BI Norwegian Business School - campus Oslo

GRA 19502

Master Thesis

Component of continuous assessment: Thesis Master of Science
Final master thesis – Counts 80% of total grade

Pricing American-Style Options by Monte Carlo Simulation

Navn: Bjarte Joleik, Vilde Rivers Marhaug

Start: 02.03.2018 09.00

Finish: 03.09.2018 12.00

Abstract

We replicate (in some parts) and extend Tompaidis and Yang's (2014) analysis by comparing the performance of Ordinary Least-Squares (OLS) Regression to Tikhonov Regularization and Classification & Regression Trees (CART), and study whether any polynomial among Chebyshev, Hermite, Laguerre, Legendre and Powers perform superiorly when used in the pricing function. We analyze each method's performance by testing five option types (of which two barrier option types are new research in this thesis) in-the-money, at-the-money and out-of-the-money, and by varying the polynomial degree between zero and five. We find no evidence of superiority among the tested polynomials. Like Tompaidis and Yang (2014), we find that OLS regression tend to underperform when the number of simulation paths is small. Despite this issue, we find that OLS regression performs best among the methods tested – which is also observable for one of the tested barrier options.

Content

1. Introduction

One of the most important problems in option pricing theory is valuing and optimally exercising American-style options. Valuing these early-exercise derivatives typically involves solving an optimal stopping problem. For simple vanilla options where only one factor affects the option value, the optimal stopping problem can be solved efficiently using conventional numerical procedures. However, the valuation becomes more complicated when additional factors affect the option value.

1.1 Problem Description

The Black and Scholes formula is commonly known as a closed-form solution for valuing European options. In contrast, no closed-form solution exists for valuing American options. When only one factor affects an American option's value, it is conventionally valued by lattice methods such as binomial and trinomial trees as well as finite difference methods. However, lattice and finite difference methods become difficult to evaluate accurately when multiple stochastic factors affect the option value. When problems with multi-dimensional features are considered, Monte Carlo methods often give better results since the convergence rate of Monte Carlo simulations is independent of the number of stochastic state variables.

The major drawback of Monte Carlo simulation is its difficulty of dealing with the early-exercise feature embedded in American options. The problem of using simulation-based methods to price American options results from the difficulty of applying a forward-based procedure to a problem requiring a backward-based procedure to be solved. Because of the early-exercise feature embedded in American options, we must know the option value at intermediate times between the simulation start and when the option expires. With Monte Carlo, this information is hard to obtain. Therefore, although Monte Carlo can handle multi-factor problems, once we must solve a problem backwards, it becomes difficult to implement. Despite these difficulties, several researchers have provided ways of valuing American options when using Monte Carlo simulations. In 2001, Longstaff and Schwartz proposed the Least Squares Monte Carlo (LSM) method. Their approach involves using a least-squares regression analysis to determine the best-fit relationship between the continuing value and the values of relevant variables at each time an early-exercise decision must be made. The method has achieved much popularity because of its intuitive regression-based approach. In 2014, Tompaidis and Yang have evaluated the LSM algorithm's performance of Ordinary Least-Squares Regression against Quantile Regression, Tikhonov Regularization, Matching Projection Pursuit, a modified version of Matching Projection Pursuit, and Classification and Regression Trees.

1.2 Research Question

This thesis's objective is to replicate (in some parts) and confirm the analysis of Tompaidis and Yang (2014) by comparing the performance of OLS to other numerical methods, and to extend and improve the methodology along different directions by finding other basis functions that can improve the pricing accuracy. The research question for our thesis is: What are strengths and weaknesses of methods proposed by Tompaidis and Yang (2014), how do they differ in terms of efficiency, robustness and precision, and what are improvements to be done.

1.3 Experimental Design

The LSM algorithm's OLS performance will be compared to Tikhonov Regularization and Classification and Regression Trees (CART). As a benchmark, we will use three out of five test-case options (i.e., Call Option with Continuous Dividends, American-Asian Call Option, and Put Option on a Jump-Diffusion Asset) that were introduced by Fu, Laprise, Madan, Su, and Wu (2001) and two other options (i.e., Up-In and Up-Out Barrier Option). We will study whether each option's obtained price converges to its true price at different simulation numbers when the option is in-the-money, at-the-money, and out-of-the-money, and when varying the polynomial degree between zero and five. We will also study whether any of the polynomials among Chebyshev, Hermite, Laguerre, Legendre and Powers perform superiorly when used in the pricing function.

1.4 Thesis Structure

The rest of the thesis is organized as follows. Section 2 contains a literature review on American-style simulation-based option pricing. Section 3 covers theory on the LSM algorithm, OLS and other numerical methods, the five test cases, and the antithetic variates technique. Section 4 includes a presentation of the numerical results. Section 5 concludes the thesis.

2. Literature Review

The modern version of the Markov Chain Monte Carlo method was invented in the late 1940s by Stanislaw Ulam, and Phelim Boyle was among the first to introduce Monte Carlo simulation into finance by it proposing it for the study of European option prices in 1977.

The first approaches to which Monte Carlo simulation can be used to value American-style options were presented in the 1990's. Tilley (1993) was the first who attempted to use Monte Carlo simulation to value American options by using a bundling technique and a backward induction algorithm to determine the early-exercise boundary. With improvements on Tiller's idea, Carriere (1996) presents a backward induction algorithm and applies it to calculate an early-exercise premium. He shows that the estimation of the early-exercise decision rule should be equivalent to the estimation of a series of conditional expectations. In his algorithm, the conditional expectations are estimated using nonparametric least-squares regression of spline functions. Other early work includes Grant, Vora and Weeks (1997) and Broadie and Glasserman (1997) who consider more general path-dependent options such as Asian options. Broadie and Glasserman show how to price Asian options by Monte Carlo, but their method does not focus on an optimal exercise strategy. Instead, they compute a confidence interval and create two biased estimators; an upper (biased high) and a lower (biased low) bound that converges asymptotically and unbiasedly to the true price of an American option.

Carriere's idea was further developed by Tsitsiklis and Van Roy (1999), Tsitsiklis and Van Roy (2001), and Longstaff and Schwartz (2001) who use least-squares regression to approximate the continuation value function by its projection on the linear span of a set of functions. Tsitsiklis and Van Roy (2001) use all the simulated paths to estimate the continuation value. In contrast to Tsitsiklis and Van Roy (2001), Longstaff and Schwartz (2001) only use in-the-money price paths to increase the efficiency of the algorithm. In their method, they apply least-squares regression in which the explanatory variables are certain polynomial functions and estimate the continuation values of several derivative types.

The convergence properties of the LSM algorithm have been studied by Clément, Lamberton and Protter (2002). They demonstrate that the estimated conditional expectation approaches the true conditional expectation as the polynomial degree goes to infinity. Glasserman and Yu (2004) study the convergence rate of the algorithm when the number of basis functions and the number of paths increase simultaneously. They show that in certain cases, to guarantee that the option price converges to its true price, the number of paths must grow exponentially with the number of polynomial basis functions when the underlying state variable follows Brownian motion. If the underlying variable follows geometric Brownian motion, the number of paths must grow faster than exponential to guarantee convergence.

Other studies of the LSM algorithm include the following authors. Moreno and Navas (2003) analyze the robustness of the algorithm with respect to basis function selection. Gamba (2003) extends it to value real options. Rasmussen (2005) and Fouque and Han (2007) attempts to improve the efficiency of LSM by including control variates.

In 2014, Tompaidis and Yang critically evaluate LSM's OLS performance against quantile regression, Tikhonov regularization, Matching Projection Pursuit (MPP), a modified version of MPP, and Classification and Regression Trees (CART. They find that LSM is inclined to over fit in several instances such as when the exercise frequency increases or when using a low number of simulation paths. Additionally, their analysis find that several of the other methods outperform LSM when including European option prices in the polynomial basis functions.

3. Theory

The first part of this section covers theory behind the LSM algorithm by Longstaff and Schwartz (2001). The second part covers OLS, Tikhonov Regularization and CART. The third part covers the test-cases. The fourth part covers the choice of basis functions. The fifth part covers the antithetic variates technique.

3.1 Least-Squares Monte Carlo (LSM) Algorithm

Tompaidis and Yang (2014) suggest using the following notation and approach for the LSM algorithm: $S_t^{(i)}$ is the value of the state variables at time t along path

i; h is the option payoff; V is the option value; $\{t_i\}_{i=0}^{N}$ are the possible exercise times.

3.1.1 Price-Path Simulation

The LSM algorithm starts by simulating M possible price paths a stock may follow during a specified time span t. The price is logarithmic and follows a geometric Brownian motion stochastic process. It can be computed by the following equation.

$$S(t_{j+1}) = S(t_j)\ exp\left[\left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma\varepsilon\sqrt{\Delta t}\right]$$

In the equation, $S(t_j)$ is stock price at time j; μ is the expected return in a risk-neutral world; σ is the volatility; ε is a random number drawn from the standard normal distribution; Δt is the length of time interval. A simulation illustration is shown below.

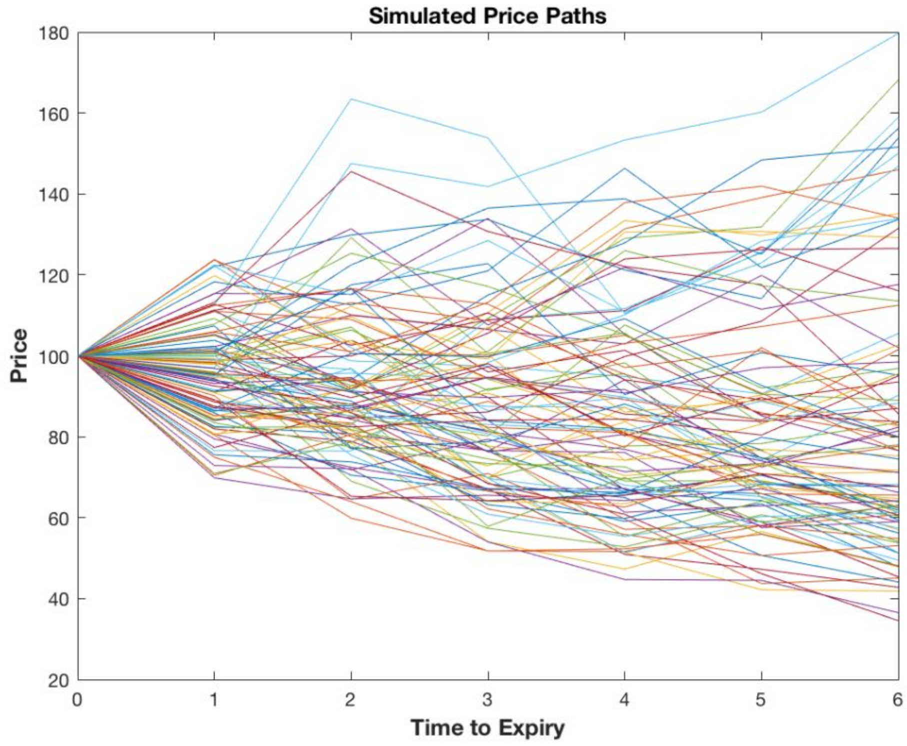

Figure I: Simulated price paths (S=100, K=100, r=0.05, sigma=0.2, δ = 0.1, T=3, N=6, M=100).

3.1.2 Payoff Computation

For all M paths and all exercise time points t, the options cash flow matrix V is computed. At the terminal value t_N, set the option value V equal to the payoff.

$$V\left(S_{t_N}^{(m)}, t_N\right) = h\left(S_{t_N}^{(m)}, t_N\right), m = 1, \dots, M$$

3.1.3 Conditional Expectation Value Computation

The algorithm then proceeds backwards by projecting the expectation of the subsequent discounted cash flows onto the basis functions for the paths where the option is in the money at time t_{j-1}.

For the set of paths $\{i_l\}_{l=1}^{L}$, for which the option is in-the-money; that is, $h\left(S_{t_{N-1}}^{i_l}, t_{N-1}\right) > 0$, find coefficients $a_j^*(t_{N-1})$ to minimize the norm

$$\left\| \sum_{j=1}^{N_b} a(t_{n-1}) \begin{matrix} \varphi_j\left(S_{t_{N-1}}^{(i_1)}\right) \\ \varphi_j\left(S_{t_{N-1}}^{(i_2)}\right) \\ \vdots \\ \varphi_j\left(S_{t_{N-1}}^{(i_L)}\right) \end{matrix} - e^{-r(t_N - t_{N-1})} \begin{pmatrix} V\left(S_{t_N}^{(i_1)}, t_N\right) \\ V\left(S_{t_N}^{(i_2)}, t_N\right) \\ \vdots \\ V\left(S_{t_N}^{(i_L)}, t_N\right) \end{pmatrix} \right\|$$

Here, $\| \cdot \|$ is the vector norm, and $\left(\varphi_{j}{}_{t_{j=1}}^{(N_b)}\right)$ is the basis function. The fitted values are selected as the expected continuation values. Longstaff and Schwartz (2001) estimate the conditional expectation value directly by OLS, while Tompaidis and Yang (2014) use OLS and additional methods that will be outlined in this section.

3.1.4 Optimal Exercise Decision

We compare the vector norm $\| \cdot \|$ with the value for immediate exercise $V\left(S_{t_{j-1}}^{i}\right)$ at each path to decide on the optimal exercise decision.

$$V\left(S_{t_{N-1}}^{(m)}, t_{N-1}\right) = \begin{cases} h\left(S_{t_{N-1}}^{(m)}, t_{N-1}\right) & if\ h\left(S_{t_{N-1}}^{(m)}, t_{N-1}\right) \geq \sum_{j=1}^{N_b} a_j^*(t_{N-1})\, \varphi_j\left(S_{t_{N-1}}^{(m)}\right) \\ e^{-r(t_N - t_{n-1})} V\left(S_{t_N}^{(m)}, t_N\right) & otherwise \end{cases}$$

3.1.5 Backwards Recursion

We recursively use the above algorithm for possible exercise times $t_{N-2}, t_{N-3}, \dots$, until time t_0 which becomes the option price.

3.2 Ordinary Least Squares and Alternative Methods

In this section, we present theories of OLS regression and alternative methods.

3.2.1 Ordinary Least-Squares Regression

OLS regression finds the coefficients $\{y_i\}_{i=1}^{N_b}$ that minimize the sum of squared errors, given observations $\{y_i\}_{i=1}^{L}$ and a set of regressors $\{x_i\}_{i=1}^{N_b}$. The sum of squared errors is given by the equation below.

$$\underset{a}{min}\left[\sum_{i=1}^{L}(y_i-\hat{y}_i)^2\right]=\underset{a}{min}\left[\sum_{i=1}^{L}\left(y_i-\left(\sum_{j=1}^{N_b}a_jx_j\right)_i\right)^2\right]$$

In the above equation, $(\,)_i$ is the i^{th} component of a vector. The regressors $\{x_i\}_{i=1}^{N_b}$ correspond to the basis functions $\{\phi_i\}_{j=1}^{N_b}$. The observed values are the discounted option values from the next possible exercise time; that is, $y_i = e^{-r(t_{j+1}-t_j)}\,V\left(S_{t_{j+1}}^{(i)},t_j\right)$.

3.2.2 Tikhonov Regularization

Tikhonov Regularization is a regularization method developed by Phillips (1962) and Tikhonov (1963) to deal with linear discrete ill-posed problems. The method can be formulated as:

$$\underset{a}{min}\left[\sum_{i=1}^{L}\left(y_i-\left(\sum_{j=1}^{N_b}a_jx_j\right)_i\right)^2+\lambda^2\sum_{i=1}^{L}\left(\sum_{j=1}^{N_b}L_{ij}\left(a_j-\bar{a}_j\right)\right)^2\right]$$

In the above equation, y is the vector of observed values, x is a matrix with columns corresponding to the predictors, a is a coefficient vector of the predictors, λ is a regularization parameter specifying the amount of regularization by determining a tradeoff between the solution size measured by

$\sum_{i=1}^{N_b}\left(\sum_{j=1}^{N_b}L_{ij}\left(a_j-\bar{a}_j\right)\right)^2$, and the solution quality measured by $\sum_{i=1}^{L}\left(y_i-\left(\sum_{J01}^{N_b}a_jx_j\right)_i\right)^2$. The vector $\bar{a}$ is a prior estimate of a, which we set to 0 in this thesis. L is a weight matrix which we set to I, the identity matrix. We use the

Regularization Tools package by Hansen (1994) to perform Tikhonov Regularization.

3.2.3 Classification and Regression Trees (CART)

The Classification and Regression Tree (CART) method was developed by Breiman, Friedman, Olshen, and Stone (1984) and can be used to construct prediction models from data. The prediction models are obtained by recursively partitioning the data space and fitting a simple prediction model within each partition. Thus, the partitioning can be represented graphically as a decision tree. Since a simulation-based American-style option's continuation value is unknown, problem dependent, nonparametric, and nonlinear, CART is a well-suited method as its nature is nonparametric and does not require knowledge about the relationship between predictors and dependent variables.

3.3 Test Cases

To test the different methods, we use three out of five test-case options (i.e. Call Option with Continuous Dividends, American-Asian Call Option, and Put Option on a Jump-Diffusion Asset) that were introduced by Fu, Laprise, Madan, Su, and Wu (2001) and two other options (i.e., Up-In and Up-Out Barrier Option).

Relevant notations include: *T:* expiration date; t_i: possible exercise time for *i* = 0, … N; *K:* strike price; *r:* interest rate; σ: volatility; S_t: stock price at time *t*; S_t^j: stock price at time *t* of stock *j*, where *j* = 1, …, n; h_t: payoff if the option is exercised at time $t \in \{t_i\}_{i=0}^{N}$; $\mathcal{E}$: random sample drawn from a normal distribution with (0,1).

3.3.1 Test Case 1: Call Option with Continuous Dividends

The payoff function is given by $h_t = (S_t - K)^+$. The stock price S_t follows geometric Brownian motion under the risk-neutral measure $dS_t = S_t[(r-\delta)dt + \sigma dZ_t]$. By Itô's lemma, the path-generating process becomes $S_{s_{j+1}}^i = S_{t_j}^i exp\left[\left(r - \delta - \frac{\sigma^2}{2}\right)dt + \sigma\mathcal{E}\sqrt{dt}\right]$.

3.3.2 Test Case 2: American-Asian Call Option

The payoff function given by $h_t = (\overline{S_t} - K)^+$, where $S_t = \frac{1}{n_t+1}\sum_{j=0}^{n_t} S_{t_j}$ and $t_j = t' + (t - t')\frac{j}{n_t}$. $\overline{S_t}$ is the discrete average of the stock prices, where averaging starts at a pre-specified date *t'* up to the exercise time t. We use daily averaging starting on day *t'* and allow early exercise at times $\{t_i\}_{i=0}^{N}$. The underlying stock price process follows geometric Brownian motion with continuous dividends, as formulated in 3.4.1.

3.3.3 Test Case 3: Put Option on a Jump Diffusion Asset

The payoff function is given by $h_t = (K - S_t)^+, t \in \{t_i\}_{i=0}^{N}$. The stock price process follows jump diffusion under the risk-neutral measure $dS_t = S_t[(r - \delta - \lambda k)dt + \sigma dZ_t + dp]$, where dz is a Wiener process and dp is a Poisson process generating jumps (depends on λ and *k*). By Itô's lemma, the path-generating process becomes $S_t = S_0 exp\left[\left(r - \frac{\sigma^2}{2}\right)t + \sigma\sqrt{t}Z_0 + \sum_{j=1}^{N(t)}\left(\delta Z_j - \frac{\delta^2}{2}\right)\right]$, where $Z_j \sim N(0,1)$, $j = 0, \ldots, N(t)$ are independent, identically distributed (i.i.d.) and $N(t) \sim Poisson\ (\lambda t)$.

