s}h(\bar{X}_s^{\zeta_n,\bar{X}_n},\bar{Y}_s^{\zeta_n,\bar{Y}_n})ds\Big]\\
&+\mathbb{E}\Big[\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}\int_{\zeta_n}^{T}\rho((\sigma_S\sigma_Y)(\bar{Y}_s^{\zeta_n,\bar{Y}_n})\\
&\qquad\qquad-(\sigma_S\sigma_Y)(m_{s-\zeta_n}(\bar{Y}_n)))\partial^2_{x,y}P_{T-s}h(\bar{X}_s^{\zeta_n,\bar{X}_n},\bar{Y}_s^{\zeta_n,\bar{Y}_n})\,ds\Big].
\end{aligned}
\tag{4.6.7}
$$

Now, from the very definition of the Markov chain $(\bar{X}_i,\bar{Y}_i)_{0\leqslant i\leqslant N_T+1}$ and of the weight sequence $(\theta_i)_{1\leqslant i\leqslant N_T+1}$ of Theorem 4.3.1, the first term of the above equality can be written as

$$
\mathbb{E}\Big[h(\bar{X}_T^{\zeta_n,\bar{X}_n},\bar{Y}_T^{\zeta_n,\bar{Y}_n})\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}\Big]=\mathbb{E}\Big[h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})\prod_{i=1}^{N_T+1}\theta_i\mathbf{1}_{\{N_T=n\}}\Big].
\tag{4.6.8}
$$

We now look at the second, third, fourth and fifth terms. Let us deal with the third and fourth terms. The others are treated in a similar manner and we will omit some technical details. We first take its conditional expectation w.r.t $\{\zeta_1=t_1,\cdots,\zeta_n=t_n,N_T=n\}$ and introduce the measurable function

$$
\begin{aligned}
G(t_1,\cdots,t_n,s,T):=\mathbb{E}\Big[\prod_{i=1}^{n+1}\theta_i\Big[&\frac{1}{2}(a_Y(\bar{Y}_s^{\zeta_n,\bar{Y}_n})-a_Y(m_{s-\zeta_n}(\bar{Y}_n)))\partial_y^2P_{T-s}h(\bar{X}_s^{\zeta_n,\bar{X}_n},\bar{Y}_s^{\zeta_n,\bar{Y}_n})\\
&+(b_Y(\bar{Y}_s^{\zeta_n,\bar{Y}_n})-b_Y(m_{s-\zeta_n}(\bar{Y}_n)))\partial_yP_{T-s}h(\bar{X}_s^{\zeta_n,\bar{X}_n},\bar{Y}_s^{\zeta_n,\bar{Y}_n})\Big]\Big|\zeta_1=t_1,\cdots,\zeta_n=t_n,N_T=n\Big]
\end{aligned}
$$

which satisfies

$$
\begin{aligned}
|G(t_1,\cdots,t_n,s,T)|&\leqslant C\mathbb{E}\Big[\prod_{i=1}^{n+1}|\theta_i|\Big(1+|\int_{t_n}^{s}\sigma_Y(m_{u-t_n})\,dB_u|\Big)\Big|\zeta_1=t_1,\cdots,\zeta_n=t_n,N_T=n\Big]\\
&\leqslant C\mathbb{E}\Big[\prod_{i=1}^{n+1}|\theta_i||\zeta_1=t_1,\cdots,\zeta_n=t_n,N_T=n\Big]
\end{aligned}
$$

where we used the boundedness of a_Y, the Lipschitz regularity of b_Y, the inequalities $\sup_{0\leqslant t\leqslant T}|\partial_y^\ell P_t h|_\infty \leqslant C$ for $\ell = 1, 2$ and, for the last inequality, the inequality $\mathbb{E}[|\int_{t_n}^s \sigma_Y(m_{u-t_n})\, dB_u||\mathcal{G}_n, \zeta_1 = t_1, \cdots, \zeta_n = t_n, N_T = n] \leqslant C|\sigma_Y|_\infty (s - t_n)^{1/2} \leqslant CT^{1/2}$. Recall now that $\mathbb{P}(N_T = n, \zeta_1 \in dt_1, \cdots, \zeta_n \in dt_n) = (1 - F(T - t_n)) \prod_{j=0}^{n-1} f(t_{j+1} - t_j) dt_1, \cdots, dt_n$ on the set $\Delta_n(T)$ so that from Lemma 4.6.2 and the estimate (4.2.22), we obtain

$$\begin{aligned}
&\mathbb{E}\Big[\int_{\zeta_n}^T |G(\zeta_1, \cdots, \zeta_n, s, T)|\mathbf{1}_{\{N_T=n\}} ds\Big] \\
&\leqslant C \int_{t_n}^T \int_{\Delta_n(T)} (T - t_n) \prod_{i=1}^n (t_i - t_{i-1})^{-1/2} dt_1 \cdots dt_n dt_{n+1} < \infty.
\end{aligned}$$

Hence, by Lemma 4.6.1, it holds

$$\begin{aligned}
&\mathbb{E}\Big[\mathbf{1}_{\{N_T=n\}} \prod_{i=1}^{n+1} \theta_i \int_{\zeta_n}^T \Big[\frac{1}{2}(a_Y(\bar{Y}_s^{\zeta_n,\bar{Y}_n}) - a_Y(m_{s-\zeta_n}(\bar{Y}_n)))\partial_y^2 P_{T-s} h(\bar{X}_s^{\zeta_n,\bar{X}_n}, \bar{Y}_s^{\zeta_n,\bar{Y}_n}) \\
&+ (b_Y(\bar{Y}_s^{\zeta_n,\bar{Y}_n}) - b_Y(m_{s-\zeta_n}(\bar{Y}_n)))\partial_y P_{T-s} h(\bar{X}_s^{\zeta_n,\bar{X}_n}, \bar{Y}_s^{\zeta_n,\bar{Y}_n})\Big] ds\Big] \\
&= \mathbb{E}\Big[\mathbf{1}_{\{N_T=n\}} \int_{\zeta_n}^T G(\zeta_1, \cdots, \zeta_n, s, T)\, ds\Big] \\
&= \mathbb{E}\Big[\prod_{i=1}^n \theta_i (1 - F(T - \zeta_{n+1}))^{-1} (f(\zeta_{n+1} - \zeta_n))^{-1} \\
&\times \Big[\frac{1}{2}(a_Y(\bar{Y}_{n+1}) - a_Y(m_n))\mathcal{D}_{n+1}^{(2,2)} P_{T-\zeta_{n+1}} h(\bar{X}_{n+1}, \bar{Y}_{n+1}) \\
&+ (b_Y(\bar{Y}_{n+1}) - b_Y(m_n))\mathcal{D}_{n+1}^{(2)} P_{T-\zeta_{n+1}} h(\bar{X}_{n+1}, \bar{Y}_{n+1})\Big]\mathbf{1}_{\{N_T=n+1\}}\Big].
\end{aligned}$$

Finally, we take the conditional expectation $\mathbb{E}_{n,n+1}[.]$ inside the above expectation and then employ the IBP formula (4.2.14), two times w.r.t. the diffusion coefficient and one time w.r.t the drift coefficient as done before. We obtain

$$\begin{aligned}
&\mathbb{E}\Big[\prod_{i=1}^n \theta_i (1 - F(T - \zeta_{n+1}))^{-1} (f(\zeta_{n+1} - \zeta_n))^{-1} \Big[\frac{1}{2}(a_Y(\bar{Y}_{n+1}) \\
&- a_Y(m_n))\mathcal{D}_{n+1}^{(2,2)} P_{T-\zeta_{n+1}} h(\bar{X}_{n+1}, \bar{Y}_{n+1}) \\
&+ (b_Y(\bar{Y}_{n+1}) - b_Y(m_n))\mathcal{D}_{n+1}^{(2)} P_{T-\zeta_{n+1}} h(\bar{X}_{n+1}, \bar{Y}_{n+1})\Big]\mathbf{1}_{\{N_T=n+1\}}\Big] \\
&= \mathbb{E}\Big[\prod_{i=1}^n \theta_i (1 - F(T - \zeta_{n+1}))^{-1} (f(\zeta_{n+1} - \zeta_n))^{-1} [\mathcal{I}_{n+1}^{(2,2)}(c_Y^{n+1}) + \mathcal{I}_{n+1}^{(2)}(b_Y^{n+1})] \\
&\times P_{T-\zeta_{n+1}} h(\bar{X}_{n+1}, \bar{Y}_{n+1})\mathbf{1}_{\{N_T=n+1\}}\Big].
\end{aligned}$$

In a completely analogous manner, we derive

$$\begin{aligned}
&\mathbb{E}\Big[\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}\int_{\zeta_n}^{T}\frac{1}{2}(a_S(\bar{Y}_s^{\zeta_n,\bar{Y}_n})-a_S(m_{s-\zeta_n}(\bar{Y}_n)))\\
&\times[\partial_x^2 P_{T-s}h(\bar{X}_s^{\zeta_n,\bar{X}_n},\bar{Y}_s^{\zeta_n,\bar{Y}_n})-\partial_x P_{T-s}h(\bar{X}_s^{\zeta_n,\bar{X}_n},\bar{Y}_s^{\zeta_n,\bar{Y}_n})]ds\Big]\\
&=\mathbb{E}\Big[\prod_{i=1}^{n}\theta_i(1-F(T-\zeta_{n+1}))^{-1}(f(\zeta_{n+1}-\zeta_n))^{-1}\\
&\times[\mathcal{I}_{n+1}^{(1,1)}(c_S^{n+1})-\mathcal{I}_{n+1}^{(1)}(c_S^{n+1})]P_{T-\zeta_{n+1}}h(\bar{X}_{n+1},\bar{Y}_{n+1})\mathbf{1}_{\{N_T=n+1\}}\Big]
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}\Big[\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}\int_{\zeta_n}^{T}\rho((\sigma_S\sigma_Y)(\bar{Y}_s^{\zeta_n,\bar{Y}_n})-(\sigma_S\sigma_Y)(m_{s-\zeta_n}(\bar{Y}_n)))\partial_{x,y}^2 P_{T-s}h(\bar{X}_s^{\zeta_n,\bar{X}_n},\bar{Y}_s^{\zeta_n,\bar{Y}_n})\,ds\Big]\\
&=\mathbb{E}\Big[\prod_{i=1}^{n}\theta_i(1-F(T-\zeta_{n+1}))^{-1}(f(\zeta_{n+1}-\zeta_n))^{-1}\mathcal{I}_{n+1}^{(1,2)}(c_{Y,S}^{n+1})P_{T-\zeta_{n+1}}h(\bar{X}_{n+1},\bar{Y}_{n+1})\mathbf{1}_{\{N_T=n+1\}}\Big].
\end{aligned}$$

Summing the three previous identities, we obtain that the sum of the second, third, fourth and fifth term in the right-hand side of (4.6.7) is equal to

$$\begin{aligned}
&\mathbb{E}\Big[\prod_{i=1}^{n}\theta_i(1-F(T-\zeta_{n+1}))^{-1}(f(\zeta_{n+1}-\zeta_n))^{-1}\Big[\mathcal{I}_{n+1}^{(1,1)}(a_S^{n+1})-\mathcal{I}_{n+1}^{(1)}(c_S^{n+1})\\
&\quad+\mathcal{I}_{n+1}^{(2,2)}(c_Y^{n+1})+\mathcal{I}_{n+1}^{(2)}(b_Y^{n+1})+\mathcal{I}_{n+1}^{(1,2)}(c_{Y,S}^{n+1})\Big]P_{T-\zeta_{n+1}}h(\bar{X}_{n+1},\bar{Y}_{n+1})\mathbf{1}_{\{N_T=n+1\}}\Big]\\
&=\mathbb{E}\Big[\prod_{i=1}^{n+2}\theta_i P_{T-\zeta_{n+1}}h(\bar{X}_{n+1},\bar{Y}_{n+1})\mathbf{1}_{\{N_T=n+1\}}\Big]
\end{aligned}$$

where we used the very definitions (4.3.3) and (4.3.4) of the weights $(\theta_i)_{1\leqslant i\leqslant N_T+1}$ on the set $\{N_T=n+1\}$. This concludes the proof of (4.6.6) at step $n+1$.

To conclude it remains to prove the absolute convergence of the first sum and the convergence to zero of the last term in (4.6.6). These two results follow directly from the boundedness of h and the general estimates on the product of weights established in Lemma 4.6.2.

