Journal of Computational Finance **22**(4), 25–71
DOI: 10.21314/JCF.2018.361

Research Paper

ε-monotone Fourier methods for optimal stochastic control in finance

Peter A. Forsyth and George Labahn

David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue, Waterloo, Ontario N2L 3G1, Canada; emails: paforsyt@uwaterloo.ca, glabahn@uwaterloo.ca

(Received October 25, 2017; revised April 3, 2018; accepted May 27, 2018)

ABSTRACT

Stochastic control problems in finance often involve complex controls at discrete times. As a result, numerically solving such problems using, for example, methods based on partial differential or integrodifferential equations inevitably gives rise to low-order (usually at most second-order) accuracy. In many cases, Fourier methods can be used to efficiently advance solutions between control monitoring dates, and numerical optimization methods can then be applied across decision times. However, Fourier methods are not monotone, and as a result they give rise to possible violations of arbitrage inequalities. This is problematic in the context of control problems, where the control is determined by comparing value functions. In this paper, we give a preprocessing step for Fourier methods that involves projecting the Green's function onto the set of linear basis functions. The resulting algorithm is guaranteed to be monotone (to within a tolerance), ℓ_∞-stable and satisfies an ε-discrete comparison principle. In addition, the algorithm has the same complexity per step as a standard Fourier method and second-order accuracy for smooth problems.

Keywords: monotonicity; Fourier methods; discrete comparison; optimal stochastic control; finance.

Corresponding author: P. A. Forsyth

Print ISSN 1460-1559 | Online ISSN 1755-2850

1 INTRODUCTION

Optimal stochastic control problems in finance often involve monitoring or making decisions at discrete points in time. These monitoring times typically cause difficulties when solving optimal stochastic control problems numerically, both for efficiency and correctness. This is because, for efficiency, numerical methods are typically applied from one monitoring time to the next; correctness arises as an issue when the decision is determined by comparing value functions, which is somewhat problematic when discrete approximations are not monotone. These optimal stochastic problems arise in many important financial applications and include problems such as asset allocation (Cong and Oosterlee 2016; Forsyth and Vetzal 2017; Huang 2010; Li and Ng 2000), pricing of variable annuities (Alonso-Garcia *et al* 2018; Bauer *et al* 2008; Chen *et al* 2008; Dai *et al* 2008; Huang *et al* 2017; Ignatieva *et al* 2018) and hedging in discrete time (Angelini and Herzel 2014; Remillard and Rubenthaler 2013).

These optimal control problems are typically modeled as the solutions of partial integrodifferential equations (PIDEs), which can be solved via numerical, finite-difference (Chen *et al* 2008) or Monte Carlo (Cong and Oosterlee 2016) methods. When cast into dynamic programming form, the optimal control problem reduces to solving a PIDE backward in time between each decision point, and then determining the optimal control at each such point. In many cases, including those mentioned above, the models are based on fairly simple stochastic processes, with the main interest being the behavior of the optimal controls. These parsimonious stochastic models can be justified if we are looking at long-term problems, eg, variable annuities or saving for retirement, where the time scales are of the order of ten to thirty years. In these situations, it is reasonable to use a parsimonious stochastic process model.

In these (and many other) situations, the characteristic function of the associated stochastic process is known in closed form. For the PIDE types that appear in financial problems, knowing the characteristic function implies that the Fourier transform of the solution is also known in closed form. By discretizing these Fourier transforms, we obtain an approximation to the solution, which can be used for effective numerical computation. A natural approach in this case is to use a Fourier scheme to advance the solution in a single time step between decision times, and then to apply a numerical optimization approach to advance the solution across the decision time. This technique is repeated until the current time is reached (Lippa 2013; Ruijter *et al* 2013). These methods are based on Fourier space time-stepping (FST) (Jackson *et al* 2008), the convolution (CONV) technique (Lord *et al* 2008) or the Fourier cosine (COS) algorithm (Fang and Oosterlee 2008). Fourier methods have been applied to, among other things, the pricing of exotic variance products and volatility derivatives

(Zheng and Kwok 2014), guaranteed minimum withdrawal benefits (Alonso-Garcia *et al* 2018; Huang *et al* 2017; Ignatieva *et al* 2018) and equity-indexed annuities (Deng *et al* 2017).

Fourier methods have a number of advantages over finite-difference and other methods; primarily, there are no time-stepping errors between decision dates. These methods also provide easy handling for stochastic processes involving jump diffusion (Lippa 2013) and regime switching (Jackson *et al* 2008). Although Fourier methods typically need a large number of discretization points, the algorithms reduce to using finite fast Fourier transforms (FFTs) that operate efficiently on most platforms (including graphics processing units). The algorithms are also quite easy to implement. For example, using Fourier methods for the pricing of variable annuities reduces to the use of discrete FFTs and local optimization. A detailed knowledge of partial differential equation algorithms is not actually required in this case. Fourier methods also easily extend to multifactor stochastic processes, where finite-difference methods have difficulties because of cross-derivative terms. Of course, Fourier methods suffer from the curse of dimensionality, and hence are restricted, except in special cases, to problems of dimension 3 or less. Finally, Fourier methods have good convergence properties for problems with noncomplex controls. For example, for European option pricing, in cases where the characteristic function of the underlying stochastic process is known, the COS method achieves exponential convergence (with regard to the number of terms in the Fourier series) (Fang and Oosterlee 2008).

A major drawback with current Fourier methods is that they are not monotone. In the contingent claims context, monotone methods preserve arbitrage inequalities, or discrete comparison properties, independent of any discretization errors. As a concrete example, consider the case of a variable annuity contract, with ratchet features and withdrawal controls at each decision date. Suppose contract A has a larger payoff at the terminal time than contract B. Then a monotone scheme generates a value for contract A that is always larger than the value of contract B, at all points in time and space, regardless of the accuracy of the numerical scheme. In a sense, the arbitrage inequality (discrete comparison) condition is the financial equivalence of conservation of mass in engineering computations. Use of nonmonotone methods is especially problematic in the context of control problems, where the control is determined by comparing value functions.

Monotonicity is also relevant for the convergence of numerical schemes. In general, optimal control problems posed as PIDEs are nonlinear and do not have unique solutions. The financially relevant solution is the viscosity solution of the PIDEs, and it is well known that a discretization of a PIDE converges to the viscosity solution if it is monotone, consistent and stable (Barles and Souganidis 1991). There are examples where nonmonotone discretizations fail to converge (Obermann 2006) and also examples where there is convergence (Pooley *et al* 2003) but not to the financially correct viscosity solution. In addition, in cases where the Green's function has a thin peak, existing nonmonotonic Fourier methods require a very small space step that often results in numerical issues. Finally, monotone schemes are more reliable for the numerical computation of Greeks (ie, derivatives of the solution), which is often important for financial instruments.

The starting point for this paper is the assumption that we have a closed-form representation of the Fourier transform of the Green's function of the stochastic process PIDE. From a practical point of view, we also assume that a spatial shift property holds. The latter assumption can be removed but at the cost of increasing the computational complexity of our method. We will discuss these assumptions further below.

In this paper, we present a new Fourier algorithm in which monotonicity can be guaranteed to within a user-specified numerical tolerance. The algorithm is for use with general optimal control problems in finance. In these general control problems, the objective function may be complex and nonsmooth; hence the optimal control at each step must be determined by a numerical optimization procedure. Indeed, in many cases, this is done by discretizing the control and using exhaustive search. Reconstructing the Fourier coefficients is typically done by assuming the control is constant over discretized intervals of the physical space, by numerically determining the control at the midpoint of these intervals and, finally, by reconstructing the Fourier coefficients by quadrature. This is equivalent to using a type of trapezoidal rule to reconstruct the Fourier coefficients, and hence this can be at most second-order accurate (in terms of the physical domain mesh size).

In fact, we show how the FST or CONV schemes can be modified to get new schemes in which monotonicity can be guaranteed to within a user-specified numerical tolerance. Our approach is similar to that used in these schemes, which first approximate the solution of a linear PIDE by a Green's function convolution, then discretize the convolution and finally carry out the dense matrix–vector multiplication efficiently using an FFT. In our case, we discretize the value function and generate a continuous approximation of the function by assuming linear basis (or alternatively piecewise constant) functions. Given this approximation, we carry out an exact integration of the convolution integral and then truncate the series approximation of this integral so that monotonicity holds to within a certain tolerance. Consequently, we prove that our algorithm has an ε-discrete comparison property, that is, given a tolerance ε, a discrete comparison (also called arbitrage inequality) holds to $O(\varepsilon)$, independent of the number of discretization nodes and time steps. This is similar in spirit to the ε-monotone schemes discussed in, for example, Bokanowski *et al* (2018). Typically, the convergence to the integral is exponential in the series truncation parameter, so it is inexpensive to specify a small tolerance. The key idea here is that the number of terms required to accurately determine the projection of the Green's function onto linear basis functions can be larger than the number of basis functions. After an initial setup cost, the complexity per step is the same as for the standard FST or CONV methods. This requires only a small change to existing codes in order to guarantee monotonicity. The desirable property of our method is that monotonicity can be guaranteed (to within a certain tolerance) independent of the number of FST (CONV) grid nodes or the time-step size.

While Fourier methods have good convergence properties for vanilla contracts or problems where controls are smooth, it is a different story for general optimal control problems. For example, if the COS method is applied to optimal control problems, then it is challenging to maintain exponential convergence, as the optimal control must be determined in the physical space. Hence, a highly accurate recursive expression for the Fourier coefficients must be found after application of the optimal controls in order to maintain exponential convergence. In the case of bang-bang controls, it is often possible to separate the physical domain into regions where the control is constant. If these regions are determined to high accuracy, then an accurate algorithm for recursive generation of the Fourier coefficients can be developed (Ruijter *et al* 2013). However, even for the case of an American option, this requires careful analysis and implementation (Fang and Oosterlee 2008). Our interest is in general problems, where the control may not be of the bang-bang type, and we expect that such good convergence properties will not hold. In addition, in the path-dependent case, the problem is usually converted to Markovian form through additional state variables. The dynamics of these state variables are typically represented by a deterministic equation (between monitoring dates). At monitoring dates, the state variable may have nonsmooth jumps (eg, cashflows), and hence the standard approach would be to discretize this state variable and then to interpolate the value function across the monitoring dates. If linear interpolation is used, this also implies that the solution is at most second-order accurate at a monitoring date.

While monotone schemes have good numerical properties, they appear to be inherently low-order methods. However, it would seem that in the most general case it is difficult to develop high-order schemes for control problems. For example, in the COS method, this difficulty can be traced to that of reconstructing the Fourier coefficients after numerically determining the optimal control at discrete points in the physical space. Consequently, in this paper we focus on FST or CONV techniques, which use straightforward procedures to move between Fourier space and the physical space (and vice versa).

We illustrate the behavior of our algorithm by comparing various implementations of FST/CONV on some model option-pricing examples, particularly European and Bermudan options. In addition, we demonstrate the use of the monotone scheme methods on a realistic asset allocation problem. Our main conclusion is that for problems with complex controls, where we can expect fairly low-order convergence to the solution, a small change to standard FST or CONV methods can be made that guarantees monotonicity, at least to within a user-specified tolerance. This does not alter the order of convergence in this case; hence, we can ensure a monotone scheme with only a slightly increased setup cost. After the initialization, the complexity per step of the monotone method is the same as that of the standard FST/CONV algorithm.

The remainder of this paper is as follows. In the next section, we describe our optimal control problem in a general setting. Section 3 describes existing Fourier methods and contrasts them with our new monotone Fourier method presented in Section 4. The monotone algorithm for solving optimal control problems is then given in Section 5, with properties of the algorithm and proofs given in Section 6. Wraparound is an important issue for Fourier methods, particularly in the case of our control problems. Our method of minimizing such an error is described in Section 7. Section 8 presents two numerical examples used to stress test the monotone algorithm. This is followed by an application of our algorithm to the multiperiod mean–variance optimal asset allocation problem, a general optimal control problem well suited to our monotone methods. The paper ends with our conclusions and topics for future research.

2 GENERAL CONTROL FORMULATION

In this section, we describe our optimal control problem in a general setting. Consider a set of intervention (or monitoring) times t_n

$$\hat{\mathcal{T}} \equiv \{t_0 \leqslant \cdots \leqslant t_M\}, \tag{2.1}$$

with $t_0 = 0$ the inception time of the investment and $t_M = T$ the terminal time. For simplicity, we specify the set of intervention times to be equidistant, that is, $t_n - t_{n-1} = \Delta t = T/M$ for each n.

Let $t_n^- = t_n - \varepsilon$ and $t_n^+ = t_n + \varepsilon$ (with $\varepsilon \to 0^+$) denote the instants before and after the nth monitoring time t_n. We define a value function $\hat{v}(x,t)$ with domain $x \in \mathbb{R}$ (we restrict our attention to one-dimensional problems for ease of exposition) that satisfies

$$\hat{v}_t + \mathcal{L}\hat{v} = 0, \quad t \in (t_n^+, t_{n+1}^-), \tag{2.2}$$

with $\mathcal{L}$ a partial integrodifferential operator. At $t_n \in \hat{\mathcal{T}}$ we find an optimal control $\hat{c}(x,t_n)$ via

$$\hat{v}(x,t_n^-) = \inf_{\hat{c} \in \mathbb{Z}} \mathcal{M}(\hat{c})\hat{v}(x,t_n^+), \tag{2.3}$$

where $\mathcal{M}(\hat{c})$ is an intervention operator and $\mathbb{Z}$ is the set of admissible controls.

It is more natural to rewrite these equations going backward in time, $\tau = T - t$, that is, in terms of time to completion. In this case, the value function is $v(x, \tau) = \hat{v}(x, T - t)$ and satisfies

$$v_\tau - \mathcal{L}v = 0, \quad \tau \in (\tau_n^+, \tau_{n+1}^-), \tag{2.4}$$

$$v(x, \tau_n^+) = \inf_c \mathcal{M}(c)v(x, \tau_n^-), \quad \tau_n \in \mathcal{T}. \tag{2.5}$$

Here the control $c(x, \tau) = \hat{c}(x, T - \tau)$, and $\mathcal{T}$ now refers to the set of backward intervention times

$$\mathcal{T} \equiv \{\tau_0 \leqslant \cdots \leqslant \tau_M\} \quad \text{with } \tau_0 = 0, \quad \tau_M = T \quad \text{and} \quad \tau_n = T - t_{M-n}.$$

A typical intervention operator has the form

$$\mathcal{M}(c)v(x, \tau_n^-) = v(x + \Gamma(x, \tau_n^-, c), \tau_n^-). \tag{2.6}$$

As an example, in the context of portfolio allocation, we can interpret $\Gamma(x, \tau_i^-, c)$ as a rebalancing rule. In general, there can also be cashflows associated with the decision process, as in the case of variable annuities. However, for simplicity, we will ignore such a generalization in this paper, and we will instead assume that the intervention operator has the form (2.6). In our asset allocation example (described later), the cashflows are modeled by updating a path-dependent variable.

3 CONVOLUTION AND FOURIER SPACE TIME-STEPPING METHODS

In this section, we derive the FST and the closely related CONV technique in an intuitive fashion. This will allow us to contrast these methods with the monotone technique developed in the next section. For ease of exposition, we will continue to restrict our attention to one-dimensional problems. However, there is no difficulty generalizing this approach to the multidimensional case. In a financial context, it is often the case that the variable $x = \log(S) \in (-\infty, \infty)$, where S is an asset price.