3.3.4 Test Case 4: Up-in Barrier Option

A barrier option is either knocked in or knocked out when the underlying asset price reaches a certain barrier level *H*, meaning that the option's "existence" depends on whether the underlying crosses a barrier. The underlying asset paths are calculated in the same way as in test case 1 and 2.

Test case 4 is an Up-In Barrier Option where the payoff is calculated from the asset paths that crosses a predetermined barrier level *H*. The option does not exist until the underlying asset price reaches *H*. After the asset price reaches *H*, the option is knocked in and exists until it expires, regardless of whether the underlying drops below the barrier.

3.3.5 Test Case 5: Up-Out Barrier Option

Test case 5 is an Up-Out Barrier Option where the payoff is calculated from the asset paths that are below the barrier level *H*. As soon as the underlying asset price reaches the barrier level *H*, it is knocked out and ceases to exist.

3.4 Choice of Basis Functions

Early papers have used simple power function polynomials as basis functions, for example Tsitsiklis and Van Roy (2001) and Longstaff and Schwartz (2001). Moreno and Navas (2003) found that other types of polynomials; that is, Chebyshev, Hermite, Laguerre, and Legendre, lead to very small variation in the option value when the largest degree of the different polynomials is the same and when using 100,000 price paths.

To compute the polynomials, Moreno and Navas (2003) suggest using the expression $f_n(x) = d_n \sum_{m=0}^{N} c_m g_m(x)$ and recurrence formula: $a_{n+1} f_{n+1}(x) = (a_n + b_n x) f_n x - a_{n-1} f_{n-1}(x)$.

Table I: Polynomial Recurrence Formulas

$T_0(x) = 1$	$T_1(x) = 1$	$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$
$He_0(x) = 1$	$He_1(x) = 1$	$He_{n+1}(x) = xHe_n(x) - nHe_{n-1}(x)$
$L_0(x) = 1$	$L_1(x) = 1 - X$	$L_{n+1}(x) = \frac{1}{n+1}[(2n+1-x)L_n(x) - nL_{n-1}(x)]$
$P_0(x) = 1$	$P_1(x) = 1$	$P_{n+1}(x) = \frac{1}{n+1}[(2n+1)xP_n(x) - nP_{n-1}(x)]$
$W_0(x) = 1$	$W_1(x) = 1$	$W_n(x) = xW_{n-1}$

Notes: *T:* Chebyshev; *He:* Hermite; *L:* Laguerre; *P:* Legendre; *W:* Powers.

3.5 Antithetic Variates

The method of antithetic variates is a variance-reduction technique, which introduces negative dependence between pairs of replications (Glasserman, 2003). The method can be used to obtain more accurate estimates from the Monte Carlo valuation, or to obtain the desired accuracy using less computational work. A simulation trial involves calculating two values; the first is calculated as usual and the second is calculated by adding a negative sign in front of all random numbers drawn from the normal distribution. This way we get complementary values for all original values. Thus, the total range of possible outcomes become better represented and we will be able to achieve lower variance in the estimates.

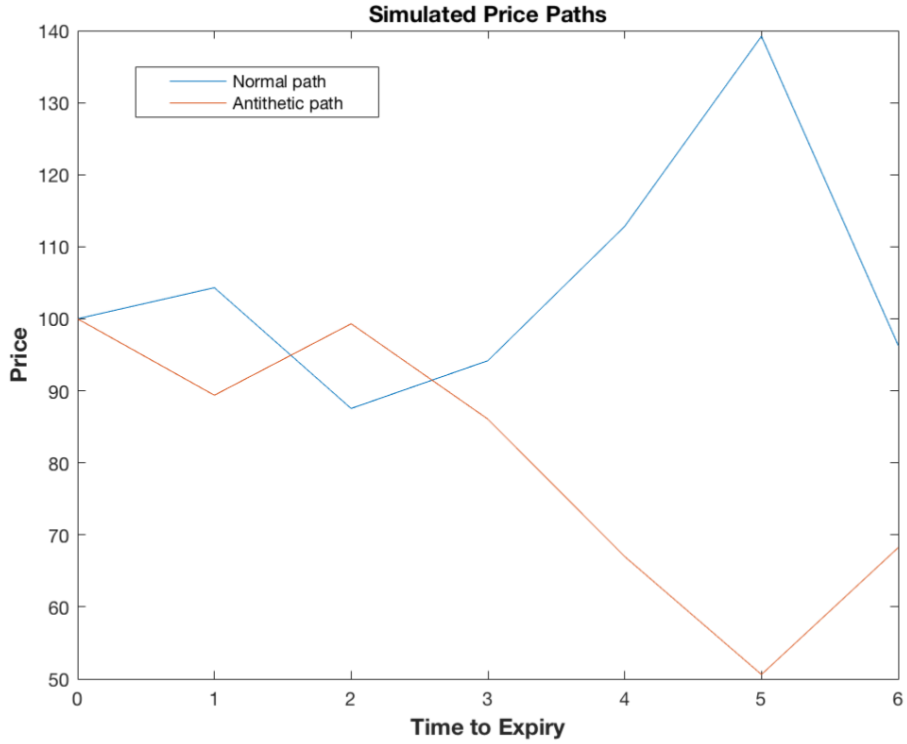

Figure II: Simulation of one normal and one antithetic price path.

4. Numerical Results

This section presents the LSM algorithm's setup. Thereafter follows a test-case analysis and a polynomial analysis of the LSM algorithm's numerical results.

4.1 Setup

We estimate the asymptotic approximation to the option price for each of the test cases using OLS regression and 100,000 simulation paths. To evaluate the estimation methods, we compare values obtained when the option is in-the-money, at-the-money, and out-of-the-money, and for different polynomials types of degrees 0 to 5 and different number of simulation paths. We use samples of 100, 1,000, 5,000, and 10,000 simulation paths. Antithetic paths are implemented to reduce estimator variance. All functions are carried out in MATLAB, where Tikhonov Regularization is from the Regularization Tools package by Hansen (1994). The detailed analysis can be found within 7.1 in the Appendix, where standard errors are computed by running each method 20 times using independent samples. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means more than three standard errors away. In this section, table II, III and IV summarize the overall performance of the different analyzes. By assigning a "+", "o" or "– ", the tables give visual representations of whether the methods perform good, average or poor at different simulation numbers.

4.2 Test-Case Analysis

This section includes the test-case analysis's numerical results when using the Laguerre polynomial of degree 0 to 5. Table II summarizes each test case's overall performance.

Table II: Test-Case Analysis

Option Type	S_0	OLS				Tikhonov				CART			
ContD	90	–	o	+	+	–	+	+	+	+	o	+	+
	100	–	+	+	+	–	+	o	o	+	+	+	+
	110	–	+	+	+	+	+	–	–	+	+	+	+
Asian	90	–	+	+	+	+	+	–	–	+	+	+	+
	100	o	+	+	+	+	–	–	–	+	+	+	+
	110	–	+	+	+	–	–	–	–	+	+	+	+
JumpD	90	–	–	+	+	–	–	–	o	+	+	o	+
	100	–	–	+	+	o	o	+	+	+	+	+	+
	110	–	o	+	+	o	+	+	+	+	+	+	o
UpInBarrier	90	–	o	+	+	–	+	+	+	+	+	+	+
	100	–	+	+	+	–	+	+	o	+	o	+	+
	110	–	+	+	+	+	+	–	–	+	+	+	+
UpOutBarrier	90	+	+	+	+	+	+	+	+	o	+	+	+
	100	+	+	+	+	+	+	+	+	+	+	+	+
	110	+	+	+	+	+	+	+	+	o	+	+	+

Notes: "+", "o", and "–" represent good, neutral, and poor performance. Each method's four columns correspond to the results from the test-case analysis with different numbers of simulation paths (from left to right; 100, 1,000, 5,000, 10,000, and 100,000 paths). Each case is analyzed when the option is out-of-the-money (S_0 = 90), at-the-money (S_0 = 100) and in-the-money (S_0 = 110).

Table II shows that OLS underperforms when the number of simulation paths is small (e.g., M = 100), but performs better as the number of paths increases. Tikhonov Regularization mostly outperforms OLS when the number of paths is small, but the performance seems to vary when the option is out-of-the-money, at-the-money and out-of-the-money. CART seems to be the superior method as it performs well in all cases. However, since CART uses only degree 0 instead of degree 0 to 5, we must look closer in the detailed analysis to understand the whole picture. It appears that the option value is further away from the asymptotic value (see table 7.1.2.2 in the appendix) when estimating using CART rather than OLS, in cases when the option is out-of-the-money, at-the-money and out-of-the-money.

4.3 Polynomial Analysis

This section includes an analysis of five polynomial types and of the polynomial degrees 0 to 5.

4.3.1 Polynomial-Type Analysis

Table III summarizes each option type's overall performance using five polynomial types (i.e., Chebyshev, Hermite, Laguerre, Legendre and Powers).

Table III: Polynomial-Type Analysis

Option Type	**Poly**	**OLS**				**Tikhonov**				**CART**			
ContD	Ch	–	+	+	+	+	+	+	+	+	+	+	+
	He	–	+	+	+	+	+	+	+	o	+	+	+
	La	–	+	+	+	–	+	+	+	+	+	+	+
	Le	–	+	+	+	+	+	+	+	o	+	+	+
	Pw	–	+	+	+	+	+	+	+	+	+	+	–
Asian	Ch	o	+	+	+	+	+	+	+	+	+	+	+
	He	o	+	+	+	+	+	+	+	+	+	+	+
	La	o	+	+	+	+	+	o	+	+	+	+	+
	Le	–	+	+	+	+	+	+	+	+	+	+	+
	Pw	–	+	+	+	+	+	+	o	–	+	+	+
JumpD	Ch	–	–	+	+	+	o	+	+	+	+	+	+
	He	–	–	+	+	+	+	+	+	+	+	+	+
	La	–	–	+	+	+	+	+	+	+	+	+	+
	Le	–	o	+	+	–	+	+	+	+	+	+	+
	Pw	–	o	+	+	+	+	+	+	+	+	+	+
UpInBarrier	Ch	–	+	+	+	o	+	+	+	+	+	+	+
	He	–	+	+	+	o	+	+	+	+	o	+	+
	La	–	+	+	+	–	+	+	+	+	+	+	+
	Le	–	+	+	+	o	+	+	+	o	–	+	+
	Pw	–	+	+	+	o	+	+	+	+	+	+	+
UpOutBarrier	Ch	+	+	+	+	+	+	+	+	+	+	o	+
	He	+	+	+	+	+	+	+	+	+	+	+	+
	La	+	+	+	+	+	+	+	+	+	+	+	+
	Le	+	+	+	+	+	+	+	+	+	+	+	+
	Pw	+	+	+	+	+	+	+	+	o	+	+	+

Notes: "+", "o", and "–" represent good, neutral, and poor performance. Each method's four columns correspond to the results from the polynomial analysis with different numbers of simulation paths (from left to right; 100, 1,000, 5,000, 10,000, and 100,000 paths). All computations use $S_0 = 100$.

Table III shows no pattern in how the different polynomial types perform. Hence, none of the polynomial types seem to perform superiorly.

4.3.2 Polynomial-Degree Analysis

Table IV summarizes the overall performance of the polynomial-degree analysis, using degrees of 0 to 5.

Table IV: Polynomial-Degree Analysis

Option Type	**N_b**	**OLS**				**Tikhonov**				**CART**			
ContD	0	+	+	+	+	+	+	+	+	+	+	+	+
	1	o	+	+	+	o	+	+	+				
	2	–	+	+	+	–	+	+	+				
	3	–	+	+	+	–	+	+	+				
	4	–	o	+	+	–	o	+	+				
	5	–	o	+	+	–	o	+	+				
Asian	0	+	+	+	+	+	+	+	+	+	+	+	+
	1	+	o	o	+	+	o	o	+				
	2	+	+	+	+	+	+	+	+				
	3	–	+	+	+	–	+	+	+				
	4	–	o	+	+	–	o	+	+				
	5	–	+	+	+	–	+	+	+				
JumpD	0	+	+	+	+	+	+	+	+	+	+	+	+
	1	+	o	o	+	+	o	o	+				
	2	–	o	+	+	–	o	+	+				
	3	–	–	+	+	–	–	+	+				
	4	–	–	+	+	–	–	+	+				
	5	–	–	+	+	–	–	+	+				
UpinBarrier	0	+	+	+	+	+	+	+	+	+	+	+	+
	1	–	o	o	+	–	o	o	+				
	2	–	+	+	+	–	+	+	+				
	3	–	o	+	+	–	o	+	+				
	4	–	+	o	+	–	+	o	+				
	5	–	–	+	+	–	–	+	+				
UpOutBarrier	0	+	+	+	+	+	+	+	+	+	+	+	+
	1	+	o	o	+	+	o	o	+				
	2	+	+	+	+	+	+	+	+				
	3	+	+	+	+	+	+	+	+				
	4	+	+	+	+	+	+	+	+				
	5	+	+	+	+	+	+	+	+				

Notes: "+", "o", and "–" represent good, neutral, and poor performance. N_b denotes the highest degree of polynomial basis functions. Each method's four columns correspond to the results from the polynomial analysis with different numbers of simulation paths (from left to right; 100, 1,000, 5,000, 10,000, and 100,000 paths). All computations use S = 100.

Table IV shows that increasing degree of polynomial basis functions causes underperformance in OLS and Tikhonov Regularization, especially when using a low number of simulation paths (e.g., M = 100). The degree of underperformance is most likely caused by worsened overfitting issues as the polynomial degree increases. In contrast, the Up-Out Barrier option seems to be unaffected, most likely because of the how the option is constructed. Because the Up-Out ceases to exist as soon as the underlying asset takes a value higher than the barrier level, the up-out function excludes nearly all extreme values and thereby comes off unaffected to overfitting.

5. Conclusion

We have compared the performance of OLS regression to Tikhonov Regularization and CART when pricing American-style options by Monte Carlo simulations. In addition to three out of five test-case options used by Tompaidis and Yang (2014), we have extended the analysis by implementing the Up-In and the Up-Out Barrier Option. Like Tompaidis and Yang (2014), we find that OLS regression is subject to overfitting and produces inaccurate estimates when the number of simulation paths is small. This result is also observable for the Up-In. In contrast, the Up-Out comes off unaffected to overfitting because of its construction and therefore seems less suitable as a test-case option. Unlike Tompaidis and Yang (2014) – who found MMPP to be the best-performing method – we find OLS to perform best since our research focuses solely on OLS, Tikhonov Regularization and CART. Lastly, we find that none of the tested polynomials perform superiorly. The next future research step would be to extend the algorithm in other ways for additional improvement. An interesting algorithm extension could be to implement control variates in the basis functions as an additional variance-reduction technique.

6. Bibliography

Barraquand, J., & Martineau, D. (1995). Numerical valuation of high dimensional multivariate American securities. *Journal of financial and quantitative analysis, 30*(3), 383-405.

Boyle, P., Broadie, M., & Glasserman, P. (1997). Monte Carlo methods for security pricing. *Journal of economic dynamics and control, 21*(8), 1267-1321.

Boyle, P. P. (1977). Options: A monte carlo approach. *Journal of financial Economics, 4*(3), 323-338.

Boyle, P. P. (1977). Options: A monte carlo approach. *Journal of financial Economics, 4*(3), 323-338.

Breiman, L., Friedman, J., & Olshen, R. (2009). Stone, cj (1984) classification and regression trees. *Wadsworth, Belmont, California*.

Broadie, M., & Glasserman, P. (1997). Pricing American-style securities using simulation. *Journal of economic dynamics and control, 21*(8), 1323-1352.

Broadie, M., & Kaya, O. (2004). *Exact simulation of option greeks under stochastic volatility and jump diffusion models.* Paper presented at the Simulation Conference, 2004. Proceedings of the 2004 Winter.

Brooks, C. (2014). *Introductory econometrics for finance*: Cambridge university press.

Carriere, J. F. (1996). Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: mathematics and Economics, 19*(1), 19-30.

Clément, E., Lamberton, D., & Protter, P. (2002). An analysis of a least squares regression method for American option pricing. *Finance and Stochastics, 6*(4), 449-471.

Cox, J. C., Ross, S. A., & Rubinstein, M. (1979). Option pricing: A simplified approach. *Journal of financial Economics, 7*(3), 229-263.

Fouque, J.-P., & Han, C.-H. (2007). A martingale control variate method for option pricing with stochastic volatility. *ESAIM: Probability and Statistics, 11*, 40-54.

Fu, M. C., Laprise, S. B., Madan, D. B., Su, Y., & Wu, R. (2001). Pricing American options: A comparison of Monte Carlo simulation approaches. *Journal of Computational Finance, 4*(3), 39-88.

Gamba, A. (2003). Real options valuation: A Monte Carlo approach.

Gamba, A. (2003). Real options valuation: A Monte Carlo approach.

Glasserman, P. (2013). *Monte Carlo methods in financial engineering* (Vol. 53): Springer Science & Business Media.

Glasserman, P., & Yu, B. (2004). Number of paths versus number of basis functions in American option pricing. *The Annals of Applied Probability, 14*(4), 2090-2119.

Grant, D., Vora, G., & Weeks, D. (1997). Path-dependent options: Extending the Monte Carlo simulation approach. *Management Science, 43*(11), 1589-1602.

Hansen, P. (2007). Regularisation Tools: A Matlab Package for Analysis and Solution of Discrete Ill-Posed Problems, Version 4.0.

Hoyle, M. (September 1, 2016). Pricing American Options. Retrieved from https://se.mathworks.com/matlabcentral/fileexchange/16476-pricing-american-options?focused=6781443&tab=function&s_tid=gn_loc_drop

Hull, J. C., & Basu, S. (2016). *Options, futures, and other derivatives*: Pearson Education India.

Longstaff, F. A., & Schwartz, E. S. (2001). Valuing American options by simulation: a simple least-squares approach. *The review of financial studies, 14*(1), 113-147.

Metropolis, N., & Ulam, S. (1949). The monte carlo method. *Journal of the American statistical association, 44*(247), 335-341.

Moreno, M., & Navas, J. F. (2003). On the robustness of least-squares Monte Carlo (LSM) for pricing American derivatives. *Review of Derivatives Research, 6*(2), 107-128.

Moreno, M., & Navas, J. F. (2003). On the robustness of least-squares Monte Carlo (LSM) for pricing American derivatives. *Review of Derivatives Research, 6*(2), 107-128.

Phillips, D. L. (1962). A technique for the numerical solution of certain integral equations of the first kind. *Journal of the ACM (JACM), 9*(1), 84-97.

Rasmussen, N. S. (2005). Control variates for Monte Carlo valuation of American options.

Tihonov, A. N. (1963). Solution of incorrectly formulated problems and the regularization method. *Soviet Math., 4*, 1035-1038.

Tilley, J. A. (1999). Valuing American options in a path simulation model.

Tompaidis, S., & Yang, C. (2014). Pricing American-style options by Monte Carlo simulation: alternatives to ordinary least squares.

