

Université de Montréal

Stochastic mesh approximations for dynamic hedging with costs

par
Pierre-Alexandre Tremblay

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en recherche opérationnelle

Juillet, 2017

© Pierre-Alexandre Tremblay, 2017.

CONTENTS

CONTENTS	ii
LIST OF TABLES	vi
LIST OF FIGURES	viii
LIST OF ABBREVIATIONS	xiv
DEDICATIONxviii
ACKNOWLEDGMENTS	xix
CHAPTER 1: INTRODUCTION	1
1.1 General overview	1
1.2 A basic example of dynamic hedging with costs	4
1.3 Dynamic Programming on a Stochastic Mesh : main ideas	11
1.4 Related works	13
1.4.1 Some important works on the hedging problem with costs	13
1.4.2 Monte Carlo based algorithms for derivative hedging	15
1.5 Plan of the thesis and main contributions	16
CHAPTER 2: STRUCTURE OF SOLUTIONS	20
2.1 Problem definition	20
2.1.1 Formulation as a Dynamic Program	20
2.1.2 State transition function	23
2.1.3 Risk measures	25
2.1.4 Derivative pricing	28
2.2 Dynamic Programming algorithm	29

2.3	General form of solutions	31
2.3.1	A direct approach	32
2.3.2	The Karush-Kuhn-Tucker conditions	36
2.4	Results specific to the negative exponential loss function	39
2.5	Existence of solutions : measurability conditions	42
2.6	Conclusion	44

**CHAPTER 3: APPROXIMATE DYNAMIC PROGRAMMING WITH
A STOCHASTIC MESH 45**

3.1	Approximate Dynamic Programming	47
3.1.1	Simulating states	47
3.1.2	Approximating J_k^*	49
3.2	Stochastic mesh methods	50
3.2.1	Basic construction	50
3.2.2	Application to the hedging problem	51
3.2.3	Choice of mesh density	53
3.2.4	Using simulated derivative values	55
3.3	Approximation of the risk function dependence on the stock quantity	57
3.3.1	A linear-quadratic approximation	57
3.3.2	A bound on the approximation error	59
3.4	Confidence interval for the optimal value	65
3.5	Convergence of mesh estimators	68
3.6	Conclusion	69

CHAPTER 4: PRICE DYNAMICS AND HEURISTIC POLICIES 71

4.1	The exponential Ornstein-Uhlenbeck stochastic volatility model	71
4.1.1	Basic form	72
4.1.2	An extension using additional information variables	75

4.2	Some heuristic policies	78
4.2.1	Whalley-Wilmott hedging policy	79
4.2.2	Parametric approximation of Zakamouline	81
4.2.3	One step lookahead and local hedging	82
4.2.4	Local hedging approximation	85
4.3	Conclusion	87

CHAPTER 5: NUMERICAL RESULTS FOR THE BASIC MESH

	CONSTRUCTION	89
5.1	MeshHedging : a Java library for computing stochastic mesh based hedging policies	89
5.2	Risk estimation algorithm	92
5.3	Illustration of the convergence of mesh estimators	94
5.4	Global performance comparison of some heuristic policies using simulated data	96
5.5	Conclusion	103

CHAPTER 6: EFFICIENCY IMPROVEMENTS 106

6.1	Control variates	107
6.1.1	Outer controls	108
6.1.2	Inner controls	111
6.2	Using a single grid	118
6.3	Quasi-Monte Carlo methods	121
6.4	Removing small weights	124
6.5	Combining the single grid and Russian roulette methods	128
6.6	Other relevant techniques not tested here	132
6.7	Conclusion	137

CHAPTER 7: EMPIRICAL COMPARISON OF HEDGING POLI-	
 CIES	139
7.1 Global performance comparison for various parameter combinations	140
7.2 Effect of various parameters on ϵ -optimal policies	141
7.2.1 Transaction costs b	145
7.2.2 Number of time steps K	145
7.2.3 Stock price s_0	147
7.2.4 Stock quantity u_0	151
7.2.5 Initial volatility σ_0 , mean volatility level $\bar{\sigma}$ and volatility of log-volatility σ_v	151
7.2.6 Time horizon T	156
7.2.7 Rate of log-volatility mean reversion κ	159
7.2.8 Correlation ρ	161
7.2.9 Risk aversion γ	161
7.3 Conclusion	166
CHAPTER 8: CONCLUSION	169
BIBLIOGRAPHY	171

LIST OF TABLES

4.I	Moment-based estimates of the parameters of the expOU model for various historical periods.	73
5.I	Estimates of the portfolio risk when hedging a call option under the exponential loss function and the discrete GBM model for various combinations of the parameters γ , σ , K and b . Mesh based estimates using $N = 512$ points.	101
5.II	Estimates of the portfolio risk when hedging a call option under the exponential loss function and the discrete expOU model for various combinations of the parameters γ , κ , σ_v and ρ (with $N = 512$, $K = 8$, $\sigma_0 = 40\%$, $\bar{\sigma} = 20\%$ and $b = 2\%$).	104
6.I	Empirical variance reduction factors when estimating \widehat{J}_0^π using the control variate $V_K^{0,\pi}$ with the coefficient $\beta = -\gamma$, under the exponential loss function and the discrete GBM model, for various policies π ($N = 256$, $n = n_R = 100$).	109
6.II	Estimates $\widehat{J}_0^\pi(X_0)$ of the portfolio risk when hedging a call option. The setup is the same as for table 5.II, but this time using control variates for estimating h_k values.	114
6.III	Computing times (CT, in seconds per replication) and 95%-level confidence interval for the efficiency improvement factor (using $n_R = 100$ replications, GBM model, $N = 1024$ mesh paths, $\sigma = 20\%$, $T = 6$ months, $\gamma = 1$ and $b = 2\%$).	133

6.IV	Computing times (CT, in seconds per replication) and 95%-level confidence interval for the efficiency improvement factor (using $n_R = 100$ replications, $N = 1024$ mesh paths, expOU model with $\sigma = 20\%$, $T = 6$ months, $\gamma = 1$ and $b = 2\%$).	134
6.V	Computing times (CT, in seconds per replication) and 95%-level confidence interval for the efficiency improvement factor (using $n_R = 100$ replications, $N = 512$ mesh paths, expOU-I model with $\sigma = 20\%$, $T = 6$ months, $\gamma = 1$ and $b = 2\%$).	135
7.I	Estimates of the portfolio risk for hedging a call option under the negative exponential loss function and the discrete expOU model. Setup : $T = 3$ months, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 2.5$, $\sigma_0 = \bar{\sigma} = \sigma$, $N = 1024$ mesh points and $n = n_R = 100$	142
7.II	Estimates of the portfolio risk for hedging a put option under the negative exponential loss function and the discrete expOU model. Setup : $T = 3$ months, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 2.5$, $\sigma_0 = \bar{\sigma} = \sigma$, $N = 1024$ mesh points and $n = n_R = 100$	143
7.III	Estimates of the portfolio risk when hedging a call option under the negative exponential loss function and the discrete expOU model. Setup : $T = 3$ months, $K = 64$ $\sigma_0 = \bar{\sigma} = 16\%$, $b = 1\%$, $N = 2048$ mesh points, and $n = n_R = 100$	144

LIST OF FIGURES

1.1	Sample paths from the geometric Brownian motion process, for $K = 64$ steps, $T = 3$ months, initial price $s_0 = 10$, drift $\mu = 0$ and volatility $\sigma = 16\%$	7
1.2	Top : Initial value and payoff of a standard call option expiring in 3 months with strike $X = 10$, under the BSM model with $r = q = 0$ and $\sigma = 16\%$. Bottom : Delta of the option (same parameters).	8
1.3	Sample paths of the portfolio value $V_k^\pi(X_0)$ and of the policy decisions u_k . Left-hand side : $\pi = \text{BSM}$, right-hand side : $\pi = Z$ (for $\gamma = 1$). Setup : $s_0 = X = 10$, $\sigma = 16\%$, $T = 3$ months, $K = 64$, $b = 1\%$	10
3.1	Estimates of the expected one step risk $\mathbf{E}_0[g_1(X_0, v, Y_1)]$, expected risk function approximation $\mathbf{E}_0[\widehat{J}_1(\psi(X_0, v, Y_1))]$ and Q -function approximation $\widehat{Q}_0(X_0, v)$ as a function of the trading decision v , for a call option under the GBM model ($s_0 = 10$, $\sigma = 40\%$, $b = 2\%$, $\gamma = 5$ and $K = 8$) and a stochastic mesh with $N = 1024$ paths. The product $\mathbf{E}_0[g_1(X_0, v, Y_1)]\mathbf{E}_0[\widehat{J}_1(\psi(X_0, v, Y_1))]$ is also indicated in comparison to $\widehat{Q}_0(X_0, v)$	60
3.2	Approximation error bound estimate $\widehat{\epsilon}_0$ as a function of N , for a call option under the GBM model with $s_0 = 10$, $\sigma_0 = 0.2$, $T = 6$ months, $K = 8$, $\gamma = 1$ and $b = 0.2\%$	63
3.3	Approximation error bound estimate $\widehat{\epsilon}_0$ as a function of K , for a call option under the GBM model with $s_0 = 10$, $\sigma_0 = 0.2$, $T = 6$ months, $N = 256$, $\gamma = 1$ and $b = 0.2\%$	64

5.1	Risk estimates $\widehat{J}_0(X_0)$ as a function of N , for a call option under the GBM model ($\sigma_0 = 20\%$, $b = 2\%$, $\gamma = 1$, $K = 8$).	97
5.2	Risk estimates $\widehat{J}_0(X_0)$ as a function of N , for a call option under the GBM model ($\sigma_0 = 40\%$, $b = 2\%$, $\gamma = 5$, $K = 8$).	97
5.3	Risk estimates $\widehat{J}_0(X_0)$ as a function of N , for a call option under the expOU model ($\sigma_0 = 40\%$, $\bar{\sigma} = 20\%$, $\sigma_v = 1.2$, $\rho = 0.5$, $\kappa = 2.6$, $b = 2\%$, $\gamma = 1$, $K = 8$).	98
5.4	Risk estimates $\widehat{J}_0(X_0)$ as a function of N , for a call option under the expOU model ($\sigma_0 = 40\%$, $\bar{\sigma} = 20\%$, $\sigma_v = 1.2$, $\rho = 0.5$, $\kappa = 2.6$, $b = 2\%$, $\gamma = 5$ and $K = 8$).	98
6.1	Inner and outer risk estimates $\widehat{J}_0(X_0)$ as a function of N , using control variates, for a call option under the expOU model ($\sigma_0 = 40\%$, $\bar{\sigma} = 20\%$, $\sigma_v = 1.2$, $\kappa = 2.6$, $\rho = -0.5$, $b = 2\%$, $\gamma = 1$, $K = 8$). Error bars indicate one standard deviation.	117
6.2	Inner and outer risk estimates \widehat{J}_0 as a function of N , using control variates, for a call option under the expOU model ($\sigma_0 = 40\%$, $\bar{\sigma} = 20\%$, $\sigma_v = 1.2$, $\kappa = 2.6$, $\rho = -0.5$, $b = 2\%$, $\gamma = 5$, $K = 8$). Error bars indicate one standard deviation.	117
6.3	Risk estimate $\widehat{J}_0(X_0)$ as a function of N , for $K = 16$ (above) and $K = 64$ (below). Setup : call option under the expOU model, with $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $b = 1\%$, $\gamma = 1$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.	125

6.4	Sample average estimates of cost function approximations $\widehat{J}_0(X_0)$ as a function of the base-2 logarithm $\log_2 C$ of the total policy computation time. Error bars indicate one standard deviation of the sample averages. The results are for a call option under the GBM model with $s_0 = 10$, $\sigma_0 = 20\%$, $T = 6$ months, $K = 64$, $b = 2\%$ and $\gamma = 1$	130
7.1	Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the transaction cost parameter b . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $\gamma = 1$, $N = 1024$ and $n = 100$, $n_R = 1000$. Error bars indicate one standard deviation of the sample averages.	146
7.2	Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the number of time steps K . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $b = 1\%$, $\gamma = 1$, $N = 2048$ for the call and $N = 4096$ for the put, and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.	148
7.3	Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the number of time steps K . The setup is similar as in figure 7.2, but only for the call option, with the transaction costs and risk aversion increased to $b = 2\%$ and $\gamma = 5$, respectively. Using $N = 1024$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages. . .	149

- 7.4 Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the stock price s_0 . Left-hand side : put option, right-hand side : call option. Setup : $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 4096$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages. 150
- 7.5 No hedging boundaries for the initial decision $\mu_1(X_0)$ for various policies as a function of the stock price s_0 , for a call option (above) and a put option (below) under the expOU model. Setup : $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 1024$ and $n = n_R = 100$ 152
- 7.6 Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the stock quantity price u_0 . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 1024$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages. . . 153
- 7.7 Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the initial volatility parameter σ_0 . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 4096$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages. . . 155

7.8	<p>Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the mean volatility parameter $\bar{\sigma}$. Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 1024$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.</p>	157
7.9	<p>Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the volatility of log-volatility parameter σ_v. Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\kappa = 2.5$, $\sigma_v = 1.25$, $\rho = -0.4$, $T = 3$ months, $K = 16$, $b = 1\%$, $N = 1024$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages. . .</p>	158
7.10	<p>Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the option's time to expiry T. Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 4096$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages. . .</p>	160
7.11	<p>Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the rate of log-volatility mean reversion parameter κ. Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 4096$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.</p>	162

- 7.12 Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the correlation parameter ρ . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 4096$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages. . . . 163
- 7.13 Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the risk aversion parameter γ . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\kappa = 2.5$, $\sigma_v = 1.25$, $\rho = -0.4$, $T = 3$ months, $K = 16$, $b = 1\%$, $N = 1024$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages. . . . 165
- 7.14 Average portfolio value at expiry time T as a function of its standard deviation, for various values of the risk aversion parameter γ . Top : call option for $K = 16$, $\sigma_0 = \bar{\sigma} = 16\%$, $\kappa = 2.5$, $\sigma_v = 1.25$, $\rho = -0.4$. Bottom : put option for $K = 64$, $\sigma_0 = \bar{\sigma} = 40\%$, $\kappa = 2.5$, $\sigma_v = 2.5$, $\rho = -0.8$. For both : $s_0 = 10$, $T = 3$ months, $b = 1\%$, $N = 1024$, $n = 100$, $n_R = 1000$. Error bars indicate one standard deviation of the sample averages. . . 167

LIST OF ABBREVIATIONS

AD	Average density
ADP	Approximate Dynamic Programming
BSM	Black-Scholes-Merton
CV	Control variate
DP	Dynamic Programming
expOU	Exponential Ornstein-Uhlenbeck
GBM	Geometric Brownian motion
HB	High-biased
LB	Low-biased
LR	Likelihood ratio
NH	No hedging
OU	Ornstein-Uhlenbeck
SAA	Sample average approximation
SDP	Stochastic Dynamic Programming
SM	Stochastic mesh
SV	Stochastic volatility
VRF	Variance reduction factor
WW	Whalley-Wilmott
Z	Zakamouline

NOTATION

$\mathbf{1}\{Z > z\}$	Indicator function
\mathcal{A}	Space of admissible trading policies
δ	Threshold for the Russian roulette technique
E	Expectation operator
E_k	Expectation conditional on time t_k information
\mathcal{F}_k	Sigma algebra generated by Y_k
γ	Risk aversion parameter
\mathcal{G}_k^N	Set of N simulated points at step k
μ_k	Decision function at step k
π	Hedging policy
$\psi(X_k, v, Y_{k+1})$	Transition function from state X_k to X_{k+1} , given a decision v and a new market state Y_{k+1}
\mathbb{R}	Set of real numbers
\mathcal{U}	Set of admissible trading decisions
\mathcal{X}	Set of possible states of the system (portfolio and market variables)
\mathcal{X}_k^N	Set of states at step k (not necessarily simulated)
\mathcal{Y}	Set of possible states of the market
a, b	Transaction costs parameters

t_k	Time at step k , expressed in years
c_k	Value of the portfolio cash account at step k
$f_k(Y_k, Y_{k+1})$	Conditional (transition) density of Y_k at step k
$g_k(Y_k)$	Unconditional density of Y_k at step k
h_k	Price of the derivative to hedge at step k
\widehat{J}_k	Approximation of the optimal risk function at step k
\widehat{J}_k^π	Monte Carlo estimate of the optimal risk function at step k based on the policy π
\widehat{J}_k^H	High biased estimator of J_k^*
\widehat{J}_k^i	Approximation of the optimal risk function at step k for a fixed market vector Y_k^i
\widehat{J}_k^L	Low biased estimator of J_k^*
J_k^*	Optimal risk function at step k
J_k^π	Risk function corresponding to the policy π
K	Number of hedging times
L	Loss function
$N(0, 1)$	Standard normal distribution with mean 0 and variance 1
$\widehat{Q}_k(X_k, v)$	Monte Carlo estimate of the function $Q_k(X_k, v)$
$Q_k(X_k, v)$	Expected risk at step k after a decision v

s_k	Stock price at step k
$U(0, 1)$	Uniform distribution over the unit interval $[0, 1]$
u_k	Stock quantity held at step k
v_k	Quantity of stock traded at step k
V_k^π	Value of the portfolio at step k under the policy π
$w_{k,i,j}$	Stochastic mesh weight between market vector Y_k^i and Y_{k+1}^j
X_k	System state vector at step k
Y_k	Market state vector at step k

À mes enfants, Laurie-Gabrielle, Aïlys et Yoram.

ACKNOWLEDGMENTS

Many people help us or influence us along a project as vast as a PhD thesis. Here I wish in particular to thank my adviser, Pierre L'Ecuyer, for his guidance and his sharp feedback. More generally, I wish to thank my parents, Jean-Guy, Claudette and Régent, for fostering my curiosity by exposing me to science at a young age and for their support throughout my life.

I would also like to acknowledge the support of the Institut de finance mathématique de Montréal (IFM2) through their doctoral scholarship program (2008-2010), as well as that of my employer, the Caisse de dépôt et placement du Québec, by allowing me to work part-time as I completed the thesis.

CHAPTER 1

INTRODUCTION

1.1 General overview

In finance, *hedging* refers to a trading activity that aims to reduce the fluctuations in the market value of some asset or portfolio. These fluctuations will stem from the evolution of factors such as stock prices, interest rates and volatility levels. To the layman, it may seem strange to want to hedge an investment; when investing money in a stock, for example, we hope for a (positive) return, so why cancel this out by hedging? But hedging can be a very useful tool for reducing risk that we do not want to take and is linked to an asset that is difficult or otherwise undesirable to trade.

An important practical application is to manage the risk of trading financial derivative instruments, such as stock options. In the case of stock options, the typical hedging decision rule, or hedging *policy*, involves trading the underlying stock in order to reduce the portfolio sensitivity to the stock price. This price sensitivity, or “delta”, can be seen as a *local* measure of risk and the associated policy, “delta hedging”, requires frequent trading (typically daily) to keep the delta close to zero. However, the presence of transaction costs, such as broker fees or bid-ask spreads in prices, could make the cumulative cost of delta hedging quite large. So it is natural to compare the performance of hedging policies based on a *global* measure of risk, i.e., which involves the total performance of the portfolio over a given time horizon. Finding a policy which reduces such a risk measure is what we refer to generally as *hedging policies*, and is the focus of this thesis.

In the case of hedging a stock option in a continuous time setting where the underlying price follows a basic one-dimensional geometric Brownian motion (GBM) process (see the example in section 1.2), policies which minimize a global risk measure

including costs have already been found. They can be computed explicitly as numerical solutions to a partial differential equation (PDE), as in the works of Hodges and Neuberger [71] and Clewlow and Hodges [40]. This is computationally intensive, but there exists simple approximations in the case of low transaction costs, as well as other heuristics, which will be discussed later in section 4.2.

It is also well-known that models using more general processes than the GBM process, such as stochastic volatility (SV) processes, can better reproduce various features of the option market, such volatility smirks and term structures (see for example Christoffersen, Heston and Jacobs [39]). Thus it is of practical interest to find methods to compute optimal policies using SV models including costs, in order to determine whether or not currently known heuristic policies could significantly be improved, depending on the context. But as more flexible market models (such as SV processes) are considered, the PDE approach becomes harder to apply, due to the increased dimensionality of the market state (stock price and volatility).

In order to try to bridge the gap between SV models and the practical computation of optimal hedging with costs, we consider a particular version of the global hedging problem in discrete time, including transaction costs, and measuring risk as the expectation of a loss function on the terminal portfolio value. We aim to

- provide a fairly general computational method to closely approximate the optimal solution to this class of problems,
- assess the magnitude of the resulting approximation error,
- compute solutions in an efficient manner, so that it could be used in practice.

As we will show, these three objectives can be attained by extending the use of the *Stochastic Mesh* method, originally introduced by Broadie and Glasserman [35], [36] for American option pricing. It involves generating market states by Monte Carlo simulation, and computing relevant conditional expectations via weighted averages

over these states. We extend its application to hedging by computing at each state of the mesh a one dimensional approximation of the optimal risk function, instead of a single number (i.e., the option value), as usually done in the case of American option pricing. Because of the various possible implementations of this general idea, we will usually refer to *stochastic mesh methods* in the plural.

Focusing on the case of the negative exponential loss function, the specific approximation used in experiments is motivated by the analysis of some general properties of the solutions. Furthermore, we provide new techniques to improve the efficiency of using stochastic mesh methods, so that the resulting solutions can be computed fast enough to be useful in practice. Important additional sources of inspiration for these techniques are the works by Rust [117] and Boyle, Kolkiewicz and Tan [30].

As will be seen in section 2.1, the problem of finding an optimal hedging policy can be framed as a discrete time stochastic dynamic program (SDP). So in theory, assuming the specific details of the problem are “nice” enough, a straightforward solution is known through the Dynamic Programming (DP) Algorithm. In practice, this solution can be calculated provided the value function or the policies can be well-approximated by some set of functions, but good approximating functions may not be so easy to find. This may depend on the details of the specific instance considered, such as the risk measure used or the specification of transaction costs. The challenge here is to make the method work using a reasonable amount of computational resources. Note that current theoretical developments use a more technical formulation of the hedging problem involving concepts such as convex duality and equivalent martingale measures, but we will not discuss these further here (see for example Cvitanić and Karatzas [43], Kramkov and Schachermayer [81], [82], and for a possible Monte Carlo implementation, Grasselli and Hurd [64]).

As we will explain in more detail in section 1.4, the hedging problem has already been addressed under many conditions. Solutions in continuous time related to our

setup include that of Hodges and Neuberger [71], Clewlow and Hodges [40], as well as the approximate solutions by Whalley and Wilmott [136] and Zakamouline [138]. But the use of stochastic mesh methods gives us flexibility in the choice of the (possibly multi-dimensional) stock price dynamics, while staying computationally tractable. Furthermore, it comes with a built-in low biased estimate of the optimal risk, which is typically not known for a heuristic policy. This can be compared to the high biased estimate provided by a Monte Carlo sample average of the loss function, to obtain a measure of the accuracy of the solution.

To summarize, there is a need for hedging methodologies that take costs into account and that can be applied to large classes of derivative payoffs and stock price dynamics, which is what motivates our use of stochastic mesh methods for computing solutions. For definiteness, we will restrict our numerical experiments to the so-called exponential Ornstein-Uhlenbeck model (or expOU, for short), which is described in some detail in section 4.1. Modeling assumptions will be explained in more detail in the next sections. The general methods we study here do not depend on a specific price model, but our examples do make use of the existence of a non-singular transition density between each pair of consecutive states of the stochastic mesh, which allows simple definitions for weights between these states.

1.2 A basic example of dynamic hedging with costs

We start by illustrating some important concepts and questions related to dynamic hedging with costs by looking at a simple concrete example, that of delta hedging a standard option, which was mentioned in section 1.1.

Consider a standard European call option, which is a contract that gives its owner the right to buy the stock at a predetermined price X and time T (the option's expiry

date). Its payoff is given by

$$h(s(T)) = \max(s(T) - X, 0), \quad (1.1)$$

where $s(T)$ is the stock price at time T . The term “European” means that the option can *only* be exercised at expiry. This is in contrast to an “American” option, which can be exercised at any time during its life, i.e., for $t \in [0, T]$.

Assume that you are a trader for a bank which bought this option from a client for a price (or premium) α_0 . There could be many reasons the client wanted to sell this option, such as implementing a so called “covered call” strategy, for example. At the time of buying this option, you could not locate another counterparty on the market willing to take the opposite side of the transaction (i.e., buying from you the same option) for a premium $\alpha_1 > \alpha_0$, which would have allowed you to remove all of the risk associated to the option and secure a profit equal to $\alpha_1 - \alpha_0$. So now you wish to hedge this option until some future time, perhaps until its expiry.

For definiteness, let $\{t_k : k = 0, 1, \dots, K\}$ be the trading dates, expressed in years, and let $\{s_k : k = 0, 1, \dots, K\}$ denote the stock prices at those dates. The exposure at time t_k of the option to the stock price s_k is characterized to first order by the *delta*, or first derivative of the option value $h_k(s_k)$ with respect to s_k , which we denote as

$$\Delta_k(s_k) = \frac{\partial h_k}{\partial s_k}(s_k). \quad (1.2)$$

The specific formula depends on the pricing model used, so assume that we are working under the Black-Scholes [22] and Merton [104] framework (hereafter, BSM). Hence, the stock price dynamics is represented by a geometric Brownian motion process with a drift equal to the risk-free interest rate r , so that the log returns are

given by

$$\ln(s_{k+1}/s_k) = r(t_{k+1} - t_k) + \sigma\sqrt{t_{k+1} - t_k}Z_k, \quad Z_k \sim N(0, 1), \quad (1.3)$$

where σ is the volatility parameter and each Z_k are independent. For simplicity, we assume there are no dividends. Some sample paths for this process are represented in figure 1.1. The top part of figure 1.2 illustrates the derivative price $h_k(s_k)$ as a function of s_k , for an option with strike price $X = 10$, maturity $T = 3$ months, interest rate and dividend yield both equal to zero ($r = q = 0$), and volatility parameters $\sigma = 16\%$. Under the BSM model, the delta is given by

$$\Delta_k(s_k) = \Phi(d_1(s_k)), \quad (1.4)$$

$$d_1(s_k) = \frac{\ln(s_k/X) - (r - \sigma^2/2)(T - t_k)}{\sigma\sqrt{T - t_k}}, \quad (1.5)$$

where $\Phi(\cdot)$ is the cumulative density function of the normal distribution with mean 0 and variance 1. The bottom part of figure 1.2 illustrates the delta as a function of the underlying price s_k , based on equations (1.4) and (1.5).

Policies will be defined more formally in section 2.1.1, but for now they should be understood as decision rules that take market information (such as the stock price) as input and returns the quantity of stock to hold in the portfolio. If we follow a policy that requires holding a quantity of stock $u_k = -\Delta_k(s_k)$ at each step k , then the total value of the portfolio (including a cash amount c_k) is given by

$$V_k(c_k, u_k, s_k) = c_k + u_k s_k + h_k(s_k) = c_k - \Delta_k(s_k) s_k + h_k(s_k) \quad (1.6)$$

and has an overall delta $\partial V_k / \partial s_k$ equal to zero at each step k . This policy is called (dynamic) *delta hedging*. Allowing for continuous time trading and some suitable conditions, dynamic delta hedging in the BSM framework will keep the portfolio

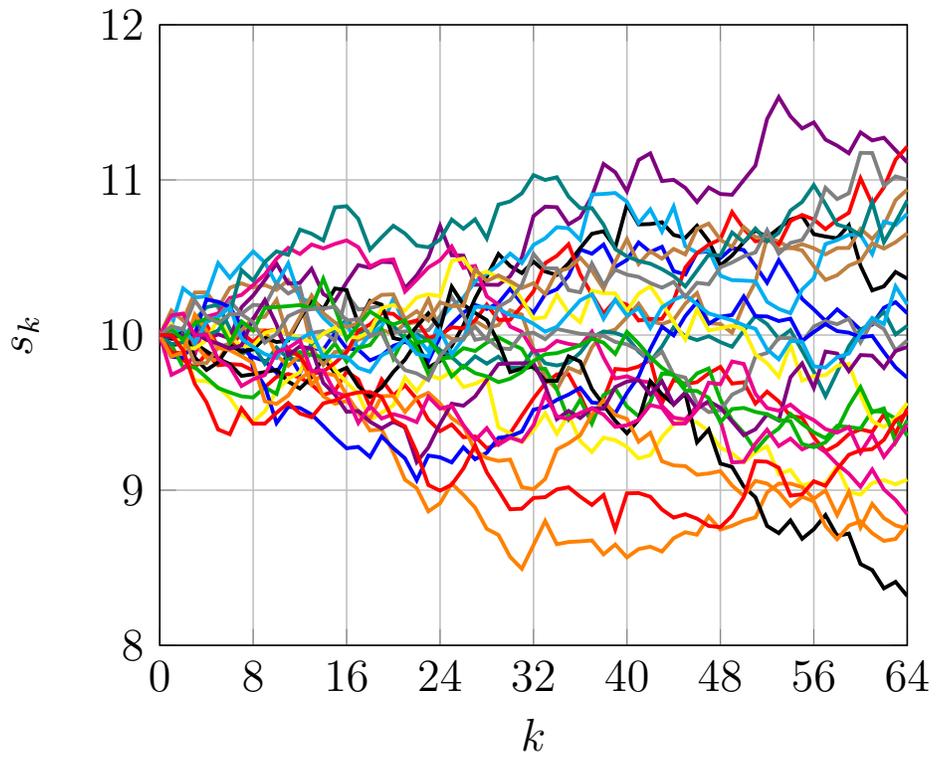


Figure 1.1: Sample paths from the geometric Brownian motion process, for $K = 64$ steps, $T = 3$ months, initial price $s_0 = 10$, drift $\mu = 0$ and volatility $\sigma = 16\%$.

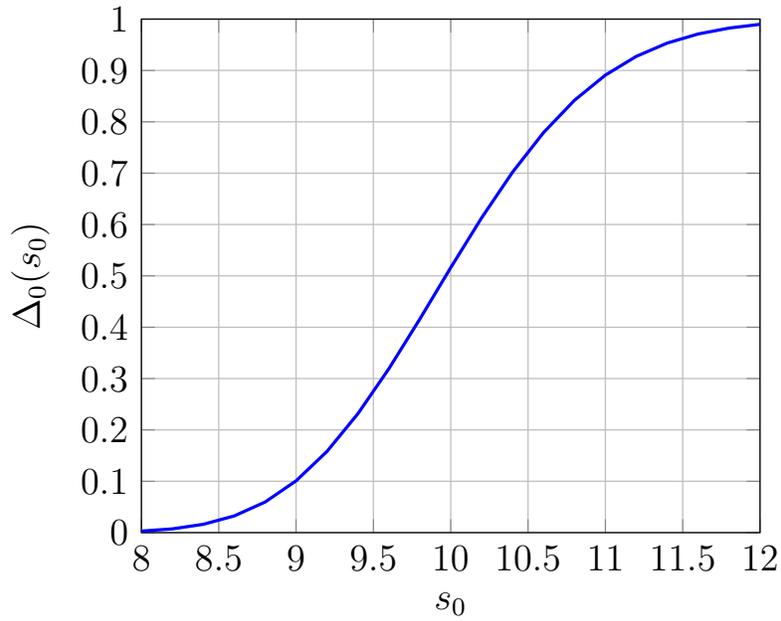
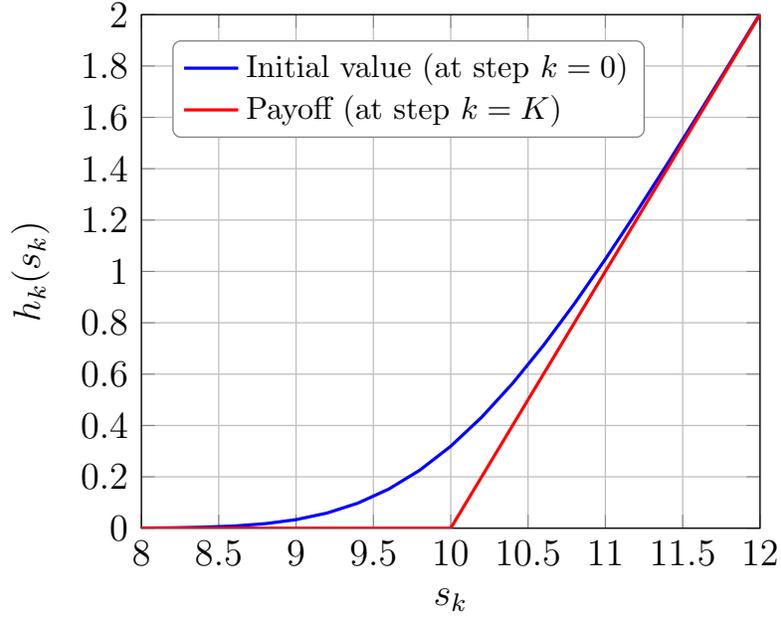


Figure 1.2: Top : Initial value and payoff of a standard call option expiring in 3 months with strike $X = 10$, under the BSM model with $r = q = 0$ and $\sigma = 16\%$. Bottom : Delta of the option (same parameters).

value constant at all times, thus removing *all* the risk (see again [22] and [104]). But in practice, it is not possible to trade continuously and, in any case, cumulative transaction costs (if non-zero) will tend to infinity as the time steps $\Delta t = t_{k+1} - t_k$ tend to 0, as pointed out first by Leland [94].

The effect of costs is visible in figure 1.3. It shows sample paths for both the policy decisions u_k and the portfolio value V_k^π for two different policies π . The first policy is delta hedging under the BSM model, which we described above. This policy reduces the variance of the final portfolio value (as viewed at the terminal step $k = K$), by trading at each step to bring the portfolio delta back to zero. But this leads to a noticeable negative drift in the portfolio value, as can be seen in the top left chart, due to transaction costs. The second policy is the hedging policy from Zakamouline [138]. As described in section 4.2.2, it is a close approximation to a policy which minimizes a particular measure of risk on the terminal portfolio value. This measure of risk, negative exponential risk, will be seen again in section 2.1.3 and will be used throughout this thesis. We denote the Zakamouline policy here by $Z(1)$, where the number 1 refers to the value used for a risk aversion parameter γ (i.e., here $\gamma = 1$). This policy makes less frequent hedging adjustments than the BSM policy (compare the two charts at the bottom of figure 1.3), depending on the value of various parameters such as risk aversion, volatility and transaction costs. Hedging less reduces the magnitude of the negative drift coming from transaction costs, but can also lead to a higher variance of the final portfolio value, as can be seen in the top right chart of figure 1.3. Typically, we cannot reduce both the transaction costs and portfolio value variance, so the optimal tradeoff we are aiming for is defined by the risk measure we choose.

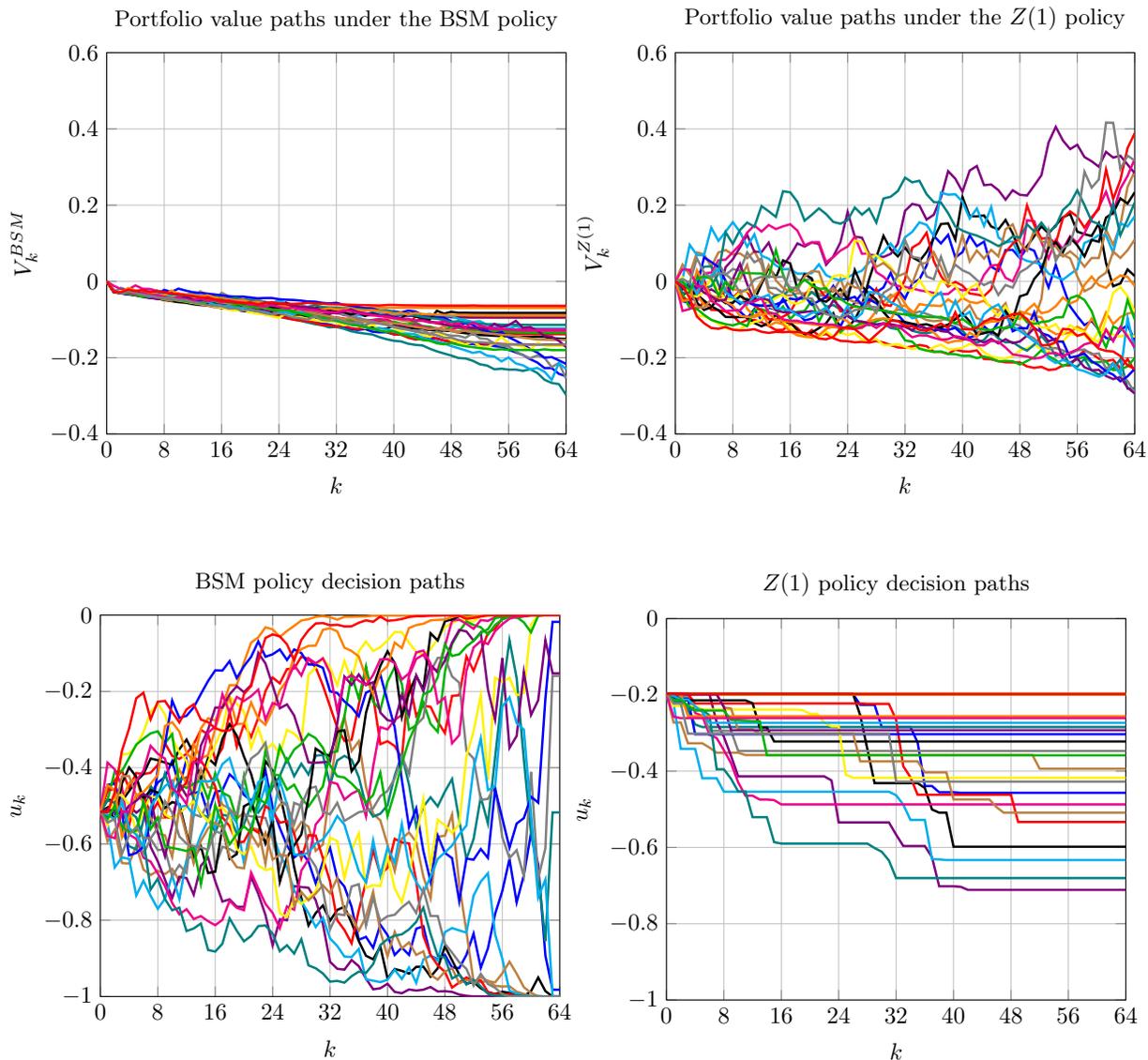


Figure 1.3: Sample paths of the portfolio value $V_k^\pi(X_0)$ and of the policy decisions u_k . Left-hand side : $\pi = \text{BSM}$, right-hand side : $\pi = \text{Z}$ (for $\gamma = 1$). Setup : $s_0 = X = 10$, $\sigma = 16\%$, $T = 3$ months, $K = 64$, $b = 1\%$.

1.3 Dynamic Programming on a Stochastic Mesh : main ideas

As stated before, the problem we are trying to solve is to minimize a global risk measure. This risk measure is defined at the last step $k = K$ of the discrete time horizon we are considering. Such a problem, where we are looking for an optimal decision policy over many steps, falls within the scope of the well-established field of discrete time DP. The general solution method is given the *DP Algorithm*, which proceeds by backward induction : one starts with a risk function $J_k(X_k)$, assumed known for any given state X_k at the last step $k = K$ (at time t_K), and uses this information to calculate the optimal risk at the previous step $k = K - 1$. This still requires optimization, but it is simpler to tackle than the full problem, since it involves only one step at a time. The process is then repeated until we reach step $k = 0$. Note that in DP, the functions $J_k(X_k)$ are usually referred to as *cost functions*, but we use the term *risk function* to avoid any confusion with transaction costs.

Here the number of possible states X_k of the system at each step k is in fact infinite. Each state depends on a vector Y_k of market information (parameters) such as the stock price and volatility, which are both defined over a continuum of possible values. They also depend on the stock quantity u_k held in the portfolio as a hedge, as well as the amount of cash c_k (although this last variable may be ignored in some formulations, depending on the specific objective function used).

A straightforward approximation method is to focus on a finite subset of states $\{X_k^i : k = 0, \dots, K, i = 1, \dots, N\}$, and compute a solution based on those states only. In our setting, the market information vector Y_k is a stochastic process, so it makes sense to simulate Y_k to help generate states X_k . Indeed, the DP Algorithm will then be applied to states that are more likely to be relevant. This idea was already used in the Stochastic Mesh method mentioned in section 1.1 for the problem of pricing American options.

In the case of the hedging problem, there is at least one extra dimension involved

in the state space, due to the need to keep track of the stock quantity u_k (the cash account c_k also may be required as a state variable, depending on the risk measure used). This is where our implementation differs from the original setup underlying the Stochastic Mesh method as applied to American options, since for American option pricing the states can then be fully generated independently from the (early-exercise) decisions. A straightforward extension could be to define a finite set $\{u^j : j = 1, \dots, M\} \subset \mathbb{R}$ of stock quantities to consider, perhaps forming a regular grid. But as we will discuss later in chapter 2, specific details of the hedging problem can lead to some simplifications, where only a few points are required to define a good estimate over all possible values of u_k .

The two main motivations for using the Stochastic Mesh method is that 1) it does not suffer from typical curses of dimensionality when the dimension of the market vector state is increased, i.e. the exponential increase in the number of states (see section 3.1.1) and 2) it is applicable for any (sufficiently integrable) payoff, including path-dependent and discontinuous payoffs. This is in contrast with PDE methods, which require deterministic grids which grow exponentially in size as a function of the dimension, and may further require some regularity conditions on the payoff. Another useful benefit is that the method yields a confidence interval for the optimal risk, through one low biased and one high biased estimate. In our case, in contrast to the basic Stochastic Mesh algorithm, we need to also take into account the approximation error coming from keeping track of the effect of the stock quantity on the optimal risk function.

But the use of a stochastic mesh in its basic form still comes with a significant computational cost, both in terms of time and memory, because the number of weights to consider grows like $O(KN^2)$. For example, using $N = 2^{11} = 2048$ mesh points over $K = 8$ time steps leads to over 32 million weights. So the use of efficiency improvement techniques becomes necessary for practical implementations. This will

be addressed further in chapter 6.

1.4 Related works

This section surveys some of the existing literature on hedging problems with various levels of complexity, to help put in context the hedging problem from this thesis. We start by highlighting a few important theoretical works, which should by no means be considered an exhaustive survey. We then provide some references to works which specifically involve Monte Carlo simulation.

1.4.1 Some important works on the hedging problem with costs

For the case where transaction costs are present, the first results were published by Leland [94], who proposed to modify the implied volatility parameter in the Black-Scholes hedging policy to account for the effect of costs on the buy and sell price for stocks. However, this did not involve explicitly minimizing a risk measure. Ideally, one would like a policy that guarantees the terminal liquidation value of the hedged portfolio, so that it effectively becomes riskless (which is the case of the BSM model without costs [22]). In discrete time, Boyle and Vorst [29] showed how this can be done using a binomial lattice model, but this is a restrictive price model and furthermore it does not generalize well in multiple dimensions (see also Bensaïd, Lesne, Pagès and Scheinkman [14]). And in continuous time, Soner, Shreve and Cvitanić [127] proved a negative result, which is that if one seeks to *super-replicate* a standard call option, that is, have a policy such that the terminal payoff is never strictly negative, then the cheapest policy is to buy one share of the underlying stock and hold it to maturity. Thus any policy that replicates perfectly a standard call option will cost at least as much as buying the actual stock, which is uninteresting in practice.

This then leads naturally to looking for policies which are optimal in some sense,

such as trying to maximize a given utility function (equivalently, minimize a risk function). For the so-called negative exponential risk function and proportional transaction costs (the setup which we consider in chapter 2), the optimal policy was first analyzed by Hodges and Neuberger [71]. They derive a differential equation for the optimal solution, which can be shown to be characterized by a *no-transaction region* where it is optimal not to trade when the stock quantity u_k is within the region. When u_k is outside the region, the optimal policy is to trade just enough to go back on the boundary. However, the resulting partial differential equation for the boundary is required to be solved numerically. Subsequent works by Edirisinghe, Naik and Uppal [53], Davis, Panas and Zariphopoulou [47] and Clewlow and Hodges [40] improved on numerical aspects, but computational complexity still remains an issue for this approach. As will be described in section 4.2, closed analytical formulas that approximate the optimal policy for this case are given by Whalley and Wilmott [135] and Zakamouline [138]. Although the original analysis was for the case of exponential utility, Andersen and Damgaard [4] showed that Hodges and Neuberger's result is somewhat independent of the specific utility function used, as was conjectured by Davis, Panas and Zariphopoulou [47].

Another objective function which has been studied extensively is quadratic risk. For the case without costs, see for example Schweizer [122], Schweizer [121], Grünewald and Trautmann [65], Černý [131], Bobrovnytska and Schweizer [24], Heath, Platen and Schweizer [69], Černý and Kallsen [132] and the references therein. Although not necessary an optimal solution to the global quadratic hedging problem, Schweizer [120] introduced a convenient local hedging approach (local risk minimization) for the case without costs. This was then extended by Lamberton, Pham and Schweizer [83] to the case with costs.

1.4.2 Monte Carlo based algorithms for derivative hedging

It is sometimes possible to calculate the optimal hedge by Monte Carlo simulation without bias, assuming there are no transaction costs and that the markets are *complete* (i.e. that derivative payoffs can be replicated by only trading the underlying asset). See for example the two different approaches by Detemple, Garcia and Rindisbacher [50], and Cvitanić, Goukasian and Zapatero [44], [45]. See also Boyle, Imai and Tan [28] for the application of quasi-Monte Carlo methods to the latter approach. To address the case of *incomplete* markets without costs, Grasselli and Hurd [64] provide a Monte Carlo based solution method for pricing and hedging an option under the exponential risk measure. It uses a dual formulation of the hedging problem and solves it by estimating hedging policies via linear regression, similarly to the Longstaff and Schwarz method for American options [98]. In numerical experiments, the results for pricing appear more stable than for the hedging policies.

When costs are included, there are a variety of methods proposed in the literature, but they can be difficult to compare due to the differences in setups and the various approximations introduced. Gondzio, Kouwenberg and Vorst [63] provide methods to compute optimal policies when the volatility is stochastic. However, these results are only applicable to a restricted number of time steps (up to 3 or 4), as the complexity of their algorithm grows exponentially in the number of steps. For the same reasons, their algorithm does not generalize well for higher dimensional state spaces. Kennedy, Forsyth and Vetzal [78] study optimal hedging for one dimensional jump-diffusion, combining delta-hedging with the selection of a set of options that minimizes the jump risk in stock prices using a quadratic risk measure. Boyle, Coleman and Li [27] investigate the problem of minimizing (centered) quadratic risk for a portfolio of derivatives by trading other more liquid derivatives (which could include stocks). They show through experiments that taking costs into account and adding trading restrictions helps in obtaining more stable solutions, but the hedging

strategies they consider are static, i.e., they do not take into account possible future trading decisions. Keppo and Peura [79] start with a general problem similar to equation (2.3), and replace it by a simplified problem involving a non-linear program to be solved, with coefficients depending on the first two moments of the price dynamics. This approach is interesting as it allows to tackle problems with many assets, but it is not clear how close the resulting approximation is to the original problem or how well its solution would behave under various conditions (for example, as the number of periods or number assets grows, or for complex price dynamics and risk measures).

1.5 Plan of the thesis and main contributions

We conclude this introduction by giving a quick outline of the chapters of this thesis, each of which considers a different aspect of the hedging problem and stochastic mesh methods. We also indicate specific contributions associated with each chapter.

In chapter 2, we recall some basic theory for discrete time DP problems following Bertsekas and Shreve [18], as applicable to the hedging problem. We then analyze the structure of solutions in terms of transaction boundaries for the case of a general convex loss function and for the particular case of the negative exponential loss function.

Main contribution 1 : Simple theoretical framework for hedging in discrete time when costs are present, applicable to a large class of market models and options payoffs, including a characterization of the no-transaction region and of the optimal risk function.

In chapter 3, we show how an approximation to the optimal risk function can be implemented using stochastic mesh methods. A central point is that the mesh allows to compute a confidence interval for the optimal risk function value. Using this, we provide some first computational results which illustrate the behavior of the mesh

based solutions and compare them to other heuristics for various setups.

Main contribution 2 : Applying stochastic mesh methods to the optimal hedging problem, where the optimization step is based on a one-dimensional function of the stock quantity.

Secondary contribution 1 : A recursive error bound formula for the approximation error at the various steps of the Dynamic Program and its use with the stochastic mesh estimator of the risk to obtain a low biased estimator of the optimal risk.

Secondary contribution 2 : Linear-quadratic approximation for the function $\ln J_k^*$ in the case of the negative exponential loss function.

In chapter 4, we start by giving details on the exponential Ornstein-Uhlenbeck (expOU) model used throughout our experiments. We also provide an extension to the model to test the possible impact of including additional information on optimal hedging decisions. We then describe some important heuristics and use them to relate our work to the existing literature on hedging, as well as to serve as benchmarks against which to compare empirical results for stochastic mesh based policies.

Secondary contribution 3 : Estimates of the expOU model parameters for various market environments (to provide realistic parameter ranges for experiments).

Secondary contribution 4 : Introduction of an extension of the expOU model including additional information variables, and its use in the context of stochastic mesh hedging.

In chapter 5, we present the *MeshHedging* Java library which was developed in the context of this thesis to implement the computation of hedging policies using stochastic mesh methods. We also provide details about the algorithm used to compare the risk of various hedging policies throughout our experiments. Some results are then presented to illustrate the performance of the basic mesh construction.

Main contribution 3 : The *MeshHedging* Java library and its application to

computing hedging policies.

In chapter 6, we will propose efficiency improvements for computing hedging policies using a stochastic mesh, so that the method be even more useful in practice. If the derivative price $h_k(Y_k)$ must be estimated along the mesh nodes, then using control variates to reduce the variance of these estimates $\widehat{h}_k(Y_k)$ is of paramount importance. Computation time is also an important element to account for, for example when computing the weights used in the weighted averages that approximate the cost functions at every state. This can take a long time because weights must be defined over all consecutive pairs of market states (Y_k^i, Y_{k+1}^j) , which means a total number of weights in the order of $O(KN^2)$. To improve on this aspect, we try using a single grid of states for all the stages of the DP Algorithm, instead of a full mesh, to reduce by a factor of K the number of weights that need to be computed. This is partially related to methods introduced by Rust [117] and Boyle, Kolkiewicz and Tan [30]. Additionally, pairs of states (Y_k^i, Y_{k+1}^j) that have very low weights are removed using a Russian roulette technique, to reduce the computation time while not introducing bias in the stochastic mesh based conditional expectations.

Main contribution 4 : Benefits of combining the single grid and Russian roulette techniques, which is new in the context of stochastic mesh methods.

Secondary contribution 5 : New applications of control variate techniques for the stochastic mesh : efficient method for reducing the variance of the in-mesh derivative value estimates, and some preliminary results involving biased controls (with expectation unknown, but close to zero).

Finally, chapter 7 applies the stochastic mesh hedging methodology developed in this thesis to investigate the properties of the optimal solution for the problem of hedging either a long call or long put option position when the market dynamics follows an expOU process. In particular, the behavior of the optimal risk function and policies is examined as a function of the various model parameters, and compared to

the associated results for various heuristic hedging policies. This allows to highlight in which setups do the heuristics and the mesh-based policy do best and in which setups they struggle.

