

# A Fourier-cosine method for an efficient computation of solutions to BSDEs

M. J. Ruijter\*

C. W. Oosterlee†

March 15, 2013

## Abstract

We develop a Fourier method to solve backward stochastic differential equations (BSDEs). General theta-discretization of the time-integrands leads to an induction scheme with conditional expectations. These are approximated by using Fourier-cosine series expansions, relying on the availability of a characteristic function. The method is applied to BSDEs with jumps. Numerical experiments demonstrate the applicability of BSDEs in financial and economic problems and show fast convergence of our efficient probabilistic numerical method.

**Keywords:** Backward stochastic differential equations, Fourier cosine expansion method, European options, market imperfections, jump-diffusion process, utility indifference pricing.

AMS Subject classification: 91G60, 60H35, 65C30, 65T50, 60E10.

## 1 Introduction

Whereas the theory and applications of classical *forward* stochastic differential equations (FSDE), with a prescribed initial value, is traditional and became widely known, we are concerned with *backward* stochastic differential equations (BSDEs). A BSDE is stochastic differential equation for which a terminal condition, instead of an initial condition, has been specified and its solution consists of a pair of processes. The linear type of equation was introduced by Bismut in [Bis73], where linear BSDEs were used in stochastic optimal control problems as adjoint equations in the stochastic version of the Pontryagin's maximum principle. The general notion of BSDE has been introduced by Pardoux and Peng [PP90]. They proved existence and uniqueness of solutions of BSDEs under some Lipschitz conditions on the driver function. Many researchers have attempted to relax these restrictions. For example, the authors in [LSM97] show existence of a minimal solution under more general assumption for the driver function, which is assumed to be continuous with linear growth in some of its arguments. Kobylanski [Kob00] provided uniqueness and existence results for a driver with quadratic growth in one of its argument. For a general introduction to BSDEs we refer to [Pha09, EKPQ97].

In recent years, BSDEs have received more attention in mathematical finance and economics. For example, the Black-Scholes formula for pricing options can be represented by a system of decoupled

---

\*Centrum Wiskunde & Informatica, Amsterdam, the Netherlands, email: [m.j.ruijter@cwi.nl](mailto:m.j.ruijter@cwi.nl).

†Centrum Wiskunde & Informatica, Amsterdam, the Netherlands, email: [c.w.oosterlee@cwi.nl](mailto:c.w.oosterlee@cwi.nl), and Delft University of Technology, Delft, the Netherlands.

forward-backward SDEs. Market imperfections can also be incorporated, such as different lending and borrowing rates for money, the presence of transaction costs or short sales constraints. These imperfections give rise to more involved nonlinear BSDEs. If the asset price follows a jump diffusion process then the option can not perfectly be replicated by assets and cash, i.e., the market is not complete. A way to value and hedge options in this setting is by utility indifference pricing, where a certain utility value is assigned to the possible profits and losses of the hedging portfolio. The pricing problem can be solved by means of a BSDE with jumps.

The well-known Feynman-Kac theorem gives a probabilistic representation for the solution of a linear parabolic partial differential equation (PDE) by means of the corresponding FSDE and a conditional expectation. The solution of a BSDE provides a probabilistic representation for *semilinear parabolic PDEs*, see for example [PP92], which is a generalisation of the Feynman-Kac theorem. Also the converse relation holds. This connection enables us to solve a semilinear PDE by probabilistic numerical methods, like Monte Carlo simulation techniques.

Probabilistic numerical methods to solve BSDEs may, for example, rely on time discretization of the stochastic process and approximations for the appearing conditional expectations. Least-squares Monte Carlo regression to approximate the conditional expectations is used in, for example, [LGW06, GLW05, BS12]. In this paper we employ a general theta-method for the time-integration ([Ise09]) and propose a new method to approximate the solution backwards in time. This approach is based on the COS method, which was developed in [FO08] for pricing financial options. The method is based on Fourier-cosine series expansions and relies on the characteristic function of the transitional density, which enables us to approximate the conditional expectations in a very efficient way. The characteristic function is in principle available for Lévy processes, or affine jump diffusion processes. The applicability of the resulting method is therefore quite general. We call the method the *BCOS method*, short for BSDE-COS method.

We start in Section 2 with notation, definitions, and a further introduction to BSDEs, where also the link with semilinear partial differential equations is stated. A general time discretization of the BSDE results in expressions with conditional expectations (Section 3). These conditional expectations are computed by the BCOS method (Section 4) and the problem is then solved backwards in time. We perform extensive numerical experiments in Section 5. Then, in Section 6, utility indifference pricing and the related maximization problems are discussed. We derive a numerical scheme for the resulting BSDE with jumps in Section 6.3. Results in Section 7 show the utility indifference ask and bid price.

## 2 Backward stochastic differential equations

We start with some notation and definitions, for which we follow the survey paper [EKPQ97]. Let  $\omega = (\omega_t)_{0 \leq t \leq T}$  be a standard one-dimensional Brownian motion on a filtered probability space  $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ , with  $\mathbb{F} = (\mathcal{F}_t)_{0 \leq t \leq T}$  the natural filtration of the Brownian motion  $\omega$ , and  $T$  a fixed finite time horizon. We denote by  $\mathbb{H}_T^2(\mathbb{R})$  the set of predictable processes  $\eta : \Omega \times [0, T] \rightarrow \mathbb{R}$  such that  $\mathbb{E} \left[ \int_0^T |\eta_t|^2 dt \right] < \infty$  and by  $\mathbb{L}_T^2(\mathbb{R})$  the set of  $\mathcal{F}_T$ -measurable random variables  $X : \Omega \rightarrow \mathbb{R}$  that are square integrable.

We consider the BSDE

$$-dY_t = f(t, Y_t, Z_t)dt - Z_t d\omega_t, \quad Y_T = \xi, \quad (2.1)$$

where function  $f : \Omega \times [0, T] \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  is  $\mathcal{P} \otimes \mathcal{B} \otimes \mathcal{B}$ -measurable.  $\mathcal{P}$  is the set of  $\mathcal{F}_t$ -progressively measurable scalar processes on  $\Omega \times [0, T]$ .  $f(\cdot)$  is the *generator* or *driver* of the process and the *terminal condition*  $\xi : \Omega \rightarrow \mathbb{R}$  is an  $\mathcal{F}_T$ -measurable random variable. For simplicity we use one-dimensional processes, but the BSDE theory can be extended to higher dimensions, with  $n$ -dimensional processes  $\omega_t$  and  $Y_t$  and an  $n \times d$ -dimensional  $Z_t$  process, as described in [EKPQ97].

A solution to BSDE (2.1) is given by a pair of processes  $(Y, Z)$ , with  $Y$  a continuous real-valued adapted process and  $Z$  a real-valued predictable process satisfying  $\int_0^T |Z_t|^2 dt < \infty$ ,  $\mathbb{P}$  a.s., satisfying

$$Y_t = \xi + \int_t^T f(s, Y_s, Z_s) ds - \int_t^T Z_s d\omega_s, \quad 0 \leq t \leq T. \quad (2.2)$$

Unlike an FSDE, the solution of a BSDE is thus a *pair* of adapted processes  $(Y, Z)$ . Note that BSDEs can not be considered as time-reversed FSDEs, because at time  $t$  the pair  $(Y_t, Z_t)$  is  $\mathcal{F}_t$ -measurable and the process does not yet “know” the terminal condition.

Function  $f$  and terminal condition  $\xi$  are called *standard parameters* for the BSDE, if

- $\xi \in \mathbb{L}_T^2(\mathbb{R})$ ,
- $f(\cdot, 0, 0) \in \mathbb{H}_T^2(\mathbb{R})$ ,
- $f$  is uniformly Lipschitz in  $y$  and  $z$ , i.e., there exists a constant  $L^{Lipz} > 0$  such that

$$|f(t, y_1, z_1) - f(t, y_2, z_2)| \leq L^{Lipz} (|y_1 - y_2| + |z_1 - z_2|), \quad \forall (y_1, z_1), \forall (y_2, z_2), \quad dt \otimes d\mathbb{P} \text{ a.s.} \quad (2.3)$$

A result from [EKPQ97, Pha09, PP90] is that, given a pair of standard parameters  $(f, \xi)$ , there exists a unique solution  $(Y, Z) \in \mathbb{H}_T^2(\mathbb{R}) \times \mathbb{H}_T^2(\mathbb{R})$  to BSDE (2.1).

**Markovian case for the BSDE** A *linear* parabolic PDE has a probabilistic representation by means of the Feynman-Kac theorem. Here, we consider a *semilinear* PDE of the form

$$-\frac{\partial v}{\partial t}(t, x) - \mathcal{L}v(t, x) - f(t, x, v(t, x), \sigma(t, x)D_x v(t, x)) = 0, \quad (t, x) \in [0, T] \times \mathbb{R}, \quad (2.4)$$

$$v(T, x) = g(x), \quad x \in \mathbb{R}, \quad (2.5)$$

with the differential operator of second order

$$\mathcal{L}v(t, x) = \mu(t, x)D_x v(t, x) + \frac{1}{2}\sigma^2(t, x)D_x^2 v(t, x). \quad (2.6)$$

This PDE also has a probabilistic representation, by means of the following FSDE,

$$X_t = x, \quad dX_s = \mu(s, X_s)ds + \sigma(s, X_s)d\omega_s, \quad t \leq s \leq T. \quad (2.7)$$

and BSDE

$$-dY_s = f(s, X_s^{t,x}, Y_s, Z_s)ds - Z_s d\omega_s, \quad Y_T = g(X_T^{t,x}), \quad (2.8)$$

whose terminal condition is determined by the terminal value of FSDE (2.7).  $X_s^{t,x}$  denotes the solution to (2.7) starting from  $x$  at time  $t$ , and  $(Y_s^{t,x}, Z_s^{t,x})$  is the corresponding solution to the BSDE.

The coefficients  $\sigma : [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$  and  $\mu : [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$  in (2.7) are assumed to be Lipschitz in  $x$  and satisfy a linear growth condition in  $x$ . Functions  $f : [0, T] \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  and  $g : \mathbb{R} \rightarrow \mathbb{R}$  are assumed to be uniformly continuous with respect to  $x$ . Moreover,  $f$  satisfies a Lipschitz condition in  $(y, z)$  and there exists a constant  $C$  such that  $|f(t, x, y, z) + |g(x)| \leq C(1 + |x|^p)$ ,  $p \geq 1/2$ .

The conditions on  $f$  and  $\xi$  guarantee the existence of a unique solution  $(Y, Z)$  to the BSDE (2.8). Together with the Markov property of the process  $X$ , we notice that there exists a deterministic function  $v(t, x)$  such that the solution  $Y$  of the BSDE is  $Y_s^{t,x} = v(s, X_s^{t,x})$ ,  $t \leq s \leq T$ . The solution of the BSDE is said to be Markovian as it can be written as a function of time and the state process  $X_s^{t,x}$ . The following results hold

**Result 2.1.** ([PP92, Pha09]) *Let  $v \in C^{1,2}$  be a classical solution to (2.4) and suppose there exists a constant  $C \geq 0$  such that, for all  $(t, x)$ ,  $|v(t, x)| + |\sigma(t, x)D_x v(t, x)| \leq C(1 + |x|)$ . Then the pair  $(Y, Z)$ , defined by*

$$Y_s^{t,x} = v(s, X_s^{t,x}), \quad Z_s^{t,x} = \sigma(s, X_s^{t,x})D_x v(s, X_s^{t,x}), \quad t \leq s \leq T, \quad (2.9)$$

*is the solution to BSDE (2.8) (a so-called verification result).*

*The converse result states: Suppose  $(Y, Z)$  is the solution to the BSDE, then the function defined by  $v(t, x) = Y_t^{t,x}$  is a viscosity solution to the PDE.*

The verification result follows from application of Itô's lemma to  $v(t, X_t)$  ([Pha09]):

$$\begin{aligned} dv(t, X_t) &= (v_t(t, X_t) + \mathcal{L}v(t, X_t))dt + \sigma(t, X_t)D_x v(t, X_t)d\omega_t \\ &= -f(t, X_t, v(t, X_t), \sigma(t, X_t)D_x v(t, X_t))dt + \sigma(t, X_t)D_x v(t, X_t)d\omega_t. \end{aligned} \quad (2.10)$$

So, solving the semilinear PDE or the corresponding BSDE results in the same solution. A PDE can be solved by applying numerical discretization techniques and for BSDEs probabilistic numerical methods are available. For example, Picard methods for  $Y$ , see [BD07, GL10], give rise to a sequence of 'easy' linear BSDEs. Another class of methods focuses on dynamic programming equations, see [BT04, Zha04]. Our probabilistic solution method to the BSDE is in this class and consists of two steps: First of all, the FSDE is simulated by an Euler scheme and the general theta-time discretization of the BSDE then results in expressions with conditional expectations (see Section 3). Secondly, the conditional expectations are computed by the BCOS method (see Section 4) and the problem is solved backwards in time.

### 3 Discretization of the BSDE

We wish to discretize the forward stochastic process,

$$X_0 = x_0 \text{ given, } X_t = X_0 + \int_0^t \mu(s, X_s)ds + \int_0^t \sigma(s, X_s)d\omega_s, \quad (3.1)$$

and the backward process<sup>1</sup>,

$$Y_t = \xi + \int_t^T f(s, Y_s, Z_s)ds - \int_t^T Z_s d\omega_s, \quad \xi = g(X_T). \quad (3.2)$$

For this, we define a partition  $\Delta : 0 = t_0 < t_1 < t_2 < \dots < t_m < \dots < t_M = T$ , with fixed time steps  $\Delta t := t_{m+1} - t_m$ . For notational convenience we write  $X_m = X_{t_m}$ ,  $Y_m = Y_{t_m}$ ,  $Z_m = Z_{t_m}$  and define  $\Delta\omega_m := \omega_{t_{m+1}} - \omega_{t_m}$ . With  $\omega_t$  a Wiener process, the increments  $\Delta\omega_m \sim \mathcal{N}(0, \sqrt{\Delta t})$  are normally distributed. The classical Euler discretization  $X^\Delta$  of the FSDE reads, in this case,

$$X_0^\Delta = x_0, \quad X_{m+1}^\Delta = X_m^\Delta + \mu(t_m, X_m^\Delta)\Delta t + \sigma(t_m, X_m^\Delta)\Delta\omega_m, \quad m = 0, \dots, M-1. \quad (3.3)$$

For the BSDE, we then start with

$$Y_m = Y_{m+1} + \int_{t_m}^{t_{m+1}} f(s, Y_s, Z_s)ds - \int_{t_m}^{t_{m+1}} Z_s d\omega_s. \quad (3.4)$$

---

<sup>1</sup>In our numerical experiments, the driver function does not depend on  $X_s$  and we omit this dependency here. However, the numerical method can be easily be generalized including this dependency.