Tsitsiklis, J. N., & Van Roy, B. (2001). Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks, 12*(4), 694-703.

Ulam, S. M. (1960). *A collection of mathematical problems* (Vol. 8): Interscience Publishers.

7. Appendices

7.1 Tables

7.1.1 Polynomial Analysis

Table 7.1.1.1: Call Option with Continuoud Dividends

M	Pol./N_b	OLS 0	OLS 1	OLS 2	OLS 3	OLS 4	OLS 5	Tikhonov 0	Tikhonov 1	Tikhonov 2	Tikhonov 3	Tikhonov 4	Tikhonov 5	CART 0
100	Ch	8.02 ± 0.14	8.09 ± 0.18	8.53 ± 0.15	8.80 ± 0.23	9.06 ± 0.17	9.01 ± 0.15	7.49 ± 0.12	7.90 ± 0.13	8.25 ± 0.12	7.94 ± 0.16	8.22 ± 0.23	8.27 ± 0.15	7.89 ± 0.13
100	He	7.73 ± 0.17	8.21 ± 0.17	8.54 ± 0.17	9.12 ± 0.19	8.91 ± 0.12	9.51 ± 0.16	7.39 ± 0.14	7.94 ± 0.16	8.14 ± 0.14	8.24 ± 0.19	8.19 ± 0.18	8.49 ± 0.13	8.15 ± 0.15
100	La	7.81 ± 0.13	8.19 ± 0.12	8.59 ± 0.14	9.07 ± 0.15	8.90 ± 0.13	9.54 ± 0.15	7.64 ± 0.14	8.26 ± 0.13	7.98 ± 0.14	8.41 ± 0.17	8.48 ± 0.13	8.24 ± 0.17	7.87 ± 0.14
100	Le	7.83 ± 0.11	8.52 ± 0.14	8.83 ± 0.14	9.08 ± 0.17	9.11 ± 0.12	8.99 ± 0.23	7.55 ± 0.14	7.97 ± 0.15	7.99 ± 0.15	8.57 ± 0.17	8.50 ± 0.13	8.15 ± 0.15	8.12 ± 0.16
100	Pw	7.68 ± 0.16	8.04 ± 0.16	8.49 ± 0.14	9.10 ± 0.23	9.09 ± 0.19	8.90 ± 0.17	7.55 ± 0.12	7.86 ± 0.12	8.04 ± 0.11	8.40 ± 0.11	8.39 ± 0.18	8.14 ± 0.16	7.84 ± 0.13
1 000	Ch	7.65 ± 0.05	7.84 ± 0.04	7.94 ± 0.05	7.96 ± 0.05	7.96 ± 0.04	7.99 ± 0.05	7.40 ± 0.04	7.84 ± 0.07	7.89 ± 0.03	7.86 ± 0.04	7.90 ± 0.05	7.74 ± 0.05	7.80 ± 0.05
1 000	He	7.76 ± 0.04	7.78 ± 0.04	7.84 ± 0.05	7.92 ± 0.05	8.02 ± 0.05	8.05 ± 0.04	7.41 ± 0.04	7.74 ± 0.05	7.87 ± 0.04	7.88 ± 0.05	7.84 ± 0.05	7.85 ± 0.05	7.79 ± 0.04
1 000	La	7.73 ± 0.05	7.82 ± 0.05	7.83 ± 0.04	7.96 ± 0.05	8.01 ± 0.06	8.04 ± 0.05	7.41 ± 0.06	7.81 ± 0.05	7.78 ± 0.04	7.89 ± 0.04	7.86 ± 0.04	7.75 ± 0.04	7.73 ± 0.05
1 000	Le	7.78 ± 0.03	7.89 ± 0.05	7.92 ± 0.04	8.01 ± 0.05	7.97 ± 0.05	7.93 ± 0.04	7.46 ± 0.03	7.74 ± 0.05	7.78 ± 0.06	7.92 ± 0.05	7.87 ± 0.04	7.65 ± 0.05	7.84 ± 0.04
1 000	Pw	7.79 ± 0.06	7.86 ± 0.07	7.89 ± 0.04	7.95 ± 0.05	7.97 ± 0.05	7.92 ± 0.04	7.42 ± 0.04	7.76 ± 0.06	7.87 ± 0.05	7.90 ± 0.06	7.89 ± 0.05	7.78 ± 0.05	7.81 ± 0.05
5 000	Ch	7.79 ± 0.02	7.83 ± 0.01	7.89 ± 0.02	7.86 ± 0.01	7.84 ± 0.02	7.90 ± 0.02	7.42 ± 0.02	7.76 ± 0.02	7.85 ± 0.02	7.89 ± 0.02	7.83 ± 0.03	7.75 ± 0.03	7.80 ± 0.02
5 000	He	7.75 ± 0.02	7.81 ± 0.02	7.89 ± 0.02	7.87 ± 0.03	7.89 ± 0.02	7.83 ± 0.02	7.44 ± 0.02	7.75 ± 0.03	7.84 ± 0.03	7.82 ± 0.02	7.82 ± 0.02	7.75 ± 0.02	7.76 ± 0.02
5 000	La	7.81 ± 0.02	7.78 ± 0.02	7.87 ± 0.02	7.88 ± 0.01	7.89 ± 0.02	7.91 ± 0.02	7.42 ± 0.02	7.76 ± 0.02	7.81 ± 0.02	7.85 ± 0.02	7.87 ± 0.01	7.76 ± 0.02	7.76 ± 0.02
5 000	Le	7.77 ± 0.02	7.80 ± 0.02	7.89 ± 0.02	7.87 ± 0.02	7.89 ± 0.02	7.87 ± 0.02	7.44 ± 0.02	7.77 ± 0.02	7.85 ± 0.03	7.86 ± 0.02	7.86 ± 0.02	7.79 ± 0.02	7.74 ± 0.02
5 000	Pw	7.74 ± 0.03	7.77 ± 0.02	7.88 ± 0.02	7.85 ± 0.02	7.88 ± 0.02	7.86 ± 0.02	7.45 ± 0.02	7.74 ± 0.02	7.88 ± 0.02	7.83 ± 0.02	7.86 ± 0.02	7.79 ± 0.02	7.81 ± 0.03
10 000	Ch	7.76 ± 0.01	7.78 ± 0.01	7.84 ± 0.02	7.85 ± 0.01	7.87 ± 0.01	7.86 ± 0.02	7.42 ± 0.01	7.74 ± 0.01	7.85 ± 0.01	7.84 ± 0.02	7.87 ± 0.01	7.77 ± 0.02	7.75 ± 0.01
10 000	He	7.77 ± 0.02	7.82 ± 0.02	7.87 ± 0.01	7.88 ± 0.02	7.86 ± 0.01	7.87 ± 0.01	7.44 ± 0.01	7.73 ± 0.01	7.83 ± 0.02	7.82 ± 0.02	7.84 ± 0.01	7.76 ± 0.02	7.79 ± 0.01
10 000	La	7.76 ± 0.02	7.81 ± 0.02	7.86 ± 0.01	7.86 ± 0.01	7.86 ± 0.02	7.86 ± 0.01	7.44 ± 0.01	7.77 ± 0.01	7.84 ± 0.01	7.85 ± 0.01	7.87 ± 0.02	7.78 ± 0.01	7.76 ± 0.02
10 000	Le	7.76 ± 0.02	7.85 ± 0.02	7.87 ± 0.01	7.90 ± 0.02	7.87 ± 0.01	7.87 ± 0.01	7.43 ± 0.01	7.78 ± 0.02	7.84 ± 0.01	7.82 ± 0.02	7.86 ± 0.01	7.78 ± 0.02	7.76 ± 0.02
10 000	Pw	7.77 ± 0.01	7.81 ± 0.01	7.86 ± 0.01	7.86 ± 0.01	7.86 ± 0.02	7.87 ± 0.02	7.44 ± 0.01	7.75 ± 0.01	7.85 ± 0.02	7.83 ± 0.02	7.85 ± 0.01	7.75 ± 0.01	7.80 ± 0.01
100 000	Ch	7.77 ± 0.01	7.82 ± 0.00	7.85 ± 0.00	7.86 ± 0.00	7.87 ± 0.00	7.86 ± 0.00	7.44 ± 0.00	7.74 ± 0.00	7.84 ± 0.01	7.85 ± 0.01	7.85 ± 0.01	7.81 ± 0.01	7.77 ± 0.01
100 000	He	7.77 ± 0.01	7.81 ± 0.00	7.85 ± 0.00	7.86 ± 0.00	7.86 ± 0.00	7.86 ± 0.01	7.44 ± 0.00	7.75 ± 0.00	7.84 ± 0.01	7.85 ± 0.01	7.85 ± 0.00	7.80 ± 0.01	7.77 ± 0.00
100 000	La	7.77 ± 0.00	7.82 ± 0.00	7.85 ± 0.01	7.86 ± 0.00	7.87 ± 0.00	7.87 ± 0.01	7.43 ± 0.00	7.75 ± 0.01	7.85 ± 0.01	7.85 ± 0.00	7.85 ± 0.00	7.80 ± 0.01	7.76 ± 0.00
100 000	Le	7.77 ± 0.00	7.82 ± 0.01	7.85 ± 0.00	7.86 ± 0.00	7.86 ± 0.00	7.86 ± 0.00	7.44 ± 0.00	7.75 ± 0.00	7.84 ± 0.00	7.85 ± 0.00	7.85 ± 0.01	7.80 ± 0.01	7.77 ± 0.01
100 000	Pw	7.77 ± 0.01	7.82 ± 0.00	7.86 ± 0.00	7.86 ± 0.01	7.86 ± 0.00	7.87 ± 0.00	7.44 ± 0.00	7.75 ± 0.00	7.84 ± 0.01	7.84 ± 0.00	7.85 ± 0.00	7.81 ± 0.01	7.76 ± 0.00

Notes: The parameters used are S0 = 100, K = 100, $\sigma = 0.2$, r = 0.05, T = 3, $\delta = 0.1$, and exercise points t $\in$ {0, 0.5, 1, 1.5, 2, 2.5, 3}. Nb denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.1.2: American-Asian Call Option

		OLS						Tikhonov						CART
M	Pol./N_b	0	1	2	3	4	5	0	1	2	3	4	5	0
100	Ch	4.76 ± 0.08	5.51 ± 0.07	5.59 ± 0.07	5.68 ± 0.06	5.73 ± 0.04	5.68 ± 0.07	4.13 ± 0.03	4.73 ± 0.04	5.42 ± 0.09	5.52 ± 0.07	5.48 ± 0.07	5.30 ± 0.07	4.75 ± 0.06
100	He	4.69 ± 0.06	5.55 ± 0.06	5.56 ± 0.06	5.80 ± 0.05	5.70 ± 0.06	5.81 ± 0.07	4.19 ± 0.05	4.67 ± 0.06	5.48 ± 0.07	5.44 ± 0.08	5.45 ± 0.06	5.48 ± 0.06	4.70 ± 0.05
100	La	4.74 ± 0.06	5.60 ± 0.08	5.56 ± 0.05	5.67 ± 0.06	5.72 ± 0.07	5.75 ± 0.07	4.24 ± 0.04	4.85 ± 0.05	5.47 ± 0.08	5.46 ± 0.06	5.39 ± 0.08	5.46 ± 0.07	4.69 ± 0.05
100	Le	4.75 ± 0.06	5.61 ± 0.07	5.57 ± 0.06	5.80 ± 0.08	5.76 ± 0.05	5.71 ± 0.07	4.12 ± 0.04	4.78 ± 0.05	5.54 ± 0.06	5.32 ± 0.05	5.53 ± 0.08	5.18 ± 0.08	4.71 ± 0.05
100	Pw	4.67 ± 0.06	5.50 ± 0.07	5.67 ± 0.07	5.79 ± 0.07	5.76 ± 0.06	5.85 ± 0.07	4.19 ± 0.05	4.83 ± 0.09	5.53 ± 0.06	5.47 ± 0.07	5.46 ± 0.08	5.28 ± 0.06	4.86 ± 0.05
1 000	Ch	4.72 ± 0.02	5.48 ± 0.03	5.50 ± 0.02	5.53 ± 0.03	5.53 ± 0.02	5.45 ± 0.02	4.10 ± 0.01	4.72 ± 0.02	5.41 ± 0.03	5.41 ± 0.03	5.38 ± 0.03	5.27 ± 0.02	4.68 ± 0.01
1 000	He	4.67 ± 0.01	5.47 ± 0.03	5.45 ± 0.03	5.50 ± 0.02	5.50 ± 0.02	5.50 ± 0.02	4.13 ± 0.01	4.71 ± 0.01	5.38 ± 0.03	5.39 ± 0.03	5.39 ± 0.02	5.27 ± 0.03	4.69 ± 0.02
1 000	La	4.70 ± 0.02	5.42 ± 0.02	5.46 ± 0.03	5.51 ± 0.03	5.50 ± 0.03	5.51 ± 0.02	4.15 ± 0.01	4.73 ± 0.02	5.39 ± 0.02	5.41 ± 0.03	5.30 ± 0.02	5.24 ± 0.02	4.69 ± 0.02
1 000	Le	4.72 ± 0.02	5.47 ± 0.02	5.53 ± 0.03	5.53 ± 0.03	5.52 ± 0.02	5.47 ± 0.03	4.13 ± 0.01	4.72 ± 0.02	5.46 ± 0.03	5.39 ± 0.03	5.37 ± 0.03	5.20 ± 0.02	4.68 ± 0.02
1 000	Pw	4.66 ± 0.02	5.51 ± 0.03	5.49 ± 0.03	5.48 ± 0.02	5.52 ± 0.02	5.46 ± 0.02	4.12 ± 0.01	4.73 ± 0.02	5.44 ± 0.02	5.37 ± 0.03	5.34 ± 0.03	5.30 ± 0.02	4.66 ± 0.02
5 000	Ch	4.67 ± 0.01	5.47 ± 0.01	5.47 ± 0.01	5.48 ± 0.01	5.46 ± 0.01	5.46 ± 0.01	4.12 ± 0.01	4.73 ± 0.01	5.45 ± 0.01	5.42 ± 0.01	5.43 ± 0.02	5.26 ± 0.01	4.69 ± 0.01
5 000	He	4.69 ± 0.01	5.46 ± 0.01	5.46 ± 0.01	5.48 ± 0.01	5.49 ± 0.01	5.45 ± 0.01	4.13 ± 0.01	4.72 ± 0.01	5.46 ± 0.01	5.41 ± 0.01	5.40 ± 0.02	5.37 ± 0.02	4.68 ± 0.01
5 000	La	4.70 ± 0.01	5.46 ± 0.01	5.46 ± 0.01	5.46 ± 0.01	5.47 ± 0.01	5.47 ± 0.01	4.13 ± 0.01	4.71 ± 0.01	5.45 ± 0.01	5.47 ± 0.02	5.40 ± 0.01	5.32 ± 0.02	4.70 ± 0.01
5 000	Le	4.68 ± 0.01	5.46 ± 0.01	5.46 ± 0.01	5.47 ± 0.01	5.48 ± 0.01	5.49 ± 0.01	4.13 ± 0.01	4.73 ± 0.01	5.46 ± 0.01	5.44 ± 0.01	5.45 ± 0.01	5.30 ± 0.02	4.69 ± 0.01
5 000	Pw	4.68 ± 0.01	5.47 ± 0.01	5.45 ± 0.01	5.47 ± 0.01	5.48 ± 0.01	5.48 ± 0.01	4.13 ± 0.01	4.73 ± 0.01	5.46 ± 0.01	5.40 ± 0.01	5.45 ± 0.02	5.30 ± 0.01	4.68 ± 0.01
10 000	Ch	4.69 ± 0.01	5.47 ± 0.01	5.46 ± 0.01	5.47 ± 0.01	5.47 ± 0.01	5.46 ± 0.01	4.12 ± 0.00	4.71 ± 0.00	5.46 ± 0.01	5.39 ± 0.01	5.44 ± 0.01	5.32 ± 0.01	4.69 ± 0.00
10 000	He	4.69 ± 0.01	5.46 ± 0.01	5.47 ± 0.01	5.47 ± 0.01	5.46 ± 0.01	5.47 ± 0.01	4.13 ± 0.00	4.70 ± 0.01	5.44 ± 0.01	5.41 ± 0.01	5.45 ± 0.01	5.32 ± 0.01	4.70 ± 0.01
10 000	La	4.69 ± 0.00	5.46 ± 0.01	5.46 ± 0.01	5.46 ± 0.01	5.45 ± 0.01	5.47 ± 0.01	4.12 ± 0.00	4.71 ± 0.01	5.44 ± 0.01	5.43 ± 0.01	5.44 ± 0.01	5.33 ± 0.01	4.69 ± 0.01
10 000	Le	4.68 ± 0.01	5.46 ± 0.01	5.46 ± 0.01	5.47 ± 0.01	5.47 ± 0.01	5.48 ± 0.01	4.13 ± 0.00	4.71 ± 0.00	5.45 ± 0.01	5.43 ± 0.01	5.47 ± 0.01	5.34 ± 0.01	4.69 ± 0.01
10 000	Pw	4.69 ± 0.00	5.46 ± 0.01	5.47 ± 0.01	5.47 ± 0.01	5.48 ± 0.01	5.46 ± 0.01	4.13 ± 0.00	4.72 ± 0.01	5.46 ± 0.01	5.43 ± 0.01	5.43 ± 0.01	5.34 ± 0.01	4.69 ± 0.00
100 000	Ch	4.69 ± 0.00	5.45 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	4.12 ± 0.00	4.71 ± 0.00	5.45 ± 0.01	5.40 ± 0.00	5.46 ± 0.00	5.41 ± 0.01	4.69 ± 0.00
100 000	He	4.69 ± 0.00	5.45 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	5.46 ± 0.00	5.47 ± 0.00	4.13 ± 0.00	4.71 ± 0.00	5.45 ± 0.01	5.40 ± 0.00	5.46 ± 0.00	5.41 ± 0.01	4.69 ± 0.00
100 000	La	4.69 ± 0.00	5.46 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	4.12 ± 0.00	4.72 ± 0.00	5.44 ± 0.01	5.40 ± 0.00	5.47 ± 0.00	5.41 ± 0.01	4.69 ± 0.00
100 000	Le	4.69 ± 0.00	5.46 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	4.13 ± 0.00	4.71 ± 0.00	5.44 ± 0.01	5.39 ± 0.00	5.46 ± 0.00	5.44 ± 0.01	4.69 ± 0.00
100 000	Pw	4.69 ± 0.00	5.46 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	5.47 ± 0.00	4.12 ± 0.00	4.71 ± 0.00	5.45 ± 0.01	5.39 ± 0.00	5.46 ± 0.00	5.41 ± 0.01	4.69 ± 0.00

Notes: The parameters used are S0 = 100, K = 100, $\sigma = 0.2$, r = 0.09, T = 120/365, t' = 91/365, $\delta = 0$, and exercise points $\in \{0, 105/365, 108/365, 111/365, 114/365, 117/365, 120/365\}$. Nb denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.1.3: Put Option on a Jump-Diffusion Asset