3.1 Green's functions

A solution of the PIDE (2.4),

$$v_\tau - \mathcal{L}v = 0, \quad \tau \in (\tau_n, \tau_{n+1}],$$

can be represented in terms of the Green's function of the PIDE, a function typically of the form $g = g(x, x', \Delta\tau)$. However, in many cases this will have the form $g = g(x - x', \Delta\tau)$; we will assume this to hold in our problems. More formally, we make the following assumptions, which we assume hold in the rest of this work.

ASSUMPTION 3.1 (The form of Green's function) *The Green's function can be written as*

$$g(x, x', \Delta\tau) = g(x - x', \Delta\tau) = \int_{-\infty}^{\infty} G(\omega, \Delta\tau) \mathrm{e}^{2\pi \mathrm{i}\omega(x-x')} \, \mathrm{d}\omega, \tag{3.1}$$

where $G(\omega, \Delta\tau)$ is known in closed form, and $G(\omega, \Delta\tau)$ is independent of (x, x').

REMARK 3.2 If we view the Green's function in Assumption 3.1 as a scaled conditional probability density f, then our assumption is that $f(y \mid x)$ only depends on x and y via their difference $f(y \mid x) = f(y - x)$. This assumption holds for Lévy processes (independent and stationary increments), but it does not hold, for example, for a Heston stochastic volatility model or for mean-reverting Ornstein–Uhlenbeck processes (but see Surkov (2010), Zhang *et al* (2012) and Shan (2014) for possible workarounds). The ε-monotonicity modifications described in this paper also hold when we do not have $g(x, x', \Delta\tau) = g(x - x', \Delta\tau)$, but at the price of reduced efficiency. This is discussed later, in Section 4.2. The second assumption, that we know the Fourier transform of our Green's function in closed form, holds, for example, in situations where the characteristic function of the underlying stochastic process is known. In the case of Lévy processes, the Lévy–Khintchine formula provides such an explicit representation of the characteristic function.

From Assumption 3.1, the exact solution of our PIDE is then

$$v(x, \tau + \Delta\tau) = \int_{\mathbb{R}} g(x - x', \Delta\tau) v(x', \tau) \, \mathrm{d}x'. \tag{3.2}$$

The Green's function has a number of important properties (Garroni and Menaldi 1992). For this work, the two properties

$$\int_{\mathbb{R}} g(x, \Delta\tau) \, \mathrm{d}x = C_1 \leqslant 1 \quad \text{and} \quad g(x, \Delta\tau) \geqslant 0 \tag{3.3}$$

are particularly important.[1] These properties are formally proven in Garroni and Menaldi (1992), but they can also be deduced from the interpretation of the Green's function as a scaled probability density.

We define the Fourier transform pair for the Green's function as

$$\left.\begin{aligned} G(\omega, \Delta\tau) &= \int_{-\infty}^{\infty} g(x, \Delta\tau) \mathrm{e}^{-2\pi \mathrm{i}\omega x} \, \mathrm{d}x, \\ g(x, \Delta\tau) &= \int_{-\infty}^{\infty} G(\omega, \Delta\tau) \mathrm{e}^{2\pi \mathrm{i}\omega x} \, \mathrm{d}\omega, \end{aligned}\right\} \tag{3.4}$$

with a closed-form expression for $G(\omega, \Delta\tau)$ being available.

[1] For the examples in this paper the constant C_1 is explicitly given (in each example) in Section 3.1.1. C_1 is a constant independent of x.

As is typically the case, we assume that the Green's function $g(x, \Delta\tau)$ decays to zero as $|x| \to \infty$, ie, $g(x, \Delta\tau)$ is negligible outside a region $x \in [-A, A]$. Choosing $x_{\min} < -A$ and $x_{\max} > A$, we localize the computational domain for the integral in (3.2) so that $x \in [x_{\min}, x_{\max}]$. We can therefore replace the Fourier transform pair (3.4) by its Fourier series equivalent:

$$\left.\begin{aligned} G(\omega_k, \Delta\tau) &\simeq \int_{x_{\min}}^{x_{\max}} g(x, \Delta\tau) \mathrm{e}^{-2\pi \mathrm{i}\omega_k x}\, \mathrm{d}x, \\ \hat{g}(x, \Delta\tau) &= \frac{1}{P} \sum_{k=-\infty}^{\infty} G(\omega_k, \Delta\tau) \mathrm{e}^{2\pi \mathrm{i}\omega_k x}, \end{aligned}\right\} \tag{3.5}$$

with $P = x_{\max} - x_{\min}$ and $\omega_k = k/P$. Here the scaling factors in (3.5) are selected to be consistent with the scaling in (3.4). The solution of the PIDE (3.2) is then approximated as

$$v(x, \tau + \Delta\tau) \simeq \int_{x_{\min}}^{x_{\max}} \hat{g}(x - x', \Delta\tau) v(x', \tau)\, \mathrm{d}x'. \tag{3.6}$$

Note that the Fourier series (3.5) implies a periodic extension of $\hat{g}$, that is, $\hat{g}(x + P, \tau) = \hat{g}(x, \tau)$. The localization assumption also then implies that $v(x, \tau)$ is periodically extended.

Substituting the Fourier series (3.5) into (3.6) gives our approximate solution as

$$v(x, \tau + \Delta\tau) \simeq \frac{1}{P} \sum_{k=-\infty}^{\infty} G(\omega_k, \Delta\tau) \mathrm{e}^{2\pi \mathrm{i}\omega_k x} \int_{x_{\min}}^{x_{\max}} v(x', \tau) \mathrm{e}^{-2\pi \mathrm{i}\omega_k x'}\, \mathrm{d}x'. \tag{3.7}$$

Let $\Delta x = P/N$ and choose points $\{x_j\}$, $\{x'_j\}$ by

$$x_j = \hat{x}_0 + j\Delta x, \quad x'_j = \hat{x}_0 + j\Delta x \quad \text{for } j = -\tfrac{1}{2}N, \dots, \tfrac{1}{2}N - 1.$$

Then, the integral in (3.7) can be approximated by a quadrature rule with weights w_ℓ, giving

$$\begin{aligned} \int_{x_{\min}}^{x_{\max}} v(x', \tau) \mathrm{e}^{-2\pi \mathrm{i}\omega_k x'}\, \mathrm{d}x' &\simeq \sum_{\ell=-N/2}^{N/2-1} w_\ell v(x'_\ell, \tau) \exp\left(-2\pi \mathrm{i}\frac{k}{P} x'_\ell\right) \Delta x \\ &= P \exp\left(-2\pi \mathrm{i}\frac{k}{P} \hat{x}_0\right) V(\omega_k, \tau), \end{aligned} \tag{3.8}$$

where

$$V(\omega_k, \tau) = \frac{1}{N} \sum_{\ell=-N/2}^{N/2-1} w_\ell v(x'_\ell, \tau) \mathrm{e}^{-2\pi \mathrm{i} k\ell/N} \tag{3.9}$$

is the discrete Fourier transform (DFT) of $\{w_j v(x'_j, \tau)\}$. In the following, we will consider two cases for the weights w_ℓ: the trapezoidal rule and Simpson's quadrature. Substituting (3.8) and (3.9) into (3.7) and truncating the infinite sum to $k \in [-\frac{1}{2}N, \frac{1}{2}N - 1]$ then gives

$$\begin{aligned} v(x_j, \tau + \Delta\tau) &\simeq \frac{1}{P} \sum_{k=-N/2}^{N/2-1} \exp\left(2\pi \mathrm{i} \frac{k}{P} \hat{x}_0\right) G(\omega_k, \Delta\tau) \exp\left(\frac{2\pi \mathrm{i} kj}{N}\right) \\ &\qquad \times P \exp\left(-2\pi \mathrm{i} \frac{k}{P} \hat{x}_0\right) V(\omega_k, \tau) \\ &= \sum_{k=-N/2}^{N/2-1} G(\omega_k, \Delta\tau) V(\omega_k, \tau) \exp\left(\frac{2\pi \mathrm{i} kj}{N}\right). \end{aligned} \tag{3.10}$$

Thus, $\{v(x_j, \tau + \Delta\tau)\}$ is the inverse DFT of the product $\{G(\omega_k, \Delta\tau) V(\omega_k, \tau)\}$.

In summary, we can obtain a discrete set of values for the solution v by first going to the Fourier domain by constructing its Fourier transform V using a set of quadrature weights and then returning to the physical domain by convolution of V with the Fourier transform of the Green's function. The cost is then that of doing a single FFT and inverse FFT (iFFT).

There are four significant approximations in these steps: localization of the computational domain; representation of the Green's function by a truncated Fourier series; a periodic extension of the solution; and approximation of the integral in (3.7) by a quadrature rule. For details of the effect of the errors from these approximations we refer the reader to the discussion in Lord *et al* (2008).

3.1.1 Examples of Green's functions

We consider the generic PIDE (which would be typical of a problem where the underlying asset follows a jump diffusion):

$$v_\tau = \tfrac{1}{2}\sigma^2 v_{xx} + (\mu - \tfrac{1}{2}\sigma^2 - \lambda\kappa) v_x - (\rho + \lambda) v + \lambda \int_{-\infty}^{+\infty} v(x + y) f(y) \, \mathrm{d}y, \tag{3.11}$$

where σ, μ, λ, κ, ρ are constants, and $f(y)$ is the jump-size density. If, for example, $\rho = \mu = r$, where r is the risk-free rate, then this is the option-pricing equation, while if $\rho = 0$, then this PIDE arises in asset allocation.

Let

$$\left.\begin{aligned} v(x, \tau) &= \int_{-\infty}^{\infty} V(\omega, \tau) \mathrm{e}^{2\pi \mathrm{i}\omega x} \, \mathrm{d}\omega, \\ f(y, \tau) &= \int_{-\infty}^{\infty} F(\omega, \tau) \mathrm{e}^{2\pi \mathrm{i}\omega y} \, \mathrm{d}\omega. \end{aligned}\right\} \tag{3.12}$$

Substituting (3.12) into (3.11) gives

$$V(\omega,\tau)_\tau = \Psi(\omega)V(\omega,\tau), \tag{3.13}$$

where

$$\Psi(\omega) = (-\tfrac{1}{2}\sigma^2(2\pi\omega)^2 + (\mu - \lambda\kappa - \tfrac{1}{2}\sigma^2)(2\pi\mathrm{i}\omega) - (\rho+\lambda) + \lambda\bar{F}(\omega)), \tag{3.14}$$

with $\bar{F}(\omega)$ being the complex conjugate of $F(\omega)$. Integrating (3.13) gives

$$V(\omega,\tau+\Delta\tau) = \mathrm{e}^{\Psi(\omega)\Delta\tau}V(\omega,\tau),$$

from which we can deduce that the Fourier transform of the Green's function $G(\omega,\Delta\tau)$ is

$$G(\omega,\Delta\tau) = \mathrm{e}^{\Psi(\omega)\Delta\tau}. \tag{3.15}$$

Two common choices for jump-size density are the double exponential (with p_u, η_1, η_2 constants)

$$f(y) = p_u\eta_1\mathrm{e}^{-\eta_1 y}\mathbf{1}_{y\geqslant 0} + (1-p_u)\eta_2\mathrm{e}^{\eta_2 y}\mathbf{1}_{y<0} \tag{3.16}$$

and the lognormal (with γ, ν constants)

$$f(y) = \frac{1}{\sqrt{2\pi}\gamma}\exp\biggl(-\frac{(y-\nu)^2}{2\gamma^2}\biggr). \tag{3.17}$$

In the case of a double exponential jump distribution, we have

$$\bar{F}(\omega) = \frac{p_u}{1-2\pi\mathrm{i}\omega/\eta_1} + \frac{1-p_u}{1+2\pi\mathrm{i}\omega/\eta_2}, \tag{3.18}$$

while in the case of a lognormal jump-size distribution, we have

$$\bar{F}(\omega) = \exp(2(\pi\mathrm{i}\omega\nu - (\pi\omega\gamma)^2)). \tag{3.19}$$

From (3.13) and (3.15), we obtain

$$G(0,\Delta\tau) = \mathrm{e}^{-\rho\Delta\tau},$$

which means that in these cases $C_1 = \int_{\mathbb{R}} g(x,\Delta\tau)\,\mathrm{d}x$ becomes

$$C_1 = \begin{cases} \mathrm{e}^{-r\Delta\tau}, & \text{option pricing,} \\ 1, & \text{mean–variance asset allocation.} \end{cases}$$

3.2 The FST/CONV algorithms

The FST and CONV algorithms are described using the previous approximations. Let $(v^n)^+$ be the vector of solution values just after τ_n, and let w_{quad} be the vector of quadrature weights:

$$\left.\begin{aligned} (v^n)^+ &= [v(x_{-N/2}, \tau_n^+), \dots, v(x_{N/2-1}, \tau_n^+)], \\ w_{\text{quad}} &= [w(x_{-N/2}), \dots, w(x_{N/2-1})]. \end{aligned}\right\} \tag{3.20}$$

Furthermore, let $\mathbb{I}_{\Delta x}(x)$, $x_k \leqslant x \leqslant x_{k+1}$, be a linear interpolation operator:

$$\mathbb{I}_{\Delta x}(x)(v^n)^+ = \theta v(x_k, \tau_n^+) + (1-\theta) v(x_{k+1}, \tau_n^+), \quad \theta = \frac{(x_{k+1} - x)}{\Delta x}. \tag{3.21}$$

The full FST/CONV algorithm applied to a control problem is illustrated in Algorithm 3.3. We refer the reader to Lippa (2013), Ignatieva *et al* (2018) and Huang *et al* (2017) for applications in finance.

ALGORITHM 3.3 (FST/CONV Fourier method) $x \circ y$ is the Hadamard product of vectors x, y.

Require $G = \{G(\omega_j, \Delta\tau)\}$, $j = -\frac{1}{2}N, \dots, \frac{1}{2}N - 1$

```
Input: number of time steps $M$ and initial solution $(v^0)^-$
$(v^0)^+ = \inf_c \mathcal{M}(c)(\mathbb{I}_{\Delta x}(x)(v^0)^-)$
for $m = 1, \dots, M$ do {time-step loop}
    $V^{m-1} = \text{FFT}[w_{\text{quad}} \circ (v^{m-1})^+]$        {frequency domain}
    $(v^m)^- = \text{iFFT}[V^{m-1} \circ G]$        {physical domain}
    $v(x_j, \tau_m^+) = \inf_c \mathcal{M}(c)(\mathbb{I}_{\Delta x}(x_j)(v^m)^-)$   $j = -\frac{1}{2}N, \dots, \frac{1}{2}N - 1$,
                {optimal control}
end for
```

REMARK 3.4 In Jackson *et al* (2008), the authors describe their FST method in slightly different terms. There, they use a continuous Fourier transform to convert the PIDE into Fourier space. The PIDE in physical space then reduces to a linear first-order differential equation in Fourier space that can be solved in closed form (as in Section 3.1.1). In this way, the method is able to produce exact pricing results between monitoring dates (if any) of an option, using a continuous domain. In practice, using a discrete computational domain leads to approximations, as a discrete Fourier transform is used to approximate the continuous Fourier transform.

