Functional Quantization-based Stratified Sampling

Alexander Platts

A dissertation submitted to the Faculty of Commerce, University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Philosophy.

August 27, 2017

MPhil in Mathematical Finance,
University of Cape Town.

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Philosophy in the University of the Cape Town. It has not been submitted before for any degree or examination in any other University.

_____

August 27, 2017

Abstract

Functional quantization-based stratified sampling is a method for variance reduction proposed by Corlay and Pagès (2015). This method requires the ability to both create functional quantizers and to sample Brownian paths from the strata defined by the quantizers. We show that product quantizers are a suitable approximation of an optimal quantizer for the formation of functional quantizers. The notion of functional stratification is then extended to options written on multiple stocks and American options priced using the Longstaff-Schwartz method. To illustrate the gains in performance we focus on geometric brownian motion (GBM), constant elasticity of variance (CEV) and constant elasticity of variance with stochastic volatility (CEV-SV) models. The pricing algorithm is used to price knock-in, knock-out, autocall, call on the max and path dependent call on the max options.

Acknowledgements

I would like to that my supervisors for their help and support during my dissertation. I am also grateful to my friends and family for their advice, help proof reading and general support over the course of the year.

Finally, I would like to thank the lecturers who have allowed me an understanding of the work and an appreciation for the financial world.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Monte-Carlo simulations are by far the most commonly implemented integration techniques in the numerical probability field. This is partly due to the easy implementation and possibility for parallelisation (Corlay and Pagès, 2015). However, Monte-Carlo simulations can often be extremely computationally expensive. Several variance reduction techniques that attempt to reduce computational time (by increasing simulation accuracy) are antithetic and common random numbers, control variates, conditioning, stratified sampling, importance sampling, splitting, etc. (Kleijnen *et al.*, 2010). While many of these methods are highly effective, the dependence on the payoff function and altered implementation means that they are often only implemented in special cases (Corlay and Pagès, 2015).

Corlay and Pagès (2015) present a method for infinite dimensional stratification based on functional quantizers of Brownian motion. This method of stratification is intended as a generic variance reduction technique.

1.1 Stratification as a method of variance reduction

Stratified sampling refers to any sampling method where observations are drawn from non-overlapping subsets (strata) of the sample space. Traditionally it is assumed that most of the variance in an option price can be removed through terminal stratification, where only the terminal points of the Brownian path are stratified. Regarding the choice of strata, Corlay (2011); Corlay and Pagès (2015) describe a strong link between optimal quantization of a random variable and the variance reduction that can be achieved by the corresponding stratification. Corlay and Pagès (2015) describe the quantization of a random variable X as an approximation by a random variable Y that takes on finitely many values. Optimal quantization is then the choice of a Y such that $||X - Y||^2$ is minimized.

Vector quantization of random variables was first implemented as a method for signal discretization. The premise of this method was that one could design a

finitely valued random variable which approximated a signal with as little error as possible (Cuperman and Gersho, 1982). In later years quantization was included in numerical integration methods and multi-dimensional stochastic processes, such as American and Swing option pricing (Corlay and Pagès, 2015).

Corlay and Pagès (2015) provide a method for stratification where, as opposed to terminal stratification, the entire space is stratified. The stratification requires infinite dimensional quantizers of the relevant space. Luschgy and Pagès (2002) demonstrate that the problem of an optimal functional quantizer can be reduced to a function of the principal components of the random variable and an optimal quantizer of some finite dimensional random variable. This stratification can then be used as a generic variance reduction method for the pricing of path dependant derivatives.

1.2 Dissertation structure

This dissertation consists of four chapters, including this introduction. The second chapter provides an overview of the theory of functional quantization and functional quantization-based stratified sampling, as presented by Corlay and Pagès (2015). It then extends the notion of functional quantization-based stratified sampling to multi-factor models. Finally, it presents a method of American option pricing based on a stratified form of the Longstaff and Schwartz (2001) method. In Chapter Three the implementation of the theory is validated against results presented in Corlay *et al.* (2005) and Corlay and Pagès (2015). The efficiency of the functional quantization is investigated using several sampling methods. This is done for the three options presented in Corlay and Pagès (2015), four multi-variate options and two American options. Chapter Four provides a summary of the results and conclusions.

Chapter 2

Overview of Theory

In this chapter, the mathematics leading to functional quantization will be presented. This includes the formation of an optimal vector quantizer, the extension of the vector quantizer to an infinite dimensional setting and the use of these quantizers to optimally stratify a Hilbert space. The idea of functional quantization is then extended to processes in multiple dimensions. Finally, the use of stratification in the Longstaff-Schwartz method for American option pricing is investigated.

2.1 Optimal vector quantization

Consider the probability space $(\Omega, \mathcal{A}, \mathbb{P})$ and an L^2–space E. A vector quantizer of this space is defined as follows:

Definition 2.1. (Quantizer) A quantizer $\Gamma = \{\gamma_1, \ldots, \gamma_N\}$ is a set which takes on finitely many values in $\mathbb{R}^d$. Quantization is then the process of finding the Γ which best approximates the distribution.

Consider Γ, a quantizer of X that takes on finitely many (N) values in E. The resulting discretization error is given by the L^p-norm of $X - \Gamma$. Naturally one wishes to minimize this error, giving the minimization problem

$$\min\{||X - \Gamma||_p, \Gamma : \Omega \to E \text{ measurable, } \operatorname{card}(\Gamma(\Omega)) \leq N\}, \tag{2.1}$$

where $\operatorname{card}(\Gamma(\Omega))$ denotes the cardinality of $\Gamma(\Omega)$. In order to create a more tractable expression, we first define the Voronoi partitions.

Definition 2.2. (Voronoi partition) Let $C = \{C_1, \ldots, C_N\}$ be a borel partition of E and $\Gamma = \{\gamma_1, \ldots, \gamma_N\}$, the associated quantizer. C_i is the Voronoi partition associated with γ_i if $\forall i \in \{1, 2, \ldots, N\}, \quad C_i = \{\xi \in E, |\xi - \gamma_i| = \min_{j \in [1,N]} |\xi - \gamma_j|\}$, which is to say that C_i is the set of all points whose nearest point in Γ is γ_i.

The minimization problem in (2.1) can then be reduced to

$$\min\{||X - \text{Proj}_\Gamma(X)||_p,\ \Gamma \subset E,\ \text{card}(\Gamma) \leq N\}, \tag{2.2}$$

where $\text{Proj}_\Gamma(X) = \sum_{i=1}^{N} \gamma_i \mathbb{1}_{C_i}(X)$ is the nearest neighbour projection of X onto Γ. Luschgy and Pagès (2002) provide a proof of the existence of this minimum.

2.1.1 Practical considerations

Here we investigate the practical implications of optimal quantization for a finitely dimensioned space. In the following, the L^p-distortion function for a quantizer of level N is defined as

$$\mathcal{D}_{p,N} = ||X - \text{Proj}_\Gamma(X)||_p. \tag{2.3}$$

The majority of literature focuses on three methods for minimizing (2.2): Competitive Learning Vector Quantization (CLVQ), Lloyd's method and the Newton Raphson method (Gray, 1984; Pagès *et al.*, 2003). These methods are explained below in increasing order of complexity.

Lloyd's method

Lloyd's method is a fixed-point algorithm which iteratively updates according to

$$\hat{\gamma}_i(k+1) = \frac{\int_{C_i(k)} x d\mathbb{P}(x)}{\int_{C_i(k)} d\mathbb{P}(x)},$$

where $\gamma_i(n)$ indicates the value for γ_i after the n^{th} iteration of Lloyd's method. In a probabilistic setting, one can use the randomized Lloyd's method, where the integrals above are evaluated using Monte Carlo integration:

$$\hat{\gamma}_i(k+1) = \frac{\sum_{m=1}^{M} X_m \mathbb{1}_{\{X_m \in C_i(k)\}}}{\sum_{m=1}^{M} \mathbb{1}_{\{X_m \in C_i(k)\}}}.$$

Pagès *et al.* (2003) note several disadvantages of Lloyd's method, namely: for dimensions greater than 1 it may not converge, it becomes numerically intractable (since it requires evaluating the d-dimensional integral $\int_{C_i} \dots d\mathbb{P}(x)$), and it is prone to getting "stuck" in local minima. Gray *et al.* (1980) note that, for a differentiable and continuous $\mathcal{D}_{p,N}$, one can escape local minima by ensuring that there is a zero probability of hitting the boundaries of a decision region — i.e., for any $X \in \mathbb{R}^d$, $\mathbb{P}\left(||X - \text{Proj}_{\gamma_i}(X)|| = ||X - \text{Proj}_{\gamma_j}(X)||\right) = 0$ for every $i \neq j$.

CLVQ

The CLVQ method is an iterative stochastic gradient descent method based on the integral representation of the gradient of the distortion, $\nabla \mathcal{D}_{2,N}$ (Keller *et al.*, 2006). The CLVQ algorithm, as described in Pagès (2014), is given by

$$\hat{\Gamma}_i(k+1) = \hat{\Gamma}_i(k) - \rho_{k+1} \nabla \mathcal{D}_{2,N}(\hat{\Gamma}_i(k)), \quad \hat{\Gamma}_i(0) \in (\text{Hull}(\text{supp}(\mathbb{P})))^N, \tag{2.4}$$

where Hull(supp($\mathbb{P}$)) is the closed convex hull support of the distribution of $\mathbb{P}$ and ρ_{k+1} is a stepping parameter. Pagès *et al.* (2003) note that while under certain assumptions the stochastic gradient descent method converges almost surely, these assumptions are not suitable for $\mathcal{D}_{2,N}$. Despite this, practical implementation has demonstrated satisfactory results. Unfortunately, for $d \geq 2$, solving $\nabla \mathcal{D}_{2,N}(\hat{\gamma}_i(k))$ becomes computationally expensive, prompting the use of the stochastic equivalent of (2.4),

$$\hat{\gamma}_i(k+1) = \hat{\gamma}_i(k) - \rho_{k+1} \left(\hat{\gamma}_i(k) - X_{k+1} \right) 1_{\{X_{k+1} \in C_i(x)\}} \Big)_{1 \leq i \leq N}. \tag{2.5}$$

Pagés and Printems (2000) suggest the use of the stepping parameter

$$\rho_{k+1} = \frac{4\rho_0 N^{\frac{1}{d}}}{4N^{\frac{1}{d}} + \rho_0 \pi^2 N^{\frac{-2}{d}} k},$$

where ρ_0 is the initial step size and d is the dimension of the quantizer.

Newton-Raphson

Note that when the Hessian of a distribution is known, one can set the stepping parameter to $\rho_{k+1} = (\nabla^2 \mathcal{D}_{2,N})^{-1}$ in (2.4), to obtain the Newton-Rhapson procedure:

$$\hat{\gamma}_i(k+1) = \hat{\gamma}_i(k) - (\nabla^2 \mathcal{D}_{2,N})^{-1} \nabla \mathcal{D}_{2,N}(\hat{\gamma}_i(k)), \tag{2.6}$$

where $\nabla \mathcal{D}_{2,N}(\hat{\gamma}(k))$ and $\nabla^2 \mathcal{D}_{2,N}(\hat{\gamma}_k)$ can be solved explicitly for one dimension. McWalter *et al.* (2016) propose the following formulation. Suppose f_X and F_X are the PDF and CDF of X, respectively. Define r^{i+} and r^{i-} by

$$r^{i+} = \frac{\gamma_i + \gamma_{i+1}}{2} \quad \text{and} \quad r^{i-} = \frac{\gamma_{i-1} + \gamma_i}{2},$$

where $r^{1-} = -\infty$ and $r^{N+} = \infty$. The elements of the vector $\nabla D_{2,N}(\hat{\gamma}_k)$ are then given by

$$\frac{\partial \mathcal{D}_{2,N}(\Gamma)}{\partial \gamma_i} = 2\gamma_i \left(F_X(r^{i+}) - F_X(r^{i-}) \right) - 2 \left(M_X^1(r^{i+}) - M_X^1(r^{i-}) \right),$$

where

$$M_X^p(x) = \mathbb{E}\left[X^p \mathbb{1}_{\{X<x\}}\right] \tag{2.7}$$

is the p-th lower partial expectation of X. Similarly, the diagonal elements of $\nabla^2 \mathcal{D}_{2,N}(\hat{\gamma}_k)$ are given by

$$\frac{\partial \mathcal{D}_{2,N}(\Gamma)}{\partial(\gamma_i)^2} = 2\left(F_X(r^{i+}) - F_X(r^{i-})\right) + \frac{1}{2}\left(f_X(r^{i+})(\gamma_i - \gamma_{i+1}) + f_X(r^{i-})(\gamma_{i-1} - \gamma_i)\right),$$

and the super- and sub-diagonal elements as

$$\frac{\partial \mathcal{D}_{2,N}(\Gamma)}{\partial \gamma_i \partial \gamma_{i+1}} = \frac{1}{2} f_X(r^{i+})(\gamma_i - \gamma_{i+1})$$

and

$$\frac{\partial D_{2,N}(\Gamma)}{\partial \gamma_i \partial \gamma_{i-1}} = \frac{1}{2} f_X(r^{i-})(\gamma_{i-1} - \gamma_i),$$

respectively.

Stability and splitting methods

Pagés and Printems (2000) mention several causes of instability when generating a guassian quantizer using CLVQ. The first is that large norms may have dramatic effects on the CLVQ step. To mitigate this, one carries out the CLVQ procedure using a spherically truncated normal distribution (with 99% of the mass) before completing the minimization with Lloyd's method. Secondly, for large N, the stability of the quantizer solution may suffer, prompting the use of 'splitting methods'. Splitting methods are heuristic procedures which attempt to stabilize the quantizer generation by sequentially adding points to the quantizer. Pagés and Printems (2000) suggest the use of the following procedure.

Algorithm 2.3. (Splitting method)

1. Initialize a vector of size n, $\hat{\Gamma}^n \sim \mathcal{N}(0; I_d)$.
2. Pass $\hat{\Gamma}^n$ into the CLVQ procedure to obtain Γ^n.
3. Add the origin to Γ^n to obtain $\hat{\Gamma}^{n+1}$.
4. Pass $\hat{\Gamma}^{n+1}$ into the CLVQ procedure to obtain Γ^{n+1}.
5. Repeat (3) and (4) until the desired cardinality is achieved.
6. Pass the resulting quantizer into Lloyd's method.

Alternatively, one can use the Linde-Buzo-Gray (LBG) algorithm with Lloyd's method (Linde *et al.*, 1980):

Algorithm 2.4. (LBG with randomized Lloyd's method)

1. Initialize a vector of size 1, $\Gamma^1 = 0$.
2. Set $\hat{\Gamma}_{n+1} = (1+\epsilon)\Gamma_n \cup (1-\epsilon)\Gamma_n$.
3. Pass $\hat{\Gamma}_{n+1}$ into Lloyd's method to obtain Γ_{n+1}.
4. Repeat (2) and (3) until the desired cardinality is achieved.