		OLS						Tikhonov						CART
M	Pol./N_b	0	1	2	3	4	5	0	1	2	3	4	5	0
100	Ch	8.53 ± 0.08	8.85 ± 0.09	9.21 ± 0.10	9.26 ± 0.09	9.41 ± 0.11	9.42 ± 0.11	7.77 ± 0.08	8.58 ± 0.09	8.75 ± 0.10	9.03 ± 0.09	9.01 ± 0.13	9.09 ± 0.13	8.53 ± 0.11
100	He	8.60 ± 0.11	8.94 ± 0.11	9.04 ± 0.09	9.09 ± 0.08	9.45 ± 0.14	9.23 ± 0.10	7.90 ± 0.09	8.42 ± 0.10	8.85 ± 0.11	9.05 ± 0.13	9.02 ± 0.12	9.07 ± 0.10	8.51 ± 0.10
100	La	8.59 ± 0.09	9.01 ± 0.12	9.01 ± 0.08	9.48 ± 0.08	9.19 ± 0.10	9.46 ± 0.12	7.72 ± 0.12	8.48 ± 0.11	8.76 ± 0.13	8.92 ± 0.11	8.96 ± 0.10	9.08 ± 0.08	8.65 ± 0.10
100	Le	8.55 ± 0.10	9.13 ± 0.12	9.20 ± 0.11	9.25 ± 0.12	9.49 ± 0.11	9.25 ± 0.11	7.86 ± 0.11	8.68 ± 0.10	8.73 ± 0.08	9.08 ± 0.10	9.30 ± 0.12	9.17 ± 0.10	8.50 ± 0.10
100	Pw	8.52 ± 0.10	8.85 ± 0.12	9.20 ± 0.09	9.31 ± 0.10	9.38 ± 0.11	9.38 ± 0.12	7.70 ± 0.10	8.61 ± 0.08	9.05 ± 0.12	8.89 ± 0.08	8.90 ± 0.10	9.29 ± 0.13	8.70 ± 0.11
1 000	Ch	8.57 ± 0.04	8.78 ± 0.03	8.80 ± 0.03	8.83 ± 0.04	8.85 ± 0.04	8.88 ± 0.03	7.74 ± 0.03	8.43 ± 0.03	8.75 ± 0.03	8.86 ± 0.03	8.80 ± 0.04	8.80 ± 0.04	8.52 ± 0.03
1 000	He	8.57 ± 0.04	8.74 ± 0.04	8.80 ± 0.03	8.85 ± 0.04	8.85 ± 0.03	8.83 ± 0.03	7.78 ± 0.02	8.48 ± 0.03	8.61 ± 0.03	8.79 ± 0.04	8.80 ± 0.03	8.83 ± 0.03	8.55 ± 0.04
1 000	La	8.53 ± 0.03	8.75 ± 0.03	8.81 ± 0.02	8.83 ± 0.04	8.84 ± 0.03	8.88 ± 0.04	7.73 ± 0.03	8.42 ± 0.04	8.69 ± 0.03	8.71 ± 0.03	8.78 ± 0.03	8.79 ± 0.04	8.53 ± 0.03
1 000	Le	8.58 ± 0.03	8.74 ± 0.03	8.84 ± 0.04	8.83 ± 0.03	8.78 ± 0.04	8.83 ± 0.03	7.77 ± 0.03	8.55 ± 0.03	8.70 ± 0.03	8.77 ± 0.03	8.81 ± 0.03	8.80 ± 0.03	8.57 ± 0.03
1 000	Pw	8.58 ± 0.03	8.76 ± 0.03	8.81 ± 0.04	8.81 ± 0.03	8.81 ± 0.04	8.82 ± 0.03	7.78 ± 0.03	8.55 ± 0.03	8.71 ± 0.03	8.76 ± 0.04	8.77 ± 0.03	8.82 ± 0.03	8.52 ± 0.03
5 000	Ch	8.54 ± 0.01	8.72 ± 0.01	8.69 ± 0.01	8.74 ± 0.02	8.74 ± 0.02	8.73 ± 0.02	7.78 ± 0.01	8.46 ± 0.02	8.67 ± 0.02	8.69 ± 0.02	8.68 ± 0.01	8.72 ± 0.02	8.55 ± 0.01
5 000	He	8.53 ± 0.01	8.70 ± 0.02	8.71 ± 0.01	8.74 ± 0.02	8.71 ± 0.01	8.73 ± 0.02	7.77 ± 0.01	8.48 ± 0.01	8.66 ± 0.01	8.69 ± 0.02	8.73 ± 0.01	8.69 ± 0.02	8.55 ± 0.01
5 000	La	8.56 ± 0.01	8.71 ± 0.02	8.74 ± 0.01	8.76 ± 0.01	8.75 ± 0.02	8.73 ± 0.01	7.79 ± 0.01	8.48 ± 0.02	8.68 ± 0.02	8.71 ± 0.02	8.73 ± 0.02	8.73 ± 0.01	8.55 ± 0.01
5 000	Le	8.55 ± 0.01	8.69 ± 0.01	8.71 ± 0.02	8.71 ± 0.02	8.74 ± 0.01	8.73 ± 0.02	7.78 ± 0.01	8.44 ± 0.02	8.71 ± 0.01	8.74 ± 0.01	8.76 ± 0.02	8.72 ± 0.01	8.52 ± 0.02
5 000	Pw	8.56 ± 0.02	8.71 ± 0.01	8.77 ± 0.02	8.75 ± 0.02	8.72 ± 0.01	8.76 ± 0.02	7.77 ± 0.01	8.46 ± 0.01	8.66 ± 0.02	8.72 ± 0.02	8.72 ± 0.02	8.72 ± 0.01	8.53 ± 0.01
10 000	Ch	8.54 ± 0.01	8.72 ± 0.01	8.72 ± 0.01	8.72 ± 0.01	8.71 ± 0.01	8.70 ± 0.01	7.76 ± 0.01	8.48 ± 0.01	8.66 ± 0.01	8.73 ± 0.01	8.69 ± 0.01	8.72 ± 0.01	8.54 ± 0.01
10 000	He	8.53 ± 0.01	8.70 ± 0.01	8.70 ± 0.01	8.72 ± 0.01	8.73 ± 0.01	8.75 ± 0.01	7.77 ± 0.01	8.49 ± 0.01	8.69 ± 0.02	8.71 ± 0.01	8.70 ± 0.01	8.71 ± 0.01	8.54 ± 0.01
10 000	La	8.54 ± 0.01	8.69 ± 0.01	8.71 ± 0.01	8.74 ± 0.01	8.71 ± 0.01	8.71 ± 0.01	7.78 ± 0.01	8.46 ± 0.01	8.68 ± 0.01	8.72 ± 0.01	8.72 ± 0.01	8.72 ± 0.01	8.53 ± 0.01
10 000	Le	8.52 ± 0.01	8.70 ± 0.01	8.72 ± 0.01	8.73 ± 0.01	8.74 ± 0.01	8.73 ± 0.01	7.78 ± 0.01	8.46 ± 0.01	8.70 ± 0.01	8.73 ± 0.01	8.71 ± 0.01	8.71 ± 0.01	8.54 ± 0.01
10 000	Pw	8.55 ± 0.01	8.68 ± 0.01	8.72 ± 0.01	8.73 ± 0.01	8.72 ± 0.01	8.72 ± 0.01	7.76 ± 0.01	8.46 ± 0.01	8.67 ± 0.01	8.71 ± 0.01	8.72 ± 0.01	8.73 ± 0.01	8.55 ± 0.01
100 000	Ch	8.54 ± 0.00	8.69 ± 0.00	8.71 ± 0.00	8.71 ± 0.00	8.72 ± 0.00	8.70 ± 0.00	7.77 ± 0.00	8.47 ± 0.00	8.68 ± 0.00	8.70 ± 0.00	8.69 ± 0.01	8.71 ± 0.00	8.54 ± 0.00
100 000	He	8.54 ± 0.00	8.70 ± 0.00	8.71 ± 0.00	8.71 ± 0.00	8.72 ± 0.00	8.70 ± 0.00	7.76 ± 0.00	8.47 ± 0.00	8.68 ± 0.00	8.71 ± 0.00	8.69 ± 0.01	8.70 ± 0.00	8.54 ± 0.00
100 000	La	8.54 ± 0.00	8.70 ± 0.00	8.71 ± 0.00	8.72 ± 0.00	8.72 ± 0.00	8.71 ± 0.00	7.76 ± 0.00	8.47 ± 0.00	8.68 ± 0.00	8.71 ± 0.00	8.71 ± 0.00	8.70 ± 0.00	8.54 ± 0.00
100 000	Le	8.55 ± 0.00	8.70 ± 0.00	8.71 ± 0.00	8.71 ± 0.00	8.71 ± 0.00	8.70 ± 0.00	7.77 ± 0.00	8.47 ± 0.00	8.68 ± 0.00	8.71 ± 0.00	8.69 ± 0.01	8.70 ± 0.00	8.54 ± 0.00
100 000	Pw	8.55 ± 0.00	8.69 ± 0.00	8.71 ± 0.00	8.71 ± 0.00	8.71 ± 0.00	8.71 ± 0.00	7.76 ± 0.00	8.47 ± 0.00	8.69 ± 0.00	8.71 ± 0.00	8.71 ± 0.01	8.71 ± 0.00	8.54 ± 0.00

Notes: The parameters used are S0 = 100, K = 100, $\sigma = \sqrt{0.08}$ r = 0.1, T = 0.5, $\delta = 0.2$, $\lambda = 2$, and exercise points $t \in \{0, 0.125, 0.25, 0.375, 0.5\}$. Nb denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.1.4: Up-In Barrier Option

		OLS						Tikhonov						CART
M	Pol./N_b	0	1	2	3	4	5	0	1	2	3	4	5	0
100	Ch	7.85 ± 0.16	7.61 ± 0.15	8.17 ± 0.19	8.60 ± 0.17	9.06 ± 0.17	8.92 ± 0.13	7.47 ± 0.17	7.65 ± 0.15	7.76 ± 0.15	8.20 ± 0.22	8.13 ± 0.14	7.71 ± 0.12	7.46 ± 0.14
100	He	7.52 ± 0.19	7.94 ± 0.14	8.26 ± 0.13	8.48 ± 0.20	8.82 ± 0.20	9.13 ± 0.17	7.32 ± 0.10	7.40 ± 0.18	7.96 ± 0.19	7.76 ± 0.17	8.02 ± 0.17	7.98 ± 0.18	7.29 ± 0.16
100	La	7.49 ± 0.21	8.12 ± 0.14	8.02 ± 0.13	8.70 ± 0.21	8.75 ± 0.17	8.80 ± 0.17	7.28 ± 0.15	7.60 ± 0.17	8.12 ± 0.13	7.79 ± 0.13	7.91 ± 0.12	8.26 ± 0.20	7.58 ± 0.15
100	Le	7.43 ± 0.22	7.78 ± 0.13	8.49 ± 0.19	8.75 ± 0.17	8.73 ± 0.15	8.49 ± 0.16	7.37 ± 0.09	7.23 ± 0.12	8.08 ± 0.21	8.02 ± 0.14	7.84 ± 0.17	7.66 ± 0.13	7.63 ± 0.17
100	Pw	7.57 ± 0.12	7.87 ± 0.12	8.37 ± 0.21	8.58 ± 0.17	8.71 ± 0.17	8.89 ± 0.20	7.31 ± 0.14	7.72 ± 0.16	7.89 ± 0.18	7.85 ± 0.15	8.09 ± 0.20	8.05 ± 0.19	7.47 ± 0.19
1 000	Ch	7.44 ± 0.04	7.34 ± 0.03	7.48 ± 0.05	7.54 ± 0.05	7.50 ± 0.04	7.52 ± 0.05	7.25 ± 0.04	7.22 ± 0.03	7.41 ± 0.04	7.45 ± 0.05	7.44 ± 0.05	7.33 ± 0.05	7.35 ± 0.07
1 000	He	7.39 ± 0.06	7.48 ± 0.03	7.42 ± 0.05	7.59 ± 0.06	7.51 ± 0.05	7.59 ± 0.05	7.28 ± 0.05	7.31 ± 0.05	7.38 ± 0.05	7.42 ± 0.05	7.56 ± 0.04	7.33 ± 0.05	7.44 ± 0.06
1 000	La	7.37 ± 0.04	7.44 ± 0.06	7.51 ± 0.04	7.54 ± 0.04	7.59 ± 0.03	7.57 ± 0.05	7.24 ± 0.04	7.27 ± 0.06	7.41 ± 0.05	7.50 ± 0.04	7.49 ± 0.04	7.38 ± 0.03	7.25 ± 0.05
1 000	Le	7.32 ± 0.04	7.42 ± 0.05	7.53 ± 0.04	7.50 ± 0.04	7.57 ± 0.04	7.52 ± 0.04	7.26 ± 0.04	7.27 ± 0.05	7.41 ± 0.04	7.37 ± 0.06	7.49 ± 0.04	7.34 ± 0.04	7.40 ± 0.03
1 000	Pw	7.29 ± 0.05	7.47 ± 0.04	7.50 ± 0.05	7.51 ± 0.04	7.51 ± 0.06	7.65 ± 0.06	7.27 ± 0.04	7.26 ± 0.04	7.51 ± 0.05	7.52 ± 0.05	7.52 ± 0.05	7.32 ± 0.05	7.40 ± 0.06
5 000	Ch	7.28 ± 0.02	7.43 ± 0.02	7.44 ± 0.03	7.40 ± 0.02	7.43 ± 0.03	7.40 ± 0.02	7.23 ± 0.01	7.24 ± 0.02	7.39 ± 0.03	7.38 ± 0.02	7.46 ± 0.02	7.32 ± 0.02	7.28 ± 0.02
5 000	He	7.29 ± 0.03	7.42 ± 0.02	7.42 ± 0.02	7.44 ± 0.02	7.43 ± 0.02	7.50 ± 0.02	7.24 ± 0.02	7.28 ± 0.02	7.40 ± 0.02	7.42 ± 0.03	7.41 ± 0.02	7.33 ± 0.02	7.28 ± 0.02
5 000	La	7.29 ± 0.03	7.40 ± 0.02	7.43 ± 0.02	7.43 ± 0.02	7.47 ± 0.02	7.42 ± 0.02	7.27 ± 0.02	7.26 ± 0.02	7.38 ± 0.03	7.45 ± 0.02	7.43 ± 0.02	7.32 ± 0.02	7.30 ± 0.02
5 000	Le	7.27 ± 0.02	7.41 ± 0.02	7.43 ± 0.02	7.44 ± 0.02	7.48 ± 0.02	7.46 ± 0.03	7.23 ± 0.02	7.28 ± 0.02	7.39 ± 0.02	7.36 ± 0.02	7.44 ± 0.02	7.35 ± 0.02	7.30 ± 0.02
5 000	Pw	7.31 ± 0.03	7.39 ± 0.02	7.46 ± 0.02	7.44 ± 0.02	7.49 ± 0.02	7.44 ± 0.02	7.23 ± 0.02	7.28 ± 0.02	7.39 ± 0.02	7.38 ± 0.02	7.43 ± 0.02	7.30 ± 0.01	7.28 ± 0.02
10 000	Ch	7.27 ± 0.01	7.40 ± 0.02	7.42 ± 0.02	7.45 ± 0.01	7.42 ± 0.01	7.44 ± 0.02	7.24 ± 0.01	7.27 ± 0.02	7.38 ± 0.01	7.38 ± 0.02	7.43 ± 0.01	7.35 ± 0.01	7.33 ± 0.02
10 000	He	7.31 ± 0.02	7.42 ± 0.01	7.45 ± 0.01	7.41 ± 0.01	7.46 ± 0.01	7.40 ± 0.01	7.22 ± 0.01	7.22 ± 0.02	7.39 ± 0.02	7.39 ± 0.02	7.44 ± 0.01	7.30 ± 0.02	7.29 ± 0.02
10 000	La	7.29 ± 0.02	7.39 ± 0.01	7.45 ± 0.01	7.45 ± 0.01	7.44 ± 0.02	7.44 ± 0.01	7.26 ± 0.01	7.24 ± 0.02	7.38 ± 0.02	7.39 ± 0.02	7.40 ± 0.01	7.33 ± 0.01	7.30 ± 0.01
10 000	Le	7.28 ± 0.02	7.39 ± 0.01	7.41 ± 0.02	7.44 ± 0.02	7.41 ± 0.01	7.45 ± 0.02	7.24 ± 0.01	7.23 ± 0.01	7.39 ± 0.01	7.40 ± 0.02	7.43 ± 0.02	7.33 ± 0.02	7.29 ± 0.01
10 000	Pw	7.29 ± 0.02	7.42 ± 0.02	7.43 ± 0.02	7.41 ± 0.02	7.42 ± 0.02	7.45 ± 0.02	7.24 ± 0.01	7.26 ± 0.02	7.41 ± 0.02	7.39 ± 0.02	7.42 ± 0.01	7.35 ± 0.01	7.29 ± 0.02
100 000	Ch	7.28 ± 0.01	7.40 ± 0.00	7.43 ± 0.00	7.42 ± 0.01	7.42 ± 0.00	7.42 ± 0.01	7.23 ± 0.00	7.26 ± 0.01	7.37 ± 0.01	7.39 ± 0.00	7.42 ± 0.01	7.35 ± 0.01	7.30 ± 0.00
100 000	He	7.29 ± 0.01	7.41 ± 0.00	7.42 ± 0.00	7.43 ± 0.00	7.43 ± 0.00	7.44 ± 0.01	7.22 ± 0.00	7.25 ± 0.01	7.38 ± 0.01	7.40 ± 0.01	7.42 ± 0.00	7.36 ± 0.01	7.30 ± 0.00
100 000	La	7.29 ± 0.00	7.40 ± 0.01	7.42 ± 0.00	7.43 ± 0.01	7.43 ± 0.00	7.42 ± 0.00	7.23 ± 0.00	7.27 ± 0.01	7.38 ± 0.01	7.40 ± 0.00	7.41 ± 0.00	7.37 ± 0.01	7.28 ± 0.00
100 000	Le	7.29 ± 0.01	7.40 ± 0.01	7.42 ± 0.00	7.42 ± 0.01	7.44 ± 0.00	7.43 ± 0.00	7.24 ± 0.00	7.26 ± 0.01	7.38 ± 0.01	7.40 ± 0.00	7.42 ± 0.00	7.35 ± 0.01	7.29 ± 0.01
100 000	Pw	7.29 ± 0.00	7.40 ± 0.00	7.42 ± 0.00	7.43 ± 0.01	7.43 ± 0.01	7.42 ± 0.01	7.24 ± 0.00	7.25 ± 0.00	7.38 ± 0.00	7.39 ± 0.00	7.42 ± 0.00	7.34 ± 0.01	7.29 ± 0.00

Notes: The parameters used are S0 = 100 K = 100, H = 110, $\sigma = 0.2$, r = 0.05, T = 3, $\delta = 0.1$, and exercise points $t \in \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$. Nb denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.1.5: Up-Out Barrier Option

