Mixing Monte Carlo and Partial Differential Equation Methods For Multi-Dimensional Optimal Stopping Problems Under Stochastic Volatility

by

David Farahany

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Department of Statistical Sciences
University of Toronto

Abstract

Mixing Monte Carlo and Partial Differential Equation Methods For Multi-Dimensional Optimal Stopping Problems Under Stochastic Volatility

David Farahany
Doctor of Philosophy
Department of Statistical Sciences
University of Toronto
2019

In this thesis, we develop a numerical approach for solving multi-dimensional optimal stopping problems (OSPs) under stochastic volatility (SV) that combines least squares Monte Carlo (LSMC) with partial differential equation (PDE) techniques. The algorithm provides dimensional reduction from the PDE and regression perspective along with variance and dimensional reduction from the MC perspective.

In Chapter 2, we begin by laying the mathematical foundation of mixed MC-PDE techniques for OSPs. Next, we show the basic mechanics of the algorithm and, under certain mild assumptions, prove it converges almost surely. We apply the algorithm to the one dimensional Heston model and demonstrate that the hybrid algorithm outperforms traditional LSMC techniques in terms of both estimating prices and optimal exercise boundaries (OEBs).

In Chapter 3, we describe methods for reducing the complexity and run time of the algorithm along with techniques for computing sensitivities. To reduce the complexity, we apply two methods: clustering via sufficient statistics and multi-level Monte Carlo (mlMC)/multi-grids. While the clustering method allows us to reduce computational run times by a third for high dimensional problems, mlMC provides an order of magnitude reduction in complexity. To compute sensitivities, we employ a grid based method for derivatives with respect to the asset, S, and an MC method that uses initial dispersions for sensitivities with respect to variance, v. To test our approximations and computation of sensitivities, we revisit the one dimensional Heston model and find our approximations introduce little-to-no error and that our computation of sensitivities is highly accurate in comparison to standard LSMC. To demonstrate the utility of our new computational techniques, we apply the hybrid algorithm to the multi-dimensional Heston model and show that the algorithm is highly accurate in terms of estimating prices, OEBs, and sensitivities, especially in comparison to standard LSMC.

In Chapter 4, we highlight the importance of multi-factor SV models and apply our hybrid

algorithm to two specific examples: the Double Heston model and a mean-reverting commodity model with jumps. Again, we were able to obtain low variance estimates of the prices, OEBs, and sensitivities.

Acknowledgements

I would like to begin by thanking my supervisors, Professors Sebastian Jaimungal and Kenneth Jackson for all of their guidance, support, and seemingly infinite amount of patience. I am very grateful to them for the opportunity to work on the fun and interesting topic of least squares Monte-Carlo methods. I would also like to thank my external examiner, Professor Tony Ware, for his thoughtful comments and careful read through of this thesis. Furthermore, I would like to thank the members of my thesis and examining committee, Professors Luis Seco, Sheldon Lin, Chi-Guhn Lee, and Leonard Wong for their suggestions and comments.

Next, I would like to thank a number of teachers and mentors that I have had during my time as a university student: Professors Binder, Karshon, Hare, Spronk, and Reid. I am deeply indebted to you for all of the time you spent on me during office hours and the opportunities to work on interesting research projects. I would also like to thank the graduate administrators in the Department of Statistical Sciences: Christine Bulguryemez, Andrea Carter and Annette Courtemanche for all of their assistance over the years.

To my mom, dad, and brother, thank you for all of the love and support throughout my life, especially over the last few years; this thesis would not have been possible without you.

Finally, I want to express appreciation to my friends and colleagues in the department, especially my fellow Math. Fin. Gladiators: Tianyi Jia, Luke Gan, Bill Huang, Philippe Casgrain, Michael Chiu, Ali Al-Aradi, and Arvind Shrivats - thank you for all the laughs and interesting discussions.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Optimal stopping problems (OSPs) are an important class of problems that arise in probability and statistics, especially in the field of mathematical finance. In statistics they arise in Bayesian quickest detection problems where one detects the presence of a drift in a Weiner process or the jump intensity of a Poisson process [51]. In mathematical finance, their applications include problems pertaining to portfolio allocation, production-consumption, irreversible investment, valuation of natural resources and American style options [52].

While the theory of such problems has been extensively analyzed within the continuous time stochastic control literature [51], [52], [60], and in some cases solved in closed form [43], [37], [51], their numerical solution still remains an active area of research. Following the classification scheme of [57], we refer to problems of dimension 1 as *low* dimensional, 3 - 5 as *medium* dimensional and problems greater than 5 as *high* dimensional. Low and most medium dimensional problems are often solved using accurate grid-based deterministic methods. More complex medium and high dimensional problems must be solved using simulation techniques developed in [47], [62] and improving their accuracy is a topic of active research. In this thesis, we develop a numerical method to effectively handle discrete time OSPs under stochastic volatility which are classified as medium to high dimensional. Our approach is a hybrid between gridded methods and simulation approaches making it ideal for the types of problems we consider.

In the sections that follow we outline the basic theory of discrete optimal stopping and discuss its connections to continuous time problems. Afterwards, we discuss the history and literature on their numerical solutions ranging from deterministic to probabilistic approaches followed by the relatively new field of hybrid approaches. Finally, we outline the main contributions of this thesis, setting the stage for developing the new hybrid algorithm.

1.1 Optimal Stopping Problems

1.1.1 Theoretical Results

We begin by discussing various classical results that theoretically characterize the solution of fully observed, Markovian, discrete time OSPs and provide the foundation for numerical approaches. Our presentation follows the manuscript of [51] which, for the interested reader, also contains

detailed proofs of the results that we discuss.

Classical Results on Optimal Stopping

Let $\{t_0 = 0, t_1, \ldots, t_{N-1}, t_N = T\}$ be an ordered set of times and $(\Omega, \mathcal{F}, (\mathcal{F}_n)_{n=1}^N, \mathbb{Q})$ a filtered probability space. Let X_n be a stochastic process on $\mathbb{R}^d$ that is adapted to $\mathcal{F}_n$, $h : \mathbb{R}^d \to \mathbb{R}$ a Borel measurable function and $Z_n = h_n(X_n)$, where Z_n is adapted to $\mathcal{F}_n$. We refer to Z_n as our reward process, h_n as our payoff function, and X_n as our underlying driving process. Finally, let $\mathcal{M}_n^N$ be the set of stopping times adapted to $\mathcal{F}_n$ that take value in $\{t_n, \ldots, t_N\}$.

We suppose that at each time, t_n, an agent has a one-time claim to the reward process Z_n and must eventually claim the reward by time t_N. We define an optimal stopping problem to be the problem of characterizing and computing V_n as defined by

$$V_n = \sup_{\tau \in \mathcal{M}_n^N} \mathbb{E}\left[Z_\tau\right]. \tag{1.1}$$

To motivate the main result regarding such problems, we describe a process S_n^N which provides a stochastic characterization of the agent's choice at each time t_n. At time N, the agent's claim on Z_n expires and so she must immediately exercise to obtain $S_N^N = Z_N$. At time t_{N-1}, the agent may choose to claim the reward and obtain Z_{N-1} or may hold on and continue until time t_N, in which case the value of her position is $\mathbb{E}[Z_N \mid \mathcal{F}_{N-1}]$. As a result, we set $S_{N-1}^N = \max\left(Z_{N-1}, \mathbb{E}[Z_N \mid \mathcal{F}_{N-1}]\right)$. This argument may be extended backwards for all preceding times which leads to the following description of S_n^N

$$\begin{aligned} S_N^N &= Z_N \\ S_n^N &= \max\left(Z_n, \mathbb{E}\left[Z_{n+1} \mid \mathcal{F}_n\right]\right) \ \forall\ n = N-1, \ldots, 0. \end{aligned}$$

This formulation of the agent's optimal choice also provides a potential candidate for the optimal stopping time in (1.1)

$$\tau_n^N = \min\left\{k \ \mid \ S_k^N = Z_k,\ k = n, \ldots, N\right\}. \tag{1.2}$$

The following theorem shows that the agent's process, S_n^N, is indeed the optimal strategy and that the supremum in (1.1) is achieved at τ_n^N. By taking expectations, Theorem 1 also shows us that $V_n = \mathbb{E}[Z_{\tau_n^N}]$.

Theorem 1 ([51],Theorem 1.2)**.** *Let $0 \le n \le N$ and suppose $\mathbb{E}\left(\sup_{n \le k \le N} |Z_k|\right) < \infty$. Then*

$$\begin{aligned} S_n^N &\ge \mathbb{E}[Z_\tau \mid \mathcal{F}_n] \ \forall\ \tau \in \mathcal{M}_n^N \\ S_n^N &= \mathbb{E}[Z_{\tau_n^N} \mid \mathcal{F}_n] \end{aligned}$$

where τ_n^N is as defined in (1.2). Furthermore:

1. *The stopping time τ_n^N is optimal.*
2. *If τ_* is an optimal stopping time, then $\tau_n^N \le \tau_*$, $\mathbb{Q}$ a.s.*

The last point shows that while we do not guarantee uniqueness of the optimal stopping time, τ_n^N is a.s. the earliest optimal stopping time.

It is worth noting an alternative way to characterize S_n^N is by means of the Snell-envelope of Z_n:

$$S_n^N = \text{ess sup}_{\tau \in \mathcal{M}_n^N} \mathbb{E}[Z_\tau \mid \mathcal{F}_n]$$

This approach is useful for extending the above results to infinite horizon and continuous time problems.

We now turn to a characterization of optimal stopping problems when the underlying process X_n is a time inhomogeneous Markov Chain. We modify our probability space to be of the form $(\Omega, \mathcal{F}, (\mathcal{F}_n), \mathbb{Q}_{n,x})$ such that $X_n = x$ a.s. and the mapping $x \mapsto \mathbb{Q}_{n,x}(F)$ is measurable on $\mathbb{R}^d$ for all $F \in \mathcal{F}$. It may then be shown that the mapping $x \mapsto \mathbb{E}_x^{\mathbb{Q}}[Y]$ is also measurable for any random variable Y on our space.

In this case, we define our optimal stopping problem to be the characterization and computation of

$$V^N(n,x) = \sup_{\tau \in \mathcal{M}_0^{N-n}} \mathbb{E}_{n,x}^{\mathbb{Q}}[h(n+\tau, X_{n+\tau})] \tag{1.3}$$

and yields a solution that is analogous to Theorem 1. As before, we set

$$\begin{aligned} S_N^N &= h(X_N) \\ S_{n+k}^N &= \max\left(h(X_{n+k}), \mathbb{E}_{n,x}^{\mathbb{Q}}\left[V^N(X_{n+k+1}) \mid \mathcal{F}_{n+k}\right]\right) \\ &\forall\ k = 0, \ldots, N-n-1 \end{aligned}$$

and

$$\tau_{n+k}^N = \min\{\ l \ \mid\ n+k \le l \le N, S_l^N = h(l, X_l)\}.$$

With these definitions, one can then show $S_{n+k}^N = V^N(n+k, X_{n+k})$ and motivates us to introduce the continuation and stopping sets C and D

$$\begin{aligned} C &= \{(n,x) \in \{0,\ldots,N\} \times \mathbb{R}^d \ \mid\ V(n,x) > h(n,x)\} \\ D &= \{(n,x) \in \{0,\ldots,N\} \times \mathbb{R}^d \ \mid\ V(n,x) = h(n,x)\} \end{aligned}$$

along with the first entry time τ_n^N into D via

$$\tau_{n,x}^N = \min\{\ k \ \mid\ n \le k \le N, (k, X_k) \in D\}.$$

Theorem 2. *Let $E_x\left[\sup_{0\le k\le N} |h(k, X_k)|\right] < \infty$ for all $x \in \mathbb{R}^d$. Then the value function V^N satisfies the Wald-Bellman equations*

$$V^N(n,x) = \max(h(n,x), TV^N(n,x)) \tag{1.4}$$

for $n = N-1, \ldots, 1, 0$ where $TV^N(n,x) = \mathbb{E}_{n,x}^{\mathbb{Q}}[h(n+1, X_{n+1})]$ and $x \in \mathbb{R}^d$. Moreover, we have

- *The stopping time $\tau_{n,x}^N$ is optimal in (1.3)*

- *If τ_* is an optimal stopping time in (1.3) then $\tau_D \leq \tau_*$ P_x-a.s. for every $x \in \mathbb{R}^d$.*

As we shall see, the Wald-Bellman equations (1.4) plays a fundamental role in the numerical solution of the optimal stopping problems we consider.

Connections to Continuous Time Problems

While we focus on discrete time problems, finite horizon continuous time problems are an important class of optimal stopping problems. Analogous to the discrete time problem, we seek to compute the value of

$$V_t^T = \sup_{\tau \in \mathcal{M}_t^T} \mathbb{E}[Z_\tau]. \tag{1.5}$$

In a continuous time framework, previous arguments for deriving the optimal strategy do not carry over and we instead define S_t using the Snell envelope for Z

$$S_t = \text{ess sup}_{\tau \geq t} \mathbb{E}\left[Z_\tau \mid \mathcal{F}_t\right]$$

along with

$$\tau_t = \inf\left\{s \geq t \mid S_s = Z_s\right\}.$$

From here, results that are similar in nature to Theorems 1 and 2 may be derived which show that under certain circumstances, τ_t is the optimal stopping for the problem (1.5).

What interests us the most is that under fairly mild assumptions, these problems may be approximated by discrete time problems where the number of exercise dates tends to infinity. Following [5] we let

$$\pi = \{0 = t_0, t_1, \ldots, t_{N-1}, t_N = T\}$$

and

$$S_t^\pi = \text{ess sup}_{\tau \in \mathcal{T}_{[t,T]}^\pi} \mathbb{E}\left[Z_\tau \mid \mathcal{F}_t\right]$$

which we think of as a π-discrete time approximation to S_t.

Omiting certain mild regularity assumptions on Z, the following result justifies our approach of using discrete approximations to numerically solve continuous time problems and shows that our approximating process converges uniformly in L^2.

Theorem 3. *Given S_t and S_t^π as before, we have*

$$\max_{i \leq N} \mathbb{E}\left[\left|S_{t_i} - S_{t_i}^\pi\right|^2\right]^{\frac{1}{2}} \leq O(|\pi|^{\frac{1}{4}})$$

where $|\pi| = \max_{i<N} |t_{i+1} - t_i|$.

Beyond these facts, the numerical analysis of continuous time processes lays outside the scope of this thesis and we continue with our discrete time framework.

1.1.2 Solutions via Numerical Approaches

As seen in Section 1.1.1, solving an OSP may be reduced to solving the Wald-Bellman equations shown in (1.4); this in turn reduces to developing an algorithm for computing conditional expectations of the form

$$\mathbb{E}^{\mathbb{Q}}_{t_n,x}[h(n+1, X_{n+1})] := \mathbb{E}^{\mathbb{Q}}[h(n+1, X_{n+1}) \mid X_{t_n} = x]. \tag{1.6}$$

Developing numerical methods for accurately computing (1.6) has been an active area of research over the past several decades, particularly in the context of mathematical finance. In such situations, the driving process $(X_{t_n})_{n=0}^N$ arises from sampling an SDE of a process $(X_t)_{t\geq 0}$. Methods for computing (1.6) can be categorized as deterministic, probabilistic, and as of late, hybrid.

Determinstic Numerical Approaches

Most deterministic approaches revolve around using the Markov property for X and Itô's Lemma to generate a PDE which the conditional expectation must solve:

$$\begin{cases} \dfrac{\partial g}{\partial t}(t,x) + \mathcal{L}g(t,x) = 0\,, \\ \qquad\qquad g(t_{n+1}, x) = V(t_{n+1}, x) \end{cases} \tag{1.7}$$

over $[t_n, t_{n+1}]$ with some additional spatial boundary conditions. Here

$$g(t_n, x) = \mathbb{E}_{t_n,x}[V(t_{n+1}, X_{t_{n+1}})]$$

and $\mathcal{L}$ is the infinitesimal generator of X_t.

This PDE may then be solved using a variety of fully numerical methods. If X_t is, for instance, an Ito process then (1.7) defines a parabolic PDE which may be solved numerically using a variety of state-of-the-art methods as outlined in [17].

The main draw back of this approach is the well known *Curse of Dimensionality*: the computational costs of such methods grow exponentially with d, and effectively limit these approaches to problems where $d \leq 3$. Examples and discussions of high dimensional implementations may be found in [14], [54].

While fully numerical methods are appealing within their dimensionality constraints they often miss special properties of the model that allow for more efficient computations. For instance if the density, or a special transform of the density, of X_t is known analytically, one may apply quadrature or various numerical transform techniques to compute (1.6). These more efficient approaches may be found in [59], [48], [21], [57], and others.

When applicable, deterministic numerical methods tend to be the best approach for solving OSPs. By obtaining non-probabilistic estimates of $\mathbb{E}_{t_n,x}\left[V(t_{n+1}, X_{t_{n+1}})\right]$, one may estimate the holding and exercise regions with excellent accuracy; an issue which plagues probabilistic approaches. Also, the solutions they produce are global in that one obtains $\mathbb{E}_{t_n,x}\left[h(t_{n+1}, X_{t_{n+1}})\right]$ for an entire grid of values for x as opposed to a relatively small region of points.

Probabilistic Numerical Approaches

In recent decades, several probabilistic approaches have emerged to deal with multi-dimensional OSPs. These methods include random trees, state based partitioning, stochastic mesh methods and least-squares/regression MC (LSMC). While our main focus in this section will be on LSMC, the reader should see [27] for a detailed overview of other methods. Also, as the literature on LSMC is vast, this introduction/survey is meant to be non-exhaustive.

LSMC is a powerful family of methods for solving OSPs that was first introduced by [9] and further popularized by [47] (LS), [62] (TV). These methods are popular due to their simplicity and applicability to a wide range of models, especially of high dimension. The classical approach is based on the following, given $\mathbb{E}\left[V_{n+1} \mid X_n = x\right]$ we attempt to write

$$\mathbb{E}\left[V_{n+1}(X_{n+1}) \mid X_n = x\right] \approx \sum_{l=1}^{L} a_{n,l}\ \phi_l(x) \tag{1.8}$$

for some coefficient vector $a_n \in \mathbb{R}^L$ where $\{\phi_l\}_{l=1}^L$ is an independent set of functions on $\mathbb{R}^d$. Assuming (1.8) holds, we project $\mathbb{E}\left[V_{n+1}(X_{n+1}) \mid X_n = x\right]$ onto $a_n \cdot \phi(x)$ which results in the following identities for a_n

$$\begin{aligned} a_n &= A_n^{-1}\mathbb{E}\left[\phi(X_n)V_{n+1}(X_{n+1})\right] \text{ where} \\ A_n &= \mathbb{E}\left[\phi(X_n)\phi(X_n)^T\right] \end{aligned}$$

and refer to a_n as the idealized coefficient. As the above expectations are usually not computable in closed form, we estimate a_n via MC and regression. LSMC then typically takes the following approach

1. Simulate N independent paths of X_t on $\{0 = t_0, \ldots, t_M = T\}$ denoted as X_m^j and set $V_M^j = h(X_M^j)$.
2. Set $A_{M-1}^N = \frac{1}{N}\sum_{j=1}^N \phi(X_{M-1}^j)\phi(X_{M-1}^j)^T$ and $a_n^N = [A_{M-1}^N]^{-1}\frac{1}{N}\sum_{j=1}^N \phi(X_{M-1}^j)V_M^j$
3. Set $C_{M-1}^N(X) = a_n^N \cdot \phi(X)$
4. At this stage the TV and LS approaches diverge slightly and we take one of two actions

TV: Set $V_{M-1}^j = \max(h(X_{M-1}^j), C_{M-1}^N(X_{M-1}^j)$

LS: Set

$$V_{M-1}^j = \begin{cases} h(X_{M-1}^j) & \text{if } h(X_{M-1}^j) \geq C_{M-1}^N(X_{M-1}^j) \\ V_M^j & \text{if } h(X_{M-1}^j) < C_{M-1}^N(X_{M-1}^j) \end{cases} \tag{1.9}$$

5. We then repeat the above steps, moving backwards in time, until we obtain $\{V_1^j\}_{j=1}^N$, set

$$V_0^{d,N} = \frac{1}{N}\sum_{j=1}^N V_1^j$$

and refer to V_0^N as the direct estimator.

This initial computation provides us with several estimates: the value $V_0^{d,N}$, continuation and exercise regions $C^N, D^N \subset \{t_1, ..., t_{M-1}\} \times \mathbb{R}^d$. The regions C^N and D^N are estimates of the true continuation and exercise regions and provide an approximation of the optimal stopping time τ_n

$$\tau_n^N = \min\{\ k \mid n \leq k \leq M, X_k \in D_k\}$$

which leads to the expression

$$V_0^{l,N} = \mathbb{E}\left[h(\tau_0^N, X_{\tau_0^N})\right]$$

that may be computed by a usual MC simulation. As τ_0^N is by definition sub-optimal, $V_0^{l,N}$ is biased low, hence we use the superscript l. The estimator $V_0^{d,N}$ for the TV approach tends to be biased high due to the convexity of the max function, however, demonstrating this mathematically often requires extra assumptions on one's model and simulation framework [40]. It is worth noting that the direct estimator in LS tends to be lower than that of TV [27]. While $V_0^{d,N}$ produces estimates that tend to be biased high, they often greatly over estimate V_0. By means of duality estimates [32], [55], one may obtain true upper bounds on V_0 that are fairly sharp.

In the LS approach, V_0 is computed using less information than TV, specifically in that one simply needs to know the sign of the statistic

$$T_n^N := h_n - C_n^N$$

to obtain V_n^N as opposed to knowing the entire function C_n^N. Thus, one only needs to know the location of the boundary at time n and this observation is exploited in [49]. The structure of the payoff in TV is more similar to PDE approaches, and lends itself to stochastic grid interpretations such as in [6], [40]. Also, the work of [20] describes an algorithm that in some sense interpolates between the TV and LS approaches.

Regarding theoretical results, the authors of [62] show that $a_n^N \to a_n$ almost surely for the TV algorithm. In [12], the authors provide a detailed analysis of the LS approach. Under certain assumptions the authors prove $a_n^N \to a_n$ a.s. and develop a CLT for a_n^N, thus establishing a rate of convergence. Both [62] and [12] keep the number of basis functions fixed at some degree L. In [29], [58] convergence is analyzed when the number of basis functions is taken into account. The work of [19] applies the language and methodology of statistical learning theory to analyze the convergence of LSMC for general learning procedures, beyond least-squares regression. In their work they provide an analysis that extends the results of [12].

Since these seminal works, numerous shortcomings have been identified which has led to a number of improvements and extensions of the original algorithms. The traditional approaches to LSMC have now been classified as *Regress-Now* algorithms in contrast to newer approaches which *Regress-Later*. Examples of regress-later algorithms may be found in [6], [40] and are further discussed in [28]. These algorithms have closed-form features built in which allow for variance reduction and tend to perform better than regress-now algorithms. Also, notable approaches for computing sensitivities may be found in [40], [64], [7].

A notable shortcoming of LSMC is referred to as the *Basis Selection Problem*: the LSMC algorithm is sensitive to the choice of basis functions, and a set of basis functions that work for one problem may not work for another [1], [5]. Thus, there is no general consensus on the choice of basis functions, and one must find a family that suits their own problem best. If the quality of the regression step is poor at any given step n, one is not able to estimate $C_n^N(X)$ accurately, and cannot determine the exercise region D_n which undermines the entire computation for later time stages and yields nonsensical results. One of the issues arises from trying to globally fit a polynomial to the data and is addressed by local regression [5], [40].

Interestingly, LSMC tends to perform relatively poorly for SV models. One needs to simulate millions of paths in order to obtain a good estimate of V_0 and the optimal exercise boundaries, even for a $2d$ problem [34], [49]. To improve the method in these circumstances, certain model specific approaches have been proposed such as in [34], [1], [24]. On a side note, the work of [53] deals with the situation of non-Markovian OSPs where the volatility process is unobservable. While this appears to be the most realistic formulation, we note that a practitioner may always calibrate their model to various sources of real-world data to measure the current level of volatility, at least up to some level of confidence. As mentioned before, the purpose of this thesis is to develop methods to improve the convergence of LSMC for fully-observed SV problems, especially in high dimensions.

Beyond OSPs, LSMC has seen applications in areas such as computing exposure profiles in counter-party credit risk [24], the numerical solutions of backwards SDEs [4] and optimal switching problems [8].

1.2 Mixed Monte Carlo-Partial Differential Equations Methods

When numerically computing expectations of the form (1.6)

$$g(x) = \mathbb{E}\left[h(X_T) \mid X_0 = x\right]$$

there are two notable numerical approaches: deterministic methods as in Section 1.1.1 and direct MC. In the case where d is large, only MC methods are applicable, however, they converge at a rate $\frac{\sigma_X}{\sqrt{N}}$, where the constant σ_X increases considerably with d.

In [35], [44] a hybrid approach, which combines probabilistic and deterministic methods, was developed for models that possess a property known as one-way coupling. In a financial context, suppose $X_t = (S_t, v_t)$ on $\mathbb{R}^2$ where S_t represents an asset price, v_t is its variance where v_t may be simulated independently of S_t. This latter property is known as *one way coupling* and will be formally described in Section 2.2. We then compute $g(S, v)$ as follows

1. Simulate N independent paths of v_t on $[0, T]$ stemming from v_0.

2. Compute
$$\mathbb{E}[\ h(S_T) \mid X_0 = (S_0, v_0), [v^j]_0^T\ ] := \mathbb{E}[\ h(S_T) \mid X_0 = (S_0, v_0), \mathcal{F}_T^v\ ]$$
in closed or semi-closed form.

3. Estimate $g(s,v)$ to be

$$\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}[h(S_T) \mid X_0 = (S,v), [v^j]_0^T].$$

By computing $\mathbb{E}[h(S_T) \mid X_0 = (S,v), [v^j]_0^T]$, one essentially "integrates out" the randomness with respect to S and the variance of our MC estimator is significantly reduced in comparison to a full MC simulation. Also, the deterministic numerical analysis problem is on $\mathbb{R}$ rather than $\mathbb{R}^2$. In the last decade this method has seen renewed interest in [46], [45], [63] where it is more fleshed out. Also, with the emergence of complex foreign exchange (FX) option pricing models this method has been applied in the works of [3], [15], [13], [16]. In this case one typically has four types of variables: an exchange rate S_t, a domestic and foreign interest rate r_t^d, r_t^f and a volatility process for the exchange rate σ_t. As interest rates may be multi-factor, these models have dimension $d \geq 4$ and so benefit greatly from the hybrid approach. For certain models, conditional on a path of σ_t, the FX system falls within the affine class of models so that Step 2 above may be computed very efficiently.

Theoretically, the algorithm's convergence is provided by the Strong Law of Large Numbers (SLLN) and Central Limit Theorem (CLT) when v_t is simulated exactly, or when its SDE is discretized for a fixed number of time steps. An interesting problem is the quanification of the variance reduction which has been discussed to some extent in [13], [15].

In [45], the hybrid algorithm is extended to allow $g(s,v)$ to be computed for a wide range of v using regression. From there, the authors show how one can also price Bermudan style options, however, the method is briefly touched on and demonstrated with a simple two period problem. The purpose of this thesis is to study this hybrid LSMC-PDE algorithm. In what follows, we provide the theoretical foundation, a proof of almost sure convergence, methods to improve the run times and compute sensitivities, along with numerous examples.

1.3 Outline of Thesis

In this thesis, we develop a numerical approach for solving multi-dimensional discrete time optimal stopping problems under SV that combines LSMC with PDE techniques. The algorithm provides dimensional reduction from the PDE and regression perspective along with variance and dimensional reduction from the MC perspective.

In Chapter 2, we lay the mathematical foundations for mixed MC-PDE techniques. Next, we prove the basic mechanics of the algorithm and, under certain mild assumptions, demonstrate it converges almost surely using methods from point-set topology. Afterwards, we apply the algorithm to the one dimensional Heston model and compare it to traditional approaches to LSMC in terms of prices and optimal exercise boundaries.

In Chapter 3, we describe methods for reducing the complexity and run time of the algorithm along with a technique for computing sensitivities. To reduce the complexity, we apply two methods: clustering via sufficient statistics and mlMC. In order to compute sensitivities, we employ a grid based method for derivatives with respect to the asset, S, and the MC based method of [64]

using initial dispersions for sensitivities with respect to v. For the standard Heston model, we validate our complexity reduction approximations and sensitivities again with respect to the reference solution. Finally, we consider the multi-dimensional Heston model and demonstrate the quality of the hybrid algorithm's new additions.

In Chapter 4, we highlight the importance of multi-factor SV models and apply our hybrid algorithm to two specific examples: the Double Heston model and a mean-reverting commodity model with jumps. For each example, we report prices, sensitivities, and optimal exercise boundaries.

1.4 Notation

In this section we briefly discuss some notational conventions that appear throughout this thesis.

- Given a function $f : \mathbb{R}_+ \times \mathbb{R}^n \mapsto \mathbb{R}$ (i.e., $f(t, Z)$) we often denote the canonical stochastic processes corresponding to it as $(f_t)_{t\in[0,T]}$ with $f_t := f(t, Z_t)$, where Z_t is some n-dimensional stochastic process.
- When defined, we denote the Fourier transform of f as $\mathcal{F}[f](\xi), \widehat{f}(\xi)$ or $(f)\widehat{\ }(\xi)$.
- For $x \in \mathbb{R}^n$ or $x \in \mathbb{M}_{n\times m}(\mathbb{R})$, we define its Euclidean norm by $|x|$.
- Given any function $h : \mathbb{R}^n \to \mathbb{R}$, we write $||h||_\infty := \sup_{x\in\mathbb{R}^n} |h(x)|$ and supp $h := \text{cl}\{x \in \mathbb{R}^n \mid |h(x)| > 0\}$.
- Let $A \subset \mathbb{R}^n$, we let $C^0(A)$ denote the set of continuous functions that vanish at infinity. That is, for $\varepsilon > 0$ if $f \in C^0(A)$ then $\{x \mid |f(x)| \geq \varepsilon\}$ is compact. Furthermore, we define $f \in C_c(A)$ to be the set of compactly supported continuous functions on A.
- We often work with a set of dates $\{t_0, ..., t_M\} \subset [0, T]$ with $\Delta t_k = t_{k+1} - t_k$ and functions $h_{t_i}(S) : \mathbb{R}^{d_S} \to \mathbb{R}$ at each date. To simplify notation, we sometimes suppress the subscript k in Δt_k and h_k.
- Throughout this thesis, we also assume a risk free rate of interest, $r > 0$.

Chapter 2

A Hybrid LSMC/PDE Algorithm

2.1 Introduction

In this chapter we introduce the hybrid LSMC/PDE algorithm that is the central focus of this thesis. We begin by providing the theoretical foundation of mixed MC-PDE approaches for computing future expected values. Once we have this formulation, we proceed to describe the workings of our algorithm followed by a discussion of its most salient features. We then proceed to study the algorithm from a theoretical perspective and prove that it converges almost surely under suitable conditions. Finally we apply the hybrid algorithm to the Heston model and compare it to standard LSMC where we see it provides considerably better estimates of prices and OEBs.

2.2 Probability Spaces, SDEs, and Conditional Expectations

Initial Probability Space

We suppose the existence of a probability space $(\Omega, \mathbb{F}, \mathbb{Q})$ which may accomodate a $d_S + d_v$ dimensional stochastic process $X = (S_t^{(1)}, ..., S_t^{(d_s)}, v_t^{(1)}, ...v_t^{(d_v)})_{t\in[0,T]}$ satisfying a system of SDEs with a strong, unique solution. We further suppose this system of SDEs exhibits one-way coupling in a sense which we describe below. We begin by defining mappings

$$\begin{aligned}
\mu_S &: [0,T] \times \mathbb{R}^{d_S+d_v} \to \mathbb{R}^{d_S}\,, \\
\mu_v &: [0,T] \times \mathbb{R}^{d_v} \to \mathbb{R}^{d_v}\,, \\
\sigma_S &: [0,T] \times \mathbb{R}^{d_S+d_v} \to \mathbb{M}_{d_S\times d_W}(\mathbb{R})\,, \\
\sigma_v &: [0,T] \times \mathbb{R}^{d_v} \to \mathbb{M}_{d_v\times d_{W,v}}(\mathbb{R})\,,
\end{aligned}$$

and a d_W-dimensional Brownian motion, $W = (W_t)_{t\in[0,T]}$, with independent components where $d_{W,v} < d_W$. We also denote the final $d_{W,v}$ components of W_t as W_t^v, i.e., $W_t^{v,i} = W_t^{i-d_W+d_{W,v}}$ $\forall\ i \in \{d_W - d_{W,v} + 1, \ldots, d_W\}$. The process $X = (X_t)_{t\in[0,T]}$ is assumed to satisfy the following

system of SDEs

$$\begin{aligned} dS_t &= \mu_S(t, S_t, v_t)dt + \sigma_S(t, S_t, v_t) \cdot dW_t\,, \qquad \text{and} \\ dv_t &= \mu_v(t, v_t)dt + \sigma_v(t, v_t) \cdot dW_t^v\,. \end{aligned}$$

While our algorithm may be formulated for such a general system of SDEs, our theoretical results and numerical examples assume that there exist mappings $\widetilde{\mu}_S, \widetilde{\sigma}_S$ such that

$$\begin{aligned} \mu_S^{(i)}(t, S, v) &= S^{(i)}\widetilde{\mu}^{(i)}(t, v) \\ \sigma_S^{(i)}(t, S, v) &= S^{(i)}\widetilde{\sigma}^{(i)}(t, v) \end{aligned} \tag{2.1}$$

where the superscript i indicates the i^{th} component of the vector S and i^{th} row of the matrices $\mu_S, \sigma_S, \widetilde{\mu}_S$ and $\widetilde{\sigma}_S$.

Finally, let $\mathcal{F}_{s,t}^v = \sigma(v_u)_{u\in[s,t]}$, $\mathcal{F}_{s,t}^S = \sigma(S_u)_{u\in[s,t]}$ and $\mathcal{F}_{s,t}^{W^v} = \sigma(W_u^v)_{u\in[s,t]}$. To extend this notation, we sometimes write $\mathcal{F}_t^Z := \mathcal{F}_{0,t}^Z$ for some process Z.

Given $t_n \in [0, T]$, we define a new class of (conditional) probability measures $\mathbb{Q}_{t_n,S}$ via $\mathbb{Q}_{t_n,S}(B) = \mathbb{Q}(B \mid S_{t_n} = S)$ for $B \in \mathcal{F}_{t_n,T}^S \vee \mathcal{F}_{t_n,T}^v$.

For a realization of v_t and W_t^v on $[t_n, t_{n+1}]$, which we denote as $[v]_{t_n}^{t_{n+1}}$, we assume there exists a finite-dimensional statistic of the path,

$$\Lambda_n : C_0([t_n, t_{n+1}])^{d_v+d_{W^v}} \to \mathbb{R}^{d_\Lambda},$$

such that the following Markovian-like relation holds

$$\begin{aligned} &\mathbb{E}^{\mathbb{Q}}[\ h(S_{t_{n+1}}, v_{t_{n+1}}) \mid S_{t_n} = S\ ,\ \mathcal{F}_{t_n,t_{n+1}}^{W^v}\ ] \\ &\quad = \mathbb{E}^{\mathbb{Q}}[\ h(S_{t_{n+1}}, v_{t_{n+1}}) \mid S_{t_n} = S\ ,\ \Lambda_n([v]_{t_n}^{t_{n+1}}),\ v_{t_{n+1}}\ ] \end{aligned} \tag{2.2}$$

where $h : \mathbb{R}^{d_S+d_v} \to \mathbb{R}$ is Borel measurable such that the expectations are well-defined. Intuitively, the conditional expectation generates a PDE over $[t_n, t_{n+1}] \times \mathbb{R}^{d_S}$ which depends on the simulated path, $[v]_{t_n}^{t_{n+1}}$ as follows

$$\begin{cases} \dfrac{\partial g}{\partial t}(t, S) + \mathcal{L}^{[v]}g(t, S) = 0\,, \\ \qquad\qquad g(t_{n+1}, S) = h_{n+1}(S, v_{t_{n+1}})\,. \end{cases}$$

The vector $\Lambda_n([v]_{t_n}^{t_{n+1}})$ captures the dependency of g on $[v]_{t_n}^{t_{n+1}}$ over $[t_n, t_{n+1})$, as induced by $\mathcal{L}^{[v]}$, and $v_{t_{n+1}}$ corresponds to the PDE's boundary conditions. For example, in the Heston model, it can be shown that Λ_n takes the following form

$$\Lambda_n([v]_n^{n+1}) = \left(\int_{t_n}^{t_{n+1}} \sqrt{v_s}\ dW_s^v,\ \int_{t_n}^{t_{n+1}} v_s\ ds \right).$$

It is worth noting that generating the conditional PDE for (2.2) is, in general, non-trivial and to the best of our knowledge there are two approaches in the literature: the drift discretization method in [45], [13], [15] and conditionally-affine decomposition of [16]. Since our numerical

examples assume that (2.1) holds, conditional on a volatility path, our model is affine, and we employ the latter technique followed by Fourier Space Time-Stepping (FST) (see [59]) to solve the associated conditional PDEs.

In our algorithm we will often consider expectations of the form

$$\mathbb{E}\left[\ \phi(v_{t_n})h(S_{t_{n+1}}, v_{t_{n+1}}) \,\middle|\, S_{t_n} = S\ ,\ \mathcal{F}^{W^v}_{t_n,t_{n+1}}\right]$$

which leads us to define

$$\Theta_n([v]_n^{n+1}) = (v_{t_n}, \Lambda_n([v]_n^{n+1}), v_{t_{n+1}})$$

where $\phi : \mathbb{R}^{d_v} \to \mathbb{R}$. In later sections we will replace $\Theta_n([v]_n^{n+1})$ with $[v]_{t_n}^{t_{n+1}}$ and simply write

$$\mathbb{E}\left[\ \phi(v_{t_n})h(\ S_{t_{n+1}}, v_{t_{n+1}}\ ) \,\middle|\, S_{t_n} = S\ ,\ [v]_{t_n}^{t_{n+1}}\ \right] := \mathbb{E}\left[\ \phi(v_{t_n})h(S_{t_{n+1}}, v_{t_{n+1}}) \,\middle|\, S_{t_n} = S\ , \Theta_n([v]_n^{n+1})\ \right].$$

Equation (2.2) gives rise to the mappings $G_{f,t_n,S}$ defined by

$$\begin{aligned} &G_{f,t_n,S} : \mathbb{R}^{d_\Theta} \to \mathbb{R}\,, \\ &G_{f,t_n,S}(\theta) = \mathbb{E}^{\mathbb{Q}}\left[\ f(S_{t_{n+1}}, v_{t_{n+1}}, v_n) \mid S_{t_n} = S,\ \Theta_n = \theta\ \right] \end{aligned} \tag{2.3}$$

and the conditional probability measures $\mathbb{Q}_{t_n,S,\theta,v}$ defined via

$$\mathbb{Q}_{t_n,S,\theta}(B) := \mathbb{Q}(\ B \mid S_{t_n} = S,\ \Theta_n = \theta)$$

for $B \in \sigma(S_{t_{n+1}}) \vee \sigma(v_{t_{n+1}}) \vee \sigma(v_{t_n})$. Letting $\widetilde{\mathbb{Q}}_{\Theta_n}$ denote the distribution of Θ_n on $\mathbb{R}^{d_\Theta}$ we have the following relation

$$\begin{aligned} &\mathbb{E}^{\mathbb{Q}}\left[\ f(S_{t_{n+1}}, v_{t_{n+1}}, v_{t_n}) \mid S_{t_n} = S\right] \\ &\quad = \int_{\mathbb{R}^{d_\Theta}} \int_{\Omega} f(S_{t_{n+1}}(\omega), v_{t_{n+1}}, v_{t_n})\ d\mathbb{Q}_{t_n,S,\theta}(\omega)\ d\widetilde{\mathbb{Q}}_{\Theta_n}(\theta). \end{aligned} \tag{2.4}$$

Inherited Sampling Probability Space

A consequence of one-way coupling is the ability to simulate paths of v_t independently of S_t. We now make sense of the notion of an iid collection of sample paths of v_t. Since we only realize v_t through the statistics Θ_n, we only describe how to generate iid copies of Θ_n.

Denote the ordered subset $\{0 = t_0, ..., t_n, t_{n+1}, ..., t_N = T\} \subset [0, T]$, and let $\{[t_n, t_{n+1}]\}_{n=0}^{N-1}$ be the corresponding intervals. Given a path v_t on $[0, T]$, define the $d_\Theta \times N$ dimensional matrix

$$\Theta([v]) = [\ \Theta_0([v]_0^1)\ , \ldots,\ \Theta_{N-1}([v]_{N-1}^N)\,].$$

This random matrix induces a measure $\widetilde{\mathbb{Q}}_\Theta$ on $\mathbb{R}^{Nd_\Theta}$. Given $\widetilde{\mathbb{Q}}_\Theta$, we introduce a new probability space $(\Omega', \mathcal{F}', \mathbb{Q}')$ equipped with a collection of independent random matrices

$$\{\Theta([v^j])\}_{j=1}^{\infty}\,,$$

such that each $\Theta([v^j])$ has distribution $\widetilde{\mathbb{Q}}_\Theta$ on $\mathbb{R}^{Nd_\Theta}$. This construction follows from Kolmogorov's Extension Theorem applied to measures on $\mathbb{R}^{Nd_\Theta}$ (see Appendix A). It then follows that each column $\Theta_n([v^j]_n^{n+1})$ has distribution $\widetilde{\mathbb{Q}}_{\Theta_n}$ on $\mathbb{R}^{d_\Theta}$. Although the process S_t is not defined on Ω', we can still compute relevant expectations involving this process using $G_{f,t_n,S}(\Theta)$ as defined in (2.3). When taking limits in our algorithm, we consider expressions of the form

$$\lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} \mathbb{E}^{\mathbb{Q}} \left[\; f(S_{n+1}, v_{t_{n+1}}, v_{t_n}) \;\; | \;\; S_{t_n} = S, [v^j]_n^{n+1} \right]$$

which, by the SLLN, will converge to $\mathbb{E}^{\mathbb{Q}'} \left[G_{f,t_n,S} \left(\Theta \left([v]_{t_n}^{t_{n+1}} \right) \right) \right]$ a.s. under $\mathbb{Q}'$. For our purposes, however, we require convergence to

$$\mathbb{E}^{\mathbb{Q}} \left[\; f(S_{t_{n+1}}, v_{t_{n+1}}, v_{t_n}) \;\; | \;\; S_{t_n} = S \right].$$

To establish the equivalence between these expressions, we note

$$\begin{aligned}
\mathbb{E}^{\mathbb{Q}'} \left[G_{f,t_n,S} \left(\Theta \left([v]_{t_n}^{t_{n+1}} \right) \right) \right] &= \int_{\Omega'} G_{f,t_n,S}(\Theta_n([v(\omega')]_n^{n+1}))\; d\mathbb{Q}'(\omega') \\
&= \int_{\mathbb{R}^{d_\Theta}} G_{f,t_n,S}(\theta)\; d\widetilde{\mathbb{Q}}_{\Theta_n}(\theta) \\
&= \int_{\mathbb{R}^{d_\Theta}} \mathbb{E}^{\mathbb{Q}} \left[f(S_{t_{n+1}}, v_{t_{n+1}}, v_{t_n}) \mid S_{t_n} = S, \Theta_n = \theta \right] d\widetilde{\mathbb{Q}}_{\Theta_n}(\theta) \\
&= \int_{\mathbb{R}^{d_\Theta}} \int_{\Omega} f(S_{t_{n+1}}(\omega), v_{t_{n+1}}, v_{t_n})\; d\mathbb{Q}_{t_n,S,\theta,n}(\omega)\; d\widetilde{\mathbb{Q}}_{\Theta_n}(\theta) \\
&= \mathbb{E}^{\mathbb{Q}} \left[f(S_{t_{n+1}}, v_{t_{n+1}}, v_{t_n}) \mid S_{t_n} = S \right] \qquad (2.5)
\end{aligned}$$

where the second equality follows from $\Theta([v]_{t_n}^{t_{n+1}})$ being $\widetilde{\mathbb{Q}}_{\Theta_n}$ distributed.

