Fourier Coupled BSDE Method for Milstein and Second-Order Taylor Schemes

Balint Negyesi[1,2], Kees Oosterlee[1]

[1]Utrecht University; [2]TU Delft

June 30, 2025

Agenda

1. Coupled BSDEs
2. High-Order Discrete Scheme for the Forward Equation
3. Discrete Characteristic Function
4. BCOS Method
5. Numerical Experiments

BSDEs

- BSDEs provide a probabilistic representation of semi–linear/nonlinear PDEs.
- Our aim is for fast and accurate solvers.
- Some relevant forward SDEs are *non-affine* (CEV, local volatility), so characteristic functions are typically unknown.
- Coupled BSDEs pose a specific challenge!
- Goal: accurate, Fourier-based method for non-affine drift/volatility.

Problem Statement

Decoupled FBSDE on $[0, T]$

$$\begin{cases} \mathrm{d}X_t = \mu(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}W_t, \ X_0 = x_0 \\ \mathrm{d}Y_t = -f(t, X_t, Y_t, Z_t)\,\mathrm{d}t + Z_t\,\mathrm{d}W_t, \ Y_T = g(X_T) \end{cases} \Rightarrow \text{solve } (Y_0, Z_0)$$

Problem Statement

Decoupled FBSDE on $[0, T]$

$$\begin{cases} \mathrm{d}X_t = \mu(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}W_t,\ X_0 = x_0 \\ \mathrm{d}Y_t = -f(t, X_t, Y_t, Z_t)\,\mathrm{d}t + Z_t\,\mathrm{d}W_t,\ Y_T = g(X_T) \end{cases} \Rightarrow \text{solve } (Y_0, Z_0)$$

Coupled FBSDE:

$$\begin{cases} \mathrm{d}X_t = \mu(t, X_t, Y_t, Z_t)\,dt + \sigma(t, X_t, Y_t, Z_t)\,dW_t \\ \mathrm{d}Y_t = -f(t, X_t, Y_t, Z_t)\,dt + Z_t\,dW_t, \quad t \in [0, T], \\ X_0 = x_0, \quad Y_T = g(X_T). \end{cases}$$

- Drift μ and volatility σ now depend on Y, Z $\Rightarrow$ fully–coupled system.

Problem Statement

Decoupled FBSDE on $[0, T]$

$$\begin{cases} \mathrm{d}X_t = \mu(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}W_t,\ X_0 = x_0 \\ \mathrm{d}Y_t = -f(t, X_t, Y_t, Z_t)\,\mathrm{d}t + Z_t\,\mathrm{d}W_t,\ Y_T = g(X_T) \end{cases} \quad \Rightarrow \text{solve } (Y_0, Z_0)$$

Coupled FBSDE:

$$\begin{cases} \mathrm{d}X_t = \mu(t, X_t, Y_t, Z_t)\,dt + \sigma(t, X_t, Y_t, Z_t)\,dW_t \\ \mathrm{d}Y_t = -f(t, X_t, Y_t, Z_t)\,dt + Z_t\,dW_t, \quad t \in [0, T], \\ X_0 = x_0, \quad Y_T = g(X_T). \end{cases}$$

- Drift μ and volatility σ now depend on Y, Z $\Rightarrow$ fully–coupled system.

$\Rightarrow$ Uniform grid $\pi = \{t_n\}_{n=0}^{N}$, $\Delta t = t_{n+1} - t_n$.

Backward Scheme with θ-Parameters (Zhao et al. 2012)

$$y(t_N,x) = g(x), \quad z(t_N,x) = \partial_x g(x)\,\sigma(t_N, y(t_N,x), z(t_N,x)),$$

$$\begin{aligned} z(t_n,x) = \frac{1}{\theta_3 \Delta t_n}\Big(&\theta_4 \Delta t_n\, \mathbb{E}_n^x[z(t_{n+1},X_{n+1})] \\ &+ (\theta_3-\theta_4)\,\mathbb{E}_n^x[y(t_{n+1},X_{n+1})\,\Delta W_n] \\ &+ (1-\theta_2)\Delta t_n\, \mathbb{E}_n^x[f(t_{n+1}, y(t_{n+1},X_{n+1}), z(t_{n+1},X_{n+1}))\,\Delta W_n]\Big), \end{aligned}$$

$$\begin{aligned} y(t_n,x) = \mathbb{E}_n^x\Big[&y(t_{n+1},X_{n+1}) \\ &+ (1-\theta_1)\Delta t_n\, f(t_{n+1}, y(t_{n+1},X_{n+1}), z(t_{n+1},X_{n+1}))\Big] \\ &+ \theta_1 \Delta t_n\, f(t_n, y(t_n,x), z(t_n,x)). \end{aligned}$$

with $\theta_1, \theta_2 \in [0,1]$, $\theta_3 \in (0,1]$, $\theta_4 \leq \theta_3$