		OLS						Tikhonov						CART
M	Pol./N_b	0	1	2	3	4	5	0	1	2	3	4	5	0
100	Ch	1.70 ± 0.03	1.64 ± 0.05	1.69 ± 0.04	1.65 ± 0.04	1.70 ± 0.03	1.61 ± 0.03	1.72 ± 0.04	1.64 ± 0.04	1.60 ± 0.04	1.60 ± 0.04	1.61 ± 0.03	1.63 ± 0.05	1.61 ± 0.04
100	He	1.63 ± 0.05	1.68 ± 0.05	1.60 ± 0.03	1.68 ± 0.04	1.67 ± 0.04	1.66 ± 0.03	1.57 ± 0.05	1.59 ± 0.05	1.61 ± 0.04	1.63 ± 0.04	1.63 ± 0.03	1.64 ± 0.04	1.63 ± 0.03
100	La	1.61 ± 0.03	1.64 ± 0.04	1.70 ± 0.03	1.70 ± 0.04	1.63 ± 0.04	1.61 ± 0.03	1.54 ± 0.04	1.64 ± 0.04	1.64 ± 0.04	1.59 ± 0.03	1.64 ± 0.06	1.65 ± 0.05	1.67 ± 0.04
100	Le	1.74 ± 0.04	1.70 ± 0.06	1.67 ± 0.03	1.65 ± 0.03	1.65 ± 0.03	1.71 ± 0.04	1.60 ± 0.04	1.61 ± 0.04	1.61 ± 0.04	1.71 ± 0.04	1.60 ± 0.04	1.69 ± 0.04	1.66 ± 0.03
100	Pw	1.58 ± 0.03	1.60 ± 0.03	1.68 ± 0.03	1.65 ± 0.04	1.66 ± 0.04	1.68 ± 0.03	1.63 ± 0.04	1.59 ± 0.04	1.68 ± 0.05	1.67 ± 0.04	1.61 ± 0.05	1.68 ± 0.04	1.71 ± 0.03
1 000	Ch	1.59 ± 0.01	1.61 ± 0.01	1.63 ± 0.01	1.62 ± 0.01	1.61 ± 0.01	1.60 ± 0.01	1.59 ± 0.01	1.60 ± 0.02	1.62 ± 0.01	1.64 ± 0.01	1.62 ± 0.01	1.64 ± 0.01	1.62 ± 0.01
1 000	He	1.62 ± 0.01	1.62 ± 0.01	1.61 ± 0.01	1.64 ± 0.01	1.63 ± 0.02	1.60 ± 0.01	1.60 ± 0.02	1.61 ± 0.01	1.64 ± 0.01	1.63 ± 0.01	1.61 ± 0.01	1.62 ± 0.01	1.61 ± 0.02
1 000	La	1.61 ± 0.01	1.62 ± 0.01	1.63 ± 0.01	1.62 ± 0.01	1.65 ± 0.01	1.62 ± 0.01	1.60 ± 0.01	1.64 ± 0.01	1.63 ± 0.02	1.62 ± 0.01	1.62 ± 0.01	1.62 ± 0.01	1.62 ± 0.01
1 000	Le	1.61 ± 0.02	1.62 ± 0.01	1.61 ± 0.01	1.63 ± 0.01	1.62 ± 0.01	1.63 ± 0.01	1.60 ± 0.01	1.65 ± 0.01	1.63 ± 0.01	1.63 ± 0.01	1.63 ± 0.01	1.61 ± 0.01	1.63 ± 0.01
1 000	Pw	1.61 ± 0.01	1.61 ± 0.01	1.63 ± 0.01	1.62 ± 0.01	1.62 ± 0.01	1.64 ± 0.01	1.59 ± 0.01	1.63 ± 0.01	1.62 ± 0.01	1.60 ± 0.01	1.63 ± 0.01	1.61 ± 0.01	1.61 ± 0.01
5 000	Ch	1.63 ± 0.01	1.62 ± 0.01	1.62 ± 0.00	1.62 ± 0.01	1.61 ± 0.01	1.63 ± 0.01	1.61 ± 0.01	1.62 ± 0.00	1.63 ± 0.01	1.63 ± 0.00	1.62 ± 0.01	1.62 ± 0.01	1.64 ± 0.01
5 000	He	1.62 ± 0.00	1.62 ± 0.01	1.61 ± 0.01	1.62 ± 0.00	1.62 ± 0.01	1.62 ± 0.01	1.61 ± 0.01	1.61 ± 0.00	1.61 ± 0.01	1.62 ± 0.01	1.62 ± 0.01	1.62 ± 0.01	1.62 ± 0.01
5 000	La	1.62 ± 0.00	1.62 ± 0.01	1.62 ± 0.00	1.61 ± 0.01	1.61 ± 0.01	1.62 ± 0.00	1.62 ± 0.01	1.62 ± 0.01	1.63 ± 0.01	1.62 ± 0.01	1.63 ± 0.01	1.62 ± 0.01	1.62 ± 0.01
5 000	Le	1.62 ± 0.00	1.62 ± 0.01	1.63 ± 0.01	1.62 ± 0.01	1.63 ± 0.01	1.63 ± 0.01	1.61 ± 0.00	1.61 ± 0.01	1.62 ± 0.01	1.62 ± 0.01	1.63 ± 0.01	1.63 ± 0.01	1.62 ± 0.01
5 000	Pw	1.62 ± 0.01	1.62 ± 0.01	1.62 ± 0.01	1.63 ± 0.01	1.61 ± 0.00	1.62 ± 0.01	1.61 ± 0.01	1.62 ± 0.01	1.63 ± 0.01	1.62 ± 0.01	1.62 ± 0.01	1.62 ± 0.01	1.61 ± 0.01
10 000	Ch	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.63 ± 0.01	1.61 ± 0.00	1.62 ± 0.00	1.61 ± 0.00	1.62 ± 0.01	1.62 ± 0.00	1.63 ± 0.00	1.62 ± 0.00
10 000	He	1.61 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.63 ± 0.00	1.62 ± 0.00	1.61 ± 0.01	1.61 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.01	1.62 ± 0.01	1.62 ± 0.00
10 000	La	1.62 ± 0.00	1.61 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.61 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.61 ± 0.01	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00
10 000	Le	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.61 ± 0.00	1.62 ± 0.00	1.61 ± 0.00	1.62 ± 0.00	1.61 ± 0.00	1.63 ± 0.01	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.01
10 000	Pw	1.61 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.60 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.61 ± 0.00	1.61 ± 0.00	1.62 ± 0.01	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00
100 000	Ch	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.60 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00
100 000	He	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.60 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00
100 000	La	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.61 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00
100 000	Le	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.61 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00
100 000	Pw	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.61 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00	1.62 ± 0.00

Notes: The parameters used are S0 = 100 K = 100, H = 110, $\sigma = 0.2$, r = 0.05, T = 3, $\delta = 0.1$, and exercise points $t \in \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$. Nb denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

7.1.2 Test-Case Analysis

Table 7.1.2.1: Call Option with Continuous Dividends

S = 90												
N_b	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
M = 100												
0	4.26	±	0.00	4.34	±	0.13	4.02	±	0.10	4.50	±	0.14
1	4.29	±	0.00	4.59	±	0.13	4.44	±	0.12			
2	4.32	±	0.00	4.90	±	0.15	4.75	±	0.13			
3	4.32	±	0.00	5.07	±	0.15	4.66	±	0.14			
4	4.32	±	0.00	5.43	±	0.14	5.09	±	0.16			
5	4.31	±	0.00	5.48	±	0.19	5.04	±	0.12			
M = 1,000												
0	4.26	±	0.00	4.24	±	0.04	4.08	±	0.03	4.36	±	0.05
1	4.29	±	0.00	4.28	±	0.03	4.24	±	0.04			
2	4.32	±	0.00	4.36	±	0.03	4.37	±	0.05			
3	4.32	±	0.00	4.44	±	0.03	4.32	±	0.03			
4	4.32	±	0.00	4.44	±	0.03	4.31	±	0.03			
5	4.31	±	0.00	4.48	±	0.04	4.29	±	0.04			
M = 5,000												
0	4.26	±	0.00	4.28	±	0.02	4.06	±	0.02	4.25	±	0.02
1	4.29	±	0.00	4.30	±	0.01	4.26	±	0.02			
2	4.32	±	0.00	4.33	±	0.02	4.32	±	0.02			
3	4.32	±	0.00	4.36	±	0.02	4.29	±	0.02			
4	4.32	±	0.00	4.32	±	0.01	4.31	±	0.02			
5	4.31	±	0.00	4.33	±	0.02	4.27	±	0.02			
M = 10,000												
0	4.26	±	0.00	4.27	±	0.01	4.05	±	0.01	4.26	±	0.01
1	4.29	±	0.00	4.28	±	0.02	4.25	±	0.01			
2	4.32	±	0.00	4.32	±	0.01	4.30	±	0.02			
3	4.32	±	0.00	4.32	±	0.01	4.31	±	0.02			
4	4.32	±	0.00	4.32	±	0.01	4.35	±	0.02			
5	4.31	±	0.00	4.31	±	0.02	4.28	±	0.01			

Notes: The parameters used are S0 = 90, K = 100, $\sigma = 0.2$, r = 0.05, T = 3, $\delta = 0.1$, and exercise points $t \in \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.2: Call Option with Continuous Dividends

N_b	Asymptotic			OLS			Tikhonov			CART		
S = 100												
M = 100												
0	7.77	±	0.00	7.81	±	0.13	7.64	±	0.14	7.87	±	0.14
1	7.82	±	0.00	8.19	±	0.12	8.26	±	0.13			
2	7.85	±	0.01	8.59	±	0.14	7.98	±	0.14			
3	7.86	±	0.00	9.07	±	0.15	8.41	±	0.17			
4	7.87	±	0.00	8.90	±	0.13	8.48	±	0.13			
5	7.87	±	0.01	9.54	±	0.15	8.24	±	0.17			
M = 1,000												
0	7.77	±	0.00	7.73	±	0.05	7.41	±	0.02	7.73	±	0.02
1	7.82	±	0.00	7.82	±	0.05	7.81	±	0.05			
2	7.85	±	0.01	7.83	±	0.04	7.78	±	0.04			
3	7.86	±	0.00	7.96	±	0.05	7.89	±	0.04			
4	7.87	±	0.00	8.01	±	0.06	7.86	±	0.04			
5	7.87	±	0.01	8.04	±	0.05	7.75	±	0.04			
M = 5,000												
0	7.77	±	0.00	7.81	±	0.02	7.42	±	0.02	7.76	±	0.02
1	7.82	±	0.00	7.78	±	0.02	7.76	±	0.02			
2	7.85	±	0.01	7.87	±	0.02	7.81	±	0.02			
3	7.86	±	0.00	7.88	±	0.01	7.85	±	0.02			
4	7.87	±	0.00	7.89	±	0.02	7.87	±	0.01			
5	7.87	±	0.01	7.91	±	0.02	7.76	±	0.02			
M = 10,000												
0	7.77	±	0.00	7.76	±	0.02	7.44	±	0.01	7.76	±	0.02
1	7.82	±	0.00	7.81	±	0.02	7.77	±	0.01			
2	7.85	±	0.01	7.86	±	0.01	7.84	±	0.01			
3	7.86	±	0.00	7.86	±	0.01	7.85	±	0.01			
4	7.87	±	0.00	7.86	±	0.02	7.87	±	0.02			
5	7.87	±	0.01	7.86	±	0.01	7.78	±	0.01			

Notes: The parameters used are S0 = 100, K = 100, σ = 0.2, r = 0.05, T = 3, δ = 0.1, and exercise points t ∈ {0, 0.5, 1, 1.5, 2, 2.5, 3}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.3: Call Option with Continuous Dividends

S = 110												
N_b	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
						M = 100						
0	12.80	±	0.00	12.95	±	0.16	12.74	±	0.13	12.96	±	0.16
1	12.86	±	0.01	13.10	±	0.21	12.90	±	0.16			
2	12.92	±	0.01	13.73	±	0.15	13.47	±	0.15			
3	12.94	±	0.00	14.06	±	0.22	13.45	±	0.12			
4	12.95	±	0.00	14.13	±	0.12	13.26	±	0.16			
5	12.94	±	0.00	14.50	±	0.18	13.20	±	0.13			
						M = 1,000						
0	12.80	±	0.00	12.74	±	0.05	12.58	±	0.04	12.74	±	0.05
1	12.86	±	0.01	12.88	±	0.04	12.82	±	0.06			
2	12.92	±	0.01	12.95	±	0.04	12.87	±	0.05			
3	12.94	±	0.00	12.96	±	0.05	13.04	±	0.05			
4	12.95	±	0.00	12.98	±	0.05	12.94	±	0.06			
5	12.94	±	0.00	13.08	±	0.05	12.75	±	0.05			
						M = 5,000						
0	12.80	±	0.00	12.80	±	0.02	12.55	±	0.02	12.78	±	0.02
1	12.86	±	0.01	12.82	±	0.03	12.76	±	0.02			
2	12.92	±	0.01	12.89	±	0.02	12.89	±	0.03			
3	12.94	±	0.00	12.94	±	0.02	12.94	±	0.02			
4	12.95	±	0.00	12.97	±	0.01	12.90	±	0.02			
5	12.94	±	0.00	12.94	±	0.02	12.73	±	0.03			
						M = 10,000						
0	12.80	±	0.00	12.78	±	0.02	12.53	±	0.01	12.81	±	0.02
1	12.86	±	0.01	12.88	±	0.01	12.76	±	0.01			
2	12.92	±	0.01	12.91	±	0.02	12.88	±	0.02			
3	12.94	±	0.00	12.94	±	0.01	12.90	±	0.02			
4	12.95	±	0.00	12.99	±	0.01	12.92	±	0.02			
5	12.94	±	0.00	12.93	±	0.01	12.80	±	0.02			

Notes: The parameters used are S0 = 110, K = 100, σ = 0.2, r = 0.05, T = 3, δ = 0.1, and exercise points t ∈ {0, 0.5, 1, 1.5, 2, 2.5, 3}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.4: American-Asian Call Option

S = 90												
N_b	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
M = 100												
0	1.13	±	0.00	1.19	±	0.04	1.07	±	0.03	1.16	±	0.03
1	1.24	±	0.00	1.37	±	0.04	1.13	±	0.04			
2	1.25	±	0.00	1.36	±	0.04	1.33	±	0.05			
3	1.25	±	0.00	1.45	±	0.05	1.27	±	0.05			
4	1.25	±	0.00	1.34	±	0.05	1.24	±	0.05			
5	1.25	±	0.00	1.43	±	0.04	1.30	±	0.03			
M = 1,000												
0	1.13	±	0.00	1.15	±	0.02	1.03	±	0.01	1.14	±	0.01
1	1.24	±	0.00	1.26	±	0.01	1.15	±	0.01			
2	1.25	±	0.00	1.26	±	0.02	1.23	±	0.02			
3	1.25	±	0.00	1.27	±	0.01	1.23	±	0.01			
4	1.25	±	0.00	1.29	±	0.02	1.23	±	0.02			
5	1.25	±	0.00	1.28	±	0.02	1.23	±	0.01			
M = 5,000												
0	1.13	±	0.00	1.13	±	0.01	1.04	±	0.00	1.13	±	0.00
1	1.24	±	0.00	1.25	±	0.01	1.13	±	0.01			
2	1.25	±	0.00	1.25	±	0.01	1.25	±	0.01			
3	1.25	±	0.00	1.26	±	0.01	1.24	±	0.01			
4	1.25	±	0.00	1.25	±	0.01	1.23	±	0.01			
5	1.25	±	0.00	1.24	±	0.01	1.23	±	0.01			
M = 10,000												
0	1.13	±	0.00	1.13	±	0.01	1.05	±	0.00	1.13	±	0.00
1	1.24	±	0.00	1.26	±	0.01	1.14	±	0.00			
2	1.25	±	0.00	1.25	±	0.00	1.25	±	0.00			
3	1.25	±	0.00	1.24	±	0.00	1.25	±	0.01			
4	1.25	±	0.00	1.26	±	0.00	1.24	±	0.01			
5	1.25	±	0.00	1.26	±	0.01	1.22	±	0.01			

Notes: The parameters used are S0 = 90, K = 100, $\sigma = 0.2$, r = 0.09, T = 120/365, t′ = 91/365, $\delta = 0$, and exercise points $\in$ {0, 105/365, 108/365, 111/365, 114/365, 117/365, 120/365}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.5: American-Asian Call Option

S = 100												
Nb	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
M = 100												
0	4.69	±	0.00	4.74	±	0.06	4.24	±	0.04	4.69	±	0.05
1	5.46	±	0.00	5.60	±	0.08	4.85	±	0.05			
2	5.47	±	0.00	5.56	±	0.05	5.47	±	0.08			
3	5.47	±	0.00	5.67	±	0.06	5.46	±	0.06			
4	5.47	±	0.00	5.72	±	0.07	5.39	±	0.08			
5	5.47	±	0.00	5.75	±	0.07	5.46	±	0.07			
M = 1,000												
0	4.69	±	0.00	4.70	±	0.02	4.15	±	0.01	4.69	±	0.01
1	5.46	±	0.00	5.42	±	0.02	4.73	±	0.02			
2	5.47	±	0.00	5.46	±	0.03	5.39	±	0.02			
3	5.47	±	0.00	5.51	±	0.03	5.41	±	0.03			
4	5.47	±	0.00	5.50	±	0.03	5.30	±	0.02			
5	5.47	±	0.00	5.51	±	0.02	5.24	±	0.02			
M = 5,000												
0	4.69	±	0.00	4.70	±	0.01	4.13	±	0.01	4.70	±	0.01
1	5.46	±	0.00	5.46	±	0.01	4.71	±	0.01			
2	5.47	±	0.00	5.46	±	0.01	5.45	±	0.01			
3	5.47	±	0.00	5.46	±	0.01	5.47	±	0.02			
4	5.47	±	0.00	5.47	±	0.01	5.40	±	0.01			
5	5.47	±	0.00	5.47	±	0.01	5.32	±	0.02			
M = 10,000												
0	4.69	±	0.00	4.69	±	0.00	4.12	±	0.00	4.69	±	0.01
1	5.46	±	0.00	5.46	±	0.01	4.71	±	0.01			
2	5.47	±	0.00	5.46	±	0.01	5.44	±	0.01			
3	5.47	±	0.00	5.46	±	0.01	5.43	±	0.01			
4	5.47	±	0.00	5.45	±	0.01	5.44	±	0.01			
5	5.47	±	0.00	5.47	±	0.01	5.33	±	0.01			

Notes: The parameters used are S0 = 100, K = 100, σ = 0.2, r = 0.09, T = 120/365, t′ = 91/365, δ = 0, and exercise points ∈ {0, 105/365, 108/365, 111/365, 114/365, 117/365, 120/365}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.6: American-Asian Call Option

N_b	Asymptotic			OLS			Tikhonov			CART		
S = 110												
M = 100												
0	11.48	±	0.00	11.31	±	0.08	11.04	±	0.04	11.59	±	0.09
1	13.11	±	0.00	13.32	±	0.04	11.55	±	0.09			
2	13.12	±	0.00	13.36	±	0.07	13.08	±	0.07			
3	13.12	±	0.00	13.51	±	0.05	12.87	±	0.05			
4	13.12	±	0.00	13.50	±	0.05	12.63	±	0.07			
5	13.12	±	0.00	13.59	±	0.05	12.82	±	0.10			
M = 1,000												
0	11.48	±	0.00	11.54	±	0.03	11.02	±	0.01	11.48	±	0.02
1	13.11	±	0.00	13.11	±	0.02	11.50	±	0.03			
2	13.12	±	0.00	13.13	±	0.01	12.97	±	0.03			
3	13.12	±	0.00	13.14	±	0.02	12.92	±	0.04			
4	13.12	±	0.00	13.13	±	0.02	12.59	±	0.04			
5	13.12	±	0.00	13.16	±	0.02	12.61	±	0.03			
M = 5,000												
0	11.48	±	0.00	11.48	±	0.01	11.03	±	0.01	11.48	±	0.02
1	13.11	±	0.00	13.10	±	0.01	11.48	±	0.01			
2	13.12	±	0.00	13.12	±	0.01	12.99	±	0.02			
3	13.12	±	0.00	13.12	±	0.01	13.10	±	0.01			
4	13.12	±	0.00	13.12	±	0.01	12.71	±	0.04			
5	13.12	±	0.00	13.13	±	0.01	12.67	±	0.03			
M = 10,000												
0	11.48	±	0.00	11.48	±	0.01	11.02	±	0.00	11.49	±	0.01
1	13.11	±	0.00	13.12	±	0.01	11.52	±	0.01			
2	13.12	±	0.00	13.12	±	0.01	12.87	±	0.04			
3	13.12	±	0.00	13.12	±	0.00	13.12	±	0.00			
4	13.12	±	0.00	13.12	±	0.01	12.82	±	0.04			
5	13.12	±	0.00	13.13	±	0.01	12.66	±	0.04			

