ϵ-Monotone Fourier Methods for Optimal Stochastic Control in Finance [*]

Peter A. Forsyth [†] George Labahn[‡]

October 23, 2017

Abstract

Stochastic control problems in finance having complex controls inevitably give rise to low order accuracy, usually at most second order. Fourier methods are efficient at advancing the solution between control monitoring dates, but are not monotone. This gives rise to possible violations of arbitrage inequalities. We devise a preprocessing step for Fourier methods which 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. The algorithm has the same complexity per step as a standard Fourier method and has second order accuracy for smooth problems.

Keywords: Monotonicity, Fourier methods, discrete comparison, optimal stochastic control, finance

Running Title: Monotone Fourier

1 Introduction

Optimal stochastic control problems in finance often involve monitoring or making decisions at discrete points in time. This happens, for example, in problems such as asset allocation (Li and Ng, 2000; Huang, 2010; Forsyth and Vetzal, 2017; Cong and Oosterlee, 2016), pricing of variable annuities (Bauer et al., 2008; Dai et al., 2008; Chen et al., 2008; Ignatieva et al., 2016; Alonso-Garcia et al., 2017; Huang et al., 2018), and hedging in discrete time (Remillard and Rubenthaler, 2013; Angelini and Herzel, 2014) to name just a few.

These optimal control problems are typically modeled as the solutions of Partial Integro Differential 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 backwards in time between each decision point, and then determining the optimal control at each such point. In many cases, including for

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

[†] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1, paforsyt@uwaterloo.ca, +1 519 888 4567 ext. 34415.

[‡] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1, glabahn@uwaterloo.ca, +1 519 888 4567 ext. 34667

example those mentioned above, the models are based on fairly simple stochastic processes, with the main interest being the behaviour of the optimal controls. These simple stochastic models can be justified if one is looking at long term problems, for example, variable annuities or saving for retirement, where the time scales are of the order of 10-30 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 type of PIDEs 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 a discrete approximation to the solution which can be used for effective numerical computation. A natural approach is then to use a Fourier scheme to advance the solution in a single time step between decision times, and then apply a numerical optimization approach to advance the solution across the decision time. This technique is repeated until the current time is reached (Ruijter et al., 2013; Lippa, 2013). These methods are based on Fourier Space Timestepping (FST) (Jackson et al., 2008), the CONV technique (Lord et al., 2008) or the COS algorithm (Fang and Oosterlee, 2008). Fourier methods have been applied to pricing of exotic variance products and volatility derivatives (Zheng and Kwok, 2014), guaranteed minimum withdrawal benefits (Ignatieva et al., 2016; Alonso-Garcia et al., 2017; Huang et al., 2018) and equity-indexed annuities (Deng et al., 2017) to name just a few.

Fourier methods have a number of advantages compared to finite difference and other methods. First and foremost is that there are no timestepping 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 these methods typically need a large number of discretization points, the algorithms reduce to using finite FFTs which are efficiently available on most platforms (including also GPUs). 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. Detailed knowledge of PDE algorithms is not actually required in this case. Fourier methods also easily extend to multi-factor stochastic process where finite difference methods have difficulties because of cross derivative terms. Finally, Fourier methods have good convergence properties for problems with non-complex 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 (in terms of number of terms of the Fourier series) (Fang and Oosterlee, 2008).

A major drawback with current, existing 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 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 which 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 non-monotone 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 (Barles and Souganidis, 1991) that a discretization of a PIDE converges to the viscosity solution if it is monotone, consistent and stable. There are examples (Obermann, 2006) where non-monotone discretizations fail to converge 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 non-monotonic Fourier methods require a very small space step which often results in numerical issues. Finally, monotone schemes are more reliable for the numeric computation of Greeks (i.e. derivatives of the solution), often important information for financial instruments.

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 non-smooth, and 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, 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 one can modify the FST or CONV schemes 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 multiply 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 tolerance. That is, we prove that our algorithm has an $\epsilon-$Discrete Comparison Property, that is, given a tolerance ϵ, then a discrete comparison (a.k.a. arbitrage inequality) holds to $O(\epsilon)$, independent of the number of discretization nodes and timesteps. 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 set up cost the complexity per step is the same as 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 tolerance) independent of the number of FST (CONV) grid nodes or 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 into 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 non-smooth jumps (e.g. cash flows) 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 inherently be 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 problem can be traced to the difficulty of reconstructing the Fourier coefficients after numerically determining the optimal control at discrete points in the physical space. Consequently, in this article 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 behaviour of our algorithm by comparing various implementations of FST on some model option pricing examples, in particular European and Bermudean 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 which 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 set up cost. After the initialization, the complexity per step of the monotone method is the same as 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 is used to describe existing Fourier methods which allows us to contrast 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 appearing in the following section. Wrap-around is an important issue for Fourier methods, particularly in the case of our control problems. Our method of minimizing such 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 a conclusion 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

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

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

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

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

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

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 backwards 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

$$\begin{aligned} v_\tau - \mathcal{L}v &= 0 \qquad\qquad ; \quad \tau \in (\tau_n^+, \tau_{n+1}^-) && (2.4)\\ v(x, \tau_n^+) &= \inf_c \mathcal{M}(c) v(x, \tau_n^-) \quad ; \quad \tau_n \in \mathcal{T} \qquad . && (2.5) \end{aligned}$$

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

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

A typical intervention operator has the form

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

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 cash flows 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 assume that the intervention operator has the form (2.6). In our asset allocation example (described later), the cash flows are modeled by updating a path dependent variable.

3 CONV and FST Methods

In this section, we derive the FST and closely related CONV technique in an intuitive fashion. This will allow us to contrast these methods with the monotone technique developed in the following section. For ease of exposition, we will continue to restrict attention to one dimensional problems. However, there is no difficulty generalizing this approach to the multi-dimensional case. In a financial context, 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)

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

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 Green's function will have the form $g = g(x - x', \Delta\tau)$ and we will assume this to hold in our problems.[1] The exact solution of our PIDE is then

$$v(x, \tau + \Delta\tau) \quad = \quad \int_{\mathbb{R}} g(x - x', \Delta\tau) v(x', \tau) \; dx' \; . \tag{3.1}$$

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

$$\int_{\mathbb{R}} g(x, \Delta\tau) \; dx \quad = \quad C_1 \;\; \leq \;\; 1 \quad \text{and} \quad g(x, \Delta\tau) \geq 0 \tag{3.2}$$

[1] If we view the Green's function as a scaled conditional probability density f then our assumption is that $f(y|x)$ only depends on x and y via their difference $f(y|x) = f(y - x)$.

are particularly important.[2] These properties are formally proven in (Garroni and Menaldi, 1992), but 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

$$\begin{aligned} G(\omega, \Delta\tau) &= \int_{-\infty}^{\infty} g(x, \Delta\tau) e^{-2\pi i \omega x} \, dx \\ g(x, \Delta\tau) &= \int_{-\infty}^{\infty} G(\omega, \Delta\tau) e^{2\pi i \omega x} \, d\omega \; . \end{aligned} \tag{3.3}$$

We will assume that a closed form expression for $G(\omega, \Delta\tau)$ is available, that is, that we know the Fourier transform of our Green's function. This is the case, for example, in situations where the characteristic function of the underlying stochastic process is known.

As is typically the case, we assume that $g(x, \Delta\tau)$ decays to zero as $|x| \to \infty$ so that $g(x, \Delta\tau)$ 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 equation (3.1) so that $x \in [x_{\min}, x_{\max}]$. We can therefore replace the Fourier transform pair (3.3) by their Fourier series equivalent

$$\begin{aligned} G(\omega_k, \Delta\tau) &\simeq \int_{x_{\min}}^{x_{\max}} g(x, \Delta\tau) \; e^{-2\pi i \omega_k x} \, dx \\ \hat{g}(x, \Delta\tau) &= \frac{1}{P} \sum_{k=-\infty}^{\infty} G(\omega_k, \Delta\tau) \; e^{2\pi i \omega_k x} \end{aligned} \tag{3.4}$$

with $P = x_{\max} - x_{\min}$ and $\omega_k = \frac{k}{P}$. Here the scaling factors in equation (3.4) are selected to be consistent with the scaling in (3.3). The solution of the PIDE (3.1) is then approximated as

$$v(x, \tau + \Delta\tau) \simeq \int_{x_{\min}}^{x_{\max}} \hat{g}(x - x', \Delta\tau) v(x', \tau) \; dx' \; . \tag{3.5}$$

Note that the Fourier series (3.4) implies a periodic extension of $\hat{g}$, that is, $\hat{g}(x + P, \tau) = g(x, \tau)$. The localization assumption also then implies that $v(x, \tau)$ is periodically extended.