By a basic Euler discretization, backwards in time, we would require the unknown value  $Y_{m+1}$  to approximate  $Y_m$ . This scheme does hence not suffice, as it would not take into account the adaptability constraints on  $Y$  and  $Z$ . To obtain a computationally viable backward induction scheme we should take conditional expectations, which will result in a similar approximation scheme to the BSDE as used in [ZWP09]. For the  $\mathcal{F}_{t_m}$ -measurable random variables  $Y_m$  and  $Z_m$  it holds that  $\mathbb{E}_m[Y_m] = Y_m$  and  $\mathbb{E}_m[Z_m] = Z_m$ , where  $\mathbb{E}_m[\cdot]$  represents the conditional expectation  $\mathbb{E}[\cdot | \mathcal{F}_{t_m}]$ . Taking conditional expectations at both sides of equation (3.4) then results in

$$\begin{aligned} Y_m &= \mathbb{E}_m[Y_{m+1}] + \int_{t_m}^{t_{m+1}} \mathbb{E}_m[f(s, Y_s, Z_s)] ds \\ &\approx \mathbb{E}_m[Y_{m+1}] + \Delta t \theta_1 f(t_m, Y_m, Z_m) + \Delta t (1 - \theta_1) \mathbb{E}_m[f(t_{m+1}, Y_{m+1}, Z_{m+1})], \quad \theta_1 \in [0, 1]. \end{aligned} \quad (3.5)$$

The integrand in the above equation (3.5) is a deterministic continuous function of time  $s$ , so that we can use the well-known theta-time discretization method to approximate the integral ([Ise09]). Multiplying both sides of equation (3.4) by  $\Delta\omega_m$ , taking the conditional expectation, and applying the theta-method also gives us

$$\begin{aligned} 0 &= \mathbb{E}_m[Y_{m+1} \Delta\omega_m] + \int_{t_m}^{t_{m+1}} \mathbb{E}_m[f(s, Y_s, Z_s)(\omega_s - \omega_{t_m})] ds - \int_{t_m}^{t_{m+1}} \mathbb{E}_m[Z_s] ds \\ &\approx \mathbb{E}_m[Y_{m+1} \Delta\omega_m] + \Delta t (1 - \theta_2) \mathbb{E}_m[f(t_{m+1}, Y_{m+1}, Z_{m+1}) \Delta\omega_m] \\ &\quad - \Delta t \theta_2 Z_m - \Delta t (1 - \theta_2) \mathbb{E}_m[Z_{m+1}], \quad \theta_2 \in [0, 1]. \end{aligned} \quad (3.6)$$

Note that for equations (3.5) and (3.6) we use two different time discretization parameters  $\theta_1$  and  $\theta_2$ , respectively. The above equations lead to a discrete-time approximation  $(Y^\Delta, Z^\Delta)$  for  $(Y, Z)$ :

$$Y_M^\Delta = g(X_M^\Delta), \quad Z_M^\Delta = \sigma D_x g(X_M^\Delta), \quad (3.7a)$$

for  $m = M - 1, \dots, 0$ :

$$\begin{aligned} Z_m^\Delta &= -\theta_2^{-1} (1 - \theta_2) \mathbb{E}_m[Z_{m+1}^\Delta] + \frac{1}{\Delta t} \theta_2^{-1} \mathbb{E}_m[Y_{m+1}^\Delta \Delta\omega_m] \\ &\quad + \theta_2^{-1} (1 - \theta_2) \mathbb{E}_m[f(t_{m+1}, Y_{m+1}^\Delta, Z_{m+1}^\Delta) \Delta\omega_m], \end{aligned} \quad (3.7b)$$

$$Y_m^\Delta = \mathbb{E}_m[Y_{m+1}^\Delta] + \Delta t \theta_1 f(t_m, Y_m^\Delta, Z_m^\Delta) + \Delta t (1 - \theta_1) \mathbb{E}_m[f(t_{m+1}, Y_{m+1}^\Delta, Z_{m+1}^\Delta)]. \quad (3.7c)$$

The use of  $\theta_1 = 0$  gives us an *explicit* scheme for  $Y_m^\Delta$ , whereas  $\theta_1 \in (0, 1]$  results in an *implicit* scheme. To solve for  $Z_m^\Delta$ , we should have obviously  $\theta_2 \neq 0$  in equation (3.7b), which gives an explicit scheme for  $Z_m^\Delta$ . For the terminal value  $Z_M^\Delta$  we use the relation from Result 2.1.

The terminal condition is a deterministic function of  $X_M^\Delta$  and  $X^\Delta$  is a Markov process. Then it is easily seen, using an induction argument, that there are deterministic functions  $y(t_m, x)$  and  $z(t_m, x)$ , so that

$$Y_m^\Delta = y(t_m, X_m^\Delta), \quad Z_m^\Delta = z(t_m, X_m^\Delta). \quad (3.8)$$

So, the random variables  $Y_m^\Delta$  and  $Z_m^\Delta$  are functions of  $X_m^\Delta$ , for each  $m = 0, \dots, M$ . The functions  $y(t_m, x)$  and  $z(t_m, x)$  are obtained in a backward manner by means of the following time iteration

$$y(t_M, x) = g(x), \quad z(t_M, x) = \sigma D_x g(x), \quad (3.9a)$$

for  $m = M - 1, \dots, 0$ :

$$\begin{aligned} z(t_m, x) &= -\theta_2^{-1} (1 - \theta_2) \mathbb{E}_m^x [z(t_{m+1}, X_{m+1}^\Delta)] + \frac{1}{\Delta t} \theta_2^{-1} \mathbb{E}_m^x [y(t_{m+1}, X_{m+1}^\Delta) \Delta\omega_m] \\ &\quad + \theta_2^{-1} (1 - \theta_2) \mathbb{E}_m^x [f(t_{m+1}, y(t_{m+1}, X_{m+1}^\Delta), z(t_{m+1}, X_{m+1}^\Delta)) \Delta\omega_m], \end{aligned} \quad (3.9b)$$

$$y(t_m, x) = \mathbb{E}_m^x [y(t_{m+1}, X_{m+1}^\Delta)] + \Delta t \theta_1 f(t_m, y(t_m, x), z(t_m, x))$$

$$+ \Delta t(1 - \theta_1) \mathbb{E}_m^x[f(t_{m+1}, y(t_{m+1}, X_{m+1}^\Delta), z(t_{m+1}, X_{m+1}^\Delta))], \quad (3.9c)$$

where the conditional expectation  $\mathbb{E}_m^x[\cdot]$  represents  $\mathbb{E}[\cdot | X_m^\Delta = x]$ . Note that functions  $y$  and  $z$  depend on the discretization partition  $\Delta$ .

Equations (3.9) provide us with a scheme to solve the BSDE backwards in time, starting at terminal time  $T$ . One could use least squares Monte Carlo methods, like the Longstaff-Schwartz method, to approximate the conditional expectations, see for example [LGW06, GLW05, BS12]. The authors of [BT04] apply a Malliavin based algorithm to solve them, whereas [MPSMT02] employs a binomial tree method. In the next section, we introduce a Fourier method to solve the BSDE.

## 4 BCOS method

In this section we explain our method of choice to compute the conditional expectations in (3.9) and solve the problem recursively, backwards in time. Our method is an extension of the COS method, which is a Fourier method developed in [FO08] to compute European option prices. The COS method for computing Bermudan options also consists of a backwards-in-time scheme to find the conditional expectations of the continuation value, see [FO09]. The method for solving BSDEs with a COS method is named the *BCOS method* here. First, in Section 4.1 we derive the COS formulas and define the Fourier-cosine coefficients. Then, Sections 4.2 and 4.3 are devoted to the approximation of functions  $z$  and  $y$ . Section 4.4 discusses the recursive recovery of the Fourier coefficients and Section 4.5 the error components.

### 4.1 COS formulas and Fourier-cosine coefficients

Without loss of generality, we assume a constant drift  $\mu$  and volatility  $\sigma$  here, and

$$X_{m+1}^\Delta = X_m^\Delta + \mu\Delta t + \sigma\Delta\omega_m. \quad (4.1)$$

In Remark 4.1 we will comment on the use of more general functions  $\mu(t, x)$  and  $\sigma(t, x)$ . Suppose we wish to approximate the expectation

$$I := \mathbb{E}_m^x[v(t_{m+1}, X_{m+1}^\Delta)], \quad (4.2)$$

where  $v$  represents a general functional. With the continuous transitional density function denoted by  $p(\zeta|x) = \mathbb{P}(X_{m+1}^\Delta = \zeta | X_m^\Delta = x)$ , we rewrite

$$I = \int_{\mathbb{R}} v(t_{m+1}, \zeta) p(\zeta|x) d\zeta. \quad (4.3)$$

We assume that the integrand decays to zero as  $\zeta \rightarrow \pm\infty$ . Because of that, we can truncate the infinite integration range to some finite interval  $[a, b] \subset \mathbb{R}$  without losing significant accuracy. This gives us the approximation

$$I_1 = \int_a^b v(t_{m+1}, \zeta) p(\zeta|x) d\zeta. \quad (4.4)$$

The notation  $I_i$  is used to denote the different approximations of  $I$  and keeps track of the numerical errors that set in at each step. Next, we replace the density function and function  $v$  by their Fourier-cosine series expansions on  $[a, b]$ , that is

$$p(\zeta|x) = \sum_{k=0}^{\infty} \mathcal{P}_k(x) \cos\left(k\pi \frac{\zeta - a}{b - a}\right), \quad (4.5a)$$

$$v(t_{m+1}, \zeta) = \sum_{k=0}^{\infty}' \mathcal{V}_k(t_{m+1}) \cos\left(k\pi \frac{\zeta - a}{b - a}\right), \quad (4.5b)$$

with series coefficients  $\{\mathcal{P}_k\}_{k=0}^{\infty}$  and  $\{\mathcal{V}_k\}_{k=0}^{\infty}$  given by

$$\mathcal{P}_k(x) := \frac{2}{b-a} \int_a^b p(\zeta|x) \cos\left(k\pi \frac{\zeta - a}{b - a}\right) d\zeta \quad \text{and} \quad (4.6a)$$

$$\mathcal{V}_k(t_{m+1}) := \frac{2}{b-a} \int_a^b v(t_{m+1}, \zeta) \cos\left(k\pi \frac{\zeta - a}{b - a}\right) d\zeta, \quad (4.6b)$$

respectively.  $\sum'$  indicates that the first term in the summation is weighted by one-half. Parseval's theorem and truncation of the series summations gives us the approximation

$$I_2 = \frac{b-a}{2} \sum_{k=0}^{N-1}' \mathcal{V}_k(t_{m+1}) \mathcal{P}_k(x). \quad (4.7)$$

The Fourier-cosine coefficients of the transitional density function,  $\mathcal{P}_k(x)$ , can now be approximated as follows ([FO08])

$$\begin{aligned} \mathcal{P}_k(x) &\approx \frac{2}{b-a} \int_{\mathbb{R}} p(\zeta|x) \cos\left(k\pi \frac{\zeta - a}{b - a}\right) d\zeta = \frac{2}{b-a} \Re\left(\varphi\left(\frac{k\pi}{b-a} \middle| x\right) e^{ik\pi \frac{x-a}{b-a}}\right) \\ &= \frac{2}{b-a} \Re\left(\phi\left(\frac{k\pi}{b-a}\right) e^{ik\pi \frac{x-a}{b-a}}\right) := \Phi_k(x). \end{aligned} \quad (4.8)$$

$\Re(\cdot)$  denotes taking the real part of the input argument and  $\varphi(\cdot|x)$  is the conditional *characteristic function* of  $X_{m+1}^{\Delta}$ , given  $X_m^{\Delta} = x$ . The characteristic function encountered here can be written as

$$\varphi(u|x) = \varphi(u|0)e^{iux} = \phi(u)e^{iux}, \quad \phi(u) := \exp\left(iu\mu\Delta t - \frac{1}{2}u^2\sigma^2\Delta t\right). \quad (4.9)$$

Inserting the above equations in (4.7) gives us the *COS formula* for approximation of  $I$ :

$$\hat{I} := I_3 = \sum_{k=0}^{N-1}' \mathcal{V}_k(t_{m+1}) \Re\left(\phi\left(\frac{k\pi}{b-a}\right) e^{ik\pi \frac{x-a}{b-a}}\right). \quad (4.10)$$

For solving the BSDE we need to deal with expectations of the form

$$\mathbb{E}_m^x[v(t_{m+1}, X_{m+1}^{\Delta})\Delta\omega_m]. \quad (4.11)$$

With the help of the equality (A.4) in Appendix A.1 (without jumps), they are computed by

$$\begin{aligned} \mathbb{E}_m^x[v(t_{m+1}, X_{m+1}^{\Delta})\Delta\omega_m] &\approx \mathbb{E}_m^x\left[\sum_{k=0}^{N-1}' \mathcal{V}_k(t_{m+1}) \cos\left(k\pi \frac{X_{m+1}^{\Delta} - a}{b - a}\right) \Delta\omega_m\right] \\ &= \sum_{k=0}^{N-1}' \mathcal{V}_k(t_{m+1}) \sigma \Delta t \Re\left(i \frac{k\pi}{b-a} \phi\left(\frac{k\pi}{b-a}\right) e^{ik\pi \frac{x-a}{b-a}}\right). \end{aligned} \quad (4.12)$$

Now we return to the BSDE problem (3.9), where we defined the deterministic functions  $y(t_m, X_m^{\Delta}) := Y_m^{\Delta}$  and  $z(t_m, X_m^{\Delta}) := Z_m^{\Delta}$ . Let  $\mathcal{Y}_k(t_{m+1})$  be the Fourier-cosine coefficients of  $y(t_{m+1}, x)$  in (3.9c), i.e.,

$$\mathcal{Y}_k(t_{m+1}) = \frac{2}{b-a} \int_a^b y(t_{m+1}, x) \cos\left(k\pi \frac{x - a}{b - a}\right) dx, \quad (4.13)$$

$\mathcal{Z}_k(t_{m+1})$  the Fourier-cosine coefficients of function  $z(t_{m+1}, x)$  in (3.9b), i.e.,

$$\mathcal{Z}_k(t_{m+1}) = \frac{2}{b-a} \int_a^b z(t_{m+1}, x) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \quad (4.14)$$

and  $\mathcal{F}_k(t_{m+1})$  the Fourier-cosine coefficients of driver  $f(t_{m+1}, y(t_{m+1}, x), z(t_{m+1}, x))$ , i.e.,

$$\mathcal{F}_k(t_{m+1}) = \frac{2}{b-a} \int_a^b f(t_{m+1}, y(t_{m+1}, x), z(t_{m+1}, x)) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \quad (4.15)$$

In Sections 4.2 and 4.3, we will assume that the above coefficients are given. In Section 4.4 the algorithm to recover these coefficients recursively, backwards in time, will be discussed.

## 4.2 COS approximation of function $z(t_m, x)$

For the computation of  $z(t_m, x)$  in (3.9b), we need to compute three expectations,  $\mathbb{E}_m^x[Z_{m+1}^\Delta]$ ,  $\mathbb{E}_m^x[Y_{m+1}^\Delta \Delta\omega_m]$ , and  $\mathbb{E}_m^x[f(t_{m+1}, Y_{m+1}^\Delta, Z_{m+1}^\Delta) \Delta\omega_m]$ . With the help of COS formulas we can derive the following approximations for these expectations:

$$\mathbb{E}_m^x[Z_{m+1}^\Delta] \approx \sum_{k=0}^{N-1} \mathcal{Z}_k(t_{m+1}) \Re\left(\phi\left(\frac{k\pi}{b-a}\right) e^{ik\pi \frac{x-a}{b-a}}\right), \quad (4.16a)$$

$$\mathbb{E}_m^x[Y_{m+1}^\Delta \Delta\omega_m] \approx \sum_{k=0}^{N-1} \mathcal{Y}_k(t_{m+1}) \sigma \Delta t \Re\left(\frac{ik\pi}{b-a} \phi\left(\frac{k\pi}{b-a}\right) e^{ik\pi \frac{x-a}{b-a}}\right), \quad (4.16b)$$

$$\mathbb{E}_m^x[f(t_{m+1}, Y_{m+1}^\Delta, Z_{m+1}^\Delta) \Delta\omega_m] \approx \sum_{k=0}^{N-1} \mathcal{F}_k(t_{m+1}) \sigma \Delta t \Re\left(\frac{ik\pi}{b-a} \phi\left(\frac{k\pi}{b-a}\right) e^{ik\pi \frac{x-a}{b-a}}\right). \quad (4.16c)$$

We then find as COS approximation

$$\begin{aligned} z(t_m, x) \approx & -\frac{1-\theta_2}{\theta_2} \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{Z}_k(t_{m+1}) \Phi_k(x) + \frac{1}{\Delta t \theta_2} \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{Y}_k(t_{m+1}) \sigma \Delta t \frac{d\Phi_k(x)}{dx} \\ & + \frac{1-\theta_2}{\theta_2} \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{F}_k(t_{m+1}) \sigma \Delta t \frac{d\Phi_k(x)}{dx}, \end{aligned} \quad (4.17)$$

with  $\Phi_k$  as defined in (4.8).