Main contribution 5 : Detailed empirical results for the optimal policy under the expOU model with costs under various settings, including an in-depth comparison of various hedging heuristics and their relative performance in various environments.

CHAPTER 2

STRUCTURE OF SOLUTIONS

This chapter makes the hedging problem more precise, starting with a formal definition in section 2.1. We review in section 2.2 some fundamental results from discrete time Dynamic Programming, considering more specifically the problem of optimal dynamic hedging. Section 2.3 then provides some useful results on the structure of solutions to the class of problems of section 2.1. We then show in section 2.4 that for the particular case of the exponential loss function, the optimal policy and the optimal risk function have a relatively simple piecewise characterization. This will be used in chapter 3 to motivate a particular risk function approximation. Finally, in section 2.5, we cover some technical points regarding the measurability of the risk function and the validity of the DP algorithm in our context.

2.1 Problem definition

We now describe the hedging problem more formally as a discrete time dynamic program. This will allow us to establish some notation for the rest of the thesis.

2.1.1 Formulation as a Dynamic Program

We work in a discrete time setting, where trading decisions can only be made at the times $t_0 < t_1 < \dots < t_K < \infty$, expressed in years. In particular, we set the initial time to $t_0 = 0$ and the horizon of the hedge is $t_K = T$ years, where $T \in [0, \infty)$. The state of the system at step k , corresponding to time t_k , can be completely described by an element X_k from the set $\mathcal{X} = \{X_k = (c, u, Y) : k = 0, \dots, K, c \in \mathbb{R}, u \in [u_{\min}, u_{\max}] \subset \mathbb{R}, Y \in [0, \infty)^d\}$, where c is the value of the cash account, u is the amount of shares of stock (possibly fractional) held in the

portfolio and $Y = (Y_1, \dots, Y_d)$ is the market state vector, i.e., a vector of variables governing the dynamics of the stock (such as price and volatility). Note that negative quantities are allowed for the cash and stock positions, which represent respectively an amount of money borrowed and a short position in the stock (i.e., the stock has been borrowed to be sold, so that the exposure to stock prices is negative). The interval $\mathcal{U} := [u_{\min}, u_{\max}] \subset \mathbb{R}$ constrains the allowable stock quantities. But this does not restrict the applicability of the formulation since, for example, one could not buy or sell more stocks than have been issued on the market at a given time.

Uncertainty is represented over a probability space (Ω, \mathcal{F}, P) and the market dynamics is described by a discrete time Markov process $\{Y_k \in [0, \infty)^d : k = 0, \dots, K\}$, adapted to a filtration $\{\mathcal{F}_k : k = 0, \dots, K\}$, $\mathcal{F}_0 \subseteq \dots \subseteq \mathcal{F}_K = \mathcal{F}$. This filtration represents the information available about the market at each time t_k . All expectations $\mathbb{E}[\cdot]$ in the following are understood to be taken with respect to the measure P and we will denote expectations conditional on \mathcal{F}_k (the time t_k information) as $\mathbb{E}_k[\cdot]$.

The portfolio contains the derivative to hedge, as well as cash and some given amount of the stock. Its value at time t_k is given by

$$V_k(X_k) = c_k + u_k s_k + h_k(Y_k), \quad (2.1)$$

where $X_k = (c_k, u_k, Y_k) \in \mathcal{X}$, and $h_k(Y_k)$ is the time t_k value of the derivative to hedge, as a function of the market information vector Y_k , i.e., $h_k : [0, \infty)^d \rightarrow \mathbb{R}$. There may in fact be more than one derivative to be hedged in the portfolio, in which case we can always take $h_k(Y_k)$ as their total value.

We define a trading policy π to be a K -tuple of functions $\pi = (\mu_1, \mu_2, \dots, \mu_K)$ where each decision $u_{k+1} = \mu_{k+1}(X_k)$ represents the stock quantity to be held in the portfolio during the period $(t_k, t_{k+1}]$, depending on the state $X_k \in \mathcal{X}$ of the system at instant k . That is, $\mu_{k+1} : \mathcal{X} \rightarrow \mathcal{U} \subseteq \mathbb{R}$ for $k = 0, \dots, K-1$, where \mathcal{U} is the space of admissible decisions (i.e., the decision space). Note that the decision

u_{k+1} is then a part of the next state $X_{k+1} = (c_{k+1}, u_{k+1}, Y_{k+1})$, as it is the stock quantity held at time t_{k+1} . We denote the set of all admissible trading policies by $\mathcal{A} = \{\pi = (\mu_1, \dots, \mu_K)$, where $\mu_k : \mathcal{X} \rightarrow \mathcal{U}$, $\forall k = 1, \dots, K\}$.

Finally, to succinctly describe the evolution of the states X_k , we define a transition function ψ at each step k

$$X_{k+1} = \psi(X_k, v, Y_{k+1}). \quad (2.2)$$

Here, $X_{k+1} = (c_{k+1}, u_{k+1}, Y_{k+1})$, where c_{k+1} is given by the financing equation (2.4), v is the stock quantity to be held at the next period (so that $u_{k+1} = v$), and $Y_{k+1} = Y_{k+1}(Y_k, \omega)$, where $\omega \in \mathcal{F}_{k+1}$. The function ψ does not depend on any particular policy π , nor does it depend on time either, which is why we write ψ and not ψ_k .

Given a trading policy π and a state X_k at some step k and the transition equation (2.2), the portfolio values $\{V_j^\pi(X_k) : j \geq k\}$ at steps $j \geq k$ form a stochastic process, i.e. a set of random variables indexed by time (see Billingsley [19]). The superscript π emphasizes that the value of the portfolio is a random variable that depends on the policy π , and the state X_k is a known state from which the policy is applied. For ease of notation, we will always write V_j^π (without specifying the starting state) to mean $V_j^\pi(X_0)$, unless otherwise specified.

We are now ready to define the hedging problem formally : We are interested in trading policies that aim to reduce some measure of risk, i.e. hedging policies. Risk is represented here by the expectation of a loss function $L : \mathbb{R} \rightarrow [0, \infty)$ applied to the value V_K^π of the portfolio at time t_K , the last observation time. An optimal hedging policy is then a trading policy π that minimizes the risk, i.e., a policy π that solves

$$\inf_{\pi \in \mathcal{A}} \mathbf{E}_0[L(V_K^\pi - V_0)]. \quad (2.3)$$

To conclude the section, we note that dynamic programming is not the only possibility to formulate discrete time hedging problems. For example, Gondzio, Kouwen-

berg and Vorst [63] apply stochastic programming to solve a hedging problem under stochastic volatility and costs. Their approach involves using aggregated states and only a few time steps, in order to reduce the computational cost of the method. Other approaches such as multistage stochastic programming could also be considered (see for example the book by Birge and Louveaux [21] or the the survey by Yu, Ji and Wang [137]), but we do not study these further here.

2.1.2 State transition function

Before we go on, let us take a look at the details of the transition function ψ from equation (2.2). The amount of cash c_{k+1} held in the portfolio at step $k + 1$ is completely defined by the policy π and the stock price s_k through the financing equation

$$c_{k+1} = (c_k - \Delta u_{k+1} s_k - \phi(s_k) |\Delta u_{k+1}|) (1 + r \Delta t_{k+1}), \quad (2.4)$$

where $\Delta u_{k+1} = u_{k+1} - u_k$ is the amount of stock traded at time t_k , $\phi(s_k) := a + b s_k$ is the transaction cost per unit stock traded (depending on the constant parameters $a, b \geq 0$) and r is a constant interest rate. For simplicity, we omit possible dividend payments from our discussion, but they could of course be included in equation (2.4). Throughout the remainder of the thesis, we will assume that the numéraire (i.e., the reference asset used to quantify the value of other assets) is taken to be a unit of the cash account, instead of a unit of a currency (see for example Delbaen and Schachermayer [49], section 2.1). Thus without loss of generality, we can effectively remove the interest rate variable r (i.e., set $r = 0$), which helps clarify the notation.

The transaction risk function $\phi(s_k)$ described above is an idealization, but nevertheless covers typical trading costs. Indeed, the constants a and b in equation (2.4) could respectively represent a brokerage cost per quantity of stock traded and a bid-offer spread on the value of the stock, for example. At the time of writing, US stock transactions for institutional investors could cost about $a = 0.005\%$ per

share whereas European stock transactions could cost about $b = 0.1\%$ of the traded amount (source: <http://www.interactivebrokers.com/en/p.php?f=commission>, as of august 2012). Bid-offer spreads vary from stock to stock, but the associated cost would be about $a = 0.01\%$ for large capitalization US stocks such as IBM. However, it may not be possible to execute a large transaction (say 20000 shares) at the quoted price, since that price may only be available for a small amount of stocks (say 500 shares). In that case, the market impact of the transaction could impact the stock price in a way that amounts to an indirect cost, which is not explicitly modeled here. Note also that brokerage costs could be much higher for ordinary investors and bid-offer spreads could be wider for less liquidly traded stocks. In all such cases with extra costs, a simple approximation could be to use higher values for a and b .

In principle, unlimited losses are possible for the portfolio if, for example, the stock quantity u_k held in the portfolio is negative and the stock price s_k is not bounded above. This can be a problem because the expected loss function may no longer be well defined, depending on the specific model used for the price process. In our setup, however, we assume that the stock price is bounded above by some value s_{\max} (and also below by 0). In numerical experiments, this can be guaranteed by using a truncated distribution for the stock price dynamics, with a bounded right tail for the stock price increments $s_{k+1} - s_k$ ($k = 0, \dots, K - 1$). That is, one can simulate the price increments $s_{k+1} - s_k = e^Z$ as lognormal variates based on a truncated normal distribution for Z (see Law and Kelton [84], for example).

Note that different formulations could be used to limit losses more directly. For example, putting a lower limit on the total portfolio value V_k^π itself, so that it remains above some value $V_{\min} \geq 0, \forall k$, or putting a lower limit on the portfolio variations $V_{k+1}^\pi - V_k^\pi$. However, such constraints make it harder to analyze the resulting risk function. In particular, even if the loss function L is convex, the resulting risk function may no longer be convex. We will not consider these constraints further in

this thesis.

2.1.3 Risk measures

In a market where trading can occur only in discrete time and/or where some risks cannot be fully hedged (such as in stochastic volatility models), pricing and hedging depend on the preferences of investors (see for example Karatzas and Shreve [77]). This can be done by specifying a risk measure to minimize, as in section 2.1.1 where we consider the expectation of a loss function L applied to the portfolio performance $V_K^\pi - V_0$.

Throughout this thesis, we will focus on *negative exponential risk*. It is defined by the loss function

$$L(V_K^\pi - V_0) = \frac{1}{\gamma} (e^{-\gamma(V_K^\pi - V_0)} - 1), \quad (2.5)$$

where $\gamma > 0$ is a parameter characterizing risk aversion. Minimizing the function $L(x)$ from equation (2.5) corresponds to maximizing $U(x) := -L(x)$, which is known as the *exponential utility function*.

In order to get a better intuition for the associated objective function for the hedging problem defined by equation (2.3), we would like to consider the first few terms of a series expansion for $\mathbb{E}_0[L(V_K^\pi - V_0)]$. To this end, we will need to use a stochastic version of Taylor's theorem.

Theorem 2.1.1 (*Stochastic Taylor's Theorem*) *Let $L : [a, b] \rightarrow \mathbb{R}$ be a function with continuous n^{th} -order derivative on $[a, b]$. Let r be a random variable defined on the probability space (Ω, \mathcal{F}, P) , and assume that $r(\omega) \in [a, b], \forall \omega \in \Omega$. Then there is a measurable function $\xi : \Omega \rightarrow \mathbb{R}$ such that $\xi(\omega)$ lies in the closed interval with endpoints 0 and $r(\omega)$ for each $\omega \in \Omega$, and*

$$L(r(\omega)) = L(0) + \sum_{k=1}^{n-1} \frac{1}{k!} L^{(k)}(0) r^k(\omega) + \frac{1}{n!} L^{(n)}(\xi(\omega)) r^n(\omega). \quad (2.6)$$

Proof: See theorem 18.18 from Aliprantis and Border [3]. \square

We now look at the first three terms of the (stochastic) Taylor series expansion of the loss function (2.5), and then take their expectation. That is, setting $r = V_K^\pi - V_0$, $L(x) = e^{-\gamma x}$ and $n = 3$ in equation (2.6), and assuming the random variable r describing the portfolio performance must be constrained to lie in some closed interval, we have

$$e^{-\gamma(V_K^\pi - V_0)} = 1 - \gamma(V_K^\pi - V_0) + \frac{\gamma^2}{2}(V_K^\pi - V_0)^2 - \frac{\gamma^3}{3!}e^{-\gamma\xi}(V_K^\pi - V_0)^3,$$

for some measurable function ξ . Hence the objective function can be expressed as

$$\mathbf{E}_0[L(V_K^\pi - V_0)] = -\mathbf{E}_0[V_K^\pi - V_0] + \frac{\gamma}{2}\mathbf{E}_0[(V_K^\pi - V_0)^2] - \frac{\gamma^2}{3!}\mathbf{E}_0[e^{-\gamma\xi}(V_K^\pi - V_0)^3]. \quad (2.7)$$

For low values of γ , the term $\mathbf{E}_0[V_K^\pi - V_0]$ will dominate. If there are no transaction costs and the portfolio value process $\{V_k^\pi : k = 0, \dots, K\}$ is a martingale, then $\mathbf{E}_0[V_K^\pi - V_0] = 0$. So in that case, again for low γ , the second term in equation (2.7) will dominate. Note that this will then be proportional to the variance of the terminal portfolio value V_K^π .

Negative exponential risk is often used in theoretical works on hedging (see Hodges and Neuberger [71], Clewlow and Hodges [40], Grasselli and Hurd [64]). As pointed out by previous authors (for example in [71]), this risk measure has many advantages:

- It is simple to define.
- It penalizes losses, but not gains (as opposed to the quadratic risk measure where $L(x) = x^2$, for example).
- As we will see in section 2.4, the optimal decision at any given step k does not depend on the cash amount c_k . Thus the state space can effectively be reduced

by one dimension.

However, the risk aversion parameter γ is not known a priori, and must be chosen by the hedger. This is not a real problem, since various values of γ will yield different risk/reward profiles (this can be observed by simulation, for example), from which the hedger can choose a suitable γ . See for example Zakamouline [138], Sinclair [125] (pages 66-68). A study by Bliss and Panigirtzoglou [23] found possible values of γ that were coherent with observed option prices on the FTSE and S&P500 stock indices during periods covering the years 1983 to 2001, and concluded that they depended on many factors, including the time horizon and the option volatility levels.

Another potential problem is that the expectation of the loss function (2.5) may not even be defined in some basic settings. As an example, consider a stock price that follows a lognormal distribution and a portfolio that has a negative exposure (or a *short* position, in financial jargon) to the stock price as the price goes to infinity (i.e., the portfolio value $V_k(X_k)$ goes to negative infinity as the stock price s_k goes up). The latter will happen for example if we sold n call options and hedge them by holding less than n units of the underlying stock. Thus the optimal solution will always involve holding at least n units of the stock, irrespective of the risk aversion level. But in practice, there is of course a limit to the amount one can lose. Furthermore, this can be addressed by truncating the stock price distribution, for example (see section 2.1.2).

There are other, more general formulations to the hedging problem. For example, Schachermayer [119] rigorously shows the existence of an optimal solution for a large class of utility functions (including exponential utility), even when allowing the portfolio value to take any value on the real line \mathbb{R} . The idea is to start by considering portfolio processes that are uniformly bounded from below, so that the utility function values can be defined, and then taking an appropriate limit to define a larger set of processes over which the utility function is to be maximized.

Other risk measures could of course be used, such as quadratic risk (see Schweizer [121] and Černý [131]) or conditional Value-at-Risk (see Rockafellar and Uryasev [113], [112], for example). These are all examples of *convex* risk measures, which are usually preferred, as the theory for solving convex optimization problems is well understood. For a more detailed discussion of other possible risk measures for use in portfolio optimization problems, we refer the reader to Prigent [110], Rockafellar and Uryasev [113], [112], Rockafellar, Uryasev and Zabarankin [114] and Ortobelli, Rachev, Stoyanov, Fabozzi and Biglova [106].

2.1.4 Derivative pricing

To simulate portfolio value paths $(V_0, V_1^\pi, \dots, V_K^\pi)$, we need to know the value $h_k(Y_k)$ of the derivative to hedge at each step k . The basic requirement from financial theory is that this pricing should not lead to arbitrage opportunities, i.e., setting up a portfolio which is guaranteed to have strictly positive return. Various versions of the Fundamental Theorem of Asset Pricing (FTAP) gives conditions for this to be true, which essentially say that a market model is arbitrage free if and only if there exists an equivalent martingale measure. See Delbaen and Schachermayer [49] for a very detailed account of this theory, at various levels of generality. In our setting (finite discrete time), a relevant version of the FTAP which applies was provided by Dalang, Morton and Willinger [46]. Under a martingale measure Q , the price $h_k(Y_k)$ of any (sufficiently integrable) derivative is given by the expectation of its payoff at expiry :

$$h_k(Y_k) = \mathbf{E}_k^Q[h_K(Y_K)]. \quad (2.8)$$

For models more general than the Black-Scholes-Merton model mentioned earlier, some option pricing formulas do exist (see for example Hull and White [73], for a standard call option under a particular stochastic volatility process where the volatility and stock price processes are uncorrelated). But in general, no exact formula is

known for the expectation given by equation (2.8). This will be addressed later in section 3.2.4, using the stochastic mesh construction to estimate derivative prices.

2.2 Dynamic Programming algorithm

The theoretical solution to problem (2.3) is well understood in terms of the DP algorithm. This section presents briefly how this algorithm can be applied in theory and sets some associated notation. See Bertsekas and Shreve [18], for example, for a rigorous treatment of the subject. We will discuss in chapter 3 some numerical aspects of DP relevant to our problem.

Recall from section 2.1.1 that given a policy π , we write the step k portfolio value as $V_k^\pi(X_0)$, which is a random variable determined by starting at state $X_0 = (c_0, u_0, Y_0)$ and applying the policy π and the transition functions ψ from equation (2.2), from step $l = 0$ until step $l = k$.

At each step k , the risk function J_k^π is defined as the expected loss on the terminal value $V_K^\pi(X_k)$ of the portfolio when the policy π is applied at steps $k, \dots, K - 1$, starting from state X_k . We write

$$J_k^\pi(X_k) = \mathbf{E}_k[L(V_K^\pi(X_k) - V_0)], \quad (2.9)$$

and the problem is then to minimize $J_0^\pi(X_0)$ over a given space \mathcal{A} of admissible policies, i.e., to compute

$$J_0^*(X_0) = \min_{\pi \in \mathcal{A}} J_0^\pi(X_0). \quad (2.10)$$

The solution to this problem can be obtained in principle by the DP algorithm, shown in algorithm 1. Given an initial state X_0 , the DP algorithm computes $J_0^*(X_0)$, the optimal risk at step $k = 0$, and the associated optimal policy π^* by backward recursion. This assumes that the risk functions J_k^* can be defined exactly for any state $X_k \in \mathcal{X}$, even if the state space \mathcal{X} is infinite. Approximate versions of this

algorithm (e.g. based on a finite subset of states) will be discussed later in chapter 3.

Algorithm 1 Dynamic Programming

1 Let $k = K$. For any $X_K \in \mathcal{X}$, define

$$J_K^*(X_K) := L(V_K(X_K) - V_0). \quad (2.11)$$

2 For $k = k - 1$ to 0. For any $X_k \in \mathcal{X}$, let

$$J_k^*(X_k) := \min_{v \in \mathcal{U}} Q_k(X_k, v), \quad (2.12)$$

where

$$Q_k(X_k, v) := \mathbf{E}_k[J_{k+1}^*(\psi(X_k, v, Y_{k+1}))], \quad (2.13)$$

for v the decision at stage k , and let

$$\mu_{k+1}^*(X_k) := \arg \min_{v \in \mathcal{U}} Q_k(X_k, v) \quad (2.14)$$

We now introduce some additional definitions that will be used only in the remainder of this section, to relate our problem explicitly to an important result from the classic text by Bertsekas and Shreve [18]. Let $H : \mathcal{X} \times \mathcal{U} \times F \rightarrow \mathbb{R}$ be given by

$$H(X_k, v, J) = \mathbf{E}_k [J(\psi(X_k, v, Y_{k+1}))] \quad (2.15)$$

and where F is the set of all extended real value functions $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$. Furthermore, let $T : F \rightarrow F$ be defined by

$$T(J)(X_k) := \min_{v \in \mathcal{U}} H(X_k, v, J). \quad (2.16)$$

The following proposition confirms that the DP algorithm is indeed valid.

Proposition 2.2.1 (*Bertsekas and Shreve [18], proposition 3.4*) *Assume that the function H is monotone, in the sense that $H(X_k, v, J) \leq H(X_k, v, J')$, $\forall X_k \in \mathcal{X}, v \in$*

$\mathcal{U}, J, J' \in F$ s.t. $J \leq J'$. Let the control space \mathcal{U} be a Hausdorff space and assume that for each $X_k \in \mathcal{X}, \lambda \in \mathbb{R}$, and $k = 0, 1, \dots, K - 1$, the set

$$U_k(X_k, \lambda) = \{u \in \mathcal{U} : H[X_k, u, T^k(J_0)] \leq \lambda\} \quad (2.17)$$

is compact. Then

$$J_K^* = T^K(J_0), \quad (2.18)$$

and there exists a uniformly K -stage optimal policy, i.e. a policy $\pi^* = (\mu_1^*, \dots, \mu_K^*)$ such that the sub-policies $\pi^{(k)} = (\mu_{k+1}^*, \dots, \mu_K^*)$ are all k -stage optimal, that is, $J_k^{\pi^{(k)}}(X_k) = J_k^*(X_k)$ for all $X_k \in \mathcal{X}$ and for all $k = 0, \dots, K - 1$.

One important technical point for algorithm 1 to be defined is to make sure that the risk functions $J_k^*(X_k)$ are integrable with respect to the measure P , so that the expectations (2.13) can be defined. This is not obvious in general because of the minimization step involved in defining the $J_k^*(X_k)$. We give more details about this in section 2.5.

2.3 General form of solutions

Under some convexity conditions for the function J_k^* , we will see below that the optimal policies can be characterized in terms of a *no-trading* interval $[b_k^-, b_k^+]$, such that the optimal decision is

1. trade to the boundary if u_k is outside of $[b_k^-, b_k^+]$, and
2. do nothing if $u_k \in [b_k^-, b_k^+]$.

As we will discuss later, the bounds b_k^+ and b_k^- may possibly be functions of the state X_k , but this will depend on the specific loss function L used in the objective. We show in two ways that the optimal decision rule behaves as described above : one

way is via an explicit characterization of the no-transaction boundary, and the other is by applying the well-known Karush-Kuhn-Tucker conditions. These results hold for a general convex loss function L . More details will later be given in section 2.4 for the particular case of the negative exponential loss function.

2.3.1 A direct approach

To obtain the bounds b_k^+ and b_k^- , there is a complication that arises when computing the minimum of Q_k as a function of v , for a fixed state X_k . Ideally, we would like the function $Q_k(X_k, \cdot)$ to be strictly convex for any given X_k , as this would simplify the analysis of the solutions. But here, the transition function $\psi(X_k, v, Y_{k+1})$ of equation (2.2) is not linear in the decision v (because of an absolute value due to the presence of transaction costs), which prevents the function $Q_k(X_k, \cdot)$ from being convex in v in general. So instead of using $\psi(X_k, v, Y_{k+1})$ directly, we first define two new *linear* transition functions ψ^+ and ψ^- as $\psi^\pm(X_k, v, Y_{k+1}) = (c_{k+1}^\pm, v, Y_{k+1})$, where

$$c_{k+1}^\pm = c_k - (v - u_k)s_k \pm (a + bs_k)(v - u_k), \quad (2.19)$$

i.e., we remove the absolute values on $v - u_k$ in the transaction cost term $(a + bs_k)|v - u_k|$ and add a plus or a minus sign in front.

We then define

$$Q_k^-(X_k, v) := \mathbf{E}_k[J_{k+1}^*(\psi^-(X_k, v, Y_{k+1}))] \quad (2.20)$$

and

$$Q_k^+(X_k, v) := \mathbf{E}_k[J_{k+1}^*(\psi^+(X_k, v, Y_{k+1}))]. \quad (2.21)$$

Thus $Q_k^-(X_k, v)$ will be the same as $Q_k(X_k, v)$ when $v \geq u_k$ and $Q_k^+(X_k, v)$ will be

the same as $Q(X_k, v)$ when $v \leq u_k$. And finally, let

$$b_k^-(X_k) := \arg \min_{v \in \mathcal{U}} Q_k^-(X_k, v), \quad (2.22)$$

$$b_k^+(X_k) := \arg \min_{v \in \mathcal{U}} Q_k^+(X_k, v). \quad (2.23)$$

Throughout this thesis, we will work with functions $Q_k^\pm(X_k, \cdot)$ that are strictly convex, from which it follows that the values $b_k^\pm(X_k)$ are uniquely defined, since we originally fixed \mathcal{U} to be a closed interval (hence a compact set). More generally, if the functions $Q_k^\pm(X_k, \cdot)$ were allowed to be convex (but not strictly convex), then they could have more than one minimum.

The following proposition describes the optimal policy in terms of $b_k^+(X_k)$ and $b_k^-(X_k)$, which turn out to define the boundary of a no-transaction region :

Proposition 2.3.1 *Assume that the $k+1$ stage risk function $J_{k+1}^*(\cdot, Y_{k+1})$ is strictly convex as a function of (c_{k+1}, u_{k+1}) , for Y_{k+1} fixed. Then for a given state $X_k = (c_k, u_k, Y_k)$, either 1) the boundaries are equal, i.e., $b_k^-(X_k) = b_k^+(X_k)$, in which case the the optimal stock quantity v^* at stage k is given by*

$$v^* = b_k^-(X_k) = b_k^+(X_k), \quad (2.24)$$

or 2) $b_k^-(X_k) \neq b_k^+(X_k)$ and the optimal stock quantity v^* at stage k is given by

$$v^* = b_k^-(X_k) \text{ for } u_k \leq b_k^-(X_k), \quad (2.25)$$

$$v^* = b_k^+(X_k) \text{ for } u_k \geq b_k^+(X_k) \quad (2.26)$$

and

$$v^* = u_k \text{ for } u_k \in [b_k^-(X_k), b_k^+(X_k)], \quad (2.27)$$

and for any fixed Y_k the optimal risk function $J_k^*(\cdot, Y_k)$ at stage k is strictly convex

in (c_k, u_k) , for all $c_k \in \mathbb{R}$, $u_k \in \mathcal{U}$.

Proof :

The case with equal boundaries is trivial, so we consider the second case, where $b_k^-(X_k) \neq b_k^+(X_k)$. For the first part (i.e., the value v^* of the optimal decision), we consider a *fixed* state $X_k = (c_k, u_k, Y_k)$, so as a shorthand, we write $Q_k(v)$ and $Q_k^\pm(v)$ instead of $Q_k(X_k, v)$ and $Q_k^\pm(X_k, v)$. We will prove the result by comparing the values $Q_k^+(v)$ and $Q_k^-(v)$ to $Q_k(v)$ over different subintervals for v . The convexity of $J_k^*(\cdot, Y_k)$ will be shown afterwards.

Assume first that $v^* \geq u_k$ and consider the following two possibilities for $b_k^-(X_k)$:

1. $b_k^-(X_k) \geq u_k$, and
2. $b_k^-(X_k) < u_k$.

For the first possibility, $Q_k(v) = Q_k^-(v)$ for $v \geq u_k$, so $v^* = b_k^-(X_k)$. For the second possibility, we must have $v^* = u_k$. Otherwise $v^* > u_k$ (by our initial assumption), and as we will show below, this would imply that there exists a quantity \bar{v} such that $Q_k^-(\bar{v}) < Q_k^-(b_k^-(X_k))$, which would be a contradiction, since $b_k^-(X_k)$ minimizes Q_k^- by definition. To see this, choose an $\alpha \in (0, 1)$ such that the quantity $\bar{v} := \alpha b_k^-(X_k) + (1 - \alpha)v^*$ satisfies $\bar{v} > u_k$. Then, using the fact that $Q_k(v) = Q_k^-(v)$ for $v \geq u_k$, we have

$$Q_k^-(v^*) = Q_k(v^*) \leq Q_k(\bar{v}) = Q_k^-(\bar{v}) \quad (2.28)$$

Now note that Q_k^- is strictly convex as a function of c_k, u_k and v , since it is defined as the expectation of the composition of the strictly convex function $J_{k+1}^*(\cdot, Y_{k+1})$ with the linear mapping $\psi^-(X_k, v, Y_{k+1})$. This implies that

$$Q_k^-(\bar{v}) < \alpha Q_k^-(b_k^-(X_k)) + (1 - \alpha)Q_k^-(v^*) \leq \alpha Q_k^-(b_k^-(X_k)) + (1 - \alpha)Q_k^-(\bar{v}), \quad (2.29)$$

where the first inequality uses the strict convexity of $Q_k^-(\cdot)$ in v and the second

inequality follows from $Q_k^-(v^*) \leq Q_k^-(\bar{v})$. Thus $Q_k^-(\bar{v}) < Q_k^-(b_k^-)$, which is what we needed to prove.

For the case $v^* \leq u_k$, we can use a similar argument to show that we have $v^* = b_k^+(X_k)$ if $b_k^+(X_k) \leq u_k$ and $v^* = u_k$ if $b_k^+(X_k) > u_k$.

To conclude the proof, we need to show the strict convexity of $J_k^*(\cdot, Y_k)$ as a function of $(c_k, u_k) \in \mathbb{R} \times \mathcal{U}$, for any given Y_k , where $J_k^*(X_k) = Q_k(X_k, v^*)$. Note that for $u_k \leq b_k^+(X_k)$, we have $Q_k(X_k, v^*) = Q_k^-(X_k, v^*)$ since $v^* = u_k$ in the range $[b_k^-(X_k), b_k^+(X_k)]$, in which case there are no transaction costs. Since $Q_k^-(\cdot, Y_k, v)$ is a strictly convex function of (c_k, u_k) for any v , then $Q_k(\cdot, Y_k, v^*)$ is strictly convex for $(c_k, u_k) \in A_1 := \{(c_k, u_k) : u_k \leq b_k^+(X_k)\}$. By symmetry, we also see that $Q_k(\cdot, Y_k, v^*)$ is strictly convex in $(c_k, u_k) \in A_2 := \{(c_k, u_k) : u_k \geq b_k^-(X_k)\}$. Note that $A_1 \cup A_2 = \mathbb{R} \times \mathcal{U}$ is also a convex set and that A_1 and A_2 have disjoint boundaries, so $Q_k(\cdot, Y_k, v^*)$ must also be strictly convex on $\mathbb{R} \times \mathcal{U}$. To see this more explicitly, suppose that $Q_k(\cdot, Y_k, v^*)$ is not strictly convex on this domain. Then there must exist some points $x \in A_1, y \in A_2, z \in A_1 \cup A_2$ and a real number $\lambda \in (0, 1)$ such that

$$Q_k((z, Y_k), v^*) \geq \lambda Q_k((x, Y_k), v^*) + (1 - \lambda) Q_k((y, Y_k), v^*). \quad (2.30)$$

Here we assume without loss of generality that x and y are not both in A_1 or A_2 , because otherwise we would have a contradiction since $Q_k(\cdot, Y_k, v)$ is already known to be convex over these two sets. But then we can simply take two other points x' and y' along the line segment \overline{xy} such that $\frac{1}{2}x' + \frac{1}{2}y' = z$ and both $x', y' \in A_1$ or $x', y' \in A_2$, which again yields a contradiction. \square

In proposition 2.3.1, nothing was said about the ordering of the bounds $b_k^-(X_k)$ and $b_k^+(X_k)$, other than implicitly considering the case $b_k^-(X_k) \leq b_k^+(X_k)$ in equation (2.27). The following simple corollary restricts the possible orderings :

Corollary 2.3.2 1) Let $X_k = (c_k, u_k, Y_k) \in \mathcal{X}$ for some k . Then it is not possible to

have $b_k^-(X_k) > u_k > b_k^+(X_k)$. 2) In particular, if $b_k^-(X_k)$ and $b_k^+(X_k)$ are independent of u_k , then we must have $b_k^-(X_k) \leq b_k^+(X_k)$.

Proof : For the first part, assume that $b_k^-(X_k) > u_k > b_k^+(X_k)$. By proposition 2.3.1, we must have that the optimal decision v^* is given by $v^* = b_k^-(X_k)$, since $b_k^-(X_k) > u_k$. But by the same proposition, we must also have $v^* = b_k^+(X_k)$, since $b_k^+(X_k) < u_k$. Thus this implies $b_k^-(X_k) = b_k^+(X_k)$, which is a contradiction. For the second part, assume that $b_k^-(X_k) > b_k^+(X_k)$. So there exists a u such that $b_k^-(X_k) > u > b_k^+(X_k)$. Hence if we set $u_k = u$ and consider the first result, we reach a contradiction. \square

2.3.2 The Karush-Kuhn-Tucker conditions

It is also possible to characterize the optimal solution of a convex program via the Karush-Kuhn-Tucker conditions. For a classic reference, see for example the text by Rockafellar [115], section 28. As mentioned earlier in this section, the objective function $Q_k(X_k, \cdot)$ in equation (2.12) will not generally be convex in v due to the presence of the absolute value term $|u_k - v|$ for the transaction costs in the financing equation (2.4). But we can recast the problem as a convex program with linear inequality constraints by introducing two non-negative variables $l, m \geq 0$ and make the following substitutions :

- $v - u_k \rightarrow l - m$,
- $|v - u_k| \rightarrow l + m$, and
- $Q_k(X_k, v) \rightarrow Q_k^L(X_k, l, m)$,

where

$$Q_k^L(X_k, l, m) := \mathbf{E}_k [J_{k+1}^*(\psi^L(X_k, l, m, Y_{k+1}))], \quad (2.31)$$

$$\psi^L(X_k, l, m, Y_{k+1}) := (c_{k+1}^L, u_k + (l - m), Y_{k+1}), \quad (2.32)$$

$$c_{k+1}^L := c_k - (l - m)s_k - (l + m)(a + bs_k). \quad (2.33)$$

Assuming for simplicity that there are no constraints on admissible decisions (i.e. $\mathcal{U} = \mathbb{R}$), the step k program to solve then becomes

$$\min_{l, m \geq 0} Q_k^L(X_k, l, m), \quad (2.34)$$

and, introducing the dual variables λ and μ , the KKT conditions can be written as

1. (Stationarity)

$$\frac{\partial Q_k^L}{\partial l}(X_k, l, m) + \lambda = 0, \quad (2.35)$$

$$\frac{\partial Q_k^L}{\partial m}(X_k, l, m) + \mu = 0, \quad (2.36)$$

2. (Primal feasibility)

$$l, m \geq 0, \quad (2.37)$$

3. (Dual feasibility)

$$\lambda, \mu \text{ unconstrained} \quad (2.38)$$

4. (Complementary slackness)

$$l \cdot \lambda = 0, m \cdot \mu = 0. \quad (2.39)$$

The two variables l and m can be interpreted as the step k quantity of stock bought, or sold, respectively. Note that it is possible to add a positive constant $C > 0$ to both l and m , to obtain the modified quantities $l' = l + C, m' = m + C$, in which case the net transaction will be the same, i.e.

$$v - u_k = l' - m' = (l + C) - (m + C) = l - m, \quad (2.40)$$

but with higher transaction costs, given by

$$\phi(s_k)(l' + m') = \phi(s_k)(l + m + 2C) > \phi(s_k)(l + m), \quad (2.41)$$

assuming non-zero unit costs $\phi(s_k)$. But if the loss function $L(x)$ is monotone decreasing, so that $x > y \Rightarrow L(x) < L(y)$, these higher cost solutions can never be optimal, and so will simply be discarded. Thus we have shown the following :

Proposition 2.3.3 *Assume that the loss function $L(x)$ is monotone decreasing and that the optima of the programs defined by equations (2.12) and (2.34), are attained respectively at the points v^* and (l^*, m^*) . Then $Q_k(X_k, v^*) = Q_k^L(X_k, l^*, m^*)$, with $v^* = l^* - m^*$.*

Let us now consider the various possibilities for the optimal values of l and m , to link these to the no-transaction region $[b_k^-(X_k), b_k^+(X_k)]$ described in section 2.3.1. The case of inactive constraints (with both $l > 0$ and $m > 0$) can never correspond to an optimum, as explained above. The three cases for active constraints (i.e., 1. $l = 0, m > 0$, 2. $l \geq 0, m = 0$ and 3. $l = m = 0$) correspond either to trading (cases 1 and 2) or not trading (case 3). Assume a fixed cash level c_k and market vector Y_k at step k . Write $l^*(u)$ and $m^*(u)$ for the optimal solution to program (2.34) when $u_k = u$ and define $\mathcal{U}^0(c_k, Y_k)$ as the set of stock quantities u_k at step k such that the optimal solution for problem (2.34) is not to trade. Now assume u^1 is another stock quantity, such that if $u_k = u^1$, then the optimal solution is given by $(l^*(u^1), m^*(u^1)) = (l^1, 0)$. And clearly, we must have $u^1 + l^1 \in \mathcal{U}^0(c_k, Y_k)$, i.e. if $u_k = u^1 + l^1$ to begin with, we attain the same stock position, but without any trading costs, so it must be optimal not to trade. Write the set of all such stock quantities A^- , i.e. let $A^- = \{u : \text{for } u_k = u, \text{ the optimal solution is given by } l^*(u) > 0 \text{ and } m^*(u) = 0\}$. For $a^- = \sup A^-$, we must then have $u + l^*(u) = a^-, \forall u \in A^-$. Similarly, we can define $A^+ = \{u : \text{for } u_k = u, \text{ the optimal solution is given by } l^*(u) = 0 \text{ and } m^*(u) > 0\}$

and $a^+ = \inf A^+$, for which we must then have $u - m^*(u) = a^+, \forall u \in A^+$. Thus we reobtain the no-transaction boundaries as $b_k^-(c_k, Y_k) = a^-$ and $b_k^+(c_k, Y_k) = a^+$, which shows again that the optimal policy can be separated into trading and no-trading regions, as in equations (2.25)-(2.27).

2.4 Results specific to the negative exponential loss function

For the remainder of this thesis, we will focus on the case of the negative exponential loss function.

To simplify the notation, we will slightly redefine the step k risk function J_k^π associated to the policy π as the function G_k^π , given by

$$G_k^\pi(X_k) := \mathbf{E}_k[e^{-\gamma(V_K^\pi(X_k) - V_k(X_k))}], \forall X_k \in \mathcal{X}. \quad (2.42)$$

The difference with the previous definition $J_k^\pi(X_k)$ is a factor $e^{-\gamma(V_k(X_k) - V_0)} \in \mathcal{F}_k$, which is known at every step k . That is,

$$J_k^\pi(X_k) = e^{-\gamma(V_k(X_k) - V_0)} G_k^\pi(X_k). \quad (2.43)$$

In particular, we have $G_0^\pi(X_0) = J_0^\pi(X_0)$ and $G_K^\pi(X_K) = 1$. The function Q_k in equation (2.13) can then conveniently be replaced by the function R_k defined by

$$R_k(X_k, v) := \mathbf{E}_k[g(X_k, v, Y_{k+1}) G_{k+1}^*(X_{k+1})], \forall X_k \in \mathcal{X}, v \in \mathcal{U}, \quad (2.44)$$

using the one-stage risk functions

$$g(X_k, v, Y_{k+1}) := e^{-\gamma(V_{k+1}(X_{k+1}) - V_k(X_k))} \quad (2.45)$$

and where $X_{k+1} = \psi(X_k, v, Y_{k+1})$.

Using this notation, we have the following :

Proposition 2.4.1 *Let $L(x) = \exp(-\gamma x)$. Then for any $k = 0, \dots, K - 1$ and $X_k = (c_k, u_k, Y_k)$, we have that a) the transaction boundaries $b_k^-(X_k)$ and $b_k^+(X_k)$ are only functions of Y_k , and are independent of u_k and c_k , and b) the optimal risk function G_k^* is independent of c_k and its logarithm $\ln G_k^*$ takes the following form*

$$\ln G_k^*(X_k) = \beta_1(s_k)u_k + \eta_1(Y_k), \text{ for } u_k \leq b_k^- \quad (2.46)$$

$$\ln G_k^*(X_k) = \beta_2(s_k)u_k + \eta_2(Y_k), \text{ for } u_k \geq b_k^+ \quad (2.47)$$

$$\ln G_k^*(X_k) = \alpha_k(Y_k, u_k), \text{ for } u_k \in [b_k^-, b_k^+], \quad (2.48)$$

where $\beta_1(s_k) = -\gamma\phi(s_k)$, $\beta_2(s_k) = +\gamma\phi(s_k)$, $\eta_j(Y_k) \in \mathbb{R}$, $j = 1, 2$, are independent of u_k and c_k and $\alpha_k : \mathcal{Y} \times \mathcal{U} \rightarrow [0, \infty)$ is a function such that $\alpha_k(Y_k, \cdot)$ is strictly convex, $\forall Y_k \in \mathcal{Y}$.

Proof: We will prove the result by backward induction on k . For $k = K$, we have $G_K^*(X_K) = 1$. Now assume the proposition holds for some $k + 1$.

To prove a), start by noting that we can always write

$$R_k^\pm(X_k, v) = e^{\mp\gamma\phi(s_k)(v-u_k)} \mathbf{E}_k[e^{-\gamma(v\Delta s_{k+1} + \Delta h_{k+1})} G_{k+1}^*(c_{k+1}, v, Y_{k+1})],$$

where c_{k+1} is given by equation (2.4), and depends on c_k and u_k . But by hypothesis $G_{k+1}^*(X_{k+1})$ is independent of c_{k+1} . Furthermore, u_k plays no role when optimizing $R_k^\pm(X_k, \cdot)$ as a function of v , since it appears only through a multiplicative factor. Thus we must have that the boundaries $b_k^\pm(X_k)$ are independent of both c_k and u_k .

For the proof of b) we consider proposition 2.3.1, in which we have the following three cases:

1. If $u_k \leq b_k^-$, then $v^* = b_k^-$, so that $G_k^*(X_k) = R_k(X_k, v^*) = R_k^-(X_k, b_k^-)$. More

explicitly, we can write

$$\begin{aligned} R_k^-(X_k, b_k^-) &= \mathbb{E}_k[e^{-\gamma(b_k^- \Delta s_{k+1} - \phi(s_k)(b_k^- - u_k) + \Delta h_{k+1})}] \\ &= f(b_k^-) e^{-\gamma \phi(s_k) u_k}, \end{aligned}$$

where

$$f(b_k^-) = \mathbb{E}_k[e^{-\gamma(b_k^- (\Delta s_{k+1} - \phi(s_k)) + \Delta h_{k+1})}]$$

is independent of u_k . Thus we can write $\ln G_k^*(X_k) = \beta_1(s_k) u_k + \eta_1(Y_k)$, with $\beta_1(s_k) = -\gamma \phi(s_k)$.

2. Similarly, if $u_k \geq b_k^+$, we have that $\ln G_k^*(X_k) = \beta_2(s_k) u_k + \eta_2(Y_k)$, with $\beta_2(s_k) = +\gamma \phi(s_k)$.
3. For the case with $u_k \in [b_k^-, b_k^+]$, it is optimal not to trade, so $G_k(X_k) = R_k(X_k, u_k)$, which implies that

$$G_k(X_k) = \mathbb{E}_k[e^{-\gamma(v \Delta s_{k+1} + \Delta h_{k+1})} G_{k+1}^*(c_{k+1}, v, Y_{k+1})]. \quad (2.49)$$

Thus $G_k(X_k)$ does not depend on c_k in this case either. And finally, the strict convexity of $\alpha_k(Y_k, \cdot) := \ln G_k(c_k, \cdot, Y_k)$ in u_k , restricted to $u_k \in [b_k^-, b_k^+]$, follows from proposition 2.3.1. \square

Proposition 2.4.1 will be used in numerical experiments to define approximations for the optimal risk function (see section 3.3.1). Note that these solutions still have parameters that depend on the state Y_k of the market at the given step k . This dependence could be complicated and related to other factors such as the specific derivative being hedged. Note also that proposition 2.3.1 is more general and applies, for example, in the case where $L(x) = x^2$, i.e. the loss function is quadratic. In that case, however, the boundaries $b_k^\pm(X_k)$ may depend on the values c_k and u_k .

2.5 Existence of solutions : measurability conditions

As mentioned earlier, an important theoretical issue associated with stochastic DP is that the policies and risk functions obtained at each step of the DP algorithm remain measurable in some sense. So in this section, we provide some explicit conditions for our problem to be well-defined.

The general theory is very technical (see Bertsekas and Shreve [18], chapters 7 and 8), and we do not attempt to cover it here. Instead, we simply show in proposition 2.5.4 below that our hedging problem using the negative exponential loss function does indeed meet the conditions given in section 11.3 of Bertsekas and Shreve [18] for a Borel model with multiplicative costs. Then, in corollary 2.5.5, the validity of the DP algorithm in our setting will follow from a result in [18].

But first let us recall some definitions from descriptive set theory.

Definition 2.5.1 (*Bertsekas and Shreve [18], definition 7.7*) *A topological space X is said to be a Borel space if there exists a complete separable metric space Y and a Borel subset $B \in \mathcal{B}_Y$ (where \mathcal{B}_Y is the Borel σ -algebra of Y) such that X is homeomorphic to B . The empty set is also regarded as a Borel space.*

Definition 2.5.2 (*Bertsekas and Shreve [18], definition 7.8*) *Let X and Y be topological spaces. A function $f : X \rightarrow Y$ is Borel-measurable if $f^{-1}(B) \in \mathcal{B}_X$ for every $B \in \mathcal{B}_Y$.*

Definition 2.5.3 (*Bertsekas and Shreve [18], definition 7.16*) *Let X be a Borel space and denote by \mathcal{F}_X the collection of closed subsets of X . The analytic subsets of X are the members of $\mathcal{S}(\mathcal{F}_X)$, the set of all nuclei of Suslin schemes for \mathcal{F}_X (see [18], definition 7.15).*

We can now state the following :

Proposition 2.5.4 *Suppose that we are using the negative exponential risk function as described in section 2.4 and that the densities $f_k(Y_k, Y_{k+1})$ are Borel-measurable functions, for all $k = 0, \dots, K - 1$. Then the hedging problem defined in section 2.1 meets the following conditions :*

1. *The state space \mathcal{X} , the control space \mathcal{U} and the market vector space \mathcal{Y} (called the disturbance space in [18]) are Borel spaces.*
2. *The set $\Gamma = \{(X_k, v) | X_k \in \mathcal{X}, v \in \mathcal{U}\}$ is analytic.*
3. *The density $f_k(Y_k, Y_{k+1})$ and the transition function $\psi(X_k, v, Y_{k+1})$ are Borel-measurable, for $X_k \in \mathcal{X}$ and $Y_{k+1} \in \mathcal{Y}$.*
4. *The one stage risk function $g(X_k, u, Y_{k+1})$ is Borel-measurable and bounded, in the sense that there exists a constant b such that $0 \leq g(X_k, u, Y_{k+1}) \leq b \forall X_k \in \mathcal{X}, u \in \mathcal{U}, Y_{k+1} \in \mathcal{Y}$.*

Proof:

1. We can take $Y = \mathbb{R}^{d+2}$ in definition (2.5.1). Then $\mathcal{U} = [u_{min}, u_{max}]$, $\mathcal{Y} = [0, \infty)^d$ and $\mathcal{X} = \mathbb{R} \times \mathcal{U} \times \mathcal{Y}$ are all Borel subsets of Y .
2. Here, we simply have $\Gamma = \mathcal{X} \times \mathcal{U}$, which is a product of Borel sets, and thus itself a Borel set (see Bertsekas and Shreve [18], proposition 7.13). And every Borel subset of a Borel space is analytic (see Bertsekas and Shreve [18], proposition 7.36).
3. The function $\psi(X_k, u, Y_{k+1})$ is a continuous functions, and thus Borel-measurable.
4. The function $g(X_k, u, Y_{k+1})$ is continuous, and thus Borel-measurable. Furthermore, note that the portfolio variations are assumed bounded (see section 2.1.2). Thus $g(X_k, u, Y_{k+1})$ must also be bounded. \square

We then have the following result, which directly follows from Bertsekas and Shreve [18], proposition 11.7, and from the definitions from section 2.2 :

Corollary 2.5.5 *The DP algorithm is valid for the hedging problem, in the sense that*

$$J_0^* = T^K(J_K). \quad (2.50)$$

Furthermore, there exists a uniformly K -stage optimal policy π^ .*

2.6 Conclusion

In this chapter, we made our hedging problem explicit, framing it as a discrete time stochastic dynamic program. We also specified two important elements of the problem : the state transition function (including linear transaction costs) and the negative exponential loss function. Using a more general convexity hypothesis on the loss function, we showed in proposition 2.3.1 how the optimal decisions at each step k are related to the two trading boundaries $b_k^-(X_k)$ and $b_k^+(X_k)$. For the particular case of the negative exponential loss function, we then provided two useful results in proposition 2.4.1. First, we showed that the trading boundaries were in fact only dependent on the market information vector Y_k , and not on the cash account c_k or the stock quantity held u_k . Second, we characterized the associated optimal risk function within the three regions delimited by the boundaries. These results will be exploited when constructing the stochastic mesh approximation described next in chapter 3. More precisely, for a fixed Y_k defining a mesh node, the optimal risk function J_k^* can then be well approximated by a simple piecewise quadratic function of the stock quantity u_k (see section 3.3.1). Finally, we verified that our formulation of the hedging problem was indeed well-defined, based on classical results by Bertsekas and Shreve [18].

CHAPTER 3

APPROXIMATE DYNAMIC PROGRAMMING WITH A STOCHASTIC MESH

It is usually not possible to obtain an exact solution to a dynamic program unless the problem has a special structure. As a basic example, if there is a *finite* (and relatively small) state space, a solution can be computed by brute force, i.e., by going through all the states. However, the state space described in section 2.1 is infinite, since states $X_k = (c_k, u_k, Y_k)$ include the quantity of stock held u_k , the corresponding market parameters Y_k (such as the price), as well as the amount of cash c_k in the portfolio, all of which are considered here as continuous variables. Obviously, one cannot evaluate explicitly the risk function J_k^* over all states $X_k \in \mathcal{X}_k$. So unless we impose additional structure on the problem, the solution must be approximated.

This chapter describes the use of *stochastic mesh methods* to compute approximate solutions to the hedging problem. This class of methods was originally developed for the problem of pricing American options (see Broadie and Glasserman [36] and Glasserman [59], chapter 8), in particular for multi-dimensional state spaces (for example, when pricing options involving five underlying assets or more). But this is the first time it is applied to the hedging problem.

