UPPSALA
UNIVERSITET

IT Licentiate theses
2016-001

Radial Basis Function Methods for Pricing Multi-Asset Options

VICTOR SHCHERBAKOV

UPPSALA UNIVERSITY
Department of Information Technology

UPPSALA
UNIVERSITET

Radial Basis Function Methods for Pricing Multi-Asset Options

Victor Shcherbakov
victor.shcherbakov@it.uu.se

January 2016

Division of Scientific Computing
Department of Information Technology
Uppsala University
Box 337
SE-751 05 Uppsala
Sweden

http://www.it.uu.se/

Dissertation for the degree of Licentiate of Philosophy in Scientific Computing with specialization in Numerical Analysis

© Victor Shcherbakov 2016
ISSN 1404-5117
Printed by the Department of Information Technology, Uppsala University, Sweden

Abstract

The price of an option can under some assumptions be determined by the solution of the Black–Scholes partial differential equation. Often options are issued on more than one asset. In this case it turns out that the option price is governed by the multi-dimensional version of the Black–Scholes equation. Options issued on a large number of underlying assets, such as index options, are of particular interest, but pricing such options is a challenge due to the "curse of dimensionality". The multi-dimensional PDE turn out to be computationally expensive to solve accurately even in quite a low number of dimensions. In this thesis we develop a radial basis function partition of unity method for pricing multi-asset options up to moderately high dimensions. Our approach requires the use of a lower number of node points per dimension than other standard PDE methods, such as finite differences or finite elements, thanks to a high order convergence rate. Our method shows good results for both European style options and American style options, which allow early exercise. For the options which do not allow early exercise, the method exhibits an exponential convergence rate under node refinement. For options that allow early exercise the option pricing problem becomes a free boundary problem. We incorporate two different approaches for handling the free boundary into the radial basis function partition of unity method: a penalty method, which leads to a nonlinear problem, and an operator splitting method, which leads to a splitting scheme. We show that both methods allow for locally high algebraic convergence rates, but it turns out that the operator splitting method is computationally more efficient than the penalty method. The main reason is that there is no need to solve a nonlinear problem, which is the case in the penalty formulation.

Acknowledgments

First of all I would like to acknowledge my main advisor Elisabeth Larsson for introducing me to the subject and for her invaluable help and patience. Secondly, I would like to thank my co-advisor Lina von Sydow and the entire Computational Finance group including Slobodan Milovanović, Josef Höök and Per Lötstedt for sharing ideas and having a nice research environment. I am also grateful to all my colleagues at TDB for providing a pleasant and friendly atmosphere at work. Finally, I would like to thank the Anna Maria Lundin Foundation for being a sponsor for many of my research trips.

List of Papers

This thesis is based on the following papers

I V. Shcherbakov and E. Larsson. *Radial basis function partition of unity methods for pricing vanilla basket options.* Accepted for publication in Computers & Mathematics with applications.

II L. von Sydow, L.J. Höök, E. Lindström, S. Milovanović, J. Persson, V. Shcherbakov, Y. Shpolyanskiy, S. Sirén, J. Toivanen, J. Waldén, M. Wiktorsson, J. Levesley, J. Li, C. Oosterlee, M. Ruijter, A. Toropov, and Y. Zhao. *BENCHOP — The BENCHmarking project in Option Pricing.* International Journal of Computer Mathematics, 92(12), 2361–2379, 2015.

III V. Shcherbakov. *Radial basis function partition of unity operator splitting method for pricing American multi-asset options.* Submitted.

Reprints were made with permission from the publishers.

Contents

1 Introduction **3**

2 The Black–Scholes theory **5**
2.1 The Black–Scholes market assumptions . . . 5
2.2 The Black–Scholes equation . . . 6
2.3 Risk neutral valuation . . . 7
2.4 The multi-dimensional Black–Scholes equation . . . 8
2.5 The European option . . . 8
2.6 The American option . . . 10
2.7 Variational inequalities for the American option . . . 12
2.8 Penalised formulation for the American option . . . 14
2.9 Overview of numerical methods . . . 14

3 Radial basis function methods **19**
3.1 Radial basis function partition of unity method . . . 21
3.2 Convergence of RBF methods . . . 22

4 Summary of papers **25**
4.1 Paper I . . . 25
4.2 Paper II . . . 26
4.3 Paper III . . . 27

5 Conclusion and outlook **31**

Chapter 1

Introduction

In the last decades derivatives have begun to play a significant role in finance. A derivative can be seen as a contract on a more basic underlying asset, such as a stock, gold or wheat. Financial contracts have been known since a long time ago, but only from 1973, when The Chicago Board Options Exchange (CBOE) started trading call options, derivatives turned into well-defined contracts. Typical derivatives are futures, forwards, options, swaps, which are now actively traded throughout the world in large amounts. Today's derivative market volume is often estimated at more than $1,200,000,000,000,000. Therefore, there is a high demand for correct contract prices.

In the same year when CBOE first started trading options, Fisher Black and Myron Scholes published their famous work "The Pricing of Options and Corporate Liabilities" [3], where they derived a partial differential equation, whose solution gives the risk neutral expectation of the price of the option over time, and a pricing formula for plain European call and put options. Robert C. Merton was the first to extend the mathematical understanding of the Black–Scholes model. Scholes and Merton received The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel for this work in 1997. Fisher Black, unfortunately, passed away just a couple of years before the award.

The academic research has dramatically increased in volume after this publication. Nowadays there is a large variety of models and methods which are used for pricing financial derivatives. Since the range of traded contracts is wide, there does not exist a closed form pricing formula for every contract. Therefore, contract prices have to be approximated numerically. Numerical methods that are commonly used can be subdivided into two groups: stochastic and deterministic. Stochastic methods, such as Monte Carlo methods [12] are applied to the stochastic differential formulations which describe the underlying asset dynamics. They simulate the underlying process to a future date, and then discount the expected payoff to the present date with a certain discount factor, that depends on the present interest rate and the time window. These methods are very flexible with respect to the underlying dynamic process, which can be a simple geometric Brownian motion (GBM), or a more complex process with, for example, jumps incorporated to better capture the market features. Binomial tree methods [4] similarly to Monte Carlo methods simulate the underlying dynamics, letting the underlying asset jump or fall at the next time step with certain probabilities, up to a future date and then discount the expected cash flow back to the present date. Both these types of methods are widely used in industry, because of their flexibility and their ease of implementation. A disadvantage of such methods is usually a low convergence rate. On the other hand, we have deterministic methods, such as finite difference methods [29], finite element methods [32], spectral methods [8], which deal with the Black–Scholes partial differential equation (if an underlying process other than the geometric Brownian motion is used to simulate the asset behaviour, then other types of PDEs, PIDEs or FPDEs can be derived to define the contract price). These methods usually have good convergence rates, but they suffer from the "curse of dimensionality", i.e., the problem size increases exponentially with the number of underlying assets. This is not the case for the stochastic methods, where the problem size scales linearly with the number of underlying assets. A more detailed overview of the numerical methods is given in Section 2.9.

Chapter 2

The Black–Scholes theory

In this chapter we introduce the Black–Scholes market model and describe its relation to risk neutral option pricing.

2.1 The Black–Scholes market assumptions

In the celebrated paper [3] Black and Scholes made a number of market assumptions, which allow to determine option prices. The assumptions are as follows:

- The asset price follows the lognormal random walk.
- The interest rate r is known and is constant over time.
- There are no transaction costs associated with hedging a portfolio.
- The asset pays no dividends (can be relaxed).
- The market is liquid, i.e., trading can take place continuously.
- There are no penalties to short selling, and fractional holdings are allowed.
- There are no arbitrage opportunities, meaning that all risk-free portfolios must earn the same return.

Definition 1. An ***arbitrage*** on a financial market is a self-financed portfolio $\{h_t : 0 \le t \le T\}$ such that

$$\begin{aligned} &U^h(0) = 0, \\ &\mathbb{P}(U^h(T) \ge 0) = 1, \\ &\mathbb{P}(U^h(T) > 0) > 0, \end{aligned}$$

where U^h denotes the value of the portfolio. This simply means that all risk-free portfolios must earn the same return. Thus, an arbitrage is a risk-free profit. The marked is called ***arbitrage-free*** if there are no arbitrage opportunities.

The Black–Scholes model has been and is still being criticised. Some of the assumptions have been disregarded. For example, dividends can be paid out discretely or continuously, and other dynamics than the GBM asset dynamics have been proposed to better capture the stock behaviour. The Merton jump-diffusion process [21] incorporates jumps into the stock dynamics to reflect sudden changes which occur in stock prices due to, e.g., a crisis. Also, the distribution of the Merton process has heavy tails. That is, the occurrence of rare events is more probable than in GBM. This can also be observed on exchanges. The Heston model [13] assumes that the volatility has a stochastic nature. The volatility is indeed a stochastic variable, and even derivatives on volatility are traded. Another commonly used asset dynamic is the Carr–Geman–Madan–Yor (CGMY) process [5], that incorporates jumps and reflects heavy tails and skewness of the asset return distributions. Nevertheless, in this work we adhere to the classical model, since this is not the main focus, but our focus is on the numerical methods.

Definition 2. Consider a financial market with vector price process X_t. Let $\mathcal{F}_t^X$ denote the σ-algebra generated by X_t on the interval $[0, t]$, i.e., it contains the information generated by the random process X_t. A ***contingent claim with the maturity time T***, also called T-claim, is any stochastic variable $\mathcal{X} \in \mathcal{F}_T^X$. A contingent claim is called a ***simple*** claim if it is of the form $\mathcal{X} = \Phi(X_T)$, where $\Phi(X_T)$ is called the ***contract function*** or the ***payoff function***.

In other words, a contingent claim is a contract which postulates that the holder will receive the ammount $\mathcal{X}$ (can be positive or negative) at the time of maturity $t = T$. The aim is to determine the arbitrage-free price of a simple T-claim for $t < T$.

2.2 The Black–Scholes equation

Assume that the market consists of two assets, one risk-free asset B_t (bank account) and one risky asset X_t (stock), with dynamics given by

$$dB_t = rB_t dt, \tag{2.1}$$

$$dX_t = \alpha X_t dt + \sigma X_t dW_t^{\mathbb{P}}, \tag{2.2}$$

where the short interest rate r, the mean return of the stock α and the volatility σ are deterministic constants, and $W_t^{\mathbb{P}}$ is a Wiener process under the objective probability measure $\mathbb{P}$, that describes our real world, i.e., the world in which the trading takes place. We consider a simple contingent claim of the form

$$\mathcal{X} = \Phi(X_T), \tag{2.3}$$

and we assume that this claim can be traded on the exchange and that its price process is defined as follows

$$\Pi(t; \mathcal{X}) = V(t, X_t), \tag{2.4}$$

where V is some smooth function. Thus, the problem is reduced to finding the form of V. Applying the Itô formula and taking into account that the market is arbitrage-free we can show that for every $t > 0$ on the entire real positive line, V has to satisfy the following (deterministic) PDE

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V}{\partial x^2} + rx\frac{\partial V}{\partial x} - rV &= 0, && (2.5) \\ V(T, x) &= \Phi(x), && (2.6) \end{aligned}$$

which is known as the ***Black–Scholes equation*** (for detailed derivations see [2]). The final condition (2.6) specifies that at the time of maturity T an amount, depending on the payoff function Φ, will be paid out. Note that here x defines the deterministic value of the underlying asset, whereas X_t is its stochastic representation.

2.3 Risk neutral valuation

In financial applications the objective probability measure $\mathbb{P}$ is often substituted by an equivalent risk neutral probability measure $\mathbb{Q}$, under which the assets B_t, X_t have the following dynamics

$$dB_t = rB_t dt, \tag{2.7}$$

$$dX_t = rX_t dt + \sigma X_t dW_t^{\mathbb{Q}}, \tag{2.8}$$

where $W_t^{\mathbb{Q}}$ is a $\mathbb{Q}$-Wiener process different from $W_t^{\mathbb{P}}$. Such a probability measure $\mathbb{Q}$ exists if the market is complete, i.e., if every contingent claim can be hedged. The probability measure $\mathbb{Q}$ describes the world in which all investors are indifferent to risk, that is why it is called risk neutral probability measure. Note that now the expected return on the stock X_t is equal to r. In the risk neutral world the expected return on all securities is the risk-free interest rate.

We know that the arbitrage free price of a contingent claim $\mathcal{X} = \Phi(X_T)$ is given by $\Pi(t; \Phi(X_T)) = V(t, X_t)$, where the function V is the solution of the pricing equation (2.5)–(2.6). We notice that the solution of (2.5)–(2.6), given the asset dynamic (2.8), can be found using the Feynman–Kac representation theorem [2]

$$V(t,x) = e^{-r(T-t)}\mathbb{E}_t^{\mathbb{Q}}\left[\Phi(X_T)|X_t = x\right], \qquad (2.9)$$

where $\mathbb{E}^{\mathbb{Q}}$ is the expected value, taken under the risk neutral probability measure $\mathbb{Q}$.

The economical interpretation of the formula (2.9) is the following: given today's date t and today's stock value x, the derivative price is taken as the expectation of the final payment $\mathbb{E}_t^{\mathbb{Q}}\left[\Phi(X_T)|X_t = x\right]$ and discounted to the present value using the discount factor $e^{-r(T-t)}$. It is important to note that the expected value in the formula (2.9) cannot be taken under the objective probability measure $\mathbb{P}$.

2.4 The multi-dimensional Black–Scholes equation

Formulation (2.5)–(2.6) can be extended to the case when the contract is written on several underlying assets. Thus, the arbitrage-free price of a simple T-claim $\mathcal{X} = \Phi(X_T^1, X_T^2, \ldots, X_T^m)$ at time $t < T$ is $\Pi(t; \mathcal{X}) = V(t, X_t^1, X_t^2, \ldots, X_t^m)$, where $V(t, x_1, x_2, \ldots, x_m)$ satisfies the multi-dimensional Black–Scholes equation

$$\begin{aligned}\frac{\partial V}{\partial t} + \frac{1}{2}\sum_{i,j=1}^{m} \sigma_i\sigma_j\rho_{ij}x_i x_j \frac{\partial^2 V}{\partial x_i \partial x_j} + \sum_{i=1}^{m} r x_i \frac{\partial V}{\partial x_i} - rV &= 0 \qquad (2.10)\\ V(T, \vec{x}) &= \Phi(\vec{x}), \qquad (2.11)\end{aligned}$$

where $\vec{x} = (x_1, \ldots, x_m)$, σ_i is the volatility of the i-th asset and ρ_{ij} is the correlation between assets i and j. Such multi-dimensional problems often arise in practice, when it is needed to price multiple-asset contracts such as index options, multi-asset swaps, cross-currency options, and become a challenge from the computational point of view.

2.5 The European option

In this works we focus on pricing simple contingent claims that are called options.

Definition 3. An ***option*** is a financial contract which gives the buyer the right, but not the obligation, to buy or sell an underlying asset at a specified strike price on or before a specified date. An option that gives to the owner the right to buy an asset at a specific price is referred to as a ***call***. An option that gives to the owner the right to sell an asset at a specific price is referred to as a ***put***. An option that may only be exercised on the expiration date is called a ***European option***. An option that may be exercised any time before or at the expiration date is called an ***American option***.

For call options the payoff function is given by

$$\Phi(x) = \max(x - K, 0) := (x - K)^+, \tag{2.12}$$

that is, if the asset value is greater than the strike price, then it is worth to exercise the option, if it is not, then the option expires worthless. For put options the payoff function is given by

$$\Phi(x) = (K - x)^+, \tag{2.13}$$

that is the opposite situation, if the asset price is less than the strike, then the option should be exercised, and if it is greater than the strike we let the option expire unexercised.

Thus, prices of a European call option C and a European put option P satisfy the Black–Scholes equation. For the European call it reads

$$\begin{aligned} \frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 C}{\partial x^2} + rx\frac{\partial C}{\partial x} - rC &= 0, \quad x > 0,\ t \in [0, T), & (2.14)\\ C(T, x) &= (x - K)^+. & (2.15) \end{aligned}$$

And for the European put we have

$$\begin{aligned} \frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 P}{\partial x^2} + rx\frac{\partial P}{\partial x} - rP &= 0, \quad x > 0,\ t \in [0, T), & (2.16)\\ P(T, x) &= (K - x)^+. & (2.17) \end{aligned}$$

Prices of European multi-asset options can be defined through multi-dimensional versions of equations (2.14), (2.16), that can be written out similarly to (2.10).

Under assumptions of risk neutral valuation, the Black–Scholes formula, which determines the value of a European single-asset option, can be derived from equation (2.9). Thus, for a European call option $C(t, x)$

$$C(t, x) = xN(d_1) - e^{-r(T-t)}KN(d_2), \tag{2.18}$$

where $N(x)$ is the standard normal cumulative distribution function

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-z^2/2} dz, \tag{2.19}$$

and

$$d_1 = \frac{\ln\left(\frac{x}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}, \tag{2.20}$$

$$d_2 = \frac{\ln\left(\frac{x}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} = d_1 - \sigma\sqrt{T-t}. \tag{2.21}$$

Similarly, the price can be defined for a European put option $P(t,x)$

$$P(t,x) = e^{-r(T-t)}KN(-d_2) - xN(-d_1). \tag{2.22}$$

Numerical approximation of option prices is necessary, unless we price a single-asset European option with constant coefficients r and σ. Numerical experiments cannot be performed on an infinite domain, therefore, in order to enable numerical experiments the infinite domain of definition is truncated at the point x_∞, which is located far enough from the origin, and boundary conditions are assigned there.

2.6 The American option

As follows from Definition 3 an American-style option can be exercised any time before or at the maturity. Merton, Samuelson and McKean [19, 20, 26, 27] showed that prices of an American call option C and an American put option P satisfy the following free boundary problems for the Black–Scholes equation. Here we assume that the asset continuously pays out an amount of proportional dividend d. This assumption is especially important to highlight some properties of American call options. Thus, for the call option we have

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 C}{\partial x^2} + (r-d)x\frac{\partial C}{\partial x} - rC = 0, \quad 0 \le x < x_c(t),\ t \in [0,T), \tag{2.23}$$

subject to the final and boundary conditions

$$C(T,x) = (x-K)^+, \tag{2.24}$$

$$C(t,0) = 0, \tag{2.25}$$

$$C(t,x_c(t)) = (x_c(t)-K)^+, \tag{2.26}$$

$$C_x(t,x_c(t)) = 1. \tag{2.27}$$

And for the put option the problem reads

$$\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 P}{\partial x^2} + (r-d)x\frac{\partial P}{\partial x} - rP = 0, \quad x > x_p(t),\ t \in [0,T) \tag{2.28}$$

subject to the final and boundary conditions

$$P(T,x) = (K-x)^+, \tag{2.29}$$
$$P(t,0) = K, \tag{2.30}$$
$$P(t,x_p(t)) = (K-x_p(t))^+, \tag{2.31}$$
$$P_x(t,x_p(t)) = -1. \tag{2.32}$$

The free boundaries $x_c(t)$ and $x_p(t)$ are called the *optimal exercise boundaries* and fulfil the properties, see, e.g., [16]

c.i $x_c(t) > \max\left(\frac{r}{d}K, K\right), \quad t \in [0,T)$,

c.ii $x \in [0, x_c(t)) \iff C(t,x) > (x-K)^+, \quad t \in [0,T)$,

c.iii $x \geq x_c(t) \iff C(t,x) = (x-K)^+, \quad t \in [0,T)$,

for the call option; and

p.i $x_p(t) < \max\left(\frac{r}{d}K, K\right), \quad t \in [0,T)$,

p.ii $x \geq x_p(t) \iff P(t,x) > (K-x)^+, \quad t \in [0,T)$,

p.iii $x \in [0, x_p(t)) \iff P(t,x) = (K-x)^+, \quad t \in [0,T)$,

for the put option. From the properties we can notice that if the dividend $d = 0$, then the value of an American call is equivalent to the value of a European call with the same strike price, and if the risk-free interest rate $r = 0$, then the value of an American put is equivalent to the value of a European put with the same strike price.

The financial interpretation of the optimal exercise boundary $x_c(t)$ is that when the asset spot price x is in the *exercise region* $[x_c(t), \infty)$, it is optimal to exercise an American call option and receive the amount $(x-K)^+$, and if the asset price x is in the *hold region* $[0, x_c(t))$, then it is optimal to hold an American call option, because its value is greater than the payoff that would be received if the option was exercised. Analogous situation for an American put option. If the asset price x is in the *exercise region* $[0, x_p(t))$ then it is worth to exercise the option and receive the payoff. If the asset price is in the *hold region* then it is optimal to hold an American put option, because it is worth more than the payoff. A sketch of the free boundary for an American call option is drawn in Figure 2.1.

Due to the early exercise opportunity, which gives more flexibility to the holder, an American option is worth more than a European option written on the same asset. Figure 2.2 shows the price difference between a European put option and an American put option issued on the same asset.

No closed form solution exists for the American option pricing problem, therefore numerical methods have to be applied to evaluate the option price.

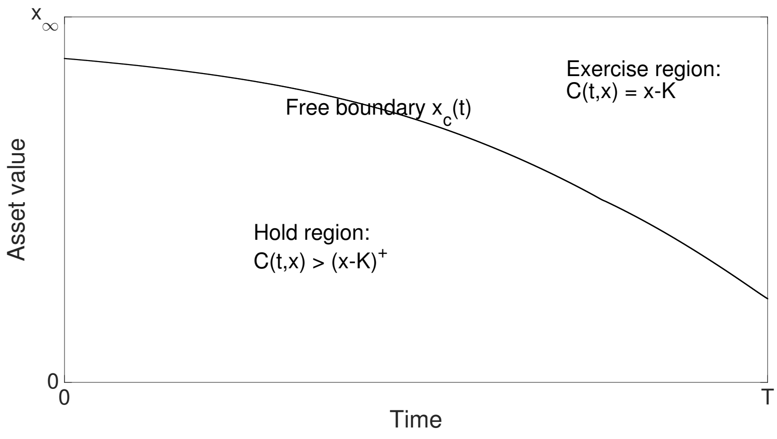


Figure 2.1: The optimal exercise boundary, the hold region and the exercise region for an American call option.

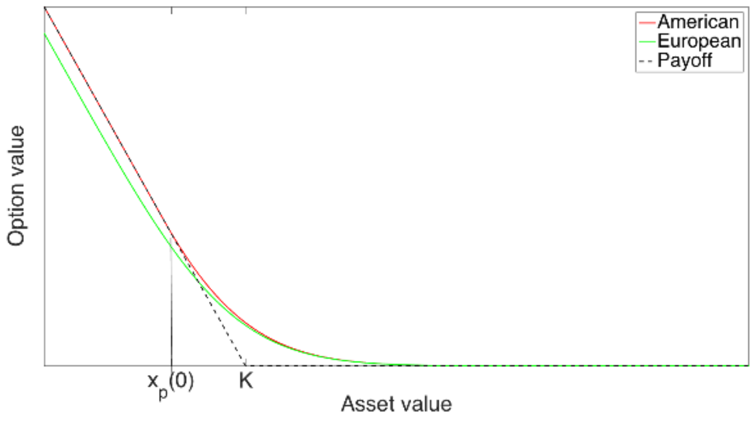


Figure 2.2: A sketch of prices for an American put option and a European put option issued on the same asset with respect to the spot price of the asset.

2.7 Variational inequalities for the American option

For the numerical approach it is not convenient to solve the problem in the domain bounded by a moving boundary. It is possible to modify the Black–Scholes formulation to a problem stated on the entire semi strip $x > 0$,

$t \in [0, T]$, including the exercise regions. In this case the American option pricing problem is viewed as variational inequalities.

Let us denote the spatial Black–Scholes operator as

$$\mathcal{L}V = \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V}{\partial x^2} + (r-d)x\frac{\partial V}{\partial x} - rV. \tag{2.33}$$

With this notation the Black–Scholes equation in the hold region reads as

$$\frac{\partial V}{\partial t} + \mathcal{L}V = 0. \tag{2.34}$$

What happens with this equation in the exercise regions? We substitute the payoff into the equation and take into account quantities (2.27), (2.32)

$$\frac{\partial V}{\partial t} + \mathcal{L}V = \begin{cases} (r-d)x - r(x-K) = -dx + rK, & \text{for call options,} \\ -(r-d)x - r(K-x) = dx - rK, & \text{for put options.} \end{cases}$$

From (c.i) we have that for call options $dx > rK$ in the exercise region and from (p.i) we have that for put options $dx < rK$ in the exercise region. Hence, for both, call and put, options we the following inequality holds in the exercise regions.

$$\frac{\partial V}{\partial t} + \mathcal{L}V < 0. \tag{2.35}$$

Thus, in the entire semi strip $x > 0$, $t \in [0, T]$ the American option price has to fulfil an inequality of the Black–Scholes type

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V}{\partial x^2} + (r-d)x\frac{\partial V}{\partial x} - rV \leq 0. \tag{2.36}$$

Another inequality can be obtained from the assumption of an arbitrage free market. For the American option this means that its value always has to be greater or equal to the payoff value, that is,

$$V \geq \Phi. \tag{2.37}$$

Thus, if we combine all these arguments, we can get the following two cases

$$\text{if } V > \Phi, \text{ then the Black–Scholes } \textit{equation } \frac{\partial V}{\partial t} + \mathcal{L}V = 0$$

$$\text{if } V = \Phi, \text{ then the Black–Scholes } \textit{inequality } \frac{\partial V}{\partial t} + \mathcal{L}V < 0.$$

Formally this can be written as a *linear complimentarity problem* (LCP)

$$V - \Phi \geq 0, \tag{2.38}$$

$$\frac{\partial V}{\partial t} + \mathcal{L}V \leq 0, \tag{2.39}$$

$$(V - \Phi)\left(\frac{\partial V}{\partial t} + \mathcal{L}V\right) = 0. \tag{2.40}$$

2.8 Penalised formulation for the American option

Another way to convert the American option pricing problem into a fixed domain problem is to introduce a penalty term $P(V) \geq 0$. Then inequality (2.36) can be written as an equality in the entire semi strip $x > 0$, $t \in [0, T]$

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V}{\partial x^2} + (r-d)x\frac{\partial V}{\partial x} - rV + P(V) = 0. \tag{2.41}$$

The penalty function should ideally be zero (or very small) in the hold region and positive in the exercise region. When calculating an approximation of V, the free boundary location is not known, but the distance from the payoff $V - \Phi$ is available and often serves as a decision-making block of a penalty function. A common choice for the penalty term, which we also use in this thesis, is [22]

$$\frac{e\,C}{V + e - q} \tag{2.42}$$

where $e \to 0$, C is some constant and q is the barrier function, which is based on the payoff function Φ, for example for the put option $q = K - x$. It is easy to understand the role of the penalty term. When the solution V is distinctly above the payoff Φ, the penalty term $P(V)$ is negligible, and the Black–Scholes equation estimates the price reasonably well. When the solution V approaches the payoff Φ, the penalty term $P(V)$ grows and eventually dominates the Black–Scholes part of the equation, keeping the solution above the payoff. However, the introduction of a penalty term into the equation leads to an error that depends on the size of the penalty term e.

2.9 Overview of numerical methods

In general, there is a wide choice of numerical methods that are used in option pricing. Monte Carlo (MC) methods deal with the stochastic formulation (2.8). In the case of Monte Carlo methods, a large number N of trajectories following (2.8) is simulated (see Figure 2.3) until the maturity time and then the Feynman–Kac formulation (2.9) is applied to obtain a risk-neutral option price. The limitation of MC methods is that they usually have low convergence rates, although the convergence rate can in certain cases be increased from $\mathcal{O}(N^{-1/2})$ to $\mathcal{O}(N^{-1})$ by using quasi-random numbers. On the other hand, they scale well with the number of dimensions and are suitable for all underlying dynamics models. However, computations of Greeks (for some examples see (2.43)),

$$\Delta = \frac{\partial V}{\partial x}, \quad \mathcal{V} = \frac{\partial V}{\partial \sigma}, \quad \Gamma = \frac{\partial^2 V}{\partial x^2}. \tag{2.43}$$

which are quantities representing the sensitivity of the price of derivatives to a change in underlying parameters, can be complicated with these methods. Greeks are important parameters which are used for designing portfolio hedging strategies. Usually to determine their values a finite difference approximation based of two nearby values obtained by Monte Carlo simulations is used. Such an approach for computing Greeks is quite inefficient. For more details on Monte Carlo methods see [12].

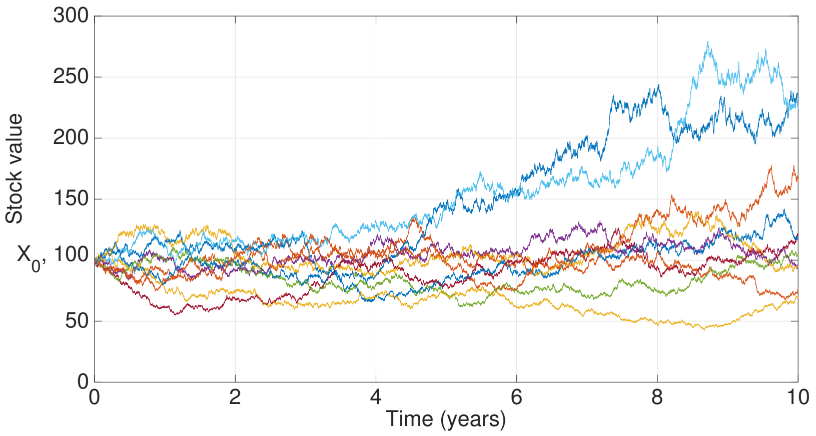


Figure 2.3: An example of a stock evolution following geometric Brownian motion.

A similar idea is used in binomial tree methods [4]. In the discrete time at the next time step the stock price can either go up with the probability p or fall down with the probability $1-p$ (see Figure 2.4). At the final time we get a set of simulated stock values to which the Feynman–Kac formulation is applied. Usually binomial tree methods experience the same problems as Monte Carlo methods.

Fourier expansion based methods [6, 17] are applied to the Feynman–Kac formulation (2.9) and perform extremely well, but the knowledge of the characteristic function of the underlying process is required, which is the case for standard dynamics, e.g., the geometric Brownian motion (2.2) or the Heston stochastic volatility model, but is not the case for more complex dynamics with, for example, time-space dependent volatilities. However, characteristic functions can be estimated numerically [23], but then the problem becomes nearly as expensive as to solve the original Black–Scholes partial differential equation using another method.

Grid methods, such as finite difference (FD) methods [29], finite element methods (FEM) [32], and mesh-free methods, such as radial basis function

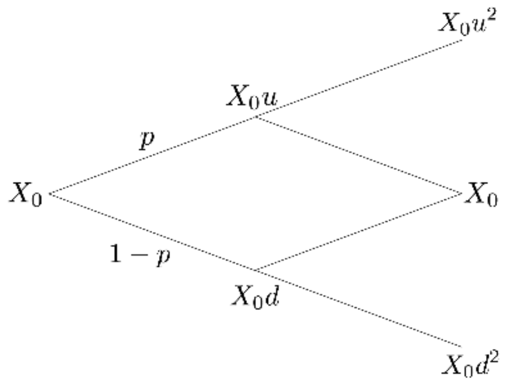


Figure 2.4: An example of a binomial tree, where the growth u is inversely proportional to the fall d, i.e., $u = 1/d$.