4 AN ε-MONOTONE FOURIER METHOD

Our monotone Fourier method proceeds in a similar fashion to the previous section, but it is based on a slightly different philosophy. We begin by discretizing the value function, and then generate a continuous approximation of the value function by assuming linear basis functions. Given this approximation, we carry out an exact integration of the convolution integral. We can then truncate the series approximation of this integral so that monotonicity holds to within a certain tolerance (using a truncation parameter to keep track of the number of terms). Typically, the convergence to the integral is exponential in the series truncation parameter, so it is inexpensive to specify a small tolerance. The key idea here is that the number of terms required to accurately determine the projection of the Green's function onto a given set of linear basis functions can be larger than the number of basis functions.

An additional important point is that, after the initial setup cost, the complexity per step is the same as for the standard FST and CONV methods. This requires only a small change to existing FST or CONV codes in order to guarantee monotonicity. The desirable property of this method is that monotonicity can be guaranteed (to within a small tolerance) independently of the number of FST (CONV) grid nodes or time-step size.

4.1 A monotone scheme

We proceed as follows. As before, we assume a localized computational domain:

$$v(x,\tau+\Delta\tau)=\int_{x_{\min}}^{x_{\max}} g(x-x',\Delta\tau)v(x',\tau)\,\mathrm{d}x' \tag{4.1}$$

and discretize this problem on the grid $\{x_j\}$, $\{x'_j\}$:

$$x_j=\hat{x}_0+j\Delta x,\quad x'_j=\hat{x}_0+j\Delta x,\quad j=-\tfrac{1}{2}N,\dots,\tfrac{1}{2}N-1,$$

where $P=x_{\max}-x_{\min}$ and $\Delta x=P/N$ with $x_{\min}=\hat{x}_0-\frac{1}{2}N\Delta x$ and $x_{\max}=\hat{x}_0+\frac{1}{2}N\Delta x$. Setting $v_j(\tau)=v(x_j,\tau)$, we can now represent the solution as a linear combination:

$$v(x,\tau)\simeq\sum_{j=-N/2}^{N/2-1}\phi_j(x)v(x_j,\tau)=\sum_{j=-N/2}^{N/2-1}\phi_j(x)v_j(\tau), \tag{4.2}$$

where the ϕ_j are piecewise linear basis functions, ie,

$$\phi_j(x)=\begin{cases}\dfrac{(x_{j+1}-x)}{\Delta x}, & x_j\leqslant x\leqslant x_{j+1},\\[2mm] \dfrac{(x-x_{j-1})}{\Delta x}, & x_{j-1}\leqslant x\leqslant x_j,\\[2mm] 0, & \text{otherwise}.\end{cases} \tag{4.3}$$

Substituting representation (4.2) into (4.1) gives

$$\begin{aligned} v_k(\tau+\Delta\tau) &= \int_{x_{\min}}^{x_{\max}} g(x_k - x, \Delta\tau) v(x,\tau)\,\mathrm{d}x \\ &= \sum_{j=-N/2}^{N/2-1} v_j(\tau) \int_{x_{\min}}^{x_{\max}} \phi_j(x) g(x_k - x, \Delta\tau)\,\mathrm{d}x \\ &= \sum_{j=-N/2}^{N/2-1} v_j(\tau) \tilde{g}(x_k - x_j, \Delta\tau)\Delta x, \end{aligned} \tag{4.4}$$

where

$$\tilde{g}(x_k - x_j, \Delta\tau) = \frac{1}{\Delta x} \int_{x_k - x_j - \Delta x}^{x_k - x_j + \Delta x} \phi_{k-j}(x) g(x, \Delta\tau)\,\mathrm{d}x. \tag{4.5}$$

Here we have used $\phi_j(x_k - x) = \phi_{k-j}(x)$, a property that follows from the properties of linear basis functions. Setting $\ell = k - j$, $y_\ell = x_k - x_j = \ell\Delta x$ for $\ell = -\frac{1}{2}N, \dots, \frac{1}{2}N - 1$ gives

$$\tilde{g}(y_\ell, \Delta\tau) = \frac{1}{\Delta x} \int_{y_\ell - \Delta x}^{y_\ell + \Delta x} \phi_\ell(x) g(x, \Delta\tau)\,\mathrm{d}x \tag{4.6}$$

as the averaged projection of the Green's function onto the basis functions ϕ_ℓ. Note that, for this projection, $\tilde{g}(y_\ell, \Delta\tau) \geqslant 0$, since the exact Green's function has $g(x) \geqslant 0$ for all x, and of course $\phi_\ell(x) \geqslant 0$. Therefore, the scheme (4.4) is monotone for any N.

REMARK 4.1 (Green's function availability in closed form) If the Green's function is available in closed form, rather than just its Fourier transform, then (4.6) can be used to compute the $\tilde{g}(y_\ell, \Delta\tau)$ terms directly, as, for example, in Tanskanen and Lukkarinen (2003). However, in general this will require a numerical integration. If the Fourier transform of the Green's function is known, we will derive a technique to efficiently compute $\tilde{g}(y_\ell, \Delta\tau)$ to an arbitrary level of accuracy.

4.2 Approximating the monotone scheme

The scheme (4.4) is monotone, since the weights $\tilde{g}(y_\ell, \Delta\tau)$ given in (4.6) are nonnegative. However it is only possible for us to approximate these weights, and this prevents us from guaranteeing monotonicity. In this subsection, we show how we overcome this issue.

Recall that our starting point is that G, the Fourier series of the Green's function, is known in closed form. We then replaced our Green's function $g(x, \Delta\tau)$ by its localized, periodic approximation,

$$\hat{g}(x, \Delta\tau) = \frac{1}{P} \sum_{k=-\infty}^{\infty} \mathrm{e}^{2\pi \mathrm{i} \omega_k x} G(\omega_k, \Delta\tau),$$
$$\omega_k = \frac{k}{P}, \quad P = x_{\max} - x_{\min},$$

and then projected the Green's function onto the linear basis functions. Replacing $g(x, \Delta\tau)$ by $\hat{g}(x, \Delta\tau)$ in (4.6), and assuming uniform convergence of the Fourier series, we integrate (4.6) term by term, resulting in

$$\tilde{g}(y_j, \Delta\tau) = \frac{1}{P} \sum_{k=-\infty}^{\infty} \left(\frac{1}{\Delta x} \int_{y_j - \Delta x}^{y_j + \Delta x} \mathrm{e}^{2\pi \mathrm{i} \omega_k x} \phi_j(x) \, \mathrm{d}x \right) G(\omega_k, \Delta\tau). \tag{4.7}$$

In the case of linear basis functions (4.3), we convert the complex exponential in (4.7) into trigonometric functions, with the resulting integration giving[2]

$$\tilde{g}(y_j, \Delta\tau) = \frac{1}{P} \sum_{k=-\infty}^{\infty} \mathrm{e}^{2\pi \mathrm{i} \omega_k y_j} \left(\frac{\sin^2 \pi \omega_k \Delta x}{(\pi \omega_k \Delta x)^2} \right) G(\omega_k, \Delta\tau). \tag{4.8}$$

This is then approximated by truncating the series.

A key point is that the truncation of the projection of the Green's function does not have to use the same number of terms as the number of basis functions. That is, set $N' = \alpha N$, with N defined in (4.2) and $\alpha = 2^k$ for $k = 1, 2, \dots$. Suppose we now truncate the Fourier series for the projected linear basis form for $\tilde{g}$ to N' terms. Let $\tilde{g}(y_k, \Delta\tau, \alpha)$ denote the use of a truncated Fourier series with truncation parameter α for a fixed value of N so that the Fourier series (4.8) truncates to

$$\tilde{g}(y_j, \Delta\tau, \alpha) = \frac{1}{P} \sum_{k=-\alpha N/2}^{\alpha N/2 - 1} \mathrm{e}^{2\pi \mathrm{i} \omega_k j \Delta x} \left(\frac{\sin^2 \pi \omega_k \Delta x}{(\pi \omega_k \Delta x)^2} \right) G(\omega_k, \Delta\tau). \tag{4.9}$$

Using the notation $\tilde{g}_j(\Delta\tau, \alpha) = \tilde{g}(y_j, \Delta\tau, \alpha)$, we have

$$\tilde{g}_{j+N}(\Delta\tau, \alpha) = \tilde{g}_j(\Delta\tau, \alpha),$$

so our sequence $\{\tilde{g}_{-N/2}(\Delta\tau, \alpha), \dots, \tilde{g}_{N/2-1}(\Delta\tau, \alpha)\}$ is periodic.

[2] For $\omega_k = 0$, we take the limit $\omega_k \to 0$.

REMARK 4.2 (Efficient computation of the projections) It remains to compute the projections. For this, we need to determine the discrete convolution (4.9). Let

$$Y_k = \left(\frac{\sin^2 \pi\omega_k \Delta x}{(\pi\omega_k \Delta x)^2}\right) G(\omega_k, \Delta\tau), \quad k = -\frac{\alpha N}{2}, \dots, \frac{\alpha N}{2} - 1.$$

Then rewriting $\mathrm{e}^{2\pi \mathrm{i}\omega_k j\Delta x} = \mathrm{e}^{2\pi \mathrm{i} k\ell/(\alpha N)}$ with $\ell = j\alpha$ and defining

$$\mathbb{Y}_\ell = \frac{1}{P} \sum_{k=-\alpha N/2}^{\alpha N/2-1} \exp\left(\frac{2\pi \mathrm{i} k\ell}{\alpha N}\right) Y_k, \quad \ell = -\frac{\alpha N}{2}, \dots, \frac{\alpha N}{2} - 1, \tag{4.10}$$

gives $\{\mathbb{Y}_\ell\}$ as the DFT of the $\{Y_k\}$ (of size $N' = \alpha N$). Consequently, using (4.9) and (4.10) yields

$$\tilde{g}(y_j, \Delta\tau, \alpha) = \mathbb{Y}_\ell, \quad \ell = j\alpha, \quad j = -\tfrac{1}{2}N, \dots, \tfrac{1}{2}N - 1. \tag{4.11}$$

Thus, the projections $\{\tilde{g}(y_j, \Delta\tau, \alpha)\}$ are computed via a single FFT of size N'.

For $k = -\frac{1}{2}N, \dots, \frac{1}{2}N - 1$, we define $\tilde{G}(\omega_k, \Delta\tau, \alpha)$ as the DFT of the $\{\tilde{g}(y_m, \Delta\tau, \alpha)\}$:

$$\tilde{G}(\omega_k, \Delta\tau, \alpha) = \frac{P}{N} \sum_{m=-N/2}^{N/2-1} \exp\left(-\frac{2\pi \mathrm{i} mk}{N}\right) \tilde{g}(y_m, \Delta\tau, \alpha). \tag{4.12}$$

Note that

$$\Delta x = \frac{P}{N} > \frac{P}{N'} = \frac{P}{\alpha N},$$

that is, the basis function is integrated over a grid of size $\Delta x > P/N'$, and so is larger than the grid spacing on the N' grid. As $\alpha \to \infty$, there is no error in evaluating these integrals (projections) for a fixed value of N. For any finite α, there is an error due to the use of a truncated Fourier series.

Again, we emphasize that the truncation for the Fourier series representation of the projection of the Green's function in (4.9) does not have to use the same number of terms (αN) as the discrete convolution (N). Instead, we can take a very accurate expansion of the Green's function projection and then translate this back to the coarse grid using (4.12). There is no further loss of information in this last step. As remarked above, we only use the Fourier representation of $\tilde{g}(y_j, \Delta\tau, \alpha)$ to carry out the discrete convolution, ie, a dense matrix–vector multiplication, efficiently. The discrete convolution in Fourier space is exactly equivalent to the discrete convolution in physical space, assuming periodic extensions.

REMARK 4.3 (Assumption 3.1 revisited) The assumption $g(x, x', \Delta\tau) = g(x - x', \Delta\tau)$ permits fast computation of a dense matrix–vector multiplication using an FFT. As mentioned earlier, this assumption holds for Lévy processes, but it does not hold, for example, for a Heston stochastic volatility model. However, the basic idea of projection of the Green's function onto linear basis functions can be used even if Assumption 3.1 does not hold. The cost, in this case, is a loss of computational efficiency. As an example, in the case of the Heston stochastic volatility model, we have a closed form for the characteristic function, but here the Green's function has the form $g = g(\nu, \nu', x - x', \Delta\tau)$, where ν is the variance and $x = \log S$, where S is the asset price. In this case, we can use an FFT effectively in the x-direction, but not in the ν-direction.

REMARK 4.4 (Relation to the COS method) In the COS method, the solution $v(x, \tau)$ is also expanded in a Fourier series. This gives an exponential convergence of the entire algorithm for smooth $v(x, \tau)$, which in turn requires that we have a highly accurate Fourier representation of $v(x, \tau)$. However, suppose $v(x, \tau)$ is obtained by applying an impulse control using a numerical optimization method at discrete points on a previous step, using linear interpolation (the only interpolation method that is monotone in general). In that case, we will not have an accurate representation of the Fourier series of $v(x, \tau)$. In addition, it does not seem possible to ensure monotonicity for the COS method. So far, we have only assumed that the $v(x, \tau)$ can be expanded in terms of piecewise linear basis functions. This property can be used to guarantee monotonicity. However, convergence will be slower than the COS method if the solution is smooth.

REMARK 4.5 (Piecewise constant basis functions) The equations and the previous discussion in this section also hold if our basis functions are piecewise constant functions, ie, basis functions ϕ_j that are nonzero over $[x_j - \frac{1}{2}\Delta x, x_j + \frac{1}{2}\Delta x]$. In this case, computing the integral in (4.7) gives

$$\tilde{g}(y_j, \Delta\tau) = \frac{1}{P} \sum_{k=-\infty}^{\infty} \mathrm{e}^{2\pi \mathrm{i} \omega_k y_j} \left(\frac{\sin \pi \omega_k \Delta x}{\pi \omega_k \Delta x} \right) G(\omega_k, \Delta\tau) \tag{4.13}$$

with the subsequent equations also requiring slight modifications.