2.3 Algorithm Overview

We now describe a hybrid-method for computing $V_k(S, v)$ which is based on the [62] approach, but uses conditional PDEs to incorporate dimensional and variance reduction. We begin by giving an intuitive explanation and provide a formal, pseudo-code based, description in 2.A.

2.3.1 Description

We simulate N paths of v starting from the initial value v_0. Each path of v_t over $[t_k, t_{k+1}]$ is represented as $[v]_k^{k+1}$. Given a product set $\mathcal{S} \subset \mathbb{R}^{d_S}$, we compute V_k over the domain $\mathcal{S} \times \mathbb{R}^{d_v}$. The set $\mathcal{S}$ is the domain of the conditional expectations that we compute; theoretically it is treated as a continuum whereas in practice it is the grid for our numerical PDE solver. We abuse notation and simply write $\mathcal{S}$ for both situations where the distinction is clear from context. We suppose the discretized form of $\mathcal{S}$ has N_s points in each dimension so that there are $N_s^{d_S}$ points in total. Given the value $V_{k+1}(S_{k+1}, v_{k+1})$ of the option at time t_{k+1} we proceed to compute the continuation value at t_k. The algorithm begins at time $t_M = T$ where $V_M(S_M, v_M) = h_M(S_M)$.

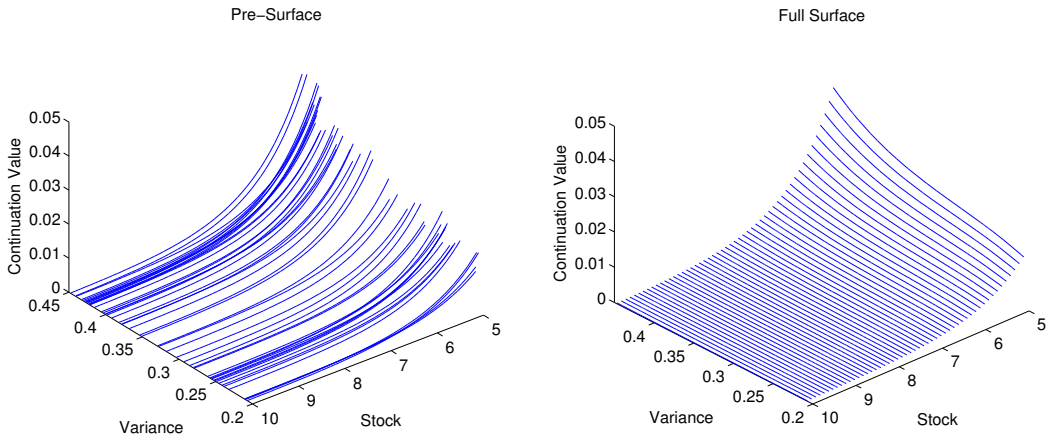

Figure 2.1: The pre and completed continuation surface. The pre-surface is generated by solving a PDE along each path. It is smooth along the S axis, and noisy across the v axis. The completed surface is generated by regressing across the v-axis.

Solving along S to obtain the pre-surface

For each simulation path $j \in \{1, \ldots, N\}$, at time $k = M - 1$, we compute

$$C_k^j(s_i) := e^{-r\Delta t}\, \mathbb{E}^{\mathbb{Q}} \left[\; V_{k+1}(S_{k+1}, v_{k+1}^j) \;\; | \, S_k = s_i \, , \; [v^j]_k^{k+1} \; \right]$$

for all $s_i \in \mathcal{S}$. These may be computed globally over $\mathcal{S}$ for each path using a numerical PDE solver, or, if the model conditional on v^j admits a closed or semi-analytic closed-form solution, then this may be used for each $s_i \in \mathcal{S}$ for each path.

Regress across v to obtain the completed surface

For each $s_j \in \mathcal{S}$, from the previous step, we have N realizations of the continuation value from each volatility path, i.e., $\{C_k^j(s_i)\}_{j=1}^N$.

Next, apply least-squares regression to project this onto a family $\{\phi_l(\cdot)\}_{l=1}^L$ of linearly independent basis functions over our volatility space. This results in a vector of coefficients $a(s_i)$ of length L, and provides the continuation value at $S_{t_k} = s_i$ for any point in the volatility space as follows:

$$C_k(s_i, v) = \sum_{l=1}^{L} a_{l,k}(s_i)\phi_l(v) = a_k(s_i) \cdot \phi(v).$$

Obtaining the Option Value

The price of the option is then given by $V_k(s_i, v) = \max(h_k(s_i), C_k(s_i, v))$. These steps are repeated for all times t_k where $k = M - 1, ..., 1$.

A Direct Estimate on the Time Zero Value

Since, at time zero, there is only a single value for v_0 we obtain an estimate for our value function as

$$V_{h,0}(s_i, v_0) = \max\left(h_{t_0}(s_i) \ , \ \tfrac{1}{N}\sum_{j=1}^{N} e^{-r\Delta t}\, \mathbb{E}^{\mathbb{Q}}\left[V_1(S_1, v_1) \ \middle| \ S_0 = s_i \ , \ [v^j]_0^1 \right] \right) .$$

This estimate tends to be biased high, although as discussed in Section 1.1.2, as with standard LSMC, it is not clear that this holds in general. In [40], additional conditions are given under which this estimator is indeed biased high and one may attempt to extend their work to the current setting. In Corollary 6, we discuss its convergence properties.

A Lower Estimate on the Time Zero Value

Given our estimated regression coefficients, we obtain a sub-optimal exercise policy $\tau(t, S, v)$ defined on $\{t_1, ..., t_{M-1}\} \times \mathcal{S} \times \mathbb{R}^{d_v}$. Thus, we may define a lower estimate via the expectation

$$\mathbb{E}^{\mathbb{Q}}\left[\ e^{-r\tau} h(S_\tau) \mid S_0, v_0 \ \right]. \tag{2.6}$$

In traditional LSMC, one simulates a new independent set of paths (S_t, v_t) to approximate (2.6). While in some cases this approach for a low estimator may be appealing, we describe a hybrid method for computing (2.6) which is reminiscent of pricing a barrier option from a financial perspective.

To this end, we denote the t_k holding and exercise regions by Γ_k and Γ_k^c, respectively. We then simulate N new independent paths of v_t on $[0, T]$, compute

$$\mathbb{E}^{\mathbb{Q}}\left[\ e^{-r\tau} h(S_\tau) \mid S_0, [v^j]_0^T \ \right] \tag{2.7}$$

via a PDE approach for each j, and take the average. To compute (2.7), for each j, first set $V_M^j(S, v) = h(S)$. Next, compute

$$U_{M-1}^j(S) = e^{-r\Delta t}\, \mathbb{E}^{\mathbb{Q}}\left[V_M^j(S_M, v_M^j) \mid S_{t_{M-1}} = S, [v^j]_{M-1}^M \right]$$

via a PDE method for all $S \in \mathcal{S}$. The value surface at time $t = t_{M-1}$ is then given by

$$V_{M-1}^j(S) = U_{M-1}^j(S) \cdot I_{(S, v_{M-1}^j) \in \Gamma_{M-1}} + h(S) \cdot I_{(S, v_{M-1}^j) \in \Gamma_{M-1}^c}.$$

After repeating this procedure for times $k = M-2, ..., 1$ we obtain the lower estimate

$$V_{l,0}(S, v_0) = \frac{1}{N}\sum_{j=1}^{N} e^{-r\Delta t}\, \mathbb{E}^{\mathbb{Q}}\left[\ V_1^j(S_1, v_1^j) \mid S_{t_1} = S, [v^j]_0^1 \ \right] \tag{2.8}$$

for all $S \in \mathcal{S}$. It should be mentioned, however, that in practice this estimator may not be truly biased low due to the bias introduced by the discretization of our PDE solver.

2.3.2 Discussion

We refer the reader to 2.A for a pseudo-code based formal description.

Although we solve a PDE over thousands of paths of v_t over each time interval $[t_n, t_{n+1}]$ and solve a linear-regression problem for each $s_i \in \mathcal{S}$ and t_n, the computational costs and run time are not as high as they may seem. First, the PDEs over each volatility path, and the regressions at $s_i \in \mathcal{S}$, are independent and can be parallelized. Also, based on (2.11), the regression problems at time t_n require only one matrix inversion, and its result is applied to each of the $N_s^{d_S}$ regression sites. Finally, in Chapter 3, we discuss two model-free methods which allow us to significantly reduce the algorithm's complexity.

We immediately see that our algorithm provides dimensional reduction from the PDE and regression perspective, and variance reduction from the MC perspective. If one employs a fully numerical scheme to solve the conditional PDEs, the algorithm is capable, in principle, of handling $3 + n$ dimensional problems where one solves a PDE over three dimensional asset space and simulates n volatility variables. Although our set up is described in the context of an asset-volatility space setting, the algorithm can be applied to any situation where certain variables appear in the payoff and others appear in the background. For these settings, one should simulate the background variables and solve PDEs over the variables that appear in the payoff function.

As we shall see, the algorithm tends to be accurate, for a given computational budget, in determining the time-zero value surface, optimal exercise regions and sensitivities with respect to S. This may be attributed to the stability provided by our PDE grid, $\mathcal{S}$. When the conditional PDEs are solved along each path, we obtain our pre-surface as described in Section 2.3.1. At this point one has two choices: global or local regression. As seen when standard LSMC is applied to SV problems, direct global regression, using polynomials, tends to perform quite poorly due to a lack of flexibility. One may then turn to some type of fixed or adaptive local regression as in [5] or [40] which is essentially what we do in Section 2.3.1. Our regression approach can be viewed as a special type of local regression which is tailored to the presence of $\mathcal{S}$ and is equivalent to local regression onto $N_s^{d_S}$ carefully chosen bundles. If $N_s = 512$ and $d_S = d_v = 2$, we are regressing onto $262,144$ families of basis functions at the cost of inverting a single matrix of size, $L \times L$ where L is only 10, and carrying out matrix multiplication at each $S \in \mathcal{S}$. Also, $C_n(s_j, v)$ is typically simple to fit as a function of v and does not require more than three or four monomials in each volatility dimension thus eliminating the Basis Selection Problem referred to in Section 1.1.2.

Working with $\mathcal{S}$ has other advantages as well. In comparison to standard approaches to LSMC, there is a fundamental shift in how we compare the continuation value to the exercise value and locate the exercise boundary. At time n, when setting the value of V_n for each $s_i \in \mathcal{S}$ we have

$$V_n^N(s_i, v) = \max\left(h_n(s_i), C_n^N(s_i, v)\right) ,$$

and note that $h_n(s_i)$ is a deterministic constant as opposed to a function of a random variable. Thus, we have reduced the problem of locating the boundary from a global problem over the variables S, v to a sequence of lower dimensional problems which are simpler in nature and exhibit less noise. Our approach essentially stores a function in v-space within each element of $\mathcal{S}$ and

allows us to write our continuation surface using the separating representation

$$C_n^N(s_i, v) = a_n^N(s_i) \cdot \phi(v).$$

This leads us to name to our algorithm *Grid Path Separation Method* (GPSM) which is in line with the naming schemes currently in the literature. Another possible name is *LSMC-PDE* which is given in [23] as the algorithm may be viewed as a down the middle hybrid between LSMC and PDE approaches. We also refer to LSMC as *Least Squares Method* (LSM).

From the design of experiments perspective, our algorithm may be viewed as a batched design nested within a probabilistic design. The simulation of v_t stemming from v_0 on $[0, T]$ and the repeated use of its paths, corresponds to a probabilistic design for the variable v. The solving of the conditional PDEs may be viewed as a sort of batched design for S in the following sense. At time t_n, we solve a conditional PDE over $[v^j]_n^{n+1}$ and obtain the pre-regression continuation value of the option along this path for each $S \in \mathcal{S}$. This procedure is equivalent to selecting batches at each $S \in \mathcal{S}$, simulating an "infinite" number of paths of S_t on $[t_n, t_{n+1}]$ conditional on $S_{t_n} = S$ and $[v^j]_n^{n+1}$ and equating the pre-regression continuation value for $[v^j]_n^{n+1}$ to the average payoff of these "infinite" number of paths.

2.4 Theoretical Aspects of the Algorithm

2.4.1 Coefficient Identities

In this section, we show that the GPSM, as described in Section 2.3, is well defined and converges probabilistically. For notational convenience, we suppose the risk free rate is 0.

Idealized Coefficients

For each n, we consider a family of *idealized continuation functions*, C_n, which are constructed by means of backwards induction. We begin by writing $C_M \equiv 0$ and $C_n(S, v) = a_n(S) \cdot \phi(v)$ for $n < M$ where $a_n(S)$ results from regressing the random variable

$$\mathbb{E}^{\mathbb{Q}}\left[\max(h_{n+1}(S_{n+1}), C_{n+1}(S_{n+1}, v_{n+1})) \mid S_n = S,\ v_n\right] \quad \text{onto the basis} \quad \{\phi_l(\cdot)\}_{l=1}^{L}$$

for each $S \in \mathcal{S}$. The coefficient vector $a_n(S)$ is the vector that minimizes the mapping $H_{n,S} : \mathbb{R}^L \to \mathbb{R}$ defined by

$$H_{n,S}(a) = \mathbb{E}^{\mathbb{Q}}\left[\left(\mathbb{E}^{\mathbb{Q}}\left[f_{n+1}(S_{n+1}, v_{n+1}) \mid S_n = S, v_n\right] - a \cdot \phi(v_n)\right)^2 \mid S_n = S\right] \tag{2.9}$$

where

$$\begin{aligned} f_n(S, v) &= \max(h_n(S), C_n(S, v)) \quad \text{for} \quad n < M\ , \text{ and} \\ f_M(S, v) &= h_M(S)\ . \end{aligned} \tag{2.10}$$

To minimize H, we obtain the first order conditions and obtain the normal equations, resulting in the coefficients

$$a_n(S) = A_n^{-1} \cdot \mathbb{E}^{\mathbb{Q}} \left[\phi(v_n) f_{n+1}(S_{n+1}, v_{n+1}) \mid S_n = S \right]$$

where $A_n = \mathbb{E}^{\mathbb{Q}}[\phi(v_{t_n})\phi(v_{t_n})^\intercal]$.

Almost Idealized Coefficients

Next, for a fixed t_n, we define a new type of continuation value, called the *almost-idealized continuation functions*, $\widetilde{C}_n^N(S, v) = \widetilde{a}_n^N(S) \cdot \phi_n(v)$. These random variable are obtained by running the dynamic programming algorithm with the idealized continuation value at all times $k = M, \ldots, n+1$. At time step n we then estimate $\widetilde{a}_n^N(S)$ using our N paths of v_t and future idealized continuation values. This gives us the following regression coefficients for each $\omega' \in \Omega'$

$$\widetilde{a}_n^N(S, \omega') = \left[A_n^N(\omega') \right]^{-1} \frac{1}{N} \sum_{j=1}^{N} \phi(v_n^j(\omega')) \cdot \mathbb{E}^{\mathbb{Q}} \left[f_{n+1}(S_{n+1}, v_{n+1}^j(\omega')) \mid S_n = S, [v^j(\omega')]_n^{n+1} \right]$$

where $A_n^N(\omega') = \frac{1}{N} \sum_{j=1}^{N} \phi(v_n^j(\omega'))\phi(v_n^j(\omega'))^T$. Note that f_{n+1} involves the idealized continuation value at time $n+1$.

Estimated Coefficients

The *estimated continuation functions* are the continuation functions produced from our algorithm: $C_n^N(S, v) = a_n^N(S) \cdot \phi_n(v)$. The regression coefficients are given by

$$a_n^N(S, \omega') = \left[A_n^N(\omega') \right]^{-1} \cdot \frac{1}{N} \sum_{j=1}^{N} \phi(v_n^j(\omega')) \cdot \mathbb{E}^{\mathbb{Q}} \left[f_{n+1}^N(S_{n+1}, v_{n+1}^j(\omega'), \omega') \mid S_n = S, [v^j(\omega')]_n^{n+1} \right] \tag{2.11}$$

where

$$\begin{aligned} &f_n^N(S, v, \omega') = \max(h_n(S), C_n^N(S, v, \omega')) \quad \text{for} \quad n < M\,, \text{ and} \\ &f_M^N(S, v, \omega') = h_M(S). \end{aligned}$$

2.4.2 Truncation Scheme

We now state the following truncation scheme for our least-squares regression. It ensures that the coefficients produced by the algorithm are well defined and converge in a sense to be described later on.

Assumption 1 (Truncation Conditions)**.**

1. *The basis functions $\{\phi_l\}_{l=1}^L$ are bounded and supported on a compact rectangle.*

2. The norm of the matrix

$$\left[A_n^N(\omega')\right]^{-1} = \left[\frac{1}{N}\sum_{j=1}^{N}\phi(v_n^j(\omega'))\phi(v_n^j(\omega'))^T\right]^{-1}$$

is uniformly bounded for all N, n, provided the inverse is defined.

3. For each $i = 1, ..., M$, the exercise values $h_{t_i}(\cdot)$ are bounded with compact support in $\mathcal{S} \subset \mathbb{R}^{d_S}$.

Condition (1) may be imposed by limiting the support of $\{\phi_l\}$ on a bounded domain as they are typically smooth. By making supp $\phi \subset \mathbb{R}^{d_v}$ to be a very large rectangle, the value function is essentially unaffected.

Condition (2) is imposed by replacing $[A_n^N]^{-1}$ with $[A_n^N]^{-1}I_R^{n,N}$ where $I_R^{n,N}$ is the indicator of the event that $[A_n^N]^{-1}$ is uniformly bounded by some constant R. If $R > |A_n^{-1}|$ then we have $[A_n^N]^{-1}I_R^{n,N} \to A_n^{-1}$ $\mathbb{Q}'$-a.s. Again, by making R a very large constant, this has essentially no effect on the values obtained by the algorithm.

Condition (3) on the functions h are always satisfied in practice as numerically solving a PDE involves truncation of h's domain. As well, many payoffs such as put and digital options are already bounded. Unfortunately, this approach rules out certain parameter regimes of SV models with low order moment explosions as in [2].

Lemma 1. *Given the truncation conditions, the functions $H_{n,S}$ defined in (2.9) are finite valued for all $n = 1, ..., M-1$.*

The proof is omitted due to its simplicity. The next lemma establishes a useful relationship between the idealized, almost idealized and estimated coefficients.

Lemma 2. *Let $n \in \{1, ..., M-2\}$, $S \in \mathcal{S}$. There exists a constant, c, which depends on our truncation conditions, such that*

$$|a_n^N(S) - a_n(S)| \leq c \cdot B_n^N(S) + \delta_n^N(S)$$

where

$$B_n^N(S) = \frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}\left[\ |a_{n+1}^N(S_{n+1}) - a_{n+1}(S_{n+1})| \ \middle|\ S_n = S, [v^j]_n^{n+1}\ \right].$$

and

$$\delta_n^N(S) = |\widetilde{a}_n^N(S) - a_n(S)| \tag{2.12}$$

Proof. Given $n \in \{1, ..., M-1\}$ and $S \in \mathcal{S}$ we have

$$|a_n^N(S) - a_n(S)| \leq |a_n^N(S) - \widetilde{a}_n^N(S)| + |\widetilde{a}_n^N(S) - a_n(S)|.$$

After simplifying, we find

$$
\begin{aligned}
&|a_n^N(S) - \widetilde{a}_n^N(S)| \\
&\quad \leq |[A_n^N]_{R_A}^{-1}| \cdot \frac{1}{N} \sum_{j=1}^{N} \mathbb{E}^{\mathbb{Q}} \left[|f_{n+1}^N(S_{n+1}, v_{n+1}^j) - f_{n+1}(S_{n+1}, v_{n+1}^j)| \mid S_n = S, [v^j]_n^{n+1} \right] \cdot |\phi(v_n^j)| \\
&\quad \leq c' \cdot \frac{1}{N} \sum_{j=1}^{N} \mathbb{E}^{\mathbb{Q}} \left[|f_{n+1}^N(S_{n+1}, v_{n+1}^j) - f_{n+1}(S_{n+1}, v_{n+1}^j)| \mid S_n = S, [v^j]_n^{n+1} \right] \qquad (2.13)
\end{aligned}
$$

We then focus on the difference within the expectation. Using the inequality $|\max(a,b) - \max(a,c)| \leq |b-c|$ we find

$$
\begin{aligned}
&\mathbb{E}^{\mathbb{Q}} \left[|f_{n+1}^N(S_{n+1}, v_{n+1}^j) - f_{n+1}(S_{n+1}, v_{n+1}^j)| \mid S_n = S, [v^j]_n^{n+1} \right] \\
&= \mathbb{E}^{\mathbb{Q}} \left[|\max(h_{n+1}(S_{n+1}), C_{n+1}^N(S_{n+1}, v_{n+1}^j)) - \max(h_{n+1}(S_{n+1}), C_{n+1}(S_{n+1}, v_{n+1}^j))| \mid S_n = S, [v^j]_n^{n+1} \right] \\
&\leq \mathbb{E}^{\mathbb{Q}} \left[|C_{n+1}^N(S_{n+1}, v_{n+1}^j) - C_{n+1}(S_{n+1}, v_{n+1}^j)| \mid S_n = S, [v^j]_n^{n+1} \right] \\
&\leq \mathbb{E}^{\mathbb{Q}} \left[|a_{n+1}^N(S_{n+1}) - a_{n+1}(S_{n+1})| \mid S_n = S, [v^j]_n^{n+1} \right] \cdot |\phi(v_{n+1})| \\
&\leq c'' \cdot \mathbb{E}^{\mathbb{Q}} \left[|a_{n+1}^N(S_{n+1}) - a_{n+1}(S_{n+1})| \mid S_n = S, [v^j]_n^{n+1} \right] \qquad (2.14)
\end{aligned}
$$

where c', c'' depend on our truncation conditions. Substituting (2.14) into (2.13), we obtain the result. □

2.4.3 Separable Models and Almost-Sure Convergence

Overview

In this section we prove the following theorem assuming certain separability properties that we discuss below. At the end of the proof, we also state two Corollaries.

Theorem 4. *Let $n \in \{1, \ldots, M-1\}$ and $S \in \mathcal{S}$ be fixed, then we have*

$$\lim_{N \to \infty} |a_n^N(S) - a_n(S)| = 0$$

$\mathbb{Q}'$-almost surely

Proving the analogous theorem for standard LSMC, as shown in [62], [12], tends to be fairly straightforward, with few assumptions on the model, and uses the following arguments

1. Bound $|a_n^N - a_n|$ in terms of $|a_{n+1}^N - a_{n+1}|$
2. Iterate this bound until the final time step, $M-1$.
3. Due to independence over time $M-1$ to M, the Strong of Large Numbers (SLLN) implies that $\lim_{N\to\infty} |a_{M-1}^N - a_{M-1}| = 0$ a.s.

Although this seems like the natural approach for proving Theorem 4, certain complications arise from the fact that the process S_t is not simulated and only "lives" in $\mathbb{Q}$ as opposed to $\mathbb{Q}'$. At time

n we may observe $S_n = S$ and the distribution of S_{n+1} conditional on $\Theta_n([v]_n^{n+1})$, but S_{n+k} for $k > 1$, is not tangible. To complicate matters further, in Step 2, we have that S_{n+1} feeds into $\mathbb{E}[\,\cdot \mid S_{n+1} = S]$ which is an unknown, non-linear, function of S. To mimic the proof of standard LSMC, we then turn to somehow trying to *separate* the non-linearities in S from the expectation. To carry out the separation, we mainly suppose that if $S_n = S$ a.s. then we may express S_{n+1} as

$$S_{n+1} = S \odot R_n$$

where $S \odot R := (S^{(1)}R^{(1)}, \ldots, S^{(d_S)}R^{(d_S)})$. From a financial perspective, we think of R_n as being the return of S_t over the nth time interval. With this assumption, we are then able to write

$$\mathbb{E}^{\mathbb{Q}}[\, h(S_{n+1}) \mid S_n = S] = \mathbb{E}^{\mathbb{Q}}[\, h(S \odot R_n)\,]$$

which takes care of the difficulties with conditioning. The next step is to bring the dependence on S outside of the expectation which may be handled using the Stone-Weierstrass (SW) theorem (See Appendix A). Loosely speaking, using the SW theorem, we can approximate the function $h(S \odot R)$ by a function of the form

$$\psi(S, R) = \sum_{i=1}^{m} \psi_{i,S}(S)\psi_{i,R}(R)$$

which, upon taking expectations, provides the separating approximation we desire. However, since the SW theorem is a topological result, one needs to impose compactness and continuity assumptions on the model which motivates the following.

Assumption 2 (Separability Conditions)**.** *Let $n \in \{1, ..., M-1\}$*

1. *The process S_t, for all $t \in [0, T]$, takes values in*

$$\mathcal{S} = \left\{(S^{(1)}, ..., S^{(d_S)}) \in \mathbb{R}^{d_S} \mid S^{(i)} > 0, \forall i = 1, ..., d_S\right\}, \qquad a.s.$$

2. *If $S_n = S$ almost surely, then $S_{n+1} = S \odot R_n = (S_n^{(1)}R_n^{(1)}, ..., S_n^{(d_s)}R_n^{(d_S)})$ where R_n is adapted to $\vee_{t=t_n}^{t_M}\mathcal{F}_t$ and does not depend on the value of S_n. R_n takes values in $\mathcal{S}$ a.s.*

3. *The exercise function, h, is continuous with compact support in $\mathcal{S}$. The basis functions ϕ are compactly supported and continuous on $\mathbb{R}^{d_v}$. That is, $h \in C_c(\mathcal{S})$ and $\phi_i \in C_c(\mathbb{R}^{d_v})$.*

Condition (1) limits our analysis to assets which take only positive values such as equities and foreign exchange rates.

Condition (2) allows us to separate our future asset price as a product of its current price and return.

The assumption that R_n take values in $\mathcal{S}$ implies that they are finite valued a.s. As a result,

letting $\varepsilon > 0$, there exists tuples $\{(r_l^{(i)}, r_h^{(i)})\}_{i=1}^{d_S}$, where $r_l^{(i)} > 0$ such that

$$\Omega^{\varepsilon}_{r_l,r_h} = \left\{ \; r_l^{(i)} \leq R_n^{(i)} \leq r_h^{(i)} \mid \forall \; n \in \{1,\dots,M-1\}, \; \forall \; i \in \{1,\dots,d_S\} \; \right\} \subset \Omega \tag{2.15}$$

satisfies $\mathbb{Q}(\Omega^{\varepsilon}_{r_l,r_h}) > 1-\varepsilon$. We also write

$$E^{\varepsilon}_{r_l,r_h} = \left\{ R \in \mathbb{R}^{d_S} \mid r_l^{(i)} \leq R^{(i)} \leq r_h^{(i)}, \forall \; i \in \{1,\dots,d_S\} \; \right\}.$$

Given $E^{\varepsilon}_{r_l,r_h}$, we may find an open set U^{ε} such that $E^{\varepsilon}_{r_l,r_h} \subset U^{\varepsilon}$ and $\overline{U^{\varepsilon}} \subsetneq \mathcal{S}$. By Urysohn's Lemma/Tietze's Extension theorem (see [50]), there exists a map $\eta_E : \mathcal{S} \to \mathbb{R}$ such that $\eta_E = 1$ on $E^{\varepsilon}_{r_l,r_h}$, $\eta_E = 0$ on $\mathcal{S} \setminus U^{\varepsilon}$ and $||\eta_E||_{\infty} \leq 1$, i.e. a bump function supported on E. In most cases, our notation will suppress dependence on r_l, r_h, ε.

Condition (3) allows us to apply the Stone-Weierstrass (SW) theorem which underlies the 'separation technique' that will be demonstrated in upcoming Lemmas. We apply the version of SW for functions on locally compact spaces that *vanish at infinity* (see [25]). Suppose we are given the payoff function for a call option, i.e. $g : \mathbb{R} \to \mathbb{R}$ where $g(x) = (x-K)_+$. To modify g such that it falls within our assumption, we first truncate its support to obtain a function $f(x) = (x-K)_+ I_{(0,R_1)}(x)$ where R_1 is large number. Finally, we continuously extend f on $\mathbb{R}$ such that $f = 0$ on (R_2, ∞) where $R_2 > R_1$. A similar construction may be done for a put option payoff near 0.

The proof takes the following steps

1. Lemma 3. Carry out a geometric construction that allows us to approximately separate functions h of S_{t_n} that are continuous and compactly supported in $\mathcal{S}$. The function that provides the approximate separation is denoted as ψ.

2. Lemma 4. Use the geometric construction to show the explicit relationship between h and the separating functions, ψ and thus demonstrate what is referred to as a separating estimate.

3. Lemma 5. Prove the theorem for $n = M-1$ and also obtain an almost-sure separating estimate for $|a^N_{M-1}(S) - a_{M-1}(S)|$.

4. Lemma 6. Prove the theorem for $n = M-2$ and also obtain an almost-sure separating estimate for $|a^N_{M-2}(S) - a_{M-2}(S)|$. The separating estimate for $n = M-2$ involves the function $\delta^N_{M-2}(S)$ as defined in (2.12).

5. Lemma 7. Develop an almost-sure separating estimate for δ^N_n for all $n \in \{1,\dots,M-2\}$.

6. Proposition 1 . Prove the theorem for $n \in \{1,\dots,M-3\}$ using Lemma 6 and Lemma 7. Also obtain an almost-sure separating estimate for $|a^N_n(S) - a_n(S)|$ which is used during the induction.

As evident from Assumption 2, our proof is for a stylized version of the algorithm where our PDE grid is the continuum, $\mathcal{S}$, as opposed to a finite grid of points. We also note that we suppose the number of basis functions, L, is fixed and that the proof is agnostic to the exact type of basis

functions. Different choices of basis functions and L will lead to more accurate results produced by the algorithm, however, we leave the quantification of this error for future study.

Proofs

Lemma 3. *Let $\varepsilon > 0$ and $h : \mathcal{S} \to \mathbb{R}$ be continuous and compactly supported in $\mathcal{S}$. Let $\widetilde{h} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ be defined via*

$$\widetilde{h}(S, R) = h(S \odot R) \cdot \eta_E(R)$$

where E results from (2.15). There exists a map $\psi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ of the form

$$\psi(S, R) = \sum_{i=1}^{k} \psi_{i,S}(S)\psi_{i,R}(R)$$

such that $\psi_{i,S}, \psi_{i,R} \in C_c^0(\mathcal{S})$ and $||\widetilde{h} - \psi||_\infty < \varepsilon$

Proof. It follows from the properties of the mapping $i(S, R) = S \odot R$ and compact support of η_E and h, that $\widetilde{h}$ is compactly supported in $\mathcal{S}$. We also have that $\widetilde{h}$ is continuous on $\mathcal{S} \times \mathcal{S}$.

To construct ψ, we begin by defining the algebra of functions

$$\mathcal{A} := \left\{ \sum_{i=1}^{k} \psi_{i,S}(S)\psi_{i,R}(R) \mid \psi_{i,S}, \psi_{i,R} \in C_c^0(\mathcal{S}),\ k \in \mathbb{N} \right\}$$

where $C_c^0(\mathcal{S})$ is the set of continuous functions that are compactly supported in $\mathcal{S}$. We now use SW to show $\mathcal{A}$ is dense in $C^0(\mathcal{S} \times \mathcal{S})$ under the uniform metric.

Given two distinct points $\{(S_i, R_i)\}_{i=1}^2$, without loss generality, we assume $S_1 \neq S_2$. We now find bump functions $\eta_{S,1}, \eta_{S,2} \in C_c^0(\mathcal{S})$ that separate S_1, S_2, i.e., $\eta_{S,1}(S_1) = 1, \eta_{S,1}(S_2) = 0$, $\eta_{S,2}(S_1) = 0, \eta_{S,2}(S_2) = 1$, along with a bump function $\eta_R \in C_c^0(\mathcal{S})$ that is supported on R_1 and R_2. Letting $\psi_1(S, R) = \eta_{S,1}(S) \cdot \eta_R(R)$ and $\psi_2(S, R) = \eta_{S,2}(S) \cdot \eta_R(R)$, we have that $\mathcal{A}$ separates points. Since $\mathcal{A}$ contains bump functions supported at each point $(S, R) \in \mathcal{S} \times \mathcal{S}$, it vanishes nowhere. □

Lemma 4. *Let $\varepsilon > 0$ and $h : \mathcal{S} \to \mathbb{R}$ be continuous and supported in $\mathcal{S}$. Let $\widetilde{h}$ be as in the statement of Lemma 3 and $\psi(S, R)$ be a separable ε-approximation of $\widetilde{h}$. There exists a random variable $F(S)$ on Ω' and function $G(S)$ such that*

$$\mathbb{E}^{\mathbb{Q}}\left[\phi(v_n)h(S_{n+1}) \mid S_n = S, [v(\omega')]_n^{n+1}\right] = F(S, \omega') + \phi(v_n(\omega')) \sum_{i=1}^{k} \psi_{S,i}(S)\mathbb{E}^{\mathbb{Q}}\left[\psi_{R,i}(R_n) \mid [v(\omega')]_n^{n+1}\right]$$

$$\mathbb{E}^{\mathbb{Q}}\left[\phi(v_n)h(S_{n+1}) \mid S_n = S\right] = G(S) + \sum_{i=1}^{k} \psi_{S,i}(S)\mathbb{E}^{\mathbb{Q}}\left[\ \phi(v_n)\psi_{R,i}(R_n)\right]$$

where $|F(S, \omega')| \leq c\ \varepsilon,\ |G(S)| \leq c'\ \varepsilon$ for all $S \in \mathcal{S}$, $\omega' \in \Omega'$ where c, c' depend on our truncation and separability conditions.

Proof. We will only show the first inequality as they are similar.

$$\begin{aligned}
&\mathbb{E}^{\mathbb{Q}}\left[\phi(v_n)h(S_{n+1}) \mid S_n = S, [v(\omega')]_n^{n+1}\right] \\
&= \mathbb{E}^{\mathbb{Q}}[\phi(v_n)h(S \odot R_n)(1-\eta_E(R_n)) \mid S_n = S, [v(\omega')]_n^{n+1}] + \mathbb{E}^{\mathbb{Q}}\left[\phi(v_n)h(S \odot R_n)\eta_E(R_n) \mid S_n = S, [v(\omega')]_n^{n+1}\right] \\
&= F_1(S,\omega') + \mathbb{E}^{\mathbb{Q}}\left[\phi(v_n)\widetilde{h}(S \odot R_n) \mid S_n = S, [v(\omega')]_n^{n+1}\right] \\
&= F_1(S,\omega') + \mathbb{E}^{\mathbb{Q}}\left[\phi(v_n)(\widetilde{h}(S \odot R_n) - \psi(S,R_n)) \mid S_n = S, [v(\omega')]_n^{n+1}\right] \\
&\quad + \mathbb{E}^{\mathbb{Q}}\left[\phi(v_n)\psi(S,R_n) \mid S_n = S, [v(\omega')]_n^{n+1}\right] \\
&= F_1(S,\omega') + F_2(S,\omega') + \mathbb{E}^{\mathbb{Q}}\left[\phi(v_n)\psi(S,R_n) \mid S_n = S, [v(\omega')]_n^{n+1}\right] \\
&= F_1(S,\omega') + F_2(S,\omega') + \mathbb{E}^{\mathbb{Q}}\left[\phi(v_n)\psi(S,R_n) \mid S_n = S, [v(\omega')]_n^{n+1}\right] \\
&= F_1(S,\omega') + F_2(S,\omega') + \phi(v_n(\omega'))\sum_{i=1}^{k}\psi_{S,i}(S)\mathbb{E}^{\mathbb{Q}}\left[\psi_{R,i}(R_n) \mid [v(\omega')]_n^{n+1}\right]
\end{aligned}$$

It then follows that $|F_i(S,\omega')| < \varepsilon$ for all $S \in \mathcal{S}$, $\omega' \in \Omega'$, $i = 1, 2$. Finally, we set $F(S,\omega') = F_1(S,\omega') + F_2(S,\omega')$. □

Lemma 5. *Let $\varepsilon > 0$. There exists a set $\Omega'_{M-1} \subset \Omega'$ with $\mathbb{Q}'(\Omega'_{M-1}) = 1$ such that for $\omega' \in \Omega'_{M-1}$ we have the following*

$$|a_{M-1}^N(S,\omega') - a_{M-1}(S,\omega')| \leq c_{M-1}\ \varepsilon + \sum_{i_{M-1}=1}^{k_{M-1}} \alpha_{M-1,M}^{i_{M-1},N}(\omega') \cdot \psi_{S,i_{M-1}}(S)$$

where $\alpha_{M-1,M}^{i_{M-1},N}$ are random variables that depend on $\{[v^j]_{M-1}^M\}_{j=1}^{\infty}$ and

$$\lim_{N\to\infty} \alpha_{M-1,M}^{i_{M-1},N}(\omega') = 0,$$

$\psi_{S,i_{M-1}}$ satisfy the conditions of Lemma 4 and c_{M-1} depends on our truncation and separability conditions.

Proof.

Given $\varepsilon > 0$ we have

$$\begin{aligned}
|a_{M-1}^N(S) - a_{M-1}(S)| = |\ [A_{M-1}^N]^{-1}\frac{1}{N}\sum_{j=1}^{N}\phi(v_{M-1}^j)\mathbb{E}^{\mathbb{Q}}[h(S_M) \mid [v^j]_{M-1}^M, S_{M-1} = S] \\
- A_{M-1}^{-1}\mathbb{E}^{\mathbb{Q}}[h(S_M)\ \phi(v_{M-1}) \mid S_{M-1} = S]\ |.
\end{aligned}$$

By Lemma 4 we can find a constant c and an separable ε-approximation, $\psi_{M-1}(S,R)$ such that

$$\begin{aligned}|a^N_{M-1}(S) - a_{M-1}(S)| \leq \mid \left[A^N_{M-1}\right]^{-1} \frac{1}{N}\sum_{j=1}^{N}\phi(v^j_{M-1}) \sum_{i_{M-1}=1}^{k_M} \psi_{S,i_{M-1}}(S)\mathbb{E}^{\mathbb{Q}}\left[\psi_{R,i_{M-1}}(R_{M-1}) \mid [v^j]^M_{M-1}\right] \\ - A^{-1}_{M-1}\sum_{i_{M-1}=1}^{k_M}\psi_{S,i_{M-1}}(S)\mathbb{E}^{\mathbb{Q}}\left[\phi(v_{M-1})\psi_{R,i_{M-1}}(R_{M-1})\right] \mid + c\,\varepsilon \\ \leq \sum_{i_{M-1}=1}^{k_{M-1}} |\psi_{S,i_{M-1}}(S)| \cdot \alpha^{i_{M-1},N}_{M-1,M} + c\,\varepsilon\end{aligned}$$

where the last line follows from interchanging the summations, c depends on our truncation and separability conditions and

$$\begin{aligned}\alpha^{i_{M-1},N}_{M-1,M} = \mid [A^N_{M-1}]^{-1}\frac{1}{N}\sum_{j=1}^{N}\phi(v^j_{M-1})\mathbb{E}^{\mathbb{Q}}\left[\psi_{R,i_{M-1}}(R_{M-1}) \mid [v^j]^M_{M-1}\right] \\ - A^{-1}_{M-1}\mathbb{E}^{\mathbb{Q}}\left[\phi(v_{M-1})\psi_{R,i_{M-1}}(R_{M-1})\right] \mid. \qquad (2.16)\end{aligned}$$

By the SLLN, there exists a set Ω'_{M-1} with $\mathbb{Q}'(\Omega'_{M-1}) = 1$ such that $\lim_{N\to\infty}\alpha^{i_{M-1},N}_{M-1,M} = 0$ on Ω'_{M-1} for all $i_{M-1} \in \{1,...,k_M\}$. □

Lemma 6. *Let $\varepsilon > 0$. There exists a set Ω'_{M-2} such that $\mathbb{Q}'(\Omega'_{M-2}) = 1$ and that for $\omega' \in \Omega'_{M-2}$*

$$\begin{aligned}|a^N_{M-2}(S,\omega') - a_{M-2}(S,\omega')| \leq c^N_{M-2}(\omega')\,\varepsilon + \sum_{i_{M-1}=1}^{k_{M-1}}\alpha^{i_{M-1},N}_{M-1,M}(\omega')\sum_{i_{M-2}}^{k_{M-2}}\alpha^{i_{M-2},i_{M-1},N}_{M-2,M-1}(\omega')\psi_{S,i_{M-2},i_{M-1}}(S) \\ + \delta^N_{M-2}(S,\omega')\end{aligned}$$

where $\alpha^{i_{M-1},N}_{M-2,M-1}$ are random variables that depend on $\{[v^j]^{M-1}_{M-2}\}_{j=1}^{\infty}$ and satisfy

$$\limsup_{N\to\infty}\alpha^{i_{M-2},N}_{M-2,M-1}(\omega') < \infty.$$

Also, $\limsup_{N\to\infty} c^N_{M-2}(\omega') < \infty$ with a bound depending only on our truncation and separability conditions and $\psi_{S,i_{M-1},i_{M-2}}$ satisfy the conditions of Lemma 4. Lastly, δ^N_{M-2} is as defined in (2.12).