Clearly this algorithm only works for an N that is a power of 2. For other N's, one can simply split the points but only if the number of points would not increase beyond the desired size. If splitting would increase the size of the quantizer beyond the desired size, one simply adds a single point to the origin instead of splitting Γ. Both methods have been shown to converge to a local minimum (Gray *et al.*, 1980; Pagès *et al.*, 2003).

2.2 Extension to infinite dimensions

The notion of quantization is extended to an infinite dimensional setting. In order to achieve optimal functional quantization, it is necessary to find a tractable objective function for minimization. With this in mind, we state several definitions and propositions, closely following the approach and method presented by Corlay and Pagès (2015).

Assume now that E is a separable Hilbert space $(H, \langle ., . \rangle_H)$. Note that in the quadratic case, an L^2-optimal quantizer the Borel partition associated with Γ, $\mathcal{C} = \{C_1, \ldots, C_N\}$, is stationary — which is to say $\mathbb{E}[X|X \in C_i] = \gamma_i$. The stationarity of $\mathcal{C}$ in the quadratic case is proved in Graf *et al.* (2007). Consequently, if Γ is an L^2-optimal quantizer and $\mathcal{C} = \{C_1, \ldots, C_N\}$ are the associated Voronoi partitions, one has $\forall \gamma_i \in \Gamma$, that $\gamma_i = \mathbb{E}[X|X \in C_i]$. In order to create a tractable minimization, one must first reduce the dimension of the minimization of $\mathcal{D}_{2,N}$ to some finite number. Proposition 2.5 below provides an upper bound for the $\mathcal{D}_{2,N}$ in terms of a finite dimensional subspace.

Proposition 2.5. *For any finite-dimensional subspace $V \subset H$, denote the orthogonal projection onto V by Π_V. The quantization error ($\mathcal{D}_{2,N}(X)$) at level N is then bounded above by*

$$\mathcal{D}^2_{2,N}(X) \leq \mathbb{E}\left[|X - \Pi_V(X)|^2\right] + \mathcal{D}^2_{2,N}(\Pi_V(X)). \tag{2.8}$$

Additionally, it is clear that the quadratic quantization error consists of projection error and quantization error of the projected random variate. Refer to Luschgy and Pagès (2002) for a proof.

Define the quantization dimension by

$$d_N(X) := \min\{\dim \operatorname{span}(\Gamma)\},$$

where $\dim \operatorname{span}(\Gamma)$ denotes the dimension of the span of Γ, an L^2–optimal quantizer with cardinality $\leq N$. It follows from Proposition 2.5 that the problem of optimal quantization corresponds to the minimization of

$$\mathcal{D}^2_{2,N}(X) = \min\left\{\mathbb{E}\left[||X - \Pi_V(X)||^2\right] + \mathcal{D}^2_{2,N}(\Pi_V(X)), \begin{array}{l} V \subset H \text{ s.t. } \dim \\ V \geq d_N(X) \end{array}\right\}. \tag{2.9}$$

This is to say for any H one can find a finite-dimensional subspace V, such that the quantization error is minimized, provided that the dimension of V is not less than the dimension of the span of Γ. One now has a finite dimensional minimization problem. With this minimization problem in mind, we define the covariance operator.

Definition 2.6. (Covariance operator of a random variable) Consider a centred H-valued L^2 random variable X. Its covariance operator $C_X : H \to H$ is given by $C_X y = \mathbb{E}[\langle y, X\rangle X]$.

Luschgy and Pagès (2002) state that one can then obtain the subspace V in (2.9) from the eigenvectors (corresponding to the largest eigenvalues) of C_X. In order to determine these eigenvectors and eigenvalues, we define the Karhunen-Loéve expansion.

Definition 2.7. (Karhunen-Loéve (KL) expansion for a Gaussian random process) Let $(e_n(t))_{n\geq 1}$ be a set of eigenfunctions of C_X (where $C_X : L([0,T], dt) \to L^2([0,T], dt)$) that form an orthonormal basis of L^2. One may then expand the paths of $(X_t)_{t\geq 0}$ on this basis, i.e.

$$X(\omega, t) = \sum_{n\geq 1} \xi_n(\omega) e_n(t) \qquad \mathbb{P}(d\omega) - a.s.\,,$$

where $(\xi_n)_{n\geq 1} \sim \mathcal{N}(0, \lambda_n^X)$ is a sequence of Gaussian random variables and λ_n is a decreasing sequence of eigenvalues.

Theorem 2.8, proved in Luschgy and Pagès (2002), now allows one to reduce the infinite dimensional quantization problem to a finite one.

Theorem 2.8. *Let Γ be an optimal quantizer of X, $V = span(\Gamma)$ and $m = \dim(V)$. Then $C_X(V) = V$ and $\mathbb{E}\left[|X - \Pi_V(X)|^2\right] = \sum_{j\geq m+1} \lambda_j^X$, where λ_i^X are the non-zero eigenvalues of C_X (in descending order of magnitude). We then have*

$$\sum_{j\geq m+1} \lambda_j^X = \inf\left\{\mathbb{E}\left[|X - \Pi_V(X)|^2\right], V \subset H \text{ linear subspace, } \dim(V) = m\right\}.$$

The minimal quadratic distortion $D_{2,N}(X)$ *is then*

$$\mathcal{D}_{2,N}(X) = \sum_{j \geq m+1} \lambda_j^X + \mathcal{D}_{2,N}\left(\bigotimes_{j=1}^{m} \mathcal{N}(0, \lambda_j^X)\right) \quad \text{for } m \geq d_N(X),$$

where $\bigotimes_{j=1}^{m} \mathcal{N}(0, \lambda_j^X)$ *is a quantizer of the* m*-dimensional Gaussian normal* $\mathcal{N}(0, \delta_{ij}\lambda_j^X)$*,* δ_{ij} *is the Kronecker delta and* λ_j^X *is determined by the KL expansion of the covariance operator* C_X*.*

Theorem 2.8 shows that, because $\sum_{j \geq m+1} \lambda_j^X$ is only dependent on the quantization dimension, the problem of optimal quantization of a Gaussian process can be reduced to the finite dimensional quantization of $\bigotimes_{j=1}^{m} \mathcal{N}(0, \lambda_j^X)$ (when the KL expansion is known).

Assume now that X is a bi-measurable Gaussian process defined on $(\Omega, \mathcal{A}, \mathbb{P})$. Corlay and Pagès (2015) presents the following closed form KL basis and eigenvalues:

1. Brownian motion:

$$e_n^W(t) := \sqrt{\frac{2}{T}} \sin\left(\pi(n - 0.5)\frac{t}{T}\right), \quad \lambda_n^W := \left(\frac{T}{\pi(n - 0.5)}\right)^2, \quad n \geq 1.$$

2. Brownian bridge:

$$e_n^B(t) := \sqrt{\frac{2}{T}} \sin\left(\pi n \frac{t}{T}\right), \quad \lambda_n^B := \left(\frac{T}{\pi n}\right)^2, \quad n \geq 1.$$

The natural form of an optimal functional quantizer path is then

$$X_i = \sum_{1 \leq n \leq d_N} \bigotimes_{j=1}^{d_N} \mathcal{N}(0, \lambda_j^X) e_n^X, \tag{2.10}$$

where d_N is defined as the truncation dimension. The truncation dimension is selected for a particular N so as to minimize the distortion. Luschgy and Pagés (2012) have shown this to be $d_N = \{\lfloor \log(N) \rfloor, \lceil \log(N) \rceil\}$. However, for the purposes of this dissertation, the critical dimension for a given N will be taken to be the optimal d_N as determined by Corlay *et al.* (2005).

Product quantization is another formulation of optimal functional quantization that attempts to reduce the complexity of the multi-dimensional minimization problem required in determining $\bigotimes_{j=1}^{d_N} \mathcal{N}(0, \lambda_j^X)$. A product quantizer approximates $\bigotimes_{j=1}^{d_N} \mathcal{N}(0, \lambda_j^X)$ as the Cartesian product of d_N 1-dimensional vector quantizers, resulting of a quantizer of the form:

$$X_i = \sum_{1 \leq n \leq d_N} \sqrt{\lambda_n^X} \xi_n^N e_n^X, \tag{2.11}$$

where ξ_n^N is the Cartesian product of one dimensional vector quantizers, denoted by $\bigotimes_{i=1}^n \Gamma_{N_i}$. In order to determine ξ_n^N, one must first find a product decomposition of N such that $N_1 \times \ldots \times N_n \leq N$. Clearly there are many possible decompositions for a given N. Corlay and Pagès (2015) suggest the use of a blind optimization procedure, where the distortion arising from the quantizer formed from each of these decompositions is tested. Once again, for the purposes of this dissertation, the optimal decompositions will be taken from Corlay *et al.* (2005). Another advantage of product quantization is that, due to the explicit nature of the Newton-Raphson method, one can easily calculate the probability associated with each path. The probabilities are given by $P_i = \sum_{k=1}^{d_N} \left(\Pi_{j=1}^n F_{ij}^{\otimes} \right)$, where $F^{\otimes}$ is the $(N \times d_N)$-dimensional matrix formed from the Cartesian product $\bigotimes_{j=1}^{d_N} \left(F(r_j^{\cdot +}) - F(r_j^{\cdot -}) \right)$ and $r_j^{\cdot +}$, $r_j^{\cdot -}$ are vectors containing all the r^{i+} and r^{i-} for the j^{th} quantizer.

2.3 Quantization-based variance reduction

2.4 Stratified sampling - Bayesian simulation

In this section we describe the use of the stratified sampling of Brownian motion as a method of variance reduction in Monte Carlo price estimation. In order to implement a stratified sampling technique, one is required to simulate the conditional distribution $\mathcal{L}(X|X \in A_i)$, where A_i is the slab associated with a particular Brownian path, X_i. The slab A_i is equivalent to C_i (the Voronoi region for the product quantizer), this nomenclature is used to distinguish between the Voronoi regions of the product quantizer and the Voronoi regions of the quantizers used to generate it. The simulation of a conditional path takes advantage of the orthonormal nature of the KL decomposition,

$$\begin{pmatrix} X_{t_1} \\ \vdots \\ X_{t_n} \end{pmatrix} = \underbrace{\sum_{k=1}^{d_N} \sqrt{\lambda_k^X}\xi_k \begin{pmatrix} e_k^X(t_1) \\ \vdots \\ e_k^X(t_n) \end{pmatrix}}_{\mathrm{Z}} + \underbrace{\sum_{k \geq d_N+1}^{\infty} \sqrt{\lambda_k^X}\xi_k \begin{pmatrix} e_k^X(t_1) \\ \vdots \\ e_k^X(t_n) \end{pmatrix}}_{\mathrm{Y}}.$$

Recalling that a functional quantizer consists of the first d_N terms of the KL expansion, it is natural that in order to generate conditional paths one must calculate the Y in the above expansion. While it is difficult to explicitly calculate Y, X is simulatable and Z can be determined through a least squares regression. If we define

$$\xi = \begin{pmatrix} \int_0^T X_s e_1(s)^X ds \\ \vdots \\ \int_0^T X_s e_{d_N}(s)^X ds \end{pmatrix}$$

then $Z = \mathbb{E}[X_s|\xi]$ is the expectation of X_t, given its first d_N KL coordinates. This yields $X = \mathbb{E}[X|\xi] + Y$, where $Z \sim \mathcal{N}(0, \text{cov}(X - \mathbb{E}[X|\xi]))$. The covariance is then

$$\begin{aligned}\text{cov}(\xi - \mathbb{E}[\xi|X]) &= \text{cov}(\xi) + \text{cov}(\mathbb{E}[\xi|X]) - 2\text{cov}(\xi, \mathbb{E}[\xi|X]) \\ &= \text{cov}(\xi) - \alpha\text{cov}(X)\alpha^t,\end{aligned}$$

where $\text{cov}(\xi) = \delta_{ij}\lambda^W$ (δ_{ij} being the Kronecker delta), $\text{cov}(X)_{ij} = \min(t_i, t_j)$. The term α is the linear part of the regression of Y on X. In the general case this can be calculated through least squares regression, and in the Brownian case Corlay and Pagès (2015) provide an analytic solution

$$\alpha = \begin{cases} \alpha_{ij} = \lambda_i^X \frac{2e_i^X(t_j) - e_i^X(t_{j-1}) - e_i^X(t_{j+1})}{\delta t} & j \notin \{0, n\} \\ \alpha_{i0} = \lambda_i^X \left(\left(e_i^X\right)'(t_0) - \frac{e_i^X(t_1) - e_i^X(t_{j+1})}{\delta t} \right) & j = 0 \\ \alpha_{in} = \lambda_i^X \left(\frac{e_i^X(t_n) - e_i^X(t_{n-1})}{\delta t} - \left(e_i^X\right)'(t_n) \right) & j = n \end{cases} . \tag{2.12}$$

The algorithm for the efficient simulation $(X|X \in A_i)$, which denotes some X simulated such that it lies in the slab A_i, is then:

Algorithm 2.9. (Simulation of $(X|X \in A_i)$)

1. Calculate $R_{k \le d_N, 0 \le j \le n} = \sqrt{\lambda_k} e_j^X$.
2. Simulate a Brownian path X.
3. Simulate $\xi_k \sim \mathcal{N}\left((RR^t)^{-1}RV, \text{cov}(\xi) - \alpha\text{cov}(V)\alpha^t\right)$.
4. Determine $Y = V - (\xi_k)^t R$.
5. Simulate $(\xi_k|X \in A_i) = F^{-1}\left((F(b_i) - F(a_i))U + F(a_i)\right)$, $U \sim \mathcal{U}[0,1]$, where a_i, b_i are the rows of the Cartesian products $a_{1 \le i \le N_n, 1 \le j \le d_N} = \bigotimes_{j=1}^{d_N} r_{N_j}^{i+}$ and $b_{1 \le i \le N_n, 1 \le j \le d_N} = \bigotimes_{j=1}^{d_N} r_{N_j}^{i-}$. This is simply simulating ξ_k from the hyper-rectangle of the i^{th} stratum.
6. Calculate $(Z|X \in A_i) = (\xi_k|X \in A_i)R$.
7. Calculate $(X|X \in A_i) = (Z|X \in A_i) + Y$.

It is important to note that the simulation of $(\xi_k|X \in A_i)$, as in step 5, is possible due to the rectangular nature of the quantizer grids generated by product quantization. If one were to use an optimal quantizer, it would be necessary to generate normal random variates in a d_N-dimensional polygon. Additionally, the relatively expensive computation $(RR^t)^{-1}R$ in step 3 need not be recalculated for each V. Algorithm 2.9 now allows for the rapid simulation of Brownian paths conditional on their association with A_i.