One advantage of using a stochastic mesh is that it yields an (in-sample) estimator $\widehat{J}_0(X_0)$ of the optimal risk $J_0^*(X_0)$ that is *low* biased. This is in contrast with some generic Approximate Dynamic Programming (ADP) methodologies, such as approximating the optimal policy or risk function using a regression over a linear combination of predefined (“basis”) functions, where in general there is no simple way to analyze the bias. In the context of American option pricing, well-known regression methods include the least-squares Monte Carlo method (LSM) of Longstaff and

Schwartz [98], and the method of Tsitsiklis and Van Roy (TvR) [130]. As pointed out in Dion and L'Ecuyer [51], the bias of the in-sample price estimates can have either sign for both methods, although the bias from the LSM method is usually positive, and that of the TvR method is usually negative.

But for any policy π , we do know that computing a standard Monte Carlo estimate $\widehat{J}_0^\pi(X_0)$ of the risk (out-of-sample) will be a high biased estimator of the (optimal) minimum risk. That is, simulate N independent paths of the market vector process, apply the hedging policy to these paths to obtain the terminal portfolio values $V_K^{\pi,i}(X_0)$, $i = 1, \dots, N$, and compute a sample average of the portfolio loss function

$$\widehat{J}_0^\pi(X_0) = \frac{1}{N} \sum_{i=1}^N L(V_K^{\pi,i}(X_0) - V_0). \quad (3.1)$$

Combining these low and high biased estimators, stochastic mesh methods thus allow us to derive confidence intervals for the value of the optimal risk function (as was done by Broadie and Glasserman [36] for the price of American options).

We start by discussing basic elements of the field of ADP in section 3.1 to provide some context. See the books by Bertsekas [16], [17], and the lecture notes by L'Ecuyer [87] for a much broader overview. Applications of ADP methods can also be found, for example, in Haurie and L'Ecuyer [68], Ben Ameur, Breton and L'Ecuyer [12] and Ben Ameur, Breton, Karoui and L'Ecuyer [13]. Next, in section 3.2, we discuss the stochastic mesh methods, both in general and in the context of the hedging problem. In section 3.3, we introduce a simple linear-quadratic approximation that is meant to be used in conjunction with a stochastic mesh, and also provide some general error bounds for the risk function approximations at each stage. Section 3.4 provides more details on the low and high biased estimates derived from the stochastic mesh, and how they can be used to determine a confidence interval for the minimum of the risk function. Finally, the convergence of the mesh estimator is discussed in section 3.5.

3.1 Approximate Dynamic Programming

A central idea of ADP which we will use here is to apply the DP algorithm to an approximation \widehat{J}_k of the risk function, defined for a finite subset of N states $\mathcal{X}_k^N \subset \mathcal{X}_k$. More precisely, for $k = K$ to 0, we apply algorithm 2 below. But which states should be used to define an approximation?

Algorithm 2 Approximate Dynamic Programming

- 1 Assume we know approximate values $\widehat{J}_k^i \approx J_k^*(X_k^i)$, $i = 1, \dots, N$ of J_k^* over some finite subset of states $\mathcal{X}_k^N = \{X_k^i : i = 1, \dots, N\} \subset \mathcal{X}_k$.
 - 2 Use the values \widehat{J}_k^i to define an approximation \widehat{J}_k of J_k^* over *all* of \mathcal{X}_k . Note that this approximation may not necessarily go through the points (X_k^i, \widehat{J}_k^i) , i.e., it is possible that $\widehat{J}_k(X_k^i) \neq \widehat{J}_k^i$ for some or all i .
 - 3 Use the approximation \widehat{J}_k to calculate approximate risk function values $\widehat{J}_{k-1}^i \approx J_{k-1}^*(X_{k-1}^i)$, $i = 1, \dots, N$ in the next step $k - 1$ of the DP algorithm, for some X_{k-1}^i , $i = 1, \dots, N$, using equations (2.12) and (2.13) (or some approximate version thereof).
-

3.1.1 Simulating states

Assuming the state space \mathcal{X}_k has dimension d , a straightforward discretization into L points of each of the d components of the space involves approximating each function J_k^* over L^d states $X_k^i \in \mathcal{X}_k$, $i = 1, \dots, L^d$, for $k = 1, \dots, K$. The problem with this approach is that the number of states considered grows exponentially with d . This is an example of a *curse of dimensionality*, a term which was coined by Bellman [10]. This expression highlights the fact that approximating the solution of a high-dimensional dynamic program typically runs into some limitations in memory space and in computing time, unless the problem has some special structure (for example, sufficiently smooth functions J_k^*). See Powell [109] for other types of curses of dimensionality affecting ADP.

For some high dimensional DP problems, defining an approximate objective function over states sampled by Monte Carlo simulation can help overcome the “state space” curse of dimensionality described above. Intuitively, this works by ensuring that computational resources are spent on states that are more likely to be visited, instead of trying to cover the state space uniformly. For example, one could simulate N independent paths $\{(Y_0^i, \dots, Y_K^i) : i = 1, \dots, N\}$ of the market vector process to obtain a set of states $\mathcal{G}_k^N = \{Y_k^i : i = 1, \dots, N\}$ at each step k over which we can then apply the DP algorithm. The Stochastic Mesh, LSM and TvR methods mentioned at the beginning of this chapter all use this idea of defining the states for the approximate DP recursion (algorithm 2) by Monte Carlo simulations, in the context of American option pricing.

Another class of DP problems that can be solved using simulated states is discussed by Rust [117] (see also section 6.2). There, a state X_k is taken to be an element of the unit hypercube $[0, 1]^d$, with a transition density between any two states depending on an action with values taken from a finite set. This is somewhat similar to the American option pricing problem, where there is at every step a binary decision to be made (whether to exercise the option or not). However, a more precise correspondence between the two classes of problems remains to be done, and is beyond the scope of this research.

For the hedging problem, an added difficulty is that the evolution of the state X_k partly depends on the policy, e.g. the stock quantity u_k and the cash account c_k . But we can define states for the DP Algorithm by simulating market vector paths $\{(Y_0^i, \dots, Y_K^i) : i = 1, \dots, N\}$, which are independent of the policy used, and combine them with a (deterministic) grid for (c_k, u_k) , i.e., let the states for the approximation be given by the set $\mathcal{X}_k^{NM} := \{(Y_k^i, c_k^j, u_k^j), i = 1, \dots, N, j = 1, \dots, M\}$.

3.1.2 Approximating J_k^*

Once we have determined over which states the risk function J_k^* will be approximated, we still need to decide what kind of approximation to use.

Given an approximate risk function \widehat{J}_k at stage k , we can use it to define an approximation \widehat{J}_{k-1} of the optimal risk function J_{k-1}^* at the previous stage $k-1$. In equation (2.12), the DP algorithm computes the value of $J_{k-1}^*(X_{k-1})$ through minimizing the function $Q_{k-1}(X_{k-1}, \cdot)$. So we can compute

$$\widehat{J}_{k-1}^i := \inf_{v \in \mathcal{U}} \widehat{Q}_{k-1}(X_{k-1}^i, v), \quad (3.2)$$

where

$$\widehat{Q}_{k-1}(X_{k-1}^i, v) = \mathbf{E}_{k-1}[\widehat{J}_k(\psi_{k-1}(X_{k-1}^i, v, Y_k))], \quad (3.3)$$

for all the states $X_{k-1}^i \in \mathcal{X}^N$. Then we can define an approximate risk function \widehat{J}_{k-1} over all of \mathcal{X} as a function such that the pairs $\{(X_{k-1}^i, \widehat{J}_{k-1}(X_{k-1}^i)) : i = 1, \dots, N\}$ pass through or near $\{(X_{k-1}^i, \widehat{J}_{k-1}^i) : i = 1, \dots, N\}$. This can be done based on theoretical considerations (as we will do later in section 3.3) or by some numerical procedure, e.g., by defining the approximation function as a linear combination of basis functions, with coefficients determined by linear regression.

Note however that the conditional expectation in equation (3.3) may not have an analytical solution in general. So a second approximation could be required, but this time for the function \widehat{Q}_{k-1} . As we will see in section 3.2, one possibility is to define such an approximation using a stochastic mesh.

It is also possible to use basis functions to approximate the function Q_k instead of J_k^* . In the context of American option pricing, this type of approach was used for example by Tsitsiklis and Van Roy [130], and more recently by Dion and L'Ecuyer [51]. More generally, see also L'Ecuyer, Haurie and Hollander [91], Haurie and L'Ecuyer [68], L'Ecuyer [85], Ben Ameur, Breton and L'Ecuyer [12] and Ben Ameur, Breton,

Karoui and L'Ecuyer [13].

As discussed in section 4.2, one can also approximate directly the optimal policies instead of the functions J_k^* or Q_k . For applications of policy approximation to portfolio optimization problems, see for example Brandt, Goyal, Santa-Clara and Stroud [33] and Bolder-Rubin [25]. As pointed out in van Binsbergen and Brandt [133], approximating the optimal policy instead of the optimal risk function can yield better results. See also the analysis by Stentoft [128] for the problem of American option pricing.

3.2 Stochastic mesh methods

We now turn to the definition of stochastic mesh methods as class of approximations. As for other Monte Carlo-based ADP methods, it applies the DP algorithm on a set of states \mathcal{X}_k^N from $k = K$ to $k = 0$.

3.2.1 Basic construction

The basic construction of a stochastic mesh starts by defining nodes Y_k^j using a set of N independent paths $\{(Y_0^j, \dots, Y_K^j) : j = 1, \dots, N\}$ of the market vector process. More generally, for each step $k = 0, \dots, K$, one could also define the mesh nodes as a set of points $\mathcal{G}_k^N := \{Y_k^i : i = 1, \dots, N\}$, sampled according to some density $g_k(Y_k)$. This density could be the unconditional density at step k of the process $\{Y_k : k = 0, \dots, K\}$, but not necessarily.

The key element is that at a given time step k , the expectation of a function $\Phi_{k+1}(Y_{k+1})$, conditional on Y_k , is approximated using a weighted average over the simulated states $\mathcal{G}_{k+1}^N := \{Y_{k+1}^j : j = 1, \dots, N\}$ at the next step $k + 1$. That is,

$$\mathbf{E}_k[\Phi_{k+1}(Y_{k+1})] \approx \frac{1}{N} \sum_{j=1}^N w_{k,j} \Phi_{k+1}(Y_{k+1}^j), \quad (3.4)$$

where the $w_{k,j}$ are weights. The weights account for the fact that the points (vectors) $Y = Y_{k+1}^j$ at which the function $\Phi_{k+1}(Y)$ is evaluated may be sampled from a different probability distribution than that used by the expectation operator $\mathbf{E}_k[\cdot]$.

More generally, we assume that there is a weight function $w_k(Y_k, Y_{k+1})$ (possibly dependent on the step k) linking two consecutive market vectors Y_k and Y_{k+1} . For simulated market vectors Y_k^i and Y_{k+1}^j , we then define

$$w_{k,i,j} := w_k(Y_k^i, Y_{k+1}^j) \quad (3.5)$$

and, in the case where the starting state vector Y_k is fixed and arbitrary, as in equation (3.4), we drop the index i to write

$$w_{k,j} := w_k(Y_k, Y_{k+1}^j). \quad (3.6)$$

Assuming that $\mathbf{E}_k[w_k(Y_k, Y_{k+1})] = 1$, the weight function $w_k(Y_k, Y_{k+1})$ acts as a change of measure. Different possibilities for the choice of the weights will be discussed in section 3.2.3.

Having defined how conditional expectations are computed on the mesh nodes, we now give a general description of the ADP algorithm using a stochastic mesh :

As mentioned earlier, Broadie and Glasserman [35], [36] originally proposed this method to evaluate American options by simulation, in which case the function $\Phi_{k+1}(Y_k)$ would be the option price at step $k + 1$ (see also Glasserman's book [59], section 8.5).

3.2.2 Application to the hedging problem

The original application of the stochastic mesh method (American option pricing) involves a pointwise estimate of the option value at every node of the mesh. Here we describe how this approach can be adapted to the hedging problem under the

Algorithm 3 ADP on a stochastic mesh

- 1 Generate sets of N market vectors $\mathcal{G}_k^N := \{Y_k^i \in [0, \infty)^d : i = 1, \dots, N\}$, for each step $k = 0, \dots, K$.
 - 2 If the state vectors X_k have more elements than the market vectors Y_k (e.g. u_k and c_k), pick M possible values from these other elements of X_k and combine them to the N points in \mathcal{G}_k^N to define NM points $\mathcal{X}_k^{NM} \subset \mathcal{X}$. Otherwise, set $M = 1$.
 - 3 Compute weights $w_{k,i,j}$ for all pair of states $(Y_k^i, Y_{k+1}^j), i, j = 1, \dots, N, k = 0, \dots, K - 1, Y_k^i \in \mathcal{G}_k^N, Y_{k+1}^j \in \mathcal{G}_{k+1}^N$.
 - 4 Apply the ADP algorithm (see algorithm 2) on the states $X_k \in \mathcal{X}^{NM}$, replacing the functions J_k^* and Q_k by their corresponding stochastic mesh approximations \widehat{J}_k and \widehat{Q}_k (as defined in section 3.2.2, for example).
-

assumption of a negative exponential loss function. In essence, the risk function approximation will be defined as a 1-dimensional function of the stock quantity at every node, i.e. $\widehat{J}_k : \mathcal{U} \times \mathcal{Y} \rightarrow [0, \infty)$. This is possible because by section 2.4 the optimal risk function does not depend on the cash account c_k .

First suppose that we are given some risk function approximation $\widehat{J}_{k+1}(c_{k+1}, u_{k+1}, Y_k)$ at step $k + 1$ defined $\forall u_{k+1} \in \mathcal{U}$ and for $Y_{k+1}^j \in \mathcal{G}_{k+1}^N, j = 1, \dots, N$, i.e. simulated market information vectors defining the mesh points. For a general state $X_k = (c_k, u_k, Y_k)$, we approximate $Q_k(X_k, v)$ using the stochastic mesh as

$$\widehat{Q}_k(X_k, v) := \frac{1}{N} \sum_{j=1}^N w_{k,j} e^{-\gamma(V_{k+1}(\psi(X_k, v, Y_{k+1}^j)) - V_k)} \widehat{J}_{k+1}(v, Y_{k+1}^j), \quad (3.7)$$

for given weight functions $w_{k,j} = w_k(Y_k, Y_{k+1}^j), j = 1, \dots, N$, taking as input consecutive pairs of market states. These will be discussed in more detail in section 3.2.3.

The associated optimal risk function approximation at step k is then defined as

$$\widehat{J}_k(u_k, Y_k) := \inf_{v \in \mathcal{U}} \widehat{Q}_k(X_k, v). \quad (3.8)$$

This is a *sample average approximation* (SAA) (see for example Shapiro [124] and the references therein) of the “exact” version of the one-step problem (2.12) of chapter 2.

When computing $\widehat{J}_k(u_k, Y_k^i)$ for a market state vector Y_k^i on the stochastic mesh itself, we are considering transitions to other market states Y_{k+1}^j ($j = 1, \dots, N$), so we label the corresponding weights $w_{k,i,j}$ and write

$$\widehat{Q}_k(c_k, u_k, Y_k^i, v) := \frac{1}{N} \sum_{j=1}^N w_{k,i,j} e^{-\gamma(V_{k+1}(\psi(c_k, u_k, Y_k^i, v, Y_{k+1}^j)) - V_k)} \widehat{J}_{k+1}(v, Y_{k+1}^j), \quad (3.9)$$

and

$$\widehat{J}_k(u_k, Y_k^i) := \inf_{v \in \mathcal{U}} \widehat{Q}_k(c_k, u_k, Y_k^i, v). \quad (3.10)$$

3.2.3 Choice of mesh density

We now turn to the question of how the mesh states Y_k^i and associated weights $w_{k,i,j}$ can be determined. Of course, this will affect the statistical properties of the estimators $\widehat{Q}_k(X_k, v)$ and $\widehat{J}_k(X_k)$. We outline here some possibilities which have already been proposed in the literature. New proposals for mesh constructions will be given in chapter 6.

A useful requirement for the weights is that the weighted sum in (3.4) approximates the expectation (conditional on information at step k) in an unbiased way. This can be guaranteed using likelihood ratio (LR) weights, under some assumptions. For example, suppose that the vectors in the grid \mathcal{G}_{k+1}^N were sampled such that we know their unconditional density, which we denote $g_{k+1}(Y_{k+1})$, and that we also know the density of any $Y_{k+1}^j \in \mathcal{G}_{k+1}^N$ conditional on any $Y_k^i \in \mathcal{G}_k^N$, which we

denote $f_k(Y_k^i, Y_{k+1}^j)$. The LR weights are then defined as

$$w_{k,i,j} = \frac{f_k(Y_k^i, Y_{k+1}^j)}{g_{k+1}(Y_{k+1}^j)}, \text{ for } i \neq j \quad (3.11)$$

and $w_{k,i,j} = 1$ for $i = j$.

However, Broadie and Glasserman [36] point out that using LR weights over many steps K may lead the variance of the mesh estimates to grow exponentially with K , by analyzing an example where a mesh is used to price a standard European option. A better alternative, which avoids this phenomenon, is to use so-called *average density* (AD) weights, defined by

$$w_{k,i,j} = \frac{f_k(Y_k^i, Y_{k+1}^j)}{\frac{1}{N} \sum_{l=1}^N f_k(Y_k^l, Y_{k+1}^j)}. \quad (3.12)$$

The average density in the denominator can be interpreted as the density for the set of mesh points \mathcal{G}_{k+1}^N at step $k+1$ obtained by first drawing uniformly (i.e., with equal probability $1/N$) a point from the set \mathcal{G}_k^N and then generating a transition using the conditional density $f_k(Y_k^l, Y_{k+1}^j)$ for $l = 1, \dots, N$.

Liu and Hong [97] obtain both LR weights and AD weights through a conditioning procedure. More specifically, they start by rewriting the mesh based conditional expectation given by equation (3.4) as a limit

$$\mathbb{E}_k[\Phi_{k+1}(Y_{k+1})|Y_k = y] = \lim_{\epsilon \rightarrow 0} \frac{\mathbb{E}[\Phi_{k+1}(Y_{k+1})1_{y-\epsilon \leq Y_k \leq y+\epsilon}]}{\mathbb{E}[1_{y-\epsilon \leq Y_k \leq y+\epsilon}]}. \quad (3.13)$$

They then introduce some conditioning with respect to Y_{k+1} , from which they obtain the weights

$$w_k(y, Y_{k+1}) = \lim_{\epsilon \rightarrow 0} \frac{\mathbb{E}[1_{y-\epsilon \leq Y_k \leq y+\epsilon}|Y_{k+1}]}{\mathbb{E}[1_{y-\epsilon \leq Y_k \leq y+\epsilon}]}. \quad (3.14)$$

Using a similar idea, but conditioning on both Y_{k-1} and Y_{k+1} , they obtain another

expression for weights, which they call *binocular weights*. These new weights can be computed in terms of a conditional density obtained by a bridge sampling technique. The authors also provide numerical evidence that in the context of estimating American option prices, the binocular weights lead to smaller bias whereas the average density weights lead to smaller variance.

Note that if the transition densities $f_k(Y_k, Y_{k+1})$ are unknown, then we cannot use either the LR weights or the AD weights. To overcome the need for a transition density, Broadie, Glasserman and Ha [37] suggest to choose weights so that they minimize a measure of dispersion, given a set of linear constraints. When considered for only one step, this is called *weighted Monte Carlo* (see for example Avellaneda and Jäckel [6] and Glasserman and Yu [60]). As constraints, one could require that some low-order moments of the stock price distribution match their theoretical value, for example, which should help reduce bias in the optimization steps. However, this does not guarantee that the stochastic mesh estimator will be unbiased for every function $\Phi_k(Y_k)$ in equation (3.4).

Glasserman [59] (section 8.6.2) shows that approximating the optimal risk function via linear regression on a set of basis functions amounts to using mesh weights with a specific form (dependent on the choice of basis functions). However, the resulting mesh based estimates may also be biased.

3.2.4 Using simulated derivative values

An important difficulty when trying to apply stochastic mesh methods to the hedging problem is that the derivative price may not be known as an analytic function of market variables such as the underlying price or volatility level. In some cases, such as for the stochastic volatility model of section 4.1, option prices can only be obtained through numerical integration or simulation.

The stochastic mesh itself can be reused to obtain an unbiased estimate of the

derivative price, instead of simulating new independent paths at each state. For example, we can estimate the derivative price $h_k^i = h_k(Y_k^i)$ for the market state Y_k^i as

$$\widehat{h}_k^i = \frac{1}{N} \sum_{j=1}^N w_{k,i,j} \widehat{h}_{k+1}^j. \quad (3.15)$$

But in any case, estimating \widehat{h}_k^i by simulation adds noise to the estimator $\widehat{Q}_k(X_k, v)$ from equations (3.7) and (3.9), which is used to estimate optimal risk function values $\widehat{J}_k(X_k)$ in equation (3.8). More precisely, assuming the error $Z_{k+1} := \widehat{h}_{k+1}(Y_{k+1}) - h_{k+1}(Y_{k+1})$ from the derivative value h_{k+1} has an expected value of 0 and is independent of the market vector Y_{k+1} , the estimate $\widehat{Q}_k(X_k, v)$ of $Q_k(X_k, v)$ is *upward* biased since we have

$$\mathbf{E}_k \left[e^{-\gamma(v\Delta s_{k+1} + \Delta \widehat{h}_{k+1})} J_{k+1}(X_{k+1}) \right] = \mathbf{E}_k [e^{-\gamma Z_{k+1}}] \mathbf{E}_k \left[e^{-\gamma(v\Delta s_{k+1} + \Delta h_{k+1})} J_{k+1}(X_{k+1}) \right] \quad (3.16)$$

for any cost function $J_{k+1}(X_{k+1})$, with $X_{k+1} = \psi(X_k, v, Y_{k+1})$ and with $\mathbf{E}_k[e^{-\gamma Z_{k+1}}] \geq 1$. To see this last inequality, note that the function $e^{-\gamma x}$ is convex in x , so by Jensen's inequality we have $\mathbf{E}_k[e^{-\gamma Z_{k+1}}] \geq e^{-\gamma \mathbf{E}_k[Z_{k+1}]}$. Furthermore, by assumption $\mathbf{E}_k[Z_{k+1}] = 0$, so the right-hand side of the inequality equals 1. The resulting bias can be interpreted intuitively in the following way : evaluating the derivative on the mesh adds portfolio noise which is independent from the stock price variation Δs_{k+1} , and thus cannot be hedged by trading the underlying stock, which increases the portfolio risk.

In chapter 6, we will see that some variance reduction techniques (such as control variates) allow to reduce significantly the error Z_{k+1} and the associated bias of $\widehat{Q}_k(X_k)$.

3.3 Approximation of the risk function dependence on the stock quantity

Because we focus on the negative exponential loss function, the cash account variable c_k does not affect the risk function value $J_k^*(X_k)$ (see section 2.1.3), so it is omitted in the following discussion. However, in the case where costs are non-zero, the optimal risk function J_k^* may depend on u_k (see section 2.4). So for a given Y_k , we need to keep track of J_k^* as a function of u_k and not just a single point. We will start by looking at a specific approximation that can be used in this case and will then give a bound on the approximation error.

3.3.1 A linear-quadratic approximation

In our numerical experiments, we use the following approximation for the optimal risk function J_k^* , based on proposition 2.4.1. The general idea is to estimate the values of the two points b_k^- and b_k^+ delimiting the no-transaction boundary, and then approximate the risk function piecewise on the three sub-intervals $(-\infty, b_k^-]$, $[b_k^-, b_k^+]$ and $[b_k^+, +\infty)$.

Approximation 3.3.1 *Linear-quadratic approximation $\ln \widehat{J}_k(\cdot, Y_k)$ for $\ln J_k^*(\cdot, Y_k)$, assuming the market vector Y_k is fixed.*

1. Compute approximate values \widehat{b}_k^+ and \widehat{b}_k^- for b_k^+ and b_k^- using equations (2.22), (2.23) and (3.4).
2. For a given tolerance δ , if $|\widehat{b}_k^+ - \widehat{b}_k^-| < \delta$, then we assume that the boundary values are equal (i.e., $b_k^- = b_k^+$) and that the risk is a constant as a function of u_k , approximated as $\widehat{J}_k(u_k, Y_k) = \frac{1}{2}(\widehat{Q}_k^+(X_k, \widehat{b}_k^+) + \widehat{Q}_k^-(X_k, \widehat{b}_k^-))$.
3. Otherwise, $|\widehat{b}_k^+ - \widehat{b}_k^-| \geq \delta$, and we solve for η_1 and η_2 in equations (2.46) and (2.47), via $\widehat{\eta}_1 = \ln \widehat{J}_k(\widehat{b}_k^-, Y_k) - \beta_1 \widehat{b}_k^-$ and $\widehat{\eta}_2 = \ln \widehat{J}_k(\widehat{b}_k^+, Y_k) - \beta_2 \widehat{b}_k^+$.

4. Let $\widehat{\eta}_3 = \ln \widehat{J}_k(\widehat{u}_k^0, Y_k)$, for $\widehat{u}_k^0 := \frac{1}{2}(\widehat{b}_k^+ + \widehat{b}_k^-)$.
5. Approximate the (strictly convex) function $\alpha_k(Y_k, \cdot)$ in equation (2.48) by the quadratic approximation $g_k(u_k) = Au_k^2 + Bu_k + C_k$, where the parameters A, B, C are determined algebraically so the function goes through the three points $(\widehat{b}_k^-, \widehat{\eta}_1)$, $(\widehat{b}_k^+, \widehat{\eta}_2)$ and $(\widehat{u}_k^0, \widehat{\eta}_3)$.

Note that the linear part (step 3) of the above approximation is *exact*, in the sense that it corresponds to equations (2.46) and (2.47) in Proposition 2.4.1. But of course, some error can be introduced by the approximation of the parameters η_1 and η_2 . As for the quadratic part in step 5, for $u_k \in [b_k^-, b_k^+]$, it is indeed an approximation. It is justified because proposition 2.4.1 shows that this part of the function is known to be (strictly) convex.

Overall, this construction ensures the strict convexity of the risk function approximation, while only requiring simple algebra to estimate and evaluate. Thus it avoids potential difficulties when using non-convex approximations (for example, a general polynomial function) in the optimization steps. It is possible to have more sophisticated approximations, such as cubic splines with constraints guaranteeing convexity. But this comes at the cost of 1) a larger parameter estimation time (since the parameters must be optimized in the cubic spline approach) and 2) a larger function evaluation time, which is a problem since the approximation must be evaluated repeatedly. For more about convex spline approximations, see for example Russell [116], or the classic text by De Boor [48] for splines in general.

Figure 3.1 shows a function $\widehat{Q}_0(X_0, \cdot)$ resulting from applying the linear-quadratic approximation to estimate the function $Q_k(X_k, \cdot)$ at step $k = 0$, over $K = 8$ steps. For comparison, the sample average estimates of the expected one-step risk $E_0[g_1(X_0, v, Y_1)]$ and the expected risk $E_0[\widehat{J}_1(\psi(X_0, v, Y_1))]$ are shown, as well as their product $E_0[g_1(X_0, v, Y_1)]E_0[\widehat{J}_1(\psi(X_0, v, Y_1))]$. The two expectations $E_0[g_1(X_0, v, Y_1)]$ and $E_0[\widehat{J}_1(\psi(X_0, v, Y_1))]$ can be loosely interpreted as the building blocks of $Q_0(X_0, v)$,

based on the definition from equation (2.44). But of course their product may not necessarily equal to $Q_0(X_0, v)$, as can be seen in the figure, since the random variables $\widehat{J}_1(\psi(X_0, v, Y_1))$ and $g_1(X_0, v, Y_1)$ may be correlated. All of these estimates are given as functions of v , and their convexity in v is apparent. The error bars represent one standard deviation of the sample average.

3.3.2 A bound on the approximation error

There are different possibilities to measure the error of the approximation \widehat{J}_k compared to the optimal risk function J_k^* . Proposition 3.3.2 below gives a basic result involving the sup norm, where the aim is only to bound the error coming from the “deterministic” part of the risk function (involving the decision variable u), while assuming that the “stochastic” part (involving expectations of functions of Y_k) can be computed exactly.

At step k of the DP backward recursion (i.e., after going from $k = K - 1$ to $l = k$), we have defined a sequence of approximate risk functions $(\widehat{J}_{K-1}, \dots, \widehat{J}_k)$. The following proposition gives a bound on the error from calculating $\widehat{J}_k(X_k)$ over the different possible values $u_k \in \mathcal{U}$ of the stock quantity in the state $X_k = (u_k, Y_k)$.

Proposition 3.3.2 *For any Y_k ,*

$$\sup_{u_k \in \mathcal{U}} |\widehat{J}_k(u_k, Y_k) - J_k^*(u_k, Y_k)| \leq \epsilon_k(Y_k), \quad (3.17)$$

where

$$\epsilon_k(Y_k) = \mathbf{E}_k[\epsilon_{k+1}(Y_{k+1})] + \sup_{u_k \in \mathcal{U}} \left| \widehat{J}_k(u_k, Y_k) - \inf_{v \in \mathcal{U}} \mathbf{E}_k \left[\widehat{J}_{k+1}(\psi_k((u_k, Y_k), v, Y_{k+1})) \right] \right|. \quad (3.18)$$

Proof: We prove the result by induction on k . For $k = K - 1$, the result is trivial, since by definition $\widehat{J}_{k+1}(X_{k+1}) = J_K^*(X_K) = L(V_K(X_K) - V_0)$. Note that we assume

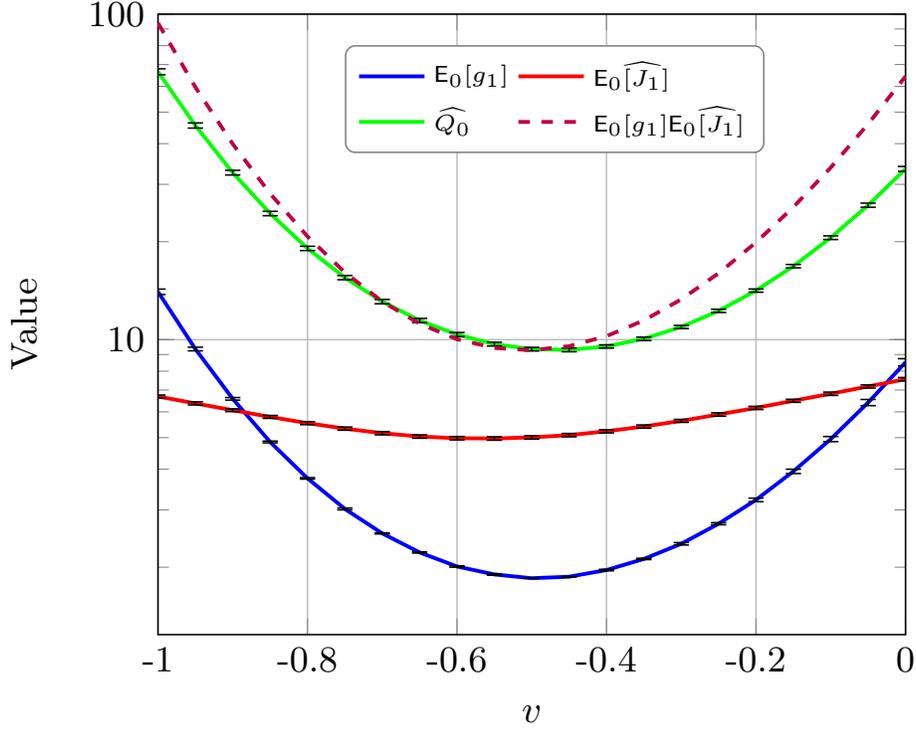


Figure 3.1: Estimates of the expected one step risk $\mathbf{E}_0[g_1(X_0, v, Y_1)]$, expected risk function approximation $\mathbf{E}_0[\widehat{J}_1(\psi(X_0, v, Y_1))]$ and Q -function approximation $\widehat{Q}_0(X_0, v)$ as a function of the trading decision v , for a call option under the GBM model ($s_0 = 10$, $\sigma = 40\%$, $b = 2\%$, $\gamma = 5$ and $K = 8$) and a stochastic mesh with $N = 1024$ paths. The product $\mathbf{E}_0[g_1(X_0, v, Y_1)]\mathbf{E}_0[\widehat{J}_1(\psi(X_0, v, Y_1))]$ is also indicated in comparison to $\widehat{Q}_0(X_0, v)$.

that Y_k is fixed, so we will write below $X_k(u) := (u, Y_k)$ when we need to specify the u dependence of the state X_k .

Now assume the result is true for $k + 1$. To prove the result for k , we will use the following two basic inequalities

$$\sup_{u \in \mathcal{U}} (f(u) + g(u)) \leq \sup_{u \in \mathcal{U}} f(u) + \sup_{u \in \mathcal{U}} g(u) \quad (3.19)$$

$$\inf_{u \in \mathcal{U}} (f(u) - g(u)) \geq \inf_{u \in \mathcal{U}} f(u) - \inf_{u \in \mathcal{U}} g(u), \quad (3.20)$$

for f and g any two bounded functions and $\mathcal{U} \subseteq \mathbb{R}$. First, by (3.19), we have

$$\begin{aligned} \sup_{u \in \mathcal{U}} |\widehat{J}_k(X_k(u)) - J_k^*(X_k(u))| &\leq \sup_{u \in \mathcal{U}} \left| \widehat{J}_k(X_k(u)) - \inf_{v \in \mathcal{U}} \mathbf{E}_k[\widehat{J}_{k+1}(\psi_k(X_k(u), v, Y_{k+1}))] \right| \\ &+ \sup_{u \in \mathcal{U}} \left| J_k^*(X_k(u)) - \inf_{v \in \mathcal{U}} \mathbf{E}_k[\widehat{J}_{k+1}(\psi_k(X_k(u), v, Y_{k+1}))] \right|. \end{aligned}$$

So the result will follow if we can show that

$$\sup_{u \in \mathcal{U}} \left| J_k^*(X_k(u)) - \inf_{v \in \mathcal{U}} \mathbf{E}_k[\widehat{J}_{k+1}(\psi_k(X_k(u), v, Y_{k+1}))] \right| \leq \mathbf{E}_k[\epsilon_{k+1}(Y_{k+1})].$$

To obtain this inequality, note that by using the induction hypothesis and the expectation operator,

$$\left| \mathbf{E}_k[\widehat{J}_{k+1}(\psi_k(X_k, v, Y_{k+1}))] - \mathbf{E}_k[J_{k+1}^*(\psi_k(X_k, v, Y_{k+1}))] \right| \leq \mathbf{E}_k[\epsilon_{k+1}(Y_{k+1})]. \quad (3.21)$$

Furthermore, by applying inequality (3.20), we have

$$\begin{aligned} &\inf_{v \in \mathcal{U}} \left| \mathbf{E}_k[\widehat{J}_{k+1}(\psi_k(X_k, v, Y_{k+1}))] \right| - \inf_{v \in \mathcal{U}} \left| \mathbf{E}_k[J_{k+1}^*(\psi_k(X_k, v, Y_{k+1}))] \right| \\ &\leq \inf_{v \in \mathcal{U}} \left| \mathbf{E}_k[\widehat{J}_{k+1}(\psi_k(X_k, v, Y_{k+1}))] - \mathbf{E}_k[J_{k+1}^*(\psi_k(X_k, v, Y_{k+1}))] \right| \end{aligned}$$

The result then follows by combining this with (3.21) and applying the sup operator

on both sides of the resulting inequality. \square

In practice, it may be quite hard to compute the bound $\epsilon_k(Y_k)$ on the right-hand side of the inequality (3.17). A first obstacle is the evaluation of the expectation operators $E_k[\cdot]$ in equation (3.18). This can be addressed by approximating the expectations using the mesh, as in equation (3.4). The results should then hold assuming that the number of paths N used to construct the mesh (or, more generally, the number of points N in each mesh grid \mathcal{G}_k^N) is high enough.

Perhaps more problematic is the computation of the supremum in equation (3.18), which is applied to a function which is costly to evaluate and may have many local maxima. As a rough proxy for the global supremum, one possibility is to measure the maximum error on a small subset of values $\{u_k^m \in \mathcal{U} : m = 1, \dots, M\}$. These ideas will be used later in section 3.4 and in numerical experiments, to help define confidence intervals for the values of the optimal risk function.

Figure 3.2 shows the behavior of an estimate of the approximation error bound from equation (3.17) as a function of the number of mesh nodes N , for various number M of values for the discretization of u_k . The estimates were computed using $n_R = 1000$ replications of a stochastic mesh for the problem of hedging a call option under the GBM model with parameters $s_0 = 10$, $\sigma = 20\%$, $T = 6$ months and $K = 8$. Overall, the error bound estimates decrease with N , which is expected as the amount of noise is reduced, although it would only go to 0 if the approximate risk function \hat{J}_k were actually the optimal risk function J_k^* . Increasing M also increases the error bound estimates $\hat{\epsilon}_0$, as it should, but this effect becomes marginally smaller as M increases.

Figure 3.3 shows the effect of increasing the number of time steps K on the error bound estimate $\hat{\epsilon}_0$ for the same setup as above, but with the number of mesh nodes set at $N = 256$. As expected, $\hat{\epsilon}_0$ grows with K , and this growth appears to be exponential, as the relationship is roughly linear on a log-log scale. For the sake of

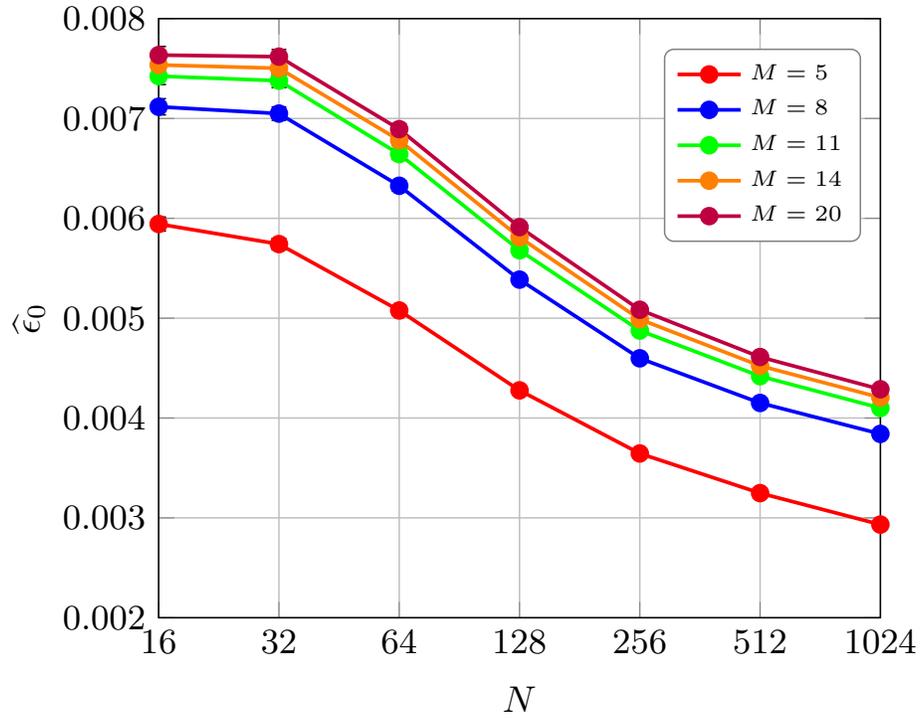


Figure 3.2: Approximation error bound estimate $\hat{\epsilon}_0$ as a function of N , for a call option under the GBM model with $s_0 = 10, \sigma_0 = 0.2, T = 6$ months, $K = 8, \gamma = 1$ and $b = 0.2\%$.

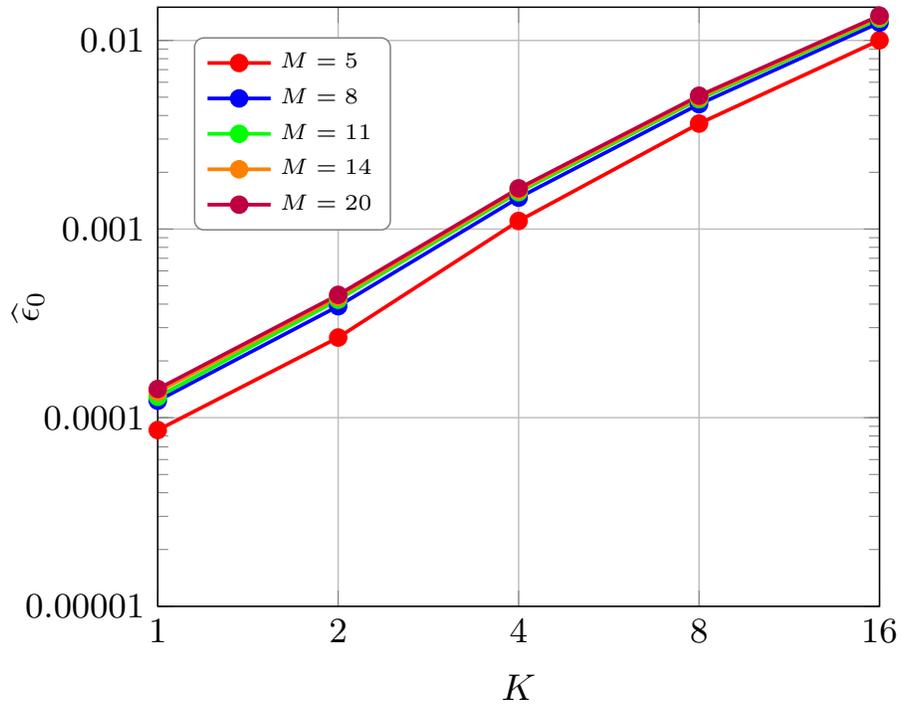


Figure 3.3: Approximation error bound estimate $\hat{\epsilon}_0$ as a function of K , for a call option under the GBM model with $s_0 = 10$, $\sigma_0 = 0.2$, $T = 6$ months, $N = 256$, $\gamma = 1$ and $b = 0.2\%$.

illustration, fitting the model $\widehat{\epsilon}_0 = \alpha K^\beta$ algebraically using the values for $K = 1$ and $K = 16$ with $M = 20$, one finds $\alpha = 0.000142$ and $\beta = 1.64$.

3.4 Confidence interval for the optimal value

We now recall how one can use a stochastic mesh to obtain a confidence interval for the optimal risk function value $J_0^*(X_0)$ via low and high biased estimators. This was originally pointed out by Broadie and Glasserman [34] for the American option pricing problem. Here however, the construction of the low and high biased estimators are inverted, due to the fact that in the hedging problem we seek to minimize an expected loss and not maximize an expected value.

First, note that any policy π is *suboptimal*, meaning that $J_0^\pi(X_0) \geq J_0^*(X_0)$. So computing a standard (unbiased) Monte Carlo estimate $\widehat{J}_0^\pi(X_0)$ of $J_0^\pi(X_0)$ will lead to a high biased estimator $\widehat{J}_0^H(X_0)$ of $J_0^*(X_0)$. That is, simulate N independent paths $\{(Y_k^i : k = 0, \dots, K) : i = 1, \dots, N\}$ and apply the policy π to compute the terminal portfolio value $V_K^{\pi, i}$ along each path, to obtain the unbiased estimator of $J_0^\pi(X_0) = \mathbf{E}_0[L(V_K^\pi - V_0)]$ given by

$$\widehat{J}_0^H(X_0) := \frac{1}{N} \sum_{i=1}^N L(V_K^{\pi, i} - V_0). \quad (3.22)$$

Broadie and Glasserman [34] refer to this as the *path estimator*.

Furthermore, as we saw earlier in section 3.2.2, the stochastic mesh method applied to our hedging problem yields an estimator of the optimal risk function via equations (3.9) and (3.10). This estimator is called the *mesh estimator* by Broadie and Glasserman [34]), and is a *low biased* estimator of the optimal risk function :

Proposition 3.4.1 *The mesh estimator $\widehat{J}_k(X_k)$ is a low biased estimator of $J_k^*(X_k)$, $\forall k = 0, \dots, K$ and $\forall X_k \in \mathcal{X}$.*

Proof : For $k = K$ and any $X_K \in \mathcal{X}$, the result is trivial since both $\widehat{J}_K(X_K)$ and $J_K^*(X_K)$ are simply equal to the loss function $L(V(X_K) - V(X_0))$. To see the low bias for a given step $k < K$, assume that the estimator is low biased for step $k + 1$, i.e.

$$\mathbf{E}_{k+1} \left[\widehat{J}_{k+1}(X_{k+1}) \right] \leq J_{k+1}^*(X_{k+1}), \forall X_{k+1} \in \mathcal{X}. \quad (3.23)$$

Consider now the following obvious inequality

$$\mathbf{E}_k \left[\inf_{v \in \mathcal{U}} \widehat{Q}_k(X_k, v) \right] \leq \mathbf{E}_k \left[\widehat{Q}_k(X_k, v) \right], \forall v \in \mathcal{U}, \quad (3.24)$$

and then minimize the right-hand side with respect to v , which yields

$$\mathbf{E}_k \left[\widehat{J}_k(X_k) \right] = \mathbf{E}_k \left[\inf_{v \in \mathcal{U}} \widehat{Q}_k(X_k, v) \right] \leq \inf_{v \in \mathcal{U}} \mathbf{E}_k \left[\widehat{Q}_k(X_k, v) \right], \quad (3.25)$$

where the equality on the left-hand side comes from the definition of $\widehat{J}_k(X_k)$. We will show that the right-hand side of inequality (3.25) is bounded above by $J_k^*(X_k)$, $\forall X_k \in \mathcal{X}$, which will complete the proof.

Using the definition of the function \widehat{Q}_k from equation (3.9) and applying the expectation and minimization operators, we see that

$$\inf_{v \in \mathcal{U}} \mathbf{E}_k \left[\widehat{Q}_k(X_k, v) \right] = \inf_{v \in \mathcal{U}} \mathbf{E}_k \left[\frac{1}{N} \sum_{j=1}^N w_{k,j} e^{-\gamma(V_{k+1}(\psi(X_k, v, Y_{k+1}^j)) - V_k)} \widehat{J}_{k+1}(v, Y_{k+1}^j) \right].$$

But now conditioning on Y_{k+1} and then using the hypothesis from equation (3.23) to the right-hand-side, we obtain

$$\begin{aligned} \inf_{v \in \mathcal{U}} \mathbf{E}_k \left[\widehat{Q}_k(X_k, v) \right] &= \inf_{v \in \mathcal{U}} \mathbf{E}_k \left[\frac{1}{N} \sum_{j=1}^N w_{k,j} e^{-\gamma(V_{k+1}(\psi(X_k, v, Y_{k+1}^j)) - V_k)} \mathbf{E}_{k+1} \left[\widehat{J}_{k+1}(v, Y_{k+1}^j) \right] \right] \\ &\leq \inf_{v \in \mathcal{U}} \mathbf{E}_k \left[\frac{1}{N} \sum_{j=1}^N w_{k,j} e^{-\gamma(V_{k+1}(\psi(X_k, v, Y_{k+1}^j)) - V_k)} J_{k+1}^*(v, Y_{k+1}^j) \right]. \end{aligned}$$

The random variable inside this last expectation is the (unbiased) mesh estimator of $Q_k(X_k, v)$, so we have

$$\inf_{v \in \mathcal{U}} \mathbb{E}_k \left[\widehat{Q}_k(X_k, v) \right] \leq \inf_{v \in \mathcal{U}} Q_k(X_k, v) \quad (3.26)$$

$$= J_k^*(X_k), \quad (3.27)$$

as required. \square

By applying the DP algorithm recursively on the approximation functions \widehat{J}_k , it follows that the mesh estimator $\widehat{J}_k(X_k)$ is low biased for every stage $k = 0, 1, \dots, K - 1$. However, in the case of the hedging problem, there is an additional source of error introduced by the risk function approximations $\widehat{J}_{k+1}(v, Y_{k+1}^i)$ used in equation (3.7). As we saw in section 3.3.2, this error can be bounded above by some ϵ_k , but this bound typically cannot be computed exactly.

Here, we define an approximation $\widehat{\epsilon}_k(Y_k^i)$ of $\epsilon_k(Y_k^i)$ at each node Y_k^i of the mesh by computing

$$\widehat{\epsilon}_k(Y_k^i) = \frac{1}{N} \sum_{j=1}^N w_{k,i,j} \widehat{\epsilon}_{k+1}(Y_{k+1}^j) + \max_{u^m \in \mathcal{U}, m=1, \dots, M} \left| \widehat{J}_k(u^m, Y_k^i) - \min_{v \in \mathcal{U}} \widehat{Q}_k(u^m, Y_k^i, v) \right|. \quad (3.28)$$

The first term may be an unbiased estimate of $\mathbb{E}_k[\epsilon_{k+1}(Y_{k+1})]$, depending on the choice of weights (see section 3.2.3). The second term will introduce some bias, but this should decrease as the number M of values for u is increased. So for N (number of paths) and M high enough, we can regain an (approximately) low biased estimator $\widehat{J}_k^L(X_k)$ for J_k^* by subtracting $\widehat{\epsilon}_k$ from the risk function approximation, i.e., let

$$\widehat{J}_k^L(X_k) := \widehat{J}_k(X_k) - \widehat{\epsilon}_k(X_k). \quad (3.29)$$

Now suppose we have done a numerical experiment where we computed n_R inde-

pendent replications of the estimates $\widehat{J}_0^L(X_0)$ and $\widehat{J}_0^H(X_0)$ of $J_0^*(X_0)$, and label their sample averages as \overline{J}_0^L and \overline{J}_0^H , respectively. Assuming normality for the distributions of \widehat{J}_0^L and \widehat{J}_0^H , we can define a level $1 - \alpha$ confidence interval on $J_0^*(X_0)$ in the usual way as

$$\left[\overline{J}_0^L - z_{\alpha/2} \frac{s^L}{\sqrt{n_R}}, \overline{J}_0^H + z_{\alpha/2} \frac{s^H}{\sqrt{n_R}} \right], \quad (3.30)$$

where $z_{\alpha/2}$ is the $1 - \alpha/2$ quantile of the normal distribution and s^L and s^H are respectively the sample standard deviations of \widehat{J}_0^L and \widehat{J}_0^H .

3.5 Convergence of mesh estimators

In section 3.4, we discussed two possible (mesh and path) estimators of the optimal risk, one low and the other high biased. In the context of American option pricing, the high and low biased stochastic mesh estimators were shown by Broadie and Glasserman [35] to converge to the true option values. Avramidis and Matzinger [8] provide further analysis that yields an asymptotic upper bound on the probability of error of the mesh based estimate as a function of N , and show that it goes to 0 as $N \rightarrow \infty$.

In a different context, Rust [117] also showed the convergence of a stochastic mesh-type algorithm (the random Bellman operator method). He showed that it allowed to solve with polynomial computational complexity a particular class of Markov decision problems, which uses a d -dimensional continuous state space but a finite action set A . Explicitly, an upper bound on the worst-case complexity of the K -stage version of the problem is given by

$$\text{comp}^{\text{wor-ran}}(\epsilon, d) = O\left(\frac{Kd^4|A|^5}{\epsilon^4}\right), \quad (3.31)$$

where ϵ denotes the approximation error, and some additional constants are absorbed in the $O(\cdot)$ notation.