4.3 Computing the monotone scheme

In order to ensure our monotone approach is effective, it remains to compute the discrete convolution (4.4) efficiently. For the DFT pair for $v_j(\tau)$ and $V(\omega_p, \tau)$, we recall that $x_j = \hat{x}_0 + j\Delta x$ and so

$$\left.\begin{aligned} v_j(\tau) &= \sum_{\ell=-N/2}^{N/2-1} V(\omega_\ell, \tau)\mathrm{e}^{2\pi \mathrm{i} \omega_l x_j} \\ &= \sum_{\ell=-N/2}^{N/2-1} (\mathrm{e}^{2\pi \mathrm{i} \omega_\ell \hat{x}_0}) V(\omega_\ell, \tau)\mathrm{e}^{2\pi \mathrm{i} j\ell/N}, \\ V(\omega_p, \tau) &= \frac{1}{N} \sum_{\ell=-N/2}^{N/2-1} \mathrm{e}^{-2\pi \mathrm{i} \omega_p x_\ell} v_\ell(\tau) \\ &= \frac{1}{N} (\mathrm{e}^{-2\pi \mathrm{i} \omega_p \hat{x}_0}) \sum_{\ell=-N/2}^{N/2-1} \exp\left(-\frac{2\pi \mathrm{i} p\ell}{N}\right) v_\ell(\tau). \end{aligned}\right\} \tag{4.14}$$

Suppose we write $\tilde{g}(x_k - x_j, \Delta\tau)$ as a DFT:

$$\tilde{g}_{k-j}(\Delta\tau, \alpha) = \frac{1}{P} \sum_{p=-N/2}^{N/2-1} \tilde{G}(\omega_p, \Delta\tau, \alpha)\mathrm{e}^{2\pi \mathrm{i}(k-j)p/N}, \tag{4.15}$$

where we use (4.12) to determine $\tilde{G}(\omega_p, \Delta\tau, \alpha)$. Substituting (4.15) and (4.14) into (4.4), we then get

$$\begin{aligned} v(x_k, \tau + \Delta\tau) &= \Delta x \sum_{j=-N/2}^{N/2-1} v_j^n \tilde{g}(x_k - x_j, \Delta\tau, \alpha) \\ &= \frac{1}{N} \sum_{p=-N/2}^{N/2-1} \sum_{\ell=-N/2}^{N/2-1} (\mathrm{e}^{2\pi \mathrm{i} \omega_\ell x_0}) \tilde{G}(\omega_p, \Delta\tau, \alpha) \\ &\qquad \times V(\omega_\ell, \tau)\mathrm{e}^{2\pi \mathrm{i} kp/N} \sum_{j=-N/2}^{N/2-1} \exp\left(\frac{2\pi \mathrm{i} j(\ell - p)}{N}\right) \\ &= \sum_{p=-N/2}^{N/2-1} (\mathrm{e}^{2\pi \mathrm{i} \omega_p x_0}) V(\omega_p, \tau) \tilde{G}(\omega_p, \Delta\tau, \alpha) \exp\left(\frac{2\pi \mathrm{i} kp}{N}\right), \end{aligned} \tag{4.16}$$

where the last equation follows from the classical orthogonality properties of Nth roots of unity.

From (4.14), we have

$$V(\omega_p, \tau) = \frac{1}{N} (\mathrm{e}^{-2\pi \mathrm{i} \omega_p \hat{x}_0}) \sum_{\ell=-N/2}^{N/2-1} \exp\left(\frac{-2\pi \mathrm{i} p\ell}{N}\right) v_\ell(\tau) = (\mathrm{e}^{-2\pi \mathrm{i} \omega_p \hat{x}_0}) \tilde{V}(\omega_p, \tau) \tag{4.17}$$

with

$$\tilde{V}(\omega_p, \tau) = \frac{1}{N} \sum_{\ell=-N/2}^{N/2-1} \exp\left(\frac{-2\pi \mathrm{i} p\ell}{N}\right) v_\ell(\tau),$$

the DFT of $\{v_\ell(\tau)\}$. Finally, substituting (4.17) into (4.16) gives

$$v(x_k, \tau + \Delta\tau) = \sum_{p=-N/2}^{N/2-1} \tilde{V}(\omega_p, \tau)\tilde{G}(\omega_p, \Delta\tau, \alpha) \exp\left(\frac{2\pi \mathrm{i} p\ell}{N}\right), \tag{4.18}$$

which we recognize as the inverse DFT of $\{\tilde{V}(\omega_p, \tau)\tilde{G}(\omega_p, \Delta\tau, \alpha)\}$.

REMARK 4.6 (Monotonicity) Equations (4.18) and (4.4) are algebraic identities (assuming periodic extensions). Hence, if we use (4.18) to advance the solution, then this is algebraically identical to using (4.4) to advance the solution. Thus, we can analyze the properties of (4.18) by analyzing (4.4). In particular, if $\tilde{g}(x_k, \Delta\tau, \alpha) \geqslant 0$, then the scheme is monotone.

REMARK 4.7 (Converting FST or CONV to monotone form) Equation (4.18) is formally identical to (3.10). This has the practical result that any FST or CONV software can be converted to a monotone form by a preprocessing step that computes $\tilde{G}(\omega_p, \Delta\tau, \alpha)$ and choosing a trapezoidal rule for the integral in (3.8).

5 A MONOTONE ALGORITHM FOR SOLUTION OF THE CONTROL PROBLEM

In this section, we describe our monotone algorithm for the control problem (2.4), (2.5). Let $(v^n)^+$ be the vector of values of our solution just after τ_n, as defined in (3.20), and let $\mathbb{I}_{\Delta x}(x)$ be the linear interpolation operator defined as in (3.21). Let

$$\tilde{V}^n = [\tilde{V}(\omega_{-N/2}, \tau_n), \dots, \tilde{V}(\omega_{N/2-1}, \tau_n)] = \mathrm{DFT}[(v^n)^-]$$

and

$$\tilde{G} = [\tilde{G}(\omega_{-N/2}, \Delta\tau, \alpha), \dots, \tilde{G}(\omega_{N/2-1}, \Delta\tau, \alpha)].$$

Let us assume that our Green's function is not an explicit function of τ, and that we instead have $g = g(x - x', \Delta\tau)$ and the time steps are all constant, ie, $\tau_{n+1} - \tau_n = \Delta\tau = \text{const}$.

In this case, we can compute $\tilde{G}(\omega_k, \Delta\tau, \alpha)$ only once. If these two assumptions do not hold, then $\tilde{G}(\cdot)$ would have to be recomputed frequently, and hence our algorithm for ensuring monotonicity becomes more costly.

Algorithm 5.1 describes the computation of $\tilde{G}(\cdot)$. Here, we test for monotonicity (up to a small tolerance) by minimizing the effect of any negative weights, by requiring that

$$\sum_j \Delta x |\min(\tilde{g}(y_j, \Delta\tau, \alpha), 0)| < \varepsilon_1 \frac{\Delta\tau}{T}.$$

The test for accuracy of the projection occurs by the comparison

$$\max_j \Delta x |\tilde{g}(y_j, \Delta\tau, \alpha) - \tilde{g}(y_j, \Delta\tau, \tfrac{1}{2}\alpha)| < \varepsilon_2.$$

Both monotonicity and convergence tests are scaled by Δx so that these quantities are bounded as $\Delta\tau \rightarrow 0$ for all Δx (the Green's function becomes unbounded as $\Delta\tau \rightarrow 0$, but the integral of the Green's function is bounded by unity). In addition, the monotonicity test scales ε_1 by $\Delta\tau / T$ in order to eliminate the number of time steps from our monotonicity bounds. This is discussed further in Section 6.

ALGORITHM 5.1 (Initialization of the monotone Fourier method)
Require: closed-form expression for $G(\omega, \Delta\tau)$, the Fourier transform of the Green's function

```
(1)  Input: $N$, $\Delta x$, $\Delta\tau$
(2)  Let $\alpha = 1$ and compute $\tilde{g}(y_j, \Delta\tau, 1)$.
(3)  for $\alpha = 2^k$, $k = 1, 2, \dots$, until convergence do {construct accurate $\tilde{g}$}
(4)      Compute $\tilde{g}(y_j, \Delta\tau, \alpha)$, $\tilde{G}(\omega_j, \Delta\tau, \alpha)$, $j = -\frac{1}{2}N, \dots, \frac{1}{2}N - 1$ using (4.11), (4.12)
(5)      $\text{test}_1 = \sum_j \Delta x \min(\tilde{g}(y_j, \Delta\tau, \alpha), 0)$        {monotonicity test}
(6)      $\text{test}_2 = \max_j \Delta x |\tilde{g}(y_j, \Delta\tau, \alpha) - \tilde{g}(y_j, \Delta\tau, \frac{1}{2}\alpha)|$        {accuracy test}
(7)      if $(|\text{test}_1| < \varepsilon_1(\Delta\tau / T))$ and $(\text{test}_2 < \varepsilon_2)$ then
(8)          break from for loop        {convergence test}
(9)          end if
(10)     end for {end accurate $\tilde{g}$ loop}
(11) Output: weights $\tilde{G}(\omega_j, \Delta\tau, \alpha)$, $j = -\frac{1}{2}N, \dots, \frac{1}{2}N - 1$ in Fourier domain.
```

In Algorithm 5.1, the test on line 5 will ensure that monotonicity holds to a user-specified tolerance and the test on line 6 ensures accuracy of the projections. The complete monotone algorithm for the control problem is given in Algorithm 5.4.

REMARK 5.2 (Convergence of Algorithm 5.1) In Section 5.1, we show that, for typical Green's functions, the test for monotonicity on line 5 in Algorithm 5.1 and the accuracy test on line 6 are usually satisfied for $\alpha = 2, 4$, for typical values of ε_1, ε_2.

REMARK 5.3 (Complexity) The complexity of using (4.18) to advance the time (excluding the cost of determining an optimal control) is $O(N \log N)$ operations, roughly the same as the usual FST/CONV methods.

ALGORITHM 5.4 (Monotone Fourier method)
Require: Weights $\tilde{G} = \{\tilde{G}(\omega_j, \Delta\tau, \alpha)\}$, for $j = -\frac{1}{2}N, \dots, \frac{1}{2}N - 1$ in Fourier domain (from Algorithm 5.1)

```
(1) Input: number of time steps $M$, initial solution $(v^0)^-$
(2) $(v^0)^+ = \inf_c \mathcal{M}(c)(\mathbb{I}_{\Delta x}(x)(v^0)^-)$
(3) for $m = 1, \dots, M$ do {time-step loop}
(4)     $\tilde{V}^{m-1} = \mathrm{FFT}[(v^{m-1})^+]$                {frequency domain}
(5)     $(v^m)^- = \mathrm{iFFT}[\tilde{V}^{m-1} \circ \tilde{G}]$                {physical domain}
(6)     $v(x_j, \tau_m^+) = \inf_c \mathcal{M}(c)(\mathbb{I}_{\Delta x}(x_j)(v^m)^-)$, $j = -\frac{1}{2}N, \dots, \frac{1}{2}N - 1$
                                                              {optimal control}
(7) end for {end time-step loop}
```

5.1 Convergence of truncated Fourier series for the projected Green's functions

Since the Green's function for (3.11) is a smooth function for any finite $\Delta\tau$, we can expect uniform convergence of the Fourier series to the exact Green's function, assuming that $\sigma > 0$. This can also be seen from the exponential decay of the Fourier coefficients, which we demonstrate in this section. Since the exact Green's function is nonnegative, the projected Green's function (4.8) converges to a nonnegative value at every point y_j. Consider the case of the truncated projection on linear basis functions

$$\tilde{g}(y_j, \Delta\tau, \alpha) = \frac{1}{P} \sum_{k=-\alpha N/2}^{\alpha N/2-1} \mathrm{e}^{2\pi \mathrm{i} \omega_k y_j} \left(\frac{\sin^2 \pi \omega_k \Delta x}{(\pi \omega_k \Delta x)^2} \right) G(\omega_k, \Delta\tau) \quad \text{with } \omega_k = \frac{k}{P}$$

and $\Delta x = P/N$. The error in the truncated series is then

$$\begin{aligned}
&|\tilde{g}(y_j, \Delta\tau, \alpha) - \tilde{g}(y_j, \Delta\tau, \infty)| \\
&\qquad = \left| \frac{1}{P} \sum_{k=\alpha N/2}^{\infty} \mathrm{e}^{2\pi \mathrm{i}\omega_k y_j} \left(\frac{\sin^2 \pi\omega_k \Delta x}{(\pi\omega_k \Delta x)^2} \right) G(\omega_k, \Delta\tau) \right. \\
&\qquad\qquad \left. + \frac{1}{P} \sum_{k=-\infty}^{-\alpha N/2-1} \mathrm{e}^{2\pi \mathrm{i}\omega_k y_j} \left(\frac{\sin^2 \pi\omega_k \Delta x}{(\pi\omega_k \Delta x)^2} \right) G(\omega_k, \Delta\tau) \right| \\
&\qquad \leqslant \frac{2}{P} \sum_{k=\alpha N/2}^{\infty} \frac{1}{(\pi\omega_k \Delta x)^2} |G(\omega_k, \Delta\tau)| \\
&\qquad \leqslant \frac{2}{P} \frac{4}{\pi^2\alpha^2} \sum_{k=\alpha N/2}^{\infty} |G(\omega_k, \Delta\tau)|. \qquad (5.1)
\end{aligned}$$

From (3.13), and noting that $\mathrm{Re}(\bar{F}(\omega)) \leqslant 1$ (this holds since $\bar{F}(\omega)$ is the Fourier transform of a density function, and can also be seen directly from (3.18), (3.19)), we then have

$$\begin{aligned}
\mathrm{Re}(\Psi(\omega)) &= -\tfrac{1}{2}\sigma^2(2\pi\omega)^2 - (\rho + \lambda) + \lambda\, \mathrm{Re}(\bar{F}(\omega)) \\
&\leqslant -\tfrac{1}{2}\sigma^2(2\pi\omega)^2 - (\rho + \lambda) + \lambda \\
&\leqslant -\tfrac{1}{2}\sigma^2(2\pi\omega)^2, \qquad (5.2)
\end{aligned}$$

since $\rho \geqslant 0$. From (3.15) and (5.2), we have

$$|G(\omega, \Delta\tau)| = |\mathrm{e}^{\Psi(\omega)\Delta\tau}| \leqslant \exp(-\tfrac{1}{2}\sigma^2(2\pi\omega)^2\Delta\tau). \qquad (5.3)$$

If we let $C_4 = 2\sigma^2\pi^2\Delta\tau/P^2$, then (5.3) and (5.1) imply

$$|\tilde{g}(y_j, \Delta\tau, \alpha) - \tilde{g}(y_j, \Delta\tau, \infty)| \leqslant \frac{8}{P\pi^2\alpha^2} \sum_{k=\alpha N/2}^{\infty} \mathrm{e}^{-C_4 k^2}.$$

Bounding the sum gives

$$|\tilde{g}(y_j, \Delta\tau, \alpha) - \tilde{g}(y_j, \Delta\tau, \infty)| \leqslant \frac{8}{P\pi^2\alpha^2} \frac{\exp(-C_4 N^2\alpha^2/4)}{1 - \mathrm{e}^{-C_4 N\alpha}}. \qquad (5.4)$$

Consider the monotonicity test in Algorithm 5.1, line 5, given by

$$\mathrm{test}_1 = \sum_j \Delta x \min(\tilde{g}(y_j, \Delta\tau, \alpha), 0).$$

Noting that $\tilde{g}(y_j, \Delta\tau, \infty) \geqslant 0$, and from (5.4) and $\sum_j \Delta x = P$, we have

$$|\text{test}_1| \leqslant \frac{8}{\pi^2\alpha^2} \frac{\exp(-C_4 N^2 \alpha^2/4)}{1 - \mathrm{e}^{-C_4 N\alpha}}, \tag{5.5}$$

so that this test is usually satisfied to within round-off for $\alpha = 2, 4$.

Consider now the accuracy test on line 6 of Algorithm 5.1, given by

$$\text{test}_2 = \max_j \Delta x |\tilde{g}(y_j, \Delta\tau, \alpha) - \tilde{g}(y_j, \Delta\tau, \tfrac{1}{2}\alpha)|,$$

which we see is bounded by

$$\begin{aligned} |\text{test}_2| &\leqslant \Delta x \max_j (|\tilde{g}(y_j, \Delta\tau, \alpha) - \tilde{g}(y_j, \Delta\tau, \infty)| \\ &\qquad\qquad + |\tilde{g}(y_j, \Delta\tau, \tfrac{1}{2}\alpha) - \tilde{g}(y_j, \Delta\tau, \infty)|) \\ &\leqslant \Delta x \max_j (2|\tilde{g}(y_j, \Delta\tau, \tfrac{1}{2}\alpha) - \tilde{g}(y_j, \Delta\tau, \infty)|) \\ &\leqslant \frac{64}{\pi^2\alpha^2} \frac{\Delta x}{P} \frac{\exp(-C_4 N^2\alpha^2/16)}{1 - \mathrm{e}^{-C_4 N\alpha/2}}. \end{aligned} \tag{5.6}$$

This test will also be satisfied for small values of α, although these α should be larger than for the monotonicity test (5.5).