The price of an option with some payoff H can then be calculated as

$$\begin{aligned} H &= B(t_0,T)\mathbb{E}\left[H((X_t)_{t\in[t_0,T]})\right] \\ &= B(t_0,T)\sum_{k=1}^{N}\mathbb{E}\left[H((X_t)_{t\in[t_0,T]});A_k\right] \\ &= B(t_0,T)\sum_{k=1}^{N}\mathbb{P}\left((X_t)_{t\in[t_0,T]}\in A_k\right)\mathbb{E}\left[H((X_t)_{t\in[t_0,T]})|A_k\right] \\ &= B(t_0,T)\sum_{k=1}^{N}P_k\mathbb{E}\left[H((X_t)_{t\in[t_0,T]})|A_k\right], \end{aligned}$$

where $B(t_0,T)$ denotes the discount factor from t_0 to T. The variance associated with the price is given by

$$\mathrm{Var}[H]=\sum_{k=1}^{N}\frac{P_k^2}{q_k}\mathrm{Var}\left[H((X_t)_{t\in[t_0,T]})|A_k\right],$$

where q_i is the weight of each stratum and P_k is the probability associated with the stratum. If one considers the case of the natural allocation, $q_i = P_i$, then

$$\begin{aligned} \mathbb{E}[Y^2] &= \sum_{i=1}^{K}P_i\mathbb{E}[Y^2|X-\in A_k] \\ &= \sum_{i=1}^{K}P_i\left(\sigma_i^2+\mu_i^2\right) \\ &= \sum_{i=1}^{K}P_i\left(\sigma_i^2+\mu_i^2\right). \end{aligned}$$

so

$$\begin{aligned} \mathrm{Var}[Y] &= \mathbb{E}[Y^2]-\mu^2 && (2.13)\\ &= \sum_{i=1}^{K}P_i\sigma_i^2+\sum_{i=1}^{K}P_i\mu_i^2-\left(\sum_{i=1}^{K}P_i\mu_i\right)^2 && (2.14)\\ &\geq \sum_{i=1}^{K}P_i\sigma_i^2. && (2.15) \end{aligned}$$

This means means that, in the case of natural allocation, one can only decrease the variance. Additionally, this makes no assumptions regarding the form of the quantizers. The choice of weights can be further refined through the minimization

problem $\min_{q_i} \mathrm{Var}[H]$, subject the $(q_i)_{1\leq i\leq N}$ being a probability vector. The solution of this is given by (Glasserman, 2003) as

$$q_i = \frac{P_i\sigma_i}{\sum_{k=1}^{N} P_k\sigma_k},$$

where σ_i is the standard deviation of the price in each stratum. The volatilities in each stratum are not known *a priori*, so one must first estimate them with a pilot run.

2.5 Functional stratification of higher dimensions

Similar to the manner in which d_N one-dimensional vector quantizers were used to construct an approximate d_N-dimensional quantizer, one can extend the notion of functional quantizers to a higher dimension. Consider a Brownian diffusion in $\mathbb{R}^n$. The distribution at any time $t_{0\leq i\leq m}$ can be approximated by the Cartesian product of the functional quantizer paths at that same time point. This method assumes that the principal components of the n-dimensional diffusion can be approximated by linear combinations of the principal components of the one-dimensional diffusion. While this method is sub-optimal in terms of both the extension of a quantizer grid to a functional quantizer in $\mathbb{R}^n$ and the d_N dimensional grid itself, it does maintain some useful properties. The probability associated with each stratum can simply be calculated as the product along the columns of the Cartesian product of the vector of probabilities associated with the n functional quantizers:

$$P_i^n = \prod_{k=1}^{n} \left(\bigotimes_{l=1}^{n} P_l \right)_{1\leq i\leq \prod_{m=1}^{n} \mathrm{card}(P_m), 1\leq k\leq n}.$$

Each dimension of the conditional paths simulated in Algorithm 2.9 can be simulated separately before being concatenated to form a path in $\mathbb{R}^n$. This allows the unmodified use of Algorithm 2.9. The implementation of this method is then relatively straight forward, and the variance of the discounted payoff will at least match that of the standard Monte Carlo estimate. Note that while in the one-dimensional case a pilot run with 30 samples per strata was sufficient, in a multidimensional case we require 30^n samples per strata. Naturally, to achieve any benefit from optimal sampling, we require that $n_p > 30^n N$, where n_p is the total number of sample paths.

2.6 Least squares Monte Carlo in American pricing

When pricing American options using Monte Carlo methods, it is necessary to price backward in time from a set of payoffs at maturity. Let $h(X_i)$ denote the payoff function for exercise at time t_i and $V_i(x)$ the value of the option at time t_i, given that $S = x$. The t_0 value of the option is then given by $V_0(X_0)$ and can be determined recursively as follows:

$$V_n(x) = h_n(x) \tag{2.16}$$

$$V_{i-1}(x) = \max\left\{h_{i-1}(x), \mathbb{E}\left[B(t_{i-1}, t_i)V_i(X_i)|X_i = x\right]\right\}. \tag{2.17}$$

We now define the continuation value of the option, which is the value of holding the option rather than exercising it, as $C_i(x) = \mathbb{E}\left[V_{i+1}(X_{i+1})|X_i = x\right]$, $C_n = 0$. We then have

$$V_i(x) = \max\left\{h_i(x), C_i(x)\right\}.$$

Longstaff and Schwartz (2001) propose the use of regression in the estimation of continuation values of a set of simulated paths as follows

$$C_i(x) = \sum_{r=1}^{M} \beta_{ir}\phi_r(x), \tag{2.18}$$

where ϕ_r are a set of parametric basis functions. Longstaff and Schwartz (2001) suggests the use of the Laguerre polynomials,

$$\phi_r(x) = \frac{e^x}{r!}\frac{d^r}{dx^r}(x^r e^{-x}).$$

The algorithm for pricing an American option then reads as follows.

Algorithm 2.10. (Longstaff Schwartz least squares)

1. Simulate M independent paths $(X_{ij})_{i\leq M, j\leq n}$.
2. At $t = t_n$ set $V_{nj} = h_m(X_{mj})$ for $j \leq M$.
3. Apply the recursive rule given by (2.17), where the continuation value is given by (2.18).
4. Set $V_0(X_0) = \frac{P_i}{M}\sum_{k=1}^{R} V_0(X_{k0})$.

While this method provides an unbiased estimator for $V_0(X_0)$, it requires that the paths $(X_{ij})_{i\leq M, j\leq n}$ are independent, which is not the case for our stratification. One then requires a weighted least squares minimization scheme. Let $F = (\phi_r(X_{ri}))_{r\leq R, i\leq M}$ and minimize

$$\min_{\beta}\left(\omega \circ (Y - F\beta)\right)'\left(\omega \circ (Y - F\beta)\right),$$

where $\omega_{i\leq M} = \sqrt{\frac{P_i}{M_i}}$ is a vector of weights, which corrects for the disproportionate strata sizes (M_i is the number of paths in the stratum, P_i is the probability associated with the slab) and Y is the vector of prices at the next time step. Denote $F^\circ = \omega \circ (F) = (\omega \circ \phi_r(X_{ri}))_{r\leq R, i\leq M}$ and $Y^\circ = \omega \circ Y$, where $\circ$ denotes the element-wise Hadamard product. One then has

$$\min_\beta (Y^\circ)'Y^\circ - 2\beta'(F^\circ)'Y^\circ + \beta'(F^\circ)'F^\circ\beta.$$

Taking the derivative with respect to β and equating to zero yields

$$-2(F^\circ)'Y^\circ + 2(F^\circ)'F^\circ\beta = 0.$$

The coefficients β can then be determined as

$$\beta = \left((F^\circ)'F^\circ\right)^{-1}(F^\circ)'Y^\circ.$$

Chapter 3

Results and Discussion

3.1 Quantization validation and analysis

3.1.1 Validation

The optimal quantization grids of the multivariate normal distribution, generated using Algorithm 2.4, were validated by comparison against those presented in Corlay *et al.* (2005). In the following, the test cases are those given by Corlay and Pagès (2015) and the 'Achieved' are those calculated in this dissertation. Figure 3.1 below compares a grid generated with Algorithm 2.4 with one presented in literature.

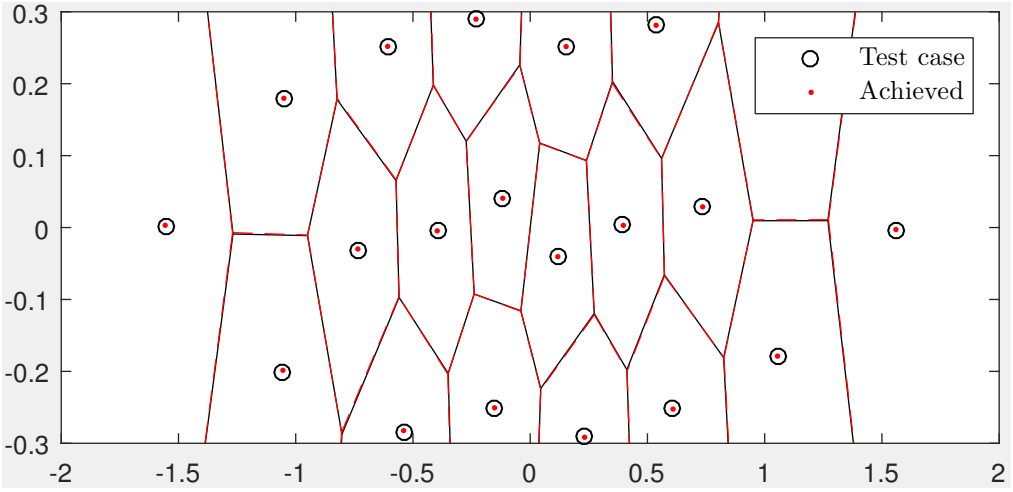

Fig. 3.1: Comparison of test case and achieved quantization grid with $N = 20, d_n = 2$.

A distortion of $\mathcal{D}_{2,20} = 0.0229$ was achieved through Algorithm 2.4, which is slightly higher than the $\mathcal{D}_{2,20} = 0.0226$ achieved in Corlay *et al.* (2005). Despite the small difference in distortion, in the two dimensional case the algorithm converged to the local optimum as determined by Corlay *et al.* (2005). Figure 3.2 below makes the same comparison for the 3-dimensional case.

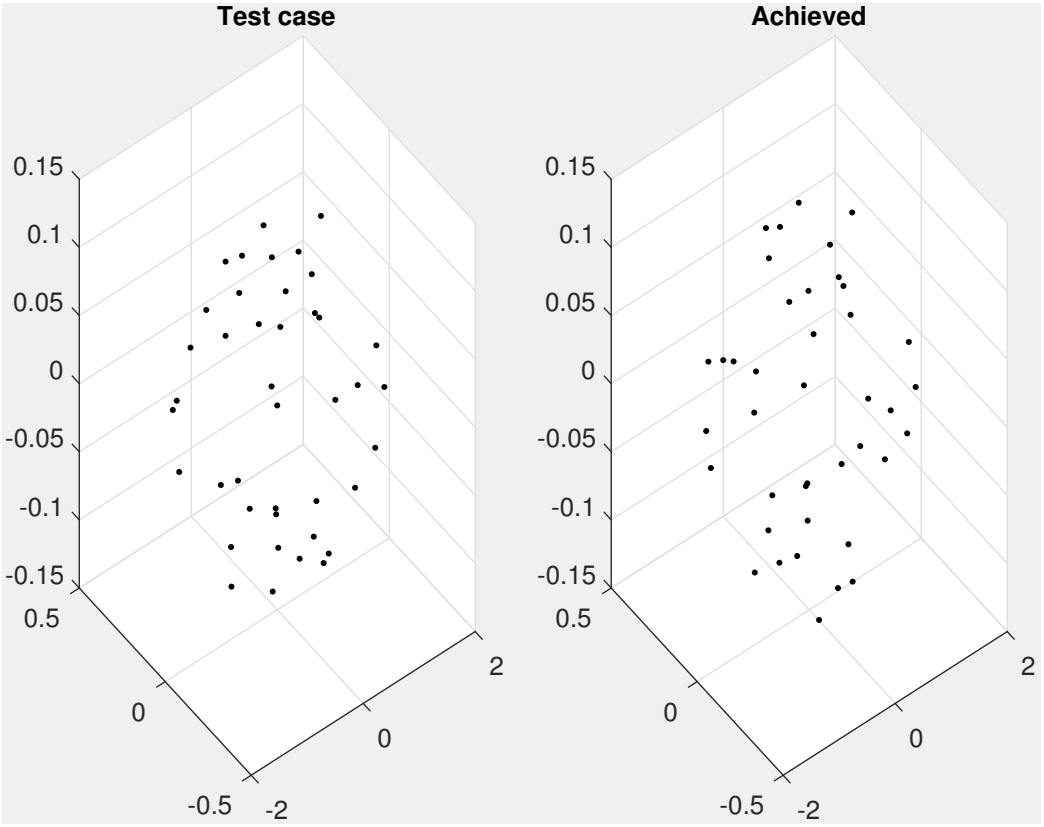

Fig. 3.2: Comparison of test case and achieved quantization grid with $N = 40, d_n = 3$.

Clearly the local optimum given by Corlay *et al.* (2005) was not achieved. Despite aesthetic differences in the grids, the distortions associated with each is relatively similar (0.0254 and 0.0256 for the test case and the achieved, respectively). The difference in appearance could be a result of the random specification of the initial grid in Algorithm 2.3 or the stochastic nature of the CLVQ method. While the CLVQ splitting method guaranties a local minimum, it does not necessarily guaranty the same local minimum on each run. In Algorithm 2.4, however, the only element of randomness is the randomized Lloyd's algorithm, which for a large enough sample size should allow for a consistent quantizer solution. Note that due to the symmetric nature of the multi-normal distribution each dimension of the grid could be reflected about 0 without altering the distortion. Algorithm 2.4 may then generate any of the d_N^2 permutations of an optimal grid. In both test cases it is likely the solution could be refined further to achieve a slightly lower distortion, however this would be at the cost of an increase in computational expense.

We now investigate the functional quantizers generated from the grids above.