Here, we provide an informal argument applicable to our setting. Let $\widehat{m} = \frac{1}{N} \sum_{i=1}^N Z^i$, with the Z^i i.i.d. samples from some random variable Z , be a standard Monte Carlo estimator of the expectation $E[Z]$. First, taking the variance $\sigma_N^2(\widehat{m})$ shows that $\sigma_N^2(\widehat{m}) \in O(N^{-1})$. Thus the standard deviation $\sigma_N(\widehat{m})$ is in $O(N^{-1/2})$. Taking the approximation error ϵ to be given by the standard deviation, we then have also $\epsilon \in O(N^{-1/2})$. And since the computational cost τ is proportional to the number of paths N (again, for standard MC), we can write $\epsilon \in O(\tau^{-1/2})$. However, in the case of the stochastic mesh, computing the weights for each consecutive pairs of states implies work τ of the order $O(N^2)$. So assuming that the standard deviation of an estimator \widehat{m} computed on a stochastic mesh is the same as for standard MC, we should have that $\tau \in O(\epsilon^{-4})$.

The convergence of the high and low biased mesh estimators for the hedging problem is illustrated in section 5.3.

3.6 Conclusion

In this chapter, we showed how the existing stochastic mesh method for American option pricing could be applied to our hedging problem from chapter 2. This was done by associating a one dimensional function $\widehat{J}_k(\cdot, Y_k^i)$ at each mesh node, instead of a single value (i.e. the option's price), in order to account for the policy dependent part of the state (the stock quantity). We made this construction explicit by proposing a linear-quadratic approximation for the function $\ln \widehat{J}_k(\cdot, Y_k^i)$. Furthermore, we provided some bounds on the error coming from the deterministic part of the approximation, assuming that there was no error from the stochastic part, i.e. supposing that all expectations were known exactly. We then discussed how a confidence interval can be constructed for the optimal risk, using a low biased mesh estimator and a high biased path estimator, as is already known from the original Stochastic Mesh method. Here, however, the fact that we require another approximation to in-

clude the stock price dependence to the risk function could make the mesh estimator no longer low biased, so we propose to regain the low bias property by subtracting an estimate of the above mentioned error bound. Finally, we discussed briefly the convergence of the mesh estimators.

CHAPTER 4

PRICE DYNAMICS AND HEURISTIC POLICIES

In this chapter, we provide some details about two separate elements required to construct the simulation experiments presented later in this thesis and to interpret their results. Section 4.1 first defines a stochastic volatility model for the stock price dynamics. Simulating paths from the model defines mesh states and the corresponding probability density is used in computation of the mesh weights (see section 3.2.3). Section 4.2 follows by describing various heuristic policies from the literature, which will serve as benchmarks for the results for stochastic mesh based policies.

4.1 The exponential Ornstein-Uhlenbeck stochastic volatility model

This section describes a stochastic volatility model for the dynamics of stock prices and the details of a discrete time sampling procedure, as well as some motivation for these choices. We focus on the exponential Ornstein-Uhlenbeck (expOU) stochastic volatility model, a model where the log of the volatility of returns follows an Ornstein-Uhlenbeck (OU) process. This specific model has been studied for example by Sandmann and Koopman [118], Jacquier, Polson and Rossi [74], Fouque, Papanicolaou and Sircar [55], [56], Masoliver and Perelló [102]. Stochastic volatility models where the log of the volatility follows an OU process can be traced back at least to Scott [123]. The model is presented below in two forms: the first one is the model itself (the basic form), and the second one introduces two additional variables that are correlated to the price and volatility shocks at each step. All of these variables are assumed to be observable by the investor, so the second form represents a model where more information is available when making decisions.

4.1.1 Basic form

In continuous time, the model is defined by the following stochastic differential equations :

$$d \ln s(t) = \left(r - q - \frac{1}{2} \sigma^2(t) \right) dt + \sigma(t) dB_1(t), \quad (4.1)$$

$$d \ln \sigma(t) = \kappa (\ln \bar{\sigma} - \ln \sigma(t)) dt + \sigma_v dB_2(t). \quad (4.2)$$

Here, $s(t)$ is the price, $\sigma(t)$ is the (instantaneous) volatility of returns, r is the risk-free rate, q is the dividend yield, the $B_i(t)$, $i = 1, 2$ are Brownian motions with correlation ρ , $\ln \bar{\sigma}$ is the long term mean of the log-volatility, κ is a speed of adjustment parameter and σ_v , the volatility of log-volatility.

In discrete time, we can integrate exactly the process for the log volatility and obtain

$$\ln \sigma_{k+1} = e^{-\kappa \Delta t} \ln \sigma_k + \ln \bar{\sigma} (1 - e^{-\kappa \Delta t}) + \sigma_v \sqrt{\frac{1 - e^{-2\kappa \Delta t}}{2\kappa}} z_{2,k}, \quad (4.3)$$

where $z_{2,k}$ is a $N(0, 1)$ variate, and where $s_k = s(t_k)$, $\sigma_k := \sigma(t_k)$. There is no known exact formula for the transitions of the discrete time process (s_0, s_1, \dots, s_K) , so we use the Euler discretization for $\ln s(t)$, given by

$$\ln s_{k+1} = \ln s_k + \left(r - q - \frac{1}{2} \sigma_k^2 \right) \Delta t_{k+1} + \sigma_k \sqrt{\Delta t_{k+1}} z_{1,k}, \quad (4.4)$$

where $z_{1,k}$ is a $N(0, 1)$ variate, with correlation ρ with $z_{2,k}$. The market information vector Y_k at step k then becomes $Y_k = (s_k, \sigma_k)^T$.

Equations (4.3) and (4.4) are used throughout this thesis to generate paths of the underlying stock price. We set $r = q = 0$, so the only parameters of interest are $(\kappa, \sigma_v, \bar{\sigma}, \rho)$, as well as the initial process values $Y_0 = (s_0, \sigma_0)^t$.

The estimation of the parameters of the above stochastic volatility model is not a simple task and could be done in different ways. We estimate the parameters

$(\kappa, \sigma_v, \bar{\sigma}, \rho)$ through a moment-based procedure described in Masoliver and Perelló [102], section 4. We use as inputs the price series of the S&P 500 index from the 1950 to 2013 inclusively. The estimation results are given in table 4.I for the full 64 year period, as well as for some 32, 16 and 8 year periods, in order to provide a sense of the stability of the parameters through time. But also, the different sets of parameter values indicate what ranges of values are reasonable to use for numerical experiments. Note that two other quantities, $\mathbf{E}[\sigma]$ and β , are given in the rightmost columns of table 4.I. They are used in the estimation procedure and represent respectively the expectation of the stationary distribution of $\sigma(t)$ and the quantity $\beta := \sigma_v^2/(2\kappa)$.

Table 4.I: Moment-based estimates of the parameters of the expOU model for various historical periods.

Period	T (years)	κ	σ_v	$\bar{\sigma}$	ρ	$\mathbf{E}[\sigma]$	β
1950-2013	64	2.7	176 %	8.7 %	-45 %	15.5 %	0.581
1950-1981	32	7.8	191 %	9.6 %	-50 %	12.1 %	0.234
1982-2013	32	2.9	183 %	10.3 %	-51 %	18.3 %	0.578
1950-1965	16	12.2	292 %	7.8 %	-19 %	11.1 %	0.35
1966-1981	16	8.8	151 %	11.4 %	-56 %	13 %	0.13
1982-1997	16	3.5	240 %	6.7 %	-24 %	15.5 %	0.834
1998-2013	16	4.6	169 %	15.2 %	-83 %	20.7 %	0.308
1950-1957	8	9.9	245 %	8.8 %	-22 %	12 %	0.305
1958-1965	8	11.1	299 %	6.7 %	-60 %	10 %	0.404
1966-1973	8	19.7	244 %	9.9 %	-78 %	11.5 %	0.151
1974-1981	8	6.5	116 %	13 %	-39 %	14.4 %	0.103
1982-1989	8	3.4	240 %	7.7 %	-36 %	18 %	0.853
1990-1997	8	2.3	110 %	9.6 %	-31 %	12.5 %	0.26
1998-2005	8	10.5	175 %	16.4 %	-54 %	19 %	0.146
2006-2013	8	4.1	171 %	15.5 %	-82 %	22.2 %	0.357
Min		2.3	110 %	6.7 %	-83 %	10 %	0.103
Max		19.7	299 %	16.4 %	-19 %	22.2 %	0.853

In empirical works about stock price dynamics, it is more typical to see discrete time models, because the parameters are then easier to estimate (see for example Harvey and Shephard [66], Pastorello, Renault and Touzi [107]). On the other hand,

continuous time models tend to be preferred in theoretical works regarding option pricing, because they can yield formulas in closed form (e.g. the Heston model [70]), although this is not necessarily the case. Ultimately, the intended application determines what is an appropriate model. In our context, the model is useful because it allows for a more realistic description of stock price dynamics than a simple geometric Brownian motion. Yet it does not introduce additional complications when applying stochastic mesh methods, because it has a well-defined transition density (as discussed in section 3.2.3).

Indeed, the choice of modeling the log of the volatility instead of the volatility itself ensures that there are no issues with negative values when sampling from the process in discrete time. Here in fact, equation (4.3) yields values from the exact distribution of $\sigma(t)$ (with the association $\sigma(t_k) = \sigma_k$). Negative values could occur for example for the Heston model [70] when discretizing using a Euler scheme. Sampling schemes that are corrected to avoid negative values have been studied (see for example Lord, Koekkoek and van Dijk [99] and Andersen [5]). But the resulting approximations typically involve singular transition densities, which make the stochastic mesh method more complicated to use (see section 3.2.3). Note that other well-known classes of models such as the discrete time GARCH family also have singular transition densities.

It has been argued by many authors that stochastic volatility models alone do not explain stock price patterns as well without jumps (see for example Cont and Tankov [41] or Gatheral [58]). In the above model, we did not include a jump term because it would require more parameters to include in the estimation process, and bring up technical questions about the modeling and simulation of rare events (jumps) which would bring us too far away from the focus of this thesis. Nevertheless, stochastic volatility models without jumps are still relevant in practice, as they are appropriate to model stock index returns (see Christoffersen, Heston and Jacobs [38]).

4.1.2 An extension using additional information variables

Investors and traders are constantly looking to take into account available information to help them improve their investment decisions. If some new information variable q_k were linked to the price variations $s_{k+1} - s_k$ or the volatility variations $v_{k+1} - v_k$ (e.g., if they are correlated), it could perhaps be exploited in the optimization problem used to determine the hedging policy, in order to reduce risk further. For example, one could consider links between macro-economic variables and stock price and volatility (see for example Corradi, Distaso and Mele [42] or Engle, Ghysels and Sohn [54]). With this in mind, we introduce below an extension to the expOU model of section 4.1.1 that will allow us to test how including additional information can impact optimal trading decisions. Furthermore, this will provide us with a higher dimensional problem on which to test stochastic mesh methods.

Consider again the discrete model given by equations (4.3) and (4.4). The uncertainty at each step k comes from the two standard normal random variables $z_{1,k}$ and $z_{2,k}$ with correlation ρ . We introduce two additional random variables $q_{1,k}$ and $q_{2,k}$, assumed defined and observable for every step $k = 0, \dots, K - 1$, so that the market information vector becomes the 4-dimensional vector $Y_k^I = (s_k, \sigma_k, q_{1,k}, q_{2,k})^T$, instead of the 2-dimensional vector $Y_k = (s_k, \sigma_k)^T$. To relate $q_{1,k}$ and $q_{2,k}$ to the expOU model, we start by rewriting $z_{1,k}$ and $z_{2,k}$ as

$$z_{1,k} = W_{1,k} \tag{4.5}$$

$$z_{2,k} = \rho W_{1,k} + \sqrt{1 - \rho^2} W_{2,k}, \tag{4.6}$$

where $W_{1,k}$ and $W_{2,k}$ are independent $N(0, 1)$ variates. Assume now that $q_{1,k} \sim N(0, 1)$ and $q_{2,k} \sim N(0, 1)$ are independent and that we can decompose $W_{1,k}$ and

$W_{2,k}$ as

$$W_{1,k} = \alpha_1 q_{1,k} + \sqrt{1 - \alpha_1^2} r_{1,k} \quad (4.7)$$

$$W_{2,k} = \alpha_2 q_{1,k} + \sqrt{1 - \alpha_2^2} r_{2,k} \quad (4.8)$$

where $r_{1,k}, r_{2,k}$ are also $N(0, 1)$ variates, independent with each other and with $q_{1,k}$ and $q_{2,k}$. The constants $\alpha_1, \alpha_2 \in [0, 1]$ represent the loadings for the factors $q_{1,k}$ and $q_{2,k}$, for all $k = 0, \dots, K - 1$. From equations (4.7) and (4.8), we see that $W_{1,k}$ and $W_{2,k}$ are linear combinations of normal variables, and thus are also normally distributed. Furthermore, it is easy to check that $W_{1,k}$ and $W_{2,k}$ have expectation 0 and variance 1 (using the independence between $q_{i,k}$ and $r_{i,k}$, for $i = 1, 2$), so the $W_{1,k}, W_{2,k}$ are indeed independent $N(0, 1)$ variates, $\forall k$.

Let $X_k^I = (c_k, u_k, Y_k^I) \in \mathcal{X}^I$ for

$$\mathcal{X}^I := \{(c_k, u_k, Y_k^I) : c_k \in \mathbb{R}, u_k \in \mathcal{U}, Y_k^I \in \mathcal{Y} \times \mathbb{R}^2\}, \quad (4.9)$$

the extended state space, and denote the corresponding optimal risk function $J_k^{I,*} : \mathcal{X}^I \rightarrow [0, \infty)$. The following proposition shows that using the extended market information vector Y_k^I instead of Y_k will not to increase the expected risk.

Proposition 4.1.1 *For all $X_k^I \in \mathcal{X}^I$, $k = 0, \dots, K - 1$, we have*

$$\mathbb{E}_k \left[J_k^{I,*}(X_k^I) \mid Y_k \right] \leq J_k^*(X_k). \quad (4.10)$$

Proof: Assume that the result is true for $k + 1$, i.e.,

$$\mathbb{E}_{k+1} \left[J_{k+1}^{I,*}(X_{k+1}^I) \mid Y_{k+1} \right] \leq J_{k+1}^*(X_{k+1}), \quad (4.11)$$

where

$$X_{k+1}^I = \psi(X_k^I, v, Y_{k+1}^I). \quad (4.12)$$

We start by noting that by definition, we have

$$J_k^{I,*}(X_k^I) := \min_{v \in \mathcal{U}} \mathbf{E}_k \left[J_{k+1}^{I,*}(X_{k+1}^I) \mid Y_k^I \right]. \quad (4.13)$$

Hence, taking expectations conditional on Y_k on both sides of equation (4.13),

$$\mathbf{E}_k \left[J_k^{I,*}(X_k^I) \mid Y_k \right] = \mathbf{E}_k \left[\min_{v \in \mathcal{U}} \mathbf{E}_k \left[J_{k+1}^{I,*}(X_{k+1}^I) \mid Y_{k+1}^I \right] \mid Y_k \right]. \quad (4.14)$$

Now, it is clear that

$$\min_{v \in \mathcal{U}} \mathbf{E}_k \left[J_{k+1}^{I,*}(X_{k+1}^I) \mid Y_k^I \right] \leq \mathbf{E}_k \left[J_{k+1}^{I,*}(X_{k+1}^I) \mid Y_k^I \right], \quad (4.15)$$

so taking expectations conditional on Y_k on both sides and minimizing the right-hand side with respect to v yields

$$\mathbf{E}_k \left[\min_{v \in \mathcal{U}} \mathbf{E}_k \left[J_{k+1}^{I,*}(X_{k+1}^I) \mid Y_k^I \right] \mid Y_k \right] \leq \min_{v \in \mathcal{U}} \mathbf{E}_k \left[J_{k+1}^{I,*}(X_{k+1}^I) \mid Y_k \right], \quad (4.16)$$

Then, combining equation (4.14) and inequality (4.16), we obtain

$$\mathbf{E}_k \left[J_k^I(X_k^{I,*}) \mid Y_k \right] \leq \min_{v \in \mathcal{U}} \mathbf{E}_k \left[J_{k+1}^{I,*}(X_{k+1}^I) \mid Y_k \right]. \quad (4.17)$$

But using the induction hypothesis from equation (4.11) and taking expectations conditional on Y_k , we see that

$$\mathbf{E}_k \left[J_{k+1}^{I,*}(X_{k+1}^I) \mid Y_k \right] \leq \mathbf{E}_k \left[J_{k+1}^*(X_{k+1}) \mid Y_k \right]. \quad (4.18)$$

So equation (4.17) becomes

$$\mathbf{E}_k \left[J_k^I(X_k^{I,*}) \mid Y_k \right] \leq \min_{v \in \mathcal{U}} \mathbf{E}_k \left[J_{k+1}^*(X_{k+1}) \mid Y_k \right]. \quad (4.19)$$

Since the right-hand side equals $J_k^*(X_k)$ by definition, this proves the result. \square .

4.2 Some heuristic policies

In practice, hedging policies are not necessarily based on some optimality criterion. For example, option traders usually hedge their portfolio by using some in-house modification of the Black-Scholes-Merton (BSM) delta hedging policy, such as using the model with an implied volatility parameter that varies as a function of the stock price or hedging more or less frequently when costs are higher (see for example Haugh and Taleb [67]). Such policies are known as *heuristics*. That is, policies that are not necessarily optimal, yet that can often provide acceptable results in practice and are relatively fast to compute.

We already described BSM delta hedging in section 1.2. We will consider two other analytical policies which takes costs into account: the Whalley-Wilmott asymptotic approximation [136] and the parametric approximation by Zakamouline [138]. These policies were developed in the (one-dimensional) geometric Brownian motion setting in continuous time, so it is not clear that they should always perform well under different conditions, such as for the two-dimensional discrete time model described later in section 4.1. With this in mind, we also consider *local hedging*, which can take more general price dynamics into account, yet requires a simpler optimization procedure than full dynamic programming. This is in fact an application of the one-step lookahead heuristic. See Bertsekas [15], sections 6.3 and 6.4, for general background on one-step lookahead policies. For an application to hedging, see Primbs [111], where the hedging problem under costs is formulated as a quadratic

regulator problem with constraints.

4.2.1 Whalley-Wilmott hedging policy

Whalley and Wilmott [135], [136] do an asymptotic analysis of the hedging problem as defined in Hodges and Neuberger [71] and Davis, Panas and Zariphopoulou [47]. This corresponds to our setting, including costs and using the exponential loss function, but in continuous time and with a constant (non-stochastic) volatility. In all of the below formulas, we use the indices k to denote the discrete time steps t_k used in our formulation of the hedging problem, but note that the original expressions are valid in continuous time.

Using formal series expansions, they derive a simple approximate expression for the hedging bandwidth defining the no-transaction region around the Black-Scholes delta (see [136], section 4). In the case of *small* proportional costs equal to some $b > 0$ in the financing equation (2.4), the no-transaction region is given by

$$-\Delta_k(s_k) - Z_k(s_k) \leq u_k \leq -\Delta_k(s_k) + Z_k(s_k), \quad (4.20)$$

where u_k is the stock quantity held at time t_k , $\Delta_k(s_k)$ is the Black-Scholes delta from equation (1.2) and

$$Z_k(s_k) = \left(\frac{3b}{2\gamma} e^{r(T-t_k)} \right)^{1/3} |\Gamma(t_k, s_k)|^{2/3}, \quad (4.21)$$

where $\Gamma(t_k, s_k) = \partial^2 h(t, s) / \partial s^2$ is the second derivative of the Black-Scholes price of the option, with respect to the stock price s , evaluated at (t_k, s_k) . Here again, γ is the risk aversion parameter for the negative exponential loss function.

In the same context, Barles and Soner [9] apply a different asymptotic analysis, but they do not provide an explicit form for the hedging bands, i.e. they are given

as

$$Z_k(s_k) = \left(\frac{1}{bs_k\gamma^{3/2}} \right) g \left((bs_k)^2 \gamma \Gamma(t_k, s_k) \right), \quad (4.22)$$

for some non-linear function $g(x)$ and using a particular volatility adjustment for computing the functions $\Delta_k(s_k)$ and $\Gamma_k(s_k)$. Kallsen and Muhle-Karbe [76] use a formal series expansion as in [135], but use it to provide an approximation of the optimal hedging bands as

$$Z_k(s_k) = \left(\frac{3bs_k}{2\gamma} \frac{d\langle\varphi\rangle_k}{\langle s \rangle_k} \right)^{1/3}, \quad (4.23)$$

where $\langle\varphi\rangle_k$ and $\langle s \rangle_k$ are the local quadratic variations of the “frictionless optimizer” (i.e. the delta in the case without costs) and of the underlying process, respectively. Albanese and Tompaidis [2] also use asymptotic analysis, but in their case the goal is to derive approximate expressions for the optimal time between transactions and the optimal volatility adjustment to make to account for transaction costs.

Simulation experiments by Mohamed [105] were the first to provide numerical evidence that the Whalley-Wilmott policy works well in practice when compared to some heuristic approaches to hedging with transaction costs. The problem considered was that of hedging a call option, using the 5th percentile of the profit and loss distribution as a risk measure, for one particular specification of the option and market parameters. Martinelli and Priaulet [101] undertook a more systematic empirical testing of hedging policies under costs. They used a mean-variance framework, focusing on the expected transaction costs and the standard-deviation of hedging errors, and considered the effect of various problem parameters such the option strike price and volatility level, as well as the effect of using a volatility level based on a GARCH(1,1) process instead of a fixed volatility. One of the hedging methods they consider is with a delta bandwidth, but this bandwidth is fixed, and not a function of the underlying price as in the Whalley-Wilmott policy. Nevertheless, their results

provide numerical evidence that hedging bands based on the option delta often provide better hedging than hedging bands around the underlying price or than using various versions of fixed time hedging (including multi-time scale hedging, where different fractions of the hedge are computed). Interestingly, the results showed that the relative performance of the delta bandwidth method was not as good under the GARCH(1,1) volatility setup in cases when a low hedging error is preferred.

4.2.2 Parametric approximation of Zakamouline

Within the negative exponential risk framework in continuous time, Zakamouline [138] posits a general parametric form for the optimal policy, based on the functional form of asymptotic solutions derived by Whalley and Wilmott [136] and Barles and Soner [9]. That is, the policy is based on two hedging bands as in equation (4.20), but with a delta $\Delta_k(s_k, \sigma_m)$ depending on a modified volatility parameter σ_m and with the hedging band width Z_k defined as a sum of two functions $Z_k = H_w + H_0$, with all of σ_m , H_w and H_0 also being functions of a set of unknown constants. For example, the function H_0 is defined by

$$H_0(t, s, \sigma) = \alpha \sigma^{\beta_1} b^{\beta_2} (\gamma s)^{\beta_3} (T - t)^{\beta_4}, \text{ where } \beta_i \in \mathbb{R}, i = 1, \dots, 4. \quad (4.24)$$

The constants β_i are specified through an optimization procedure based on linear regression to yield a good fit to the optimal policy, as approximated by the numerical solution of a PDE (see Clewlow and Hodges [40]). This was done over a wide range of values for the various (known) parameters of the problem, such as the risk aversion $\gamma \in [0.05, 15]$, maturity $T \in [0, 1.5]$, costs $b \in [0.001, 0.02]$ and volatility $\sigma \in [0.1, 0.4]$ for the problem of hedging a call option under the BSM model (for $s_0 = 100$).

Explicitly, the Zakamouline approximation, as defined by equations (10), (16),

(17) and (18) in [138], uses a delta given by

$$\Delta_k(s_k, \sigma_m) = \frac{\partial h_k(s_k, \sigma_m)}{\partial s_k}, \quad (4.25)$$

with

$$\begin{aligned} \sigma_m^2(t_k, s_k, \sigma) &= \sigma^2 (1 + H_\sigma(t_k, s_k, \sigma)), \\ H_\sigma(t_k, s_k, \sigma) &= 4.76 \frac{b^{0.78}}{\sigma^{0.25}} (\gamma s_k^2 |\Gamma(t_k, s_k, \sigma)|)^{0.15}. \end{aligned}$$

Furthermore, the hedging band size Z_k from (4.20) is defined by

$$Z_k(t_k, s_k, \sigma) = H_w(t_k, s_k, \sigma) + H_0(t_k, s_k, \sigma) \quad (4.26)$$

where

$$\begin{aligned} H_0(t_k, s_k, \sigma) &= \frac{b}{\gamma s_k \sigma^2 (T - t_k)}, \\ H_w(t_k, s_k, \sigma) &= 1.12 \frac{b^{0.31}}{\sigma^{0.25}} \left(\frac{|\Gamma_k(s_k, \sigma)|}{\gamma} \right)^{0.5}. \end{aligned}$$

The empirical performance is very good for the case of hedging a call option under the BSM model (by construction), but it is not clear how well the policy should do under other setups (i.e. different derivative to hedge or price dynamics).

4.2.3 One step lookahead and local hedging

We are also interested in heuristics that can take more complex (multi-dimensional) price dynamics into account if necessary, but without trying to solve the full dynamic program given by equation (2.3). One possibility, which we refer to as *local hedging*,

is to consider only the one-step programs

$$\min_{v \in \mathcal{U}} \mathbf{E}_k [e^{-\gamma(V_{k+1}(X_{k+1}) - V_k(X_k))}], \quad (4.27)$$

where $X_{k+1} = \psi(X_k, v, Y_{k+1})$, at each time step $k = 0, \dots, K - 1$. Then define a policy $\pi = (\mu_1, \dots, \mu_K)$ with decisions $\mu_{k+1}(X_k) = v^*$ given by the optimal solution to equation (4.27). This strategy is simple to implement (as there is no optimal risk function to keep track of at later steps), yet still takes into account the tradeoff between reducing portfolio volatility and incurring transaction costs.

This policy has already been studied extensively in the quadratic risk setting under the name *local risk-minimization* (see Lambertson, Pham and Schweizer [83], Schweizer [122] and the references therein). Among other results, Schweizer [121] showed that the local hedging policy is optimal for the global problem when the objective is quadratic, there are no transaction costs and the price process is a martingale (but not if the process has a drift). The performance of this policy in the case without costs was compared to that of a policy minimizing a global quadratic objective by Heath, Platen and Schweizer [69].

Here we show a similar result applicable to the negative exponential risk function, but with an added condition on the increments of the market process :

Proposition 4.2.1 *Let the loss function be given by $L(x) = e^{-\gamma x}$. Assume that there are no transaction costs and that the market process is a martingale. Assume also that that the market process increments $Y_{k+1} - Y_k$ are independent. Finally, let π be the local hedging policy, with decisions $\mu_{k+1}(X_k)$ at each step $k = 0, \dots, K - 1$ defined as the values v^* that solve the one-step program (4.27). Then π is optimal for the global hedging problem.*

Proof: Here we use again the notation from section 2.4, specific to the case of the negative exponential loss function. Assuming there are no costs, we have that

the optimal transaction at stage $k + 1$ is $v^* = b_{k+1}^+ = b_{k+1}^-$. So there are no no-transaction boundaries and the optimal risk function $G_{k+1}^*(X_{k+1})$ only depends on the market vector Y_{k+1} . But if there are no costs and the market vector process $\{Y_k : k = 0, \dots, K\}$ is a martingale with independent increments, then the portfolio value process $\{V_k^\pi : k = 0, \dots, K\}$ is also a martingale with independent increments. Now note that the risk function $G_k^\pi(X_k)$ from equation (2.42) can be expressed as

$$G_k^\pi(X_k) = \mathbf{E}_k[e^{-\gamma \sum_{j=k}^{K-1} (V_{j+1}^\pi - V_j^\pi)}], \quad (4.28)$$

for any policy π , and in particular for the optimal policy π^* . So by the independence of the increments $V_{k+1}^\pi - V_k^\pi$, the expected risk function R_k from equation (2.44) can be separated as the product

$$R_k(X_k, v) = \mathbf{E}_k[e^{-\gamma(V(\psi(X_k, v, Y_{k+1})) - V(X_k))}] \times \mathbf{E}_k[G_{k+1}^*(\psi(X_k, v, Y_{k+1}))], \quad (4.29)$$

where the second term only depends on Y_k and not on v . It follows that the decisions for the local hedging policy are the same as those for the (global) optimal hedging policy, which proves the result. \square .

In the case with costs, local hedging could still do well if costs are low relative to the underlying asset's volatility σ . More precisely, this should hold true especially if transaction costs $a + bs_k$ are low relative to the variance of the price process over one step (approximately $s_k^2 \sigma_k^2 \Delta t_k$). Thus we can expect that local hedging under non-zero costs should be closest to the optimal solution as the size of the time steps Δt_k gets larger.

More generally, given an approximation \widehat{J}_k of the optimal risk function J_k^* , we can define decision functions μ_{k+1} by

$$\mu_{k+1}(X_k) = \arg \min_{v \in \mathcal{U}} \mathbf{E}_k[e^{-\gamma(V_{k+1}(X_{k+1}) - V_k(X_k))} \widehat{J}_{k+1}(X_{k+1})], \quad \forall X_k \in \mathcal{X}, \quad (4.30)$$

where $X_{k+1} = \psi(X_k, v, Y_{k+1})$, for $k = 0, \dots, K-1$. This yields the *one-step lookahead* (1SL) policy $\pi^{(1)} = (\mu_1, \dots, \mu_K)$ associated with \widehat{J}_{k+1} . Local hedging can thus be interpreted as 1SL applied to a constant approximate risk function (say $\widehat{J}_{k+1}(X_{k+1}) = 1, \forall X_{k+1} \in \mathcal{X}$).

4.2.4 Local hedging approximation

To obtain a better intuitive sense about the local hedging policy, we would like to consider the first few terms of a series expansion of the two boundary Q-functions Q_k^\pm for the local hedging objective. For this, we first write $Q_k^\pm(X_k, v)$ as $Q_k^\pm(X_k, v) = \mathbf{E}_k[L(r^\pm(\omega))]$, for $L(x) := e^{-\gamma x}$ and $r^\pm(\omega) := v\Delta s_{k+1}(\omega) + \Delta h_{k+1}(\omega) \pm \phi(s_k)(v - u_k)$, and $\omega \in \mathcal{F}_{k+1}$. We then apply the stochastic Taylor's Theorem (see theorem 2.1.1) to the functions $L(r^+(\omega))$ and $L(r^-(\omega))$, which yields

$$L(r^\pm(\omega)) = L(0) + L^{(1)}(0)r^\pm(\omega) + \frac{1}{2}L^{(2)}(0)(r^\pm(\omega))^2 + \frac{1}{3!}L^{(3)}(\xi^\pm(\omega))(r^\pm(\omega))^3, \quad (4.31)$$

for two measurable functions $\xi^+(\omega)$ and $\xi^-(\omega)$. Using the fact that $L(0) = 1$, $L^{(1)}(0) = -\gamma$, $L^{(2)}(0) = \gamma^2$ and $L^{(3)}(x) = -\gamma^3 e^{-\gamma x}$, we can write more explicitly

$$L(r^\pm(\omega)) = 1 - \gamma r^\pm(\omega) + \frac{\gamma^2}{2}(r^\pm(\omega))^2 - \frac{\gamma^3}{3!}(e^{-\gamma \xi^\pm(\omega)})(r^\pm(\omega))^3. \quad (4.32)$$

And finally, after taking expectations, we see that the functions Q_k^\pm can be written as

$$Q_k^\pm(X_k, v) = 1 - \gamma \mathbf{E}_k[r^\pm(\omega)] + \frac{\gamma^2}{2} \mathbf{E}_k[(r^\pm(\omega))^2] - \frac{\gamma^3}{3!} \mathbf{E}_k[(e^{-\gamma \xi^\pm(\omega)})(r^\pm(\omega))^3]. \quad (4.33)$$

Thus, ignoring the term in $O(\gamma^3)$, we have the approximation

$$Q_k^\pm(X_k, v) \approx 1 \mp \gamma \phi(s_k)(v - u_k) - \frac{\gamma^2}{2} \mathbf{E}_k \left[(v \Delta s_{k+1} + \Delta h_{k+1} \pm \phi(s_k)(v - u_k))^2 \right]. \quad (4.34)$$

Recall from equations (2.22) and (2.23) that the transaction boundaries $b_k^-(X_k)$ and $b_k^+(X_k)$ minimize the functions $Q_k^+(X_k, \cdot)$ and $Q_k^-(X_k, \cdot)$, respectively. Thus assuming that $b_k^-(X_k)$ and $b_k^+(X_k)$ both lie inside an open set $\mathcal{U}^\circ \subset \mathcal{U}$, with \mathcal{U} being the set of admissible trading decisions, then the $b_k^\pm(X_k)$ are given as the solutions of the first order optimality condition $\partial Q_k^\pm(X_k, v)/\partial v = 0$. It then follows easily from approximation (4.34) that $b_k^-(X_k)$ and $b_k^+(X_k)$ are given (again, approximately) by

$$b_k^\pm(X_k) \approx \frac{1}{\mathbf{E}_k[\Delta s_{k+1}^2] + \phi(s_k)^2} \left(-(\mathbf{E}_k[\Delta s_{k+1} \Delta h_{k+1}] - \phi^2(s_k)u_k) \pm \frac{\phi(s_k)}{\gamma} \right). \quad (4.35)$$

Note that $\beta := \mathbf{E}_k[\Delta s_{k+1} \Delta h_{k+1}]/\mathbf{E}_k[\Delta s_{k+1}^2]$ corresponds to the linear regression coefficient of Δh_{k+1} with respect to Δs_{k+1} (known in finance as the *beta*). So making the approximation $\beta \approx \Delta_k^{\text{BSM}}$ (the Black-Scholes-Merton delta), which is reasonable if the time interval $\Delta t = t_{k+1} - t_k$ is small, we can re-write equation (4.35) as

$$b_k^\pm(X_k) \approx \frac{1}{1 + Z} \left(-\Delta_k^{\text{BSM}} + u_k Z \pm \frac{Z}{\phi(s_k)\gamma} \right), \quad (4.36)$$

where $Z = \phi(s_k)^2/\mathbf{E}_k[\Delta s_{k+1}^2]$.

The local hedging approximation policy should only do well in the restrictive case where both unit costs $\phi(s_k)$ and risk aversion γ are low. Nevertheless, we try it out as a policy since it is even easier to compute than local hedging, as no numerical optimization is involved. Furthermore, if the local hedging objective needs to be estimated by Monte Carlo simulation, the policy will contain some noise, which is not the case with the analytic approximation given by equation (4.36).

4.3 Conclusion

In the first part of this chapter, we presented the exponential Ornstein-Uhlenbeck model, which is an extension of the basic geometric Brownian motion model that allows for a mean-reverting stochastic volatility. We also estimated the value of its parameters for the S&P 500 index over various time periods, so that our simulation experiments are based on realistic parameter values, though parameters chosen for experiments may not necessarily correspond to a specific time period.

We further described one way in which this model could be extended by including two additional variables, representing information available to the trader that is correlated to the price and volatility movements. This extension has not been studied previously in the literature, and was introduced here to allow both to test the possible impact of including extra information on the hedging solutions and to test stochastic mesh methods on a higher dimensional problem, i.e. 4 dimensions for the mesh states instead of 2. To our knowledge, the stochastic mesh method has not yet been tested on models other than (possibly multi-dimensional) geometric Brownian motion. But for our hedging problem, we consider only one underlying asset, because of the difficulty of modelling the risk as function of the stock quantity. Hence a geometric Brownian motion process with $d > 1$ dimensions is not a natural example to use in this setting.

In the second part of the chapter, we described some well-known heuristics for the problem of hedging under transaction costs. The heuristics presented all include the notion of hedging bands and an associated no-transaction region. Two of them are specifically designed for the context of negative exponential risk, namely the Whalley-Wilmott and the Zakamouline approximations. We also considered local hedging, which corresponds to the one-step lookahead policy applied to a constant risk function. Similar to the results from Schweizer [121] for the quadratic risk case, we showed in proposition 4.2.1 for the case of negative exponential risk that local

hedging was optimal when there are no costs. Other application of the one-step lookahead approach to more general risk function approximations were not considered due to time constraints. Finally, we derived in section 4.2.4 an analytical approximation to local hedging in terms of hedging bands.

CHAPTER 5

NUMERICAL RESULTS FOR THE BASIC MESH CONSTRUCTION

In this chapter, we provide some numerical illustrations to the theory developed so far and details as to how these numerical results are obtained. We start in section 5.1 by an overview of the *MeshHedging* Java library, which was developed specifically to perform numerical experiments for this thesis. Section 5.2 then presents a general algorithm used in the *MeshHedging* library for estimating the risk of various policies. This includes both (low biased) estimates of the optimal risk obtained from the stochastic mesh construction and (high biased) out-of-sample estimates of the risk from heuristics and stochastic mesh based policies. Numerical results illustrating the convergence of the high and low biased stochastic mesh estimators are then presented in section 5.3. And finally, in section 5.4, we compare a basic stochastic mesh based policy (using AD weights and a linear-quadratic approximation for the cost function) to the heuristics from section 4.2 under different settings, involving high or low costs, volatility, etc. This serves to highlight the cases for which the stochastic mesh approximation can be expected to work well and the cases that would require further improvements.

5.1 MeshHedging : a Java library for computing stochastic mesh based hedging policies

In order to run numerical experiments in the context of the present thesis, a Java language library, which we call the *MeshHedging* library, was developed. It is publicly available on-line on the GitHub repository at <https://github.com/average3101/MeshHedging>. This library makes use of external classes from existing libraries :

- The *SSJ* library [89, 92], for random number generation, basic probability

distributions and stochastic processes,

- The *COLT* library [72] for linear algebra,
- The `FastMath` class of the *Commons Math* library [1], for fast implementations of some standard mathematical functions (such as the exponential function),
- The `Fmin` class [134], which is an implementation of Brent’s method for finding the minimum of a 1-dimensional function. The code from this class was included directly in the class `BrentSolver` of the *MeshHedging* library. It has been modified to include a limit on the number of iterations.

The *MeshHedging* library itself is composed of various packages, each dealing with a separate aspect of implementing stochastic mesh methods for hedging :

- *derivatives* : This package is used to define options and their associated prices. Classes representing derivatives (such as `CallOption` and `PutOption`) are subclasses of the abstract class `Derivative`. By default, prices from `CallOption` and `PutOption` are based on methods from the static class `OptionPricing`, which provides standard option pricing methods based on the Black-Scholes-Merton model. When pricing options on a stochastic mesh based on another model than the BSM model, the class `DerivativePricedOnMesh` should be used instead. Its constructor takes a `Derivative` object as its argument
- *dynprog* : This package contains classes defining the risk function approximations, which are classes extending the abstract class `RiskFunctionApproximation`. The Q_k functions are represented by the class `QFunction` and the linearized versions Q_k^+ and Q_k^- used in finding the no-transaction boundaries are represented by the class `QFunctionForTradeBoundaries`. The abstract class `DynamicHedgingProgram` and its extension `DynamicHedgingProgramUsingLQApproximation` contain as members the various problem elements (loss

function, risk function approximation, etc) and implement the risk estimation algorithm from section 5.2.

- *experiments* : This package helps to set up simulation experiments regarding the estimation of the expected risk of hedging policies. In particular, the class `HedgingComparisons` was used as the entry point to generate many of the results presented in this thesis. It takes as input a file defining the experiment parameters and interprets these values to construct an instance of the `RiskMeasureEstimator` class, which is used to compute sample path estimates of the risk for hedging policies.
- *meshmethods* : The two most important classes of this package are `StochasticMesh`, which contains the market vector values used to define the mesh, and `Weights`, which deals with storing the relevant information about weights, including how they are computed. Subclasses of these two classes then give the user the flexibility to implement the different mesh setups described in this thesis (e.g. `SingleGrid` or `MeshWeightsAverageDensity`).
- *numerics* : This package is targeted at the optimization of 1-dimensional functions. The abstract class `OneDimSolver` is designed to have as a member variable a one-dimensional function, represented as an object that implements the `OneDimFunction` interface. After calling the method `evaluate` from a `OneDimSolver` object, the information about the solution of the minimization are returned as a `OneDimSolverResults` object.
- *policies* : Here, policies are defined as classes extending the abstract class `HedgingPolicy`. The class `RiskFunctionBasedPolicy` is a special case which computes the hedging decisions through a `DynamicHedgingProgram` object from the *dynprog* package (i.e. it computes the decision that minimizes the risk associated to a given stochastic dynamic program, given the current state X_k

and the risk function approximations \widehat{J}_{k+1} for the next step). The class `HedgingPortfolio` is used to define the portfolio value path $\{V_k^\pi : k = 0, \dots, K\}$ for a given path of the market information process $\{Y_k : k = 0, \dots, K\}$.

- *stochprocess* : This package defines market vectors and market vector processes, through extensions of the abstract classes `MarketVector` and `MarketVectorProcess`, respectively.
- *tests* : This packages contains various executables that allow to run validation tests on other packages, such as *meshmethods*, *stochprocess*, *dynprog*. The tests are basic, such as checking that the sum of stochastic mesh weights are equal to one, or that the risk functions are convex (for a convex loss function). In the cases where the values tested are estimates, thresholds are used to determine whether or not the test passed or failed. For the package *meshmethods*, it is expected that some tests will fail if the number of mesh points is too low (e.g. for likelihood ratio weights).

5.2 Risk estimation algorithm

The comparison of the out-of-sample performance of mesh-based hedging policies with the performance of heuristic policies is done using algorithm 4. Note that the estimation of the risk is done in two steps. First, an estimate is computed at step 1.e as an average over n sample paths. This is repeated over n_R replications, each using an independent mesh, and combined in the average at step 3.

A more direct way of obtaining such results would be to look at the average over the total number $n_{total} = n_R \times n$ of out-of-sample paths, each using an independent mesh to define the mesh-based policy. But in the case of a stochastic mesh based policy, a large amount of computation time is spent on evaluating the mesh weights. Thus separating the experiment in two loops allows to cut on simulation time, while

Algorithm 4 Policy comparison

1 For $j = 1, \dots, n_R$ replications :

a: Define a stochastic mesh using N independent sample paths of the market vector process and using the AD weights from equation (3.12).

b: Apply the stochastic mesh DP algorithm (algorithm 3) from section 3.2.1 using approximation 3.3.1 to obtain a low biased estimate $\widehat{J}_0^{L,j}(X_0)$ of $J_0(X_0)$.

c: Simulate n independent paths of the market vector process $\{Y_k^i : k = 0, \dots, K, i = 1, \dots, n\}$.

d: Apply each policy π_p ($p = 1, \dots, n_P$) to each $i = 1, \dots, n$ path to obtain the terminal portfolio values $V_K^{\pi_p, i}(X_0)$

e: Use these values to compute an estimate of the risk as the sample average

$$\widehat{J}_0^{\pi_p, j}(X_0) = \frac{1}{n} \sum_{i=1}^n L \left(V_K^{\pi_p, i}(X_0) - V_0 \right). \quad (5.1)$$

2 For each policy π_p ($p = 1, \dots, n_P$), take an average of the estimators from step 1.e to obtain the high biased risk estimates

$$\widehat{J}_0^{\pi_p}(X_0) = \frac{1}{n_R} \sum_{j=1}^{n_R} \widehat{J}_0^{\pi_p, j}(X_0). \quad (5.2)$$

3 Compute the stochastic mesh low biased risk estimate as the average of the estimates from step 1.b

$$\widehat{J}_0^L(X_0) = \frac{1}{n_R} \sum_{j=1}^{n_R} \widehat{J}_0^{L, j}(X_0). \quad (5.3)$$

still obtaining precise estimates \widehat{J}_0 and $\widehat{\epsilon}_0$ for the risk function and associated error. To be more explicit, we can write the total computation time as

$$C(\text{total}) = n_R [C(\text{mesh generation}) + n \times C(\text{policy evaluation})], \quad (5.4)$$

where $C(\cdot)$ represents the computation time for the element (\cdot) . If

$$C(\text{mesh generation}) \gg C(\text{policy evaluation}) \quad (5.5)$$

and $n_{\text{total}} := n_R \times n$ is fixed, then the total computation time is minimized by letting $n_R = 1$ and $n = n_{\text{total}}$. But with $n_R = 1$, we cannot estimate the error of the low biased (in-sample) mesh estimate $\widehat{J}_0^L(X_0)$ of $J_0(X_0)$. So in order to balance these considerations and have a good estimate of the error of $\widehat{J}_0^L(X_0)$, we choose both $n_R \gg 1$ and $n \gg 1$.

5.3 Illustration of the convergence of mesh estimators

Figures 5.1 and 5.2 show approximate risk function values $\widehat{J}_0(X_0)$ for the problem of hedging a call option under the GBM model, and figures 5.3 and 5.4 show the same under the expOU model. All are given as functions of the number of mesh paths N . The point here is to compare the low biased (Mesh-LB) and high biased (Mesh-HB) stochastic mesh based estimates, in order to observe evidence of their convergence as N increases. For reference, we compare the mesh-based estimates to the (high biased) risk estimates associated to the following policies: Black-Scholes-Merton (BSM), Whalley-Wilmott (WW) and Zakamouline (Z) and local hedging (Local). In the case of the high biased estimate for the stochastic mesh, the policy used is the one described in proposition 2.3.1, using the linear-quadratic approximation from section 3.3.1 to define the no-transaction bounds b_k^- and b_k^+ . The details of the experiment, including the mesh construction, are given in algorithm 4 of section 5.2.

Figures 5.1 and 5.2 both use $n_R = 1\,000$ replications and $n = 1\,000$ paths per replication. The error bars correspond to the sample standard deviation of each of these averages. And in all cases, the initial stock price is $s_0 = 10$, the option's expiry is in $T = 6$ months, the number of time steps is $K = 8$ and the proportional transaction costs are $b = 2\%$.

In figures 5.1 and 5.2, the dashed red line (Mesh-LB) is a the stochastic mesh based risk estimate (low-biased) and the full red line (Mesh-HB) is the out-of-sample estimate using mesh policy (high-biased). The fact that they tend to get very close for higher values of N shows that the optimum is nearly attained by the stochastic mesh based policy. In the case with low volatility ($\sigma = 20\%$) and low risk aversion ($\gamma = 1$), the Whalley-Wilmot (WW), Zakamouline (Z), local hedging (Local) heuristics do well and are close to the optimum. The Black-Scholes-Merton (BSM) delta hedging policy does poorly, as it does not take costs into account. Its performance is in fact worst than doing nothing (as the no-hedging policy NH) ! However, when the volatility and risk aversion parameters are higher ($\sigma = 40\%$, $\gamma = 5$), the BSM policy does better than other heuristics. The local hedging policy risk estimate also tends to be close to the BSM policy risk estimate for higher N , but this is not necessarily observed in other setups. Here it could be explained by the fact that the price uncertainty is then higher relative to the actual transaction costs, so that the local hedging objective will favor trading over the “fear” of incurring transaction costs. By price uncertainty, we mean that for $T = 6$ months and $K = 8$, we have time steps of $\Delta t = 0.5/8 = 0.075$ years, and thus a price standard deviation of about $s_0 \times \sigma \times \sqrt{\Delta t} = 10 \times 40\% \times \sqrt{0.075} \approx 1.1$ at each step, compared to costs of the order of $s_0 \times b = 10 \times 2\% = 0.2$.

In figures 5.3 and 5.4, we show results of similar experiments, but for the case of the expOU model of section 4.1. The model parameters are chosen to be $\kappa = 2.6$, $\sigma_v = 0.6$, $\sigma_0 = 40\%$ and $\bar{\sigma} = 20\%$ for both figures, the only difference being the

value for γ (respectively, $\gamma = 1$ and $\gamma = 5$). First note that there is a (high) bias in $\widehat{J}_0^\pi(X_0)$ that is apparent for any policy π for lower values of N , in both figures, not just the stochastic mesh based estimates. Indeed, the estimates of $\widehat{J}_0^\pi(X_0)$ should be independent of N for the heuristic policies BSM, WW, Z and NH, but observe that they follow a curve instead of a straight line. This bias is caused by the use of the stochastic mesh itself to approximate the value the call option, instead of using an exact analytic formula, as is available for the GBM model. As explained in section 3.2.4, such additional noise cannot be hedged, thus increasing the value of the risk measure. An additional consequence is that the estimate Mesh-LB is no longer low biased. More precisely, the low bias of the Mesh-LB estimator came from the minimization step of the DP algorithm, assuming that all expectations were unbiased. But again, as seen in section 3.2.4, the noise from the estimator \widehat{h}_k of $h_k(Y_k)$ introduces a high bias in the estimator $\widehat{Q}_k(X_k, v)$, so that the overall bias not necessarily low anymore (and indeed, it is clearly high in the case of figure 5.4).

These bias issues are not specific to the expOU model, as similar results are obtained if simulating under the GBM model and evaluating a call option using the mesh (not shown here). In the case of figure 5.4, the effect of the bias is magnified by the higher value of γ in the exponential loss function (that is, $\gamma = 5$ instead of $\gamma = 1$ in figure 5.3).

5.4 Global performance comparison of some heuristic policies using simulated data

In this section, we present some results from using stochastic mesh methods to approximate solutions to the problem of hedging a call option under transaction costs (see section 1.2). We compare the mesh based policy to the heuristics described in section 4.2, for both the optimal decisions and risk functions, for a wide range of experimental setups.

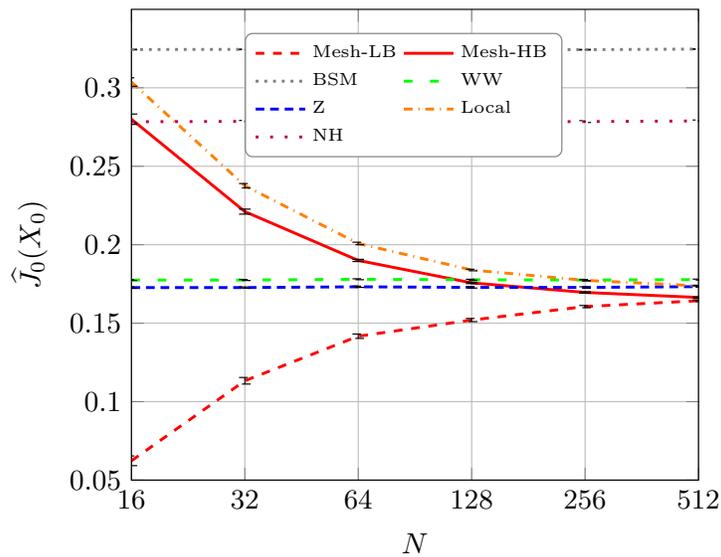


Figure 5.1: Risk estimates $\hat{J}_0(X_0)$ as a function of N , for a call option under the GBM model ($\sigma_0 = 20\%$, $b = 2\%$, $\gamma = 1$, $K = 8$).

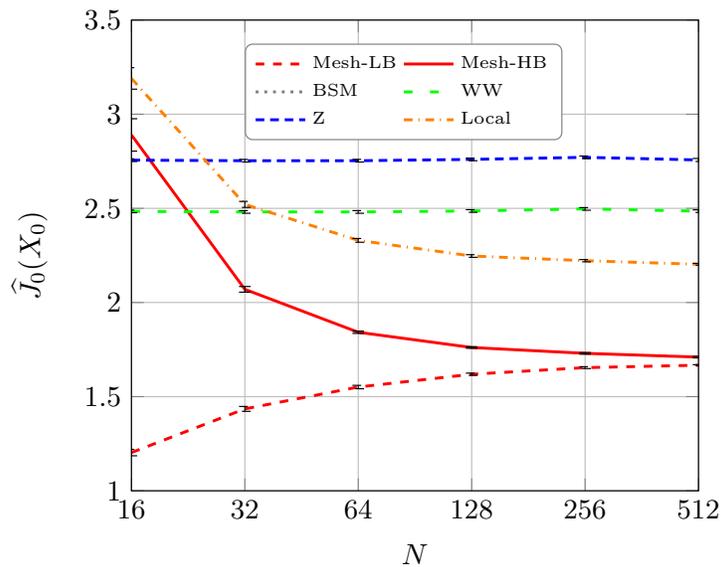


Figure 5.2: Risk estimates $\hat{J}_0(X_0)$ as a function of N , for a call option under the GBM model ($\sigma_0 = 40\%$, $b = 2\%$, $\gamma = 5$, $K = 8$).