Indeed, from Lemma 4.6.2, the estimate (4.2.22), the tower property of conditional expectation and the identity (4.2.6), we obtain

$$\begin{aligned}
&\mathbb{E}\Big[\,|h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})|\prod_{i=1}^{N_T+1}|\theta_i|\,\mathbf{1}_{\{N_T=j\}}\Big]\\
&\leqslant C^j|h|_\infty\mathbb{E}\Big[(1-F(T-\zeta_j))^{-1}\prod_{i=1}^{j}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-\frac{1}{2}}\mathbf{1}_{\{N_T=j\}}\Big]\\
&=C^j|h|_\infty\int_{\Delta_j(T)}\prod_{i=1}^{j}(s_i-s_{i-1})^{-\frac{1}{2}}\,d\mathbf{s}_j\\
&=C^j|h|_\infty T^{\frac{j}{2}}\frac{\Gamma^j(1/2)}{\Gamma(1+j/2)}
\end{aligned}$$

which in turn yields

$$\sum_{j=0}^{n-1} \mathbb{E}\Big[\,|h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})|\prod_{i=1}^{N_T+1}|\theta_i|\,\mathbf{1}_{\{N_T=j\}}\Big] \leqslant |h|_\infty \sum_{n\geqslant 0}\frac{(CT^{1/2})^n}{\Gamma(1+n/2)} = |h|_\infty E_{1/2,1}(CT^{1/2})$$

so that the series converge absolutely. Similarly,

$$\Big|\mathbb{E}\Big[P_{T-\zeta_n}h(\bar{X}_n,\bar{Y}_n)\prod_{i=1}^{n+1}\theta_i\mathbf{1}_{\{N_T=n\}}\Big]\Big| \leqslant C^n|h|_\infty T^{\frac{n}{2}}\frac{\Gamma^n(1/2)}{\Gamma(1+n/2)}$$

so that the remainder indeed vanishes as n goes to infinity. We thus conclude

$$\begin{aligned} P_Th(x_0,y_0) &= \sum_{n\geqslant 0}\mathbb{E}\Big[h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})\prod_{i=1}^{N_T+1}\theta_i\mathbf{1}_{\{N_T=n\}}\Big] \\ &= \mathbb{E}\Big[h(\bar{X}_{N_T+1},\bar{Y}_{N_T+1})\prod_{i=1}^{N_T+1}\theta_i\Big] \end{aligned} \tag{4.6.9}$$

for any $h\in\mathcal{C}_b^2(\mathbb{R}^2)$. We eventually extend the above representation formula to any bounded and continuous function h using a standard approximation argument. The remaining technical details are omitted.

Step 2: Extension to measurable map h satisfying the growth assumption (4.3.1)

We first extend the previous result to any bounded and measurable h. This follows from a monotone class argument that we now detail.

Let us first consider $h\in\mathcal{C}_b(\mathbb{R}^2)$. From Fubini's theorem, it holds

$$\begin{aligned} &\mathbb{E}\Big[h(\bar{X}_{n+1},\bar{Y}_{n+1})\prod_{i=1}^{n+1}\theta_i\Big|N_T=n,\zeta^{n+1}\Big] \\ &= \int_{\mathbb{R}^2}h(x,y)\,\mathbb{E}\Big[\bar{p}(T-\zeta_n,\bar{X}_n,\bar{Y}_n,x,y) \\ &\qquad \prod_{i=1}^{n+1}\theta_i(\bar{X}_{i-1},\bar{Y}_{i-1},\bar{X}_i,\bar{Y}_i,\zeta^{n+1})\Big|N_T=n,\zeta^{n+1}\Big]\,dxdy \end{aligned}$$

which can be justified as follows. From the upper-bound estimate (4.2.4), Lemma

4.6.2, Lemma 4.6.3, it holds

$$
\begin{aligned}
&\Big|\mathbb{E}\Big[\bar{p}(T-\zeta_n,\bar{X}_n,\bar{Y}_n,x,y)\prod_{i=1}^{n+1}\theta_i(\bar{X}_{i-1},\bar{Y}_{i-1},\bar{X}_i,\bar{Y}_i,\zeta^{n+1})\Big|N_T=n,\zeta^{n+1}\Big]\Big| \\
&\leqslant (1-F(T-\zeta_n))^{-1}\int_{(\mathbb{R}^2)^n}\bar{p}(T-\zeta_n,x_n,y_n,x,y) \\
&\quad\times\prod_{i=1}^{n}|\theta_i(x_{i-1},y_{i-1},x_i,y_i,\zeta^{n+1})|\bar{p}(\zeta_i-\zeta_{i-1},x_{i-1},y_{i-1},x_i,y_i)\,d\mathbf{x}_n d\mathbf{y}_n \\
&\leqslant (C_T)^{n+1}(1-F(T-\zeta_n))^{-1}\int_{(\mathbb{R}^2)^n}\bar{q}_c(T-\zeta_n,x_n,y_n,x,y) \\
&\quad\prod_{i=1}^{n}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-\frac{1}{2}}\times\bar{q}_c(\zeta_i-\zeta_{i-1},x_{i-1},y_{i-1},x_i,y_i)\,d\mathbf{x}_n d\mathbf{y}_n \\
&\leqslant (C_T)^{n+1}(1-F(T-\zeta_n))^{-1}\prod_{i=1}^{n}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-\frac{1}{2}}\bar{q}_c(T,x_0,y_0,x,y)
\end{aligned}
$$

for some $c := c(T,b_Y,\kappa) > 4\kappa$. Hence, from (4.6.9) and again Fubini's theorem, justified by the previous estimate and the fact that $\mathbb{E}\Big[(C_T)^{N_T+1}(1-F(T-\zeta_{N_T}))^{-1}\prod_{i=1}^{N_T}(f(\zeta_i-\zeta_{i-1}))^{-1}(\zeta_i-\zeta_{i-1})^{-\frac{1}{2}}\Big]<\infty$, one has

$$
P_T h(x_0,y_0)=\int_{\mathbb{R}^2}h(x,y)\mathbb{E}\Big[\bar{p}(T-\zeta_{N_T},\bar{X}_{N_T},\bar{Y}_{N_T},x,y)\prod_{i=1}^{N_T+1}\theta_i\Big]\,dxdy \tag{4.6.10}
$$

for any $h\in\mathcal{C}_b(\mathbb{R}^2)$. Moreover, from the previous computations, the following upper-bound holds

$$
\begin{aligned}
&\Big|\mathbb{E}\Big[\bar{p}(T-\zeta_{N_T},\bar{X}_{N_T},\bar{Y}_{N_T},x,y)\prod_{i=1}^{N_T+1}\theta_i\Big]\Big| \\
&=\Big|\sum_{n\geqslant 0}\int_{\Delta_n(T)}\mathbb{E}\Big[\bar{p}(T-s_n,\bar{X}_n,\bar{Y}_n,x,y)\prod_{i=1}^{n+1}\theta_i\,\Big|N_T=n,\zeta^{n+1}=(0,s_1,\cdots,s_n,T)\Big] \\
&\quad\times(1-F(T-s_n))\prod_{i=1}^{n}f(s_i-s_{i-1})\,d\mathbf{s}_n\Big| \\
&\leqslant\Big(\sum_{n\geqslant 0}\int_{\Delta_n(T)}(C_T)^{n+1}\prod_{i=1}^{n}(s_i-s_{i-1})^{-\frac{1}{2}}\,d\mathbf{s}_n\Big)\bar{q}_c(T,x_0,y_0,x,y) \\
&=CE_{1/2,1}(CT^{1/2})\bar{q}_c(T,x_0,y_0,x,y).
\end{aligned}
\tag{4.6.11}
$$

It now follows from (4.6.10) and a monotone class argument that the probabilistic representation formula (4.3.2) is valid for any real-valued bounded and measurable map h defined over $\mathbb{R}^2$. The extension to any measurable map h satisfying the growth assumption: $|h(x,y)|\leqslant C\exp(\gamma(|x|^2+|y|^2))$ for any $0\leqslant\gamma<(2cT)^{-1}$, c being the constant appearing on the right-hand side of (4.6.11), follows from the integral representation (4.6.10), the upper-bound (4.6.11) combined with a standard

approximation argument. Remaining technical details are omitted.

Step 3: Finite $L^p(\mathbb{P})$-moment for the probabilistic representation

If N is a renewal process with $Beta(\alpha,1)$ jump times then $f(s_i - s_{i-1}) = \frac{1-\alpha}{\bar\tau^{1-\alpha}} \frac{1}{(s_i-s_{i-1})^\alpha} \mathbf{1}_{[0,\bar\tau]}(s_i - s_{i-1})$ and $1 - F(T - s_n) = 1 - \left(\frac{T-s_n}{\bar\tau}\right)^{1-\alpha} \geqslant 1 - (\frac{T}{\bar\tau})^{1-\alpha}$, similarly to step 2, by Fubini's theorem, we get

$$\begin{aligned}
&\mathbb{E}\Big[|h(\bar X_{n+1}, \bar Y_{n+1})|^p \prod_{i=1}^{n+1} |\theta_i|^p \Big| N_T = n, \zeta^{n+1}\Big] \\
&= \int_{\mathbb{R}^2} |h(x,y)|^p \, \mathbb{E}\Big[\bar p(T - \zeta_n, \bar X_n, \bar Y_n, x, y) \\
&\qquad \prod_{i=1}^{n+1} |\theta_i(\bar X_{i-1}, \bar Y_{i-1}, \bar X_i, \bar Y_i, \zeta^{n+1})|^p \Big| N_T = n, \zeta^{n+1}\Big] \, dxdy.
\end{aligned}$$

The above formula is justified by Lemma 4.6.2 and Lemma 4.6.3 which yield

$$\begin{aligned}
&\mathbb{E}\Big[\bar p(T - \zeta_n, \bar X_n, \bar Y_n, x, y) \prod_{i=1}^{n+1} |\theta_i(\bar X_{i-1}, \bar Y_{i-1}, \bar X_i, \bar Y_i, \zeta^{n})|^p \Big| N_T = n, \zeta^{n+1}\Big] \\
&\leqslant C(1 - F(T - \zeta_n))^{-1} \int_{(\mathbb{R}^2)^n} \bar p(T - \zeta_n, x_n, y_n, x, y) \\
&\quad \times \prod_{i=1}^{n} |\theta_i(x_{i-1}, y_{i-1}, x_i, y_i, \zeta^{n+1})|^p \bar p(\zeta_i - \zeta_{i-1}, x_{i-1}, y_{i-1}, x_i, y_i) \, d\mathbf{x}_n d\mathbf{y}_n \\
&\leqslant C^{n+1}(1 - F(T - \zeta_n))^{-1} \int_{(\mathbb{R}^2)^n} \bar q_c(T - \zeta_n, x_n, y_n, x, y) \\
&\quad \times \prod_{i=1}^{n} (f(\zeta_i - \zeta_{i-1}))^{-p} (\zeta_i - \zeta_{i-1})^{-\frac{p}{2}} \bar q_c(\zeta_i - \zeta_{i-1}, x_{i-1}, y_{i-1}, x_i, y_i) \, d\mathbf{x}_n d\mathbf{y}_n \\
&\leqslant C^{n+1}(1 - F(T - \zeta_n))^{-1} \prod_{i=1}^{n} (f(\zeta_i - \zeta_{i-1}))^{-1} (\zeta_i - \zeta_{i-1})^{-\frac{p}{2}+\alpha p-\alpha} \bar q_c(T, x_0, y_0, x, y)
\end{aligned}$$

for some $c := c(T, b_Y, \kappa) > 4\kappa$. Now, using the fact that $\mathbb{E}\Big[C^{N_T+1}(1 - F(T - \zeta_{N_T}))^{-1} \prod_{i=1}^{N_T} (f(\zeta_i - \zeta_{i-1}))^{-1} (\zeta_i - \zeta_{i-1})^{-\frac{p}{2}+\alpha p-\alpha}\Big] < \infty$ as soon as $p(\frac{1}{2} - \alpha) < 1 - \alpha$

and that $h \in \mathcal{B}_\gamma(\mathbb{R}^2)$, from the previous computation, we obtain

$$\begin{aligned}
&\mathbb{E}[|h(\bar{X}_{N_T+1}, \bar{Y}_{N_T+1})|^p \prod_{i=1}^{N_T+1} |\theta_i|^p] \\
&= \sum_{i=0}^{+\infty} \mathbb{E}\Big[\mathbb{E}\Big[|h(\bar{X}_{n+1}, \bar{Y}_{n+1})|^p \prod_{i=1}^{n+1} |\theta_i|^p \Big| N_T, \zeta^{n+1}\Big] \mathbf{1}_{N_T=n}\Big] \\
&\leqslant \mathbb{E}\Big[C^{N_T+1}(1 - F(T - \zeta_{N_T}))^{-1} \prod_{i=1}^{N_T} (f(\zeta_i - \zeta_{i-1}))^{-1} (\zeta_i - \zeta_{i-1})^{-\frac{p}{2}+\alpha p+\alpha}\Big] \\
&\quad \times \int_{\mathbb{R}^2} e^{\gamma p(|x|^2+|y|^2)} \bar{q}_c(T, x_0, y_0, x, y)\, dxdy.
\end{aligned}$$

To conclude the proof, it suffices to note that the above space integral is finite as soon as $0 \leqslant \gamma p < (2cT)^{-1}$.

4.6.2 Proof of Lemma 4.4.1

Since $h \in \mathcal{C}_p^1(\mathbb{R}^2)$ and $\mathbb{E}_{i,n}\Big[|\bar{X}_{i+1}|^q + |\bar{Y}_{i+1}|^q\Big] < \infty$ a.s., for any $q \geqslant 1$, under **(AR)**, one may differentiate under the (conditional) expectation and deduce that $(x, y) \mapsto \mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1}, \bar{Y}_{i+1})\theta_{i+1} | (\bar{X}_i, \bar{Y}_i) = (x, y)\Big] \in \mathcal{C}_p^1(\mathbb{R}^2)$ for any $i \in \{0, \cdots, n\}$ *a.s.* The rest of the proof is divided into three parts.