Figure 3.3 below compares the functional quantizers generated from the two dimensional grids shown in Figure 3.1. As expected, the functional quantizers are very similar, both with a distortion of 0.0718. In the three dimensional case (shown

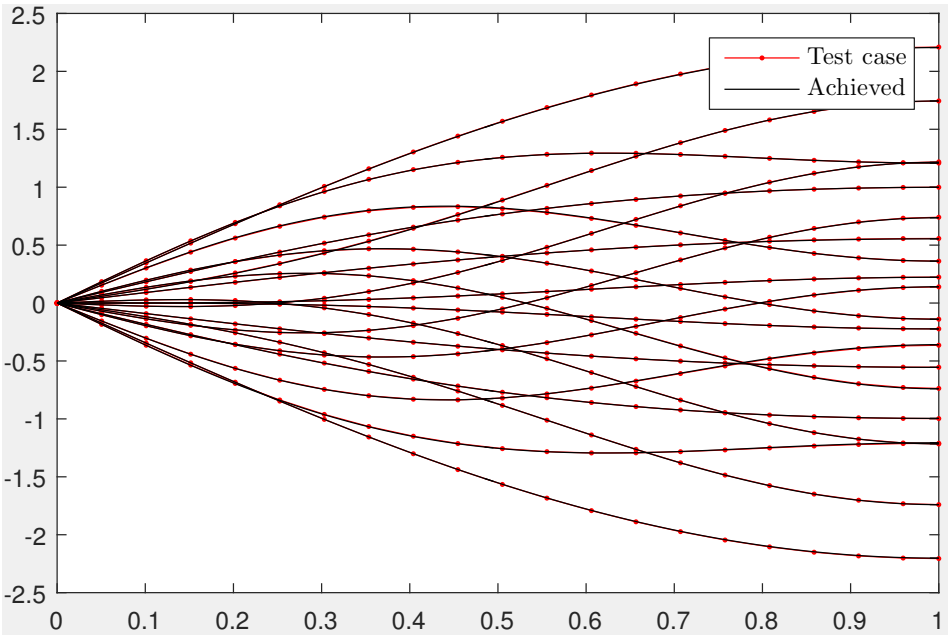

Fig. 3.3: Comparison of test case and achieved functional quantization with $N = 20, d_n = 2$.

in Figure 3.4), the functional quantizer does not exactly match the test case. At the extreme values the two quantizers coincide, however towards the centre there is a significant deviation. Despite this, both quantizers have an equal distortion of 0.0584, indicating that both local minima presented in Figure 3.2 provide an equally valid functional quantizer.

3.1.2 Comparison of product and optimal quantization

Recall that a product quantizer is an approximation of the optimal quantizer generated via the Cartesian product of d_N vector quantizers. Due to the rapid convergence of Algorithm 2.6, a product quantizer can be generated with little computational expense. It would be desirable that the distortion of the product quantizer and the resulting functional quantizer is not significantly higher than the optimal distortion. Figure 3.5 below compares the distortion for the grid and functional cases. Product and optimal quantizer grids and the corresponding functional quantizers were generated for $1 \leq N \leq 128$ and $1 \leq d_N \leq 3$. The distortions of the grid quantizers and functional quantizers were then calculated.

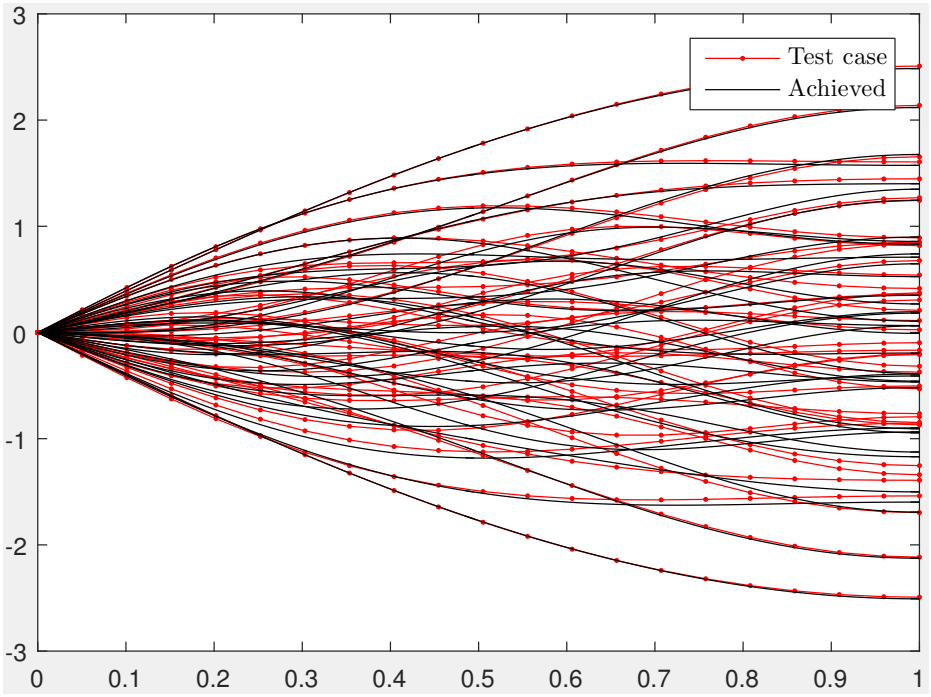

Fig. 3.4: Comparison of test case and achieved functional quantization with $N = 40, d_n = 3$.

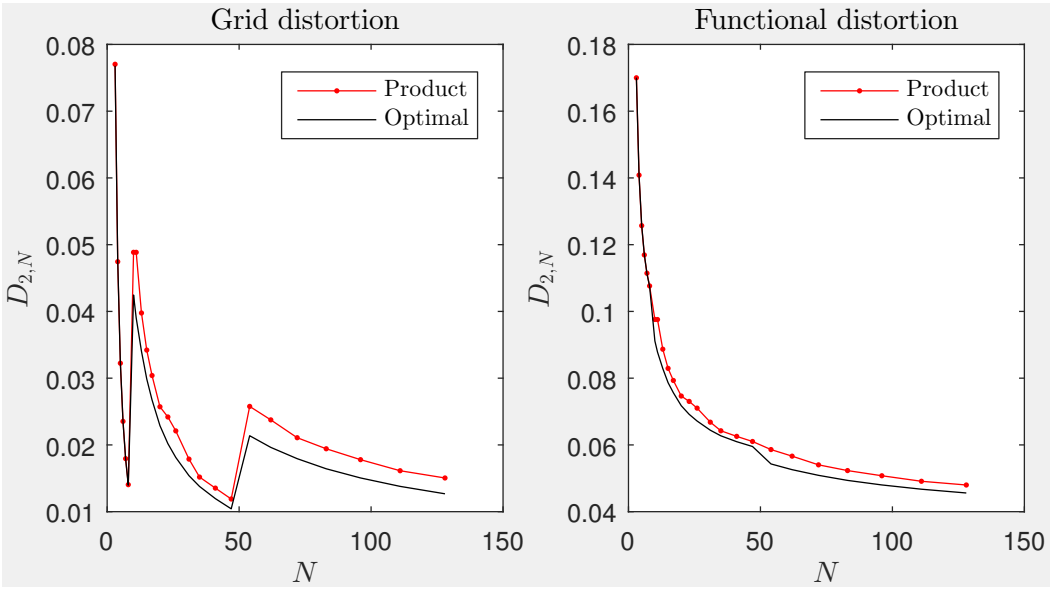

Fig. 3.5: Comparison of the distortion arising from optimal and product quantizers.

In all cases the product grid performed almost as well as the optimal quantizer in terms of grid and functional distortion. The jumps in the grid distortion occur

each time the quantization dimension increases. Note that the product and optimal distortion coincide for $1 \leq N \leq 9$, as in these cases $d_N = 1$ and the product and optimal grid are exactly equal.

It is clear that the product quantizer is capable of achieving the same level of distortion as the optimal quantizer; all that is required is a larger N. Additionally, the time taken to calculate the optimal quantizer is many orders of magnitude more than that required to calculate the product quantizer. The product quantizer then has two major benefits over the optimal quantizer:

1. The explicit calculation of the probabilities associated with each slab.
2. The rectangular Voronoi cells required for Algorithm 2.9.

As such, product quantization was the natural choice for the rest of the dissertation.

3.2 Validation of Bayesian simulation

Once the functional quantizer generation was validated it was necessary to ensure that Algorithm 2.9 was, in fact, simulating Brownian motions conditional on belonging to a particular slab. Figure 3.6 overleaf shows the simulation of 5 random paths conditional on belonging to the slab of the quantizer shown in red. While the simulation appears to have Brownian paths in the correct slab, it was necessary to ensure that the generated paths are in fact Brownian. This validation was done by first simulating a set of paths and Algorithm 2.9 using the natural weights (i.e. $\lfloor N_i = P_i N \rceil$). It is required that the variance and mean are given by $\text{Var}(X_t) = t$ and $\mathbb{E}(X_t) = 0$, respectively. Figure 3.7 overleaf shows that this is the case and that Algorithm 2.9 is working as desired.

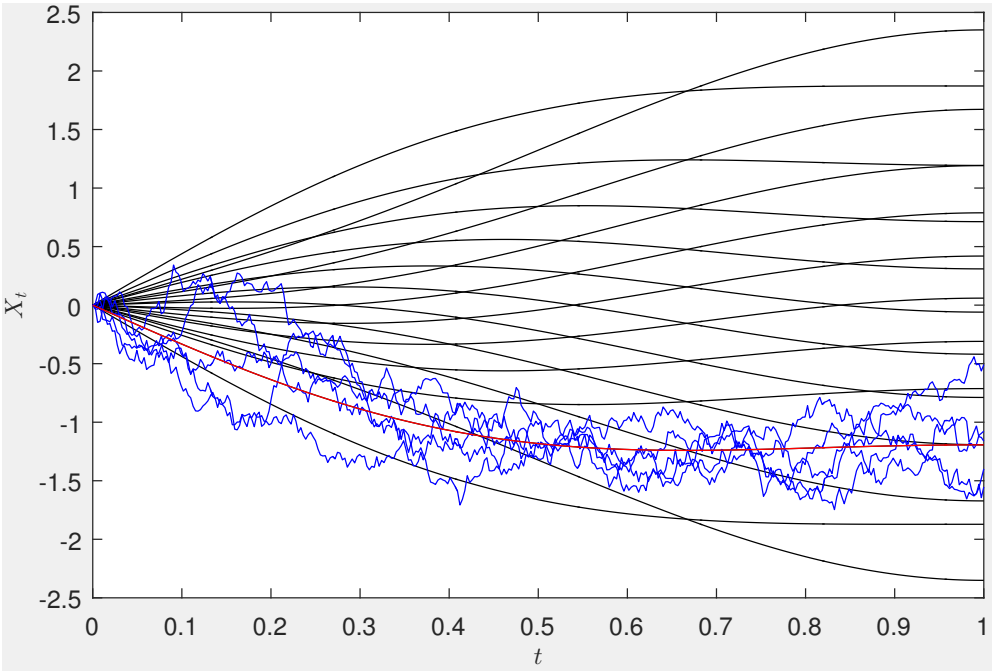

Fig. 3.6: Demonstration of the generation of conditional Brownian paths.

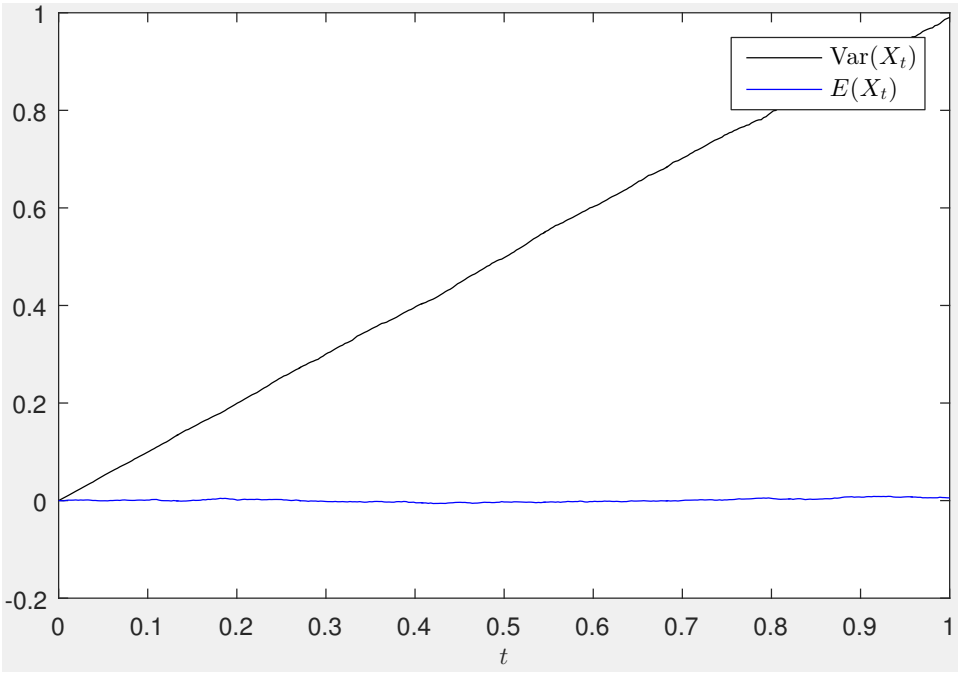

Fig. 3.7: Distributional properties of naturally weighted paths generated using Algorithm 2.9.

3.3 Validation and efficiency of pricing

The pricing algorithm was validated against examples presented in Corlay and Pagès (2015). The convergence of all tests to the literature prices can be found in Section A.1. Higher sample size prices can be seen in Tables 3.1, 3.2 and 3.3. The first two test cases, shown in Tables 3.1 and 3.2, feature up-and-in call options with payoff

$$f\left((S)_{t\leq T}\right) = \max(S_T - K, 0)\mathbb{1}_{\max\left((S)_{t\leq T}\geq H\right)},$$

and underlying dynamics governed by GBM,

$$d\,S(t) = \mu S(t)\,dt + \sigma S(t)\,dW_t,$$

where μ is the drift and σ is the volatility of the stock. The third test case featured an autocall option with the following payoff structure: for some observation times $(T_i)_{1\leq i\leq n}$, if $S_t > K$ at $(T_i)_{i<n}$ the holder receives $P(1+C)$ and the option ends but if the option survives until T_n the holder receives

$$f\left(S_{T_n}\right) = \begin{cases} P(1+C) & S_{T_n} > K, \\ P & H \leq S_{T_n} \leq K, \\ \frac{PS_{T_n}}{K} & S_{T_n} < H. \end{cases}$$

For this option the stock prices were modelled under the CEV dynamics

$$d\,S(t) = \lambda\sqrt{V(t)}S(t)^{\frac{\beta}{2}}\,dW_t,$$

where $V(t) = \sigma^2$ and $0 \leq \beta \leq 2$ is the elasticity parameter. The stock prices are then simulated using the Euler discretization scheme,

$$\ln S(t_i) = \ln S(t_{i-1}) - \frac{1}{2}\lambda\sigma S(t_{i-1})^{\beta-2}(t_i - t_{i-1}) + \sigma S(t_{i-1})^{\frac{\beta}{2}-1}\Delta W_S(t_i)). \quad (3.1)$$

3.3.1 Up-and-in call option

While it is a certainty that the stratification would lead to at most equal variance in the estimation of the price, one must also consider the time taken to achieve this reduction in variance. Figure 3.8 overleaf compares the 99.7% confidence interval associated with a price achieved at some number of paths N_p, as well as the time taken to calculate the price.