(RBF) methods [8], are used to solve the differential formulation (2.5)–(2.6), (2.38)–(2.40) or (2.41). For the PDE methods we obtain the following semi-discrete problem

$$\frac{dV}{dt} = -LV, \tag{2.44}$$

where L is the discrete representation of the spatial Black–Scholes operator $\mathcal{L}$.

An implicit finite difference scheme is usually used to discretise (2.44) in time. Explicit schemes can also be applied, but are not preferred, because the arising conditions on the time step length is quite severe, whilst the Black–Scholes equation has a strongly diffusive nature, and not many steps are needed to resolve the problem in time. An advantage of the PDE methods is that they allow to obtain option values for the entire range of asset values, and, moreover, automatically find estimates for sensitivities, but they become computationally too expensive already for moderately high-dimensional problems due to the “curse of dimensionality”.

The early exercise property in the American option pricing problem is addressed by several techniques. For methods which deal with the stochastic formulation (2.8) the Longstaff–Schwartz method [18] is used. The main idea is to estimate the payoff at the current time step and compare it with the future payoff, which is predicted by a simple regression algorithm, and then decide whether it is optimal to exercise the option now or it is worth to hold it longer.

Fourier expansion based methods employ Newton’s method to detect the free boundary location [7] and then solve the problem in two parts, one in the hold region and one in the exercise region. It turns out that only one of these subproblems (in the hold region) has to be solved and for the other one there exists an analytical representation.

Grid methods may exploit the operator splitting method [15], which is applied to formulation (2.38)–(2.40), to resolve the early exercise property. This approach splits the scheme in time into two substeps. In the first substep the Black–Scholes equation is solved and in the second substep the American constraint (2.37) is examined and if needed the solution is projected to the payoff function in order to avoid arbitrage. A penalised formulation of the Black–Scholes equation, as described in section 2.8, is also a common way to deal with the free boundary. The problem is stated in the entire domain, and the free boundary location does not play any role. Thus, numerical methods can be applied straightforwardly. However, it is important to notice that the penalised formulation is a nonlinear problem. Therefore the numerical approximation has to involve either nonlinear solvers or some special types of discretisation schemes with explicit treatment of the penalty term. As was mentioned previously, fully explicit schemes are not advantageous, therefore implicit-explicit schemes are a popular selection. Nevertheless, they still lead to some time constraint, which however is milder than in the fully explicit case. Simply ignoring the free boundary and applying the American constraint (2.37) at each time step is also possible [14].

In this thesis we aim to construct a method which is suitable for problems of low to moderately high dimensions. For this purpose we use the radial basis function partition of unity method (RBF–PUM), which is a localised version of the global RBF method. The locality of the partition of unity technique allows for a significant reduction of the computational effort compared with the global method. For the American option pricing problem we apply and compare both the operator splitting approach and the penalty approach, to handle the free boundary.

Chapter 3

Radial basis function methods

Here we give an introduction to the theory of RBF methods. Assume that we would like to approximate a function $u(x)$ inside a domain $\Omega \subset \mathbb{R}^d$, given N scattered nodes $\vec{x}_1, \ldots, \vec{x}_N \in \Omega$. The RBF approximation of the function $u(x)$ with the given values $u(\vec{x}_n), \ldots, u(\vec{x}_N)$ defined at the node points takes the form

$$\mathcal{J}_u(\vec{x}) = \sum_{j=1}^{N} \lambda_j \phi(\|\vec{x} - \vec{x}_j\|), \quad \vec{x} \in \Omega, \tag{3.1}$$

where λ_j are unknown coefficients, $\|\cdot\|$ is the Euclidian norm and $\phi(r)$ is a real-valued radial basis function, whose value depends only on the distance from its centerpoint. To determine coefficients λ_j we enforce the interpolation conditions

$$\mathcal{J}_u(\vec{x}_j) = u(\vec{x}_j), \quad j = 1, \ldots, N, \tag{3.2}$$

and as a result we obtain a linear system

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{u}, \tag{3.3}$$

where $A_{ij} = \phi(\|\vec{x}_i - \vec{x}_j\|)$, $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_N]^T$, $\mathbf{u} = [u(\vec{x}_1), \ldots, u(\vec{x}_N)]^T$. In order to distinguish between multi-dimensional objects, we denote d-dimensional objects by $\vec{y}$ and N-dimensional objects by $\mathbf{y}$.

If a time dependent function $u(t, \vec{x})$ is approximated then we let the coefficients λ_j be time-dependent. In this case, the approximant takes the form

$$\mathcal{J}_u(\vec{x}, t) = \sum_{j=1}^{N} \lambda_j(t) \phi(\|\vec{x} - \vec{x}_j\|), \quad \vec{x} \in \Omega, \ t \geq 0. \tag{3.4}$$

Table 3.1: *Examples of radial basis functions.*

RBF	$\phi(r)$
Multiquadric	$(1+(\varepsilon r)^2)^{1/2}$
Inverse multiquadric	$(1+(\varepsilon r)^2)^{-1/2}$
Inverse quadratic	$(1+(\varepsilon r)^2)^{-1}$
Gaussian	$e^{-(\varepsilon r)^2}$
Thin plate spline	$r^2\ln(r)$

Commonly used radial basis functions are presented in Table 4.1. We notice that most of the basis functions depend on the so-called shape parameter ε, which determines the width of the basis function. In Figure 3.1 the one-dimensional Gaussian and multiquadric basis functions of different widths are presented. If flat radial basis functions are used, then the matrix $\mathbf{A}$ is ill-conditioned, because the rows of the matrix $\mathbf{A}$ become linearly dependent. Curiously, the best approximation is often achieved when ε is small, which leads to flat basis functions. In order to solve this issue some stable methods, such as the Contour–Padé [11] or the RBF–QR method [9, 10], were designed.

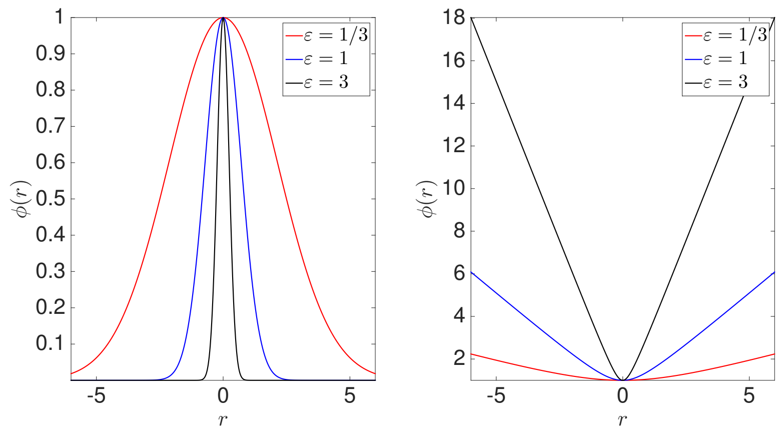


Figure 3.1: Left: The Gaussian radial basis function of different widths. Right: The multiquadric basis function of different widths.

For our experiments we use the multiquadric radial basis function, because it is, in general, less sensitive to changes of the shape parameter and, therefore, allows for more robust computations.

3.1 Radial basis function partition of unity method

A property of global RBF methods is that they exhibit high convergence rates. For smooth problems the convergence rate can be exponential if infinitely smooth radial basis functions are used [24]. However, the "price" for the high convergence rate is that the linear system (3.3) is dense, i.e., the matrix **A** consists of 100% non-zero elements. Taking into account that the problem size grows exponentially with the number of dimensions, we can conclude that solving the system (3.3) will require a large computational effort. Therefore, we apply a partition of unity technique [1] to introduce locality and sparsify the system. The main idea is to subdivide the domain Ω into M smaller overlapping subdomains Ω_i, which form an open cover of Ω (see Figure 3.2), that is,

$$\Omega \subseteq \bigcup_{i=1}^{M} \Omega_i. \tag{3.5}$$

Then we construct local RBF approximations inside each subdomain and combine them into a global approximant by the partition of unity functions $\{w_i\}_{i=1}^{M}$, subordinated to the open cover $\{\Omega_i\}_{i=1}^{M}$ of Ω. In mathematical terms it takes the form

$$\mathcal{J}_u(\vec{x}) = \sum_{i=1}^{M} w_i(\vec{x})\mathcal{J}_u^i(\vec{x}) = \sum_{i=1}^{M} w_i(\vec{x}) \sum_{j=1}^{N_i} \lambda_j^i \phi(\|\vec{x} - \vec{x}_j^i\|), \quad \vec{x} \in \Omega, \tag{3.6}$$

where $\mathcal{J}_u^i$ are local approximants constructed over N_i node points.

The partition of unity functions w_i can be constructed using Shepard's method [28] as follows:

$$w_i(\vec{x}) = \frac{\varphi_i(\vec{x})}{\sum_{k=1}^{M} \varphi_k(\vec{x})}, \quad i = 1, \ldots, M, \tag{3.7}$$

where $\varphi_i(\vec{x})$ is a function that is compactly supported on Ω_i, which, for example, can be a compactly supported Wendland's function [31]

$$\varphi(r) = \begin{cases} (1-r)^4(4r+1), & \text{if } 0 \le r \le 1, \\ 0, & \text{if } r > 1. \end{cases} \tag{3.8}$$

The linear system for RBF–PUM arising in standard financial applications, with the domain parameters that we used, is about 20 times more sparse than for the global RBF method, that is, only around 5% elements remain non-zero, while maintaining a similarly high accuracy. Furthermore, a partition based formulation is also well suited for parallel implementation.

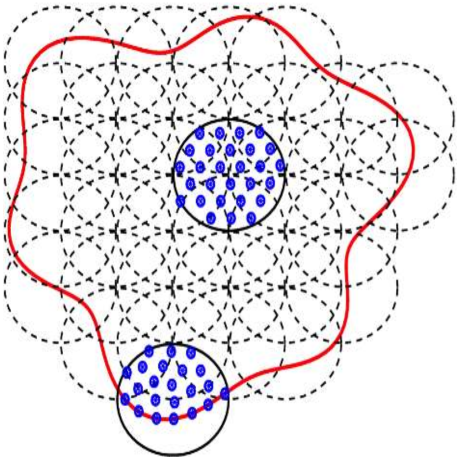


Figure 3.2: An example of a domain, scattered nodes, and a partitioning.

3.2 Convergence of RBF methods

As was previously mentioned, RBF methods exhibit good convergence properties. Although theoretical a priori error estimates were found for just a limited number of basis functions. For example an error estimate for the global RBF method, when Gaussian kernels are used, was derived in [24] under conditions that the problem is smooth and the solution lives in the native space generated by the kernels

$$\|E(t)\|_\infty \leq Ce^{\gamma \log(h)/h} \max_{0\leq\tau\leq t} \|u(\tau)\|_{\mathcal{N}(\Omega)}, \tag{3.9}$$

where h is the distance between nodes, γ is the exponential convergence order, and $\mathcal{N}(\Omega)$ is the native space generated by the radial basis function. In financial applications the initial condition is often only a C^0 function. Hence exponential approximation accuracy at the initial time is not possible as this requires smoothness of the solution [24]. However, due to the smoothing properties of parabolic problems, the solution can be approximated with high accuracy at later times [25]. It has been proved in [30], that solutions of parabolic problems with non-smooth initial condition can be approximated with optimal order when time is positive.

RBF–PUM inherits high convergence rates from the global method, and it can be shown for parabolic problems, particularly, for the convection-diffusion equation [25] that the convergence is exponential under node refinement, if the distance between partition centers H is fixed, i.e., the total number of partitions is kept fixed, and the convergence is algebraic under patch refinement, if the ratio H/h is fixed, i.e., the number of nodes points is adjusted such that each partition always contains the same number of node points. The error estimates, if inverse multiquadric basis functions are used, look as follows

$$\|E(t)\|_\infty \leq CH^{m-\frac{d}{2}-k} \max_{0\leq\tau\leq t} \max_i \|u(\tau)\|_{\mathcal{N}(\Omega_i)}, \tag{3.10}$$

$$\|E(t)\|_\infty \leq Ce^{-\mu/\sqrt{h}} \max_{0\leq\tau\leq t} \max_i \|u(\tau)\|_{\mathcal{N}(\Omega_i)}, \tag{3.11}$$

where m is the maximal polynomial degree which can be supported by the number of nodes located in each partition and determines the algebraic convergence order, k is the order of the partial differential equation, the constant μ defines the rate of exponential convergence, and $\mathcal{N}(\Omega_i)$ is the native space generated by the radial basis function.

Chapter 4

Summary of papers

Most of the attention in this thesis is dedicated to a design of an efficient numerical method for option pricing using the radial basis function partition of unity approach. Primarily, we focus on European- and American-style vanilla basket option contracts, because they have most of the interesting features, while being quite simple financial instruments. The American-style options are of a particular interest, because of the early exercise property, which turns the pricing problem into a free boundary problem. We test different techniques to handle the free boundary. Also a large comparison of various methods was conducted within BENCHOP — The BENCHmarking Project on Option Pricing, where our approach has been shown competitive, especially for higher-dimensional problems. A follow up work, where we improve a weak point of our approach, was done following the results of the BENCHOP project.

4.1 Paper I

V. Shcherbakov and E. Larsson. *Radial basis function partition of unity methods for pricing vanilla basket options.* Accepted for publication in Computers & Mathematics with applications.

In this paper we develop a radial basis function partition of unity method for pricing vanilla (plain European and American) multi-asset options. We numerically study convergence properties of the method with respect to the size of the shape parameter, to the number of partitions per dimension as well as to the number of node points per dimension. It turns out that for the European option case, we can observe a spectral converge rate despite of the discontinuity in the first derivative of the initial condition (see Figure 4.1). In order to price American multi-asset options we design a suitable penalty function. We find an error estimate for the continuous solution with respect

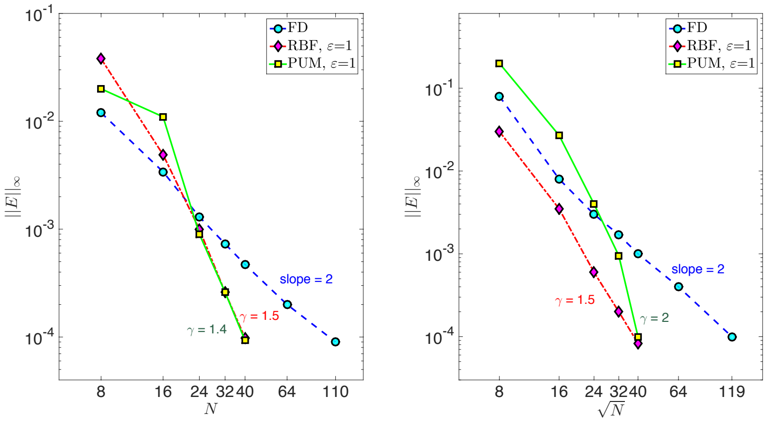


Figure 4.1: Left: Error convergence in l_∞-norm for a European option on one underlying asset. Right: Error convergence in l_∞-norm for a basket European option on two underlying assets. ε is the shape parameter.

to the penalty parameter. This directly reflects the dependence of the error on the size of the penalty parameter and justifies that the selected penalty function is of the correct form. We numerically study convergence of the solution against the size of the penalty parameter as well as the number of node points per dimension. It turns out that RBF–PUM achieves a locally high algebraic convergence rate. All error convergence and time efficiency results we compare to the results obtained by the central second order finite difference method, which is a standard method for such kind of problems. Our approach in most of the cases outperforms the standard finite difference method and its advantage become more evident in higher-dimensional cases.

4.2 Paper II

L. von Sydow, L.J. Höök, E. Lindström, S. Milovanović, J. Persson, V. Shcherbakov, Y. Shpolyanskiy, S. Sirén, J. Toivanen, J. Waldén, M. Wiktorsson, J. Levesley, J. Li, C. Oosterlee, M. Ruijter, A. Toropov, and Y. Zhao. *BENCHOP — The BENCHmarking project in Option Pricing.* International Journal of Computer Mathematics, 92(12), 2361–2379, 2015.

This was a large benchmarking project, where a number of methods commonly used for financial computations were tested on six benchmark problems (with quite many subproblems). RBF–PUM showed itself competitive in all test cases where it was applied, except the American-style option pri-

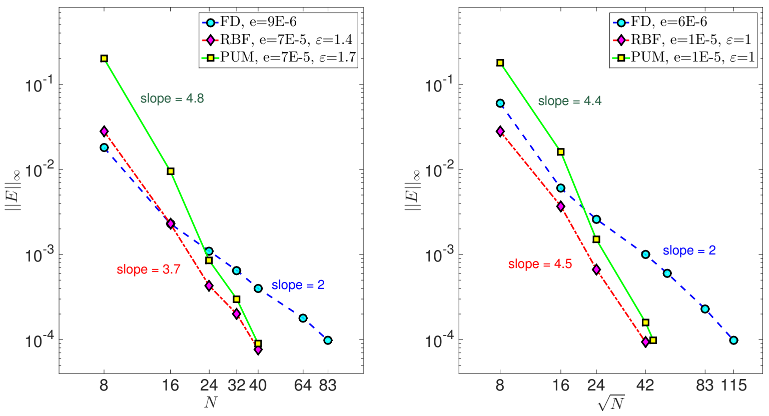


Figure 4.2: Left: Error convergence in l_∞-norm for an American option on one underlying asset. Right: Error convergence in l_∞-norm for a basket American option on two underlying assets. ε is the shape parameter, e is the penalty parameter.

cing problem, while for the two dimensional problems RBF–PUM was the best among the PDE methods. The reason for such a failure in pricing American options is that RBF–PUM used a penalty approach to resolve the free boundary issue, while most of the other methods used the operator splitting method. The penalty function used was nonlinear, and, therefore, in order to avoid nonlinear iterations we chose to treat it explicitly, while the rest of the spatial operator was treated implicitly. The use of the semi-implicit scheme imposed a condition on the time step, which resulted in a necessity to select a tiny time step to maintain stability. In the next paper we approach the free boundary problem with the operator splitting method, and this leads to a significant increase in the computational efficiency.

4.3 Paper III

V. Shcherbakov. *Radial basis function partition of unity operator splitting method for pricing American multi-asset options.* Submitted.

We correct our approach to the American option pricing problem and we extend the operator splitting formulation (OS) to the radial basis function partition of unity method. Also we implement a fully implicit method for the penalty approach (PIM), i.e., we involve nonlinear iterations, and it turns out that the Newton method requires just a few iterations to converge.

These two methods allow for a significant improvement in terms of computational efficiency. However, the operator splitting method outperforms the penalty method, and this advantage increases with the problem dimension. Some numerical results for a single-asset problem can be found in Table 4.1. Just for comparison we present the results of the semi-implicit (PIMEX), where the penalty function is defined explicitly, and fully explicit (PEX) implementations of the penalty method. For the double-asset problem the semi-implicit and the explicit penalty formulation become noncompetitive, therefore in Table 4.2 we display the results only for the implicit version of the penalty method and for the operator splitting method.

Table 4.1: *The American put option reference values V_r and the deviation of the approximate solution V from V_r for one underlying asset. The grid sizes and the execution times for the operator splitting method and for the three versions of the penalty method are also presented. The* * *denotes the reference solution, which was obtained by the Fourier Gauss Laguerre (FGL) method [17]. N denotes the number of node points in space, and N_t denotes the number of time steps.*

				Values	
Method	$N \times N_t$	Time (s)	$x = 90$	$x = 100$	$x = 110$
FGL*	-	-	10.726487	4.820608	1.828208
PEX	232 × 14000	1.5421	0.000515	0.000083	-0.000100
PIMEX	604 × 6500	0.4977	0.000729	0.000406	0.000177
PIM	74 × 75	0.0378	0.000601	0.000151	0.000126
OS	87 × 480	0.0249	0.000228	-0.000474	-0.000112

It is worth to notice that the number of grid points is different for the three penalty implementations. This is mostly due to the size of the penalty parameter combined with the focus on the time efficiency of each implementation. Let us consider PIM. This method has no constraint on the time step implied by the penalty size. Therefore we can select $e = 10^{-5}$, that is, the error introduced by the penalty approach will be an order of magnitude lower than the required tolerance. It simply means that we need 74 space nodes and 75 time step to have an accurate approximation. On the other hand, the efficiency of the PIMEX implementation suffers from the choices of small penalty parameters. The smaller the parameter is the smaller time steps we have to take. In the experiment for PIMEX we used $e = 10^{-4}$ and it led to 6500 time steps and 604 space points to suppress the error introduced by the penalty, which is roughly of the same size as the tolerance. If in this case we chose $e = 10^{-5}$ we would need around 74 space points (as in the PIM case), but then the constraint on the time step size would become even more severe and we would require around 150000 steps. This would negatively affect the overall computational time. A similar argument applies to the PEX case.

Table 4.2: *The American put option reference values V_r and the deviation of the approximate solution V from V_r for two underlying asset. The grid sizes and the execution times for the operator splitting method and for the implicit penalty method are also presented. The * denotes the reference solution, which was obtained by the Fourier–cosine series expansion [6]. N denotes the number of node points in space, and N_t denotes the number of time steps.*

				Values	
Method	$\sqrt{N} \times N_t$	Time (s)	$x = [90, 100]$	$x = [100, 100]$	$x = [100, 110]$
COS*	-	-	6.649395	4.051099	2.325955
PIM	73 × 22	62.9980	0.000173	0.000115	0.000105
OS	76 × 160	14.6647	0.000366	0.000420	-0.000054

Chapter 5

Conclusion and outlook

In this thesis we have developed the radial basis function partition of unity method for pricing multi-asset options with and without early exercise features. We have also compared a large number of methods that are widely used both in industry and in academia. Of course the main focus was on the comparison of RBF–PUM with finite difference methods, because these methods, roughly speaking, belong to the same class. In Paper I we clearly could see the advantage of using RBF–PUM instead of finite difference methods for pricing European options. Moreover, RBF–PUM achieves an exponential convergence rate, even though the initial condition has a discontinuous first derivative. Finite difference methods are able to achieve only a second order convergence rate, despite the order of a scheme, if no other special measures are taken, such as for example smoothing of the initial condition by a Rannacher step. In the case of American options, RBF–PUM should also be preferred over finite difference methods, if both use the penalty approach for handling the free boundary. We also extend the operator splitting formulation to RBF–PUM for pricing American multi-asset options. As observed in Paper III, this allows for a significant increase in computational efficiency since no nonlinear problem has to be solved, in contrast with the penalty formulation.

In future work we would like to further continue the development of RBF–PUM and to incorporate the least squares method instead of pure collocation. That is, we will have more evaluation points than node points. We expect that it will lead to problems with a smaller number of degrees of freedom, but more importantly, this will lead to more robust and efficient computations, since we reduce sensitivity to node point locations.

In order to accelerate computations even more, we plan to apply a new partition of unity approach compared with the way it is used now. Currently, given a domain we first scatter node points and then we choose an open cover of the domain and construct the partition of unity subordinated to the open cover. We would like to propose an approach, where given a domain we cover it by patches which include scattered nodes and partition of unity weights. That is, node points are scattered over the patches and then the patches cover the domain. Such an approach will allow us to precompute patches and local differentiation matrices in advance and save a significant amount of the computational work.

Mesh free methods facilitate the implementation of refinement strategies, where extra node points can easily be added without remeshing. Therefore another idea for the future work is to better exploit the advantage of RBF methods to work on a set of scattered node points and implement an adaptive RBF–PUM solver. In Papers II and III we used unstructured grids with node points clustered in the regions where it was needed the most, i.e., in the regions where a higher computational accuracy was required or in the regions where the numerical error was expected to be large and needed to be reduced by having more node points to allow for an appropriate order of approximation. In reality it is quite hard to predict where the highest error should be expected. Therefore, an adaptive solver can be designed. In the partition of unity frameworks the grid refinement could be done inside a separate patch or a group of neighbouring patches. Moreover, using precomputed patches and differentiation matrices would allow for just a simple substitution of elements in the global RBF matrix.

We expect that a combination of the two above mentioned ideas will result in an efficient tool for solving time-dependent partial differential equations. As a test case we are going to use a relevant financial problem, for example, computing transition densities for a two-factor interest rate model.

Bibliography

[1] I. Babuška and J. M. Melenk. The partition of unity method. *Internat. J. Numer. Methods Engrg.*, 40(4):727–758, 1997.

[2] T. Björk. *Arbitrage Theory in Continuous Time.* Oxford University Press, New York, 3 edition, 2009.

[3] F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. Polit. Econ.*, 81(3):637–654, 1973.

[4] S. Borovkova, F. Permana, and J. van der Weide. American basket and spread option pricing by a simple binomial tree. *J. Derivatives*, 19:29–38, 2012.

[5] P. Carr, H. Geman, D. B. Madan, and M. Yor. Stochastic volatility for Lévy processes. *Math. Finance*, 13(3):345–382, 2003.

[6] F. Fang and C. W. Oosterlee. A novel pricing method for European options based on Fourier-cosine series expansions. *SIAM J. Sci. Comput.*, 31(2):826–848, 2008.

[7] F. Fang and C. W. Oosterlee. Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. *Numer. Math.*, 114(1):27–62, 2009.

[8] G. E. Fasshauer, A. Q. M. Khaliq, and D. A. Voss. Using meshfree approximation for multi-asset American option problems. *J. Chinese Inst. Engrs.*, 27(4):563–571, 2004.

[9] B. Fornberg, E. Larsson, and N. Flyer. Stable computations with Gaussian radial basis functions. *SIAM J. Sci. Comput.*, 33(2):869–892, 2011.

[10] B. Fornberg and C. Piret. A stable algorithm for flat radial basis functions on a sphere. *SIAM J. Sci. Comput.*, 30(1):60–80, 2007.

[11] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Comput. Math. Appl.*, 48(5-6):853–867, 2004.

[12] P. Glasserman. *Monte Carlo methods in financial engineering.* Springer-Verlag, New York, 2004.

[13] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financ. Stud.*, 6(2):327–343, 1993.

[14] Y. C. Hon. A quasi-radial basis functions method for American options pricing. *Comput. Math. Appl.*, 43(3-5):513–524, 2002.

[15] S. Ikonen and J. Toivanen. Operator splitting methods for American option pricing. *Appl. Math. Lett.*, 17(7):809–814, 2004.

[16] F. Jamshidian. An analysis of American options. *Rev. Futures Markets*, 11(1):72–80, 1992.

[17] E. Lindström, J. Ströjby, M. Brodén, M. Wiktorsson, and J. Holst. Sequential calibration of options. *Comput. Statist. Data Anal.*, 52(6):2877–2891, 2008.

[18] F. A. Longstaff and E. S. Schwartz. Valuing American options by simulation: A simple least-squares approach. *Rev. Financ. Stud.*, 14(1):113–147, 2001.

[19] H. P. McKean. A free boundary problem for the heat equation arising from a problem in mathematical economics. *Ind. Manag. Rev.*, 6(2):32–39, 1965.

[20] R. C. Merton. Theory of rational option pricing. *Bell J. Econom. and Management Sci.*, 4:141–183, 1973.

[21] R. C. Merton. Option pricing when underlying stock returns are discontinuous. *J. Financ. Econ.*, 3(1):125 – 144, 1976.

[22] B. F. Nielsen, O. Skavhaug, and A. Tveito. Penalty and front-fixing methods for the numerical solution of American option problems. *J. Comput. Finance*, 5(4):69–97, 2002.

[23] S. Pagliarani, A. Pascucci, and R. Candia. Expansion formulae for local Lévy models. Technical Report 34571, Munich Personal RePEc Archive, 2011.

[24] C. Rieger and B. Zwicknagl. Sampling inequalities for infinitely smooth functions, with applications to interpolation and machine learning. *Adv. Comput. Math.*, 32(1):103–129, 2010.

[25] A. Safdari-Vaighani, A. Heryudono, and E. Larsson. A radial basis function partition of unity collocation method for convection-diffusion equations. *J. Sci. Comput.*, 64(2):341–367, 2015.

[26] P. A. Samuelson. Rational theory of warrant pricing. *Ind. Manag. Rev.*, 6(2):13–31, 1965.

[27] P. A. Samuelson and R. C. Merton. A complete model of warrant pricing that maximizes utility. *Ind. Manag. Rev.*, 10(2):17–46, 1969.

[28] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, ACM '68, pages 517–524, New York, NY, USA, 1968. ACM.

[29] D. Tavella and C. Randall. *Pricing Financial Instruments: The Finite Difference Method.* Wiley Series in Financial Engineering. Wiley, 2000.

[30] V. Thomée. *Galerkin finite element methods for parabolic problems*, volume 25 of *Springer Series in Computational Mathematics.* Springer-Verlag, Berlin, second edition, 2006.

[31] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.*, 4(4):389–396, 1995.

[32] P. Wilmott, J. Dewynne, and S. Howison. *Option Pricing: Mathematical Models and Computations.* Oxford Financial Press, Oxford, 1998.

Paper I

Radial basis function partition of unity methods for pricing vanilla basket options

Victor Shcherbakov[a,*], Elisabeth Larsson[a]

[a]*Uppsala University, Department of Information Technology, Box 337, SE-751 05 Uppsala, Sweden*

Abstract

Meshfree methods based on radial basis function (RBF) approximation are becoming widely used for solving PDE problems. They are flexible with respect to the problem geometry and highly accurate. A disadvantage of these methods is that the linear system to be solved becomes dense for globally supported RBFs. A remedy is to introduce localisation techniques such as partition of unity. RBF partition of unity methods (RBF–PUM) allow for a significant sparsification of the linear system and lower the computational effort. In this work we apply a global RBF method as well as RBF–PUM to problems in option pricing. We consider one- and two-dimensional vanilla options. In order to price American options we employ a penalty approach. A penalty term, suitable for American multi-asset call options, has been designed. RBF–PUM is shown to be competitive compared with a finite difference method and a global RBF method. It is as accurate as the global RBF method, but significantly faster. The results for RBF–PUM look promising for extension to higher-dimensional problems.

Keywords: radial basis function, partition of unity, option pricing, basket option, penalty method
2010 MSC: 65M70, 35K15

1. Introduction

Option contracts have been used for many centuries, but trading of options, as well as academic research on option pricing, increased dramatically in volume after 1973, when Black and Scholes published their market model [1]. Nowadays a variety of options are traded at the world exchanges, starting with simple vanilla options and continuing to multi-dimensional index options. Therefore, there is a high demand for correct option prices. Moreover, option prices play an important role in risk management, hedging, and parameter estimation.