6 PROPERTIES OF THE MONOTONE FOURIER METHOD

In this section, we prove a number of properties satisfied by our ε-monotone Fourier algorithm. The main properties include ℓ_∞ stability and a type of ε-discrete comparison principle.

LEMMA 6.1 *Let C_1 be a constant such that the exact Green's function satisfies $C_1 = \int_{\mathbb{R}} g(x, \Delta\tau)\,\mathrm{d}x$. Then, for all k,*

$$\Delta x \sum_{j=-N/2}^{N/2-1} \tilde{g}(x_k - x_j, \Delta x, \alpha) = C_1 \quad \textit{with } \Delta x = \frac{P}{N}.$$

PROOF For $y_\ell = x_k - x_j$, $\ell = k - j$, we have

$$\begin{aligned} &\Delta x \sum_{j=-N/2}^{N/2-1} \tilde{g}(x_k - x_j, \Delta x, \alpha) \\ &\qquad = \frac{P}{N} \sum_{\ell=-N/2}^{N/2-1} \tilde{g}(y_\ell, \Delta\tau, \alpha) \\ &\qquad = \frac{P}{N} \sum_{\ell=-N/2}^{N/2-1} \frac{1}{P} \sum_{k=-\alpha N/2}^{\alpha N/2-1} \mathrm{e}^{2\pi\mathrm{i}\omega_k \ell\Delta x} \biggl(\frac{\sin^2 \pi\omega_k\Delta x}{(\pi\omega_k\Delta x)^2}\biggr) G(\omega_k, \Delta\tau) \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{N} \sum_{k=-\alpha N/2}^{\alpha N/2-1} \left(\frac{\sin^2 \pi \omega_k \Delta x}{(\pi \omega_k \Delta x)^2} \right) G(\omega_k, \Delta\tau) \sum_{\ell=-N/2}^{N/2-1} \exp\left(\frac{2\pi \mathrm{i} \ell k}{N} \right) \\
&= G(0, \Delta\tau) = \int_{-\infty}^{\infty} g(x, \Delta\tau)\, \mathrm{d}x \\
&= C_1.
\end{aligned}
$$

□

THEOREM 6.2 (ℓ_∞ stability) *Assume that $\tilde{G}$ is computed using Algorithm 5.1, that $(v^n)^-$ is computed from*

$$
(v_k^n)^- = \sum_{j=-N/2}^{N/2-1} \Delta x \tilde{g}_{k-j} (v_j^{n-1})^+ \tag{6.1}
$$

and that

$$
\|(v^n)^+\|_\infty \leqslant \|(v^n)^-\|_\infty. \tag{6.2}
$$

Then, for every $0 \leqslant n \leqslant M$, we have

$$
\|(v^n)^+\|_\infty \leqslant C_2 = \mathrm{e}^{2\varepsilon_1} \|(v^0)^-\|_\infty.
$$

PROOF From (6.1), we obtain

$$
\begin{aligned}
(v_k^n)^- &= \sum_{j=-N/2}^{N/2-1} \Delta x \tilde{g}_{k-j} (v_j^{n-1})^+ \\
&= \sum_{j=-N/2}^{N/2-1} \Delta x \max(\tilde{g}_{k-j}, 0)(v_j^{n-1})^+ + \sum_{j=-N/2}^{N/2-1} \Delta x \min(\tilde{g}_{k-j}, 0)(v_j^{n-1})^+,
\end{aligned} \tag{6.3}
$$

which then implies

$$
\begin{aligned}
|(v_k^n)^-| \leqslant \|(v^{n-1})^+\|_\infty &\sum_{j=-N/2}^{N/2-1} \Delta x \max(\tilde{g}_{k-j}, 0) \\
&+ \|(v^{n-1})^+\|_\infty \sum_{j=-N/2}^{N/2-1} \Delta x |\min(\tilde{g}_{k-j}, 0)|.
\end{aligned}
$$

From Lemma 6.1, we get

$$
\sum_{j=-N/2}^{N/2-1} \Delta x \max(\tilde{g}_{k-j}, 0) = C_1 + \sum_{j=-N/2}^{N/2-1} \Delta x |\min(\tilde{g}_{k-j}, 0)|, \tag{6.4}
$$

and so, noting lines 5 and 7 in Algorithm 5.1, we have

$$\begin{aligned} |(v_k^n)^-| &\leqslant \|(v^{n-1})^+\|_\infty\bigg(C_1 + 2\sum_{j=-N/2}^{N/2-1} \Delta x|\min(\tilde{g}_{k-j}, 0)|\bigg) \\ &\leqslant \|(v^{n-1})^+\|_\infty\bigg(C_1 + 2\varepsilon_1\frac{\Delta\tau}{T}\bigg). \end{aligned} \tag{6.5}$$

Since (6.5) is true for any k, we have that

$$\|(v^n)^-\|_\infty \leqslant \|(v^{n-1})^+\|_\infty\bigg(C_1 + 2\varepsilon_1\frac{\Delta\tau}{T}\bigg),$$

which, combined with (6.2) and using $C_1 \leqslant 1$, gives

$$\|(v^n)^+\|_\infty \leqslant \|(v^{n-1})^+\|_\infty\bigg(1 + 2\varepsilon_1\frac{\Delta\tau}{T}\bigg).$$

Iterating the above bound and using (6.2) at $n = 0$ gives

$$\begin{aligned} \|(v^n)^+\|_\infty &\leqslant \|(v^0)^-\|_\infty\bigg(1 + 2\varepsilon_1\frac{\Delta\tau}{T}\bigg)^n \\ &\leqslant \|(v^0)^-\|_\infty \exp\bigg(2\varepsilon_1 n\frac{\Delta\tau}{T}\bigg) \\ &\leqslant \|(v^0)^-\|_\infty \mathrm{e}^{2\varepsilon_1} \\ &= C_2. \end{aligned}$$

□

REMARK 6.3 (Jump condition) We remark that the jump condition $\|(v^n)^+\|_\infty \leqslant \|(v^n)^-\|_\infty$ (see 6.2) is trivially satisfied if $\mathbb{I}_{\Delta x}(x)$ in line 6 in Algorithm 5.4 is a linear interpolant.

From Theorem 6.2 and Remark 6.3, we immediately obtain the following result.

COROLLARY 6.4 (Stability of Algorithm 5.4) *Algorithm 5.4 is ℓ_∞-stable.*

LEMMA 6.5 (Minimum value of solution) *Let $(v^n)^+$ be generated using (6.1), and set*

$$(v^n)^+_{\min} = \min_k (v_k^n)^+.$$

If the conditions for Lemma 6.1 are satisfied and

$$(v^n)^+_{\min} \geqslant (v^n)^-_{\min}, \tag{6.6}$$

then

$$(v^n)^+_{\min} \geqslant (v^0)^-_{\min}(C_3)^n - C_2(\mathrm{e}^{\varepsilon_1} - 1),$$

where $C_2 = \|(v^0)^-\|_\infty \mathrm{e}^{2\varepsilon_1}$ *is given in Lemma 6.1 and*

$$C_3 = \sum_{j=-N/2}^{N/2-1} \Delta x \max(\tilde{g}_{k-j}, 0).$$

PROOF From (6.1), and using (6.3) along with the definition of C_3, we obtain

$$\begin{aligned}(v_k^n)^- &\geqslant (v^{n-1})^+_{\min} \sum_{j=-N/2}^{N/2-1} \Delta x \max(\tilde{g}_{k-j}, 0) + \sum_{j=-N/2}^{N/2-1} \Delta x \min(\tilde{g}_{k-j}, 0)(v_j^n)^+ \\ &\geqslant (v^{n-1})^+_{\min} \sum_{j=-N/2}^{N/2-1} \Delta x \max(\tilde{g}_{k-j}, 0) \\ &\quad - \|(v^{n-1})^+\|_\infty \sum_{j=-N/2}^{N/2-1} \Delta x |\min(\tilde{g}_{k-j}, 0)| \\ &= (v^{n-1})^+_{\min} C_3 - \|(v^{n-1})^+\|_\infty \bigg(\sum_{j=-N/2}^{N/2-1} \Delta x |\min(\tilde{g}_{k-j}, 0)| \bigg).\end{aligned}$$

Using Lemma 6.1 and lines 5 and 7 in Algorithm 5.1 then gives

$$(v_k^n)^- \geqslant (v^{n-1})^+_{\min} C_3 - C_2 \varepsilon_1 \frac{\Delta\tau}{T}$$

and, since this is valid for any k, using (6.6) we obtain

$$(v^n)^+_{\min} \geqslant (v^{n-1})^+_{\min} C_3 - C_2 \varepsilon_1 \frac{\Delta\tau}{T}.$$

Iterating implies

$$\begin{aligned}(v^n)^+_{\min} &\geqslant (v^0)^+_{\min} C_3^n - C_2 \varepsilon_1 \frac{\Delta\tau}{T} \bigg(\frac{1 - C_3^n}{1 - C_3} \bigg) \\ &\geqslant (v^0)^-_{\min} C_3^n - C_2 \varepsilon_1 \frac{\Delta\tau}{T} \bigg(\frac{1 - C_3^n}{1 - C_3} \bigg),\end{aligned} \tag{6.7}$$

where we again use (6.6) in the last line. From (6.4) and the definition of C_3, we have

$$C_3 = C_1 + \sum_{j=-N/2}^{N/2-1} \Delta x |\min(\tilde{g}_{k-j}, 0)| \leqslant 1 + \varepsilon_1 \frac{\Delta\tau}{T}, \tag{6.8}$$

where the last inequality follows from lines 5 and 7 in Algorithm 5.1 (and recalling that $C_1 \leqslant 1$). Combining (6.6)–(6.8), and noting that $n\Delta\tau \leqslant T$, gives

$$(v^n)^+_{\min} \geqslant (v^0)^+_{\min} C_3^n - C_2(\mathrm{e}^{\varepsilon_1} - 1) \geqslant (v^0)^-_{\min} C_3^n - C_2(\mathrm{e}^{\varepsilon_1} - 1).$$

□

REMARK 6.6 We note that condition (6.6), that is, $(v^n)^+_{\min} \geqslant (v^n)^-_{\min}$, is satisfied if $\mathbb{I}_{\Delta x}(x)$ in line 6 in Algorithm 5.4 is a linear interpolant.

THEOREM 6.7 (ε-discrete comparison principle) *Suppose we have two independent discrete solutions:*

$$\left.\begin{aligned} (u^n)^+ &= [u(x_{-N/2}, \tau_n^+), \dots, u(x_{+N/2-1}, \tau_n^+)], \\ (w^n)^+ &= [w(x_{-N/2}, \tau_n^+), \dots, w(x_{+N/2-1}, \tau_n^+)], \end{aligned}\right\} \tag{6.9}$$

with

$$(u^0)^- \geqslant (w^0)^-,$$

where the inequality is understood in the component-wise sense, and $(u^n)^+, (w^n)^+$ *are computed using Algorithm 5.4. If* $\tilde{G}$ *is computed using Algorithm 5.1 and* $\mathbb{I}_{\Delta x}(x)$ *is a linear interpolant, then*

$$(u^n)^+ - (w^n)^+ \geqslant -\varepsilon_1 \|(u^0 - w^0)^-\|_\infty + O(\varepsilon_1^2), \quad \varepsilon_1 \to 0. \tag{6.10}$$

PROOF Let $(z^n)^+ = (u^n)^+ - (w^n)^+$, $(z^n)^- = (u^n)^- - (w^n)^-$. Then,

$$(z_k^n)^- = \sum_{j=-N/2}^{N/2-1} \Delta x \tilde{g}_{k-j} (z_j^{n-1})^+.$$

Noting that

$$z_j(\tau_n^+) = \inf_c \mathcal{M}(c)(\mathbb{I}_{\Delta x}(x_j)(u^m)^-) - \inf_c \mathcal{M}(c)(\mathbb{I}_{\Delta x}(x_j)(w^m)^-), \tag{6.11}$$

we then have

$$|z_j(\tau_n^+)| \leqslant \sup_c \mathcal{M}(c) |\mathbb{I}_{\Delta x}(x_j)((u^m)^- - (w^m)^-)|. \tag{6.12}$$

Hence, using the definition of the intervention operator (2.6), we obtain

$$\|(z^n)^+\|_\infty \leqslant \|(z^n)^-\|_\infty. \tag{6.13}$$

Similarly,

$$\begin{aligned}(z^n)^+_{\min} &= \min_j z_j(\tau_n^+) \\ &\geqslant \min_j \inf_c \mathcal{M}(c)\mathbb{I}_{\Delta x}(x_j)((u^m)^- - (w^m)^-) \\ &\geqslant (z^n)^-_{\min}.\end{aligned} \tag{6.14}$$

Hence, condition (6.2) of Lemma 6.1 and condition (6.6) of Lemma 6.5 are satisfied. Applying Lemma 6.5 to $(z^n)^+$, $(z^n)^-$, we get

$$(z^n)^+_{\min} \geqslant (z^0)^-_{\min}(C_3)^n - \mathrm{e}^{2\varepsilon_1}\|(u^0 - w^0)^-\|_\infty(\mathrm{e}^{\varepsilon_1} - 1), \tag{6.15}$$

where $C_3 = \sum_{j=-N/2}^{N/2-1} \Delta x \max(\tilde{g}_{k-j}, 0)$. Since $(z^0)^-_{\min} \geqslant 0$ and $0 \leqslant C_3^n \leqslant \mathrm{e}^{\varepsilon_1}$, the result follows. □

REMARK 6.8 If Algorithm 5.1 is used to construct $\tilde{G}$ for use in Algorithm 5.4, then the ε-discrete comparison property is satisfied for any N, $\Delta\tau$, M up to order ε_1. Since typically $\tilde{g}(y_j, \Delta\tau, \alpha) \to \tilde{g}(y_j, \Delta\tau, \infty) \geqslant 0$ exponentially in α, in practice it is very inexpensive to make ε_1 as small as desired.

REMARK 6.9 (Continuously observed impulse control problems) By determining the optimal control at each time step, we can apply our monotone Fourier method to the continuously observed impulse control problem:

$$\max\Big[v_\tau - \mathcal{L}v, v - \inf_c \mathcal{M}(c)v\Big] = 0. \tag{6.16}$$

This is effectively a method whereby the optimal control is applied explicitly, as in Chen and Forsyth (2008). Using the methods developed in this paper combined with those from Chen and Forsyth (2008), it is straightforward to show that the ε-monotone Fourier technique is ℓ_∞-stable and consistent in the viscosity sense as $\Delta\tau, \Delta x \to 0$. The ε-monotone Fourier method is also monotone to $O(h)$, where $h = O(\Delta x) = O(\Delta\tau)$ is the discretization parameter. Thus, it is possible to show convergence to the viscosity solution using the results in Barles and Souganidis (1991) extended as in Azimzadeh *et al* (2018), using the ε-monotonicity property as in Bokanowski *et al* (2018).