While Figure 3.8 provides a clear representation of the efficiency of the methods at varying times, in the interests of compact results a new metric was introduced. Consider the form of the confidence interval, as given by the central limit theorem, $[V_0 - \frac{\alpha\sigma}{\sqrt{N_p}}, V_0 + \frac{\alpha\sigma}{\sqrt{N_p}}]$, where V_0 is the option price. The size of the α-confidence

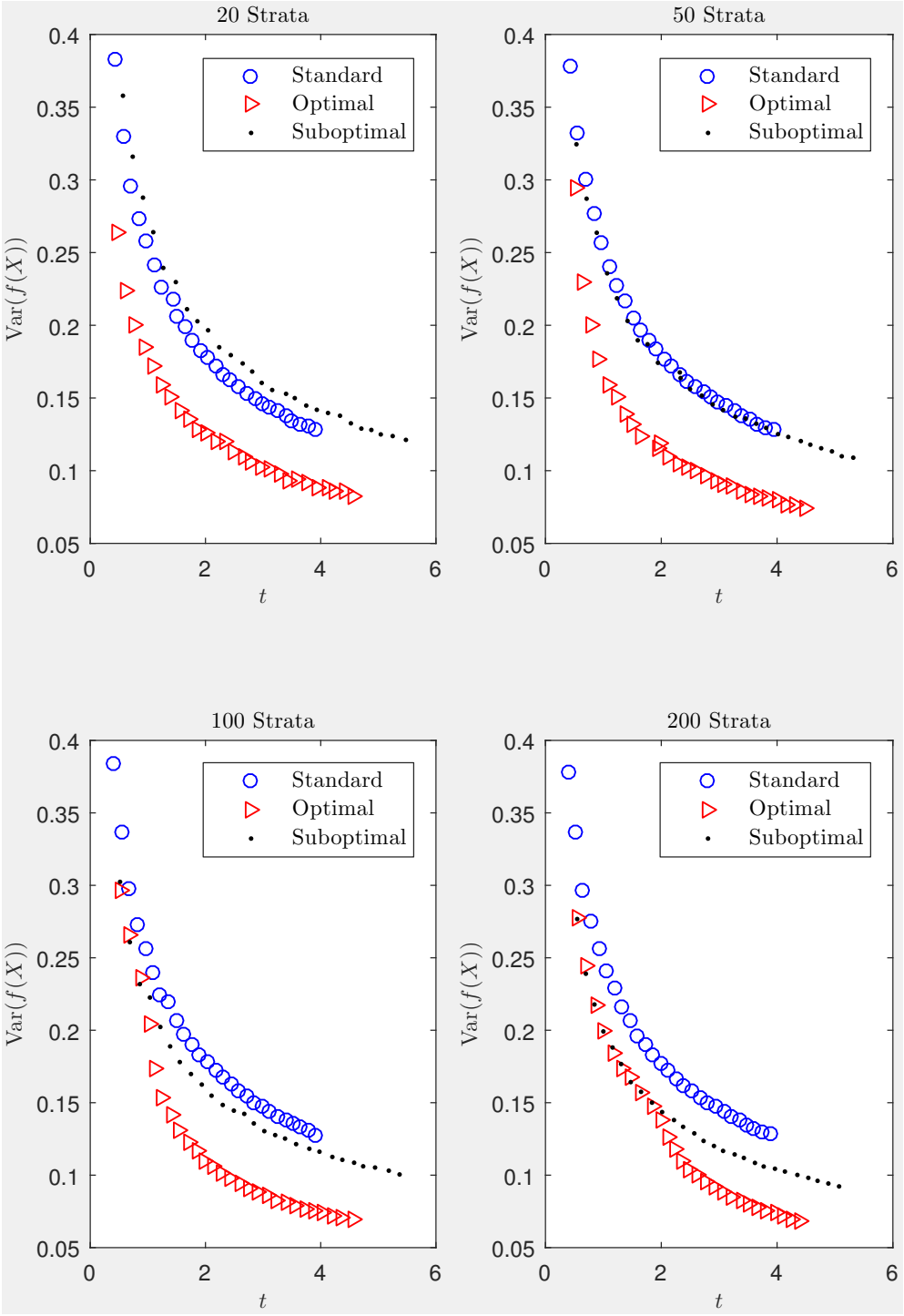

Fig. 3.8: Comparison of variance reduction achieved at a given time for an up-and-in call option: $S_0 = 100$, $K = 100$, $H = 125$, $\sigma = 0.3$, $T = 1.5$, $r = 0$, $M = 365$.

interval is then directly proportional to $\frac{\sigma}{\sqrt{N_p}} := \eta$ and if the time taken to compute a price from N paths is known, one can construct a metric which compares the rate at which confidence is achieved. One has $\frac{\alpha\sigma}{\sqrt{N_p}} = \frac{\alpha\sigma}{\sqrt{t_p N_p . s^{-1}}} = \frac{\alpha\sigma}{\sqrt{t_p}}\sqrt{\frac{s}{N_p}} \propto \sigma\sqrt{\frac{s}{N_p}}$, where s is the time taken to calculate a price with N_p sample paths and t_p is the total run time. This metric makes the assumption that the time taken per path is linearly increasing with N_p. This is not strictly true — while Algorithm 2.9 has computational cost of order N_p, it does not consider the time taken to calculate the strata and matrices required for stratification. However, for a relatively high N_p this time becomes negligible.

Note that the payoff functions and functions that convert from Brownian motions to stock paths are also of order N_p. Then, for a large N_p it is expected that η provides an appropriate measure of efficiency for a given sampling method. For the remainder of this section $N_p = 1\ 000\ 000$ was used.

Table 3.1 below shows the option price, the variance associated with the price, the time taken to calculate the price and the η associated with the price for four strata sizes. Note that the achieved prices match the analytic prices calculated by Corlay and Pagès (2015).

Tab. 3.1: Comparison of sampling methods for an up-and-in call with parameters: $S_0 = 100,\ K = 100,\ H = 125,\ \sigma = 0.3,\ T = 1.5,\ r = 0,\ M = 365$

Strata	Method	$V_0(S_0)$	σ^2	$t_p\ (s)$	$\eta\ (\times 10^3)$
$V_0^{an} = 13.9597$	Unstratified	13.9865	729.0537	11.0773	89.8663
20 Strata	Suboptimal	13.9488	162.3504	51.6159	91.5416
	Optimal	13.9542	81.2641	52.1827	65.1197
50 Strata	Suboptimal	13.9438	132.1495	50.7451	81.8898
	Optimal	13.9432	60.8392	51.5272	55.9900
100 Strata	Suboptimal	13.9520	114.0308	52.1543	77.1180
	Optimal	13.9533	50.1935	52.6177	51.3913
200 Strata	Suboptimal	13.9461	95.8493	51.3555	70.1597
	Optimal	13.9425	43.1031	51.9060	47.3002

If one compares η from Table 3.1 and the trends observed in Figure 3.8, it is apparent that η is a suitable indicator of efficiency. For 20 strata, the Suboptimal results approximately match the variance reduction of the unstratified estimate. As the number of strata increases there is a clear increase in performance of the suboptimal case, outperforming the unstratified Monte Carlo results at 50 strata and beyond. The optimal sampling outperforms the unstratified sampling for all numbers of strata, with a slight decrease in variance with increasing numbers of

strata.

The disturbances were most likely a result of a decrease in computational priority during simulation. Despite this, the trends are still clearly observable. The prices in Table 3.1, however, were calculated with much higher sample sizes; the effects of these small disturbances on η should then be negligible. Note that in Figure 3.8, for the higher strata simulations, the Suboptimal and optimal perform equally well for several of smallest times. This is a result of the minimum allowable pilot run size (which is dependent on strata size). With 200 strata, a minimum pilot run size of 14000 paths was enforced, meaning that for any total sample size less than this equal weightings were used. This is true for all plots shown in Appendix A. For the remainder of the discussion, results will be presented as in Table 3.1. The figure equivalents can be found in Appendix A.

The results for the an out of the money up-and-in call can be seen in Table 3.2 below. Once again, the analytic prices provided by Corlay and Pagès (2015) are matched.

Tab. 3.2: Comparison of sampling methods for an up-and-in call with parameters: $S_0 = 100,\ K = 100,\ H = 200,\ \sigma = 0.3,\ T = 1,\ r = 0,\ M = 365$

Strata	Method	$V_0(S_0)$	σ^2	$t_p\ (s)$	$\eta\ (\times 10^3)$
$V_0^{an} = 1.3665$	Unstratified	1.3607	146.8029	10.5659	39.38
20 Strata	Suboptimal	1.3786	81.2796	62.4109	71.2231
	Optimal	1.3667	5.0040	60.6442	17.4203
50 Strata	Suboptimal	1.3555	62.8028	59.2121	60.9810
	Optimal	1.3507	2.7245	51.4196	11.8360
100 Strata	Suboptimal	1.3655	58.7310	59.7364	59.2315
	Optimal	1.3605	2.5462	50.6465	11.3559
200 Strata	Suboptimal	1.3714	43.3615	59.8809	50.9561
	Optimal	1.3636	1.8876	52.5683	9.9614

Table 3.2 shows similar trends to those seen in 3.1, however in this case the Suboptimal weights were outperformed by the standard Monte Carlo for all strata sizes. The optimal weights outperformed both the unstratified and Suboptimally stratified weights by a large margin. This can be explained by the low moneyness of the up-and-in call, with a knock in of twice the initial stock price. The low moneyness leads to a low price, reducing the term $\sum_{i=1}^{K} P_i\mu_i^2 - \left(\sum_{i=1}^{K} P_i\mu_i\right)^2$ in Equation 2.13 and thus reducing the benefit from Suboptimal stratification. So, in the Suboptimal case there is a decrease in variance when compared to the unstratified Monte Carlo, but the additional computation time outweighs the advantage

obtained from the stratification. While the Suboptimal allocation samples from all strata, the optimal weighting only samples from in the money strata, giving a much better estimate of the expectation in each strata.

3.3.2 Autocall option

The Euler discretization, required for the pricing of the autocall option, added significant computation time to the pricing algorithm. This addition lead to the introduction of another variable, the pricing batch sizes. To explain this, one must consider the implementation of the Euler discretization that takes the Brownian paths, simulated by Algorithm 2.9, to stock prices in the CEV model. The discretization scheme in (3.1) lends itself to a vectorised implementation which, in *MATLAB*, leads to a significantly faster computation. The implementation used for this simulation, however, runs each stratum as a batch, i.e., each set of Brownian paths in a stratum are converted to a stock price in a separate call to the Euler function. This means that as the number of strata is increased, the Euler function is called more times, reducing the degree to which the whole calculation is vectorized and ultimately slowing it down. In an attempt to mitigate the effects of this on the comparison between unstratified and stratified approaches, the unstratified Monte Carlo price was calculated in the same sized batches as the Suboptimal allocation. An implementation without batching is possible, but for larger sample sizes it may lead to memory issues. The results for the autocall option under the CEV model can be seen in Table 3.3 below.

Firstly note that all prices fell within the target range of $[V_0^{tar} = 99.0179, 99.0709]$ given in Corlay and Pagès (2015). In Table 3.3 the effects of the vectorization are apparent, with higher strata runs taking slightly longer (a feature that was not seen in Tables 3.1 and 3.2). Both stratified weights consistently outperformed the unstratified Monte Carlo with the optimal weights performing better than the Suboptimal weights. Despite the increasing run time, associated with increasing numbers of strata the additional variance reduction lead to a lower η.

Tab. 3.3: Comparison of sampling methods for an autocall option under CEV dynamics with parameters: $S_0 = 100,\ K = 110,\ H = 80,\ P = 100,\ C = 0.07,\ \sigma = 0.3,\ T = 1.5,\ r = 0,\ M = 365$.

Strata	Method	$V_0(S_0)$	σ^2	$t_p\ (s)$	$\eta\ (\times 10^3)$
20 Strata	Suboptimal	99.0548	41.9440	91.3624	61.9040
	Optimal	99.0623	21.7607	81.2248	42.0418
	Unstratified	99.0638	130.8190	38.9041	71.3400
50 Strata	Suboptimal	99.0634	37.0959	89.1493	57.5072
	Optimal	99.0523	18.9521	79.8258	38.8956
	Unstratified	99.0597	131.0787	39.2729	71.7485
100 Strata	Suboptimal	99.0607	33.6947	92.0874	55.7033
	Optimal	99.0570	15.1409	85.8173	36.0465
	Unstratified	99.0578	131.2843	41.7766	74.0581
200 Strata	Suboptimal	99.0517	31.1936	95.9731	54.7151
	Optimal	99.0573	13.5873	88.1257	34.6033
	Unstratified	99.0453	131.4336	43.6314	75.7273

3.4 Stratification in higher dimensions

To test the use of stratification in higher dimensions, four multivariate options were priced: an autocall option with dynamics governed by a mean reverted CEV-SV model, a call on the max with two stocks under GBM, a knock-in call on the max with two stocks under GBM and a knock-out call on the max with two stocks under the CEV model. Each option was priced with $1\ 000\ 000$ sample paths, $500\ 000$ for each random variable. In each case, both Brownian paths were stratified to the same extent, however a more judicious choice of stratification would likely increase the performance.

For the first option the stock paths were simulated under a CEV-SV model, with a mean reverting variance and dynamics defined by Lord (2008),

$$\begin{aligned} dS(t) &= \lambda\sqrt{V(t)}S(t)^{\frac{\beta}{2}}dW_S(t) \\ dV(t) &= -\kappa\left(V(t)-\theta\right)dt + \omega V(t)^{\alpha}dW_V(t), \end{aligned}$$

where $0 < \beta \leq 2$ and $\alpha > 0$ are elasticity parameters and λ, κ, ω are scaling

parameters. The paths were simulated according to the Euler scheme,

$$\begin{aligned} V(t_i) &= V(t_{i-1}) - \kappa\left(V(t_{i-1}) - \theta\right)(t_i - t_{i-1}) + \omega V(t_{i-1})^{\alpha}\Delta W_V(t_i)) \\ \ln S(t_i) &= \ln S(t_{i-1}) - \frac{1}{2}\lambda\sqrt{V(t_i)}S(t_{i-1})^{\beta-2}(t_i - t_{i-1}) \\ &\quad + \sqrt{V(t_i)}S(t_{i-1})^{\frac{\beta}{2}-1}\Delta W_S(t_i)). \end{aligned}$$

The results are shown in Table 3.4 overleaf.