Proof. Given $\varepsilon > 0$, and using Lemma 2 and 5 we have that for $\omega' \in \Omega'_{M-1}$

$$\begin{aligned}|a^N_{M-2}(S,\omega') - a_{M-2}(S,\omega')| &\leq \widetilde{c}\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}\left[|a^N_{M-1}(S_{M-1},\omega') - a_{M-1}(S_{M-1},\omega')| \mid S_{M-2} = S, [v^j(\omega')]^{M-1}_{M-2}\right] \\ &\quad + \delta^N_{M-2}(S,\omega') \\ &\leq c'\varepsilon + \sum_{i_{M-1}}^{k_{M-1}}\alpha^{i_{M-1},N}_{M-1,M}(\omega')\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}\left[\psi_{S,i_{M-1}}(S_{M-1}) \mid S_{M-2} = S\ [v^j(\omega')]^{M-1}_{M-2}\right] \\ &\quad + \delta^N_{M-2}(S,\omega')\end{aligned}$$

For each $\psi_{S,i_{M-1}}$, we apply Lemma 4, and obtain a separable ε-approximating function $\psi_{i_{M-1}}(S,R)$ of the form

$$\psi_{i_{M-1}}(S,R) = \sum_{i_{M-2}=1}^{k_{M-2}} \psi_{S,i_{M-2},i_{M-1}}(S)\psi_{R,i_{M-2},i_{M-1}}(R)$$

and apply it to the expectation. This results in

$$|a^N_{M-2}(S,\omega') - a_{M-2}(S,\omega')| \leq (c' + \widetilde{c}^N_{M-2}(\omega'))\varepsilon + \sum_{i_{M-1}=1}^{k_{M-1}} \alpha^{i_{M-1},N}_{M-1,M}(\omega') \sum_{i_{M-2}=1}^{k_{M-2}} \alpha^{i_{M-2},i_{M-1},N}_{M-2,M-1}(\omega')\psi_{i_{M-2},i_{M-1}}(S) + \delta^N_{M-2}(S,\omega')$$

where $\lim_{N\to\infty} \widetilde{c}^N_{M-2}(\omega') = 0$ and

$$\alpha^{i_{M-2},N}_{M-2,M-1} = \frac{1}{N}\sum_{j=1}^{N} \mathbb{E}^{\mathbb{Q}}\left[\ \psi_{R,i_{M-2},i_{M-1}}(R_{M-2}) \mid [v^j]^{M-1}_{M-2}\ \right]. \tag{2.17}$$

By the SLLN, for each tuple (i_{M-1}, i_{M-2}) there exists a set of full measure such that $\alpha^{i_{M-2},N}_{M-2,M-1}$ converges. Thus, we may find a set $\widetilde{\Omega}'_{M-2}$ with $\mathbb{Q}'(\widetilde{\Omega}'_{M-2}) = 1$ on which they all converge. Also, $\delta^N_{M-2}(S)$ converges to 0 on a set of full measure Ω'^{δ}_{M-2}. We then define $\Omega'_{M-2} = \widetilde{\Omega}'_{M-2} \cap \Omega'_{M-1} \cap \Omega'^{\delta}_{M-2}$ and $c^N_{M-2} = c' + \widetilde{c}^N_{M-2}$ which implies the lemma. □

Lemma 7. *Let $n \in \{1, ..., M-2\}$, $\delta^N_n(S)$ as in (2.12) and $\varepsilon > 0$. There exists a set $\Omega'^{\delta}_n \subset \Omega'$ with $\mathbb{Q}'(\Omega'^{\delta}_n) = 1$ such that for $\omega' \in \Omega'^{\delta}_n$ we have*

$$\delta^N_n(S,\omega') \leq c \cdot \varepsilon + \sum_{i_n=1}^{k_n} \beta^{i_n,N}_{n,n+1}(\omega')\psi_{i_n}(S)$$

where $\beta^{i_n,N}_{n,n+1}$ are random variables that depend on $\{[v^j]^{n+1}_n\}_{j=1}^{\infty}$, $\lim_{N\to\infty} \beta^{i_n,N}_{n,n+1}(\omega') = 0$, c depends on our truncation and separability conditions, and ψ_{i_n} satisfy the conditions of Lemma 4.

Proof. We break the proof up into multiple stages.

Preliminary estimates on a_k, f_k.

Let f_k be as defined in (2.10). We begin by showing that for each $j \in \{n, ..., M-1\}$ that

$$a_j(S) = F_j(S) + G_j(S)$$
$$f_j(S,v) = H_j(S,v) + J_j(S,v)$$

where $|F_k(S)| \leq c_k\varepsilon$, $|H_k(S,v)| \leq c'_k\varepsilon$ with c_k, c'_k depending on our truncation conditions and separability conditions. The functions G_j admit the representation

$$G_j(S) = \sum_{i_j=1}^{k_j} c_{i_j,j}\psi_{S,i_j}(S)$$

where $c_{i_j,j} \in \mathbb{R}^L$ and $\psi_{S,i_j} \in C_c(\mathcal{S})$. Finally $J_j(S,v) = \max(h(S), G_j(S)\cdot\phi(v))$. To this end, we

let $j = M-1$ and have that

$$\begin{aligned} a_{M-1}(S) =& A_{M-1}^{-1}\mathbb{E}^{\mathbb{Q}}\left[\phi(v_{M-1})h(S_M) \mid S_{M-1} = S\right] \\ =& A_{M-1}^{-1}\mathbb{E}^{\mathbb{Q}}[\phi(v_{M-1})h(S \odot R_{M-1})(1-\eta_E(R_{M-1}))\ ] \\ &+ A_{M-1}^{-1}\mathbb{E}^{\mathbb{Q}}\left[\phi(v_{M-1})h(S \odot R_{M-1})\eta_E(R_{M-1})\right] \end{aligned} \tag{2.18}$$

where E is as in (2.4.3) and focus on the second term. We let

$$\begin{aligned} \widetilde{\Xi}_{M-1} &: \mathbb{R}^{2d_S+d_v} \to \mathbb{R}^L \\ \widetilde{\Xi}_{M-1}(S,R,v_1) &= \phi(v_1)h(S \odot R)\eta_E(R) \\ &= \phi(v_1)\widetilde{h}(S,R) \end{aligned}$$

and apply Lemma 3 to find an ε-separating function of the form

$$\psi_{M-1}(S,R) = \sum_{i_{M-1}=1}^{k_{M-1}} \psi_{S,i_{M-1}}(S)\psi_{R,i_{M-1}}(R)$$

where $\psi_{S,i_{M-1}}, \psi_{R,i_{M-1}} \in C_c^0(\mathcal{S})$ and $||\widetilde{h} - \psi_{M-1}||_\infty < \varepsilon$. This allows us to write

$$\widetilde{\Xi}_{M-1}(S,R,v_1) = F_{2,M-1}(S,R,v_1) + \phi(v_1)\psi_{M-1}(S,R)$$

where $F_{2,M-1}(S,R,v_1) = \phi(v_1)(\widetilde{h}(S,R) - \psi_{M-1}(S,R))$ and $\mathbb{E}^{\mathbb{Q}}[F_{2,M-1}(S,R_{M-1},v_{M-1})] < c\,\varepsilon$ with c depending on our truncation and separability conditions.

Returning to our expression in (2.18), we find

$$\begin{aligned} a_{M-1}(S) &= F_{M-1}(S) + \sum_{i_{M-1}=1}^{k_{M-1}} c_{i_{M-1},M-1}\psi_{S,i_{M-1}}(S) \\ &=: F_{M-1}(S) + G_{M-1}(S) \end{aligned}$$

where $|F_{M-1}(S)| \leq c_{M-1}\ \varepsilon$ for all $S \in \mathcal{S}$, c_{M-1} depends on our truncation and separability conditions, and

$$c_{i_{M-1},M-1} = \mathbb{E}^{\mathbb{Q}}\left[\phi(v_{M-1})\psi_{R,i_{M-1}}(R_{M-1})\right].$$

We now turn to $f_{M-1}(S,v)$

$$\begin{aligned} f_{M-1}(S,v) &= \max(h(S), a_{M-1}(S)\cdot\phi(v)) \\ &= \max(h(S), F_{M-1}(S)\cdot\phi(v) + G_{M-1}(S)\cdot\phi(v)) \\ &= \Big[\max(h(S), F_{M-1}(S)\cdot\phi(v) + G_{M-1}(S)\cdot\phi(v)) - \max(h(S), G_{M-1}(S)\cdot\phi(v))\Big] \\ &+ \max(h(S), G_{M-1}(S)\cdot\phi(v)). \end{aligned}$$

Equating $H_{M-1}(S,v)$ to the term in brackets and

$$J_{M-1}(S,v) := \max(h(S), G_{M-1}(S)\cdot\phi(v)),$$

gives us the required form and concludes the claim for $j = M-1$.

Now let $j \in \{1,...,M-2\}$, we have

$$\begin{aligned}
a_j(S) &= A_j^{-1}\mathbb{E}^{\mathbb{Q}}\left[\phi(v_j)f_{j+1}(S_{j+1},v_{j+1})\right] \\
&= A_j^{-1}\mathbb{E}^{\mathbb{Q}}\left[\phi(v_j)H_{j+1}(S_{j+1},v_{j+1})\right] \\
&+ A_j^{-1}\mathbb{E}^{\mathbb{Q}}\left[\phi(v_j)\max(h(S\odot R_j), G_{j+1}(S\odot R_j)\cdot\phi(v_{j+1}))(1-\eta_{E_{j+1}}(R_n)))\right] \\
&+ A_j^{-1}\mathbb{E}^{\mathbb{Q}}\left[\phi(v_j)\max(h(S\odot R_j), G_{j+1}(S\odot R_j)\cdot\phi(v_{j+1}))\eta_{E_{j+1}}(R_n)\right],
\end{aligned}$$

where E_{j+1} is as in (2.15) so that $\mathbb{Q}(\Omega^{\varepsilon_{j+1},c}) < \frac{\varepsilon}{||J_{j+1}||_\infty}$, and focus on the final term. By assumption, $G_{j+1}(S) = \sum_{i_{j+1}=1}^{k_{j+1}} c_{i_{j+1}}\psi_{S,i_{j+1}}(S)$ where $c_{i_{j+1}} \in \mathbb{R}^L$. Letting $d_{i_{j+1}}(v) = c_{i_{j+1}}\cdot\phi(v)$, we are led to consider the function

$$\begin{aligned}
\widetilde{\Xi}_j &: \mathbb{R}^{2d_S+2d_v} \to \mathbb{R}^L \\
\widetilde{\Xi}_j(S,R,v_1,v_2) &= \phi(v_1)\max(h(S\odot R)\eta_{E_{j+1}}(R), \sum_{i_{j+1}=1}^{k_{j+1}} d_{i_{j+1}}(v_2)\psi_{S,i_{j+1}}(S\odot R)\eta_{E_{j+1}}(R)) \\
&= \phi(v_1)\max(\widetilde{h}(S,R), \sum_{i_{j+1}=1}^{k_{j+1}} d_{i_{j+1}}(v_2)\widetilde{\psi}_{S,i_{j+1}}(S,R))
\end{aligned}$$

where $\widetilde{\psi}_{S,i_{j+1}}(S,R) = \psi_{S,i_{j+1}}(S,R)\eta_{E_{j+1}}(R)$. Before carrying out a construction very similar to the $j = M-1$ case, we let

$$A_{j+1} = ||\widetilde{h}||_\infty + \sum_{i_{j+1}=1}^{k_{j+1}} ||d_{i_{j+1}}||_\infty ||\widetilde{\psi}_{S,i_{j+1}}||_\infty.$$

Next, we note that

$$\max\left(\widetilde{h}, \sum_{i_{j+1}=1}^{k_{j+1}} d_{i_{j+1}}\widetilde{\psi}_{S,i_{j+1}}\right) \in C_c^0(\mathcal{S}^2\times\mathbb{R}^{d_v})$$

and find a function of the form

$$\psi_j(S,R,v) = \sum_{i_j=1}^{k_j}\psi_{S,i_j}(S)\psi_{R,v,i_j}(R,v)$$

such that $\psi_{S,i_j} \in C_c^0(\mathcal{S})$, $\psi_{R,v,i_j} \in C_c^0(\mathcal{S} \times \mathbb{R}^{d_v})$ and

$$\left|\left| \max\left(\widetilde{h}, \sum_{i_{j+1}=1}^{k_{j+1}} d_{i_{j+1}} \widetilde{\psi}_{S,i_{j+1}}\right) - \psi_j \right|\right|_\infty < \varepsilon.$$

Finally, we write

$$a_j(S) = F_{j,1}(S) + F_{j,2}(S) + F_{j,3}(S) + \sum_{i_j=1}^{k_j} c_{i_j,j} \psi_{S,i_j}(S)$$

where

$$\begin{aligned}
F_{j,1}(S) &= A_j^{-1} \mathbb{E}^{\mathbb{Q}} \left[\phi(v_j) H_{j+1}(S_{j+1}, v_{j+1}) \mid S_j = S\right], \\
F_{j,2}(S) &= A_j^{-1} \mathbb{E}^{\mathbb{Q}} \left[\phi(v_j) \max(h(S \odot R_j), G_{j+1}(S \odot R_j) \cdot \phi(v_{j+1}))(1 - \eta_{E_{j+1}}(R_n)))\right], \\
F_{j,3}(S) &= \mathbb{E}^{\mathbb{Q}} \left[\widetilde{\Xi}_j(S, R_j, v_j, v_{j+1}) - \phi(v_j) \psi_j(S, R_j, v_{j+1}) \right]
\end{aligned}$$

so that for $i = 1, 2, 3$, we have $||F_{j,i}||_\infty \leq c'_{j,i} \varepsilon$ where $c'_{j,i}$ depend on our truncation and separability conditions and

$$c_{i_j,j} = \mathbb{E}\left[\phi(v_j) \psi_{R,v,i_j}(R_j, v_{j+1})\right].$$

Also we have that $\psi_{S,i_j}(S) \in C_c^0(\mathcal{S})$. Writing $F_j = F_{j,1} + F_{j,2} + F_{j,3}$ and $G_j(S)$ to be the final term completes the claim for $a_j(S)$. Showing the result for $f_j(S_j, v_j)$ is analogous to the base case and so we omit the proof.

Estimates of $\delta_n^N(S)$

We write

$$\begin{aligned}
\delta_n^N(S) = |\ & \left[A_n^N\right]^{-1} \frac{1}{N} \sum_{j=1}^N \mathbb{E}^{\mathbb{Q}} \left[\phi(v_n) f_{n+1}(S_{n+1}, v_{n+1}) \mid S_n = S, [v^j]_n^{n+1}\right] \\
& - A_n^{-1} \mathbb{E}^{\mathbb{Q}} \left[\phi(v_n) f_{n+1}(S_{n+1}, v_{n+1}) \mid S_n = S\right] \ | \\
\leq |\ & \left[A_n^N\right]^{-1} \frac{1}{N} \sum_{j=1}^N \mathbb{E}^{\mathbb{Q}} \left[\phi(v_n) J_{n+1}(S_{n+1}, v_{n+1}) \mid S_n = S, [v^j]_n^{n+1}\right] \\
& - A_n^{-1} \mathbb{E} \left[\phi(v_n) J_{n+1}(S_{n+1}, v_{n+1}) \mid S_n = S\right] \ + c\, \varepsilon \\
\leq |\ & \left[A_n^N\right]^{-1} \frac{1}{N} \sum_{j=1}^N \phi(v_n^j) \mathbb{E}^{\mathbb{Q}} \left[\max(h(S_{n+1}), G_n(S_{n+1}) \cdot \phi(v_{n+1})) \eta_E(R_n) \mid S_n = S, [v^j]_n^{n+1}\right] \\
& - A_n^{-1} \mathbb{E}^{\mathbb{Q}} \left[\phi(v_n) \max(h(S_{n+1}), G_n(S_{n+1}) \cdot \phi(v_{n+1})) \eta_E(R_n) \mid S_n = S\right] \ | \ + c'\, \varepsilon
\end{aligned}$$

where c and c' depend on our truncation conditions. We now focus on the expression within our

expectations and define the function

$$\begin{aligned}\widetilde{\Xi}'_n &: \mathbb{R}^{2d_S+2d_v} \to \mathbb{R}^L \\ \widetilde{\Xi}'_n(S,R,v_1,v_2) &= \phi(v_1)\max(h(S\odot R), G_n(S\odot R)\cdot\phi(v))\ \eta_E(R) \\ &= \phi(v_1)\max(h(S\odot R)\ \eta_E(R), G_n(S\odot R)\eta_E(R)\cdot\phi(v_2)) \\ &= \phi(v_1)\max(\widetilde{h}(S,R), \widetilde{G}_n(S,R)\cdot\phi(v_2)).\end{aligned}$$

Applying techniques that are exaclty analogous to previous steps, we obtain a separating estimate for $\widetilde{\Xi}'$ of the form

$$\widetilde{\Xi}'_n(S,R,v_1,v_2) = F(S,R,v_1,v_2) + \phi(v_1)\sum_{i_n=1}^{k_n}\psi_{S,i_n}(S)\psi_{R,v,i_n}(R,v_2)$$

where F is appropriately bounded and $\psi_{S,i_n} \in C_c^0(\mathcal{S})$. This leads to

$$\delta_n^N(S) \leq\ c\,\varepsilon + \sum_{i_n=1}^{k_n}|\psi_{S,i_n}(S)|\cdot\beta_{n,n+1}^{i_n,N}$$

where c again depends on our truncation conditions and

$$\beta_{n,n+1}^{l,N} = |\ \left[A_n^N\right]^{-1}\ \frac{1}{N}\sum_{j=1}^{N}\phi(v_n^j)\mathbb{E}^{\mathbb{Q}}\left[\psi_{R,v,i_n}(R_n,v_{n+1})\mid [v^j]_n^{n+1}\ \right] - A_n^{-1}\,\mathbb{E}^{\mathbb{Q}}\left[\phi(v_n^j)\psi_{R,v,i_n}(R_n,v_{n+1})\ \right]\ |.$$

By the SLLN, for each i_n, we have that $\lim_{N\to\infty}\beta_{n,n+1}^{i_n,N} = 0$ a.s. As a result, we may find a set $\Omega_n'^{\delta}$ with $\mathbb{Q}'(\Omega_n'^{\delta}) = 1$ such that they all converge which completes the proof. □

Proposition 1. *Let $n \in \{1,...,M-3\}$ and $\varepsilon > 0$. There exists a set Ω'_n such that $\mathbb{Q}'(\Omega'_n) = 1$ and for $\omega' \in \Omega'_n$ we have*

$$|a_n^N(S,\omega') - a_n(S,\omega')| \leq c_n^N(\omega')\ \varepsilon + \delta_n^N(S,\omega') + \alpha_n^N(S,\omega') + \sum_{l=n+1}^{M-2}\beta_{l,n}^N(S,\omega') \tag{2.19}$$

where

$$\alpha_n^N(S,\omega') = \sum_{i_{M-1}=1}^{k_{M-1}}\alpha_{M-1,M}^{i_{M-1},N}(\omega')\sum_{i_{M-2}=1}^{k_{M-2}}\alpha_{M-2,M-1}^{i_{M-2},i_{M-1},N}(\omega')\ \ldots\ \sum_{i_n=1}^{k_n}\alpha_{i_n,i_{n+1}}^{i_n,\ldots,i_{M-1},N}(\omega')\cdot\psi_{i_n,\ldots,i_{M-1}}^{n,\alpha}(S),$$

$$\beta_{l,n}^N(S,\omega') = \sum_{i_l=1}^{p_l}\beta_{l-1,l}^{l,i_l,N}(\omega')\sum_{i_{l-1}=1}^{p_{l-1}}\beta_{l-2,l-1}^{l,i_l,i_{l-1},N}(\omega')\ \ldots\ \sum_{i_n=1}^{p_n}\beta_{n,n+1}^{l,i_l,\ldots,i_n,N}(\omega')\cdot\psi_{i_l,\ldots,i_n}^{n,\beta}(S),$$

$$\lim_{N\to\infty}\alpha_{M-1,M}^{i_{M-1},N}(\omega') = 0,\ \limsup_N \alpha_{k-1,k}^{i_j,\ldots,i_l,N}(\omega') < \infty,\ \forall\ k \in \{n,...,M-1\}$$

and

$$\forall\, l \in \{n+1,\dots,M-2\}, \forall j \in \{n,\dots,l\}$$
$$\lim_{N\to\infty} \beta_{l-1,l}^{l,i_l,N}(\omega') = 0,\ \limsup_N \beta_{k-1,k}^{l,i_j,\dots,i_l,N}(\omega') < \infty.$$

The bounds on $\limsup_N c_n^N$ *depend on our truncation and separability conditions and* $\psi_{i_l,\dots,i_n}^{k,\alpha}, \psi_{i_l,\dots,i_n}^{k,\beta}$ *satisfy the conditions of Proposition 3.*

Proof. We begin by letting $n = M-3$ and by Lemma 2 and Lemma 6 we have on $\omega' \in \Omega'_{M-2} \cap \Omega'^{\delta}_{M-2}$

$$\begin{aligned}
&|a_{M-3}^N(S,\omega') - a_{M-3}(S,\omega')| \\
&\le \delta_{M-3}^N(S,\omega') + \widetilde{c}\frac{1}{N}\sum_{j=1}^N \mathbb{E}^{\mathbb{Q}}\left[|a_{M-2}^N(S_{M-2},\omega') - a_{M-2}(S_{M-2},\omega')| \mid S_{M-3} = S, [v^j(\omega')]_{M-3}^{M-2}\right] \\
&\le \delta_{M-3}^N(S,\omega') + c_{M-2}^N(\omega') \\
&+ \widetilde{c}\sum_{i_{M-1}=1}^{k_{M-1}} \alpha_{M-1,M}^{i_{M-1},N}(\omega') \sum_{i_{M-2}}^{k_{M-2}} \alpha_{M-2,M-1}^{i_{M-2},i_{M-1},N}(\omega') \frac{1}{N}\sum_{j=1}^N \mathbb{E}^{\mathbb{Q}}\left[\psi_{S,i_{M-2},i_{M-1}}(S_{M-2}) \mid S_{M-3} = S, [v^j(\omega')]_{M-3}^{M-2}\right] \qquad (2.20)
\end{aligned}$$

$$+ \widetilde{c}\frac{1}{N}\sum_{j=1}^N \mathbb{E}^{\mathbb{Q}}[\delta_{M-2}^N(S_{M-2},\omega') \mid S_{M-3} = S, [v^j(\omega')]_{M-3}^{M-2}\,] \qquad (2.21)$$

Line (2.20) may be handled just as in the proof of Lemma 6. As for line (2.21), we use Lemma 7 and write

$$\begin{aligned}
&\frac{1}{N}\sum_{j=1}^N \mathbb{E}^{\mathbb{Q}}[\delta_{M-2}^N(S_{M-2},\omega') \mid S_{M-3} = S, [v^j(\omega')]_{M-3}^{M-2}\,] \\
&\le c\,\varepsilon + \sum_{i_{M-2}=1}^{p_{M-2}} \beta_{M-2,M-1}^{M-2,i_{M-2},N}(\omega') \frac{1}{N}\sum_{j=1}^N \mathbb{E}^{\mathbb{Q}}[\psi_{i_{M-2}}(S_{M-2}) \mid S_{M-3} = S, [v^j(\omega')]_{M-3}^{M-2}]
\end{aligned}$$

where $\lim_{N\to\infty} \beta_{M-3,M-2}^{M-2,i_{M-2},N}(\omega') = 0$ and we apply the usual separation technique, the details of which we omit. We then obtain a set Ω'_{M-3} such that $\mathbb{Q}'(\Omega'_{M-3}) = 1$ and for $\omega' \in \Omega'_{M-3}$

$$\begin{aligned}
&|a_{M-3}^N(S,\omega') - a_{M-3}(S,\omega')| \\
&\le c_{M-3}^N(\omega') + \delta_{M-3}^N(S,\omega') \\
&+ \sum_{i_{M-1}=1}^{k_{M-1}} \alpha_{M-1,M}^{i_{M-1},N}(\omega') \sum_{i_{M-2}}^{k_{M-2}} \alpha_{M-2,M-1}^{i_{M-2},i_{M-1},N}(\omega') \sum_{i_{M-3}}^{k_{M-3}} \alpha_{M-3,M-2}^{i_{M-3},i_{M-2},i_{M-1},N}(\omega') \cdot \psi_{S,i_{M-3},i_{M-2},i_{M-1}}^{M-3,\alpha}(S) \\
&+ \sum_{i_{M-2}=1}^{k_{M-2}} \beta_{M-2,M-1}^{M-2,i_{M-2},N}(\omega') \sum_{i_{M-3}=1}^{m_{M-3}} \beta_{M-3,M-2}^{M-2,i_{M-2},i_{M-3},N}(\omega') \psi_{i_{M-3},i_{M-2}}^{M-2,\beta}(S)
\end{aligned}$$

which corresponds to $l = M-2$ and the above random variables satisfy the necessary conditions.

We now suppose the claim is true for $n = m+1$ where $m+1 \le M-3$ and show it holds for

$n = m$. Again, using Lemma 2 for $\omega' \in \Omega'_m$ we have

$$
\begin{aligned}
&|a_m^N(S,\omega') - a_m(S,\omega')| \\
&\leq \delta_m^N(S,\omega') + \widetilde{c}\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}\left[\,|a_{m+1}^N(S_{m+1},\omega') - a_{m+1}(S_{m+1},\omega')| \mid S_m = S, [v^j(\omega')]_m^{m+1}\,\right]
\end{aligned}
$$

applying our induction hypothesis to the summation in the second term, we find

$$
\begin{aligned}
&\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}\left[\,|a_{m+1}^N(S_{m+1},\omega') - a_{m+1}(S_{m+1},\omega')| \mid S_m = S, [v^j(\omega')]_m^{m+1}\,\right] \\
&\leq c_{m+1}^N(\omega')\varepsilon + \frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}[\delta_{m+1}^N(S_{m+1},\omega') \mid S_m = S, [v^j(\omega')]_m^{m+1}] \qquad (2.22) \\
&+ \frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}[\alpha_{m+1}^N(S_{m+1},\omega') \mid S_m = S, [v^j(\omega')]_k^{m+1}] \qquad (2.23) \\
&+ \sum_{l=m+2}^{M-2}\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}[\beta_{l,m}^N(S_{m+1},\omega') \mid S_m = S, [v^j(\omega')]_m^{m+1}] \qquad (2.24)
\end{aligned}
$$

and focus on each average separately. For line (2.22), again, we apply Lemma 7 to obtain an ε-separation, followed by the usual separation technique and find a set $\Omega_m^{'\delta}$ such that $\mathbb{Q}'(\Omega_m^{'\delta}) = 1$ and

$$
\begin{aligned}
&\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}[\delta_{m+1}^N(S_{m+1},\omega') \mid S_m = S, [v^j(\omega')]_m^{m+1}\ ] \\
&\leq c\,\varepsilon + \sum_{i_{m+1}=1}^{k_{m+1}}\beta_{m+1,m+2}^{m+1,i_{m+1},N}(\omega')\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}[\psi_{i_{m+1}}(S_{m+1}) \mid S_m = S, [v^j(\omega')]_m^{m+1}] \\
&\leq c_{\delta,m}^N(\omega')\,\varepsilon + \sum_{i_{m+1}=1}^{k_{m+1}}\beta_{m+1,m+2}^{m+1,i_{m+1},N}(\omega')\sum_{i_m=1}^{k_m}\beta_{m,m+1}^{m+1,i_m,i_{m+1},N}(\omega')\psi_{i_m,i_{m+1}}^{m+1,\beta}(S) \qquad (2.25)
\end{aligned}
$$

where $\lim_{N\to\infty}\beta_{m+1,m+2}^{m+1,i_{m+1},N}(\omega') = 0$, and $\limsup_{N\to\infty} c_{m+1}^N(\omega')$ only depends on our truncation and separability conditions along with the other usual conditions. We now label the second term in (2.25) as $\beta_{m+1,m}^N$. Then we turn to line (2.23) and find

$$
\begin{aligned}
&\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}[\alpha_{m+1}^N(S_{m+1},\omega') \mid S_m = S, [v^j(\omega')]_m^{m+1}] \\
&= \sum_{i_{M-1}=1}^{k_{M-1}}\alpha_{M-1,M}^{i_{M-1},N}(\omega')\sum_{i_{M-2}=1}^{k_{M-2}}\alpha_{M-2,M-1}^{i_{M-2},i_{M-1},N}(\omega')\ \ldots \\
&\sum_{i_{m+1}=1}^{k_{m+1}}\alpha_{i_{m+1},i_{m+2}}^{i_{m+1},\ldots,i_{M-1},N}(\omega')\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}[\psi_{i_{m+1},\ldots,i_{M-1}}^{m+1,\alpha}(S_{m+1}) \mid S_m = S, [v^j(\omega')]_m^{m+1}]
\end{aligned}
$$

which may be handled using the usual separation technique to obtain a set Ω'_m such that $\mathbb{Q}'(\Omega'_m) = 1$ and

$$\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}[\alpha^N_{m+1}(S_{m+1},\omega') \mid S_m = S, [v^j(\omega')]_m^{m+1}]$$
$$\leq c^N_{\alpha,m}(\omega')\varepsilon + \sum_{i_{M-1}=1}^{k_{M-1}} \alpha^{i_{M-1},N}_{M-1,M}(\omega') \sum_{i_{M-2}=1}^{k_{M-2}} \alpha^{i_{M-2},i_{M-1},N}_{M-2,M-1}(\omega') \ \cdots \sum_{i_m=1}^{k_m} \alpha^{i_m,\ldots,i_{M-1},N}_{i_m,i_{m+1}}(\omega')\psi^{m,\alpha}_{i_m,i_{m+1}}(S).$$

Next we turn to line (2.24) and analyze each term of the form $\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}[\beta^N_{l,m}(S_{m+1},\omega') \mid S_m = S, [v^j(\omega')]_m^{m+1}]$ for $l = m+2,\ldots,M-2$. Again, applying the usual techniques, we obtain a set $\Omega'^{\beta}_{l,m}$ with $\mathbb{Q}'(\Omega'^{\beta}_{l,m}) = 1$ such that

$$\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}[\beta^N_{l,m+1}(S_{m+1},\omega') \mid S_m = S, [v^j(\omega')]_m^{m+1}]$$
$$= \sum_{i_l=1}^{p_l} \beta^{l,i_l,N}_{l-1,l}(\omega') \sum_{i_{l-1}=1}^{p_{l-1}} \beta^{l,i_l,i_{l-1},N}_{l-2,l-1}(\omega') \ \cdots$$
$$\sum_{i_{m+1}=1}^{p_{m+1}} \beta^{l,i_l,\ldots,i_m,N}_{m+1,m+2}(\omega')\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}^{\mathbb{Q}}[\psi^{m+1,\beta}_{i_l,\ldots,i_{m+1}}(S_{m+1}) \mid S_m = S, [v^j(\omega')]_m^{m+1}]$$
$$\leq c^N_{\beta,m}(\omega')\varepsilon + \sum_{i_l=1}^{p_l} \beta^{l,i_l,N}_{l-1,l}(\omega') \sum_{i_{l-1}=1}^{p_{l-1}} \beta^{l,i_l,i_{l-1},N}_{l-2,l-1}(\omega') \ \cdots \sum_{i_m=1}^{p_m} \beta^{l,i_l,\ldots,i_m,N}_{m,m+1}(\omega')\psi^{m,\beta}_{i_l,\ldots,i_m}(S)$$

with the usual conditions being satisfied. We then set

$$\beta^N_{l,m}(S,\omega') = \sum_{i_l=1}^{p_l} \beta^{l,i_l,N}_{l-1,l}(\omega') \sum_{i_{l-1}=1}^{p_{l-1}} \beta^{l,i_l,i_{l-1},N}_{l-2,l-1}(\omega') \ \cdots \sum_{i_m=1}^{p_m} \beta^{l,i_l,\ldots,i_m,N}_{m,m+1}(\omega')\psi^{m,\beta}_{i_l,\ldots,i_m}(S)$$

for each $l = m+2,\ldots,M-2$. Upon collecting all terms of the form c^N_m and $\beta^N_{l,m}$, for $l = m+1,\ldots,M-2$ and intersecting our sets of full measure, we obtain the result for $n = m$ which implies the result.

□

Now that we are finished proving Theorem 4 we turn to showing Corollary 5 holds.

Corollary 5. *Let $S \in \mathcal{S}$ and $n \in \{1,...,M-1\}$, then we have* $\lim_{N\to\infty}\mathbb{E}^{\mathbb{Q}'}\left[\ \left|a^N_n(S) - a_n(S)\right|\ \right] = 0.$

Proof of Corollary 5. We prove the result for $n \in \{1,...,M-3\}$, the cases where $n = M-1, M-2$ are, of course, simpler. Let $\varepsilon > 0$ and recall (2.19). We begin with the expression

$$\alpha^N_n(S) = \sum_{i_{M-1}=1}^{k_{M-1}} \alpha^{i_{M-1},N}_{M-1,M} \sum_{i_{M-2}=1}^{k_{M-2}} \alpha^{i_{M-2},i_{M-1},N}_{M-2,M-1} \ \cdots \sum_{i_n=1}^{k_n} \alpha^{i_n,\ldots,i_{M-1},N}_{i_n,i_{n+1}} \cdot \psi^{n,\alpha}_{i_n,\ldots,i_{M-1}}(S)$$

and note that, by (2.17), for $k \leq M-2$, each random variable of the form $\alpha_{k,k+1}^{i,N}$ is uniformly bounded over N and the $\psi_i^{n,\alpha}$ are bounded as well. Here the superscript i represents an indexing tuple. As well, recalling equation (2.16), the boundedness of $[A_{M-1}^N]^{-1}$ and the SLLN imply that $\alpha_{M-1,M}^{i_{M-1},N} \to 0$ in L^1. Thus we conclude $\alpha_n^N(S) \to 0$ in L^1. Next, we turn to the expressions $\beta_{l,n}^N(S)$ where

$$\beta_{l,n}^N(S) = \sum_{i_l=1}^{m_l} \beta_{l-1,l}^{l,i_l,N} \sum_{i_{l-1}=1}^{m_{l-1}} \beta_{l-2,l-1}^{l,i_l,i_{l-1},N} \cdots \sum_{i_n=1}^{m_n} \beta_{n,n+1}^{l,i_l,\ldots,i_n,N} \cdot \psi_{i_l,\ldots,i_n}^{n,\beta}(S)$$

and note again, for $k \leq l-2$, the random variables $\beta_{k,k+1}^{l,i,N}$ are uniformly bounded over N and $\psi_i^{n,\beta}$ are bounded functions. Also, just as with $\alpha_{M-1,M}^{i_{M-1},N}$, we have that $\beta_{l-1,l}^{l,i_l,N} \to 0$ in L^1. Thus, we conclude each $\beta_{l,n}^N \to 0$ in L^1.

The same reasoning shows that $\delta_n^N(S) \to 0$ in L^1 as well.

Finally we turn to c_n^N and note that $c_n^N = c' + \widetilde{c}_n^N$ where c' depends on our truncation and separability conditions and

$$|\widetilde{c}_n^N| \leq C \left(\sum_{i_{M-1}}^{k_{M-1}} \alpha_{M-1,M}^{i_{M-1},N} + \sum_{l=n+1}^{k_n} \sum_{i_l=1}^{m_l} \beta_{l-1,l}^{l,i_l,N} \right)$$

for some $C > 0$. As each $\alpha_{M-1,M}^{i_{M-1},N}, \beta_{l-1,l}^{l,i_l,N} \to 0$ in L^1, this completes the proof.

□

The next Corollary, which we state without proof due to its simplicitiy, states that our continuation functions converge almost surely and in L^1. This implies that our direct estimator also converges in each sense.

Corollary 6. *Let $S \in \mathcal{S}, v \in \mathbb{R}^{d_v}$ and $n \in \{0, 1, ..., M-1\}$, then $\lim_{N\to\infty} C_{n,d}^N(S,v) = C_n(S,v)$ $\mathbb{Q}'$ a.s. and $\lim_{N\to\infty} \mathbb{E}^{\mathbb{Q}'}[C_{n,d}^N(S,v)] = C_n(S,v)$.*

2.5 A $1d+1d$ Example: Heston Model

We consider a Bermudan put option written on the Heston model [33] as the typical type of problem the GPSM is designed to handle. Due to its low dimensionality, well established deterministic methods may be applied which form a reference to check the GPSM's performance. The low dimensionality also allows us to fully visualize the exercise regions produced by the hybrid algorithm which may be compared to deterministic implementations.

2.5.1 Model Description and Framework

We suppose the existence of a probability space $(\Omega, \mathcal{F}, \mathbb{Q})$, just as in Section 2.2, such that $\mathbb{Q}$ is a risk neutral measure. On this space, we have the following process (S_t, v_t) defined as

$$dS_t = S_t(r\ dt + \sqrt{v_t}\ dW_t^S)\,, \qquad \text{and} \qquad dv_t = \kappa(\theta - v_t)\ dt + \gamma\sqrt{v_t}\ dW_t^v$$

where $d[W^S, W^v]_t = \rho\ dt$. Writing $W_t^S = \rho\ W_t^v + \sqrt{1-\rho^2}\ W_t^{\perp,v}$, where $W^{\perp,v}$ is a Brownian motion independent of W^v, we have the model falls into the class described in Section 2.2. Next, let $h(S) = (K-S)_+$ be the payoff of a put option and suppose we have a collection of dates $\{t_0 = 0, \ldots, t_M = T\}$ where the holder of the option may exercise. We refer to K and T as the option's strike and maturity, respectively.

2.5.2 Derivation of Pricing Formulas

To compute $\mathbb{E}\left[e^{-r\Delta t_n} f(S_{n+1}, v_{n+1}) \mid S_n = s_i, v_n = v\right]$ we begin by iterating the total expectation and writing our expression in terms of $X_t = \log S_t$ where

$$dX_t = \left(r - \tfrac{1}{2}v_t\right)\ dt + \sqrt{1-\rho^2}\sqrt{v_t}\ dW_t^{\perp,v} + \rho\sqrt{v_t}\ dW_t^v\,.$$

We then consider a payoff of the form

$$g(X, v) = f(\exp(X), v)\,,$$

so that

$$\mathbb{E}\left[e^{-r\Delta t_n} f(S_{n+1}, v_{n+1}) \mid S_n, v_n\right] = e^{-r\Delta t_n}\mathbb{E}\left[\mathbb{E}\left[g(X_{n+1}, v_{n+1}) \mid [v]_n^{n+1}, X_n\right] \mid X_n, v_n\right]\,. \quad (2.26)$$

Deriving the Conditional PDE

Focusing on the inner expectation, treating $[v]_n^{n+1}$ as a deterministic path on $[t_n, t_{n+1}]$, and representing it as a function of time v_t, we then write

$$\begin{aligned} &X_t = Y_t + Z_t\,, \quad \text{where} \\ &Y_t = X_{t_n} + \int_{t_n}^t \left(r - \tfrac{1}{2}v_s\right)\ ds + \int_{t_n}^t \sqrt{1-\rho^2}\sqrt{v_s} dW_s^{v,\perp}\,, \quad \text{and} \quad Z_t = \int_{t_n}^t \rho\sqrt{v_s} dW_s^v\,. \end{aligned}$$

Re-writing the inner expectation in the rhs of (2.26) in terms of the process Y_t we have

$$\mathbb{E}\left[g(X_{n+1}, v_{n+1}) \mid [v]_n^{n+1}, X_n\right] = \mathbb{E}\left[g(Y_{n+1} + Z_{n+1}, v_{n+1}) \mid [v]_n^{n+1}, Y_n\right] := u(t, Y_n).$$

By the Feynman-Kac theorem, the function $u(t,y)$ can be written as the solution to the following PDE

$$\begin{cases} \partial_t u(t,y) + a_t\ \partial_y u(t,y) + b_t\ \partial_y^2 u(t,y) = 0\,, \\ \qquad\qquad\qquad\qquad\qquad u(t_{n+1}, y) = g(y + Z_{t_{n+1}}, v_{t_{n+1}})\,, \end{cases} \quad (2.27)$$

where $a_t = r - \frac{1}{2}v_t$ and $b_t = (1-\rho^2)v_t$ are deterministic functions of time.

From here, we may also identify our finite dimensional path statistic Λ as

$$\Lambda([v]_n^{n+1}) = \left(\int_{t_n}^{t_{n+1}} v_s\ ds, \int_{t_n}^{t_{n+1}} \sqrt{v_s}\ dW_s^v\right)$$

which is interesting for theoretical purposes and will be further utilized in Chapter 3.

FFT Based Solution

Taking the Fourier transform of (2.27), as in the FST method of [59], and letting $\widehat{u}(t,\omega)$ denote the Fourier transform of $u(t,y)$ in the second argument, we have

$$\begin{cases} \partial_t \widehat{u}(t,\omega) + (a_t i\omega - \frac{1}{2} b_t \omega^2)\widehat{u}(t,\omega) = 0\,, \\ \widehat{u}(t_{n+1},\omega) = e^{i\omega Z_{t_{n+1}}}\ \widehat{g}(\omega, v_{n+1})\,, \end{cases}$$

which is an ODE wrt time for $\widehat{u}$. Thus,

$$\widehat{u}(t_n,\omega) = \widehat{u}(t_{n+1},\omega)\exp\left\{ i\omega \int_{t_n}^{t_{n+1}} a_s ds - \frac{1}{2}\omega^2 \int_{t_n}^{t_{n+1}} b_s ds \right\}.$$

Writing $\Psi(\omega, t_n, t_{n+1}) = i\omega Z_{t_{n+1}} + i\omega \int_{t_n}^{t_{n+1}} a_s\ ds - \frac{1}{2}\omega^2 \int_{t_n}^{t_{n+1}} b_s\ ds$ or more compactly $\Psi_{n,n+1} := \Psi(\cdot, t_n, t_{n+1})$, and using the FST's discretization methodology with fast Fourier transforms (FFT), we have the recursion

$$u_n = \mathrm{FFT}^{-1}\Big[\mathrm{FFT}\Big[g_{n+1}\Big]\exp(\Psi_{n,n+1})\Big]\,,$$

where g_k, u_k are discretizations of $g(t_k,y), u(t_k,y)$.

2.5.3 Numerical Experiments, Results, and Discussion

In this section we carry out various tests to check the performance of the GPSM in comparison to both LSM and a finite difference implementation which we use as a reference.

For each set of model parameters that we test, we carry out the following procedure for LSM, GPSM, and finite differences.

Procedure for LSM

We run N_{trial} of an LSM implementation as follows.

1. Simulate N_{sims} paths of (S,v) on $[0,T]$ with Euler discretization $1/N_{step}$.

2. Carry out the high estimate of the TV approach to LSM using the $N_{lsm}^{S,v}$ simulation of (S,v) and a two state-variable basis with dimension L_{lsm}.

   - Retain the estimated coefficients.

3. Independently simulate $N_{lsm}^{S,v}$ paths of (S,v) on $[0,T]$ with Euler discretization $1/N_{step}$.

4. Carry out the low estimate of the TV approach to LSM using the simulation of (S,v) and the coefficients from the previous stages.

For the hybrid algorithm, we use a monomial basis $\{x^l\}_{l=0}^{deg_{hyb}}$ and define $L_{hyb} = deg_{hyb} + 1$ to be the total number of basis functions used.

Procedure for the GPSM

We run N_{trial} of the GPSM as follows.

1. Simulate N^v_{gpsm} paths of $(v, \Delta W^v)$ on $[0, T]$ with Euler discretization $1/N_{step}$.
2. Carry out the direct estimate for GPSM with $N^{\mathcal{S}} \in \{N^{\mathcal{S}}_1, N^{\mathcal{S}}_2, N^{\mathcal{S}}_3\}$ points in the log-space grid. Use a one state-variable basis of dimension L_{mix}.
   - Retain the estimated coefficients for each resolution level.
3. Independently simulate N^v_{gpsm} paths of $(v, \Delta W^v)$ on $[0, T]$ with Euler discretization $1/N_{step}$.
4. Carry out the low estimate version of the GPSM in the same manner using estimated coefficients from the previous stage at each resolution.