Backward Scheme with θ-Parameters (Zhao et al. 2012)

$$y(t_N, x) = g(x), \quad z(t_N, x) = \partial_x g(x)\, \sigma(t_N, y(t_N, x), z(t_N, x)),$$

$$\begin{aligned} z(t_n, x) = \frac{1}{\theta_3 \Delta t_n}\Big(& \theta_4 \Delta t_n \, \mathbb{E}_n^x [z(t_{n+1}, X_{n+1})] \\ & + (\theta_3 - \theta_4)\, \mathbb{E}_n^x [y(t_{n+1}, X_{n+1})\, \Delta W_n] \\ & + (1 - \theta_2) \Delta t_n \, \mathbb{E}_n^x [f(t_{n+1}, y(t_{n+1}, X_{n+1}), z(t_{n+1}, X_{n+1}))\, \Delta W_n] \Big), \end{aligned}$$

$$\begin{aligned} y(t_n, x) = \mathbb{E}_n^x \Big[& y(t_{n+1}, X_{n+1}) \\ & + (1 - \theta_1) \Delta t_n \, f(t_{n+1}, y(t_{n+1}, X_{n+1}), z(t_{n+1}, X_{n+1}) \Big] \\ & + \theta_1 \Delta t_n \, f(t_n, y(t_n, x), z(t_n, x)). \end{aligned}$$

with $\theta_1, \theta_2 \in [0, 1]$, $\theta_3 \in (0, 1]$, $\theta_4 \leq \theta_3$

$\theta_1 = \theta_2 = \theta_3 = 1/2; \theta_4 = -1/2$, 2^{nd}-order accuracy

Motivation and Objective

We aim to extend the COS method to fully coupled FBSDEs using higher-order Taylor schemes.

This enables improved convergence:

- Strong convergence of order 1 (Milstein)
- Weak convergence of order 2 (simplified 2.0 weak Taylor)

COS Approximation Equations

We use the COS method to approximate conditional expectations:

$$\mathbb{E}_n^x\left[v(t_{n+1}, X_{n+1})\right] \approx \sum_{k=0}^{K-1}{}' \mathcal{V}_k(t_{n+1})\, \mathrm{Re}\left\{\varphi_{X_{n+1}}\left(\frac{k\pi}{b-a}\middle| t_n, x\right) \exp\left(-i\frac{k\pi a}{b-a}\right)\right\}, \tag{14}$$

where the Fourier cosine expansion coefficients are

$$\begin{aligned} \mathcal{V}_k(t_{n+1}) &= \frac{2}{b-a}\int_a^b v(t_{n+1}, x)\cos\left(\frac{k\pi(x-a)}{b-a}\right)\mathrm{d}x. \\ v(t_{n+1}, x) &= \sum_{k=0}^{\infty}{}' \mathcal{V}_k(t_{n+1})\cos\left(\frac{k\pi(x-a)}{b-a}\right). \end{aligned}$$

Here, the notation $\sum{}'$ indicates that the $k = 0$ term has weight $1/2$.

When we don't know the ChF, Decoupled Case

Rewrite the Euler, Milstein, and 2.0 weak Taylor discretization schemes:

$$X_{n+1} = x + \bar{m}(x)\Delta t + \bar{s}(x)\Delta W_{n+1} + \bar{\kappa}(x)(\Delta W_{n+1})^2, \quad X_m = x.$$

When we don't know the ChF, Decoupled Case

Rewrite the Euler, Milstein, and 2.0 weak Taylor discretization schemes:

$$X_{n+1} = x + \bar{m}(x)\Delta t + \bar{s}(x)\Delta W_{n+1} + \bar{\kappa}(x)(\Delta W_{n+1})^2, \quad X_m = x.$$

For the Euler scheme: $\bar{m}(x) = \mu(x), \quad \bar{s}(x) = \sigma(x), \quad \bar{\kappa}(x) = 0.$