## 4.3 COS approximation of function $y(t_m, x)$

For the computation of function  $y(t_m, x)$  in equation (3.9c) there are two explicit parts,  $\mathbb{E}_m^x[Y_{m+1}^\Delta]$  and  $\mathbb{E}_m^x[f(t_{m+1}, Y_{m+1}^\Delta, Z_{m+1}^\Delta)]$ , that are approximated by the following COS formulas

$$\mathbb{E}_m^x[Y_{m+1}^\Delta] \approx \sum_{k=0}^{N-1} \mathcal{Y}_k(t_{m+1}) \Re\left(\phi\left(\frac{k\pi}{b-a}\right) e^{ik\pi \frac{x-a}{b-a}}\right), \quad (4.18a)$$

$$\mathbb{E}_m^x[f(t_{m+1}, Y_{m+1}^\Delta, Z_{m+1}^\Delta)] \approx \sum_{k=0}^{N-1} \mathcal{F}_k(t_{m+1}) \Re\left(\phi\left(\frac{k\pi}{b-a}\right) e^{ik\pi \frac{x-a}{b-a}}\right). \quad (4.18b)$$

Besides, when  $\theta_1 > 0$ , we also have an implicit part, for which we define

$$h(t_m, x) := \mathbb{E}_m^x[Y_{m+1}^\Delta] + \Delta t (1 - \theta_1) \mathbb{E}_m^x[f(t_{m+1}, Y_{m+1}^\Delta, Z_{m+1}^\Delta)]. \quad (4.19)$$



The function  $h(t_m, x)$  can be approximated by

$$h(t_m, x) \approx \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{Y}_k(t_{m+1}) \Phi_k(x) + \Delta t (1 - \theta_1) \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{F}_k(t_{m+1}) \Phi_k(x), \quad (4.20)$$

with  $\Phi_k$  from (4.8). Now we can write

$$y(t_m, x) = \Delta t \theta_1 f(t_m, y(t_m, x), z(t_m, x)) + h(t_m, x). \quad (4.21)$$

In order to determine function  $y(t_m, x)$  in equation (4.21), we will perform *P Picard iterations*, starting with an initial guess,  $y^0(t_m, x) := \mathbb{E}_m^x[Y_{m+1}^\Delta]$  (see equation (4.18a)). The Picard method can be written as

$$\begin{aligned} y^1(t_m, x) &= \Delta t \theta_1 f(t_m, y^0(t_m, x), z(t_m, x)) + h(t_m, x), \\ y^2(t_m, x) &= \Delta t \theta_1 f(t_m, y^1(t_m, x), z(t_m, x)) + h(t_m, x), \\ &\dots \\ y^P(t_m, x) &= \Delta t \theta_1 f(t_m, y^{P-1}(t_m, x), z(t_m, x)) + h(t_m, x). \end{aligned}$$

The convergence properties of the Picard iterations to the “true” values  $y(t_m, x)$  is discussed in Section 4.5.

#### 4.4 Recovery of coefficients and algorithm

The computation of functions  $z(t_m, x)$  and  $y(t_m, x)$  at time-point  $t_m$  requires the Fourier-cosine coefficients  $\mathcal{Z}_k(t_{m+1})$ ,  $\mathcal{Y}_k(t_{m+1})$ , and  $\mathcal{F}_k(t_{m+1})$  of time-point  $t_{m+1}$ . In the next time step of the BCOS method we wish to compute functions  $z(t_{m-1}, x)$  and  $y(t_{m-1}, x)$  at time-point  $t_{m-1}$ , for which we need the Fourier-cosine coefficients of time-point  $t_m$ . The coefficients can be computed recursively backwards in time, as we explain in this section.

Firstly, the computation of the coefficients

$$\mathcal{Z}_k(t_m) = \frac{2}{b-a} \int_a^b z(t_m, x) \cos\left(k\pi \frac{x-a}{b-a}\right) dx \quad (4.23)$$

can be divided into three parts, similar as equation (4.16). We then use the following approximations

$$\frac{2}{b-a} \int_a^b \mathbb{E}_m^x[Z_{m+1}^\Delta] \cos\left(k\pi \frac{x-a}{b-a}\right) dx \approx \Re\left(\sum_{j=0}^{N-1} \mathcal{Z}_j(t_{m+1}) \phi\left(\frac{j\pi}{b-a}\right) \mathcal{M}_{k,j}\right), \quad (4.24a)$$

$$\frac{2}{b-a} \int_a^b \mathbb{E}_m^x[Y_{m+1}^\Delta \Delta\omega_m] \cos\left(k\pi \frac{x-a}{b-a}\right) dx \approx \Re\left(\sum_{j=0}^{N-1} \frac{ij\pi}{b-a} \sigma \Delta t \mathcal{Y}_j(t_{m+1}) \phi\left(\frac{j\pi}{b-a}\right) \mathcal{M}_{k,j}\right), \quad (4.24b)$$

$$\frac{2}{b-a} \int_a^b \mathbb{E}_m^x[f(t_{m+1}, Y_{m+1}^\Delta, Z_{m+1}^\Delta) \Delta\omega_m] \cos\left(k\pi \frac{x-a}{b-a}\right) dx \approx \Re\left(\sum_{j=0}^{N-1} \frac{ij\pi}{b-a} \sigma \Delta t \mathcal{F}_j(t_{m+1}) \phi\left(\frac{j\pi}{b-a}\right) \mathcal{M}_{k,j}\right), \quad (4.24c)$$

with matrix elements

$$\mathcal{M}_{k,j} := \frac{2}{b-a} \int_a^b e^{ij\pi \frac{x-a}{b-a}} \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \quad (4.25)$$

These approximations can be found by inserting COS formulas (4.16) in (4.24). Note that the approximation signs “ $\approx$ ” are due to the errors of the COS formulas, i.e., truncation of the integration range to a finite interval  $[a, b]$  and truncation of the infinite sums to a finite number of terms  $N$ . The coefficients  $\mathcal{Z}_k(t_m)$  are then computed as follows

$$\begin{aligned} \mathcal{Z}_k(t_m) \approx & \Re \left( \sum_{j=0}^{N-1} \left[ -\frac{1-\theta_2}{\theta_2} \mathcal{Z}_j(t_{m+1}) \phi \left( \frac{j\pi}{b-a} \right) + \frac{1}{\Delta t \theta_2} \frac{ij\pi}{b-a} \sigma \Delta t \mathcal{Y}_j(t_{m+1}) \phi \left( \frac{j\pi}{b-a} \right) \right. \right. \\ & \left. \left. + \frac{1-\theta_2}{\theta_2} \frac{ij\pi}{b-a} \sigma \Delta t \mathcal{F}_j(t_{m+1}) \phi \left( \frac{j\pi}{b-a} \right) \right] \mathcal{M}_{k,j} \right). \end{aligned} \quad (4.26)$$

Secondly, the coefficients  $\mathcal{H}_k(t_m)$  of function  $h(t_m, x)$  in (4.20) are computed by

$$\begin{aligned} \mathcal{H}_k(t_m) &= \frac{2}{b-a} \int_a^b h(t_m, x) \cos \left( k\pi \frac{x-a}{b-a} \right) dx \\ &\approx \Re \left( \sum_{j=0}^{N-1} \left[ \mathcal{Y}_j(t_{m+1}) \phi \left( \frac{j\pi}{b-a} \right) + \Delta t (1-\theta_1) \mathcal{F}_j(t_{m+1}) \phi \left( \frac{j\pi}{b-a} \right) \right] \mathcal{M}_{k,j} \right). \end{aligned} \quad (4.27)$$

The Fourier-cosine coefficients  $\mathcal{Z}_k(t_m)$  in (4.26) and  $\mathcal{H}_k(t_m)$  in (4.27), for  $k = 0, 1, \dots, N-1$ , can thus be computed by one matrix-vector multiplication. These matrix-vector multiplications  $\mathcal{M}\mathbf{u}$  can be done efficiently with the use of an FFT algorithm, see [FO09]. With this the computational complexity is reduced from order  $O(N^2)$  to  $O(N \log N)$ , with  $N$  the number of terms in the summations.

**Remark 4.1.** For general drift  $\mu(t, x)$  and volatility  $\sigma(t, x)$  in (4.1) we need to compute the following integrals to recover the Fourier-cosine coefficients

$$\begin{aligned} & \frac{2}{b-a} \int_a^b \varphi \left( \frac{j\pi}{b-a} \middle| x \right) e^{ij\pi \frac{x-a}{b-a}} \cos \left( k\pi \frac{x-a}{b-a} \right) dx, \\ & \varphi(u|x) = \exp \left( x + u\mu(t_m, x)\Delta t - \frac{1}{2}u^2\sigma^2(t_m, x)\Delta t \right). \end{aligned} \quad (4.28)$$

As the integration kernel is smooth, we can approximate the integrals efficiently by, for example, a Clenshaw-Curtis quadrature rule ([Gen72]).

At last, the coefficients  $\mathcal{F}_k^{P-1}(t_m)$  of function  $f(t_m, y^{P-1}(t_m, x), z(t_m, x))$  are given by

$$\mathcal{F}_k^{P-1}(t_m) := \frac{2}{b-a} \int_a^b f(t_m, y^{P-1}(t_m, x), z(t_m, x)) \cos \left( k\pi \frac{x-a}{b-a} \right) dx. \quad (4.29)$$

They are approximated by a *discrete Fourier-cosine transform (DCT)*. For this we need to compute the integrand  $f(t_m, y^{P-1}(t_m, x), z(t_m, x))$  on an *equidistant x-grid* with  $Q \geq N$  grid points, as explained in Appendix B. With a converging Picard method, we have  $\mathcal{F}_k(t_m) \approx \mathcal{F}_k^{P-1}(t_m)$  for sufficiently many iterations  $P$ . Then,

$$\mathcal{Y}_k(t_m) \approx \Delta t \theta_1 \mathcal{F}_k^{P-1}(t_m) + \mathcal{H}_k(t_m). \quad (4.30)$$

With the aforementioned formulas we approximate the Fourier-cosine coefficients  $\mathcal{Z}_k(t_m)$ ,  $\mathcal{Y}_k(t_m)$ , and  $\mathcal{F}_k(t_m)$  by using the coefficients of time-point  $t_{m+1}$ . Starting with the coefficients at the terminal time we can solve them recursively, backwards in time. The evolution of the extra

error introduced by approximation of the coefficients has been discussed in [FO09]. The final approximations of the functions  $y(t_m, x)$  and  $z(t_m, x)$  by the BCOS method are denoted by  $\hat{y}(t_m, x)$  and  $\hat{z}(t_m, x)$ . The overall algorithm to solve the BSDE (3.9) backwards in time can be summarized as:

BCOS method:

*Initial step:* Compute, or approximate, the terminal coefficients  $\mathcal{Y}_k(t_M)$ ,  $\mathcal{Z}_k(t_M)$ , and  $\mathcal{F}_k(t_M)$ .

*Loop:* For  $m = M - 1$  to  $m = 0$ : Compute the functions  $\hat{z}(t_m, x)$ ,  $f(t_m, \hat{y}(t_m, x), \hat{z}(t_m, x))$ , and  $\hat{y}(t_m, x)$  and determine the corresponding Fourier-cosine coefficients  $\mathcal{Z}_k(t_m)$ ,  $\mathcal{F}_k(t_m)$ , and  $\mathcal{Y}_k(t_m)$ , as described in Sections 4.2, 4.3, and 4.4.

*Terminal step:* Compute  $\hat{y}(t_0, x_0)$ .

#### 4.5 Errors and computational complexity

In the BCOS method when solving BSDEs several approximation errors are encountered. In the first place there are discretization errors, due to the discrete-time approximation of the stochastic processes. Moreover, errors are introduced by the COS formulas and the Picard method. These error components and the computational complexity are discussed in this section.

**Discretization error of the FSDE** If the functions  $\sigma(t, x)$  and  $\mu(t, x)$  are Lipschitz continuous w.r.t.  $x$ , satisfy a linear growth condition in  $x$ , and are  $1/2$ -Hölder continuous with respect to  $t$ , then the following standard estimate holds ([KP92]). There exists a constant  $C > 0$ , independent of  $\Delta$ , such that

$$\max_{0 \leq m \leq M} \mathbb{E}_0 [|X_m - X_m^\Delta|] \leq C\sqrt{\Delta t}. \quad (4.31)$$

**Discretization error of the BSDE** We define the local theta-discretization errors in equations (3.5) and (3.6) by

$$R_m^y := \int_{t_m}^{t_{m+1}} \mathbb{E}_m [f(s, Y_s, Z_s)] ds - \Delta t \theta_1 f(t_m, Y_m, Z_m) - \Delta t (1 - \theta_1) \mathbb{E}_m [f(t_{m+1}, Y_{m+1}, Z_{m+1})], \quad (4.32a)$$

$$\begin{aligned} R_m^z &:= \int_{t_m}^{t_{m+1}} \mathbb{E}_m [f(s, Y_s, Z_s) (\omega_s - \omega_{t_m})] ds - \Delta t (1 - \theta_2) \mathbb{E}_m [f(t_{m+1}, Y_{m+1}, Z_{m+1}) \Delta \omega_m] \\ &\quad - \int_{t_m}^{t_{m+1}} \mathbb{E}_m [Z_s] ds + \Delta t \theta_2 Z_m + \Delta t (1 - \theta_2) \mathbb{E}_m [Z_{m+1}]. \end{aligned} \quad (4.32b)$$

The order of these errors depends on the smoothness of the integrands with respect to time  $s$ . If functions  $f$  and  $g$  are sufficiently smooth and bounded, with bounded derivatives, then (see [ZWP09])

$$R_m^y = O((\Delta t)^2) \quad \text{and} \quad R_m^z = O((\Delta t)^2), \quad \text{for } \theta_i \in [0, 1], \quad (4.33)$$

$$R_m^y = O((\Delta t)^3), \quad \text{for } \theta_1 = 1/2, \quad \text{and} \quad R_m^z = O((\Delta t)^3), \quad \text{for } \theta_2 = 1/2. \quad (4.34)$$

The global errors due to the theta-time discretization in (3.7c) and (3.7b) are denoted by

$$\epsilon_m^y := Y_m - Y_m^\Delta, \quad \epsilon_m^z := Z_m - Z_m^\Delta, \quad \epsilon_m^f := f(t_m, Y_m, Z_m) - f(t_m, Y_m^\Delta, Z_m^\Delta). \quad (4.35)$$

For the  $y$ -component we have

$$\epsilon_m^y = \mathbb{E}_m [\epsilon_{m+1}^y] + \Delta t \theta_1 \epsilon_m^f + \Delta t (1 - \theta_1) \mathbb{E}_m [\epsilon_{m+1}^f] + R_m^y. \quad (4.36)$$

We can assume, without loss of generality, that  $f(t, y, z)$  is Lipschitz in  $y$  and  $z$  on our computational domain with Lipschitz constant  $L^{Lipz}$ . Then the global error can be bounded by

$$|\epsilon_m^y| \leq \frac{1 + (1 - \theta_1)\Delta t L^{Lipz}}{1 - \theta_1 \Delta t L^{Lipz}} \mathbb{E}_m[|\epsilon_{m+1}^y|] + \frac{\theta_1 \Delta t L^{Lipz}}{1 - \theta_1 \Delta t L^{Lipz}} |\epsilon_m^z| + \frac{1}{1 - \theta_1 \Delta t L^{Lipz}} |R_m^y|. \quad (4.37)$$

For the  $z$ -component we have

$$\epsilon_m^z = \frac{1}{\Delta t} \theta_2^{-1} \mathbb{E}_m[\epsilon_{m+1}^y \Delta \omega_m] + \theta_2^{-1} (1 - \theta_2) \mathbb{E}_m[\epsilon_{m+1}^f \Delta \omega_m] - \theta_2^{-1} (1 - \theta_2) \mathbb{E}_m[\epsilon_{m+1}^z] + \frac{1}{\Delta t} \theta_2^{-1} R_m^z. \quad (4.38)$$

Hölders inequality gives

$$|\mathbb{E}_m[\epsilon_{m+1}^y \Delta \omega_m]| \leq \mathbb{E}_m[|\epsilon_{m+1}^y|^2]^{1/2} \sqrt{\Delta t}, \quad (4.39a)$$

$$|\mathbb{E}_m[\epsilon_{m+1}^f \Delta \omega_m]| \leq \mathbb{E}_m[|\epsilon_{m+1}^f|^2]^{1/2} \sqrt{\Delta t}, \quad (4.39b)$$

so that

$$\begin{aligned} |\epsilon_m^z| &\leq \frac{1}{\sqrt{\Delta t}} \theta_2^{-1} \mathbb{E}_m[|\epsilon_{m+1}^y|^2]^{1/2} + \theta_2^{-1} (1 - \theta_2) \mathbb{E}_m[|\epsilon_{m+1}^f|^2]^{1/2} \sqrt{\Delta t} \\ &\quad - \theta_2^{-1} (1 - \theta_2) \mathbb{E}_m[|\epsilon_{m+1}^z|] + \frac{1}{\Delta t} \theta_2^{-1} |R_m^z|. \end{aligned} \quad (4.40)$$

The error depends on the choice of  $\theta_1$  and  $\theta_2$  and on the smoothness of driver function  $f$  and terminal condition  $g$ . For example, for a sufficiently smooth function  $f(y)$ , only depending on  $y$ , and  $g(y)$  sufficiently smooth, we find by induction

$$\mathbb{E}_0[|\epsilon_m^y|] = \begin{cases} O(\Delta t), & \text{for } \theta_1 \neq 1/2, \\ O((\Delta t)^2), & \text{for } \theta_1 = 1/2. \end{cases} \quad (4.41)$$

The authors in [ZWP09] obtain the convergence rate  $\mathbb{E}_0[|\epsilon_m^z|] = O(\Delta t)$ . In our numerical examples we find second order convergence for  $\theta_1 = \theta_2 = \frac{1}{2}$ .