For LSM, our basis consists of functions of all functions of the form $\phi_{j,k}(S, v) = S^j v^k$ and $\phi_i(S) = (K - S)^i_+$ such that $j + k + i \leq deg_{lsm}$. The quantity L_{lsm} denotes the total dimension of the basis.

Details for the Finite Difference Method

We employ an explicit finite difference scheme with equally spaced grid points. Our boundary conditions are as outlined in [36]. Let N_s and N_v be the number of points in our spatial and volatility grids, N_t be the number of time steps taken, and $S_{min}, S_{max}, v_{min}, v_{max}$ be the end points of our grids. These parameters have been found to be stable in determining the time-zero prices and exercise boundary. We use cubic spline interpolation to obtain the time zero prices.

Settings for Trials

We carry out three main sets of tests that vary the maturity and number of exercise dates. Our finite difference parameters are as in Table 2.2. Our model parameters are fixed across trials and

Trial Type	Maturity (yrs)	No. Exercise Dates
1	0.25	10
2	1.00	12
3	2.50	30

Table 2.1: Settings for the three main trials. Subtrial settings will be presented in Table 2.4. All exercise opportunities are equally spaced out.

Trial	N_s	N_v	N_t	v_{min}	v_{max}	S_{min}	S_{max}
1	2^9	2^7	10^5	0	1.00	0	20
2	2^9	2^7	10^6	0	1.25	0	53
3	2^9	2^7	10^6	0	1.50	0	112

Table 2.2: parameters used in of our finite difference computation.

S_0	v_0	κ	θ	γ	ρ	K	r
10	0.15	5	0.16	0.9	0.1	10	0.02

Table 2.3: Model parameters used across trials.

N_{trials}	$(\log S/S_0)_{min}$	$(\log S/S_0)_{max}$
100	-3	3

Subtrial Type	N_{sims}^{hyb}	deg_{hyb}	L_{hyb}	N_{sims}^{lsm}	deg_{lsm}	L_{lsm}
a	$1.0 \cdot 10^4$	3	4	$1.0 \cdot 10^5$	3	13
b	$2.5 \cdot 10^4$	4	5	$2.5 \cdot 10^5$	4	19
c	$5.0 \cdot 10^4$	5	6	$5.0 \cdot 10^5$	5	28

Table 2.4: Simulation and regression parameters used across subtrials for LSM and GPSM.

listed in Table 2.3. These parameters are similar to [36] except that we set r to be 2% as opposed to 10% which is more in line with current financial circumstances. Within each trial, we carry out three subtrials where we increase the number of paths and basis functions according to Table 2.4.

Numerical Results

We refer the reader to Appendices 2.B and 2.C where we collect all pricing statistics and optimal exercise boundaries (OEBs) respectively.

Discussion of Results

As indicated in Table 2.5 to Table 2.7, the LSM is capable of computing the time zero value for only two digits when simulating 100 000 to 500 000 paths. As expected, the standard deviation is fairly low given the relatively high number of paths, however, a bias persists in both the direct and low estimator from the inability to accurately locate the exercise boundaries as seen in Figure 2.3. While the boundary is accurate for the final exercise date, the middle dates are fairly inaccurate while the early dates are completely inaccurate. Our plot in Figure 2.3 corresponds to Trial 2(c) and is meant to showcase the LSM's best case scenario which took, on average, about 48 seconds for the direct estimator, and 38 seconds for the low estimator.

In Tables 2.9 to 2.10 we see that the GPSM is able to match the reference price to 3 digits when $N_S = 2^7$ and to 4 digits when $N_S = 2^9$. In Table 2.8, however, one is able to match 2 digits when $N_S = 2^7$ and 3 digits when $N_S = 2^9$. This shows that the GPSM is slightly less accurate for earlier maturities. Also, the level of variance at 10 000 simulated paths is comparable to that of LSM with 100 000 paths. This is of course owed to the conditioning technique of our algorithm. Interestingly, while the variance decreases as the number of paths increase from 10 000 to 50 000, we see that the bias is unaffected and is mostly a function of our grid resolution. One thing we note here is that the direct and low estimators that we produce are identical for up until 3 or

4 digits with comparable levels of accuracy. We conclude that our low estimator is better than that of LSMC, and that the GPSM's regression component introduces very little bias due to the accuracy of the direct estimator.

In Figure 2.4, we show the OEB for the GPSM for Trial 2(a) with $N_S = 2^9$ and should be compared to the OEB for LSM pictured in Figure 2.3. We see the GPSM is highly accurate in locating the exercise boundary across exercise dates, with only a slight error on the interface of the exercise and holding regions. Beyond the variance reduction obtained by solving conditional PDEs, we can attribute this accuracy to our regression method as discussed in Section 2.3.2 and our simplification of how the boundary is located. Although we don't show it, this discrepancy is further diminished in Trial 2(c) where we used 50 000 simulations. We show the result of Trial 2(a) to indicate that the worst-case run for GPSM is considerably better than the best-case run for LSM.

In Section 3.2, we will use the low-biased estimation procedure as a means to compute

$$\frac{\partial V_0^N}{\partial v}(S_0, v_0)$$

as opposed to a redundant estimate of $V_0^N(S_0, v_0)$.

We use a Fourier based approach due to its numerical simplicity and speed. The approach of separating X into components Y, Z is also employed in [16]. It is worth noting that the FST is likely not the most efficient known approach as it requires a large uniform grid and is used solely for its simplicity. It seems one may be able to solve the conditional PDEs more efficiently using the method outlined in [21].

This $1d+1d$ example is used to show how LSM, GPSM and deterministic methods compare in terms of computing time-zero prices and determining the exercise boundaries. It is worth noting that a deterministic method is likely the most efficient way to solve this problem, either using PDE methods as described in [36] or quadrature based approach in [22]. Later on, in Section 3.4, we apply the GPSM to a $2d + 2d$ analogue in which deterministic approaches will not apply.

2.A Pseudo-code Description

Algorithm 1 GPSM: Exercise Boundaries and Direct Estimate

Simulate N paths of v_t
for $n = M : 2$ **do**
  if $n == M$ **then**
    Set $V_n(s_i, v) = h_n(s_i)$ ▷ For $i = 1, ..., N_s^{d_S}$
  end if ▷ Initialize boundary for PDE solver
  for $j = 1 : N$ **do**
    Compute $\mathbb{E}\left[V_n(S_n, v_n^j) \mid S_{n-1} = s_i, [v^j]_{n-1}^n\right]$ ▷ For $i = 1, ..., N_s^{d_S}$
  end for
  for $i = 1 : N_s^{d_S}$ **do**
    Regress $\{ e^{-r\Delta t}\mathbb{E}\left[V_n(S_n, v_n^j) \mid S_{n-1} = s_i, [v^j]_{n-1}^n\right] \}_{j=1}^N$ onto $\{\phi_l(v)\}_{l=1}^L$ ▷ Obtain $[a_n(s_i)]_{i=1}^{N_s^{d_S}}$
    Set $C_{n-1}(s_i, v) = a_{n-1}(s_i) \cdot \phi(v)$
  end for ▷ Obtain a matrix of dimension $N_s^{d_S} \times L$
  Set $V_{n-1}(s_i, v) = \max(h_{n-1}(s_i), C_{n-1}(s_i, v))$ ▷ For $i = 1, ..., N_s^{d_S}$
end for
for $j = 1 : N$ **do**
  Compute $e^{-r\Delta t}\mathbb{E}\left[V_1(S_1, v_1^j) \mid S_0 = s_i, [v^j]_0^1\right]$ ▷ For $i = 1, ..., N_s^{d_S}$
end for
Set $V_0^h(s_i, v_0) = \max\left(h_0(s_i), \frac{1}{N}\sum_{i=1}^N e^{-r\Delta t}\mathbb{E}\left[V_1(S_1, v_1^i) \mid S_0 = s_i, [v^i]_0^1\right]\right)$ ▷ For $i = 1, ..., N_s^{d_S}$ ▷ High estimate of the time-zero prices
return $V_{h,0}(s_i, v_0)$ for $i = 1, ..., N_s^{d_S}$

Algorithm 2 GPSM: Lower Price

Simulate $N_{v,2}$ paths of v_t
for $j = 1, ..., N_{v,2}$ **do**
  for $n = M : 2$ **do**
    if $n == M$ **then**
      Set $V_n^j(s_i, v_M^j) = h_n(s_i)$ ▷ For $i = 1, ..., N_s^{d_S}$
    end if
    Compute $U_{n-1}^j(s_i) = e^{-r\Delta t}\mathbb{E}[V_n(S_n, v_n^j) \mid S_{n-1} = s_i, [v^i]_{n-1}^n]$
    Set $V_{n-1}^j(s_i) = U_{n-1}^j(s_i) I_{\Gamma_n}(s_i, v_{n-1}^j) + h(s_i) I_{\Gamma_n^c}(s_i, v_{n-1}^j)$ ▷ For $i = 1, ..., N_s^{d_S}$
  end for
  Set $V_{l,0}^j(s_i) = e^{-r\Delta t}\mathbb{E}[V_1(S_1, v_1^j) \mid S_0 = s_i, [v^i]_{n-1}^n]$ ▷ For $i = 1, ..., N_s^{d_S}$
end for
Set $V_{l,0}(s_i) = \frac{1}{N_{v,2}}\sum_{j=1}^{N_{v,2}} V_{l,0}^j(s_i)$ ▷ For $i = 1, ..., N_s^{d_S}$
return $V_{l,0}(s_i, v_0)$ for $i = 1, ..., N_s^{d_S}$

2.B Pricing Statistics

Subtrial Type	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	Direct	0.7390	$2.560 \cdot 10^{-3}$	5.66
	Low	0.7381	$2.833 \cdot 10^{-3}$	5.17
b	Direct	0.7399	$2.293 \cdot 10^{-3}$	19.27
	Low	0.7392	$1.822 \cdot 10^{-3}$	17.13
c	Direct	0.7399	$1.325 \cdot 10^{-3}$	40.26
	Low	0.7395	$1.199 \cdot 10^{-3}$	35.60

Table 2.5: Numerical Results for LSM Trial 1. $T = 0.25$ with 10 equally spaced exercise dates. The reference value is 0.7416

Subtrial Type	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	Direct	1.4481	$5.5530 \cdot 10^{-3}$	6.68
	Low	1.4473	$5.2402 \cdot 10^{-3}$	5.27
b	Direct	1.4483	$3.3610 \cdot 10^{-3}$	19.96
	Low	1.4477	$3.1173 \cdot 10^{-3}$	17.42
c	Direct	1.4494	$2.0117 \cdot 10^{-3}$	48.36
	Low	1.4487	$2.3288 \cdot 10^{-3}$	38.38

Table 2.6: Numerical Results for LSM Trial 2. $T = 1$ with 12 equally spaced exercise dates. The reference value is 1.4528.

Subtrial Type	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	Direct	2.1998	$7.6403 \cdot 10^{-3}$	7.76
	Low	2.1979	$7.9606 \cdot 10^{-3}$	5.23
b	Direct	2.2008	$4.5664 \cdot 10^{-3}$	26.63
	Low	2.1988	$4.2684 \cdot 10^{-3}$	19.25
c	Direct	2.2025	$3.1398 \cdot 10^{-3}$	72.27
	Low	2.2013	$3.2480 \cdot 10^{-3}$	43.24

Table 2.7: Numerical Results for LSM Trial 3. $T = 2.5$ with 30 equally spaced exercise dates. The reference value is 2.2111.

Subtrial Type	N_S	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	2^7	Direct	0.7438	$9.5027 \cdot 10^{-4}$	4.64
		Low	0.7436	$8.6641 \cdot 10^{-4}$	4.81
	2^8	Direct	0.7423	$9.5351 \cdot 10^{-4}$	5.83
		Low	0.7421	$8.7197 \cdot 10^{-4}$	4.88
	2^9	Direct	0.7419	$9.5426 \cdot 10^{-4}$	8.11
		Low	0.7417	$8.7139 \cdot 10^{-4}$	6.69
b	2^7	Direct	0.7435	$5.5137 \cdot 10^{-4}$	11.79
		Low	0.7436	$5.8161 \cdot 10^{-4}$	9.07
	2^8	Direct	0.7420	$5.5341 \cdot 10^{-4}$	15.09
		Low	0.7421	$5.8552 \cdot 10^{-4}$	12.01
	2^9	Direct	0.7416	$5.5387 \cdot 10^{-4}$	20.54
		Low	0.7417	$5.8625 \cdot 10^{-4}$	17.37
c	2^7	Direct	0.7437	$4.0581 \cdot 10^{-4}$	23.35
		Low	0.7435	$3.9400 \cdot 10^{-4}$	18.75
	2^8	Direct	0.7422	$4.0731 \cdot 10^{-4}$	30.48
		Low	0.7420	$3.9555 \cdot 10^{-4}$	24.89
	2^9	Direct	0.7418	$4.0765 \cdot 10^{-4}$	41.16
		Low	0.7417	$3.9583 \cdot 10^{-4}$	34.95

Table 2.8: Numerical Results for GPSM Trial 1. $T = 0.25$ with 10 equally spaced exercise dates. The reference value is 0.7416.

Subtrial Type	N_S	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	2^7	Direct	1.4539	$1.3654 \cdot 10^{-3}$	5.23
		Low	1.4537	$1.3923 \cdot 10^{-3}$	4.24
	2^8	Direct	1.4532	$1.3667 \cdot 10^{-3}$	6.82
		Low	1.4530	$1.3921 \cdot 10^{-3}$	5.74
	2^9	Direct	1.4530	$1.3670 \cdot 10^{-3}$	9.59
		Low	1.4528	$1.3934 \cdot 10^{-3}$	8.31
b	2^7	Direct	1.4541	$8.9978 \cdot 10^{-4}$	13.71
		Low	1.4539	$9.6078 \cdot 10^{-4}$	10.95
	2^8	Direct	1.4534	$9.0053 \cdot 10^{-4}$	18.46
		Low	1.4532	$9.6214 \cdot 10^{-4}$	14.43
	2^9	Direct	1.4532	$9.0072 \cdot 10^{-4}$	24.87
		Low	1.4530	$9.6251 \cdot 10^{-4}$	21.28
c	2^7	Direct	1.4539	$6.3971 \cdot 10^{-4}$	27.34
		Low	1.4537	$5.8203 \cdot 10^{-4}$	21.74
	2^8	Direct	1.4532	$6.4043 \cdot 10^{-4}$	35.96
		Low	1.4530	$5.8366 \cdot 10^{-4}$	28.63
	2^9	Direct	1.4530	$6.4055 \cdot 10^{-4}$	49.61
		Low	1.4529	$5.8330 \cdot 10^{-4}$	41.88

Table 2.9: Numerical Results for GPSM Trial 2. $T = 1$ with 12 equally spaced exercise dates. The reference value is 1.4528.

Subtrial Type	N_S	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	2^7	Direct	2.2117	$1.1953 \cdot 10^{-3}$	12.64
		Low	2.2118	$1.2116 \cdot 10^{-3}$	9.68
	2^8	Direct	2.2114	$1.1960 \cdot 10^{-3}$	16.53
		Low	2.2114	$1.2124 \cdot 10^{-3}$	13.53
	2^9	Direct	2.2113	$1.1962 \cdot 10^{-3}$	23.88
		Low	2.2113	$1.2128 \cdot 10^{-3}$	19.71
b	2^7	Direct	2.2119	$5.9960 \cdot 10^{-4}$	33.05
		Low	2.2119	$8.1975 \cdot 10^{-4}$	24.87
	2^8	Direct	2.2116	$6.0000 \cdot 10^{-4}$	42.85
		Low	2.2116	$8.2126 \cdot 10^{-4}$	34.17
	2^9	Direct	2.2115	$6.0001 \cdot 10^{-4}$	60.73
		Low	2.2114	$8.2071 \cdot 10^{-4}$	49.25
c	2^7	Direct	2.2119	$5.2746 \cdot 10^{-4}$	65.04
		Low	2.2118	$5.4734 \cdot 10^{-4}$	51.14
	2^8	Direct	2.2115	$5.2777 \cdot 10^{-4}$	86.93
		Low	2.2115	$5.4745 \cdot 10^{-4}$	68.37
	2^9	Direct	2.2114	$5.2787 \cdot 10^{-4}$	120.18
		Low	2.2114	$5.4781 \cdot 10^{-4}$	101.94

Table 2.10: Numerical Results for GPSM Trial 3. $T = 2.5$ with 30 equally spaced exercise dates. The reference value is 2.2111.

2.C Optimal Exercise Boundaries

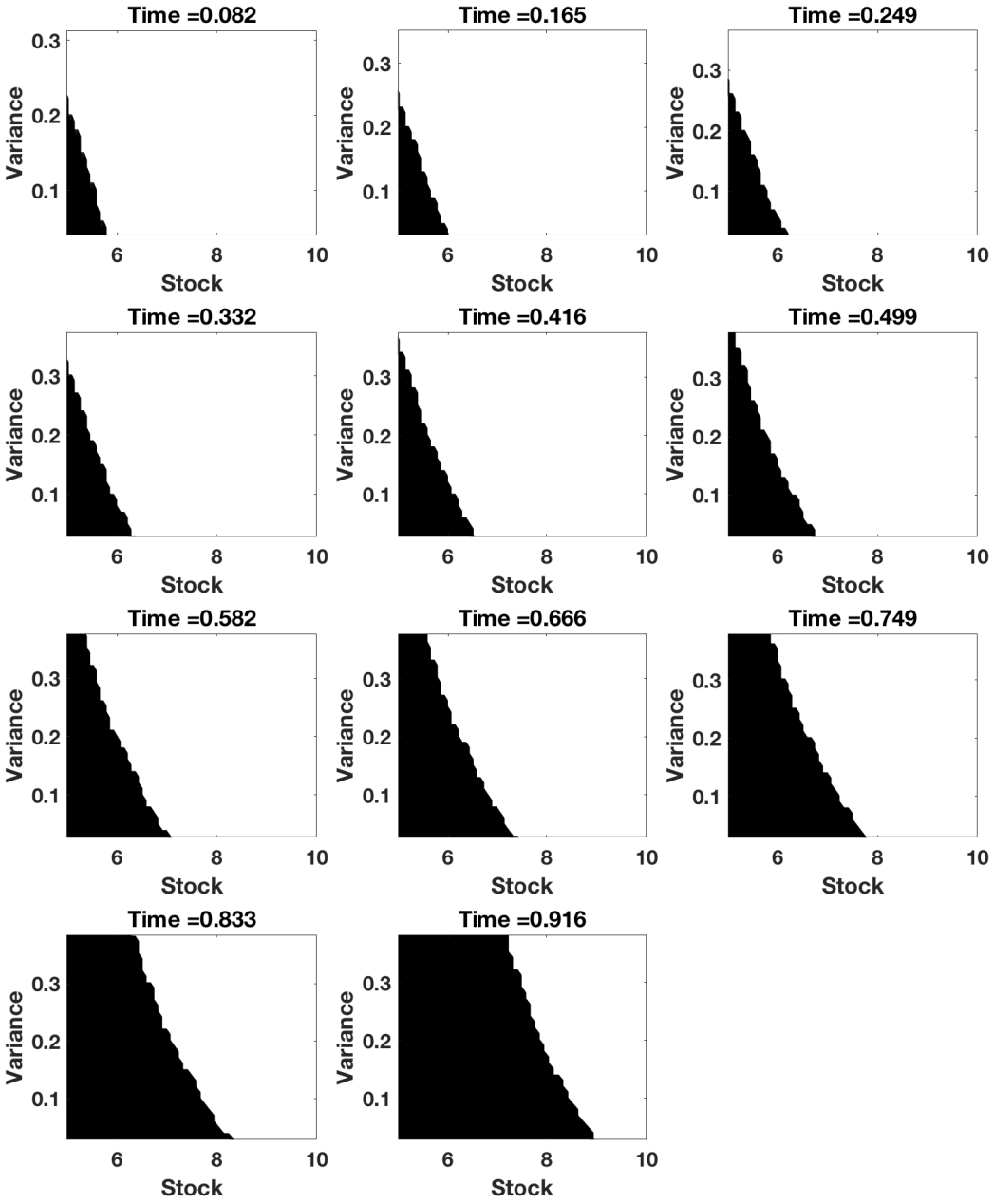

Figure 2.2: Reference optimal exercise boundaries generated by a finite difference scheme. Black indicates exercise, white indicates hold.

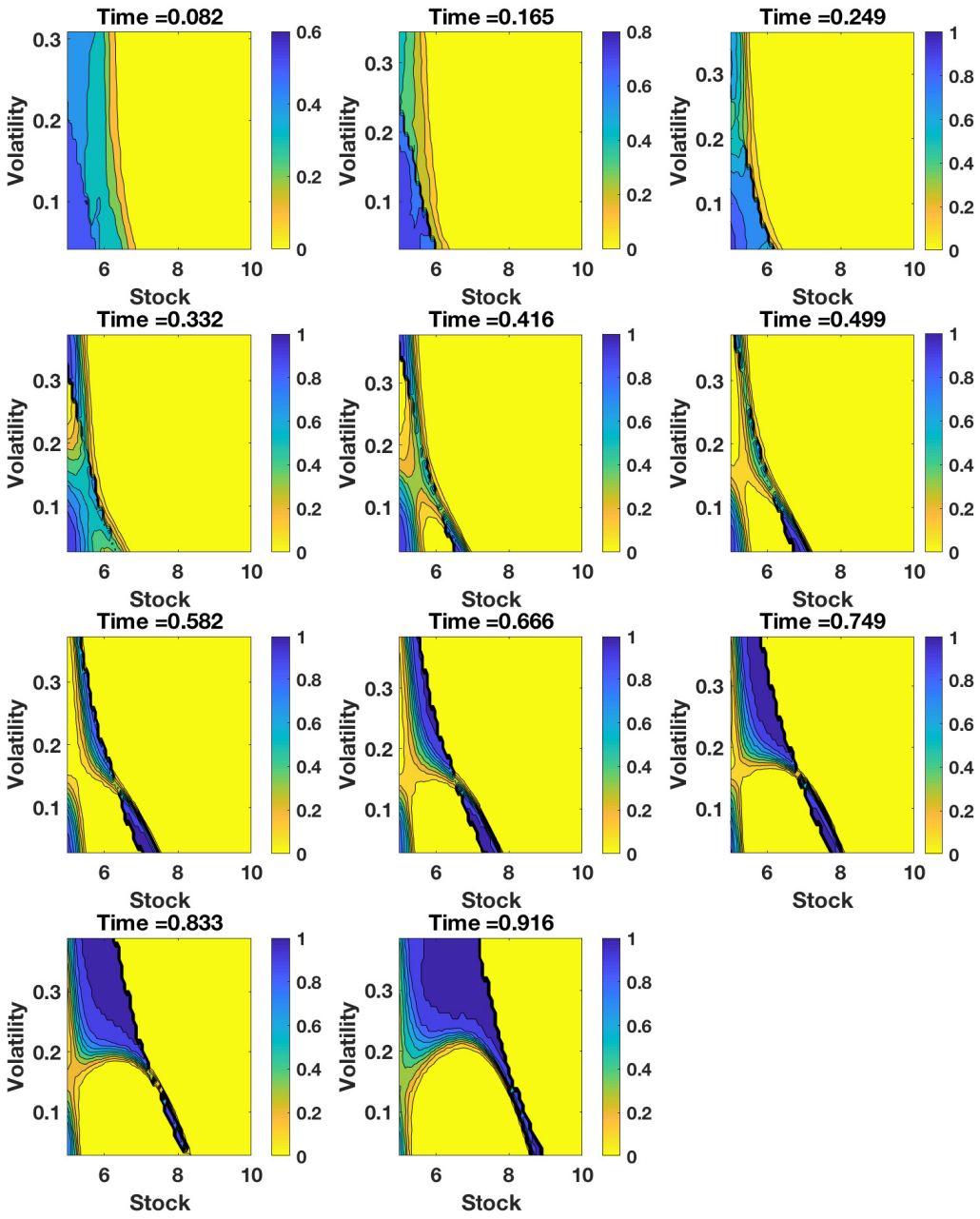

Figure 2.3: Difference optimal exercise boundaries generated by LSM Trial 2(c) compared to the reference. Dark blue indicates incorrectness with probability 1, yellow indicates correctness with probability 1.

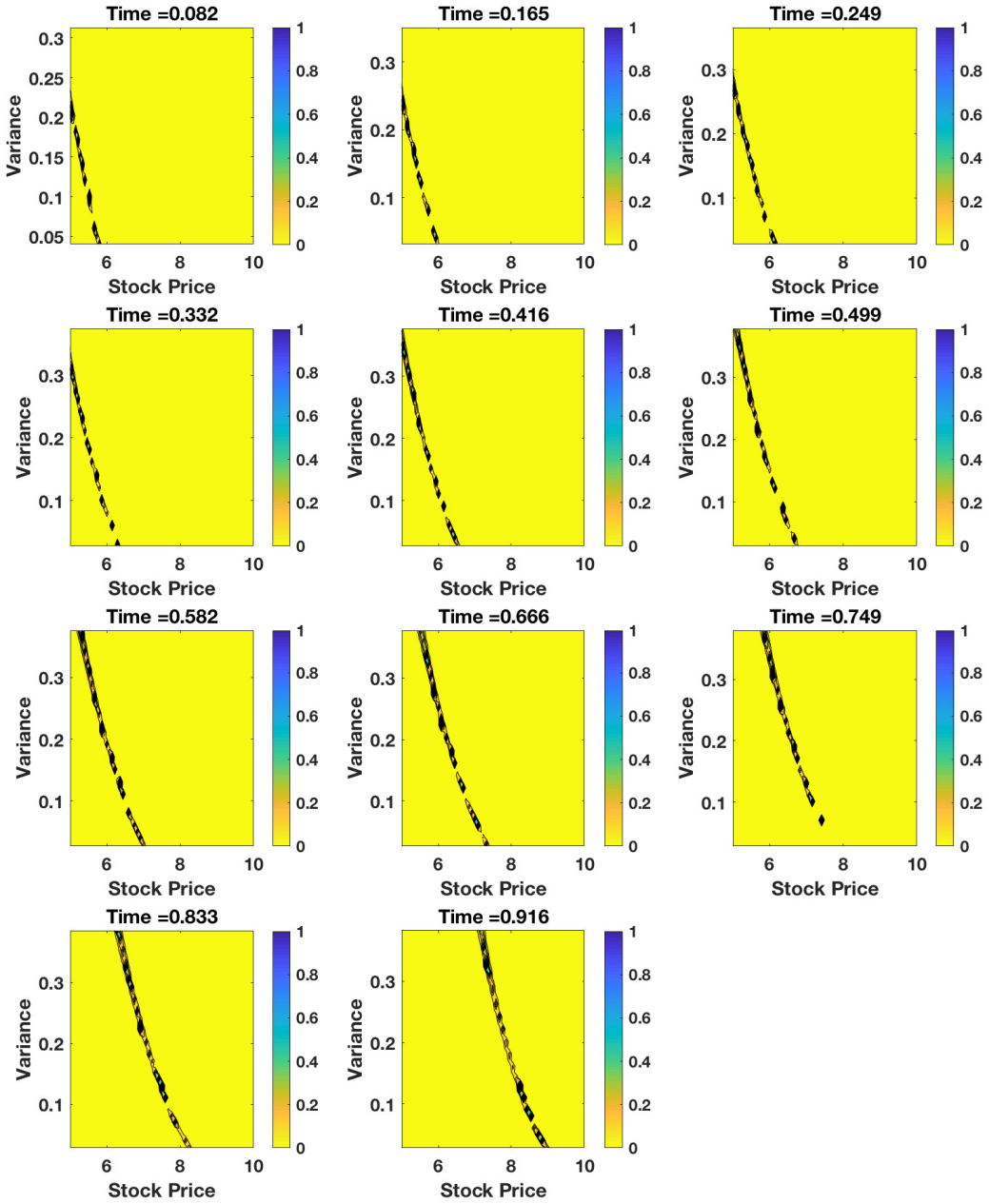

Figure 2.4: Difference optimal exercise boundaries generated by GPSM Trial 2(a), $N_S = 2^9$, compared to the reference. Dark blue indicates incorrectness with probability 1, yellow indicates correctness with probability 1.

Chapter 3

Complexity Reduction Methods and Sensitivities

In this chapter we address secondary issues surrounding the hybrid algorithm such as reducing the complexity and runtimes along with the computation of sensitivities.

As mentioned in Chapter 2, if $d_S > 1$ and one does not have a closed form solution, solving several thousand conditional PDEs at the required resolution can be computationally demanding. To improve matters, we introduce two approximation methods to speed up our computations. The first method is a clustering approach which allows us to reduce the number of paths for which we solve conditional PDEs. The second is a multi-level Monte Carlo/multi-grid method that allows us to solve the majority of our conditional PDEs at a coarse resolution and relatively few conditional PDEs at the required fine resolution. We also show how the two methods may be combined to further reduce the complexity.

Computing the value function's sensitivities to underlying state variables is an important topic in real world applications, especially mathematical finance. When a financial institution writes a Bermudan style option, they will first value it using the hybrid algorithm and afterwards hedge their position using the option's sensitivities. Given their importance, we describe a technique to compute derivatives of our value function with respect to our state variables, S and v. For S-sensitivities we apply a gridded finite difference approach, whereas for v-sensitivities we apply a common MC method that uses initial dispersions.

After describing our complexity reduction approximations and sensitivity techniques, we return to the $1d + 1d$ Heston model to check their effectiveness. We also compute sensitivities using standard LSMC and compare them to the estimates obtained by the hybrid algorithm. Finally, we apply our hybrid algorithm to the multi-dimensional Heston model, which is a $2d + 2d$ example, and compare the results to standard LSMC.

3.1 Complexity Reduction

3.1.1 Clustering

As described in Section 2.2, we suppose our model satisfies the following Markov-like property

$$\mathbb{E}\left[\ \phi(v_{t_n})h(S_{t_{n+1}}, v_{t_{n+1}})\ \middle|\ S_{t_n} = S\ ,\ \mathcal{F}^{W^v}_{t_n,t_{n+1}}\ \right] = \mathbb{E}\left[\ \phi(v_{t_n})h(S_{t_{n+1}}, v_{t_{n+1}})\ \middle|\ S_{t_n} = S\ ,\ \Theta_n([v])\ \right]$$

where

$$\Theta_n([v]) = (v_{t_n}, \Lambda_n([v]), v_{t_{n+1}})$$

takes values in some $\mathbb{R}^{d_\Theta}$. This property indicates that the numerical solutions to conditional expectations do not depend on every point of the path $[v]$, which is an infinite dimensional object, but rather on a finite-dimensional statistic, $\Theta_n([v])$. With this observation, if we simulate two paths of v, $[v^1]$ and $[v^2]$, and have $\Theta_n([v^1]) \approx \Theta_n([v^2])$ then we may achieve computational savings by only solving a PDE over one of the paths, or perhaps over

$$\Theta^*_n = \tfrac{1}{2}(\Theta_n([v^1]) + \Theta_n([v^2])),$$

and use the resulting numerical solution as a representative for both paths. More generally, given a time stage $[t_n, t_{n+1}]$, suppose we simulate N paths $\{[v^j]\}_{j=1}^N$ over which we must solve conditional PDEs and compute coefficients $a^N_n(S)$. Instead of solving a PDE over each path, we first compute $\{\Theta^j_n([v])\}_{j=1}^N$ and cluster them into L_n clusters of size $L_{n,k}$. Given the kth cluster, we obtain a representative $\Theta^*_{n,k}$ as

$$\Theta^*_{n,k} = (v^*_n, \Lambda^*_n([v]), v^*_{n+1}) = \frac{1}{L_{n,k}} \sum_{j=1}^{L_{n,k}} \Theta_n([v^{k,j}])$$

where $\{[v^{k,j}]\}_{j=1}^{L_{n,k}}$ are the paths within the kth cluster. Finally we compute the coefficient $a^N_n(S)$ via the modified formula

$$a^N_n(S) \approx [A^N_n]^{-1} \frac{1}{N} \sum_{k=1}^{L_n} L_{N,k}\ \phi(v^{k,*}_n)\ \mathbb{E}^{\mathbb{Q}}\left[f^N_{n+1}(S_{n+1}, v^{k,*}_{n+1})\ \middle|\ S_n = S, \Theta^*_{n,k}\right] \tag{3.1}$$

where the matrix A^N_n is constructed using the unclustered paths $\{[v^k]\}_{j=1}^N$. There are numerous approaches and software packages available for clustering the paths such as k-means, Gaussian mixture models, or hierarchical clustering. As we typically need to cluster about 10^4 paths into a few thousand clusters, the most practical method is agglomerative hierarchical clustering. 'Top down' methods struggle with a large number of clusters, whereas agglomerative hierarchical clustering is only sensitive to the number of paths and not the number of clusters.

Agglomerative clustering produces a hierarchical tree with successive levels joining together clusters from lower levels based on their proximity. The method begins by joining individual paths, observed as $\Theta([v])$, into pairs. The next step is to join pairs of paths that are closest to each other and to continue this process until the data is consolidated into a single cluster. To define

proximity, given two path statistics $\Theta_n([v^{(1)}]), \Theta_n([v^{(2)}])$ we set

$$d(\Theta_n([v^{(1)}]), \Theta_n([v^{(2)}])) = |\Theta_n([v^{(1)}]) - \Theta_n([v^{(2)}])|$$

and for clusters $C^{(1)}, C^{(2)}$

$$d(C^{(1)}, C^{(2)}) = \max \left\{ d(\Theta_n([v^{(1)}]), \Theta_n([v^{(2)}])) \;\; | \;\; \Theta_n([v^{(i)}]) \in C^{(i)} \right\}$$

Given a hierarchical tree as in Figure 3.1, there are at least two approaches to selecting the

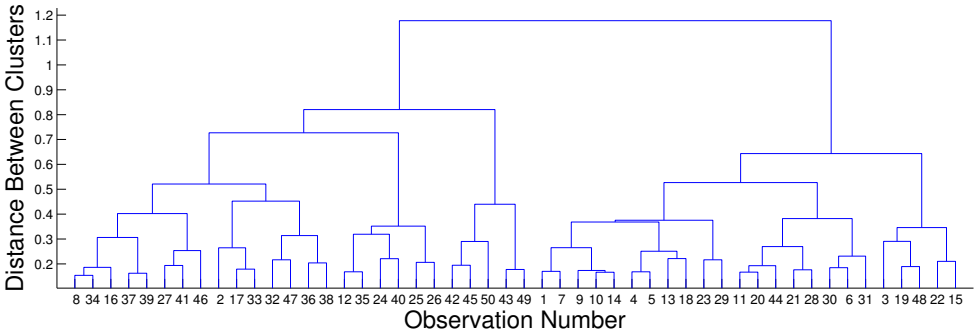

Figure 3.1: An example of a hierarchical tree produced for 100 observations of $\Theta_n([v])$ for the $1d + 1d$ Heston model of Section 2.5. In practice one typically has at least 10^4 observations.

clusters.

1. Select a number, N_{cl}, of clusters and make a branch cut at the lowest height such that there are at most N_{cl} clusters. Unfortunately, the value N_{cl} must be chosen based on user-experience. As one lowers N_{cl} the quality of their cluster representative depreciates and our experiments in this thesis show that one can typically reduce the number of paths by approximately a half.

2. Select a cutoff number c, such that if a node in the tree has *inconsistency* value less than c, take all observations corresponding to this node as a cluster. The inconsistency value measures the height of a node compared to the average height of all other nodes in its level. A higher inconsistency value indicates that a node and its observations are less similar to all other nodes and their observations

Approach (1) is more convenient to implement than (2), however, the adaptive nature of (2) makes it more efficient across time intervals. In this thesis, we use Approach (1) and leave Approach (2) for future study. Our use of clustering is applied in obtaining the direct estimate and not in the computation of the low biased estimates. The low biased estimator takes the following form

$$V_{l,0}(S, v_0) = \frac{1}{N} \sum_{j=1}^{N} P_0^j(S)$$

where P_0^j is the solution obtained from the low estimator. As $P_0^j(S)$ depends on $\{\Theta_n([v^j])\}_{n=0}^{M-1}$, where M is the number of time intervals, applying our clustering adjustment would involve clustering N observations in $\mathbb{R}^{Md_\Theta}$ which is prohibitively high dimensional.

3.1.2 Multi-Level Monte Carlo (mlMC)/Multi-Grids

Overview of mlMC

Suppose we wish to compute $\mathbb{E}^{\mathbb{Q}'}[X]$ where X is a random variable that is computationally expensive to simulate. If we know of another random variable Y that closely approximates X and is less expensive to simulate, we may write

$$\begin{aligned}\mathbb{E}^{\mathbb{Q}'}[X] &= \mathbb{E}^{\mathbb{Q}'}[Y] + \mathbb{E}^{\mathbb{Q}'}[X-Y] \\ &\approx \frac{1}{N_0}\sum_{j=1}^{N_0} Y_0^{(j)} + \frac{1}{N_1}\sum_{j=1}^{N_1}\left(X_1^{(j)} - Y_1^{(j)}\right)\end{aligned}$$

where $N_0 \gg N_1$ and the samples in each batch are independent of each other. If X and Y have similar variances and high positive correlation, then the variance of the second summation will be low and N_1 does not need to be made large. To further extend this notion, given a sequence of random variables $\{X_k\}_{k=0}^K$, with X_k less expensive to simulate than X_{k+1}, we may similarly write

$$\mathbb{E}^{\mathbb{Q}'}[X_K] \approx \frac{1}{N_0}\sum_{j=1}^{N_0} X_0^{(j)} + \sum_{k=1}^{K}\frac{1}{N_k}\sum_{j=1}^{N_k}\left(X_k^{(j)} - X_{k-1}^{(j)}\right) \tag{3.2}$$

where $N_k > N_{k+1}$ and $N_0 \gg N_k$ for $k \geq 1$ and the batches are independent of each other.

In [26], this idea is developed and numerous types of applications and examples are presented. One of the more interesting applications of mlMC is the numerical solution of stochastic PDEs (SPDEs). In such examples one considers X_l to be the numerical solution of a PDE, conditional on a random sample, with grid resolution an increasing function of l. In this case, X_l has a vector output and one must interpolate the PDE grid for (3.2) to hold. As seen in (2.11), on each time interval, $[t_n, t_{n+1}]$, the estimated coefficients we obtain may be viewed as the expectation of the solution of an SPDE generated by our finite-dimensional statistics, $\Theta_n([v])$, multiplied by $\phi(v_{t_n})$. With this observation we proceed to apply mlMC to improve the complexity in computing our estimated coefficients.

mlMC for Computing the Estimated Coefficients

Assume we have K independent sets of simulations of v_t on $[0,T]$, denoted $\{(v_t^{(j,k)})_{j=1}^{\infty}\}_{k=0}^K$ with N_k^v paths where $N_k^v > N_{k+1}^v$. Also, let $P_n^{(j,k)}(S)$ denote the numerical solution of the conditional PDE on the jth path with grid resolutions $N_k^{\mathcal{S}}$, in each dimension, over $[t_n, t_{n+1}]$ where $N_k^{\mathcal{S}} < N_{k+1}^{\mathcal{S}}$.

Applying mlMC as outlined in (3.2), we may write

$$a_n^N(S) \approx \left[A_n^{N_0}\right]^{-1}\left[\frac{1}{N_0^v}\sum_{j=1}^{N_0}\phi(v_n^{j,0})\cdot P_n^{(j,0)}(S) + \sum_{k=1}^{K}\frac{1}{N_k}\sum_{j=1}^{N_k^v}\phi(v_n^{j,k})\cdot\left(P_n^{(j,k)}(S) - P_n^{(j,k-1)}(S)\right)\right] \tag{3.3}$$

where each of $P_n^{(j,k)}(S)$ is interpolated to have a resolution that matches level K.

As mentioned before, the grids in (3.3) must be interpolated to the highest level in order for the expression to be well defined. Below we describe an efficient scheme that is independent of the

number of simulated paths.

I1. Instead of interpolating $P_n^{(j,k+1)}(S)$ and $P_n^{(j,k)}(S)$ for each $j \geq 1$ directly to a level K resolution, it is far more efficient to first compute

$$\frac{1}{N_k^v}\sum_{j=1}^{N_k^v}\phi(v_n^{j,k})P_n^{(j,k+1)}(S) \text{ and } \frac{1}{N_k^v}\sum_{j=1}^{N_k^v}\phi(v_n^{j,k})P_n^{(j,k)}(S),$$

interpolate to level K, and then calculate their difference.

I2. When one has carried out the computation over $[t_{n+1}, t_{n+2}]$ and turns to the interval $[t_n, t_{n+1}]$ their terminal condition is given by the expression

$$f_{n+1}^N(S_{n+1}, v_{n+1}^j) = \max(h_{n+1}(S_{n+1}), a_{n+1}^N(S_{n+1}) \cdot \phi(v_{n+1}^j))$$

where $a_{n+1}^N(S)$ is of level K resolution. When computing $P_n^{(j,k)}(S)$ for $k < K$ one must interpolate $f_{n+1}^N(S, v_{n+1}^j)$ to a level k resolution. To avoid having to do this for each $j \in \{1, \ldots, N_k\}$ one may pre-interpolate $a_{n+1}^N(S)$ to a level k resolution and use it to compute $f_{n+1}^N(S_{n+1}, v_{n+1}^j)$ and eventually $\{P_n^{(j,k)}(S)\}_{j=1}^{N_k}$.

mlMC for Low Estimates of the Time Zero Price

We again carry out K independent simulations of v_t on $[0, T]$, denoted as $\{(v_t^{(k)})\}_{k=0}^K$, with N_k^v paths where $N_k^v > N_{k+1}^v$. Letting $P_0^{(j,k)}(S)$ denote the numerical solution of the conditional PDE on the jth path with grid resolution $N_k^{\mathcal{S}}$ over $[0, T]$ where $N_k^{\mathcal{S}} < N_{k+1}^{\mathcal{S}}$ we have

$$V_{l,0}(S, v_0) \approx \frac{1}{N_0^v}\sum_{j=1}^{N_0^v}P_0^{(j,0)}(S) + \sum_{k=0}^{K-1}\frac{1}{N_k^v}\sum_{j=1}^{N_k^v}(P_n^{(j,k+1)}(S) - P_n^{(j,k)}(S))$$

where again we interpolate the lower resolution grids to match the highest resolution grid. In this case, we make use of (I1) but do not require (I2). We do not provide a pseudo-code description for this part as it is relatively simple.

A Comment On Optimized Path Allocations and Number of Levels

An important question in the application of mlMC is how to choose the number of paths N_k^v that are allocated to the kth level. In the case of a single period problem, it is relatively easy to derive the optimal allocation using Lagrange multipliers [26]. In our setting, the main computation is the generation of our OEBs and direct estimates of the time-zero value. As this is a multi-period computation carried out over a grid of dimension $L \times N_{\mathcal{S}}^{d_S}$, the approach in [26] does not appear to carry over. We leave this interesting and important question of how to optimally allocate paths for the direct estimator for future study. It is possible, however, to apply the traditional allocation scheme to the low estimator, although we do not implement it here.

3.1.3 Combining Clustering and mlMC

Clustering and mlMC may be combined to further improve the algorithm's complexity.