*Corresponding author
Email addresses: `victor.shcherbakov@it.uu.se` (Victor Shcherbakov), `elisabeth.larsson@it.uu.se` (Elisabeth Larsson)

In this paper we consider the problem of pricing so called vanilla basket options, i.e., European and American options, with several underlying assets. A European option is a contract with a fixed exercise date, while an American option can be exercised at any time before maturity. Among the different available models of the underlying behaviour, such as the Heston model with stochastic volatility or the Merton model with jump diffusion, we select the standard Black-Scholes model, since it is a basic test case. Under the Black-Scholes model the price of European and American options can be determined by solving either a partial differential equation or a stochastic differential equation [2]. In the case of a single-asset European option the price is known analytically, while for multi-assets options the prices have to be computed numerically. The American option is more difficult due to the opportunity to exercise the option at any time. Such an opportunity introduces a free exercise boundary, which complicates the problem. The price for an American option needs to be computed numerically even in the single-asset case.

There are several techniques to handle the free exercise boundary. The most commonly used technique consists in rewriting the free boundary problem as a linear complementarity problem (LCP) and then solving it by a standard method, such as projected successive over-relaxation (PSOR) [3]. The drawback of this method is that it is relatively slow. Another method, that is used in industry, is the operator splitting (OS) method [4]. It is fast and effective for one-dimensional problems. Alternatively, a penalty approach can be taken as proposed in [5], and further developed in [6, 7, 8]. A penalty term designed to approximately enforce the early exercise condition is added to the PDE, which allows for removing the free boundary and solving the problem on a fixed domain. In combination with radial basis function (RBF) methods, variations of the penalty approach have been popular for handling American options, see [9, 10, 11, 12, 13]. It is also possible to in which in each time step ignore the free boundary and then apply the American constraint explicitly. This has been done for RBF methods in [14, 15, 16]. In this paper, we evaluate the performance of the penalty approach in the RBF setting with respect to accuracy and computational cost.

There are various numerical methods, which are used for option pricing in industry as well as in academia. Perhaps the most popular methods are Monte Carlo (MC) methods [17] and finite difference (FD) methods [3]. Both of them have their own strengths and weaknesses. MC methods converge slowly but are effective for pricing high-dimensional options, because the computational cost scales linearly with the number of underlying assets. On the other hand, FD methods have a better convergence rate, while the computational cost grows exponentially with the number of underlying assets. Other types of methods that are used are binomial tree methods [18] and Fourier expansion based methods [19].

We aim to construct a method for option pricing, based on radial basis function approximation, that can be competitive for low-dimensional to moderately high-dimensional problems. RBF methods can achieve high order algebraic, or for some problems even exponential, convergence rates [12, 20]. It means that in order to get the same accuracy the problem size will be smaller than with FD, which is crucial if we work in a many-dimensional space. A global RBF method was shown to compare favourably with an adaptive FD method in [21] in one and two dimensions.

Another advantage of RBF methods is that they are meshfree and therefore can accommodate non-trivial geometries. In financial applications, the computational domains that are used in the literature are often regular. For example, squares, cubes or hypercubes can easily be used. However, depending on the nature of the contract function, using another shape of the domain can lead to substantial computational savings, see, e.g., [21], where a simplex domain is used instead. Furthermore, with a meshfree method, the discretisation can easily be adapted to resolve local features in the solution.

A drawback of global RBF methods is that the linear system that needs to be solved is dense and often ill-conditioned. The situation can be improved by introducing localisation techniques, see e.g., [22, 23, 24]. One way to introduce locality is to employ a partition of unity framework, which was proposed by Babuška and Melenk in 1997 [25]. A partition based formulation is also well suited for parallel implementation. Some work on parallelisation for localised RBF methods has been done, see for example [23, 26, 27]. The ill-conditioning can be addressed by, for example, the RBF–QR technique [28, 29, 30].

In this paper we consider the problem of pricing dividend paying vanilla basket call options. In order to solve the problem we use global RBF and RBF partition of unity methods (RBF–PUM). We show that RBF based methods provide a good alternative to already existing methods. All comparisons of the solutions are made against a standard FD solution for European options an FD–OS solution for American options.

The outline of the paper is as follows. In Section 2, we introduce the Black-Scholes model for European and American basket call options. In Section 3, we discuss the penalty approach for American options and its form in the case of call options. Then in Section 4, we give an overview of RBF methods and RBF–PUM. Section 5 contains numerical experiments and comparisons. Finally, Section 6 concludes the paper.

2. The Black-Scholes model

The multi-dimensional Black-Scholes equation takes the form

$$\frac{\partial V}{\partial t} = \mathcal{L}V, \quad \mathbf{x} \in \Omega,\, t \in (0, T]\,, \tag{2.1}$$

where V is the value of the option, $\mathbf{x} = (x_1, \ldots, x_d)$ defines the spot prices of the d underlying assets, Ω is the domain of definition, t is the backward time, i.e., time to maturity, and T is the maturity time of the option. The spatial operator $\mathcal{L}$ takes the form

$$\mathcal{L} = \frac{1}{2}\sum_{i,j=1}^{d} \Sigma_{ij} x_i x_j \frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i=1}^{d}(r - D_i)x_i \frac{\partial}{\partial x_i} - r, \tag{2.2}$$

where D_i is the continuous dividend yield paid out by the ith asset, the matrix $\Sigma = [\sigma\sigma^*]$, where σ is the volatility matrix, and r is the risk-free interest rate.

The payoff function for the call option is given by:

$$\Phi(\mathbf{x}) = \max\left(\sum_{i=1}^{d} \alpha_i x_i - K, 0\right), \tag{2.3}$$

where K is the strike price and α_i is the weight of the ith asset in the portfolio. This is the value of the option at the time of maturity, but since we use backward time the initial condition becomes

$$V(\mathbf{x}, 0) = \Phi(\mathbf{x}), \quad \mathbf{x} \in \Omega. \tag{2.4}$$

2.1. The European case

In the case of the European option $\Omega = \Omega_E = \mathbb{R}_+^d$, but in order to enable numerical simulations we truncate $\mathbb{R}_+^d$ sufficiently far away from the origin that asymptotical results hold to high accuracy. We denote the truncated domain by $\hat{\Omega}_E$ and the far-field (truncation) boundary is given by $\Gamma^F = \bigcup_{i=1}^d \Gamma_i^F$, where $\Gamma_i^F = \{\mathbf{x} \mid \mathbf{x} \in \Omega_E,\ x_i = x_\infty\}$. The near-field boundary can be seen as the single point $\mathbf{x} = 0$, and there we have the condition

$$V(0, t) = 0, \quad t \in [0, T], \tag{2.5}$$

and at the far-field boundary we use the asymptotic condition

$$V(\mathbf{x}, t) = \sum_{i=1}^{d} \alpha_i x_i e^{-D_i t} - K e^{-rt}, \quad \mathbf{x} \in \Gamma^F,\, t \in [0, T]. \tag{2.6}$$

At the boundaries $\Gamma_i = \{\mathbf{x} \,|\, \mathbf{x} \in \Omega_E, \mathbf{x} \neq 0, x_i = 0\}$, the spatial operator (2.2) is degenerate and reduces to a $(d-1)$-dimensional operator. Fichera [31] derived general conditions for when to impose boundary conditions for parabolic PDEs with degenerate diffusion operators (see also the Feller condition [32]). In the case of the Black–Scholes operator, boundary conditions should not be imposed at Γ_i unless required for numerical purposes.

2.2. The American case

In the case of the American option, $\Omega = \Omega_A$ is a subdomain of $\mathbb{R}_+^d$, which falls inside the free early exercise boundary $\Gamma(\mathbf{x}, t)$. We use the same near-field boundary condition as for the European option

$$V(0, t) = 0, \quad t \in [0, T]. \tag{2.7}$$

At the free boundary we have

$$V(\mathbf{x}, t) = \Phi(\mathbf{x}), \quad \mathbf{x} \in \Gamma(\mathbf{x}, t),\, t \in [0, T], \tag{2.8}$$

$$\frac{\partial V}{\partial x_i}(\mathbf{x}, t) = \alpha_i, \qquad \mathbf{x} \in \Gamma(\mathbf{x}, t),\, t \in [0, T]. \tag{2.9}$$

Outside the free boundary the solution is given by $V(\mathbf{x}, t) = \Phi(\mathbf{x})$.

3. The penalty method

Penalty methods can be used for solving boundary value problems. An early reference to the penalty method appears in 1943 in Courant's work on motion in a bounded domain [33]. In relation to option pricing, the penalty method was introduced by Zvan et al. in [5], where a penalty approach for a Black-Scholes model with stochastic volatility for American options is discussed. Then, Nielsen et al. [7] proposed a new form of the penalty term for American put options, which has subsequently been used by several authors, combined with finite differences [34] and radial basis functions [9, 10, 11, 12, 13].

In this paper we consider a penalty method for pricing American basket call options. In the case of call options, dividends must be present, otherwise the American call is equivalent to the European call [35], while in the case of put options dividends may be zero. Hence, we propose a penalty term for the American basket option with dividends,

$$P = \frac{e\left(rK - \sum_{i=1}^{d} \alpha_i D_i x_i\right)}{V + e - q}, \tag{3.1}$$

where e is the penalty parameter, which has to be chosen sufficiently small, and $q(\mathbf{x})$ is the barrier function, which is the non-zero part of the payoff function,

$$q(\mathbf{x}) = \sum_{i=1}^{d} \alpha_i x_i - K. \tag{3.2}$$

Adding the penalty term to the Black-Scholes equation allows us to convert the free boundary problem to a fixed domain problem. The error introduced by the penalty is expected to be $\mathcal{O}(e)$. The modified equation takes the form

$$\frac{\partial V}{\partial t} = \mathcal{L}V - P(V), \quad \mathbf{x} \in \hat{\Omega}_E,\, t \in (0, T], \tag{3.3}$$

where $\hat{\Omega}_E$ is the same domain as for the European option, since the free boundary has been removed and the modified problem is defined on the entire extended domain $\hat{\Omega}_E$. The equation is subject to the following initial and boundary conditions

$$V(\mathbf{x}, 0) = \Phi(\mathbf{x}), \quad \mathbf{x} \in \hat{\Omega}_E, \tag{3.4}$$

$$V(0, t) = 0, \quad t \in [0, T], \tag{3.5}$$

$$V(\mathbf{x}, t) = \Phi(\mathbf{x}), \quad \mathbf{x} \in \Gamma^F,\, t \in [0, T]. \tag{3.6}$$

3.1. Substantiation of the form of the penalty term

In this subsection, we will show why our choice of the form of the penalty term for the American basket call option is motivated.

The value of an American call option must be larger or equal to the payoff value in order to exclude all arbitrage opportunities. The penalty function is designed in a way that it is negligible (of order e) when the solution is away from the barrier q, but it increases and penalises when the solution approaches the barrier. Now we want to show that the solution of such a penalised equation does not fall below the payoff and does not permit arbitrage opportunities. Moreover the solution will stick to the payoff after crossing the free boundary, that is, it will mimic the behaviour of the true solution. We consider the single-asset case. Equation (3.3) takes form

$$\frac{\partial V}{\partial t} = \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V}{\partial x^2} + (r-D)x\frac{\partial V}{\partial x} - rV - \frac{e\,(rK - Dx)}{V + e - q}. \tag{3.7}$$

We assume that the solution V is close to the payoff function, i.e., $V \approx x - K$, or we can write it as $V = x - K + \delta$, for some $0 < \delta < e$. Inserting this representation of V into the right part of (3.7) we obtain

$$\frac{\partial V}{\partial t} = (r-D)x - r(x - K + \delta) - \frac{e(rK - Dx)}{x - K + \delta + e - (x - K)}. \tag{3.8}$$

We reorganise the terms

$$(e+\delta)\frac{\partial V}{\partial t} = -Dx\delta + rK\delta - r\delta^2 - re\delta \tag{3.9}$$

and use the fact that $Dx > rK$ when x is above the free boundary [35], thus we get

$$(e+\delta)\frac{\partial V}{\partial t} = -(Dx - rK)\delta - r\delta^2 - re\delta < 0, \tag{3.10}$$

which as $e + \delta > 0$ implies $\frac{\partial V}{\partial t} < 0$. That is, positive perturbations are quickly recovered and the solution is pulled down to the payoff. Now doing a similar analysis we show that the solution is not able to fall below the payoff. We assume that V experiences negative perturbations, i.e., $V = x - K - \delta$, for some $0 < \delta < e$. Inserting this form of V into the right hand side of (3.7) we obtain

$$\frac{\partial V}{\partial t} = (r-D)x - r(x - K - \delta) - \frac{e(rK - Dx)}{x - K - \delta + e - (x - K)}. \tag{3.11}$$

Rearranging the summands and using the same fact that $Dx > rK$ when x is above the free boundary we get

$$(e-\delta)\frac{\partial V}{\partial t} = (Dx - rK)\delta + r\delta(e - \delta) > 0. \tag{3.12}$$

This as well shows that negative perturbations are immediately recovered with time, and therefore the solution of the penalised equation with such a choice of a penalty function is not allowed to fall below the payoff, that is, $V \geq q$ after the free boundary. However, going to (3.10) and (3.12) we see that for the region where $Dx - rK < 0$ a solution that comes close to q is repelled from q, meaning that whether the solution is repelled from or attracted to q, it will never cross the barrier function q, and thus as the initial condition is above q, we conclude that $V \geq q$ in the entire domain. We use this result for deriving an energy estimate of the error.

Now we are going to derive an estimate for the error $\eta(x,t) = V(x,t) - V_a(x,t)$, where $V(x,t)$ is the solution of the one-dimensional penalised equation (3.7) and $V_a(x,t)$ is the analytical solution of the Black–Scholes which is prolonged by the payoff after the free boundary. We know that $V_a(x,t)$ satisfies the homogeneous Black–Scholes equation in Ω_A, and if we plug in the payoff function into the Black–Scholes equation we will see that it will satisfy the non-homogeneous Black-Scholes equation in $\hat{\Omega}_E \backslash \Omega_A$ with the right hand side $f(x,t) = Dx - rK$. Thus in the entire $\hat{\Omega}_E$ we can write that V_a is the solution of

$$\begin{cases} V_t = \frac{1}{2}\sigma^2 x^2 V_{xx} + (r-D)xV_x - rV, & \text{if } x \in \Omega_A, \\ V_t = \frac{1}{2}\sigma^2 x^2 V_{xx} + (r-D)xV_x - rV + f, & \text{if } x \in \hat{\Omega}_E \backslash \Omega_A, \end{cases} \tag{3.13}$$

and V_a has a continuous first derivative in $\hat{\Omega}_E$ due to the smooth pasting condition for the American option (2.9).

The error fulfils the following differential equation

$$\begin{cases} \eta_t = \frac{1}{2}\sigma^2 x^2 \eta_{xx} + (r-D)x\eta_x - r\eta - \frac{e(rK-Dx)}{V_a+\eta+e-q}, & \text{if } x \in \Omega_A, \\ \eta_t = \frac{1}{2}\sigma^2 x^2 \eta_{xx} + (r-D)x\eta_x - r\eta - \frac{e(rK-Dx)}{V_a+\eta+e-q} + f, & \text{if } x \in \hat{\Omega}_E \backslash \Omega_A, \end{cases} \tag{3.14}$$

subject to initial and boundary conditions

$$\eta(x,0) = 0, \quad x \in \hat{\Omega}_E, \tag{3.15}$$

$$\eta(0,t) = 0, \quad t \in [0,T], \tag{3.16}$$

$$\eta(x,t) = 0, \quad x \in \Gamma^F,\ t \in [0,T], \tag{3.17}$$

and it is a $C^1(\hat{\Omega}_E)$-function due to the smoothness properties of the analytical and approximate solutions (V_a and V).

Thus, now we can write out an energy estimate for the error in $L_2(\hat{\Omega}_E)$-norm taking into account that in the weak form we are able to combine the two domains despite of the discontinuity in the second derivative of V_a

$$\frac{d}{dt}||\eta||^2 = 2(\eta_t, \eta) =$$
$$(\sigma^2 x^2 \eta_{xx}, \eta) + 2\left((r-D)x\eta_x, \eta\right) - 2(r\eta, \eta) - 2(F, \eta) - 2\left(\frac{e(rK-Dx)}{V_a+\eta+e-q}, \eta\right), \tag{3.18}$$

where

$$F = \begin{cases} 0, & \text{if } x \in \Omega_A, \\ f, & \text{if } x \in \hat{\Omega}_E \backslash \Omega_A. \end{cases} \tag{3.19}$$

We take a closer look at each term individually.

1. Integrating the first summand by parts gives us

$$(\sigma^2 x^2 \eta_{xx}, \eta) = -\sigma^2(2x\eta_x, \eta) - \sigma^2(x\eta_x, x\eta_x) \le -2\sigma^2(x\eta_x, \eta). \tag{3.20}$$

If we then integrate $2(x\eta_x, \eta)$ by parts we see that

$$2(x\eta_x, \eta) = (\eta, \eta), \tag{3.21}$$

Therefore,

$$(\sigma^2 x^2 \eta_{xx}, \eta) \le \sigma^2 ||\eta||^2. \tag{3.22}$$

2. Using (3.21) the second summand becomes

$$2\left((r-D)x\eta_x,\eta\right) = -(r-D)||\eta||^2. \tag{3.23}$$

3. The third term

$$-2(r\eta,\eta) = -2r||\eta||^2. \tag{3.24}$$

4+5. The last two terms we consider in two parts: before and after the free boundary, which for the one-dimensional case we denote as x^*.

   (a) Before the free boundary we have

$$\begin{aligned} Q_1 = -2\int_0^{x^*} F\eta\,dx - 2\int_0^{x^*} \frac{e\,(rK-Dx)}{V_a+\eta+e-q}\,\eta\,dx = \\ -2\int_0^{x^*} 0\cdot\eta\,dx + 2\int_0^{x^*} \frac{e\,(Dx-rK)}{V_a+\eta+e-q}\,(V-V_a)\,dx \le \\ 2\int_0^{x^*} \frac{|e\,(Dx-rK)\,|}{|V_a+\eta+e-q|}|V-V_a|\,dx \end{aligned} \tag{3.25}$$

   Since $V_a \ge q$, $V \ge q$, we have that $|V-V_a| \le |V-q|$. Also using that $V_a+\eta-q = V-q \ge 0$ and $e \ge 0$ then we obtain

$$Q_1 \le 2\int_0^{x^*} \frac{|e\,(Dx-rK)\,|}{|V-q|}|V-q|\,dx. \tag{3.26}$$

   Extending the integration region to $\hat{\Omega}_E$ we have

$$Q_1 \le 2e||(Dx-rK)|| \le 2e||Dx-rK||_\infty \le 2e(Dx_\infty - rK). \tag{3.27}$$

   (b) After the free boundary we have

$$\begin{aligned} Q_2 = -2\int_{x^*}^{x_\infty} F\eta\,dx - 2\int_{x^*}^{x_\infty} \frac{e\,(rK-Dx)}{V_a+\eta+e-q}\,\eta\,dx = \\ -2\int_{x^*}^{x_\infty} (Dx-rK)\eta\,dx + 2\int_{x^*}^{x_\infty} \frac{e\,(Dx-rK)}{V_a+\eta+e-q}\,\eta\,dx. \end{aligned} \tag{3.28}$$

   As $V \ge q$ and $V_a = q$ in this region we know that $\eta \ge 0$. Furthermore, $Dx-rK \ge 0$. Therefore, we can write

$$Q_2 = -2\int_{x^*}^{x_\infty} |(Dx-rK)||\eta|\,dx + 2\int_{x^*}^{x_\infty} \left|\frac{e\,(Dx-rK)}{V_a+\eta+e-q}\right| |\eta|\,dx. \tag{3.29}$$

   Since $V_a = q$ and $\eta \ge 0$

$$\frac{e}{V_a+\eta+e-q} \le 1. \tag{3.30}$$

   Applying this fact to (3.29) we obtain that

$$Q_2 \le -2\int_{x^*}^{x_\infty} |(Dx-rK)||\eta|\,dx + 2\int_{x^*}^{x_\infty} |(Dx-rK)||\eta|\,dx = 0. \tag{3.31}$$

Thus, summing up all the terms we get an estimate

$$\frac{d}{dt}||\eta||^2 \leq (\sigma^2 - 3r + D)||\eta||^2 + 2(Dx_\infty - rK)e, \tag{3.32}$$

or after integration taking into account that $||\eta(0)|| = 0$

$$||\eta||^2 \leq \begin{cases} \frac{\mu e}{\nu}(\exp(\nu t) - 1), & \text{if } \nu \neq 0, \\ 2\mu e t, & \text{if } \nu = 0, \end{cases} \tag{3.33}$$

where $\nu = \sigma^2 - 3r + D$ and $\mu = 2(Dx_\infty - rK)$. We have a dependence of the error on the penalty parameter size. That is, as $e \to 0$ the solution of the penalised problem will converge to the solution of the original problem. This dependence is investigated numerically in section 6.4. A similar analysis could be done in the multi-dimensional case, but an estimate in line with the result in [35] would be needed.

4. Radial basis function methods

RBF methods are meshfree and based on scattered nodes, therefore they are very flexible in terms of the geometry of the computational domain. Given N scattered nodes $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \Omega \subset \mathbb{R}^d$, the RBF interpolant of a function with values $u(\mathbf{x}_1), \ldots, u(\mathbf{x}_N)$ defined at those points takes the form

$$\mathcal{J}_u(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|), \quad \mathbf{x} \in \Omega, \tag{4.1}$$

where λ_j is an unknown coefficient, $\|\cdot\|$ is the Euclidean norm and $\phi(r)$ is a real-valued radial basis function, such as the Gaussian $\phi(r) = e^{-(\varepsilon r)^2}$ or the multiquadric $\phi(r) = \sqrt{1 + (\varepsilon r)^2}$, which we use for our numerical experiments. In order to determine λ_j, $j = 1, \ldots, N$, we enforce the interpolation conditions $\mathcal{J}_u(\mathbf{x}_j) = u(\mathbf{x}_j)$ and as a result we obtain a linear system

$$A\bar{\lambda} = \bar{u}, \tag{4.2}$$

where $A_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$, $\bar{\lambda} = [\lambda_1, \ldots, \lambda_N]^T$, $\bar{u} = [u(\mathbf{x}_1), \ldots, u(\mathbf{x}_N)]^T$.

When we approximate a time dependent function $u(\mathbf{x}, t)$, we let λ_j be time-dependent, such that

$$\mathcal{J}_u(\mathbf{x}, t) = \sum_{j=1}^{N} \lambda_j(t) \phi(\|\mathbf{x} - \mathbf{x}_j\|), \quad \mathbf{x} \in \Omega, t \geq 0. \tag{4.3}$$

4.1. RBF partition of unity methods

In spite of the many advantages of RBF methods, there is one computationally expensive disadvantage. The interpolation matrix A becomes dense when globally supported RBFs are used. Employing a partition of unity method (PUM) is one way to introduce locality and sparsity. A collocation RBF–PUM is introduced in the forthcoming paper [22] for elliptic PDEs, and applied to option pricing problems in [12]. The main idea is to subdivide a larger domain into smaller overlapping subdomains. Then a local RBF approximation is used within each subdomain. Local approximations in neighbouring subdomains are coupled, but the overall matrix structure is sparse and the computational complexity is reduced. Furthermore, there is an opportunity for parallel implementation.

We define a partition of unity $\{w_i\}_{i=1}^M$, subordinated to the open cover $\{\Omega_i\}_{i=1}^M$ of Ω, i.e., $\Omega \subseteq \bigcup_{i=1}^M \Omega_i$, such that

$$\sum_{i=1}^{M} w_i(\mathbf{x}) = 1, \quad \mathbf{x} \in \Omega. \tag{4.4}$$

Now, for each subdomain we construct a local RBF interpolant $\mathcal{J}_u^i$, and then form the global interpolant for the entire domain Ω:

$$\mathcal{J}_u(\mathbf{x}) = \sum_{i=1}^{M} w_i(\mathbf{x})\mathcal{J}_u^i(\mathbf{x}) = \sum_{i=1}^{M} w_i(\mathbf{x}) \sum_{j=1}^{N_i} \lambda_j^i \phi(\|\mathbf{x} - \mathbf{x}_j^i\|), \quad \mathbf{x} \in \Omega. \tag{4.5}$$

The partition of unity functions w_i can be constructed using Shepard's method [36] as follows:

$$w_i(\mathbf{x}) = \frac{\varphi_i(\mathbf{x})}{\sum_{k=1}^{M} \varphi_k(\mathbf{x})}, \quad i = 1, \ldots, M, \tag{4.6}$$

where $\varphi_i(\mathbf{x})$ is a function that is compactly supported on Ω_i, which we choose to be a C^2 compactly supported Wendland function [37]

$$\varphi(r) = \begin{cases} (1-r)^4(4r+1), & \text{if } 0 \le r \le 1 \\ 0, & \text{if } r > 1. \end{cases} \tag{4.7}$$

The elements Ω_i of the open cover of Ω will be chosen as circular patches. Therefore, the Wendland functions will be scaled to get

$$\varphi_i(\mathbf{x}) = \varphi\left(\frac{\|\mathbf{x} - \mathbf{c}_i\|}{r_i}\right), \quad i = 1, \ldots, M, \tag{4.8}$$

where r_i is the radius of the patch Ω_i and $\mathbf{c}_i$ is its centre point.

5. Time discretisation and space approximations

When we solve the option pricing problem numerically, we collocate the different RBF approximations in space as described in Sections 5.2 and 5.3. In time we use a standard ODE solver. We define the discrete times t^n, $n = 0, \ldots, N_t$ and denote the approximate solution at time t^n by $V^n(\mathbf{x}) \approx V(t^n, \mathbf{x})$.

5.1. The BDF-2 time stepping scheme

For the time discretisation we choose the second order backward differentiation formula (BDF-2). That is, for the European option the time discretisation is entirely implicit. A fully implicit time discretisation for the American option will lead to unconditional stability, but we will need to solve a nonlinear system of equations at each time step, and the total computational cost may become high. Another option is to use either an explicit scheme or a semi-implicit scheme with the penalty term evaluated explicitly at the middle time level, see equations (5.1–5.2). We have chosen to use the semi-implicit scheme. We show the discretisation for the American option only, as the scheme for the European option is identical, except for the presence of the penalty term.

We divide the time interval $[0, T]$ into N_t steps of length $k^n = t^n - t^{n-1}$, $n = 1, \ldots, N_t$. The BDF-2 scheme has the form [38, p. 401]

$$\begin{aligned} (E - \beta_0^n \mathcal{L})\, V_I^1 &= V_I^0, && (5.1)\\ (E - \beta_0^n \mathcal{L})\, V_I^n &= \beta_1^n V_I^{n-1} - \beta_2^n V_I^{n-2} - \beta_0^n P(V_I^{n-1}), \quad n = 2, \ldots, N_t, && (5.2) \end{aligned}$$

where V_I^n is the solution in the interior, E is an identity operator and

$$\beta_0^n = k^n \frac{1+\omega_n}{1+2\omega_n}, \quad \beta_1^n = \frac{(1+\omega_n)^2}{1+2\omega_n}, \quad \beta_2^n = \frac{\omega_n^2}{1+2\omega_n}, \tag{5.3}$$

where $\omega_n = k^n/k^{n-1}$, $n = 2, \ldots, N_t$. In [39] it is shown how the time steps can be chosen in such a way that $\beta_0^n \equiv \beta_0$. Then the coefficient matrix is the same in all time steps and only one matrix factorisation is needed.

The boundary conditions are enforced at each new time level through

$$V_B^n = f_B^n, \quad n = 1, \ldots, N_t. \tag{5.4}$$

The semi-implicit scheme will put a restriction on the time step size of the following form:

$$\Delta t \le \frac{Ce}{\left| rK - \sum_{i=1}^d \alpha_i D_i x_{i,\infty} \right|}, \tag{5.5}$$

where $\Delta t = \max\{k^n\}_{n=1}^{N_t}$, $x_{i,\infty}$ is the point, at which we truncate the domain in the direction of i-th asset and C is some constant. This condition is obtained empirically, but performing a simple linearisation of the penalty term and some heuristic calculations we can obtain a similar result with $C = k^n/\beta_0^n = 3/2$ for the BDF-2 scheme on a uniform time grid. This aligns with the condition $\Delta t \le \frac{e}{rK}$, which can be found in [7] for the case when finite differences are used to price an American put without dividends. Condition (5.5) does not depend on the grid, therefore for some choices of e it is less severe than the condition imposed by the explicit scheme.

In section 6.4 we will see that condition (5.5) holds numerically with an observed constant that is larger than 3/2.

5.2. Approximation in space using RBF

When using a collocation approach, we work with the nodal solution values $v_j^n = V_\varepsilon^n(\mathbf{x}_j) \approx V(t_n, \mathbf{x}_j)$. We build the approximation at time t_n according to (4.3)

$$V_\varepsilon^n(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j^n \phi(\varepsilon \|\mathbf{x} - \mathbf{x}_j\|). \tag{5.6}$$

The nodal values v_j^n and the coefficients λ_j^n fulfil the following relation:

$$A\bar{\lambda}^n = \bar{v}^n, \tag{5.7}$$

where the interpolation matrix A has elements $a_{pq} = \phi(\varepsilon\|\mathbf{x}_p - \mathbf{x}_q\|)$ and

$$\bar{\lambda}^n = [\lambda_1^n, \ldots, \lambda_N^n]^T, \quad \bar{v}^n = [v_1^n, \ldots, v_N^n]^T.$$

For RBFs such as Gaussians, multiquadrics, and inverse multiquadrics, A is non-singular as long as the node points are distinct. Hence, we can invert the relation to get

$$\bar{\lambda}^n = A^{-1}\bar{v}^n. \tag{5.8}$$

This allows us to construct differentiation matrices to evaluate derivatives of the RBF approximation in terms of the nodal values

$$\frac{\partial \bar{v}^n}{\partial x_k} = A^{(k)}\bar{\lambda}^n = A^{(k)}A^{-1}\bar{v}^n, \quad \frac{\partial^2 \bar{v}^n}{\partial x_k \partial x_m} = A^{(km)}\bar{\lambda}^n = A^{(km)}A^{-1}\bar{v}^n, \tag{5.9}$$

where $A^{(k)}$ and $A^{(km)}$ are matrices of derivatives of radial basis functions with elements $a_{pq}^{(k)} = \phi'_{x_k}(\varepsilon\|\mathbf{x}_p - \mathbf{x}_q\|)$ and $a_{pq}^{(km)} = \phi''_{x_k x_m}(\varepsilon\|\mathbf{x}_p - \mathbf{x}_q\|)$ respectively.