Convergence of  $(Y^\Delta, Z^\Delta)$  to  $(Y, Z)$  is discussed in [BT04, Zha04, LGW06, GLW05, BE08] for the special case  $\theta_1 = \theta_2 = 1$ . Under certain conditions for functions  $f$  and  $g$ , error convergence of order  $O((\Delta t)^{1/2})$  in the  $L^2$ -sense was found.

**Error COS formulas** In Section 3 we encountered deterministic functions  $y$  and  $z$  such that

$$y(t_m, X_m^\Delta) = Y_m^\Delta, \quad z(t_m, X_m^\Delta) = Z_m^\Delta. \quad (4.42)$$

These functions are approximated by COS formulas and the corresponding Fourier coefficients are recovered backwards in time, resulting in the approximations  $\hat{y}(t_m, x)$  and  $\hat{z}(t_m, x)$ . Fourier series expansions and their convergence properties have been discussed in [Boy01]. The error of the COS method was been discussed extensively in [FO08, FO09]. The error component  $I - \hat{I}$  (equations (4.10)) converges exponentially in the number of terms in the series expansions for smooth density functions and a sufficiently wide integration interval. A density function with a discontinuity in one of its derivatives gives rise to an algebraic convergence in  $N$ .

**Convergence of Picard iterations** With  $P$  Picard iterations we find the fixed-point  $y$  of the equation

$$y = \Delta t \theta_1 f(t_m, y, z(t_m, x)) + h(t_m, x). \quad (4.43)$$

We can assume, without loss of generality, that the driver function  $f$  is bounded, as  $y$  and  $z$  are bounded on our computational domain, and Lipschitz in  $y$  with Lipschitz constant  $L^{Lipz}$ . For  $\Delta t$  small enough, i.e.  $L^{Lipz} \Delta t \theta_1 < 1$ , a unique fixed-point exists, and the Picard iterations converge towards that point for any initial guess. The fixed-point technique converges to the true solution at the geometric rate  $\Delta t \theta_1 L^{Lipz}$ , which depends on the Lipschitz condition of the driver function.

**Computational complexity** The computation time of the BCOS method is linear in the number of timesteps  $M$ . For each discrete time-point  $t_m$  we perform the following operations:

- Computation of  $\hat{z}(t_m, x)$  and  $\hat{h}(t_m, x)$  on an  $x$ -grid, in  $O(NQ)$  operations.
- Initialization of the Picard method: Computation of  $\hat{y}^0(t_m, x)$  on an  $x$ -grid, in  $O(NQ)$  operations.
- Computation of  $\hat{y}^P(t_m, x)$  on an  $x$ -grid by  $P$  Picard iterations, in  $O(PQ)$  operations.
- Computation of  $Z_k(t_m)$  and  $\mathcal{H}_k(t_m)$  by the FFT algorithm, in  $O(N \log N)$  operations.
- Computation of  $\mathcal{F}_k(t_m) \approx \mathcal{F}_k^{P-1}(t_m)$  by DCT (see Appendix B), in  $O(Q \log Q)$  operations.
- Computation of  $\mathcal{Y}_k(t_m) \approx \mathcal{Y}_k^P(t_m)$ , in  $O(N)$  operations.

For the approximation of the coefficients  $\mathcal{F}_k^{P-1}(t_m)$  with a DCT we first need to compute  $\hat{z}(t_m, x)$ ,  $\hat{h}(t_m, x)$ , and  $\hat{y}^0(t_m, x)$  on an  $x$ -grid with  $Q \geq N$  equidistant points, which is of order  $O(NQ)$ . This is the most time-consuming part of the algorithm. However, solving these functions is uncoupled and can be computed in parallel. In total, the complexity of the BCOS algorithm is  $O(M(NQ + PQ + N \log N + Q \log Q + N))$ .

## 5 Numerical experiments

In this section we discuss several numerical experiments. MATLAB 7.11.0 is used for the computations, with an Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz and 7.7 GB RAM. To test the general theta-method we distinguish between four discretization schemes:

$$\begin{array}{ll} \text{Scheme A:} & \theta_1 = 0, \quad \theta_2 = 1, \\ \text{Scheme B:} & \theta_1 = 0.5, \quad \theta_2 = 1, \\ \text{Scheme C:} & \theta_1 = 1, \quad \theta_2 = 1, \\ \text{Scheme D:} & \theta_1 = 0.5, \quad \theta_2 = 0.5. \end{array}$$

For all four schemes,  $z(t_m, x)$  can be solved explicitly and  $y(t_m, x)$  is solved explicitly for scheme A and implicitly with  $P = 5$  Picard iterations for the other schemes.

Similar as in [FO08], we prescribe a computational domain  $[a, b]$

$$[a, b] = [x_0 + \kappa_1 - L\sqrt{\kappa_2}, x_0 + \kappa_1 + L\sqrt{\kappa_2}], \quad (5.1)$$

with cumulants  $\kappa_1 = \mu T$  and  $\kappa_2 = \sigma^2 T$ , and  $L = 10$ . Furthermore we set the number of terms in the Fourier-cosine series expansions equal to  $N = 2^9$  and  $Q = N$  grid points for the DCT. For these values the BCOS method has converged in  $N$  to machine precision.

### 5.1 Example 1

The first example is taken from [WLZ09]. The underlying process is the Wiener process, i.e.  $X_t = \omega_t$ . The BSDE reads

$$dY_t = -f(t, Y_t, Z_t)dt + Z_t d\omega_t, \quad (5.2a)$$

$$f(t, Y_t, Z_t) = -Y_t(1 - Y_t)(\gamma - Y_t), \quad (5.2b)$$

$$Y_T = g(X_T) = 1 / (1 + \exp(-X_T - T/4)). \quad (5.2c)$$

The exact solution is given by

$$(Y_t, Z_t) = \left( 1 / (1 + \exp(-X_t - t/4)), \exp(-X_t - t/4) / (1 + \exp(-X_t - t/4))^2 \right), \quad (5.3)$$

which gives  $(Y_0, Z_0) = (1/2, 1/4)$ . Note that driver  $f(\cdot)$  is nonlinear in  $y$  and does not depend on  $z$ . We take terminal time  $T = 1$  and  $\gamma = 3/4$ . For the results of the BCOS method, we refer to Figure 5.1. We observe that the approximated value  $\hat{y}(t_0, x_0)$  converges with  $O(\Delta t)$  for schemes A and C and with  $O((\Delta t)^2)$  for schemes B and D. The approximated value  $\hat{z}(t_0, x_0)$  converges with  $O((\Delta t)^2)$  for scheme D and with  $O(\Delta t)$  for the other three schemes.

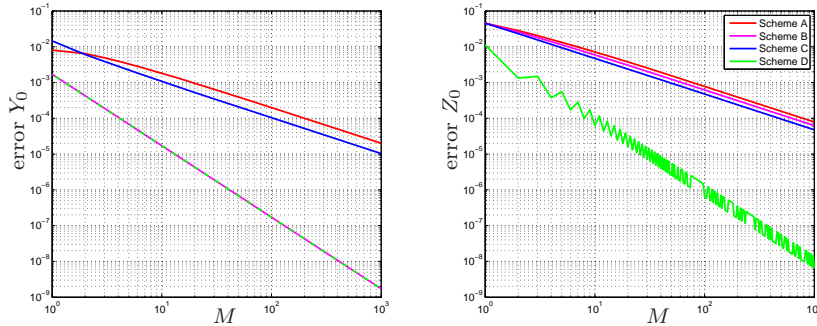


Figure 5.1: Results example 1 ( $N = 2^9$ ), left: error  $\hat{y}(t_0, x_0)$ , right: error  $\hat{z}(t_0, x_0)$ .

Table 5.1 shows the CPU times, for scheme  $D$ , for different values of  $M$  and  $N$ . Each test required less than one second. Computation of the functions  $\hat{z}(t_m, x)$ ,  $\hat{h}(t_m, x)$ , and  $\hat{y}^0(t_m, x)$  on an  $x$ -grid is the most time consuming part of the algorithm. The computation time is linear in the number of timesteps  $M$  and  $O(N \log N)$  in the number of terms in the Fourier-cosine series expansions.

$M$	4	8	16	32	64	128	256	512
$N = 2^9$	0.0301	0.0304	0.0412	0.0639	0.1071	0.1966	0.3736	0.7292
$N$	$2^6$	$2^7$	$2^8$	$2^9$				
$M = 256$	0.0940	0.1109	0.1552	0.3736				

Table 5.1: CPU time (s).

## 5.2 Example 2, Black-Scholes call option

In this example we compute the price  $v(t, S_t)$  of a call option by a BSDE where the underlying asset follows a geometric Brownian motion,

$$dS_t = \mu S_t dt + \sigma S_t d\omega_t. \quad (5.4)$$

The exact solution is given by the Black-Scholes price, which can be found analytically by solving the Black-Scholes equation, with the help of the Feynman-Kac theorem for linear PDEs ([Shr04]). For the derivation of the Black-Scholes PDE we set up a self-financing portfolio  $Y_t$  with  $a_t$  assets and bonds with risk-free return rate  $r$ . Markets are assumed to be complete in this model, there are no trading restrictions, and the option can be exactly replicated by the hedging portfolio, that is  $Y_T = \max(S_T - K, 0)$ . Then, the option value at initial time should be equal to the initial value of the portfolio. The portfolio evolves according to the SDE

$$dY_t = r(Y_t - a_t S_t) dt + a_t dS_t = \left( rY_t + \frac{\mu - r}{\sigma} \sigma a_t S_t \right) dt + \sigma a_t S_t d\omega_t. \quad (5.5)$$

If we set  $Z_t = \sigma a_t S_t$ , then  $(Y, Z)$  solves the BSDE,

$$dY_t = -f(t, Y_t, Z_t) dt + Z_t d\omega_t, \quad (5.6a)$$

$$f(t, Y_t, Z_t) = -rY_t - \frac{\mu - r}{\sigma} Z_t, \quad (5.6b)$$

$$Y_T = \max(S_T - K, 0). \quad (5.6c)$$

$Y_t$  corresponds to the value of the portfolio and  $Z_t$  is related to the hedging strategy. In this case, the driver function  $f(\cdot)$  is Lipschitz continuous and linear with respect to  $y$  and  $z$ . The option value is given by  $v(t, S_t) = Y_t$  and  $\sigma S_t v_S(t, S_t) = Z_t$ . For the tests, we use the following parameter values

$$S_0 = 100, K = 100, r = 0.1, \mu = 0.2, \sigma = 0.25, T = 0.1, \quad (5.7)$$

with the exact solutions  $Y_0 = v(t_0, S_0) = 3.65997$  and  $Z_0 = \sigma S_0 v_S(t_0, S_0) = 14.14823$ . For the numerical approximation, we switch to log-asset domain  $X_t = \log S_t$ , with

$$dX_t = (\mu - \frac{1}{2}\sigma^2)dt + \sigma d\omega_t. \quad (5.8)$$

Results of the BCOS method for all four schemes are presented in Figure 5.2. The approximated values  $\hat{y}(t_0, x_0)$  and  $\hat{z}(t_0, x_0)$  both converge with  $O(\Delta t)$  for schemes A, B, and C and with  $O((\Delta t)^2)$  for scheme D, as expected.

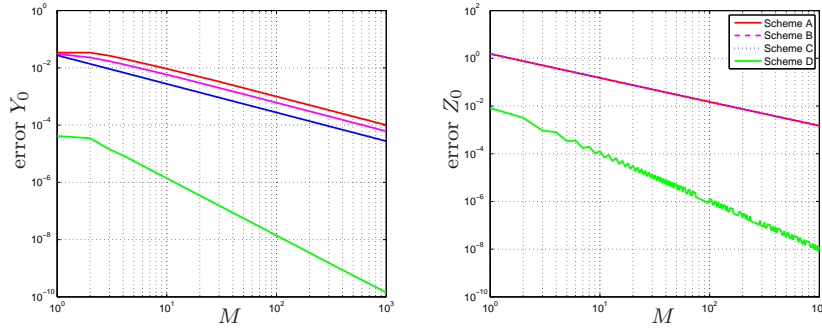


Figure 5.2: Results example 2 ( $N = 2^9$ ), left: error  $\hat{y}(t_0, x_0)$ , right: error  $\hat{z}(t_0, x_0)$ .

We would like to emphasize that solving the BSDE is done under the historical, real-world  $\mathbb{P}$ -measure. However, the exact Black-Scholes solution does not depend on  $\mu$ . In Figure 5.3 we see results for different values of drift  $\mu$ . The convergence rates in  $M$  are the same, but a higher value of  $\mu$  gives a larger error for the same number of timesteps  $M$ . This is due to the Lipschitz constant  $L^{Lipz} = \max(\frac{\mu - r}{\sigma}, r)$ , which is increasing in  $\mu$ .

### 5.3 Example 3, bid-ask spread for interest rates

For the pricing problem in the previous section, the driver  $f(\cdot)$  was linear and the initial option value  $Y_0$  reduced to the expectation of the discounted option payoff under the risk-neutral measure, i.e., the Black-Scholes price. We now consider a model introduced by Bergman [Ber95] with different interest rates for lending and borrowing a bond. This market imperfection results in a driver function which is nonlinear.