On a time increment $[t_n, t_{n+1}]$, when computing the coefficients in equation (2.11), we begin with the approximation (3.3). Next, since typically $N_0^v \gg N_k^v$ for $k = 1, ..., K$, we cluster the paths of the 0-level simulation of v_t and apply approximation (3.1) to compute $\frac{1}{N_0^v}\sum_{j=1}^{N_0^v} \phi(v_n^{j,0}) \cdot P_n^{j,0}(S)$.

We refer to the new algorithm with clustering and mlMC as LSMC-PDE-PRML (path reduced, multi-level), as named in [23], or prml-GPSM. To better understand the effects of clustering and mlMC, suppose at each time stage one must solve $10,000$ 2-dimensional PDEs at a grid resolution of 2^9. If we apply the prml-GPSM, we may set $N_0^v = 10,000$, $N_1^v = 100$, $N_{cl} = 4,500$, $N_0^{\mathcal{S}} = 2^6$ and $N_1^{\mathcal{S}} = 2^9$. Assuming the complexity and run time of the numerical PDE solver is proportional to the number of points in the PDE grid, using a two level approximation we may find

$$\frac{\text{Complexity}_0}{\text{Complexity}_{prml}} \approx \frac{10000 \cdot 2^9 \times 2^9}{4500 \cdot 2^6 \times 2^6 + 100 \cdot (2^6 \times 2^6 + 2^9 \times 2^9)} \approx 58$$

As a result, the prml algorithm may decrease the complexity by a factor of 58 times for this particular example. This back of the envelope calculation ignores the costs of interpolations and clustering which tend to be outweighed by PDE solving costs, at least for higher dimensional problems.

3.2 Computing Sensitivities

In this section we discuss how one can compute the sensitivities of the value function, $V_n^N(S, v)$, with respect to the variables S and v for value functions whose continuation surface is twice differentiable in S and once in v. As V_n^N is computed via a mix of numerical PDE methods and MC, one may expect a hybrid approach for computing sensitivities as well. As we shall see, sensitivities with respect to S are handled from a numerical PDE perspective, whereas sensitivities with respect to v, are computed using familiar MC methods. Mixed sensitivities take a hybrid approach between the two, as we will see.

From the MC perspective there are a number of well known approaches for computing sensitivities for expected values, many of which may be extended to value functions arising from optimal stopping problems. These methods include bumping, pathwise differentiation and likelihood ratios and are discussed in [27]. The method of bumping is computationally expensive and is typically used as a last resort. Pathwise differentiation is based on interchanging derivatives and expectations and sometimes requires smoothing to avoid singularities [7]. The method of likelihood ratios requires one to know the density of their model in closed form which presents challenges. When computing sensitivities with respect to v, we adopt the methodology of [64] which is a model-free approach that requires a slight modification of the GPSM but ultimately adds little extra computation. While the associated variances tend to be relatively high, the extensive benchmarking presented in [64] shows that they are usually on the same order of magnitude as the pathwise differentiation and likelihood ratio methods.

From the PDE perspective, a natural approach for computing sensitivities is, upon solving the

PDE numerically, to use finite differences or interpolating splines on the PDE grid. This approach is appealing as it requires essentially zero extra computation and tends to be very accurate. We use this approach for computing our sensitivities with respect to S.

3.2.1 Modifying the Low Estimator

Initial Dispersions and Limitations

Before discussing our methodology for taking derivatives of the value function, we discuss how the modified-LSM (MLSM) of [64] may be applied to the GPSM, leading to the modified GPSM (MGPSM). This alteration of the algorithm will allow us to compute sensitivities of the value function with respect to v at time zero when carrying out the low estimator.

As mentioned in Section 2.3, we select an initial point for $v(t)$, v_0, and have our N simulations stem from this initial point. This provides an estimate for the time zero continuation value

$$V_{l,0}^N(S, v_0) = \mathbb{E}^{\mathbb{Q}}\left[e^{-r\tau_N} h(S_{\tau_N}) \mid S_0 = S, v_0 = v\right] \tag{3.4}$$

for all $S \in \mathcal{S}$ but only $v_0 \in \mathbb{R}^{d_v}$ with no information for $v \neq v_0$ where τ_N is determined from the direct estimator. To obtain sensitivities with respect to v, at v_0, we must modify our simulation to stem from a neighbourhood around v_0 which is commonly referred to as the *initial dispersion*, A.

For each $S \in \mathcal{S}$, we then obtain a collection of values $\{V_{0,l}^{j,N}(S, v_0^j)\}_{j=1}^N$ which we again regress onto a basis $\{\phi_l\}_{l=1}^L$ yielding coefficients

$$a_0^N(S) = [A_0^N]^{-1} \frac{1}{N} \sum_{j=1}^N \phi(v_0^j) V_{0,l}^{j,N}(S, v_0^j)$$

where $A_0^N = \frac{1}{N}\sum_{j=1}^N \phi(v_0^j)\phi(v_0^j)^T$. Finally we set

$$\begin{aligned} C_0^N(S, v) &= a_0^N(S) \cdot \phi(v) \quad \text{and} \\ V_{l,0}^N(S, v) &= \max(h_0(S), C_{l,0}^N(S, v)). \end{aligned}$$

As documented in [64], [42], [40], selecting an appropriate choice for A is a non-trivial task even in one dimension. The original work of [64] is focused on options written on one or more stocks, $S(t)$, following a geometric Brownian motion (GBM). In the one dimensional example, they suggest an initial dispersion generated via

$$S^j(0) = S(0)e^{\alpha\sigma\sqrt{T}Z^j}$$

where $Z^j \sim N(0,1)$, σ is the stock's volatility, T is the option's maturity, and α is a tunable parameter. One then sets $A = \{S^j(0)\}_{j=1}^N$ where N is the number of simulations. Intuitively, this initial dispersion mimics the true distribution of $S(t)$ although a proper theoretical justification for this choice remains missing from the literature and is beyond the scope of this thesis. To further complicate matters, when the dimension of S is two or more, the variances tend to increase

by at least an order of magnitude. To remedy the situation, [64] recommend computing multi-dimensional sensitivities one at a time. Although this approach may provide some improvements, it requires an extra computation for each sensitivity and does not assist with mixed sensitivities. Another issue documented in [42] is a variance-bias trade off that occurs in one dimensional problems when the spread, α, is varied. To obtain low biased estimates with low variance they recommend a "structured" approach which replaces estimating sensitivities based on (3.4) with

$$\mathbb{E}^{\mathbb{Q}}\left[e^{-r\Delta t}V_{t_1}(S_{t_1}, v_{t_1}) \mid S_0 = S, v_0 = v\right]$$

as both are available upon carrying out the direct estimator. The quality of this estimate relies on an accurate estimate of the regression coefficients at time t_1, which is never guaranteed, and its quality will diminish as the number of exercise dates increase as is shown in [40] for a similar type of computation. Thus, we proceed with the approach discussed in [64] without structuring.

Initial Dispersion for Mean Reverting Volatility Processes

In our setting we must select an initial dispersion for randomizing the initial value of $v(t)$ which is often a mean reverting process with level $\theta \in \mathbb{R}^{d_v}$. Following the approach of [64], we wish to mimic the distribution of $v(t)$, however, in our case, the exact distribution of $v(t)$ may not be known. As a result, we take an empirical approach as follows

1. Set $\theta = v(0)$
2. Simulate N paths of $v(t)$ on $[0, T^*]$
3. Set $A = \{v^j(T^*)\}_{j=1}^N$.

The final horizon T^* functions in a manner similar to the spread parameter α, and our backtests show that setting $T^* = 1$ provides reasonable results for most purposes.

3.2.2 Approaches for Computing Sensitivities

Due to the inherently high variances associated with computing sensitivities with respect to v, we offer two approaches for computing prices and sensitivities depending on which quantities the user requires. The main difference between the two is how many paths the user simulates. If one simulates N paths in approach (1) to obtain reasonable results, they typically need about $k \cdot N$ paths for approach (2) where k is roughly 5 or 10 or so. In what follows $\beta \in \{0, 1, 2\}^{d_S}$.

1. Computing the time-zero value and sensitivities with respect to S
   (a) Direct estimator
   - Uses the standard procedure and may be coupled with the prml additions.
   - Obtain $V_{0,d}^N(S, v)$ for $v = v_0$ and all $S \in \mathcal{S}$ and the OEBs.
   - Obtain estimates of

$$\frac{\partial^\beta V_{0,d}^N}{\partial S^\beta}(S, v_0)$$

(b) Low estimator

- Uses the standard procedure and may be coupled with mlMC additions.
- Obtain $V_{0,l}^N(S,v)$ for $v = v_0$ and all $S \in \mathcal{S}$.
- Obtain estimates of
$$\frac{\partial^\beta V_{0,l}^N}{\partial S^\beta}(S, v_0)$$
for all $S \in \mathcal{S}$

2. Computing the time-zero value, sensitivities with respect to S *and* v

(a) Direct estimator

- Uses the standard procedure and may be coupled with the prml additions.
- Obtain $V_{0,d}^N(S,v)$ for $v = v_0$ and all $S \in \mathcal{S}$ and the OEBs.
- Obtain estimates of
$$\frac{\partial^\beta V_{0,d}^N}{\partial S^\beta}(S, v_0)$$

(b) A modified low estimate of the time-zero surface using the methodology of [64]. This method involves randomizing the initial value of $v(t)$ with initial dispersion A as outlined in Section 3.2.1.

- May be coupled with mlMC as discussed before.
- May use a "structured" approach as discussed in [42]
- Obtain $V_{0,l}^N(S,v)$ for a small neighbourhood of v_0 and all $S \in \mathcal{S}$.
- Obtain an estimate of
$$\frac{\partial V_{0,l}^N}{\partial v}(S, v_0) \quad , \quad \frac{\partial V_{0,l}^N}{\partial S v}(S, v_0) \quad \text{and} \quad \frac{\partial^\beta V_{0,l}^N}{\partial S^\beta}(S, v_0)$$
for $S \in \mathcal{S}$.

3.2.3 Details for Computing Partial Derivatives

As our value function is determined using a max of two functions

$$V_n^N(S,v) = \max(h_n(S), C_n^N(S,v))$$

we have that V_n^N may not be smooth on $\mathcal{S} \times \mathbb{R}^{d_v}$. Given $(S,v) \in \mathcal{S} \times \mathbb{R}^{d_v}$ we must first check whether it is an interior point of either the holding or exercise region. Once we determine the region, we then differentiate the appropriate component of V_n^N. In what follows, we suppose we have detected (S_0, v_0) to be in the continuation region. As the approaches are relatively simple, we do not provide a pseudo-code description in Appendix 3.4.3.

Computing Sensitivities with Respect to S

We compute sensitivities with respect to S using the direct, low and modified low estimation procedures.

Direct and Low Estimator

By construction, our time-zero continuation value function is a grid $C^N_{0,j,v_0} := \{C^N_{0,j}(S_i, v_0)\}_{i=1}^{|\mathcal{S}|}$ where $|\mathcal{S}| = N_S^{d_S}$ and $j \in \{d, l\}$. We then construct a gridded-interpolant for C^N_{0,j,v_0}, denoted as $\overline{C}^N_{0,j}(S, v_0)$, and compute derivatives with respect to S via finite differences. As an example, if $d_S = 1$, we set

$$\frac{\partial V^N_{0,j}(S_0, v_0)}{\partial S} = \frac{\overline{C}^N_{0,j}(S + h, v_0) - \overline{C}^N_{0,j}(S - h, v_0)}{2h}. \tag{3.5}$$

Higher order and mixed derivatives may be computed via analogous finite difference formulas. In practice, a difference spacing of $h = 10^{-3}$ is sufficient.

Modified Low Estimator

By construction, our time zero continuation value is a function of our time-zero coefficient grid $\{a^N_0(S_i)\}_{i=1}^{|\mathcal{S}|}$. We obtain a grid for the continuation value $C^N_{0,l,v_0} := \{C^N_{0,l}(S_i, v_0)\}_{i=1}^{|\mathcal{S}|}$ where $C^N_{0,l} = a^N_0(S_i) \cdot \phi(v_0)$ and again construct a gridded-interpolant $\overline{C}^N_{0,l}(S, v_0)$. Finally, we take derivatives with respect to S as before.

Computing Sensitivities with Respect to v

Modified Low Estimator

For notational simplicity, we suppose $d_v = 1$. As $\phi(v)$ is known analytically, and is often a polynomial, we begin by computing $\frac{\partial \phi(v_0)}{\partial v}$. Next, we compute the grid $\{\partial_v C^N_{0,l}(S_i, v_0)\}_{i=1}^{|\mathcal{S}|}$ via the formula $\partial_v C^N_{0,l}(S_i, v_0) := a^N_0(S_i) \cdot \frac{\partial \phi(v_0)}{\partial v}$. Finally, to extend our computation for $S \notin \mathcal{S}$, we again generate a gridded-interpolant $\overline{\partial_v C}^N_{0,l,v_0}(S)$.

Computing Sensitivities with Respect to (S, v)

Modified Low Estimator

Following the procedure for computing sensitivities with respect to v, we obtain $\overline{\partial_v C}^N_{0,l,v_0}(S)$ which may be further differentiated with respect to S using finite differences as before.

3.3 Revisiting the $1d + 1d$ Heston Model

To test our complexity reduction and sensitivity computation methods, we re-consider the $1d+1d$ Heston model from Section 2.5.

3.3.1 Testing Clustering and mlMC

We test out the effects of clustering and mlMC, first separately, and then in a combined manner. We present our results with time horizons indicated in Table 2.1, repeated in Table 3.3.1, while working within Subtrial (a) from Table 2.4 in terms of number of basis functions and FFT spatial discretization. We keep our model parameters fixed as in Table 2.3 and set $N_{trial} = 100$ throughout. Our overall procedure follows the one listed in Section 2.5.3 except that our ml- and prml-implementations.

Trial Type	Maturity (yrs)	No. Exercise Dates
1	0.25	10
2	1.00	12
3	2.50	30

Table 3.1: Maturity and exercise dates for the three main trials that are used for testing clustering, mlMC, and combined clustering and mlMC.

Sub-trial Type	Number of Clusters
a	3 000
b	4 500
c	6 000

Table 3.2: Subtrial settings for the pr-GPSM indicating the number of clusters. We set $N_{sims} =$ 10 000 throughout.

3.3.2 Testing the Computation of Sensitivities

We now implement our methods for computing sensitivities for the MGPSM, prml-GPSM and MLSM. For the MLSM we apply an approach that is analogous to our sensitivities for MGPSM: we carry out the direct estimator as usual and obtain the boundary, afterwards we modify the low estimator with an initial dispersion constructed as in Section 3.2.1.

We work with the same parameters as in Table 2.3 and set $N_{trial} = 100$ throughout. In terms of time horizons and algorithm settings, for the MGPSM we work with the parameters of Trial 2 in Table 3.3.1, and Subtrial (a) of Table 2.4. For the MLSM we use the same settings as in Subtrial (a) of Table 2.4 as well. For the MGPSM approach, we keep $T^* = 1$ fixed throughout our tests and vary our grid resolutions as in Table 3.7. For MLSM, we set $T^* = 0.25$ which provides a good balance between bias and variance. For the MLSM, when applying the direct estimator, we include powers of our payoff function within our basis, however, as these functions are not differentiable, we do not include them to the modified low estimator when carrying out the time-zero regression. Thus, we set $L^d_{lsm} = 19$ and $L^l_{lsm} = 15$

In order to obtain a reference for the sensitivities, we use our finite difference scheme as outlined in Trial 2 of Table 2.2. Our sensitivities are obtained via generating a gridded-interpolant for the time-zero surface that is obtained. Afterwards, we differentiate the surface that we obtain.

3.3.3 Results and Discussion

In Appendix 3.B and 3.C we present the pricing statistics, sensitivities and OEBs for testing clustering, mlMC and sensitivities for the standard Heston model.

When viewing the results of clustering in Tables 3.9, 3.10, 3.11 in comparison to the standard GPSM in Table 2.9, we find relatively low amounts of added bias or variance. We also find that the variance is largely unaffected as one varies the number of clusters from 3 000 to 6 000. Our run times are higher as computing thousands of $1d$ FFTs is faster than clustering at each time stage. For problems where $d_S > 1$ the costs of solving the conditional PDEs will largely outweigh that

Sub-trial Type	N_1^S
a	2^7
b	2^8
c	2^9

Table 3.3: Subtrial settings for the ml-GPSM indicating the grid resolutions for the higher levels. We keep $N_0^S = 2^6, N_{sim,0}^v = 10\ 000, N_{sim,1}^v = 100$ fixed throughout subtrials.

Sub-trial Type	N_1^S
a	2^7
b	2^8
c	2^9

Table 3.4: Subtrial settings for the prml-GPSM indicating the grid resolutions for the higher levels. We keep $N_0^S = 2^6, N_{sim,0}^v = 10\ 000, N_{sim,1}^v = 100,\ N_{cl} = 4\ 500$ fixed throughout subtrials.

of clustering and so we will reap the advantages in those situations. When regarding our mlMC results in Tables 3.15, 3.16, 3.17 in comparison to the standard GPSM results in Table 2.9, we again see that the bias and variance of the algorithm is essentially unchanged. The same may be said of the prml version whose results are presented in Tables 3.9, 3.10, 3.11. Our run times for the Direct estimator are somewhat lower for mlMC due to the savings provided by carrying most of our FFTs at a resolution of 2^6. While there are some added interpolations in these computations, we find they do not affect the run times. Our OEBs for the prml version of the algorithm in Figure 3.3 show that the prml-GPSM is also able to closely approximate the true OEBs. Naturally, the OEBs for the pr-GPSM and ml-GPSM will be of even higher quality, and so we omit presenting their OEBs.

In Tables 3.18, 3.19, 3.20, 3.21 we present our results for computing sensitivities using finite differences, the MLSM, MGPSM and prml-MGPSM, respectively. First, from Tables 3.20, 3.21 we note that including the prml adjustments again has almost no effect on the sensitivities and their associated variances. Moreover, we see that the hybrid algorithm is highly accurate when compared to the finite difference solution in Table 3.18. Turning to the MLSM shown in Table 3.19, we see that, in comparison to Table 2.6 , our initial dispersion has added a slight bias to the price $V_{0,l}^N$ and interestingly made it closer to the reference value, although with some added variance; interestingly, the MGPSM has very little bias or variance added. In terms of the MLSM sensitivities, we see that they are mostly unbiased except for the computation of $\partial_{Sv} V_{0,l}^N$. In comparison to the MGPSM's sensitivities, however, they are an order of magnitude worse in terms of standard deviations. Another striking feature is the variance that is added as a consequence of using an initial dispersion to estimate prices and sensitivities in the modified low estimator. The fact that the GPSM is able to get low variance estimates of S-sensitivities, without an initial dispersion, using the direct estimator is a major advantages over the MLSM.

As alluded to before, our sensitivities with respect to v tend to be considerably worse than those with respect to S. Intuitively, there is a certain "stability" that arises when working with a

grid for computing S-sensitivities that is not present when computing v-sensitivities. As we shall see, being able to only determine $\partial_v V^N_{0,l}$ to one decimal place with a moderately high number of paths will be a recurrent issue. Interestingly, the variance of $\partial_{Sv} V^N_{0,l}$ decreases for both the MLSM and MGPSM, although the decrease is more significant for the MGPSM, possibly owing to the presence of the grid.

3.4 A $2d + 2d$ Example: Multi-dimensional Heston Model

In this section we consider the multidimensional Heston model which is an example of a high dimensional optimal stopping problem. The model is a natural and important extension of the standard Heston model that arises in financial applications.

3.4.1 Model Description and Framework

We work with the following version of the two stock price Heston model

$$\begin{aligned} dS^{(1)}_t &= S^{(1)}_t(rdt + \sqrt{v^{(1)}_t}dW^{(1)}_t)\,, & dv^{(1)}_t &= \kappa_1(\theta_1 - v^{(1)}_t)dt + \eta_1\sqrt{v^{(1)}_t}dW^{(3)}_t\,, \\ dS^{(2)}_t &= S^{(2)}_t(rdt + \sqrt{v^{(2)}_t}dW^{(2)}_t)\,, & dv^{(2)}_t &= \kappa_2(\theta_2 - v^{(2)}_t)dt + \eta_2\sqrt{v^{(2)}_t}dW^{(4)}_t \end{aligned}$$

where $(W^{(i)})^4_{i=1}$ is a 4-dimensional Brownian motion with full correlation structure $\rho = [\rho_{i,j}]$. Switching to log-space for our asset prices $X^{(i)}_t = \log S^{(i)}_t$, $i = 1, 2$, and applying the Cholesky decomposition to the matrix ρ, we obtain the following system:

$$\begin{aligned} dX^{(1)}_t &= (r - \tfrac{1}{2}v^{(1)}_t)dt + \sqrt{v^{(1)}_t}\sum_{j=1}^{4} a_{1,j}dB^{(j)}_t\,, \\ dX^{(2)}_t &= (r - \tfrac{1}{2}v^{(2)}_t)dt + \sqrt{v^{(2)}_t}\sum_{j=2}^{4} a_{2,j}dB^{(j)}_t\,, \\ dv^{(1)}_t &= \kappa_1(\theta_1 - v^{(1)}_t)dt + \eta_1\sqrt{v^{(1)}_t}\sum_{j=3}^{4} a_{3,j}dB^{(j)}_t\,, \\ dv^{(2)}_t &= \kappa_2(\theta_2 - v^{(2)}_t)dt + \eta_2\sqrt{v^{(2)}_t}a_{4,4}dB^{(4)}_t\,, \end{aligned}$$

where $(B^{(i)}_t)^4_{i=1}$ are independent Brownian motions and $[a_{i,j}]$ is an upper triangular matrix satisfying the usual conditions with $a_{4,4} = 1$, and $\rho = aa^\intercal$.

3.4.2 Derivation of Pricing Formulas

We follow the same approach as in Section 2.5, to compute $\mathbb{E}[e^{-r\Delta t_n} f(S_{n+1}, v_{n+1}) \mid S_n = S, v_n = v]$ where $S_t = (S^{(1)}_t, S^{(2)}_t)$, $v_t = (v^{(1)}_t, v^{(2)}_t)$. Again, writing $g(X, v) = f(\exp(X), v)$ we have

$$\mathbb{E}\left[f(S_{n+1}, v_{n+1}) \mid S_n, v_n\right] = \mathbb{E}\left[e^{-r\Delta t}\mathbb{E}\left[g(X_{n+1}, v_{n+1}) \mid [v]^{n+1}_n, X_n\right] \mid X_n, v_n\right]. \tag{3.6}$$

Deriving the Conditional PDE

Focusing on the inner expectation, and treating $[v]_n^{n+1} = ([v^{(1)}]_n^{n+1}, [v^{(2)}]_n^{n+1})$ as a deterministic path in $\mathbb{R}^2$, leads to the following decompositions

$$\begin{aligned}
X_t^{(1)} &= Y_t^{(1)} + Z_t^{(1)}\,, \\
Y_t^{(1)} &= X_{t_n}^{(1)} + \int_{t_n}^t (r - \tfrac{1}{2} v_s^{(1)}) ds + \int_{t_n}^t \sqrt{v_t^{(1)}} \sum_{j=1}^{2} a_{1,j} dB_t^{(j)}\,, \\
Z_t^{(1)} &= \int_{t_n}^t \sqrt{v_t^{(1)}} \sum_{j=3}^{4} a_{1,j} dB_t^{(j)}\,,
\end{aligned}$$

and similarly

$$\begin{aligned}
X_t^{(2)} &= Y_t^{(2)} + Z_t^{(2)}\,, \\
Y_t^{(2)} &= X_{t_n}^{(2)} + \int_{t_n}^t (r - \tfrac{1}{2} v_s^{(2)}) ds + \int_{t_n}^t \sqrt{v_t^{(2)}} a_{2,2} dB_t^{(2)}\,, \\
Z_t^{(2)} &= \int_{t_n}^t \sqrt{v_t^{(2)}} \sum_{j=3}^{4} a_{2,j} dB_t^{(j)}.
\end{aligned}$$

Writing the inner expectation on the rhs of (3.6) in terms of $(Y_t^{(1)}, Y_t^{(2)})$ leads to

$$\begin{aligned}
&\mathbb{E}\left[g(X_{n+1}^{(1)}, X_{n+1}^{(2)}, v_{n+1}^{(1)}, v_{n+1}^{(2)}) \mid [v]_n^{n+1}, X_n \right] \\
&\quad = \mathbb{E}\left[g(Y_{n+1}^{(1)} + Z_{n+1}^{(1)}, Y_{n+1}^{(2)} + Z_{n+1}^{(2)}, v_{n+1}^{(1)}, v_{n+1}^{(2)}) \mid [v]_n^{n+1}, Y_n = y \right]. =: u(t,y)
\end{aligned}$$

By the Feynman-Kac theorem, the function $u(t, y_1, y_2)$ can be written as the solution to the following PDE

$$\begin{cases}
\begin{aligned} 0 &= \partial_t u(t,y) + A_t\ \partial_{y_1} u(t,y) + B_t\ \partial_{y_2} u(t,y) \\ &\quad + C_t \partial_{y_1}^2 u(t,y) + D_t \partial_{y_1,y_2}^2 u(t,y) + E_t \partial_{y_2}^2 u(t,y) \end{aligned} \\
u(t_{n+1}, y_1, y_2) = g(y_1 + Z_{t_{n+1}}^{(1)}, y_2 + Z_{t_{n+1}}^{(2)}, v_{t_{n+1}}^{(1)}, v_{t_{n+1}}^{(2)})\,,
\end{cases} \tag{3.7}$$

where

$$\begin{aligned}
A_t &= r - \frac{1}{2} v_t^{(1)} & & & B_t &= r - \frac{1}{2} v_t^{(2)} \\
C_t &= \frac{1}{2}(a_{1,1}^2 + a_{2,2}^2) v_t^{(1)} & D_t &= a_{1,2} a_{2,2} \sqrt{v_t^{(1)}} \sqrt{v_t^{(2)}} & E_t &= \frac{1}{2} a_{2,2}^2 v_t^{(2)}.
\end{aligned}$$

FFT Based Solution

Taking the Fourier transform of (3.7), as in the FST method of [59] , and letting $\widehat{u}(t,\omega)$ denote the Fourier transform of $u(t,y)$ in the second argument, we have

$$\begin{cases} \partial_t \widehat{u}(t,\omega) + (iA_t\omega_1 + iB_t\omega_2 - C_t\omega_1^2 - D_t\omega_1\omega_2 - E_t\omega_2^2)\widehat{u}(t,\omega) = 0\,, \\ \qquad\qquad \widehat{u}(t_{n+1},\omega) = e^{i\omega_1 Z^{(1)}_{t_{n+1}} + i\omega_2 Z^{(2)}_{t_{n+1}}}\ \widehat{g}(\omega_1,\omega_2,v^{(1)}_{n+1},v^{(2)}_{n+1})\,, \end{cases}$$

which is an ODE in Fourier space, wrt time, for each fixed $\omega = (\omega_1,\omega_2)$. Solving this ODE in closed form we find

$$\widehat{u}(t_n,\omega) = \widehat{u}(t_{n+1},\omega)\exp\left\{\int_{t_n}^{t_{n+1}} \left(iA_s\omega_1 + iB_s\omega_2 - C_s\omega_1^2 - D_s\omega_1\omega_2 - E_s\omega_t^2\right) ds\right\}.$$

Using the FST's discretization methodology, we then have

$$u_n = \mathrm{FFT}_2^{-1}[\mathrm{FFT}_2[g_{n+1}]\exp(\Psi_{n,n+1})]$$

where FFT_2 denotes the $2d$ FFT and

$$\Psi_{n,n+1}(\omega_1,\omega_2) = i\omega_1 Z^{(1)}_{t_{n+1}} + i\omega_2 Z^{(2)}_{t_{n+1}} + \int_{t_n}^{t_{n+1}} \left(iA_s\omega_1 + iB_s\omega_2 - C_s\omega_1^2 - D_s\omega_1\omega_2 - E_s\omega_2^2\right) ds.$$

Statistics for Clustering and Path Reduction

From (3.4.2), we identify the following path statistic for $[v]_n^{n+1} = ([v^{(1)}]_n^{n+1}, [v^{(2)}]_n^{n+1})$:

$$\Lambda_n([v]_n^{n+1}) = \left(\int_{t_n}^{t} \sqrt{v_t^{(1)}} \sum_{j=3}^{4} a_{1,j} dB_t^{(j)},\ \int_{t_n}^{t} \sqrt{v_t^{(2)}} \sum_{j=3}^{4} a_{2,j} dB_t^{(j)},\right.$$
$$\left. \int_{t_n}^{t_{n+1}} v_t^{(1)} dt, \int_{t_n}^{t_{n+1}} v_t^{(2)} dt,\ \int_{t_n}^{t_{n+1}} \sqrt{v_t^{(1)}}\sqrt{v_t^{(2)}} dt \right)$$

so that

$$\Theta([v]_n^{n+1}) = (v^{(1)}_{t_n}, v^{(2)}_{t_n}, \Lambda([v]_n^{n+1}), v^{(1)}_{t_{n+1}}, v^{(2)}_{t_{n+1}}).$$

3.4.3 Numerical Experiments, Results, and Discussion

In this section we carry out various tests to check the performance of the prml-GPSM in comparison to both LSM. In a separate test, we also compute sensitivities using the modified GPSM. From a deterministic perspective, this a $4d$ problem and so it is not tractable from a purely finite difference approach. Also, due to the correlations, the model is not of the affine class and so is not amenable to direct Fourier techniques.

Procedure for LSM and prml-GPSM

We run N_{trial} of an LSM and prml-MGPSM implementation as in Section 2.5.3 and Section 3.3.1. Our basis for LSM consists of all functions of the form $\phi_{i,j,k,l}(S_1,S_2,v_1,v_2) = S_1^i \cdot S_2^j \cdot v_1^k \cdot v_2^l$

and $\phi_m(S_1, S_2) = h(S_1, S_2)^m$ such that $i + j + k + l \leq deg_{lsm}$ and $m \leq deg_{lsm}$, where ϕ_m is the exercise function. Our basis for GPSM consists of all functions of the form $\phi_{i,j}(v_1, v_2) = v_1^i \cdot v_2^j$ with $i + j \leq deg_{gpsm}$.

The quantities L_{lsm}, L_{gpsm} denote the total dimension of the basis for LSM and GPSM, respectively.

Procedure for the MLSM and prml-MGPSM sensitivities

We run N_{trial} of the MLSM and prml-MGPSM with modified low estimator. For the MLSM we again apply the analogous approach as the GPSM with an initial dispersion constructed as in Section 3.2.1.

For the MGPSM and MLSM, we set $T^* = 1$ and $T^* = 0.25$, respectively. We find these approaches provide a good balance between bias and variance. As mentioned in Section 3.2.2, when $d_v > 1$ the variances associated with $\partial_v V_{0,l}^N(S_0, v_0)$ tend to increase substantially. Thus, to obtain sensible results, we solve considerably more conditional PDEs as shown in Table 3.6. To provide a fair comparison with MLSM, we also simulate many more paths than before as shown in Table 3.6.

Settings for Trials

We work with a Bermudan max-put option with exercise function

$$h(S_1, S_2) = (K - \max(S_1, S_2))_+.$$

Our model and option parameters are fixed across trials and listed in Table 3.5. The parameters for the numerical aspect of our computations are provided in Table 3.6.

T	K	Exercise Frequency	r	$S_0^{(1)}$	$v_0^{(1)}$	κ_1	θ_1	η_1	$S_0^{(2)}$	$v_0^{(2)}$	κ_2	θ_2	η_2
1	10	$T/12$	0.025	10	0.45	1.52	0.45	0.4	10	0.3	1.3	0.30	0.43

$$[\rho_{i,j}] = \begin{bmatrix} 1 & \rho_{S^1,S^2} & \rho_{S^1,v^1} & \rho_{S^1,v^2} \\ \rho_{S^2,S^1} & 1 & \rho_{S^2,v^1} & \rho_{S^2,v^2} \\ \rho_{v^1,S^1} & \rho_{v^1,S^2} & 1 & \rho_{v^1,v^2} \\ \rho_{v^2,S^1} & \rho_{v^2,S^2} & \rho_{v^2,v^1} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.2 & -0.3 & -0.15 \\ 0.2 & 1 & -0.11 & -0.35 \\ -0.3 & -0.11 & 1 & 0.2 \\ -0.15 & -0.35 & 0.2 & 1 \end{bmatrix}$$

Table 3.5: Parameters used in our option pricing model.

Results and Discussion

In Appendices 3.B and 3.C we present the pricing statistics and OEBs for LSM and GPSM. We also present sensitivities for GPSM.

We begin by comparing our pricing statistics for the LSM and prml-GPSM as shown in Tables 3.22 and 3.23. We find that the LSM's low estimator is 1.17 with a very high direct estimate, whereas the prml-GPSM provides a better low estimate of 1.18 that is also attainable from the

N_{trial}	$N_{LSM}^{S,v}$	$N_{sim,0}^{v}$	$N_{sim,1}^{v}$	N_{clust}	N_{step}
100	500 000	10 000	100	4 500	1 000

$\deg_{full}$	$\deg_{mix}$	L_{full}	L_{mix}
4	4	61	10

$N_0^{\mathcal{S}}$	$(\log S^{(i)}/S_0^{(i)})_{min}$	$(\log S^{(i)}/S_0^{(i)})_{max}$
2^6	-3	3

Table 3.6: Parameters fixed when pricing using the prml-GPSM and LSM.

Sub-trial Type	N_{l}^{S}
a	2^7
b	2^8
c	2^9

Table 3.7: Subtrial settings for pricing using the prml-GPSM indicating the grid resolutions for the higher levels. We keep $N_0^S = 2^6, N_{sim,0}^v = 10\ 000, N_{sim,1}^v = 100$ fixed throughout subtrials. These settings have no effect on the computation of sensitivities.

N_{trial}	$\deg_{full}$	$\deg_{mix}$	L_{full}^{d}	L_{full}^{l}	L_{mix}
100	4	4	61	57	10

Level	0	1	2	3
N_k^v	50 000	500	250	100
$N_k^{\mathcal{S}}$	2^5	2^6	2^7	2^8

N_{LSM}^{v}	1 250 000

Table 3.8: ml-GPSM and MLSM settings used for the computation of sensitivities. Only a single type of trial is used for GPSM sensitivities unlike pricing. We fix $T^* = 1$ and $T^* = 0.25$ in our initial dispersions for the GPSM and MLSM, respectively.

direct estimator. The agreement of the direct and low estimators for the prml-GPSM is similar to the $1d + 1d$ example considered in Section 2.5. In that case, we saw the prices matched the reference value to two or three decimal places. Both algorithms produce a standard deviation that is on the order of 10^{-3}, except that the LSM used 500 000 paths of (S, v) and prml-GPSM effectively only used $4\ 500 + 100$ paths of $(v, \Delta W^v)$. We also note that the prml-GPSM provides estimates of the price for all $S \in \mathcal{S}$ and not just a single value (along with S-sensitivities), unlike the LSM.

In Figures 3.4 - 3.9, we plot three types of slices of the OEBs for both types of algorithms. As the four dimensional objects may be parametrized via d_S-dimensional v-slices, we may gain some insight by looking at slices for certain meaningful choices of v. Also, for fixed v, we have

some intuition with regards to how the slices should look considering the structure of the max-put option. Thus, for this example, and also the examples in Chapter 4, we consistently plot our boundaries at three levels:

1. $v_1 = 0.5 \cdot \theta_{v_1}, v_2 = 1.5 \cdot \theta_{v_2}$
   - Figures 3.4 and 3.5
2. $v_1 = \theta_{v_1}, v_2 = \theta_{v_2}$
   - Figures 3.6 and 3.7
3. $v_1 = 1.5 \cdot \theta_{v_1}, v_2 = 0.5 \cdot \theta_{v_2}$
   - Figures 3.8 and 3.9

Based on our heat map colour scheme, we see that the LSM tends to be $75 - 80\%$ inconsistent in deciding whether to exercise or not in many different regions whereas prml-GPSM is almost always consistent, except for the very earliest dates in Figures 3.5, 3.9. We also note that for many dates, the LSM and prml-GPSM describe different exercise policies. As we are more inclined to trust the prml-GPSM, this suggests that some of the regions described by the LSM are consistently inaccurate which explains the slight pricing differences that we observe and mirrors the situation in Chapter 2.

We now consider our results for computing sensitivities for the prml-MGPSM and MLSM as shown in Tables 3.24 and 3.25. Firstly, our initial dispersion of $T^* = 0.25$ for MLSM has added a very slight upwards bias to $V_{0,l}^N$, whereas, our dispersion adds a negligible bias to the MGPSM, similar to the standard Heston model. While we do not have a reference value for our sensitivities, the agreement of our direct and low-type sensitivity estimates in Table 3.25 suggest that we correctly determined our S-sensitivities to three or four decimal places. This is further confirmed by the fact that the MLSM produces similar values until two decimal places for $\partial_{S_1} V_{0,l}^N, \partial_{S_2} V_{0,l}^N$ before being drowned out by the variance. Second order S-sensitivities for the MLSM have fairly low variance, however, we see they differ somewhat from the MGPSM. We also observed this, however, in the standard Heston model when comparing to the reference solution. For sensitivities wrt to v, we see again both the MGPSM and MLSM have considerably higher variances, than S-sensitivities, however, they are not unreasonably poor. These estimates resulted from 50 000 paths at level-0 for MGPSM and 1 250 000 paths for LSM, which is much higher than the number of paths needed to get reasonable price and boundary estimates. In the case of the standard Heston model, we obtained roughly the same level of accuracy in our v-sensitivities, except we used the same number of paths as for pricing: 10 000 and 100 000 for MGPSM and MLSM, respectively. This reveals the major problem that arises when $d_v > 1$. In our backtests, we found that setting $\rho = I$ eliminates the disproportionate increase and sheds some light on the causes. Thus, the choice of the number of paths should be chosen to reflect the underlying correlations of the model. It also seems possible that other time-zero regression approaches on the modified estimator along with different choices of the initial dispersion may improve results.

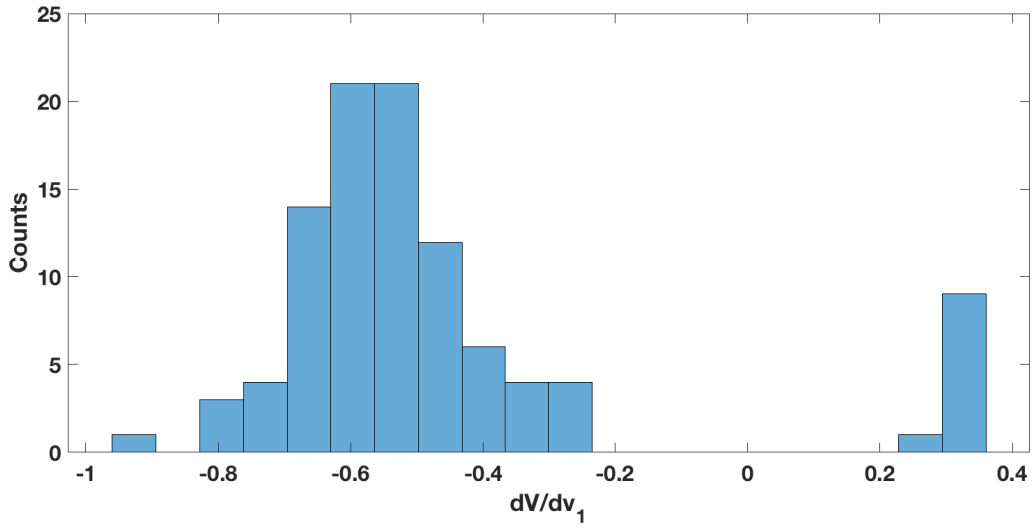

Figure 3.2: Numerical issues with the computation of $\partial_{v_1} V_{0,l}^N$ for the MLSM. The mean of the second mode is 0.3215.

Finally, we notice a discrepancy in the MLSM and GPSM's estimate of $\partial_v V_{0,l}^N$ along with the fact that the MLSM's is negative, which is certainly incorrect. To analyze the results, we consider Figure 3.2 above. We see that there is some numerical instability with this estimated quantity. Upon closer inspection, the second mode has 10 points with a mean of 0.3215, and standard deviation of 0.0181. This shows that 10% of the time, the MLSM computes $\partial_{v_1} V_{0,l}^N$ with relatively high accuracy, but has numerical inconsistencies which need to be studied more in the future. We note, that the GPSM does not demonstrate such stability issues.

In terms of run times, again, we note that the FST is not the most efficient way to solve our conditional PDEs and that our prml features are not optimized. Thus, we should view our prml-GPSM and ml-MGPSM run times as strict upper bounds. Our choice of the number of simulated paths for the LSM and MLSM were such that their run times were comparable with the prml-GPSM and ml-MGPSM, respectively. With this in mind, we make the following observations. First, the LSM's price estimates have variances that are comparable to that of the GPSM, which suggests that there is no variance reduction in prices. However, since the prices are wide in their range, it is difficult to reach any conclusions and one may need to carry out a duality estimate as in [32], [55], adding to the LSM's run time, to get a tight upper bound. Beyond time-zero prices, however, we clearly see variance reduction in OEBs and most sensitivities. Thus, overall, we may conclude that the prml-GPSM and ml-MGPSM are more efficient than the LSM and MLSM, respectively.

3.A Pseudo-code Descriptions

3.A.1 Clustering

Here we describe the clustering procedure that takes place when solving over intervals $[t_n, t_{n+1}]$. We suppose the algorithm has been run for all times $\{t_{n+1}, ..., t_M\}$.

Algorithm 3 Clustering

1: Fix $[\Theta]_{i,j}$ where $i = 1, ..., N$, $j = 1, ...m$ ▷ m is the dimension of the path statistic that we cluster.
2: Apply agglomerative (bottom-up) hierarchical clustering on the N observations with maximum distance joining.
▷ Obtain a hierarchical tree for the clusters.
3: Cut the hierarchical tree to obtiain a maximum of L_n clusters.
4: For each cluster
- record the number of paths, $L_{n,k}$
- Compute the average of the cluster, $\Theta_k^* \in \mathbb{R}^{d_\Theta}$

5: Use the values $L_n, \{L_{n,k}\}$ and the paths $[v^{k*}]_n^{n+1}$ when computing (3.1)

3.A.2 mlMC/Multi-grids

We provide guidelines for computing $a_n(S)$ over the time interval $[t_n, t_{n+1}]$ in a manner that keeps the number of interpolations to a minimum. Given our coefficient matrix $a_{n+1}(S)$ at level K, we carry out the following.

Algorithm 4 mlMC/Multi-Grids For $a_n(S)$

for $k = 0 : K - 1$ **do**
  Interpolate $a_{n+1}(S)$ from resolution level K to level k.
end for
for $k = K - 1 : 0$ **do**
  for $j = 1 : N_k^v$ **do**
    Generate boundary conditions for the PDEs to be solved at resolutions k and $k + 1$.
      ▷ Requires computing the matrix $a_n(S) \cdot \phi(v_n^j)$ using $a_n(S)$ at grid level $k, k + 1$.
    Compute $P_n^{j,k}(S)$ and $P_n^{j,k+1}(S)$ via a numerical PDE technique.
  end for
  Compute $\frac{1}{N_k^v} \sum_{j=1}^{N_k^v} \phi(v_n^{j,k}) P_n^{j,k}(S)$ and $\frac{1}{N_k^v} \sum_{j=1}^{N_k^v} \phi(v_n^{j,k}) P_n^{j,k+1}(S)$
  Interpolate both to grid level K
  Compute $\frac{1}{N_k^v} \sum_{j=1}^{N_k^v} \phi(v_n^{j,k}) \Big(P_n^{j,k+1}(S) - P_n^{j,k}(S) \Big)$
end for
Repeat the above procedure for level 0 and compute $a_n(S)$ which is defined on grid level K as in (3.3).