Tab. 3.4: Comparison of sampling methods for an autocall option under CEV-SV dynamics with parameters: $S_0 = 100,\ \beta = 1.5,\ \theta = \sigma^2,\ \alpha = 0.9,\ \lambda = 1,\ \kappa = 0.8,\ \omega = 0.05,\ K = 110,\ H = 80,\ P = 100,\ C = 0.07,\ \sigma = 0.3,\ T = 1.5,\ r = 0,\ M = 365$

Strata	Method	$V_0(S_0)$	σ^2	t_p (s)	η $(\times 10^3)$
25 Strata	Suboptimal	97.2510	81.4892	80.6213	114.6278
	Optimal	97.2440	44.2443	83.0004	85.7005
	Unstratified	97.2489	170.2652	34.3368	108.1329
100 Strata	Suboptimal	97.2603	71.5886	77.9980	105.6765
	Optimal	97.2386	38.1362	79.6397	77.9379
	Unstratified	97.2397	170.3356	37.3387	112.7839
400 Strata	Suboptimal	97.2386	70.4715	81.1602	106.9531
	Optimal	97.2569	49.4434	86.6981	92.5921
	Unstratified	97.2569	170.1515	46.7331	126.1087
2500 Strata	Suboptimal	97.2523	80.0753	138.1083	148.7216
	Optimal	97.2508	78.7335	138.1506	147.4929
	Unstratified	97.2737	169.6352	88.6679	173.4428

For all strata sizes, the optimal allocation performed the best and the Suboptimal weights outperformed the unstratified Monte Carlo for 100 strata and above. In the case of 400 and $2\ 500$ strata, the η of the optimal allocation increases. This results from the necessity of a pilot run with at least 900 samples per stratum. As the number of strata increases, the number of optimal samples decreases and the number of samples in the pilot run increases. For the largest number of strata, the number of optimal samples is essentially negligible, and there is no significant reduction in variance over the Suboptimal sampling. For the remainder of the results, the $2\ 500$ strata results are omitted. The increasing run times with increasing number of strata results from the batched calculation of stock price paths (as previously discussed).

For the CEV-SV model, it is expected that the stochastic variance will have a lesser effect on the option price than the stock price. The variance reduction

achieved from the stratification of the variance will then not be as effective as that for a stratified stock price. Ideally, one would stratify the variance paths to a lesser extent, but this was not investigated in this dissertation.

Table 3.5 shows results for a call on the max with two stocks, with a payoff

$$H(X_T) = \max(S_T^1 - K, S_T^2 - K, 0),$$

where S_t^1 and S_t^2 are driven by GBM. Note that this payoff is not path dependent

Tab. 3.5: Comparison of sampling methods for a call on the max option with two stocks driven by GBM and parameters: $S_0 = [100\ 80]$, $K = 180$, $\sigma = [0.3\ 0.35]$, $T = 1.5$, $r = 0$, $M = 365$

Strata	Method	$V_0(S_0)$	σ^2	$t_p\ (s)$	$\eta\ (\times 10^3)$
	Unstratified	1.6755	103.6293	16.1244	40.8728
25 Strata	Suboptimal	1.6815	57.8846	62.5945	60.1935
	Optimal	1.6796	23.8794	64.3111	39.1881
100 Strata	Suboptimal	1.6767	54.5067	57.2201	55.8469
	Optimal	1.6882	20.9100	57.7563	34.7517
2500 Strata	Suboptimal	1.6833	26.1919	54.3275	37.7219
	Optimal	1.6906	16.8910	53.2801	29.9993

and the conversion from a Brownian path to a stock price path is extremely rapid. This is then a worst case scenario for functional quantization. Despite the apparently unsuitable option for functional stratification, the optimal sampling outperforms the unstratified in all cases. At $2\ 500$ strata, the Suboptimal weights outperform the unstratified Monte Carlo. The poor performance of the Suboptimal allocation is likely a result of the low moneyness of the option.

Table 3.6 shows results for a knock-in call on the max with two stocks, with payoff

$$H(X_T) = \max(S_T^1 - K, S_T^2 - K, 0)\mathbb{1}_{max_{t\leq T}(S_t^1, S_t^2)\geq B},$$

where S_t^1 and S_t^2 are driven by GBM. Note that this option also has a relatively low moneyness due to the high knock-in boundary.

The optimal sampling provides a lower η than the unstratified for all strata sizes, however, the Suboptimal weights require a much higher number of strata to outperform the unstratified Monte Carlo. Once again, the poor performance of the Suboptimal allocation is likely a result of a combination of the low moneyness of the option and the use of GBM to convert the Brownian paths to stock price paths.

Table 3.7 shows results for a knock-out call on the max with two stocks, with payoff

$$H(X_T) = \max(S_T^1 - K, S_T^2 - K, 0)\mathbb{1}_{max_{t\leq T}(S_t^1, S_t^2)\geq B},$$

Tab. 3.6: Comparison of sampling methods for a call on the max with knock-in option with two stocks driven by GBM and parameters: $S_0 = [100\ 110],\ K = 105,\ B = 180,\ \sigma = [0.3\ 0.35],\ T = 1.5,\ r = 0,\ M = 365$

Strata	Method	$V_0(S_0)$	σ^2	$t_p\ (s)$	$\eta\ (\times 10^3)$
	Unstratified	19.3741	1710.8299	17.0076	241.2186
25 Strata	Suboptimal	19.4717	684.8206	63.7959	295.5967
	Optimal	19.4461	393.3842	65.1477	226.3983
100 Strata	Suboptimal	19.4994	605.7690	58.2307	265.6099
	Optimal	19.4413	342.5059	57.8198	199.0157
2500 Strata	Suboptimal	19.4500	466.6827	55.2160	227.0169
	Optimal	19.4547	342.5081	53.9130	192.1751

where S_t^1 and S_t^2 are driven by a CEV model. In this, the unstratified Monte Carlo

Tab. 3.7: Comparison of sampling methods for a call on the max with knock-out option with two stocks driven by CEV and parameters: $S_0 = [100\ 110],\ K = 105,\ B = 140,\ \beta = 1.5,\ \sigma = [0.3\ 0.35],\ T = 1.5,\ r = 0,\ M = 365$

Strata	Method	$V_0(S_0)$	σ^2	$t_p\ (s)$	$\eta\ (\times 10^3)$
25 Strata	Suboptimal	1.8911	28.8798	92.0168	72.9031
	Optimal	1.8865	10.8430	97.1850	45.9081
	Unstratified	1.8774	67.2457	46.1181	78.7559
100 Strata	Suboptimal	1.8878	26.8273	89.4118	69.2630
	Optimal	1.8949	9.7059	89.3916	41.6564
	Unstratified	1.8867	67.8311	47.7068	80.4488
400 Strata	Suboptimal	1.8855	19.0897	90.4038	58.7499
	Optimal	1.8942	10.0544	91.5073	42.8965
	Unstratified	1.8856	67.7004	52.3272	84.1733

method was outperformed by both the Suboptimal and optimal weights, the reason being the computationally expensive use of the CEV model. Unlike in Table 3.6, the Monte Carlo simulation requires a similar run time to the stratified, as the difference in time arising from the path generation is mitigated by the expensive conversion of Brownian paths to stock price paths. In the 400 strata case, the optimal sampling performance decreases. This is a result of the increasing variance associated with the increased size of the pilot run as well as a decrease in the extent to which the calculation is vectorized.

3.5 American options

In the context of the Longstaff Schwartz method, it no longer makes sense to calculate the variance as outlined in Section 2.4, as this requires that the prices in each stratum are independent of those in other stratums and that is no longer the case. To compare the efficiency of the stratification, then, it was necessary to repeatedly calculate the price of the American option and determine a Monte Carlo estimate of the variance of the price. Note that the variance mentioned in Section 2.4 was the variance of the payoffs and not of the price. Once again, it was assumed that the efficiency of the stratification could be estimated by $\eta = \sigma\sqrt{\frac{t_p}{N}}$, except in this case σ is the volatility of the price (as opposed to the payoffs) and N is the number of repetitions in the price calculation.

In the case of optimal sampling, it is likely that no samples were drawn from several strata. This means that these samples will not feature in the Longstaff Schwartz pricing method. The regression is then only over a subset of the possible paths. Despite the fact that only paths with a positive payoff at the relevant time slice are considered in the regression, one cannot say with certainty that entirely out of the money strata are never in the money. Certainly in the case of an at the money or in the money put, one expects that at least some paths from the out of the money stratum would have a positive exercise value. It is then likely that the use of optimal sampling will induce some bias in the regression in the Longstaff Schwartz method, especially near inception. In an attempt to remove this bias, a new sampling method was introduced: the floored optimal. The floored optimal weights are simply the optimal weights but with a minimum number of samples in each stratum. The following were calculated with a floor equal to 20% of the samples in the highest weighted stratum. Each price calculation was repeated 200 times with $100\ 000$ sample paths.

Table 3.8 below shows the results for an American put option with parameters $S_0 = 40,\ \sigma = 0.4,\ r = 0,\ K = 40,\ T = 1,\ N_{div} = 365.$

Firstly note that all of the prices are consistent with each other and the unstratified Monte Carlo price. The Suboptimal sampling method has the lowest average run time, followed by the optimal and then the floored optimal. The unstratified Monte Carlo sample size was selected so that its run time was similar to the stratified run times. The increase in run time for the optimal sampling can be explained by the additional calculations required to calculate the optimal weights, as well as the necessity for a pilot run. While the total sample size for the optimal weights and Suboptimal are the same, the additional call to the pricing function from the pilot run means that the algorithm is not vectorized to the same extent. The floored

Tab. 3.8: Comparison of sampling methods for an at the money American option: $S_0 = 40,\ \sigma = 0.4,\ r = 0,\ K = 40,\ T = 1,\ N_{div} = 365$

Strata	Method	$V_0(S_0)$	$\sigma^2\ (\times 10^{-5})$	$t_p\ (s)$	$\eta\ (\times 10^{-3})$
	Unstratified	6.3022	50.7755	5.7173	3.8085
20 Strata	Suboptimal	6.3010	19.5075	4.1909	2.0218
	Optimal	6.2964	66.9657	4.3675	3.8241
	Floored optimal	6.3026	16.8242	5.1787	2.0872
50 Strata	Suboptimal	6.3036	16.6613	4.0755	1.8426
	Optimal	6.2955	26.5795	4.3008	2.3907
	Floored optimal	6.3022	14.2111	5.8469	2.0383
100 Strata	Suboptimal	6.3022	14.1165	4.0760	1.6962
	Optimal	6.2964	44.0176	4.2651	3.0638
	Floored optimal	6.3008	13.9443	6.0440	2.0528
200 Strata	Suboptimal	6.3016	14.9349	4.0228	1.7332
	Optimal	6.3040	54.4275	4.0551	3.3220
	Floored optimal	6.3005	15.7703	5.7967	2.1379

optimal run time was significantly higher as the floor was simply added to the existing optimal allocation, increasing the sample size. The η of the sampling methods shows that the Suboptimal allocation performs best, followed by the floored optimal and then the unstratified approaches. The optimal weights had a relatively high and unstable η. The increase in volatility for the optimal weights can be explained by considering strata with very few samples. Points in a stratum with few samples will have a large weight in the Longstaff Schwartz least squares procedure. These points lead to a high variance in the estimation of the continuation values, causing a high variance in the prices. Further evidence of this explanation can be seen in the lower η of the floored optimal. Introducing the floor prevents any one strata from having few points with overly large weights, reducing the variance associated with the estimation of the continuation values. Despite the low variance of the floored optimal approach, the slow run time meant it was less efficient than the Suboptimal approach. The use of the floored optimal served as a means to validate the idea that the high variance in the optimal sampling approach was a result of highly weighted points in the estimation of the continuation values. As a result of their poor performance, the floored optimal and optimal weightings received no further investigation.

Table 3.9 below shows the results of the simulations for an out of the money American put option with parameters $S_0 = 40,\ \sigma = 0.4,\ r = 0,\ K = 20,\ T =$

1, $N_{div} = 365$ at four levels of stratification.

Tab. 3.9: Comparison of sampling methods for an out of the money American option: $S_0 = 40,\ \sigma = 0.4,\ r = 0,\ K = 20,\ T = 1,\ N_{div} = 365$

Strata	Method	$V_0(S_0)$	$\sigma^2\ (\times 10^{-7})$	$t_p\ (s)$	$\eta\ (\times 10^{-3})$
	Unstratified	0.18804	164.4358	1.2131	0.31553
20 Strata	Suboptimal	0.1879	65.5972	2.9103	0.3090
	Optimal	0.1867	14.2829	3.3338	0.1543
50 Strata	Suboptimal	0.1878	58.6451	2.8132	0.2872
	Optimal	0.1866	8.3601	3.2637	0.1168
100 Strata	Suboptimal	0.1878	41.9556	2.8263	0.2435
	Optimal	0.1865	9.4661	3.2316	0.1237
200 Strata	Suboptimal	0.1879	42.9506	2.7826	0.2445
	Optimal	0.1866	9.6326	2.9991	0.1202

The optimal sampling was included in Table 3.9 above to demonstrate the bias in the price arising from the missing strata. The price obtained by the optimal weights is consistently lower than both the unstratified and Suboptimally stratified approaches. The bias is much more apparent in this example than in Table 3.8, as very few strata were in the money; in the 200 strata case only four strata were sampled. The high proportion of missing strata had a much larger influence on the bias than what was observed in Table 3.8. While the variance of the optimal sampling was much more stable in this example it was a result of equally weighted strata, which is a special case (and a fluke) and is by no means representative of the optimal sampling methods stability.

The Suboptimal allocation showed no bias in the price (as expected) and provided a more significant reduction in the variance than what was seen in Table 3.8. In both Table 3.8 and Table 3.9, the lowest η was achieved at 100 strata; the subsequent increase at 200 strata could be a result of an insufficient sample size for the number of strata (i.e., strata with few highly weighted points). This suggests the existence of an optimal number of strata for a given sample size, however this was not investigated. This optimal number of strata was a feature unique to the pricing of American options as all other options tested did not have any interaction between strata in the pricing function.

The high efficiency of the suboptimal sampling when compared to the unstratified Monte Carlo is a result of the computationally intense payoff function, the Longstaff Schwartz least squares Monte Carlo. The more computationally intense the payoff function and conversion from a Brownian path to a stock path, the less

noticeable the additional time associated with generating conditional paths will be. This can also be seen in a comparison between Tables 3.2 and 3.3 where the expensive Euler discretization masks the difference in time taken to generate paths between the unstratified Monte Carlo and the stratified approach.

Chapter 4

Summary and Conclusion

The randomized Lloyd's splitting method, for the generation of optimal grid quantizers, was implemented and validated against those given in Corlay *et al.* (2005). This algorithm proved to be capable of matching the optimal quantizer presented in literature in the two-dimensional case. In the three-dimensional case, a different local minimum was obtained, however the distortion was equal to that of the literature case and it was concluded that the splitting method was suitable for use in this dissertation.