Suppose that an agent can invest in bonds at risk-free rate  $r$  and borrow money at rate  $R > r$ . The amount invested at time  $t$  is equal to  $\max(Y_t - a_t S_t, 0)$  and the amount borrowed is  $\min(Y_t - a_t S_t, 0)$ . Then, the replication portfolio follows the following dynamics

$$\begin{aligned} dY_t &= r \max(Y_t - a_t S_t, 0)dt + R \min(Y_t - a_t S_t, 0) + a_t dS_t \\ &= \left( rY_t + \frac{\mu - r}{\sigma} \sigma a_t S_t + (R - r) \min(Y_t - a_t S_t, 0) \right) + \sigma a_t S_t d\omega_t. \end{aligned} \quad (5.9)$$

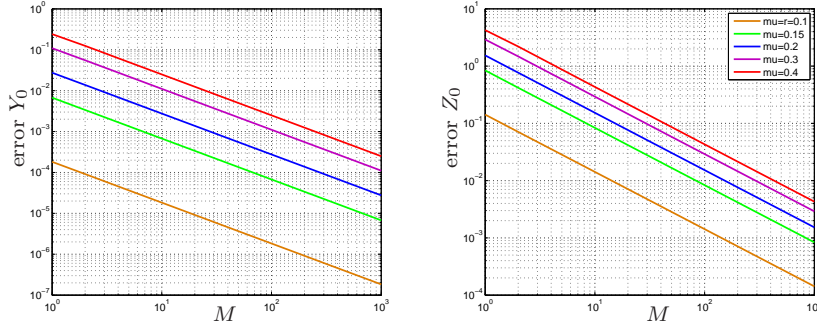


Figure 5.3: Results example 2 for different values of  $\mu$  (Scheme C).

Again, with  $Z_t = \sigma a_t S_t$ , we obtain the BSDE

$$dY_t = -f^{r,R}(t, Y_t, Z_t)dt + Z_t d\omega_t, \quad (5.10a)$$

$$f^{r,R}(t, Y_t, Z_t) := -rY_t - \frac{\mu - r}{\sigma} Z_t - (R - r) \min(Y_t - Z_t/\sigma, 0), \quad (5.10b)$$

$$Y_T = g(S_T). \quad (5.10c)$$

The driver is now a non-linear Lipschitz function and the dependence on the different return rates is emphasized by the superscripts. The Lipschitz constant of the driver function is given by  $L^{Lipz} = (R - r)/\sigma$ . The corresponding semilinear PDE for  $v(t, S_t) = Y_t$  reads

$$\begin{aligned} \frac{\partial v}{\partial t}(t, S) + \mathcal{L}v(t, S) + f^{r,R}(t, v(t, S), \sigma S D_S v(t, S)) &= 0, \quad (t, S) \in [0, T] \times \mathbb{R}, \\ v(T, S) &= g(S), \quad S \in \mathbb{R}. \end{aligned} \quad (5.11)$$

This semilinear PDE is solved in [FL07] by using a PDE discretization method. According to [Gob10] the following lower bound is valid

$$Y_t^{r,R} \geq \max(Y_t^{r_1, r_1}, Y_t^{R, R}), \quad \forall t \in [0, T], \quad (5.12)$$

where  $Y_t^{r_1, r_1}$  denotes the solution with driver  $f^{r_1, r_1}$ . In other words, the option price with different interest rates is larger than the price with fixed interest rates. For a call option it follows that  $Y_t^{r,R} = Y_t^{R,R}$ , in other words, the exact solution is given by the Black-Scholes call price with interest rate  $R$  and the BSDE becomes linear, as in Section 5.2. The authors in [BS12] examine a combination of a long call with strike  $K_1 = 95$  and two short calls with strike  $K_2 = 105$ , with payoff function

$$g(S) = (S - K_1)^+ - 2(S - K_2)^+, \quad (5.13)$$

and parameter values

$$S_0 = 100, r = 0.01, \mu = 0.05, \sigma = 0.2, T = 0.25. \quad (5.14)$$

An analytic solution is not available then. We perform tests for borrow rates  $R = 0.06$  and  $R = 3.01$  with corresponding Lipschitz constants  $L^{Lipz} = 0.25$  and  $L^{Lipz} = 15$ , respectively. The authors in [BS12] note that an interest rate of 301% is not relevant for financial applications, but they propose it in order to test their algorithm under an extreme situation, as we will do.

For  $R = 0.06$ , the reference value obtained in [BS12] is  $Y_0^{r,R} = 2.96$ . Error results of the BCOS method are shown in Figure 5.4. For this we used reference values  $Y_0^{r,R} = 2.9584544$  and  $Z_0^{r,R} =$



0.55319, obtained with  $M = 10000$ . The approximated value  $\hat{y}(t_0, x_0)$  converges with  $O(\Delta t)$  for schemes A, B, and C and approximately with  $O((\Delta t)^{3/2})$  for scheme D. The approximated value  $\hat{z}(t_0, x_0)$  converges with  $O(\Delta t)$  for schemes B and C and with a higher convergence rate for schemes A and D.

For  $R = 3.01$ , the authors of [BS12] find  $Y_0^{r,R} = 6.4$ . For large  $M$  the BCOS method gives the reference values  $Y_0^{r,R} = 6.3748$  and  $Z_0^{r,R} = -4.690$ . The results of our numerical approximations are shown in Figure 5.5. Convergence is slower and the errors are larger compared to  $R = 0.06$ , because the driver function has a larger Lipschitz constant. The convergence rates are not clearly readable, however for large  $M$  the orders seem to correspond to the case with  $R = 0.06$ .

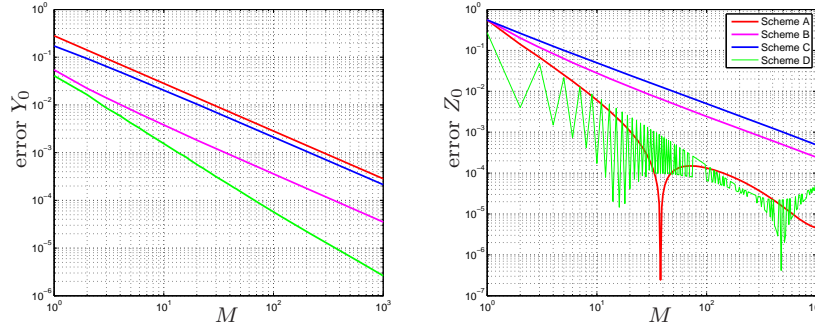


Figure 5.4: Results example 3,  $R = 0.06$  ( $N = 2^9$ ), left: error  $\hat{y}(t_0, x_0)$ , right: error  $\hat{z}(t_0, x_0)$ .

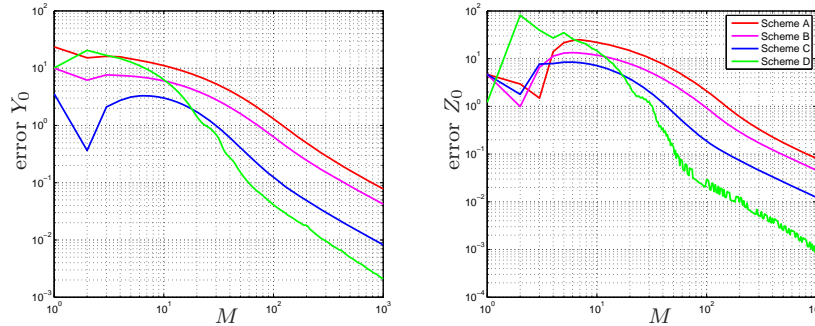


Figure 5.5: Results example 3,  $R = 3.01$  ( $N = 2^9$ ), left: error  $\hat{y}(t_0, x_0)$ , right: error  $\hat{z}(t_0, x_0)$ .

## 6 Exponential utility maximization and utility indifference price

In a financial market with jumps or with constrained hedging strategies it is usually not possible to perform a perfect hedge which attains exactly the option payoff as the final value, there is a so-called replication error. If markets are not complete there are different ways to value options ([EKPQ97]), for example by

- *Super-strategies* are strategies with a positive replication error. The super-replicating option price is the minimal initial investment to find a strategy that always dominates the payoff of the option. ([EKQ95])

- *Risk-minimizing strategies*, where the problem is to find a strategy with minimal variance for the replication error. They were first introduced by Föllmer and Sondermann in [FS86].
- *Utility indifference pricing*, where the utility of the replication error is maximized. The corresponding price makes an agent indifferent in terms of expected utility between selling the option or not selling it. Utility indifference pricing was introduced by Hodges and Neuberger in [HN89].

We focus on utility indifference pricing, which basically consists of solving two utility maximization problems, one with and one without an option liability. In the next section we consider a general utility maximization problem. We employ the model of Morlais in [Mor10], making use of an exponential utility function and jumps in the asset price. The problem can be defined by a BSDE including jumps. We refer to [REK00, Sek06, HIM05, MS05, Pha09] for the setting where asset prices follow only a diffusion process, this model is generalized with jumps in [Bec06, Mor10].

### 6.1 Exponential utility maximization under jump-diffusion with option payoff

Following the notation in [Mor10], the probability space  $(\Omega, \mathbb{F}, \mathbb{P})$  is now equipped with two independent stochastic processes: the standard Brownian motion  $\omega$  and a real-valued Poisson point process defined on  $\Omega \times [0, T] \times E$ , with  $E := \mathbb{R}/\{0\}$ . We denote by  $N(dJ, dt)$  the associated Poisson random measure whose compensator is assumed to be of the form  $\nu(dJ)dt$ , where  $\nu(dJ)$  stands for the Lévy measure, which is positive and satisfies

$$\nu(\{0\}) = 0 \quad \text{and} \quad \int_E (1 \wedge |J|)^2 \nu(dJ) < \infty. \quad (6.1)$$

$N(B, t)$ ,  $B \in \mathbb{R}$ , is the number of jumps with size in set  $B$  which occur before or at time  $t$  and  $\nu(B)$  counts the expected number of jumps in a unit time interval.  $\mathcal{F}$  is the completed filtration generated by both processes  $\omega$  and  $N$ . The so-called compensated Poisson random measure,  $\tilde{N}$ , is given by

$$\tilde{N}(dJ, dt) = N(dJ, dt) - \nu(dJ)dt. \quad (6.2)$$

The asset price is supposed to follow the following jump-diffusion process,

$$dS_t/S_{t-} = b(t)dt + \sigma(t)d\omega_t + \int_E \beta(t, J)\tilde{N}(dJ, dt), \quad (6.3)$$

$S_{t-}$  represents the value of the asset just before a possible jump occurs. The jumps may model the occurrence of, for example, market crashes or default losses. A writer sells a bounded  $\mathcal{F}_T$ -measurable option payoff  $\xi = g(S_T)$  at time  $t = 0$ . He is endowed with some initial capital  $w$  and then invests  $\alpha_t$ ,  $t \in [0, T]$  of his portfolio  $W_t^\alpha$  in assets, where the superscript emphasized the dependence on  $\alpha$ . The remaining part is invested in a risk-free opportunity with zero rate of return, i.e.,  $r = 0$ . The dynamics of his self-financing portfolio read

$$W_0^\alpha = w, \quad dW_t^\alpha = \alpha_t \frac{dS_t}{S_{t-}} = \alpha_t b(t)dt + \alpha_t \sigma(t)d\omega_t + \alpha_t \int_E \beta(t, J)\tilde{N}(dJ, dt). \quad (6.4)$$

At terminal time  $T$  there is an uncertain claim  $\xi$  and the agent is able to reduce the risk by his trading strategy. The attitude of the agent towards possible profits and losses is measured by an exponential utility,

$$\mathfrak{U}(x) = -\exp(-\eta x), \quad \eta > 0. \quad (6.5)$$

The utility function is monotonically increasing and concave;  $\eta$  is the coefficient of absolute risk aversion and represents the degree of risk aversion. A higher value of  $\eta$  corresponds to a higher

level of risk aversion. Figure 6.1 shows the utility function for different values of  $\eta$ . A negative amount of final wealth has a higher weight than a positive amount, in other words, more weight is given to unfavourable losses.  $\eta = 0$  corresponds to risk neutrality and  $\eta = \infty$  to absolute risk aversion.

The agent wants to maximize his expected utility at time  $T$  and his *objective function* now reads

$$V(w) = \max_{\alpha \in \mathcal{A}} \mathbb{E} [\mathfrak{U}(W_T^\alpha - \xi)] = \max_{\alpha \in \mathcal{A}} \mathbb{E} \left[ \mathfrak{U} \left( w + \int_0^T \alpha_t \frac{dS_t}{S_{t-}} - \xi \right) \right], \quad (6.6)$$

where we maximize over the investment opportunities  $\alpha$  in the constraint set  $\mathcal{A}$  with admissible strategies. Possible trading strategies may be restricted, for example a writer may be forced not to hold a negative amount of assets. For the tests in Section 7 we will take  $\mathcal{A} = [\alpha_{min}, \alpha_{max}]$ .

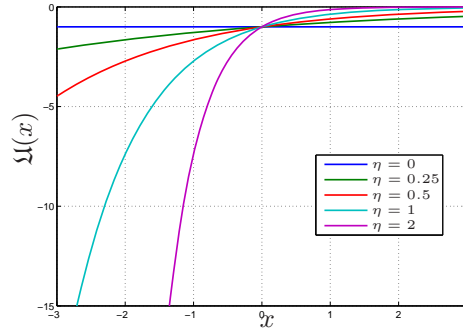


Figure 6.1: Utility function  $\mathfrak{U}(x) = -\exp(-\eta x)$  for different values of  $\eta$ .

The objective function can also be characterized by a backward stochastic differential equation with jumps (BSDEJ), as follows

$$V(w) = \mathfrak{U}(w - Y_0) = -e^{-\eta w} e^{\eta Y_0}, \quad (6.7)$$

where  $(Y, Z, U)$  is the solution to a BSDEJ, given by ([Mor10])

$$dY_t = -f(t, Z_t, U_t)dt + Z_t d\omega_t + \int_E U_t(J) \tilde{N}(dJ, dt), \quad Y_T = \xi, \quad (6.8a)$$

$$f(t, z, u) = -z \frac{b(t)}{\sigma(t)} - \frac{1}{2\eta} \left| \frac{b(t)}{\sigma(t)} \right|^2 + \min_{\alpha \in \mathcal{A}} \left[ \frac{\eta}{2} \left( \alpha \sigma(t) - \left( z + \frac{1}{\eta} \frac{b(t)}{\sigma(t)} \right) \right)^2 + |u(\cdot) - \alpha \beta(t, \cdot)|_\eta \right], \quad (6.8b)$$

$$|u(\cdot)|_\eta = \int_E \frac{\exp(\eta u(J)) - \eta u(J) - 1}{\eta} \nu(dJ), \quad (6.8c)$$

The solution of the above BSDEJ consists of a triplet of processes  $(Y, Z, U)$  in  $\mathcal{S}^\infty(\mathbb{R}) \times L^2(\omega) \times L^2(\tilde{N})$ <sup>2</sup>. Existence and uniqueness results for this BSDEJ are provided in [Mor10]. For more information about existence and uniqueness for BSDEJs, we refer to [TL94, BBP97, Roy06]. Furthermore, there exists an optimal predictable strategy  $\alpha_t^* \in \mathcal{A}$  that attains the minimum in (6.8b) for  $(t, z, u) = (t, Z_t, U_t)$ .

<sup>2</sup>Following [Mor10],  $\mathcal{S}^\infty(\mathbb{R})$  is the set of all adapted processes  $Y$  with càdlàg paths such that  $\sup_\Omega (\sup_{t \in [0, T]} |Y_t|) < \infty$ .  $L^2(\omega)$  is the set of all predictable processes  $Z$  such that  $\mathbb{E}[\int_0^T |Z_s|^2 ds] < \infty$ .  $L^2(\tilde{N})$  is the set of all  $\mathcal{P} \otimes \mathcal{B}(E)$ -measurable processes  $U$  such that  $\mathbb{E}[\int_0^T \int_E |U_s(J)|^2 \nu(dJ) ds] < \infty$ .  $\mathcal{P}$  stands for the  $\sigma$ -field of all predictable sets of  $[0, T] \times \Omega$  and  $\mathcal{B}(E)$  the Borel field of  $E$ .

## 6.2 Utility indifference price

Now we start with the utility indifference price, where the idea is the following. The writer of an option receives the option premium and hedges the option with an optimal strategy that maximizes the utility of the portfolio value at the terminal time minus the payoff. We also determine the expected utility without the option trade. The *utility indifference price* of the option is defined as the additional initial wealth with which the writer can achieve the same utility as without the option.