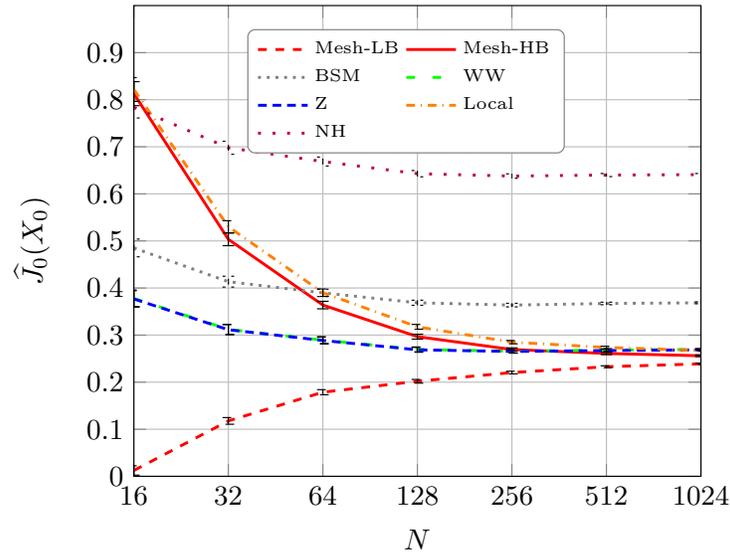


Figure 5.3: Risk estimates $\hat{J}_0(X_0)$ as a function of N , for a call option under the expOU model ($\sigma_0 = 40\%$, $\bar{\sigma} = 20\%$, $\sigma_v = 1.2$, $\rho = 0.5$, $\kappa = 2.6$, $b = 2\%$, $\gamma = 1$, $K = 8$).

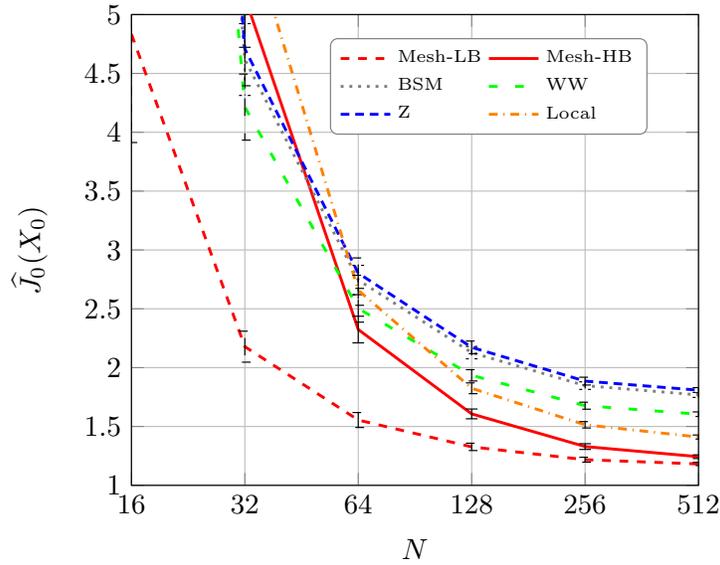


Figure 5.4: Risk estimates $\hat{J}_0(X_0)$ as a function of N , for a call option under the expOU model ($\sigma_0 = 40\%$, $\bar{\sigma} = 20\%$, $\sigma_v = 1.2$, $\rho = 0.5$, $\kappa = 2.6$, $b = 2\%$, $\gamma = 5$ and $K = 8$).

The experiments aim to highlight that :

1. There are some settings where known heuristics do not do well compared to the optimal solution, which is known to lie between the low and high biased mesh estimators. This shows that there is something to be gained by trying to compute the optimal solution, i.e., there is room for improving the hedging policies compared to the heuristics, and that using a stochastic mesh is a possible way to do it.
2. There are setups where convergence of the stochastic mesh is slower and/or not attained in reasonable time. This shows the need for efficiency improvement techniques, some of which will be introduced in chapter 6.

Table 5.I provides estimates of the portfolio risk when the stock price dynamics is given by the discrete time GBM model. Overall, the experiments are constructed as in section 5.3, using algorithm 4, unless specified otherwise below. We consider a fixed expiry date $T = 6$ months, but various values for the number of time steps K , the risk aversion parameter γ , the volatility σ , and the transaction cost parameter b . For the low biased (Mesh-LB) and high biased (Mesh-HB) path estimates, we use a stochastic mesh with average density weights, $N = 512$ nodes and the linear-quadratic approximation of section 3.3.1 for the risk function. A rough estimate $\hat{\epsilon}_0$ of the approximation error bound from section 3.3.2 is provided using $M = 11$ points. The path estimators (out-of-sample, high biased) for Mesh-HB and the various heuristic policies, are computed using an average over $n_R = 1\,000$ independent replications of $n = 1\,000$ Monte Carlo estimates of the risk. In the case of the stochastic mesh estimate Mesh-LB (in-sample, low biased), a new mesh is computed for each of the n_R replications. The heuristics considered are : Black-Scholes-Merton delta hedging (BSM), Whalley-Wilmott (WW), Zakamouline (Z), local hedging (Local), local hedging approximation (Local-A) and no hedging (NH).

On the right-hand side of the table, each cell has two values: the sample average estimate and the sample standard deviation of the average (i.e., $\bar{s} = s/\sqrt{n_R}$), which are given above and below, respectively. The values for the high biased estimators that are within $\delta = 0.05$ of the lowest high biased estimate are highlighted in various shades of green (the lower the value is, the darker the color). So if the lowest high biased estimate is $\hat{J}^\pi = 0.025$, then this value will be dark green, and a value of $\hat{J}^\pi + \delta = 0.075$ will be white. However, if an estimate is above $\hat{J}^\pi + \delta$, it is simply highlighted in red, to indicate that it is somewhat poor compared to the best obtained values. The value of the threshold of δ is arbitrary, but was chosen to help illustrate the relative performance of each policy, for this particular experimental setup.

Similarly, the low biased estimator \hat{J} (Mesh-LB) is colored with various shades of green, depending on its distance with the smallest high biased estimate \hat{J}^π . Values below $\hat{J}^\pi - \delta$ are colored red and values above $\hat{J}^\pi - \delta$ are colored with darker green shades as they get closer to \hat{J}^π . It is important to note that if none of the values of the high biased estimators are close to the low biased estimator, it does not necessarily mean that they are all far from the actual optimum, as the low biased stochastic mesh estimator could itself be far from the optimum.

Here are some observations on these results :

- The sample averages for the error bound estimates $\hat{\epsilon}_0$ of the linear-quadratic approximation are relatively small. Among the observed cases, they correspond to at most 5% of the objective value (for the case $\sigma = 20\%$, $\gamma = 5$, $K = 8$ and $b = 2\%$).
- The low biased mesh estimator (Mesh-LB) is always within $\delta = 0.05$ of the smallest high biased estimate of the risk function value. Visually, this can be seen by the absence of red cells in the Mesh-LB column of the table, although of course a different choice of threshold δ would change the overall picture. For this estimator, the worst relative performances (when comparing with the

Table 5.I: Estimates of the portfolio risk when hedging a call option under the exponential loss function and the discrete GBM model for various combinations of the parameters γ , σ , K and b . Mesh based estimates using $N = 512$ points.

γ	σ	K	b	$\hat{\epsilon}_0$	Mesh-LB	Mesh-HB	Local	Local-A	Z	WW	BSM	NH
1	0.2	4	0	0	0.023	0.028	0.028	0.025	0.025	0.025	0.025	0.278
				-	-	-	-	-	-	-	-	0.001
1	0.2	4	0.01	0	0.107	0.111	0.115	0.112	0.123	0.115	0.135	0.278
				-	-	-	-	-	-	-	-	0.001
1	0.2	4	0.02	0.001	0.164	0.166	0.199	0.184	0.182	0.173	0.257	0.278
				-	-	-	0.001	-	-	-	-	0.001
1	0.2	8	0	0	0.007	0.02	0.019	0.014	0.014	0.014	0.014	0.279
				-	-	-	-	-	-	-	-	0.001
1	0.2	8	0.01	0.001	0.1	0.107	0.146	0.136	0.114	0.111	0.158	0.279
				-	-	-	0.001	-	-	-	-	0.001
1	0.2	8	0.02	0.004	0.157	0.163	0.257	0.253	0.173	0.178	0.324	0.279
				-	-	-	0.001	0.001	-	-	-	0.001
1	0.4	4	0	0	0.096	0.102	0.102	0.101	0.101	0.101	0.101	1.064
				-	-	-	-	-	-	-	-	0.001
1	0.4	4	0.01	0	0.208	0.213	0.215	0.213	0.253	0.243	0.219	1.064
				-	-	0.001	0.001	0.001	0.001	0.001	0.001	0.001
1	0.4	4	0.02	0	0.31	0.315	0.321	0.317	0.356	0.338	0.351	1.064
				-	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
1	0.4	8	0	0	0.046	0.061	0.059	0.054	0.054	0.054	0.054	1.066
				-	-	-	-	-	-	-	-	0.001
1	0.4	8	0.01	0	0.173	0.184	0.194	0.188	0.212	0.2	0.205	1.066
				-	-	-	0.001	-	0.001	0.001	-	0.001
1	0.4	8	0.02	0.002	0.28	0.29	0.326	0.318	0.323	0.305	0.379	1.066
				-	0.001	0.001	0.001	0.001	0.001	0.001	-	0.001
5	0.2	4	0	0	0.138	0.14	0.142	0.149	0.149	0.149	0.149	1.749
				-	-	-	-	-	-	-	-	0.002
5	0.2	4	0.01	0.001	0.332	0.335	0.344	0.348	0.409	0.396	0.36	1.749
				-	-	0.001	0.001	0.001	0.001	0.001	0.001	0.002
5	0.2	4	0.02	0.009	0.557	0.56	0.582	0.588	0.651	0.602	0.705	1.749
				-	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.002
5	0.2	8	0	0	0.069	0.073	0.074	0.075	0.075	0.075	0.075	1.752
				-	-	-	-	-	-	-	-	0.002
5	0.2	8	0.01	0.002	0.265	0.27	0.293	0.287	0.33	0.312	0.326	1.752
				-	-	-	0.001	-	0.001	0.001	-	0.002
5	0.2	8	0.02	0.024	0.487	0.494	0.561	0.549	0.584	0.531	0.82	1.752
				-	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.002
5	0.4	4	0	0	0.945	0.953	1.024	1.193	1.193	1.193	1.193	32.066
				-	0.002	0.003	0.003	0.004	0.004	0.004	0.004	0.027
5	0.4	4	0.01	0	1.662	1.674	1.818	2	2.616	2.629	1.983	32.066
				-	0.003	0.004	0.005	0.005	0.007	0.007	0.005	0.027
5	0.4	4	0.02	0.005	2.678	2.694	2.965	3.175	4.205	3.933	3.24	32.066
				-	0.005	0.006	0.007	0.008	0.011	0.01	0.008	0.027
5	0.4	8	0	0	0.384	0.395	0.434	0.476	0.476	0.476	0.476	32.115
				-	0.001	0.001	0.001	0.002	0.002	0.002	0.002	0.026
5	0.4	8	0.01	0	0.887	0.905	1.028	1.067	1.477	1.472	1.072	32.115
				-	0.001	0.002	0.003	0.003	0.004	0.004	0.003	0.026
5	0.4	8	0.02	0.008	1.653	1.68	1.984	2.001	2.761	2.49	2.221	32.115
				-	0.002	0.004	0.005	0.005	0.007	0.007	0.005	0.026

Mesh-HB estimator) seem to be associated with higher K values. This behavior will be seen more clearly later in other results (see section 7.2.2).

- The high biased mesh estimator (Mesh-HB) is also within $\delta = 0.05$ of the smallest high biased estimate of the risk function value, for each setup considered. In particular, the mesh based policy is the only one that performs well in the in high risk aversion and high volatility case ($\gamma = 5, \sigma = 40\%$). All the other heuristics considered for that setup achieve a risk function value that is higher than $\delta = 0.05$ from the smallest high biased estimate.
- Local hedging (Local) and its approximation (Local-A) do very well overall (except for $\gamma = 5, \sigma = 40\%$), and are the most consistent performer among the heuristics. As expected from the discussion in section 4.2.3, their performances are better when costs are low and are poorer when costs are higher ($b = 2\%$).
- As for the remaining heuristics (BSM, WW and Z), note first that they are all the same, by definition, in the absence of costs ($b = 0$). The BSM policy is nearly optimal when costs are low and risk aversion is low. When costs are higher, it tends to do poorly, which is expected, especially for high numbers of steps K . The WW and Z estimators do generally well when there are costs, except for the high risk aversion case $\gamma = 5$. The performances of these policies seem poor when both γ and the volatility level σ are higher.

Table 5.II provides results for the case of the expOU model, using a similar methodology. Because it is a stochastic volatility model, we can expect the risk estimates to be higher than for the GBM model. There is another important difference, which is that the derivative (call option) values $h_k(Y_k)$ are estimated using mesh-based conditional expectations, instead of evaluated by an analytical formula. This results in additional “risk” introduced in the derivative price as explained in section 3.2.4 and also contributes to higher risk estimates compared to those of table 5.I.

A similar effect was observed in results (not shown here) for the GBM model using such a mesh-based pricing for the derivative price. Finally, note that much higher values of the approximation error bound estimates $\hat{\epsilon}_0$ are observed, especially when the risk aversion γ or the volatility σ are high.

5.5 Conclusion

In this chapter, we explained in some detail how the numerical experiments were performed through the *MeshHedging* Java library and the risk estimation algorithm from section 5.2. We also provided some numerical illustrations to the theory developed so far. Overall, numerical results indicate that approximate solutions to the hedging problem based on stochastic methods does provide useful information about the optimal risk associated with different setups via the Mesh-LB and Mesh-HB estimators. In particular, we noted some setups under which the performance of some known heuristics can be improved upon. For example, the *Z* and *WW* heuristics have a poorer performance when the risk aversion γ or the volatility σ are higher. However, there are some setups that are more difficult for the stochastic mesh based policy:

- Higher number of time steps K : all else being equal, when increasing K , the risk estimates become lower, but the gap between Mesh-LB and Mesh-HB becomes larger, indicating a larger bias of the Mesh-LB and/or Mesh-HB estimators.
- Using derivative values estimated on the mesh adds noise, which introduces an upward bias on risk estimates.

Increasing the number N of paths when building the mesh would reduce these biases, but would also lead to a steep increase in computation time, as discussed in

Table 5.II: Estimates of the portfolio risk when hedging a call option under the exponential loss function and the discrete expOU model for various combinations of the parameters γ , κ , σ_v and ρ (with $N = 512$, $K = 8$, $\sigma_0 = 40\%$, $\bar{\sigma} = 20\%$ and $b = 2\%$).

γ	κ	σ_v	ρ	$\hat{\epsilon}_0$	Mesh-LB	Mesh-HB	Local	Z	WW	BSM	NH
1	2.6	0.6	-0.5	0.1E+02	0.233	0.261	0.274	0.267	0.268	0.367	0.64
				6.2E+00	0.002	0.003	0.002	0.003	0.003	0.003	
1	2.6	0.6	0	0.9E+04	0.241	0.267	0.284	0.275	0.276	0.372	0.693
				0.6E+04	0.002	0.003	0.003	0.003	0.003	0.003	0.003
1	2.6	0.6	0.5	0.3E+08	0.243	0.272	0.294	0.28	0.282	0.374	0.742
				0.3E+08	0.003	0.003	0.003	0.003	0.003	0.003	0.003
1	2.6	1.2	-0.5	7.8E+16	0.262	0.316	0.319	0.3	0.307	0.413	0.697
				7.8E+16	0.002	0.003	0.003	0.003	0.003	0.003	0.003
1	2.6	1.2	0	0.6E+54	0.287	0.322	0.337	0.324	0.331	0.431	0.824
				0.6E+54	0.003	0.003	0.003	0.003	0.003	0.004	0.005
1	2.6	1.2	0.5	7.8E+34	0.293	0.335	0.367	0.337	0.347	0.436	0.932
				7.8E+34	0.004	0.004	0.006	0.004	0.004	0.005	0.006
1	5.2	0.6	-0.5	0.8E+02	0.213	0.236	0.247	0.239	0.247	0.353	0.517
				0.5E+02	0.002	0.002	0.002	0.002	0.002	0.002	0.002
1	5.2	0.6	0	5.7E+02	0.218	0.239	0.252	0.244	0.251	0.356	0.548
				5.5E+02	0.002	0.002	0.002	0.002	0.002	0.003	0.003
1	5.2	0.6	0.5	0.4E+04	0.219	0.242	0.259	0.247	0.254	0.357	0.576
				0.4E+04	0.002	0.002	0.002	0.003	0.003	0.003	0.003
1	5.2	1.2	-0.5	0.3E+02	0.225	0.252	0.261	0.254	0.265	0.375	0.542
				0.2E+02	0.002	0.002	0.002	0.002	0.002	0.002	0.002
1	5.2	1.2	0	1.6E+02	0.237	0.259	0.273	0.265	0.275	0.383	0.611
				1.4E+02	0.002	0.003	0.003	0.003	0.003	0.003	0.003
1	5.2	1.2	0.5	2.2E+10	0.24	0.266	0.289	0.271	0.283	0.385	0.67
				2.0E+10	0.002	0.003	0.003	0.003	0.003	0.003	0.003
5	2.6	0.6	-0.5	3.6E+78	1.182	1.243	1.41	1.808	1.604	1.768	10.102
				3.6E+78	0.014	0.016	0.017	0.022	0.019	0.021	0.021
5	2.6	0.6	0	8.0E+78	1.256	1.318	1.485	1.859	1.657	1.851	11.063
				8.0E+78	0.016	0.018	0.02	0.025	0.022	0.025	0.025
5	2.6	0.6	0.5	7.8E+68	1.273	1.342	1.511	1.849	1.658	1.853	11.909
				7.8E+68	0.018	0.02	0.023	0.027	0.025	0.027	0.027
5	2.6	1.2	-0.5	4.7E+80	1.693	1.818	2.391	2.573	2.341	2.736	13.174
				4.7E+80	0.019	0.027	0.333	0.031	0.028	0.033	0.15
5	2.6	1.2	0	0.2E+184	2.025	2.12	2.332	2.878	2.637	3.174	16.424
				NaN	0.029	0.033	0.036	0.043	0.04	0.047	0.237
5	2.6	1.2	0.5	0.5E+232	NaN	2.241	2.462	2.916	2.706	3.231	19.468
				NaN	NaN	0.051	0.052	0.061	0.057	0.067	0.389
5	5.2	0.6	-0.5	7.1E+62	0.935	0.978	1.074	1.354	1.201	1.43	6.12
				7.1E+62	0.01	0.011	0.012	0.015	0.014	0.016	0.061
5	5.2	0.6	0	0.4E+56	0.964	1.005	1.099	1.365	1.213	1.462	6.474
				0.4E+56	0.011	0.012	0.013	0.016	0.015	0.017	0.069
5	5.2	0.6	0.5	0.4E+50	0.971	1.017	1.11	1.357	1.21	1.466	6.782
				0.4E+50	0.012	0.013	0.014	0.017	0.016	0.018	0.077
5	5.2	1.2	-0.5	0.1E+66	1.122	1.175	1.289	1.639	1.483	1.818	7.106
				0.1E+66	0.012	0.013	0.014	0.018	0.016	0.02	0.071
5	5.2	1.2	0	0.5E+86	1.223	1.274	1.382	1.705	1.546	1.949	8.083
				0.5E+86	0.015	0.016	0.018	0.021	0.02	0.024	0.093
5	5.2	1.2	0.5	0.2E+88	1.235	1.294	1.405	1.683	1.54	1.946	8.872
				0.2E+88	0.017	0.019	0.021	0.025	0.023	0.028	0.12

section 3.5. With these issues in mind, we will examine some efficiency improvement techniques in chapter 6.

CHAPTER 6

EFFICIENCY IMPROVEMENTS

We have mentioned in section 1.3 that computational cost is an important practical problem when using stochastic mesh approximations. It may be difficult to reach a desired level of accuracy ϵ , because of the $O(\epsilon^{1/4})$ convergence rate discussed in section 3.5. As pointed out by Broadie and Glasserman [36], pages 44-45,

”... the stochastic mesh method leaves a lot of latitude in implementation. For the method to be practically viable, it is essential to exploit efficiencies in the computation of the estimators wherever possible. This requires, in particular, careful choice of the density used to generate the mesh. It also motivates the use of control variates, ... ”.

Fu, Laprise, Madan, Su and Wu [57] and Avramidis and Matzinger [8] also indicate that stochastic mesh estimators have large bias and likely requires variance reduction techniques for practical applications.

This chapter investigates the effect of various efficiency improvement techniques on the performance of stochastic mesh methods when applied to hedging problems. We start in section 6.1 by applying control variates to the mesh estimates. To try to address the problem of computational space and time constraints, we propose in section 6.2 using a *single grid* instead of a full mesh. The idea is to recycle the same set of weights at every step $k = 1, \dots, K$, so that the number of weights to store and the number of associated computations is divided by a factor K . We then describe possible applications of quasi-Monte Carlo (QMC) methods in section 6.3, including randomized quasi-Monte Carlo (RQMC). These can potentially speed up the computation, as they have a theoretically faster rate of convergence than standard Monte Carlo (see for example L’Ecuyer [86]).

To further reduce the computation time, we consider in section 6.4 a Russian roulette technique that ignores weights that are too small to influence the result, in a way that we make precise later. Small weights occur naturally on any stochastic mesh, but even more so on a single grid implementation. The performance gains from the single grid and Russian roulette techniques are somewhat complementary, as we will see in section 6.5. This will allow us to compute results for high values of the number of time steps K . Finally, in section 6.6, we briefly describe some other variance reduction techniques that were not yet tested for our hedging problem, but that could be useful to increase the computational efficiency.

6.1 Control variates

Control variates (CVs) are a well known tool to reduce the variance of a Monte Carlo estimator. See for example L'Ecuyer [88], Law and Kelton [84], Glasserman [59] or Szechtman [129] for an introduction, or, for more advanced applications, L'Ecuyer and Buist [90], Lemieux and La [95] or Bolia and Juneja [26]. We consider here some possible applications for the estimation of

- the risk function value $J_0^\pi(X_0)$ associated to a policy π by (out-of-sample) Monte Carlo simulations,
- the derivative values $h_k(Y_k)$ on the stochastic mesh, and
- the expected risk $Q_k(X_k, v)$ via the weighted Monte Carlo average from equation (3.7), i.e.

$$\widehat{Q}_k(X_k, v) = \frac{1}{N} \sum_{j=1}^N w_{k,j} e^{-\gamma(V_{k+1}(\psi(X_k, v, Y_{k+1}^j)) - V_k)} \widehat{J}_{k+1}(v, Y_{k+1}^j). \quad (6.1)$$

6.1.1 Outer controls

When computing $J_0^\pi(X_0)$ by Monte Carlo simulation for a given hedging policy π using the sample average estimator

$$\widehat{J}_0^\pi(X_0) = \frac{1}{N} \sum_{i=1}^N L(V_K^\pi(X_0) - V_0), \quad (6.2)$$

it is in principle straightforward to apply the control variates technique to reduce the variance of $\widehat{J}_0^\pi(X_0)$. One simply needs to find a vector $C \in \mathbb{R}^m$ of random variables whose components are correlated with $L(V_K^\pi(X_0) - V_0)$ and then consider

$$\widehat{J}_0^{\pi,CV}(X_0) = \widehat{J}_0^\pi(X_0) - \beta^{*T}(C - \mathbf{E}_0[C]), \quad (6.3)$$

where

$$\beta^* = \arg \min_{\beta} \text{Var} \left[\widehat{J}_k^\pi(X_0) - \beta^T(C - \mathbf{E}_0[C]) \right]. \quad (6.4)$$

In the case of the negative exponential loss function, one potentially useful control is simply

$$C = V_K^{0,\pi}(X_0), \quad (6.5)$$

which we define as the value of the portfolio at step K after applying the policy π , but assuming that transaction costs are zero. It corresponds (up to a constant) to the linear part of the Taylor series expansion of $L(V_K^{0,\pi}(X_0) - V_0) = e^{-\gamma(V_K^{0,\pi}(X_0) - V_0)}$, so it should be well correlated with the function $L(V_K^\pi(X_0) - V_0)$, especially for small values of γ and small transaction costs. Note that the expectation is known : as discussed in section 2.1.3, if the market process is a martingale, then so is the portfolio value process, and thus $\mathbf{E}_0[V_K^{0,\pi} - V_0] = 0, \forall \pi$. For the present example, setting $\beta = -\gamma$ should always reduce the variance because the correlation between $-\gamma(V_K^\pi(X_0) - V_0)$ and $L(V_K^\pi(X_0) - V_0)$ is non-negative, by the convexity of the function L .

Clewwell and Hodges [40] also make use of a Taylor series expansion of the exponential function to improve computational efficiency, although their solution method (a binomial tree) is different than that used here.

Table 6.I: Empirical variance reduction factors when estimating \widehat{J}_0^π using the control variate $V_K^{0,\pi}$ with the coefficient $\beta = -\gamma$, under the exponential loss function and the discrete GBM model, for various policies π ($N = 256, n = n_R = 100$).

γ	σ	K	b	BSM	WW	Z	Local	Mesh-HB	NH
1	0.2	4	0	28.4	28.4	28.4	5.4	3	1.7
1	0.2	4	0.01	35	27.9	19	10.5	7.5	2.2
1	0.2	4	0.02	17.3	16	8.7	5.3	7.1	2.4
1	0.2	8	0	53.3	53.3	53.3	8	3.8	2.1
1	0.2	8	0.01	17.8	29.5	22.7	10	7.7	1.9
1	0.2	8	0.02	6.8	19.1	11.2	6.3	5.3	2.1
1	0.4	4	0	11.1	11.1	11.1	7.4	5.1	1.7
1	0.4	4	0.01	8.3	6.7	7.1	5.6	4.9	1.1
1	0.4	4	0.02	8.5	8.9	9.3	8	6.8	1.8
1	0.4	8	0	13.1	13.1	13.1	7.9	3.8	1.8
1	0.4	8	0.01	14.1	11.6	11	5.4	3.9	1.3
1	0.4	8	0.02	9.7	10	8	7	6.1	1
5	0.2	4	0	1.2	1.2	1.2	1.2	1.2	1.2
5	0.2	4	0.01	1.1	1.1	1.1	1.1	1.1	1.2
5	0.2	4	0.02	1.1	1.1	1.1	1.1	1.1	1.2
5	0.2	8	0	1.3	1.3	1.3	1.3	1.3	1.2
5	0.2	8	0.01	1.1	1.1	1.1	1.2	1.2	1.2
5	0.2	8	0.02	1.1	1.1	1.1	1.1	1.1	1.2
5	0.4	4	0	1	1	1	1.1	1	1
5	0.4	4	0.01	1	1	1	1	1	1
5	0.4	4	0.02	1	1	1	1	1	1
5	0.4	8	0	1.1	1.1	1.1	1.1	1.1	1
5	0.4	8	0.01	1	1	1	1.1	1	1
5	0.4	8	0.02	1	1	1	1	1	1

Table 6.I shows some variance reduction factors (VRF) observed empirically when applying this control variate for various parameter values for risk aversion γ , proportional transaction costs b , volatility σ , and number of steps K . The results show the

ratio F of the sample variances of the uncontrolled and the controlled estimators, i.e.

$$F = S_{n_R}^2 \left(\widehat{J}_0^\pi \right) / S_{n_R}^2 \left(\widehat{J}_0^{\pi^{CV}} \right), \quad (6.6)$$

where for a random variable $X \in \mathbb{R}$ with samples values $\{X_i : i = 1, \dots, n_R\}$, we write the sample variance as

$$S_{n_R}^2(X) = \frac{1}{n_R - 1} \sum_{i=1}^{n_R} \left(X_i - \frac{1}{n_R} \sum_{j=1}^{n_R} X_j \right)^2. \quad (6.7)$$

The sample variances were based on $n_R = 100$ independent estimates of the risk function. And these risk function estimates $\widehat{J}_0^\pi(X_0)$ were computed as sample averages of $n = 100$ independent simulations of values $L(V_K^\pi(X_0) - V_0)$ of the loss function applied to the terminal portfolio performance, for various policies π . The stochastic mesh policy (Mesh-HB) and the local hedging policy (Local) are both based on a stochastic mesh constructed using $N = 256$ independent sample paths, and an independent mesh was used for each of the n_R replications.

We note that

- the observed VRFs are at least greater than or equal to 1 for each setting,
- the VRFs are (often) larger in the case without costs ($b = 0$),
- the VRFs are much smaller when the risk aversion parameter γ is larger.

These results are not surprising, given the discussion at the beginning of this section about the definition of this control variate. We also observe that the VRFs are larger for the policies BSM, WW and Z. This could be due to the lower variance of the terminal portfolio value V_K^π when applying these policies, compared to Local, Mesh or NH. The Local and Mesh policies have some added noise due to their being computed based on a stochastic mesh, and in the case of NH, there is no hedging at

all, by definition. This is coherent with the poorer performance observed when the volatility parameter σ is higher.

6.1.2 Inner controls

This section focuses on applying CV techniques specifically for reducing the variance of the mesh estimators. This is what Broadie and Glasserman [36] refer to as *inner controls*.

6.1.2.1 Control for \widehat{h}_k

Applying CV techniques to the estimates $\widehat{h}_k(Y_k)$ of the derivative value evaluated on the stochastic mesh can help to substantially reduce the bias in $\widehat{Q}_k(X_k, v)$, as discussed in section 3.2.4. Following Broadie and Glasserman [36] (section 4.1), we consider the controlled estimator

$$\widehat{h}_k^{CV}(Y_k) = \frac{\frac{1}{N} \sum_{j=1}^N w_{k,j} \left(\widehat{h}_{k+1}^j - \beta^T C_{k+1}^j \right)}{\frac{1}{N} \sum_{j=1}^N w_{k,j}}, \quad (6.8)$$

where the $\widehat{h}_{k+1}^j := \widehat{h}_{k+1}(Y_{k+1}^j)$ are mesh based estimates of the derivative value and the $C_{k+1}^j \in \mathbb{R}^m$ are sample realizations of the control vector C , corresponding to the market state Y_{k+1}^j . The vector $\beta \in \mathbb{R}^m$ is then chosen to be the solution $\widehat{\beta}_N$ of the weighted least-squares problem

$$\min_{\alpha, \beta} \sum_{j=1}^N w_{k,j} \left(\widehat{h}_{k+1}^j - (\alpha + \beta^T C_{k+1}^j) \right)^2, \quad (6.9)$$

which is given by

$$\widehat{\beta}_N = \left(\frac{1}{N} \sum_{j=1}^N w_{k,j} C_{k+1}^j (C_{k+1}^j)^T \right)^{-1} \left(\frac{1}{N} \sum_{j=1}^N w_{k,j} \widehat{h}_{k+1}^j C_{k+1}^j \right). \quad (6.10)$$

The resulting controlled estimate $\widehat{h}_k^{CV}(Y_k)$ of $h_k(Y_k)$ from equation (6.8) may have some bias because $\widehat{\beta}_N$ may be correlated with the C_{k+1}^j . But the solution $\widehat{\beta}_N$ is known to be an asymptotically consistent estimator of β^* (see for example L'Ecuyer [88], section 6.5.3). Therefore, the estimator \widehat{h}_k^{CV} will also be asymptotically consistent (see Glynn and Szechtman [62], theorem 1).

However, the error on individual estimates of $\widehat{h}_k(Y_k)$ gets magnified by the exponential loss function, as seen in section 3.2.4. This can be particularly problematic in cases where there are large outliers in the sampled $\widehat{\beta}_N$ values. In numerical experiments not shown here, this affected the stochastic mesh estimates of the portfolio risk $J_0^*(X_0)$ in the case of the expOU model, but not the GBM model (both using the AD sampling procedure of section 3.2.3).

Fortunately, the theoretical solution $\beta = \beta^*$ may be known in some cases, thus eliminating the problem of outliers in the estimator $\widehat{\beta}_N$. For example, consider $\mathbf{E}_k[h_{k+1}(Y_{k+1})|\mathcal{H}_{k+1}]$, the conditional expectation of the derivative value on some sub- σ -algebra $\mathcal{H}_{k+1} \subset \mathcal{F}_{k+1}$. We can take $C_{k+1} = h_{k+1}(Y_{k+1}) - \mathbf{E}_k[h_{k+1}(Y_{k+1})|\mathcal{H}_{k+1}]$ as a control variate. In that case, clearly $\mathbf{E}_k[C_{k+1}] = 0$ and the optimal β coefficient can be shown to be simply $\beta^* = 1$ (see Theorem 2 of Glynn and Szechtman [62]). When using the expOU model, where the σ -algebra \mathcal{F}_{k+1} is generated by the random vector $Y_{k+1}(\omega) = (s_{k+1}(\omega), \sigma_{k+1}(\omega))^T$, one possibility is to take \mathcal{H}_{k+1} to be the σ -algebra generated by the random variable $\sigma_{k+1}(\omega)$ (the value of the volatility parameter at step $k+1$). Since the coefficient β^* is a constant, the resulting controlled estimator \widehat{h}_k^{CV} of h_k is clearly unbiased.

Table 6.II shows various estimates $\widehat{J}_0^\pi(X_0)$ of the portfolio risk under the expOU model, using the above control variate when estimating the derivative values $h_k(Y_k)$ over the mesh nodes. The setup is the same as for table 5.II seen in section 5.4, which used only a sample average estimator for the derivative values $h_k(Y_k)$. Overall, the estimates $\widehat{J}_0^\pi(X_0)$ have smaller (average) values and also have reduced sample

standard deviation, as should be expected. The approximation error bound estimates $\widehat{\epsilon}_0$ are now much smaller, which indicates that the previously high values observed were due to the error in the $h_k(Y_k)$ estimates. These improvements are more apparent when the risk aversion parameter γ is higher (i.e., for $\gamma = 5$).

Note that using linear controls can be interpreted as modifying the weights of the simulated paths in the standard Monte Carlo average estimator, as explained in Glasserman [59] (section 4.2). In the case of the controlled estimator (6.8), it is not hard to show that

$$\widehat{h}_k^{CV}(Y_k) = \frac{1}{N} \sum_{j=1}^N \widetilde{w}_{k,j} \widehat{h}_{k+1}^j, \quad (6.11)$$

with

$$\widetilde{w}_{k,j} := w_{k,j} \left(1 - \left(\frac{1}{N} \sum_{i=1}^N C_{k+1}^i \right)^T \left(\frac{1}{N} \sum_{i=1}^N C_{k+1}^i (C_{k+1}^i)^T \right)^{-1} C_{k+1}^j \right). \quad (6.12)$$

In the context of a stochastic mesh, we could apply the method by modifying the weights once (at each state) and then reuse the modified weights in future computations. The overall impact on other mesh estimators such as $\widehat{Q}_k(X_k, v)$ should be positive, in the sense that the variance of the estimator should be reduced. Preliminary experiments (not shown here) did not show a large improvement, but this would deserve further study.

6.1.2.2 Control for \widehat{Q}_k

We would also like to apply CV techniques when computing $\widehat{Q}_k(X_k, v)$ at each step $k = K - 1, \dots, 0$ of the construction of the stochastic mesh based policy. The

Table 6.II: Estimates $\hat{J}_0^\pi(X_0)$ of the portfolio risk when hedging a call option. The setup is the same as for table 5.II, but this time using control variates for estimating h_k values.

γ	κ	σ_v	ρ	$\hat{\epsilon}_0$	M-LB	M-HB	LOC	LOC-A	Z	WW	BSM	NH
1	2.6	0.6	-0.5	0.002 -	0.236 -	0.253 -	0.278 0.001	0.282 0.001	0.265 0.001	0.266 0.001	0.364 -	0.637 0.001
1	2.6	0.6	0	0.002 -	0.242 -	0.258 -	0.274 0.001	0.282 0.001	0.272 0.001	0.272 0.001	0.368 -	0.689 0.001
1	2.6	0.6	0.5	0.003 -	0.244 -	0.262 -	0.271 0.001	0.28 0.001	0.276 0.001	0.277 0.001	0.369 -	0.736 0.001
1	2.6	1.2	-0.5	0.002 -	0.279 0.001	0.325 0.001	0.367 0.001	0.321 0.001	0.311 0.001	0.318 0.001	0.424 0.001	0.712 0.001
1	2.6	1.2	0	0.002 -	0.291 0.001	0.316 0.001	0.338 0.001	0.328 0.001	0.323 0.001	0.329 0.001	0.429 0.001	0.822 0.001
1	2.6	1.2	0.5	0.003 -	0.3 0.001	0.328 0.001	0.339 0.001	0.336 0.001	0.338 0.001	0.347 0.001	0.436 0.001	0.933 0.001
1	5.2	0.6	-0.5	0.003 -	0.216 -	0.23 -	0.249 0.001	0.259 0.001	0.238 0.001	0.245 -	0.351 -	0.516 0.001
1	5.2	0.6	0	0.003 -	0.219 -	0.233 -	0.245 0.001	0.257 0.001	0.242 0.001	0.248 -	0.353 -	0.546 0.001
1	5.2	0.6	0.5	0.003 -	0.22 -	0.236 -	0.242 0.001	0.255 0.001	0.244 0.001	0.251 -	0.354 -	0.573 0.001
1	5.2	1.2	-0.5	0.003 -	0.232 -	0.249 -	0.271 0.001	0.271 0.001	0.256 0.001	0.266 -	0.376 -	0.544 0.001
1	5.2	1.2	0	0.003 -	0.238 -	0.252 -	0.265 0.001	0.273 0.001	0.262 0.001	0.272 0.001	0.379 -	0.607 0.001
1	5.2	1.2	0.5	0.003 -	0.242 -	0.26 -	0.264 0.001	0.274 0.001	0.269 0.001	0.281 0.001	0.383 -	0.667 0.001
5	2.6	0.6	-0.5	0.011 -	1.125 0.001	1.151 0.002	1.278 0.003	1.381 0.003	1.689 0.004	1.498 0.003	1.655 0.003	9.519 0.009
5	2.6	0.6	0	0.012 -	1.181 0.002	1.206 0.002	1.334 0.003	1.426 0.003	1.717 0.004	1.529 0.003	1.712 0.003	10.305 0.01
5	2.6	0.6	0.5	0.012 -	1.18 0.002	1.209 0.002	1.363 0.003	1.408 0.003	1.683 0.004	1.508 0.003	1.689 0.003	10.944 0.01
5	2.6	1.2	-0.5	0.015 -	1.717 0.003	916.159 914.277	191.136 189.054	2.177 0.005	2.555 0.006	2.326 0.006	2.723 0.006	13.129 0.019
5	2.6	1.2	0	0.045 0.029	1.914 0.003	1.956 0.005	2.166 0.006	2.343 0.006	2.672 0.006	2.448 0.006	2.951 0.006	15.326 0.019
5	2.6	1.2	0.5	0.016 -	1.937 0.003	1.986 0.005	2.272 0.006	2.318 0.006	2.622 0.006	2.432 0.006	2.908 0.006	17.598 0.023
5	5.2	0.6	-0.5	0.014 -	0.897 0.001	0.916 0.002	1.006 0.002	1.099 0.002	1.279 0.003	1.134 0.002	1.353 0.002	5.825 0.006
5	5.2	0.6	0	0.014 -	0.918 0.001	0.936 0.002	1.026 0.002	1.113 0.002	1.281 0.003	1.138 0.002	1.375 0.002	6.126 0.006
5	5.2	0.6	0.5	0.014 -	0.919 0.001	0.939 0.002	1.043 0.002	1.106 0.002	1.263 0.003	1.126 0.002	1.369 0.002	6.375 0.006
5	5.2	1.2	-0.5	0.015 -	1.097 0.002	1.123 0.002	1.249 0.003	1.378 0.003	1.575 0.003	1.425 0.003	1.751 0.003	6.872 0.008
5	5.2	1.2	0	0.015 -	1.161 0.002	1.183 0.002	1.308 0.003	1.425 0.003	1.593 0.003	1.444 0.003	1.825 0.003	7.612 0.008
5	5.2	1.2	0.5	0.015 -	1.164 0.002	1.191 0.002	1.344 0.003	1.406 0.003	1.558 0.003	1.425 0.003	1.805 0.003	8.277 0.008

usual controlled estimator $\widehat{Q}_k^{CV}(X_k, v)$ of $\widehat{Q}_k(X_k, v)$ would be

$$\widehat{Q}_k^{CV}(X_k, v) = \left(\frac{1}{N} \sum_{j=1}^N w_{k,j} e^{-\gamma(V_{k+1}(X_{k+1}^j) - V_k)} \widehat{J}_{k+1}(v, Y_{k+1}^j) \right) - \beta^T C,$$

where $X_{k+1}^j = \psi(X_k, v, Y_{k+1}^j)$ and $C \in \mathbb{R}^m$ is a random vector, assumed without loss of generality to have known expectation $\mathbf{E}_k^P[C] = (0, \dots, 0)^T \in \mathbb{R}^m$ (i.e., a zero vector). However, it is not obvious in general to find a control C that is well correlated to $w_{k,j} e^{-\gamma(V_{k+1}(X_{k+1}^j) - V_k)} \widehat{J}_{k+1}(v, Y_{k+1}^j)$. For the hedging problem, we have the added difficulty that the $\widehat{J}_{k+1}(v, Y_{k+1}^j)$ depend on the stock quantity v . Thus computing an optimal β for a fixed value $v = v^0$ may reduce the variance around v^0 , but increase it for other values.

For such a setup, where the estimator $\widehat{Q}_k(X_k, v)$ and the control $C(v)$ are both considered as functions of another variable v , the optimal β should also be a function of v . As described in L'Ecuyer and Buist [90], the optimal value $\beta^*(v)$ could be approximated as the ratio $q_1(v)/q_2(v)$, where q_1, q_2 are two functions, which in our context would be given by $q_1(v) = \mathbf{E}[C(v)\widehat{Q}_k(X_k, v)]$ and $q_2(v) = \mathbf{E}[C(v)^2]$. These two functions could be obtained as approximations, based on sample realizations of $C(v)$ and $\widehat{Q}_k(X_k, v)$, for m values of the variable v , and using a fitting procedure, such as smoothing splines (see de Boor [48]).

With these issues in mind, we now consider as a concrete example the following controlled estimator

$$\widehat{Q}_k^{CV}(X_k, v) = \frac{1}{N} \sum_{j=1}^N w_{k,j} \left(e^{-\gamma(V_{k+1}(X_{k+1}^j) - V_k)} - \beta C(v) \right) \widehat{J}_{k+1}(v, Y_{k+1}^j), \quad (6.13)$$

with

$$C(v) = V_{k+1}(X_{k+1}^j) - V_k = v \Delta s_{k+1} - \Delta h_{k+1}, \quad (6.14)$$

where v denotes the new stock quantity at step $k + 1$ (the decision). In numerical experiments described below, we evaluate it for a fixed $\beta = -\gamma$. It corresponds to the linear part of the Taylor series expansion of $e^{-\gamma\Delta(V_{k+1}(v, Y_{k+1}^j) - V_0)}$, so it should be well correlated to it, especially for small values of γ . Note that the expectation of $C(v)$ is known, with $\mathbb{E}_k[C(v)] = 0, \forall v$, but that the resulting controlled estimator \widehat{Q}_k^{CV} may be biased, since $C(v)$ could be correlated with $\widehat{J}_{k+1}(v, Y_{k+1}^j)$.

Figures 6.1 and 6.2 illustrate what can happen in practice using such a control variate. Both use a similar setup involving the expOU model, the only difference being in the value for the risk aversion parameter γ . The convergence of the controlled estimators (AD-CV) is clearly faster as a function of the number of mesh nodes N , both in sample and out-of-sample. However, in figure 6.2 the controlled estimators tend to a higher value than the standard estimators (for the values of N tested here). Even worse, the inner mesh estimator (dashed red curve) crosses over the (always) high biased outer mesh estimator (full red curve), which shows that the inner mesh estimator is no longer low biased. This behavior is also apparent in the $\gamma = 1$ case, but to a lesser extent.

We have not tried the more general approach described earlier, taking $\beta^*(v) = q_1(v)/q_2(v)$, but it would likely still suffer from the bias issue and from the fact that $C(v)$ should only be correlated to $\widehat{J}_{k+1}(X_{k+1})$ for low values of γ .

6.1.2.3 Concluding remarks on inner controls

As we have just seen, inner control variates can sometimes prove useful (e.g. to estimate $\widehat{h}_k(Y_k)$, as in section 6.1.2.1), but they can also have adverse effects (e.g., as in section 6.1.2.2). When they can help reduce the variance of mesh based estimators $\widehat{Q}_k(X_k, v)$ of the $Q_k(X_k, v)$ without introducing new bias, they help reduce the low bias from the mesh estimator $\widehat{J}_0(X_0)$ of the risk. This is because the optimization step minimizes $\widehat{Q}_k(X_k, \cdot)$ for a finite number of states X_k , so the optimal decisions

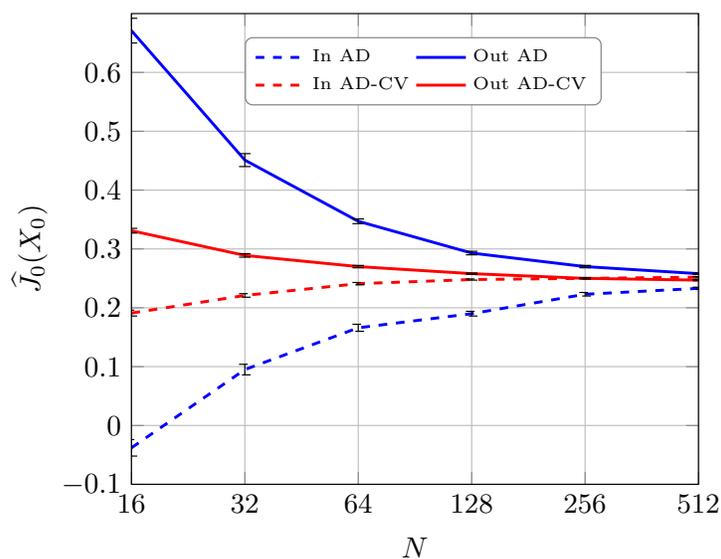


Figure 6.1: Inner and outer risk estimates $\hat{J}_0(X_0)$ as a function of N , using control variates, for a call option under the expOU model ($\sigma_0 = 40\%$, $\bar{\sigma} = 20\%$, $\sigma_v = 1.2$, $\kappa = 2.6$, $\rho = -0.5$, $b = 2\%$, $\gamma = 1$, $K = 8$). Error bars indicate one standard deviation.

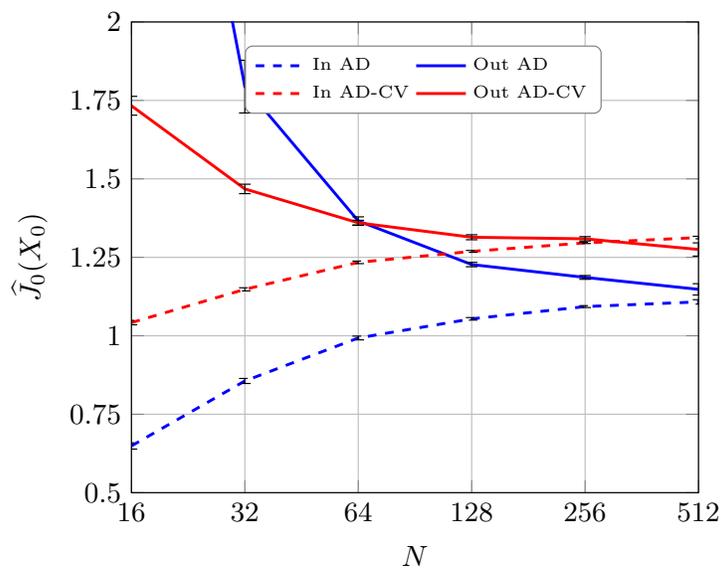


Figure 6.2: Inner and outer risk estimates \hat{J}_0 as a function of N , using control variates, for a call option under the expOU model ($\sigma_0 = 40\%$, $\bar{\sigma} = 20\%$, $\sigma_v = 1.2$, $\kappa = 2.6$, $\rho = -0.5$, $b = 2\%$, $\gamma = 5$, $K = 8$). Error bars indicate one standard deviation.

focus on these states, at the expense of out-of-sample performance. Another way to see this is that return expectations obtained when using mesh based weighted averages are no longer zero, which can then be directly exploited if one is trying to minimize a risk function. This was already pointed out in other contexts by Klaassen [80] or Duan and Simonato [52], for example. For a severe example of this behavior in high dimensions, see also Birge [20].

6.2 Using a single grid

When applying the stochastic mesh method, many similar computations must be repeated, such as computing weights at every step k . Since this takes a lot of time, we would like to reduce the number of such computations. Assuming that the transition density $f_k(Y_k, Y_{k+1})$ from Y_k to Y_{k+1} is independent of the time step k , then we can write $f_k(Y_k, Y_{k+1}) = f(Y_k, Y_{k+1})$ for some fixed density function $f : \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \infty)$. This is the case for the discrete time version expOU model of section 4.1, using constant time steps $\Delta t_{k+1} = \Delta t = T/K, \forall k = 0, \dots, K-1$. So it is possible to define a mesh using only one set of states $\mathcal{G}^N := \{Y^i : i = 1, \dots, N\}$ for all time steps, i.e. $\mathcal{G}^N = \mathcal{G}^N, \forall k = 1, \dots, K$, and to use step independent weights $w_{k,i,j} = w_{i,j}, \forall k$.

This is similar to the stochastic mesh-type method introduced by Rust [117] (independently from Broadie and Glasserman [35]) : the *random Bellman operator method*. Rust showed that a randomized algorithm with polynomial computational complexity can solve (within a given error ϵ) a class of Markov decision problems called *discrete decision processes*. A precise definition can be found in [117], but here we note that the term *discrete* refers to the decision space \mathcal{U} being a finite set, whereas the state space is assumed continuous. In the terminology of this thesis, the stochastic mesh used is based on a single random grid, defined by N points $\{U_i : i = 1, \dots, N\}$ sampled uniformly from the unit hypercube $[0, 1]^d$. The objective function at every step is computed recursively over this grid, using the DP algorithm.