When we don't know the ChF, Decoupled Case

Rewrite the Euler, Milstein, and 2.0 weak Taylor discretization schemes:

$$X_{n+1} = x + \bar{m}(x)\Delta t + \bar{s}(x)\Delta W_{n+1} + \bar{\kappa}(x)(\Delta W_{n+1})^2, \quad X_m = x.$$

For the Euler scheme: $\bar{m}(x) = \mu(x), \quad \bar{s}(x) = \sigma(x), \quad \bar{\kappa}(x) = 0.$

For the Milstein scheme:

$$\bar{m}(x) = \mu(x) - \frac{1}{2}\sigma\sigma_x(x), \quad \bar{s}(x) = \sigma(x), \quad \bar{\kappa}(x) = \frac{1}{2}\sigma\sigma_x(x).$$

For the order 2.0 weak Taylor scheme:

When we don't know the ChF, Decoupled Case

Rewrite the Euler, Milstein, and 2.0 weak Taylor discretization schemes:

$$X_{n+1} = x + \bar{m}(x)\Delta t + \bar{s}(x)\Delta W_{n+1} + \bar{\kappa}(x)(\Delta W_{n+1})^2, \quad X_m = x.$$

For the Euler scheme: $\bar{m}(x) = \mu(x), \quad \bar{s}(x) = \sigma(x), \quad \bar{\kappa}(x) = 0.$

For the Milstein scheme:

$$\bar{m}(x) = \mu(x) - \frac{1}{2}\sigma\sigma_x(x), \quad \bar{s}(x) = \sigma(x), \quad \bar{\kappa}(x) = \frac{1}{2}\sigma\sigma_x(x).$$

For the order 2.0 weak Taylor scheme:

$$\bar{m}(x) = \mu(x) - \frac{1}{2}\sigma\sigma_x(x) + \frac{1}{2}\left(\mu\mu_x(x) + \frac{1}{2}\mu_{xx}\sigma^2(x)\right)\Delta t,$$

$$\bar{s}(x) = \sigma(x) + \frac{1}{2}\left(\mu_x\sigma(x) + \mu\sigma_x(x) + \frac{1}{2}\sigma_{xx}\sigma^2(x)\right)\Delta t,$$

$$\bar{\kappa}(x) = \frac{1}{2}\sigma\sigma_x(x).$$

Discrete Characteristic Function

$$X_{n+1} = x + \bar{m}(x)\Delta t + \bar{\kappa}(x)\left(\Delta W_{n+1} + \frac{1}{2}\frac{\bar{s}(x)}{\bar{\kappa}(x)}\right)^2 - \frac{1}{4}\frac{\bar{s}^2(x)}{\bar{\kappa}(x)}$$
$$\stackrel{d}{=} x + \bar{m}(x)\Delta t - \frac{1}{4}\frac{\bar{s}^2(x)}{\bar{\kappa}(x)} + \bar{\kappa}(x)\Delta t\left(U_{n+1} + \sqrt{\lambda(x)}\right)^2,$$

$\lambda(x) := \frac{1}{4}\frac{\bar{s}^2(x)}{\bar{\kappa}^2(x)\Delta t}$, $U_{n+1} \sim \mathcal{N}(0,1)$.

$(U_{n+1} + \sqrt{\lambda(x)})^2 \sim \chi_1'^2(\lambda(x))$ non-central chi-squared distributed.

The characteristic function of X_{n+1}, given $X_n = x$

$$\varphi(u|X_n = x) = \mathbb{E}\left[\exp\left(iuX_{n+1}\right)\Big|X_n = x\right]$$
$$= \exp\left(iux + iu\bar{m}(x)\Delta t - \frac{\frac{1}{2}u^2\bar{s}^2(x)\Delta t}{1 - 2iu\bar{\kappa}(x)\Delta t}\right)(1 - 2iu\bar{\kappa}(x)\Delta t)^{-1/2}.$$

Simplified 2.0 Weak Taylor Scheme, Coupled Case

- Explicit strategy (Huijskens et al., 2017; Negyesi et al. 2025): at step n use the already computed coefficients at t_{n+1}:

$$\xi(x) := y_{n+1}(x), \quad \eta(x) := z_{n+1}(x).$$

- Works for any forward discretisation once $\partial_x \xi$, $\partial_x^2 \xi$ are available