Substituting the Fourier series (3.4) into (3.5) gives our approximate solution as

$$v(x, \tau + \Delta\tau) = \frac{1}{P} \sum_{k=-\infty}^{\infty} G(\omega_k, \Delta\tau) e^{2\pi i \omega_k x} \int_{x_{\min}}^{x_{\max}} v(x', \tau) e^{-2\pi i \omega_k x'} \, dx'. \tag{3.6}$$

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

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

Then the integral in (3.6) can be approximated by a quadrature rule with weights w_ℓ giving

$$\begin{aligned} \int_{x_{\min}}^{x_{\max}} v(x', \tau) e^{-2\pi i \omega_k x'} \, dx' &\simeq \sum_{\ell=-N/2}^{N/2-1} w_\ell \; v(x'_\ell, \tau) e^{-2\pi i \frac{k}{P} x'_\ell} \Delta x \\ &= P e^{-2\pi i \frac{k}{P} \hat{x}_0} V(\omega_k, \tau) \end{aligned} \tag{3.7}$$

[2] For the examples in this paper the constant C_1 is explicitly given (in each example) in Appendix A

where

$$V(\omega_k, \tau) \;=\; \frac{1}{N} \sum_{\ell=-N/2}^{N/2-1} w_\ell \; v(x'_\ell, \tau) e^{-2\pi i k \ell / N} \tag{3.8}$$

is the DFT of $\{w_j \; v(x'_j, \tau)\}$. Using equations (3.7) and (3.8) in equation (3.6), and truncating the sum to $k \in [-N/2, N/2-1]$ then gives

$$\begin{aligned} v(x_j, \tau + \Delta\tau) \;&=\; \frac{1}{P} \sum_{k=-N/2}^{N/2-1} e^{2\pi i \frac{k}{P} \hat{x}_0} G(\omega_k, \Delta\tau) e^{2\pi i k j / N} P e^{-2\pi i \frac{k}{P} \hat{x}_0} V(\omega_k, \tau) \\ &=\; \sum_{k=-N/2}^{N/2-1} G(\omega_k, \Delta\tau) V(\omega_k, \tau) e^{2\pi i k j / N}. \end{aligned} \tag{3.9}$$

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

In summary, one can obtain a discrete set of values for the solution v by first going to the Fourier domain via 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 the cost of doing a single FFT and IFFT.

There are four significant approximations in these steps. These include localization of the computational domain, representation of the Green's function by a truncated Fourier series, a periodic extension of the solution and, finally, approximation of the integral in equation (3.6) by a quadrature rule. The effect of the errors from these approximations has been previously discussed in (Lord et al., 2008) and we refer the reader there for details.

3.2 The FST/CONV Algorithms

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

$$\begin{aligned} (v^n)^+ \;&=\; [v(x_{-N/2}, \tau_n^+), \ldots, v(x_{N/2-1}, \tau_n^+)] \\ w_{quad} \;&=\; [w(x_{-N/2}), \ldots, w(x_{N/2-1})] \end{aligned} \tag{3.10}$$

and let $\mathbb{I}_{\Delta x}(x)$, with $x_k \leq x \leq x_{k+1}$, be a linear interpolation operator

$$\mathbb{I}_{\Delta x}(x)(v^n)^+ \;=\; \theta \cdot v(x_k, \tau_n^+) \;+\; (1-\theta) \cdot v(x_{k+1}, \tau_n^+) \quad \text{with} \quad \theta = \frac{(x_{k+1} - x)}{\Delta x}. \tag{3.11}$$

The full FST/CONV algorithm applied to a control problem is illustrated in Algorithm 1. We refer the reader to (Lippa, 2013; Ignatieva et al., 2016; Huang et al., 2018) for applications in finance.

Remark 3.1. *In (Jackson et al., 2008), the authors describe their FST method in slightly different terms. 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 which can be solved in closed-form as long as the characteristic function of the associated stochastic process is known in closed form (See Appendix A). 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.*

Algorithm 1 FST/CONV Fourier method. $x \circ y$ is the Hadamard product of vectors x,y.

```
Require: $G = \{G(\omega_j, \Delta\tau)\},\ j = -N/2, \ldots, N/2-1$
Input: number of timesteps $M$ and initial solution $(v^0)^-$
$(v^0)^+ = \inf_c \mathcal{M}(c)\big(\ \mathbb{I}_{\Delta x}(x)(v^0)^-\ \big)$
for $m = 1, \ldots, M$ do { Timestep loop}
    $V^{m-1} = FFT[\ w_{quad} \circ (v^{m-1})^+\ ]$                                {Frequency domain}
    $(v^m)^- = IFFT[\ V^{m-1} \circ G\ ]$                                         {Physical domain}
    $v(x_j, \tau_m^+) = \inf_c \mathcal{M}(c)\big(\ \mathbb{I}_{\Delta x}(x_j)(v^m)^-\ \big)\ ;\ j = -\frac{N}{2}, \ldots, j = \frac{N}{2} - 1$    {Optimal control}
end for
```

4 An ϵ-Monotone Fourier Method

Our monotone Fourier method proceeds in a similar fashion as in the previous section, but is based on a slightly different philosophy. We begin by discretizing the value function, and then generate a continuous approximation of the 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 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 set-up cost, the complexity per step is the same as the standard FST or 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) independent of the number of FST (CONV) grid nodes or timestep size.

4.1 A Monotone Scheme

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

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

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

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

where $P = x_{\max} - x_{\min}$ and $\Delta x = \frac{P}{N}$ with $x_{\min} = \hat{x}_0 - \frac{N\Delta x}{2}$ and $x_{\max} = \hat{x}_0 + \frac{N\Delta x}{2}$. We 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, that is, hat functions over $[x_j - \Delta x, x_j + \Delta x]$ which are one at x_j and zero at all other $x_i \neq x_j$. Substituting representation (4.2) into equation

(4.1) gives

$$
\begin{aligned}
v_k(\tau + \Delta\tau) &= \int_{x_{\min}}^{x_{\max}} g(x_k - x, \Delta\tau)\ v(x,\tau)\ dx \\
&= \sum_{j=-N/2}^{N/2-1} v_j(\tau) \int_{x_{\min}}^{x_{\max}} \phi_j(x)\ g(x_k - x, \Delta\tau)\ dx \\
&= \sum_{j=-N/2}^{N/2-1} v_j(\tau)\ \widetilde{g}(x_k - x_j, \Delta\tau)\Delta x\ , \qquad (4.3)
\end{aligned}
$$

where

$$
\widetilde{g}(x_k - x_j, \Delta\tau) \ = \ \frac{1}{\Delta x} \int_{x_k - x_j - \Delta x}^{x_k - x_j + \Delta x} \phi_{k-j}(x)\ g(x, \Delta\tau)\ . \qquad (4.4)
$$

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

$$
\widetilde{g}(y_\ell, \Delta\tau) \ = \ \frac{1}{\Delta x} \int_{y_\ell - \Delta x}^{y_\ell + \Delta x} \phi_\ell(x)\ g(x, \Delta\tau)\ dx \qquad (4.5)
$$

as the averaged projection of the Green's function onto the basis functions ϕ_ℓ. Note that for this projection $\widetilde{g}(y_\ell, \Delta\tau) \geq 0$ since the exact Green's function has $g(x) \geq 0$ for all x, and of course $\phi_\ell(x) \geq 0$. Therefore the scheme (4.3) is monotone for any N.

4.2 Approximating the Monotone Scheme

The scheme (4.3) is monotone since the weights $\widetilde{g}(y_\ell, \Delta\tau)$ given in (4.5) 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 have then replaced our Green's function $g(x, \Delta\tau)$ by its localized, periodic approximation

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

with the projection of the Green's function onto the linear basis functions. Assuming uniform convergence of the Fourier series (see Appendix), we integrate equation (4.5) term by term, which gives

$$
\widetilde{g}(y_j, \Delta\tau) \ = \ \frac{1}{P} \sum_{k=-\infty}^{\infty} e^{2\pi i \omega_k y_j} \left(\frac{\sin^2 \pi\omega_k \Delta x}{(\pi\omega_k\Delta x)^2} \right) G(\omega_k, \Delta\tau). \qquad (4.6)
$$

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 equation (4.2) and $\alpha \in \{2,4,8,\ldots\}$, and suppose we now truncate the Fourier series for the projected linear basis form for $\widetilde{g}$ to N' terms. Let $\widetilde{g}(y_k, \Delta\tau, \alpha)$, denote the use of a truncated Fourier series with truncation parameter α for a fixed value of N. In this case the Fourier series (4.6) truncates to

$$\widetilde{g}(y_j, \Delta\tau, \alpha) = \frac{1}{P} \sum_{k=-\alpha N/2}^{\alpha N/2-1} e^{2\pi i \omega_k j \Delta x} \left(\frac{\sin^2 \pi\omega_k \Delta x}{(\pi \omega_k \Delta x)^2} \right) G(\omega_k, \Delta\tau) \ . \tag{4.7}$$

Using the notation $\widetilde{g}_j(\Delta\tau, \alpha) = \widetilde{g}(y_j, \Delta\tau, \alpha)$, then

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

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

Remark 4.1 (Efficient computation of the projections)**.** *It remains to compute the projections. For this one needs to determine the discrete convolution (4.7). Let*

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

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

$$\mathbb{Y}_\ell \ = \ \frac{1}{P} \sum_{k=-\alpha N/2}^{\alpha N/2-1} e^{2\pi i k \ell/(\alpha N)} Y_k \ ; \ \ell = -\frac{\alpha N}{2}, \ldots, \frac{\alpha N}{2} - 1 \tag{4.8}$$

gives $\{\mathbb{Y}_\ell\}$ *as the DFT of the* $\{Y_k\}$ *(of size* N'*). Consequently, using equations (4.7) and (4.8)*

$$\widetilde{g}(y_j, \Delta\tau, \alpha) = \mathbb{Y}_\ell \ ; \ \ell = j\alpha \ ; \ j = -N/2, \ldots, N/2 - 1 \ . \tag{4.9}$$

Thus the projections $\{\widetilde{g}(y_j, \Delta\tau, \alpha)\}$ *are computed via a single FFT.*

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

$$\widetilde{G}(\omega_k, \Delta\tau, \alpha) \ = \ \frac{P}{N} \sum_{m=-N/2}^{N/2-1} e^{-2\pi i m k/N} \ \widetilde{g}(y_m, \Delta\tau, \alpha). \tag{4.10}$$

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 (4.7) does not have to use the same number of terms (αN) as used in 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 (4.10). There is no further loss of information in this last step. As remarked above, we only use the Fourier representation of $\widetilde{g}(y_j, \Delta\tau, \alpha)$ to carry out the discrete convolution, that is a dense matrix vector multiply, efficiently. The discrete convolution in Fourier space is exactly equivalent to the discrete convolution in physical space, assuming periodic extensions.