3.B Pricing Statistics

Clustering Results for the $1d + 1d$ Heston model.

Subtrial Type	N_{cl}	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	3 000	Direct	0.7420	$8.4562 \cdot 10^{-4}$	19.51
		Low	0.7418	$1.0134 \cdot 10^{-3}$	
b	4 500	Direct	0.7420	$9.1536 \cdot 10^{-4}$	21.25
		Low	0.7417	$8.2428 \cdot 10^{-4}$	
c	6 000	Direct	0.7421	$8.2547 \cdot 10^{-4}$	21.71
		Low	0.7417	$8.8071 \cdot 10^{-4}$	

Table 3.9: Numerical Results for pr-GPSM Trial 1. $T = 0.25$ with 10 equally spaced exercise dates. $N_S = 2^9$ and $N_{sims} = 10\ 000$. The reference value is 0.7416.

Subtrial Type	N_{cl}	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	3 000	Direct	1.4534	$1.3606 \cdot 10^{-3}$	23.64
		Low	1.4530	$1.5887 \cdot 10^{-3}$	
b	4 500	Direct	1.4533	$1.4194 \cdot 10^{-3}$	25.60
		Low	1.4530	$1.3965 \cdot 10^{-3}$	
c	6 000	Direct	1.4534	$1.2893 \cdot 10^{-3}$	26.77
		Low	1.4529	$1.3549 \cdot 10^{-3}$	

Table 3.10: Numerical Results for pr-GPSM Trial 2. $T = 1.0$ with 12 equally spaced exercise dates. $N_S = 2^9$ and $N_{sims} = 10\ 000$. The reference value is 1.4528.

Subtrial Type	N_{cl}	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	3 000	Direct	2.2121	$1.1538 \cdot 10^{-3}$	58.58
		Low	2.2113	$1.3435 \cdot 10^{-3}$	
b	4 500	Direct	2.2117	$1.1625 \cdot 10^{-3}$	62.45
		Low	2.2115	$1.2465 \cdot 10^{-3}$	
c	6 000	Direct	2.2117	$1.1086 \cdot 10^{-3}$	65.81
		Low	2.2114	$1.1589 \cdot 10^{-3}$	

Table 3.11: Numerical Results for pr-GPSM Trial 3. $T = 2.5$ with 30 equally spaced exercise dates. $N_S = 2^9$ and $N_{sims} = 10\ 000$. The reference value is 2.2111.

mlMC Results for the $1d + 1d$ Heston model.

Subtrial Type	N_1^S	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	2^7	Direct	0.7439	$8.6588 \cdot 10^{-4}$	6.81
		Low	0.7435	$9.8859 \cdot 10^{-4}$	5.24
b	2^8	Direct	0.7436	$8.3096 \cdot 10^{-4}$	6.72
		Low	0.7435	$9.1579 \cdot 10^{-4}$	5.33
c	2^9	Direct	0.7438	$8.2342 \cdot 10^{-4}$	6.68
		Low	0.7435	$9.1904 \cdot 10^{-4}$	5.29

Table 3.12: Numerical Results for ml-GPSM Trial 1. $T = 0.25$ with 10 equally spaced exercise dates. $N_0^v = 10\ 000$, $N_0^S = 2^6$, $N_1^v = 100$. The reference value is 0.7416.

Subtrial Type	N_1^S	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	2^7	Direct	1.4538	$1.3852 \cdot 10^{-3}$	7.68
		Low	1.4539	$1.4801 \cdot 10^{-3}$	6.40
b	2^8	Direct	1.4532	$1.4691 \cdot 10^{-3}$	8.02
		Low	1.4535	$1.4200 \cdot 10^{-3}$	5.93
c	2^9	Direct	1.4531	$1.2420 \cdot 10^{-3}$	7.83
		Low	1.4528	$1.4329 \cdot 10^{-3}$	6.15

Table 3.13: Numerical Results for ml-GPSM Trial 2. $T = 1$ with 12 equally spaced exercise dates. $N_0^v = 10\ 000$, $N_0^S = 2^6$, $N_1^v = 100$. The reference value is 1.4528.

Subtrial Type	N_1^S	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	2^7	Direct	2.2118	$1.1207 \cdot 10^{-3}$	16.68
		Low	2.2119	$1.2268 \cdot 10^{-3}$	12.91
b	2^8	Direct	2.2114	$1.1117 \cdot 10^{-3}$	16.68
		Low	2.2115	$1.2100 \cdot 10^{-3}$	12.98
c	2^9	Direct	2.2113	$1.1405 \cdot 10^{-3}$	17.12
		Low	2.2115	$1.2191 \cdot 10^{-3}$	12.78

Table 3.14: Numerical Results for ml-GPSM Trial 3. $T = 2.5$ with 30 equally spaced exercise dates. $N_0^v = 10\ 000$, $N_0^S = 2^6$, $N_1^v = 100$. The reference value is 2.2111.

prml Results for the $1d + 1d$ Heston model.

Subtrial Type	N_1^S	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	2^7	Direct	0.7437	$9.4084 \cdot 10^{-4}$	18.94
		Low	0.7435	$1.0106 \cdot 10^{-3}$	
b	2^8	Direct	0.7422	$9.4893 \cdot 10^{-4}$	18.91
		Low	0.7421	$1.0181 \cdot 10^{-3}$	
c	2^9	Direct	0.7418	$9.9186 \cdot 10^{-4}$	18.88
		Low	0.7416	$1.0327 \cdot 10^{-3}$	

Table 3.15: Numerical Results for prml-GPSM Trial 1. $T = 0.25$ with 10 equally spaced exercise dates. $N_0^v = 10\ 000$, $N_0^S = 2^6$, $N_1^v = 100$, $N_{cl} = 4\ 500$. The reference value is 0.7416.

Subtrial Type	N_1^S	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	2^7	Direct	1.4540	$1.3834 \cdot 10^{-3}$	22.89
		Low	1.4538	$1.5052 \cdot 10^{-3}$	
b	2^8	Direct	1.4534	$1.3946 \cdot 10^{-3}$	23.07
		Low	1.4532	$1.5198 \cdot 10^{-3}$	
c	2^9	Direct	1.4531	$1.3948 \cdot 10^{-3}$	22.97
		Low	1.4530	$1.5032 \cdot 10^{-3}$	

Table 3.16: Numerical Results for prml-GPSM Trial 2. $T = 1$ with 12 equally spaced exercise dates. $N_0^v = 10\ 000$, $N_0^S = 2^6$, $N_1^v = 100$, $N_{cl} = 4\ 500$. The reference value is 1.4528.

Subtrial Type	N_1^S	Estimate Type	Mean Price	Std Dev	Mean Run Time (s)
a	2^7	Direct	2.2120	$1.1290 \cdot 10^{-3}$	56.73
		Low	2.2119	$1.2280 \cdot 10^{-3}$	
b	2^8	Direct	2.2117	$1.1443 \cdot 10^{-3}$	56.69
		Low	2.2115	$1.2500 \cdot 10^{-3}$	
c	2^9	Direct	2.2116	$1.1344 \cdot 10^{-3}$	56.48
		Low	2.2115	$1.2225 \cdot 10^{-3}$	

Table 3.17: Numerical Results for prml-GPSM Trial 3. $T = 2.5$ with 30 equally spaced exercise dates. $N_0^v = 10\ 000$, $N_0^S = 2^6$, $N_1^v = 100$, $N_{cl} = 4\ 500$. The reference value is 2.2111.

Sensitivities Results for the $1d + 1d$ Heston model.

V_0	$\partial_S V_0$	$\partial_{SS} V_0$	$\partial_v V_0$	$\partial_{Sv} V_0$
1.4528	-0.4142	0.1050	0.9961	0.03255

Table 3.18: Numerical results for computing sensitivities using finite differences. $T = 1$ with 12 equally spaced exercise dates.

Estimated	$V_{0,l}^N$	$\partial_S V_{0,l}^N$	$\partial_{SS} V_{0,l}^N$	$\partial_v V_{0,l}^N$	$\partial_{Sv} V_{0,l}^N$
Mean	1.4526	-0.4149	0.09630	0.9794	0.04462
Std Dev	$7.4085 \cdot 10^{-3}$	$3.9961 \cdot 10^{-3}$	$2.0228 \cdot 10^{-3}$	$9.4700 \cdot 10^{-2}$	$3.1261 \cdot 10^{-2}$

Table 3.19: Numerical results for computing sensitivities using the MLSM where $T^* = 0.25$. For each run, we use $N = 100\,000$ paths of $S(t), v(t)$. We do not report prices and standard deviations for the direct estimator as they are the same as in Section 2.5.3

N_1^S	Estimated	$V_{0,d}^N$	$\partial_S V_{0,d}^N$	$\partial_{SS} V_{0,d}^N$
2^7	Mean	1.4542	-0.4141	0.1048
	Std Dev	$1.4486 \cdot 10^{-3}$	$2.7866 \cdot 10^{-4}$	$1.6487 \cdot 10^{-4}$
2^8	Mean	1.4533	-0.4142	0.1049
	Std Dev	$1.5571 \cdot 10^{-3}$	$3.0417 \cdot 10^{-4}$	$1.7428 \cdot 10^{-4}$
2^9	Mean	1.4530	-0.4142	0.1049
	Std Dev	$1.4215 \cdot 10^{-3}$	$2.8221 \cdot 10^{-4}$	$1.6168 \cdot 10^{-4}$

N_1^S	Estimated	$V_{0,l}^N$	$\partial_S V_{0,l}^N$	$\partial_{SS} V_{0,l}^N$	$\partial_v V_{0,l}^N$	$\partial_{Sv} V_{0,l}^N$
2^7	Mean	1.4542	-0.4142	0.1048	0.9983	0.03301
	Std Dev	$1.8779 \cdot 10^{-3}$	$3.7349 \cdot 10^{-4}$	$2.0964 \cdot 10^{-4}$	$1.6607 \cdot 10^{-2}$	$3.3528 \cdot 10^{-3}$
2^8	Mean	1.4529	-0.4143	0.1050	0.9963	0.03254
	Std Dev	$1.8079 \cdot 10^{-3}$	$3.6504 \cdot 10^{-4}$	$2.0152 \cdot 10^{-4}$	$1.6630 \cdot 10^{-2}$	$3.3466 \cdot 10^{-3}$
2^9	Mean	1.4525	-0.4144	0.1050	0.9945	0.03220
	Std Dev	$1.9412 \cdot 10^{-3}$	$3.8712 \cdot 10^{-4}$	$2.1930 \cdot 10^{-4}$	$1.8011 \cdot 10^{-2}$	$3.5712 \cdot 10^{-3}$

Table 3.20: Numerical results for computing sensitivities using the modified GPSM where $T^* = 1$. The top table uses the standard direct estimator. The bottom table corresponds to the modified low estimator. For each run, we use $N = 10\,000$ paths of $v(t)$.

N_1^S	Estimated	$V_{0,d}^N$	$\partial_S V_{0,d}^N$	$\partial_{SS} V_{0,d}^N$
2^7	Mean	1.4542	-0.4141	0.1048
	Std Dev	$1.3650 \cdot 10^{-3}$	$2.6980 \cdot 10^{-4}$	$1.5209 \cdot 10^{-4}$
2^8	Mean	1.4538	-0.4142	0.1049
	Std Dev	$1.3400 \cdot 10^{-3}$	$2.6413 \cdot 10^{-4}$	$1.5252 \cdot 10^{-4}$
2^9	Mean	1.4533	-0.4142	0.1049
	Std Dev	$1.3258 \cdot 10^{-3}$	$2.5783 \cdot 10^{-4}$	$1.5212 \cdot 10^{-4}$

N_1^S	Estimated	$V_{0,l}^N$	$\partial_S V_{0,l}^N$	$\partial_{SS} V_{0,l}^N$	$\partial_v V_{0,l}^N$	$\partial_{Sv} V_{0,l}^N$
2^7	Mean	1.4536	-0.4142	0.1048	0.9926	0.03189
	Std Dev	$1.9438 \cdot 10^{-3}$	$3.7831 \cdot 10^{-4}$	$2.1841 \cdot 10^{-4}$	$1.8305 \cdot 10^{-2}$	$3.5825 \cdot 10^{-3}$
2^8	Mean	1.4532	-0.4143	0.1050	0.9963	0.03265
	Std Dev	$1.9548 \cdot 10^{-3}$	$3.8643 \cdot 10^{-4}$	$2.2225 \cdot 10^{-4}$	$1.8232 \cdot 10^{-2}$	$3.5012 \cdot 10^{-3}$
2^9	Mean	1.4528	-0.4143	0.1050	0.9932	0.03174
	Std Dev	$1.986 \cdot 10^{-3}$	$4.1660 \cdot 10^{-4}$	$2.3852 \cdot 10^{-4}$	$1.7744 \cdot 10^{-2}$	$3.4623 \cdot 10^{-3}$

Table 3.21: Numerical results for computing sensitivities using the modified prml-GPSM where $T^* = 1$. The top table uses the standard direct estimator. The bottom table corresponds to the modified low estimator. We set $N_0^{\mathcal{S}} = 2^6$, $N_0^v = 10\ 000$, $N_{cl} = 4\ 500$

Pricing Results for the $2d + 2d$ Heston model

Estimate Type	Mean Price	Std Dev	Run Time (s)
Direct	1.2121	$5.4457 \cdot 10^{-3}$	88.56
Low	1.1765	$2.9584 \cdot 10^{-3}$	74.13

Table 3.22: Resulting price statistics for the LSM algorithm.

$N_{1,i}^{\mathcal{S}}$	Estimate Type	Mean Price	Std Dev	Run Time (s)
2^9	Direct	1.1852	$5.7030 \cdot 10^{-3}$	157.25
	Low	1.1818	$6.0549 \cdot 10^{-3}$	129.53
2^8	Direct	1.1858	$5.6317 \cdot 10^{-3}$	83.96
	Low	1.1823	$6.090 \cdot 10^{-3}$	82.95
2^7	Direct	1.1878	$5.6099 \cdot 10^{-3}$	67.88
	Low	1.1842	$5.5971 \cdot 10^{-3}$	74.34

Table 3.23: Resulting price statistics for the prml-GPSM algorithm. We fix $N_0^{\mathcal{S}} = 2^6, N_{sim,0}^v = 10000, N_{sim,1}^v = 100$, and $N_{cl} = 4500$.

Sensitivities Results for the $2d + 2d$ Heston model

Estimated	$V^N_{0,l}$	$\partial_{S_1} V^N_{0,l}$	$\partial_{S_1S_1} V^N_{0,l}$	$\partial_{S_2} V^N_{0,l}$	$\partial_{S_2S_2} V^N_{0,l}$	$\partial_{S_1S_2} V^N_{0,l}$
Mean	1.1838	-0.1388	0.009937	-0.1995	0.02597	0.026786
Std Dev	$3.0243 \cdot 10^{-3}$	$9.4522 \cdot 10^{-3}$	$4.1055 \cdot 10^{-4}$	$1.2464 \cdot 10^{-3}$	$5.4466 \cdot 10^{-4}$	$2.7114 \cdot 10^{-4}$

Estimated	$\partial_{v_1} V^N_{0,l}$	$\partial_{v_2} V^N_{0,l}$	$\partial_{v_1S_1} V^N_{0,l}$	$\partial_{v_1S_2} V^N_{0,l}$	$\partial_{v_2S_1} V^N_{0,l}$	$\partial_{v_2S_2} V^N_{0,l}$
Mean	-0.4617	0.7462	0.07030	-0.07537	-0.1505	0.1130
Std Dev	$2.8673 \cdot 10^{-1}$	$3.4498 \cdot 10^{-2}$	$7.1656 \cdot 10^{-3}$	$7.8605 \cdot 10^{-3}$	$1.3510 \cdot 10^{-1}$	$9.9115 \cdot 10^{-3}$

Estimate Type	Run Time (s)
Direct	232.71
Modified Low	199.88

Table 3.24: Numerical results for computing sensitivities using the MLSM with $T^* = 0.25$. $T = 1$ with 12 equally spaced exercise dates. The top table gives the settings for our multi-level computation.

Estimated	$V^N_{0,d}$	$\partial_{S_1} V^N_{0,d}$	$\partial_{S_1S_1} V^N_{0,d}$	$\partial_{S_2} V^N_{0,d}$	$\partial_{S_2S_2} V^N_{0,d}$	$\partial_{S_1S_2} V^N_{0,d}$
Mean	1.1837	-0.1316	0.01152	-0.1912	0.02858	0.02821
Std Dev	$2.4314 \cdot 10^{-3}$	$2.1060 \cdot 10^{-4}$	$7.3302 \cdot 10^{-5}$	$2.4665 \cdot 10^{-4}$	$1.080 \cdot 10^{-4}$	$5.6083 \cdot 10^{-5}$

Estimated	$V^N_{0,l}$	$\partial_{S_1} V^N_{0,l}$	$\partial_{S_1S_1} V^N_{0,l}$	$\partial_{S_2} V^N_{0,l}$	$\partial_{S_2S_2} V^N_{0,l}$	$\partial_{S_1S_2} V^N_{0,l}$
Mean	1.1835	-0.1317	0.01156	-0.1912	0.02870	0.02824
Std Dev	$3.8697 \cdot 10^{-3}$	$6.3512 \cdot 10^{-4}$	$2.2764 \cdot 10^{-4}$	$9.8522 \cdot 10^{-4}$	$3.9627 \cdot 10^{-4}$	$3.8970 \cdot 10^{-4}$

Estimated	$\partial_{v_1} V^N_{0,l}$	$\partial_{v_2} V^N_{0,l}$	$\partial_{v_1S_1} V^N_{0,l}$	$\partial_{v_1S_2} V^N_{0,l}$	$\partial_{v_2S_1} V^N_{0,l}$	$\partial_{v_2S_2} V^N_{0,l}$
Mean	0.3724	0.8110	0.09030	-0.08909	-0.1370	0.1465
Std Dev	$3.1264 \cdot 10^{-2}$	$3.5558 \cdot 10^{-2}$	$4.7515 \cdot 10^{-3}$	$7.0874 \cdot 10^{-3}$	$4.5716 \cdot 10^{-3}$	$6.4022 \cdot 10^{-3}$

Estimate Type	Mean Run Time (s)
Direct	218.29
Modified Low	173.31

Table 3.25: Numerical results for computing sensitivities using the ml-GPSM with $T^* = 1$. $T = 1$ with 12 equally spaced exercise dates. The top table gives the settings for our multi-level computation.

3.C Optimal Exercise Boundaries

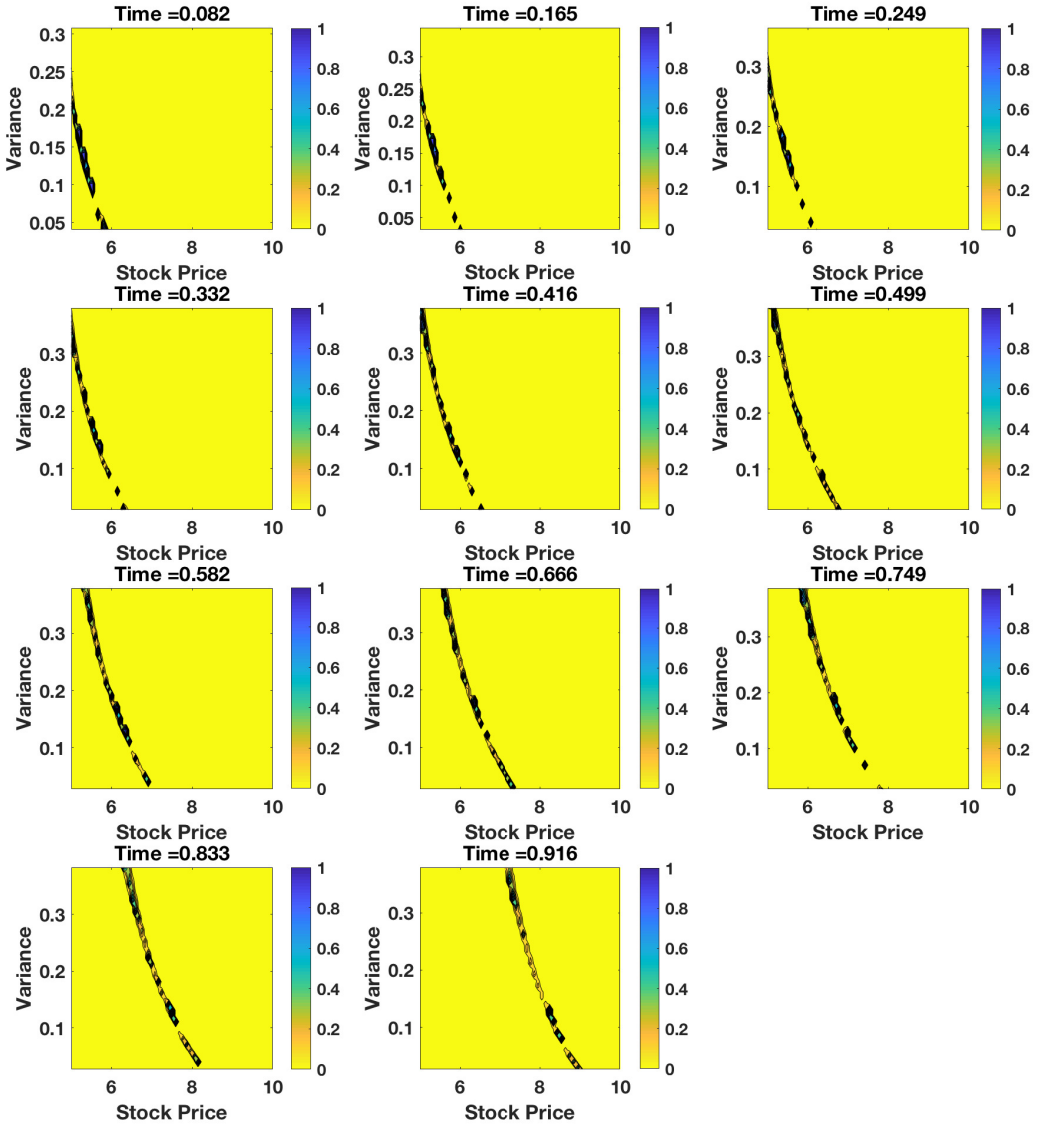

Figure 3.3: Difference optimal exercise boundaries generated by the prml-GPSM Trial 2(c), $N_S = 2^9$, compared to the reference. Dark blue indicates incorrectness with probability 1, yellow indicates correctness with probability 1.

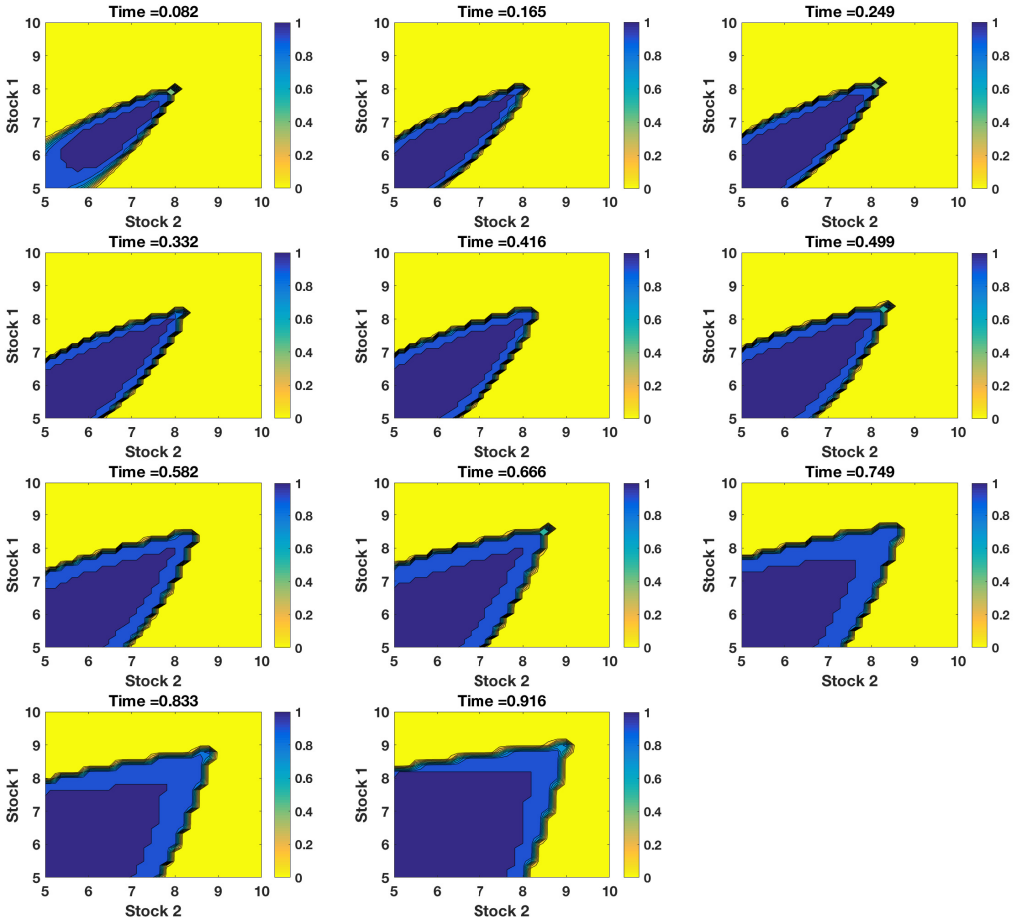

Figure 3.4: A slice of the LSM OEB with $v_1 = 0.5 \cdot \theta_{v_1}$ and $v_2 = 1.5 \cdot \theta_{v_2}$. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^8$.

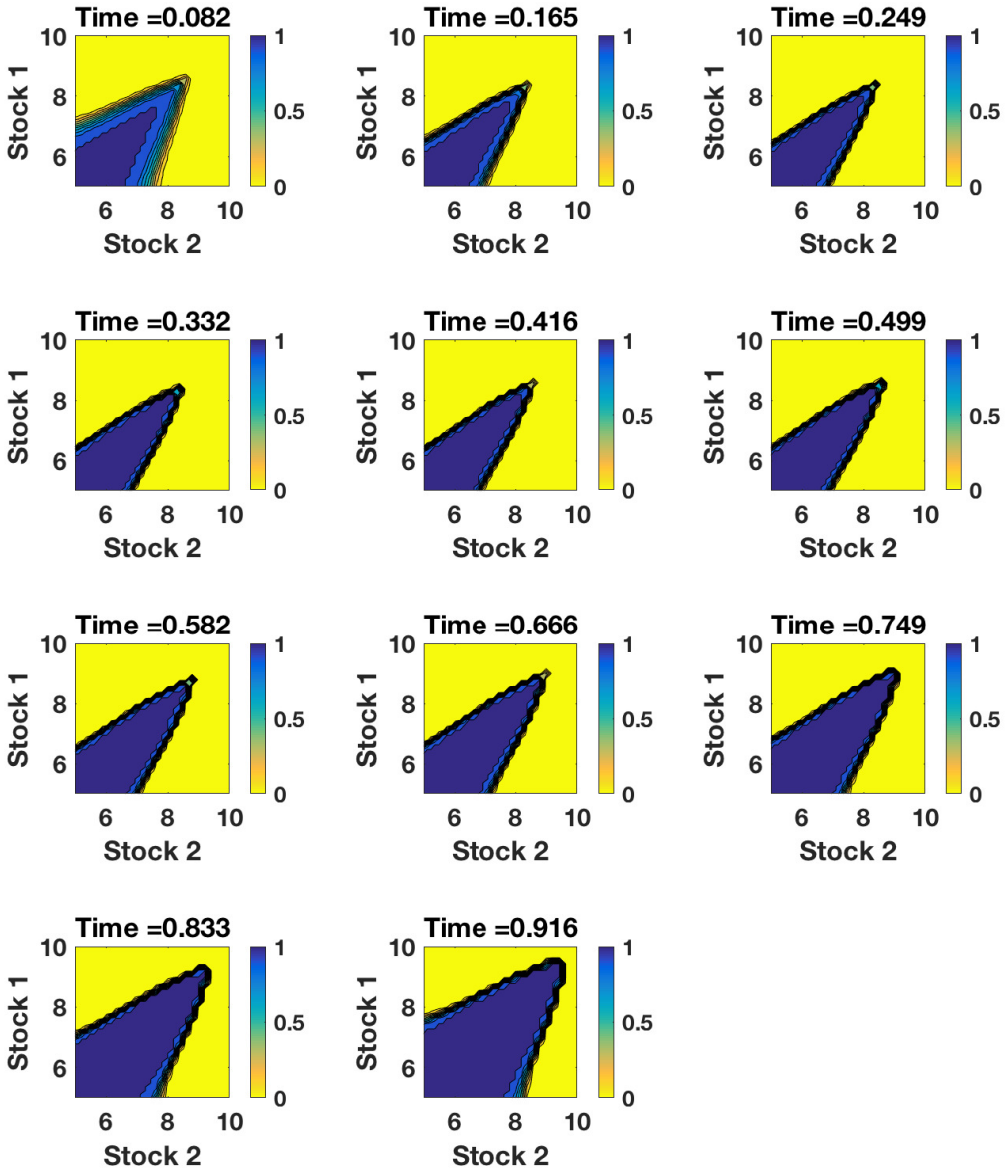

Figure 3.5: A slice of the prml-GPSM OEB with $v_1 = 0.5 \cdot \theta_{v_1}$ and $v_2 = 1.5 \cdot \theta_{v_2}$. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^8$.

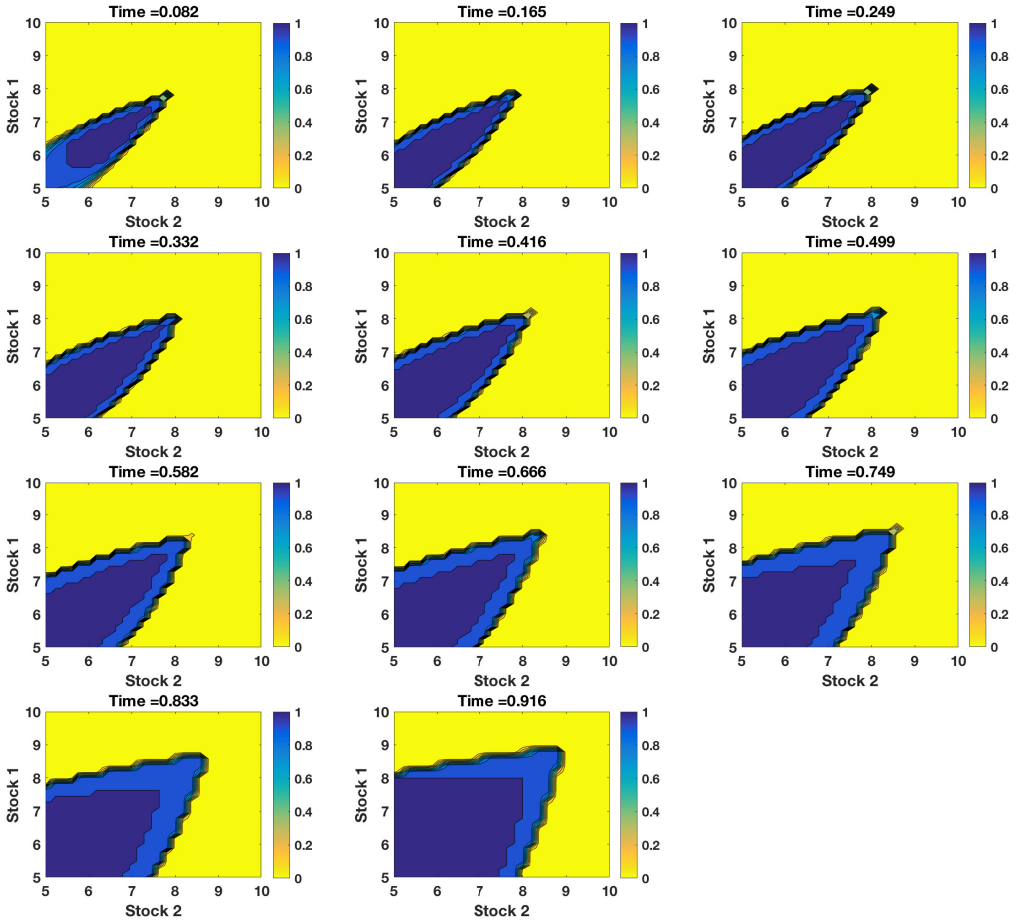

Figure 3.6: A slice of the LSM OEB with $v_1 = \theta_{v_1}$ and $v_2 = \theta_{v_2}$. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^8$.

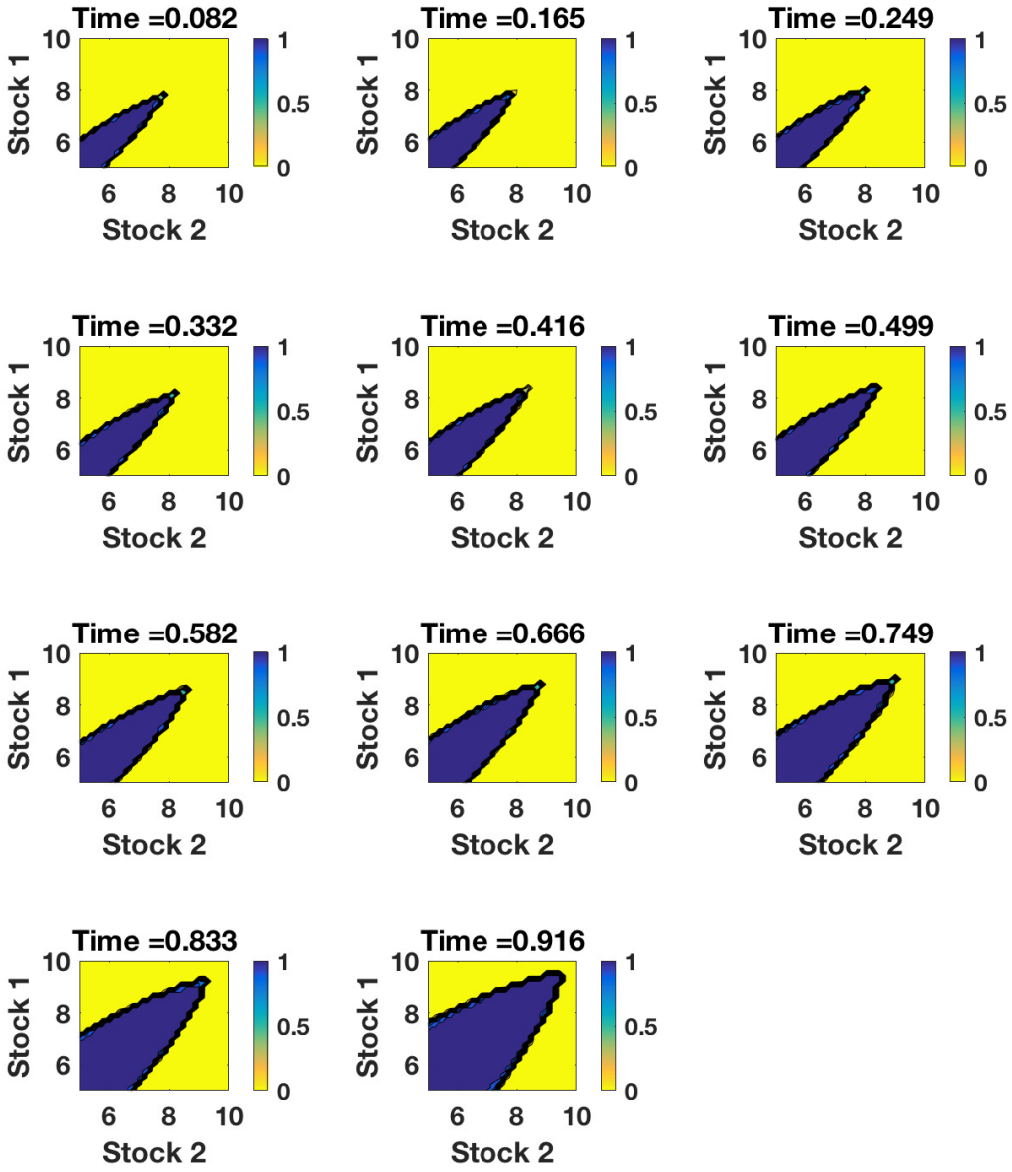

Figure 3.7: A slice of the prml-GPSM OEB with $v_1 = \theta_{v_1}$ and $v_2 = \theta_{v_2}$. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^8$.

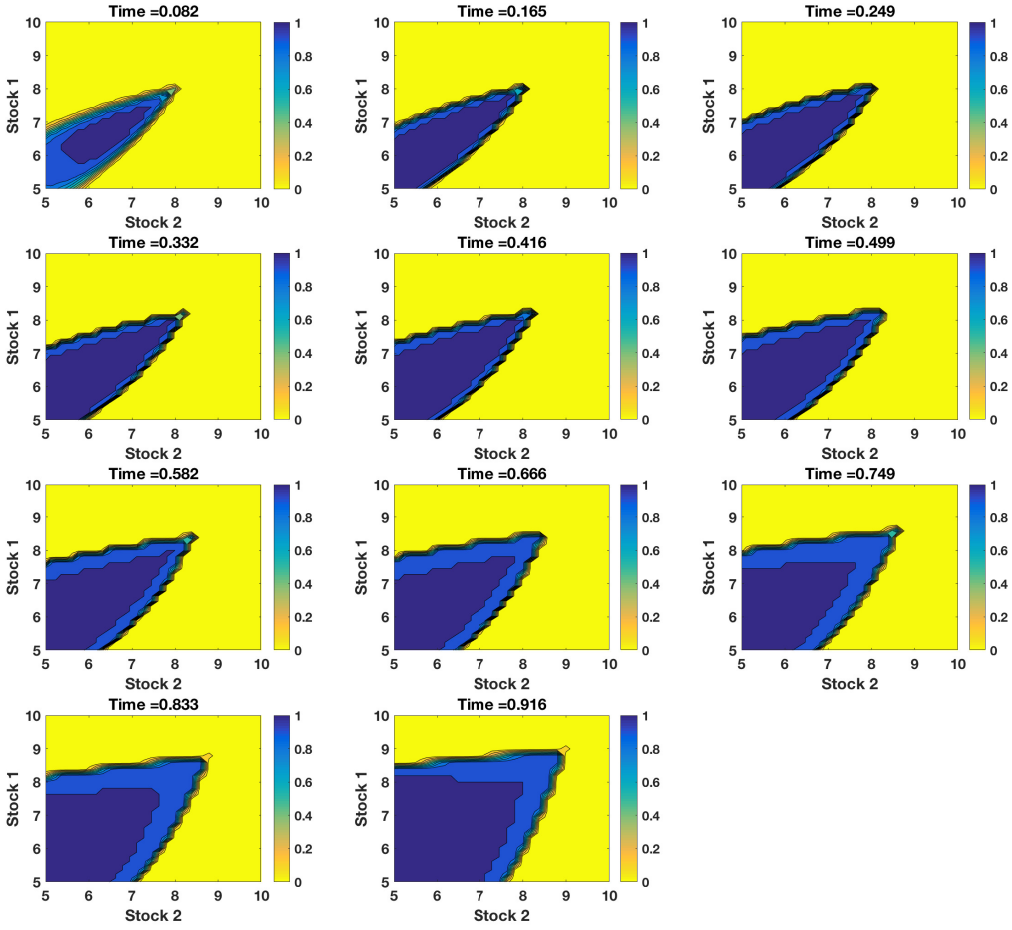

Figure 3.8: A slice of the LSM OEB with $v_1 = 1.5 \cdot \theta_{v_1}$ and $v_2 = 0.5 \cdot \theta_{v_2}$. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^8$.

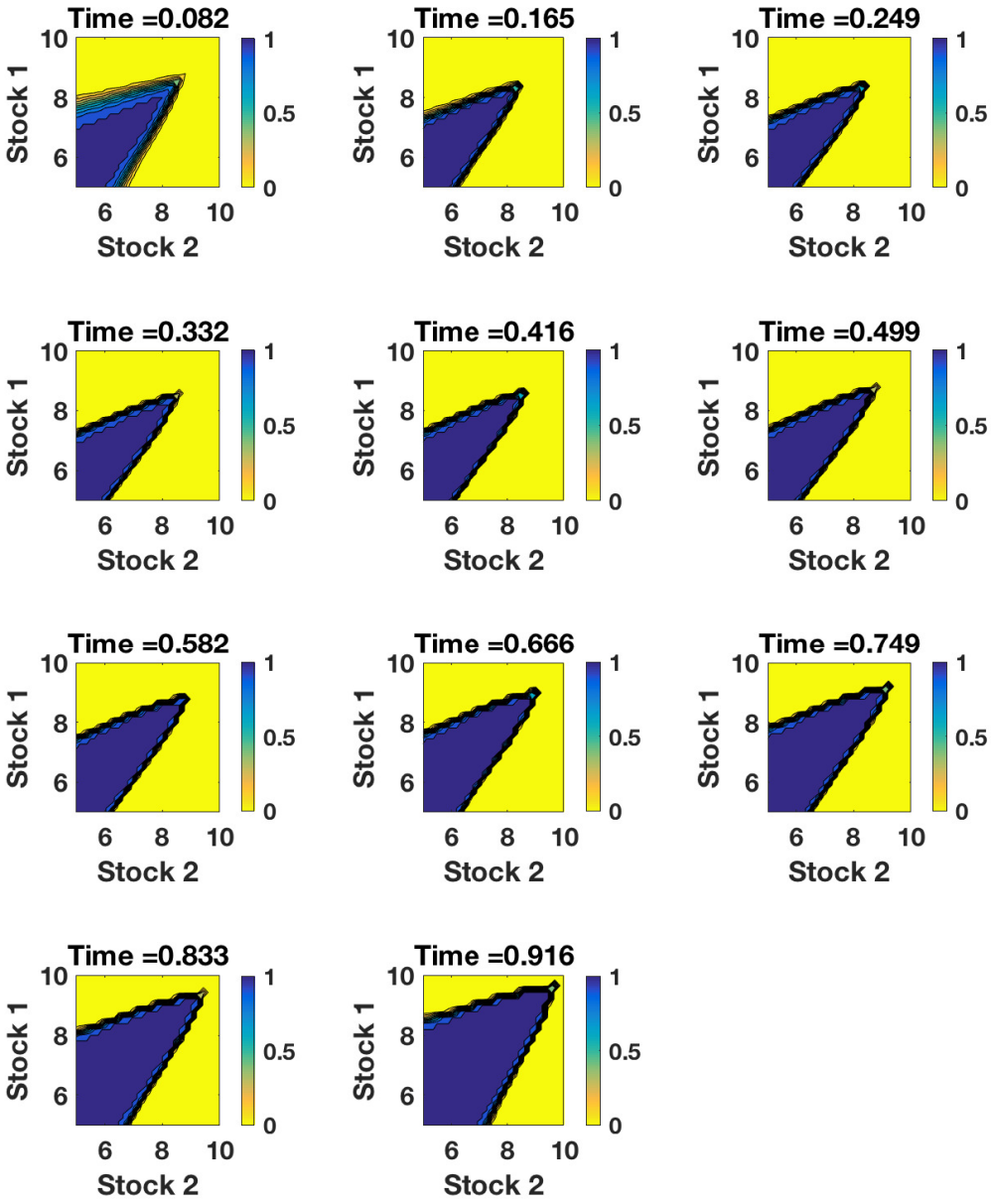

Figure 3.9: A slice of the prml-GPSM OEB with $v_1 = 1.5 \cdot \theta_{v_1}$ and $v_2 = 0.5 \cdot \theta_{v_2}$. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^8$.