Step 1: proofs of (4.4.1) and (4.4.2)

The transfer of derivatives formulae (4.4.1) and (4.4.2) are easily obtained by differentiating under expectation (which is allowed by the polynomial growth at infinity of h) noting from the definition of the Markov chain $\bar{X}$ that $\partial_{s_0}\bar{X}_1 = \partial_{s_0} \ln(s_0) = \frac{1}{s_0}$ and $\partial_{\bar{X}_i} h(\bar{X}_{i+1}, \bar{Y}_{i+1}) = \partial_{\bar{X}_{i+1}} h(\bar{X}_{i+1}, \bar{Y}_{i+1}) \partial_{\bar{X}_i} \bar{X}_{i+1} = \partial_{\bar{X}_{i+1}} h(\bar{X}_{i+1}, \bar{Y}_{i+1})$. Observe as well that from (4.2.15), the fact that $\partial_{\bar{X}_i} c_S^{i+1} = \partial_{\bar{X}_i} c_Y^{i+1} = \partial_{\bar{X}_i} b_Y^{i+1} = \partial_{\bar{X}_i} c_{Y,S}^{i+1} = \partial_{\bar{X}_i} \mathcal{I}_{i+1}^{(1)}(1) = \partial_{\bar{X}_i} \mathcal{I}_{i+1}^{(2)}(1) = 0$ and the very definition of the random variables $(\theta_i)_{1 \leqslant i \leqslant n+1}$, one has $\partial_{\bar{X}_i} \theta_{i+1} = 0$. This gives the identities (4.4.1) and (4.4.2).

Step 2: proofs of (4.4.3) and (4.4.4)

The proofs of (4.4.3) and (4.4.4) are more involved. Let us prove (4.4.3). We proceed by considering the difference between the term appearing on the left-hand side and the first two terms appearing on the right-hand side of (4.4.3). On the one

hand, using the IBP formula (4.2.14) and (4.2.8), we get

$$\begin{aligned}
&\partial_{\bar{Y}_i}\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\theta_{i+1}\Big]\\
&=\mathbb{E}_{i,n}\Big[\partial_{\bar{X}_{i+1}}h(\bar{X}_{i+1},\bar{Y}_{i+1})\partial_{\bar{Y}_i}\bar{X}_{i+1}\theta_{i+1}\Big]+\mathbb{E}_{i,n}\Big[\partial_{\bar{Y}_{i+1}}h(\bar{X}_{i+1},\bar{Y}_{i+1})\partial_{\bar{Y}_i}\bar{Y}_{i+1}\theta_{i+1}\Big]\\
&\quad+\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\partial_{\bar{Y}_i}\theta_{i+1}\Big]\\
&=\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\Big[\mathcal{I}^{(1)}_{i+1}(\partial_{\bar{Y}_i}\bar{X}_{i+1}\theta_{i+1})+\mathcal{I}^{(2)}_{i+1}(m_i'\theta_{i+1})\Big]\Big]\\
&\quad+\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\mathcal{I}^{(2)}_{i+1}\Big(\sigma'_{Y,i}\Big(\rho_i Z^1_{i+1}+\sqrt{1-\rho_i^2}Z^2_{i+1}\Big)\theta_{i+1}\Big)\Big]\\
&\quad+\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\mathcal{I}^{(2)}_{i+1}\Big(\sigma_{Y,i}\frac{\rho_i'}{\sqrt{1-\rho_i^2}}\Big(\sqrt{1-\rho_i^2}Z^1_{i+1}-\rho_i Z^2_{i+1}\Big)\theta_{i+1}\Big)\Big]\\
&\quad+\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\partial_{\bar{Y}_i}\theta_{i+1}\Big].
\end{aligned}$$

On the other hand, again from the IBP formula (4.2.14), we obtain

$$\begin{aligned}
&\mathbb{E}_{i,n}\Big[\partial_{\bar{X}_{i+1}}h(\bar{X}_{i+1},\bar{Y}_{i+1})\,\overrightarrow{\theta}^{e,X}_{i+1}\Big]+\mathbb{E}_{i,n}\Big[\partial_{\bar{Y}_{i+1}}h(\bar{X}_{i+1},\bar{Y}_{i+1})\,\overrightarrow{\theta}^{e,Y}_{i+1}\Big]\\
&\quad+\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\,\overrightarrow{\theta}^{c}_{i+1}\Big]\\
&=\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\Big[\mathcal{I}^{(1)}_{i+1}(\overrightarrow{\theta}^{e,X}_{i+1})+\mathcal{I}^{(2)}_{i+1}(\overrightarrow{\theta}^{e,Y}_{i+1})+\overrightarrow{\theta}^{c}_{i+1}\Big]\Big].
\end{aligned}$$

Combining the two previous identities, we see that the difference

$$\begin{aligned}
&\partial_{\bar{Y}_i}\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\theta_{i+1}\Big]-\Big(\mathbb{E}_{i,n}\Big[\partial_{\bar{X}_{i+1}}h(\bar{X}_{i+1},\bar{Y}_{i+1})\,\overrightarrow{\theta}^{e,X}_{i+1}\Big]\\
&\qquad+\mathbb{E}_{i,n}\Big[\partial_{\bar{Y}_{i+1}}h(\bar{X}_{i+1},\bar{Y}_{i+1})\,\overrightarrow{\theta}^{e,Y}_{i+1}\Big]+\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\,\overrightarrow{\theta}^{c}_{i+1}\Big]\Big)\\
&=\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\mathcal{I}^{(2)}_{i+1}\Big(m_i'\theta_{i+1}-\overrightarrow{\theta}^{e,Y}_{i+1}\Big)\Big]+\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\partial_{\bar{Y}_i}\theta_{i+1}\Big]\\
&\quad-\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\mathcal{I}^{(1)}_{i+1}(\overrightarrow{\theta}^{e,X}_{i+1})\Big]+\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\mathcal{I}^{(1)}_{i+1}(\partial_{\bar{Y}_i}\bar{X}_{i+1}\theta_{i+1})\Big]\\
&\quad+\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\mathcal{I}^{(2)}_{i+1}\Big(\sigma'_{Y,i}\Big(\rho_i Z^1_{i+1}+\sqrt{1-\rho_i^2}Z^2_{i+1}\Big)\theta_{i+1}\Big)\Big]\\
&\quad+\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\mathcal{I}^{(2)}_{i+1}\Big(\sigma_{Y,i}\frac{\rho_i'}{\sqrt{1-\rho_i^2}}\Big(\sqrt{1-\rho_i^2}Z^1_{i+1}-\rho_i Z^2_{i+1}\Big)\theta_{i+1}\Big)\Big] \qquad (4.6.12)\\
&\quad-\mathbb{E}_{i,n}\Big[h(\bar{X}_{i+1},\bar{Y}_{i+1})\,\overrightarrow{\theta}^{c}_{i+1}\Big].
\end{aligned}$$

Before proceeding, we provide the explicit expression for the quantity $\partial_{\bar{Y}_i}\theta_{i+1}$. Using the chain rule formula of Lemma 4.2.2, after some standard but cumbersome computations, we obtain

$$
\begin{aligned}
&\partial_{\bar{Y}_i}\theta_{i+1}\\
&= (f(\zeta_{i+1}-\zeta_i))^{-1}\Big[\mathcal{I}^{(1,1)}_{i+1}(\partial_{\bar{Y}_i}c^{i+1}_S) - \mathcal{I}^{(1)}_{i+1}(\partial_{\bar{Y}_i}c^{i+1}_S) + \mathcal{I}^{(2,2)}_{i+1}(\partial_{\bar{Y}_i}c^{i+1}_Y) + \mathcal{I}^{(2)}_{i+1}(\partial_{\bar{Y}_i}b^{i+1}_Y)\\
&\quad + \mathcal{I}^{(1,2)}_{i+1}(\partial_{\bar{Y}_i}c^{i+1}_{Y,S}) - \Big(\frac{\sigma'_{S,i}}{\sigma_{S,i}}\Big(2\mathcal{I}^{(1,1)}_{i+1}(c^{i+1}_S) - \mathcal{I}^{(1)}_{i+1}(c^{i+1}_S) + \mathcal{I}^{(1,2)}_{i+1}(c^{i+1}_{Y,S})\Big)\\
&\quad + \Big(\frac{\sigma'_{Y,i}}{\sigma_{Y,i}} - \frac{\rho'_i\rho_i}{1-\rho_i^2}\Big)\Big(2\mathcal{I}^{(2,2)}_{i+1}(c^{i+1}_Y) + \mathcal{I}^{(2)}_{i+1}(b^{i+1}_Y) + \mathcal{I}^{(1,2)}_{i+1}(c^{i+1}_{Y,S})\Big)\Big)\\
&\quad - \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}\Big(\mathcal{I}^{(1,2)}_{i+1}(c^{i+1}_S) + \mathcal{I}^{(2,1)}_{i+1}(c^{i+1}_S) - \mathcal{I}^{(2)}_{i+1}(c^{i+1}_S) + \mathcal{I}^{(2,2)}_{i+1}(c^{i+1}_{Y,S})\Big)\Big].
\end{aligned}
$$

Also, after some simple algebraic simplifications using the definitions of $\overrightarrow{\theta}^{e,Y}_{i+1}$ and $\overrightarrow{\theta}^{e,X}_{i+1}$ in (4.4.3), one obtains

$$
\mathcal{I}^{(2)}_{i+1}\Big(m'_i\theta_{i+1} - \overrightarrow{\theta}^{e,Y}_{i+1}\Big) = -(f(\zeta_{i+1}-\zeta_i))^{-1}\Big[\mathcal{I}^{(2,2)}_{i+1}\Big(\partial_{\bar{Y}_i}c^{i+1}_Y\Big) + \mathcal{I}^{(1,2)}_{i+1}\Big(\partial_{\bar{Y}_i}c^{i+1}_{Y,S}\Big)\Big]
$$

and

$$
\mathcal{I}^{(1)}_{i+1}(\overrightarrow{\theta}^{e,X}_{i+1}) = (f(\zeta_{i+1}-\zeta_i))^{-1}\mathcal{I}^{(1,1)}_{i+1}\Big(\partial_{\bar{Y}_i}c^{i+1}_S\Big).
$$

Combining the three previous identities and gathering similar terms, we obtain

$$
\begin{aligned}
&\mathcal{I}^{(2)}_{i+1}\Big(m'_i\theta_{i+1} - \overrightarrow{\theta}^{e,Y}_{i+1}\Big) + \partial_{\bar{Y}_i}\theta_{i+1} - \mathcal{I}^{(1)}_{i+1}(\overrightarrow{\theta}^{e,X}_{i+1})\\
&= (f(\zeta_{i+1}-\zeta_i))^{-1}\Big[-\mathcal{I}^{(1)}_{i+1}(\partial_{\bar{Y}_i}c^{i+1}_S) + \mathcal{I}^{(2)}_{i+1}(\partial_{\bar{Y}_i}b^{i+1}_Y)\\
&\quad - \Big(\frac{\sigma'_{S,i}}{\sigma_{S,i}}\Big(2\mathcal{I}^{(1,1)}_{i+1}(c^{i+1}_S) - \mathcal{I}^{(1)}_{i+1}(c^{i+1}_S) + \mathcal{I}^{(1,2)}_{i+1}(c^{i+1}_{Y,S})\Big)\\
&\quad + \Big(\frac{\sigma'_{Y,i}}{\sigma_{Y,i}} - \frac{\rho'_i\rho_i}{1-\rho_i^2}\Big)\Big(2\mathcal{I}^{(2,2)}_{i+1}(c^{i+1}_Y) + \mathcal{I}^{(2)}_{i+1}(b^{i+1}_Y) + \mathcal{I}^{(1,2)}_{i+1}(c^{i+1}_{Y,S})\Big)\Big)\\
&\quad - \frac{\rho'_i}{1-\rho_i^2}\frac{\sigma_{Y,i}}{\sigma_{S,i}}\Big(\mathcal{I}^{(1,2)}_{i+1}(c^{i+1}_S) + \mathcal{I}^{(2,1)}_{i+1}(c^{i+1}_S) - \mathcal{I}^{(2)}_{i+1}(c^{i+1}_S) + \mathcal{I}^{(2,2)}_{i+1}(c^{i+1}_{Y,S})\Big)\Big]. \quad (4.6.13)
\end{aligned}
$$

The previous identity will be used in the next step of the proof. Coming back to (4.6.12) and using the definition of the weight $\overrightarrow{\theta}^{c}_{i+1}$ allows to conclude the proof of the identity (4.4.3).

Step 3: The weight sequences $(\overrightarrow{\theta}^{e,Y}_i)_{1\leqslant i\leqslant n+1}$, $(\overrightarrow{\theta}^{e,X}_i)_{1\leqslant i\leqslant n+1}$ *and* $(\overrightarrow{\theta}^{c}_i)_{1\leqslant i\leqslant n+1}$ *and the related spaces* $\mathbb{M}_{i,n}(\bar{X},\bar{Y},\ell/2)$, $\ell\in\mathbb{Z}$.