Let  $u_0(w)$  denote the utility maximization value without the option payoff

$$u_0(w) = \max_{\alpha_t \in \mathcal{A}} \mathbb{E} [\mathfrak{U}(W_T^\alpha)], \quad (6.9)$$

and  $u_\xi(w)$  denotes the utility maximization value in presence of the option,

$$u_\xi(w) = \max_{\alpha_t \in \mathcal{A}} \mathbb{E} [\mathfrak{U}(W_T^\alpha - \xi)]. \quad (6.10)$$

The *seller's indifference price (ask price)*  $v^a$  satisfies

$$u_0(w) = u_\xi(w + v^a). \quad (6.11)$$

In other words, it is the price at which a writer is indifferent, in the sense that the expected utility under optimal trading remains the same, between selling the option for price  $v^a$  and not selling any option. We need to solve for  $v^a$  and with the theory in Section 6.1 we find

$$\mathfrak{U}(w - Y_0^0) = \mathfrak{U}(w + v^a - Y_0^\xi) \implies v^a = Y_0^\xi - Y_0^0, \quad (6.12)$$

where  $Y_t^0$  and  $Y_t^\xi$  follow BSDEJ (6.8) with terminal conditions  $Y_T = 0$  and  $Y_T = \xi$ , respectively. With this we can value an option under jump-diffusion and when the trading strategies are constraint, for example  $\mathcal{A} = [\alpha_{min}, \alpha_{max}]$ , with the help of BSDEJs.

The *buyer's indifference price (bid price)*  $v^b$  is defined in a similar way and satisfies

$$u_0(w) = u_{-\xi}(w - v^b). \quad (6.13)$$

Again with the BSDE approach we find

$$\mathfrak{U}(w - Y_0^0) = \mathfrak{U}(w - v^b - Y_0^{-\xi}) \implies v^b = Y_0^0 - Y_0^{-\xi}. \quad (6.14)$$

Below is a list of properties of utility indifference prices (see, for example, [HH09, REK00]). We here denote by  $v(\eta, \xi)$  the option price with coefficient of absolute risk aversion  $\eta$  and option  $\xi$ .

- Prices  $v^b$  and  $v^a$  are independent of initial wealth  $w$ .
- Bid and ask prices are related via  $v^b(\eta, \xi) = -v^a(\eta, -\xi)$ .
- The ask price is larger than the bid price:  $v^a \geq v^b$ .
- If the market is complete, i.e. there are no jumps and  $\mathcal{A} = \mathbb{R}$ , then the option is perfectly replicable. The driver function reduces to  $f(t, y, z) = -z \frac{b}{\sigma} - \frac{1}{2\eta} \left| \frac{b}{\sigma} \right|^2$  and the utility indifference prices reduce to the Black-Scholes price.
- Convexity:  $v^a(\eta, \rho\xi_1 + (1 - \rho)\xi_2) \leq \rho v^a(\eta, \xi_1) + (1 - \rho)v^a(\eta, \xi_2)$ ,  $\rho \in (0, 1)$ .
- Concavity:  $v^b(\eta, \rho\xi_1 + (1 - \rho)\xi_2) \geq \rho v^b(\eta, \xi_1) + (1 - \rho)v^b(\eta, \xi_2)$ ,  $\rho \in (0, 1)$ .

### 6.3 Discretization and BCOS method for BSDEJs

In this section, we explain the BCOS method to solve BSDEJ (6.8). We suppose that the asset price follows the following FSDE:

$$dS_t/S_{t-} = bdt + \sigma d\omega_t + \int_E \beta(J)\tilde{N}(dJ, dt), \quad \text{with } \beta(J) = e^J - 1. \quad (6.15)$$

Moreover,  $E$  is assumed to be a finite set,  $E = \{j_1, j_2, \dots, j_{n_j}\}$ , with Lévy measure  $\nu(\{j_\ell\}) = \lambda p_\ell$ , where  $\lambda = \nu(\mathbb{R})$  is the intensity rate. In other words,  $p_\ell$  is the probability of jump size  $j_\ell$  and  $\nu(dJ) = \lambda \sum_{\ell=1}^{n_j} p_\ell \delta_{j_\ell}(dJ)$ . So,

$$\int_E \beta(J)\tilde{N}(dJ, dt) = \sum_{\ell=1}^{n_j} \beta(j_\ell)\tilde{N}(\{j_\ell\}, dt). \quad (6.16)$$

We define  $\mu := b - \int_E \beta(J)\nu(dJ)$  and switch to the log-asset domain  $X_t = \log S_t$ , i.e.,

$$dX_t = \left( \mu - \frac{1}{2}\sigma^2 + \int_E J\nu(dJ) \right) dt + \sigma d\omega_t + \int_E J\tilde{N}(dJ, dt). \quad (6.17)$$

The Euler discretization of the FSDE (6.17) reads

$$X_{m+1}^\Delta = X_m^\Delta + \left( \mu - \frac{1}{2}\sigma^2 + \int_E J\nu(dJ) \right) \Delta t + \sigma \Delta\omega_m + \int_E J\tilde{N}(dJ, \Delta t), \quad (6.18)$$

where we defined  $\tilde{N}(dJ, \Delta t) := \tilde{N}(dJ, (t_m, t_{m+1}]) = \tilde{N}(dJ, t_{m+1}) - \tilde{N}(dJ, t_m)$ . The characteristic function of  $X_{m+1}^\Delta$ , given  $X_m^\Delta = x$ , reads

$$\begin{aligned} \varphi(u|x) &= \varphi(u|0)e^{iux} = \phi(u)e^{iux}, \quad \text{with} \\ \phi(u) &:= \exp\left(iu\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t - \frac{1}{2}u^2\sigma^2\Delta t\right) e^{\lambda\Delta t(\varphi_J(u)-1)}, \quad \varphi_J(u) = \sum_{\ell=1}^{n_j} p_\ell e^{iuj_\ell}. \end{aligned} \quad (6.19)$$

For the discretization of the BSDEJ, we start from

$$Y_m = Y_{m+1} + \int_{t_m}^{t_{m+1}} f(s, Z_s, U_s) ds - \int_{t_m}^{t_{m+1}} Z_s d\omega_s - \int_{t_m}^{t_{m+1}} \int_E U_s(J)\tilde{N}(dJ, ds). \quad (6.20)$$

Both processes  $\omega$  and  $\tilde{N}$  are independent. Taking conditional expectations of both sides of (6.20) and applying the theta-method results, similar as equation (3.5), in

$$Y_m \approx \mathbb{E}_m[Y_{m+1}] + \Delta t\theta_1 f(t_m, Z_m, U_m) + \Delta t(1 - \theta_1)\mathbb{E}_m[f(t_{m+1}, Z_{m+1}, U_{m+1})], \quad \theta_1 \in [0, 1]. \quad (6.21)$$

Multiplying both sides of equation (6.20) by  $\Delta\omega_m$  and taking conditional expectations gives us, similar as equation (3.6),

$$\begin{aligned} 0 &\approx \mathbb{E}_m[Y_{m+1}\Delta\omega_m] + \Delta t(1 - \theta_2)\mathbb{E}_m[f(t_{m+1}, Z_{m+1}, U_{m+1})\Delta\omega_m] \\ &\quad - \Delta t\theta_2 Z_m - \Delta t(1 - \theta_2)\mathbb{E}_m[Z_{m+1}], \quad \theta_2 \in [0, 1]. \end{aligned} \quad (6.22)$$

Multiplying both sides of equation (6.20) by  $\tilde{N}(\{j_\ell\}, \Delta t)$  and taking conditional expectations gives

$$0 = \mathbb{E}_m[Y_{m+1}\tilde{N}(\{j_\ell\}, \Delta t)] + \int_{t_m}^{t_{m+1}} \mathbb{E}_m \left[ f(s, Z_s, U_s)\tilde{N}(\{j_\ell\}, s - t_m) \right] ds - \int_{t_m}^{t_{m+1}} p_\ell \lambda \mathbb{E}_m [U_s(j_\ell)] ds, \quad (6.23)$$

where we used the Itô isometry for

$$\mathbb{E}_m \left[ \int_{t_m}^{t_{m+1}} \int_E U_s(J) \tilde{N}(dJ, ds) \tilde{N}(\{j_\ell\}, \Delta t) \right] = \mathbb{E}_m \left[ \int_{t_m}^{t_{m+1}} p_\ell \lambda U_s(j_\ell) ds \right]. \quad (6.24)$$

By the theta-discretization we get

$$\begin{aligned} 0 &\approx \mathbb{E}_m[Y_{m+1} \tilde{N}(\{j_\ell\}, \Delta t)] + \Delta t(1 - \theta_3) \mathbb{E}_m[f(t_{m+1}, Z_{m+1}, U_{m+1}) \tilde{N}(\{j_\ell\}, \Delta t)] \\ &\quad - p_\ell \lambda \Delta t \theta_3 U_m(j_\ell) - p_\ell \lambda \Delta t(1 - \theta_3) \mathbb{E}_m[U_{m+1}(j_\ell)], \quad \theta_3 \in [0, 1], \text{ for } \ell = 1, \dots, n_j. \end{aligned} \quad (6.25)$$

The above equations lead to a time discretization  $(Y^\Delta, Z^\Delta, U^\Delta)$  for  $(Y, Z, U)$ , as follows

$$Y_M^\Delta = g(X_M^\Delta), \quad (6.26a)$$

for  $m = M - 1, \dots, 0$ :

$$\begin{aligned} Z_m^\Delta &= -\theta_2^{-1}(1 - \theta_2) \mathbb{E}_m[Z_{m+1}^\Delta] + \frac{1}{\Delta t} \theta_2^{-1} \mathbb{E}_m[Y_{m+1}^\Delta \Delta \omega_m] \\ &\quad + \theta_2^{-1}(1 - \theta_2) \mathbb{E}_m[f(t_{m+1}, Z_{m+1}^\Delta, U_{m+1}^\Delta) \Delta \omega_m], \end{aligned} \quad (6.26b)$$

$$\begin{aligned} U_m^\Delta(j_\ell) &= -\theta_3^{-1}(1 - \theta_3) \mathbb{E}_m[U_{m+1}^\Delta(j_\ell)] + \frac{1}{p_\ell \lambda \Delta t} \theta_3^{-1} \mathbb{E}_m[Y_{m+1}^\Delta \tilde{N}(\{j_\ell\}, \Delta t)] \\ &\quad + \frac{1}{p_\ell \lambda} \theta_3^{-1}(1 - \theta_3) \mathbb{E}_m[f(t_{m+1}, Z_{m+1}^\Delta, U_{m+1}^\Delta) \tilde{N}(\{j_\ell\}, \Delta t)], \quad \ell = 1, \dots, n_j \end{aligned} \quad (6.26c)$$

$$Y_m^\Delta = \mathbb{E}_m[Y_{m+1}^\Delta] + \Delta t \theta_1 f(t_m, Z_m^\Delta, U_m^\Delta) + \Delta t(1 - \theta_1) \mathbb{E}_m[f(t_{m+1}, Z_{m+1}^\Delta, U_{m+1}^\Delta)]. \quad (6.26d)$$

As the terminal condition is a deterministic function of  $X_M^\Delta$  and because  $X^\Delta$  is a Markov process, it is easily seen that there are deterministic functions  $y(t_m, x)$ ,  $z(t_m, x)$ , and  $u(t_m, x, j_\ell)$  so that

$$Y_m^\Delta = y(t_m, X_m^\Delta), \quad Z_m^\Delta = z(t_m, X_m^\Delta), \quad U_m^\Delta(j_\ell) = u(t_m, X_m^\Delta, j_\ell), \quad \ell = 1, \dots, n_j. \quad (6.27)$$

So, the random variables  $Y_m^\Delta$ ,  $Z_m^\Delta$ , and  $U_m^\Delta(j_\ell)$  are functions of  $X_m^\Delta$ , for each  $m = 0, \dots, M$ . The functions  $y(t_m, x)$ ,  $z(t_m, x)$ , and  $u(t_m, x, j_\ell)$  are obtained in a backward manner, by means of

$$y(t_M, x) = g(x), \quad (6.28a)$$

for  $m = M - 1, \dots, 0$ :

$$\begin{aligned} z(t_m, x) &= -\theta_2^{-1}(1 - \theta_2) \mathbb{E}_m^x[z(t_{m+1}, X_{m+1}^\Delta)] + \frac{1}{\Delta t} \theta_2^{-1} \mathbb{E}_m^x[y(t_{m+1}, X_{m+1}^\Delta) \Delta \omega_m] \\ &\quad + \theta_2^{-1}(1 - \theta_2) \mathbb{E}_m^x[f(t_{m+1}, z(t_{m+1}, X_{m+1}^\Delta), u(t_{m+1}, X_{m+1}^\Delta, (\cdot))) \Delta \omega_m], \end{aligned} \quad (6.28b)$$

$$\begin{aligned} u(t_m, x, j_\ell) &= -\theta_3^{-1}(1 - \theta_3) \mathbb{E}_m^x[u(t_{m+1}, X_{m+1}^\Delta, j_\ell)] + \frac{1}{p_\ell \lambda \Delta t} \theta_3^{-1} \mathbb{E}_m^x[y(t_{m+1}, X_{m+1}^\Delta) \tilde{N}(\{j_\ell\}, \Delta t)] \\ &\quad + \frac{1}{p_\ell \lambda} \theta_3^{-1}(1 - \theta_3) \mathbb{E}_m^x[f(t_{m+1}, z(t_{m+1}, X_{m+1}^\Delta), u(t_{m+1}, X_{m+1}^\Delta, (\cdot))) \tilde{N}(\{j_\ell\}, \Delta t)], \\ &\quad \ell = 1, \dots, n_j \end{aligned} \quad (6.28c)$$

$$\begin{aligned} y(t_m, x) &= \mathbb{E}_m^x[y(t_{m+1}, X_{m+1}^\Delta)] + \Delta t \theta_1 f(t_m, z(t_m, x), u(t_m, x, (\cdot))) \\ &\quad + \Delta t(1 - \theta_1) \mathbb{E}_m^x[f(t_{m+1}, z(t_{m+1}, X_{m+1}^\Delta), u(t_{m+1}, X_{m+1}^\Delta, (\cdot)))]. \end{aligned} \quad (6.28d)$$

Similar as in Section 4, the Fourier-cosine coefficients of the functions  $z(t_m, x)$ ,  $f(t_m, y(t_m, x), z(t_m, x), u(t_m, x, \cdot))$ , and  $y(t_m, x)$  are denoted by  $\mathcal{Z}_k(t_m)$ ,  $\mathcal{F}_k(t_m)$ , and  $\mathcal{Y}_k(t_m)$ , respectively. Let  $\mathcal{U}_k^\ell(t_m)$  be the Fourier-cosine coefficients of  $u(t_m, x, j_\ell)$ , i.e.,

$$\mathcal{U}_k^\ell(t_m) = \frac{2}{b-a} \int_a^b u(t_m, x, j_\ell) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \quad \ell = 1, \dots, n_j. \quad (6.29)$$

We obtain the following COS formulas to approximate the conditional expectations in equation (6.28c), see Appendix A.2 for details,

$$\mathbb{E}_m^x[U_{m+1}^\Delta(j_\ell)] \approx \sum_{k=0}^{N-1} \mathcal{U}_k^\ell(t_{m+1}) \Re \left( \phi \left( \frac{k\pi}{b-a} \right) e^{ik\pi \frac{x-a}{b-a}} \right), \quad (6.30a)$$

$$\mathbb{E}_m^x \left[ Y_{m+1}^\Delta \tilde{N}(\{j_\ell\}, \Delta t) \right] \approx \sum_{k=0}^{N-1} \mathcal{Y}_k(t_{m+1}) \Re \left\{ \phi \left( \frac{k\pi}{b-a} \right) e^{ik\pi \frac{x-a}{b-a}} p_\ell \lambda \Delta t \left[ \exp \left( i \frac{k\pi j_\ell}{b-a} \right) - 1 \right] \right\}, \quad (6.30b)$$

$$\mathbb{E}_m^x \left[ f(t_{m+1}, Z_{m+1}^\Delta, U_{m+1}^\Delta) \tilde{N}(\{j_\ell\}, \Delta t) \right] \approx \sum_{k=0}^{N-1} \mathcal{F}_k(t_{m+1}) \Re \left\{ \phi \left( \frac{k\pi}{b-a} \right) e^{ik\pi \frac{x-a}{b-a}} p_\ell \lambda \Delta t \left[ \exp \left( i \frac{k\pi j_\ell}{b-a} \right) - 1 \right] \right\}. \quad (6.30c)$$