Chapter 4

Multi-Factor Stochastic Volatility Models

4.1 Introduction

In previous chapters we applied our hybrid algorithm to the single and multi-variable Heston model and tested the effectiveness of our algorithm along with various extensions that we have developed. In this chapter, we apply our algorithm to single asset multi-factor SV models. As noted by [11], single factor SV models may capture smiles and smirks via the parameter ρ, but they are unable to model the differing of skewness across maturities. This limitation stems from to the correlational parameter being constant. To improve model fits, the authors in [11] introduce a two factor Heston model, sometimes referred to as the Double Heston model, which allows for stochastic correlation between the underlying returns and the volatility process, and effectively captures the maturity dependent slope. They calibrate their model to a time series of volatility surfaces, and they succeed in meaningfully characterizing the contributions of their volatility factors. They find one volatility component typically has high mean reversion and is able to describe short term correlations between returns and volatility, whereas the second component has low mean reversion and is able to describe long term correlations.

Due to the importance of multi-factor SV models, we explore how our hybrid algorithm performs for two different examples. The first example is the Double Heston model as described in [11] where we also use real world calibrated parameters from [56]. Our second example is a model for the spot price of a commodity that is a hybrid between the models presented in [61] and [10]. The work of [61] introduces a multi-factor SV model for the spot price of a commodity is used to study whether volatility is unspanned in commodity markets. There, they model the spot price and cost of carry, analyze a time series of option data written on futures contracts, and find that including a second volatility factor considerably enhances their calibration. The model in [10] is driven by a mean-reverting jump and mean-reverting diffusive process where the diffusive component contains a SV process. The authors then use the model to study ambiguity aversion in commodity markets. Our model is essentially that of [10] except that it includes a second volatility factor driving the diffusive component, similar to [61]. Admittedly, the model has not

been calibrated to the market and it seems that excellent calibration to observed option data can be obtained with two volatility factors. On the other hand, it is well documented that commodity prices exhibit jumping behaviour, including spikes, and we aim to capture this feature through the jump component. We price a Bermudan option on the spot price as opposed to the futures. The model contains some novel properties such as jumps and mean reversion in the asset dynamics, and serves as an interesting example for the final application considered in this thesis. Finally, while our theoretical formulation of the algorithm assumed purely diffusive drivers of our assets, we note that replacing our Brownian motions with Lévy processes has no effect on our proofs of convergence.

Since, in previous chapters, we have already demonstrated that the GPSM is more effective than the LSM in determining prices, OEBs and sensitivities, we dispense with the LSM and focus on how the GPSM performs for multi-factor SV models.

4.2 The Double Heston Model

4.2.1 Model Description and Framework

We work with the following system of SDEs

$$\begin{aligned} dS_t &= S_t(r dt + \sqrt{v_t^{(1)}} dW_t^{(1)} + \sqrt{v_t^{(2)}} dW_t^{(2)}), \\ dv_t^{(1)} &= \kappa_1(\theta_1 - v_t^{(1)})dt + \eta_1 \sqrt{v_t^{(1)}} dW_t^{(3)}, \\ dv_t^{(2)} &= \kappa_2(\theta_2 - v_t^{(2)})dt + \eta_2 \sqrt{v_t^{(2)}} dW_t^{(4)}, \end{aligned}$$

where $(W^{(i)})_{i=1}^4$ is a 4-dimensional risk-neutral Brownian motion with full correlation structure $\rho = [\rho_{i,j}]$. Switching to log-space for our asset prices $X_t = \log S_t$ and applying the Cholesky decomposition to the matrix ρ, we obtain the following system:

$$\begin{aligned} dX_t &= (r - \tfrac{1}{2}q_t)dt + \sqrt{v_t^{(1)}} \sum_{j=1}^{4} a_{1,j} dB_t^{(j)} + \sqrt{v_t^{(2)}} \sum_{j=2}^{4} a_{2,j} dB_t^{(j)}, \\ dv_t^{(1)} &= \kappa_1(\theta_1 - v_t^{(1)})dt + \eta_1 \sqrt{v_t^{(1)}} \sum_{j=3}^{4} a_{3,j} dB_t^{(j)}, \\ dv_t^{(2)} &= \kappa_2(\theta_2 - v_t^{(2)})dt + \eta_2 \sqrt{v_t^{(2)}} a_{4,4} dB_t^{(4)}, \end{aligned}$$

with

$$q_t = v_s^{(1)} + v_s^{(2)} + 2\rho_{1,2}\sqrt{v_s^{(1)}}\sqrt{v_s^{(2)}},$$

where $(B^{(i)})_{i=1}^4$ are independent Brownian motions and $[a_{i,j}]$ is an upper triangular matrix satisfying the usual conditions with $a_{4,4} = 1$, and $\rho = aa^\intercal$.

4.2.2 Derivation of Pricing Formulas

We follow the same approach as in Section 2.5, to compute $\mathbb{E}[e^{-r\Delta t} f(S_{n+1}, v_{n+1}) \mid S_n = S, v_n = v]$ where $v_t = (v_t^{(1)}, v_t^{(2)})$. Again, writing $g(X, v) = f(\exp(X), v)$ we have

$$\mathbb{E}\left[f(S_{n+1}, v_{n+1}) \mid S_n, v_n\right] = \mathbb{E}\left[e^{-r\Delta t}\mathbb{E}\left[g(X_{n+1}, v_{n+1}) \mid [v]_n^{n+1}, X_n\right] \mid X_n, v_n\right]. \tag{4.1}$$

Deriving the Conditional PDE

Focusing on the inner expectation, and treating $[v]_n^{n+1} = ([v^{(1)}]_n^{n+1}, [v^{(2)}]_n^{n+1})$ as a deterministic path in $\mathbb{R}^2$, leads to the following decompositions

$$\begin{aligned}
X_t &= Y_t + Z_t\,,\\
Y_t &= X_{t_n} + \int_{t_n}^{t} (r - \tfrac{1}{2}q_s)ds + \int_{t_n}^{t} a_{1,1}\sqrt{v_s^{(1)}}dB_s^{(1)} + \int_{t_n}^{t}\left(a_{1,2}\sqrt{v_s^{(1)}} + a_{2,2}\sqrt{v_s^{(2)}}\right)dB_s^{(2)},\\
Z_t &= \int_{t_n}^{t} (a_{1,3}\sqrt{v_s^{(1)}} + a_{2,3}\sqrt{v_s^{(2)}})dB_s^{(3)} + \int_{t_n}^{t}\left(a_{1,4}\sqrt{v_s^{(1)}} + a_{2,4}\sqrt{v_s^{(2)}}\right)dB_s^{(4)}.
\end{aligned}$$

Writing the inner expectation on the rhs of (4.1) in terms of Y_t leads to

$$\begin{aligned}
&\mathbb{E}\left[g(X_{n+1}^{(1)}, v_{n+1}^{(1)}, v_{n+1}^{(2)}) \mid [v]_n^{n+1}, X_n\right]\\
&\quad = \mathbb{E}\left[g(Y_{n+1} + Z_{n+1}, v_{n+1}^{(1)}, v_{n+1}^{(2)}) \mid [v]_n^{n+1}, Y_n = y_n\right] =: u(t, y)
\end{aligned}$$

By the Feynman-Kac theorem, the function $u(t, y)$ can be written as the solution to the following PDE

$$\begin{cases} 0 = \partial_t u(t,y) + A_t\ \partial_y u(t,y) + B_t\ \partial_y^2 u(t,y) \\ u(t_{n+1}, y) = g(y + Z_{t_{n+1}}, v_{t_{n+1}}^{(1)}, v_{t_{n+1}}^{(2)})\,, \end{cases} \tag{4.2}$$

where

$$A_t = r - \frac{1}{2}q_t \qquad\qquad B_t = a_{1,1}^2 v_t^{(1)} + \left(a_{1,2}\sqrt{v_t^{(1)}} + a_{2,2}\sqrt{v_t^{(2)}}\right)^2.$$

FFT Based Solution

Taking the Fourier transform of (4.2), as in the FST method of [59] , and letting $\widehat{u}(t, \omega)$ denote the Fourier transform of $u(t, y)$ in the second argument, we have

$$\begin{cases} \partial_t \widehat{u}(t,\omega) + (iA_t\omega - B_t\omega^2)\widehat{u}(t,\omega) = 0\,, \\ \qquad\qquad \widehat{u}(t_{n+1}, \omega) = e^{i\omega Z_{t_{n+1}}}\ \widehat{g}(\omega, v_{n+1}^{(1)}, v_{n+1}^{(2)})\,, \end{cases}$$

which is an ODE in Fourier space, wrt time, for each fixed $\omega \in \mathbb{R}$. Solving this ODE in closed form we find

$$\widehat{u}(t_n, \omega) = \widehat{u}(t_{n+1}, \omega) \exp\left\{\int_{t_n}^{t_{n+1}} \left(iA_s\omega - B_s\omega^2\right) ds\right\}. \tag{4.3}$$

Using the FST's discretization methodology, we then have

$$u_n = \text{FFT}^{-1}[\text{FFT}[g_{n+1}]\exp(\Psi_{n,n+1})]$$

where

$$\Psi_{n,n+1}(\omega) = i\omega Z_{t_{n+1}} + \int_{t_n}^{t_{n+1}} \left(iA_s\omega - B_s\omega^2\right) ds.$$

Statistics for Clustering and Path Reduction

From (4.3), we identify the following path statistic for $[v]_n^{n+1} = ([v^{(1)}]_n^{n+1}, [v^{(2)}]_n^{n+1})$:

$$\Lambda_n([v]_n^{n+1}) = \left(\int_{t_n}^{t_{n+1}} q_s ds, \quad \int_{t_n}^{t_{n+1}} a_{1,1}^2 v_s^{(1)} + \left(a_{1,2}\sqrt{v_s^{(1)}} + a_{2,2}\sqrt{v_s^{(2)}}\right)^2 ds, \quad Z_{t_{n+1}}\right)$$

so that

$$\Theta([v]_n^{n+1}) = (v_{t_n}^{(1)}, v_{t_n}^{(2)}, \Lambda([v]_n^{n+1}), v_{t_{n+1}}^{(1)}, v_{t_{n+1}}^{(2)}).$$

4.2.3 Numerical Experiments, Results, and Discussion

Procedure for pricing and sensitivites for the GPSM

Since $d_S = 1$ we use ml-GPSM without clustering and carry out N_{trial} trials for all of our computations. Again, our basis for GPSM consists of all functions of the form $\phi_{i,j}(v_1, v_2) = v_1^i \cdot v_2^j$ with $i + j \leq deg_{gpsm}$. The quantity L_{gpsm} denotes the total dimension of the basis for the GPSM.

Settings for Trials

We work with a Bermudan put option with exercise function $h(S) = (K - S)_+$. Our model and option parameters are fixed across trials and listed in Table 4.1. These parameters are taken from [56] where the model is calibrated to the volatility surface corresponding to put options written on the ticker DIA, an ETF that tracks the Dow Jones Industrial Average, on May 10, 2012. The surface has strikes ranging from $K = 124$ to $K = 136$ with $\Delta K = 1$ and maturities of 37, 72, 135 and 226 days. In [11], the authors show that one can price vanilla European options in closed-form when certain cross-correlations are set to zero, as the model fits within the affine class. As calibration often requires a highly efficient pricing formula, [56] use the same correlation settings as in [11]. Our mixed MC-PDE approach for computing conditional expectations is, however, unaffected by non-zero cross correlations.

The parameters for the numerical aspect of our computations are provided in Table 4.1.

Results and Discussion

We present our results for pricing, sensitivities in Table 4.9, 4.10, and OEBS in Figures 4.4 - 4.6. As mentioned before, since $d_S = 1$ in this case, clustering is counter-productive as its computational costs outweigh the cost of computing $1d$-FFTs. Hence we only use ml-GPSM for all computations.

T	K	Exercise Frequency	r	S_0
1	130	$T/12$	0.025	130

$v_0^{(1)}$	κ_1	θ_1	η_1	$v_0^{(2)}$	κ_2	θ_2	η_2
0.0258	3.059	0.0317	1.985	0.0092	1.8467	0.0605	0.7149

$$[\rho_{i,j}] = \begin{bmatrix} 1 & \rho_{W^1,W^2} & \rho_{W^1,W^3} & \rho_{W^1,W^4} \\ \rho_{W^2,W^1} & 1 & \rho_{W^2,W^3} & \rho_{W^2,W^4} \\ \rho_{W^3,W^1} & \rho_{W^3,W^2} & 1 & \rho_{W^3,W^4} \\ \rho_{W^4,W^1} & \rho_{W^4,W^2} & \rho_{W^4,W^3} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0.0643 & 0 \\ 0 & 1 & 0 & -0.975 \\ 0.0643 & 0 & 1 & 0 \\ 0 & -0.975 & 0 & 1 \end{bmatrix}$$

Table 4.1: parameters used in our option pricing model.

N_{trial}	N_{step}	$\deg_{mix}$	L_{mix}	$(\log S/S_0)_{min}$	$(\log S/S_0)_{max}$
100	1 000	4	10	-3	3

Table 4.2: parameters fixed when testing ml-GPSM for price and sensitivty computations.

Sub-trial Type	N_1^S
a	2^7
b	2^8
c	2^9

Table 4.3: Subtrial settings for pricing using the ml-GPSM indicating the grid resolutions for the higher levels. We keep $N_0^S = 2^6, N_{sim,0}^v = 50\ 000, N_{sim,1}^v = 100$ fixed throughout subtrials. These settings have no effect on the computation of sensitivities.

Level	0	1
N_k^v	100 000	500
N_k^S	2^5	2^8

Table 4.4: mlMC settings used for the computation of sensitivities. Only a single type of trial is used for sensitivities unlike pricing.

For the choice of parameters in Table 4.1 we observe much higher variances, especially for the low estimator, compared to previous sections and hence require more paths than before. We use 50 000 for level-0 when computing prices and boundaries, and 100 000 paths at level-0 for sensitivities. These computations are not prohibitively expensive. Another issue we observe is a slightly higher than average discrepancy between the direct and low biased time-zero prices, perhaps due to the high variance demonstrated by the low estimator. Our backtesting shows that the root cause is the very high correlation of -97.5% for ρ_{W^2,W^4}. When this parameter is lowered, these high variances and discrepancies vanish. The financial meaning of this parameter is that the DIA's risk-neutral skewness at later maturities is highly negative, and this feature stems from the second volatility component describes the behaviour for later maturities as confirmed in [56]. Thus, for equities with less skewed distributions at later maturities, we expect improved performance.

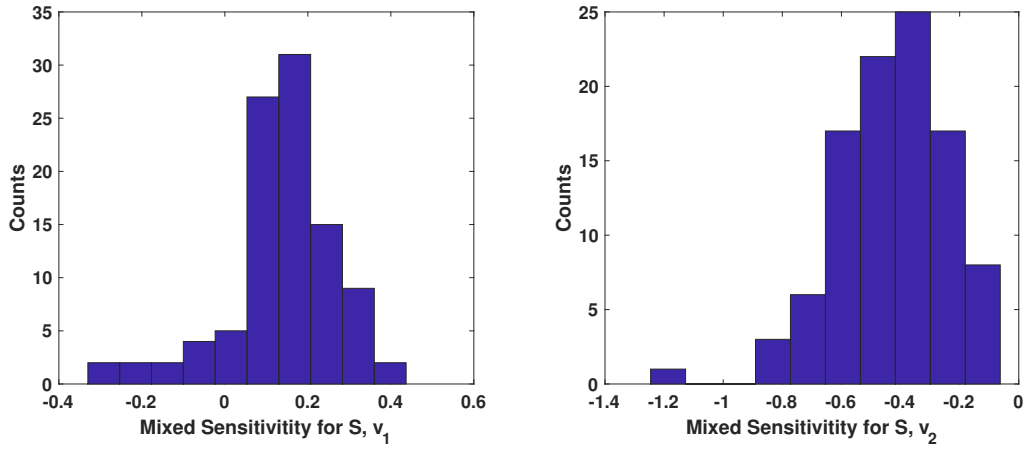

Figure 4.1: Histograms for computed mixed sensitivities

With regards to sensitivities, we see accurate results for S-sensitivities, and moderately accurate results for $\partial_{v_1} V^N_{0,l}, \partial_{v_2} V^N_{0,l}$, but this time, mixed partial derivatives have relatively high variances which again we may attribute to the high negative correlation. We show the results in Figures 4.1, and note that the outliers we see do not play a significant role.

The OEBs shown in Figures 4.4 - 4.6, appear excellent with only a little noise along the interface of the holding and exercise regions.

4.3 A Mean Reverting Commodity Model with Jumps

4.3.1 Model Description and Framework

We work with the following risk-neutral model for the spot price of a commodity $S_t = e^{X_t + Y_t + \theta_t}$ where

$$\begin{aligned} dY_t &= -\kappa_Y Y_t \, dt + d\tilde{J}_t \\ dX_t &= -\kappa_X X_t \, dt + \sqrt{v_t^{(1)}} dW_t^{X^1} + \sqrt{v_t^{(2)}} dW_t^{X^2} \\ dv_t^1 &= \kappa_{v,1}(\theta_1 - v_t^{(1)}) \, dt + \eta_1 \sqrt{v_t^{(1)}} dW_t^{v^1} \\ dv_t^2 &= \kappa_{v,2}(\theta_2 - v_t^{(2)}) \, dt + \eta_2 \sqrt{v_t^{(2)}} dW_t^{v^2} \end{aligned}$$

with $X_0 = Y_0 = 0$, $(W^{X^i}, W^{v^i})_{i=1}^2$ are risk-neutral Brownian-motions with correlations $\rho = [\rho_{i,j}]$, $\tilde{J}_t$ is a compensated compound Poisson process with risk-neutral Levy measure ν, θ_t is a deterministic function that describes the seasonal component of S_t, and the risk free rate is $r > 0$. Carrying out a Cholesky decomposition, we obtain an upper triangular matrix, a, satisfying the usual conditions

with $a_{4,4} = 1$, $\rho = aa^\intercal$, and standard Brownian vector $(W_t^{(i)})_{i=1}^4$ such that

$$\begin{aligned}
dY_t &= -\kappa_Y Y_t \; dt + d\tilde{J}_t \\
dX_t &= -\kappa_X X_t \; dt + \sqrt{v_t^{(1)}} \sum_{j=1}^4 a_{1,j} dW_t^{(j)} + \sqrt{v_t^{(2)}} \sum_{j=2}^4 a_{2,j} dW_t^{(j)} \\
dv_t^{(1)} &= \kappa_{v,1}(\theta_1 - v_t^{(1)}) \; dt + \eta_1 \sum_{j=3}^4 a_{3,j} \sqrt{v_t^{(1)}} dW_t^{(j)} \\
dv_t^{(2)} &= \kappa_{v,2}(\theta_2 - v_t^{(2)}) \; dt + \eta_2 \; a_{4,4} \sqrt{v_t^{(2)}} dW_t^{(4)}.
\end{aligned}$$

4.3.2 Derivation of Pricing Formulas

We follow the same approach as in Section 2.5, to compute

$$\mathbb{E}[e^{-r\Delta t} f(t_{n+1}, X_{n+1}, Y_{n+1}, v_{n+1}) \mid X_n = X, Y_n = Y, v_n = v]$$

where $v_t = (v_t^{(1)}, v_t^{(2)})$. Again, writing $g(t, x, y, v) = f(\exp(x + y + \theta_t), v)$ we have

$$\begin{aligned}
&\mathbb{E}\left[f(t_{n+1}, X_{n+1}, Y_{n+1}, v_{n+1}) \mid X_n, Y_n, v_n \right] \\
&= \mathbb{E}\left[e^{-r\Delta t} \mathbb{E}\left[g(t_{n+1}, X_{n+1}, Y_{n+1}, v_{n+1}) \mid [v]_n^{n+1}, X_n, Y_n \right] \mid X_n, Y_n, v_n \right].
\end{aligned} \tag{4.4}$$

Deriving the Conditional PDE

Focusing on the inner expectation, and treating $[v]_n^{n+1} = ([v^{(1)}]_n^{n+1}, [v^{(2)}]_n^{n+1})$ as a deterministic path in $\mathbb{R}^2$, leads to the decomposition $X_t = \widetilde{X}_t + X'_t$ where $\widetilde{X}$ and X' satisfy the SDEs

$$\begin{aligned}
d\tilde{X}_t &= -\kappa_X X_t dt + \sum_{j=1}^2 a_{1,j} \sqrt{v_t^{(1)}} dW_t^{(j)} + a_{2,2} \sqrt{v_t^{(2)}} dW_t^{(2)} \\
&= -\kappa_X X_t dt + a_{1,1} \sqrt{v_t^{(1)}} dW_t^{(1)} + \left(a_{1,2} \sqrt{v_t^{(1)}} + a_{2,2} \sqrt{v_t^{(2)}} \right) dW_t^{(2)} \\
dX'_t &= \sqrt{v_t^{(1)}} \sum_{j=3}^4 a_{1,j} dW_t^{(j)} + \sqrt{v_j^{(2)}} \sum_{j=3}^4 a_{2,j} dW_t^{(j)},
\end{aligned}$$

We then have

$$\begin{aligned}
&\mathbb{E}\left[g(t_{n+1}, X_{n+1}, Y_{n+1}, v_{n+1}^{(1)}, v_{n+1}^{(2)}) \mid [v]_n^{n+1}, X_n = x, Y_n = y \right] \\
&= \mathbb{E}\left[g(t_{n+1}, \widetilde{X}_{n+1} + X'_{n+1}, Y_{n+1}, v_{n+1}^{(1)}, v_{n+1}^{(2)}) \mid [v]_n^{n+1}, \widetilde{X}_n = x, Y_n = y \right] =: u(t, x, y)
\end{aligned}$$

By the Feynman-Kac theorem, the function $u(t,x,y)$ can be written as the solution to the following PDE

$$\begin{aligned} 0 =& \partial_t u(t,x,y) - \kappa_X x \partial_x u(t,x,y) - (\kappa_Y y + \gamma)\partial_y u(t,x,y) \\ &+ \frac{1}{2}\Big(a_{1,1}^2 v_t^{(1)} + (a_{1,2}\sqrt{v_t^{(1)}} + a_{2,2}\sqrt{v_t^{(2)}})^2\Big)\partial_{xx}^2 u(t,x,y) \\ &+ \int_{\mathbb{R}} \left(u(t,x,y+z) - u(t,x,y)\right) d\nu(z) \\ u(t_{n+1},x,y) =& \, g(t_{n+1}, x + X'_{n+1}, y) \end{aligned} \tag{4.5}$$

where $\gamma = \int_{\mathbb{R}} z \; d\nu(z)$.

FFT Based Solution

We now take the Fourier transform of (4.5), as in the mean-reverting FST method of [38]. We note the identity

$$(x \; \partial_x f(x))\,\widehat{}\,(\omega) = -\hat{f}(\omega) - \omega \partial_\omega \hat{f}(\omega)$$

which follows from integration by parts and basic properties of the Fourier transform. Letting $\widehat{u}(t,\omega_1,\omega_2)$ denote the Fourier transform of $u(t,x,y)$ over real space, we have

$$\begin{aligned} 0 =& \, \partial_t \hat{u}(t,\omega) + \kappa_X \omega_1 \partial_{\omega_1}\hat{u}(t,\omega) + \kappa_Y \omega_2 \partial_{\omega_2}\hat{u}(t,\omega) \\ &+ \left(\kappa_X + \kappa_Y - \gamma i\omega_2 - \frac{1}{2}\Big(a_{1,1}^2 v_t^{(1)} + (a_{1,2}\sqrt{v_t^{(1)}} + a_{2,2}\sqrt{v_t^{(2)}})^2\Big)\omega_1^2\right. \\ &+ \left.\int_{\mathbb{R}} (e^{iz\omega_2} - 1) d\nu(z)\right)\hat{u}(t,\omega) \\ \hat{u}(t_{n+1},\omega_1,\omega_2) =& \, e^{i\omega_1 X'_{t_{n+1}}} \widehat{g}(t_{n+1},\omega_1,\omega_2, v_{t_{n+1}}^{(1)}, v_{t_{n+1}}^{(2)}) \end{aligned}$$

which is a PDE in Fourier space. Solving this PDE in closed form we find

$$\hat{u}(t_n,\omega_1,\omega_2) = \exp\Big(\int_{t_n}^{t_{n+1}} h(s,\alpha(s,\omega_1),\beta(s,\omega_2))ds\Big)\; \widehat{u}(t_{n+1},\alpha(t_{n+1},\omega_1),\beta(t_{n+1},\omega_2))$$

where

$$h(t,\omega_1,\omega_2) = \kappa_X + \kappa_Y - \gamma i\omega_2 - \frac{1}{2}\Big(a_{1,1}^2 v_t^{(1)} + (a_{1,2}\sqrt{v_t^{(1)}} + a_{2,2}\sqrt{v_t^{(2)}})^2\Big)\omega_1^2 + \int_{\mathbb{R}} (e^{iz\omega_2} - 1) d\nu(z) \tag{4.6}$$

and

$$\begin{aligned} \alpha(s,\omega_1) &= \omega_1 e^{\kappa_X (s - t_n)} \\ \beta(s,\omega_2) &= \omega_2 e^{\kappa_Y (s - t_n)}. \end{aligned}$$

Using the FST's discretization methodology, we then have

$$u_n = \text{FFT}_2^{-1}\left[\text{FFT}_2\left[u_{n+1}\left(\alpha_{n+1}(\cdot), \beta_{n+1}(\cdot)\right)\right]\exp(\Psi_{n,n+1})\right] \tag{4.7}$$

where

$$\Psi_{n,n+1}(\omega_1, \omega_2) = i\omega_1 X'_{t_{n+1}} + \int_{t_n}^{t_{n+1}} h(s, \alpha(s, \omega_1), \beta(s, \omega_2))\, ds.$$

A Note on Extrapolation

As discussed in [38], since $|\alpha(s, \omega_1)| > |\omega_1|$ and $|\beta(s, \omega_2)| > |\omega_2|$ for $s \in [t_n, t_{n+1}]$, equation (4.7) implies that we must extrapolate our Fourier space grid. To convert extrapolations into interpolations, we may use the following Fourier identity for compositions with linear maps

$$[f \circ T]\,\widehat{\ }\,(\omega) = |\det\ T|^{-1}(\widehat{f} \circ S)(\omega) \tag{4.8}$$

where $S = (T^*)^{-1}$ and * denotes the conjugate-transpose [25]. In our case

$$S = \begin{bmatrix} \exp(\kappa_X(t_{n+1} - t_n)) & 0 \\ 0 & \exp(\kappa_Y(t_{n+1} - t_n)) \end{bmatrix},$$

$$T = \begin{bmatrix} \exp(-\kappa_X(t_{n+1} - t_n)) & 0 \\ 0 & \exp(-\kappa_Y(t_{n+1} - t_n)) \end{bmatrix},$$

and $|\det T| = \exp(-(\kappa_X + \kappa_Y)(t_{n+1} - t_n))$.

Applying (4.8) to equation 4.7

$$\begin{aligned} &\hat{u}(t_{n+1}, \alpha(t_{n+1}, \omega_1), \beta(t_{n+1}, \omega_2)) \\ =&e^{-(\kappa_X+\kappa_Y)(t_{n+1}-t_n)}[\ u(t_{n+1}, \tilde{\alpha}(t_{n+1}, x), \tilde{\beta}(t_{n+1}, y)\ ]\ \widehat{\ }\ (\omega_1, \omega_2) \end{aligned}$$

so that

$$\begin{aligned} &\text{FFT}_2[u](t_{n+1}, \alpha(t_{n+1}, \omega_1), \beta(t_{n+1}, \omega_2)) \\ =&e^{-(\kappa_X+\kappa_Y)(t_{n+1}-t_n)}\text{FFT}_2[\ u(t_{n+1}, \tilde{\alpha}(t_{n+1}, x), \tilde{\beta}(t_{n+1}, y))\ ]\ \ (\omega_1, \omega_2) \end{aligned}$$

where

$$\begin{aligned} \tilde{\alpha}(t_{n+1}, x) &= x \cdot e^{-\kappa_X(t_{n+1}-t_n)} \\ \tilde{\beta}(t_{n+1}, y) &= y \cdot e^{-\kappa_Y(t_{n+1}-t_n)} \end{aligned}$$

allowing us to interpolate our real space grid as opposed to extrapolating our Fourier space grid.

Statistics for Clustering and Path Reduction

From (3.4.2), we identify the following path statistic for $[v]_n^{n+1} = ([v^{(1)}]_n^{n+1}, [v^{(2)}]_n^{n+1})$:

$$\Lambda_n([v]_n^{n+1}) = \left(a_{1,1}^2 v_t^{(1)} + \left(a_{1,2}\sqrt{v_t^{(1)}} + a_{2,2}\sqrt{v_t^{(2)}} \right)^2, X'_{t_{n+1}} \right)$$

so that

$$\Theta([v]_n^{n+1}) = (v_{t_n}^{(1)}, v_{t_n}^{(2)}, \Lambda([v]_n^{n+1}), v_{t_{n+1}}^{(1)}, v_{t_{n+1}}^{(2)}).$$

4.3.3 Numerical Experiments, Results, and Discussion

Procedure for pricing and sensitivites for the GPSM

Since $d_S = 2$ we use the prml-GPSM for pricing and for sensitivities we use the ml-MGPSM. Again, our basis for GPSM consists of all functions of the form $\phi_{i,j}(v_1, v_2) = v_1^i \cdot v_2^j$ with $i + j \leq deg_{gpsm}$. The quantity L_{gpsm} denote the total dimension of the basis the GPSM.

Settings for Trials

We examine a Bermudan put option with exercise function $h(t, X, Y) = (K - e^{X+Y+\theta(t)})_+$. Our model and option parameters are fixed across trials and listed in Table 4.5. Our jump process J_t is assumed to be compound Poisson with double exponential jumps as in [41], so that

$$d\nu(z) = \lambda \left(p f_+(z) + (1-p) f_-(z) \right) dz$$

where $f_+(z) = \frac{1}{a^+} \exp(-\frac{1}{a^+} z)$ and $f_-(z) = \frac{1}{a^-} \exp(\frac{1}{a^-} z)$

Our parameters for $v^{(1)}, v^{(2)}$ reflect the observation that one volatility process typically has high values for κ, η and the other has low values which enhance the fit of the term structure of volatility and we allow for a full correlational structure between the Brownian motions. The process Y_t jumps 10 times a year with an average up and down jump size of 0.3 and 0.25, respectively. Our choice of mean-reversion parameter, κ_Y, in this case, does not capture spikes, however, our back-tests show that a higher value for this parameter yields results of the same quality in terms of prices and OEBs.

The full set of parameters for the numerical aspect of our computations are provided in Table 4.5. We also display the seasonal component along with five sample paths of the commodity price in Figures 4.2 and 4.3, respectively.

Results and Discussion

We present our results for pricing and sensitivities in 4.11, 4.12, and OEBs in Figures 4.7 - 4.9.

From Table 4.7, we see the usual consistency in our price estimates. This $2d + 2d$ example has relatively high run times in comparison to the multi-dimensional Heston model in Chapter 3 seeing as our exercise frequency is only $T/6$ as opposed to $T/12$. The slow down is due to the extra extrapolations carried out for each path when computing integrals of h as defined in (4.6).

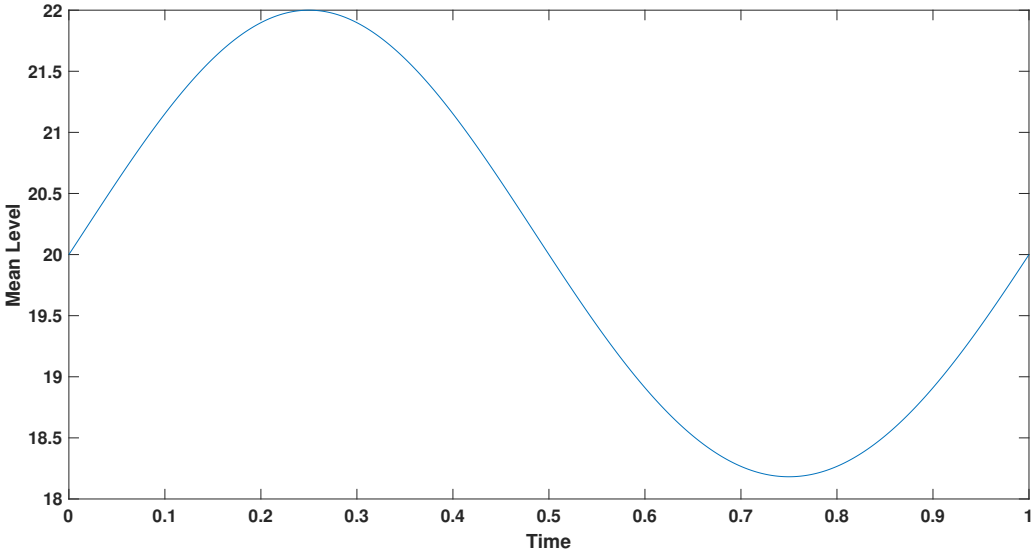

Figure 4.2: Mean level of the commodity price given by $\exp(\theta(t))$ which represents seasonal changes in the price.

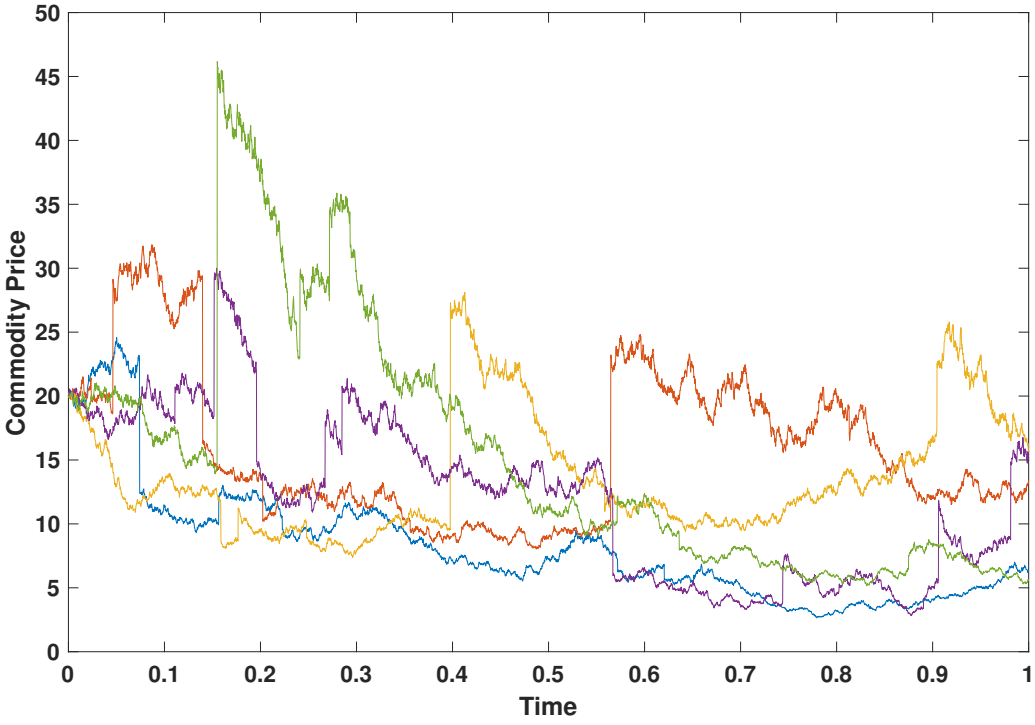

Figure 4.3: Five sample paths of $S(t)$ on the time interval $[0, 1]$

T	K	Exercise Frequency	r
1	20	$T/6$	0.025

κ_X	κ_Y	p	a^+	a^-	λ
1.2	3.7	0.45	0.3	0.25	10

$v_0^{(1)}$	κ_1	θ_1	η_1	$v_0^{(2)}$	κ_2	θ_2	η_2
0.03	0.4	0.3	0.3	0.21	3.86	0.21	0.85

$$[\rho_{i,j}] = \begin{bmatrix} 1 & \rho_{W^{X1},W^{X2}} & \rho_{W^{X1},W^{v1}} & \rho_{W^{X1},W^{v2}} \\ \rho_{W^{X2},W^{X1}} & 1 & \rho_{W^{X2},W^{v1}} & \rho_{W^{X2},W^{v2}} \\ \rho_{W^{v1},W^{X1}} & \rho_{W^{v3},W^{X2}} & 1 & \rho_{W^{v1},W^{v2}} \\ \rho_{W^{v2},W^{X1}} & \rho_{W^{v2},W^{X2}} & \rho_{W^{v2},W^{v1}} & 1 \end{bmatrix} = \begin{bmatrix} 1 & -0.3 & -0.25 & 0.2 \\ -0.3 & 1 & 0.3 & -0.4 \\ -0.25 & 0.3 & 1 & 0.1 \\ 0.2 & -0.4 & 0.1 & 1 \end{bmatrix}$$

$$\theta(t) = \log(1.1)\sin(2\pi t) + \log(20)$$

Table 4.5: parameters used in our option pricing model.

N_{trial}	N_{step}	$\deg_{mix}$	L_{mix}	$(\log S^{(i)}/S_0^{(i)})_{min}$	$(\log S^{(i)}/S_0^{(i)})_{max}$	N_{cl}
100	1 000	4	10	-3	3	4 500

Table 4.6: parameters fixed when testing the GPSM for price and sensitivity computations.

Sub-trial Type	N_1^S
a	2^7
b	2^8
c	2^9

Table 4.7: Subtrial settings for pricing using the prml-GPSM indicating the grid resolutions for the higher levels. We keep $N_0^S = 2^6, N_{sim,0}^v = 10\ 000, N_{sim,1}^v = 100, N_{cl} = 4\ 500$ fixed throughout subtrials. These settings have no effect on the computation of sensitivities.

Level	0	1	2	3
N_k^v	50 000	500	250	100
N_k^S	2^5	2^6	2^7	2^8

Table 4.8: mlMC settings used for the computation of sensitivities. Only a single type of trial is used for sensitivities unlike pricing.

The sensitivities provided in Table 4.12, show strong results for gridded sensitivities in the direct estimator, with higher variance for the low estimator, especially for the jump component, Y. We also see that $\partial_{YY} V_{0,d}^N, \partial_{YY} V_{0,l}^N$ differ considerably, unlike in other examples. These discrepancies are due to the presence of the initial dispersion. Sensitivities with respect to the variable v behave similar to other examples where $d_v > 1$, and mixed sensitivities involving X and v_i are fairly accurate. Unfortunately, we see mixed sensitivities with respect to Y and v_i are not very accurate, similar to other Y-sensitivities using the low estimator and initial dispersion. An in-depth analysis

of this phenomenon, however, lies beyond the scope of this study.

Fortunately, we see the prml-GPSM is able to consistently locate the exercise boundaries across volatility slices and exercise dates as seen in Figures 4.7 - 4.9. These boundaries are of higher quality than those shown for the multi-dimensional Heston model Chapter 3.

4.A Pricing Statistics

Pricing and Sensitivity Results for the Double Heston Model

$N_{1,i}^{\mathcal{S}}$	Estimate Type	Mean Price	Std Dev	Run Time (s)
2^9	Direct	11.969	$6.246 \cdot 10^{-2}$	47.07
	Low	11.819	$1.091 \cdot 10^{-1}$	38.87
2^8	Direct	11.972	$6.236 \cdot 10^{-2}$	47.72
	Low	11.818	$1.105 \cdot 10^{-1}$	37.54
2^7	Direct	11.986	$6.231 \cdot 10^{-2}$	44.51
	Low	11.841	$1.113 \cdot 10^{-1}$	36.01

Table 4.9: Resulting price statistics for the ml-GPSM algorithm. We fix $N_0^{\mathcal{S}} = 2^6, N_{sim,0}^v = 50000, N_{sim,1}^v = 100$.

Estimated	$V_{0,d}^N$	$\partial_S V_{0,d}^N$	$\partial_{SS} V_{0,d}^N$
Mean	11.971	-0.3478	0.01351
Std Dev	$7.968 \cdot 10^{-2}$	$1.5755 \cdot 10^{-3}$	$7.5221 \cdot 10^{-5}$

Estimated	$V_{0,l}^N$	$\partial_S V_{0,l}^N$	$\partial_{SS} V_{0,l}^N$
Mean	11.841	-0.3439	0.01381
Std Dev	$2.9899 \cdot 10^{-1}$	$9.3165 \cdot 10^{-3}$	$1.0199 \cdot 10^{-3}$

Estimated	$\partial_{v_1} V_{0,l}^N$	$\partial_{v_2} V_{0,l}^N$	$\partial_{Sv_1} V_{0,l}^N$	$\partial_{Sv_2} V_{0,l}^N$
Mean	15.144	43.08	0.1366	-0.4199
Std Dev	3.2965	7.0346	$1.2790 \cdot 10^{-1}$	$1.9465 \cdot 10^{-1}$

Estimate Type	Run Time (s)
Direct	80.00
Modified Low	74.14

Table 4.10: Numerical results for computing sensitivities using the ml-GPSM with $T^* = 1$. $T = 1$ with 12 equally spaced exercise dates. We fix $N_0^{\mathcal{S}} = 2^5, N_1^{\mathcal{S}} = 2^8, N_{sim,0}^v = 100\ 000, N_{sim,1}^v = 500$.

Pricing and Sensitivity Results for the Commodity Spot Model

$N_{1,i}^{\mathcal{S}}$	Estimate Type	Mean Price	Std Dev	Run Time (s)
2^9	Direct	9.6196	$4.9821 \cdot 10^{-3}$	114.61
	Low	9.6161	$5.2557 \cdot 10^{-3}$	124.21
2^8	Direct	9.6222	$5.0093 \cdot 10^{-3}$	63.00
	Low	9.6186	$5.2157 \cdot 10^{-3}$	88.04
2^7	Direct	9.6284	$5.0339 \cdot 10^{-3}$	51.99
	Low	9.6251	$5.2443 \cdot 10^{-3}$	74.89

Table 4.11: Resulting price statistics for the ml-GPSM algorithm. We fix $N_0^{\mathcal{S}} = 2^6, N_{sim,0}^v = 10000, N_{sim,1}^v = 100$, and $N_{cl} = 4500$.