In this last step, we prove the last statement of Lemma 4.4.1 concerning the weight sequences $(\overrightarrow{\theta}^{e,Y}_i)_{1\leqslant i\leqslant n+1}$, $(\overrightarrow{\theta}^{e,X}_i)_{1\leqslant i\leqslant n+1}$ and $(\overrightarrow{\theta}^{c}_i)_{1\leqslant i\leqslant n+1}$.

Following similar lines of reasonings as those used in the proof of Lemma 4.6.2, namely using the fact that d^{i+1}_S, d^{i+1}_Y, $d^{i+1}_{Y,S}$, $e^{Y,i+1}_S$, $e^{Y,i+1}_Y \in \mathbb{M}_{i,n}(\bar{X},\bar{Y},1/2)$ and

$\mathcal{D}^{(1)}_{i+1} d^{i+1}_S$, $\mathcal{D}^{(1,1)}_{i+1} d^{i+1}_S$, $\mathcal{D}^{(2)}_{i+1} d^{i+1}_Y$, $\mathcal{D}^{(2,2)}_{i+1} d^{i+1}_Y$, $\mathcal{D}^{(1)}_{i+1} d^{i+1}_{Y,S}$, $\mathcal{D}^{(2)}_{i+1} d^{i+1}_{Y,S}$, $\mathcal{D}^{(1,2)}_{i+1} d^{i+1}_{Y,S}$, $\mathcal{D}^{(1)}_{i+1} e^{Y,i+1}_S$, $\mathcal{D}^{(2)}_{i+1} e^{Y,i+1}_Y \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, 0)$ as well as Lemma 4.2.3, we conclude

$$f(\zeta_{i+1} - \zeta_i) \overrightarrow{\theta}^{e,Y}_{i+1} \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, -1/2), \quad i \in \{0, \cdots, n-1\}.$$

Note also that

$$\begin{aligned} e^{X,i+1}_S &= \partial_{\bar{Y}_i} c^{i+1}_S \\ &= \frac{1}{2}(a'_S(\bar{Y}_{i+1}) \partial_{\bar{Y}_i} \bar{Y}_{i+1} - a'_S(m_i) m'_i) \\ &= \frac{1}{2}(a'_S(\bar{Y}_{i+1}) - a'_S(m_i)) \partial_{\bar{Y}_i} \bar{Y}_{i+1} + \frac{1}{2} a'_S(m_i)(\partial_{\bar{Y}_i} \bar{Y}_{i+1} - m'_i) \end{aligned}$$

so that, using on the one hand the Lipschitz regularity of a'_S and on the other hand (4.2.8), from similar arguments as those used in the proof of Lemma 4.6.2, we conclude that

$$\frac{1}{2}(a'_S(\bar{Y}_{i+1}) - a'_S(m_i)) \partial_{\bar{Y}_i} \bar{Y}_{i+1}, \quad \frac{1}{2} a'_S(m_i)(\partial_{\bar{Y}_i} \bar{Y}_{i+1} - m'_i) \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, 1/2)$$

which in turn implies that $e^{X,i+1}_S \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, 1/2)$. Moreover, standard computations that we omit show that $\mathcal{D}^{(1)}_{i+1} e^{X,i+1}_S \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, 0)$ so that by Lemma 4.2.3 we deduce

$$f(\zeta_{i+1} - \zeta_i) \overrightarrow{\theta}^{e,X}_{i+1} \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, 0).$$

We now prove that $f(\zeta_{i+1} - \zeta_i) \overrightarrow{\theta}^{c}_{i+1} \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, -1/2)$ for any $i \in \{0, \cdots, n-1\}$. We use the decomposition

$$\begin{aligned} f(\zeta_{i+1} - \zeta_i) \overrightarrow{\theta}^{c}_{i+1} = f(\zeta_{i+1} - \zeta_i) \Big(& \mathcal{I}^{(2)}_{i+1} \Big(m'_i \theta_{i+1} - \overrightarrow{\theta}^{e,Y}_{i+1} \Big) + \partial_{\bar{Y}_i} \theta_{i+1} - \mathcal{I}^{(1)}_{i+1} (\overrightarrow{\theta}^{e,X}_{i+1}) \Big) \\ & + \mathcal{I}^{(1)}_{i+1} \Big(\partial_{\bar{Y}_i} \bar{X}_{i+1} f(\zeta_{i+1} - \zeta_i) \theta_{i+1} \Big) \\ & + \mathcal{I}^{(2)}_{i+1} \Big(\Big(\sigma'_{Y,i} \Big(\rho_i Z^1_{i+1} + \sqrt{1 - \rho_i^2} Z^2_{i+1} \Big) \\ & + \sigma_{Y,i} \frac{\rho'_i}{\sqrt{1 - \rho_i^2}} \Big(\sqrt{1 - \rho_i^2} Z^1_{i+1} - \rho_i Z^2_{i+1} \Big) \Big) f(\zeta_{i+1} - \zeta_i) \theta_{i+1} \Big). \end{aligned}$$

We first prove that $f(\zeta_{i+1} - \zeta_i) \Big(\mathcal{I}^{(2)}_{i+1} \Big(m'_i \theta_{i+1} - \overrightarrow{\theta}^{e,Y}_{i+1} \Big) + \partial_{\bar{Y}_i} \theta_{i+1} - \mathcal{I}^{(1)}_{i+1} (\overrightarrow{\theta}^{e,X}_{i+1}) \Big) \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, -1/2)$. We investigate each term appearing on the right-hand side of (4.6.13).

In particular, we first use the fact that c^{i+1}_S, c^{i+1}_Y, b^{i+1}_Y, $c^{i+1}_{Y,S}$ $\partial_{\bar{Y}_i} c^{i+1}_S, \partial_{\bar{Y}_i} b^{i+1}_Y \in \mathbb{M}_{i,n}(\bar{X}, \bar{Y}, 1/2)$ and the fact that when one applies the differential operators $\mathcal{D}^{(\alpha_1)}_{i+1}$, $\mathcal{D}^{(\alpha_1,\alpha_2)}_{i+1}$ to these elements the resulting random variables belong to $\mathbb{M}_{i,n}(\bar{X}, \bar{Y}, 0)$ for any $(\alpha_1, \alpha_2) \in \{1, 2\}^2$. From Lemma 4.2.3, we thus conclude that the elements $\mathcal{I}^{(1,1)}_{i+1}(c^{i+1}_S)$, $\mathcal{I}^{(1,2)}_{i+1}(c^{i+1}_S)$, $\mathcal{I}^{(2,1)}_{i+1}(c^{i+1}_S)$, $\mathcal{I}^{(2,2)}_{i+1}(c^{i+1}_Y)$, $\mathcal{I}^{(1,2)}_{i+1}(c^{i+1}_{Y,S})$, $\mathcal{I}^{(2,2)}_{i+1}(c^{i+1}_{Y,S})$ belong to $\mathbb{M}_{i,n}(\bar{X}, \bar{Y}, -1/2)$ and that $\mathcal{I}^{(1)}_{i+1}(c^{i+1}_S), \mathcal{I}^{(2)}_{i+1}(b^{i+1}_Y)$, $\mathcal{I}^{(1)}_{i+1}(\partial_{\bar{Y}_i} c^{i+1}_S), \mathcal{I}^{(2)}_{i+1}(\partial_{\bar{Y}_i} b^{i+1}_Y)$ belong to $\mathbb{M}_{i,n}(\bar{X}, \bar{Y}, 0)$. Moreover, using **(ND)**, one gets that there exists $C > 0$ such that for any $i \in$

$\{0,\cdots,n-1\}$, $|\sigma'_{S,i}/\sigma_{S,i}|+|\sigma'_{Y,i}/\sigma_{Y,i}|+|\sigma_{Y,i}/\sigma_{S,i}|+|\rho'_i/(1-\rho_i^2)|+|\rho'_i\rho_i/(1-\rho_i^2)|\leqslant C$. We thus conclude that $f(\zeta_{i+1}-\zeta_i)\Big(\mathcal{I}_{i+1}^{(2)}\Big(m'_i\theta_{i+1}-\overrightarrow{\theta}^{e,Y}_{i+1}\Big)+\partial_{\bar{Y}_i}\theta_{i+1}-\mathcal{I}_{i+1}^{(1)}(\overrightarrow{\theta}^{e,X}_{i+1})\Big)\in\mathbb{M}_{i,n}(\bar{X},\bar{Y},-1/2)$.

It thus suffices to prove $\mathcal{I}_{i+1}^{(1)}\Big(\partial_{\bar{Y}_i}\bar{X}_{i+1}f(\zeta_{i+1}-\zeta_i)\theta_{i+1}\Big)$, $\mathcal{I}_{i+1}^{(2)}\Big(\sigma'_{Y,i}\ (\rho_iZ_{i+1}^1+\sqrt{1-\rho_i^2}Z_{i+1}^2)f(\zeta_{i+1}-\zeta_i\theta_{i+1}\Big)$ and $\mathcal{I}_{i+1}^{(2)}\Big(\sigma_{Y,i}\frac{\rho'_i}{\sqrt{1-\rho_i^2}}\Big(\sqrt{1-\rho_i^2}Z_{i+1}^1-\rho_iZ_{i+1}^2\Big)\Big)f(\zeta_{i+1}-\zeta_i)\theta_{i+1}\Big)$ belong to $\mathbb{M}_{i,n}(\bar{X},\bar{Y},-1/2)$. In order to do this, we remark that

$$
\begin{aligned}
&\partial_{\bar{Y}_i}\bar{X}_{i+1}=-\frac{1}{2}a'_{S,i}+\sigma'_{S,i}Z_{i+1}^1\in\mathbb{M}_{i,n}(\bar{X},\bar{Y},1/2),\\
&\mathcal{D}_{i+1}^{(1)}(\partial_{\bar{Y}_i}\bar{X}_{i+1})=\frac{\sigma'_{S,i}}{\sigma_{S,i}}\in\mathbb{M}_{i,n}(\bar{X},\bar{Y},0),\\
&\sigma'_{Y,i}(\rho_iZ_{i+1}^1+\sqrt{1-\rho_i^2}Z_{i+1}^2)\in\mathbb{M}_{i,n}(\bar{X},\bar{Y},1/2),\\
&\mathcal{D}_{i+1}^{(2)}\Big(\sigma'_{Y,i}(\rho_iZ_{i+1}^1+\sqrt{1-\rho_i^2}Z_{i+1}^2\Big)=\frac{\sigma'_{Y,i}}{\sigma_{Y,i}}\in\mathbb{M}_{i,n}(\bar{X},\bar{Y},0),\\
&\sigma_{Y,i}\frac{\rho'_i}{\sqrt{1-\rho_i^2}}\Big(\sqrt{1-\rho_i^2}Z_{i+1}^1-\rho_iZ_{i+1}^2\Big)\in\mathbb{M}_{i,n}(\bar{X},\bar{Y},1/2),\\
&\mathcal{D}_{i+1}^{(2)}\Big(\sigma_{Y,i}\frac{\rho'_i}{\sqrt{1-\rho_i^2}}\Big(\sqrt{1-\rho_i^2}Z_{i+1}^1-\rho_iZ_{i+1}^2\Big)\Big)=-\frac{\rho_i\rho'_i}{1-\rho_i^2}\in\mathbb{M}_{i,n}(\bar{X},\bar{Y},0)
\end{aligned}
$$

and from Lemma 4.6.2, $f(\zeta_{i+1}-\zeta_i)\theta_{i+1}\in\mathbb{M}_{i,n}(\bar{X},\bar{Y},-1/2)$. From Lemma 4.2.3, it follows that $f(\zeta_{i+1}-\zeta_i)\theta_{i+1}\partial_{\bar{Y}_i}\bar{X}_{i+1}$, $f(\zeta_{i+1}-\zeta_i)\theta_{i+1}\sigma'_{Y,i}(\rho Z_{i+1}^1+\sqrt{1-\rho^2}Z_{i+1}^2)$, $f(\zeta_{i+1}-\zeta_i)\theta_{i+1}\sigma_{Y,i}\frac{\rho'_i}{\sqrt{1-\rho_i^2}}\Big(\sqrt{1-\rho_i^2}Z_{i+1}^1-\rho_iZ_{i+1}^2\Big)\in\mathbb{M}_{i,n}(\bar{X},\bar{Y},0)$. Now following similar computations as those employed in the proof of Lemma 4.6.2 and omitting some technical details we obtain $\mathcal{D}_{i+1}^{(\alpha)}(f(\zeta_{i+1}-\zeta_i)\theta_{i+1})\in\mathbb{M}_{i,n}(\bar{X},\bar{Y},-1)$ so that from the chain rule formula and Lemma 4.2.3, the random variables $\mathcal{D}_{i+1}^{(\alpha)}(f(\zeta_{i+1}-\zeta_i)\theta_{i+1}\partial_{\bar{Y}_i}\bar{X}_{i+1})$, $\mathcal{D}_{i+1}^{(\alpha)}(\sigma'_{Y,i}(\rho_iZ_{i+1}^1+\sqrt{1-\rho_i^2}Z_{i+1}^2)f(\zeta_{i+1}-\zeta_i)\theta_{i+1})$ and $\mathcal{D}_{i+1}^{(\alpha)}(\sigma_{Y,i}\frac{\rho'_i}{\sqrt{1-\rho_i^2}}(\sqrt{1-\rho_i^2}Z_{i+1}^1-\rho_iZ_{i+1}^2)f(\zeta_{i+1}-\zeta_i)\theta_{i+1})$ belong to $\mathbb{M}_{i,n}(\bar{X},\bar{Y},-1/2)$. From Lemma 4.2.3, we thus conclude that $\mathcal{I}_{i+1}^{(1)}\Big(\partial_{\bar{Y}_i}\bar{X}_{i+1}f(\zeta_{i+1}-\zeta_i)\theta_{i+1}\Big)$, $\mathcal{I}_{i+1}^{(2)}\Big(\sigma'_{Y,i}(\rho_iZ_{i+1}^1+\sqrt{1-\rho_i^2}Z_{i+1}^2)f(\zeta_{i+1}-\zeta_i)\theta_{i+1}\Big)$ and $\mathcal{I}_{i+1}^{(2)}\Big(\sigma_{Y,i}\frac{\rho'_i}{\sqrt{1-\rho_i^2}}\ (\sqrt{1-\rho_i^2}Z_{i+1}^1-\rho_iZ_{i+1}^2)f(\zeta_{i+1}-\zeta_i)\theta_{i+1}\Big)$ belong to $\mathbb{M}_{i,n}(\bar{X},\bar{Y},-1/2)$. From the preceding arguments, we eventually deduce that $f(\zeta_{i+1}-\zeta_i)\overrightarrow{\theta}^{c}_{i+1}\in\mathbb{M}_{i,n}(\bar{X},\bar{Y},-1/2)$ for any $i\in\{0,\cdots,n-1\}$.