The weights used to compute the conditional expectations are Markovian transition densities $p(X_{k+1}|X_k, v)$, which depend on the decision $v \in \mathcal{U}$. For a given decision v , these densities can be interpreted as likelihood ratio weights, since the unconditional density used to generate the points of the grid \mathcal{G}^N is the uniform density, i.e. the function $g_{k+1}(Y)$ in equation (3.11) is a constant. For our hedging problem, the setup is different, as we require the decision space \mathcal{U} to be continuous. Furthermore, we focus on a multiplicative risk function (exponential utility), whereas the risk function in [117] is additive. So the conclusions about the polynomial computational complexity do not carry over directly to our setting, and we do not attempt to adapt the relevant proofs.

The downside to using a single grid is that only a small fraction of the grid points are likely to be reached, especially for the early stages (i.e., for small k). Intuitively, mesh weights $w_{k,i,j}$ will be very close to zero if and only if the corresponding market vectors Y_k^i and Y_{k+1}^j are far apart. More concretely, suppose the weights were either LR or AD weights, as given by equations (3.11) and (3.12) respectively. Then, for given destination market vectors Y_{k+1}^j ($j = 1, \dots, N$), the weights will be close to zero only if the density $f_k(Y_k^i, Y_{k+1}^j)$ is close to zero. Now assume that the market vector process $\{Y_k : k = 0, \dots, K\}$ is such that the random vector

$$r(Y_k, Y_{k+1}) := (\ln(Y_{k+1,1}/Y_{k,1}), \dots, \ln(Y_{k+1,d}/Y_{k,d}))$$

follows a d -dimensional normal distribution $r(Y_k, Y_{k+1}) \sim N(\mu, \Sigma)$, for some $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^d \times \mathbb{R}^d$. Then, the sampled market vectors Y_k^i and Y_{k+1}^j can be considered far apart if and only if the distance $\delta_{i,j}$ defined by

$$\delta_{i,j} := (X_{i,j} - \mu)^t \Sigma^{-1} (X_{i,j} - \mu), \text{ where } X_{i,j} = r(Y_k^i, Y_{k+1}^j),$$

is large, since the multinormal density function is then proportional to $e^{-\delta_{i,j}/2}$. The

two dimensional expOU process described in section 4.1 has $Y_k^i = (s_k^i, \sigma_k^i)^t$ and is an example of such a process with lognormal increments. In the case of the GBM process, we have simply $Y_k^i = s_k^i$ and the distance $\delta_{i,j}$ can be written explicitly as

$$\delta_{i,j} = \frac{(\ln(s_{k+1}^j/s_k^i) - \mu)^2}{\sigma^2(t_{k+1} - t_k)}, \text{ where } \mu = (r - q - \frac{1}{2}\sigma^2)(t_{k+1} - t_k).$$

To apply the single grid idea to our problem, we use a modified version that aims to increase the likelihood that the mesh points come from paths simulated from our stock price model. It is defined by considering the unconditional distribution F_K of the market information vector Y_k at time $t_K = T$ and defining the grid points $Y^i \in \mathcal{G}^N$ by inversion. More explicitly, if $Y^i = g(Z^{i,1}, \dots, Z^{i,d})$ for some function $g : \mathbb{R}^d \rightarrow \mathbb{R}$ and independent $N(0, 1)$ variates $Z^{i,j}$, then we generate the $Z^{i,j}$ as $Z^{i,j} = \Phi^{-1}(U^{i,j})$, where Φ is the standard normal cumulative distribution function and $U^{i,j} \sim U(0, 1)$, for $i = 1, \dots, N$ and $j = 1, \dots, d$. The distribution for step $k = K$ is used instead of some other $k < K$ since it guarantees that we will sample points in a wide enough range to cover likely paths of the process. The weights are defined using equation (3.11) with $g_{k+1}(Y) = g(Y)$ taken as the density associated to the distribution $F_K(Y)$. As we will discuss in section 6.3, this is also closely related to the mesh generation method from Boyle, Kolkiewicz and Tan [31, 32].

The distribution function F_K is known explicitly in some cases of interest, such as for the lognormal distribution, but not for the expOU model of section 4.1, where the market vector $Y_k = (s_k, \sigma_k)^T$ is defined by equations (4.3) and (4.4) over many steps. So as a crude approximation, we use the distribution corresponding to only one step and apply it over the entire horizon $\Delta t = T$, instead of $\Delta t = T/K$ (i.e., we use the distribution for the case $K = 1$).

Thus by construction, $Y_{k+1}^i = Y_k^i$, $\forall i = 1, \dots, N$ and $\forall k = 1, \dots, K - 1$, so clearly those two vectors are not independent. This could lead the associated weight to

dominate all other weights and introduce some bias in the mesh based estimates. To avoid such problems, we set

$$w_{i,j} = 0, \text{ if } i = j. \quad (6.15)$$

Finally, we normalize the weights as

$$w_{i,j}^N := \frac{w_{i,j}}{\frac{1}{N} \sum_{j=1}^N w_{i,j}}, \quad (6.16)$$

so that the sum of weights starting from a given state Y_i will always equal one. This reduces the variance of the mesh based estimators, at the expense of introducing some small bias.

6.3 Quasi-Monte Carlo methods

Recall that standard Monte Carlo (MC) is used to estimate d -dimensional integrals of the form

$$I = \int_{[0,1]^d} f(U) dU \quad (6.17)$$

as

$$\hat{I} = \frac{1}{n} \sum_{i=0}^{n-1} f(U^i) \quad (6.18)$$

using (pseudo-)random numbers, whereas quasi-Monte Carlo (QMC) replaces the U^i 's by the elements u_i of a deterministic set $P_n = \{u_0, \dots, u_{n-1}\}$ which covers the unit cube $[0, 1]^d$ more uniformly than would random points. See for example L'Ecuyer [86] and the references therein. The performance of QMC depends on the dimension d of the problem and the smoothness of the function f to be integrated, but good results have been obtained in financial applications (see L'Ecuyer [86], [88], Glasserman [59]). Therefore it is natural to try to use QMC methods instead of standard MC when constructing a stochastic mesh to estimate solutions to our

hedging problem.

In the case of American option pricing, this has already been done by Boyle, Kolkiewicz and Tan [31, 32], who described a way to generate a stochastic mesh using a low-discrepancy mesh (LDM) method. As explained in more detail in an associated research report [30], it involves fitting the parameters of a log-normal distribution F_k at each step $k = 1, \dots, K$ to match the mixture of log-normal distributions that follows from the average density construction described in section 3.2.3. The grid points Y_k^i are then generated by inversion using elements of a low discrepancy sequence for the u^i instead of pseudo-random numbers.

Here we will test the application of QMC methods in a similar way, in that we generate a single grid of points \mathcal{G}^N (as in section 6.2), using the inversion method on a QMC point set P_n , where $n = N$, the number of points in the grid. However, our implementation differs from [32] in that we use *only one* grid instead of K grids and do not use a fitting procedure. Thus we take the dimension d of the QMC points to be $d = m$, where m is the dimension of the space \mathcal{Y} of market vectors, which is much lower than the mK dimensions required if the mesh points were generated by independent path simulations. However, as explained in section 6.2, using a single grid implies that a relatively large proportion of grid points may be unreachable or have a very small weight $w_{k,j}$, especially for the early steps (i.e., for $k \ll K$). The method is otherwise well defined and yields unbiased estimates, as before.

As opposed to standard MC, QMC estimates are deterministic, so it not possible to measure the approximation error using the standard deviation of the estimates. However, this can be addressed by using a randomized version of quasi-Monte Carlo (RQMC), as explained in L'Ecuyer [86], for example. See also Ben Hameur, L'Ecuyer and Lemieux [11], who give examples where combining RQMC, control variates and other techniques can reduce the variance of basic MC estimators by large factors.

A general algorithm for applying RQMC to the single grid method is given by

algorithm 5 below. In our experiments, we apply RQMC in the following way: We choose the QMC point set P_n to be defined by the first $n = N$ points of a Sobol sequence [126] in dimension d . The explicit definition of the Sobol sequences depends on a set of parameters called *direction numbers* (see for example L’Ecuyer [88] for more details). In the Java programs used to compute the results, the point set P_n is generated using the class `SobolSequence` from the *SSJ* Java library [89], which in turn uses as default direction numbers the values given in Lemieux, Cieslak, and Luttmner [96]. The randomization is provided by a *random shift modulo 1* (see L’Ecuyer [88]), where a single point U is generated uniformly over the unit hypercube $[0, 1)^d$ and is added to each point of P_n coordinate-wise, modulo 1. That is, writing $U = (U_0, \dots, U_{d-1})$, the shifted version \tilde{u}_i of a point $u_i \in P_n$ is defined by

$$\tilde{u}_i = (\tilde{u}_{i,0}, \dots, \tilde{u}_{i,d-1}), \text{ where } \tilde{u}_{i,j} = u_{i,j} + U_j \pmod{1}, \text{ for } j = 0, \dots, d-1. \quad (6.19)$$

Algorithm 5 Single grid with RQMC

- 1 Generate a QMC point set $P_n = \{u_i \in [0, 1)^d, i = 0, \dots, n\}$ in d dimensions.
 - 2 For $j = 1$ to n_R replications
 - a: Generate uniformly a random point $U \in [0, 1)^d$ and use it to randomize P_n as $\tilde{P}_n \leftarrow P_n + U \pmod{1}$.
 - b: Use the (randomized) point set \tilde{P}_n to generate the states $Y_i = F_K^{-1}(\tilde{u}_i)$, with $\tilde{u}_i \in \tilde{P}_n$, for $i = 1, \dots, N$.
 - c: Compute associated mesh weights $w_{i,j}$.
 - d: Apply the DP algorithm to estimate the optimal risk at $k = 0$: $\hat{J}_0^{(j)}(X_0)$.
 - 3 Using the estimates $\hat{J}_0^{(j)}(X_0)$, $j = 1, \dots, n_R$, compute the desired statistics (e.g. mean, standard deviation).
-

Figure 6.3 illustrates the convergence of various versions of the low biased (LB)

and high biased (HB) of the mesh based estimators. AD is the average density weights and ADN is as AD, but also including the normalization from equation (6.16). SGN is the single grid implementation from section 6.2, including the normalization. SGNQ is the same as SGN, but using the RQMC method as described above. Two setups are used : $K = 16$ (above) and $K = 64$ (below). The SGNQ approximation appears to do best when N is higher, especially for the HB version.

6.4 Removing small weights

To reduce waste of computational resources due to small weights, we can apply a *Russian roulette* procedure, where a weight w less than some given threshold $\delta > 0$ is set to 0 with probability $1 - w/\delta$ and is boosted as $w = \delta$ otherwise. This idea has been commonly used for a long time to improve the efficiency of MC simulations, for example in radiation transport problems (see for example McGrath and Irving [103]), and dates back at least to Kahn and Harris [75]. The roulette procedure may increase the variance of the estimator $\frac{1}{N} \sum_{j=1}^N w_{k,j} Z_{k+1}^j$ of $\mathbf{E}_k[Z_{k+1}]$, but for some values of δ , the reduced computational effort may lead to an overall increased efficiency.

The roulette is often considered along with *splitting*, where destination states Y_{k+1}^j corresponding to large weights are resampled as two or more states to improve the uniformity of the weights. See for example L'Ecuyer, Demers and Tuffin [93] and the references given there. Introducing a new state Y_{k+1}^j on the mesh may correspond to both small and large weights $w_{k,i,j}$, depending on the starting point Y_k^i , so the possible gain from splitting is not obvious (compared to simply using more mesh points, for example). However, we will not study the splitting technique further, and focus only on the roulette.

Denote by $\tilde{w}(\delta)$ weights that go through the roulette procedure with a threshold

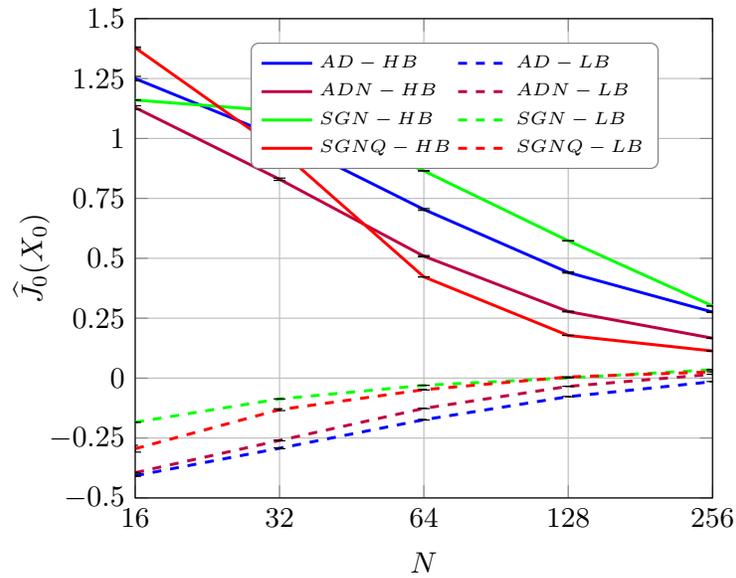
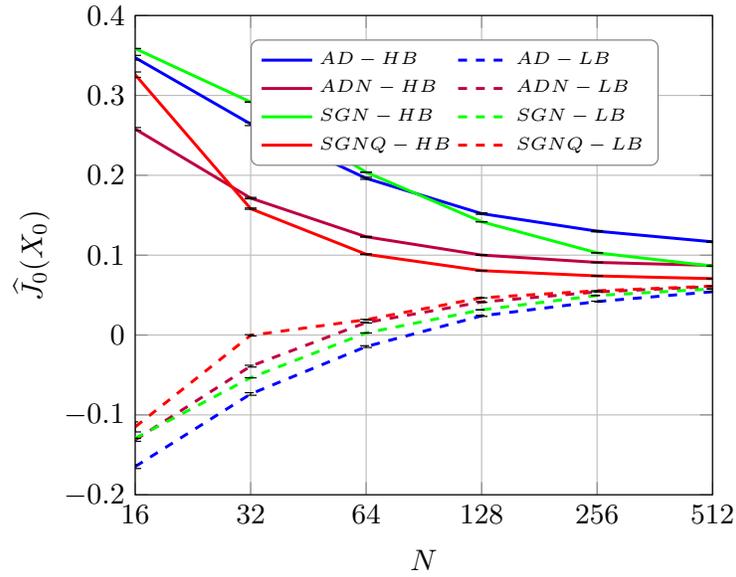


Figure 6.3: Risk estimate $\hat{J}_0(X_0)$ as a function of N , for $K = 16$ (above) and $K = 64$ (below). Setup : call option under the expOU model, with $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $b = 1\%$, $\gamma = 1$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

δ . More formally, we can write

$$\tilde{w}(\delta) = w\mathbf{1}_{\{w \geq \delta\}} + \delta\mathbf{1}_{\{u < w/\delta\}}\mathbf{1}_{\{w < \delta\}}, \text{ for } u \sim U(0, 1), \text{ independent of } w \text{ and } Z. \quad (6.20)$$

Now define G as the measure under which the points of the mesh are sampled, as opposed to the measure P from section 2.1.1 used to define the expectation for the problem (2.3). We usually suppose that the mesh based estimates of conditional expectations are unbiased, in the sense that

$$\mathbb{E}_k^G[Zw] = \mathbb{E}_k^P[Z], \text{ for any (sufficiently integrable) } Z \in \mathcal{F}_{k+1}, \quad (6.21)$$

and where $w = w(Y_k, Y_{k+1}) \in \mathcal{F}_{k+1}$. The following proposition guarantees that the roulette procedure does not introduce bias in the mesh based estimates:

Proposition 6.4.1 *For any sufficiently integrable random variable $Z \in \mathcal{F}_{k+1}$, we have*

$$\mathbb{E}_k^G[Z\tilde{w}(\delta)] = \mathbb{E}_k^G[Zw]. \quad (6.22)$$

Proof: The result can be derived easily by conditioning on w . That is, we first consider $Z\tilde{w}(\delta)$ for a fixed w

$$Z\tilde{w}(\delta) = Z(w\mathbf{1}_{\{w \geq \delta\}} + \delta\mathbf{1}_{\{u < w/\delta\}}\mathbf{1}_{\{w < \delta\}}). \quad (6.23)$$

Taking expectations on both sides, conditional on w , we have

$$\mathbb{E}_k^G[Z\tilde{w}(\delta)|w] = \mathbb{E}_k^G[Z|w](w\mathbf{1}_{\{w \geq \delta\}} + \delta P[u < w/\delta]\mathbf{1}_{\{w < \delta\}}), \quad (6.24)$$

where we have used the independence of the random variable u and the fact that $\mathbb{E}_k^G[\mathbf{1}_{\{u < w/\delta\}}] = P[u < w/\delta]$. Now note that $P[u < w/\delta] = w/\delta$ if $w/\delta \leq 1$ (since

$w \geq 0$ and $\delta > 0$), otherwise $P[u < w/\delta] = 1$ for $w/\delta > 1$. Hence

$$\mathbb{E}_k^G[Z\tilde{w}(\delta)|w] = \mathbb{E}_k^G[Z|w](w\mathbf{1}_{\{w \geq \delta\}} + w\mathbf{1}_{\{w < \delta\}}) = \mathbb{E}_k^G[Z|w](w). \quad (6.25)$$

Therefore, taking w inside the conditional expectation on the right-hand side,

$$\mathbb{E}_k^G[Z\tilde{w}(\delta)|w] = \mathbb{E}_k^G[Zw|w] \quad (6.26)$$

and taking unconditional expectations on both sides completes the proof. \square

We now compare the variance of the stochastic mesh estimator, with and without the roulette procedure. The variance under the measure G of the stochastic mesh estimator of $\mathbb{E}_k^P[Z]$ can be written

$$\text{Var}_k^G \left[\frac{1}{N} \sum_{j=1}^N w_{k,j} Z^j \right] = \frac{1}{N^2} \sum_{j=1}^N \text{Var}_k^G [w_{k,j} Z^j] = \frac{1}{N} \text{Var}_k^G [wZ].$$

Roulette weights may lead to greater variance, but this can be controlled by the choice of the threshold parameter δ :

Proposition 6.4.2 *The increase of the variance can be bounded as*

$$0 \leq \text{Var}_k^G[Z\tilde{w}(\delta)] - \text{Var}_k^G[Zw] \leq \frac{\delta^2}{4} \mathbb{E}_k^G[Z^2 \mathbf{1}_{\{w < \delta\}}] \leq \frac{\delta^2}{4} \mathbb{E}_k^G[Z^2], \quad (6.27)$$

and thus goes to zero at a rate of $O(\delta^2)$.

Proof: To see this, note that because of (6.22), we have

$$\begin{aligned} \text{Var}_k^G[Z\tilde{w}(\delta)] - \text{Var}_k^G[Zw] &= \mathbb{E}_k^G[Z^2\tilde{w}(\delta)^2] - \mathbb{E}_k^G[Z^2w^2] \\ &= \mathbb{E}_k^G [Z^2(\delta^2 \mathbf{1}_{\{u < w/\delta\}} \mathbf{1}_{\{w < \delta\}} - w^2 \mathbf{1}_{\{w < \delta\}})] \\ &= \mathbb{E}_k^G [Z^2(\delta - w)w \mathbf{1}_{\{w < \delta\}}]. \end{aligned}$$

The expression $(\delta - w)w\mathbf{1}_{\{w < \delta\}}$ in the last equation cannot be negative, thus the entire expectation is bounded below by zero. Now the maximum of the same expression is attained by $w = \delta/2$, which yields the upper bound. \square

In general the term $\mathbb{E}_k^G[Z^2\mathbf{1}_{\{w < \delta\}}]$ shows that it also helps if events where Z^2 is high and w is small have low probability under the measure G (i.e., not too many “tail events”). Note that when $\delta \gg 1$, most points will go through the roulette procedure. Since surviving weights all come out being equal to the constant δ , applying the normalization procedure from equation (6.16) will turn them to 1, but if δ is too large, then no points will survive.

6.5 Combining the single grid and Russian roulette methods

The computational cost of constructing a mesh is dominated by the computation of the weights, which requires $O(KN^2)$ operations in general (see section 1.3 or Glasserman [59]), absorbing the (polynomial) dependence on the dimension d of the vector Y_k in the O notation. For the SG method of section 6.2, this will require $O(N^2)$ operations as the relevant computations need only be done for one time step. However, the DP recursion still requires summing over $O(KN^2)$ weights to approximate the functions J_k^* . Thus the time complexity of algorithm 3 remains in $O(KN^2)$.

The combination of the SG method with roulette weights (SG+R) can nevertheless be useful in practice by reducing the computational cost by a large multiplicative factor. To see this more precisely, we write the overall computational effort as the sum of the computational efforts C_{mesh} for computing the mesh (states and weights), and C_{DP} for applying the DP Algorithm.

$$\text{Computational effort} = C_{mesh}MN^2 + C_{DP}KN^2, \quad (6.28)$$

where $M = 1$ for the SG method and $M = K$ for the AD method, and C_{mesh} and C_{DP} are interpreted as the average cost per weight for the mesh and the DP recursion, respectively. Thus the SG method will reduce the overall effect of the mesh cost C_{mesh} , without affecting C_{DP} . Using the roulette weights will increase C_{mesh} slightly, but reduce the total number of weights by a factor depending on δ , d and K , which will reduce C_{DP} accordingly.

Figure 6.4 shows an example of $\widehat{J}_0(X_0)$ versus the base 2 logarithm of the total policy computation time C , for various setups, all using a stochastic mesh with AD weights, with and without the roulette procedure, for various threshold values ($\delta = 0$, 10^{-1} and 10^{-4}). For a given level of precision (horizontal line), the use of the base 2 logarithm implies that the distance between two points can be used to quickly estimate the efficiency improvement factor. For example, at the bottom right of the figure, both the basic estimator without roulette and the estimator with roulette parameter $\delta = 10^{-1}$ have similar lowest values of $\widehat{J}_0(X_0)$, but the roulette based estimator shows a lower computational cost by a factor of approximately $2^2 = 4$. Overall, we see that this estimator (roulette with $\delta = 10^{-1}$) yields similar values for the objective function $\widehat{J}_0(X_0)$ as the alternative estimators considered, but at a lower computational cost, and is thus more efficient.

Tables 6.III, 6.IV and 6.V show more detailed results for three models, respectively the GBM model, the expOU model and the expOU model extended by information variables (hereafter, expOU-I), as described in section 4.1.2. Both versions of the expOU model use the parameter values $\bar{\sigma} = 20\%$, $\kappa = 2.6$, $\sigma_v = 60\%$, $\rho = -50\%$, and the expOU-I model uses the loadings $\alpha_1 = \alpha_2 = 0.33$. The results show, for different mesh construction methods, the values of the low biased (M-LB) and high biased (M-HB) mesh-based risk estimate, along with their difference (Gap). The fraction of non-zero weights on the mesh is shown under f_w , and CT-M, CT-DP, CT-Tot correspond respectively to the computing times for the mesh, the DP recur-

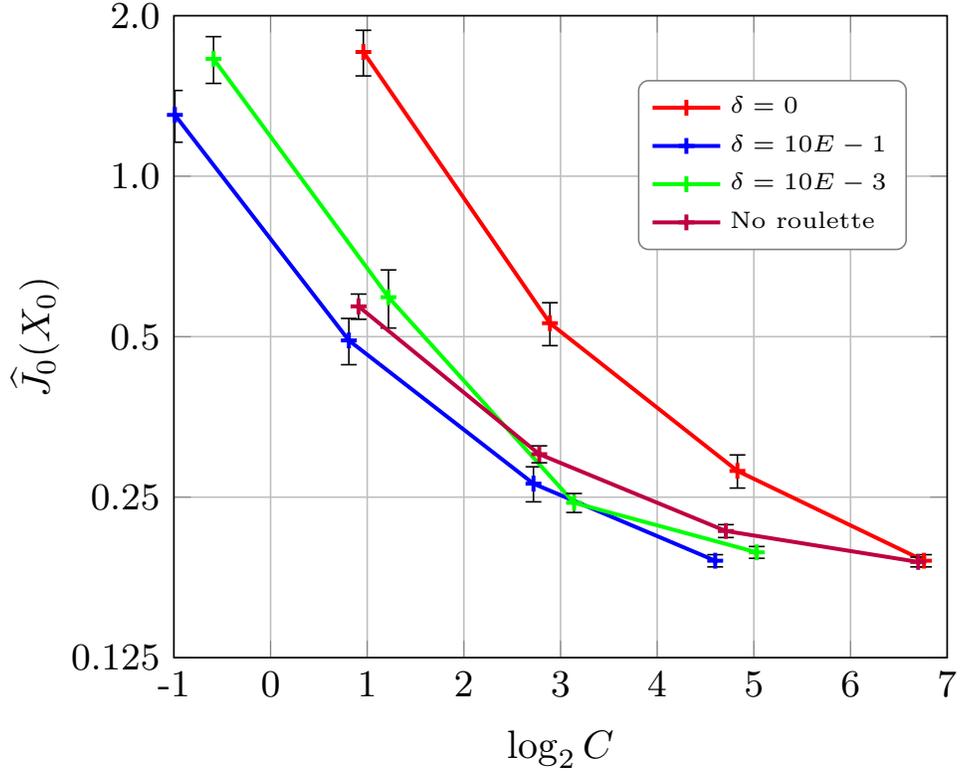


Figure 6.4: Sample average estimates of cost function approximations $\hat{J}_0(X_0)$ as a function of the base-2 logarithm $\log_2 C$ of the total policy computation time. Error bars indicate one standard deviation of the sample averages. The results are for a call option under the GBM model with $s_0 = 10$, $\sigma_0 = 20\%$, $T = 6$ months, $K = 64$, $b = 2\%$ and $\gamma = 1$.

sion for the risk estimation and their total. The last two columns show the computing time improvement (CT Improv.), defined as the ratio of the total computing time for the AD method divided by that of the method considered, and the efficiency improvement (Eff. Improv.).

Following L'Ecuyer ??, we define the efficiency $\text{Eff}(X)$ of an estimator X as the inverse of the work-normalized mean-square error,

$$\text{Eff}(X) = \frac{1}{C(X) \cdot \text{MSE}(X)}, \quad (6.29)$$

where $\text{MSE}(X) = (\mathbb{X} - \mu)^\#$, for μ the quantity to estimate. Here, $\mu = J_0^*(X_0)$ and we approximate the MSE as the square of the difference between the M-HB and M-LB estimators, i.e., we set $\text{MSE} \approx \text{Gap}^2$. To compute a confidence interval for the efficiency improvement ratio, we assume it follows an F -distribution, with parameters d_1 and d_2 given by $d_1 = d_2 = n_R = 100$.

Note that the high biased estimator M-HB was estimated for each of the n_R replications as an average over $n = 1000$ independent paths, using the outer control variate described in section 6.1.1, using an optimal beta coefficient $\beta = \beta^*$ estimated over $n_p = 250$ independent pilot runs. This reduced the variance of the estimator, but it had the undesired effect of introducing a bias in the M-HB estimate for the expOU-I model in table 6.V, because the control (the portfolio performance without costs) no longer had an expectation equal to 0. So for that table, the indicated efficiency improvement factors are unreliable. However, the computation time estimates are unaffected.

The different components considered for the mesh generation method are :

- the independent path construction with average density weights (AD) or the single grid construction with likelihood ratio weights (SG),
- the use of QMC point sets (Q), where a Brownian bridge is used to compute

the sample paths in the case of the independent path construction with AD weights,

- the use of the Russian roulette procedure (R), with R(1) indicating a threshold of $\delta = 10^{-1}$ and R(3), a threshold of $\delta = 10^{-3}$.

The computing times are indicated in seconds per replication and the experiments were executed on a server using x86-64 processors with a Fedora 14 operating system.

Overall, we see that the efficiency improvements are mostly explained by the computation time improvements, as there is not much difference in the gap between M-HB and M-LB for the various methods. Other results not shown here did highlight a larger gap for a lower number of mesh paths N , because the SG method then shows a higher bias. The best setup among those tested is the SG-Q-R(1) method, which combines the various efficiency improvement techniques mentioned in this chapter. Its performance is most noticeable for the expOU model with $K = 64$, where an efficiency improvement factor within the interval $(19, 37)$ is obtained, at the 95%-confidence level. Note also that the efficiency gains observed are better for higher K , as expected from the discussion at the beginning of this section. But these gains are hindered by the relatively high computation times for the risk estimate (CT-DP) compared to the mesh construction (CT-M), due to the cost of computing the exponential function when constructing the risk function approximation at every step.

6.6 Other relevant techniques not tested here

In the context of American option pricing, Avramidis and Hyden [7] describe three relevant techniques neither analyzed nor tested in this thesis : 1) computing a mesh based *low-biased* estimator, as opposed to the basic *high-biased* mesh estimator described by Broadie and Glasserman [36], 2) using importance sampling and 3) using

Table 6.III: Computing times (CT, in seconds per replication) and 95%-level confidence interval for the efficiency improvement factor (using $n_R = 100$ replications, GBM model, $N = 1024$ mesh paths, $\sigma = 20\%$, $T = 6$ months, $\gamma = 1$ and $b = 2\%$).

K	Method	M-LB	M-HB	Gap	f_w	CT-M	CT-DP	CT-Tot	CT Imp.	Eff. Imp.
16	AD	0.151 0.002	0.161 0.001	0.01 0.002	1	4.9	64.8	69.6	1	
16	AD-Q	0.152 0.002	0.161 0.001	0.009 0.002	1	4.8	64.4	69.2	1	(0.8, 1.6)
16	AD-Q-R(1)	0.152 0.002	0.161 0.001	0.009 0.002	0.59	4.4	39.8	44.2	1.6	(1.3, 2.5)
16	AD-Q-R(3)	0.152 0.002	0.161 0.001	0.009 0.002	0.74	4.5	49.4	54	1.3	(1, 2)
16	SG	0.152 0.004	0.162 0.001	0.01 0.004	0.99	1.1	64.8	65.9	1.1	(0.7, 1.4)
16	SG-Q	0.152 0.004	0.163 0.001	0.011 0.005	0.99	1.1	64.8	65.9	1.1	(0.6, 1.2)
16	SG-Q-R(1)	0.152 0.004	0.163 0.001	0.011 0.005	0.42	0.8	27.5	28.3	2.5	(1.4, 2.7)
16	SG-Q-R(3)	0.152 0.004	0.163 0.001	0.011 0.005	0.56	0.9	37.6	38.5	1.8	(1, 2)
64	AD	0.141 0.002	0.163 0.001	0.022 0.002	0.91	21	253.4	274.4	1	
64	AD-Q	0.144 0.002	0.163 0.001	0.019 0.002	0.91	19.4	252.3	271.7	1	(1, 1.9)
64	AD-Q-R(1)	0.144 0.002	0.163 0.001	0.019 0.002	0.37	18.9	99.5	118.3	2.3	(2.3, 4.4)
64	AD-Q-R(3)	0.144 0.002	0.163 0.001	0.019 0.002	0.48	18.4	133	151.3	1.8	(1.8, 3.5)
64	SG	0.144 0.004	0.164 0.001	0.021 0.004	0.79	2.7	222.9	225.6	1.2	(1, 2)
64	SG-Q	0.144 0.005	0.165 0.001	0.021 0.005	0.79	3	225	228.1	1.2	(0.9, 1.8)
64	SG-Q-R(1)	0.144 0.005	0.165 0.001	0.021 0.005	0.23	1.3	57.7	59	4.7	(3.7, 7.1)
64	SG-Q-R(3)	0.144 0.005	0.165 0.001	0.021 0.005	0.31	1.4	81.7	83.2	3.3	(2.6, 5)

Table 6.IV: Computing times (CT, in seconds per replication) and 95%-level confidence interval for the efficiency improvement factor (using $n_R = 100$ replications, $N = 1024$ mesh paths, expOU model with $\sigma = 20\%$, $T = 6$ months, $\gamma = 1$ and $b = 2\%$).

K	Method	M-LB	M-HB	Gap	f_w	CT-M	CT-DP	CT-Tot	CT Imp.	Eff. Imp.
16	AD	0.154 0.004	0.168 0.001	0.014 0.004	0.99	6.9	64.6	71.5	1	
16	AD-Q	0.147 0.004	0.167 0.001	0.02 0.004	0.99	6.2	66	72.2	1	(0.3, 0.7)
16	AD-Q-R(1)	0.147 0.004	0.167 0.001	0.02 0.004	0.42	5.9	29.6	35.4	2	(0.7, 1.4)
16	AD-Q-R(3)	0.147 0.004	0.167 0.001	0.02 0.004	0.61	5.8	40.2	46	1.6	(0.5, 1.1)
16	SG	0.157 0.004	0.17 0.001	0.013 0.004	0.96	1.2	63.9	65.2	1.1	(0.9, 1.8)
16	SG-Q	0.155 0.004	0.17 0.001	0.015 0.005	0.96	1.3	62.4	63.7	1.1	(0.7, 1.3)
16	SG-Q-R(1)	0.155 0.004	0.17 0.001	0.015 0.005	0.27	0.8	17.3	18.2	3.9	(2.3, 4.4)
16	SG-Q-R(3)	0.155 0.004	0.17 0.001	0.015 0.005	0.42	1	28.1	29	2.5	(1.4, 2.8)
64	AD	0.13 0.004	0.21 0.003	0.08 0.005	0.85	29.1	242.7	271.7	1	
64	AD-Q	0.129 0.004	0.209 0.003	0.08 0.005	0.85	26.5	240	266.5	1	(0.7, 1.4)
64	AD-Q-R(1)	0.129 0.004	0.209 0.003	0.08 0.005	0.19	23.7	49.4	73.1	3.7	(2.7, 5.2)
64	AD-Q-R(3)	0.129 0.004	0.209 0.003	0.08 0.005	0.29	24.1	75.8	99.9	2.7	(1.9, 3.8)
64	SG	0.15 0.007	0.208 0.004	0.057 0.007	0.7	2.7	198.5	201.2	1.4	(1.9, 3.7)
64	SG-Q	0.154 0.006	0.21 0.004	0.056 0.007	0.69	2.4	199.8	202.2	1.3	(2, 3.8)
64	SG-Q-R(1)	0.154 0.006	0.209 0.004	0.055 0.007	0.09	1	20.9	21.9	12.4	(19.1, 37.1)
64	SG-Q-R(3)	0.154 0.006	0.21 0.004	0.056 0.007	0.15	1.1	34.2	35.3	7.7	(11.4, 22.1)

Table 6.V: Computing times (CT, in seconds per replication) and 95%-level confidence interval for the efficiency improvement factor (using $n_R = 100$ replications, $N = 512$ mesh paths, expOU-I model with $\sigma = 20\%$, $T = 6$ months, $\gamma = 1$ and $b = 2\%$).

K	Method	M-LB	M-HB	Gap	f_w	CT-M	CT-DP	CT-Tot	CT Imp.	Eff. Imp.
16	AD	0.087 0.012	0.262 0.002	0.175 0.012	0.98	3.4	16	19.3	1	
16	AD-Q	0.084 0.011	0.263 0.002	0.179 0.011	0.98	3.1	15.9	19	1	(0.7, 1.3)
16	AD-Q-R(1)	0.084 0.011	0.263 0.002	0.179 0.011	0.39	3.2	6.3	9.5	2	(1.4, 2.7)
16	AD-Q-R(3)	0.084 0.011	0.263 0.002	0.179 0.011	0.57	3.1	9.1	12.2	1.6	(1.1, 2.1)
16	SG	0.091 0.01	0.274 0.004	0.183 0.011	0.95	0.5	15.3	15.8	1.2	(0.8, 1.5)
16	SG-Q	0.102 0.009	0.277 0.003	0.175 0.01	0.95	0.5	14.2	14.7	1.3	(0.9, 1.8)
16	SG-Q-R(1)	0.102 0.009	0.277 0.003	0.175 0.01	0.25	0.4	3.3	3.7	5.3	(3.8, 7.3)
16	SG-Q-R(3)	0.102 0.009	0.277 0.003	0.175 0.01	0.39	0.4	5.7	6.1	3.1	(2.3, 4.4)
64	AD	0.071 0.011	0.301 0.006	0.23 0.013	0.83	13.4	56.4	69.8	1	
64	AD-Q	0.069 0.01	0.303 0.007	0.234 0.012	0.83	14.5	56	70.5	1	(0.7, 1.3)
64	AD-Q-R(1)	0.069 0.01	0.303 0.007	0.234 0.012	0.17	13.6	10.6	24.2	2.9	(2, 3.9)
64	AD-Q-R(3)	0.069 0.01	0.303 0.007	0.234 0.012	0.26	12.9	16	28.9	2.4	(1.7, 3.2)
64	SG	0.091 0.011	0.379 0.013	0.288 0.017	0.66	0.8	44	44.9	1.6	(0.7, 1.4)
64	SG-Q	0.103 0.011	0.382 0.011	0.279 0.016	0.66	0.8	42.2	43	1.6	(0.8, 1.5)
64	SG-Q-R(1)	0.103 0.011	0.379 0.011	0.276 0.016	0.08	0.4	4.7	5.1	13.7	(6.8, 13.2)
64	SG-Q-R(3)	0.103 0.011	0.38 0.011	0.277 0.016	0.13	0.5	7.2	7.7	9.1	(4.5, 8.7)

a variable number of mesh points at every stage.

To obtain the mesh based low-biased estimator, the idea is to split the mesh points \mathcal{G} into two sets, $I \subset \mathcal{G}$ and $I' = \mathcal{G} - I$. The set I is used to compute the continuation value of the American option (i.e. its value at the next stage $k + 1$ if not exercised at stage k), in order to test if it is optimal to exercise the option. And the set I' is used to recompute the continuation value *if* it is not optimal to exercise the option. These are then combined through an averaging procedure to obtain an estimator with provably low bias. Intuitively, the original high bias is turned into a low bias by estimating the option value using a suboptimal exercise policy with Monte Carlo simulations independent from the ones used to estimate the future value. This estimator has also been studied by Boyle, Kolkiewicz and Tan [31, 32].

In the derivative hedging context of our thesis, this would correspond to computing the expected risk $\widehat{Q}_k(X_k, v)$ from equation (3.7) using the set I , and use this to obtain the optimal decision v^* . Then, use the set I' to compute $\widehat{J}_k(X_k) = \widehat{Q}_k(X_k, v^*)$. This new estimator would have to be high biased (since using a suboptimal decision), as opposed to low biased.

A related estimator suggested by Avramidis and Hyden [7] is to simply take an average of both (in-sample) high and low biased mesh based estimators. Their numerical results show that this average mesh estimator does reduce the bias. As seen in the numerical results from earlier in this chapter, the bias from the mesh based estimator is also a critical problem when applying stochastic mesh methods to the hedging problem. So it would be worthwhile to test the impact on computational efficiency of both the splitted mesh based estimator, and the associated average mesh estimator.

The second technique discussed by Avramidis and Hyden [7] is to adapt some importance sampling ideas from Glasserman, Heidelberger and Shahabuddin [61]

to the stochastic mesh. This implies a modification for the weights used. Their numerical results indicate that the technique works well for pricing smooth American option with smooth payoffs (e.g. a geometric average option), but less so for non-smooth payoffs (e.g. a max option).

The last technique discussed in [7] is to allow more flexibility on the number of mesh points used, because having a fixed number of nodes per stage implies a lower density of nodes at the later stages (for the independent path construction). For example, by introducing a growth rate for the number of nodes at every stage. This is more general than the stochastic tree based method from Broadie and Glasserman [34], and various possible implementations are given in [7]. The idea is to allocate the computational budget more efficiently, without having an exponential growth in the number of mesh nodes like for the stochastic tree.

6.7 Conclusion

In this chapter we looked at different ways to improve on the computational efficiency of the mesh based hedging policy. We first considered the control variates technique, which is well-known for Monte Carlo simulations and was already applied to stochastic mesh methods by Broadie and Glasserman [36]. Our contribution here was to show how it can be adapted to the context of mesh hedging. More specifically, for inner controls, we saw that it makes good sense to try to reduce the variance of mesh based estimators $\widehat{h}_k(Y_k)$ of the derivative price through control variates. In this case, good controls are known (e.g. the derivative price under the BSM model), the optimal coefficient β^* is known, and the associated noise reduction makes a noticeable difference in the optimization step. However, applying the control variates technique to the estimator $\widehat{Q}_k(X_k, v)$ is more difficult, since it depends on the stock quantity v .

We also discussed the idea of using a single grid instead of a stochastic mesh,

which is already present in the random Bellman operator method described by Rust [117]. Applying this here is not expected in itself to reduce the variance, and it can in fact increase it. But it does reduce the time associated with the computation of the weights over the entire mesh by a factor of K , the number of time steps. So the overall efficiency could be improved. We also saw that the quasi-Monte Carlo method can naturally be used to generate states on a single grid. Then the dimension d of the point set is then equal to $m = \dim \mathcal{Y}$, the dimension of the market vectors, instead of the dimension $d = mK$ that would be required to generate one path (from step $k = 1$ to K).

The single grid method can create many small weights between pairs of market vectors. Since these weights may not contribute much to the approximation of conditional expectations, we considered applying a Russian roulette technique to reduce the number of such weights without introducing bias. As far as we know, it is the first time that this technique has been used in the context of a stochastic mesh-type approximation.

Finally, we discussed how the single grid and Russian roulette ideas could be combined, and how these provide complementary computation time reductions. However, the specific efficiency improvement results observed in practice depend on the various details of the problem, such as the number of time steps K or the market vector process used. Results not shown here indicated that the loss function used can also have a large impact on the observed efficiency improvements, for example using a quadratic loss function instead of the negative exponential, because of the associated computation time (i.e. the quadratic loss function is faster to compute, so the dynamic programming part of the algorithm takes less time, whereas the stochastic mesh construction itself is unaffected).

CHAPTER 7

EMPIRICAL COMPARISON OF HEDGING POLICIES

Having established a methodology that allows us to approximate optimal policies relatively well under a stochastic volatility model (the exponential Ornstein-Uhlenbeck model) and for many time steps (experimental results in the previous chapters were up to $K = 64$ steps), we now apply it to investigate the empirical performance of various hedging heuristics.

More specifically, we study the following questions for the case of the expOU model :

1. What is the effect of varying the parameters of the model on the optimal risk $J_0^*(X_0)$ and on the optimal decision $\mu_1^*(X_0)$ when hedging a simple call or put option ?
2. How do the various heuristics perform under various setups compared to the stochastic mesh based approximation (and vice versa) ?

The experimental setups are defined by many possible parameter values. So of course it is not possible to map out all possibilities. To recall, the parameters we consider for hedging under the expOU model and proportional transaction costs are $s_0, u_0, T, K, \gamma, b, \sigma_0, \bar{\sigma}, \sigma_v, \rho$. Perelló, Sircar and Masoliver [108] also provide some numerical results highlighting the impact of various combinations of parameters of the expOU model. However the focus of their paper is option prices and not hedging policies (with or without costs).

We start in section 7.1 by presenting a global overview of the effect on hedging performance of various combinations of problem parameters. This is followed by section 7.2, which presents more in depth results, where the value of only one parameter

at a time is changed while keeping the others fixed. Section 7.3 concludes.

7.1 Global performance comparison for various parameter combinations

Tables 7.I and 7.II show global results for hedging a call and a put option, respectively. Simulations are under the expOU model, starting at-the-money ($s_0 = X = 10$), for various values of the risk aversion γ , volatility σ , number of time steps K and transaction costs b parameters. Derivative values $h_k(Y_k)$ are estimated on the mesh using control variates as described in section 6.1.2.1. The mesh is computed using a single grid generated using uniform random numbers drawn from the Sobol sequence and the roulette technique is applied with threshold $\delta = 10^{-7}$. See sections 6.2, 6.3 and 6.4 for more details.

Here are some observations on the results :

- The Mesh-LB estimates are not very close to the Mesh-HB estimates in general (especially when there are no costs, i.e. for $b = 0$). This indicates that the number of mesh points used here is too low for the mesh-based policy to be close to being optimal. This is likely related to the number of steps K considered being relatively high ($K = 32$ and $K = 64$).
- The policies BSM, WW, Z and Local-A are all equivalent when costs are 0, so their results are the same.
- The NH policy is nearly optimal only in cases where costs are high ($b = 2\%$), volatility is low ($\sigma = 8\%$) and risk aversion is low ($\gamma = 1$).
- The WW policy has poorer results when the costs b are higher, which is expected since this approximation is devised for a low b regime. Note also that WW performs relatively poorly also here when γ is high.
- The BSM policy does well only in the case when costs are zero.

- The Z policy is generally very good. There is a slight underperformance when all of b, σ and γ are high.
- The Local policy is nearly optimal when the volatility is lower ($\sigma = 8\%$) and when there are no costs. Otherwise it tends to underperform.
- The policy approximation Local-A generally performs better than the policy Local itself, perhaps because it is less noisy.
- The approximation error estimate $\hat{\epsilon}_0$ is larger when the risk aversion parameter γ is larger.

Table 7.III shows more global results for the call option hedging problem, but with transaction costs and number of time steps fixed at $b = 1\%$ and $K = 64$ respectively, and with volatility parameters $\sigma_0 = \bar{\sigma} = 16\%$. Instead, we test different combinations of the parameters κ, σ_v and ρ . Generally, both the Mesh and Z policies have a very good performance across the various setups. But the Mesh policy runs into problems for $\kappa = 10, \sigma_v = 10, \rho = -0.8$. For the Z policy, there is a slight underperformance for the lower κ cases. Note however that the NH policy also does well for low γ . This indicates that the volatility stays low enough in the examples that there is not much incentive to hedge for the given level of costs ($b = 1\%$). The number of mesh points used ($N = 2048$) is higher than for table 7.I and 7.II, so the resulting values for the Mesh-LB and Mesh-HB estimators tend to be closer, even though this is also for a high number of steps ($K = 64$).

7.2 Effect of various parameters on ϵ -optimal policies

We now look more precisely at the impact that various parameters can have on the initial hedging decision $\mu_1(X_0)$ of various policies π , and the corresponding risk estimate $\hat{J}_0^\pi(X_0)$ at step $k = 0$.

Table 7.I: Estimates of the portfolio risk for hedging a call option under the negative exponential loss function and the discrete expOU model. Setup : $T = 3$ months, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 2.5$, $\sigma_0 = \bar{\sigma} = \sigma$, $N = 1024$ mesh points and $n = n_R = 100$.

γ	σ	K	b	$\hat{\epsilon}_0$	Mesh-LB	Mesh-HB	Local	Local-A	Z	WW	BSM	NH
1	0.08	32	0	0	-0.021	0.018	0.018	0.011	0.011	0.011	0.011	0.047
				-	-	-	-	-	-	-	-	-
1	0.08	32	0.01	0.003	0.027	0.047	0.052	0.045	0.039	0.08	0.28	0.047
				-	-	-	-	-	-	-	-	-
1	0.08	32	0.02	0.001	0.038	0.052	0.053	0.047	0.044	0.136	0.627	0.047
				-	-	-	0.001	-	-	-	0.001	-
1	0.08	64	0	0	-0.054	0.023	0.024	0.011	0.011	0.011	0.011	0.047
				-	-	-	-	-	-	-	-	-
1	0.08	64	0.01	0.003	0.024	0.048	0.056	0.045	0.04	0.085	0.389	0.047
				-	-	-	0.001	-	-	-	-	-
1	0.08	64	0.02	0.002	0.037	0.054	0.053	0.046	0.044	0.145	0.925	0.047
				-	-	-	-	-	-	-	0.001	-
1	0.16	32	0	0	-0.006	0.049	0.05	0.04	0.04	0.04	0.04	0.165
				-	-	0.001	0.001	-	-	-	-	0.001
1	0.16	32	0.01	0.006	0.074	0.11	0.155	0.137	0.101	0.131	0.315	0.165
				-	-	0.001	0.001	0.001	-	-	-	0.001
1	0.16	32	0.02	0.01	0.106	0.136	0.17	0.153	0.126	0.205	0.669	0.165
				-	-	0.001	0.001	0.001	-	-	0.001	0.001
1	0.16	64	0	0	-0.056	0.063	0.064	0.038	0.038	0.038	0.038	0.164
				-	-	0.001	0.001	-	-	-	-	0.001
1	0.16	64	0.01	0.011	0.065	0.118	0.179	0.149	0.101	0.135	0.423	0.164
				-	-	0.001	0.001	0.001	-	-	0.001	0.001
1	0.16	64	0.02	0.014	0.101	0.14	0.182	0.158	0.127	0.215	0.969	0.164
				-	-	0.001	0.002	0.001	-	-	0.001	0.001
5	0.08	32	0	0	0.026	0.046	0.045	0.043	0.043	0.043	0.043	0.172
				-	-	-	-	-	-	-	-	0.001
5	0.08	32	0.01	0.036	0.106	0.116	0.161	0.149	0.12	0.165	0.61	0.172
				-	-	-	0.001	0.001	-	-	0.001	0.001
5	0.08	32	0.02	0.059	0.137	0.145	0.168	0.163	0.154	0.298	2.785	0.172
				-	-	-	0.001	0.001	-	-	0.007	0.001
5	0.08	64	0	0	0.006	0.053	0.053	0.042	0.042	0.042	0.042	0.172
				-	-	0.001	-	-	-	-	-	0.001
5	0.08	64	0.01	0.057	0.103	0.12	0.19	0.162	0.121	0.174	1.064	0.172
				-	-	-	0.002	0.001	-	-	0.003	0.001
5	0.08	64	0.02	0.084	0.136	0.149	0.192	0.169	0.158	0.323	7.932	0.172
				-	-	-	0.005	0.001	-	-	0.034	0.001
5	0.16	32	0	0	0.114	0.154	0.155	0.156	0.156	0.156	0.156	0.78
				-	-	0.001	0.001	0.001	0.001	0.001	0.001	0.002
5	0.16	32	0.01	0.035	0.295	0.326	0.549	0.51	0.354	0.392	0.97	0.78
				-	-	0.001	0.003	0.002	0.001	0.001	0.002	0.002
5	0.16	32	0.02	0.113	0.417	0.444	0.667	0.638	0.506	0.662	4.054	0.78
				0.001	-	0.001	0.003	0.002	0.001	0.001	0.013	0.002
5	0.16	64	0	0	0.069	0.173	0.174	0.153	0.153	0.153	0.153	0.795
				-	-	0.001	0.001	0.001	0.001	0.001	0.001	0.002
5	0.16	64	0.01	0.072	0.281	0.333	0.699	0.61	0.351	0.403	1.608	0.795
				-	-	0.001	0.007	0.002	0.001	0.001	0.005	0.002
5	0.16	64	0.02	0.182	0.402	0.451	0.832	0.719	0.506	0.703	11.231	0.795
				-	-	0.001	0.011	0.003	0.002	0.001	0.057	0.002

Table 7.II: Estimates of the portfolio risk for hedging a put option under the negative exponential loss function and the discrete expOU model. Setup : $T = 3$ months, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 2.5$, $\sigma_0 = \bar{\sigma} = \sigma$, $N = 1024$ mesh points and $n = n_R = 100$.