Thus,

$$L\bar{v}^n = \left[\frac{1}{2}\sum_{k,m=1}^{d} \Sigma_{km} x_k x_m A^{(km)} + \sum_{k=1}^{d}(r - D_k)x_k A^{(k)} - rA\right] A^{-1}\bar{v}^n, \tag{5.10}$$

where L is a matrix representation of the spatial operator $\mathcal{L}$ and

$$P(v_j^n) = \frac{e\left(rK - \sum_{k=1}^{d} \alpha_k D_k x_k\right)}{v_j^n + e - q}. \tag{5.11}$$

These expressions are then used for populating the blocks in the system of form (5.12), that arises when collocating (5.1)-(5.2) at interior nodes (I) and (5.4) at boundary nodes (B).

$$\begin{pmatrix} E_I - \beta_0 L_{II} & -\beta_0 L_{IB} \\ 0 & E_B \end{pmatrix} \begin{pmatrix} \bar{v}_I^n \\ \bar{v}_B^n \end{pmatrix} = \begin{pmatrix} \bar{f}_I^n \\ \bar{f}_B^n \end{pmatrix}, \tag{5.12}$$

where

$$\bar{f}_I^n = \beta_1^n \bar{v}_I^{n-1} - \beta_2^n \bar{v}_I^{n-2} - \beta_0^n P(\bar{v}_I^{n-1}). \tag{5.13}$$

5.3. Approximation in space using RBF–PUM

We define the nodal solution values $v_j^n = V_\varepsilon^n(\mathbf{x}_j) \approx V(t_n, \mathbf{x}_j)$. For the RBF partition of unity method we build an interpolant as described in (4.5)

$$V_\varepsilon^n(\mathbf{x}) = \sum_{i=1}^{M} w_i(\mathbf{x}) V_{loc}^{i,n}(\mathbf{x}) = \sum_{i=1}^{M} w_i(\mathbf{x}) \sum_{j=1}^{N_i} \lambda_j^{i,n} \phi(\varepsilon \|\mathbf{x} - \mathbf{x}_j^i\|). \tag{5.14}$$

Now as in the global case we can enforce interpolation conditions and obtain a linear system

$$\bar{v}^n = \sum_{i=1}^{M} R_i W_i A_i \bar{\lambda}^{i,n}, \tag{5.15}$$

where R_i is a permutation operator which maps the local index set $\mathcal{I}_i = \{1, \ldots, N_i\}$ corresponding to the nodes in the i-th partition into the global index set $\mathcal{I} = \{1, \ldots, N\}$, W_i is a diagonal matrix with element $w_i(\mathbf{x}_j)$ on it, and A_i is a local RBF matrix.

By requiring the local nodal values $v_j^{i,n}$ to coincide with the global nodal values v_j^n, we simplify the coupling together of the local solutions (otherwise, there would be more unknown values than equations, requiring extra conditions). Through the local interpolation property we have

$$\bar{v}^{i,n} = A_i \bar{\lambda}^{i,n}, \quad \Rightarrow \quad \bar{\lambda}^{i,n} = A_i^{-1} \bar{v}^{i,n}. \tag{5.16}$$

Then we construct approximations for the derivatives

$$\begin{aligned}
\frac{\partial \bar{v}^n}{\partial x_k} =& \sum_{i=1}^{M} R_i \left[W_i^{(k)} A_i + W_i A_i^{(k)} \right] \bar{\lambda}^{i,n} = \sum_{i=1}^{M} R_i \left[W_i^{(k)} A_i + W_i A_i^{(k)} \right] A_i^{-1} \bar{v}^{i,n}, \\
\frac{\partial^2 \bar{v}^n}{\partial x_k \partial x_m} =& \sum_{i=1}^{M} R_i \left[W_i^{(km)} A_i + W_i^{(k)} A_i^{(m)} + W_i^{(m)} A_i^{(k)} + W_i A_i^{(km)} \right] \bar{\lambda}^{i,n} = \\
& \sum_{i=1}^{M} R_i \left[W_i^{(km)} A_i + W_i^{(k)} A_i^{(m)} + W_i^{(m)} A_i^{(k)} + W_i A_i^{(km)} \right] A_i^{-1} \bar{v}^{i,n},
\end{aligned}$$

where $W_i^{(k)}$, $W_i^{(km)}$ are diagonal matrices containing the derivatives of w_i and $A_i^{(k)}$, $A_i^{(km)}$ are local derivative RBF matrices. Note that the partition of unity $\{w_i\}_{i=1}^M$ must be at least two times differentiable.

Thus,

$$\begin{aligned}
L\bar{v}^n = \sum_{i=1}^{M} R_i \Bigg[& \frac{1}{2} \sum_{k,m=1}^{d} \Sigma_{km} x_k x_m \left(W_i^{(km)} A_i + W_i^{(k)} A_i^{(m)} + W_i^{(m)} A_i^{(k)} + W_i A_i^{(km)} \right) + \\
& + \sum_{k=1}^{d} (r - D_k) x_k \left(W_i^{(k)} A_i + W_i A_i^{(k)} \right) - r W_i A_i \Bigg] A_i^{-1} \bar{v}^{i,n},
\end{aligned}$$

and

$$P(v_j^n) = \frac{e\left(rK - \sum_{k=1}^{d} \alpha_k D_k x_k\right)}{v_j^n + e - q}.$$

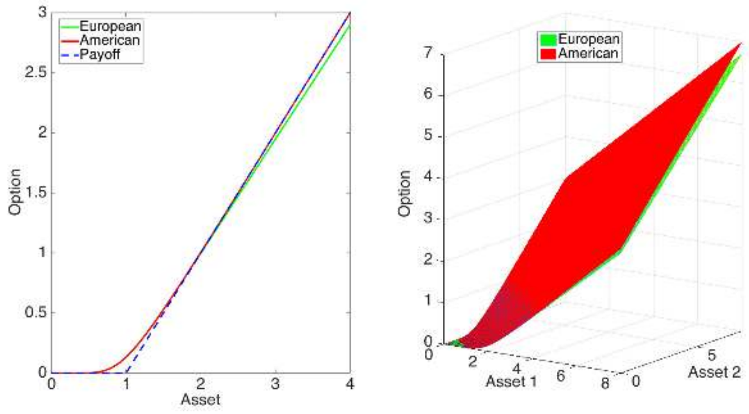


Figure 1: Left: Price of an option on one underlying dividend paying asset. Right: Price of a basket option on two underlying dividend paying assets.

6. Numerical results

In order to solve the option pricing problems numerically, we truncate the domain where the problem is defined. For call options we truncate the domain at $x_\infty = 4dK$ in each direction, at this distance the true solution is close enough to the asymptotic value. Therefore we will carry out numerical experiments on $\Omega = [0, 4dK]^d$. Figure 1 displays typical solutions for European and American options on one and two underlying dividend paying assets.

For the numerical experiments we use the semi-implicit discretisation described in the previous section. The type of basis functions we select is the multiquadric RBF $\phi(r) = \sqrt{1 + \varepsilon^2 r^2}$. It is infinitely smooth and less sensitive to the choice of the shape parameter than, e.g., the Gaussian RBF. We use the following set of parameters: $K = 1$, $T = 1$, $r = 0.1$, $D = 0.05$, $\sigma = 0.3$ for one underlying asset, and $\alpha_{1,2} = 0.5$, $D_{1,2} = 0.05$ and

$$\sigma = \begin{pmatrix} 0.3 & 0.05 \\ 0.05 & 0.3 \end{pmatrix}$$

for two underlying assets.

In order to assess the errors in the numerical solutions the results were compared with accurate reference solutions. In the one-dimensional case for the European call option we use the closed-form solution and for the American call option we use an operator splitting finite difference solution with 2048 discretisation points in space and 8192 points in time. In the two-dimensional case for the European call we use a finite difference solution on a 256×256 grid with 2000 steps in time, and for the American call we use an operator splitting solution on the same 256×256 grid with 2000 time steps. The error in the uniform norm was measured over the around-strike area $\mathcal{U}$, which in the one-dimensional case is $\mathcal{U} = [\frac{K}{3}, \frac{5K}{3}]$ and for the two-dimensional case $\mathcal{U} = [\frac{K}{3}, \frac{8K}{3}] \times [\frac{K}{3}, \frac{8K}{3}]$. These are the relevant regions from the financial point of view.

All methods were implemented in MATLAB R2014b. The codes can be downloaded from `http://www.it.uu.se/research/project/rbf/software/rbfpu_amop_penalty`. All experiments were performed on a laptop with a 2.3 GHz Intel Core i7 processor.

6.1. *Choice of shape parameter ε*

The accuracy of RBF methods highly depends upon the shape parameter ε of the basis functions, which is responsible for the flatness of the functions. For smooth functions, the best accuracy is typically achieved when ε is small, but then the condition number of the linear system becomes very large. In this section we try to find the best compromise for the size of ε for our problem. Figure 2 displays the dependence of the error on the size of the shape parameter for European options issued on one and two assets. In 1D the error is measured against the analytical solution, while in 2D a finite difference solution on a fine grid is used as the reference.

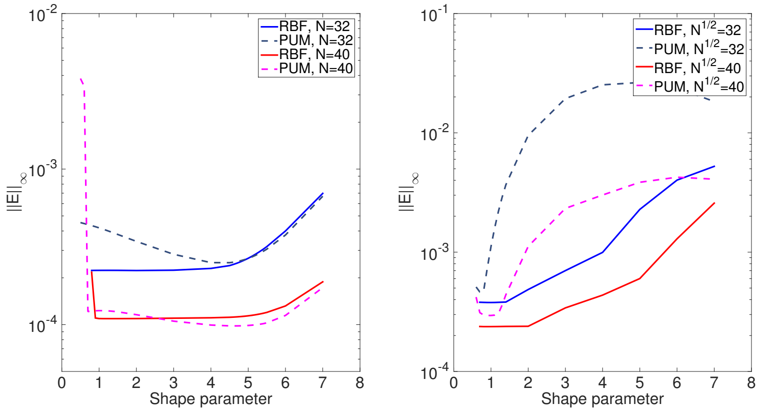


Figure 2: Left: Error in the price of the European option on one underlying asset against the shape parameter ε. Right: Error in the price of the European option on two underlying assets against the shape parameter ε. RBF denotes the global RBF method and PUM denotes RBF–PUM.

For the rest of the experiments in this paper, for each method, we use the ε that was optimal for the finest grid that was used. For example to study the convergence of the global RBF method for the European option on two underlying assets we choose $\varepsilon = 1$, because our finest grid in that experiment is 40×40 nodes, and it turns out that $\varepsilon = 1$ is the optimal choice for that grid.

Error bounds in terms of the number of nodes and the number of partitions for RBF–PUM were derived in [12] based on the results in [20]. These are valid in the case of constant ε. That is, if for the global RBF method we refine the grid and keep $\varepsilon = \varepsilon_0$ then we can expect exponential convergence; if we seek the optimal ε for each grid then the convergence behaviour is less clear.

We use $\varepsilon = 1$ for all European option experiments, $\varepsilon = 1.4$ for the American option on one asset with the global RBF method, $\varepsilon = 1.7$ for the American option on one asset with RBF–PUM, and $\varepsilon = 1$ for the American option on two assets with both methods.

6.2. Refinement strategies for RBF–PUM

For the global RBF method exponential convergence in space with respect to the number of nodes can be expected [12, 20]. For RBF–PUM there are two general methods of refinement: the number of partitions is kept fixed, this means that the number of nodes per partition is increasing under refinement, or the number of points per partition is kept fixed, this means that the number of partitions is growing under refinement. Error estimates were found in [12] of the form:

$$\|E(t)\|_\infty \leq CH^{m-\frac{d}{2}-2} \max_{0\leq\tau\leq t} \max_i \|u(\tau)\|_{\mathcal{N}(\Omega_i)}, \tag{6.1}$$

$$\|E(t)\|_\infty \leq Ce^{-\gamma/\sqrt{h}} \max_{0\leq\tau\leq t} \max_i \|u(\tau)\|_{\mathcal{N}(\Omega_i)}, \tag{6.2}$$

where H is the distance between partition centres, h is the distance between nodes, m is the maximal polynomial degree which can be supported by the number of nodes located in each partition and determines the algebraic convergence order, and γ determines the exponential convergence order. Inequality (6.2) identifies an exponential convergence rate for the case when the number of partitions is fixed, while inequality (6.1) identifies an algebraic convergence rate when the number of points per partition is fixed.

In Figure 3 we test the above estimates for the basket European option on two underlying assets. In the right plot we can see the convergence rate $h^{1.6}$ for nearly 16 points in each partition and $h^{3.5}$ for nearly 33 points per partition; expected convergence rates are h^2 and h^4 respectively. In the left plot we see an exponential convergence with $\gamma = 2$ for 36 partitions over the domain, and $\gamma = 2.1$ for 64 partitions.

This leads us to a reasonable question of what number of partitions (points per partition) is optimal in the sense of computational efficiency? From Figure 3 we can conclude that the fewer the number of partitions (points per partition) the lower (higher) the error becomes. However, the linear system becomes denser (sparser) and requires more (less) time to solve. This trade-off we study in the following subsection.

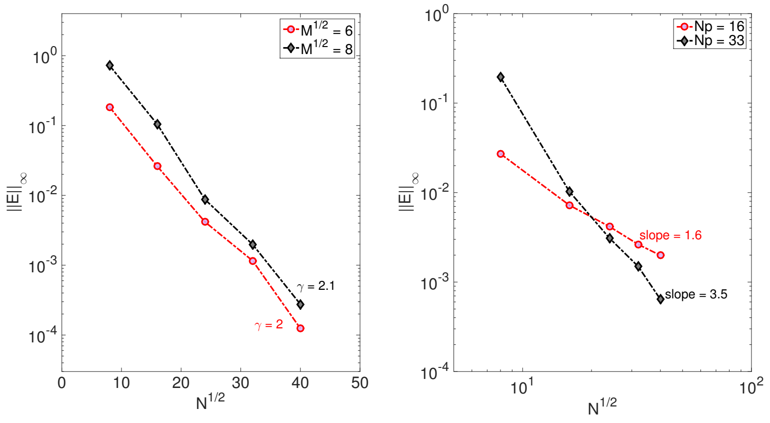


Figure 3: Left: Error in the price of the European option on two underlying assets against the problem size with respect to the number of partitions. Right: Error in the price of the European option on two underlying assets against the problem size with respect to the number of points per partition. Shape parameter $\varepsilon = 1$.

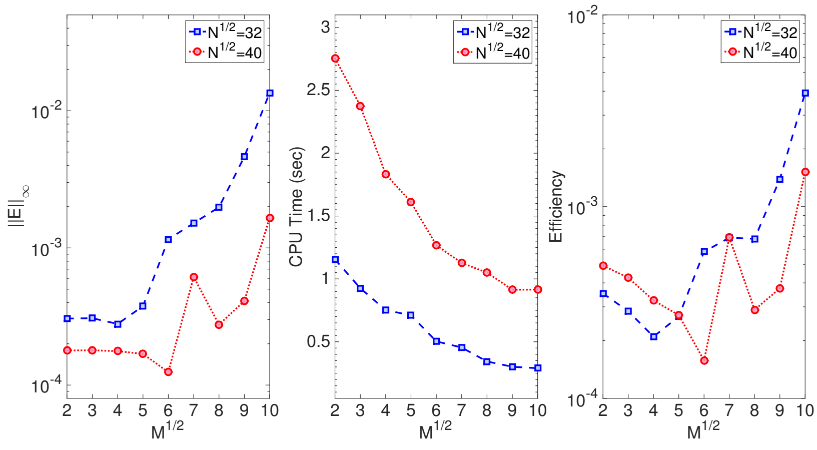


Figure 4: Left: Error in the price of the European option on 2 underlying assets against the number of partitions in one spatial dimension. Centre: Computational time against the number of partitions in one spatial dimension. Right: Efficiency computed as product between the error and CPU time. Shape parameter $\varepsilon = 1$.

6.3. Number of partitions

In the case of RBF–PUM there is a freedom to choose the number of partitions that will cover the domain. A cover with smaller partitions will lead to worse approximation results, but will on the other hand be computationally cheaper, because the linear system will be more sparse.

In Figure 4 the error in the price of the European option on two underlying assets versus the number of partitions in one spatial dimension is shown on the left, the corresponding computational time is shown in the centre, and computational efficiency as a product of the two on the right. The efficiency gives us a flavour of which number of partitions is optimal in terms of error–time.

From the figure we see for example that for 100 partitions the computational time is low while the error is large. Then the product will be moderately large. For four partitions it is the other way around, the time is high and the error is low. The optimum is found at 36 partitions, where the error is the lowest and the computational time is average. Based on this we select $\sqrt{M} = 6$ for our two-dimensional experiments. This leads to about 100 nodes in each partition for the finest grid (40×40 nodes) and about 10% non-zero elements in the linear system. For the one-dimensional experiments we choose $M = 4$. Note that the optimal RBF shape parameter is not sensitive to the number of partitions and for this experiment $\varepsilon = 1$.

6.4. Penalty parameter

In this section we study the dependence of the solution and the numerical scheme on the penalty parameter e. We have already mentioned that the error is expected to decay linearly with the penalty size, even if our theoretical result for the continuous problem shows a bound involving $\sqrt{e}$. Figure 5 confirms our expectations. The dependence is roughly linear in both the one-dimensional and two-dimensional case.

When we designed the numerical scheme we mentioned that the semi-implicit scheme may impose a less severe condition on the time step size than a fully explicit scheme. This is true for some choices of e. In the right part of Figure 5, we show the dependence of the time step size on the penalty parameter size together with the level of the time step for the explicit scheme. Here we should not forget that there is no sense in using a small penalty parameter for coarse grids and *vice versa*, because the two types of errors should be balanced.

The experiment shows that the use of the semi-implicit scheme does not always have an advantage in terms of time step size for the RBF methods, because the condition imposed by treating only the penalty explicitly is more severe than the condition in the fully explicit scheme, which depends on the space discretisation. As RBF methods have high convergence rates, few points are needed in space and hence a relatively large time step can be used also in the explicit scheme.

The right part of Figure 5 also displays that the time step should be chosen according to condition (5.5). The purple line indicates the analytical time step limit obtained from (5.5) with $C = 3/2$, and the turquoise line indicates the largest time step for which a stable numerical result was computed.

6.5. Convergence study: European option

Here we study the convergence rates of the global RBF method and RBF–PUM and compare them with a standard second order central finite difference (FD) method

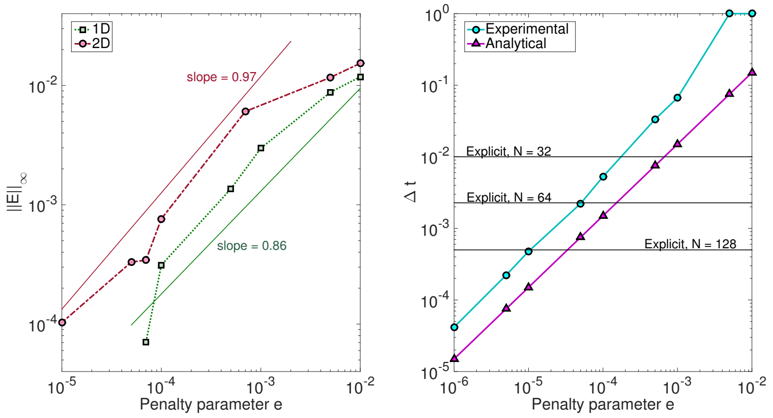


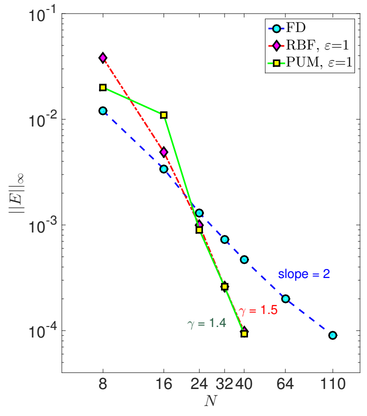
Figure 5: Left: Error measured in the region $\mathcal{U}$ against the penalty parameter size for the one asset and two asset cases. Right: Stable time step size for different sizes of the penalty parameter. Analytical—obtained from inequality (5.5) with $C = 3/2$, Experimental—experimentally obtained maximal time step for which stability holds. The three black lines show the time step size required for stability with the fully explicit scheme.

on a uniform grid. In one dimension a closed-form solution for the European option exists, whilst in two dimensions it does not, and we have to use a reference solution obtained by the FD method on a fine enough (256×256) grid to compare with. For this experiment we choose a large number of discretisation points in time ($N_t = 1000$) in order to avoid any influence of the time discretisation on the convergence rates of the methods.

As expected, in Figure 6, we observe a second order algebraic convergence rate for the FD method and exponential convergence for both RBF methods with $\gamma = 1.5$ for the global method, and $\gamma = 1.5$ in 1D and $\gamma = 2$ in 2D for RBF–PUM.

For the European option pricing problem, the initial condition is only a C^0 function. Hence exponential approximation accuracy at the initial time is not possible as this requires smoothness of the solution [20]. However, due to the smoothing properties of parabolic problems, the solution can be approximated with high accuracy at larger times [12]. It has been proved in [40], that solutions of parabolic problems with non-smooth initial condition can be approximated with optimal order when time is positive.

For financial applications an error of the size 10^{-4} is considered to be precise enough, and it is clear that to reach the desired accuracy the FD method requires a larger number of node points. In order to reach this error level, the global RBF method and RBF–PUM require 40 nodes (40 in each direction in 2D), while the FD method needs 100 nodes (112 in each direction in 2D). However, the computational



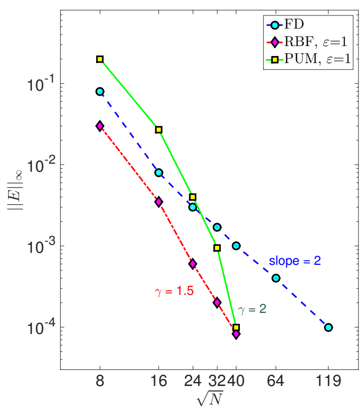


Figure 6: Left: Error convergence in l_∞-norm for a European option on one underlying asset. Right: Error convergence in l_∞-norm for a basket European option on two underlying asset.

cost per time-step is very different for the three methods and a time-comparison is therefore performed in section 6.7.

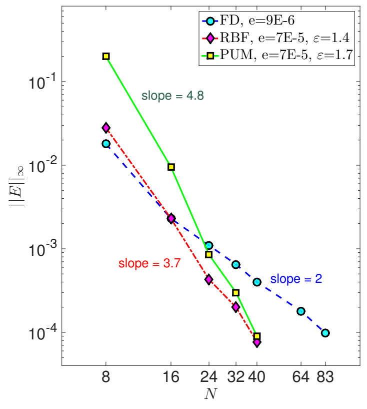
A property of the global RBF method and RBF–PUM is that they can easily reach error levels of $10^{-4} - 10^{-5}$, but then the system becomes ill-conditioned and lower error levels cannot be reached [30]. To overcome this problem the RBF–QR method was invented. It allows stable computations when the shape parameter $\varepsilon \to 0$ and it allows for achieving higher accuracy. We do not employ the RBF–QR technique because our error target can be attained without it, but it can be useful when a low price of an option is expected and a higher precision in the result is required. More details about RBF–QR can be found in [28, 29, 30].

6.6. Convergence study: American option

Here we study the convergence rates of the global RBF and RBF–PUM penalty methods and compare them with the FD penalty method. Since no closed-form solution exists in the case of American options, as a reference to measure the error we use a solution obtained by second order central finite differences combined with the operator splitting (OS) method [4] on a fine enough grid (2048 points in 1D and 256×256 points in 2D). Note that the OS method approximates the original PDE, and therefore the error introduced by the penalty term is not present. The number of required discretisation points in time is governed by the stability condition (5.5), for example if the chosen $e = 10^{-5}$, then to maintain stability the required $N_t \approx 10^4$.

In the case of American options, the second derivative of the solution has a discontinuity at the free boundary. This will limit the order of convergence.

As we said before, we aim for error of the order 10^{-4} which is sufficient for financial



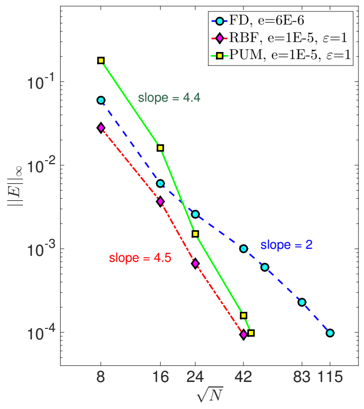


Figure 7: Left: Error convergence in l_∞-norm for an American option on one underlying asset. Right: Error convergence in l_∞-norm for a basket American option on two underlying asset. All the three methods use the penalty approach.

applications. The error introduced by the penalty term is $\mathcal{O}(e)$. Therefore we have to choose the penalty parameter e smaller than 10^{-4}.

In Figure 7 we see that all the three methods reach the specified error limit, but the FD-penalty method requires a smaller penalty parameter (which leads to a larger number of time steps to fulfil the stability condition) as well as a higher number of computational nodes in space.

As expected, the discontinuity in the second derivative of the solution does not allow for exponential convergence, but for the error range investigated here we get a high order algebraic convergence rate both for the global RBF method and RBF–PUM.

6.7. Computational efficiency

As we mentioned previously, RBF methods require fewer computational nodes than standard FD methods, but the cost of each time step is higher. Table 1 shows computational times needed to achieve a certain level of accuracy for the FD method, the global RBF method, and RBF–PUM for pricing a double-asset European option. The number of time steps is adjusted to be nearly optimal (in terms of computational time and accuracy) for each run. We can see that in order to get to the error level 10^{-4}, RBF–PUM requires 40 times less time than the standard FD method and 26 times less time that the global RBF method. The main reason for the significant time gain with RBF–PUM is that the number of time steps needed is much lower than for the global RBF method. We cannot fully quantify this effect, but an advantage of RBF–PUM compared with the global method is that we avoid at least parts of the high frequency oscillations induced in the strike region and at the boundaries when using a global RBF approximation [41]. As low frequency components in a parabolic PDE propagates at a slower time scale, we can then use a lower resolution in time.

For the experiments we measured only the time corresponding to the time-stepping loop, while the setup cost is not included. For RBF–PUM, the computations of the local matrices and the assembly can easily be parallelised and will not greatly affect the overall time. Therefore we do not take it into account.

Table 1: European double-asset option. The CPU time (sec) required to achieve the given error levels and the discretisation parameters used. Shape parameter $\varepsilon = 1$ for the global RBF method and RBF–PUM.

	FD		RBF		RBF–PUM	
$\|\|E\|\|_\infty$	Time	$\sqrt{N} \times N_t$	Time	$\sqrt{N} \times N_t$	Time	$\sqrt{N} \times N_t$
1e–2	0.0028	19×20	0.0024	12×40	0.0026	12×3
5e–3	0.0076	39×30	0.0034	14×40	0.0033	14×3
1e–3	0.0684	47×200	0.0374	22×100	0.0043	22×4
5e–4	0.3439	67×300	0.1581	26×200	0.0050	22×5
1e–4	4.1762	119×680	2.6778	42×500	0.1044	42×10

Table 2 displays computational times for the double-asset American option. The number of time steps N_t is chosen to be as small as possible while preserving stability. It is different for different methods, because some methods require a smaller penalty size, therefore they need a larger N_t to still remain stable. As we can see, to reach a 10^{-4} error level RBF–PUM requires roughly 4 times less time than the standard finite difference method and 2 times less time than the global RBF method.

Table 2: American double-asset option. The CPU time (sec) required to achieve the given error levels and the discretisation parameters used. Shape parameter $\varepsilon = 1$ for the global RBF method and RBF–PUM.

	FD		RBF		RBF–PUM	
$\|\|E\|\|_\infty$	Time	$\sqrt{N} \times N_t$	Time	$\sqrt{N} \times N_t$	Time	$\sqrt{N} \times N_t$
1e–2	0.0026	15×30	0.0036	12×50	0.0053	12×50
5e–3	0.0088	23×70	0.0054	14×60	0.0061	14×50
1e–3	0.1640	43×600	0.1527	22×500	0.2155	28×800
5e–4	1.1127	59×1500	0.9032	30×800	0.4825	32×1000
1e–4	87.552	115×15000	42.996	42×10000	24.238	46×10000

7. Summary

RBF methods provide an alternative to already existing methods for solving problems in financial applications. RBF–PUM allows to overcome the high computational cost associated with the global RBF method, while maintaining high accuracy. RBF–PUM also allows to reach a given level of accuracy with significantly less computational effort than the standard FD method and the global RBF method for both European-style and American-style multi-asset options. One way to reduce the computational time even more is to use the geometrical flexibility of RBF methods. For example, the two-dimensional problem can be easily solved on a triangular domain instead of a square domain, thus, halving the problem size.

The fact that RBF methods are meshfree allows an easy implementation of adaptive grids, which can be clustered around critical regions such as the strike area or the free boundary, in order to improve accuracy or reduce overall computational cost. In the case of RBF–PUM, refinements can be made independently within the partitions, increasing the flexibility.

With either of the RBF methods, solutions with errors of the order 10^{-4} can be stably computed with the direct RBF evaluation method described here. If lower errors are required, a different evaluation method, such as for example the RBF–QR method, is needed. However, in the case of American options the accuracy is also limited by the size of the penalty parameter.

The penalty method combined with RBFs is a good approach for pricing American options. It allows for removing the free boundary and transforming the problem to a fixed boundary problem. It facilitates the computations, in the sense that we do not have to track the free boundary location. It can be used in high dimensions and the introduced error can easily be adjusted to a desirable level.

References

[1] F. Black, M. Scholes, The pricing of options and corporate liabilities, J. Polit. Econ. 81 (3) (1973) 637–654.

[2] J. Hull, Options, Futures and Other Derivatives, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2009.

[3] D. Tavella, C. Randall, Pricing Financial Instruments: The Finite Difference Method, Wiley Series in Financial Engineering, Wiley, New York, 2000.

[4] S. Ikonen, J. Toivanen, Operator splitting methods for American option pricing, Appl. Math. Lett. 17 (7) (2004) 809–814.

[5] R. Zvan, P. A. Forsyth, K. R. Vetzal, Penalty Methods for American Options with Stochastic Volatility, J. Comput. Appl. Math. 91 (2) (1998) 199–218.

[6] P. A. Forsyth, K. R. Vetzal, Quadratic convergence of a penalty method for valuing American options, SIAM J. Sci. Comput. 23 (2002) 2095–2122.

[7] B. F. Nielsen, O. Skavhaug, A. Tveito, Penalty and front-fixing methods for the numerical solution of American option problems, J. Comput. Finance 5 (4) (2002) 69–97.

[8] Y. d'Halluin, P. A. Forsyth, G. Labahn, A penalty method for American options with jump diffusion processes, Numer. Math. 97 (2) (2004) 321–352.

[9] G. E. Fasshauer, A. Q. M. Khaliq, D. A. Voss, Using meshfree approximation for multi-asset American option problems, J. Chinese Inst. Engrs. 27 (4) (2004) 563–571.

[10] A. Q. M. Khaliq, R. H. Liu, New numerical scheme for pricing American option with regime-switching, Int. J. Theor. Appl. Finance 12 (3) (2009) 319–340.

[11] A. Belova, T. Shmidt, M. Ehrhardt, Meshfree methods in option pricing, in: Embedded Computing (MECO), 2012 Mediterranean Conference, 243–246, 2012.