Remark 4.2 (Relation to the COS method)**.** *In the COS method, the solution $v(x,\tau)$ is also expanded in a Fourier series. This gives 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 from applying an impulse control using a numerical optimization method at discrete points on a previous step, using linear interpolation (the only interpolation method which 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 4.3 (Piecewise constant basis functions)**.** *The equations and previous discussion in this section also holds if our basis functions are piecewise constant functions, that is, basis functions ϕ_j which are nonzero over $[x_j - \Delta x/2, x_j + \Delta x/2]$. In this case we would have*

$$\widetilde{g}(y_j, \Delta\tau) \;=\; \frac{1}{P}\sum_{k=-\infty}^{\infty} e^{2\pi i\omega_k y_j}\left(\frac{\sin \pi\omega_k \Delta x}{\pi\omega_k \Delta x}\right) G(\omega_k, \Delta\tau) \tag{4.11}$$

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 (4.3) 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

$$\begin{aligned} v_j(\tau) &= \sum_{\ell=-N/2}^{N/2-1} V(\omega_\ell,\tau) e^{2\pi i\omega_l x_j} &&= \sum_{\ell=-N/2}^{N/2-1}\left(e^{2\pi i\omega_\ell \hat{x}_0}\right) V(\omega_\ell,\tau) e^{2\pi i j\ell/N} \\ V(\omega_p,\tau) &= \frac{1}{N}\sum_{\ell=-N/2}^{N/2-1} e^{-2\pi i\omega_p x_\ell} v_\ell(\tau) &&= \frac{1}{N}\left(e^{-2\pi i\omega_p \hat{x}_0}\right)\sum_{\ell=-N/2}^{N/2-1} e^{-2\pi i p\ell/N} v_\ell(\tau)\ . \end{aligned} \tag{4.12}$$

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

$$\widetilde{g}_{k-j}(\Delta\tau,\alpha) \;=\; \frac{1}{P}\sum_{p=-N/2}^{N/2-1} \widetilde{G}(\omega_p,\Delta\tau,\alpha) e^{2\pi i(k-j)p/N}\ , \tag{4.13}$$

where we use equation (4.10) to determine $\widetilde{G}(\omega_p,\Delta\tau,\alpha)$. Substituting equations (4.13) and (4.12) into equation (4.3) we then get

$$\begin{aligned} v(x_k,\tau+\Delta\tau) &= \Delta x \sum_{j=-N/2}^{N/2-1} v_j^n\ \widetilde{g}(x_k - x_j,\Delta\tau,\alpha) \\ &= \frac{1}{N}\sum_{p=-N/2}^{N/2-1}\ \sum_{\ell=-N/2}^{N/2-1}\left(e^{2\pi i\omega_\ell x_0}\right)\widetilde{G}(\omega_p,\Delta\tau,\alpha) V(\omega_\ell,\tau) e^{2\pi i kp/N}\sum_{j=-N/2}^{N/2-1} e^{2\pi i j(\ell-p)/N} \\ &= \sum_{p=-N/2}^{N/2-1}\left(e^{2\pi i\omega_p x_0}\right) V(\omega_p,\tau)\widetilde{G}(\omega_p,\Delta\tau,\alpha) e^{2\pi i kp/N} \end{aligned} \tag{4.14}$$

where the last equation follows from the classical orthogonality properties of N^{th} roots of unity.

From equation (4.12) we have

$$V(\omega_p, \tau) \;=\; \frac{1}{N}\left(e^{-2\pi i \omega_p \hat{x}_0}\right) \sum_{\ell=-N/2}^{N/2-1} e^{-2\pi i p\ell/N} v_\ell(\tau) = \left(e^{-2\pi i \omega_p \hat{x}_0}\right) \widetilde{V}(\omega_p, \tau) \tag{4.15}$$

with

$$\widetilde{V}(\omega_p, \tau) \;=\; \frac{1}{N} \sum_{\ell=-N/2}^{N/2-1} e^{-2\pi i p\ell/N} v_\ell(\tau)$$

the DFT of $\{v_\ell(\tau)\}$. Finally substituting equation (4.15) into (4.14) gives

$$v(x_k, \tau + \Delta\tau) = \sum_{p=-N/2}^{N/2-1} \widetilde{V}(\omega_p, \tau)\; \widetilde{G}(\omega_p, \Delta\tau, \alpha)\; e^{2\pi i p\ell/N} \;, \tag{4.16}$$

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

Remark 4.4 (Monotonicity)**.** *Equations (4.16) and (4.3) are algebraic identities (assuming periodic extensions). Hence if we use (4.16) to advance the solution, then this is algebraically identical to using (4.3) to advance the solution. Thus we can analyse the properties of equation (4.16) by analysing equation (4.3). In particular, if* $\widetilde{g}(x_k, \Delta\tau, \alpha) \;\geq\; 0$ *then the scheme is monotone.*

Remark 4.5 (Converting FST or CONV to monotone form)**.** *Equation (4.16) is formally identical with equation (3.9). This has the practical result that any FST or CONV software can be converted to monotone form by a preprocessing step which computes* $\widetilde{G}(\omega_p, \Delta\tau, \alpha)$*, and choosing a trapezoidal rule for the integral in equation (3.7).*

5 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 earlier in equation (3.10) and $\mathbb{I}_{\Delta x}(x)$ the linear interpolation operator defined as in equation (3.11). Let

$$\widetilde{V}^n \;=\; [\widetilde{V}(\omega_{-N/2}, \tau_n), \ldots, \widetilde{V}(\omega_{N/2-1}, \tau_n)] \;=\; DFT[\; (v^n)^- \;]$$

and

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

Let us assume that our Green's function is not an explicit function of τ but rather we have $g \;=\; g(x - x', \Delta\tau)$ and that the time steps are all constant, that is, $\tau_{n+1} - \tau_n \;=\; \Delta\tau \;=\; const.$ In this case we can compute $\widetilde{G}(\omega_k, \Delta\tau, \alpha)$ only once. If these two assumptions do not hold, then $\widetilde{G}(\cdot)$ would have to be recomputed frequently and hence our algorithm for ensuring monotonicity becomes more costly.

Algorithm 2 describes the computation of $\widetilde{G}(\cdot)$. Here we test for monotonicity (up to a small tolerance) by minimizing the effect of any negative weights which are determined via

$$\sum_j \Delta x |\min(\widetilde{g}(y_j, \Delta\tau, \alpha), 0)| < \epsilon_1 \frac{\Delta\tau}{T}.$$

The test for accuracy of the projection occurs by the comparison

$$\max_j \Delta x |\widetilde{g}(y_j, \Delta\tau, \alpha) - \widetilde{g}(y_j, \Delta\tau, \alpha/2)| < \epsilon_2.$$

Both monotonicity and convergence tests are scaled by Δx so that these quantities are bounded as $\Delta\tau \rightarrow 0$ (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 timesteps from our monotonicity bounds. This is also discussed further in Section 6.

Algorithm 2 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 $\widetilde{g}(y_j, \Delta\tau, 1)$ .
 3: for $\alpha = 2, 4, \ldots$ until convergence do {Construct accurate $\widetilde{g}$}
 4:    Compute $\widetilde{g}(y_j, \Delta\tau, \alpha)$, $\widetilde{G}(\omega_j, \Delta\tau, \alpha)$, $j = -\frac{N}{2}, \ldots, \frac{N}{2} - 1$ using (4.9)-(4.10)
 5:    $test_1 = \sum_j \Delta x \min(\widetilde{g}(y_j, \Delta\tau, \alpha), 0)$            { Monotonicity test }
 6:    $test_2 = \max_j \Delta x |\widetilde{g}(y_j, \Delta\tau, \alpha) - \widetilde{g}(y_j, \Delta\tau, \alpha/2)|$            { Accuracy test }
 7:    if $(|test_1| < \epsilon_1(\Delta\tau/T))$ and $(test_2 < \epsilon_2)$ then
 8:       break from for loop                                   { Convergence test }
 9:    end if
10: end for{End accurate $\widetilde{g}$ loop}
11: Output: Weights $\widetilde{G}(\omega_j, \Delta\tau, \alpha)$, $j = -\frac{N}{2}, \ldots, \frac{N}{2} - 1$ in Fourier domain.
```

In Algorithm 2, 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 3.

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

Remark 5.2 (Complexity)**.** *The complexity of using (4.16) 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 3 Monotone Fourier method.