γ	σ	K	b	$\hat{\epsilon}_0$	Mesh-LB	Mesh-HB	Local	Local-A	Z	WW	BSM	NH
1	0.08	32	0	0	-0.022	0.019	0.018	0.012	0.012	0.012	0.012	0.073
				-	-	-	-	-	-	-	-	0.001
1	0.08	32	0.01	0.003	0.038	0.058	0.067	0.066	0.054	0.074	0.278	0.072
				-	-	-	-	0.001	-	-	-	0.001
1	0.08	32	0.02	0.001	0.054	0.067	0.069	0.07	0.062	0.109	0.624	0.072
				-	-	-	0.001	0.001	-	-	0.001	0.001
1	0.08	64	0	0	-0.057	0.024	0.025	0.011	0.011	0.011	0.011	0.075
				-	-	-	-	-	-	-	-	0.001
1	0.08	64	0.01	0.003	0.033	0.058	0.07	0.069	0.053	0.077	0.385	0.072
				-	-	-	-	0.001	-	-	-	0.001
1	0.08	64	0.02	0.001	0.045	0.072	0.072	0.073	0.063	0.117	0.922	0.074
				-	-	-	0.001	0.001	-	-	0.001	0.001
1	0.16	32	0	0	-0.007	0.055	0.055	0.041	0.041	0.041	0.041	0.218
				-	-	0.001	0.001	-	-	-	-	0.002
1	0.16	32	0.01	0.006	0.084	0.12	0.177	0.174	0.122	0.136	0.312	0.218
				-	-	0.001	0.001	0.001	0.001	-	0.001	0.001
1	0.16	32	0.02	0.009	0.125	0.155	0.199	0.193	0.152	0.193	0.66	0.218
				-	-	0.001	0.001	0.001	0.001	0.001	0.001	0.002
1	0.16	64	0	0	-0.06	0.065	0.065	0.039	0.039	0.039	0.039	0.221
				-	0.001	0.001	0.001	-	-	-	-	0.002
1	0.16	64	0.01	0.011	0.074	0.123	0.203	0.193	0.118	0.138	0.418	0.218
				-	-	0.001	0.001	0.001	-	-	-	0.002
1	0.16	64	0.02	0.014	0.115	0.158	0.211	0.211	0.152	0.202	0.961	0.224
				-	-	0.001	0.001	0.001	0.001	-	0.001	0.002
5	0.08	32	0	0	0.026	0.047	0.047	0.044	0.044	0.044	0.044	0.224
				-	-	-	-	-	-	-	-	0.001
5	0.08	32	0.01	0.036	0.119	0.133	0.189	0.185	0.135	0.168	0.599	0.224
				-	-	-	0.001	0.001	-	-	0.001	0.001
5	0.08	32	0.02	0.058	0.161	0.173	0.21	0.203	0.175	0.268	2.728	0.223
				-	-	-	0.001	0.001	-	-	0.007	0.001
5	0.08	64	0	0	0.005	0.055	0.054	0.043	0.043	0.043	0.043	0.227
				-	-	0.001	0.001	-	-	-	-	0.001
5	0.08	64	0.01	0.057	0.113	0.133	0.211	0.202	0.134	0.175	1.044	0.224
				-	-	-	0.001	0.001	-	-	0.002	0.001
5	0.08	64	0.02	0.087	0.154	0.179	0.223	0.218	0.179	0.293	7.853	0.228
				-	-	-	0.001	0.001	-	-	0.029	0.001
5	0.16	32	0	0	0.113	0.161	0.16	0.156	0.156	0.156	0.156	0.941
				-	-	0.001	0.001	0.001	0.001	0.001	0.001	0.002
5	0.16	32	0.01	0.035	0.312	0.343	0.559	0.575	0.363	0.406	0.945	0.944
				-	-	0.001	0.002	0.001	0.001	0.001	0.002	0.002
5	0.16	32	0.02	0.11	0.448	0.483	0.869	0.722	0.512	0.648	3.895	0.944
				-	-	0.001	0.047	0.002	0.001	0.001	0.012	0.002
5	0.16	64	0	0	0.067	0.18	0.188	0.152	0.152	0.152	0.152	0.955
				-	-	0.003	0.005	0.001	0.001	0.001	0.001	0.002
5	0.16	64	0.01	0.073	0.297	0.35	0.7	0.692	0.361	0.415	1.566	0.952
				-	-	0.001	0.003	0.002	0.001	0.001	0.005	0.002
5	0.16	64	0.02	0.182	0.429	5.163	36.795	0.821	0.517	0.686	10.82	0.968
				-	-	1.478	10.008	0.002	0.001	0.001	0.046	0.003

Table 7.III: Estimates of the portfolio risk when hedging a call option under the negative exponential loss function and the discrete expOU model. Setup : $T = 3$ months, $K = 64$ $\sigma_0 = \bar{\sigma} = 16\%$, $b = 1\%$, $N = 2048$ mesh points, and $n = n_R = 100$.

γ	σ_v	κ	ρ	$\hat{\epsilon}_0$	Mesh-LB	Mesh-HB	Local	Local-A	Z	WW	BSM	NH
1	1.25	2.5	-0.8	0.008	0.053	0.068	0.089	0.079	0.063	0.093	0.365	0.079
				-	-	-	-	-	-	-	-	-
1	1.25	2.5	-0.4	0.008	0.059	0.073	0.108	0.101	0.07	0.096	0.369	0.101
				-	-	-	0.001	0.001	-	-	-	0.001
1	1.25	2.5	0	0.008	0.063	0.077	0.124	0.121	0.076	0.098	0.372	0.122
				-	-	-	0.001	0.001	-	-	-	0.001
1	1.25	10	-0.8	0.009	0.055	0.073	0.093	0.084	0.066	0.092	0.364	0.084
				-	-	-	-	-	-	-	-	-
1	1.25	10	-0.4	0.008	0.059	0.068	0.1	0.096	0.069	0.092	0.367	0.096
				-	-	-	0.001	0.001	-	-	-	0.001
1	1.25	10	0	0.008	0.06	0.07	0.11	0.108	0.071	0.092	0.369	0.108
				-	-	-	0.001	0.001	-	-	-	0.001
1	2.5	2.5	-0.8	0.011	0.035	0.091	0.121	0.096	0.074	0.12	0.397	0.098
				-	-	0.001	0.001	-	-	-	-	0.001
1	2.5	2.5	-0.4	0.009	0.074	0.113	0.175	0.152	0.103	0.137	0.423	0.169
				-	-	0.001	0.001	0.001	-	-	0.001	0.001
1	2.5	2.5	0	0.008	0.094	0.131	0.208	0.193	0.122	0.149	0.437	0.23
				-	-	0.001	0.001	0.001	0.001	-	0.001	0.003
1	2.5	10	-0.8	0.011	0.053	0.08	0.104	0.087	0.067	0.104	0.383	0.087
				-	-	-	0.001	-	-	-	-	-
1	2.5	10	-0.4	0.009	0.065	0.08	0.127	0.118	0.078	0.106	0.393	0.119
				-	-	-	0.001	0.001	-	-	-	0.001
1	2.5	10	0	0.009	0.07	0.087	0.152	0.144	0.085	0.107	0.398	0.149
				-	-	-	0.001	0.001	-	-	-	0.001
5	1.25	2.5	-0.8	0.057	0.124	0.149	0.434	0.384	0.159	0.179	0.894	0.386
				-	-	0.001	0.005	0.001	-	-	0.002	0.001
5	1.25	2.5	-0.4	0.056	0.143	0.178	0.493	0.425	0.173	0.192	0.941	0.449
				-	-	0.004	0.011	0.001	-	-	0.002	0.001
5	1.25	2.5	0	0.057	0.153	0.176	0.483	0.454	0.18	0.199	0.962	0.5
				-	-	0.001	0.003	0.001	-	-	0.002	0.001
5	1.25	10	-0.8	0.061	0.122	0.166	0.565	0.388	0.151	0.167	0.866	0.388
				-	-	0.002	0.037	0.001	-	-	0.002	0.001
5	1.25	10	-0.4	0.057	0.128	0.148	0.438	0.415	0.155	0.169	0.882	0.42
				-	-	-	0.003	0.001	-	-	0.002	0.001
5	1.25	10	0	0.057	0.131	0.149	0.453	0.433	0.157	0.171	0.891	0.449
				-	-	-	0.004	0.001	-	-	0.002	0.001
5	2.5	2.5	-0.8	0.067	0.167	0.231	0.6	0.472	0.251	0.306	1.23	0.519
				-	-	0.004	0.014	0.002	0.001	0.001	0.003	0.001
5	2.5	2.5	-0.4	0.076	0.289	0.326	0.681	0.603	0.35	0.401	1.583	0.792
				0.003	-	0.001	0.008	0.002	0.001	0.001	0.005	0.002
5	2.5	2.5	0	0.069	0.351	0.398	0.793	0.713	0.4	0.453	1.749	1.007
				-	-	0.003	0.011	0.002	0.001	0.001	0.006	0.003
5	2.5	10	-0.8	0.065	0.138	0.264	0.57	0.427	0.187	0.223	1.035	0.447
				-	-	0.023	0.006	0.001	0.001	-	0.002	0.001
5	2.5	10	-0.4	0.059	0.17	0.197	0.572	0.463	0.209	0.236	1.13	0.548
				-	-	0.001	0.013	0.001	0.001	0.001	0.003	0.001
5	2.5	10	0	0.059	0.184	0.22	0.624	0.49	0.218	0.243	1.168	0.628
				-	-	0.003	0.02	0.001	0.001	0.001	0.003	0.002

7.2.1 Transaction costs b

Figure 7.1 shows the effect of proportional transaction costs b on risk estimates $\widehat{J}_0^\pi(X_0)$ for various policies π . For the optimal policy (approximated on the stochastic mesh), the optimal risk increases with b until reaching a ceiling, corresponding to the no-hedging policy (NH), when costs are too high. Whether costs are “too high” or not is relative, as it depends on the value of other parameters such as the volatility level σ and the number of time steps K . The Z policy behaves very well for all levels of costs tested here. The divergence of the BSM policy as b increases is not a surprise, as mentioned in section 1.2. We also see that the WW policy is nearly optimal when b is low, which is expected since it should be optimal (at least for the GBM model) in the limit of low transaction costs. However it seems to diverge as b increases, even underperforming the NH policy. The Local hedging policy and its approximation Local-A only do well in the very low and very high cost regimes, significantly underperforming in between. For low costs, they both are similar to the BSM policy, whereas for high costs, they quickly tend towards the NH policy.

7.2.2 Number of time steps K

We now consider the effect of increasing the number of time steps K for a fixed option maturity T (so the time steps T/K become smaller). Obviously, this should not impact the decisions of the policies NH, BSM, WW and Z since they are not defined in terms of time steps. However, it will affect the associated risk estimate $\widehat{J}_0^\pi(X_0)$, since the policy decisions are then applied more often as K increases.

Figure 7.2 shows that the mesh based policy does well for low K . But the convergence of the mesh based estimators is slower as K increases, as can be seen from the growing gap between the high biased (Mesh-HB) and low biased (Mesh-LB) estimators. Perhaps surprisingly, the WW policy tends to diverge as K is increased. The Z policy does not have this problem, and seems to go down with K , which is a

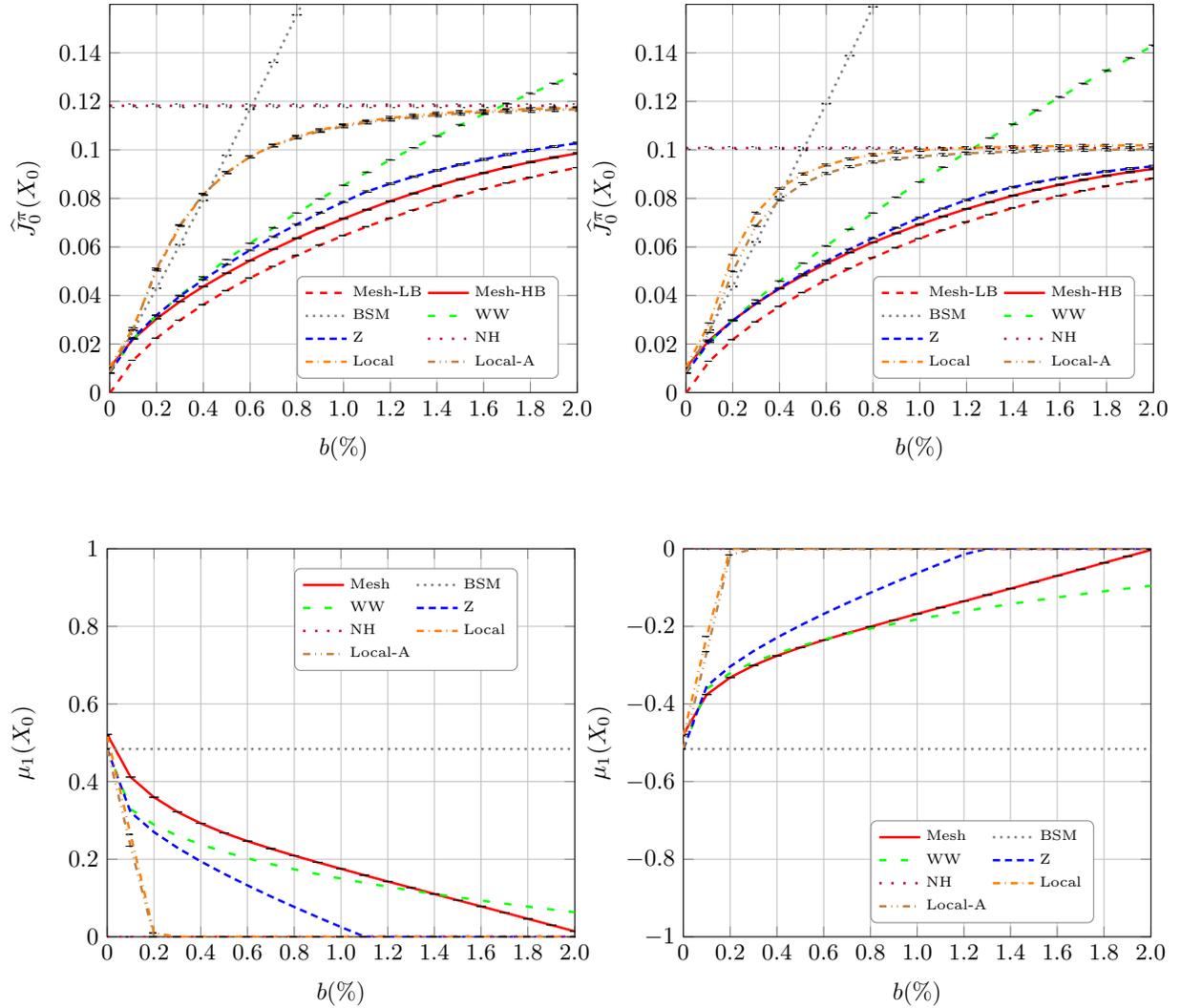


Figure 7.1: Risk estimate $\hat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the transaction cost parameter b . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $\gamma = 1$, $N = 1024$ and $n = 100$, $n_R = 1000$. Error bars indicate one standard deviation of the sample averages.

behavior one would expect from a policy that is close to the optimum in continuous time.

The risk estimate for the Z policy decreases with K in figure 7.2. But for other setups, the risk estimates may in fact increase with K (see figure 7.3, with $b = 2\%$ and $\gamma = 5$).

7.2.3 Stock price s_0

Figure 7.4 shows the effect of the initial stock price s_0 . For low values of s_0 ($s_0 < X = 10$), the call option is out-of-the-money and its price has a lower sensitivity to the moves of the stock price. Hence there is less risk. However, for values of s_0 near the strike price $X = 10$ and above, the call option is in the money and the risk increases, As the stock price increases, so do the proportional transaction costs $bs_0|v - u_0|$ required for trading v shares, where u_0 denotes the initial stock quantity. This explains the rising trend on the right hand side of the graph for $\widehat{J}_0^\pi(X_0)$ for the call option (since in the example, the stock quantity is fixed at $u_0 = 0$). There is a growing gap between the low biased and high biased mesh estimators as s_0 increases, indicating that the optimization step is harder to solve.

For the put option, the behavior is reversed. The price sensitivity is high for low values of s_0 , which is why the initial decisions $\mu_1(X_0)$ are closer to 1 than 0. The downward trend of the graph for $\widehat{J}_0^\pi(X_0)$ is due to the reduced transaction costs if the stock price is lower. Interestingly, there is also a growing gap between the low biased and high biased estimators, which increases with s_0 . At this point, the precise cause of these two gaps (for the call and put option cases) is not clear.

Note that the policies Z and SM are very similar for all values of s_0 . The policy WW is similar to Z and SM for $s_0 \leq X = 10$, but then tends to the BSM policy for $s_0 > X$.

Finally, figure 7.5 shows hedging bands for three policies : WW, Z and Mesh.

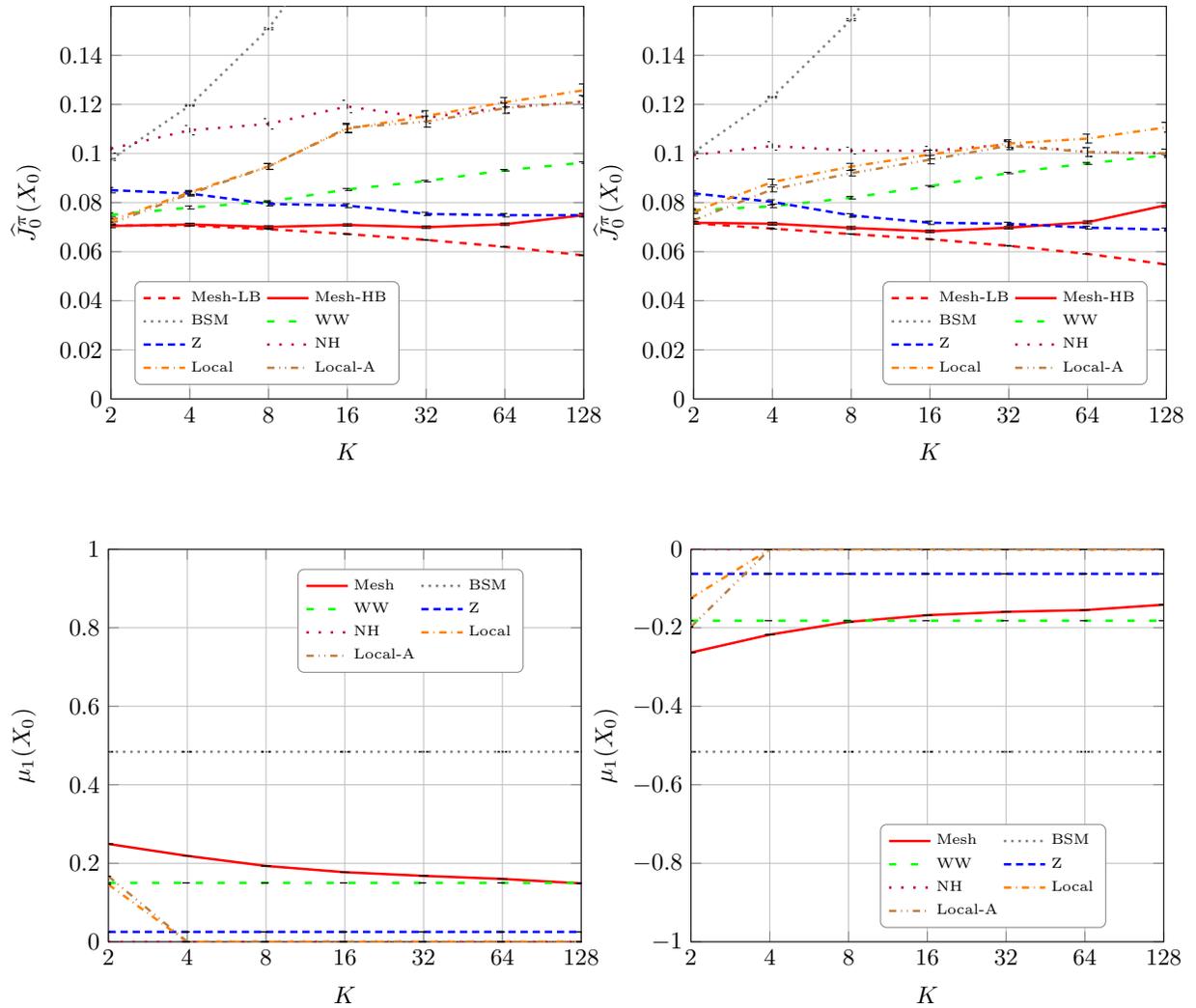


Figure 7.2: Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the number of time steps K . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $b = 1\%$, $\gamma = 1$, $N = 2048$ for the call and $N = 4096$ for the put, and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

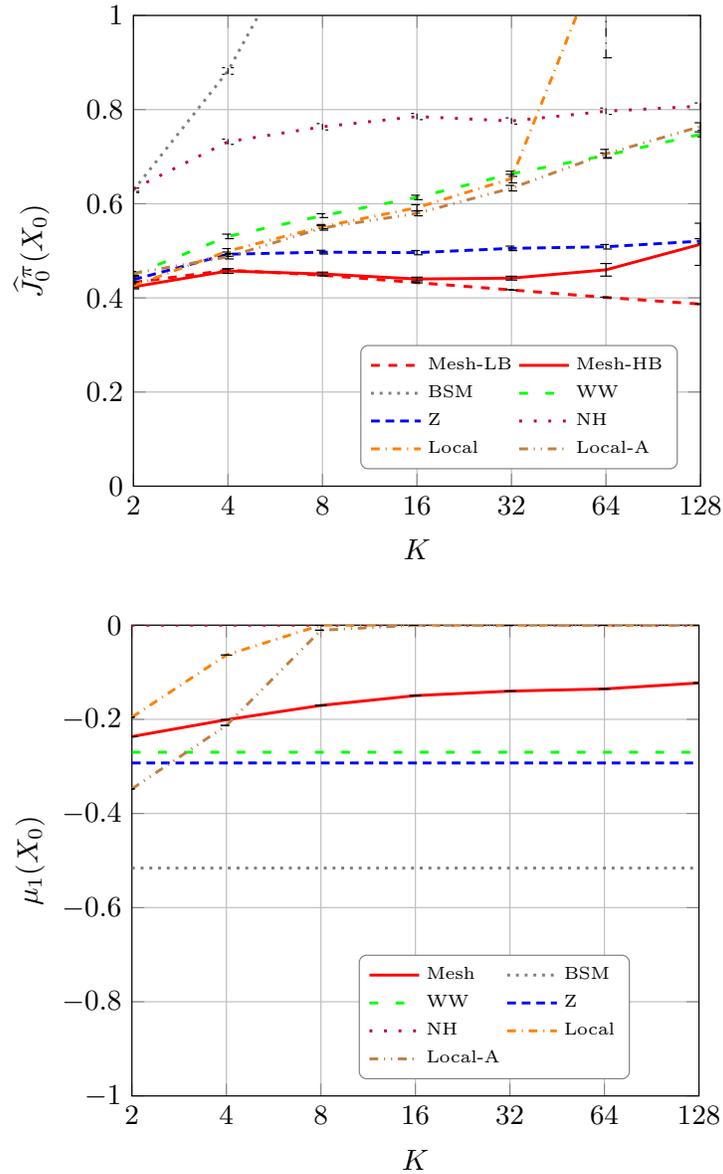


Figure 7.3: Risk estimate $\hat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the number of time steps K . The setup is similar as in figure 7.2, but only for the call option, with the transaction costs and risk aversion increased to $b = 2\%$ and $\gamma = 5$, respectively. Using $N = 1024$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

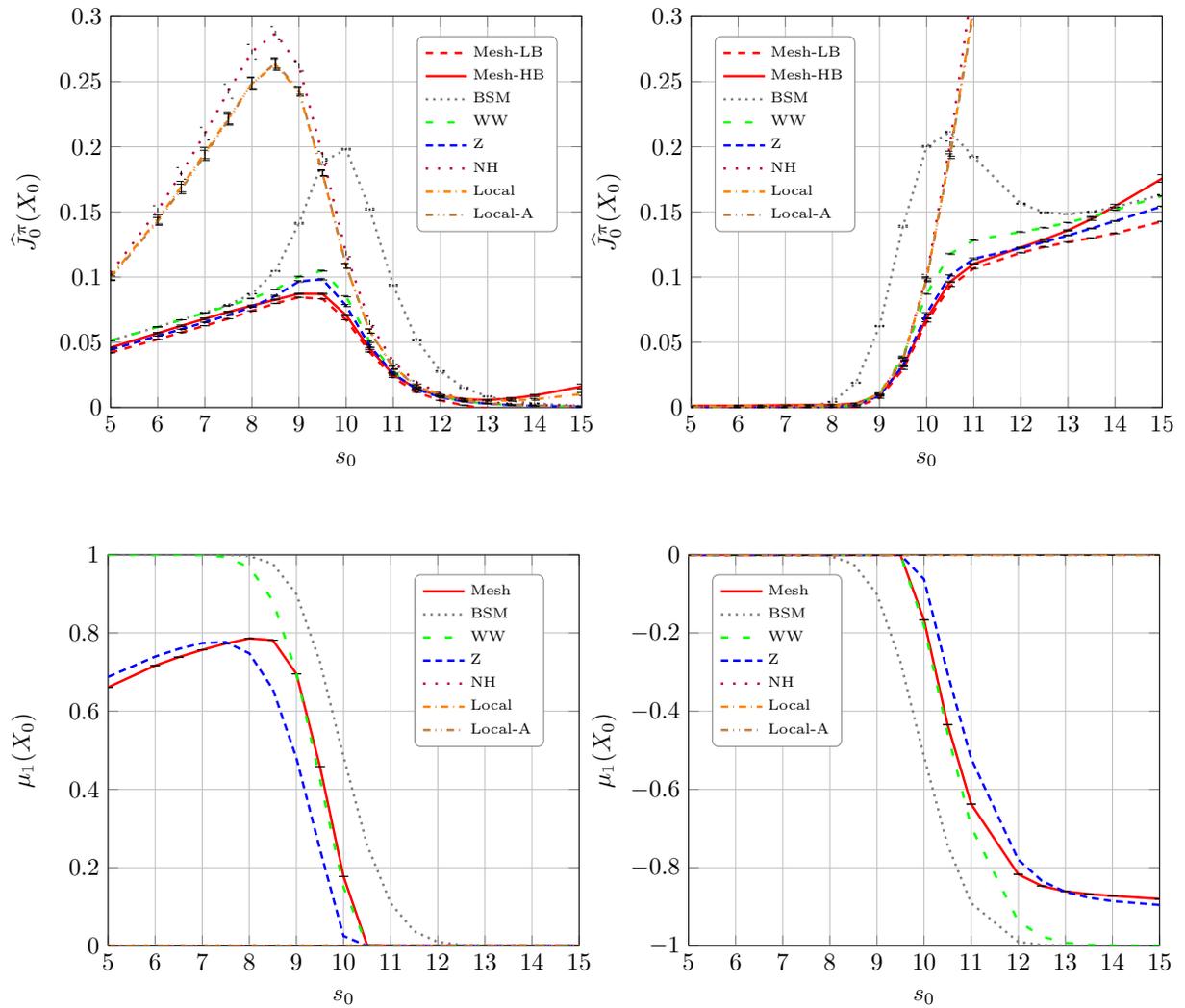


Figure 7.4: Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the stock price s_0 . Left-hand side : put option, right-hand side : call option. Setup : $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 4096$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

Interestingly, the mesh based policy is closer to the WW policy when the stock price s_0 is closer to the strike price X , and closer to the Z policy when the stock price is away from X .

7.2.4 Stock quantity u_0

Figure 7.6 shows the effect of the initial stock quantity u_0 . The shape of the risk estimate $\widehat{J}_0^\pi(X_0)$ is convex as a function of u_0 for the various policies π , and reflects the fact that for more “extreme” values of u_0 (i.e., close to 0 or -1), the portfolio is not well hedged, which leads to more expected risk coming from price movements (e.g. for the NH policy), some trading costs (e.g. for the BSM policy), or both.

Here the risk estimates $\widehat{J}_0^\pi(X_0)$ look similar for Z and SM, but not the policies π (for extreme values of u_0). However, the Mesh and WW policies look very similar as a function of u_0 , but the WW policy is clearly not optimal when we look at the $\widehat{J}_0^\pi(X_0)$ graph. This highlights a potential problem when comparing policies only from the decisions at a given step. The difference between the Z and Mesh policies is apparent for $\mu_1(X_0)$ (at $k = 0$) and indicates that Z could be improved. Yet overall the potential improvement on Z is relatively small, as can be seen from looking at the small gap between Z and Mesh-LB. On the other hand, the initial decision of the WW policy are similar to the Mesh policy at $k = 0$, but as time goes by the difference between the two policies grows, which has an important effect on their respective risk estimates. See also the discussion later in sections 7.2.5 and 7.2.6.

7.2.5 Initial volatility σ_0 , mean volatility level $\bar{\sigma}$ and volatility of log-volatility σ_v

Figure 7.7 shows the effect of the initial volatility level σ_0 . Overall, the risk estimate $\widehat{J}_0^\pi(X_0)$ increases with σ_0 for the various policies π . The mesh based estimate performs better than the alternatives, especially in the case of the put option (left-

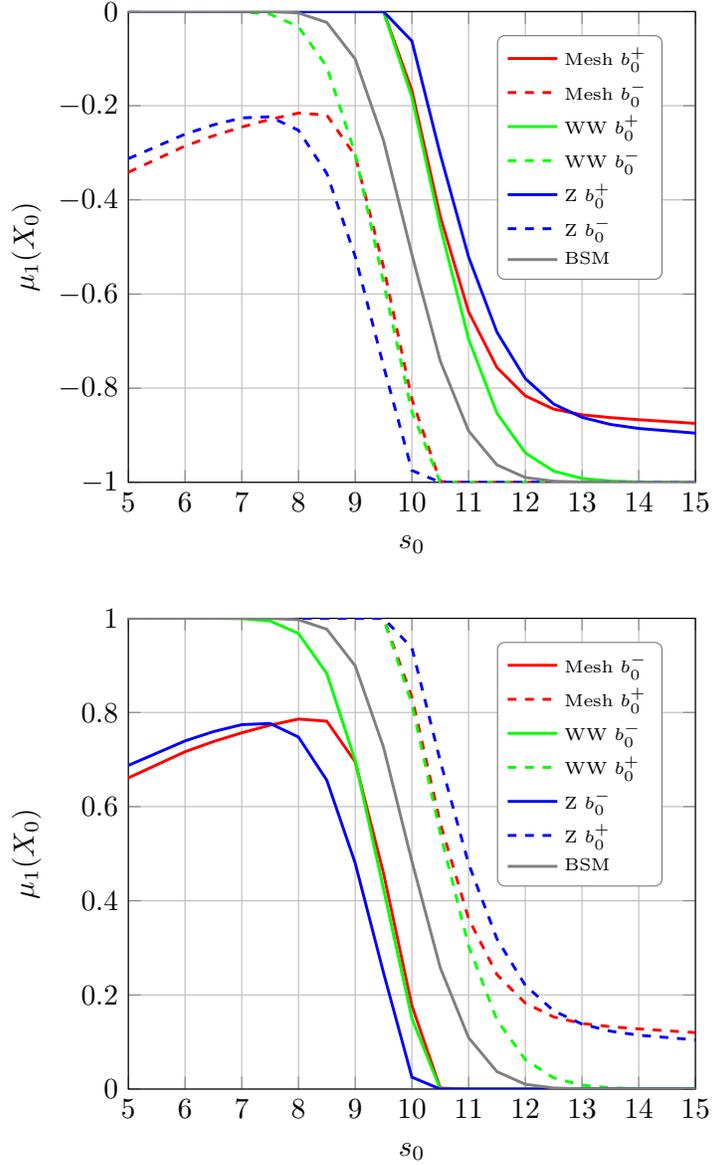


Figure 7.5: No hedging boundaries for the initial decision $\mu_1(X_0)$ for various policies as a function of the stock price s_0 , for a call option (above) and a put option (below) under the expOU model. Setup : $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 1024$ and $n = n_R = 100$.

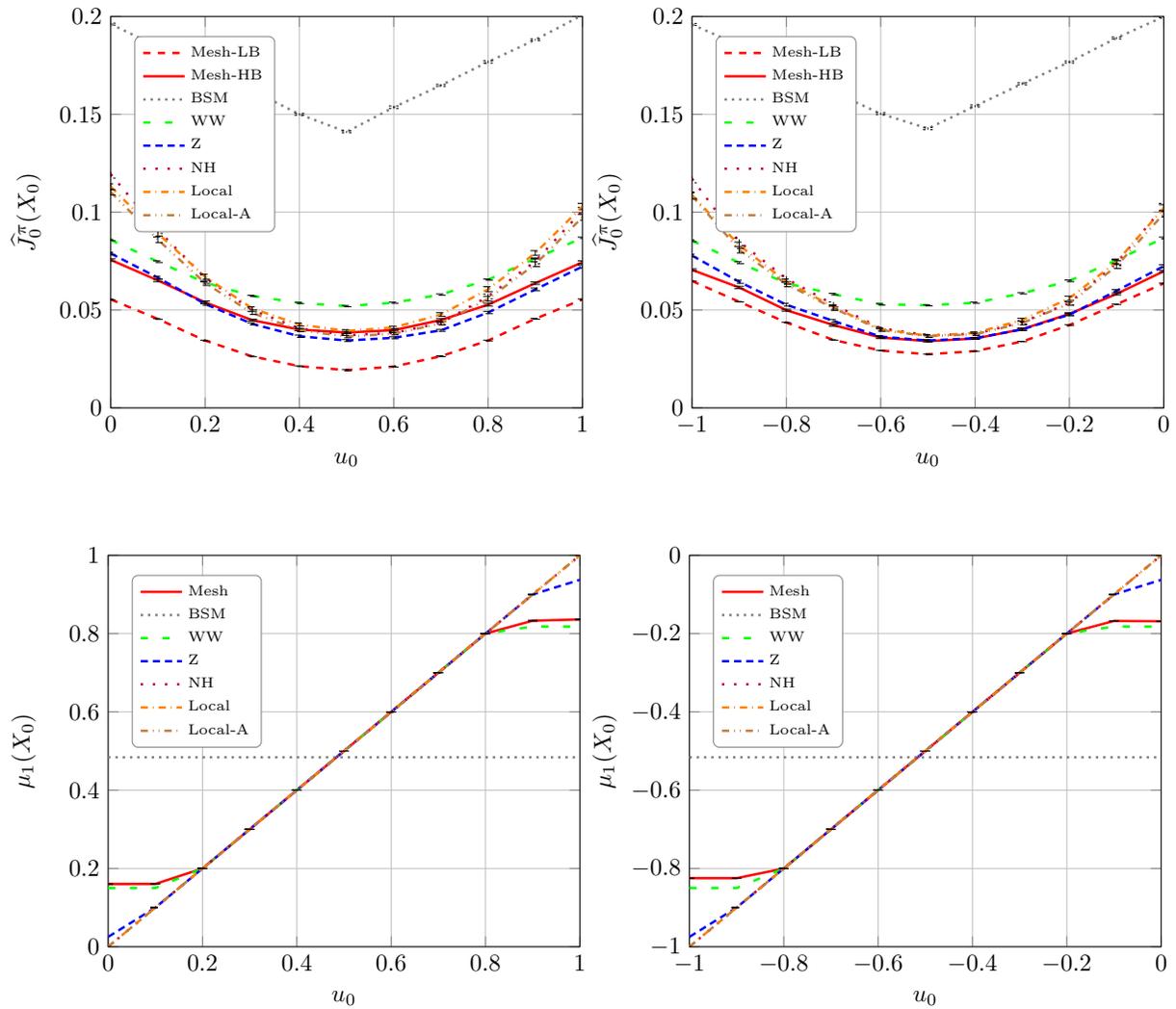


Figure 7.6: Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the stock quantity price u_0 . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 1024$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

hand side), and for higher σ_0 . The relatively small gap between the Mesh-LB and Mesh-HB estimates indicates that the mesh based policy is indeed close to being optimal.

Note that although the WW policy decisions $\mu_1(X_0)$ are closest to that of the mesh based policy on the graphs, the Z policy tends to have a smaller risk, as measured by $\widehat{J}_0^\pi(X_0)$. This is not a contradiction since the decision functions illustrated by the lower graphs are for the initial step only ($k = 0$) and for a fixed stock price s_0 , so for other values of (k, s_k) the Z policy could be closer to the mesh based policy than the WW policy (as can be already seen in figure 7.4). Note also that there is a noticeable “dip” in the Z policy around $\sigma_0 = 16\%$, when compared to the WW-policy. This is not a statistical error and reflects the functional form of the policy, since the policy decisions shown are exact for all except the Mesh-based policy. For larger σ_0 , however, both the risk estimate and the policy decision for the WW and Z policy tend to be closer.

As seen in figure 7.8, the effect of the long-term volatility parameter $\bar{\sigma}$ on the risk estimates is similar to that of σ_0 , as $\widehat{J}_0^\pi(X_0)$ increases with $\bar{\sigma}$. The impact of $\bar{\sigma}$ is however less pronounced, with an apparently smaller rate of increase. The Mesh-based policy is again seen to outperform the heuristics, especially in the case of the put option. But the gap between Mesh-LB and Mesh-HB is larger than in the figures 7.7, and gets larger with $\bar{\sigma}$. This indicates a stronger bias in the Mesh-based estimates.

As in the case of the parameter σ_0 seen above, the difference in performance between WW and Z gets smaller for higher $\bar{\sigma}$. The WW policy even does slightly better in the case of the put option for $\bar{\sigma} = 0.40$. The policy decisions for all heuristics are constants as functions of the parameter $\bar{\sigma}$, by definition. But interestingly, we notice that $\bar{\sigma}$ does have an effect on the optimal decision estimate associated to the mesh based policy. More specifically, one should hedge more (i.e. higher absolute

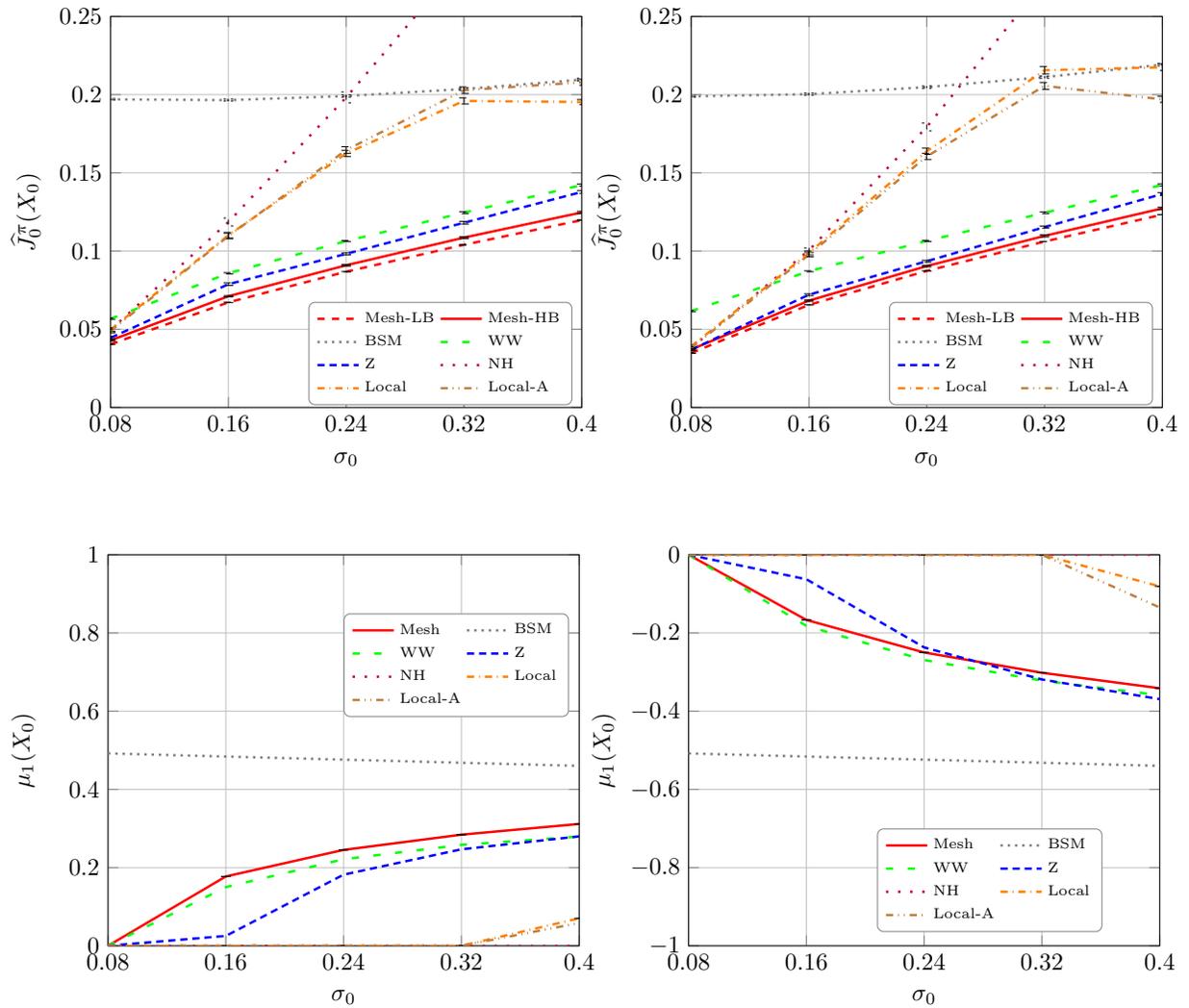


Figure 7.7: Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the initial volatility parameter σ_0 . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 4096$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

value of the hedging decision) as $\bar{\sigma}$ is increased. The impact on the optimal decision tends to level off for higher $\bar{\sigma}$, but of course this behavior could depend on the values of other model parameters.

Figure 7.9 shows that the risk estimate $\widehat{J}_0^\pi(X_0)$ tends to increase with the volatility of log-volatility parameter σ_v . The gap between the high and low biased stochastic mesh estimates is smaller for small σ_v . For $\sigma_v = 1$, it is apparent that there is clearly some (small) gain from using an optimal policy instead of the Z policy. For $\sigma_v = 2.5$, a higher number of mesh points would be required to reach a conclusion, as the out-of-sample estimates Mesh-HB and Z are quite close (especially in the case of the call option), and the gap between the Mesh-HB and Mesh-LB estimates is large. The mesh based policy decisions seems to increase slowly with σ_v (i.e., become less negative) for both the call and the put example. Interestingly, in the case of the call option, however, this is in the opposite direction to that of σ_0 or $\bar{\sigma}$ (i.e. the mesh based decision $\mu_1(X_0)$ tends to increase as function of σ_v , but decrease as a function of σ_0 or $\bar{\sigma}$). As increases in either $\bar{\sigma}$ or σ_v can both be roughly interpreted as increases in the volatility of the model, it is not intuitive that the (approximately optimal) mesh based policy should behave qualitatively differently as a function of σ_v . But here the results could also be influenced by the larger bias in the estimates for higher σ_v (as indicated from the gap between Mesh-HB and Mesh-LB), so more precise experiments would be required to clarify the situation.

7.2.6 Time horizon T

Figure 7.10 shows the behavior of the policy decisions and associated risk estimates for different values of the time to expiry T for the options to hedge. As for the volatility-related parameters σ_0 , $\bar{\sigma}$ and σ_v , there is a significant improvement (i.e. reduction) in the risk associated with the mesh based policy compared to the other heuristics tested. This is especially true in the case of the put option, whereas the

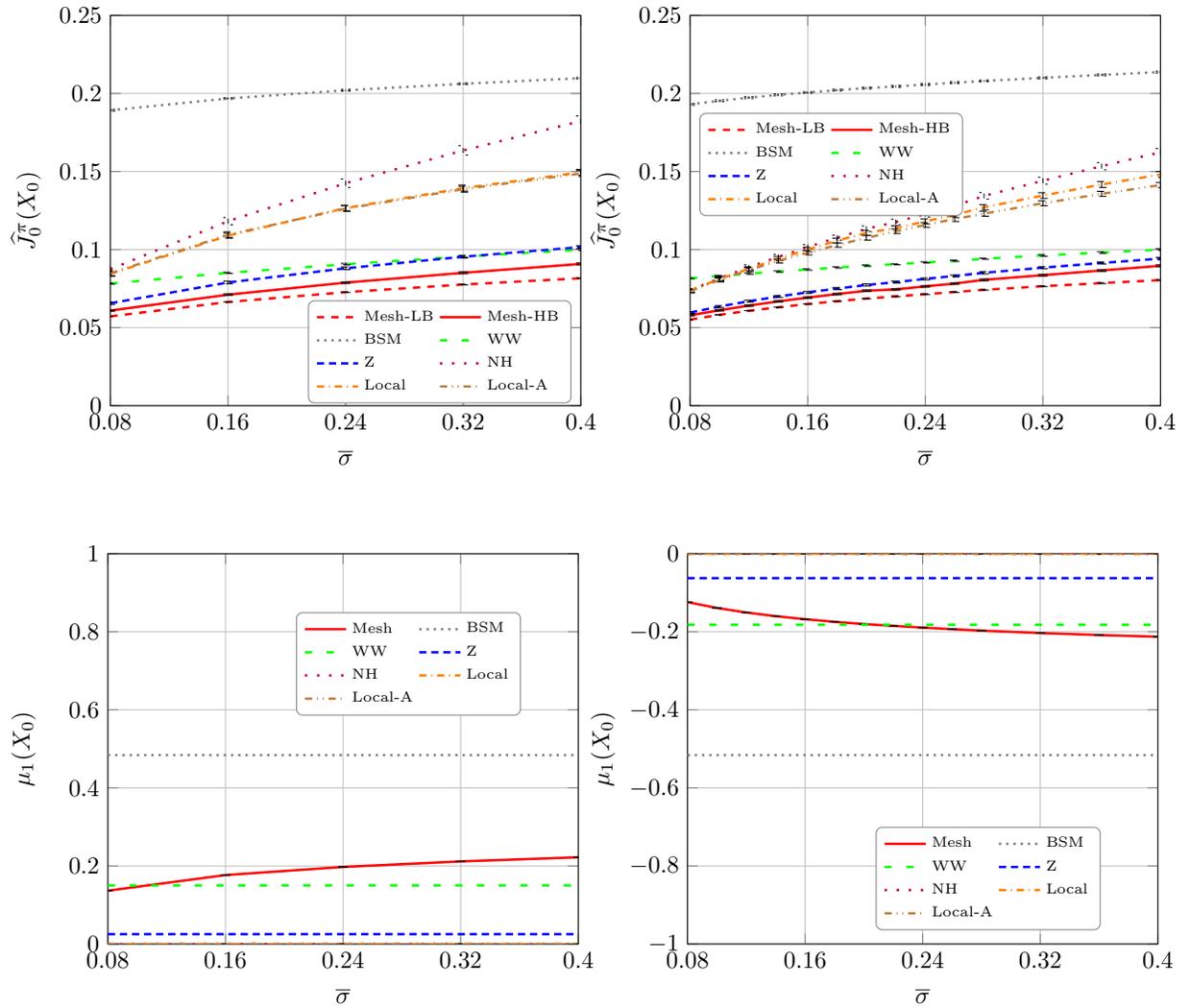


Figure 7.8: Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the mean volatility parameter $\bar{\sigma}$. Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 1024$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

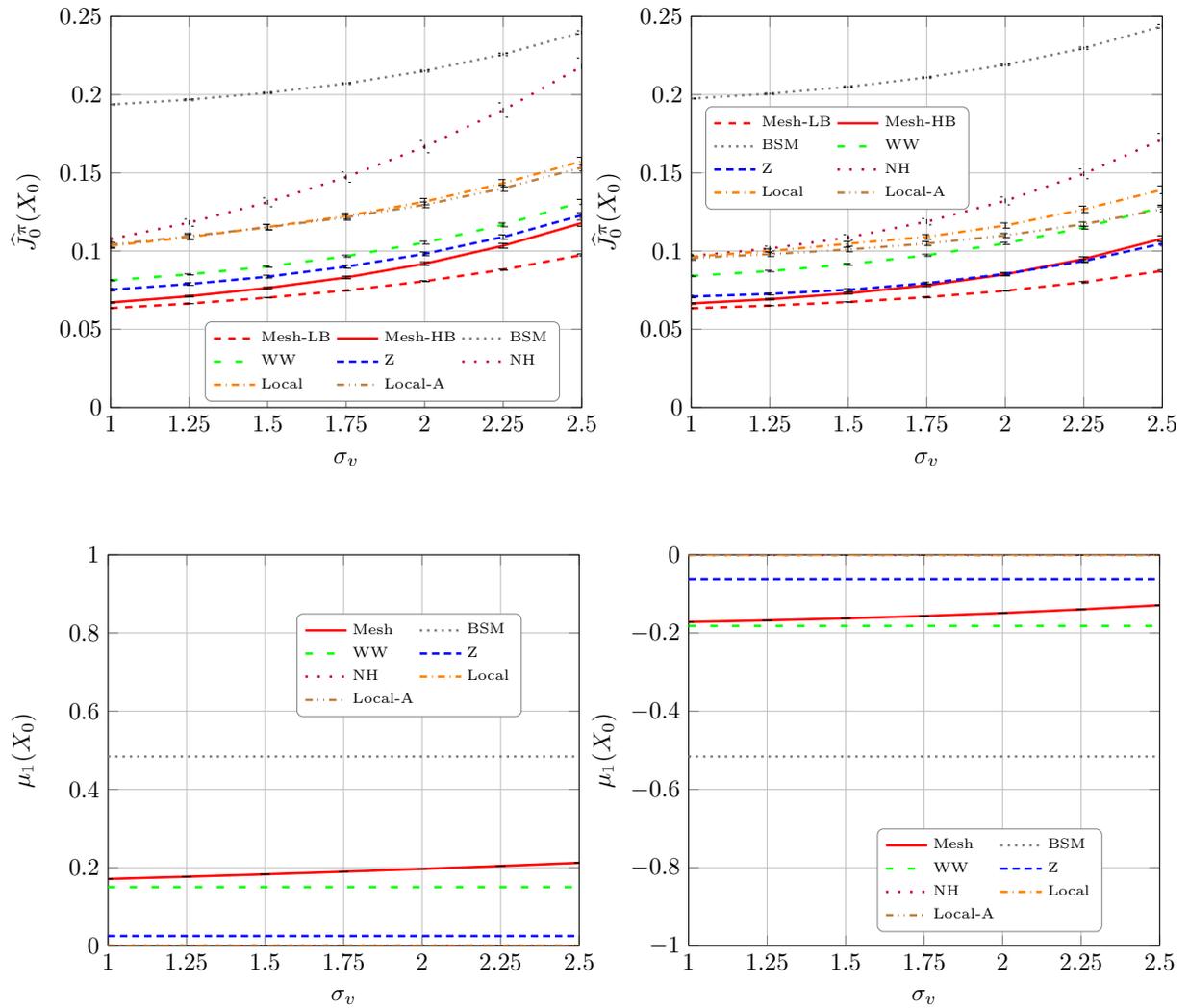


Figure 7.9: Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the volatility of log-volatility parameter σ_v . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\kappa = 2.5$, $\sigma_v = 1.25$, $\rho = -0.4$, $T = 3$ months, $K = 16$, $b = 1\%$, $N = 1024$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

impact is less pronounced in the case of the call option.

Overall, the similarity between the effects of T and the volatility parameters makes sense, since if we assume that for the model of section 4.1 (which we are using) the volatility level σ_k is roughly equal to a constant σ , then we would have that the variance of the (log-normal) distribution of prices at time $t_K = T$ is approximately given by $s_0^2(e^{\sigma^2 T} - 1) \approx s_0^2 \sigma^2 T$ (assuming zero interest rates and dividends, i.e. $r = q = 0$). Hence increasing the value of T for a fixed volatility σ can be interpreted as increasing the volatility level σ for a fixed T .