[12] A. Safdari-Vaighani, A. Heryudono, E. Larsson, A radial basis function partition of unity collocation method for convection-diffusion equations, J. Sci. Comput. 64 (2) (2015) 341–367.

[13] J. A. Rad, K. Parand, L. Ballestra, Pricing European and American options by radial basis point interpolation, Appl. Math. Comput. 251 (2015) 363–377.

[14] Y. C. Hon, A quasi-radial basis functions method for American options pricing, Comput. Math. Appl. 43 (3-5) (2002) 513–524.

[15] Y. C. Hon, X. Z. Mao, A radial basis function method for solving options pricing model, Financial Engineering 8 (1999) 31–49.

[16] Z. Wu, Y. C. Hon, Convergence error estimate in solving free boundary diffusion problem by radial basis functions method, Eng. Anal. Bound. Elem. 27 (1) (2003) 73 – 79.

[17] P. Glasserman, Monte Carlo methods in financial engineering, Springer-Verlag, New York, 2004.

[18] S. Borovkova, F. Permana, J. van der Weide, American basket and spread option pricing by a simple binomial tree, J. Derivatives 19 (2012) 29–38.

[19] F. Fang, C. W. Oosterlee, A novel pricing method for European options based on Fourier-cosine series expansions, SIAM J. Sci. Comput. 31 (2) (2008) 826–848.

[20] C. Rieger, B. Zwicknagl, Sampling inequalities for infinitely smooth functions, with applications to interpolation and machine learning, Adv. Comput. Math. 32 (1) (2010) 103–129.

[21] U. Pettersson, E. Larsson, G. Marcusson, J. Persson, Improved radial basis function methods for multi-dimensional option pricing, J. Comput. Appl. Math. 222 (1) (2008) 82–93.

[22] E. Larsson, A. Heryudono, A partition of unity radial basis function collocation method for partial differential equations, in preparation, 2015.

[23] M. Tillenius, E. Larsson, E. Lehto, N. Flyer, A scalable RBF–FD method for atmospheric flow, J. Comput. Phys. 298 (2015) 406–422.

[24] L. V. Ballestra, G. Pacelli, Pricing European and American options with two stochastic factors: A highly efficient radial basis function approach, J. Econ. Dyn. Control 37 (6) (2013) 1142–1167.

[25] I. Babuška, J. M. Melenk, The partition of unity method, Internat. J. Numer. Methods Engrg. 40 (4) (1997) 727–758.

[26] E. F. Bollig, N. Flyer, G. Erlebacher, Solution to PDEs Using Radial Basis Function Finite-differences (RBF-FD) on Multiple GPUs, J. Comput. Phys. 231 (21) (2012) 7133–7151.

[27] G. Kosec, M. Depolli, A. Rashkovska, R. Trobec, Super linear speedup in a local parallel meshless solution of thermo-fluid problems, Comput. Struct. 133 (0) (2014) 30–38.

[28] B. Fornberg, C. Piret, A stable algorithm for flat radial basis functions on a sphere, SIAM J. Sci. Comput. 30 (1) (2007) 60–80.

[29] B. Fornberg, E. Larsson, N. Flyer, Stable computations with gaussian radial basis functions, SIAM J. Sci. Comput. 33 (2) (2011) 869–892.

[30] E. Larsson, E. Lehto, A. Heryudono, B. Fornberg, Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions, SIAM J. Sci. Comput. 35 (4) (2013) 2096–2119.

[31] G. Fichera, Sulle equazioni differenziali lineari ellittico-paraboliche del secondo ordine, Atti Accad. Naz. Lincei. Mem. Cl. Sci. Fis. Mat. Nat. Sez. I. VIII, Ser. 5 (1956) 3–30.

[32] W. Feller, Two singular diffusion problems, Ann. of Math. (2) 54 (1) (1951) 173–182.

[33] R. Courant, Variational methods for the solution of problems of equilibrium and vibrations, Bull. Amer. Math. Soc. 49 (1943) 1–23.

[34] B. F. Nielsen, O. Skavhaug, A. Tveito, Penalty methods for the numerical solution of American multi-asset option problems, J. Comput. Appl. Math. 222 (1) (2008) 3–16.

[35] V. A. Kholodnyi, A nonlinear partial differential equation for American options in the entire domain of the state variable, Nonlinear Anal. 30 (8) (1997) 5059–5070.

[36] D. Shepard, A Two-dimensional Interpolation Function for Irregularly-spaced Data, in: Proceedings of the 1968 23rd ACM National Conference, ACM '68, ACM, New York, NY, USA, 517–524, 1968.

[37] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. Comput. Math. 4 (4) (1995) 389–396.

[38] E. Hairer, S. Nørsett, G. Wanner, Solving Ordinary Differential Equations I. Nonstiff problems, Springer-Verlag, Berlin, second edn., 2000.

[39] E. Larsson, K. Åhlander, A. Hall, Multi-dimensional option pricing using radial basis functions and the generalized Fourier transform, J. Comput. Appl. Math. 222 (1) (2008) 175–192.

[40] V. Thomée, Galerkin Finite Element Methods For Parabolic Problems, vol. 25 of *Springer Series in Computational Mathematics*, Springer-Verlag, Berlin, second edn., 2006.

[41] B. Fornberg, J. Zuev, The Runge phenomenon and spatially variable shape parameters in RBF interpolation, Comput. Math. Appl. 54 (3) (2007) 379–398.

Paper II

International Journal of Computer Mathematics, 2015
Vol. 92, No. 12, 2361–2379, http://dx.doi.org/10.1080/00207160.2015.1072172

BENCHOP – The BENCHmarking project in option pricing[†]

Lina von Sydow[a*], Lars Josef Höök[a], Elisabeth Larsson[a], Erik Lindström[b], Slobodan Milovanović[a], Jonas Persson[c], Victor Shcherbakov[a], Yuri Shpolyanskiy[d], Samuel Sirén[c], Jari Toivanen[e,f], Johan Waldén[g], Magnus Wiktorsson[b], Jeremy Levesley[h], Juxi Li[h], Cornelis W. Oosterlee[i,j], Maria J. Ruijter[i], Alexander Toropov[d] and Yangzhang Zhao[h]

[a]*Department of Information Technology, Uppsala University, Uppsala, Sweden;* [b]*Centre for Mathematical Sciences, Lund University, Lund, Sweden;* [c]*SUNGARD FRONT ARENA, Stockholm, Sweden www.sungard.com/frontarena;* [d]*ORC GROUP, Stockholm, Sweden, www.orc-group.com and ITMO University, St Petersburg, Russia;* [e]*Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland;* [f]*The Institute for Computational and Mathematical Engineering (ICME), Stanford University, Stanford, CA, USA;* [g]*Haas School of Business, University of California Berkeley, Berkeley, CA, USA;* [h]*Department of Mathematics, University of Leicester, Leicester, UK;* [i]*Centrum Wiskunde & Informatica, Amsterdam, The Netherlands;* [j]*Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands*

(*Received 30 January 2015; revised version received 18 June 2015; accepted 23 June 2015*)

The aim of the BENCHOP project is to provide the finance community with a common suite of benchmark problems for option pricing. We provide a detailed description of the six benchmark problems together with methods to compute reference solutions. We have implemented fifteen different numerical methods for these problems, and compare their relative performance. All implementations are available on line and can be used for future development and comparisons.

Keywords: option pricing; numerical methods; benchmark problem; Monte Carlo method; Fourier-method; finite difference method; radial basis function

2010 AMS Subject Classifications: 65-02; 91G60; 91G20

1. Introduction

The research on numerical methods for option pricing problems has been extensive over the last decades and there is now a plethora of methods targeting various types of options. However, there is a lack of cross comparisons between methods and a similar lack of common benchmarks to evaluate new approaches.

The aim of BENCHOP is to provide the finance community with a set of common benchmark problems that can be used both for comparisons between methods and for evaluation of new methods. Furthermore, in order to facilitate comparisons, MATLAB implementations of a wide range of existing methods for each benchmark problem will be made available through the BENCHOP web site www.it.uu.se/research/project/compfin/benchop.

*Corresponding author. Email: lina@it.uu.se
[†]The list of authors is organized in the following way: Project leader; Main contributors in alphabetical order; Code contributors in alphabetical order.

We also aim for BENCHOP to serve as a takeoff for future development of methods in option pricing. We expect future papers in the field to use the BENCHOP codes and problems to evaluate performance. In this way, we can contribute to a more uniform and comparable evaluation of the relative strengths and weaknesses of proposed methods.

The benchmark problems have been chosen in such a way as to be relevant both for practitioners and researchers. They should also be possible to implement with a reasonable effort. We have selected problems with respect to a number of features that may be numerically challenging. These are early exercise properties, barriers, discrete dividends, local volatility, stochastic volatility, jump diffusion, and two underlying assets. We have also included evaluation of hedging parameters in one of the problems, as this adds additional difficulties.

In this paper, we present the benchmark problems with sufficient detail so that other people can solve them in the future. We also provide analytical solutions where such are available or methods for computing accurate reference solutions otherwise. Each problem is solved using MATLAB implementations of a number of already existing numerical methods, and timing results are provided as well as error plots. For details of the methods, we refer to the original papers and additional notes at the BENCHOP web site. The codes are not fully optimized, and the numerical results should not be interpreted as competition scores. We rather see it as a synoptical exposition of the qualities of the different methods.

In Section 2, we state and motivate the benchmark problems while the numerical methods are briefly presented in Section 3. Section 4 is dedicated to the presentation of the numerical results and finally in Section 5, we discuss the results. In Appendix 1, we present how the reference values are computed for the different problems and in Appendix 2, we discuss how the local volatility surface is computed for one of the problems.

2. Benchmark problems

In this section, we state each of the six benchmark problems. In the mathematical formulations, we let S represent the actual (stochastic) asset price realization, whereas s is the asset price variable in the PDE formulation of the problem, t is the time (with $t = 0$ representing today), r is the risk free interest rate, σ is the volatility, W is a Wiener process, u is the option price as a function of s, K is the strike price, and T is the time of maturity. The payoff function $\phi(s)$ is the value of the option at time T. In Problem 4, V is the stochastic variance variable, and v is the variance value in the PDE-formulation.

In practice, the asset value today, S_0, is a known quantity, while the strike price K can take on different values. In the benchmark problem descriptions below we have chosen to fix all parameters, and then solve for different values of S_0 to simulate pricing of options that are 'in the money', 'at the money' and 'out of the money'. The initial values are not given in the problem descriptions, but for each table and figure in the numerical results section, the values that were used are listed.

2.1 *Problem 1: The Black–Scholes–Merton model for one underlying asset*

The celebrated Black–Scholes–Merton [4,39] option pricing model, developed in the early 1970's, is arguably the most successful quantitative model ever introduced in social sciences, even initiating the new field of Financial Engineering, which occupies thousands of researchers in financial institutions and universities across the world.

A key property of the model is that by building on so-called no-arbitrage arguments, it allows the price of plain vanilla call and put options to be calculated using variables that are either directly observable or can be easily estimated. The model is still widely used as a benchmark, although more advanced models have been developed over the years to take into account real-world features of asset price dynamics, such as jumps and stochastic volatility (see below).

The Black–Scholes–Merton model has the advantage that closed form solutions exist for prices, as well as for hedging parameters, for some types of options. It has therefore been extensively used to test numerical methods that are then applied to more advanced problems. The computation of the hedging parameters (Greeks) is included in this benchmark problem as they are of significant practical interest and can be expensive and/or difficult to compute for some numerical methods.

Mathematical formulation

$$\text{SDE-setting:}\quad \mathrm{d}S = rS\,\mathrm{d}t + \sigma S\,\mathrm{d}W. \tag{1}$$

$$\text{PDE-setting:}\quad \frac{\partial u}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 u}{\partial s^2} + rs\frac{\partial u}{\partial s} - ru = 0. \tag{2}$$

Deliverables

The pricing problem should be solved for three types of options; (a) a European call option, (b) an American put option, and (c) a barrier option. For the European option also the most common hedging parameters $\Delta = \partial u/\partial s$, $\Gamma = \partial^2 u/\partial s^2$ and $\mathcal{V} = \partial u/\partial \sigma$ should be computed.

Parameter and problem specifications

For this problem we have two sets of model parameters, representing less and more numerically challenging situations, respectively.

$$\text{Standard parameters:}\quad \sigma = 0.15, \quad r = 0.03, \quad T = 1.0, \quad \text{and} \quad K = 100. \tag{3}$$

$$\text{Challenging parameters:}\quad \sigma = 0.01, \quad r = 0.10, \quad T = 0.25, \quad \text{and} \quad K = 100. \tag{4}$$

The three types of options are characterized by their exercise properties and payoff functions.

$$\text{(a) European call :}\qquad \phi(s) = \max(s-K,0).$$

$$\text{(b) American put :}\qquad \phi(s) = \max(K-s,0),$$
$$u(s,t) \geq \phi(s), \quad 0 \leq t \leq T.$$

$$\text{(c) Barrier call up-and-out :}\qquad \phi(s) = \begin{cases} \max(s-K,0), & 0 \leq s < B \\ 0, & s \geq B \end{cases}, \quad B = 1.25K.$$

2.2 *Problem 2: The Black–Scholes–Merton model with discrete dividends*

A shortcoming of the classical Black-Scholes formula is that it is only valid if the underlying stock does not pay dividends, invalidating the approach for many stocks in practice. In some special cases, for example, when dividend yields are constant and paid continuously over time, closed form solutions can be derived for dividend paying stocks too, see [39]. Usually, however, numerical methods are needed to calculate the option's value.

In practice, dividends are paid at discrete points in time, and the size of the dividend payments depends on the performance of the firm. For example, a firm whose performance has been poor may be capital constrained and therefore choose not to make a dividend payment, as may a company that needs its capital for a new investment opportunity. Fairly advanced stochastic modeling may therefore be needed in practice to capture the dividend dynamics of a company. In numerical tests, it is common to abstract away from these issues and simply assume that the firm makes discrete proportional dividend payments (i.e. has a constant dividend yield).

Mathematical formulation

$$\text{SDE-setting:} \quad \mathrm{d}S = rS\,\mathrm{d}t + \sigma S\,\mathrm{d}W - \delta(t-\tau)D\,S\,\mathrm{d}t. \tag{5}$$

$$\text{PDE-setting:} \quad \frac{\partial u}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 u}{\partial s^2} + rs\frac{\partial u}{\partial s} - ru = 0, \tag{6}$$

In the SDE case, the (single) dividend at time τ enters explicitly, whereas in the PDE case, it is implicitly taken into account by enforcing

$$u(s, \tau^-) = u(s(1-D), \tau^+). \tag{7}$$

Deliverables

Prices should be computed for a) a European call option and b) an American call option.

Parameter and problem specifications

The dividend is defined by $\tau = 0.4$ and $D = 0.03$. We use the standard parameters (3) except for the expiration time, set to $T = 0.5$, together with standard payoff function $\phi(s) = \max(s - K, 0)$ and European and American exercise properties respectively, see Problem 1.

2.3 *Problem 3: The Black–Scholes–Merton model with local volatility*

As mentioned earlier, the Black–Scholes–Merton model with a constant volatility does not reproduce market prices very well in practice. One discrepancy is the so-called volatility smile (which after the October 1987 crash is known to have turned into a smirk). If the implied volatility—the volatility in the Black–Scholes–Merton model that is consistent with the observed option price—is calculated for several options with the same exercise date but different strike prices, all options should under the classical assumptions of Black–Scholes–Merton have the same implied volatility. Instead, when plotted against the different strike prices, the curve is usually that of a U-shaped smile (or an L-shaped smirk).

As discussed in [10], an approach to address this discrepancy between model and data is to assume local volatility, that is, to allow the volatility of the underlying asset to depend instantaneously on the stock price s, and time t, generating a whole volatility surface. It is shown in [10] how to reverse engineer such a volatility surface from observed option prices.

Given a volatility surface, the general Black–Scholes–Merton no-arbitrage approach can be used to derive the option price, although closed form solutions will typically no longer exist. We provide two volatility surfaces with different properties in order to see how the numerical methods handle such variable volatility coefficients.

Mathematical formulation

The equations for the option price are identical to (1) and (2) except that here $\sigma = \sigma(s, t)$.

Deliverables

The price for a European call option should be computed in each case.

Parameter and problem specifications

The first local volatility surface is given by an explicit function

$$\sigma_I(s, t) = 0.15 + 0.15(0.5 + 2t)\frac{(s/100 - 1.2)^2}{(s/100)^2 + 1.44}. \tag{8}$$

The second local volatility surface $\sigma_{II}(s, t)$ is based on market data and does not have an explicit form. The local surface is computed from a parametrization of the implied volatility data. The exact steps in the computation and the specific parametrization are given in Appendix 2.

In both cases, we use $K = 100$ and $r = 0.03$, but the expiration times are different, with $T = 1$ for σ_I, and $T = 0.5$ for σ_{II}. The payoff function for a European call option is as before $\phi(s) = \max(s - K, 0)$.

2.4 *Problem 4: The Heston model for one underlying asset*

The local volatility model allows for perfect matching of prices of European-style options but, just like the Black–Scholes–Merton model, also has its weaknesses. It does not perform very well for path dependent options and, moreover, there is clear evidence that in practice the volatility of asset prices is in itself random, beyond what can be simply be described as a function of time and underlying strike price [9,32,45]. The Heston model [20] assumes that in addition to the risk-factor that drives the value of the underlying asset, there is a another risk-factor that determines the underlying's instantaneous variance, V. The PDE formulation of the model is therefore two-dimensional. Note that in contrast to the previous models, the market in Heston's model is incomplete, and therefore additional assumptions about the market price of volatility risk are needed to determine the option price. The specific assumptions in [20] leads to the model below.

Mathematical formulation

SDE-setting:

$$\begin{aligned} \mathrm{d}S &= rS\,\mathrm{d}t + \sqrt{V}S\,\mathrm{d}W_1, \\ \mathrm{d}V &= \kappa(\theta - V)\,\mathrm{d}t + \sigma\sqrt{V}\,\mathrm{d}W_2, \end{aligned} \tag{9}$$

where W_1 and W_2 have correlation ρ.

PDE-setting:

$$\frac{\partial u}{\partial t} + \frac{1}{2}vs^2\frac{\partial^2 u}{\partial s^2} + \rho\sigma vs\frac{\partial^2 u}{\partial s\partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 u}{\partial v^2} + rs\frac{\partial u}{\partial s} + \kappa(\theta - v)\frac{\partial u}{\partial v} - ru = 0. \tag{10}$$

Deliverables

The price for a European call option should be computed.

Parameter and problem specifications

The model parameters are here given by $r = 0.03$, $\kappa = 2$, $\theta = 0.0225$, $\sigma = 0.25$, $\rho = -0.5$, $K = 100$, and $T = 1$. The payoff function for the European call option is $\phi(s, v) = \max(s - K, 0)$. With these parameters, the Feller condition is satisfied.

2.5 *Problem 5: The Merton jump diffusion model for one underlying asset*

The Merton model [40] addresses another difference between real world asset price dynamics and the (local volatility) Black–Scholes–Merton model. What was identified early on, is that stock prices occasionally experience dramatic movements over very short time periods, that is, they sometimes 'jump'. Such jumps make return distributions heavier-tailed than for pure diffusion processes, also in line with empirical observations and, as in the Heston model, causes the market to be incomplete, necessitating additional assumptions to price the option. The assumption used in [40] is that the underlying stock price follows a jump-diffusion process, where there is no risk-premium associated with jump risk. Under these conditions, the option price can be computed from a Partial-Integro Differential Equation (PIDE).

Mathematical formulation

SDE-setting:

$$\mathrm{d}S = (r - \lambda\xi)S\,\mathrm{d}t + \sigma S\,\mathrm{d}W + S\,\mathrm{d}Q, \tag{11}$$

where Q is a compound Poisson process with intensity $\lambda > 0$ and jump ratios that are log-normally distributed as $p(y) = 1/\sqrt{2\pi}y\delta\mathrm{e}^{-(\log y - \gamma)^2/2\delta^2}$ [50].

PIDE-setting:

$$\frac{\partial u}{\partial t}+\frac{1}{2}\sigma^2 s^2\frac{\partial^2 u}{\partial s^2}+(r-\lambda\xi)s\frac{\partial u}{\partial s}-(r+\lambda)u+\lambda\int_0^\infty u(sy,\tau)p(y)\,\mathrm{d}y=0. \tag{12}$$

Deliverables

The price for a European call option should be computed.

Parameter and problem specifications

The parameters to use are $r=0.03$, $\lambda=0.4$, $\gamma=-0.5$, $\delta=0.4$, $\xi=\mathrm{e}^{\gamma+\delta^2/2}-1$, $\sigma=0.15$, $K=100$, and $T=1$. The payoff function is $\phi(s)=\max(s-K,0)$.

2.6 *Problem 6: The Black–Scholes–Merton model for two underlying assets*

As an example of an option with more than one underlying, we use a spread option, which for $K=0$ is called a Margrabe option [38]. This classic rainbow option has a payoff function that depends on two underlying assets, so the option price dynamic therefore depends on two risk-factors (as long as the two stocks' returns are not perfectly correlated). In contrast to the Heston and Merton models, the model is still within the class of complete market models, and the option price is therefore completely determined without further assumptions. The reason that the market is complete in this case, in contrast to the other multi risk-factor models we have introduced, is that two underlying risky assets may be used in the formation of a hedging portfolio, whereas only one such asset is available with the Heston and Merton Models.

Mathematical formulation

SDE-setting:

$$\begin{aligned}\mathrm{d}S_1&=rS_1\,\mathrm{d}t+\sigma_1 S_1\,\mathrm{d}W_1,\\ \mathrm{d}S_2&=rS_2\,\mathrm{d}t+\sigma_2 S_2\,\mathrm{d}W_2,\end{aligned} \tag{13}$$

where W_1 and W_2 have correlation ρ.

PDE-setting:

$$\frac{\partial u}{\partial t}+\frac{1}{2}\sigma_1^2 s_1^2\frac{\partial^2 u}{\partial s_1^2}+\rho\sigma_1\sigma_2 s_1 s_2\frac{\partial^2 u}{\partial s_1\partial s_2}+\frac{1}{2}\sigma_2^2 s_2^2\frac{\partial^2 u}{\partial s_2^2}+rs_1\frac{\partial u}{\partial s_1}+rs_2\frac{\partial u}{\partial s_2}-ru=0. \tag{14}$$

Deliverables

The price for a European spread call option should be computed.

Parameter and problem specifications

The model parameters to use are $r=0.03$, $\sigma_1=\sigma_2=0.15$, $\rho=0.5$, $K=0$, and $T=1$. The payoff function for the European call spread option is $\phi(s_1,s_2)=\max(s_1-s_2-K,0)$.

3. Numerical methods

In Table 1, we display all methods that we have used. We also provide references to the original papers describing the methods. More information about the particular implementations used here can be found at www.it.uu.se/research/project/compfin/benchop.

4. Numerical results

For each benchmark problem, we have decided on three (or five) evaluation points s_i (or (s_i,s_j)). Each method must be tuned such that it delivers a solution $u(s_i)$ with a *relative error* less than

Table 1. List of methods used with abbreviations, marker symbol used in figures, and references.

Abbr.	Symbol	Method	References
Monte Carlo methods			
MC	--✱--	Monte Carlo with Euler-Maruyama in time	[16]
MC-S	--△--	Monte Carlo with analytical solution/Euler-Maruyama/quadratic scheme in time and stratified sampling	[2,16,17,36,42,50]
QMC-S	--×--	Quasi Monte Carlo with analytical solution/Euler-Maruyama/quadratic scheme in time, stratified sampling, and precomputed quasi random numbers	[2,16,17,21,36,42,44,50]
Fourier methods			
FFT	-·□-·	Fourier method with FFTs	[6,31,33]
FGL	-·×-·	Fourier method with Gauss-Laguerre quadrature	[1 (Page 890),7,18,31], [34 (Section 2.1–2.2), 35]
COS	-·○-·	Fourier method based on Fourier cosine series and the characteristic function.	[11,12,52,53]
Finite difference methods			
FD	——	Finite differences on uniform grids with Rannacher smoothed CN in time	[23,51,59,64, Chapter 78]
FD-NU	- - - - -	Finite differences on quadratically refined grids with Rannacher smoothed CN / IMEX-CNAB in time	[22,49,51,55,56]
FD-AD	-·-·-·	Adaptive finite differences with discontinuous Galerkin / BDF-2 in time	[19,22,26,46,47,62]
Radial basis function methods			
RBF	—×—	Global radial basis functions with non-uniform nodes and BDF-2 in time	[19,48]
RBF-FD	—◇—	Radial basis functions generated finite differences with BDF-2 in time	[13,14,19,41,58,63,65]
RBF-PUM	—△—	Radial basis functions partition of unity method with BDF-2 in time	[19,54,57]
RBF-LSML	—✱—	Least-squares multi-level radial basis functions with BDF-2 in time	[19,27,28]
RBF-AD	—○—	Adaptive RBFs with CN in time	[8,24,43]
RBF-MLT	—□—	Multi-level radial basis functions treating time as a spatial dimension	[25,30,60]

10^{-4} in these points. We have not put any restrictions on the error in the rest of the domain. Due to this freedom, some codes have been tuned to narrowly target these points, while others (sometimes automatically) are tuned to achieve an evenly distributed error. Then the codes are run (on the same computer system) and the execution times are recorded. Each code is run four times, and the execution time reported in Tables 2–5 is the average of the last three runs. This is because the first time a MATLAB script is executed in a session it takes a bit longer.

In the tables, we also show the approximate number of correct digits p in the result. This quantity is computed as $p = [-\log_{10} e_r]$, where e_r is the maximum relative error and $[\cdot]$ indicates rounding. The maximum relative error is computed as

$$e_r = \max_i \left| \frac{u(s_i) - u^{\text{ref}}(s_i)}{u^{\text{ref}}(s_i)} \right|.$$

For the Monte Carlo methods where the error is not deterministic, the errors are averaged over the different runs. In some cases, a method was not able to reach a relative error of 10^{-4} within reasonable time (1 hour), but a lower target 10^{-3} was attainable. These results are marked with a * in Tables 2–5. The execution time we report in the tables for Monte Carlo methods is the

Table 2. Problem 1. Computational time to compute a solution u that has a relative error $< 10^{-4}$ at $t = 0$ and $s = 90$, 100, 110 for the standard parameters and at $s = 97, 98, 99$ for the challenging parameters.

	Standard parameters			Challenging parameters		
Method	(a) European	(b) American	(c) Up-and-out	(a) European	(b) American	(c) Up-and-out
MC	*5.7e+02 (3)	–	–	×	–	–
MC-S	1.5e+01 (4)	×	×	1.6e+01 (4)	9.8e−02 (16)	×
QMC-S	5.7e−01 (5)	×	–	6.3e−01 (6)	1.2e+02 (16)	–
FFT	1.3e−03 (6)	–	–	1.3e−03 (7)	–	–
FGL	3.5e−03 (14)	7.6e−01 (5)	–	1.8e−02 (13)	2.2e+00 (16)	–
COS	1.8e−04 (5)	2.7e−02 (4)	1.8e−02 (4)	2.6e−04 (4)	2.3e−01 (5)	2.5e−04 (4)
FD	1.8e−02 (4)	7.6e−02 (4)	2.5e−02 (4)	5.1e+01 (4)	1.1e−02 (11)	5.1e+01 (4)
FD-NU	9.2e−03 (4)	5.8e−02 (4)	1.6e−02 (4)	5.0e−02 (5)	6.0e−03 (11)	5.2e−02 (5)
FD-AD	9.7e−03 (4)	4.3e−02 (4)	9.6e−03 (4)	1.0e+01 (4)	9.1e−03 (4)	2.0e+00 (4)
RBF	6.2e−02 (4)	4.6e+00 (4)	1.4e−01 (4)	7.7e+01 (4)	–	6.6e+01 (4)
RBF-FD	2.9e−01 (4)	1.3e+00 (4)	2.8e−01 (4)	3.3e+01 (5)	9.6e−01 (4)	5.1e+00 (4)
RBF-PUM	2.8e−02 (4)	3.6e+00 (4)	5.4e−02 (4)	3.4e+00 (5)	4.5e+00 (4)	1.9e+00 (4)
RBF-LSML	4.2e−02 (4)	–	3.0e−02 (4)	7.5e+00 (5)	–	–
RBF-AD	7.9e−01 (4)	1.7e+01 (5)	2.4e+01 (5)	3.3e+00 (4)	1.2e+00 (7)	1.6e+01 (4)
RBF-MLT	1.6e+01 (4)	–	2.4e+02 (4)	*3.3e+02 (4)	–	*1.9e+03 (3)

Note: The numbers within parentheses indicate the approximate number of correct digits in the result. A ‘ – ’ indicates not implemented, while ‘ × ’ means implemented, but not accurate.

Table 3. Problem 1. Computational time to compute hedging parameters $\Delta = \frac{\partial u}{\partial s}$, $\Gamma = \frac{\partial^2 u}{\partial s^2}$ and $\mathcal{V} = \frac{\partial u}{\partial \sigma}$ that have a relative error $< 10^{-4}$ at $t = 0$ and $s = 90, 100, 110$ for the standard parameters and at $s = 97, 98, 99$ for the challenging parameters.

	Standard parameters			Challenging parameters		
Method	$\Delta = \frac{\partial u}{\partial s}$	$\Gamma = \frac{\partial^2 u}{\partial s^2}$	$\mathcal{V} = \frac{\partial u}{\partial \sigma}$	$\Delta = \frac{\partial u}{\partial s}$	$\Gamma = \frac{\partial^2 u}{\partial s^2}$	$\mathcal{V} = \frac{\partial u}{\partial \sigma}$
MC-S	3.0e+00 (5)	*5.7e+01 (4)	1.7e+01 (4)	3.3e+00 (5)	×	*1.8e+01 (3)
QMC-S	5.7e−01 (5)	*1.0e+00 (4)	6.3e−01 (5)	6.3e−01 (6)	×	*6.7e−01 (4)
FFT	1.3e−03 (6)	1.2e−03 (5)	2.1e−03 (5)	1.2e−03 (7)	1.2e−03 (5)	2.5e−03 (6)
FGL	3.2e−03 (14)	2.9e−03 (14)	2.9e−03 (14)	1.8e−02 (14)	1.8e−02 (11)	2.2e−02 (11)
COS	1.8e−04 (4)	2.4e−04 (5)	2.6e−04 (5)	2.8e−04 (4)	4.4e−04 (5)	4.6e−04 (5)
FD	1.9e−02 (5)	1.4e−02 (5)	2.8e−02 (4)	4.9e+01 (5)	2.5e+02 (4)	5.0e+02 (5)
FD-NU	7.8e−03 (4)	8.9e−03 (4)	1.8e−02 (4)	6.5e−02 (4)	5.0e−01 (4)	1.6e+00 (4)
FD-AD	1.0e−02 (4)	9.9e−03 (4)	2.4e−02 (4)	1.0e+01 (4)	4.9e+01 (4)	9.0e+01 (4)
RBF	6.6e−02 (4)	6.9e−02 (4)	7.9e−02 (4)	9.3e+01 (4)	3.1e+02 (5)	7.0e+02 (4)
RBF-FD	3.0e−01 (4)	2.5e−01 (4)	5.6e−01 (4)	3.4e+01 (4)	3.7e+01 (4)	1.0e+02 (4)
RBF-PUM	2.7e−02 (4)	3.2e−02 (4)	1.0e−01 (4)	3.1e+00 (5)	6.2e+00 (4)	1.1e+01 (4)
RBF-LSML	1.3e−01 (4)	2.7e−01 (4)	7.1e−02 (5)	–	–	–
RBF-AD	3.4e+00 (4)	3.4e+01 (4)	5.0e+01 (4)	3.6e+00 (5)	1.8e+01 (4)	2.1e+01 (4)
RBF-MLT	1.8e+01 (4)	1.8e+01 (4)	*3.2e+01 (4)	4.2e+02 (4)	*3.4e+02 (3)	×

Note: The numbers within parentheses indicate the approximate number of correct digits in the result. A ‘ – ’ indicates not implemented, while ‘ × ’ means implemented, but not accurate.

time to compute the result for one evaluation point, whereas for other methods it is the time to compute the result for all evaluation points.