Finally, from the very definition of the weights on the last time interval $\overrightarrow{\theta}^{e,Y}_{n+1}$ and $\overrightarrow{\theta}^{e,X}_{n+1}$ one directly gets that

$$
\begin{aligned}
(1-F(T-\zeta_n))\,\overrightarrow{\theta}^{e,Y}_{n+1}&=m'_n+\sigma'_{Y,n}\Big(\rho_nZ_{n+1}^1+\sqrt{1-\rho_n^2}Z_{n+1}^2\Big)\\
&\quad+\sigma_{Y,n}\frac{\rho'_n}{\sqrt{1-\rho_n^2}}\Big(\sqrt{1-\rho_n^2}Z_{n+1}^1-\rho_nZ_{n+1}^2\Big)
\end{aligned}
$$

belongs to $\mathbb{M}_{n,n}(\bar{X},\bar{Y},0)$ and that

$$(1-F(T-\zeta_n))\,\overrightarrow{\theta}^{e,X}_{n+1} = -\frac{1}{2}a'_{S,n} + \sigma'_{S,n}Z^1_{n+1}$$

belongs to $\mathbb{M}_{n,n}(\bar{X},\bar{Y},1/2)$. The proof is now complete.

4.6.3 Emergence of jumps in the renewal process N

The next result is used in the proof of the probabilistic representation in Theorem 4.3.1 to express that time integrals add jumps to the renewal process N. In what follows, N is a renewal process in the sense of Definition 4.2.1.

Lemma 4.6.1 *Let $n\in\mathbb{N}$ and $G:\{(t_1,\dots,t_{n+2}) : 0<t_1<\cdots<t_{n+1}<t_{n+2}:=T\}\to\mathbb{R}$ be a measurable function such that $\mathbb{E}\Big[\int_{\zeta_n}^T |G(\zeta_1,\dots,\zeta_n,s,T)|\mathbf{1}_{\{N_T=n\}}ds\Big]<\infty$. Then, it holds*

$$\begin{aligned}
&\mathbb{E}\Big[\int_{\zeta_n}^T G(\zeta_1,\dots,\zeta_n,s,T)\mathbf{1}_{\{N_T=n\}}ds\Big]\\
&=\mathbb{E}\Big[G(\zeta_1,\dots,\zeta_n,\zeta_{n+1},T)(1-F(T-\zeta_{n+1}))^{-1}\\
&\qquad(1-F(T-\zeta_n))(f(\zeta_{n+1}-\zeta_n))^{-1}\mathbf{1}_{\{N_T=n+1\}}\Big].
\end{aligned}$$

Proof. The proof follows by rewriting the above expectations using (4.2.6). We rewrite the expectation on the right-hand side in integral form. By Fubini's theorem, we obtain

$$\begin{aligned}
&\mathbb{E}\Big[G(\zeta_1,\dots,\zeta_n,\zeta_{n+1},T)\\
&\quad(1-F(T-\zeta_{n+1}))^{-1}(1-F(T-\zeta_n))(f(\zeta_{n+1}-\zeta_n))^{-1}\mathbf{1}_{\{N_T=n+1\}}\Big]\\
&=\int_{\Delta_{n+1}(T)} G(s_1,\cdots,s_{n+1},T)(1-F(T-s_{n+1}))^{-1}(1-F(T-s_n))(f(s_{n+1}-s_n))^{-1}\\
&\quad\times(1-F(T-s_{n+1}))\prod_{j=0}^{n} f(s_{j+1}-s_j)\,d\mathbf{s}_{n+1}\\
&=\int_{\Delta_n(T)}\int_{s_n}^T G(s_1,\cdots,s_{n+1},T)\,ds_{n+1}\,(1-F(T-s_n))\prod_{j=0}^{n-1} f(s_{j+1}-s_j)\,d\mathbf{s}_n.
\end{aligned}$$

This completes the proof.

Lemma 4.6.2 *Let $n\in\mathbb{N}$. On the set $\{N_T=n\}$, the sequence of weights $(\theta_i)_{1\leqslant i\leqslant n+1}$ defined by (4.3.3) and (4.3.4) satisfy: $\forall i\in\{1,\cdots,n\}$,*

$$f(\zeta_i-\zeta_{i-1})\theta_i\in\mathbb{M}_{i-1,n}(\bar{X},\bar{Y},-1/2),\quad (1-F(T-\zeta_n))\theta_{n+1}\in\mathbb{M}_{n,n}(\bar{X},\bar{Y},0). \tag{4.6.14}$$

Proof. We investigate each term appearing in the definition of $f(\zeta_i-\zeta_{i-1})\theta_i\in\mathbb{S}_{i-1,n}(\bar{X},\bar{Y})$ and seek to apply Lemma 4.2.3. From the Lipschitz property of a_S

and the space-time inequality (4.1.3), for any $c > 0$ and any $c' > 0$, the map $(x_{i-1}, y_{i-1}, x_i, y_i, \mathbf{s}_{n+1}) \mapsto c_S^i(x_{i-1}, y_{i-1}, x_i, y_i, \mathbf{s}_{n+1})$ satisfies

$$\begin{aligned}
&|c_S^i(x_{i-1}, y_{i-1}, x_i, y_i, \mathbf{s}_{n+1})|^p \bar{q}_c(s_i - s_{i-1}, x_{i-1}, y_{i-1}, x_i, y_i) \\
&\leqslant C|y_i - m_{i-1}(y_{i-1})|^p \bar{q}_c(s_i - s_{i-1}, x_{i-1}, y_{i-1}, x_i, y_i) \\
&\leqslant C(s_i - s_{i-1})^{p/2} \bar{q}_{c'}(s_i - s_{i-1}, x_{i-1}, y_{i-1}, x_i, y_i)
\end{aligned}$$

so that, the random variables $c_S^i \in \mathbb{M}_{i-1,n}(\bar{X}, \bar{Y}, 1/2)$, for any $i \in \{1, \cdots, n+1\}$. Moreover, since c_S^i does not depend on $\bar{X}_i$ and $\partial_{\bar{X}_i} \bar{Y}_i = 0$, one has

$$\mathcal{D}_i^{(1)} c_S^i = \mathcal{D}_i^{(1,1)} c_S^i = 0.$$

From Lemma 4.2.3, we thus conclude

$$\mathcal{I}_i^{(1)}(c_S^i) \in \mathbb{M}_{i-1,n}(\bar{X}, \bar{Y}, 0) \quad \text{and} \quad \mathcal{I}_i^{(1,1)}(c_S^i) \in \mathbb{M}_{i-1,n}(\bar{X}, \bar{Y}, -1/2), \quad i \in \{1, \cdots, n\}.$$

In a completely analogous manner, omitting some technical details, we derive

$$\mathcal{I}_i^{(2)}(b_Y^i) \in \mathbb{M}_{i-1,n}(\bar{X}, \bar{Y}, 0), \quad \text{and} \quad \mathcal{I}_i^{(1,2)}(c_{Y,S}^i), \mathcal{I}_i^{(2,2)}(c_Y^i) \in \mathbb{M}_{i-1,n}(\bar{X}, \bar{Y}, -1/2).$$

Hence, we obtain $f(\zeta_i - \zeta_{i-1})\theta_i \in \mathbb{M}_{i-1,n}(\bar{X}, \bar{Y}, -1/2)$, for any $i \in \{1, \cdots, n\}$. We finally observe that $(1 - F(T - \zeta_n))\theta_{n+1} = 1 \in \mathbb{M}_{n,n}(\bar{X}, \bar{Y}, 0)$. The proof is now complete.

Lemma 4.6.3 *Let $T > 0$ and n a positive integer. For any $\mathbf{s}_n = (s_1, \cdots, s_n) \in \Delta_n(T)$, any $(x, y) \in \mathbb{R}^2$ and any positive constant c there exist two positive constants C and $c' > c$ such that the transition density $(t, x, y) \mapsto \bar{q}_c(t, x_0, y_0, x, y)$ defined by (4.2.5) satisfies the following semigroup property:*

$$\begin{aligned}
&\int_{(\mathbb{R}^2)^n} \bar{q}_c(T - s_n, x_n, y_n, x, y) \times \bar{q}_c(s_n - s_{n-1}, x_{n-1}, y_{n-1}, x_n, y_n) \times \cdots \\
&\qquad \times \bar{q}_c(s_1, x_0, y_0, x_1, y_1)\, d\mathbf{x}_n d\mathbf{y}_n \\
&\leqslant C^n \bar{q}_{c'}(T, x_0, y_0, x, y).
\end{aligned}$$

Proof. The $dx_1 \cdots dx_n$ integrals are treated using the standard semigroup property of Gaussian kernels so that from the very definition of $\bar{q}_c$, it directly follows

$$\begin{aligned}
&\int_{(\mathbb{R}^2)^n} \bar{q}_c(T - s_n, x_n, y_n, x, y) \times \bar{q}_c(s_n - s_{n-1}, x_{n-1}, y_{n-1}, x_n, y_n) \times \cdots \\
&\qquad \times \bar{q}_c(s_1, x_0, y_0, x_1, y_1)\, d\mathbf{x}_n d\mathbf{y}_n \\
&= \frac{1}{\sqrt{2\pi c T}} e^{-\frac{(x - x_0)^2}{2cT}} \int_{\mathbb{R}^n} \frac{1}{\sqrt{2\pi c(T - s_n)}} e^{-\frac{(y - m_{T-s_n}(y_n))^2}{2c(T-s_n)}} \\
&\qquad\qquad \times \cdots \times \frac{1}{\sqrt{2\pi c s_1}} e^{-\frac{(y_1 - m_{s_1}(y_0))^2}{2cs_1}}\, d\mathbf{y}_n
\end{aligned}$$

In order to upper-bound the integral appearing in the right-hand side of the above identity. We now perform the change of variables $y_1 = m_{s_1}(z_1), y_2 = m_{s_2}(z_2), \cdots, y_n =$

$m_{s_n}(z_n)$. Observe that since b_Y admits a bounded first derivative the determinants of the Jacobians $J_{s_1}(z_1) := m'_{s_1}(z_1), \cdots, J_{T-s_n}(z_n) = m'_{T-s_n}(z_n)$ are uniformly bounded for any $(s_1, \cdots, s_n) \in \Delta_n(T)$. Remark also that from the semigroup property $m_{s_{i+1}-s_i}(m_{s_i}(z_i)) = m_{s_{i+1}}(z_i)$, for $1 \leqslant i \leqslant n$ with the convention $s_{n+1} = T$. Hence, for some positive constants C and $c' > c$ that may change from line to line, we get

$$
\begin{aligned}
&\frac{1}{\sqrt{2\pi cT}} e^{-\frac{(x-x_0)^2}{2cT}} \int_{\mathbb{R}^n} \frac{1}{\sqrt{2\pi c(T-s_n)}} e^{-\frac{(y-m_{T-s_n}(y_n))^2}{2c(T-s_n)}} \times \cdots \\
&\quad \times \frac{1}{\sqrt{2\pi c s_1}} e^{-\frac{(y_1 - m_{s_1}(y_0))^2}{2cs_1}} d\mathbf{y}_n \\
&\leqslant C^n \frac{1}{\sqrt{2\pi cT}} e^{-\frac{(x-x_0)^2}{2cT}} \int_{\mathbb{R}^n} \frac{1}{\sqrt{2\pi c(T-s_n)}} e^{-\frac{(y-m_T(z_n))^2}{2c(T-s_n)}} \times \cdots \\
&\quad \times \frac{1}{\sqrt{2\pi c s_1}} e^{-\frac{(m_{s_1}(z_1) - m_{s_1}(y_0))^2}{2cs_1}} d\mathbf{z}_n \\
&\leqslant C^n \frac{1}{\sqrt{2\pi cT}} e^{-\frac{(x-x_0)^2}{2cT}} \int_{\mathbb{R}^n} \frac{1}{\sqrt{2\pi c(T-s_n)}} e^{-C^{-1}\frac{(m_T^{-1}(y) - z_n)^2}{2c(T-s_n)}} \times \cdots \\
&\quad \times \frac{1}{\sqrt{2\pi c s_1}} e^{-C^{-1}\frac{(z_1 - y_0)^2}{2cs_1}} d\mathbf{z}_n \\
&= C^n \frac{1}{2\pi cT\sqrt{C}} e^{-\frac{(x-x_0)^2}{2cT}} e^{-\frac{(m_T^{-1}(y) - y_0)^2}{2cCT}} \\
&\leqslant C^n \bar{q}_{c'}(T, x_0, y_0, x, y)
\end{aligned}
$$

where we first used the bi-Lipschitz property of the flow $(s, x) \mapsto m_s(x)$ which yields

$$
\forall t \in [0, T],\ C^{-1}|x - z|^2 \leqslant |m_t(x) - m_t(z)|^2 \leqslant C|x - z|^2
$$

for some positive constant $C \geqslant 1$ and then the semigroup property satisfied by Gaussian kernels. This completes the proof.