```
Require: Weights $\widetilde{G} = \{\widetilde{G}(\omega_j, \Delta\tau, \alpha)\}$, for $j = -\frac{N}{2}, \ldots, \frac{N}{2} - 1$ in Fourier domain (from Algor. 2).
Input: number of timesteps $M$, initial solution $(v^0)^-$
$(v^0)^+ = \inf_c \mathcal{M}(c)\big(\ \mathbb{I}_{\Delta x}(x)(v^0)^-\ \big)$
for $m = 1, \ldots, M$ do {Timestep loop}
   $\widetilde{V}^{m-1} = FFT[\ (v^{m-1})^+\ ]$                              {Frequency domain}
   $(v^m)^- = IFFT[\ \widetilde{V}^{m-1} \circ \widetilde{G}\ ]$                          {Physical domain}
   $v(x_j, \tau_m^+) = \inf_c \mathcal{M}(c)\big(\ \mathbb{I}_{\Delta x}(x_j)(v^m)^-\ \big)$ ; $j = -\frac{N}{2}, \ldots, \frac{N}{2} - 1$     {Optimal control}
end for{End timestep loop}
```

6 Properties of the Monotone Fourier Method

In this section we prove a number of properties satisfied by our ϵ-monotone Fourer 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 $C_1 = \int_{\mathbb{R}} g(x, \Delta\tau)\, dx$. Then for all k*

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

Proof.

$$\begin{aligned}
\Delta x \sum_{j=-N/2}^{N/2-1} \widetilde{g}(x_k - x_j, \Delta x, \alpha) \quad &= \quad \frac{P}{N} \sum_{\ell=-N/2}^{N/2-1} \widetilde{g}(y_\ell, \Delta\tau, \alpha) \ ; \quad \text{where } y_\ell = x_k - x_j \ , \ \ell = k - j \\
&= \quad \frac{P}{N} \sum_{\ell=-N/2}^{N/2-1} \frac{1}{P} \sum_{k=-\alpha N/2}^{\alpha N/2-1} e^{2\pi i \omega_k \ell \Delta x} \left(\frac{\sin^2 \pi\omega_k \Delta x}{(\pi\omega_k \Delta x)^2} \right) G(\omega_k, \Delta\tau) \\
&= \quad \frac{1}{N} \sum_{k=-\alpha N/2}^{\alpha N/2-1} \left(\frac{\sin^2 \pi\omega_k \Delta x}{(\pi\omega_k \Delta x)^2} \right) G(\omega_k, \Delta\tau) \sum_{\ell=-N/2}^{N/2-1} e^{2\pi i \ell k/N} \\
&= \quad G(0, \Delta\tau) \ = \ \int_{-\infty}^{\infty} g(x)\ dx = C_1 \ .
\end{aligned}$$

□

Theorem 6.1 (ℓ_∞ stability)**.** *Assume that $\widetilde{G}$ is computed using Algorithm 2, that $(v^n)^-$ is computed from*

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

and that

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

Then for every $0 \leq n \leq M$ we have

$$\|(v^n)^+\|_\infty \leq C_2 = e^{2\epsilon_1} \|(v^0)^-\|_\infty \ .$$

Proof. From equation (6.1)

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

which then implies

$$|(v_k^n)^-| \quad \leq \quad \|(v^{n-1})^+\|_\infty \sum_{j=-N/2}^{N/2-1} \Delta x \ \max(\widetilde{g}_{k-j}, 0) + \|(v^{n-1})^+\|_\infty \sum_{j=-N/2}^{N/2-1} \Delta x \ |\min(\widetilde{g}_{k-j},0)| \ .$$

From Lemma 6.1

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

and so

$$\begin{aligned} |(v_k^n)^-| \quad &\leq \quad \|(v^{n-1})^+\|_\infty \bigg(C_1 + 2 \sum_{j=-N/2}^{N/2-1} \Delta x \ |\min(\widetilde{g}_{k-j}, 0)| \bigg). \\ &\leq \quad \|(v^{n-1})^+\|_\infty (C_1 + 2\epsilon_1 \frac{\Delta \tau}{T}) \end{aligned} \tag{6.5}$$

using lines 5 and 7 in Algorithm 2. Since equation (6.5) is true for any k we have that

$$\|(v^n)^-\|_\infty \quad \leq \quad \|(v^{n-1})^+\|_\infty (C_1 + 2\epsilon_1 \frac{\Delta \tau}{T}) \ ,$$

which combined with equation (6.2) and using $C_1 \leq 1$ gives

$$\|(v^n)^+\|_\infty \quad \leq \quad \|(v^{n-1})^+\|_\infty (1 + 2\epsilon_1 \frac{\Delta \tau}{T}) \ .$$

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

$$\begin{aligned} \|(v^n)^+\|_\infty \quad &\leq \quad \|(v^0)^-\|_\infty (1 + 2\epsilon_1 \frac{\Delta \tau}{T})^n \\ &\leq \quad \|(v^0)^-\|_\infty \ e^{2\epsilon_1 n \frac{\Delta \tau}{T}} \quad \leq \quad \|(v^0)^-\|_\infty \ e^{2\epsilon_1} \quad = \quad C_2 \ . \end{aligned}$$

□

Remark 6.1 (Jump condition)**.** *We remark that (6.2), the jump condition $\|(v^n)^+\|_\infty \leq \|(v^n)^-\|_\infty$, is trivially satisfied if $\mathbb{I}_{\Delta x}(x)$ in line 6 in Algorithm 3 is a linear interpolant.*

From Theorem 6.1 and Remark 6.1 we have the immediate result,

Corollary 6.1 (Stability of Algorithm 3)**.** *Algorithm 3 is ℓ_∞ stable.*

Lemma 6.2 (Minimum value of solution.)**.** *Let $(v^n)^+$ be generated using equation 6.1, and set*

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

If the conditions for Lemma 6.1 are satisfied and

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

then

$$(v^n)^+_{\min} \quad \geq \quad (v^0)^-_{\min} \ (C_3)^n - C_2(e^{\epsilon_1} - 1)$$

where $C_2 \ = \ \|(v^0)^-\|_\infty e^{2\epsilon_1}$ is given in Lemma 6.1 and $C_3 = \sum_{j=-N/2}^{N/2-1} \Delta x \ \max(\widetilde{g}_{k-j},0)$.

Proof. From equation (6.1) and using equation (6.3) along with the definition of C_3 we obtain

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

Using Lemma 6.1 and lines 5 and 7 in Algorithm 2 then gives

$$(v_k^n)^- \quad \geq \quad (v^{n-1})^+_{\min} C_3 - C_2 \epsilon_1 \frac{\Delta\tau}{T}$$

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

$$(v^n)^+_{\min} \quad \geq \quad (v^{n-1})^+_{\min} C_3 - C_2 \epsilon_1 \frac{\Delta\tau}{T}.$$

Iterating implies

$$\begin{aligned}
(v^n)^+_{\min} &\geq (v^0)^+_{\min} {C_3}^n - C_2 \epsilon_1 \frac{\Delta\tau}{T} \left(\frac{1-{C_3}^n}{1-C_3} \right) \\
&\geq (v^0)^-_{\min} {C_3}^n - C_2 \epsilon_1 \frac{\Delta\tau}{T} \left(\frac{1-{C_3}^n}{1-C_3} \right) , \qquad (6.7)
\end{aligned}$$

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

$$C_3 \quad = \quad C_1 + \sum_{j=-N/2}^{N/2-1} \Delta x \ |\min(\widetilde{g}_{k-j},0)| \leq 1 + \epsilon_1 \frac{\Delta\tau}{T} \ , \qquad (6.8)$$

where the last inequality follows lines 5 and 7 in Algorithm 2 (and recalling that $C_1 \leq 1$). Combining equations (6.6), (6.7) and (6.8) and noting that $n\Delta\tau \leq T$ gives

$$(v^n)^+_{\min} \quad \geq \quad (v^0)^+_{\min} {C_3}^n - C_2(e^{\epsilon_1} - 1) \quad \geq \quad (v^0)^-_{\min} {C_3}^n - C_2(e^{\epsilon_1} - 1) \ .$$

□

Remark 6.2. *We note that condition 6.6, that is,* $(v^n)^+_{\min} \geq (v^n)^-_{\min}$ *is satisfied if* $\mathbb{I}_{\Delta x}(x)$ *in line 6 in Algorithm 3 is a linear interpolant.*

Theorem 6.2 (ϵ-Discrete Comparison Principle)**.** *Suppose we have two independent discrete solutions*

$$\begin{aligned}
(u^n)^+ &= [u(x_{-N/2}, \tau_n^+), \ldots, u(x_{+N/2-1}, \tau_n^+)] \\
(w^n)^+ &= [w(x_{-N/2}, \tau_n^+), \ldots, w(x_{+N/2-1}, \tau_n^+)] \qquad (6.9)
\end{aligned}$$

with

$$(u^0)^- \quad \geq \quad (w^0)^- \tag{6.10}$$

where the inequality is understood in the component-wise sense, and $(u^n)^+, (w^n)^+$ *are computed using Algorithm 3. If* $\widetilde{G}$ *is computed using Algorithm 2 and* $\mathbb{I}_{\Delta x}(x)$ *is a linear interpolant then*

$$(u^n)^+ - (w^n)^+ \quad \geq -\epsilon_1 \|(u^0 - w^0)^-\|_\infty + O(\epsilon_1^2) \ ; \ \epsilon_1 \to 0 \ . \tag{6.11}$$

Proof. Let $(z^n)^+ = (u^n)^+ - (w^n)^+$, $(z^n)^- = (u^n)^- - (w^n)^-$, then

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

Noting that

$$z_j(\tau_n^+) = \inf_c \mathcal{M}(c)\big(\ \mathbb{I}_{\Delta x}(x_j)(u^m)^- \ \big) - \inf_c \mathcal{M}(c)\big(\ \mathbb{I}_{\Delta x}(x_j)(w^m)^- \ \big) \tag{6.12}$$

then

$$|z_j(\tau_n^+)| \quad \leq \quad \sup_c \mathcal{M}(c)\big| \ \mathbb{I}_{\Delta x} \ (x_j)\big((u^m)^- - (w^m)^- \ \big) \ \big| \tag{6.13}$$

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

$$\|(z^n)^+\|_\infty \leq \|(z^n)^-\|_\infty \ . \tag{6.14}$$

Similarly

$$\begin{aligned} (z^n)^+_{\min} \quad &= \quad \min_j z_j(\tau_n^+) \\ &\geq \quad \min_j \inf_c \mathcal{M}(c) \ \mathbb{I}_{\Delta x}(x_j)\big((u^m)^- - (w^m)^- \ \big) \\ &\geq \quad (z^n)^-_{\min} \ . \end{aligned} \tag{6.15}$$

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

$$(z^n)^+_{\min} \quad \geq \quad (z^0)^-_{\min}(C_3)^n - e^{2\epsilon_1}\|(u^0 - w^0)^-\|_\infty(e^{\epsilon_1} - 1) \tag{6.16}$$

where $C_3 = \sum_{j=-N/2}^{N/2-1} \Delta x \ \ \max(\widetilde{g}_{k-j}{,}0)$. Since $(z^0)^-_{\min} \geq 0$ and $0 \leq C_3^n \leq e^{\epsilon_1}$, the result follows. □

Remark 6.3. *If Algorithm 2 is used to construct* $\widetilde{G}$ *for use in Algorithm 3, then the* ϵ*-discrete comparison property is satisfied for any* $N, \Delta\tau, M$ *up to order* ϵ_1. *Since typically* $\widetilde{g}(y_j, \Delta\tau, \alpha) \to \widetilde{g}(y_j, \Delta\tau, \infty) \geq 0$ *exponentially in* α*, in practice it is very inexpensive to make* ϵ_1 *as small as desired.*

Remark 6.4 (Continuously observed impulse control problems)**.** *By determining the optimal control at each timestep, we can apply our monotone Fourier method to the continuously observed impulse control problem*

$$\max\Big[v_\tau - \mathcal{L}v, v - \inf_c \mathcal{M}(c)v\Big] = 0 \ . \tag{6.17}$$

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. (2017), using the* ϵ *monotonicity property as in Bokanowski et al. (2018).*

7 Minimization of wrap-around error

The use of the convolution form for our solution (4.16) is rigorously correct for a periodic extension of the solution and the Green's function. In normal option pricing applications, the *wrap-around* 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 hence large errors may result (Lippa, 2013; Ruijter et al., 2013; Ignatieva et al., 2016). Hence we need to consider methods to reduce errors associated with wrap-around.

In order to minimize the effect of wrap-around 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

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

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 $\widetilde{G}(\omega_p, \Delta\tau, \alpha), p = -N^a/2, \ldots, p = N^a/2-1$ on this auxiliary grid. We then replace line 4 in Algorithm 3 by applying the DFT to the solution v on the auxiliary grid

$$\begin{aligned}
v(x_k, \tau_n^+)^a &= v(x_k, \tau_n^+) \;; \qquad k = -N/2, \ldots, k = N/2-1 \\
&= v(x_{-N/2}, \tau_n^+) \;; \quad k = -N^a/2, \ldots, -N/2-1 && (7.2)\\
&= A(x_k, \tau_n^+) \;; \qquad k = N/2, \ldots, N^a/2-1 \;, && (7.3)
\end{aligned}$$

where $A(x,\tau)$ is an asymptotic form of the solution, which we assume to be available from 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 $S = e^x$) is very small. We then carry out lines 4 - 5 of Algorithm 3 on the auxiliary grid and generate $(v^n)^-$ by discarding all the values on the auxiliary grid which are not on the original grid (as these are contaminated by wrap-around errors). The errors incurred by using extensions (7.2) and (7.3) can be made small by choosing $|x_{\min}|$ and $x_{\max}$ sufficiently large.