Notes: The parameters used are S0 = 110, K = 100, σ = 0.2, r = 0.09, T = 120/365, t′ = 91/365, δ = 0, and exercise points ∈ {0, 105/365, 108/365, 111/365, 114/365, 117/365, 120/365}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.7: Put-Option on a Jump-Diffusion Asset

S = 90												
N_b	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
M = 100												
0	13.77	±	0.00	14.05	±	0.13	12.98	±	0.10	13.88	±	0.09
1	13.91	±	0.00	14.39	±	0.11	13.69	±	0.10			
2	13.94	±	0.00	14.37	±	0.09	14.05	±	0.10			
3	13.95	±	0.00	14.53	±	0.09	14.35	±	0.10			
4	13.95	±	0.00	14.62	±	0.10	14.46	±	0.08			
5	13.94	±	0.00	14.71	±	0.07	14.65	±	0.09			
M = 1,000												
0	13.77	±	0.00	13.79	±	0.03	12.85	±	0.03	13.78	±	0.04
1	13.91	±	0.00	13.97	±	0.03	13.70	±	0.02			
2	13.94	±	0.00	14.07	±	0.03	13.68	±	0.02			
3	13.95	±	0.00	14.08	±	0.03	14.04	±	0.03			
4	13.95	±	0.00	14.11	±	0.02	14.06	±	0.03			
5	13.94	±	0.00	14.08	±	0.03	14.02	±	0.03			
M = 5,000												
0	13.77	±	0.00	13.77	±	0.01	12.88	±	0.01	13.74	±	0.01
1	13.91	±	0.00	13.96	±	0.01	13.69	±	0.02			
2	13.94	±	0.00	13.94	±	0.01	13.68	±	0.01			
3	13.95	±	0.00	13.97	±	0.01	13.96	±	0.01			
4	13.95	±	0.00	13.98	±	0.01	13.96	±	0.01			
5	13.94	±	0.00	13.97	±	0.01	13.98	±	0.01			
M = 10,000												
0	13.77	±	0.00	13.79	±	0.01	12.87	±	0.01	13.77	±	0.01
1	13.91	±	0.00	13.92	±	0.01	13.67	±	0.01			
2	13.94	±	0.00	13.94	±	0.01	13.69	±	0.01			
3	13.95	±	0.00	13.95	±	0.01	13.95	±	0.01			
4	13.95	±	0.00	13.96	±	0.01	13.97	±	0.01			
5	13.94	±	0.00	13.97	±	0.01	13.95	±	0.01			

Notes: The parameters used are S0 = 90, K = 100, $\sigma = \sqrt{0.08}$ r = 0.1, T = 0.5, δ = 0.2, λ = 2, and exercise points t $\in$ {0,0.125,0.25,0.375,0.5}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.8: Put-Option on a Jump-Diffusion Asset

S = 100												
N_b	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
						M = 100						
0	8.54	±	0.00	8.59	±	0.09	7.72	±	0.12	8.65	±	0.10
1	8.70	±	0.00	9.01	±	0.12	8.48	±	0.11			
2	8.71	±	0.00	9.01	±	0.08	8.76	±	0.13			
3	8.72	±	0.00	9.48	±	0.08	8.92	±	0.11			
4	8.72	±	0.00	9.19	±	0.10	8.96	±	0.10			
5	8.71	±	0.00	9.46	±	0.12	9.08	±	0.08			
						M = 1,000						
0	8.54	±	0.00	8.53	±	0.03	7.73	±	0.01	8.53	±	0.01
1	8.70	±	0.00	8.75	±	0.03	8.42	±	0.04			
2	8.71	±	0.00	8.81	±	0.02	8.69	±	0.03			
3	8.72	±	0.00	8.83	±	0.04	8.71	±	0.03			
4	8.72	±	0.00	8.84	±	0.03	8.78	±	0.03			
5	8.71	±	0.00	8.88	±	0.04	8.79	±	0.04			
						M = 5,000						
0	8.54	±	0.00	8.56	±	0.01	7.79	±	0.01	8.55	±	0.01
1	8.70	±	0.00	8.71	±	0.02	8.48	±	0.02			
2	8.71	±	0.00	8.74	±	0.01	8.68	±	0.02			
3	8.72	±	0.00	8.76	±	0.01	8.71	±	0.02			
4	8.72	±	0.00	8.75	±	0.02	8.73	±	0.02			
5	8.71	±	0.00	8.73	±	0.01	8.73	±	0.01			
						M = 10,000						
0	8.54	±	0.00	8.54	±	0.01	7.78	±	0.01	8.53	±	0.01
1	8.70	±	0.00	8.69	±	0.01	8.46	±	0.01			
2	8.71	±	0.00	8.71	±	0.01	8.68	±	0.01			
3	8.72	±	0.00	8.74	±	0.01	8.72	±	0.01			
4	8.72	±	0.00	8.71	±	0.01	8.72	±	0.01			
5	8.71	±	0.00	8.71	±	0.01	8.72	±	0.01			

Notes: The parameters used are S0 = 100, K = 100, $\sigma = \sqrt{0.08}$ r = 0.1, T = 0.5, $\delta = 0.2$, $\lambda = 2$, and exercise points $t \in \{0, 0.125, 0.25, 0.375, 0.5\}$. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.9: Put-Option on a Jump-Diffusion Asset

S = 110												
N_b	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
M = 100												
0	5.10	±	0.00	4.99	±	0.09	4.57	±	0.09	5.27	±	0.12
1	5.19	±	0.00	5.65	±	0.12	5.09	±	0.10			
2	5.18	±	0.00	5.37	±	0.09	5.46	±	0.10			
3	5.19	±	0.00	5.65	±	0.09	5.49	±	0.10			
4	5.19	±	0.00	5.58	±	0.10	5.35	±	0.11			
5	5.19	±	0.00	5.68	±	0.12	5.51	±	0.12			
M = 1,000												
0	5.10	±	0.00	5.02	±	0.03	4.67	±	0.02	5.07	±	0.03
1	5.19	±	0.00	5.23	±	0.04	5.05	±	0.03			
2	5.18	±	0.00	5.27	±	0.04	5.20	±	0.04			
3	5.19	±	0.00	5.24	±	0.04	5.28	±	0.04			
4	5.19	±	0.00	5.30	±	0.03	5.27	±	0.04			
5	5.19	±	0.00	5.32	±	0.03	5.26	±	0.04			
M = 5,000												
0	5.10	±	0.00	5.12	±	0.02	4.70	±	0.02	5.09	±	0.02
1	5.19	±	0.00	5.19	±	0.02	5.06	±	0.02			
2	5.18	±	0.00	5.21	±	0.02	5.21	±	0.02			
3	5.19	±	0.00	5.22	±	0.02	5.18	±	0.02			
4	5.19	±	0.00	5.20	±	0.02	5.17	±	0.02			
5	5.19	±	0.00	5.23	±	0.02	5.23	±	0.01			
M = 10,000												
0	5.10	±	0.00	5.10	±	0.01	4.70	±	0.01	5.13	±	0.01
1	5.19	±	0.00	5.16	±	0.01	5.08	±	0.01			
2	5.18	±	0.00	5.18	±	0.01	5.18	±	0.01			
3	5.19	±	0.00	5.20	±	0.01	5.19	±	0.01			
4	5.19	±	0.00	5.21	±	0.01	5.18	±	0.01			
5	5.19	±	0.00	5.20	±	0.01	5.19	±	0.01			

Notes: The parameters used are S0 = 110, K = 100, $\sigma = \sqrt{0.08}$ r = 0.1, T = 0.5, δ = 0.2, λ = 2, and exercise points t ∈ {0,0.125,0.25,0.375,0.5}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.10: Up-In Barrier Option

S = 90												
N_b	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
M = 100												
0	3.78	±	0.00	4.07	±	0.16	3.84	±	0.10	4.02	±	0.12
1	3.86	±	0.00	4.70	±	0.10	4.19	±	0.12			
2	3.86	±	0.00	4.90	±	0.15	4.57	±	0.12			
3	3.85	±	0.01	5.02	±	0.13	4.34	±	0.18			
4	3.85	±	0.00	5.03	±	0.18	4.44	±	0.18			
5	3.86	±	0.00	5.30	±	0.20	4.46	±	0.13			
M = 1,000												
0	3.78	±	0.00	3.81	±	0.04	3.80	±	0.04	3.83	±	0.05
1	3.86	±	0.00	3.96	±	0.04	3.87	±	0.03			
2	3.86	±	0.00	3.98	±	0.05	3.90	±	0.04			
3	3.85	±	0.01	3.97	±	0.05	3.88	±	0.05			
4	3.85	±	0.00	4.04	±	0.05	3.88	±	0.03			
5	3.86	±	0.00	4.05	±	0.05	3.88	±	0.03			
M = 5,000												
0	3.78	±	0.00	3.76	±	0.02	3.76	±	0.01	3.78	±	0.02
1	3.86	±	0.00	3.87	±	0.02	3.76	±	0.02			
2	3.86	±	0.00	3.86	±	0.03	3.84	±	0.02			
3	3.85	±	0.01	3.86	±	0.02	3.81	±	0.02			
4	3.85	±	0.00	3.90	±	0.02	3.85	±	0.01			
5	3.86	±	0.00	3.88	±	0.02	3.84	±	0.02			
M = 10,000												
0	3.78	±	0.00	3.81	±	0.01	3.77	±	0.01	3.80	±	0.01
1	3.86	±	0.00	3.85	±	0.02	3.77	±	0.01			
2	3.86	±	0.00	3.88	±	0.02	3.83	±	0.01			
3	3.85	±	0.01	3.86	±	0.01	3.86	±	0.01			
4	3.85	±	0.00	3.86	±	0.01	3.84	±	0.01			
5	3.86	±	0.00	3.84	±	0.01	3.85	±	0.01			

Notes: The parameters used are S0 = 90, K = 100, H = 110, σ = 0.2, r = 0.05, T = 3, δ = 0.1, and exercise points t ∈ {0, 0.5, 1, 1.5, 2, 2.5, 3}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.10: Up-In Barrier Option

N_b	Asymptotic			OLS			Tikhonov			CART		
S = 100												
M = 100												
0	7.29	±	0.00	7.81	±	0.21	7.28	±	0.15	7.58	±	0.15
1	7.40	±	0.01	8.19	±	0.14	7.60	±	0.17			
2	7.42	±	0.00	8.59	±	0.13	8.12	±	0.13			
3	7.43	±	0.01	9.07	±	0.21	7.79	±	0.13			
4	7.43	±	0.00	8.90	±	0.17	7.91	±	0.12			
5	7.42	±	0.00	9.54	±	0.17	8.26	±	0.20			
M = 1,000												
0	7.29	±	0.00	7.37	±	0.04	7.24	±	0.02	7.25	±	0.02
1	7.40	±	0.01	7.44	±	0.06	7.27	±	0.06			
2	7.42	±	0.00	7.51	±	0.04	7.41	±	0.05			
3	7.43	±	0.01	7.54	±	0.04	7.50	±	0.04			
4	7.43	±	0.00	7.59	±	0.03	7.49	±	0.04			
5	7.42	±	0.00	7.57	±	0.05	7.38	±	0.03			
M = 5,000												
0	7.29	±	0.00	7.29	±	0.03	7.27	±	0.02	7.30	±	0.02
1	7.40	±	0.01	7.40	±	0.02	7.26	±	0.02			
2	7.42	±	0.00	7.43	±	0.02	7.38	±	0.03			
3	7.43	±	0.01	7.43	±	0.02	7.45	±	0.02			
4	7.43	±	0.00	7.47	±	0.02	7.43	±	0.02			
5	7.42	±	0.00	7.42	±	0.02	7.32	±	0.02			
M = 10,000												
0	7.29	±	0.00	7.29	±	0.02	7.26	±	0.01	7.30	±	0.01
1	7.40	±	0.01	7.39	±	0.01	7.24	±	0.02			
2	7.42	±	0.00	7.45	±	0.01	7.38	±	0.02			
3	7.43	±	0.01	7.45	±	0.01	7.39	±	0.02			
4	7.43	±	0.00	7.44	±	0.02	7.40	±	0.01			
5	7.42	±	0.00	7.44	±	0.01	7.33	±	0.01			

Notes: The parameters used are S0 = 100, K = 100, H = 110, σ = 0.2, r = 0.05, T = 3, δ = 0.1, and exercise points t ∈ {0, 0.5, 1, 1.5, 2, 2.5, 3}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.12: Up-In Barrier Option

	S = 110											
N_b	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
	M = 100											
0	12.81	±	0.00	13.13	±	0.17	12.55	±	0.16	12.95	±	0.16
1	12.87	±	0.01	13.23	±	0.12	12.83	±	0.18			
2	12.92	±	0.01	13.70	±	0.16	13.16	±	0.17			
3	12.94	±	0.00	13.85	±	0.22	13.20	±	0.16			
4	12.95	±	0.00	14.16	±	0.15	13.51	±	0.19			
5	12.94	±	0.01	14.67	±	0.18	13.32	±	0.19			
	M = 1,000											
0	12.81	±	0.00	12.81	±	0.05	12.58	±	0.04	12.86	±	0.04
1	12.87	±	0.01	12.98	±	0.06	12.73	±	0.05			
2	12.92	±	0.01	12.88	±	0.04	12.92	±	0.05			
3	12.94	±	0.00	12.91	±	0.05	12.94	±	0.05			
4	12.95	±	0.00	13.07	±	0.06	12.90	±	0.05			
5	12.94	±	0.01	12.98	±	0.05	12.73	±	0.04			
	M = 5,000											
0	12.81	±	0.00	12.80	±	0.02	12.53	±	0.02	12.79	±	0.02
1	12.87	±	0.01	12.82	±	0.02	12.80	±	0.02			
2	12.92	±	0.01	12.95	±	0.02	12.86	±	0.02			
3	12.94	±	0.00	12.98	±	0.03	12.92	±	0.02			
4	12.95	±	0.00	12.94	±	0.02	12.93	±	0.02			
5	12.94	±	0.01	12.94	±	0.02	12.78	±	0.02			
	M = 10,000											
0	12.81	±	0.00	12.78	±	0.02	12.55	±	0.01	12.81	±	0.01
1	12.87	±	0.01	12.87	±	0.02	12.74	±	0.01			
2	12.92	±	0.01	12.92	±	0.01	12.88	±	0.02			
3	12.94	±	0.00	12.96	±	0.01	12.91	±	0.01			
4	12.95	±	0.00	12.95	±	0.01	12.89	±	0.01			
5	12.94	±	0.01	12.93	±	0.01	12.81	±	0.02			

Notes: The parameters used are S0 = 110, K = 100, H = 110, $\sigma = 0.2$, r = 0.05, T = 3, $\delta = 0.1$, and exercise points $t \in \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.13: Up-Out Barrier Option

						S = 90						
N_b	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
						M = 100						
0	1.12	±	0.00	1.16	±	0.03	1.16	±	0.04	1.20	±	0.03
1	1.12	±	0.00	1.16	±	0.03	1.07	±	0.04			
2	1.12	±	0.00	1.14	±	0.03	1.13	±	0.03			
3	1.12	±	0.00	1.14	±	0.04	1.16	±	0.03			
4	1.12	±	0.00	1.22	±	0.03	1.20	±	0.03			
5	1.12	±	0.00	1.21	±	0.03	1.16	±	0.03			
						M = 1,000						
0	1.12	±	0.00	1.13	±	0.01	1.11	±	0.01	1.12	±	0.01
1	1.12	±	0.00	1.13	±	0.01	1.11	±	0.01			
2	1.12	±	0.00	1.11	±	0.01	1.12	±	0.01			
3	1.12	±	0.00	1.11	±	0.01	1.13	±	0.01			
4	1.12	±	0.00	1.10	±	0.01	1.13	±	0.01			
5	1.12	±	0.00	1.13	±	0.01	1.13	±	0.01			
						M = 5,000						
0	1.12	±	0.00	1.12	±	0.01	1.11	±	0.00	1.12	±	0.01
1	1.12	±	0.00	1.12	±	0.00	1.13	±	0.00			
2	1.12	±	0.00	1.12	±	0.00	1.12	±	0.01			
3	1.12	±	0.00	1.13	±	0.00	1.12	±	0.00			
4	1.12	±	0.00	1.12	±	0.00	1.13	±	0.01			
5	1.12	±	0.00	1.13	±	0.00	1.12	±	0.00			
						M = 10,000						
0	1.12	±	0.00	1.12	±	0.00	1.11	±	0.00	1.12	±	0.00
1	1.12	±	0.00	1.12	±	0.00	1.12	±	0.00			
2	1.12	±	0.00	1.12	±	0.00	1.12	±	0.00			
3	1.12	±	0.00	1.13	±	0.00	1.12	±	0.00			
4	1.12	±	0.00	1.13	±	0.00	1.12	±	0.00			
5	1.12	±	0.00	1.12	±	0.00	1.13	±	0.00			

Notes: The parameters used are S0 = 90, K = 100, H = 110, σ = 0.2, r = 0.05, T = 3, δ = 0.1, and exercise points t ∈ {0, 0.5, 1, 1.5, 2, 2.5, 3}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.14: Up-Out Barrier Option

S = 100												
N_b	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
M = 100												
0	1.62	±	0.00	1.61	±	0.03	1.54	±	0.04	1.67	±	0.04
1	1.62	±	0.00	1.64	±	0.04	1.64	±	0.04			
2	1.62	±	0.00	1.70	±	0.03	1.64	±	0.04			
3	1.62	±	0.00	1.70	±	0.04	1.59	±	0.03			
4	1.62	±	0.00	1.63	±	0.04	1.64	±	0.06			
5	1.62	±	0.00	1.61	±	0.03	1.65	±	0.05			
M = 1,000												
0	1.62	±	0.00	1.61	±	0.01	1.60	±	0.01	1.62	±	0.01
1	1.62	±	0.00	1.62	±	0.01	1.64	±	0.01			
2	1.62	±	0.00	1.63	±	0.01	1.63	±	0.02			
3	1.62	±	0.00	1.62	±	0.01	1.62	±	0.01			
4	1.62	±	0.00	1.65	±	0.01	1.62	±	0.01			
5	1.62	±	0.00	1.62	±	0.01	1.62	±	0.01			
M = 5,000												
0	1.62	±	0.00	1.62	±	0.00	1.62	±	0.01	1.62	±	0.01
1	1.62	±	0.00	1.62	±	0.01	1.62	±	0.01			
2	1.62	±	0.00	1.62	±	0.00	1.63	±	0.01			
3	1.62	±	0.00	1.61	±	0.01	1.62	±	0.01			
4	1.62	±	0.00	1.61	±	0.01	1.63	±	0.01			
5	1.62	±	0.00	1.62	±	0.00	1.62	±	0.01			
M = 10,000												
0	1.62	±	0.00	1.62	±	0.00	1.61	±	0.00	1.62	±	0.00
1	1.62	±	0.00	1.61	±	0.00	1.62	±	0.00			
2	1.62	±	0.00	1.62	±	0.00	1.62	±	0.00			
3	1.62	±	0.00	1.62	±	0.00	1.61	±	0.01			
4	1.62	±	0.00	1.62	±	0.00	1.62	±	0.00			
5	1.62	±	0.00	1.62	±	0.00	1.62	±	0.00			