In order to see how the errors behave away from the evaluation points, solutions are also plotted for a range of values in Figures 1–8. The figures show the *absolute errors* evaluated at the integer values between $s = 60$ and $s = 160$. No figure is shown for the second local volatility case in Problem 3, because there the local volatility result is only valid for a particular S_0, not

Table 4. Problems 2 and 3. Computational time to compute a solution u that has a relative error $< 10^{-4}$ at $t = 0$ and $s = 90, 100, 110$.

	Discrete dividends				Local volatility			
Method	European call		American call		Smooth		Implied	
MC-S	*6.0e + 01	(3)	–		×		×	
QMC-S	1.6e + 00	(4)	–		–		–	
FFT	1.5e − 03	(7)	–		–		–	
FGL	3.6e − 03	(14)	8.1e − 02	(6)	–		–	
COS	8.8e − 04	(5)	2.4e − 03	(4)	1.7e − 02	(4)	–	
FD	2.0e − 02	(4)	1.5e − 02	(4)	2.2e − 02	(4)	1.2e + 00	(4)
FD-NU	1.6e − 02	(4)	1.5e − 02	(4)	3.7e − 02	(4)	8.9e − 01	(4)
FD-AD	2.1e − 02	(4)	2.3e − 02	(5)	3.6e − 02	(4)	8.8e − 01	(4)
RBF	2.3e − 01	(4)	1.1e − 01	(4)	5.5e − 02	(4)	2.4e + 00	(4)
RBF-FD	4.2e − 01	(4)	2.7e + 00	(4)	2.2e + 01	(4)	1.1e + 02	(4)
RBF-PUM	3.3e − 02	(4)	3.0e − 02	(4)	1.4e − 01	(4)	9.0e − 01	(4)
RBF-LSML	5.0e − 01	(4)	1.2e + 00	(4)	1.7e − 01	(4)	4.0e + 00	(4)

Note: The numbers within parentheses indicate the approximate number of correct digits in the result. A ‘ – ’ indicates not implemented, while ‘ × ’ means implemented, but not accurate.

Table 5. Problems 4, 5, and 6. Computational time to compute a solution u that has a relative error $< 10^{-4}$ at $t = 0$ and $s = 90, 100, 110$ for the Heston and Merton models, and to compute a solution u that has a relative error $< 10^{-4}$ at $t = 0$ and $(s_1, s_2) = (100, 90), (100, 100), (100, 110), (90, 100), (110, 100)$ for the spread option.

Method	Heston		Merton		Spread	
MC-S	×		1.6e + 01	(4)	*2.9e + 01	(4)
QMC-S	–		3.4e − 01	(4)	1.8e + 00	(5)
FFT	3.3e − 03	(5)	2.1e − 03	(5)	–	
FGL	3.8e − 03	(14)	3.1e − 03	(13)	3.1e − 03	(14)
COS	3.4e − 04	(4)	2.2e − 04	(4)	1.5e − 03	(4)
FD	–		–		–	
FD-NU	4.3e + 00	(4)	1.4e − 01	(4)	7.4e + 01	(4)
FD-AD	–		–		4.7e + 01	(4)
RBF	1.9e + 01	(4)	–		7.7e + 01	(4)
RBF-FD	–		–		2.2e + 03	(4)
RBF-PUM	4.3e + 00	(4)	–		1.3e + 01	(5)

Note: The numbers within parentheses indicate the approximate number of correct digits in the result. A ‘ – ’ indicates not implemented, while ‘ × ’ means implemented, but not accurate.

over a range. The vertical axis range in the figures is adjusted to the values that are plotted, but the lower limit is not allowed to be lower than 10^{-20}. Errors falling below that value are not visible in the figures.

The experiments have been performed on the Tintin cluster at Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX), Uppsala University. The cluster consists of 160 dual AMD Opteron 6220 (Bulldozer) nodes. All codes are implemented (serially) in MATLAB. The names of the respective codes for each problem are indicated by the boldfaced heading over each (group of) plot(s). This generic name is then combined with an acronym for the particular method as for example `BSeuCallUI_RBF.m`.

5. Discussion

Monte Carlo methods. MC methods are easy to implement in any number of dimensions, but the slow convergence rate, $\mathcal{O}(1/\sqrt{N})$ for standard MC, makes it computationally expensive to reach

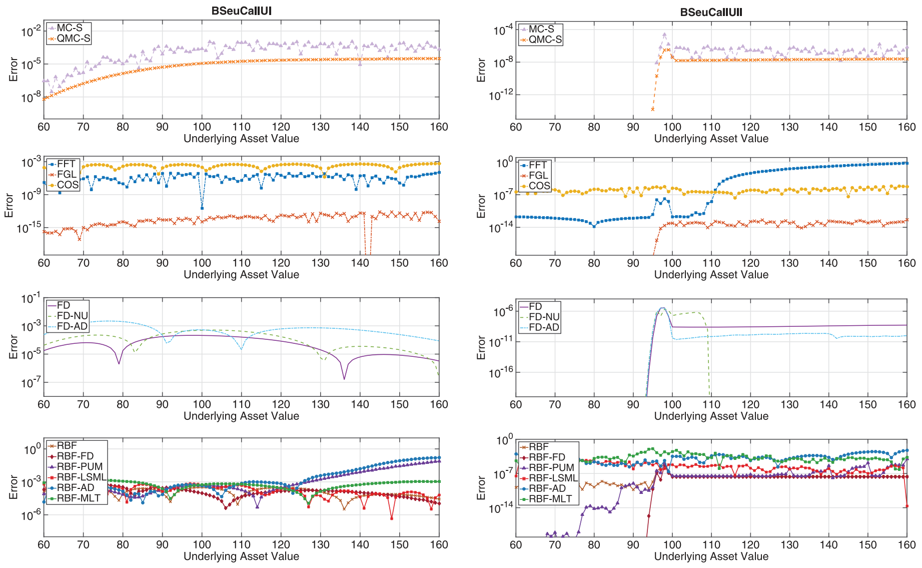


Figure 1. Problem 1(a) European call option. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute error in the solution u for $t = 0$ and $60 \le s \le 160$ when the relative error in u is less than 10^{-4} at $t = 0$ and $s = 90, 100, 110$ for standard parameters and $s = 97, 98, 99$ for challenging parameters.

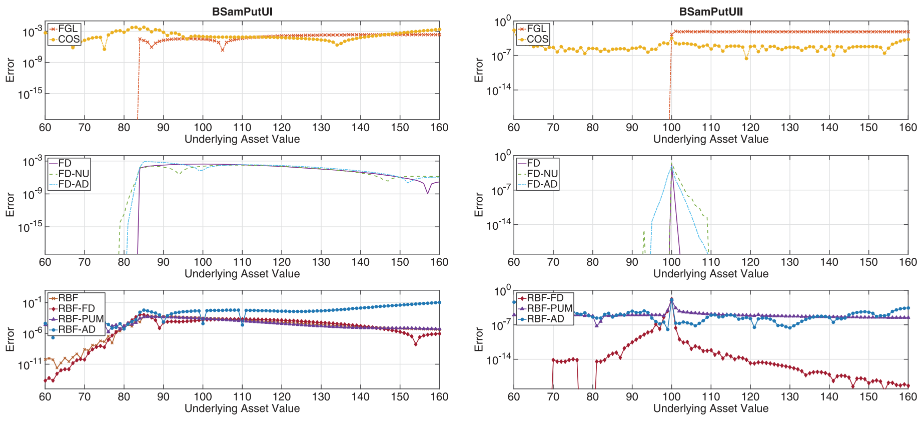


Figure 2. Problem 1(b) American put option. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute error in the solution u for $t = 0$ and $60 \le s \le 160$ when the relative error in u is less than 10^{-4} at $t = 0$ and $s = 90, 100, 110$ for standard parameters and $s = 97, 98, 99$ for challenging parameters.

the requested tolerance of 10^{-4}. The most challenging problems for the MC methods were the path dependent options, the hedging parameter Γ, and the local volatility.

Because MC methods scale linearly with the number of dimensions, they are increasingly competitive in higher dimensions. Furthermore, there are a lot of specialized techniques that can be applied to improve performance. An example of this is the QMC-S method that is comparable to some of the other methods already in one dimension, and the fastest method apart from the Fourier methods for the two-dimensional spread option.

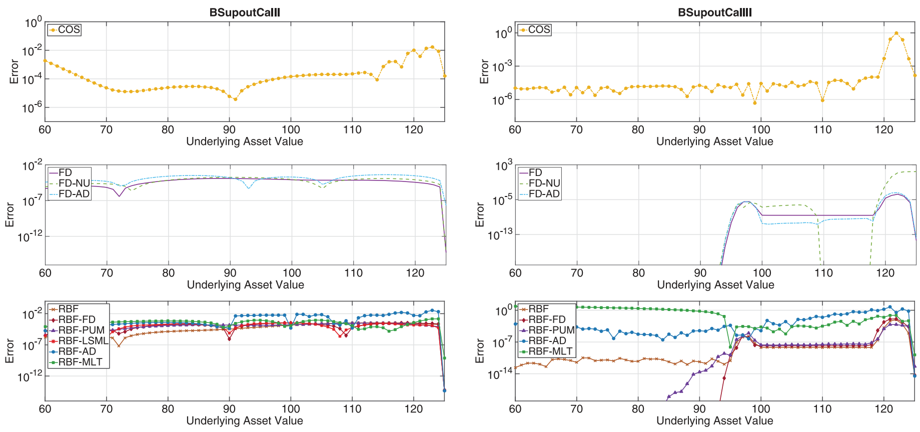


Figure 3. Problem 1(c) Up-and-out call option. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute error in the solution u for $t = 0$ and $60 \leq s \leq 125$ when the relative error in u is less than 10^{-4} at $t = 0$ and $s = 90$, 100, 110 for standard parameters and $s = 97$, 98, 99 for challenging parameters.

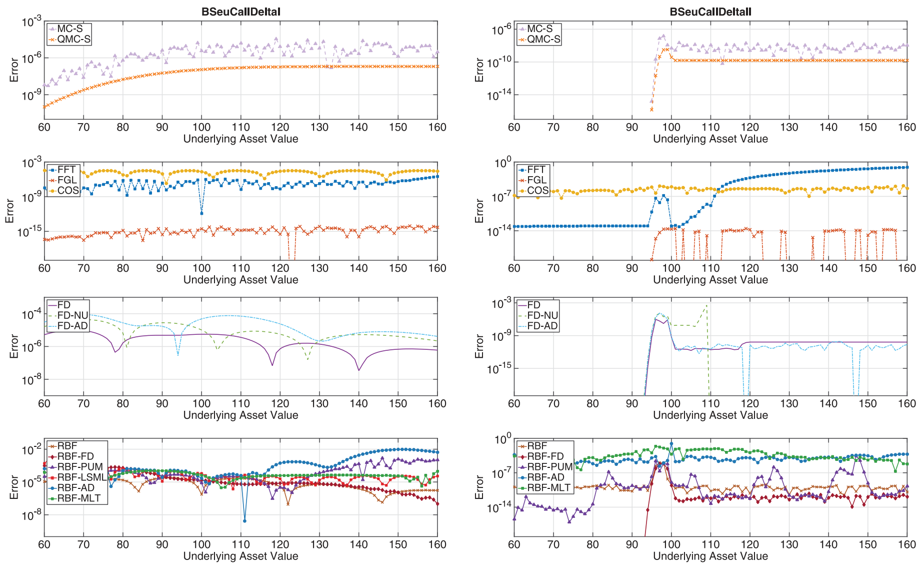


Figure 4. Problem 1(a) European call option. Errors in the hedging parameter Δ. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute errors in the hedging parameters for $t = 0$ and $60 \leq s \leq 160$ when the relative error is less than 10^{-4} at $t = 0$ and $s = 90$, 100, 110 for standard parameters and $s = 97$, 98, 99 for challenging parameters.

Fourier methods. Fourier methods (FM) rely on the availability of the characteristic function (ChF) of the underlying stochastic process. These are available for all problems except Problem 3, local volatility. However, in the recent publication [53], the stochastic process is approximated by a second order weak Taylor scheme, for which there exists an analytic solution for the ChF. This method was applied to the smooth local volatility function, but could not easily be used for the implied local volatility. Apart from Problem 3, the problems that were most challenging for the FM were the American and Up-and-out options.

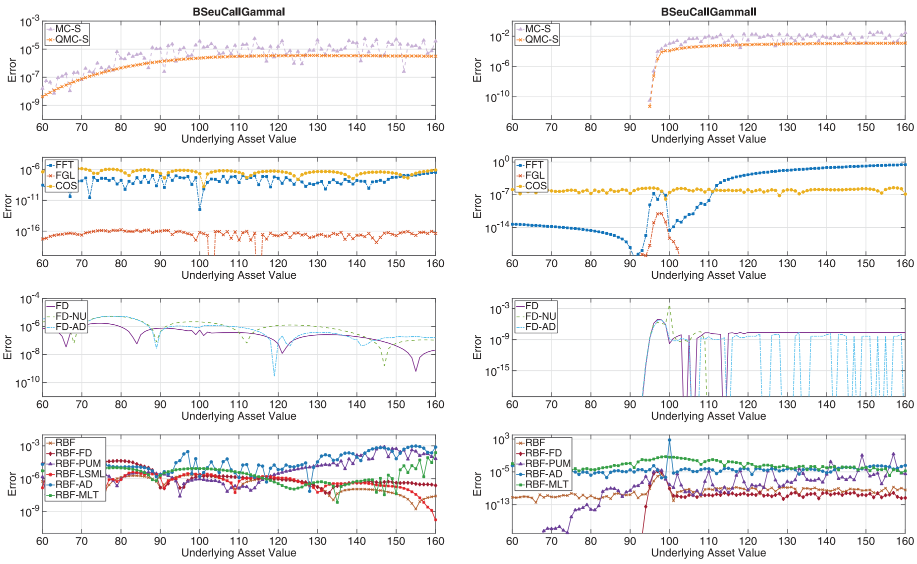


Figure 5. Problem 1(a) European call option. Errors in the hedging parameter Γ. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute errors in the hedging parameters for $t = 0$ and $60 \leq s \leq 160$ when the relative error is less than 10^{-4} at $t = 0$ and $s = 90, 100, 110$ for standard parameters and $s = 97, 98, 99$ for challenging parameters.

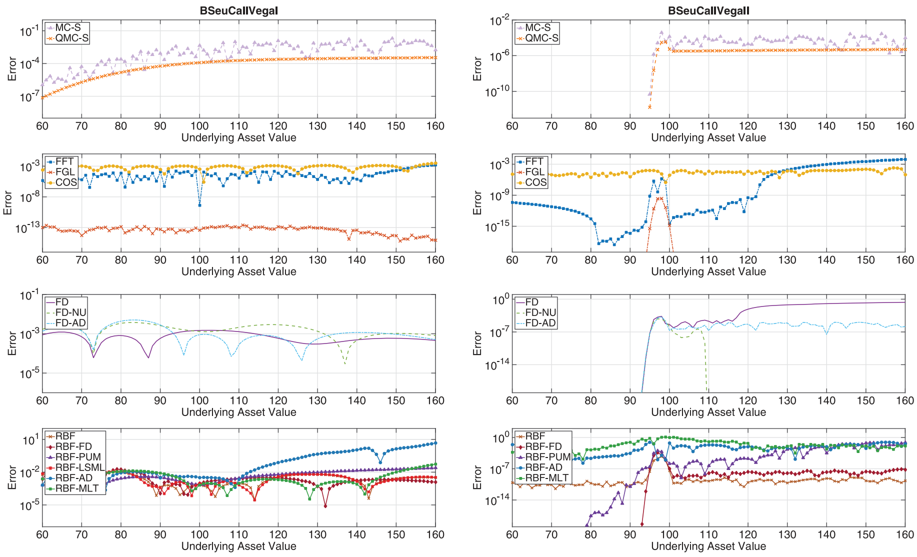


Figure 6. Problem 1(a) European call option. Errors in the hedging parameter $\mathcal{V}$. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute errors in the hedging parameters for $t = 0$ and $60 \leq s \leq 160$ when the relative error is less than 10^{-4} at $t = 0$ and $s = 90, 100, 110$ for standard parameters and $s = 97, 98, 99$ for challenging parameters.

The FM are all very fast and especially the FGL-method is also highly accurate. The fastest FM, the COS method, is the overall fastest method in all cases but two. The competitiveness of the FM is even more pronounced for the two-dimensional problems, the Heston model and the spread option.

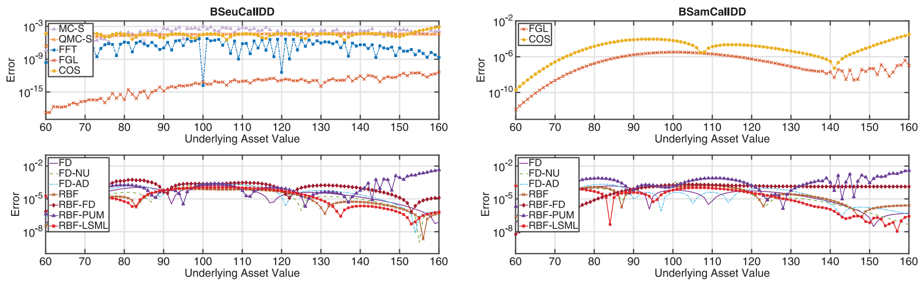


Figure 7. Problem 2(a) European call with dividends (left) and 2(b) American call with dividends (right). Error in the solution u for $t = 0$ and $60 \leq s \leq 160$ when the relative error in u is less than 10^{-4} at $t = 0$ and $s = 90, 100, 110$.

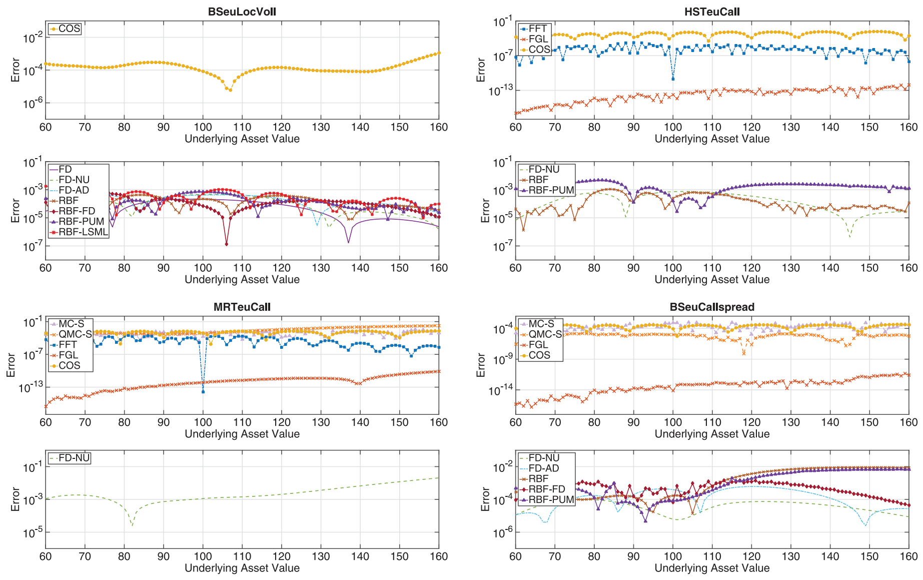


Figure 8. Problem 3 Local volatility, smooth function (top left), Problem 4 Heston (top right), Problem 5 Merton (bottom left), and Problem 6 Spread option (bottom right). Absolute error in the solution u for $t = 0$ and $60 \leq s \leq 160$ when the relative error in u is less than 10^{-4} at $t = 0$ and $s = 90, 100, 110$ for Local volatility and Merton, and for Heston with variance $v = 0.0225$. For the spread option, the error is measured in $(s_1, s_2) = (100, 90), (100, 100), (100, 110), (90, 100), (110, 100)$, and is plotted for $60 \leq s_1 \leq 160$, and $s_2 = 100$.

Finite difference methods. FD methods rely on the use of structured (possibly non-uniform) grids and are straightforward to implement. The FD-NU is the only BENCHOP method that solved all the problems. The computational times are low in all cases, and for two problems FD-NU or FD-AD is the overall fastest method. For the FD methods in general the challenging parameter set in Problem 1 is the most difficult feature to handle.

For Problem 1 the usage of a nonuniform grid, (FD-NU and FD-AD), is superior to using a uniform grid (FD) but for Problems 2 and 3, using a uniform grid is sometimes faster. The FD method has not been implemented for the two-dimensional problems, but we believe that FD-NU and FD-AD would be faster than FD in these cases thanks to the possibility of local refinement.

Radial basis function methods. RBF methods are flexible with respect to node locations and choice of basis function. This makes it possible to tune the methods to achieve well for particular targets, but it can also be hard to make a good choice. Problem 5, Merton jump diffusion has not been implemented in any RBF method, but this could be done. The problems that are challenging for RBF methods have non-smooth solutions or very sharp gradients, such as American options and the challenging parameter set in Problem 1.

The RBF methods as a group are slower than the FD methods for the one-dimensional problems. However, the results of the fastest RBF method, RBF-PUM, is in the favorable cases of the same order as those of the FD methods. In two dimensions the RBF-PUM method is as fast as or faster than the implemented FD methods. Potentially, the RBF-PUM method will be even more competitive in higher dimensions.

Acknowledgments

We would like to thank Institut Mittag-Leffler for supporting the workshop we organized on 'Mathematical and Numerical modeling in Finance' (http://www.mittag-leffler.se/?q = 0609), where the idea for the BENCHOP project was conceived and where the work was initiated.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) under Project snic2014-3-73.

References

[1] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 9th printing*, Dover publications, New York, 1972.
[2] L.B.G. Andersen, *Efficient simulation of the Heston stochastic volatility model*, Tech. rep., 2007.
[3] T. Björk, *Arbitrage Theory in Continuous Time*, 3rd ed., Oxford University Press, New York, 2009.
[4] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, J. Polit. Econ. 81 (1973), pp. 637–654.
[5] P. Carr, R. Jarrow, and R. Myneni, *Alternative characterizations of American put options*, Math. Finance 2 (1992), pp. 87–106.
[6] P. Carr and D. Madan, *Option valuation using the fast Fourier transform*, J. Comput. Finance 2 (1999), pp. 61–73.
[7] P. Carr, H. Geman, D. Madan, L. Wu, and M. Yor, *Option Pricing Using Integral Transforms*, 2003, a slideshow presentation available at: http://www.math.nyu.edu/research/carrp/papers/pdf/integtransform.pdf.
[8] R. Chan, *Pricing options under jump-diffusion models by adaptive radial basis functions*, Tech. rep., University of Bath, Department of Economics, 2010, Bath Economics Research Working papers; 06/10.
[9] R. Cont, *Empirical properties of asset returns: stylized facts and statistical issues*, Quant. Finance 1 (2001), pp. 223–236.
[10] B. Dupire, *Pricing with a smile*, Risk 7 (1994), pp. 18–20.
[11] F. Fang and C.W. Oosterlee, *A novel pricing method for European options based on Fourier-cosine series expansions*, SIAM J. Sci. Comput. 31 (2008), pp. 826–848.
[12] F. Fang and C.W. Oosterlee, *Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions*, Numer. Math. 114 (2009), pp. 27–62.
[13] N. Flyer, E. Lehto, and S. Blaise, *A guide to RBF-generated finite differences for nonlinear transport: Shallow water simulations on a sphere*, J. Comput. Phys. 231 (2012), pp. 4078–4095. Available at http://www.sciencedirect.com/science/article/pii/S0021999112000587.
[14] B. Fornberg and E. Lehto, *Stabilization of RBF-generated finite difference methods for convective PDEs*, J. Comput. Phys. 230 (2011), pp. 2270–2285. Available at http://www.sciencedirect.com/science/article/pii/S0021999110006789.
[15] J. Gatheral and A. Jacquier, *Arbitrage-free SVI volatility surfaces*, Quant. Finance 14 (2014), pp. 59–71.
[16] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Applications of mathematics: stochastic modelling and applied probability, Springer, 2004. Available at http://books.google.se/books?id = e9GWUsQkPNMC.
[17] P. Glasserman and J. Staum, *Conditioning on one-step survival for barrier option simulations*, Oper. Res. 49 (2001), pp. 923–937. Available at http://dx.doi.org/10.1287/opre.49.6.923.10018.
[18] G.H. Golub and J.H. Welsch, *Calculation of Gauss quadrature rules*, Math. Comp. 23 (1969), pp. 221–230 + s1–s10.
[19] E. Hairer, S. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Solving Ordinary Differential Equations, Springer, 2008. Available at https://books.google.se/books?id = cfZDAAAAQBAJ.

[20] S.L. Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Rev. Financial Stud. 6 (1993), pp. 327–343.
[21] L.J. Höök, T. Johnson, and T. Hellsten, *Randomized quasi-Monte Carlo simulation of fast-ion thermalization*, Comput. Sci. Disc. 5 (2012), pp. 014010. Available at http://stacks.iop.org/1749-4699/5/i = 1/a = 014010.
[22] S. Ikonen and J. Toivanen, *Operator splitting methods for American option pricing*, Appl. Math. Lett. 17 (2004), pp. 809–814. Available at http://dx.doi.org/10.1016/j.aml.2004.06.010.
[23] S. Ikonen and J. Toivanen, *Pricing American options using LU decomposition*, Appl. Math. Sci. 1 (2007), pp. 2529–2551.
[24] A. Iske and J. Behrens, *Grid-free adaptive semi-Lagrangian advection using radial basis functions*, Comput. Math. Appl. 43 (2002), pp. 319–327.
[25] A. Iske and J. Levesley, *Multilevel scattered data approximation by adaptive domain decomposition*, 39 (2005), pp. 187–198.
[26] E. Larsson, *Option pricing using the discontinuous Galerkin method for time-integration*, Tech. rep., Department of Information Technology, Uppsala University, 2013.
[27] E. Larsson and S.M. Gomes, *A Least Squares Multi-Level Radial Basis Function Method with Applications in Finance*, 2015, manuscript in preparation.
[28] E. Larsson, S. Gomes, A. Heryudono, and A. Safdari-Vaighani, *Radial basis function methods in computational finance*, in *Proc. CMMSE 2013*, Almería, Spain, 12pp., 2013.
[29] M. Lauko and D. Ševčovič, *Comparison of numerical and analytical approximations of the early exercise boundary of American put options*, ANZIAM J. 51 (2010), pp. 430–448. Available at http://dx.doi.org/10.1017/S1446181110000854.
[30] Q.T. Le Gia, I.H. Sloan, and H. Wendland, *Multiscale approximation for functions in arbitrary Sobolev spaces by scaled radial basis functions on the unit sphere*, Appl. Comput. Harmon. Anal. 32 (2012), pp. 401–412.
[31] R.W. Lee, *Option pricing by transform methods: Extensions, unification, and error control*, J. Comput. Finance 7 (2004), pp. 51–86, an augmented version is available at http://www.math.uchicago.edu/~rl/dft.pdf.
[32] E. Lindström, H. Madsen, and J. Nielsen, *Statistics for Finance*, Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis, 2015.
[33] E. Lindström and H. Wu, *A Fourier method for valuation of options under parameter uncertainty*, submitted for publication .
[34] E. Lindström, J. Ströjby, M. Brodén, M. Wiktorsson, and J. Holst, *Sequential calibration of options*, Tech. Rep. 2006:21, Centre for Mathemtical Sciences, Lund University, 2006.
[35] E. Lindström, J. Ströjby, M. Brodén, M. Wiktorsson, and J. Holst, *Sequential calibration of options*, Comput. Statist. Data Anal. 52 (2008), pp. 2877–2891.
[36] F.A. Longstaff and E.S. Schwartz, *Valuing American options by simulation: A simple least-squares approach*, Rev. Financ. Stud. 14 (2001), pp. 113–147. Available at http://rfs.oxfordjournals.org/content/14/1/113.abstract.
[37] R. Lord and C. Kahl, *Optimal fourier inversion in semi-analytical option pricing*, J. Comput. Finance 10 (2007), pp. 1–30.
[38] W. Margrabe, *The value of an option to exchange one asset for another*, J. Finance 33 (1978), pp. 177–186.
[39] R.C. Merton, *Theory of rational option pricing*, Bell J. Econom. Man. Sci. 4 (1973), pp. 141–183.
[40] R.C. Merton, *Option pricing when underlying stock returns are discontinuous*, J. Financial Econ. 3 (1976), pp. 125–144.
[41] S. Milovanović and L. von Sydow, *Radial basis function generated finite differences for option pricing problems*, 2015. Manuscript in preparation.
[42] K.-S. Moon, *Efficient Monte Carlo algorithm for pricing barrier options*, Commun. Korean Math. Soc. 23 (2008), pp. 285–294.
[43] S.L. Naqvi, *Adaptive radial basis function interpolation for time-dependent partial differential equation*, Ph.D. thesis, University of Leicester, 2013.
[44] H. Niederreiter, *Constructions of (t,m,s)–nets and (t,s)–sequences*, Finite Fields Appl. 11 (2005), pp. 578–600. Available at http://www.sciencedirect.com/science/article/pii/S1071579705000043.
[45] P. Nystrup, H. Madsen, and E. Lindström, *Stylized Facts of Financial Time Series and Hidden Markov Models in Continuous Time*, Quant. Finance, 2015.
[46] J. Persson and L. von Sydow, *Pricing European multi-asset options using a space-time adaptive FD-method*, Comput. Vis. Sci. 10 (2007), pp. 173–183.
[47] J. Persson and L. von Sydow, *Pricing American options using a space-time adaptive finite difference method*, Math. Comput. Simulation 80 (2010), pp. 1922–1935.
[48] U. Pettersson, E. Larsson, G. Marcusson, and J. Persson, *Improved radial basis function methods for multi-dimensional option pricing*, J. Comput. Appl. Math. 222 (2008), pp. 82–93.
[49] O. Pironneau, *On the transport-diffusion algorithm and its applications to the Navier-Stokes equations*, Numer. Math. 38 (1981/82), pp. 309–332. Available at http://dx.doi.org/10.1007/BF01396435.
[50] E. Platen and N. Bruti-Liberati, *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*, Stochastic Modelling and Applied Probability, Vol. 64, Springer-Verlag, Berlin, 2010.
[51] R. Rannacher, *Finite element solution of diffusion problems with irregular data*, Numer. Math. 43 (1984), pp. 309–327.
[52] M.J. Ruijter and C.W. Oosterlee, *Two-dimensional Fourier cosine series expansion method for pricing financial options*, SIAM J. Sci. Comput. 34 (2012), pp. B642–B671.