The optimal quantizer grids were then converted to functional quantizers and, once again, compared to those presented in Corlay *et al.* (2005). The two dimensional case was equal to the literature case. While the three dimensional case was aesthetically different, the distortion was equal to the literature case. It was concluded that the generation of the functional quantizer was working as expected and that the randomized Lloyd's splitting method was an acceptable method for generating the grids required for functional quantization.

The Bayesian algorithm for the simulation of Brownian paths, conditional on their association with a particular functional quantizer, was implemented. The distributional properties of these conditional paths were validated and shown to be consistent with Brownian motion.

Three options were then priced at varying levels of stratification. The prices were validated against those given Corlay and Pagès (2015). A metric, η, was proposed to reasonably measure the efficiency of a given method of stratification. This metric closely resembled the levels of efficiency evident in plots comparing the size of the confidence interval at varying times. Using this new metric, it was shown that, in all cases, optimally allocated strata performed as well as or better than an unstratified Monte Carlo estimate. Suboptimal allocations were unable to consistently outperform the unstratified Monte Carlo, especially in the case of out of the money options. Additionally, it was noted that for cases where the conversion of Brownian paths to stock price paths were computationally expensive (as in the case

of the CEV model), the performance of the stratified Monte Carlo relative to the unstratified Monte Carlo was improved.

The use of functional stratification in two dimensions was then investigated and it was shown that, as in one dimension, the optimal allocation was able to consistently outperform the unstratified estimator. Once again, the suboptimal weights were unable to consistently outperform the unstratified Monte Carlo except in the case where the CEV model was used.

Two American options were priced using a stratified version of the Longstaff Schwartz method. It was noted that the use of optimal allocation was likely to induce an increase in variance as well as a bias in the price. The increase in variance was shown to be a result of highly weighted points in the regression in the Longstaff Schwartz algorithm. This was confirmed through the introduction of a floored optimal which ensured no stratum had too few points. In both test cases, the suboptimal allocation led to a significant increase in performance, even in the case of an out of the money option. The strong performance of the suboptimal allocation was attributed to the computationally expensive payoff function.

Functional stratification then allows for an efficient and reliable method for variance reduction for path dependent options.

Bibliography

Corlay, S. (2011). *Quelques aspects de la quantification optimale et applications la finance*, PhD thesis, de l'universit Pierre et Marie Curie.

Corlay, S. and Pagès, G. (2015). Functional quantization-based stratified sampling methods, *Monte Carlo Methods and Applications* **21**(1): 1–32.

Corlay, S., Pagés, G. and Printems, J. (2005). The optimal quantization website. **URL:** *http://www.quantize.math-fi.com*

Cuperman, V. and Gersho, A. (1982). Adaptive differential vector coding of speech, *Conf. Rec. IEEE GLOBECOM* .

Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering*, Springer.

Graf, S., Luschgy, H. and Pagés, G. (2007). Optimal quantizers for random vectors in a Banach space, *J. Approx. Theory* **144**(1).

Gray, R. (1984). Vector quantization, *IEEE ASSP Magazine* **1**(2): 4 – 29.

Gray, R. M., Kieffer, J. C. and Linde, Y. (1980). Locally optimal block quantizer design, *Information and Control* **45**: 178–198.

Keller, A., Heinrich, S. and Niederreiter, H. (2006). *Monte Carlo and Quasi-Monte Carlor Methods 2006*, Springer.

Kleijnen, J. P. C., Riddler, A. A. N. and Rubinstein, R. Y. (2010). Variance Reduction Techniques in Monte Carlo Methods, *CentER Discussion Paper Series* (2010-117).

Linde, Y., Buzo, A. and Gray, R. (1980). An Algorithm for Vector Quantizer Design, *IEEE Transactions on Communications* **28**(84).

Longstaff, F. A. and Schwartz, E. S. (2001). Valuing American Options by Simulation: a Simple Least-Squares Approach, *Review of Financial Studies* **14**(1).

Lord, R. (2008). *Efficient pricing algorithms for exotic derivatives*, PhD thesis, Erasmus Universiteit Rotterdam.

Luschgy, H. and Pagès, G. (2002). Functional quantization of Gaussian processes, *Journal of Functional Analysis* **196**(2).

Luschgy, H. and Pagés, G. (2012). Critical dimension for quadratic functional quantization, *hal-007731523* .

McWalter, T. A., Rudd, R., Kienitz, J. and Platen, E. (2016). Recursive marginal quantization of higher-order schemes.
URL: *https://arxiv.org/abs/1701.02681*

Pagès, G. (2014). Introduction to optimal vector quantization and its applications for numerics.

Pagès, G., Pham, H. and Printems, J. (2003). *Optimal quantization methods and applications to numerical problems in finance*, Birkhuser Boston, chapter 1, pp. 253–297.

Pagés, G. and Printems, J. (2000). Optimal quadratic quantization for numerics: the Gaussian case.

Appendix A

Additional Figures

A.1 Option price convergence and efficiency

Each simulation used a uniform time grid with 365 divisions. Prices and the corresponding variances were calculated using optimal and sub optimal (natural) weights as well as standard Monte Carlo with no stratification. This was done for 20, 50, 100 and 200 strata and $1000 \leq N \leq 100000$. The prices obtained were plotted against N with a 99.7% confidence interval around the literature value for each weighting method.

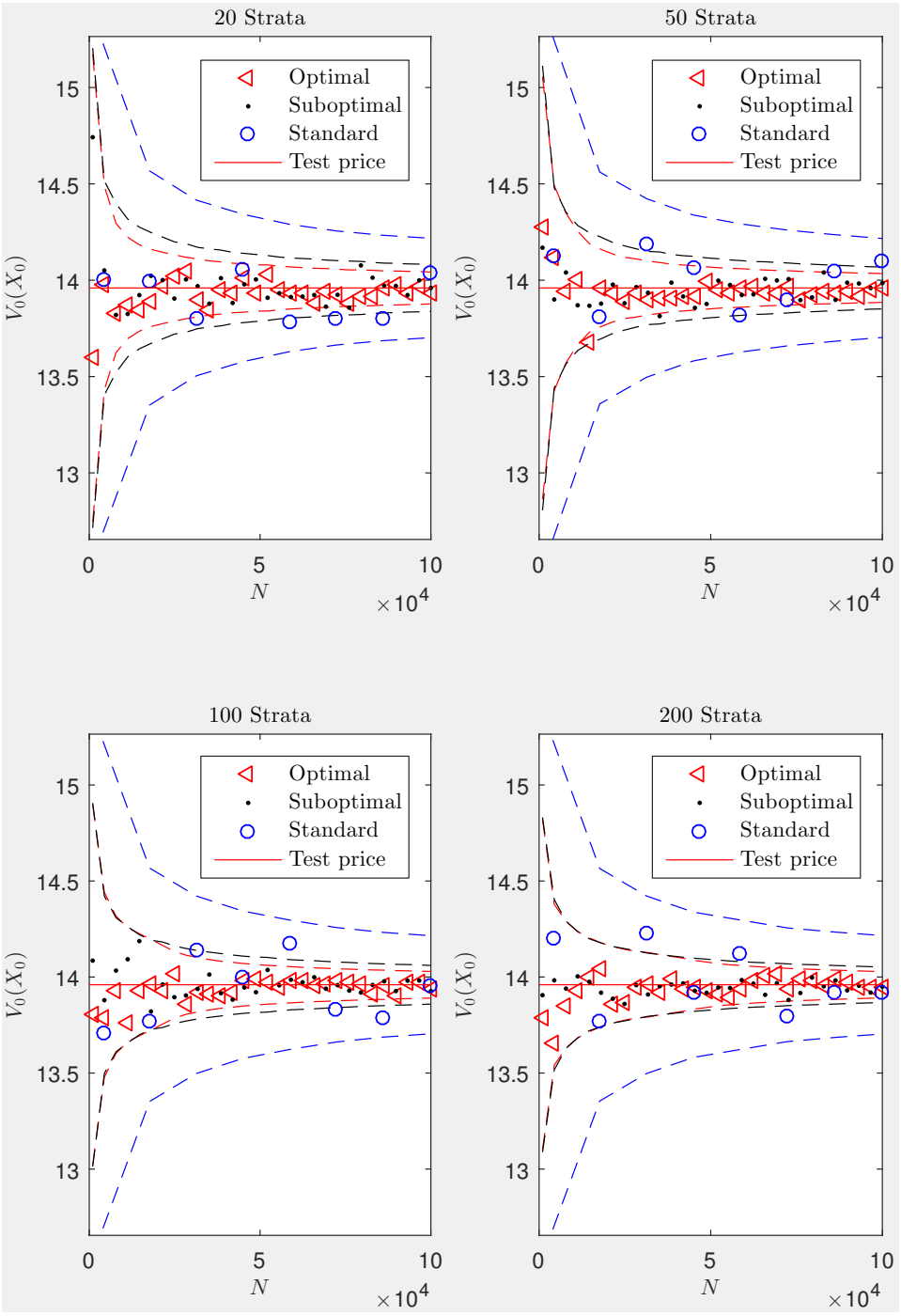

Fig. A.1: Validation of stratification for the pricing of an up-and-in call option with dynamics governed by GBM: $S_0 = 100,\ K = 100,\ H = 125,\ \sigma = 0.3,\ T = 1.5,\ r = 0,\ M = 365$

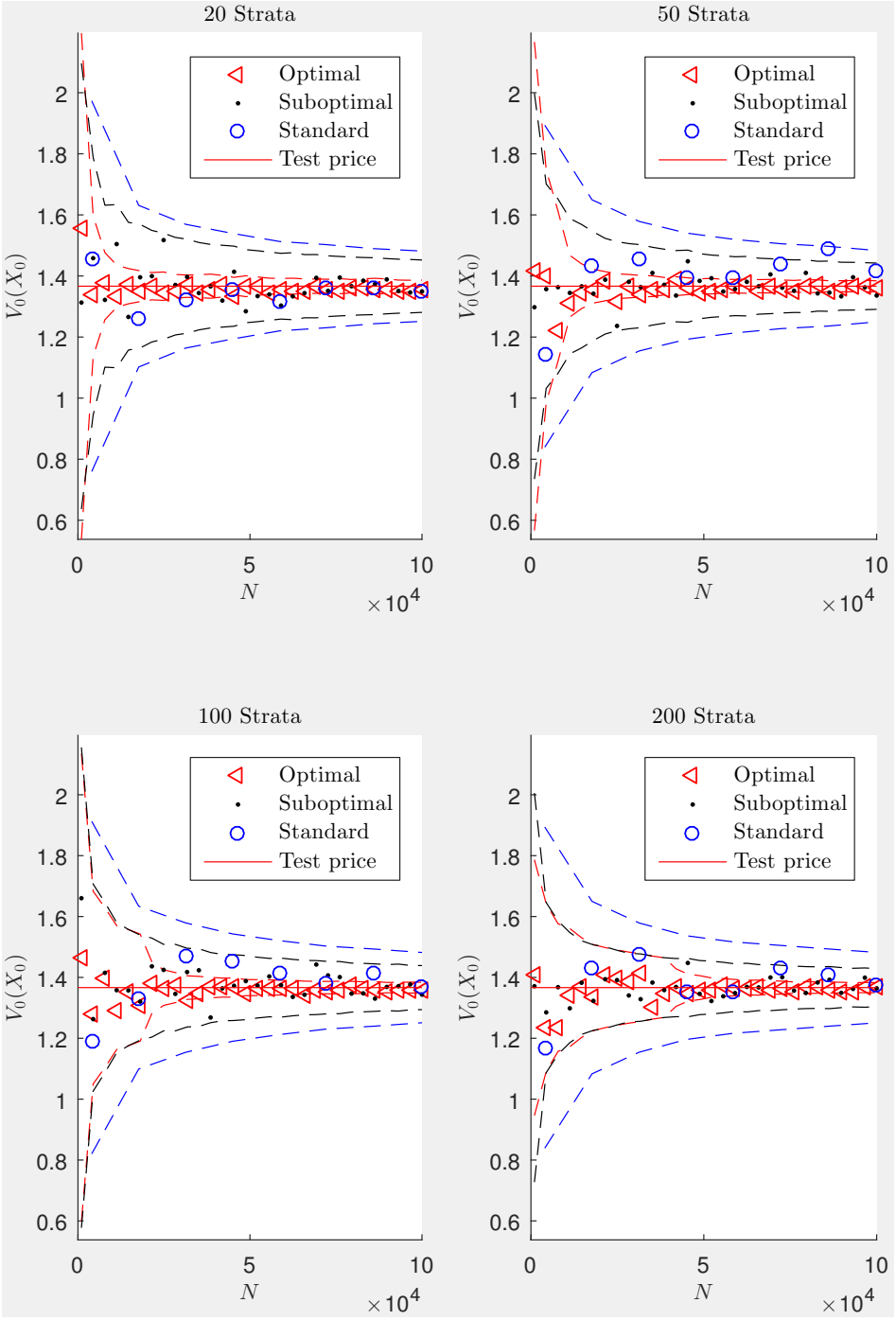

Fig. A.2: Validation of stratification for the pricing of an up-and-in call option with dynamics governed by GBM: $S_0 = 100,\ K = 100,\ H = 200,\ \sigma = 0.3,\ T = 1,\ r = 0,\ M = 365$