As for the behavior of the heuristics, notice that WW does more poorly when T is small. Since the NH policy is close to being optimal for small T , this indicates that it is best to simply not hedge when close to the expiry, although the precise time at which this starts to be true could depend on other parameters such as the stock price s_k , the stock quantity held u_k or the proportional costs b .

7.2.7 Rate of log-volatility mean reversion κ

Figure 7.11 shows the effect of various values of the parameter κ , the rate of log-volatility mean reversion. As for some other parameters such as $\bar{\sigma}$ or σ_v , the various heuristics are independent of κ , by definition, so the associated decisions μ_0 appear as straight lines in the two lower charts. The mesh based policy does appear to have a dependence on κ , but it is barely noticeable for the setup considered. One possible reason is that the initial volatility σ_0 is equal to the long-run volatility $\bar{\sigma}$. Hence the volatility level σ_k does not have a chance to deviate significantly enough from $\bar{\sigma}$ for the mean-reversion to take effect. So either using $\sigma_0 \gg \bar{\sigma}$ (say) or a higher level for the volatility of log-volatility σ_v could reveal a stronger effect, but we have not tested such setups in our experiments. Note that in table 7.III of section 7.1, we also had $\sigma_0 = \bar{\sigma}$, but different combinations of (κ, σ_v) were shown. There was a poor performance of the mesh method associated to high σ_v when κ was high (especially

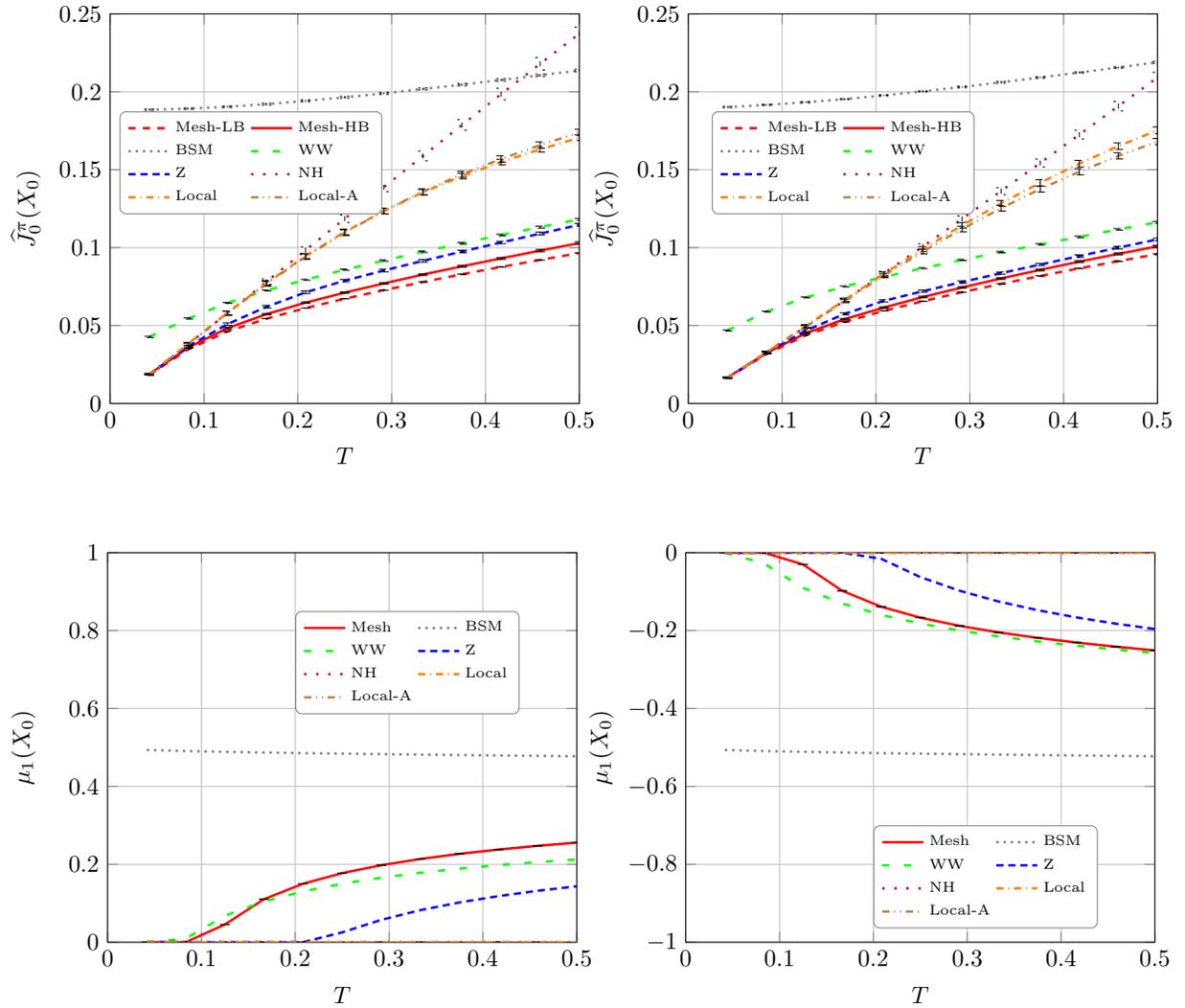


Figure 7.10: Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the option's time to expiry T . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 4096$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

for $\rho = -0.8$), and there was a good performance for high σ_v when κ was low.

7.2.8 Correlation ρ

Figure 7.12 shows the effect of the correlation parameter ρ between the price log-returns and the volatility log-returns (see again the model definition in section 4.1). Only negative correlations are considered here (i.e. a drop in prices is associated with an increase in volatility). Note that the first data point on the left is for $\rho = -0.99$ and not $\rho = -1$, to avoid numerical difficulties. Still, there is a large bias in the associated mesh based estimator for $\rho = -0.99$, as can be seen from the gap between Mesh-LB and Mesh-HB risk estimates on the two upper graphs.

There is a noticeable impact from ρ on the performance of the mesh based policy and the various heuristics. Overall, for $\rho \in [-0.9, 0]$, we see that the mesh based policy does improve on the heuristics, especially in the case of the put option, with high negative correlation parameter (e.g. $\rho = -0.9$). This makes sense because a put option is mostly exposed to price movements when the stock price s_k goes below the option's strike price X (i.e. becomes *in-the-money*, in financial jargon). So this should be associated with an increase in the level of volatility, and hence in an increase of the portfolio risk. In the case of the call option, the Z policy performs nearly as well as the mesh based policy, which is also plausible because then the negative correlation ρ has the opposite effect. That is, for the option to be in-the-money, the stock price s_k has to go above than the strike price X , so these positive price movements will tend to be associated with a decrease in volatility.

7.2.9 Risk aversion γ

Figure 7.13 shows the effect of varying the risk aversion parameter γ . Note that it is on a log-scale, to show various orders of magnitude of the parameter γ .

For low values of γ , the NH, Z and Local-A policies have similar risk estimates

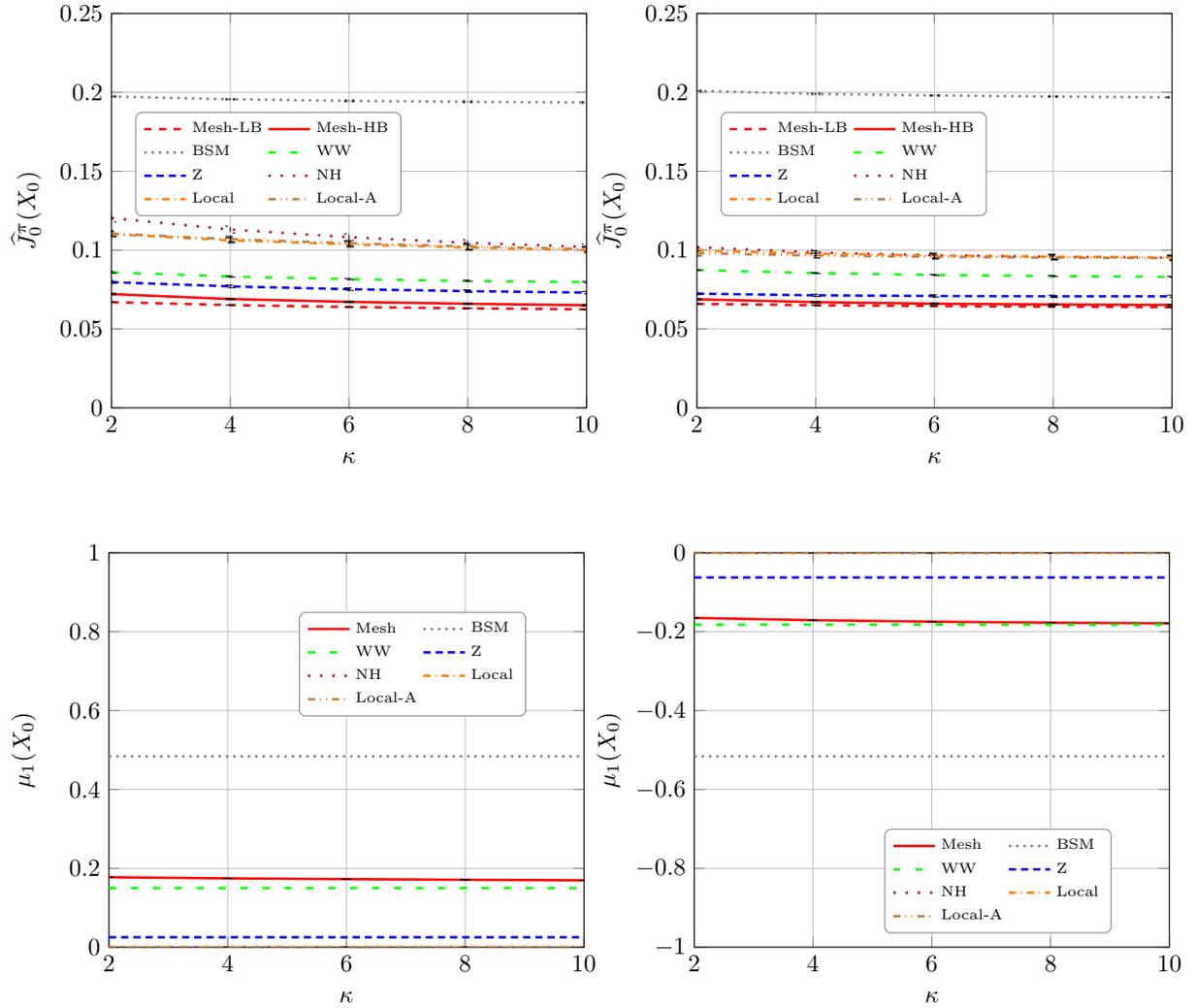


Figure 7.11: Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the rate of log-volatility mean reversion parameter κ . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\rho = -0.4$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 4096$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

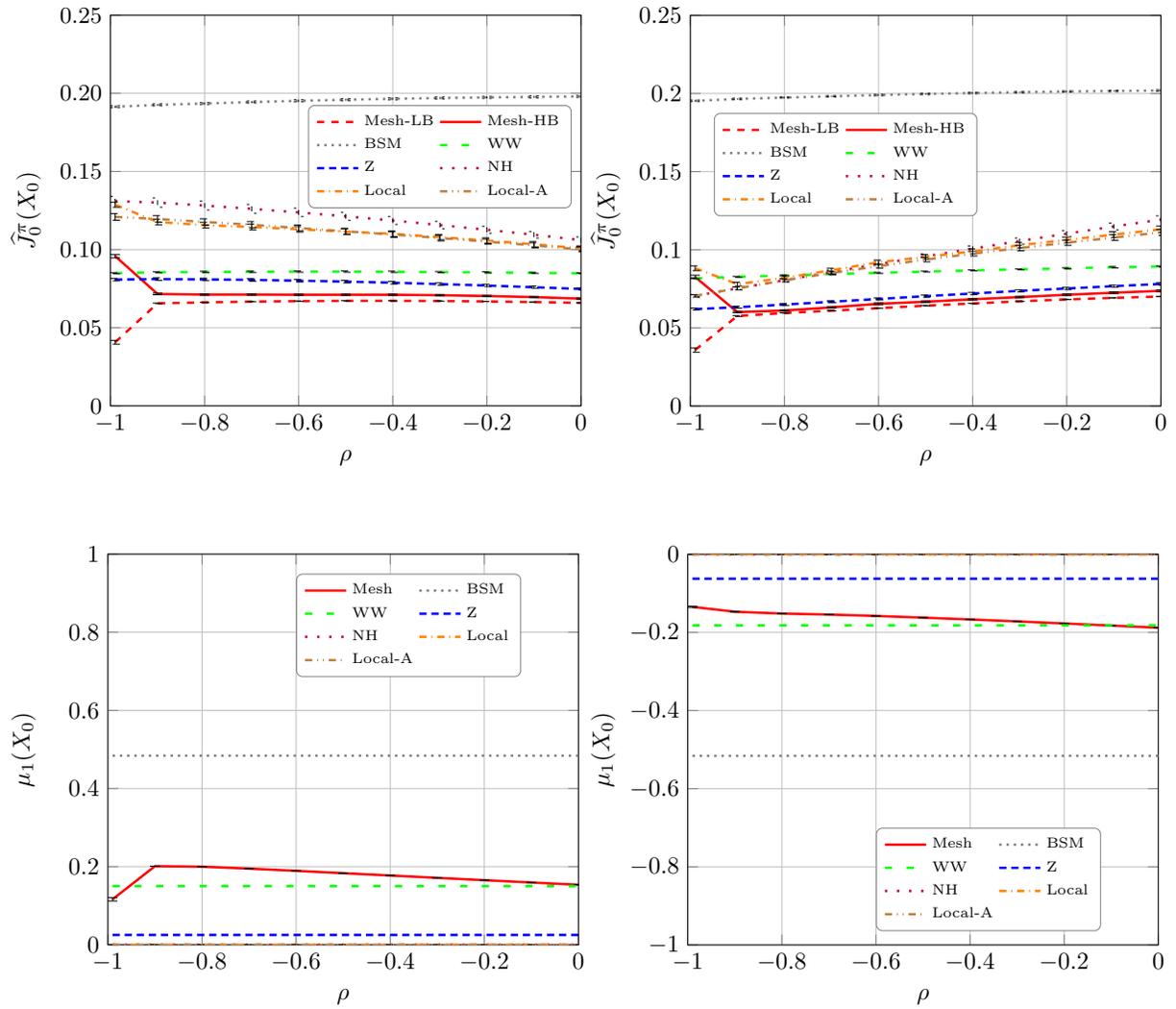


Figure 7.12: Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the correlation parameter ρ . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\kappa = 2.5$, $\sigma_v = 1.25$, $T = 3$ months, $K = 16$, $b = 1\%$, $\gamma = 1$, $N = 4096$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

and dominate the alternative heuristics and the mesh based policy. This indicates that the Z and Local-A policies mostly do not hedge in that case. The Mesh-HB and Local risk estimates are very close for low γ , which in turn points to those two policies being approximately equal. So even though the mesh based policy uses Dynamic Programming over the full number of steps K , this becomes roughly equivalent to the one-step Local hedging policy for small γ . In contrast, for high values of γ (say $\gamma \geq 10$), the mesh based policy clearly dominates the other heuristics, and in particular Local hedging.

The WW policy is closer to the optimum when γ is higher, under-performing for low γ . But recall that WW is a *low cost* approximation, not low γ . In contrast, the best setup for the Local and Local-A policies is γ to be very low (even $\gamma = 1$ is still too high), which is expected since they were defined as approximations for low γ .

Recall also that for low γ , equation (2.7) implies that the negative exponential risk is approximately equal to :

$$\mathbf{E}_0[L(V_K^\pi - V_0)] \approx -\mathbf{E}_0[V_K^\pi - V_0] + \frac{\gamma}{2}\mathbf{E}_0[(V_K^\pi - V_0)^2], \quad (7.1)$$

which can be interpreted as a tradeoff between the expectation and the second moment about 0 of the portfolio performance (or variance if $\mathbf{E}_0[V_K^\pi]$ is not too far from V_0). This tradeoff can be seen in figure 7.14, which represents the sample mean $m(V_K^\pi)$ and sample standard deviation $s(V_K^\pi)$ of the terminal portfolio value V_K^π . Each curve corresponds to a different policy π , and each point of a given curve corresponds to a different value for the risk aversion parameter γ . The top graph shows the case $K = 16$ for a call option, for which most curves appear similar (with the exception of the WW policy), whereas the bottom graph shows the case $K = 64$ for a put option, for which there is much more differentiation between results for the various policies.

In the mean-variance framework of modern portfolio theory (introduced by Markowitz [100]), portfolios that maximize expected return for a given standard deviation should

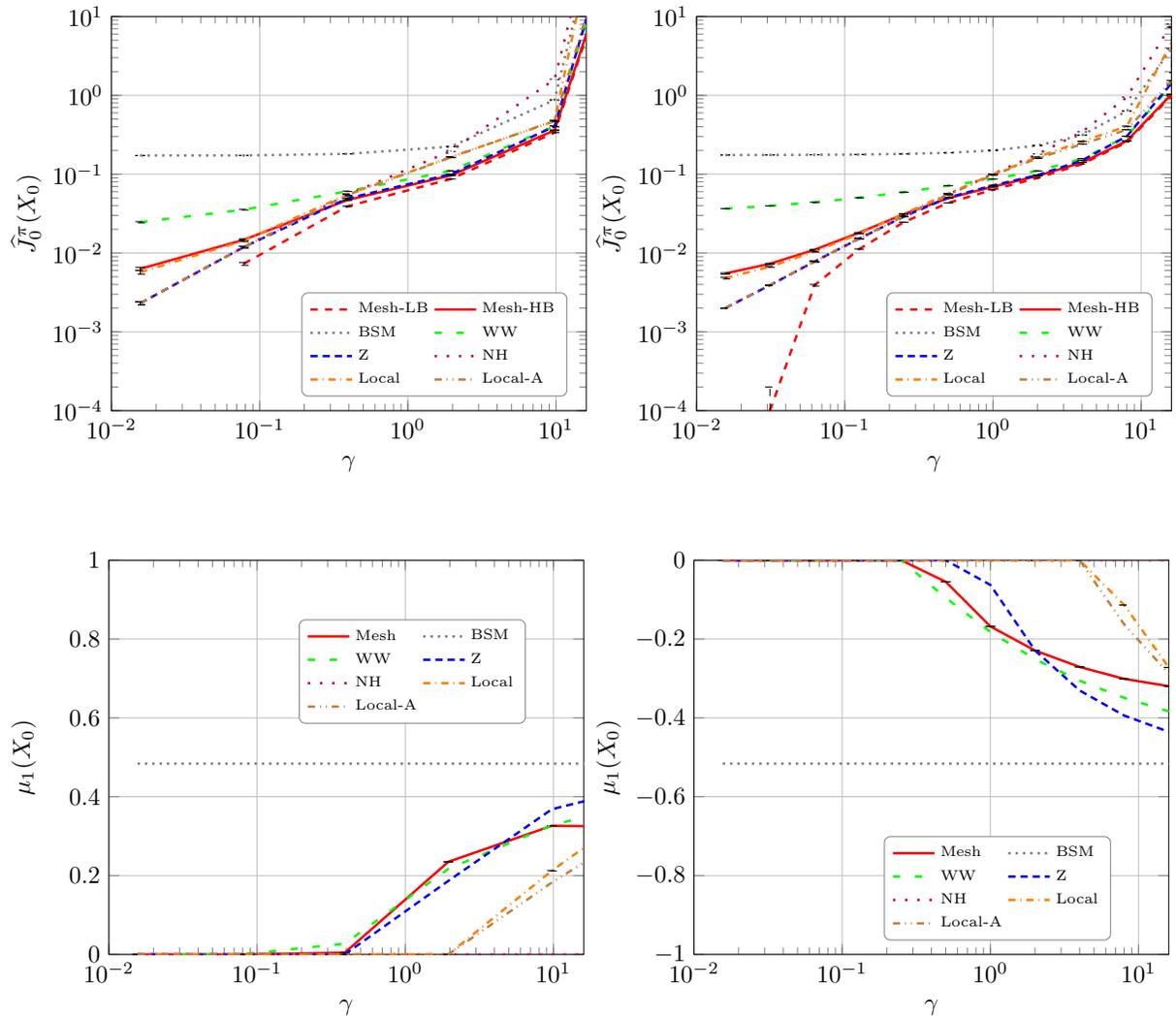


Figure 7.13: Risk estimate $\widehat{J}_0^\pi(X_0)$ and initial decision $\mu_1(X_0)$ for various policies π as a function of the risk aversion parameter γ . Left-hand side : put option, right-hand side : call option. Setup : $s_0 = 10$, $\sigma_0 = \bar{\sigma} = 16\%$, $\kappa = 2.5$, $\sigma_v = 1.25$, $\rho = -0.4$, $T = 3$ months, $K = 16$, $b = 1\%$, $N = 1024$ and $n = n_R = 100$. Error bars indicate one standard deviation of the sample averages.

be preferred (i.e. *efficient portfolios*). Under this criterion, both Local and Local-A actually do very well in figure 7.14, which is expected because they are designed for low γ , i.e. a case where the negative exponential risk objective becomes approximately equivalent to a mean-variance objective. But this objective does not take into account higher moments of the portfolio performance $V_K^\pi - V_0$, as opposed to the negative exponential risk function. Note that the NH policy results appear as a point at the top right of the graphs, corresponding to very low costs, but high variance, whereas the BSM policy results appear as a point at the bottom left, corresponding to low variance but high costs.

7.3 Conclusion

Generally speaking, we cannot expect the heuristic policies tested here, such as BSM, WW and Z, to do well in every setup, since they were designed to work under the GBM price dynamics and for small or no costs. Conversely, for any setup with a high enough number of mesh points N , the stochastic mesh based policy is supposed to dominate the heuristics, since it should converge to the optimal policy as N increases. Given the numerical evidence from the previous sections, here are the cases which could most benefit from using approximately optimal policies (such as stochastic mesh based policies), as opposed to the tested heuristics :

- Higher costs b ,
- Higher risk aversion γ ,
- Longer horizon T , larger volatility parameters $\sigma_0, \bar{\sigma}$,
- Put options (as opposed to call options), and
- Smaller number of time steps K

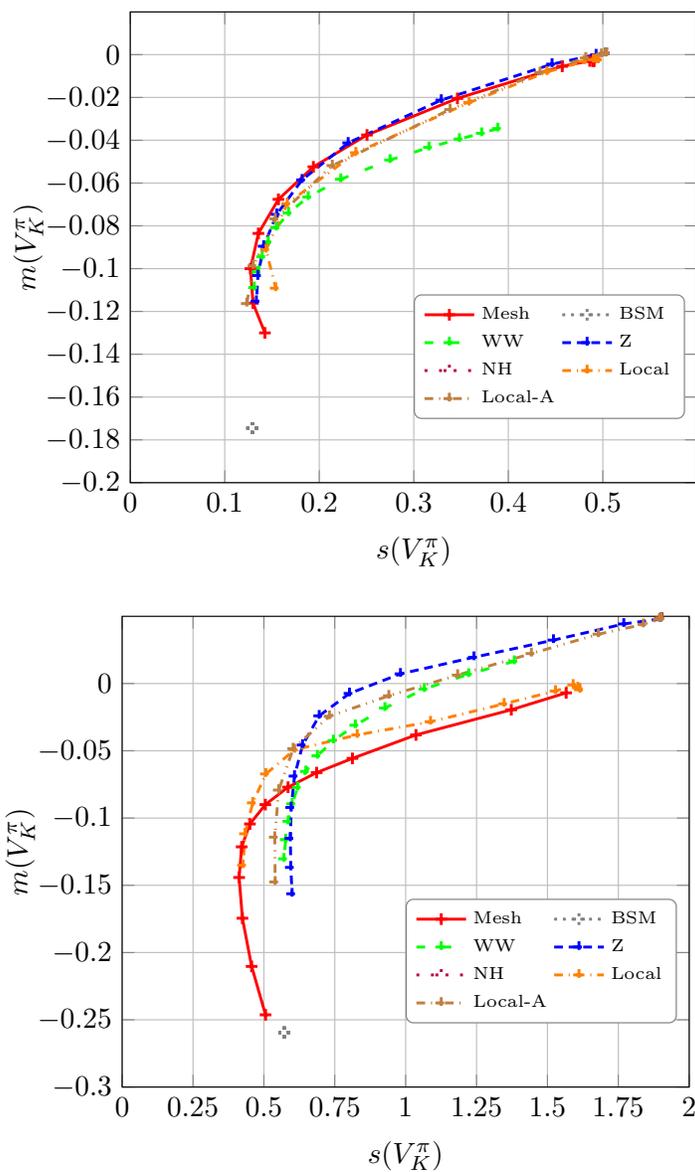


Figure 7.14: Average portfolio value at expiry time T as a function of its standard deviation, for various values of the risk aversion parameter γ . Top : call option for $K = 16$, $\sigma_0 = \bar{\sigma} = 16\%$, $\kappa = 2.5$, $\sigma_v = 1.25$, $\rho = -0.4$. Bottom : put option for $K = 64$, $\sigma_0 = \bar{\sigma} = 40\%$, $\kappa = 2.5$, $\sigma_v = 2.5$, $\rho = -0.8$. For both : $s_0 = 10$, $T = 3$ months, $b = 1\%$, $N = 1024$, $n = 100$, $n_R = 1000$. Error bars indicate one standard deviation of the sample averages.

For high K and no costs ($b = 0$), results for the stochastic mesh method are generally poor (for a fixed value of N), i.e. the bias is high for the estimates Mesh-LB and Mesh-HB. But in that context, computing an optimal policy over many steps appears unnecessary. Indeed, although convergence of the mesh-based policy is still obtained eventually, local hedging was shown to yield an optimal policy (see section 4.2.3), but at a much smaller computational cost since there is no need to compute a full stochastic mesh.

In practice, the heuristics can still do very well compared to the mesh based method when there is a strong time constraint, because of the reduced noise. The noise in the mesh based policy can be reduced by increasing the number of mesh nodes N , but with an accompanying increase in computation time. One possibility around this problem is to compute the optimal policies offline (i.e. overnight). It is not realistic to assume the option positions to hedge will not change daily, but it could still be useful to precompute approximately optimal policies if the option positions to hedge don't change too much. That is, one could bear a small mishedging cost for one day.

The performance of the Z policy is the closest to mesh-based policy in the case of the call option in general, for both the GBM and expOU models, but less so in the case of the put option. By construction, Z is optimized on a call option, so this can be seen as an example of what can happen if a policy is optimized on one derivative and applied to another. Furthermore, the difference in performance is more pronounced for the general expOU model than for the GBM examples. This indicates that the additional market risk coming from the stochastic volatility in the expOU model does have an impact on the portfolio risk, which can somewhat be reduced by the optimal policy (as approximated by the mesh-based policy).

CHAPTER 8

CONCLUSION

Building on existing works, this thesis has provided a new methodology for computing the optimal solutions to dynamic hedging problems in discrete time, including costs. Our motivation was to be able to compute such hedging policies for various underlying processes and derivative payoffs. But, knowing that the resulting policies would only be approximately optimal, we also wanted to have some kind of lower bound on the risk, to ascertain that the policies could not be improved upon by much more.

The central innovation was the application of stochastic mesh methods in this context, and the use of the associated high and low biased estimators. A realistic example, based on a stochastic volatility process and an exponential utility objective, was worked out in detail. Furthermore, we studied suitable efficiency improvement techniques (single grid, Russian roulette and RQMC) that provided appreciable computation time reductions for a given level of precision, especially when used simultaneously. But as was apparent along the various chapters, the hedging problem has many components, and the analysis from this thesis could be extended in various directions.

In terms of applying the methods presented here, using a different stochastic process for the underlying stock should not cause any additional difficulty, provided the transition densities are known (see section 3.2.3). In this regard, it would be interesting to try stochastic mesh hedging on interest rate derivatives (such as swaptions, for example), where the state space is naturally multi-dimensional when modelling the dynamics of the yield curve and its associated volatility.

As for the choice of the loss function, another natural case to consider for stochas-

tic mesh hedging is that of the quadratic loss function, or the portfolio variance. Such a function is much faster to compute than the negative exponential loss function and is widely used as a risk measure. But this would require to take the cash variable into account into the state during the DP recursion, whereas this was not necessary when the negative exponential loss function was considered in our examples. Thus a suitable two-dimensional approximation would have to be used for the risk function for a given value of the market information vector Y_k , instead of the one-dimensional approximation used in this thesis.

Another extension in this vein could be to allow hedging with more than one instrument. But then again, the challenge is that a higher dimensional approximation would be required to keep track of the effect of the quantities held of the various hedging instruments at each step.

BIBLIOGRAPHY

- [1] Commons Math: The Apache Commons Mathematics Library, August 2016. Available at <http://commons.apache.org/proper/commons-math/>.
- [2] C. Albanese and S. Tompaidis. Small transaction cost asymptotics and dynamic hedging. *European Journal of Operational Research*, 185:1404–1414, March 2008.
- [3] C. D. Aliprantis and K. C. Border. *Infinite Dimensional Analysis : A Hitchhiker's Guide*. Springer, Berlin, 2006.
- [4] E. D. Andersen and A. Damgaard. Utility based option pricing with proportional transaction costs and diversification problems: An interior-point optimization approach. *Applied Numerical Mathematics*, 29(3):395–422, 1999.
- [5] L. Andersen. Simple and efficient simulation of the Heston stochastic volatility model. *Journal of Computational Finance*, 11(3):1–42, Spring 2008.
- [6] M. Avellaneda and P. Jäckel. Weighted Monte Carlo. In *Encyclopedia of Quantitative Finance*. John Wiley & Sons, Ltd, 2010.
- [7] A. N. Avramidis and P. Hyden. Efficiency improvements for pricing American options with a stochastic mesh. In P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, editors, *Proceedings of the 1999 Winter Simulation Conference*, pages 344–350, Piscataway, New Jersey, 1999. IEEE Press.
- [8] A. N. Avramidis and H. Matzinger. Convergence of the stochastic mesh estimator for pricing American options. In E. Yücesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, editors, *Proceedings of the 2002 Winter Simulation Conference*, pages 1560–1567, Piscataway, New Jersey, 2002. IEEE Press.

- [9] G. Barles and H. M. Soner. Option pricing with transaction costs and a non-linear Black-Scholes equation. *Finance and Stochastics*, 2:369–397, 1998.
- [10] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.
- [11] H. Ben Ameur, P. L’Ecuyer, and C. Lemieux. Variance reduction of Monte Carlo and randomized quasi-Monte Carlo estimators for stochastic volatility models in finance. In *Proceedings of the 1999 Winter Simulation Conference*, pages 336–343. IEEE Press, 1999.
- [12] H. Ben Ameur, M. Breton, and P. L’Ecuyer. A dynamic programming procedure for pricing American-style Asian options. *Management Science*, 48: 625–643, 2002.
- [13] H. Ben Ameur, M. Breton, L. Karoui, and P. L’Ecuyer. A dynamic programming approach for pricing options embedded in bonds. *Journal of Economic Dynamics and Control*, 31:2212–2233, 2007.
- [14] B. Bensaïd, J.-P. Lesne, H. Pagès, and J. Scheinkman. Derivative asset pricing with transaction costs. *Mathematical finance*, 2(2):63–86, April 1992.
- [15] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Volume I*. Athena Scientific, Belmont, Massachusetts, second edition, 2000.
- [16] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Volume I*. Athena Scientific, Belmont, Mass., third edition, 2005.
- [17] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Volume II*. Athena Scientific, Belmont, Mass., third edition, 2007.

- [18] D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control : The Discrete Time Case*, volume 139 of *Mathematics in Science and Engineering*. Academic Press, New York, 1978.
- [19] P. Billingsley. *Probability and measure*. John Wiley & Sons, New York, third edition, 1995.
- [20] J. R. Birge. Unattained convergence for sampling methods in large-scale optimization models and a remedy with batch means. *INFORMS Simulation Society Research Workshop*, 2007.
- [21] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 1997.
- [22] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–654, 1973.
- [23] R. R. Bliss and N. Panigirtzoglou. Option-implied risk aversion estimates. *The Journal of Finance*, 59(1):407–446, February 2004.
- [24] O. Bobrovnytska and M. Schweizer. Mean-variance hedging and stochastic control : Beyond the Brownian setting. *IEEE Transactions on Automatic Control*, 49(3):396–408, March 2004.
- [25] D. J. Bolder and T. Rubin. Optimization in a simulation setting: Use of function approximation in debt strategy analysis. Working Paper, 2007.
- [26] N. Bolia and S. Juneja. Function-approximation-based perfect control variate for pricing American options. In *Proceedings of the 2005 Winter Simulation Conference*. IEEE Press, 2005.

- [27] K. A. Boyle, T. F. Coleman, and Y. Li. Hedging a portfolio of derivatives by modelling cost. In *Proceedings of the 2003 International Conference on Computational Intelligence for Financial Engineering*. IEEE Press, 2003.
- [28] P. Boyle, J. Imai, and K. S. Tang. Computation of optimal portfolios using simulation-based dimension reduction. *Insurance: Mathematics and Economics*, 43(3):327–338, 2008.
- [29] P. P. Boyle and T. Vorst. Option replication in discrete time with transaction costs. *The Journal of Finance*, 47(1):271–293, March 1992.
- [30] P. P. Boyle, A. W. Kolkiewicz, and K. S. Tan. Pricing American style options using low discrepancy mesh methods. IIPR - Research report 00-07, November 2000.
- [31] P. P. Boyle, A. W. Kolkiewicz, and K. S. Tan. Pricing American derivatives using simulation: A biased low approach. In K.-T. Fang, F.J. Hickernell, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 181–200, Berlin, 2002. Springer-Verlag.
- [32] P. P. Boyle, A. W. Kolkiewicz, and K. S. Tan. An improved simulation method for pricing high-dimensional American derivatives. *Mathematics and Computers in Simulation*, 62:315–322, March 2003.
- [33] M. W. Brandt, A. Goyal, P. Santa-Clara, and J. R. Stroud. A simulation approach to dynamic portfolio choice with an application to learning about return predictability. *Review of Financial Studies*, 18:831–873, 2005.
- [34] M. Broadie and P. Glasserman. Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, 21:1323–1352, 1997.

- [35] M. Broadie and P. Glasserman. A stochastic mesh method for pricing high-dimensional American options. PaineWebber Working papers in Money, Economics and Finance #PW9804, 1997.
- [36] M. Broadie and P. Glasserman. A stochastic mesh method for pricing high-dimensional American options. *The Journal of Computational Finance*, 7(4): 35–72, 2004.
- [37] M. Broadie, P. Glasserman, and Z. Ha. Pricing American options by simulation using a stochastic mesh with optimized weights. In S.P. Uryasev, editor, *Probabilistic Constrained Optimization: Methodology and Applications*, pages 32–50. Kluwer Academic Publishers, 2000.
- [38] P. Christoffersen and K. Jacobs. The importance of the loss function in option valuation. *Journal of Financial Econometrics*, 72:291–318, 2004.
- [39] P. Christoffersen, S. Heston, and K. Jacobs. The shape and term structure of the index option smirk: why multifactor stochastic volatility models work so well. *Management Science*, 55:1914–1932, 2009.
- [40] L. Clewlow and S. Hodges. Optimal delta-hedging under transaction costs. *Journal of Economic Dynamics and Control*, 21:1353–1376, 1997.
- [41] R. Cont and P. Tankov. *Financial modelling with jump processes*. Chapman & Hall/CRC, 2004.
- [42] V. Corradi, W. Distaso, and A. Mele. Macroeconomic determinants of stock volatility and volatility premiums. *Journal of Monetary Economics*, 60:203–220, 2013.
- [43] J. Cvitanić and I. Karatzas. Hedging and portfolio optimization under transaction costs : a martingale approach. *Mathematical Finance*, 6:113–165, 1996.

- [44] J. Cvitanić, L. Goukasian, and F. Zapatero. Hedging with Monte Carlo simulation. In E. Kontoghiorghes, B. Rustem, and S. Siokos, editors, *Computational methods in decision-making, economics and finance*, pages 339–353. Kluwer academic publishers, 2002.
- [45] J. Cvitanić, L. Goukasian, and F. Zapatero. Monte Carlo computation of optimal portfolios in complete markets. *Journal of Economic Dynamics and Control*, 27:971–986, 2003.
- [46] R. C. Dalang, A. Morton, and W. Willinger. Equivalent martingale measures and no-arbitrage in stochastic securities market models. *Stochastics and Stochastic Reports*, 29(2):185–201, 1990.
- [47] M. H. A. Davis, V. G. Panas, and T. Zariphopoulou. European option pricing with transaction costs. *SIAM Journal of Control and Optimization*, 31(2): 470–493, March 1993.
- [48] C. de Boor. *A Practical Guide to Splines*. Springer Verlag, New York, 1978.
- [49] F. Delbaen and W. Schachermayer. *The mathematics of arbitrage*. Springer-Verlag, Germany, 2006.
- [50] J. B. Detemple, R. Garcia, and M. Rindisbacher. A Monte Carlo method for optimal portfolios. *Journal of Finance*, 58:401–446, 2003.
- [51] M. Dion and P. L’Ecuyer. American option pricing with randomized quasi-Monte Carlo simulation. In *Proceedings of the 2010 Winter Simulation Conference*, Piscataway, NJ, 2010. IEEE Press.
- [52] J.-C. Duan and J.-G. Simonato. Empirical martingale simulation for asset prices. *Management Science*, 44(9):1218–1233, september 1998.

- [53] C. Edirisinghe, V. Naik, and R. Uppal. Optimal replication of options with transaction costs and trading restrictions. *Journal of financial and quantitative analysis*, 28(1):117–138, March 1993.
- [54] R. F. Engle, E. Ghysels, and B. Sohn. Stock market volatility and macroeconomic fundamentals. *The Review of Economics and Statistics*, 95(3):776–797, July 2013.
- [55] J.-P. Fouque, G. Papanicolaou, and K. R. Sircar. Mean-reverting stochastic volatility. *International Journal of Theoretical and Applied Finance*, 3(1):101–142, 2000.
- [56] J.-P. Fouque, G. Papanicolaou, and K. R. Sircar. *Derivatives in Financial Markets with Stochastic Volatility*. Cambridge University Press, New York, 2000.
- [57] M. C. Fu, S. B. Laprise, D. B. Madan, Y. Su, and R. Wu. Pricing American options: A comparison of Monte Carlo simulation approaches. *Journal of Computational Finance*, 4(3):39–88, Spring 2001.
- [58] J. Gatheral. *The volatility surface*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2006.
- [59] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York, 2004.
- [60] P. Glasserman and B. Yu. Large sample properties of weighted monte carlo estimators. *Operations Research*, 53(2):298–312, March-April 2005.
- [61] P. Glasserman, P. Heidelberger, and P. Shahabuddin. Asymptotically optimal importance sampling and stratification for pricing path dependent options. *Journal of Mathematical Finance*, 9(2):117–152, 1999.

- [62] P. W. Glynn and R. Szechtman. Some new perspectives on the method of control variates. In K.T. Fang, F.J.Hickernell, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 27–49. Springer Verlag, 2002.
- [63] J. Gondzio, R. Kouwenberg, and T. Vorst. Hedging options under transaction costs and stochastic volatility. *Journal of Economic Dynamics and Control*, 27(6):1045–1068, 2003.
- [64] M. R. Grasselli and T. R. Hurd. A Monte Carlo method for exponential hedging of contingent claims. 2004. Working paper.
- [65] B. Grünewald and S. Trautmann. Varianzminimierende Hedgingstrategien für Optionen bei möglichen Kurssprüngen. *Zeitschrift für betriebswirtschaftliche Forschung*, Sonderheft 38:43–87, 1997.
- [66] A. C. Harvey and N. Shephard. Estimation of an asymmetric stochastic volatility model for asset returns. *Journal of Business & Economic Statistics*, 14(4):429–434, October 1996.
- [67] E. G. Haugh and N. N. Taleb. Option traders use (very) sophisticated heuristics, never the Black-Scholes-Merton formula. *Journal of Economic Behavior and Organization*, 77(2):97–106, 2011.
- [68] A. Haurie and P. L’Ecuyer. Approximation and bounds in discrete event dynamic programming. *IEEE Trans. on Automatic Control*, 31(3):227–235, 1986.
- [69] D. Heath, E. Platen, and M. Schweizer. A comparison of two quadratic hedging approaches to hedging in incomplete markets. *Mathematical Finance*, 11(4):385–413, October 2001.

- [70] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6: 327–343, 1993.
- [71] S. D. Hodges and A. Neuberger. Optimal replication of contingent claims under transaction costs. *Review of Futures Markets*, 8:222–239, 1989.
- [72] Wolfgang Hoschek. *The Colt Distribution: Open Source Libraries for High Performance Scientific and Technical Computing in Java*. CERN, Geneva, 2004. Available at <http://acs.lbl.gov/software/colt/>.
- [73] J. Hull and A. White. The pricing of options on assets with stochastic volatilities. *Journal of Finance*, 42:281–300, 1987.
- [74] E. Jacquier, N. G. Polson, and P. E. Rossi. Bayesian analysis of stochastic volatility models with fat-tails and correlated errors. *Journal of Econometrics*, 122(1):185–212, september 2004.
- [75] H. Kahn and T. E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards Applied Mathematical Series*, 12:27–30, 1951.
- [76] J. Kallsen and J. Muhle-Karbe. Option pricing and hedging with small transaction costs. *Mathematical Finance*, 25(4):702–723, 2015.
- [77] I. Karatzas and S. E. Shreve. *Methods of Mathematical Finance*. Springer, New York, 1998.
- [78] J. S. Kennedy, P. A. Forsyth, and K. R. Vetzal. Dynamic hedging under jump diffusion with transaction costs. *Operations Research*, 57(3):541–559, May 2009.

- [79] J. Keppo and S. Peura. Optimal portfolio hedging with nonlinear derivatives and transaction costs. *Computational Economics*, 13:117–145, 1999.
- [80] P. Klaassen. Discretized reality and spurious profits in stochastic programming models for asset/liability management. *European Journal of Operational Research*, 101(2):374–392, 1997.
- [81] D. Kramkov and W. Schachermayer. The asymptotic elasticity of utility functions and optimal investment in incomplete markets. *The Annals of Applied Probability*, 9(3):904–950, 1999.
- [82] D. Kramkov and W. Schachermayer. Necessary and sufficient conditions in the problem of optimal investment in incomplete markets. *The Annals of Applied Probability*, 13(4):1504–1516, 2003.
- [83] D. Lamberton, H. Pham, and M. Schweizer. Local risk-minimization under transaction costs. *Mathematics of Operations Research*, 23:585–612, 1998.
- [84] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, third edition, 2000.
- [85] P. L’Ecuyer. Computing approximate solutions to markov renewal programs with continuous state spaces. Technical Report DIUL-RT-8912, Département d’informatique, Univ. Laval, Canada, 1989. URL <http://www.iro.umontreal.ca/~lecuyer/papers.html>.
- [86] P. L’Ecuyer. Quasi-Monte Carlo methods with applications in finance. *Finance and Stochastics*, 13(3):307–349, 2009.
- [87] P. L’Ecuyer. IFT6521, Programmation dynamique. Lecture Notes, 2013. URL <http://www.iro.umontreal.ca/~lecuyer/ift6521.html>.

- [88] P. L'Ecuyer. IFT6561, Stochastic simulation. Lecture Notes, November 2016. URL <https://www.iro.umontreal.ca/~lecuyer/ift6561/prive/book.pdf>.
- [89] P. L'Ecuyer. SSJ: Stochastic simulation in Java, software library, 2016. <http://simul.iro.umontreal.ca/ssj/>.
- [90] P. L'Ecuyer and E. Buist. On the interaction between stratification and control variates, with illustrations in a call center simulation. *Journal of Simulation*, 2(1):29–40, 2008.
- [91] P. L'Ecuyer, A. Haurie, and A. Hollander. Optimal research and development expenditures under an incremental tax incentive scheme. *Operations Research Letters*, 4(2):85–90, 1985.
- [92] P. L'Ecuyer, L. Meliani, and J. Vaucher. SSJ: A framework for stochastic simulation in Java. In E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, editors, *Proceedings of the 2002 Winter Simulation Conference*, pages 234–242. IEEE Press, 2002. Available at <http://simul.iro.umontreal.ca/ssj/indexe.html>.
- [93] P. L'Ecuyer, V. Demers, and B. Tuffin. Splitting for rare-event simulation. In L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, editors, *Proceedings of the 2006 Winter Simulation Conference*, pages 137–148. IEEE Press, 2006.
- [94] H. Leland. Option pricing and replication with transaction costs. *The Journal of Finance*, 40(5):1283–1301, 1985.
- [95] C. Lemieux and J. La. A study of variance reduction techniques for American option pricing. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A.

- Joines, editors, *Proceedings of the 2005 Winter Simulation Conference*, pages 1884–1891, 2005.
- [96] C. Lemieux, M. Cieslak, and K. Luttmer. Randqmc user’s guide: A package for randomized quasi-monte carlo methods in c. Software user’s guide, 2004. URL <http://www.math.uwaterloo.ca/~clemieux/randqmc.html>.
- [97] G. Liu and L. J. Hong. Revisit of stochastic mesh method for pricing American options. *Operations Research Letters*, 37:411–414, 2009.
- [98] F. A. Longstaff and E. S. Schwartz. Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001.
- [99] R. Lord, R. Koekkoek, and D. van Dijk. A comparison of biased simulation schemes for stochastic volatility models. *Quantitative Finance*, 10(2):177–194, 2010.
- [100] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.
- [101] L. Martinelli and P. Priaulet. Competing methods for option hedging in the presence of transaction costs. *Journal of Derivatives*, 9(3):26–38, 2002.
- [102] J. Masoliver and J. Perelló. Multiple time scales and the exponential Ornstein-Uhlenbeck stochastic volatility model. *Quantitative finance*, 6(5):423–433, 2006.
- [103] E. J. McGrath and D. C. Irving. *Techniques for efficient Monte Carlo Simulation, Volume III, Variance reduction*. Oak Ridge national laboratory, Oak Ridge, Tennessee, 1975.
- [104] R. Merton. The theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4:141–183, 1973.

- [105] B. Mohamed. Simulations of transaction costs and optimal reheding. *Applied Mathematical Finance*, 1:49–62, 1994.
- [106] S. Ortobelli, S. T. Rachev, S. Stoyanov, F. J. Fabozzi, and A. Biglova. The proper use of risk measures in portfolio theory. *International Journal of Theoretical and Applied Finance*, 8(8):1107–1133, 2005.
- [107] S. Pastorello, E. Renault, and N. Touzi. Statistical inference for random-variance option pricing. *Journal of Business & Economic Statistics*, 18(3):358–367, July 2000.
- [108] J. Perelló, R. Sircar, and J. Masoliver. Option pricing under stochastic volatility: the exponential Ornstein-Uhlenbeck model. December 2008. URL <http://arxiv.org/pdf/0804.2589>.
- [109] W. B. Powell. *Approximate Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley Interscience, New Jersey, 2007.
- [110] J.-L. Prigent. *Portfolio Optimization and Performance Analysis*. Chapman & Hall/CRC, Boca Raton, Florida, 2007.
- [111] J. A. Primbs. Dynamic hedging of basket options under proportional transaction costs using receding horizon control. *International Journal of Control*, 82(10):1841–1855, October 2009.
- [112] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000.
- [113] R. T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26(7):1443–1471, 2002.

- [114] R. T. Rockafellar, S. Uryasev, and M. Zabarankin. Generalized deviations in risk analysis. Research report #2004-4, Risk Management and Financial Engineering Lab, University of Florida, 2004.
- [115] T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.
- [116] D. L. Russell. Computing convex spline approximations. *International Journal of Information and Systems Sciences*, 5(1):83–97, 2009.
- [117] J. Rust. Using randomization to break the curse of dimensionality. *Econometrica*, 65(3):487–516, May 1997.
- [118] G. Sandmann and S. J. Koopman. Estimation of stochastic volatility models via Monte Carlo maximum likelihood. *Journal of Econometrics*, 87(2):271–301, september 1998.
- [119] W. Schachermayer. Optimal investment in incomplete markets when wealth may become negative. *Annals of applied probability*, 11(3):694–734, 2001.
- [120] M. Schweizer. Hedging of options in a general semimartingale model. Diss. ETH Zürich 8615, 1988.
- [121] M. Schweizer. Variance-optimal hedging in discrete time. *Mathematics of Operations Research*, 20:1–32, 1995.
- [122] M. Schweizer. A guided tour through quadratic hedging approaches. In E. Jouini, J. Cvitanic, and M. Musiela, editors, *Option Pricing, Interest Rates and Risk Management*, pages 538–574. Cambridge University Press, 2001.
- [123] L. O. Scott. Option pricing when the variance changes randomly: Theory, estimation, and an application. *Journal of Financial and Quantitative Analysis*, 22:419–438, 1987.

- [124] A. Shapiro. Stochastic programming approach to optimization under uncertainty. *Mathematical Programming, Series B*, 112:183–220, 2008.
- [125] E. Sinclair. *Volatility Trading*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2008.
- [126] I. M. Sobol'. The distribution of points in a cube and the approximate evaluation of integrals. *U.S.S.R. Comput. Math. and Math. Phys.*, 7(4):86–112, 1967.
- [127] H. M. Soner, S. E. Shreve, and J. Cvitanić. There is no nontrivial hedging portfolio for option pricing with transaction costs. *The Annals of Applied Probability*, 5(2):327–355, 1995.
- [128] L. Stentoft. Value function approximation or stopping time approximation: A comparison of two recent numerical methods for American option pricing using simulation and regression. 18(1), 2014. *Journal of Computational Finance*.
- [129] R. Szechtman. Control variates techniques for Monte Carlo simulation. In S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, editors, *Proceedings of the 2003 Winter Simulation Conference*, pages 144–149. IEEE Press, 2003.
- [130] J. N. Tsitsiklis and B. Van Roy. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks*, 12(4):694–703, 2001.
- [131] A. Černý. Dynamic programming and mean-variance hedging in discrete time. *Applied Mathematical Finance*, 11(1):1–25, March 2004.
- [132] A. Černý and J. Kallsen. On the structure of general mean-variance hedging strategies. *Annals of Probability*, 35(4):1479–1531, 2007.

- [133] J. H. van Binsbergen and M. W. Brandt. Solving dynamic portfolio choice problems by recursing on optimized portfolio weights or on the value function? *Computational Economics*, 29:355–367, 2007.
- [134] S. P. Verrill. Fmin — java class, 1998. Available at <https://www1.fpl.fs.fed.us/Fmin.java>.
- [135] A. E. Whalley and P. Wilmott. An asymptotic analysis of an optimal hedging model for option pricing with transaction costs. *Mathematical finance*, 7(3): 307–324, July 1997.
- [136] A. E. Whalley and P. Wilmott. Optimal hedging of options with small but arbitrary transaction cost structure. *European Journal of Applied Mathematics*, 10:117–139, 1999.
- [137] L.-Y. Yu, X.-D. Ji, and S.-Y. Wang. Stochastic programming models in financial optimization: A survey. *Advanced Modeling and Optimization*, 5(1), 2003.
- [138] V. I. Zakamouline. Efficient analytic approximation of the optimal hedging strategy for a European call option with transaction costs. *Quantitative Finance*, 6(5):435–445, October 2006.