7 MINIMIZATION OF WRAPAROUND ERROR

The use of the convolution form for our solution (4.18) is rigorously correct for a periodic extension of the solution and the Green's function. In normal option-pricing applications, the wraparound error due to periodic extension causes little error. However, in control applications, the values used in the optimization step (2.5) may be near the ends of the grid, and thus large errors may result (Ignatieva *et al* 2018; Lippa 2013; Ruijter *et al* 2013). Hence, we need to consider methods to reduce errors associated with wraparound.

In order to minimize the effect of wraparound, we proceed in the following manner. Given the localized problem on $[x_{\min}, x_{\max}]$ with N nodes, we construct an auxiliary grid with $N^a = 2N$ nodes, on the domain $[x^a_{\min}, x^a_{\max}]$, where

$$x^a_{\min} = x_{\min} - \tfrac{1}{2}(x_{\max} - x_{\min}) \quad \text{and} \quad x^a_{\max} = x_{\max} + \tfrac{1}{2}(x_{\max} - x_{\min}) \tag{7.1}$$

with $(x^a_{\max} - x^a_{\min}) = 2(x_{\max} - x_{\min})$. We construct and store the DFT of the projection of the Green's function $\tilde{G}(\omega_p, \Delta\tau, \alpha)$, $p = -\frac{1}{2}N^a, \dots, \frac{1}{2}N^a - 1$, on this auxiliary grid. We then replace line 4 in Algorithm 5.4 by applying the DFT to the solution v on the auxiliary grid

$$\begin{aligned} v(x_k, \tau_n^+)^a &= v(x_k, \tau_n^+) && (k = -\tfrac{1}{2}N, \dots, \tfrac{1}{2}N - 1) \\ &= v(x_{-N/2}, \tau_n^+) && (k = -\tfrac{1}{2}N^a, \dots, -\tfrac{1}{2}N - 1) \qquad (7.2\text{a}) \\ &= A(x_k, \tau_n^+) && (k = \tfrac{1}{2}N, \dots, \tfrac{1}{2}N^a - 1), \qquad (7.2\text{b}) \end{aligned}$$

where $A(x, \tau)$ is an asymptotic form of the solution, which we assume to be available by financial reasoning. On the auxiliary grid near $x \to -\infty$, we simply extend the solution by the constant value at $x = x_{\min}$, which is expected to generate a small error, since the grid spacing (in terms of $S = \mathrm{e}^x$) is very small. We then carry out lines 4 and 5 of Algorithm 5.4 on the auxiliary grid and generate $(v^n)^-$ by discarding all the values on the auxiliary grid that are not on the original grid (as these are contaminated by wraparound errors). The errors incurred by using extensions (7.2a) and (7.2b) can be made small by choosing $|x_{\min}|$ and $x_{\max}$ sufficiently large.

REMARK 7.1 (Use of asymptotic form to reduce wraparound error) Use of the above technique necessitates some changes to the proof of Theorem 6.7. However, the main result is the same, with adjustments to some of the constants in the bounds. This is a tedious algebraic exercise, which we omit.

REMARK 7.2 (Additional complexity to reduce wraparound) For a one-dimensional problem, the complexity for one time step is

$$O(N^a \log N^a) = O(2N \log(2N)),$$

where N is the number of nodes in the original grid. In the case of the path-dependent problem in Section 9, if there are N_x nodes in the $\log S$-direction and N_b nodes in the bond direction, then the complexity for one time step is $O(2N_b N_x \log(2N_x))$.

TABLE 1 European call option test.

Expiry time	0.25 years
Strike K	100
Payoff	Call
Initial asset price S_0	100
Risk-free rate r	0.05
Volatility σ	0.15
λ	0.1
η_1	3.0465
η_2	3.0775
p_u	0.3445
$x_{\max}$	$\log(S_0) + 10$
$x_{\min}$	$\log(S_0) - 10$
$\varepsilon_1, \varepsilon_2$	10^{-6}
Asymptotic form $x \to \infty$	$A(x) = \mathrm{e}^x$

8 NUMERICAL EXAMPLES

8.1 European option

Consider a European option written on an underlying stock whose price S follows a jump–diffusion process. Denote by ξ the random number representing the jump multiplier, so that when a jump occurs we have $S_t = \xi S_{t-}$. The risk-neutral process followed by S_t is

$$\frac{\mathrm{d}S_t}{S_{t-}} = (r - \lambda\kappa)\,\mathrm{d}t + \sigma\,\mathrm{d}Z + \mathrm{d}\bigg(\sum_{i=1}^{\pi_t}(\xi_i - 1)\bigg) \quad \text{with } \kappa = E[\xi] - 1, \tag{8.1}$$

where $E[\cdot]$ denotes the expectation operator. Here, $\mathrm{d}Z$ is the increment of a Wiener process, r is the risk-free rate, σ is the volatility, π_t is a Poisson process with positive intensity parameter λ, and ξ_i are independent and identically distributed positive random variables. The density function $f(y)$, $y = \log(\xi)$, is assumed to be double exponential (Kou and Wang 2004):

$$f(y) = p_u\eta_1 \mathrm{e}^{-\eta_1 y}\mathbf{1}_{y\geqslant 0} + (1 - p_u)\eta_2\mathrm{e}^{\eta_2 y}\mathbf{1}_{y<0} \tag{8.2}$$

with the expectation

$$E[\xi] = \frac{p_u\eta_1}{\eta_1 - 1} + \frac{(1 - p_u)\eta_2}{\eta_2 + 1}. \tag{8.3}$$

Given that a jump occurs, p_u is the probability of an upward jump and $(1 - p_u)$ is the probability of a downward jump.

TABLE 2 European call option test: value at $t = 0$, $S = S_0$.

(a) Monotone methods

	Piecewise linear		**Piecewise constant**	
N	**Value**	**Ratio**	**Value**	**Ratio**
2^9	3.9808516210		3.9443958729	
2^{10}	3.9753205007		3.9662547470	
2^{11}	3.9739391670	4.0	3.9716756819	4.0
2^{12}	3.9735939225	4.0	3.9730282349	4.0
2^{13}	3.9735076171	4.0	3.9733662066	4.0
2^{14}	3.9734860412	4.0	3.9734506895	4.0

(b) FST/CONV

	Trapezoidal		**Simpson**	
N	**Value**	**Ratio**	**Value**	**Ratio**
2^9	3.9075619850		3.9784907318	
2^{10}	3.9571661688		3.9737010716	
2^{11}	3.9694107823	4.1	3.9734923202	23.0
2^{12}	3.9724624589	4.0	3.9734796846	17.0
2^{13}	3.9732247908	4.0	3.9734789013	16.0
2^{14}	3.9734153372	4.0	3.9734788524	16.0

Parameters are given in Table 1. N denotes the number of nodes. "Ratio" is the ratio of successive changes.

The price of a European call option $v(x, \tau)$ with $x = \log S$ is then given as the solution to

$$v_\tau = \tfrac{1}{2}\sigma^2 v_{xx} + (r - \tfrac{1}{2}\sigma^2 - \lambda\kappa)v_x - (r + \lambda)v + \lambda \int_{-\infty}^{+\infty} v(x + y) f(y)\,\mathrm{d}y \quad \text{with } v(x, 0) = \max(\mathrm{e}^x - K, 0). \tag{8.4}$$

The Green's function for this problem is given in Section 3.1.1.

The particular parameters for this test are given in Table 1, with the results appearing in Table 2. All methods obtain smooth second-order convergence, with the exception of the FST/CONV Simpson rule method, which gives fourth-order convergence, due to the higher-order quadrature method. This is to be expected in this case, since there is a node at the strike. Increasing $x_{\max}$ and $|x_{\min}|$ alters the last two digits of the results in the table. This is due to the effects of both localizing the problem to $[x_{\min}, x_{\max}]$ and FFT wraparound.

TABLE 3 European call option test: value at $t = 0$, $S = S_0$ for $T = 0.001$.

(a) Monotone methods

	Piecewise linear		Piecewise constant	
N	**Value**	**Ratio**	**Value**	**Ratio**
2^9	0.19662316859		0.94284763015	
2^{10}	0.19467436458		0.041410269769	
2^{11}	0.19376651687	2.1	0.15335986938	−8.0
2^{12}	0.19346709107	3.0	0.18477993505	3.6
2^{13}	0.19339179620	4.0	0.19127438852	4.8
2^{14}	0.19337297842	4.0	0.19284673379	4.1

(b) FST/CONV

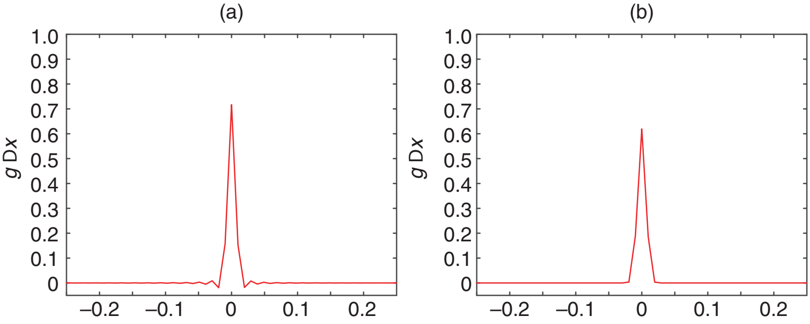
	Trapezoidal		Simpson	
N	**Value**	**Ratio**	**Value**	**Ratio**
2^9	0.24774086499		319.45747026	
2^{10}	0.21909081933		521.62802838	
2^{11}	0.18611676723	0.87	439.13444172	−2.5
2^{12}	0.17728640855	3.7	27.002978049	0.2
2^{13}	0.18913280108	−0.75	0.19367805822	15.0
2^{14}	0.19231903134	3.7	0.19338110881	9.0×10^4

Parameters as given in Table 1 but $T = 0.001$. N denotes the number of nodes. "Ratio" is the ratio of successive changes.

In order to stress these Fourier methods, we repeat this example using an expiry time of $T = 0.001$. Since the Green's function in the physical space converges to a delta function as $T \to 0$, we can expect this to be challenging for Fourier methods, as a large number of terms will be required in the Fourier series in order to get an accurate representation of the Green's function in the physical space. The results for this test are shown in Table 3. The monotone method with piecewise linear basis functions gives reasonable results for all grid sizes. The standard FST/CONV methods are quite poor, except for very large numbers of nodes. Indeed, using Simpson's rule on coarse grids even results in values larger than S_0 at $S = S_0 = 100$, which violates the provable bound for a call option.

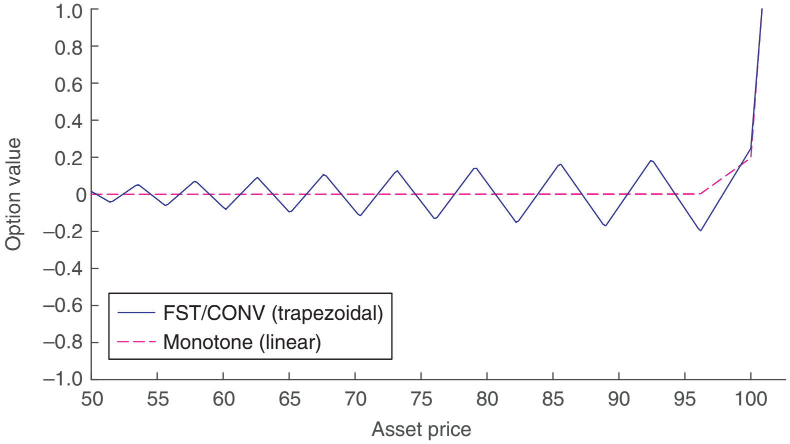
This phenomenon can be explained by examining Figure 1, which shows the projection of the Green's functions for the monotone method (piecewise linear basis function) and the truncated Green's function for the FST/CONV method. The

FIGURE 1 European call option test.



Parameters are as in Table 1 except $T = 0.001$. (a) FST/CONV Green's function, truncated Fourier series, scaled by Δx; $N = 2048$. (b) Monotone method, Green's function projected on linear basis functions; $\tilde{g}(x, \Delta\tau, \alpha)\Delta x$ with $N = 2048$, $\alpha = 4$.

FIGURE 2 European call option test with $T = 0.001$, $N = 512$.



Other parameters are as in Table 1. For the monotone solution, $\alpha = 4$ (see Algorithm 5.1).

projection of the Green's function for the monotone method in Figure 1(b) clearly has the expected properties: very peaked near $x = 0$ and nonnegative for all x. In contrast, the FST/CONV numerical Green's function is oscillatory and negative for some values of x. Figure 2 compares the FST/CONV (trapezoidal) solution with the monotone (piecewise linear) solution, on a coarse grid with 512 nodes. The monotone

TABLE 4 Bermudan put option test.

Expiry time	10 years
Strike K	100
Payoff	Put
Initial asset price S_0	100
Risk-free rate r	0.05
Volatility σ	0.15
Dividend D	1.00
Monitoring frequency $\Delta\tau$	1.0 years
λ	0.1
ν	-1.08
γ	0.4
$x_{\max}$	$\log(S_0) + 10$
$x_{\min}$	$\log(S_0) - 10$
$\varepsilon_1, \varepsilon_2$	10^{-6}
Asymptotic form $x \to \infty$	$A(x) = 0$

solution can never produce a value less than zero (to within the tolerance). Note that monotonicity is clearly violated for the FST/CONV solution, with negative values for a call option. The oscillations are even more pronounced if Simpson's quadrature is used for the FST/CONV method.

REMARK 8.1 (Error in approximating (4.6) using (4.11)) An estimate of the error in computing the projected Green's function is given in (5.6). We can see that a very small time step affects the exponent in (5.6). For the extreme case of $T = 0.001$, $N = 2056$ (the problem in Table 1), we observe that, for $\alpha = 8$, test_2 in Algorithm 5.1 is approximately 10^{-12}, indicating that a very high accuracy projection can be achieved under extreme situations. For the same problem (2056 nodes) with $T = 0.25$, we find that test_2 in Algorithm 5.1 is approximately 10^{-16} for $\alpha = 2$.

From these tests, we can conclude both that the monotone method is robust for all time-step sizes and that for smooth problems and large time steps the monotone method exhibits a slower rate of convergence than high-order techniques, as expected.

8.2 Bermudan option with nonproportional discrete dividends

Let us now assume that we have the same underlying process (8.1) as in the previous subsection, except that the density function for $y = \log(\xi)$ is assumed to be normal:

$$f(y) = \frac{1}{\sqrt{2\pi}\gamma} \exp\left(-\frac{(y-\nu)^2}{2\gamma^2}\right), \tag{8.5}$$

TABLE 5 Bermudan put option test: value at $t = 0, S = S_0$.