4.6.4 Some useful formulas

We here provide some useful formulas in order to device the unbiased Monte Carlo algorithms based on Theorem 4.3.1 and Theorem 4.4.1. Their proofs follow from standard computations as those used in subsection 4.2.4 and are omitted.

We first provide some basis results to the computations of formulas of the price and the Greeks.

$$
\begin{aligned}
\mathcal{D}_i^{(2)}(c_S^i) &= \sigma_S(\bar{Y}_i)\sigma'_S(\bar{Y}_i), \\
\mathcal{D}_i^{(2)}(c_S^{i+1}) &= m'_i\Big[\sigma_S(\bar{Y}_{i+1})\sigma'_S(\bar{Y}_{i+1}) - \sigma_S(m_i)\sigma'_S(m_i)\Big], \\
\mathcal{D}_i^{(2)}(c_{Y,S}^i) &= \rho\sigma_Y\sigma'_S(\bar{Y}_i), \\
\mathcal{D}_i^{(2,2)}(c_{Y,S}^i) &= \rho\sigma_Y\sigma''_S(\bar{Y}_i), \\
\mathcal{D}_i^{(2)}(c_{Y,S}^{i+1}) &= \rho\sigma_Y m'_i(\sigma'_S(\bar{Y}_{i+1}) - \sigma'_S(m_i)), \\
\mathcal{D}_i^{(2)}(\mathcal{D}_{i+1}^{(2)}(c_{Y,S}^{i+1})) &= \mathcal{D}_{i+1}^{(2)}(\mathcal{D}_i^{(2)}(c_{Y,S}^{i+1})) = \rho\sigma_Y m'_i\sigma''_S(\bar{Y}_{i+1}),
\end{aligned}
$$

$$\begin{aligned}
\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1) &= \frac{1}{(1-\rho^2)(\sigma_S(m_{i-1}))^2(\zeta_i-\zeta_{i-1})},\\
\mathcal{D}_i^{(2)}\mathcal{I}_i^{(1)}(1) &= -\frac{\rho}{1-\rho^2}\frac{1}{\sigma_S(m_{i-1})\sigma_Y(\zeta_i-\zeta_{i-1})},\\
\mathcal{D}_i^{(1)}\mathcal{I}_i^{(2)}(1) &= -\frac{\rho}{1-\rho^2}\frac{1}{\sigma_S(m_{i-1})\sigma_Y(\zeta_i-\zeta_{i-1})},\\
\mathcal{D}_i^{(2)}\mathcal{I}_i^{(2)}(1) &= \frac{1}{(1-\rho^2)\sigma_Y^2(\zeta_i-\zeta_{i-1})},
\end{aligned}$$

$$\begin{aligned}
\mathcal{D}_{i-1}^{(2)}\mathcal{I}_i^{(1)}(1) &= -\frac{\sigma_S'(m_{i-1})m_{i-1}'}{\sigma_S(m_{i-1})}\mathcal{I}_i^{(1)}(1),\\
\mathcal{D}_{i-1}^{(2)}\mathcal{I}_i^{(2)}(1) &= 0,\\
\mathcal{D}_{i-1}^{(2)}\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1) &= -\frac{2\sigma_S'(m_{i-1})m_{i-1}'}{\sigma_S(m_{i-1})}\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1),\\
\mathcal{D}_{i-1}^{(2)}\mathcal{D}_i^{(2)}\mathcal{I}_i^{(1)}(1) &= -\frac{\sigma_S'(m_{i-1})m_{i-1}'}{\sigma_S(m_{i-1})}\mathcal{D}_i^{(2)}\mathcal{I}_i^{(1)}(1).
\end{aligned}$$

The following formulae are required in order to compute the weights $(\theta_i)_{1\leqslant i\leqslant N_T}$ appearing in the identity (4.3.2). Note that in our examples since $a_Y(.)$ is constant, one has $c_Y^i(.)\equiv 0$ for $i\in\{1,\cdots,N_T\}$. Hence, for $i\in\{1,\cdots,N_T\}$, one has:

$$\begin{aligned}
\mathcal{I}_i^{(1)}(c_S^i) &= c_S^i\mathcal{I}_i^{(1)}(1),\\
\mathcal{I}_i^{(1,1)}(c_S^i) &= c_S^i((\mathcal{I}_i^{(1)}(1))^2-\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1)),\\
\mathcal{I}_i^{(2)}(b_Y^i) &= b_Y^i\mathcal{I}_i^{(2)}(1)-b_Y'(\bar{Y}_i),\\
\mathcal{I}_i^{(1,2)}(c_{Y,S}^i) &= \mathcal{I}_i^{(2)}(c_{Y,S}^i\mathcal{I}_i^{(1)}(1)) = c_{Y,S}^i\mathcal{I}_i^{(1)}(1)\mathcal{I}_i^{(2)}(1)-\mathcal{I}_i^{(1)}(1)\mathcal{D}_i^{(2)}(c_{Y,S}^i)-c_{Y,S}^i\mathcal{D}_i^{(2)}\mathcal{I}_i^{(1)}(1).
\end{aligned}$$

The following formulae are needed in order to compute the weights for the Delta appearing in the identity (4.4.5), for $i\in\{1,\cdots,N_T\}$ one has:

$$\begin{aligned}
\mathcal{D}_i^{(1)}(\mathcal{I}_i^{(1,1)}(c_S^i)) &= 2c_S^i\mathcal{I}_i^{(1)}(1)\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1),\\
\mathcal{D}_i^{(1)}(\mathcal{I}_i^{(1)}(c_S^i)) &= c_S^i\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1),\\
\mathcal{D}_i^{(1)}(\mathcal{I}_i^{(2)}(b_Y^i)) &= b_Y^i\mathcal{D}_i^{(1)}\mathcal{I}_i^{(2)}(1),\\
\mathcal{D}_i^{(1)}(\mathcal{I}_i^{(1,2)}(c_{Y,S}^i)) &= c_{Y,S}^i\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1)\mathcal{I}_i^{(2)}(1)+c_{Y,S}^i\mathcal{D}_i^{(1)}\mathcal{I}_i^{(2)}(1)\mathcal{I}_i^{(1)}(1)-\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1)\mathcal{D}_i^{(2)}(c_{Y,S}^i).
\end{aligned}$$

$$\begin{aligned}
\mathcal{D}_i^{(1)}\theta_i &= (f(\zeta_i-\zeta_{i-1}))^{-1}\Big[\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1,1)}(c_S^i)-\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(c_S^i)+\mathcal{D}_i^{(1)}\mathcal{I}_i^{(2)}(b_Y^i)+\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1,2)}(c_{Y,S}^i)\Big],\\
\mathcal{I}_k^{(1)}(\theta_k) &= \mathcal{I}_k^{(1)}(1)\theta_k-\mathcal{D}_k^{(1)}\theta_k,\ k\leqslant N_T,\\
\mathcal{I}_{N_T+1}^{(1)}(\theta_{N_T+1}) &= \theta_{N_T+1}\mathcal{I}_{N_T+1}^{(1)}(1)-\mathcal{D}_{N_T+1}^{(1)}\theta_{N_T+1} = \theta_{N_T+1}\mathcal{I}_{N_T+1}^{(1)}(1).
\end{aligned}$$

The following formulae are required for the computation of the weights for the Vega

appearing in the identity (4.4.6), for $i \in \{1, \cdots, N_T\}$ it holds:

$$
\begin{aligned}
\mathcal{I}_{i+1}^{(1,1)}(d_S^{i+1}) &= m_i' \mathcal{I}_{i+1}^{(1,1)}(c_S^{i+1}),\\
\mathcal{I}_{i+1}^{(1)}(e_S^{Y,i+1}) &= -m_i' \mathcal{I}_{i+1}^{(1)}(c_S^{i+1}) + \mathcal{D}_i^{(2)}(c_{Y,S}^{i+1})\mathcal{I}_{i+1}^{(1)}(1) - \mathcal{D}_{i+1}^{(1)}\mathcal{D}_i^{(2)}(c_{Y,S}^{i+1}),\\
\mathcal{I}_{i+1}^{(2)}(e_Y^{Y,i+1}) &= m_i' \mathcal{I}_{i+1}^{(2)}(b_Y^{i+1}),\\
\mathcal{I}_{i+1}^{(1,2)}(d_{Y,S}^{i+1}) &= m_i' \mathcal{I}_{i+1}^{(1,2)}(c_{Y,S}^{i+1}),\\
\mathcal{I}_{i+1}^{(1)}(e_S^{X,i+1}) &= e_S^{X,i+1}\mathcal{I}_{i+1}^{(1)}(1) - \mathcal{D}_{i+1}^{(1)} e_S^{X,i+1} = \mathcal{D}_i^{(2)}(c_S^{i+1})\mathcal{I}_{i+1}^{(1)}(1) - \mathcal{D}_{i+1}^{(1)}\mathcal{D}_i^{(2)}(c_S^{i+1}),
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{D}_i^{(2)}(\mathcal{I}_i^{(1,1)}(c_S^i)) &= \mathcal{D}_i^{(2)}(c_S^i)(\mathcal{I}_i^{(1)}(1))^2 + 2c_S^i\mathcal{I}_i^{(1)}(1)\mathcal{D}_i^{(2)}\mathcal{I}_i^{(1)}(1) - \mathcal{D}_i^{(2)}(c_S^i)\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1),\\
\mathcal{D}_i^{(2)}(\mathcal{I}_i^{(1)}(c_S^i)) &= \mathcal{D}_i^{(2)}(c_S^i)\mathcal{I}_i^{(1)}(1) + c_S^i\mathcal{D}_i^{(2)}\mathcal{I}_i^{(1)}(1),\\
\mathcal{D}_i^{(2)}(\mathcal{I}_i^{(2)}(b_Y^i)) &= b_Y'(\bar{Y}_i)\mathcal{I}_i^{(2)}(1) + b_Y^i\mathcal{D}_i^{(2)}\mathcal{I}_i^{(2)}(1) - b_Y''(\bar{Y}_i),\\
\mathcal{D}_i^{(2)}(\mathcal{I}_i^{(1,2)}(c_{Y,S}^i)) &= \mathcal{D}_i^{(2)}(c_{Y,S}^i)\mathcal{I}_i^{(1)}(1)\mathcal{I}_i^{(2)}(1) + c_{Y,S}^i\mathcal{D}_i^{(2)}\mathcal{I}_i^{(2)}(1)\mathcal{I}_i^{(1)}(1) + c_{Y,S}^i\mathcal{I}_i^{(2)}(1)\mathcal{D}_i^{(2)}\mathcal{I}_i^{(1)}(1)\\
&\quad - \mathcal{I}_i^{(1)}(1)\mathcal{D}_i^{(2,2)}(c_{Y,S}^i) - 2\mathcal{D}_i^{(2)}(c_{Y,S}^i)\mathcal{D}_i^{(2)}\mathcal{I}_i^{(1)}(1),
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{D}_{i-1}^{(2)}(\mathcal{I}_i^{(1,1)}(c_S^i)) &= \mathcal{D}_{i-1}^{(2)}(c_S^i)(\mathcal{I}_i^{(1)}(1))^2 + 2c_S^i\mathcal{I}_i^{(1)}(1)\mathcal{D}_{i-1}^{(2)}\mathcal{I}_i^{(1)}(1)\\
&\quad - \mathcal{D}_{i-1}^{(2)}(c_S^i)\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1) - c_S^i\mathcal{D}_{i-1}^{(2)}(\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1)),\\
\mathcal{D}_{i-1}^{(2)}(\mathcal{I}_i^{(1)}(c_S^i)) &= \mathcal{D}_{i-1}^{(2)}(c_S^i)\mathcal{I}_i^{(1)}(1) + c_S^i\mathcal{D}_{i-1}^{(2)}\mathcal{I}_i^{(1)}(1),\\
\mathcal{D}_{i-1}^{(2)}(\mathcal{I}_i^{(2)}(b_Y^i)) &= b_Y^i\mathcal{D}_{i-1}^{(2)}\mathcal{I}_i^{(2)}(1) + \mathcal{I}_i^{(2)}(1)(b_Y'(\bar{Y}_i)\partial_{\bar{Y}_{i-1}}\bar{Y}_i - b_Y'(m_{i-1})m_{i-1}') - b_Y''(\bar{Y}_i)\partial_{\bar{Y}_{i-1}}\bar{Y}_i,\\
\mathcal{D}_{i-1}^{(2)}(\mathcal{I}_i^{(1,2)}(c_{Y,S}^i)) &= \mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i)\mathcal{I}_i^{(1)}(1)\mathcal{I}_i^{(2)}(1) + c_{Y,S}^i(\mathcal{I}_i^{(2)}(1)\mathcal{D}_{i-1}^{(2)}\mathcal{I}_i^{(1)}(1) + \mathcal{I}_i^{(1)}(1)\mathcal{D}_{i-1}^{(2)}\mathcal{I}_i^{(2)}(1))\\
&\quad - \mathcal{I}_i^{(1)}(1)\mathcal{D}_{i-1}^{(2)}(\mathcal{D}_i^{(2)}(c_{Y,S}^i)) - \mathcal{D}_{i-1}^{(2)}(\mathcal{I}_i^{(1)}(1))\mathcal{D}_i^{(2)}(c_{Y,S}^i) - c_{Y,S}^i\mathcal{D}_{i-1}^{(2)}\mathcal{D}_i^{(2)}\mathcal{I}_i^{(1)}(1)\\
&\quad - \mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i)\mathcal{D}_i^{(2)}\mathcal{I}_i^{(1)}(1).
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{D}_i^{(2)}\theta_i &= (f(\zeta_i - \zeta_{i-1}))^{-1}\Big[\mathcal{D}_i^{(2)}(\mathcal{I}_i^{(1,1)}(c_S^i)) - \mathcal{D}_i^{(2)}(\mathcal{I}_i^{(1)}(c_S^i)) + \mathcal{D}_i^{(2)}(\mathcal{I}_i^{(2)}(b_Y^i)) + \mathcal{D}_i^{(2)}(\mathcal{I}_i^{(1,2)}(c_{Y,S}^i))\Big],\\
\mathcal{D}_{i-1}^{(2)}\theta_i &= (f(\zeta_i - \zeta_{i-1}))^{-1}\Big[\mathcal{D}_{i-1}^{(2)}(\mathcal{I}_i^{(1,1)}(c_S^i)) - \mathcal{D}_{i-1}^{(2)}(\mathcal{I}_i^{(1)}(c_S^i)) + \mathcal{D}_{i-1}^{(2)}(\mathcal{I}_i^{(2)}(b_Y^i)) + \mathcal{D}_{i-1}^{(2)}(\mathcal{I}_i^{(1,2)}(c_{Y,S}^i))\Big],
\end{aligned}
$$