[53] M.J. Ruijter and C.W. Oosterlee, *Numerical Fourier method and second-order Taylor scheme for backward SDEs in finance*, Working paper available at SSRN, submitted for publication, 2014.
[54] A. Safdari-Vaighani, A. Heryudono, and E. Larsson, *A radial basis function partition of unity collocation method for convection–diffusion equations arising in financial applications*, J. Sci. Comp. (2014), pp. 1–27.
[55] S. Salmi and J. Toivanen, *IMEX schemes for pricing options under jump-diffusion models*, Appl. Numer. Math. 84 (2014), pp. 33–45. Available at http://dx.doi.org/10.1016/j.apnum.2014.05.007.
[56] S. Salmi, J. Toivanen, and L. von Sydow, *An IMEX-scheme for pricing options under stochastic volatility models with jumps*, SIAM J. Sci. Comput. 36 (2014), pp. B817–B834. Available at http://dx.doi.org/10.1137/130924905.
[57] V. Shcherbakov and E. Larsson, *Radial basis function partition of unity methods for pricing vanilla basket options*, Tech. Rep. 2015-001, Division of Scientific Computing, Department of Information Technology, Uppsala University, 2015.
[58] A.I. Tolstykh and D.A. Shirobokov, *On using radial basis functions in a 'finite difference mode' with applications to elasticity problems*, Comput. Mech. 33 (2003), pp. 68–79.
[59] A. Toropov, D. Ivanov, and Y. Shpolyanskiy, *Application of theoretical options pricing to algorithmic trading strategies*, Sci. Tech. J. Infor. Tech. Mech. Opt. (2013), pp. 136–141.
[60] A. Townsend and H. Wendland, *Multiscale analysis in Sobolev spaces on bounded domains with zero boundary values*, IMA J. Numer. Anal. 33 (2013), pp. 1095–1114.
[61] R. Villiger, *Valuation of American call options*, Wilmott Magazine 3 (2006), pp. 64–67.
[62] L. von Sydow, *On discontinuous Galerkin for time integration in option pricing problems with adaptive finite differences in space*, in *Numerical Analysis and Applied Mathematics: ICNAAM 2013*, *AIP Conference Proceedings*, vol. 1558, American Institute of Physics (AIP), Melville, NY, 2013, pp. 2373–2376.
[63] X. Wang, *Finite differences based on radial basis functions to price options*, Master's thesis, Uppsala University, 2014.
[64] P. Wilmott, *On Quantitative Finance*, 2nd ed., John Wiley & Sons Ltd., Chichester, 2006.
[65] G.G.B. Wright and B. Fornberg, *Scattered node compact finite difference-type formulas generated from radial basis functions*, J. Comput. Phys. 212 (2006), pp. 99–123. Available at http://www.sciencedirect.com/science/article/pii/S0021999105003116.

Appendix 1. The methods used for computing the reference values used in the comparisons

For many of the benchmark problems here, there are analytical or semi-analytical solutions. For these cases, we state the closed form expressions. For the cases lacking analytical solutions, we describe the numerical method that was used for accurate enough computation of a reference solution. MATLAB codes for each problem are available at the BENCHOP web page.

A.1 *Problem 1: The Black–Scholes–Merton model for one underlying asset*

For the *European call* option, the closed form expression for the option price is given in [4]

$$\Pi^c_{BS}(t,S,K,T,r,\sigma^2) = SN\left(d^+\left(\frac{S}{K},T-t\right)\right) - Ke^{-r(T-t)}N\left(d^-\left(\frac{S}{K},T-t\right)\right), \tag{A1}$$

where

$$d^+(x,y) = \frac{1}{\sigma\sqrt{y}}\left(\log(x) + \left(r+\frac{\sigma^2}{2}\right)y\right), \tag{A2}$$

$$d^-(x,y) = \frac{1}{\sigma\sqrt{y}}\left(\log(x) + \left(r-\frac{\sigma^2}{2}\right)y\right), \tag{A3}$$

and where $N(x)$ is the cumulative distribution function for the standard normal distribution. The hedging parameters can be found through differentiation, leading to

$$\Delta_{BS} = \frac{\partial \Pi^c_{BS}}{\partial S} = N\left(d^+\left(\frac{S}{K},T-t\right)\right), \tag{A4}$$

$$\Gamma^c_{BS} = \frac{\partial^2 \Pi^c_{BS}}{\partial S^2} = \frac{\phi(d^+(S/K,T-t))}{S\sqrt{(T-t)\sigma^2}}, \tag{A5}$$

$$\mathcal{V}^c_{BS} = \frac{\partial \Pi^c_{BS}}{\partial \sigma} = S\phi\left(d^+\left(\frac{S}{K},T-t\right)\right)\sqrt{T-t}, \tag{A6}$$

where $\phi(x) = (2\pi)^{-1/2}\exp(-x^2/2)$ is the density of the standard normal distribution.

For the *American put* option, there is no closed form solution. Different analytical and semi-analytical approximations to the location of the early exercise boundary are analyzed and compared in [29]. Here, we use a relation from [5, Theorem 1.1], where it is shown that the American put price can be decomposed into a European put price and the early exercise premium. The general European put price is defined by

$$\Pi^p_{BS}(t,S,K,T,r,\sigma^2) = -SN\left(-d^+\left(\frac{S}{K},T-t\right)\right) + K\mathrm{e}^{-r(T-t)}N\left(-d^-\left(\frac{S}{K},T-t\right)\right), \tag{A7}$$

The American put price becomes

$$\Pi^p_{BS-A}(t,S,K,T,r,\sigma^2) = \Pi^p_{BS}(t,b_p(t),K,T,r,\sigma^2) + rK\int_t^T \mathrm{e}^{-r(u-t)}N\left(-d^-\left(\frac{b_p(t)}{b_p(u)},u-t\right)\right)\mathrm{d}u, \tag{A8}$$

where $b_p(t)$, $0 \le t < T$ is the (unknown) optimal exercise level at time t, and $b_p(T) = K$. The location of the exercise boundary is determined by solving the following non-linear integral equation numerically:

$$\Pi^p_{BS-A}(t,S,K,T,r,\sigma^2) = K - b_p(t). \tag{A9}$$

This equation is solved by an implicit trapezoidal method, where the implicit step due to the monotonicity in $b_p(t)$ can be found by binary search. This leads to a robust albeit slow method for obtaining the optimal exercise level b_p on a fine time grid from $t = T$ to $t = 0$. Then, for arbitrary initial stock values satisfying $S(0) > b_p(0)$, (A8) provides the option value. For $S(0) \le b_p(0)$ the value is set to the exercise value $K - S(0)$.

For the *barrier call up–and–out* option, a closed form expression for the option price can be found in [3, Theorem 18.12 p. 271].

$$\begin{aligned}\Pi^c_{BS-UO}(t,S,K,T,r,\sigma^2) &= \Pi^c_{BS}(t,S,K,T,r,\sigma^2) - \Pi^c_{BS}(t,S,B,T,r,\sigma^2)\\ &\quad - \left(\frac{B}{S}\right)^{2r/\sigma^2-1}\left(\Pi^c_{BS}\left(t,\frac{B^2}{S},K,T,r,\sigma^2\right) - \Pi^c_{BS}\left(t,\frac{B^2}{S},B,T,r,\sigma^2\right)\right).\end{aligned} \tag{A10}$$

A.2 *Problem 2: The Black–Scholes–Merton model with discrete dividends*

For the *European call* option with one proportional discrete dividend payment, there is a closed form solution [3, Proposition 16.6 p. 235]. Assuming that the dividend is paid out at time τ, where $t < \tau < T$, we have

$$\Pi^c_{BS-EDD}(t,S,K,T,r,\sigma^2) = \Pi^c_{BS}(t,S(1-D),K,T,r,\sigma^2). \tag{A11}$$

Note that the option price is independent of when during the contract period the dividend occurs.

The *American Call* option with one dividend payment at $\tau = \alpha T$, where $0 < \alpha < 1$, can be valuated semi analytically [61].

$$\begin{aligned}\Pi^c_{BS-ADD}(0,S,K,T,r,\sigma,D,\alpha) &= (1-D)SN_2\left(-d^+\left(\frac{S}{S^*_\tau},\alpha T\right), d^+\left(\frac{(1-D)S}{K},T\right), -\sqrt{\alpha}\right)\\ &\quad - K\mathrm{e}^{-rT}N_2\left(-d^-\left(\frac{S}{S^*_\tau},\alpha T\right), d^-\left(\frac{(1-D)S}{K},T\right), -\sqrt{\alpha}\right)\\ &\quad + \Pi^c_{BS}((1-\alpha)T,S,S^*_\tau,T,r,\sigma^2),\end{aligned} \tag{A12}$$

where $N_2(x,y,\rho)$ is the cumulative distribution function for the bi-variate normal distribution with zero mean and covariance matrix

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

The above formula depends on the unknown variable S^*_t, which can be estimated from the following nonlinear problem:

$$S^*_\tau - K = \Pi^c_{BS}(T-\tau,(1-D)S^*_\tau,K,T,r,\sigma^2).$$

A.3 *Problem 3: The Black–Scholes–Merton model with local volatility*

For a general local volatility function, there is no closed form solution. The reference values have in this case been computed to high accuracy by a selection of the contributed deterministic methods with different numerical approximations in space, different treatments of the boundary, and different approximations in time. In this way, we can be reasonably certain that numerical bias has been eliminated.

A.4 *Problem 4: The Heston model for one underlying asset*

The (almost exact) Heston price is computed through inverse Fourier transform using the Gauss-Laguerre quadrature method (FGL) with 1000 quadrature points in combination with an optimized choice of integration path in the complex plane stock value for stock value. The integration path is chosen so that is goes through the uniquely given saddle point of the integrand (see [34,37]).

A.5 *Problem 5: The Merton jump diffusion model for one underlying asset*

From [40] we have that the price under Merton jump diffusion is given by a weighted sum of modified Black-Scholes prices

$$\Pi^c_{ME}(t,S,K,T,r,\lambda,\gamma,\delta^2,\sigma^2) = \sum_{n=0}^{\infty} \mathrm{e}^{-\lambda(T-t)} \frac{(\lambda(T-t))^n}{n!} \Pi^c_{BSE}\left(t,(\xi+1)^n \mathrm{e}^{-\lambda(T-t)\xi} S, K, T, \sigma^2 + \delta^2 \frac{n}{T-t}\right), \quad \text{(A13)}$$

where $\xi = \mathrm{e}^{\gamma+\delta^2/2} - 1$. By using a few hundred terms, a highly accurate price approximation can be computed.

A.6 *Problem 6: The Black–Scholes–Merton model for two underlying assets*

For the particular case of $K = 0$, a closed form solution for the European call spread option is given in [38] as

$$\Pi^c_{BS-SO}(t,S_1,S_2,T,\sigma_1,\sigma_2,\rho) = \Pi^c_{BS}(t,S_1,S_2,T,0,\sigma_1^2 - 2\rho\sigma_1\sigma_2 + \sigma_2^2). \quad \text{(A14)}$$

Appendix 2. The second local volatility function used in Problem 3

The second local volatility function is based on a stochastic volatility inspired (SVI) parametrization [15] of the implied volatility surface. This approach to local volatility is widely used by practitioners, because it is relatively easy to calibrate to data, and there are techniques to eliminate different modes of arbitrage.

A.7 *The given SVI parametrization*

The global total implied variance surface in terms of the time of maturity T and the log moneyness $x = \log(K/F_T)$ in the forward price $F_T = S_0\mathrm{e}^{rT}$ is given by

$$w_g(T,x) = a + \frac{r-\ell}{2}(x-m) + \frac{r+\ell}{2}\sqrt{(x-m)^2 + p^2}, \quad \text{(A15)}$$

where a, r, ℓ, m, p are parameters that depend on T. Here, the calibrated parameters are given by

$$\begin{aligned} &a = 0.01 + 0.03\sqrt{T+0.04}, \quad r = 0.06(1-0.87\sqrt{T}),\\ &\ell = 0.31(1-0.7\sqrt{T}), \quad m = 0.03 + 0.01T,\\ &p = 0.15(0.4 + 0.6\sqrt{T+0.04}). \end{aligned}$$

A.8 *Constructing the local volatility surface*

In order to compute the local surface, we apply a transformation that corresponds to Dupire's formula [10] for the SVI parametrization.

$$w_{\text{local}}(x,T) = \frac{w_g(T,x) + T\dfrac{\partial w_g}{\partial T}(T,x)}{\left(1 - \dfrac{x\frac{\partial w_g}{\partial x}}{2w_g}\right)^2 - \left(\dfrac{\frac{\partial w_g}{\partial x}}{2w_g}\right)^2\left(\left(\dfrac{w_g T}{2}+1\right)^2 - 1\right) + \dfrac{T\frac{\partial^2 w_g}{\partial x^2}}{2}}. \quad \text{(A16)}$$

This gives us a local volatility surface in terms of T and x.

A.9 *Using the local volatility surface*

When we use the local volatility surface, we replace K by s and T by t. In order to see what that implies for the log moneyness, we need to go back to the definition. There we had $x = \log(K/F_T)$, then when we use the local volatility surface

$$x(s, S_0, t) = \log \frac{s}{F_t(S_0)} = \log \frac{s}{S_0 \mathrm{e}^{rt}}.$$

The local volatility is now given by

$$\sigma(s,t) = \sqrt{w_{\text{local}}(x(s, S_0, t), t)}. \tag{A17}$$

Note that $\sigma(s,t)$ cannot be directly evaluated for very small values of s. A function that evaluates this volatility surface can be downloaded from the BENCHOP web site.

Paper III

Radial basis function partition of unity operator splitting method for pricing multi-asset American options

Victor Shcherbakov

Abstract The operator splitting method in combination with finite differences has been shown to be an efficient approach for pricing American options numerically. Here, the operator splitting formulation is extended to the radial basis function partition of unity method. An approach that has previously often been used together with radial basis function methods to deal with the free boundary arising in American option pricing is to solve a penalised version of the Black–Scholes equation. It is shown that the operator splitting technique outperforms the penalty approach when used with the radial basis function partition of unity method. Numerical experiments are performed for one, two and three underlying assets. The advantage of the operator splitting technique grows with the number of dimensions.

Keywords American option · Multi-asset option · Radial basis function · Partition of unity · Operator splitting method · Penalty method

Mathematics Subject Classification (2000) 65M70 · 35K15

1 Introduction

The Benchmark Project in Option Pricing (BENCHOP) [16] was recently launched. In this project established methods for option pricing were tested and compared on different benchmark problems. The metric of the comparison was the time to achieve a relative error tolerance of 10^{-4}.

This work is inspired by the BENCHOP project, where among other methods, the radial basis function partition of unity method (RBF–PUM) was tested. For many of the benchmark problems RBF–PUM was competitive with finite difference (FD) methods, but for the American option problems it performed significantly worse. In the project RBF–PUM employed the penalty method [10] for handling the free exercise boundary, while the FD methods used the operator splitting (OS) formulation [7].

V. Shcherbakov
Department of Information Technology, Uppsala University, Box 337, 751 05 Uppsala, Sweden
E-mail: victor.shcherbakov@it.uu.se

The penalty approach allows to remove the early exercise boundary and solve the problem on a fixed domain by adding a (usually non-linear) penalty function, whose size is adjusted by a penalty parameter e. A property of this approach is that the error introduced by the penalty is proportional to the penalty parameter size. In order to avoid solving a non-linear problem in the BENCHOP we used an implicit-explicit numerical scheme, where the penalty function was explicitly treated. This type of scheme was proposed in [10] stating that the condition imposed by the explicit treatment of the penalty would be rather mild. Nevertheless, the time step is strongly dependent on and proportional to the penalty parameter size. Thus, in order to reach the given tolerance (10^{-4}) we had to use a very small penalty parameter, which imposed a severe condition on the time step size to maintain stability. This resulted in long execution times.

The operator splitting approach works on the linear complementarity problem (LCP), where a fully implicit scheme can be easily applied. Consequently, it does not enforce any constraint on the time step size. We believe that the use of the OS technique made all FD methods noticeably faster than RBF–PUM for the American option pricing problem.

Thus, the primary goal of this work is to extend the operator splitting approach to RBF–PUM, which to the author's knowledge has not yet been done. Then, we compare the results of the new method to those obtained by the penalty approach in the BENCHOP project. Furthermore, for the penalty approach we also implement a fully implicit scheme as well as a fully explicit scheme in order to compare against the previously implemented implicit-explicit scheme. In the fully implicit case, we employ Newton's method to handle the non-linearity. It turns out that Newton's method on average requires only a few iterations to converge, regardless of the problem's dimension. This provides an overall efficient scheme.

The layout of this paper is structured as follows. In section 2, we introduce the Black–Scholes equation for the American multi-asset put option. In section 3, we discuss the operator splitting method and the penalty method for American options. In section 4, we give an overview of RBF–PUM, set up the discrete formulation and briefly discuss implementation details. In section 5, we test the three penalty schemes and the operator splitting scheme for the one-dimensional case, and then we proceed to higher dimensional cases with the best candidates. In section 6, we give some conclusion.

2 The Black–Scholes equation

Under the Black–Scholes market assumptions [1] we consider the d-dimensional Black–Scholes equation, whose solution defines the price of an American put option written on d assets

$$\frac{\partial V}{\partial t} - \mathscr{L}V = 0, \quad \mathbf{x} \in \Omega, t \in (0,T], \tag{2.1}$$

$$\mathscr{L} = \frac{1}{2}\sum_{i,j=1}^{d} \sigma_i \sigma_j \rho_{ij} x_i x_j \frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i=1}^{d} r x_i \frac{\partial}{\partial x_i} - r, \tag{2.2}$$

where V is the value of the option, $\mathbf{x} = (x_1, \ldots, x_d)$ defines the spot prices of the d underlying assets, d is the number of assets in the portfolio, σ_i is the volatility of asset i, ρ_{ij} is the correlation between assets i and j, r is the risk-free interest rate, t is the backward time, i.e., time to maturity, and T is the maturity time of the option. Dividends could also be taken into account, but since we consider only American put options, the dividends are not necessary in order to allow for the possibility of early exercise, as for example in the case of call options. The domain Ω is a subdomain of $\mathbb{R}^d_+$, which falls inside the early exercise boundary $\Gamma(\mathbf{x}, t)$.

The payoff function for the put option is given by:

$$\Phi(\mathbf{x}) = \max\left(K - \sum_{i=1}^{d} \theta_i x_i, 0\right),$$

where K is the strike price and θ_i is the weight of the i-th asset in the portfolio. The payoff defines the option value at the time of maturity. Since the problem is stated in backward time, the payoff provides the initial condition for equation (2.1)

$$V(\mathbf{x}, 0) = \Phi(\mathbf{x}), \quad \mathbf{x} \in \Omega. \tag{2.3}$$

In addition, equation (2.1) is subject to the following boundary condition at the near-field boundary

$$V(0, t) = K, \quad t \in [0, T], \tag{2.4}$$

and at the free boundary

$$V(\mathbf{x}, t) = \Phi(\mathbf{x}), \quad \mathbf{x} \in \Gamma(\mathbf{x}, t), t \in [0, T], \tag{2.5}$$

$$\frac{\partial V}{\partial x_i}(\mathbf{x}, t) = -\theta_i, \quad \mathbf{x} \in \Gamma(\mathbf{x}, t), t \in [0, T]. \tag{2.6}$$

Outside the free boundary the solution is given by $V(\mathbf{x}, t) = \Phi(\mathbf{x})$.

Remark 2.1 The near-field boundary can be represented by the point $\mathbf{x} = 0$. At the boundaries $\nu_i = \{\mathbf{x} \,|\, \mathbf{x} \in \Omega, \mathbf{x} \neq 0, x_i = 0\}$, the spatial operator (2.2) is degenerate and reduces to a $(d-1)$-dimensional operator. Fichera [4] derived general conditions for when to impose boundary conditions for parabolic PDEs with degenerate diffusion operators. In the case of the Black–Scholes operator, boundary conditions should not be imposed at ν_i unless required for numerical purposes.

The above problem can also be reformulated as a linear complimentarity problem

$$(V - \Phi) \geq 0, \tag{2.7}$$

$$\left(\frac{\partial V}{\partial t} - \mathscr{L}V\right) \geq 0, \tag{2.8}$$

$$\left(\frac{\partial V}{\partial t} - \mathscr{L}V\right)(V - \Phi) = 0. \tag{2.9}$$

The formulations (2.1)–(2.6) and (2.7)–(2.9) are equivalent and have the same solution.

3 Methodology

In this section we introduce two approaches for solving the Black–Scholes equation for multi-asset American put options.

3.1 The penalty method

We use the type of penalty function for American put options that was introduced by Nielsen et al. [10] and has been used later by several authors together with finite differences [11] as well as radial basis functions [3,13,14].

The penalty function is given by

$$P(V) = \frac{e\,rK}{V+e-q},$$

where e is a penalty parameter, which has to be chosen sufficiently small, and $q(\mathbf{x})$ is a barrier function equal to the non-zero part of the payoff,

$$q(\mathbf{x}) = K - \sum_{i=1}^{d} \theta_i x_i. \tag{3.1}$$

The form of the penalty term has also been generalised for application to American call options in [14].

Adding the penalty term to the Black–Scholes equation allows us to convert the free boundary problem to a fixed domain problem. The error introduced by the penalty is expected to be $\mathscr{O}(e)$. The modified equation takes the form

$$\frac{\partial V}{\partial t} - \mathscr{L}V - P(V) = 0, \quad \mathbf{x} \in \tilde{\Omega}, t \in [0,T]. \tag{3.2}$$

Equation (3.2) is defined over the entire $\mathbb{R}^d_+$, but in order to enable numerical simulations we use a truncated domain $\tilde{\Omega}$, which is truncated at $x_i = x_\infty$ in each direction. We impose the following boundary conditions

$$V(0,t) = K,\ t \in [0,T], \tag{3.3}$$

$$V(\mathbf{x},t) = 0,\ \mathbf{x} \in \gamma(\mathbf{x}),\ t \in [0,T], \tag{3.4}$$

where $\gamma(\mathbf{x}) = \{\mathbf{x} \mid \mathbf{x} \in \tilde{\Omega},\ x_i = x_\infty,\ i = 1,\ldots,d\}$ is called the far-field boundary. The initial condition becomes

$$V(\mathbf{x},0) = \Phi(\mathbf{x}), \quad \mathbf{x} \in \tilde{\Omega}.$$

A typical choice for x_∞ falls between $2dK$ and $4dK$, depending on the problem parameters, where d is the problem dimension and K is the strike price.

3.2 The operator splitting method

In contrast to the penalty approach, the operator splitting method [7] deals with the Black–Scholes equation in the linear complementarity form. An auxiliary variable λ is introduced to force the option value to be higher then q, which is defined in equation (3.1). The problem then reads

$$\begin{aligned}
&\lambda = \frac{\partial V}{\partial t} - \mathscr{L}V, && \mathbf{x} > 0,\ t \in [0,T], && (3.5)\\
&(V-q)\lambda = 0, && \mathbf{x} > 0,\ t \in [0,T], && (3.6)\\
&(V-q) \geq 0, \quad \lambda \geq 0, && \mathbf{x} > 0,\ t \in [0,T], && (3.7)\\
&V = \Phi, && \mathbf{x} > 0,\ t = 0, && (3.8)\\
&V = K, && \mathbf{x} = 0,\ t \in [0,T], && (3.9)\\
&V = 0. && \mathbf{x} \in \gamma(\mathbf{x}),\ t \in [0,T], && (3.10)
\end{aligned}$$

4 Time discretisation and space approximation

The space approximation is performed by the radial basis function partition of unity method. This is a localised modification of the global RBF method, where local approximants are constructed in overlapping subdomains $\{\Omega_i\}_{i=1}^M$, which form an open cover of the entire computational domain $\tilde{\Omega}$. The local approximants are combined by the partition of unity functions $\{w^i\}_{i=1}^M$ subordinated to the open cover $\{\Omega_i\}_{i=1}^M$ into a global approximation function. The locality of the partition of unity technique allows for a significant reduction of the computational effort compare with the global method.

We divide our computational domain into M overlapping spherical patches $\{\Omega_i\}_{i=1}^M$ with radii R_i and N_i computational nodes in each. Thus, in every patch we can construct a local time-dependent RBF approximation over the nodes $\mathbf{x}_j$, $j = 1,\ldots,N_i$,

$$v^i(t,\mathbf{x}) = \sum_{j=1}^{N_i} \alpha_j^i(t)\phi(\varepsilon||\mathbf{x}-\mathbf{x}_j||) = \sum_{j=1}^{N_i} \alpha_j^i(t)\phi_j(\mathbf{x}),$$

where $\phi_j(\mathbf{x})$ is a radial basis function centered at node $\mathbf{x}_j$, $\alpha_j^i(t)$ are coefficients to be determined and ε is the shape parameter, which defines the width of the basis functions.

The local approximants are combined into the global approximant by

$$v(t,\mathbf{x}) = \sum_{i=1}^{M} w^i(\mathbf{x})v^i(t,\mathbf{x}) = \sum_{i=1}^{M} w^i(\mathbf{x}) \sum_{j=1}^{N_i} \alpha_j^i(t)\phi_j(\mathbf{x}), \tag{4.1}$$

where w^i is a partition of unity function compactly supported on Ω_i and satisfying the property

$$\sum_{i=1}^{M} w^i(\mathbf{x}) = 1.$$

The partition of unity can be constructed by Shepard's method [15]

$$w^i(\mathbf{x}) = \frac{\psi^i(\mathbf{x})}{\sum_{k=1}^M \psi^k(\mathbf{x})},$$

where $\psi^i(\mathbf{x})$ is a compactly supported function on Ω_i. We use C^2 compactly supported Wendland functions [17]

$$\psi(r) = \begin{cases} (1-r)^4(4r+1), & \text{if } 0 \le r \le 1 \\ 0, & \text{if } r > 1. \end{cases}$$

For a more detailed description see [14].

4.1 The penalty method

The total number of node point in the entire $\tilde{\Omega}$ is $N = N_I + N_B$, where N_I is the number of interior nodes and N_B is the number of boundary nodes. In order to determine the values of $\alpha_j^i(t)$ we use collocation at the N node points. For the interior nodes $\mathbf{x}_j$, $j = 1,\dots,N_I$ we use equation (3.2), and for the near-field and far-field boundary points $\mathbf{x}_j$, $j = N_I+1,\dots,N$ we use equations (3.3) and (3.4), respectively. Thus, $\mathbf{v}_I(t) = [v(t,\mathbf{x}_1),\dots,v(t,\mathbf{x}_{N_I})]^T$ and $\mathbf{v}_B(t) = [v(t,\mathbf{x}_{N_I+1}),\dots,v(t,\mathbf{x}_N)]^T$. By evaluating (4.1) at the node points, we get

$$\begin{bmatrix} \mathbf{v}_I(t) \\ \mathbf{v}_B(t) \end{bmatrix} = \begin{bmatrix} A_{II} & A_{BI} \\ A_{IB} & A_{BB} \end{bmatrix} \begin{bmatrix} \alpha_I(t) \\ \alpha_B(t) \end{bmatrix},$$

where the global coefficient matrix A consists of elements $a_{jk} = w^i(\mathbf{x}_k)\phi(\varepsilon||\mathbf{x}_j - \mathbf{x}_k||)$ and has a nearly block-diagonal structure, where the i-th block corresponds to the i-th patch. The blocks overlap due to the overlapping structure of the open cover $\{\Omega_i\}_{i=1}^M$. Consequently,

$$\begin{aligned} \mathscr{L}\mathbf{v}_I(t) &= \begin{bmatrix} L_{II} & L_{IB} \end{bmatrix} \begin{bmatrix} \alpha_I(t) \\ \alpha_B(t) \end{bmatrix} = \begin{bmatrix} L_{II} & L_{IB} \end{bmatrix} A^{-1} \begin{bmatrix} \mathbf{v}_I(t) \\ \mathbf{v}_B(t) \end{bmatrix} \\ &= \begin{bmatrix} C_{II} & C_{IB} \end{bmatrix} \begin{bmatrix} \mathbf{v}_I(t) \\ \mathbf{v}_B(t) \end{bmatrix}, \end{aligned}$$

where the matrix L consists of elements $l_{jk} = \mathscr{L}\left(w^i(\mathbf{x}_k)\phi(\varepsilon||\mathbf{x}_j - \mathbf{x}_k||)\right)$, $j = 1,\dots,N_I$ and $k = 1,\dots,N$. Note that A^{-1} exists, because for standard choices of basis functions A is non-singular.

For time marching we select the unconditionally stable second order backward differentiation formula (BDF-2) [6, p. 401]. The BDF-2 scheme involves three time levels. To initiate the method, often the BDF-1 scheme (Euler backward) is used for the first time step. Therefore two different matrices need to be factorised. In order to avoid this we use the BDF-2 scheme as described in [8].

We split the time interval $[0,T]$ into N_t non-uniform steps of length $k^n = t^{n+1} - t^n$, $n = 1,\dots,N_t$, then

$$\begin{aligned}(E-\beta_0^n\mathscr{L})\,\mathbf{v}_I^1 &= \mathbf{v}_I^0,\\ (E-\beta_0^n\mathscr{L})\,\mathbf{v}_I^n &= \beta_1^n\mathbf{v}_I^{n-1} - \beta_2^n\mathbf{v}_I^{n-2} + \beta_0^n P(\mathbf{v}_I^{n-1}), \quad n = 2,\dots,N_t,\end{aligned}$$

where E is an identity matrix of the proper size and

$$\beta_0^n = k^n\frac{1+\omega_n}{1+2\omega_n}, \quad \beta_1^n = \frac{(1+\omega_n)^2}{1+2\omega_n}, \quad \beta_2^n = \frac{\omega_n^2}{1+2\omega_n},$$

where $\omega_n = k^n/k^{n-1}, n = 2,\dots,N_t$. In [8] it is shown how the time steps can be chosen in such a way that $\beta_0^n \equiv \beta_0$. Then the coefficient matrix is the same in all time steps and only one matrix factorization is needed.