Remark 7.1 (Use of asymptotic form to reduce wrap-around error)**.** *Use of the above technique necessitates some changes to the proof of Theorem 6.2. 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.*

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

$$\frac{dS_t}{S_{t^-}} = (r - \lambda\kappa)dt + \sigma dZ + d\Bigg(\sum_{i=1}^{\pi_t}(\xi_i - 1)\Bigg) \quad \text{with} \quad \kappa = E[\xi] - 1 \tag{8.1}$$

where $E[\cdot]$ denotes the expectation operator. Here, dZ 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

Expiry time	.25 years
Strike K	100
Payoff	call
Initial asset price S_0	100
Risk-free rate r	.05
Volatility σ	.15
λ	.1
η_1	3.0465
η_2	3.0775
p_u	0.3445
$x_{\max}$	$\log(S_0) + 10$
$x_{\min}$	$\log(S_0) - 10$
ϵ_1, ϵ_2	10^{-6}
Asymptotic form $x \to \infty$	$A(x) = e^x$

TABLE 8.1: *European call option test.*

ξ_i are i.i.d. positive random variables. The density function $f(y)$, $y = \log(\xi)$ is assumed double exponential (Kou and Wang, 2004)

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

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.

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

$$\begin{aligned} v_\tau \;\; &= \;\; \frac{\sigma^2}{2} v_{xx} + (r - \frac{\sigma^2}{2} - \lambda \kappa) v_x - (r + \lambda) v + \lambda \int_{-\infty}^{+\infty} v(x + y) f(y) \; dy \\ &\quad \text{with} \quad v(x, 0) = \max(e^x - K, 0) \; . \end{aligned} \tag{8.4}$$

The Green's function for this problem is given in Appendix A.

The particular parameters for this test are given in Table 8.1 with the results appearing in Table 8.2. 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 $x_{\max}, |x_{\min}|$ altered results in the last 2 digits in the table.

In order to stress these Fourier methods, we repeat this example, except now using an expiry time of $T = .001$. Since the Green's function converges to a delta function as $T \to 0$, we can expect that this will 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 8.3. 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

	Monotone Methods				FST/CONV			
	Piecewise linear		Piecewise constant		Trapezoidal		Simpson	
N	Value	Ratio	Value	Ratio	Value	Ratio	Value	Ratio
2^9	3.9808516210		3.9443958729		3.9075619850		3.9784907318	
2^{10}	3.9753205007		3.9662547470		3.9571661688		3.9737010716	
2^{11}	3.9739391670	4.0	3.9716756819	4.0	3.9694107823	4.1	3.9734923202	23
2^{12}	3.9735939225	4.0	3.9730282349	4.0	3.9724624589	4.0	3.9734796846	17
2^{13}	3.9735076171	4.0	3.9733662066	4.0	3.9732247908	4.0	3.9734789013	16
2^{14}	3.9734860412	4.0	3.9734506895	4.0	3.9734153372	4.0	3.9734788524	16

TABLE 8.2: *European call option test: value at $\tau = 0, S = S_0$. Parameters in Table 8.1. N = number of nodes. Ratio is the ratio of successive changes.*

	Monotone Methods				FST/CONV			
	Piecewise linear		Piecewise constant		Trapezoidal		Simpson	
N	Value	Ratio	Value	Ratio	Value	Ratio	Value	Ratio
2^9	.19662316859		.94284763015		.24774086499		319.45747026	
2^{10}	.19467436458		.041410269769		.21909081933		521.62802838	
2^{11}	.19376651687	2.1	.15335986938	-8.0	.18611676723	.87	439.13444172	-2.5
2^{12}	.19346709107	3.0	.18477993505	3.6	.17728640855	3.7	27.002978049	0.2
2^{13}	.19339179620	4.0	.19127438852	4.8	.18913280108	-.75	.19367805822	15
2^{14}	.19337297842	4.0	.19284673379	4.1	.19231903134	3.7	.19338110881	9×10^4

TABLE 8.3: *European call option test: value at $\tau = 0, S = S_0$. Parameters in Table 8.1 but $T = .001$. N = number of nodes. Ratio is the ratio of successive changes.*

even results in values larger than S_0 at $S = S_0 = 100$, which violates the provable bound for a call option.

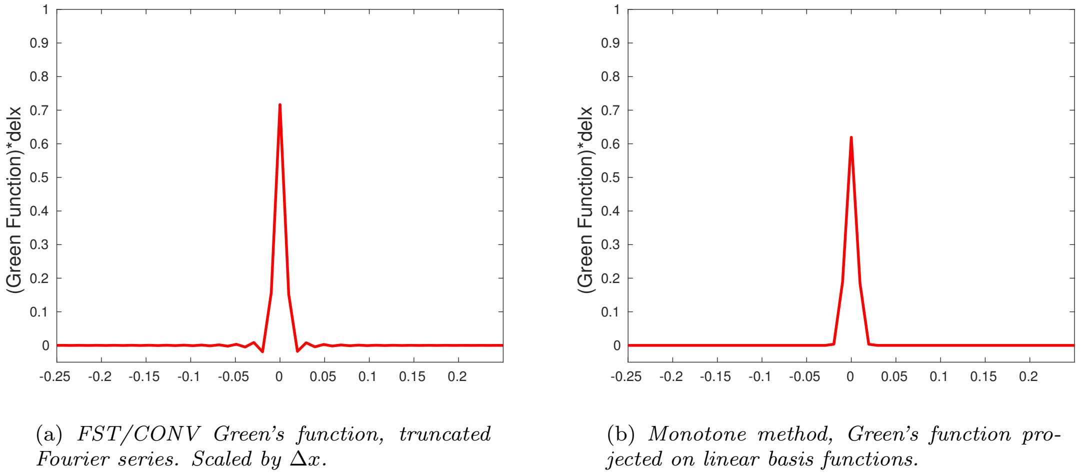
This phenomenon can be explained by examining Figure 8.1, which shows the projection of the Green's functions for the monotone method (piecewise linear basis function) and the truncated Green's function for the FST/CONV method. The projection of the Green's function for the monotone method in Figure 8.1(b) clearly has the expected properties: very peaked near $x = 0$ and non-negative for all x. In contrast, the FST/CONV numerical Green's function is oscillatory and negative for some values of x.

From these tests we can conclude both that the monotone method is robust for all timestep sizes and that for smooth problems and large timesteps, the monotone method exhibits the expected slower rate of convergence compared to high order techniques.

8.2 Bermudan option with non-proportional 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 normal

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



(a) *FST/CONV Green's function, truncated Fourier series. Scaled by Δx.*

(b) *Monotone method, Green's function projected on linear basis functions.*

FIGURE 8.1: *European call option test: parameters in Table 8.1 but $T = .001$. FST/CONV method truncated Fourier series ($N = 2048$). Monotone method shows $\widetilde{g}(x, \Delta\tau, \alpha)\Delta x$, with $N = 2048$, $\alpha = 4$.*

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

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

The expression $\max(e^x - D, e^{x_{\min}})$ in equation (8.6) ensures that the no-arbitrage condition holds, that is, 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 equation (8.6). The parameters for this problem are listed in Table 8.4 with the numerical results given in Table 8.5. All methods perform similarly, with second order convergence. We can see here 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

$$\frac{dS_t}{S_{t^-}} = (\mu - \lambda\kappa)dt + \sigma dZ + d\left(\sum_{i=1}^{\pi_t}(\xi_i - 1)\right) \tag{9.1}$$

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

$$dB_t = rB_t \ dt \ . \tag{9.2}$$

Expiry time	10 years
Strike K	100
Payoff	put
Initial asset price S_0	100
Risk-free rate r	.05
Volatility σ	.15
Dividend D	1.00
Monitoring frequency $\Delta\tau$	1.0 years
λ	.1
ν	-1.08
γ	.4
$x_{\max}$	$\log(S_0) + 10$
$x_{\min}$	$\log(S_0) - 10$
ϵ_1, ϵ_2	10^{-6}
Asymptotic form $x \to \infty$	$A(x) = 0$

TABLE 8.4: *Bermudan put option test.*

	Monotone Methods				FST/CONV			
	Piecewise linear		Piecewise constant		Trapezoidal		Simpson	
N	Value	Ratio	Value	Ratio	Value	Ratio	Value	Ratio
2^9	24.811127744		24.806532754		24.801967268		24.802639420	
2^{10}	24.789931363		24.788800257		24.787670043		24.787731820	
2^{11}	24.782264461	2.8	2.4.781982815	2.6	24.781701225	2.4	24.781787212	2.5
2^{12}	24.781134292	6.8	24.781063962	7.4	24.780993635	8.4	24.781007785	7.6
2^{13}	24.780822977	3.6	24.780805394	3.6	24.780787811	3.4	24.780788678	3.6
2^{14}	24.780744620	4.0	24.780740225	4.0	24.780735831	4.0	24.780737159	4.3

TABLE 8.5: *Bermudan put option test: value at $\tau = 0, S = S_0$. Parameters in Table 8.1. N = number of nodes. Ratio is the ratio of successive changes.*

The investor injects cash q_n at time 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

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

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

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

Since only cash withdrawals are allowed we have $c_n^* \geq 0$. 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

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

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

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

$$\text{subject to} \quad \begin{cases} (S_t, B_t) \text{ follow processes } (8.1), (9.2) \ ; \ t \notin \mathcal{T} \\ W_n^+ = S_n^- + B_n^- + q_n - c_n^*, \\ \qquad S_n^+ = W_n^+ - b_n^*, B_n^+ = b_n^* \ ; \ t \in \mathcal{T} \\ 0 \leq b_n^* \leq W_n^+ \\ c_n^* \geq 0 \end{cases} , \tag{9.6}$$

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

Let

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

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

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

then the optimal strategy is to withdraw cash $c_n^* = W_n^- + q_n - (W^* e^{-r(T-t_n)} - Q_n)$ from the portfolio, and invest the remainder $\big(W^* e^{-r(T-t_n)} - Q_n\big)$ in the risk free asset. This is optimal in this case since then $E[(W^* - W_T)^2] = 0$ (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 cash flow* (Bauerle and Grether, 2015). For the sake of discussion, 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 problem (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 (that is, 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 (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

$$v(x, b, \tau) \quad = \quad \inf_{\{(p_0,c_0^w),\ldots,(p_M,c_M^w)\}} \left\{ E\left[\ (\min(W_T - W^*, 0))^2 \ \middle| \ \log S(t) = x, B(t) = b \right] \right\} . \tag{9.9}$$

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

$$\begin{aligned} v_\tau &= \mathcal{L}v + rbv_b \quad \text{where} \\ \mathcal{L}v &\equiv \frac{\sigma^2}{2} v_{xx} + (\mu - \frac{\sigma^2}{2} - \lambda\kappa) v_x - (\mu + \lambda) v + \lambda \int_{-\infty}^{\infty} v(x+y) f(y) \ dy \\ v(x, b, 0) &= (\min(e^x + b - W^*, 0))^2 \end{aligned} \tag{9.10}$$

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 equation (9.10) at τ_{n+1}^-, given the solution at τ_n^+, $\tau_n \in \mathcal{T}$ is

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

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 $\tau_n \in \mathcal{T}$,

$$\begin{aligned} v(x, b, \tau_n^+) &= \min_{(b^*, c^*)} v(x', b^*, \tau_n^-) \\ \text{subject to} & \quad \begin{cases} c^* = \max(e^x + b + q_{M-n} - Q_{M-n}, 0) \\ W' = e^x + b + q_{M-n} - c^* \\ 0 \le b^* \le W' \\ x' = \log\big(\max(W' - b^*, e^{x_{\min}})\big) \end{cases} \end{aligned} \tag{9.12}$$

where Q_ℓ is defined in equation (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 an unequally spaced grid in the B direction with N_b nodes. Set $B_{\max} = e^{x_{\max}}$ and denote the discrete solution at (x_m, b_j, τ_n^+) by

$$\begin{aligned} (v_{m,j}^n)^+ &= v(x_m, b_j, \tau_n^+) \\ (v^n)^+ &= \{(v_{m,j}^n)^+\}_{m=-N_x/2,\ldots,N_x/2-1; j=1,\ldots,N_b} \\ (v_j^n)^+ &= [(v_{-N_x/2,j}^n)^+, \ldots, (v_{N_x/2-1,j}^n)^+]. \end{aligned} \tag{9.13}$$

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 3 to advance the solution to τ_{n+1}^-. For the mean variance problem, we extend this algorithm to approximate equation (9.11), which is described in Algorithm 4.

Algorithm 4 Advance time $(v^n)^+ \to (v^{n+1})^-$.