$$
\begin{aligned}
\overrightarrow{\theta}_i^{e,Y} &= m_{i-1}'\theta_i + (f(\zeta_i - \zeta_{i-1}))^{-1}(\mathcal{I}_i^{(1)}(1)\mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i) - \mathcal{D}_i^{(1)}\mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i)),\\
\mathcal{D}_i^{(2)}\overrightarrow{\theta}_i^{e,Y} &= m_{i-1}'\mathcal{D}_i^{(2)}\theta_i + (f(\zeta_i - \zeta_{i-1}))^{-1}\Big[\mathcal{D}_i^{(2)}(\mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i))\mathcal{I}_i^{(1)}(1)\\
&\quad + \mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i)\mathcal{D}_i^{(2)}\mathcal{I}_i^{(1)}(1) - \mathcal{D}_i^{(2)}\mathcal{D}_i^{(1)}\mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i)\Big],\\
\mathcal{I}_i^{(2)}(\overrightarrow{\theta}_i^{e,Y}) &= \overrightarrow{\theta}_i^{e,Y}\mathcal{I}_i^{(2)}(1) - \mathcal{D}_i^{(2)}\overrightarrow{\theta}_i^{e,Y},\\
\mathcal{I}_i^{(2)}(m_{i-1}'\theta_i - \overrightarrow{\theta}_i^{e,Y}) &= -(f(\zeta_i - \zeta_{i-1}))^{-1}\mathcal{I}_i^{(1,2)}(\mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i))\\
&= -(f(\zeta_i - \zeta_{i-1}))^{-1}\Big(\mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i)\mathcal{I}_i^{(1)}(1)\mathcal{I}_i^{(2)}(1) - \mathcal{I}_i^{(1)}(1)\mathcal{D}_i^{(2)}\mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i)\\
&\quad - \mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i)\mathcal{D}_i^{(1)}\mathcal{I}_i^{(2)}(1) - \mathcal{D}_i^{(1)}\mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i)\mathcal{I}_i^{(2)}(1) + \mathcal{D}_i^{(2)}\mathcal{D}_i^{(1)}\mathcal{D}_{i-1}^{(2)}(c_{Y,S}^i)\Big),
\end{aligned}
$$

$$
\begin{aligned}
\overrightarrow{\theta}_i^{e,X} &= (f(\zeta_i-\zeta_{i-1}))^{-1}\mathcal{I}_i^{(1)}(e_S^{X,i}) = (f(\zeta_i-\zeta_{i-1}))^{-1}(\mathcal{I}_i^{(1)}(1)\mathcal{D}_{i-1}^{(2)}(c_S^i) - \mathcal{D}_i^{(1)}\mathcal{D}_{i-1}^{(2)}(c_S^i)),\\
\mathcal{D}_i^{(1)}\overrightarrow{\theta}_i^{e,X} &= (f(\zeta_i-\zeta_{i-1}))^{-1}e_S^{X,i}\mathcal{D}_i^{(1)}\mathcal{I}_i^{(1)}(1),\\
\mathcal{I}_i^{(1)}(\overrightarrow{\theta}_i^{e,X}) &= \overrightarrow{\theta}_i^{e,X}\mathcal{I}_i^{(1)}(1) - \mathcal{D}_i^{(1)}\overrightarrow{\theta}_i^{e,X},\\
\mathcal{I}_i^{(1)}(\theta_i\mathcal{D}_{i-1}^{(2)}\bar{X}_i) &= (\theta_i\mathcal{I}_i^{(1)}(1) - \mathcal{D}_i^{(1)}\theta_i)\mathcal{D}_{i-1}^{(2)}\bar{X}_i - \mathcal{D}_i^{(1)}(\mathcal{D}_{i-1}^{(2)}\bar{X}_i)\theta_i,
\end{aligned}
$$

$$
\mathcal{I}_i^{(2)}\Big((\sqrt{1-\rho_{i-1}^2}Z_i^1-\rho_{i-1}Z_i^2)\theta_i\Big) = \Big(\sqrt{1-\rho_{i-1}^2}Z_i^1 - \rho_{i-1}Z_i^2\Big)(\mathcal{I}_i^{(2)}(1)\theta_i - \mathcal{D}_i^{(2)}\theta_i) + \frac{\rho_{i-1}\theta_i}{\sigma_{Y,i-1}\sqrt{1-\rho_{i-1}^2}},
$$

$$
\begin{aligned}
\overrightarrow{\theta}_i^{c} =& \mathcal{I}_i^{(2)}(m'_{i-1}\theta_i - \overrightarrow{\theta}_i^{e,Y}) + \sigma_{Y,i-1}\frac{\rho'_{i-1}}{\sqrt{1-\rho_{i-1}^2}}\mathcal{I}_i^{(2)}\Big((\sqrt{1-\rho_{i-1}^2}Z_i^1 - \rho_{i-1}Z_i^2)\theta_i\Big)\\
&+ \mathcal{I}_i^{(1)}(\theta_i\mathcal{D}_{i-1}^{(2)}\bar{X}_i) - \mathcal{I}_i^{(1)}(\overrightarrow{\theta}_i^{e,X}) + \mathcal{D}_{i-1}^{(2)}\theta_i,
\end{aligned}
$$

$$
\begin{aligned}
\overrightarrow{\theta}_{N_T+1}^{e,Y} =& \theta_{N_T+1} = \Big(m'_{N_T} + \sigma_{Y,N_T}\frac{\rho'_{N_T}}{\sqrt{1-\rho_{N_T}^2}}(\sqrt{1-\rho_{N_T}^2}Z_{N_T+1}^1 - \rho_{N_T}Z_{N_T+1}^2)\Big),\\
\overrightarrow{\theta}_{N_T+1}^{e,X} =& \theta_{N_T+1} = \Big(-\frac{1}{2}a'_{S,N_T} + \sigma'_{S,N_T}Z_{N_T+1}^1\Big),\\
\overrightarrow{\theta}_{N_T+1}^{c} =& 0,\\
\mathcal{I}_{N_T+1}^{(2)}(\overrightarrow{\theta}_{N_T+1}^{e,Y}) =& \overrightarrow{\theta}_{N_T+1}^{e,Y}\mathcal{I}_{N_T+1}^{(2)}(1) - \mathcal{D}_{N_T+1}^{(2)}\overrightarrow{\theta}_{N_T+1}^{e,Y} = \overrightarrow{\theta}_{N_T+1}^{e,Y}\mathcal{I}_{N_T+1}^{(2)}(1) + \theta_{N_T+1}\frac{\rho'_{N_T}\rho_{N_T}}{1-\rho_{N_T}^2},\\
\mathcal{I}_{N_T+1}^{(2)}(\overrightarrow{\theta}_{N_T+1}^{e,X}) =& \overrightarrow{\theta}_{N_T+1}^{e,X}\mathcal{I}_{N_T+1}^{(2)}(1) - \mathcal{D}_{N_T+1}^{(2)}\overrightarrow{\theta}_{N_T+1}^{e,X} = \overrightarrow{\theta}_{N_T+1}^{e,X}\mathcal{I}_{N_T+1}^{(2)}(1) - \theta_{N_T+1}\frac{\sigma'_{S,N_T}}{\sigma_{S,N_T}}.
\end{aligned}
$$

Bibliography

[1] Abdulle, A., Cohen, D., Vilmart, G., and Zygalakis, K. C. High weak order methods for stochastic differential equations based on modified equations. *SIAM Journal on Scientific Computing 34*, 3 (2012), A1800–A1823.

[2] Agarwal, A., and Gobet, E. Finite variance unbiased estimation of stochastic differential equations. *Proceedings of the 2017 Winter Simulation Conference* (2017), 1950–1961.

[3] Andersson, P., and Kohatsu-Higa, A. Unbiased simulation of stochastic differential equations using parametrix expansions. *Bernoulli 23*, 3 (2017), 2028–2057.

[4] Bally, V., and Kohatsu-Higa, A. A probabilistic interpretation of the parametrix method. *Ann. Appl. Probab. 25*, 6 (12 2015), 3095–3138.

[5] Bally, V., and Pagès, G. Error analysis of the optimal quantization algorithm for obstacle problems. *Stochastic Processes and their Applications 106*, 1 (2003), 1 – 40.

[6] Bally, V., and Pagès, G. A quantization algorithm for solving multidimensional discrete-time optimal stopping problems. *Bernoulli 9*, 6 (2003), 1003–1049.

[7] Beck, C., E, W., and Jentzen, A. Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *Journal of Nonlinear Science* (2017), 1–57.

[8] Beck, C., Hutzenthaler, M., Jentzen, A., and Kuckuck, B. An overview on deep learning-based approximation methods for partial differential equations. *arXiv preprint arXiv:2012.12348* (2020).

[9] Bender, C., and Denk, R. A forward scheme for backward sdes. *Stochastic Processes and their Applications 117*, 12 (2007), 1793–1812.

[10] Benveniste, A., Métivier, M., and Priouret, P. *Adaptive algorithms and stochastic approximations*, vol. 22. Springer Science & Business Media, 2012.

[11] Bergman, Y. Z. Option pricing with differential interest rates. *The Review of Financial Studies 8*, 2 (1995), 475–500.

[12] Bernis, G., Gobet, E., and Kohatsu-Higa, A. Monte Carlo evaluation of Greeks for multidimensional barrier and lookback options. *Math. Finance 13*, 1 (2003), 99–113. Conference on Applications of Malliavin Calculus in Finance (Rocquencourt, 2001).

[13] Bohn, B. *Error analysis of regularized and unregularized least-squares regression on discretized function spaces*. PhD thesis, Institute for Numerical Simulation, University of Bonn, (2017).

[14] Bohn, B. On the convergence rate of sparse grid least squares regression. In *Sparse Grids and Applications-Miami 2016*. Springer, (2018), pp. 19–41.

[15] Bouchard, B., Elie, R., and Touzi, N. Discrete-time approximation of bsdes and probabilistic schemes for fully nonlinear pdes. In *Advanced financial modelling*. De Gruyter, 2009, pp. 91–124.

[16] Bouchard, B., and Touzi, N. Discrete-time approximation and monte-carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Applications 111*, 2 (2004), 175–206.

[17] Bouchard, B., and Touzi, T. Discrete-time approximation and monte-carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Applications 111*, 2 (2004), 175 – 206.