(a) Monotone methods

	Piecewise linear		**Piecewise constant**	
N	**Value**	**Ratio**	**Value**	**Ratio**
2^9	24.811127744		24.806532754	
2^{10}	24.789931363		24.788800257	
2^{11}	24.782264461	2.8	2.4.781982815	2.6
2^{12}	24.781134292	6.8	24.781063962	7.4
2^{13}	24.780822977	3.6	24.780805394	3.6
2^{14}	24.780744620	4.0	24.780740225	4.0

(b) FST/CONV

	Trapezoidal		**Simpson**	
N	**Value**	**Ratio**	**Value**	**Ratio**
2^9	24.801967268		24.802639420	
2^{10}	24.787670043		24.787731820	
2^{11}	24.781701225	2.4	24.781787212	2.5
2^{12}	24.780993635	8.4	24.781007785	7.6
2^{13}	24.780787811	3.4	24.780788678	3.6
2^{14}	24.780735831	4.0	24.780737159	4.3

Parameters are as in Table 1. N denotes the number of nodes. "Ratio" is the ratio of successive changes.

with expectation $E[\xi] = \mathrm{e}^{\nu+\gamma^2/2}$. Rather than a European option, we will now consider a Bermudan put option that can be early exercised at fixed monitoring times τ_n. In addition, the underlying asset pays a fixed dividend amount D at τ_n^-, ie, immediately after the early exercise opportunity in forward time. Between monitoring dates, the option price is given by (8.4). At monitoring dates, we have the condition

$$v(x, \tau_n^+) = \max(v(\log(\max(\mathrm{e}^x - D, \mathrm{e}^{x_{\min}})), \tau_n^-), P(x)), \\ \text{with } P(x) = \text{payoff} = \max(K - \mathrm{e}^x, 0). \quad (8.6)$$

The expression $\max(\mathrm{e}^x - D, \mathrm{e}^{x_{\min}})$ in (8.6) ensures that the no-arbitrage condition holds, ie, the dividend cannot be larger than the stock price, taking into account the localized grid. Linear interpolation is used to evaluate the option value in (8.6). The parameters for this problem are listed in Table 4, with numerical results given in Table 5. All methods perform similarly, with second-order convergence. We can see that once we use a linear interpolation to impose the control there is no benefit, in terms of convergence order, to using a high-order method.

9 MULTIPERIOD MEAN–VARIANCE OPTIMAL ASSET ALLOCATION PROBLEM

In this section, we give an example of a realistic problem with complex controls: the multiperiod mean–variance optimal asset allocation problem. Here we consider the case of an investor with a portfolio consisting of a bond index and a stock index. The amount invested in the stock index follows the process under the objective measure

$$\frac{\mathrm{d}S_t}{S_{t^-}} = (\mu - \lambda\kappa)\,\mathrm{d}t + \sigma\,\mathrm{d}Z + \mathrm{d}\biggl(\sum_{i=1}^{\pi_t}(\xi_i - 1)\biggr) \tag{9.1}$$

with the double exponential jump-size distribution (8.2), while the amount in the bond index follows

$$\mathrm{d}B_t = rB_t\,\mathrm{d}t. \tag{9.2}$$

The investor injects cash q_n at time $t_n \in \hat{\mathcal{T}}$, with total wealth at time t being $W_t = S_t + B_t$. Let $W_n^- = S_n^- + B_n^-$ be the total wealth before cash injection. It turns out that, in the multiperiod mean–variance case, in some circumstances, it is optimal to withdraw cash from the portfolio (Cui *et al* 2014; Dang and Forsyth 2016). Denote this optimal cash withdrawal as c_n^*. The total wealth after cash injection and withdrawal is then

$$W_n^+ = W_n^- + q_n - c_n^*. \tag{9.3}$$

We then select an amount b_n^* to invest in the bond, so that

$$B_n^+ = b_n^* \quad \text{and} \quad S_n^+ = W_n^+ - b_n^*. \tag{9.4}$$

Since only cash withdrawals are allowed, we have $c_n^* \geqslant 0$. The control at rebalancing time t_n consists of the pair (b_n^*, c_n^*). That is, after withdrawing c_n^* from the portfolio we rebalance to a portfolio with S_n^+ in stock and B_n^+ in bonds. A no-leverage and no-shorting constraint is enforced by

$$0 \leqslant b_n^* \leqslant W_n^+. \tag{9.5}$$

In order to determine the mean–variance optimal solution to this asset allocation problem, we make use of the embedding result (Li and Ng 2000; Zhou and Li 2000).

The mean–variance optimal strategy can be posed as

$$\min_{\{(b_0^*,c_0^*),\dots,(b_{M-1}^*,c_{M-1}^*)\}} \quad E[(W^* - W_T)^2]$$

$$\text{subject to} \begin{cases} (S_t, B_t) \text{ follow processes (9.1), (9.2);} \quad t \notin \hat{\mathcal{T}}, \\ W_n^+ = S_n^- + B_n^- + q_n - c_n^*, \\ \qquad S_n^+ = W_n^+ - b_n^*, B_n^+ = b_n^*; \quad t \in \hat{\mathcal{T}}, \\ 0 \leqslant b_n^* \leqslant W_n^+, \\ c_n^* \geqslant 0, \end{cases} \tag{9.6}$$

where W^* can be viewed as a parameter that traces out the efficient frontier.

Let

$$Q_\ell = \sum_{j=\ell+1}^{M-1} \mathrm{e}^{-r(t_j - t_\ell)} q_j \tag{9.7}$$

be the discounted future contributions to the portfolio at time t_ℓ. If

$$(W_n^- + q_n) > W^* \mathrm{e}^{-r(T-t_n)} - Q_n, \tag{9.8}$$

then the optimal strategy is to withdraw cash $c_n^* = W_n^- + q_n - (W^* \mathrm{e}^{-r(T-t_n)} - Q_n)$ from the portfolio, and invest the remainder $(W^* \mathrm{e}^{-r(T-t_n)} - Q_n)$ in the risk-free asset. This is optimal in this case since $E[(W^* - W_T)^2] = 0$ (Cui *et al* 2012; Dang and Forsyth 2016), which is the minimum of problem (9.6).

In the following, we will refer to any cash withdrawn from the portfolio as a surplus or free cashflow (Bauerle and Grether 2015). For the sake of argument, we will assume that the surplus cash is invested in a risk-free asset but does not contribute to the computation of the terminal mean and variance. Other possibilities are discussed in Dang and Forsyth (2016).

The solution of (9.6) is the so-called pre-commitment solution. We can interpret the pre-commitment solution in the following way. At $t = 0$, we decide which Pareto point is desirable (ie, a point on the efficient frontier). This fixes the value of W^*. At any time $t > 0$, we can regard the optimal policy as the time-consistent solution to the problem of minimizing the expected quadratic loss with respect to the fixed target wealth W^*, which can be viewed as a useful practical objective function (Menoncin and Vigna 2017; Vigna 2014).

9.1 Optimal control problem

A brief overview of the PIDE for the solution of the mean–variance optimal control problem is given below (we refer the reader to Dang and Forsyth (2014) for additional details).

Let the value function $v(x, b, \tau)$ with $\tau = T - t$ be defined as

$$\begin{aligned} v(x, b, \tau) \\ = \inf_{\{(b_0^*, c_0^*), \dots, (b_M^*, c_M^*)\}} \{E[(\min(W_T - W^*, 0))^2 \mid \log S(t) = x,\ B(t) = b]\}. \end{aligned} \tag{9.9}$$

Let the set of observation times backward in time be $\mathcal{T} = \{\tau_0, \tau_1, \dots, \tau_M\}$. For $\tau \notin \mathcal{T}$, v satisfies

$$\left.\begin{aligned} v_\tau &= \mathcal{L}v + rbv_b, \\ \mathcal{L}v &\equiv \tfrac{1}{2}\sigma^2 v_{xx} + (\mu - \tfrac{1}{2}\sigma^2 - \lambda\kappa)v_x - (\mu + \lambda)v \\ &\qquad + \lambda \int_{-\infty}^{\infty} v(x + y) f(y)\,\mathrm{d}y, \\ v(x, b, 0) &= (\min(\mathrm{e}^x + b - W^*, 0))^2, \end{aligned}\right\} \tag{9.10}$$

on the localized domain $(x, b) \in [x_{\min}, x_{\max}] \times [0, b_{\max}]$.

If $g(x, \tau)$ is the Green's function of $v_\tau = \mathcal{L}v$, then the solution of (9.10) at τ_{n+1}^-, given the solution at τ_n^+, $\tau_n \in \mathcal{T}$, is

$$v(x, b, \tau_{n+1}^-) = \int_{x_{\min}}^{x_{\max}} g(x - x', \Delta\tau) v(x', b\mathrm{e}^{rb\Delta\tau}, \tau_n^+) \quad \text{with } \Delta\tau = \tau_{n+1} - \tau_n. \tag{9.11}$$

Equation (9.11) can be regarded as a combination of a Green's function step for the PIDE $v_\tau = \mathcal{L}v$ and a characteristic technique to handle the rbv_b term. At rebalancing times $\tau_n \in \mathcal{T}$,

$$v(x, b, \tau_n^+) = \min_{(b^*, c^*)} v(x', b^*, \tau_n^-)$$

$$\text{subject to } \begin{cases} c^* = \max(\mathrm{e}^x + b + q_{M-n} - Q_{M-n}, 0), \\ W' = \mathrm{e}^x + b + q_{M-n} - c^*, \\ 0 \leqslant b^* \leqslant W', \\ x' = \log(\max(W' - b^*, \mathrm{e}^{x_{\min}})), \end{cases} \tag{9.12}$$

where Q_ℓ is defined in (9.7).

9.2 Computational details

We solve problem (9.9) combined with the optimal control (9.12) on the localized domain $(x, b) \in [x_{\min}, x_{\max}] \times [0, b_{\max}]$. We discretize in the x-direction using an equally spaced grid with N_x nodes, and in the B-direction using an unequally spaced grid with N_b nodes. Set $B_{\max} = \mathrm{e}^{x_{\max}}$ and denote the discrete solution at (x_m, b_j, τ_n^+) by

$$\left.\begin{aligned} (v_{m,j}^n)^+ &= v(x_m, b_j, \tau_n^+), \\ (v^n)^+ &= \{(v_{m,j}^n)^+\}_{m=-N_x/2,\dots,N_x/2-1;j=1,\dots,N_b}, \\ (v_j^n)^+ &= [(v_{-N_x/2,j}^n)^+, \dots, (v_{N_x/2-1,j}^n)^+]. \end{aligned}\right\}$$

Let $\mathcal{I}_{\Delta x,\Delta b}(x,b)(v^n)^-$ be a two-dimensional linear interpolation operator acting on the discrete solution values $(v^n)^-$. Given the solution at τ_n^+, we use Algorithm 5.4 to advance the solution to τ_{n+1}^-. For the mean–variance problem, we extend this algorithm to approximate (9.11), as described in Algorithm 9.1.

ALGORITHM 9.1 (Advance time $(v^n)^+ \to (v^{n+1})^-$)
Require: $(v^n)^+$, $\tilde{G} = \{\tilde{G}(\omega_m, \Delta\tau, \alpha)\}$, $m = -\frac{1}{2}N_x, \dots, \frac{1}{2}N_x - 1$ (from Algorithm 5.1)

```
for $j = 1, \dots, N_b$ do                                                   {advance time loop}
    $v_{m,j}^{\text{int}} = \mathcal{I}_{\Delta x,\Delta b}(x_m, b_j \mathrm{e}^{r\Delta\tau})(v^n)^+, m = -\frac{1}{2}N_x, \dots, \frac{1}{2}N_x - 1,$
    $\tilde{V} = \text{FFT}[v_j^{\text{int}}]$
    $(v_j^{n+1})^- = \text{iFFT}[\tilde{V} \circ \tilde{G}]$                   {iFFT(Hadamard product)}
end for                                                                  {end advance time loop}
```

In order to advance the solution from τ_{n+1}^- to τ_{n+1}^+, we approximate the solution to the optimal control problem (9.12). The optimal control is approximated by discretizing the candidate control b^* using the discretized b grid and exhaustive search:

$$v(x_m, b_j, \tau_n^+) = \min_{(b^*,c^*)} \mathcal{I}_{\Delta x,\Delta b}(x^*, b^*)(v^{n+1})^-$$

$$\text{subject to} \begin{cases} c^* = \max(\mathrm{e}^{x_m} + b_j + q_{M-n} - Q_{M-n}, 0), \\ W' = \mathrm{e}^{x_m} + b_j + q_{M-n} - c^*, \\ b^* \in \{b_1, \dots, \min(b_{\max}, W')\}, \\ x^* = \log(\max(W' - b^*, \mathrm{e}^{x_{\min}})). \end{cases} \tag{9.13}$$

This algorithm converges to the solution of the original control problem as $N_x, N_b \to \infty$. This can be proved using similar steps to the finite-difference case (Dang and Forsyth 2014). For brevity, we omit the proof.

Using the control determined from solving (9.9), we can determine $E[W_T]$ and standard deviation $\text{SD}[W_T]$ by solving an additional linear PIDE (for details, see Dang and Forsyth (2014)).

REMARK 9.2 (Practical implementation enhancements) As noted by several authors, since the Green's function and the solution are real, the Fourier coefficients satisfy symmetry relations. Hence, $\tilde{G}(\omega_k, \Delta\tau, \alpha)$ and $\tilde{V}$ need to be computed and stored only for $\omega_k \geqslant 0$. It is also possible to arrange the step in line 2 of Algorithm 9.1 and the optimal control step of (9.13) so that only a single interpolation error is introduced at each node. Note that the Fourier series representation of the Green's function is only used to compute the projection of the Green's function onto linear basis functions. After this initial step, we use FFTs only to efficiently carry out a dense matrix–vector multiplication (the convolution) at each step. Use of the FFT here is algebraically identical to carrying out the convolution in the physical space. The only approximation being used in this step is the periodic extension of the solution.

9.3 Numerical example

The data for this problem is given in Table 6. It was determined by fitting to the monthly returns from the Center for Research in Security Prices (CRSP) through Wharton Research Data Services, for the period January 1926 to December 2015. We use the monthly CRSP value-weighted (capitalization weighted) total return index ("vwretd"), which includes all distributions for all domestic stocks trading on major US exchanges, and the monthly ninety-day Treasury bill return index from CRSP. Both this index and the equity index are in nominal terms, so we adjust them for inflation by using the US Consumer Price Index (also supplied by CRSP). We use real indexes, since investors saving for retirement are focused on real (not nominal) wealth goals.

As a first test, we fix $W^* = 1022$, and then increase the number of nodes in the x-direction (N_x) and in the b-direction (N_b). We use the monotone scheme, with linear basis functions. In Table 7, we show the value function $v(0, 0, T)$ and the mean $E[W_T]$ and standard deviation $\mathrm{SD}[W_T]$ of the final wealth, which are of practical importance. The value function shows smooth second-order convergence, which is to be expected. Even though the optimal control is correct only to order Δb (since we optimize by discretizing the controls and using exhaustive search), the value function is correct to $O(\Delta b)^2$ (since it is an extreme point).

We expect that the derived quantities $E[W_T]$, $\mathrm{SD}[W_T]$, which are based on the controls computed as a by-product of computing the value function, should show a lower-order convergence. Recall that these quantities are evaluated by storing the controls and then solving a linear PIDE. In fact, we do see somewhat erratic convergence for these quantities. As an independent check, we used the stored controls from solving for the value function (on the finest grid), and then carried out Monte

TABLE 6 Multiperiod mean–variance example.

Expiry time T	30 years
Initial wealth	0
Rebalancing frequency	Yearly
Cash injection $\{q_i\}_{i=0,\dots,29}$	10
Real interest rate r	0.00827
Volatility σ	0.14777
μ	0.08885
λ	0.3222
η_1	4.4273
η_2	5.262
p_u	0.2758
$x_{\max}$	$\log(100) + 5$
$x_{\min}$	$\log(100) - 10$
$\varepsilon_1, \varepsilon_2$	10^{-6}
Asymptotic form $E[(W_T - W^*)^2]$, $x \to \infty$	$A(x) = 0$

Parameters are determined by fitting to the real (inflation adjusted) CRSP data for the period January 1926 to December 2015. Interest rate is the average real return on ninety-day Treasury bills.