Estimated	$V_{0,d}^N$	$\partial_X V_{0,d}^N$	$\partial_{XX} V_{0,d}^N$	$\partial_Y V_{0,d}^N$	$\partial_{YY} V_{0,d}^N$	$\partial_{XY} V_{0,d}^N$
Mean	9.6199	-3.6406	0.1822	-0.7706	1.4409	0.5244
Std Dev	$2.0330 \cdot 10^{-3}$	$9.7139 \cdot 10^{-4}$	$6.0315 \cdot 10^{-4}$	$6.9026 \cdot 10^{-4}$	$6.3511 \cdot 10^{-2}$	$6.2546 \cdot 10^{-4}$

Estimated	$V_{0,l}^N$	$\partial_X V_{0,l}^N$	$\partial_{XX} V_{0,l}^N$	$\partial_Y V_{0,l}^N$	$\partial_{YY} V_{0,l}^N$	$\partial_{XY} V_{0,l}^N$
Mean	9.6191	-3.6406	0.1825	-0.7710	1.3722	0.5246
Std Dev	$4.5448 \cdot 10^{-3}$	$2.4648 \cdot 10^{-3}$	$3.0841 \cdot 10^{-3}$	$1.3161 \cdot 10^{-2}$	$9.7223 \cdot 10^{-2}$	$3.0313 \cdot 10^{-3}$

Estimated	$\partial_{v_1} V_{0,l}^N$	$\partial_{v_2} V_{0,l}^N$	$\partial_{v_1 X} V_{0,l}^N$	$\partial_{v_1 Y} V_{0,l}^N$	$\partial_{v_2 X} V_{0,l}^N$	$\partial_{v_2 Y} V_{0,l}^N$
Mean	0.2045	0.03610	0.5431	-0.001768	0.1281	0.005875
Std Dev	$4.8380 \cdot 10^{-2}$	$3.8423 \cdot 10^{-2}$	$2.4009 \cdot 10^{-2}$	$1.2338 \cdot 10^{-1}$	$2.0699 \cdot 10^{-2}$	$1.0373 \cdot 10^{-1}$

Estimate Type	Run Time (s)
Direct	215.79
Modified Low	215.90

Table 4.12: Numerical results for computing sensitivities using the ml-GPSM with $T^* = 1$. $T = 1$ with 6 equally spaced exercise dates. The top table gives the settings for our multi-level computation.

4.B Optimal Exercise Boundaries

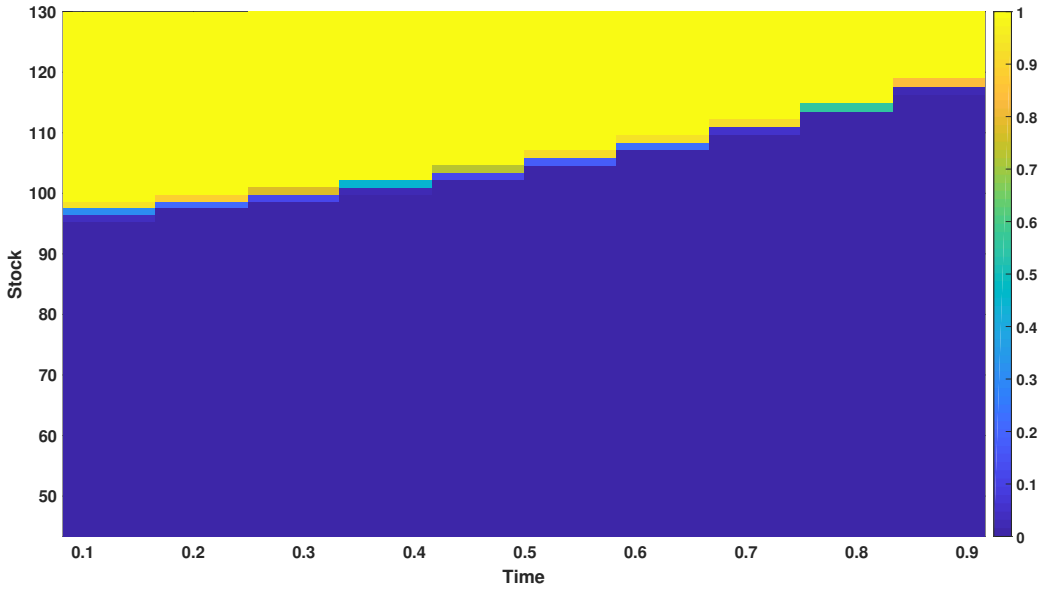

Figure 4.4: A slice of the GPSM OEB with $v_1 = 0.5 \cdot \theta_{v_1}$ and $v_2 = 1.5 \cdot \theta_{v_2}$ for the double Heston model. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^9$.

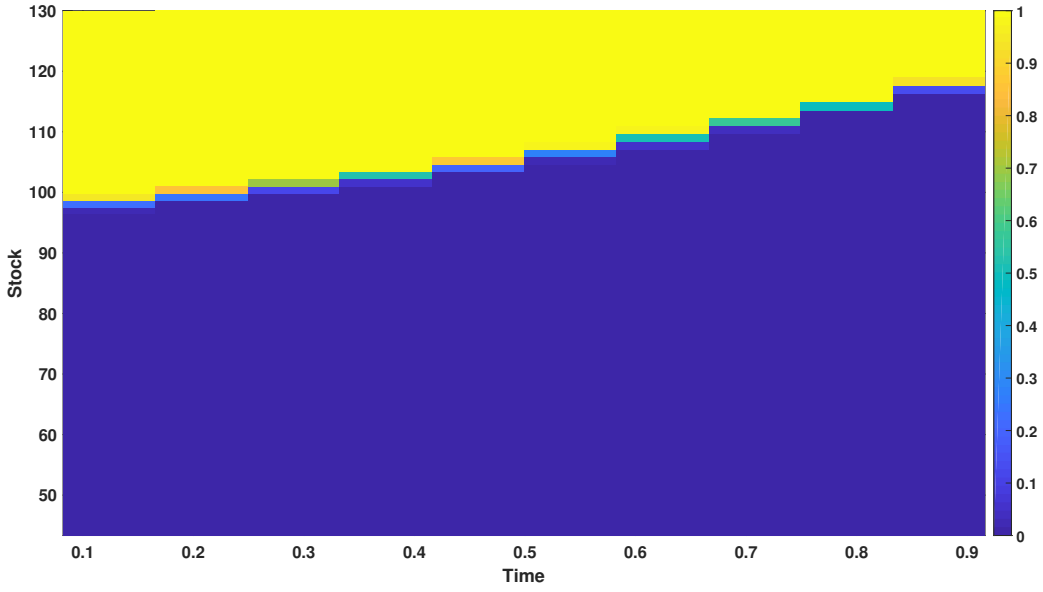

Figure 4.5: A slice of the GPSM OEB with $v_1 = \theta_{v_1}$ and $v_2 = \theta_{v_2}$ for the double Heston model. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^9$.

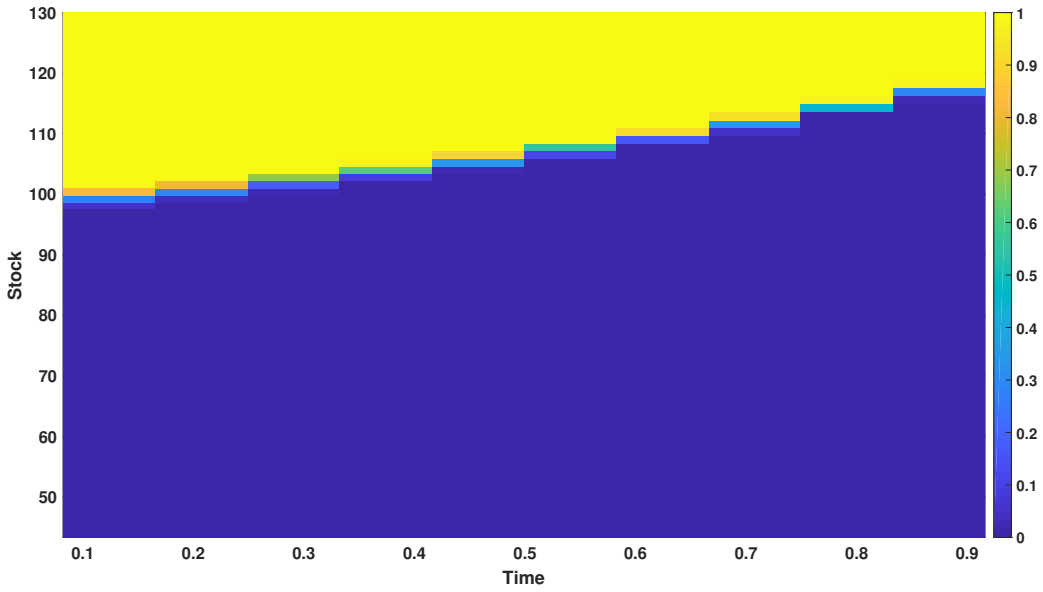

Figure 4.6: A slice of the GPSM OEB with $v_1 = 1.5 \cdot \theta_{v_1}$ and $v_2 = 0.5 \cdot \theta_{v_2}$ for the double Heston model. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^9$.

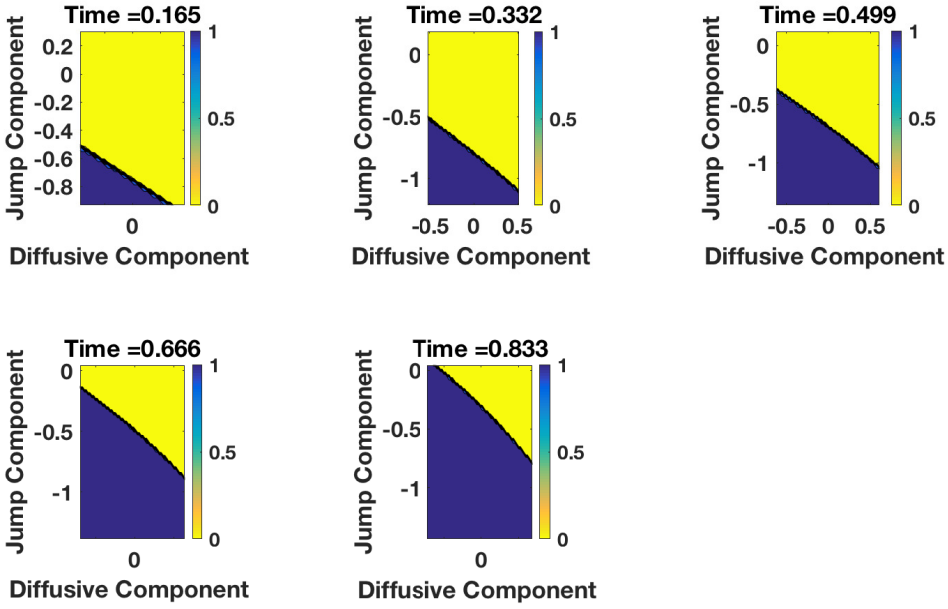

Figure 4.7: A slice of the GPSM OEB with $v_1 = 0.5 \cdot \theta_{v_1}$ and $v_2 = 1.5 \cdot \theta_{v_2}$ for the commodity spot model. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^8$.

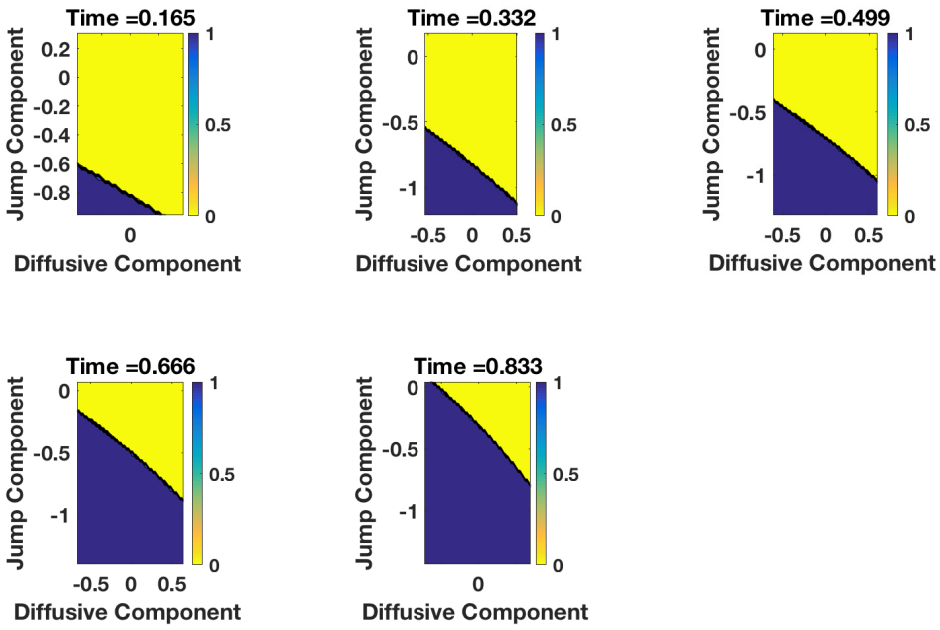

Figure 4.8: A slice of the GPSM OEB with $v_1 = \theta_{v_1}$ and $v_2 = \theta_{v_2}$ for the commodity spot model. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^8$.

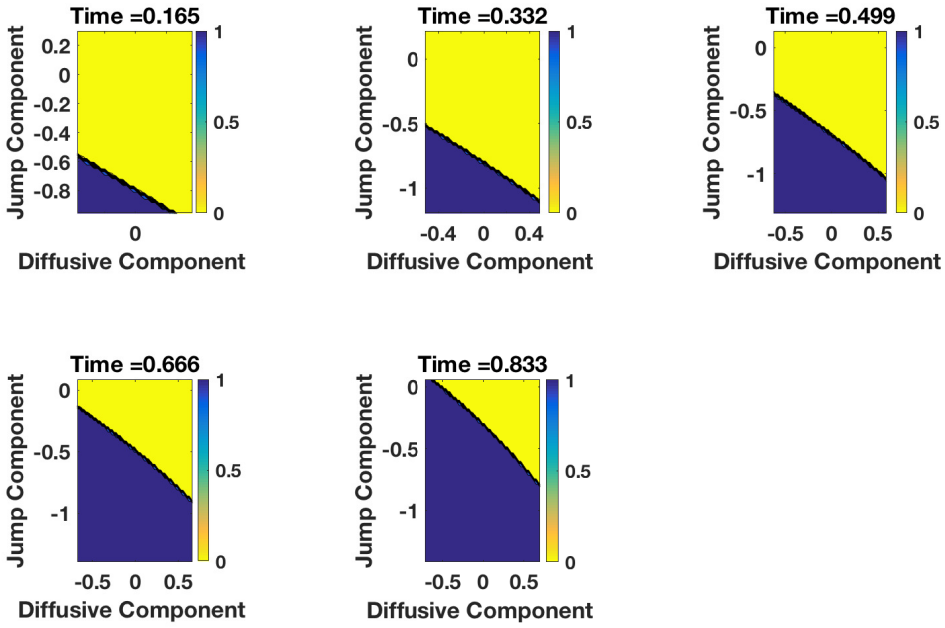

Figure 4.9: A slice of the GPSM OEB with $v_1 = 1.5 \cdot \theta_{v_1}$ and $v_2 = 0.5 \cdot \theta_{v_2}$ for the commodity spot model. Dark blue regions indicate exercising with probability 1, yellow regions indicate holding with probability 1. This computation corresponds to a grid resolution of $N_1^{\mathcal{S}} = 2^8$.

Chapter 5

Conclusions

5.1 Summary of Contributions

In this thesis, we developed a numerical approach for solving multi-dimensional discrete time optimal stopping problems under SV that combines LSMC with PDE techniques. The algorithm provides dimensional reduction from the PDE and regression perspective along with variance and dimensional reduction from the MC perspective.

In Chapter 2, we began by making rigorous the mathematical foundation for mixed MC-PDE techniques. Next, we provided the basic mechanics of the algorithm and, under certain mild assumptions, proved it converges almost surely using methods from point-set topology. Afterwards, we applied the algorithm to the one dimensional Heston model and showed that the hybrid algorithm outperforms traditional LSMC techniques in terms of estimating prices and optimal exercise boundaries.

In Chapter 3, we described methods for reducing the complexity and run time of the algorithm along with a technique for computing sensitivities. To reduce the complexity, we applied two methods: clustering via sufficient statistics and mlMC. While the clustering method allows us to reduce computational run times by a factor of a half for high dimensional problems, mlMC provides an order of magnitude reduction in complexity. In order to compute sensitivities, we employ a grid based method for derivatives with respect to the asset, S, and the MC based method of [64] using initial dispersions for sensitivities with respect to v. We find that S-sensitivities are highly accurate, whereas v-sensitivities are noisy when $d_v > 1$. To test our approximations and computation of sensitivities we revisited the one dimensional Heston model and found our approximations introduced little-to-no error and that our computation of sensitivities were highly accurate in comparison to the reference values and standard LSMC. To demonstrate the utility of our new computational techniques, we apply the hybrid algorithm to the multi-dimensional Heston model and show that the algorithm is highly accurate in terms of estimating prices, OEBs, and sensitivities, especially in comparison to standard LSMC.

In Chapter 4, we highlight the importance of multi-factor SV models and apply our hybrid algorithm to two specific examples: the Double Heston model and a mean-reverting commodity model with jumps. Again, we were able to obtain low variance estimates of the prices, OEBs, and

sensitivities with respect to S, along with reasonable estimates of sensitivities with respect to v.

5.2 Future Work

Modified Simulation Frameworks

An interesting framework that remains to be tested are Bermudan style options written on futures prices or interest rate swaps on models with SV and one-way coupling. Such frameworks have an exercise function that depends on the simulated process v and not just the underlying asset or interest rate. It would be interesting to see how the algorithm applies to this class of problems along with extensive benchmarking and backtesting.

Throughout this thesis we focused on SV models with one-way coupling, $X_t = (S_t, v_t)$, simulated paths of v_t in $\mathbb{R}^{d_v}$ and solved conditional PDEs over $\mathbb{R}^{d_S}$. When solving PDEs over $\mathbb{R}^{d_S}$ we work with a grid, $\mathcal{S}$, that provides stability in terms of locating the boundary and computing sensitivities with respect to S. The idea of treating some variables on a grid and simulating others may potentially be useful in the context of basket options written on a multi-dimensional GBM. For instance, we may consider the following model

$$\frac{dS_t^{(i)}}{S_t^{(i)}} = r\ dt + \sigma^{(i)}\ dW_t^{(i)}$$

for $i \in \{1, \ldots, d\}$ where W_t is a d-dimensional Brownian motion with mean zero and correlation matrix ρ. Upon switching to log-space variables and applying a Cholesky decomposition, one may observe that this model also has one-way coupling. It may be useful to apply the GPSM to a Bermudan style option written on this system where one simulates $d-2$ or $d-3$ of the processes $S^{(i)}$ and solves conditional PDEs over the remaining two or three variables. It may be that, based on the values of $\sigma^{(i)}$ and ρ, that it is advantageous to integrate over certain stocks and simulate others. In this case, our exercise function $h(S)$ will, similar to the case described above, depend on the simulated processes $\{S^{(i)}\}_{i=1}^{d-2}$ which could potentially compromise results, although this remains to be tested.

Optimized Implementations

An issue not addressed in this thesis is ways in which the hybrid algorithm may be optimized. Throughout this thesis, we encountered one and two dimensional conditional PDEs that were generated via Feynmann-Kac for Levy processes and solved them using Fourier Space Time-stepping, an FFT technique. While FFT methods have complexity $O(N^d\ \log\ N)$, methods such as the FST or CONV [48] require N to be relatively large. Also, the presence of numerically stored functions hinders the efficiency of the 2d-COS method [57]. One possibility could be to return to the models presented in this thesis and apply a highly optimized finite difference scheme that is second order in space and time, using a scheme which requires few points in $\mathcal{S}$ to obtain a comparable bias. Such an algorithm would benefit from the multi-index MC schemes described in [26]. Furthermore, one can implement the algorithm on GPUs to exploit the independence of the conditional PDEs

we solve.

As described in Chapter 3, it still remains to develop a scheme for optimally allocating paths to each level for mlMC applied to the direct estimator. An important topic not discussed in this thesis this is computational complexity that results from applying mlMC. Future investigations will involve fixing a model, implementing a highly efficient PDE solver and analyzing, both theoretically and numerically, the savings brought about by mlMC using the celebrated Multi-Level Monto-Carlo Theorem [26]. Some partial results on this front may be found in [23].

There is also the question of adaptive approaches to clustering which remains to be developed and tested.

Rates of Convergence, Variance Reduction Analysis, and Initial Dispersions

In Chapter 2 we proved Theorem 4 which showed that under certain conditions the hybrid algorithm's coefficients converge almost surely to the idealized coefficients. Our proof borrowed techniques from the works of [12] and [62] and generalized them using the Stone-Weierstrass Theorem. While [62] only discusses almost sure convergence of their algorithm, the work of [12] also proves a Central Limit Theorem (CLT) result and thus establishes a rate of convergence for the LS version of LSMC. The analogous result remains to be proven for our algorithm, and it seems the methods we have developed will not transfer.

Based on [12], the starting point for proving the CLT for our algorithm is to prove the following weak-type estimate:

Proposition 2 (conjecture)**.** *Given certain truncation and separability conditions, let* $n \in \{1, \dots, M-1\}$*,* $S \in \mathcal{S}$*, and* $\delta > 0$*. There exists a constant* $C_n(S) > 0$ *such that*

$$\mathbb{Q}'\left(\ |a_n^N(S) - a_n(S)| \geq \delta \right) \leq \frac{C_n(S)}{\delta^4 N^2}.$$

At a first glance, one may hope to obtain the above conjecture via Proposition 1. However, the lack of control over the bounding term, $\varepsilon \cdot c_n^N$, makes this infeasible. One needs to be more careful when constructing c_n^N and must take the rate at which it decreases into account. When applying SW, we have a function $h(S, R)$ that is compactly supported and continuous on its support and we obtain a function of the form

$$\psi_n^N(S, R) = \sum_{i=1}^{k_n^N} \psi_{N,n,i}(S)\psi_{N,n,i}(R) \tag{5.1}$$

such that $||h - \psi_n^N||_\infty < \varepsilon_N$. At the next step of our induction, we approximate each $\psi_{N,n,i}$ by another sum of separable functions that is ε_N close to $\psi_{N,n,i}$. Since SW is an existence theorem, it seems we cannot control the number of separable functions k_n^N in our approximation which interferes with obtaining a rate. Presumably, as $\varepsilon_N \to 0$, we have $k_n^N \to \infty$ although it is not clear how this relationship behaves. As a result, we likely need an approach for carrying out the

separating approximations that uses an integral as opposed to a discrete sum. For instance, we *may* be able to construct an approximation to the identity of the following form

$$\psi_n^N(S,R) = \int_{\mathbb{R}^{2d_S}} \phi_{S,N}(S-S')\phi_{R,N}(R-R')h(S',R')dS'dR'$$

where $\phi_{S,N}, \psi_{R,N}$ are compactly supported and smooth on $\mathbb{R}^{d_S}$ such that $\psi_n^N(S,R) \to h(S,R)$ uniformly on compact sets of $\mathbb{R}^{2d_S}$. Such a family of functions has the same separability properties as that of (5.1) and may ultimately allow us to prove Proposition 2, although there may be other complications. Lastly, to establish the CLT, one may use the L^2 martingale technique presented in the latter part of [12] along with Proposition 2.

Another topic of interest is quantifying the variance reduction that is obtained in computing time-zero values, OEBs and sensitivities. A common observation is that working with our PDE grid, $\mathcal{S}$, provides a certain form of "stability" for most of our estimated quantities. Ultimately, one must compute the variances of the estimated quantities from the LSM and GPSM and compare their ratios. On a related topic, the variances of our estimator for $\partial_v V_{0,l}(S,v)$ when $d_v > 1$ remains difficult to explain and warrants future study. It may be that the choice of initial dispersion and regression method are not suited for the correlation structure of $v(t)$.

Optimal Switching Problems

Optimal switching problems are a natural generalization of optimal stopping problems where instead of deciding on whether to hold or stop at each time stage, the agent may alternate between different payoff regimes with constraints on how many times they may switch. These problems arise in the context of energy production and consumption where an electricity company may hedge their risks to changes in demand or price of their commodity. An important example of optimal switching problems are swing options. The reader may consult [8] for an in-depth discussion of swing options as a stochastic control problem along with a discrete dynamic programming principle (DPP) and numerical solution via LSMC. In terms of PDE approaches, we refer the reader to [38].

As one may expect, it is possible to extend the GPSM to such problems as the DPP reduces their solution to the evaluation of conditional expectations that are analogous to the optimal stopping case.

Appendix A

Mathematical and Computational Background

In this chapter we review various mathematical and computational methods that are used throughout this thesis.

A.1 Probability Theory and Point Set Topology

Kolmogorov Extension Theorem

In this thesis, we use the Kolmogorov Extension Theorem (KET) in Chapter 2 to construct the "inherited sampling probability space" $(\Omega', \mathcal{F}', \mathbb{Q}')$ from the original space $(\Omega, \mathcal{F}, \mathbb{Q})$. The sampling space allows us to makes sense of an iid collection of paths, $\{[v^j]\}_{j=1}^{\infty}$.

The need for the KET is often motivated by the following example: When studying general probability spaces we make statements such as "Let $\{X_i\}_{i=1}^{\infty}$ be independent random variables on $\mathbb{R}$ with distribution $\{\nu_i\}_{i=1}^{\infty}$, such that ... ." The reader may then wonder if there even exists a space that may accommodate such random variables with these distributions. The KET answers this question affirmatively assuming certain consistency conditions are met. The following results are taken from [18].

We begin by letting

$$\mathbb{R}^{\mathbb{N}} = \{(\omega_1, \omega_2, \ldots) : \omega_i \in \mathbb{R}\}$$

and let $R^{\mathbb{N}}$ be the σ-algebra on $\mathbb{R}^{\mathbb{N}}$ generated by finite dimensional rectangles.

Theorem 7 (Kolmogorov's Extension Theorem)**.** *Suppose we are given probability measures μ_n on $\mathbb{R}^n$ that are consistent, that is,*

$$\mu_{n+1}((a_1, b_1] \times \ldots \times (a_n, b_n] \times \mathbb{R}) = \mu_n((a_1, b_1] \times \ldots \times (a_n, b_n])$$

Then there is a unique probability measure P on $(\mathbb{R}^{\mathbb{N}}, R^{\mathbb{N}})$ with

$$P\left(\{\omega \ \mid \ \omega_i \in (a_i, b_i], 1 \leq i \leq n\}\right) = \mu_n((a_1, b_1] \times \ldots \times (a_n, b_n]).$$

To see how the result answers the concern we raised above, we first set $\Omega' = \mathbb{R}^{\mathbb{N}}$ and $\mathcal{F}' = R^{\mathbb{N}}$. Next, we consider the measures μ_n on $\mathbb{R}^n$ defined via

$$\mu_n((a_1, b_1] \times \ldots \times (a_n, b_n]) = \prod_{i=1}^{n} \nu_i((a_i, b_i])$$

and note that, because they are consistent, by the KET, they induce a measure $\mathbb{Q}'$ on Ω'. Finally, we define our random variables $X_n : \Omega' \to \mathbb{R}$ via $X_n(\omega) = \omega_n$ and note that they are independent and have distribution ν_n.

The above construction is specific to random variables taking values in $\mathbb{R}$, however, in Chapter 2 our construction is for random vectors in $\mathbb{R}^{d_\Theta}$. For the analogous statement and proof on more general spaces than $\mathbb{R}$, we refer the reader to [31].

Stone-Weierstrass Theorem

The Stone-Weierstrass Theorem (SWT) plays a fundamental role in our proof of Theorem 4. While we only require the result on spaces like $\mathbb{R}^d$, we state it in a more general context. The following is taken from [25].

Let X be a non-compact locally compact, Hausdorff space, and $C^0(X)$ be the space of continuous functions on X that vanish at infinity, endowed with the uniform metric. The Stone-Weierstrass Theorem is a result that gives sufficient conditions for a collection of functions, $\mathcal{A} \subset C^0(X)$, to be dense. We say $\mathcal{A}$ is an *algebra of functions* if $\mathcal{A}$ is a vector space that is also closed under point-wise multiplication, $\mathcal{A}$ *vanishes at a point*, x, if $f(x) = 0$ for all $f \in \mathcal{A}$ and $\mathcal{A}$ *separates points* if for $x, y \in K$, there exists $f \in \mathcal{A}$ such that $f(x) \neq f(y)$. The SWT for says the following

Theorem 8. *If X is a non-compact, locally compact Hausdorff space and $\mathcal{A} \subset C^0(K)$ is an algebra that vanishes nowhere and separates points then $\mathcal{A}$ is dense in $C(K)$.*

Tietze-Extension Theorem and Urysohn's Lemma

Some other topological results that we require for our proof of Theorem 4 are the Tietze Extension Theorem and Urysohn's Lemma. While the two results are equivalent, we present them in the order that they are usually developed. The theorems are viewed as powerful existence results that allow one to extend a continuous map defined on a closed subset of a space X, where X may be a fairly general topological space. The following is adapted from [50], stated on a metric space X.

Theorem 9 (Urysohn's Lemma)**.** *Let X be a metric space; let A and B be disjoint closed subsets of X. Let $[a, b]$ be a closed interval in the real line. Then there exists a continuous map*

$$f : X \to [a, b]$$

such that $f(x) = a$ for every x in A, and $f(x) = b$ for every x in B.

Theorem 10 (Tietze Extension Theorem)**.** *Let X be metric space; let A be a closed subspace of X. Any continuous map of A into the closed interval $[a, b]$ of $\mathbb{R}$ may be extended to a continuous map of all of X into $[a, b]$.*

A.2 Fourier-Space Time Stepping

Throughout this thesis, using the Feynmann-Kac theorem, we derived conditional partial (integro)-differential equations (PIDEs) generated from stochastic processes driven by Levy processes. To solve such conditional PIDEs, we resorted to a numerical Fourier transform method known as Fourier Space Time-stepping as developed in [59] and [38]. In this section, we summarize the basic mechanics of the algorithm. For notational simplicity, we state the standard FST algorithm for a one-dimensional Levy process.

Theoretical Set Up

Given a Levy process X_t defined on $[0,T]$ with triple (γ,σ,ν), we generate the following PIDE to compute $\mathbb{E}[h(X_{t_{n+1}}) \mid X_{t_n} = x]$ where $[t_n, t_{n+1}] \subset [0,T]$:

$$\begin{aligned}(\partial_t + \mathcal{L})g(t,x) &= 0\\ g(t_{n+1},x) &= h(x)\end{aligned}$$

and

$$\mathcal{L}g(x) = \left(\gamma\partial_x + \frac{1}{2}\sigma\partial_x^2\right)g(x) + \int_{\mathbb{R}\setminus\{0\}} \left(g(x+y) - g(y) - y\partial_x f(x) I_{|y|<1}\right) d\nu(y).$$

To solve the PIDE, we take continuous Fourier transforms and, for each $\omega \in \mathbb{R}$, obtain the following ODE in Fourier space

$$\begin{aligned}\partial_t \mathcal{F}[g](t,\omega) + \Psi(\omega)\mathcal{F}[g](t,\omega) &= 0\,,\\ \mathcal{F}[g](t_{n+1},\omega) &= \mathcal{F}[h](\omega)\,.\end{aligned}$$

where

$$\Psi(\omega) = i\gamma\omega - \frac{1}{2}\sigma\omega^2 + \int_{\mathbb{R}} \left(e^{i\omega y} - 1 - iy\omega I_{|y|<1}\right) d\nu(y)$$

For each $\omega \in \mathbb{R}$, we solve the ODE and invert back to real space to find

$$g(t_n,x) = \mathcal{F}^{-1}\left[\mathcal{F}[g](t_{n+1},\omega)e^{\Psi(\omega)(t_{n+1}-t_n)}\right](x) \tag{A.1}$$

Numerical Approximation

We now turn to numerically computing $g(t_n,x)$ via the formula (A.1). We begin by setting a domain in real space $\Omega = [t_n, t_{n+1}] \times [x_{min}, x_{max}]$ and partition the spatial component as $x_n = x_{min} + n\Delta x$ where $\Delta x = (x_{max} - x_{min})/(N-1)$ and $n \in \{0,\dots,N-1\}$. Next, we consider our Fourier domain $\widehat{\Omega} = [t_n, t_{n+1}] \times [\omega_{min}, \omega_{max}]$ where we set $\omega_{max} = \pi/\Delta x$, $\omega_{min} = -\omega_{max}$ and discretize via $\omega_n = \omega_{min} + n\Delta\omega$ where $n \in \{1,\dots,N\}$ and $\Delta\omega = 2\omega_{max}/N$. It may then be shown

that

$$g_n = \mathrm{FFT}^{-1}\left[\mathrm{FFT}[g_{n+1}]e^{\Psi\Delta t}\right]$$

where g_n, g_{n+1} are restictions of g to our real-space discretization at times t_n, t_{n+1}, and Ψ is our characteristic exponent restricted to our Fourier space discretization.

Bibliography

[1] Farid AitSahlia, Manisha Goswami, and Suchandan Guha. American option pricing under stochastic volatility: an efficient numerical approach. *Computational Management Science*, 7(2):171–187, 2010.

[2] Leif BG Andersen and Vladimir V Piterbarg. Moment explosions in stochastic volatility models. *Finance and Stochastics*, 11(1):29–50, 2007.

[3] Xing Xian Ang. A mixed PDE/Monte Carlo approach as an efficient way to price under high-dimensional systems. Master's thesis, University of Oxford, 2013.

[4] Christian Bender and Jessica Steiner. Least-squares Monte-Carlo for backward sdes. In *Numerical methods in finance*, pages 257–289. Springer, 2012.

[5] Bruno Bouchard and Xavier Warin. Monte-Carlo valuation of American options: facts and new algorithms to improve existing methods. In *Numerical methods in finance*, pages 215–255. Springer, 2012.

[6] Mark Broadie, Paul Glasserman, et al. A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance*, 7:35–72, 2004.

[7] Luca Capriotti, Yupeng Jiang, and Andrea Macrina. AAD and least-square Monte Carlo: Fast Bermudan-style options and XVA Greeks. *Algorithmic Finance*, 6(1-2):35–49, 2017.

[8] René Carmona and Michael Ludkovski. Valuation of energy storage: An optimal switching approach. *Quantitative finance*, 10(4):359–374, 2010.

[9] Jacques F Carriere. Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: mathematics and Economics*, 19(1):19–30, 1996.

[10] Álvaro Cartea, Sebastian Jaimungal, and Zhen Qin. Model uncertainty in commodity markets. *SIAM Journal on Financial Mathematics*, 7(1):1–33, 2016.

[11] Peter Christoffersen, Steven Heston, and Kris Jacobs. The shape and term structure of the index option smirk: Why multifactor stochastic volatility models work so well. *Management Science*, 55(12):1914–1932, 2009.

[12] Emmanuelle Clément, Damien Lamberton, and Philip Protter. An analysis of a least squares regression method for american option pricing. *Finance and Stochastics*, 6(4):449–471, 2002.

[13] Andrei Cozma and Christoph Reisinger. A mixed monte carlo and pde variance reduction method for foreign exchange options under the heston-cir model. *arXiv preprint arXiv:1509.01479*, 2015.

[14] Duy Minh Dang. *Modeling multi-factor financial derivatives by a Partial Differential Equation approach with efficient implementation on Graphics Processing Units*. PhD thesis, University of Toronto, 2011.

[15] Duy-Minh Dang, Kenneth R Jackson, and Mohammadreza Mohammadi. Dimension and variance reduction for Monte Carlo methods for high-dimensional models in finance. *Applied Mathematical Finance*, 22(6):522–552, 2015.

[16] Duy-Minh Dang, Kenneth R Jackson, and Scott Sues. A dimension and variance reduction Monte-Carlo method for option pricing under jump-diffusion models. *Applied Mathematical Finance*, 24(3):175–215, 2017.

[17] Daniel J Duffy. *Finite Difference methods in financial engineering: a Partial Differential Equation approach*. John Wiley & Sons, 2013.

[18] Rick Durrett. *Probability: theory and examples*. Cambridge university press, 2010.

[19] Daniel Egloff et al. Monte-Carlo algorithms for optimal stopping and statistical learning. *The Annals of Applied Probability*, 15(2):1396–1432, 2005.

[20] Daniel Egloff, Michael Kohler, Nebojsa Todorovic, et al. A dynamic look-ahead Monte-Carlo algorithm for pricing bermudan options. *The Annals of Applied Probability*, 17(4):1138–1171, 2007.

[21] Fang Fang and Cornelis W Oosterlee. A novel pricing method for European options based on Fourier-cosine series expansions. *SIAM Journal on Scientific Computing*, 31(2):826–848, 2008.

[22] Fang Fang and Cornelis W Oosterlee. A Fourier-based valuation method for Bermudan and barrier options under Heston's model. *SIAM Journal on Financial Mathematics*, 2(1):439–463, 2011.

[23] David Farahany, Kenneth Jackson, and Sebastian Jaimungal. Mixing LSMC and PDE methods to price Bermudan options. *arXiv preprint arXiv:1803.07216*, 2018.

[24] Qian Feng and CW Oosterlee. Monte-Carlo calculation of exposure profiles and Greeks for Bermudan and barrier options under the Heston Hull-White model. *arXiv preprint arXiv:1412.3623*, 2014.

[25] Gerald B Folland. *Real analysis: modern techniques and their applications*. John Wiley & Sons, 2013.

[26] Michael B Giles. Multilevel monte carlo methods. *Acta Numerica*, 24:259–328, 2015.

[27] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media, 2013.

[28] Paul Glasserman and Bin Yu. Simulation for American options: Regression now or regression later? In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 213–226. Springer, 2004.

[29] Paul Glasserman, Bin Yu, et al. Number of paths versus number of basis functions in American option pricing. *The Annals of Applied Probability*, 14(4):2090–2119, 2004.

[30] Martino Grasselli. The 4/2 stochastic volatility model: a unified approach for the Heston and the 3/2 model. *Mathematical Finance*, 27(4):1013–1034, 2017.

[31] A Hitchhiker's Guide. *Infinite dimensional analysis*. Springer, 2006.

[32] Martin B Haugh and Leonid Kogan. Pricing American options: a duality approach. *Operations Research*, 52(2):258–270, 2004.

[33] Steven L Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2):327–343, 1993.

[34] Y Hilpisch. Fast Monte-Carlo valuation of American options under stochastic volatility and interest rates. 2009.

[35] John Hull and Alan White. The pricing of options on assets with stochastic volatilities. *The Journal of Finance*, 42(2):281–300, 1987.

[36] Samuli Ikonen and Jari Toivanen. Efficient numerical methods for pricing American options under stochastic volatility. *Numerical Methods for Partial Differential Equations*, 24(1):104–126, 2008.

[37] Sebastian Jaimungal, Max O de Souza, and Jorge P Zubelli. Real option pricing with mean-reverting investment and project value. *The European Journal of Finance*, 19(7-8):625–644, 2013.

[38] Sebastian Jaimungal and Vladimir Surkov. Lévy-based cross-commodity models and derivative valuation. *SIAM Journal on Financial Mathematics*, 2(1):464–487, 2011.

[39] Shashi Jain, Alvaro Leitao Rodriguez, and Cornelis Oosterlee. Rolling adjoints: Fast greeks along Monte-Carlo scenarios for early-exercise options. 2017.

[40] Shashi Jain and Cornelis W Oosterlee. The stochastic grid bundling method: Efficient pricing of Bermudan options and their Greeks. *Applied Mathematics and Computation*, 269:412–431, 2015.

[41] Steven G Kou. A jump-diffusion model for option pricing. *Management science*, 48(8):1086–1101, 2002.

[42] Pascal Letourneau and Lars Stentoft. Improved Greeks for American options using simulation. 2016.

[43] Tim Leung and Ronnie Sircar. Accounting for risk aversion, vesting, job termination risk and multiple exercises in valuation of employee stock options. *Mathematical Finance*, 19(1):99–128, 2009.

[44] Alan Lewis. The mixing approach to stochastic volatility and jump models. *Wilmott*, 2002.

[45] Tobias Lipp, Grégoire Loeper, and Olivier Pironneau. Mixing Monte-Carlo and partial differential equations for pricing options. In *Partial Differential Equations: Theory, Control and Approximation*, pages 323–347. Springer, 2014.

[46] Grégoire Loeper and Olivier Pironneau. A mixed PDE/Monte-Carlo method for stochastic volatility models. *Comptes Rendus Mathematique*, 347(9-10):559–563, 2009.

[47] Francis A Longstaff and Eduardo S Schwartz. Valuing American options by simulation: a simple least-squares approach. *The review of financial studies*, 14(1):113–147, 2001.

[48] Roger Lord, Fang Fang, Frank Bervoets, and Cornelis W Oosterlee. A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM Journal on Scientific Computing*, 30(4):1678–1705, 2008.

[49] Michael Ludkovski. Kriging metamodels and experimental design for Bermudan option pricing. *arXiv preprint arXiv:1509.02179*, 2015.

[50] James R Munkres. *Topology*. Prentice Hall, 2000.

[51] Goran Peskir and Albert Shiryaev. *Optimal stopping and free-boundary problems*. Springer, 2006.

[52] Huyên Pham. *Continuous-time stochastic control and optimization with financial applications*, volume 61. Springer Science & Business Media, 2009.

[53] Bhojnarine R Rambharat, Anthony E Brockwell, et al. Sequential Monte Carlo pricing of American-style options under stochastic volatility models. *The Annals of Applied Statistics*, 4(1):222–265, 2010.

[54] Christoph Reisinger and Rasmus Wissmann. Finite difference methods for medium-and high-dimensional derivative pricing PDEs. *High-Performance Computing in Finance: Problems, Methods, and Solutions, edited by E. Vynckier, J. Kanniainen and J. Keane. Chapman and Hall/CRC, London*, 2017.

[55] Leonard CG Rogers. Monte Carlo valuation of american options. *Mathematical Finance*, 12(3):271–286, 2002.

[56] Fabrice D Rouah. *The Heston Model and Its Extensions in Matlab and C*. John Wiley & Sons, 2013.

[57] Marjon J Ruijter and Cornelis W Oosterlee. Two-dimensional Fourier cosine series expansion method for pricing financial options. *SIAM Journal on Scientific Computing*, 34(5):B642–B671, 2012.

[58] Lars Stentoft. Convergence of the least squares Monte-Carlo approach to American option valuation. *Management Science*, 50(9):1193–1203, 2004.

[59] Vladimir Surkov. *Option pricing using Fourier space time-stepping framework*. PhD thesis, 2009.

[60] Nizar Touzi. *Optimal stochastic control, stochastic target problems, and backward SDE*, volume 29. Springer Science & Business Media, 2012.

[61] Anders B Trolle and Eduardo S Schwartz. Unspanned stochastic volatility and the pricing of commodity derivatives. *The Review of Financial Studies*, 22(11):4423–4461, 2009.

[62] John N Tsitsiklis and Benjamin Van Roy. Regression methods for pricing complex american-style options. *IEEE Transactions on Neural Networks*, 12(4):694–703, 2001.

[63] Binbin Wang. A study in hybrid Monte-Carlo methods in computing derivative prices. Master's thesis, University of Calgary, 2012.

[64] Yang Wang and Russel Caflisch. Pricing and hedging American-style options: a simple simulation-based approach. 2009.