Furthermore we use the COS formulas (4.16a), (4.16b), (4.16c), (4.18a), and (4.18b) from Section 4.2 and 4.3, obtained with the equality in Appendix A.1, to find

$$\begin{aligned} z(t_m, x) &\approx -\frac{1-\theta_2}{\theta_2} \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{Z}_k(t_{m+1}) \Phi_k(x) + \frac{1}{\Delta t \theta_2} \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{Y}_k(t_{m+1}) \sigma \Delta t \frac{d\Phi_k(x)}{dx} \\ &\quad + \frac{1-\theta_2}{\theta_2} \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{F}_k(t_{m+1}) \sigma \Delta t \frac{d\Phi_k(x)}{dx}, \end{aligned} \quad (6.31a)$$

$$\begin{aligned} u(t_m, x, j_\ell) &\approx -\frac{1-\theta_3}{\theta_3} \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{U}_k^\ell(t_{m+1}) \Phi_k(x) + \frac{1}{\Delta t \theta_3} \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{Y}_k(t_{m+1}) \left[ \exp \left( i \frac{k\pi j_\ell}{b-a} \right) - 1 \right] \Phi_k(x) \\ &\quad + \frac{1-\theta_3}{\theta_3} \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{F}_k(t_{m+1}) \left[ \exp \left( i \frac{k\pi j_\ell}{b-a} \right) - 1 \right] \Phi_k(x), \end{aligned} \quad (6.31b)$$

$$\begin{aligned} y(t_m, x) &\approx \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{Y}_k(t_{m+1}) \Phi_k(x) + \Delta t \theta_1 f(t_m, z(t_m, x), u(t_m, x, \cdot)) \\ &\quad + \Delta t (1-\theta_1) \frac{b-a}{2} \sum_{k=0}^{N-1} \mathcal{F}_k(t_{m+1}) \Phi_k(x). \end{aligned} \quad (6.31c)$$

The coefficients  $\mathcal{Z}_k(t_m)$ ,  $\mathcal{F}_k(t_m)$ , and  $\mathcal{Y}_k(t_m)$  are recovered in a similar way as explained in Section 4.4. The computation of the Fourier-cosine coefficients  $\mathcal{U}_k^\ell(t_m)$  of function  $u(t_m, x, j_\ell)$  can be decomposed into three parts

$$\frac{2}{b-a} \int_a^b \mathbb{E}_m^x[U_{m+1}^\Delta(j_\ell)] \cos \left( k\pi \frac{x-a}{b-a} \right) dx \approx \Re \left( \sum_{j=0}^{N-1} \mathcal{U}_j^\ell(t_{m+1}) \phi \left( \frac{j\pi}{b-a} \right) \mathcal{M}_{k,j} \right), \quad (6.32a)$$

$$\begin{aligned} \frac{2}{b-a} \int_a^b \mathbb{E}_m^x[Y_{m+1}^\Delta N(\{j_\ell\}, \Delta t)] \cos \left( k\pi \frac{x-a}{b-a} \right) dx \\ \approx \Re \left( \sum_{j=0}^{N-1} p_\ell \lambda \Delta t \left[ \exp \left( i \frac{k\pi j_\ell}{b-a} \right) - 1 \right] \mathcal{Y}_j(t_{m+1}) \phi \left( \frac{j\pi}{b-a} \right) \mathcal{M}_{k,j} \right), \end{aligned} \quad (6.32b)$$

$$\frac{2}{b-a} \int_a^b \mathbb{E}_m^x[f(t_{m+1}, Z_{m+1}^\Delta, U_{m+1}^\Delta) N(\{j_\ell\}, \Delta t)] \cos \left( k\pi \frac{x-a}{b-a} \right) dx$$

$$\approx \Re \left( \sum_{j=0}^{N-1} p_\ell \lambda \Delta t \left[ \exp \left( i \frac{k\pi j \ell}{b-a} \right) - 1 \right] \mathcal{F}_j(t_{m+1}) \phi \left( \frac{j\pi}{b-a} \right) \mathcal{M}_{k,j} \right), \quad (6.32c)$$

In summary, this results in

$$\begin{aligned} \mathcal{U}_k^\ell(t_m) \approx & \Re \left( \sum_{j=0}^{N-1} \left[ -\frac{1-\theta_3}{\theta_3} \mathcal{U}_j^\ell(t_{m+1}) \phi \left( \frac{j\pi}{b-a} \right) + \frac{1}{\Delta t \theta_3} \Delta t \left[ \exp \left( i \frac{k\pi j \ell}{b-a} \right) - 1 \right] \mathcal{Y}_j(t_{m+1}) \phi \left( \frac{j\pi}{b-a} \right) \right. \right. \\ & \left. \left. + \frac{1-\theta_3}{\theta_3} \Delta t \left[ \exp \left( i \frac{k\pi j \ell}{b-a} \right) - 1 \right] \mathcal{F}_j(t_{m+1}) \phi \left( \frac{j\pi}{b-a} \right) \right] \mathcal{M}_{k,j} \right). \end{aligned} \quad (6.33)$$

With the above equations we can recover the Fourier-cosine coefficients recursively and solve the BSDEJ backwards in time.

#### 6.4 Reference values

We first explain briefly how we can use the original COS method, in a completely different way, to obtain reference values for the numerical tests in Section 7. The utility maximization problem,

$$V(w) = \max_{\alpha_t \in \mathcal{A}} \mathbb{E} [\mathfrak{U}(W_T^\alpha - g(S_T))], \quad (6.34)$$

is a two-dimensional stochastic control problem with the following underlying processes

$$dS_t/S_{t-} = bdt + \sigma d\omega_t + \int_E \beta(J) \tilde{N}(dJ, dt), \quad (6.35)$$

$$dW_t^\alpha = \alpha_t bdt + \alpha_t \sigma d\omega_t + \alpha_t \int_E \beta(J) \tilde{N}(dJ, dt). \quad (6.36)$$

We can solve this problem by the 2D-COS method. This method was developed in [RO12] for pricing rainbow options, for which the payoff depends on two or more asset price processes, and can also be applied to stochastic control problems.

If it is not possible to invest in assets and to hedge the risky option, i.e.  $\alpha_t = 0, \forall t \in [0, T]$ , then the portfolio value  $W_t^\alpha = w$  is constant and the problem reduces to

$$V(w) = \mathbb{E} [\mathfrak{U}(w - g(S_T))] = -e^{-\eta w} \mathbb{E} \left[ e^{\eta g(S_T)} \right]. \quad (6.37)$$

We can approximate this one-dimensional expectation by using the one-dimensional COS formula from [FO08].

## 7 Numerical experiments BSDEJ

In this section we use the BCOS method to value a put option under jump-diffusion asset prices by using utility indifference pricing, as explained in Section 6.2. For the numerical tests, we use the following parameter values

$$S_0 = 1, K = 1, b = 0.1779, \sigma = 0.2, T = 0.1. \quad (7.1)$$

The jumps occurring are assumed to be bivariate distributed with possible jump sizes  $j_1$  and  $j_2$ , with

$$j_1 = -0.1338, j_2 = -0.9838, p_1 = p_2 = 0.5, \lambda = 0.0228, \quad (7.2)$$



so that the expected value is  $-0.5588$  and the standard deviation is  $0.4250$ . These values correspond to the real-world  $\mathbb{P}$ -measure for the jump-diffusion asset price in [KfV09].

Similar as in [FO08], we choose the computational domain

$$[a, b] = \left[ x_0 + \kappa_1 - L\sqrt{\kappa_2 + \sqrt{\kappa_4}}, x_0 + \kappa_1 + L\sqrt{\kappa_2 + \sqrt{\kappa_4}} \right], \quad L \in [12]. \quad (7.3)$$

The cumulants  $\kappa_1$ ,  $\kappa_2$ , and  $\kappa_4$  of the Brownian motion and the Merton jump-diffusion process are, for example, given in [FO08]. Again we set the number of terms in the Fourier-cosine series expansions equal to  $N = 2^9$  and  $Q = N$  grid points for the DCT.

We distinguish between three theta-discretization schemes:

$$\begin{aligned} \text{Scheme E:} & \quad \theta_1 = 1, \quad \theta_2 = 1, \quad \theta_3 = 1, \\ \text{Scheme F:} & \quad \theta_1 = 0.5, \quad \theta_2 = 1, \quad \theta_3 = 1, \\ \text{Scheme G:} & \quad \theta_1 = 0.5, \quad \theta_2 = 0.5, \quad \theta_3 = 0.5. \end{aligned}$$

For solving equations (6.28) in the first time iteration,  $m = M - 1$ , we set  $\theta_1 = \theta_2 = \theta_3 = 1$ , because the driver function  $f(\cdot)$  depends on the unprescribed values  $z(t_M, x)$  and  $u(t_M, x, \cdot)$ .

**No hedge** We start with the setting where it is not possible to invest in assets and to hedge the risky option, i.e.  $\alpha_t = 0, \forall t \in [0, T]$ . In Figure 7.1 results of the BCOS method are shown. The left-side plot shows the initial values of the BSDEs,  $Y_0^\xi$ ,  $Y_0^{-\xi}$ , and  $Y_0^0$ , for different values of  $\eta$ , and the right-side plots gives the bid and ask prices. The dots are the values obtained by the BCOS method, while the circles give the reference value obtained by the COS method as described in Section 6.4. The approximated values are equal.

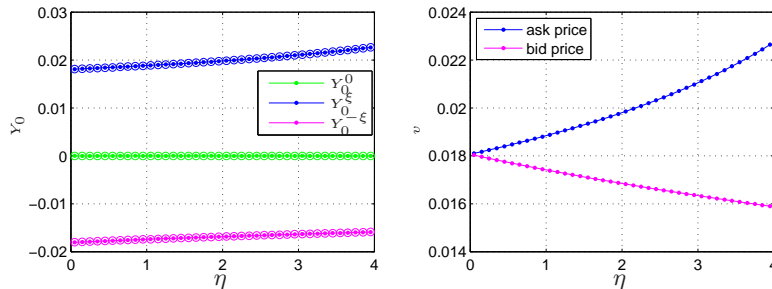


Figure 7.1: Results  $Y_0$  and utility indifference prices (Scheme G,  $N = 2^9$ ,  $M = 64$ ).

**Restricted hedging strategy** For the second test we assume that the set of admissible strategies is given by  $\mathcal{A} = [-15, 15]$ . In other words, a maximum of 15 Euro is used to buy or sell assets. We use the Newton's method to find the optimal strategy in equation (6.8b). Figure 7.2 presents the results of the BCOS method. The reference values (circles) are obtained by the 2D-COS method.

**Convergence in  $M$**  For the last test we investigate the convergence of the error in the number of timesteps  $M$  for  $\eta = 1$  and with terminal conditions  $\xi$  and  $-\xi$ . Reference values are obtained by choosing a high number of timesteps  $M$ . The approximated value  $\hat{y}(t_0, x_0)$  converges with  $O(\Delta t)$  for schemes E and F and with  $O((\Delta t)^2)$  for scheme G, as expected. The values  $\hat{z}(t_0, x_0)$ ,  $\hat{u}(t_0, x_0, j_1)$ , and  $\hat{u}(t_0, x_0, j_2)$  converge with  $O(\Delta t)$  for all three schemes. Again the scheme with  $\theta_i = 1/2, i = 1, 2, 3$  gives the best convergence rate. The CPU times for different values of  $N$  and  $M$  are shown in Table 7.1.

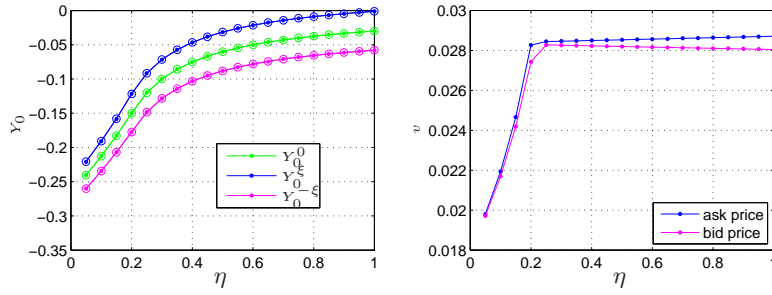


Figure 7.2: Results  $Y_0$  and utility indifference prices (Scheme G,  $N = 2^9$ ,  $M = 64$ ).

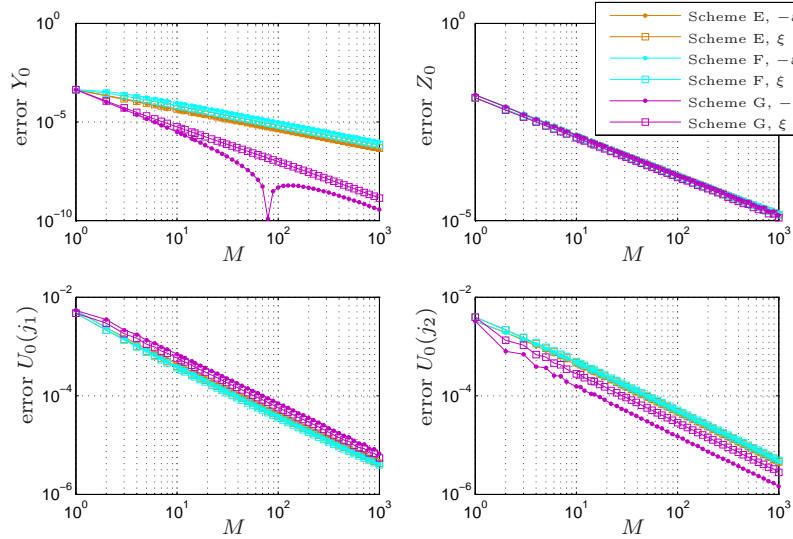


Figure 7.3: Convergence in  $M$  ( $N = 2^9$ ), upper left: error  $\hat{y}(t_0, x_0)$ , upper right: error  $\hat{z}(t_0, x_0)$ , lower left: error  $\hat{u}(t_0, x_0, j_1)$ , lower right: error  $\hat{u}(t_0, x_0, j_2)$ .

$M$	4	8	16	32	64	128	256	512
$N = 2^9$	0.0694	0.1086	0.1908	0.3358	0.6428	1.2555	2.4931	4.9387
$N$	$2^6$	$2^7$	$2^8$	$2^9$				
$M = 256$	0.7897	1.0745	1.5204	2.4931				

Table 7.1: CPU time (s).

## 8 Conclusion

In this paper we proposed a probabilistic numerical method for solving backward stochastic differential equations (BSDEs). The first step consists of discretizing the BSDE by taking conditional expectations and applying a general theta-discretization for the time-integrals. Then, the BCOS method solves the problem backwards in time by approximating the conditional expectations with the help of COS formulas. The Fourier-cosine coefficients are recovered recursively in an efficient way by using discrete Fourier-cosine transforms and an FFT algorithm.

Numerical tests demonstrate the applicability of the BCOS method for BSDEs in economic and

financial problems. In the tests we observed different convergence results for  $Z_0$  and  $Y_0$ . The convergence of the error in the number of timesteps depends on the smoothness and Lipschitz constant of the driver function and the terminal condition. In general, we achieve the highest convergence rate for the theta-scheme with  $\theta_1 = \theta_2 = \frac{1}{2}$ .