Notes: The parameters used are S0 = 100, K = 100, H = 110, σ = 0.2, r = 0.05, T = 3, δ = 0.1, and exercise points t ∈ {0, 0.5, 1, 1.5, 2, 2.5, 3}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

Table 7.1.2.15: Up-Out Barrier Option

						S = 110						
N_b	**Asymptotic**			**OLS**			**Tikhonov**			**CART**		
						M = 100						
0	1.68	±	0.00	1.68	±	0.06	1.72	±	0.05	1.55	±	0.05
1	1.68	±	0.00	1.75	±	0.06	1.62	±	0.06			
2	1.68	±	0.00	1.67	±	0.05	1.60	±	0.06			
3	1.68	±	0.00	1.71	±	0.04	1.64	±	0.06			
4	1.68	±	0.00	1.85	±	0.05	1.65	±	0.04			
5	1.68	±	0.00	1.66	±	0.05	1.62	±	0.05			
						M = 1,000						
0	1.68	±	0.00	1.69	±	0.02	1.71	±	0.02	1.68	±	0.02
1	1.68	±	0.00	1.69	±	0.02	1.68	±	0.02			
2	1.68	±	0.00	1.67	±	0.02	1.70	±	0.02			
3	1.68	±	0.00	1.69	±	0.01	1.70	±	0.02			
4	1.68	±	0.00	1.68	±	0.02	1.68	±	0.02			
5	1.68	±	0.00	1.67	±	0.01	1.71	±	0.02			
						M = 5,000						
0	1.68	±	0.00	1.69	±	0.01	1.68	±	0.01	1.67	±	0.01
1	1.68	±	0.00	1.68	±	0.01	1.70	±	0.01			
2	1.68	±	0.00	1.68	±	0.01	1.68	±	0.01			
3	1.68	±	0.00	1.68	±	0.01	1.68	±	0.01			
4	1.68	±	0.00	1.68	±	0.01	1.68	±	0.01			
5	1.68	±	0.00	1.67	±	0.00	1.68	±	0.01			
						M = 10,000						
0	1.68	±	0.00	1.68	±	0.01	1.67	±	0.01	1.68	±	0.01
1	1.68	±	0.00	1.68	±	0.00	1.68	±	0.00			
2	1.68	±	0.00	1.68	±	0.01	1.69	±	0.00			
3	1.68	±	0.00	1.68	±	0.01	1.68	±	0.01			
4	1.68	±	0.00	1.69	±	0.01	1.68	±	0.00			
5	1.68	±	0.00	1.68	±	0.01	1.67	±	0.00			

Notes: The parameters used are S0 = 110, K = 100, H = 110, σ = 0.2, r = 0.05, T = 3, δ = 0.1, and exercise points t ∈ {0, 0.5, 1, 1.5, 2, 2.5, 3}. N_b denotes the highest degree of polynomial basis functions. Standard errors are reported after the ± sign based on 20 replications. The asymptotic value is computed by OLS regression with 100,000 paths. Green fill color means zero to two standard errors away from the asymptotic value. Yellow means two to three standard errors away. Red means above three standard errors away.

7.2 MATLAB code

7.2.1 Function FullAnalysis

```
function FullAnalysis

%% INPUT PARAMETERS
WriteOutputTo = 'MATLAB'; % 'MATLAB' / 'TEXTFILE'
S0 = [100,90,110];
K = 100;
H = 110;
M = [100000,10000,5000,1000,100];
b = 5;
Replications = 20;
Antithetic = 'YES';
method = {'OLS','tikh','CART'};
optionType = {'ContD','Asian','JumpD','UpInBarrier','UpOutBarrier'};
poly = {'Ch','He','La','Le','Pw'};

for METHOD = 1:length(method)
    for OPTIONTYPE = 1:length(optionType)

        if strncmp(optionType{OPTIONTYPE},'Asian',4)
            r = 0.09;
            T = 120/365;
            N = 6;
            dt = T/N;
            div = 0;
            tao = 91/365; % Pre-specified date
            t = [0 105/365 108/365 111/365 114/365 117/365 120/365];
            sigma = 0.2;
            lambda = 0;
        elseif strncmp(optionType{OPTIONTYPE},'JumpD',4)
            r = 0.01;
            T = 0.5;
            N = 4;
            dt = T/N;
            div = 0.2;
            tao = 0;
            sigma = sqrt(0.08);
            lambda = 2;
            t = 0:dt:0.5;
        else
            r = 0.05;
            T = 3;
            N = 6;
            div = 0.1;
            dt = T/N;
            tao = 0;
            sigma = 0.2;
            t = 0:dt:T;
            lambda = 0;
        end

        for PRICEPATHS = 1:length(S0)

            for m = 1:length(M)
                for POLY = 1:length(poly)

                    if strncmp(method{METHOD},'OLS',3) ...
                            || strncmp(method{METHOD},'tikh',3)
                        d = b+1;
                    elseif strncmp(method{METHOD},'CART',3)
                        d = 1;
                    end

                    S = zeros(M(m),N+1);
                    S_AV = zeros(M(m),N+1);
                    Estimator = zeros(Replications,d);
                    StdErr = zeros(Replications,d);
                    time = zeros(Replications,d);

                    for k = 1:d
```

```
                        tic
                        for R = 1:Replications

                            % Definition of b
                            b = k-1;

                            %% SIMULATION OF ASSET PATHS
                            [S,S_AV] = AssetPaths(optionType,S0,div,r,...
                                sigma,N,M,dt,lambda,Antithetic,S,S_AV,...
                                OPTIONTYPE,PRICEPATHS,m,H);

                            %% CASH FLOW
                            if strcmp(Antithetic,'YES') == 1
                                CF = LSM(method,optionType,K,r,M,N,b,poly,...
                                    S,S_AV,dt,tao,t,Antithetic,OPTIONTYPE,...
                                    METHOD,m,POLY);
                                CF_AV = LSM_AV(optionType,Antithetic,t,tao,...
                                    method,K,r,M,N,b,poly,S,S_AV,dt,...
                                    OPTIONTYPE,METHOD,m,POLY);
                                CF = (CF(:,1) + CF_AV(:,1))./2;
                                PVCF = CF*exp(-r*dt);
                            else
                                CF = LSM(method,optionType,K,r,M,N,b,poly,...
                                    S,S_AV,dt,tao,t,Antithetic,OPTIONTYPE,...
                                    METHOD,m,POLY);
                                PVCF = CF(:,1)*exp(-r*dt);
                            end


                            %% OUTPUT COMPUTATIONS
                            [OptionPrice,STD] = normfit(PVCF);
                            Estimator(R,k) = OptionPrice;

                            %% CPU TIME
                            time(R,k) = toc/R;
                        end

                        StdErr = std(Estimator(:,k))/sqrt(Replications);

                        %% STORE OUTPUT IN TABLE
                        if strcmp(WriteOutputTo,'MATLAB') == 1
                            formatSpec = ('%8s %10s %4.0f %8.0f  %-3s ...
                            %6.0f %4.2f %4.2f %4.2f\n');
                            fprintf(formatSpec,method{METHOD},...
                                optionType{OPTIONTYPE},S0(PRICEPATHS),...
                                M(m),poly{POLY},b,mean(Estimator(:,k)),...
                                StdErr,mean(time(:,k)));
                        elseif strcmp(WriteOutputTo,'TEXTFILE') == 1
                            f = fopen('Output.txt','a+');
                            formatSpec = ('%8s %10s %4.0f %4.0f  %-3s ...
                            %6.0f %4.5f %4.5f %4.5f\n');
                            fprintf(f,formatSpec,method{METHOD},...
                                optionType{OPTIONTYPE},...
                                S0(PRICEPATHS),M(m),poly{POLY},b,...
                                mean(Estimator(:,k)),...
                                StdErr,mean(time(:,k)));
                            fclose(f);
                        end
                    end
                end
            end
        end
    end
end
```

7.2.2 Function AssetPaths

```
function [S,S_AV] = AssetPaths(optionType,S0,div,r,sigma,N,M,dt,lambda,...
    Antithetic,S,S_AV,OPTIONTYPE,PRICEPATHS,m,H)

z = randn(M(m),N);
z_AV = -z;
drift = r-div-0.5*(sigma^2);
S(:,1) = S0(PRICEPATHS); % Placing S0 in first column
S_AV(:,1) = S0(PRICEPATHS);

if (strncmp(optionType{OPTIONTYPE},'ContD',5) || ...
        strncmp(optionType{OPTIONTYPE},'Asian',5) || ...
        strncmp(optionType{OPTIONTYPE},'Paris',5) || ...
        strncmp(optionType{OPTIONTYPE},'LookB',5) || ...
        strncmp(optionType{OPTIONTYPE},'Barrier',5))
    for i = 1:M(m)
        for j = 1:N
            S(i,j+1) = S(i,j)*exp((drift)*dt+sigma*sqrt(dt)*z(i,j));
```

```
            S_AV(i,j+1) = S_AV(i,j)*exp((drift)*dt+sigma*sqrt(dt)*z_AV(i,j));
        end
    end
    S(:,1) = []; % Removing redundant column of S0
    S_AV(:,1) = []; % Removing redundant column of S0

elseif strncmp(optionType{OPTIONTYPE},'JumpD',5)
    Poisson = poissrnd(lambda*dt,M(m),N);
    for i = 1:M(m)
        for j = 1:N
            S(i,j+1) = S(i,j)*exp((r-sigma^2/2)*dt+sigma*sqrt(dt)*z(i,j)+...
                Poisson(i,j)*sqrt(dt)*(div^2/2+div*z(i,j)));
            if strcmp(Antithetic,'YES') == 1
                S_AV(i,j+1) = S_AV(i,j)*exp((r-sigma^2/2)*dt+sigma*...
                    sqrt(dt)*z_AV(i,j)+Poisson(i,j)*sqrt(dt)*...
                    (div^2/2+div*z_AV(i,j)));
            end
        end
    end
    S(:,1) = []; % Removing redundant column of S0
    S_AV(:,1) = []; % Removing redundant column of S0

elseif strncmp(optionType{OPTIONTYPE},'UpInBarrier',5)
    for i = 1:M(m)
        for j = 1:N

            S(i,j+1) = S(i,j)*exp((drift)*dt+sigma*sqrt(dt)*z(i,j));

            if N > 1
                if S(i,j) > H
                    S(i,j+1) = S(i,j)*exp((drift)*dt+sigma*sqrt(dt)*z(i,j));
                elseif S(i,j:-1:1) < H
                    S(i,j) = 0;
                    S(i,end)= 0;
                end

            elseif N == 1
                if S(i,j+1) > H
                    S(i,j+1) = S(i,j)*exp((drift)*dt+sigma*sqrt(dt)*z(i,j));
                elseif S(i,j+1) < H
                    S(i,j+1) = 0;
                end
            end

            if strcmp(Antithetic,'YES') == 1
                S_AV(i,j+1) = S_AV(i,j)*exp((drift)*dt+sigma*sqrt(dt)*z_AV(i,j));

                if N > 1
                    if S_AV(i,j) > H
                        S_AV(i,j+1) = S_AV(i,j)*exp((drift)*dt+sigma*sqrt(dt)*...
                            z_AV(i,j));
                    elseif S_AV(i,j:-1:1) < H
                        S_AV(i,j) = 0;
                        S_AV(i,end)= 0;
                    end

                elseif N == 1
                    if S_AV(i,j+1) > H
                        S_AV(i,j+1) = S_AV(i,j)*exp((drift)*dt+sigma*sqrt(dt)*...
                            z_AV(i,j));
                    elseif S_AV(i,j+1) < H
                        S_AV(i,j+1) = 0;
                    end
                end
            end
        end
    end
    S(:,1) = []; % Removing redundant column of S0
    S_AV(:,1) = []; % Removing redundant column of S0

elseif strncmp(optionType{OPTIONTYPE},'UpOutBarrier',5)
    for i = 1:M(m)
        for j = 1:N

            S(i,j+1) = S(i,j)*exp((drift)*dt+sigma*sqrt(dt)*z(i,j));
            if S(i,j+1) > H
                S(i,j+1) = 0;
            end

            if strcmp(Antithetic,'YES') == 1
                S_AV(i,j+1) = S_AV(i,j)*exp((drift)*dt+sigma*sqrt(dt)*z_AV(i,j));

                if S_AV(i,j+1) > H
                    S_AV(i,j+1) = 0;
                end
```

```
            end
        end
    end
    S(:,1) = []; % Removing redundant column of S0
    S_AV(:,1) = []; % Removing redundant column of S0
end
```

7.2.3 Function LSM

```
function CF = LSM(method,optionType,K,r,M,N,b,poly,S,S_AV,dt,tao,t,...
    Antithetic,OPTIONTYPE,METHOD,m,POLY)

[CF,~,~] = Payoff(optionType,OPTIONTYPE,S,S_AV,K,tao,t,N,M,Antithetic,m);

if (strncmp(optionType{OPTIONTYPE},'ContD',5) | ...
        strncmp(optionType{OPTIONTYPE},'Asian',5) | ...
        strncmp(optionType{OPTIONTYPE},'Paris',5) | ...
        strncmp(optionType{OPTIONTYPE},'LookB',5) | ...
        strncmp(optionType{OPTIONTYPE},'UpInBarrier',5) | ...
        strncmp(optionType{OPTIONTYPE},'UpOutBarrier',5))
    for j = N-1:-1:1
        itmP = find(S(:,j) > K);
        X = S(itmP,j); % Store in-the-money paths
        Y = CF(itmP,j+1)*exp(-r*dt); % Discounting in-the-money cash flows
        XM = BasisFunct(X,b,poly,POLY); % Weighted Laguerre polynomials
        Beta = pinv(XM)*Y; % Linear regression
        ExpCont = Regression(method,Y,Beta,b,XM,METHOD); ...
            % Conditional expected continuation value
        E = X-K; % Immediate exercise Value
        exP = itmP(ExpCont<E); % Identify immediate exercise
        rest = setdiff(1:M(m),exP);
        CF(exP,j) = E(ExpCont<E);
        CF(rest,j) = CF(rest,j+1)*exp(-r*dt);
    end

elseif strncmp(optionType{OPTIONTYPE},'JumpD',5)
    for j = N-1:-1:1
        itmP = find(S(:,j) < K);
        X = S(itmP,j); % Store in-the-money paths
        Y = CF(itmP,j+1)*exp(-r*dt); % Discounting in-the-money cash flows
        XM = BasisFunct(X,b,poly,POLY); % Weighted Laguerre polynomials
        Beta = pinv(XM)*Y; % Linear regression
        ExpCont = Regression(method,Y,Beta,b,XM,METHOD); ...
            % Conditional expected continuation value
        E = K-X; % Immediate exercise Value
        exP = itmP(ExpCont < E); % Identify immediate exercise
        rest = setdiff(1:M(m),exP);
        CF(exP,j) = E(ExpCont < E);
        CF(rest,j) = CF(rest,j+1)*exp(-r*dt);
    end
end
```

7.2.4 Function LSM_AV

```
function CF_AV = LSM_AV(optionType,Antithetic,t,tao,method,K,r,M,N,b,poly,...
    S,S_AV,dt,OPTIONTYPE,METHOD,m,POLY)

[~,CF_AV,~] = Payoff(optionType,OPTIONTYPE,S,S_AV,K,tao,t,N,M,Antithetic,m);

if (strncmp(optionType{OPTIONTYPE},'ContD',5) | ...
        strncmp(optionType{OPTIONTYPE},'Asian',5) | ...
        strncmp(optionType{OPTIONTYPE},'Paris',5) | ...
        strncmp(optionType{OPTIONTYPE},'LookB',5) | ...
        strncmp(optionType{OPTIONTYPE},'UpInBarrier',5) | ...
        strncmp(optionType{OPTIONTYPE},'UpOutBarrier',5))
    for j = N-1:-1:1
        itmP_AV = find(S_AV(:,j) > K);
        X_AV = S_AV(itmP_AV,j); % Store in-the-money paths
        Y_AV = CF_AV(itmP_AV,j+1)*exp(-r*dt); % Discounting in-the-money cash flows
        XM_AV = BasisFunct_AV(X_AV,b,poly,POLY); % Weighted Laguerre polynomials
        Beta_AV = pinv(XM_AV)*Y_AV; % Linear regression
        ExpCont_AV = Regression_AV(method,Y_AV,Beta_AV,b,XM_AV,METHOD); ...
            % Conditional expected continuation value
        E_AV = X_AV-K; % Immediate exercise Value
        exP_AV = itmP_AV(ExpCont_AV<E_AV); % Identify immediate exercise
        rest_AV = setdiff(1:M(m),exP_AV);
        CF_AV(exP_AV,j) = E_AV(ExpCont_AV<E_AV);
        CF_AV(rest_AV,j) = CF_AV(rest_AV,j+1)*exp(-r*dt);
    end
```

```
elseif strncmp(optionType{OPTIONTYPE},'JumpD',5)
    for j = N-1:-1:1
        itmP_AV = find(S_AV(:,j) < K);
        X_AV = S_AV(itmP_AV,j); % Store in-the-money paths
        Y_AV = CF_AV(itmP_AV,j+1)*exp(-r*dt); % Discounting in-the-money cash flows
        XM_AV = BasisFunct_AV(X_AV,b,poly,POLY); % Weighted Laguerre polynomials
        Beta_AV = pinv(XM_AV)*Y_AV; % Linear regression
        ExpCont_AV = Regression_AV(method,Y_AV,Beta_AV,b,XM_AV,METHOD); ...
            % Conditional expected continuation value
        E_AV = K-X_AV; % Immediate exercise Value
        exP_AV = itmP_AV(ExpCont_AV < E_AV); % Identify immediate exercise
        rest_AV = setdiff(1:M(m),exP_AV);
        CF_AV(exP_AV,j) = E_AV(ExpCont_AV < E_AV);
        CF_AV(rest_AV,j) = CF_AV(rest_AV,j+1)*exp(-r*dt);
    end
end
```