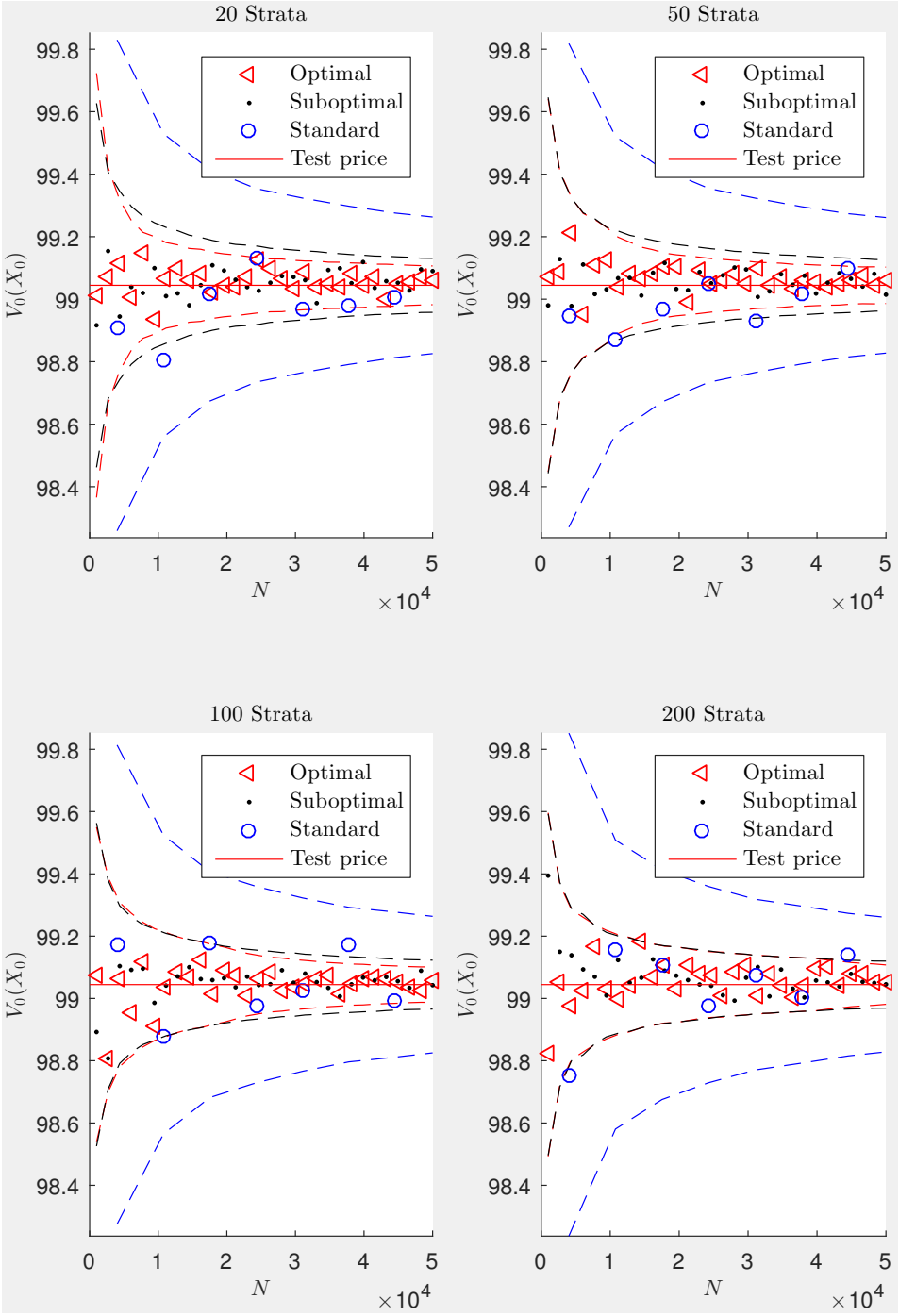

Fig. A.3: Validation of stratification for the pricing of an autocall option with dynamics governed by the CEV model: $S_0 = 100,\ K = 110,\ H = 80,\ P = 100,\ C = 0.07,\ \sigma = 0.3,\ T = 1.5,\ r = 0,\ M = 365$

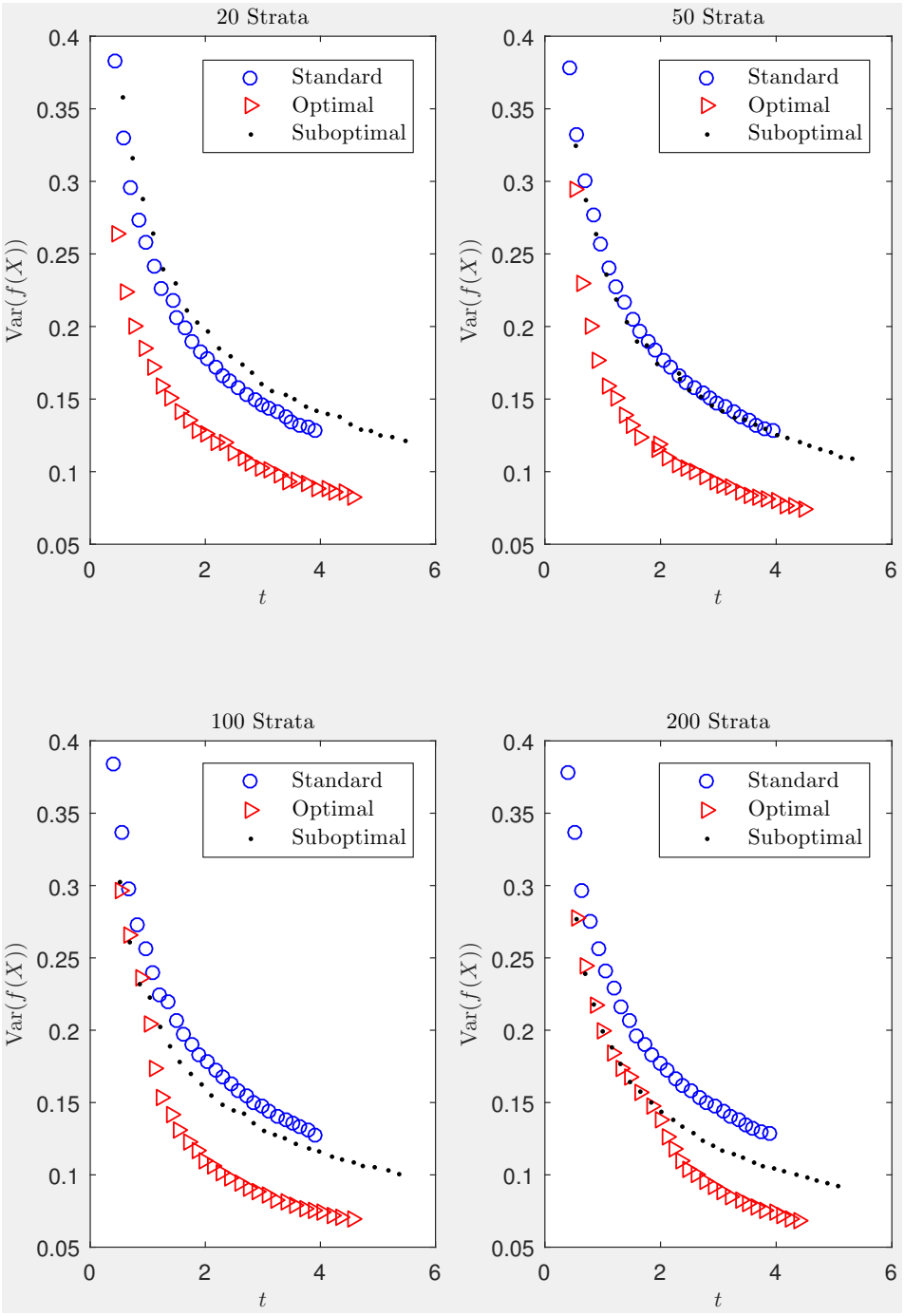

Fig. A.4: Efficiency of stratification for the pricing of an up-and-in call option with dynamics governed by GBM: $S_0 = 100,\ K = 100,\ H = 125,\ \sigma = 0.3,\ T = 1.5,\ r = 0,\ M = 365$

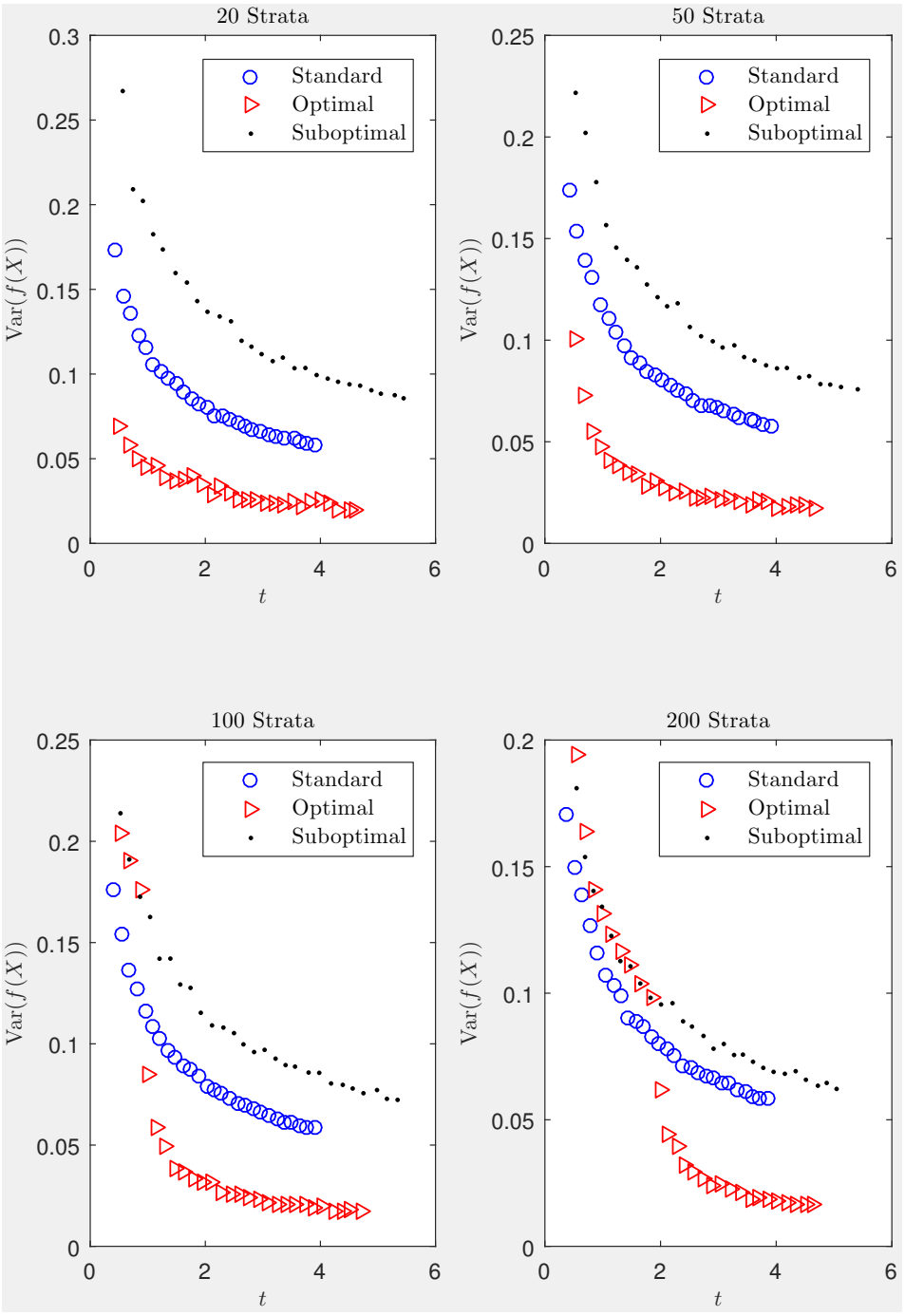

Fig. A.5: Efficiency of stratification for the pricing of an up-and-in call option with dynamics governed by GBM: $S_0 = 100,\ K = 100,\ H = 200,\ \sigma = 0.3,\ T = 1,\ r = 0,\ M = 365$

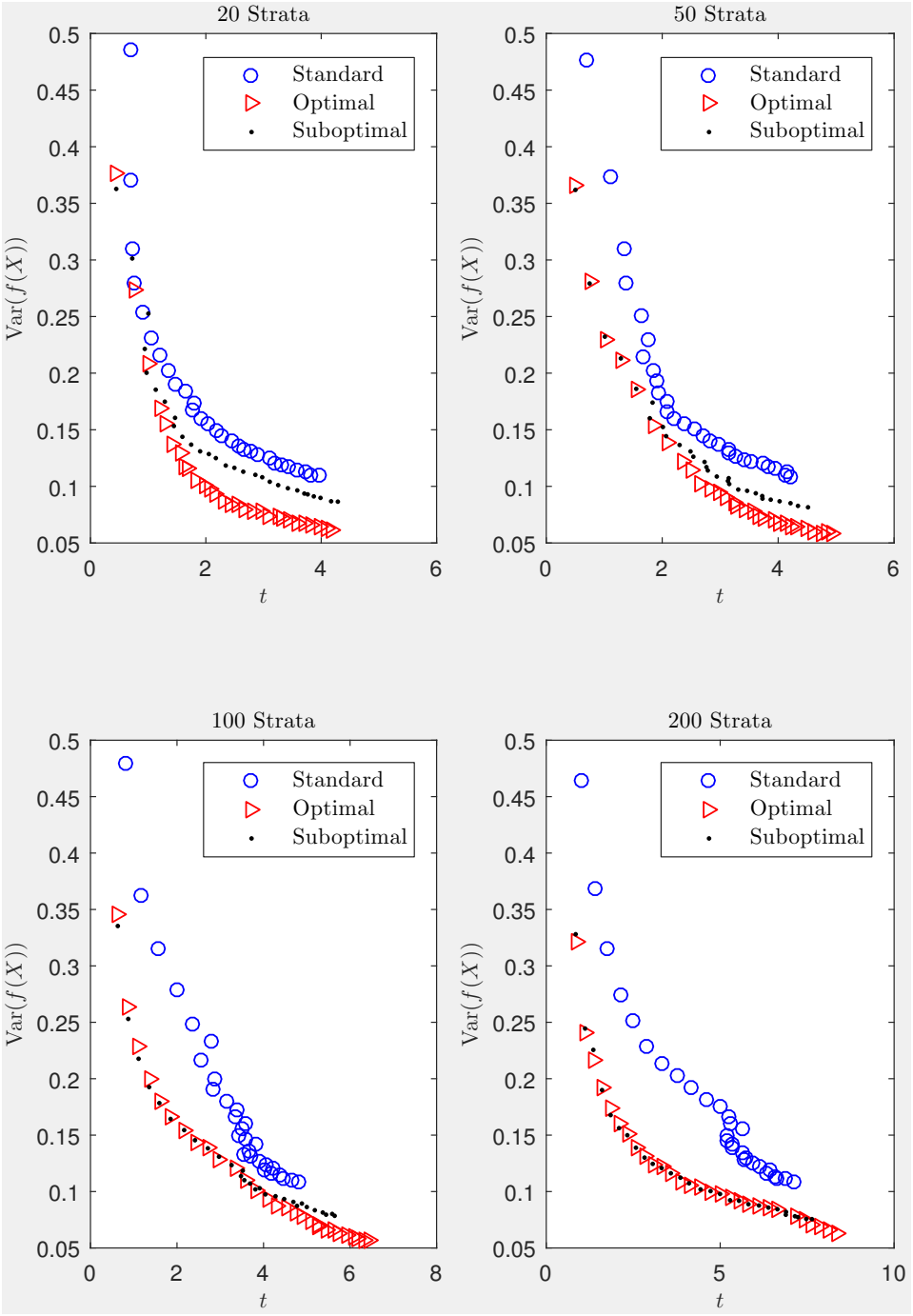

Fig. A.6: Efficiency of stratification for the pricing of an auto-call option with dynamics governed by the CEV model:
$S_0 = 100,\ K = 110,\ H = 80,\ P = 100,\ C = 0.07,\ \sigma = 0.3,\ T = 1.5,\ r = 0,\ M = 365$