$\Rightarrow$ The forward discretization for the simplified weak 2.0 scheme reads:

$$X_{n+1} = x + \bar{m}_{\xi,\eta}(t_n, x)\Delta t + \bar{s}_{\xi,\eta}(t_n, x)\Delta W_n + \bar{\kappa}_{\xi,\eta}(t_n, x)(\Delta W_n)^2 \quad (28)$$

- The coefficients are:

$$\begin{aligned} \bar{m}_{\xi,\eta}(t,x) = \bar{\mu}_{\xi,\eta}(t,x) - \bar{\sigma}_{\xi,\eta}(t,x)\partial_x \bar{\sigma}_{\xi,\eta}(t,x)/2 \\ + (\partial_t \bar{\mu}_{\xi,\eta}(t,x) + \bar{\mu}_{\xi,\eta}(t,x)\partial_x \bar{\mu}_{\xi,\eta}(t,x) \\ + \partial_{xx}^2 \bar{\mu}_{\xi,\eta}(t,x)(\bar{\sigma}_{\xi,\eta}(t,x))^2/2)\Delta t_n/2, \quad \text{etc.} \end{aligned} \quad (9)$$

Derivatives via Fourier Cosine Expansion

The decoupling fields $\xi(t_{n+1}, x)$ and $\eta(t_{n+1}, x)$ are approximated as:

$$\xi(t_{n+1}, x) = \sum_{k=0}^{K-1}{}' \mathcal{Y}_k(t_{n+1}) \cos\left(\frac{k\pi(x-a)}{b-a}\right), \tag{1}$$

$$\eta(t_{n+1}, x) = \sum_{k=0}^{K-1}{}' \mathcal{Z}_k(t_{n+1}) \cos\left(\frac{k\pi(x-a)}{b-a}\right) \tag{2}$$

Their derivatives are available analytically:

$$\partial_x \xi(t_{n+1}, x) = -\sum_{k=0}^{K-1}{}' \left(\frac{k\pi}{b-a}\right) \mathcal{Y}_k(t_{n+1}) \sin\left(\frac{k\pi(x-a)}{b-a}\right)$$

$$\partial_{xx} \xi(t_{n+1}, x) = -\sum_{k=0}^{K-1}{}' \left(\frac{k\pi}{b-a}\right)^2 \mathcal{Y}_k(t_{n+1}) \cos\left(\frac{k\pi(x-a)}{b-a}\right)$$

Algorithm : BCOS Method

Goal: Compute (Y_0, Z_0) from the terminal condition $Y_T = g(X_T)$ using a Fourier-based approach.

Step 1. Initialize $y_N = g$, $z_N = 0$ on grid $\mathcal{X}_N \subset [a, b]$.

Step 2. For $n = N - 1, \ldots, 0$:

- Set $\xi := y_{n+1}$, $\eta := z_{n+1}$
- Compute COS coefficients of y_{n+1}, z_{n+1}
- Compute derivatives $\partial_x y_{n+1}, \partial_x^2 y_{n+1}$ from coefficients
- Evaluate $\bar{m}, \bar{s}, \bar{\kappa}$
- Use COS method to evaluate:

$$y_n(x) = \mathbb{E}_x \left[y_{n+1}(X_{n+1}) + f(\cdots)\Delta t \right], \tag{3}$$

$$z_n(x) = \mathbb{E}_x \left[y_{n+1}(X_{n+1})\Delta W_n \right] / \Delta t \tag{4}$$

Step 3. Output: $Y_0 = y_0(x_0)$, $Z_0 = z_0(x_0)$.

Derivation: COS Approximation of $\mathbb{E}[(\Delta W_n)^k v(X_{n+1})]$

Suppose:

$$\rightarrow h(t_{n+1}, X_{n+1}) = \sum_{k=0}^{K-1}{}' \hat{\mathcal{H}}_k(t_{n+1}) \cos\left(\frac{k\pi(X_{n+1}-a)}{b-a}\right)$$

Define:

$$J_k(x|u) := \mathbb{E}_n^x\left[e^{iuX_{n+1}}(\Delta W_n)^k\right]$$

Then, via integration by parts and recurrence:

$$J_k(x|u) = \frac{iu\bar{s}\Delta t_n}{1-2iu\bar{\kappa}\Delta t_n} J_{k-1}(x|u) + 1_{k\geq 1}\frac{(k-1)\Delta t_n}{1-2iu\bar{\kappa}\Delta t_n} J_{k-2}(x|u)$$