```
Require: $(v^n)^+$ ; $\widetilde{G} = \{\widetilde{G}(\omega_m, \Delta\tau, \alpha)\}$, $m = -N_x/2, \ldots, N_x/2 - 1$ (from Algorithm 2)
for $j = 1, \ldots, N_b$ do {Advance time loop}
    $v_{m,j}^{int} = \mathcal{I}_{\Delta x, \Delta b}(x_m, b_j e^{r\Delta\tau})(v^n)^+$ ; $m = -N_x/2, \ldots, N_x/2 - 1$
    $\widetilde{V} = FFT[\ v_j^{int}\ ]$
    $(v_j^{n+1})^- = IFFT[\ \widetilde{V} \circ \widetilde{G}\ ]$    {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:

$$\begin{aligned} v(x_m, b_j, \tau_n^+) &= \min_{(b^*,c^*)} \mathcal{I}_{\Delta x,\Delta b}((x^*, b^*)(v^{n+1})^- \\ \text{subject to} &\quad \begin{cases} c^* = \max(e^{x_m} + b_j + q_{M-n} - Q_{M-n}, 0) \\ W' = e^{x_m} + b_j + q_{M-n} - c^* \\ b^* \in \{b_1, \ldots, \min(b_{\max}, W')\} \\ x^* = \log\bigl(\max(W' - b^*, e^{x_{\min}})\bigr) \end{cases} . \end{aligned} \tag{9.14}$$