[18] Briand, P., and Labart, C. Simulation of bsdes by wiener chaos expansion. *The Annals of Applied Probability 24*, 3 (2014), 1129–1171.

[19] Brooks, C. Rats handbook to accompany introductory econometrics for finance. *Cambridge Books* (2008).

[20] Bungartz, H.-J., and Griebel, M. Sparse grids. *Acta numerica 13* (2004), 147–269.

[21] Chan-Wai-Nam, Q., Mikael, J., and Warin, X. Machine learning for semi linear pdes. *Journal of Scientific Computing 79*, 3 (2019), 1667–1712.

[22] Chassagneux, J.-F. Linear multistep schemes for bsdes. *SIAM Journal on Numerical Analysis 52*, 6 (2014), 2815–2836.

[23] Chassagneux, J.-F., Chen, J., Frikha, N., and Zhou, C. A learning scheme by sparse grids and picard approximations for semilinear parabolic pdes. *arXiv preprint arXiv:2102.12051* (2021).

[24] Chassagneux, J.-F., and Crisan, D. Runge–kutta schemes for backward stochastic differential equations. *Annals of Applied Probability 24*, 2 (2014), 679–720.

[25] Chassagneux, J.-F., and Garcia Trillos, C. Cubature method to solve bsdes: Error expansion and complexity control. *Mathematics of Computation 89*, 324 (2020), 1895–1932.

[26] Chassagneux, J.-F., and Richou, A. Numerical stability analysis of the euler scheme for bsdes. *SIAM Journal on Numerical Analysis 53*, 2 (2015), 1172–1193.

[27] Chassagneux, J.-F., and Richou, A. Numerical simulation of quadratic bsdes. *The Annals of Applied Probability 26*, 1 (2016), 262–304.

[28] Chassagneux, J.-F., and Yang, M. Numerical approximation of singular forward-backward sdes. *arXiv preprint arXiv:2106.15496* (2021).

[29] Chen, J., Frikha, N., and Li, H. Probabilistic representation of integration by parts formulae for some stochastic volatility models with unbounded drift. *arXiv preprint arXiv:2011.10453* (2020).

[30] Cox, J. C. The constant elasticity of variance option pricing model. *Journal of Portfolio Management* (1996), 15.

[31] Crisan, D., and Manolarakis, K. Solving backward stochastic differential equations using the cubature method: application to nonlinear pricing. *SIAM Journal on Financial Mathematics 3*, 1 (2012), 534–571.

[32] Crisan, D., and Manolarakis, K. Second order discretization of backward sdes and simulation with the cubature method. *The Annals of Applied Probability 24*, 2 (2014), 652–678.

[33] Crisan, D., Manolarakis, K., and Nizar, T. On the monte carlo simulation of bsdes: An improvement on the malliavin weights. *Stochastic Processes and their Applications 120*, 7 (2010), 1133 – 1158.

[34] Crisan, D., Manolarakis, K., and Touzi, N. On the monte carlo simulation of bsdes: An improvement on the malliavin weights. *Stochastic Processes and their Applications 120*, 7 (2010), 1133–1158.

[35] Duflo, M. *Algorithmes stochastiques*, vol. 23 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer-Verlag, Berlin, (1996).

[36] E, W., Han, J., and 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 (2017), 349–380.

[37] El Karoui, N., Peng, S., and Quenez, M. C. Backward stochastic differential equations in finance. *Mathematical finance 7*, 1 (1997), 1–71.

[38] Feuersänger, C. *Sparse grid methods for higher dimensional approximation.* PhD thesis, University of Bonn, (2010).

[39] Fournié, E., Lasry, J.-M., Lebuchoux, J., and Lions, P.-L. Applications of Malliavin calculus to Monte-Carlo methods in finance. II. *Finance Stoch. 5*, 2 (2001), 201–236.

[40] Fournié, E., Lasry, J.-M., Lebuchoux, J., Lions, P.-L., and Touzi, N. Applications of Malliavin calculus to Monte Carlo methods in finance. *Finance Stoch. 3*, 4 (1999), 391–412.

[41] Friedman, A. *Partial differential equations of parabolic type*. Prentice-Hall, Inc., Englewood Cliffs, N.J., (1964).

[42] Frikha, N., Kohatsu-Higa, A., and Li, L. Integration by parts formula for killed processes: a point of view from approximation theory. *Electron. J. Probab. 24* (2019), 44 pp.

[43] Frommert, M., Pflüger, D., Riller, T., Reinecke, M., Bungartz, H.-J., and Ensslin, T. Efficient cosmological parameter sampling using sparse grids. *Monthly Notices of the Royal Astronomical Society 406*, 2 (2010), 1177–1189.

[44] Gatheral, J. *The volatility surface: a practitioner's guide*, vol. 357. John Wiley & Sons, 2011.

[45] Geiss, C., and Labart, C. Simulation of bsdes with jumps by wiener chaos expansion. *Stochastic Processes and their Applications 126*, 7 (2016), 2123–2162.

[46] Germain, M., Pham, H., and Warin, X. Deep backward multistep schemes for nonlinear pdes and approximation error analysis. *arXiv preprint arXiv:2006.01496* (2020).

[47] Germain, M., Pham, H., and Warin, X. Neural networks-based algorithms for stochastic control and pdes in finance. *arXiv preprint arXiv:2101.08068* (2021).

[48] Gobet, E., and Kohatsu-Higa, A. Computation of Greeks for barrier and look-back options using Malliavin calculus. *Electron. Comm. Probab. 8* (2003), 51–62.

[49] Gobet, E., Lemor, J.-P., and Warin, X. A regression-based monte carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability 15*, 3 (2005), 2172–2202.

[50] Gobet, E., López-Salas, J. G., Turkedjiev, P., and 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 (2016), C652–C677.

[51] Gobet, E., and Turkedjiev, P. Approximation of backward stochastic differential equations using malliavin weights and least-squares regression. *Bernoulli 22*, 1 (2016), 530–562.

[52] Griebel, M., and Oswald, P. Tensor product type subspace splittings and multilevel iterative methods for anisotropic problems. *Advances in Computational Mathematics 4*, 1 (1995), 171.

[53] Gu, Y., Yang, H., and Zhou, C. Selectnet: Self-paced learning for high-dimensional partial differential equations. *arXiv preprint arXiv:2001.04860* (2020).

[54] Hagan, P. S., Kumar, D., Lesniewski, A. S., and Woodward, D. E. Managing smile risk. *The Best of Wilmott 1* (2002), 249–296.

[55] Han, J., Jentzen, A., and E, W. Overcoming the curse of dimensionality: Solving high-dimensional partial differential equations using deep learning. *arXiv preprint arXiv:1707.02568* (2017).

[56] Han, J., Jentzen, A., and E, W. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences 115*, 34 (2018), 8505–8510.

[57] Han, J., Jentzen, A., and E, W. Algorithms for solving high dimensional pdes: From nonlinear monte carlo to machine learning. *arXiv preprint arXiv:2008.13333* (2020).

[58] Han, J., and Long, J. Convergence of the deep bsde method for coupled fbsdes. *Probability, Uncertainty and Quantitative Risk 5*, 1 (2020), 1–33.

[59] Henry-Labordère, P., Oudjane, N., Tan, X., Touzi, N., and Warin, X. Branching diffusion representation of semilinear pdes and monte carlo approximation. *Ann. Inst. H. Poincaré Probab. Statist. 55*, 1 (02 2019), 184–210.

[60] Henry-Labordère, P., Tan, X., and Touzi, N. Unbiased simulation of stochastic differential equations. *Ann. Appl. Probab. 27*, 6 (12 2017), 3305–3341.

[61] Heston, S. L. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies 6*, 2 (1993), 327–343.

[62] Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks 2*, 5 (1989), 359–366.

[63] Hornik, K., Stinchcombe, M., and White, H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks 3*, 5 (1990), 551–560.

[64] Hu, Y., Nualart, D., and Song, X. Malliavin calculus for backward stochastic differential equations and application to numerical solutions. *The Annals of Applied Probability 21*, 6 (2011), 2379–2423.

[65] Huré, C., Pham, H., and Warin, X. Deep backward schemes for high-dimensional nonlinear pdes. *Mathematics of Computation 89*, 324 (2020), 1547–1579.

[66] Hutzenthaler, M., Jentzen, A., Kruse, T., and E, W. Multilevel picard iterations for solving smooth semilinear parabolic heat equations. *arXiv preprint arXiv:1607.03295* (2016).

[67] Jiang, Y., and Li, J. Convergence of the deep bsde method for fbsdes with non-lipschitz coefficients. *arXiv preprint arXiv:2101.01869* (2021).

[68] Kapllani, L., and Teng, L. Deep learning algorithms for solving high dimensional nonlinear backward stochastic differential equations. *arXiv preprint arXiv:2010.01319* (2020).

[69] Kloeden, P. E., and Platen, E. Stochastic differential equations. In *Numerical Solution of Stochastic Differential Equations*. Springer, 1992, pp. 103–160.

[70] Kushner, H. J., and Yin, G. G. *Stochastic approximation and recursive algorithms and applications*, second ed., vol. 35 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, (2003). Stochastic Modelling and Applied Probability.

[71] Kusuoka, S., and Stroock, D. Applications of the Malliavin calculus. II. *J. Fac. Sci. Univ. Tokyo Sect. IA Math. 32*, 1 (1985), 1–76.

[72] Ladyženskaja, O. A., Solonnikov, V. A., and Ural'ceva, N. N. *Linear and quasilinear equations of parabolic type*. Translated from the Russian by S. Smith. Translations of Mathematical Monographs, Vol. 23. American Mathematical Society, Providence, R.I., (1968).

[73] Ma, J., and Zhang, J. Path regularity for solutions of backward stochastic differential equations. *Probability Theory and Related Fields 122*, 2 (2002), 163–190.

[74] Malliavin, P., and Thalmaier, A. *Stochastic Calculus of Variations in Mathematical Finance*. Springer Finance. Springer Berlin Heidelberg, 2005.

[75] Menozzi, S., and Lemaire, V. On some non asymptotic bounds for the euler scheme. *Electron. J. Probab. 15* (2010), 1645–1681.

[76] Ninomiya, S., and Victoir, N. Weak approximation of stochastic differential equations and application to derivative pricing. *Applied Mathematical Finance 15*, 2 (2008), 107–121.

[77] Nmeir, R. E. Quantization-based approximation of reflected bsdes with extended upper bounds for recursive quantization. *arXiv preprint arXiv:2105.07684* (2021).

[78] Nualart, D. *The Malliavin calculus and related topics*, second ed. Probability and its Applications (New York). Springer-Verlag, Berlin, 2006.

[79] Pagès, G. *Numerical Probability: An Introduction with Applications to Finance*. Universitext. Springer International Publishing, (2018).

[80] Pagès, G., and Sagna, A. Improved error bounds for quantization based numerical schemes for bsde and nonlinear filtering. *Stochastic Processes and their Applications 128*, 3 (2018), 847 – 883.

[81] Pardoux, E., and Peng, S. Adapted solution of a backward stochastic differential equation. *Systems Control Lett. 14*, 1 (1990), 55–61.

[82] Pardoux, E., and Peng, S. Backward stochastic differential equations and quasilinear parabolic partial differential equations. In *Stochastic partial differential equations and their applications (Charlotte, NC, 1991)*, vol. 176 of *Lect. Notes Control Inf. Sci.* Springer, Berlin, (1992), pp. 200–217.

[83] Pardoux, E., and Tang, S. Forward-backward stochastic differential equations and quasilinear parabolic PDEs. *Probab. Theory Related Fields 114*, 2 (1999), 123–150.

[84] Peng, S. Probabilistic interpretation for systems of quasilinear parabolic partial differential equations. *Stochastics and Stochastics Reports (Print)* (1991).

[85] Pham, H., Warin, X., and Germain, M. Neural networks-based backward scheme for fully nonlinear pdes. *arXiv preprint arXiv:1908.00412* (2019).

[86] Pham, H., Warin, X., and Germain, M. Neural networks-based backward scheme for fully nonlinear pdes. *SN Partial Differential Equations and Applications 2*, 1 (2021), 1–24.

[87] Reisinger, C. Analysis of linear difference schemes in the sparse grid combination technique. *IMA Journal of Numerical Analysis 33*, 2 (2013), 544–581.

[88] Robbins, H., and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics* (1951), 400–407.

[89] Stein, E. M., and Stein, J. C. Stock price distributions with stochastic volatility: an analytic approach. *The review of financial studies 4*, 4 (1991), 727–752.

[90] Takahashi, A., Tsuchida, Y., and Yamada, T. A new efficient approximation scheme for solving high-dimensional semilinear pdes: control variate method for deep bsde solver. *arXiv preprint arXiv:2101.09890* (2021).

[91] Teng, L. Gradient boosting-based numerical methods for high-dimensional backward stochastic differential equations. *arXiv preprint arXiv:2107.06673* (2021).

[92] Yamada, T., and Yamamoto, K. A second-order weak approximation of sdes using a markov chain without lévy area simulation. *Monte Carlo Methods and Applications 24*, 4 (2018), 289–308.

[93] Zhang, J. A numerical scheme for bsdes. *The Annals of Applied Probability 14*, 1 (2004), 459–488.