With this recursion, we find expressions for

$\mathbb{E}_n^x[h(t_{n+1}, X_{n+1})]$, $\mathbb{E}_n^x[h(t_{n+1}, X_{n+1})\Delta W_n]$ and $\mathbb{E}_n^x[h(t_{n+1}, X_{n+1})(\Delta W_n)^2]$

Coefficient Recovery and Derivatives via DCT

$$\to h(t_{n+1}, x) = {\sum_{k=0}^{\infty}}' \mathcal{H}(t_{n+1})_k \cos\left(\frac{k\pi(x-a)}{b-a}\right)$$

$$\mathcal{H}(t_{n+1}) = \frac{2}{a-b}\int_a^b h(t_{n+1}, x)\cos\left(\frac{k\pi(x-a)}{b-a}\right)\mathrm{d}x;$$

Given a truncated integration range $[a, b]$ and grid:

$$x_l = a + \left(l + \tfrac{1}{2}\right)\frac{b-a}{K}, \quad l = 0, \ldots, K-1,$$

approximate COS coefficients $\mathcal{H}_k$ of $h(\cdot)$ by a Discrete Cosine Transform:

$$\mathcal{H}_k(t_{n+1}) \approx \frac{2}{K}\sum_{l=0}^{K-1} h(t_{n+1}, x_l)\cos\left(\frac{k\pi(2l+1)}{2K}\right)$$

which is a DCT approximation of type 2.

BCOS Backward Algorithm

→ With the terminal condition, the backward recursion is implementable!

1. Initialise: COS coefficients of payoff $g(x)$ at $t = T$.
2. Loop $n = N - 1 \to 0$
   1. Evaluate $\mathbb{E}[Y], \mathbb{E}[Y\Delta W], \mathbb{E}[f], \mathbb{E}[f\Delta W]$ via COS.
   2. Update Z_n, then Y_n using θ-scheme.
   3. Recover new COS coefficients with DCT.
3. Output (Y_0, Z_0).

Runs in highly efficiently in Python and MATLAB.

⇒ Balint's code is open souce!

See: https://github.com/balintnegyesi/coupled-BCOS

Computational Complexity of the BCOS Method

The computational cost of the BCOS approximation is given by:

$$\mathcal{O}\left(N\left(K+K^2+PK+K\log K\right)\right),$$

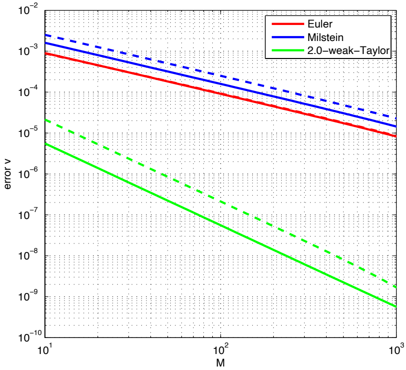
where:

- K: number of COS coefficients used in the spectral expansion;
- $\mathcal{O}(K^2)$: cost for coefficient recovery of the backward component;
- $\mathcal{O}(K\log K)$: cost of computing the COS coefficients via a Discrete Cosine Transform (DCT);
- $\mathcal{O}(PK)$: cost from applying P Picard iterations per time step;
- the outer factor $\mathcal{O}(N)$: accounts for the total number of time steps over the time grid.

Despite higher-order terms, the method remains efficient due to small N typically required, and the availability of fast transforms (DCT).

Example 1 – Bermudan Put (CEV), vol-term X^γ

- Decoupled BSDE!
- CEV parameter $\gamma = 0.2$ and 0.8, 10 exercise dates.
- 2.0-WT error $\leq 10^{-5}$ with $M{=}20$ timesteps. Euler requires ≈ 900.
- Observe 2nd-order slope for both Y_0 and Z_0.