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

Using the control determined from solving problem (9.9), we can determine $E[W_T]$ and $std[W_T]$ by solving an additional linear PIDE, see (Dang and Forsyth, 2014) for details.

Remark 9.1 (Practical implementation enhancements)**.** *As noted by several authors, since the Green's function and the solution is real, the Fourier coefficients satisfy symmetry relations. Hence* $\widetilde{G}(\omega_k, \Delta\tau, \alpha)$ *and* $\widetilde{V}$ *need to be computed and stored only for* $\omega_k \geq 0$. *It is also possible to arrange the semi-Lagrangian step and the optimal control step 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 multiply (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 9.1. The data 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 *1926:1- 2015:12*.[3] 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 90-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 CPI 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 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 9.2, we show the value function $v(0, 0,T)$ and the mean $E[W_T]$ and standard deviation $std[W_T]$ 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 $E[W_T], std[W_T]$, which are based on the controls computed as a byproduct of computing the value function, should show a lower order convergence.

[3] More specifically, results presented here 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 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.

Expiry time T	30 years
Initial wealth	0
Rebalancing frequency	yearly
Cash injection $\{q_i\}_{i=0,\ldots,29}$	10
Real interest rate r	.00827
Volatility σ	.14777
μ	.08885
λ	.3222
η_1	4.4273
η_2	5.262
p_u	0.2758
$x_{\max}$	$\log(100) + 5$
$x_{\min}$	$\log(100) - 10$
ϵ_1, ϵ_2	10^{-6}
Asymptotic form $E[(W_T - W^*)^2], x \to \infty$	$A(x) = 0$
Asymptotic form $E[W_T], x \to \infty$	$A(x) = W^*$

Table 9.1: *Multiperiod mean variance example. Parameters determined by fitting to the real (inflation adjusted) CRSP data for the period* 1926:1-2015:12. *Interest rate is the average real return on 90 day T-bills.*

N_x	N_b	Value function	Ratio	$E[W_T]$	Ratio	$std[W_T]$	Ratio
512	305	97148.899100	N/A	824.02599269	N/A	240.73884508	N/A
1024	609	97042.740997	N/A	824.07104985	N/A	240.55534019	N/A
2048	1217	97014.471301	3.8	824.09034690	2.3	240.51245396	4.3
4096	2433	97007.286530	3.9	824.08961667	-26	240.49691620	2.7
8192	4865	97005.451814	3.9	824.09295889	-.22	240.49585213	14.6

Table 9.2: *Test of convergence of optimal multiperiod mean variance investment strategy. Monotone method, linear basis functions. Parameters in Table 9.1. Fixed $W^* = 1022$. Ratio is the ratio of successive changes.*

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 Carlo simulations to directly compute the mean and standard deviation of the final wealth. The results are shown in Table 9.3.

Of more practical interest is the following computation. In Table 9.4 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 .60 in stocks and .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 solve for the value function and the expected value equation. The results of this test are shown in Table 9.5. In this case, fixing the mean and allowing W^* to vary, results in smooth convergence of the standard deviation. From a practical point of view, we can see that the optimal strategy has

N_{sim}	$E[W_T]$	$std[W_T]$
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

TABLE 9.3: *Monte Carlo simulation results, based on optimal controls from solving for the value function using the monotone Fourier technique. Numbers in brackets are the standard error, 99% confidence level, for the mean. Compare with Table 9.2. Parameters in Table 9.1. Fixed* $W^* = 1022$.

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

TABLE 9.4: *Portfolio rebalanced to* .60 *in stocks and* .40 *in bonds at each monitoring date. Closed form expression for mean and standard deviation. Median computed using Monte Carlo simulation. Parameters in Table 9.1.*

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 to a median of 704 for the constant proportion strategy. A heat map of the optimal strategy is shown in Figure 9.1.

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. It then happens that a numerical procedure must be used to determine the optimal control at discrete 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 get at most second order accuracy.

Traditional FST/CONV methods assume knowledge 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 Green's function is known, then we carry out a pre-processing step by projecting

N_x	N_b	$E[W_T]$	$std[W_T]$	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

TABLE 9.5: *At each refinement level* W^* *is determined so that* $E[W_T] = 824.10047$. *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 in Table 9.1.*

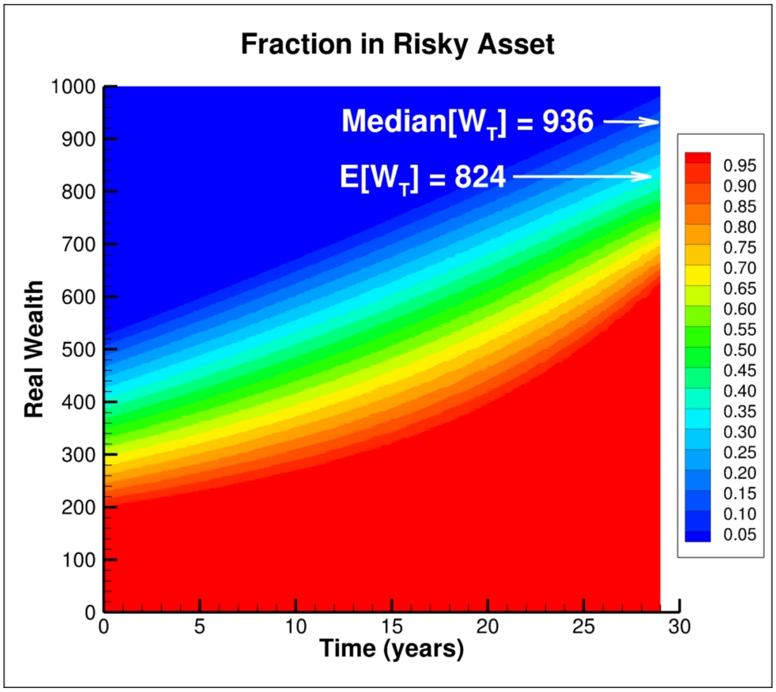


FIGURE 9.1: *Optimal strategy, fraction of portfolio invested in stock, as a function of current total real wealth* $W_t = S_t + B_t$ *and forward time* t. *Parameters in Table 9.1.*

the Green's function onto a set of linear basis functions. These integrals can then be computed to within a specified tolerance and this allows us to guarantee a monotone scheme to within the tolerance. This monotone scheme is robust to small timesteps, which is observably not the case for the standard FST/CONV methods, and indeed is a major pitfall of the latter methods.

When the Green's function depends on time only through the timestep size and the monitoring dates for the control are equally spaced (which is typically the case), then 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.

Appendices

A Green's functions

Consider the PIDE

$$v_\tau \;=\; \frac{\sigma^2}{2} v_{xx} + (\mu - \frac{\sigma^2}{2} - \lambda\kappa) v_x - (\rho + \lambda) v + \lambda \int_{-\infty}^{+\infty} v(x+y) f(y)\, dy \ . \tag{A.1}$$

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 the right hand side of equation (A.1) is $\mathcal{L}v$ in the mean variance case.