7.2.5 Function Payoff

```
function [CF,CF_AV,SBar] = Payoff(optionType,OPTIONTYPE,S,S_AV,K,tao,...
    t,N,M,Antithetic,m)

CF = zeros(M(m),N);
CF_AV = zeros(M(m),N);
SBar = zeros(M(m),N);
SBar_AV = zeros(M(m),N);
IntermMat=zeros(M(m),N-1);

% CONTINUOUS DIVIDENDS
if strncmp(optionType{OPTIONTYPE},'ContD',5) | ...
        strncmp(optionType{OPTIONTYPE},'UpInBarrier',5) | ...
        strncmp(optionType{OPTIONTYPE},'UpOutBarrier',5)
    CF(:,end) = max((S(:,end)-K),0); % Computes payoff at time T
    CF_AV(:,end) = max((S_AV(:,end)-K),0); % Computes payoff at time T

    % JUMP DIFFUSION OPTION
elseif (strncmp(optionType{OPTIONTYPE},'JumpD',5))
    CF(:,end) = max(K-S(:,end),0);
    if strcmp(Antithetic,'YES') == 1
        CF_AV(:,end) = max(K-S_AV(:,end),0); % Computes payoff at time T
    else
        CF_AV = 0;
    end

    % ASIAN OPTION
elseif (strncmp(optionType{OPTIONTYPE}, 'Asian',5))
    for i = 1:M(m)
        for j = 1:N
            SBar(i,j) = S(i,j)./(t(j)-tao+1);
        end
    end
    CF = max(SBar-K,0);

    if strcmp(Antithetic,'YES') == 1
        for i = 1:M(m)
            for j = 1:N
                SBar_AV(i,j) = S_AV(i,j)./(t(j)-tao+1);
            end
        end
        CF_AV = max(SBar_AV-K,0);
    else
        CF_AV = 0;
    end

else
    error('No option of this type found')
end
```

7.2.6 Function BasisFunct

```
function XM = BasisFunct(X,b,poly,POLY)

XM = ones(size(X));

% Chebyshev
if poly{POLY} == 'Ch'
    if b == 0
        XM = ones(size(X));
    else
        XM(:,2) = X;
        XM1 = 1;
        XM2 = X;
        if b >= 2
```

```
            A = XM1;
            B = XM2;
            for f=2:b
                LegMat = 2*X.*B-A;
                XM(:,f+1) = LegMat;
                A = B;
                B = LegMat;
            end
        end
    end

    % Hermite
elseif poly{POLY} == 'He'
    if b == 0
        XM = ones(size(X));
    else
        XM(:,2) = X;
        XM1 = 1;
        XM2 = X;
        if b >= 2
            A = XM1;
            B = XM2;
            for f = 2:b
                LegMat = X.*B-(f-1)*A;
                XM(:,f+1) = LegMat;
                A = B;
                B=LegMat;
            end
        end
    end

    % Laguerre
elseif poly{POLY} == 'La'
    if b == 0
        XM = ones(size(X));
    else
        XM(:,2)=X;
        XM1 = 1;
        XM2=1-X;
        if b>=2
            A=XM1;
            B=XM2;
            for f=2:b
                LegMat= (1/f) * ( (2* (f-1) +1-X) .*B- (f-1) *A) ;
                XM(:,f+1)=LegMat;
                A=B;
                B=LegMat;
            end
        end
    end

    % Legendre
elseif poly{POLY} == 'Le'
    if b == 0
        XM = ones(size(X));
    else
        XM(:,2)=X;
        XM1=1;
        XM2=X;
        if b>=2
            A=XM1;
            B=XM2;
            for f=2:b
                LegMat=(1/f)*((2*(f-1)+1)*X.*B-(f-1)*A);
                XM(:,f+1)=LegMat;
                A=B;
                B=LegMat;
            end
        end
    end

    % Powers
elseif poly{POLY} == 'Pw'
    if b == 0
        XM = ones(size(X));
    else
        for f=1:b
            XM(:,f+1)=X.^f;
        end
    end
else
    error('Illegal polynomial')
end
end
```

7.2.7 Function BasisFunct_AV

```
function XM_AV = BasisFunct_AV(X_AV,b,poly,POLY)

XM_AV = ones(size(X_AV));

% Chebyshev
if poly{POLY} == 'Ch'
    if b == 0
        XM_AV = ones(size(X_AV));
    else
        XM_AV(:,2) = X_AV;
        XM1 = 1;
        XM2 = X_AV;
        if b >= 2
            A = XM1;
            B = XM2;
            for f=2:b
                LegMat = 2*X_AV.*B-A;
                XM_AV(:,f+1) = LegMat;
                A = B;
                B = LegMat;
            end
        end
    end

    % Hermite
elseif poly{POLY} == 'He'
    if b == 0
        XM_AV = ones(size(X_AV));
    else
        XM_AV(:,2) = X_AV;
        XM1 = 1;
        XM2 = X_AV;
        if b >= 2
            A = XM1;
            B = XM2;
            for f = 2:b
                LegMat = X_AV.*B-(f-1)*A;
                XM_AV(:,f+1) = LegMat;
                A = B;
                B=LegMat;
            end
        end
    end

    % Laguerre
elseif poly{POLY} == 'La'
    if b == 0
        XM_AV = ones(size(X_AV));
    else
        XM_AV(:,2)=X_AV;
        XM1 = 1;
        XM2=1-X_AV;
        if b>=2
            A=XM1;
            B=XM2;
            for f=2:b
                LegMat= (1/f) * ( (2* (f-1) +1-X_AV) .*B- (f-1) *A) ;
                XM_AV(:,f+1)=LegMat;
                A=B;
                B=LegMat;
            end
        end
    end

    % Legendre
elseif poly{POLY} == 'Le'
    if b == 0
        XM_AV = ones(size(X_AV));
    else
        XM_AV(:,2)=X_AV;
        XM1=1;
        XM2=X_AV;
        if b>=2
            A=XM1;
            B=XM2;
            for f=2:b
                LegMat=(1/f)*((2*(f-1)+1)*X_AV.*B-(f-1)*A);
                XM_AV(:,f+1)=LegMat;
                A=B;
                B=LegMat;
            end
        end
    end
```

```
    % Powers
elseif poly{POLY} == 'Pw'
    if b == 0
        XM_AV = ones(size(X_AV));
    else
        for f=1:b
            XM_AV(:,f+1)=X_AV.^f;
        end
    end
else
    error('Illegal polynomial')
end
end
```

7.2.8 Function Regression

```
function ExpCont = Regression(method,Y,Beta,b,XM,METHOD)

if strncmp(method{METHOD},'OLS',3)
    ExpCont = XM*Beta;

elseif strncmp(method{METHOD},'tikh',3)
    L = eye(size(XM)); % Weight matrix set to the identity matrix
    [U,s,V] = csvd (XM);
    lambda_l = l_curve(U,s,Y);
    Beta = tikhonov(U,s,V,Y,lambda_l);
    ExpCont = XM*Beta;
    if Beta == 0
        Beta = zeros(b+1,1);
    end
    if isempty(ExpCont)
        ExpCont = zeros (0,1);
    end

elseif strncmp(method{METHOD},'CART',3)
    if isempty(XM)
        ExpCont = zeros(0,1);
    else
        Beta = classregtree(XM,Y);
        ExpCont = eval(Beta,XM);
    end

end
end
```

7.2.9 Function Regression_AV

```
function ExpCont_AV = Regression_AV(method,Y_AV,Beta_AV,b,XM_AV,METHOD)

if strncmp(method{METHOD},'OLS',3)
    ExpCont_AV = XM_AV*Beta_AV;

elseif strncmp(method{METHOD},'tikh',3)
    L = eye(size(XM_AV));
    [U,s,V] = csvd (XM_AV);
    lambda_l = l_curve(U,s,Y_AV);
    Beta_AV = tikhonov(U,s,V,Y_AV,lambda_l);
    ExpCont_AV = XM_AV*Beta_AV;
    if Beta_AV == 0
        Beta_AV = zeros(b+1,1);
    end
    if isempty(ExpCont_AV)
        ExpCont_AV = zeros (0,1);
    end

elseif strncmp(method{METHOD},'CART',3)
    if isempty(XM_AV)
        ExpCont_AV = zeros(0,1);
    else
        Beta_AV = classregtree(XM_AV,Y_AV);
        ExpCont_AV = eval(Beta_AV,XM_AV);
    end

end
end
```

7.2.10 Function csvc

```
function [U,s,V] = csvd(A,tst)

% This function computes the compact form of the SCD of A
```

```
% CSVD Compact singular value decomposition.
% s = csvd(A)
% [U,s,V] =csvd(A)
% fU,s,V3 = csvd(A,'full')

% Computes the compact form of the SVD of A:
%      A = U*diag(s)*V',
% t where
%     U is m-by-min(m,n)
%     s is min(m,n)-by-1
%     V is n-by-min(m,n).

% If a second argument is present, the full U and V are returned.


% copyright Per Christian Hansen, IMM, 06/22/93.


if (nargin==1)
 if (nargout > 1)
   [m,n] = size(A);
   if (m >= n)
     [U,s,V] = svd(full(A),0); s = diag(s);
   else
     [V,s,U] = svd(full(A)',0); s = diag(s);
   end
 else
   U = svd(full(A));
 end
else
 if (nargout > 1)
   [U,s,V] = svd(full(A)); s = diag(s);
 else
   U = svd(full(A));
 end
end
```

7.2.11 Function l_curve

```
function [reg_corner, rho, eta, reg_param] = l_curve(U,sm,Y)

% This function computes recursively the polynomial

% L CURVE Plot the L-curve and find its "corner".
% [reg_corner,rho,eta,reg_param] = l_curve(U,sm,b)
% sm = [sigma, mu]
% Plots the L-shaped curve of eta, the solution norm 11 x or
% semi-norm II L x       as a function of rho, the residual norm
% ||^ x - b li, for the Tikhonov regularization method
% The corresponding reg. parameters are returned in reg_param.
% Note that 'Tikh'require either U and s (standard-
% form regularization) or U and sm (general-form regularization),
% while 'mtvsd' requires U and s as well as L and V.
%  Since the output arguments are specified, then the corner of the
% L-curve is identified and the corresponding reg. parameter reg corner is returned.

% Reference: P. C. Hansen & D. P. O'Leary, "The use of the L-curve
% in the regularization of discrete ill-posed problems", SIAM J. Sci.
% Comput. 14 (1993), pp. 1487-1503.

% copyright Per Christian Hansen, IMM, July 26, 2007.

% Set defaults.

 npoints = 200; % Number of points on the L-curve for Tikh
 smin_ratio = 16*eps; % Smallest regularization parameter.



 % Initialization.
 [m, n] = size(U); [p,ps] = size(sm);
 if (nargout > 0), locate = 1; else locate = 0; end
 beta = U'*Y; beta2 = norm(Y)^2 - norm(beta)^2;
 if isempty (sm) || isempty (U)
   reg_corner = zeros (size(Y,2),1);
 else
   if (ps==1)
  s = sm; beta = beta(1:p);
   else
  s = sm(p:-1:1,1)./sm(p:-1:1,2); beta = beta(p:-1:1);
   end
 xi = beta(1:p)./s;
```

```
%Regularization
 eta = zeros(npoints,1); rho = eta; reg_param = eta; s2 = s.^2;
 reg_param(npoints) = max([s(p),s(1)*smin_ratio]);
 ratio = (s(1)/reg_param(npoints))^(1/(npoints-1));
 for i=npoints-1:-1:1, reg_param(i) = ratio*reg_param(i+1); end
 for i=1:npoints
   f = s2./(s2 + reg_param(i)^2);
   eta(i) = norm(f.*xi);
   rho(i) = norm((1-f).*beta(1:p));
 end
 if (m > n && beta2 > 0), rho = sqrt(rho.^2 + beta2); end

 % Locate the "corner" of the L-curve, if required.
  [reg_corner,rho_c,eta_c] = l_corner(rho,eta,reg_param,U,sm,Y);

% Make plot
marker = '-';
plot_lc(rho,eta,marker,ps,reg_param)

 ax = axis;
 hold on;
 loglog([min(rho)/100,rho_c],[eta_c,eta_c],':r',...
     [rho_c,rho_c],[min(eta)/100,eta_c],':r');
 title(['L-curve,','Tikhonov Regularization','corner at',num2str(reg_corner)]);
 axis (ax)
 hold off;

 end
```

7.2.12 Function l_corner

```
function [reg_c,rho_c,eta_c] = l_corner(rho,eta,reg_param,U,s,Y)

% This function computes recursively the polynomial.

% L_CORNER Locate the "corner" of the L-curve.
%  [reg_c,rho_c,eta_c] = l_corner(rho,eta,reg_param,U,s,Y)
% Locates the "corner" of the L-curve in log-log scale.
% It is assumed that corresponding values of
% || Ax - b 11, || Lx H1
% and the regularization parameter are stored in the arrays rho, eta,
% and reg_param, respectively (such as the output
% from routine l_curve).
% If nargin = 3, then no particular method is assumed, and if
% nargin = 2 then it is issued that reg_param = 1:length(rho).

% This function needs the Spline Toolbox

% copyright Per Christian Hansen, 1MM, July 26, 2007.

% Set default regularization method.
  if (nargin <= 3)
    method = 'none';
    if (nargin==2), reg_param = (1:length(rho))'; end
  end

  % Set this logical variable to 1 (true) if the corner algorithm
  % should always be used, even if the Spline Toolbox is available.
  alwayscorner = 0;

  % Set threshold for skipping very small singular values in the
  % analysis of a discrete L-curve.
  s_thr = eps; % Neglect singular values less than s_thr.

  % Set default parameters for treatment of discrete L-curve.
  deg = 2; % Degree of local smoothing polynomial.
  q    = 2; % Half-width of local smoothing interval.
  order = 4; % Order of fitting 2-D spline curve.

 % Initialization
  if (length(rho) < order)
   error('Too few data points for L-curve analysis')
  end
  if (nargin > 3)
    [p,ps] = size(s); [m,n] = size(U);
   beta = U'*Y;
   if (m>n), b0 = Y - U*beta; end
   if (ps==2)
     s = s(p:-1:1,1)./s(p:-1:1,2);
     beta = beta(p:-1:1);
   end
   xi = beta./s;
 end
```

```
 % Restrict the analysis of the L-curve according to M (if specified).
 if (nargin==8)
   index = find(eta < m);
   rho = rho(index); eta = eta(index); reg_param = reg_param(index);
 end

 % The L-curve is differentiable; computation of curvature in
 % log-log scale is easy.

% Compute g = - curvature of L-curve.
 g = lcfun(reg_param,s,beta,xi);

 % Locate the corner. If the curvature is negative everywhere,
 % then define the leftmost point of the L-curve as the corner.
 [gmin,gi] = min(g);
 reg_c = fminbnd('lcfun', reg_param(min(gi+1,length(g))), ...
     reg_param(max(gi-1,1)), optimset('Display','off'),s,beta,xi); % Minimizer.
kappa_max = - lcfun(reg_c,s,beta,xi); % Maximum curvature.

if (kappa_max < 0)
    lr = length(rho);
    reg_c = reg_param(lr); rho_c = rho(lr); eta_c = eta(lr);
else
    f = (s.^2)./(s.^2 + reg_c^2);
    eta_c = norm(f.*xi);
    rho_c = norm((1-f).*beta);
if (m>n), rho_c = sqrt(rho_c^2 + norm(b0)^2); end
end
```

7.2.13 Function tikhonov

```
function [x_lambda,rho,eta] = tikhonov(U,s,V,b,lambda,x_0)

% This function computes recursively the polynomial
% (applies to Tikhonov Regularization only)

%TIKHONOV Tikhonov regularization.
% [xlambda,rho,eta] = tikhonov(U,s,V,b,lambda,x_0)
% [x_lambda,rho,eta] = tikhonov(U,sm,X,b,lambda,x_0)
% sm = (sigma,mu]
% Computes the Tikhonov regularized solution x_lambda. If the SVD
% is used, i.e. if U, s, and V are specified, then standard-form
% regularization is applied:
% min { || A x - b 11'2 + lambda^2 11 x - x_O ||^2 )
% If, on the other hand, the GSVD is used, i.e. if U, sm, and X
% are specified, then general-form regularization is applied:
% min { || ^ x b ||^2 lambda^2 11 L (x    ||^2
% If x_O is not specified, then x_O = 0 is used
% Note that x 0 cannot be used if A is underdetermined and L -= I.
% If lambda is a vector, then x lambda is a matrix such that
% W   x_lambda = ( xlambda(1), Tclambda(2), -- I ?
% The solution norm (standard-form case) or seminorm (general-form
% case) and the residual norm are returned in eta and rho.

% copyright Per Christian Hansen, IMM, April 14, 2003.

% Reference: A. N. Tikhonov & V. Y. Arsenin, "Solutions of
% Ill-Posed Problems", Wiley, 1977.

% Initialization.
if (min(lambda) < 0)
  error('Illegal regularization parameter lambda')
end

if isempty (s) || isempty (U) || isempty (V)
    x_lambda = 0;
else

m = size(U,1);
n = size(V,1);
[p,ps] = size(s);
beta = U(:,1:p)'*b;
zeta = s(:,1).*beta;
ll = length(lambda); x_lambda = zeros(n,ll);
rho = zeros(ll,1); eta = zeros(ll,1);

% Treat each lambda separately.
if (ps==1)

  % The standard-form case.
  if (nargin==6), omega = V'*x_0; end
  for i=1:ll
    if (nargin==5)
      x_lambda(:,i) = V(:,1:p)*(zeta./(s.^2 + lambda(i)^2));
```

```
      rho(i) = lambda(i)^2*norm(beta./(s.^2 + lambda(i)^2));
    else
      x_lambda(:,i) = V(:,1:p)*...
        ((zeta + lambda(i)^2*omega)./(s.^2 + lambda(i)^2));
      rho(i) = lambda(i)^2*norm((beta - s.*omega)./(s.^2 + lambda(i)^2));
    end
    eta (i) = norm(x_lambda(:,i));
  end
  if (nargout > 1 && size(U,1) > p)
    rho = sqrt(rho.^2 + norm(b - U(:,1:n)*[beta;U(:,p+1:n)'*b])^2);
  end

elseif (m>=n)

  % The overdetermined or square general-form case.
  gamma2 = (s(:,1)./s(:,2)).^2;
  if (nargin==6), omega = V\x_0; omega = omega(1:p); end
  if (p==n)
   x0 = zeros(n,1);
  else
   x0 = V(:,p+1:n)*U(:,p+1:n)'*b;
  end

    for i=1:ll
      if (nargin==5)
        xi = zeta./(s(:,1).^2 + lambda(i)^2*s(:,2).^2);
        x_lambda(:,i) = V(:,1:p)*xi + x0;
        rho(i) = lambda(i)^2*norm(beta./(gamma2 + lambda(i)^2));
      else
        xi = (zeta + lambda(i)^2*(s(:,2).^2).*omega)./...
              (s(:,1).^2 + lambda(i)^2*s(:,2).^2);
        x_lambda(:,i) = V(:,1:p)*xi + x0;
        rho(i) = lambda(i)^2*norm((beta - s(:,1).*omega)./...
                  (gamma2 + lambda(i)^2));
      end
      eta(i) = norm(s(:,2).*xi);
    end
    if (nargout > 1 && size(U,1) > p)
      rho = sqrt(rho.^2 + norm (b - U(:,1:n)*[beta;U(:,p+1:n)'*b])^2);
    end

  else

    % The underdetermined general-form case.
    gamma2 = (s(:,1)./s(:,2)).^2;
    if (nargin==6), error('x_0 not allowed'), end
    if (p==m)
      x0 = zeros(n,1);
    else
      x0 = V(:,p+1:m)*U(:,p+1:m)'*b;
    end
    for i=1:ll
      xi = zeta./(s(:,1).^2 + lambda(i)^2*s(:,2).^2);
      x_lambda(:,i) = V(:,1:p)*xi + x0;
      rho(i) = lambda(i)^2*norm(beta./(gamma2 + lambda(i)^2));
      eta(i) = norm(s(:,2).*xi);
    end

end

 end
```