Example 2 – Fully coupled linear–quadratic regulator

Stochastic optimal control

$$\mu(t,x,y,z) = \left(A - B\frac{R_{xu}}{R_u}\right)x + \frac{B^2}{R_u}y + \frac{BD}{R_u}z + \beta,$$

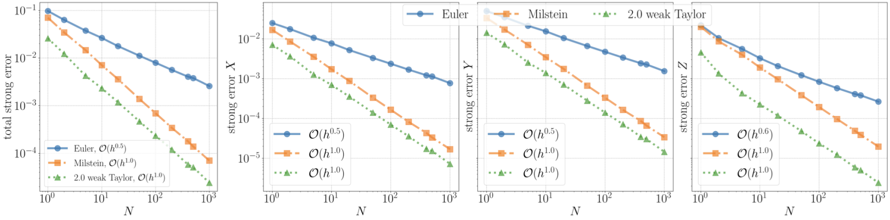
$$\sigma(t,x,y,z) = \left(C - D\frac{R_{xu}}{R_u}\right)x + \frac{DB}{R_u}y + \frac{D^2}{R_u}z + \Sigma, \quad g(x) = -Gx,$$

$$f(t,x,y,z) = \left(A - \frac{BR_{xu}}{R_u}\right)y + \left(C - \frac{DR_{xu}}{R_u}\right)z - \left(R_x - \frac{R_{xu}}{R_u}\right)x.$$

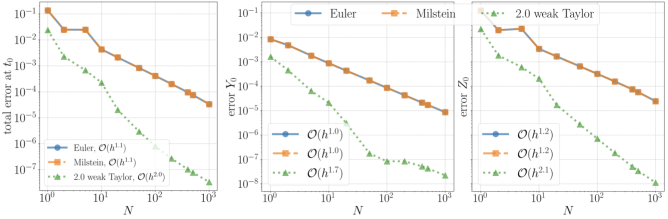
Coupling enters both drift and diffusion via optimal control u_t.

$$A=-1,\ B=0.1,\ \beta=0, C=1, D=0.01,\ \Sigma=0.05,$$
$$R_x=2, R_{xu}=0, R_u=2, G=2.$$

Closed-form solution from Riccati ODE available.



(a) strong convergence



(b) weak convergence at t_0

Euler vs. Milstein

Number of time steps N, cosine terms K.

$N \setminus K$	strong error X			strong error Y			strong error Z		
	128	512	1024	128	512	1024	128	512	1024
10	7.7e−3	7.7e−3	7.7e−3	1.5e−2	1.5e−2	1.5e−2	3.3e−3	3.3e−3	3.3e−3
100	2.4e−3	2.4e−3	2.4e−3	4.7e−3	4.7e−3	4.7e−3	8.3e−4	8.3e−4	8.3e−4
400	1.2e−3	1.2e−3	1.2e−3	2.4e−3	2.4e−3	2.4e−3	4.1e−4	4.1e−4	4.1e−4
1000	7.7e−4	7.7e−4	7.7e−4	1.5e−3	1.5e−3	1.5e−3	2.7e−4	2.7e−4	2.7e−4

(a) Euler with (7)

$N \setminus K$	strong error X			strong error Y			strong error Z		
	128	512	1024	128	512	1024	128	512	1024
10	1.7e−3	1.7e−3	1.7e−3	3.5e−3	3.5e−3	3.5e−3	1.9e−3	1.9e−3	1.9e−3
100	1.7e−4	1.7e−4	1.7e−4	3.3e−4	3.3e−4	3.3e−4	1.9e−4	1.9e−4	1.9e−4
400	4.3e−5	4.3e−5	4.3e−5	8.7e−5	8.7e−5	8.7e−5	4.9e−5	4.9e−5	4.9e−5
1000	1.7e−5	1.7e−5	1.7e−5	3.4e−5	3.4e−5	3.4e−5	2.0e−5	2.0e−5	2.0e−5

(b) Milstein with (8)

Conclusions & Outlook

⇒ Discrete characteristic function + COS gives a **fast, 2nd-order** BSDE solver for scalar FBSDEs.

- The challenge is the multi-D generalization!

References:

B. Negyesi, C.W. Oosterlee, A numerical Fourier cosine expansion method with higher order Taylor schemes for fully coupled FBSDEs. ArXiv 2025

T. P. Huijskens, M. J. Ruijter, C. W. Oosterlee. "Efficient numerical Fourier methods for coupled forward–backward SDEs". en. In: Journal of Computational and Applied Mathematics 296 (Apr. 2016), pp. 593–612.

W. Zhao, Y. Li, G. Zhang, "A generalized theta-scheme for solving backward stochastic differential equations". In: Discrete Continuous Dynamical Systems - B 17.1531-3492 (2012), pp. 1585– 1585.