Utility indifference pricing is used to value options in an incomplete market under a jump-diffusion asset price process, possibly with a restricted hedging portfolio. The bid and ask prices are represented by BSDEs with jumps. We extended our BCOS method to solving these BSDEJs under jump-diffusion with a finite number of jump sizes. Numerical experiments show highly satisfactorily and efficient pricing results. The theta-scheme with  $\theta_1 = \theta_2 = \theta_3 = \frac{1}{2}$  gives the fastest convergence.

## Appendix

### A COS formulas

In this section we explain how to approximate several conditional expectations under the discrete process

$$X_{m+1}^\Delta = X_m^\Delta + \mu\Delta t + \sigma\Delta\omega_{m+1} + \int_E JN(dJ, \Delta t), \quad (\text{A.1})$$

with characteristic function

$$\begin{aligned} \varphi(u|x) &= \varphi(u|0)e^{iux} = \phi(u)e^{iux}, \quad \text{with} \\ \phi(u) &:= \exp\left(iu\mu\Delta t - \frac{1}{2}u^2\sigma^2\Delta t\right) e^{\lambda\Delta t(\varphi_J(u)-1)}, \end{aligned} \quad (\text{A.2})$$

where  $\varphi_J(\cdot)$  denotes the characteristic function of jump size  $J$ . We define  $u_k := \frac{k\pi}{b-a}$ .

#### A.1 Computation of expectation $\mathbb{E}_m^x[\cdot \Delta\omega_m]$

For equations (4.12), (4.16b), and (4.16c) we need to compute the expectation

$$\begin{aligned} \mathbb{E}_m^x[\cos(u_k(X_{m+1}^\Delta - a)) \Delta\omega_m] &= \Re\left(\mathbb{E}_m^x\left[\exp(iu_k(X_{m+1}^\Delta - a)) \Delta\omega_m\right]\right) \\ &= \Re\left(\mathbb{E}_m^x\left[\exp\left(iu_k\left(x + \mu\Delta t + \sigma\Delta\omega_{m+1} + \int_E JN(dJ, \Delta t) - a\right)\right) \Delta\omega_{m+1}\right]\right) \\ &= \Re\left(\frac{1}{\sqrt{2\pi}\sqrt{\Delta t}} \int_{\mathbb{R}} \exp\left(iu_k\left(x + \mu\Delta t + \sigma\theta + \int_E JN(dJ, \Delta t) - a\right)\right) \theta e^{-\frac{1}{2}\left(\frac{\theta}{\sqrt{\Delta t}}\right)^2} d\theta\right). \end{aligned} \quad (\text{A.3})$$

Integration by parts gives us

$$\begin{aligned} \mathbb{E}_m^x[\cos(u_k(X_{m+1}^\Delta - a)) \Delta\omega_m] &= \Re\left(iu_k\sigma\Delta t \frac{1}{\sqrt{2\pi}\sqrt{\Delta t}} \int_{\mathbb{R}} \exp\left(iu_k\left(x + \mu\Delta t + \sigma\theta + \int_E JN(dJ, \Delta t) - a\right)\right) e^{-\frac{1}{2}\left(\frac{\theta}{\sqrt{\Delta t}}\right)^2} d\theta\right) \\ &= \sigma\Delta t \Re\left(\mathbb{E}_m^x\left[iu_k \exp\left(iu_k\left(x + \mu\Delta t + \sigma\Delta\omega_m + \int_E JN(dJ, \Delta t) - a\right)\right)\right]\right) \\ &= \sigma\Delta t \Re\left(iu_k \phi(u_k) e^{iu_k(x-a)}\right). \end{aligned} \quad (\text{A.4})$$

The derivation for diffusion processes can be found by omitting the jump part in the derivation.

## A.2 Computation of expectation $\mathbb{E}_m^x \left[ \cdot \tilde{N}(\{j_\ell\}, \Delta t) \right]$

Equation (6.30b), and similarly equation (6.30c), require the computation of

$$\mathbb{E}_m^x \left[ Y_{m+1}^\Delta \tilde{N}(\{j_\ell\}, \Delta t) \right] = \mathbb{E}_m^x \left[ Y_{m+1}^\Delta N(\{j_\ell\}, \Delta t) \right] - \mathbb{E}_m^x \left[ Y_{m+1}^\Delta \right] \nu(\{j_\ell\}) \Delta t. \quad (\text{A.5})$$

The second part in (A.5) can be written as

$$\mathbb{E}_m^x \left[ Y_{m+1}^\Delta N(\{j_\ell\}, \Delta t) \right] \approx \sum_{k=0}^{N-1} \mathcal{Y}_k(t_{m+1}) \Re \left( \mathbb{E}_m^x \left[ N(\{j_\ell\}, \Delta t) \exp(iu_k(X_{m+1}^\Delta - a)) \right] \right), \quad (\text{A.6})$$

with

$$\begin{aligned} & \mathbb{E}_m^x \left[ N(\{j_\ell\}, \Delta t) \exp(iu_k(X_{m+1}^\Delta - a)) \right] \\ &= \mathbb{E}_m^x \left[ \exp(iu_k(x + \mu\Delta t + \sigma\Delta\omega_m - a)) \right] \mathbb{E}_m^x \left[ N(\{j_\ell\}, \Delta t) \exp\left(iu_k \int_E JN(dJ, \Delta t)\right) \right]. \end{aligned} \quad (\text{A.7})$$

Now let  $\tau_q, q = 1, 2, \dots, N_{\Delta t}$  denote the jump times between  $t_m$  and  $t_{m+1}$ , with jump sizes  $J_{\tau_q}$ . Then, we find by the law of iterated expectations the following equality

$$\begin{aligned} & \mathbb{E} \left[ N(\{j_\ell\}, \Delta t) \exp\left(iu_k \int_E JN(dJ, \Delta t)\right) \right] = \mathbb{E} \left[ \sum_{q=1}^{N_{\Delta t}} \mathbb{I}_{j_\ell}(J_{\tau_q}) \exp\left(iu_k \sum_{q=1}^{N_{\Delta t}} J_{\tau_q}\right) \right] \\ &= \mathbb{E} \left[ \mathbb{E} \left[ \sum_{q=1}^{N_{\Delta t}} \mathbb{I}_{j_\ell}(J_{\tau_q}) \exp\left(iu_k \sum_{q=1}^{N_{\Delta t}} J_{\tau_q}\right) \middle| N_{\Delta t} \right] \right] \\ &= \sum_{n=0}^{\infty} e^{-\lambda\Delta t} \frac{(\lambda\Delta t)^n}{n!} \mathbb{E} \left[ \sum_{q=1}^n \mathbb{I}_{j_\ell}(J_{\tau_q}) \exp\left(iu_k \sum_{q=1}^n J_{\tau_q}\right) \right] \\ &= \sum_{n=0}^{\infty} e^{-\lambda\Delta t} \frac{(\lambda\Delta t)^n}{n!} n p_\ell e^{iu_k j_\ell} (\varphi_J(u_k))^{n-1} \\ &= e^{iu_k j_\ell} p_\ell \lambda \Delta t e^{\lambda\Delta t (\varphi_J(u_k) - 1)}. \end{aligned} \quad (\text{A.8})$$

We end up with the approximation

$$\mathbb{E}_m^x \left[ Y_{m+1}^\Delta \tilde{N}(\{j_\ell\}, \Delta t) \right] \approx \sum_{k=0}^{N-1} \mathcal{Y}_k(t_{m+1}) \Re \left( \phi(u_k) e^{iu_k(x-a)} [\exp(iu_k j) - 1] p_\ell \lambda \Delta t \right). \quad (\text{A.9})$$

## B Discrete Fourier-cosine Transform

In this section, we explain the idea of using discrete Fourier-cosine transforms (DCTs) to approximate the Fourier-cosine coefficients  $\mathcal{F}_k$  of function  $f(x)$ , i.e.,

$$\mathcal{F}_k = \frac{2}{b-a} \int_a^b f(x) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \quad (\text{B.1})$$

For this, we take  $Q \geq N$  grid-points and define

$$x^n := a + (n + \frac{1}{2}) \frac{b-a}{Q} \quad \text{and} \quad \Delta x := \frac{b-a}{Q}. \quad (\text{B.2})$$

We determine the value of function  $f(x)$  on the grid-points. The midpoint-rule integration gives us

$$\begin{aligned}\mathcal{F}_k &\approx \sum_{n=0}^{Q-1} \frac{2}{b-a} f(x^n) \cos\left(k\pi \frac{x^n - a}{b-a}\right) \Delta x \\ &= \sum_{n=0}^{Q-1} f(x^n) \cos\left(k\pi \frac{2n+1}{2Q}\right) \frac{2}{Q}.\end{aligned}\tag{B.3}$$

The above DCT (Type II) can be calculated efficiently by, for example, MATLAB's function `dct`.

## References

- [BBP97] G. Barles, R. Buckdahn, and E. Pardoux. Backward stochastic differential equations and integral-partial differential equations. *Stochastics: An International Journal of Probability and Stochastic Processes*, 60(1-2):57–83, 1997.
- [BD07] C. Bender and R. Denk. A forward scheme for backward SDEs. *Stochastic Processes and their Applications*, 117(12):1793–1812, 2007.
- [BE08] B. Bouchard and R. Elie. Discrete-time approximation of decoupled forward–backward SDE with jumps. *Stochastic Processes and their Applications*, 118(1):53–75, 2008.
- [Bec06] D. Becherer. Bounded solutions to backward SDEs with jumps for utility optimization and indifference hedging. *The Annals of Applied Probability*, 16(4):2027–2054, 2006.
- [Ber95] Y. Z. Bergman. Option pricing with differential interest rates. *Review of Financial Studies*, 8(2):475–500, 1995.
- [Bis73] J.-M. Bismut. Conjugate convex functions in optimal stochastic control. *Journal of Mathematical Analysis and Applications*, 44(2):384–404, 1973.
- [Boy01] J. P. Boyd. *Chebyshev and Fourier Spectral Methods: Second Revised Edition*. Dover Books on Mathematics Series. Dover Publ., 2001.
- [BS12] C. Bender and J. Steiner. Least-squares Monte Carlo for backward SDEs. *Numerical Methods in Finance*, 12:257–289, 2012.
- [BT04] B. Bouchard and N. Touzi. Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations. *Stochastic Processes and their applications*, 111(2):175–206, 2004.
- [EKPQ97] N. El Karoui, S. Peng, and M. C. Quenez. Backward stochastic differential equations in finance. *Mathematical finance*, 7(1):1–71, 1997.
- [EKQ95] N. El Karoui and M. C. Quenez. Dynamic programming and pricing of contingent claims in an incomplete market. *SIAM journal on Control and Optimization*, 33(1):29–66, 1995.
- [FL07] P. A. Forsyth and G. Labahn. Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. *Journal of Computational Finance*, 11(2):1–43, 2007.
- [FO08] F. Fang and C. W. Oosterlee. A novel pricing method for European options based on Fourier-cosine series expansions. *SIAM Journal on Scientific Computing*, 31(2):826–848, 2008.
- [FO09] F. Fang and C.W. Oosterlee. Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. *Numerische Mathematik*, 114(1):27–62, 2009.
- [FS86] H. Föllmer and D. Sondermann. Hedging of non-redundant contingent claims. In *Contributions to mathematical economics*, pages 205–223. North-Holland, Amsterdam, 1986.
- [Gen72] W. M. Gentleman. Implementing clenshaw-curtis quadrature, i methodology and experience. *Communications of the ACM*, 15(5):337–342, 1972.
- [GL10] Emmanuel Gobet and Céline Labart. Solving bsde with adaptive control variate. *SIAM Journal on Numerical Analysis*, 48(1):257–277, 2010.
- [GLW05] E. Gobet, J.P. Lemor, and X. Warin. A regression-based Monte Carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability*, 15(3):2172–2202, 2005.
- [Gob10] E. Gobet. Numerics of Backward SDEs. Presentation slides, 2010. Summer School in Probability Theory - Disentis - 26-30 July 2010.
- [HH09] V. Henderson and D. Hobson. Utility indifference pricing - an overview. In R. Carmona, editor, *Indifference Pricing: Theory and Applications*, pages 44–74. Princeton University Press, 2009.

- [HIM05] Y. Hu, P. Imkeller, and M. Müller. Utility maximization in incomplete markets. *The Annals of Applied Probability*, 15(3):1691–1712, 2005.
- [HN89] S. Hodges and A. Neuberger. Optimal replication of contingent claims under transactions costs. *Review of Futures Markets*, 8:222–239, 1989.
- [Ise09] A. Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2009.
- [KfV09] J.S. Kennedy, P.A. Forsyth, and K.R. Vetzal. Dynamic hedging under jump diffusion with transaction costs. *Operations Research*, 57(3):541–559, 2009.
- [Kob00] M. Kobylanski. Backward stochastic differential equations and partial differential equations with quadratic growth. *Annals of Probability*, 28(2):558–602, 2000.
- [KP92] P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Applications of Mathematics Series. Springer, 1992.
- [LGW06] J.P. Lemor, E. Gobet, and X. Warin. Rate of convergence of an empirical regression method for solving generalized backward stochastic differential equations. *Bernoulli*, 12(5):889–916, 2006.
- [LSM97] J.P. Lepeltier and J. San Martín. Backward stochastic differential equations with continuous coefficient. *Statistics & Probability Letters*, 32(4):425–430, 1997.
- [Mor10] M. A. Morlais. A new existence result for quadratic BSDEs with jumps with application to the utility maximization problem. *Stochastic Processes and their Applications*, 120(10):1966–1995, 2010.
- [MPSMT02] J. Ma, P. Protter, J. San Martín, and S. Torres. Numerical method for backward stochastic differential equations. *The Annals of Applied Probability*, 12(1):302–316, 2002.
- [MS05] M. Mania and M. Schweizer. Dynamic exponential utility indifference valuation. *The Annals of Applied Probability*, 15(3):2113–2143, 2005.
- [Pha09] H. Pham. *Continuous-time stochastic control and optimization with financial applications*. Stochastic modelling and applied probability. Springer-Verlag, 2009.
- [PP90] E. Pardoux and S. G. Peng. Adapted solution of a backward stochastic differential equation. *Systems & Control Letters*, 14(1):55–61, 1990.
- [PP92] E. Pardoux and S. Peng. Backward stochastic differential equations and quasilinear parabolic partial differential equations. In *Stochastic partial differential equations and their applications*, volume 176 of *Lecture Notes in Control and Information Sciences.*, pages 200–217. Springer, Berlin, 1992.
- [REK00] R. Rouge and N. El Karoui. Pricing via utility maximization and entropy. *Mathematical Finance*, 10(2):259–276, 2000.
- [RO12] M. J. Ruijter and C. W. Oosterlee. Two-dimensional Fourier cosine series expansion method for pricing financial options. *SIAM Journal on Scientific Computing*, 34(5):B642–B671, 2012.
- [Roy06] M. Royer. Backward stochastic differential equations with jumps and related non-linear expectations. *Stochastic processes and their applications*, 116(10):1358–1376, 2006.
- [Sek06] J. Sekine. On exponential hedging and related quadratic backward stochastic differential equations. *Applied Mathematics & Optimization*, 54(2):131–158, 2006.
- [Shr04] S. E. Shreve. *Stochastic Calculus for Finance II: Continuous-Time Models*. Springer Finance. Springer, 2004.
- [TL94] S. Tang and X. Li. Necessary conditions for optimal control of stochastic systems with random jumps. *SIAM Journal on Control and Optimization*, 32(5):1447–1475, 1994.
- [WLZ09] J. Wang, C. Luo, and W. Zhao. Crank-Nicolson scheme and its error estimates for backward stochastic differential equations. *Acta Mathematicae Applicatae Sinica, English Series*, pages 1–10, 2009.
- [Zha04] J. Zhang. A numerical scheme for BSDEs. *The Annals of Applied Probability*, 14(1):459–488, 2004.
- [ZWP09] W. Zhao, J. Wang, and S. Peng. Error estimates of the theta-scheme for backward stochastic differential equations. *Discrete and Continuous Dynamical Systems - Series B*, 12(4):905–924, 2009.