TABLE 7 Test of convergence of optimal multiperiod mean–variance investment strategy.

N_x	N_b	Value function	Ratio	$E[W_T]$	Ratio	SD$[W_T]$	Ratio
512	305	97 148.899100	N/A	824.02599269	N/A	240.73884508	N/A
1024	609	97 042.740997	N/A	824.07104985	N/A	240.55534019	N/A
2048	1217	97 014.471301	3.8	824.09034690	2.3	240.51245396	4.3
4096	2433	97 007.286530	3.9	824.08961667	−26.0	240.49691620	2.7
8192	4865	97 005.451814	3.9	824.09295889	−0.22	240.49585213	14.6

Monotone method, linear basis functions. Parameters are as in Table 6. Fixed $W^* = 1022$. "Ratio" is the ratio of successive changes.

Carlo simulations to directly compute the mean and standard deviation of the final wealth. The results are shown in Table 8.

Of more practical interest is the following computation. In Table 9, we show the results obtained by rebalancing to a constant weight in equities at each monitoring date. We specify that the portfolio is rebalanced to 0.60 in stocks and 0.40 in bonds (a common default recommendation). We then solve for the value function using the monotone Fourier method, allowing W^* to vary, but fixing the expected value so that $E[W_T]$ is the same as for the 60:40 constant proportion strategy. This is done by using a Newton iteration, where each evaluation of the residual function requires a solution for the value function and the expected value equation. The results of this test are shown in Table 10. In this case, fixing the mean and allowing W^* to

TABLE 8 Monte Carlo simulation results, based on optimal controls from solving for the value function using the monotone Fourier technique.

N_{sim}	$E[W_T]$	SD$[W_T]$
1.6×10^5	824.3425 (1.55)	240.2263
6.4×10^5	823.6719 (0.78)	240.7278
2.56×10^6	824.0077 (0.39)	240.4336
1.024×10^7	824.1043 (0.19)	240.5217

Numbers in brackets are the standard error, at a 99% confidence level, for the mean. Compare with Table 7. Parameters are as in Table 6. Fixed $W^* = 1022$.

TABLE 9 Portfolio rebalanced to 0.60 in stocks and 0.40 in bonds at each monitoring date.

$E[W_T]$	SD$[W_T]$	Median$[W_T]$
824.10047	511.8482	704

Closed-form expressions for mean and standard deviation. Median is computed using Monte Carlo simulation. Parameters are as in Table 6.

TABLE 10 At each refinement level W^* is determined so that $E[W_T] = 824.10047$.

N_x	N_b	$E[W_T]$	SD$[W_T]$	Ratio
512	305	824.10047	240.79440842	N/A
1024	609	824.10047	240.57925928	N/A
2048	1217	824.10047	240.52022512	3.6
4096	2433	824.10047	240.50571976	4.1
8192	4865	824.10047	240.50220544	4.1

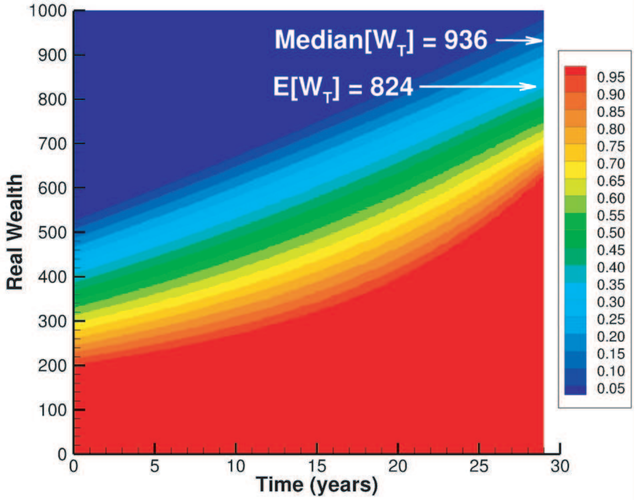
The median on the finest grid is computed by storing the controls and using Monte Carlo simulation. Median$[W_T] =$ 936. "Ratio" is the ratio of successive changes. Parameters are as in Table 6.

vary results in smooth convergence of the standard deviation. From a practical point of view, we can see that the optimal strategy has the same expected value as the constant proportion strategy, but the standard deviation is reduced from 512 to 241, and the median of the optimal strategy is 936 compared with a median of 704 for the constant proportion strategy. A heat map of the optimal strategy is shown in Figure 3.

10 CONCLUSIONS

Many problems in finance give rise to discretely monitored complex control problems. In many cases, the optimal controls are not of a simple bang-bang type. A numerical procedure must then be used to determine the optimal control at discrete

FIGURE 3 Optimal strategy, fraction of portfolio invested in stock, as a function of current total real wealth $W_t = S_t + B_t$ and forward time t.



Parameters are as in Table 6.

points in the physical domain. In these situations, there is little hope of obtaining a high-order accurate solution after the control is applied. If we desire a monotone scheme, which increases robustness and reliability for our computations, then we are limited to the use of linear interpolation; hence, we can obtain at most second-order accuracy.

Traditional FST/CONV methods assume knowledge of the Fourier transform of the Green's function, but then approximate this function by a truncated Fourier series. As a result, these methods are not monotone. Instead, when the Fourier transform of the Green's function is known, we carry out a preprocessing step by projecting the Green's function (in the physical space) onto a set of linear basis functions. These integrals can then be computed to within a specified tolerance, allowing us to guarantee a monotone scheme to within the tolerance. This monotone scheme is robust to small time steps, which is observably not the case for the standard FST/CONV methods, and indeed this lack of robustness is a major pitfall of the latter.

When the Green's function depends on time only through the time-step size and the monitoring dates for the control are equally spaced (which is typically the case), the final monotone algorithm has the same complexity per step as the original FST/CONV algorithms, and the same order of convergence for smooth control problems.

It is a simple process to add this preprocessing step to existing FST/CONV software. This results in more robust and more reliable algorithms for optimal stochastic control problems.

DECLARATION OF INTEREST

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper.

ACKNOWLEDGEMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The results presented in Section 9 were calculated based on data from Historical Indexes, © 2015 Center for Research in Security Prices (CRSP), The University of Chicago Booth School of Business. Wharton Research Data Services (WRDS) was used in preparing this article. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers.

REFERENCES

Alonso-Garcia, J., Wood, O., and Ziveyi, J. (2018). Pricing and hedging guaranteed minimum withdrawal benefits under a general Lévy framework using the COS method. *Quantitative Finance* **18**(6), 1049–1075 (https://doi.org/10.1080/14697688.2017.1357832).

Angelini, F., and Herzel, S. (2014). Delta hedging in discrete time under stochastic interest rate. *Journal of Computational and Applied Mathematics* **259**, 385–393 (https://doi.org/10.1016/j.cam.2013.06.022).

Azimzadeh, P., Bayraktar, E., and Labahn, G. (2018). Convergence of implicit schemes for Hamilton–Jacobi–Bellman quasi-variational inequalities. *SIAM Journal on Control and Optimization* **56**(6), 3994–4016.

Barles, G., and Souganidis, P. (1991). Convergence of approximation schemes for fully nonlinear equations. *Asymptotic Analysis* **4**, 271–283.

Bauer, D., Kling, A., and Russ, J. (2008). A universal pricing framework for guaranteed minimum benefits in variable annuities. *ASTIN Bulletin* **38**, 621–651 (https://doi.org/10.1017/S0515036100015312).

Bauerle, N., and Grether, S. (2015). Complete markets do not allow free cash flow streams. *Mathematical Methods of Operations Research* **81**, 137–146 (https://doi.org/10.1007/s00186-014-0489-2).

Bokanowski, O., Picarelli, A., and Reisinger, C. (2018). High-order filtered schemes for time dependent second order HJB equations. *ESAIM: Mathematical Modelling and Numerical Analysis* **52**(1), 69–97 (https://doi.org/10.1051/m2an/2017039).

Chen, Z., and Forsyth, P. A. (2008). A numerical scheme for the impulse control formulation for pricing variable annuities with a guaranteed minimum withdrawal benefit (GMWB). *Numerische Mathematik* **109**, 535–569 (https://doi.org/10.1007/s00211-008-0152-z).

Chen, Z., Vetzal, K., and Forsyth, P. A. (2008). The effect of modelling parameters on the value of GMWB guarantees. *Insurance: Mathematics and Economics* **43**(1), 165–173 (https://doi.org/10.1016/j.insmatheco.2008.04.003).

Cong, F., and Oosterlee, C. W. (2016). Multi-period mean–variance portfolio optimization based on Monte-Carlo simulation. *Journal of Economic Dynamics and Control* **64**, 23–38 (https://doi.org/10.1016/j.jedc.2016.01.001).

Cui, X., Li, D., Wang, S., and Zhu, S. (2012). Better than dynamic mean–variance: time inconsistency and free cash flow stream. *Mathematical Finance* **22**(2), 346–378 (https://doi.org/10.1111/j.1467-9965.2010.00461.x).

Cui, X., Gao, J., Li, X., and Li, D. (2014). Optimal multi-period mean–variance policy under no-shorting constraint. *European Journal of Operational Research* **234**, 459–468 (https://doi.org/10.1016/j.ejor.2013.02.040).

Dai, M., Kwok, Y., and Zong, J. (2008). Guaranteed minimum withdrawal benefit in variable annuities. *Mathematical Finance* **184**, 595–611 (https://doi.org/10.1111/j.1467-9965.2008.00349.x).

Dang, D.-M., and Forsyth, P. A. (2014). Continuous time mean–variance optimal portfolio allocation under jump diffusion: a numerical impulse control approach. *Numerical Methods for Partial Differential Equations* **30**, 664–698 (https://doi.org/10.1002/num.21836).

Dang, D.-M., and Forsyth, P. A. (2016). Better than pre-commitment mean–variance portfolio allocation strategies: a semi-self-financing Hamilton–Jacobi–Bellman equation approach. *European Journal of Operational Research* **250**, 827–841 (https://doi.org/10.1016/j.ejor.2015.10.015).

Deng, G., Dulaney, T., McCann, C., and Yan, M. (2017). Efficient valuation of equity-indexed annuities under Lévy processes using Fourier cosine series. *The Journal of Computational Finance* **21**(2), 1–27 (https://doi.org/10.21314/JCF.2017.331).

Fang, F., and Oosterlee, C. (2008). A novel pricing method for European options based on Fourier cosine series expansions. *SIAM Journal on Scientific Computing* **31**, 826–848 (https://doi.org/10.1137/080718061).

Forsyth, P., and Vetzal, K. (2017). Robust asset allocation for long-term target-based investing. *International Journal of Theoretical and Applied Finance* **20**(3), 1750017 (https://doi.org/10.1142/S0219024917500170).

Garroni, M. G., and Menaldi, J. L. (1992). *Green Functions for Second Order Parabolic Integro-Differential Problems*. Longman Scientific, New York.

Huang, H.-C. (2010). Optimal multiperiod asset allocation: matching assets to liabilities in a discrete model. *Journal of Risk and Insurance* **77**, 451–472 (https://doi.org/10.1111/j.1539-6975.2009.01350.x).

Huang, Y., Zeng, P., and Kwok, Y. (2017). Optimal initiation of a guaranteed lifelong withdrawal with dynamic controls. *SIAM Journal on Financial Mathematics* **8**, 804–840 (https://doi.org/10.1137/16M1089575).

Ignatieva, K., Song, A., and Ziveyi, J. (2018). Fourier space time-stepping algorithm for valuing guaranteed minimum withdrawal benefits in variable annuities under regime-switching and stochastic mortality. *ASTIN Bulletin* **48**, 139–169 (https://doi.org/10.1017/asb.2017.23).

Jackson, K., Jaimungal, S., and Surkov, V. (2008). Fourier space time-stepping for option pricing with Lévy models. *The Journal of Computational Finance* **12**(2), 1–29 (https://doi.org/10.21314/JCF.2008.178).

Kou, S., and Wang, H. (2004). Option pricing under a double exponential jump diffusion model. *Management Science* **50**(9), 1178–1192 (https://doi.org/10.1287/mnsc.1030.0163).

Li, D., and Ng, W.-L. (2000). Optimal dynamic portfolio selection: multiperiod mean–variance formulation. *Mathematical Finance* **10**, 387–406 (https://doi.org/10.1111/1467-9965.00100).

Lippa, J. (2013). A Fourier space time-stepping approach applied to problems in finance. MMath Thesis, University of Waterloo.

Lord, R., Fang, F., Bervoets, R., and Oosterlee, C. (2008). A fast and accurate FFT based method for pricing early-exercise options under Lévy processes. *SIAM Journal on Scientific Computing* **30**, 1678–1705 (https://doi.org/10.1137/070683878).

Menoncin, F., and Vigna, E. (2017). Mean–variance target based optimisation for defined contribution pension schemes in a stochastic framework. *Insurance: Mathematics and Economics* **76**, 172–184 (https://doi.org/10.1016/j.insmatheco.2017.08.002).

Obermann, A. (2006). Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton–Jacobi equations and free boundary problems. *SIAM Journal of Numerical Analysis* **44**(2), 879–895 (https://doi.org/10.1137/S0036142903435235).

Pooley, D., Forsyth, P., and Vetzal, K. (2003). Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis* **23**, 241–267 (https://doi.org/10.1093/imanum/23.2.241).

Remillard, B., and Rubenthaler, S. (2013). Optimal hedging in discrete time. *Quantitative Finance* **13**, 819–825 (https://doi.org/10.1080/14697688.2012.745012).

Ruijter, M., Oosterlee, C., and Albers, R. (2013). On the Fourier cosine series expansion method for stochastic control problems. *Numerical Linear Algebra with Applications* **20**, 598–625 (https://doi.org/10.1002/nla.1866).

Shan, C. (2014). Commodity options pricing by the Fourier space time-stepping method. MMath Essay, University of Waterloo.

Surkov, A. (2010). Option pricing using Fourier space time-stepping framework. PhD Thesis, University of Toronto.

Tanskanen, A., and Lukkarinen, J. (2003). Fair valuation of path-dependent participating life insurance contracts. *Insurance: Mathematics and Economics* **33**, 595–609 (https://doi.org/10.1016/j.insmatheco.2003.08.003).

Vigna, E. (2014). On efficiency of mean–variance based portfolio selection in defined contribution pension schemes. *Quantitative Finance* **14**, 237–258 (https://doi.org/10.1080/14697688.2012.708778).

Zhang, B., Grezelak, L., and Oosterlee, C. (2012). Efficient pricing of commodity options with early-exercise under the Ornstein–Uhlenbeck process. *Applied Numerical Mathematics* **62**, 91–111 (https://doi.org/10.1016/j.apnum.2011.10.005).

Zheng, W., and Kwok, Y. (2014). Fourier transform algorithms for pricing and hedging discretely sampled exotic variance products and volatility derivatives under additive processes. *The Journal of Computational Finance* **18**(2), 3–30 (https://doi.org/10.21314/JCF.2014.278).

Zhou, X. Y., and Li, D. (2000). Continuous-time mean–variance portfolio selection: a stochastic LQ framework. *Applied Mathematics and Optimization* **42**, 19–33 (https://doi.org/10.1007/s002450010003).