Let

$$
\begin{aligned}
v(\omega,\tau) &= \int_{-\infty}^{\infty} V(\omega,\tau)e^{2\pi i\omega x}\ d\omega \\
f(\omega,\tau) &= \int_{-\infty}^{\infty} F(\omega,\tau)e^{2\pi i\omega x}\ d\omega\ . \qquad \text{(A.2)}
\end{aligned}
$$

Substituting equation (A.2) into equation (A.1) gives

$$
\begin{aligned}
V(\omega,\tau)_\tau &= \Psi(\omega)V(\omega,\tau) \quad \text{where} \qquad \text{(A.3)} \\
\Psi(\omega) &= \left(-\frac{\sigma^2}{2}(2\pi\omega)^2 + (\mu - \lambda\kappa - \frac{\sigma^2}{2})(2\pi i\omega) - (\rho+\lambda) + \lambda\overline{F}(\omega)\right)\ ,
\end{aligned}
$$

with $\overline{F}(\omega)$ being the complex conjugate of $F(\omega)$. Integrating equation (A.3) gives

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

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

$$
G(\omega,\Delta\tau) \;=\; e^{\Psi(\omega)\Delta\tau}\ . \qquad \text{(A.4)}
$$

In the case of a double exponential jump distribution (8.2), then

$$
\overline{F}(\omega) \;=\; \frac{p_u}{1-2\pi i\omega/\eta_1} + \frac{1-p_u}{1+2\pi i\omega/\eta_2}
$$

while in the case of a log-normal jump size distribution (8.5)

$$
\overline{F}(\omega) \;=\; e^{2(\pi i\omega\nu - (\pi\omega\gamma)^2)}\ .
$$

From equation (A.3) and (A.4) we have

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

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

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

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

Since the Green's function for equation (A.1) 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 Appendix. Since the exact Green's function is non-negative, the projected Green's function (4.6) then converges to a non-negative value at every point y_j. Consider the case of the truncated projection on linear basis functions

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

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

$$\begin{aligned}
|\widetilde{g}(y_j,\Delta\tau,\alpha) - \widetilde{g}(y_j,\Delta\tau,\infty)| &= \Bigg| \frac{1}{P}\sum_{k=\alpha N/2}^{\infty} e^{2\pi i \omega_k y_j}\left(\frac{\sin^2 \pi\omega_k\Delta x}{(\pi\omega_k\Delta x)^2}\right) G(\omega_k,\Delta\tau) \\
&\quad + \frac{1}{P}\sum_{k=-\infty}^{-\alpha N/2-1} e^{2\pi i \omega_k y_j}\left(\frac{\sin^2 \pi\omega_k\Delta x}{(\pi\omega_k\Delta x)^2}\right) G(\omega_k,\Delta\tau)\Bigg| \\
&\le \frac{2}{P}\sum_{k=\alpha N/2}^{\infty} \frac{1}{(\pi\omega_k\Delta x)^2}|G(\omega_k,\Delta\tau)| \\
&\le \frac{2}{P}\cdot\frac{4}{\pi^2\alpha^2}\sum_{k=\alpha N/2}^{\infty} |G(\omega_k,\Delta\tau)| \ . \qquad \text{(B.1)}
\end{aligned}$$

Noting that $Re(\overline{F}(\omega)) \le 1$, we then have

$$\begin{aligned}
Re(\Psi(\omega)) &= -\frac{\sigma^2(2\pi\omega)^2}{2} - (\rho+\lambda) + \lambda Re(\overline{F}(\omega)) \\
&\le -\frac{\sigma^2(2\pi\omega)^2}{2} - (\rho+\lambda) + \lambda \le -\frac{\sigma^2(2\pi\omega)^2}{2}
\end{aligned}$$

since $\rho \ge 0$. Hence

$$|G(\omega,\Delta\tau)| = |e^{\Psi(\omega)\Delta\tau}| \le e^{-\frac{\sigma^2(2\pi\omega)^2\Delta\tau}{2}}. \qquad \text{(B.2)}$$

If we let $C_4 = \frac{2\sigma^2\pi^2\Delta\tau}{P^2}$ then equations (B.2) and (B.1) implies

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

Bounding the sum gives

$$|\widetilde{g}(y_j,\Delta\tau,\alpha) - \widetilde{g}(y_j,\Delta\tau,\infty)| \le \frac{8}{P\pi^2\alpha^2}\cdot\frac{e^{-C_4N^2\alpha^2/4}}{1-e^{-C_4N\alpha}} \ . \qquad \text{(B.3)}$$

Consider the monotonicity test in Algorithm 2, line 5, given by

$$test_1 = \sum_j \Delta x \min(\widetilde{g}(y_j,\Delta\tau,\alpha),0).$$

Noting that $\widetilde{g}(y_j,\Delta\tau,\infty) \ge 0$, and from equation (B.3) and $\sum_j \Delta x = P$, we have

$$|test_1| \le \frac{8}{\pi^2\alpha^2}\cdot\frac{e^{-C_4N^2\alpha^2/4}}{1-e^{-C_4N\alpha}} \ , \qquad \text{(B.4)}$$

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

Consider now the accuracy test on line 6 of Algorithm 2, given by

$$test_2 = \max_j \Delta x|\widetilde{g}(y_j,\Delta\tau,\alpha) - \widetilde{g}(y_j,\Delta\tau,\alpha/2)|$$

which we see is bounded by

$$\begin{aligned} |test_2| &\leq \Delta x \max_j \Big(|\widetilde{g}(y_j, \Delta\tau, \alpha) - \widetilde{g}(y_j, \Delta\tau, \infty)| + |\widetilde{g}(y_j, \Delta\tau, \alpha/2) - \widetilde{g}(y_j, \Delta\tau, \infty)| \Big) \\ &\leq \Delta x \max_j \Big(2|\widetilde{g}(y_j, \Delta\tau, \alpha/2) - \widetilde{g}(y_j, \Delta\tau, \infty)| \Big) \\ &\leq \frac{64}{\pi^2 \alpha^2} \cdot \frac{\Delta x}{P} \cdot \frac{e^{-C_4 N^2 \alpha^2/16}}{1 - e^{-C_4 N \alpha/2}} \ . \end{aligned}$$

This test will also be satisfied for small values of α, although this will require larger values of α than for the monotonicity test (B.4).

References

Alonso-Garcia, J., O. Wood, and J. Ziveyi (2017). Pricing and hedging guaranteed minimum withdrawal benefits under a general Levy framework using the COS method. Working paper, ARC center of excellence in population ageing research.

Angelini, F. and S. Herzel (2014). Delta hedging in discrete time under stochastic interest rate. *Journal of Computational and Applied Mathematics 259*, 385–393.

Azimzadeh, P., E. Bayraktar, and G. Labahn (2017). Convergence of approximation schemes for weakly nonlocal second order equations. Working paper, University of Waterloo, arXiv-1705-02922.

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

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

Bauerle, N. and S. Grether (2015). Complete markets do not allow free cash flow streams. *Mathematical Methods of Operations Research 81*, 137–146.

Bokanowski, O., A. Picarelli, and C. Reisinger (2018). High order filtered schemes for time dependent 2nd order HJB equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 27 pages. To appear.

Chen, Z. and P. A. Forsyth (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.

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

Cong, F. and C. Oosterlee (2016). Multi-period mean variance portfolio optimization based on Monte-Carlo simulation. *Journal of Economic Dynamics and Control 64*, 23–38.

Cui, X., J. Gao, X. Li, and D. Li (2014). Optimal multi-period mean variance policy under no-shorting constraint. *European Journal of Operational Research 234*, 459–468.

Cui, X., D. Li, S. Wang, and S. Zhu (2012). Better than dynamic mean-variance: Time inconsistency and free cash flow stream. *Mathematical Finance 22*, 346–378.

Dai, M., Y. Kwok, and J. Zong (2008). Guaranteed minimum withdrawal benefit in variable annuities. *Mathematical Finance 184*, 595–611.

Dang, D.-M. and P. Forsyth (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.

Dang, D. M. and P. A. Forsyth (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.

Deng, G., T. Dulaney, C. McCann, and M. Yan (2017). Efficient valuation of equity-indexed annuities under Lévy processes using Fourier cosine series. *Journal of Computational Finance 21*, 1–27.

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

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

Garroni, M. G. and J. L. Menaldi (1992). *Green functions for second order parabolic integro-differential problems*. New York: Longman Scientific.

Huang, H.-C. (2010). Optimal multiperiod asset allocation: matching assets to liabilities in a discrete model. *The Journal of Risk and Insurance 77*, 451–472.

Huang, Y., P. Zeng, and Y. Kwok (2018). Optimal initiation of a guaranteed lifelong withdrawal with dynamic controls. *SIAM Journal on Financial Mathematics*. to appear.

Ignatieva, K., A. Song, and J. Ziveyi (2016). Fourier space time-stepping algorithm for valuing guaranteed minimum withdrawal benefits in variable annuities under regime-switching and stochastic mortality. Working paper, University of New South Wales.

Jackson, K., S. Jaimungal, and V. Surkov (2008). Fourier space time-stepping for option pricing with Levy models. *Journal of Computational Finance 12:2*, 1–29.

Kou, S. and H. Wang (2004). Option pricing under a double exponential jump diffusion model. *Management Science 50*, 1178–1192.

Li, D. and W.-L. Ng (2000). Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical Finance 10*, 387–406.

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

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

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.

Pooley, D., P. Forsyth, and K. Vetzal (2003). Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis 23*, 241–267.

Remillard, B. and S. Rubenthaler (2013). Optimal hedging in discrete time. *Quantitative Finance 13*, 819–825.

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

Vigna, E. (2014). On efficiency of mean-variance based portfolio selection in defined contribution pension schemes. *Quantitative Finance 14*, 237–258.

Zheng, W. and Y. Kwok (2014). Fourier transform algorithms for pricing and hedging discretely sampled exotic variance products and volatility derivatives under additive processes. *Journal of Computational Finance 18*, 3–30.

Zhou, X. Y. and D. Li (2000). Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics and Optimization 42*, 19–33.