For the boundary node points we enforce the following condition

$$\mathbf{v}_B^n = \mathbf{f}_B^n, \quad n = 1,\dots,N_t,$$

where

$$\mathbf{f}_B = [f(t,\mathbf{x}_{N_I+1}),\dots,f(t,\mathbf{x}_N)]^T, \tag{4.2}$$

and

$$f(t,\mathbf{x}) = \begin{cases} K, & \text{if } \mathbf{x} = 0\\ 0, & \text{if } \mathbf{x} \in \gamma(\mathbf{x}).\end{cases}$$

This leads to a linear system, which needs to be solved at each time step

$$\begin{bmatrix} E_I - \beta_0 C_{II} & -\beta_0 C_{IB}\\ 0 & E_B\end{bmatrix}\begin{bmatrix}\mathbf{v}_I^n\\ \mathbf{v}_B^n\end{bmatrix} = \begin{bmatrix}\mathbf{f}_I^n\\ \mathbf{f}_B^n\end{bmatrix}, \tag{4.3}$$

where

$$\mathbf{f}_I^n = \beta_1^n\mathbf{v}_I^{n-1} - \beta_2^n\mathbf{v}_I^{n-2} + \beta_0^n P(\mathbf{v}_I^{n-1}).$$

In short notations we write it as

$$B\mathbf{v}^n = \mathbf{f}^n. \tag{4.4}$$

Note that in order to avoid solving a non-linear system the penalty term is treated explicitly here. It imposes some condition on the time step size. Nielsen shows [10] that in case of finite differences for put options the condition is

$$\Delta t \le \frac{e}{rK},$$

where e is the penalty parameter, r is the risk-free interest rate, K is the strike price and in our case $\Delta t = \max\{k^n\}_{n=1}^{N_t}$. Experiments for RBFs seem to give nearly the same bound [14].

We also implement an entirely implicit BDF-2 scheme, where the penalty is defined at the new time level. In this case instead of the linear system (4.3) we obtain the following non-linear system

$$\begin{bmatrix} E_I - \beta_0 C_{II} & -\beta_0 C_{IB}\\ 0 & E_B\end{bmatrix}\begin{bmatrix}\mathbf{v}_I^n\\ \mathbf{v}_B^n\end{bmatrix} - \beta_0^n\begin{bmatrix}P(\mathbf{v}_I^n)\\ 0\end{bmatrix} = \begin{bmatrix}\mathbf{f}_I^n\\ \mathbf{f}_B^n\end{bmatrix},$$

where

$$\mathbf{f}_I^n = \beta_1^n \mathbf{v}_I^{n-1} - \beta_2^n \mathbf{v}_I^{n-2},$$

and $\mathbf{f}_B^n$ remains as in (4.2). In order to solve the arising non-linear system we employ the Newton method.

To complete the picture we also consider a second-order explicit scheme. For this purpose we select Heun's scheme. This results in the following linear system

$$\begin{aligned}
\mathbf{w}_I^n &= \mathbf{v}_I^{n-1} + \Delta t \left[C\mathbf{v}^{n-1} + P(\mathbf{v}_I^{n-1}) \right], \\
\mathbf{w}_B^n &= \mathbf{v}_B^{n-1}, \\
\mathbf{v}_I^n &= \mathbf{v}_I^{n-1} + 0.5\Delta t \left[C\mathbf{w}^n + P(\mathbf{w}_I^n) + C\mathbf{v}^{n-1} + P(\mathbf{v}_I^{n-1}) \right], \\
\mathbf{v}_B^n &= \mathbf{f}_B^n,
\end{aligned}$$

where $\mathbf{w}^n$ is an auxiliary variable, $\mathbf{f}_B^n$ as in (4.2), and

$$C = \left[C_{II}\ C_{IB} \right].$$

Apart from the constraint on the time step size enforced by the penalty, there will be an additional constraint which comes from the explicit time discretisation.

A test of the above three schemes will tell us which method is better suited for the penalised Black–Scholes equation. The point of view which has been common in the literature is that the semi-implicit scheme is an optimal choice, because it exempts us from solving a non-linear system, while the enforced condition on the time step is not as severe as it could be if one used an entirely explicit discretisation scheme.

We will see that it turns out that the implicit penalty method is the most efficient out of the three modifications. Therefore we decide to present an algorithm only for this version.

Algorithm to evaluate an American option by the implicit penalty method

```
for $n = 2 : N_t$
    while $err \geq \tau$  % Newton iterations
        $P = e*r*K./(y+e-q)$; % Evaluate the penalty
        $s = spdiags(P./(y+e-q),0,N,N)$; % Matrix needed for
                                          Newton iterations
        $v = y-(B-\beta_0*s)\backslash(B*y+\beta_0*P+f)$;
        $err = max(abs(v-y))$;    $y = v$;
    end while
    $v_0 = v_1$;  $v_1 = v$;  % Update to next time step
end for
```

4.2 The operator splitting method

As in the previous section we use collocation to determine the coefficients $\alpha_j^i(t)$ and the BDF-2 scheme for time evolution. The main steps of the discretisation for the operator splitting method can be repeated in the same manner as for the penalty method.

The operator splitting method has to be solved in two fractional steps at each time step. At the first fractional step, the linear system is solved, and then, at the second fractional step, the auxiliary variables λ^n are updated. Thereby, problem (3.5)–(3.10) in the discrete case the takes form

$$\begin{aligned}
&B\tilde{\mathbf{v}}^n = \mathbf{g}^n + \beta_0^n \lambda^{n-1} \\
&\mathbf{v}^n - \tilde{\mathbf{v}}^n = \beta_0^n \left(\lambda^n - \lambda^{n-1}\right) \\
&(\mathbf{v}^n - \mathbf{q})\lambda^n = 0, \\
&(\mathbf{v}^n - \mathbf{q}) \geq 0, \quad \lambda^n \geq 0,
\end{aligned}$$

where $n = 1, \ldots, N_t$, B is the same matrix as in equation (4.4), $\mathbf{g}^n = [\mathbf{g}_I^n, \mathbf{g}_B^n]^T$ with

$$\mathbf{g}_I^n = \beta_1^n \mathbf{v}_I^{n-1} - \beta_2^n \mathbf{v}_I^{n-2}$$

and $\mathbf{g}_B^n = \mathbf{f}_B^n$. The initial value for the auxiliary variable is $\lambda^0 = 0$.

Below we present an algorithm for the OS method for the American option problem, which highlights the main implementation points.

Algorithm to evaluate an American option by the operator splitting method

```
[L,U] = lu(B);
 for n = 2 : $N_t$
      b = g − $\beta_0$ ∗ $\lambda$;
      $\tilde{v}$ = U\(L\b);
      $\lambda_1$ = $\lambda$;   $\lambda$ = zeros(N,1);  % Update $\lambda$ to next time step
      v = $\tilde{v}$ + $\beta_0$ ∗ ($\lambda$ − $\lambda_1$);
      ind = find(v − $\Phi$ < 0);  % Find where solution falls below
                                  the payoff
      v(ind) = $\Phi$(ind);  % Project to the payoff
      $\lambda$(ind) = $\lambda_1$(ind) + ($\tilde{v}$(ind) − v(ind))/$\beta_0$;
      $v_0$ = $v_1$;   $v_1$ = v;  % Update to next time step
 end for
```

5 Numerical experiments

In this section we compute American put option values V with the above mentioned methods. We evaluate option values at three predefined asset values, which correspond to “in-the-money”, “at-the-money” and “out-of-the-money” cases. All solutions are benchmarked against accurate reference solutions V_r, which are obtained by Fourier methods. The relative deviation of the approximated values V from the reference values V_r is restricted to be less than 10^{-4}. By practitioners, this precision is considered sufficient for financial purposes. Therefore, if the methods succeed in meeting this restriction, then the main criteria, which determines the advantage of one method over another, would be the *shorter computational time*.

In other words, all numerical experiments are performed with the aim *to get the relative error* δ *below the tolerance* 10^{-4} *in the shortest time*,

$$\delta = \left\| \frac{V - V_r}{V_r} \right\|_\infty .$$

All corresponding errors and run times are recorded and presented in the tables. To eliminate randomness, each code was run 10 times, then the two longest and the two shortest times were removed and the mean over the remaining six times was measured. Error profiles for the option values evaluated on the cube $[80, 120]^d$, where d denotes the dimension, are also presented in the figures below.

All reference solutions are obtained by using highly accurate versions of Fourier methods. The FGL method [9] in the single-asset case, and the COS method [12] in the double-asset case. For the RBF–PUM experiments we select the multiquadric basis function $\phi = \sqrt{1+\varepsilon^2 r^2}$.

The experiments are carried out on a machine with an Intel Core i7 processor, 2.3 GHz and 16 GB RAM. All codes are implemented serially in MATLAB R2014b.

To facilitate an easy reproduction of our results, we publish parameter values, that were used in the experiments, in Table 5.1. Parameters for different methods within the same experiment may differ, because they are fine-tuned to give an accurate solution within "optimal" time.

Remark 5.1 We do not guarantee that the chosen parameters and domain sizes are the true optimal ones and nothing more optimal can be found. We use quotation marks in order to denote "minimal", "optimal", "shortest" in terms of what we could find within a reasonable amount of fine-tuning.

5.1 The non-uniform grid

In all our experiments we exploit the flexibility of RBF–PUM to easily handle non-uniform grids. The non-uniform grid which we used has grid points clustered around the strike price K. The node coordinates in each direction i are defined by

$$x_{i,j} = K + l \sinh(\xi_j), \quad j = 1, \dots, \sqrt[d]{N},$$

where $\xi_j \in [\xi_*, \xi^*]$ are equidistant points, and l is the parameter which defines the amount of clustering. Requiring that the nodes $x_{i,j}$ should fall into the interval $[0, x_\infty]$ we can find ξ_* and ξ^*

$$\begin{aligned} \xi_* &= \sinh^{-1}(-K/l), \\ \xi^* &= \sinh^{-1}\left((x_\infty - K)/l\right). \end{aligned}$$

The amount of clustering $l = 0.4$ for all the experiments. We choose equal size partitions in order to enable the construction of a higher order approximant in the strike region, which is critical from the financial point of view.

Table 5.1 Parameters that were used in the experiments. PEX denotes the explicit penalty method, PIMEX denotes the IMEX penalty method, PIM denotes the implicit penalty method, and OS denotes the operator splitting method.

Method		1D	2D	3D
PEX	Domain	$\mathbf{x} \in [0, 3K]$		
	Shape param	$\varepsilon = 0.17$		
	Pen param	$e = 10^{-4}$		
	Num parts	$M = 12$		
PIMEX	Domain	$\mathbf{x} \in [0, 3K]$		
	Shape param	$\varepsilon = 0.2$		
	Pen param	$e = 10^{-4}$		
	Num parts	$M = 35$		
PIM	Domain	$\mathbf{x} \in [0.5K, 2K]$	$\mathbf{x} \in [0, 4K]^2$	
	Shape param	$\varepsilon = 0.2$	$\varepsilon = 0.1$	
	Pen param	$e = 10^{-5}$	$e = 10^{-6}$	
	Num parts	$M = 5$	$M = 144$	
	Newton tol	$\tau = 10^{-4}$	$\tau = 10^{-4}$	
OS	Domain	$\mathbf{x} \in [0.5K, 2K]$	$\mathbf{x} \in [0, 4K]^2$	$\mathbf{x} \in [0, 6K]^3$
	Shape param	$\varepsilon = 0.6$	$\varepsilon = 0.1$	$\varepsilon = 0.1$
	Num parts	$M = 3$	$M = 144$	$M = 512$

In Figure 5.1 we can see the non-uniform grid and partitioning that were used in the two-dimensional experiments for the OS method. The use of the partition of unity technique allows for a significant sparsification of the linear system. For example, with 144 partitions only 5.6% of the elements remain. For the numerical experiments we permute the matrix elements according to the sparse reverse Cuthill-McKee ordering [2].

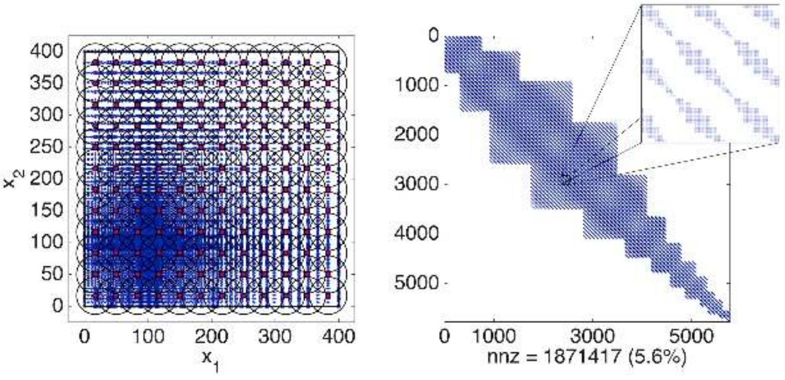


Fig. 5.1 Left: The non-uniform grid with 5776 computational nodes (76 per dimension) that was used in the two-dimensional OS experiments. The partition of unity is performed using 144 circular patches of equal size. Right: The RBF–PUM coefficient matrix that was obtained on this grid. It contains only 5.6% non-zero elements.

Remark 5.2 If a global RBF method was used then the system would be dense with all 100% of the elements non-zero.

5.2 The single-asset case

We evaluate the solution at three points $x = [90; 100; 110]$, which correspond to "in-the-money", "at-the-money" and "out-of-the-money" cases. The strike price $K = 100$, the volatility $\sigma = 0.15$, the risk-free interest rate $r = 0.03$, and the time of maturity $T = 1$ year.

Table 5.2 The American put option reference values and the difference $V - V_r$ for the approximations for one underlying asset. The grid sizes and the execution times for the operator splitting method and for the three versions of the penalty method are also presented. The * denotes the reference solution.

				Values	
Method	$N \times N_t$	Time (s)	$x = 90$	$x = 100$	$x = 110$
FGL*	-	-	10.726487	4.820608	1.828208
PEX	232×14000	1.5421	0.000515	0.000083	-0.000100
PIMEX	604×6500	0.4977	0.000729	0.000406	0.000177
PIM	74×75	0.0378	0.000601	0.000151	0.000126
OS	87×480	0.0249	0.000228	-0.000474	-0.000112

The first column of Table 5.2 contains names of the methods that were used. The FGL (Fourier–Gauss–Laguerre) method [9] is the reference solution. The second column displays the "minimal" grid size that allows to get the relative error down to 10^{-4}. The third column displays the execution times of the four methods in seconds, and the last three columns show the reference option values for the indicated spot prices, and the differences $V - V_r$ of the numerical solutions from the reference.

We can see that among the penalty methods the implicit version is preferable. Its run time is at least 13 times shorter than any of the other versions. A key factor that makes PIM faster than PEX and PIMEX is that it requires less grid points and time steps to achieve the given tolerance. This is a consequence of the implicit interpretation of the penalty function, which allows to remove any dependence of the time step on the penalty parameter. Furthermore, for PIM only a few Newton iterations per time step are needed and the overall computational time is not severely affected.

Remark 5.3 It is worth to notice that the number of grid points is different for the three penalty implementations. This is mostly due to the size of the penalty parameter combined with the focus on the time efficiency of each implementation. Let us consider PIM. This method has no constraint on the time step implied by the penalty size. Therefore we can select $e = 10^{-5}$, that is, the error introduced by the penalty approach will be an order of magnitude lower than the required tolerance. It simply means that we need 74 space nodes and 75 time step to have an accurate approximation. On the other hand, the efficiency of the PIMEX implementation suffers from the penalty parameter size. The smaller the parameter is the smaller time steps we have to take. In the experiment for PIMEX we used $e = 10^{-4}$ and it led to 6500 time steps and 604 space points to suppress the error introduced by the penalty, which roughly of the same size as the tolerance. If in this case we chose $e = 10^{-5}$ we would need around 74 space points (as in the PIM case), but then the constraint on the time step size would become even more severe and we would require around 150000 steps. This would negatively affect the overall computational time. A similar argument applies to the PEX case.

However, the operator splitting method performs 1.5 times faster than the fastest of the penalty modifications. If RBF–PUM had employed the operator splitting method, instead of the IMEX penalty method, in the BENCHOP project, it would have been significantly more competitive to finite difference methods.

In the BENCHOP project PBF–PUM equipped with the IMEX penalty was on average 60 times slower than the FD operator splitting based methods. In our experiment operator splitting based RBF–PUM is 20 times faster than PBF–PUM equipped with the IMEX penalty. That is, by simple calculation we might conclude that if in the BENCHOP project RBF–PUM instead employed the operator splitting method it would have been on average just 3 times slower than the FD methods. But even this 3 times difference will be diminished in higher dimensions, because already for two-dimensional problems RBF–PUM performed better than any of the FD methods [16].

Figure 5.2 displays the error profiles for the given four methods. We can see that towards the boundaries the errors are increasing. This is caused by the use of the adapted grid in combination with equal size partitions. The outer partitions contain fewer computational nodes, consequently the local approximants are less accurate.

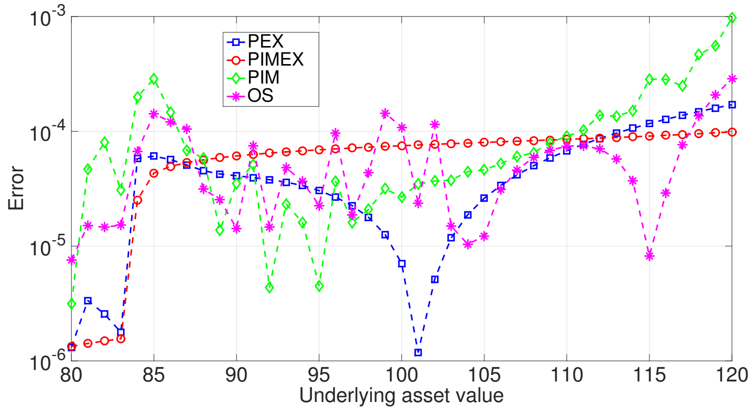


Fig. 5.2 The single-asset case. Relative error profiles of the operator splitting method and the three versions of the penalty method.

5.3 The double-asset case

Similarly, we evaluate the solution at three points $x = [90\,100;\,100\,100;\,100\,110]$, which correspond to "in-the-money", "at-the-money" and "out-of-the-money" cases. The strike price $K = 100$, the volatility $\sigma_1 = \sigma_2 = 0.15$, the correlation between the assets $\rho = 0.5$, the risk-free interest rate $r = 0.03$, and the time of maturity $T = 1$ year.

Table 5.3 The American put option reference values and the difference $V - V_r$ for the approximations for two underlying asset. The grid sizes and the execution times for the operator splitting method and for the implicit penalty method are also presented. The * denotes the reference solution.

			Values		
Method	$\sqrt{N} \times N_t$	Time (s)	$x = [90, 100]$	$x = [100, 100]$	$x = [100, 110]$
COS*	-	-	6.649395	4.051099	2.325955
PIM	73×22	62.9980	0.000173	0.000115	0.000105
OS	76×160	14.6647	0.000366	0.000420	-0.000054

For the two-dimensional experiments we keep only one version of the penalty method—the implicit version—because it performed better for the one-dimensional problem. In Table 5.3, we can see the reference option values obtained by the COS method [12] (based on a cosine expansion) and the difference $V - V_r$ between the approximated values and the reference values, together with the "minimal" grid sizes, which allow for getting the error down to 10^{-4} at the three specified points, and the execution times in seconds. In this test, the advantage of the operator splitting method over the penalty method becomes even more evident. The OS method is 4 times more efficient than PIM.

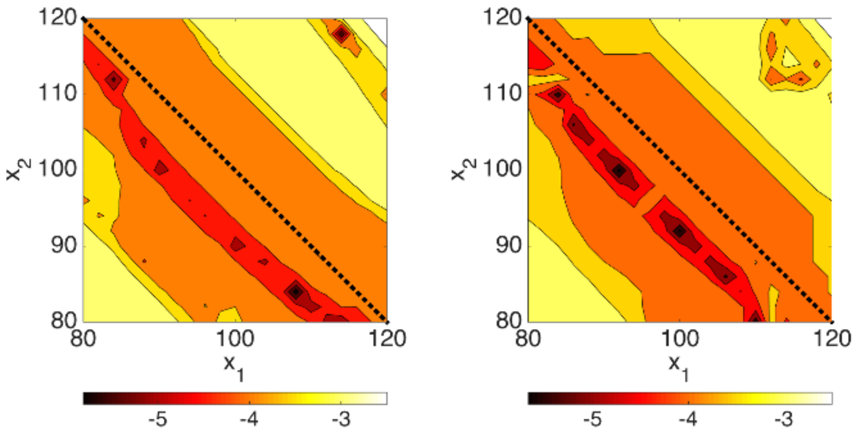


Fig. 5.3 The double-asset case. Left: The error contour in logarithmic scale for the implicit penalty method. Right: The error contour in logarithmic scale for the operator splitting method. The black dashed line denotes the strike.

In Figure 5.3 we plot contours of the error in the option values in logarithmic scale. We see that the error is larger in the lower left and in the upper right corners. This depends on the lower accuracy of the local approximants in the partitions covering those areas, because they contain fewer computational node points.

5.4 The multi-asset case

We do not possess a reference solution for the three-dimensional problem, therefore we just display the price surface of an American option written on three assets (see Figure 5.4), and option values at three points $x = [90\,100\,90;\ 100\,100\,100;\ 110\,100\,110]$, that correspond to 'in-the-money", "at-the-money" and "out-of-the-money" cases (see Table 5.4). The strike price $K = 100$, the volatility $\sigma_1 = \sigma_2 = \sigma_3 = 0.15$, the correlation between the assets $\rho_{12} = \rho_{13} = \rho_{23} = 0.5$, the risk-free interest rate $r = 0.03$, and the time of maturity $T = 1$ year.

The calculations are only performed for the operator splitting method, because the execution time of the implicit penalty method becomes unreasonably long even with only a few grid nodes per dimension. Hence, it makes the use of the implicit penalty method computationally unfeasible, while the operator splitting method allows computations on moderately large numbers of grid nodes per dimension (≈ 40) within reasonable time. Based on the experience from the one- and two-dimensional problems, this number of nodes would allow for calculating the price up to the third decimal digit, i.e., sufficiently accurately.

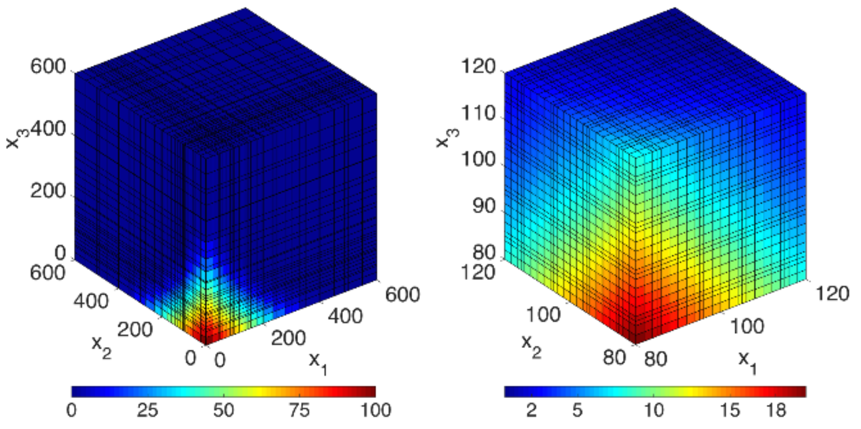


Fig. 5.4 Left: Visualisation of the three-dimensional option price surface. Right: Option values evaluated on the cube $[80, 120]^3$. The bars denote option values, and the axes denote values of the three underlying assets.

Table 5.4 Option values of an American put option written on three assets obtained by the operator splitting method.

	Values		
Method	$x = [90, 100, 90]$	$x = [100, 100, 100]$	$x = [110, 100, 110]$
OS	7.536933	3.727650	1.673915

5.5 Convergence

To validate that the methods are numerically well-behaved we study the error convergence of the OS and PIM methods in the one- and two-dimensional cases with respect to the number of grid points. The error is taken as the maximum of the relative errors at the three points indicated above. We also consider the execution time with respect to the number of computational nodes per dimension. The shape parameter and the number of partitions are kept fixed (see Table 5.1 for the values) while the convergence is being tested.

All the methods locally exhibit a high algebraic convergence rate (see Figure 5.5), which is expected for RBF based methods [13]. Although the implicit penalty method converges a little faster than the operator splitting method, it is slower in terms of run time. The reason is that PIM performs Newton iterations in order to deal with the non-linearity in each time step.

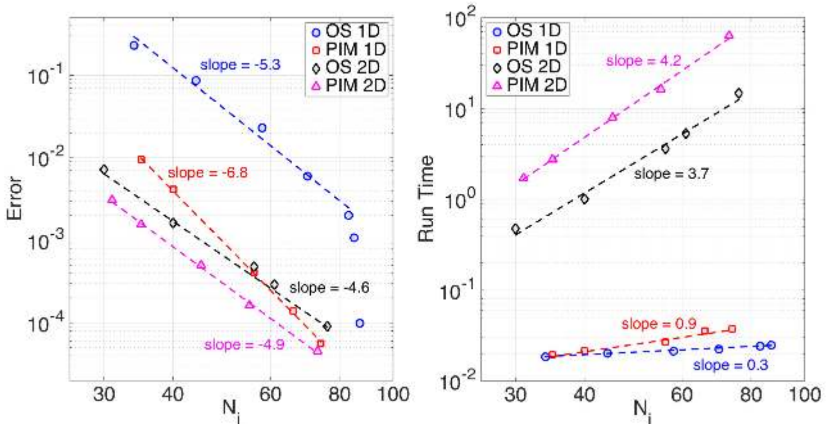


Fig. 5.5 Left: The error convergence with respect to the number of points per dimension in logarithmic scale. Right: The execution time with respect to the number of points per dimension in logarithmic scale. $N_i = N$ in the one-dimensional case and $N_i = \sqrt{N}$ in the two-dimensional case.

The execution time for both methods is almost independent of the number of points in the single-asset case, because the time spent on solving the system is negligible compared with the time for construction of local RBF matrices. In the double-asset case, solving the system of equations becomes the most time consuming operation, and we observe a nearly quadratic growth of the run time with the total number of grid points (4th order growth corresponding to $\sqrt{N}$) for both methods, which goes in line with the theory for sparse banded matrices. The total theoretical number of operations to solve the system in both cases would be $\mathscr{O}(Np^2)$, where p is the band width of the matrix [5]. In our case the band width is roughly proportional to the square root of the total size, i.e., $p^2 \sim N$. That is, the total cost would be proportional to $\mathscr{O}(N^2)$.

6 Conclusion

The operator splitting approach for pricing American options has been extended to the radial basis function partition of unity method. It has been shown suitable for multi-asset options. The penalty approach has been implemented in three versions: implicit, implicit-explicit and explicit. Previously the implicit-explicit formulation has been commonly used with radial basis functions [3, 13, 14], because it allowed to avoid solving a non-linear problem, while enforcing a "rather mild" constraint on the time step size, which still led to unnecessary extra calculations in the time integration. This work shows that, if the partition of unity technique is employed, the implicit version should be preferred. The Newton method requires only a few iterations to converge regardless of the problem dimension. Thus, it allows to avoid restriction of the time step by the penalty size and preclude unnecessary calculations.

The operator splitting method has been compared with its penalty counterparts, where we could observe a significant advantage of the OS method in terms of computational time. In three dimensions and higher, only the OS method is practically useful, because the execution times for the penalty methods become unreasonably long.

We have also studied the convergence of the OS and the PIM methods in one and two dimensions. The methods are consistent with the change of grid sizes and exhibit high algebraic convergence rates, that is characteristic for localised RBF methods.

In conclusion, we can say that if RBF–PUM had been used in combination with the operator splitting approach in the BENCHOP project, the results of RBF–PUM would have been comparable to the FD methods. We also assume that it will be more beneficial to use the operator splitting method with RBF–PUM rather than with FD methods for pricing options written on several assets, because radial basis function methods are better suited for higher-dimensional problems, due higher order accuracy and ease of implementation.

Acknowledgements The author would like to thank Marjon Ruijter (Rabobank, Utrecht) for the provided reference solution for the two-dimensional problem, Elisabeth Larsson (Uppsala University) for valuable discussions, and Slobodan Milovanović (Uppsala University) for help with visualising results.

References

1. Black, F., Scholes, M.: The pricing of options and corporate liabilities. J. Polit. Econ. **81**(3), 637–654 (1973). DOI 10.1086/260062
2. Cuthill, E., McKee, J.: Reducing the bandwidth of sparse symmetric matrices. In: Proceedings of the 1969 24th National Conference, ACM '69, pp. 157–172. ACM, New York, NY, USA (1969). DOI 10.1145/800195.805928
3. Fasshauer, G.E., Khaliq, A.Q.M., Voss, D.A.: Using mesh free approximation for multi-asset American option problems. J. Chinese Inst. Engrs. **27**(4), 563–571 (2004)
4. Fichera, G.: Sulle equazioni differenziali lineari ellittico-paraboliche del secondo ordine. Atti Accad. Naz. Lincei. Mem. Cl. Sci. Fis. Mat. Nat. Sez. I. VIII, Ser. 5 pp. 3–30 (1956)
5. Golub, G.H., Van Loan, C.F.: Matrix Computations, fourth edn. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD (2013)
6. Hairer, E., Nørsett, S., Wanner, G.: Solving Ordinary Differential Equations I. Nonstiff problems, second edn. Springer-Verlag, Berlin (2000). DOI 10.1007/978-3-540-78862-1
7. Ikonen, S., Toivanen, J.: Operator splitting methods for American option pricing. Appl. Math. Lett. **17**(7), 809–814 (2004). DOI 10.1016/j.aml.2004.06.010
8. Larsson, E., Åhlander, K., Hall, A.: Multi-dimensional option pricing using radial basis functions and the generalized Fourier transform. J. Comput. Appl. Math. **222**(1), 175–192 (2008). DOI 10.1016/j.cam.2007.10.039
9. Lindström, E., Ströjby, J., Brodén, M., Wiktorsson, M., Holst, J.: Sequential calibration of options. Comput. Statist. Data Anal. **52**(6), 2877–2891 (2008). DOI 10.1016/j.csda.2007.08.009
10. Nielsen, B.F., Skavhaug, O., Tveito, A.: Penalty and front-fixing methods for the numerical solution of American option problems. J. Comput. Finance **5**(4), 69–97 (2002)
11. Nielsen, B.F., Skavhaug, O., Tveito, A.: Penalty methods for the numerical solution of American multi-asset option problems. J. Comput. Appl. Math. **222**(1), 3–16 (2008). DOI 10.1016/j.cam.2007.10.041
12. Ruijter, M.J., Oosterlee, C.W.: Two-dimensional Fourier cosine series expansion method for pricing financial options. SIAM J. Sci. Comput. **34**(5), B642–B671 (2012). DOI 10.1137/120862053
13. Safdari-Vaighani, A., Heryudono, A., Larsson, E.: A radial basis function partition of unity collocation method for convection-diffusion equations. J. Sci. Comput. **64**(2), 341–367 (2015). DOI 10.1007/s10915-014-9935-9
14. Shcherbakov, V., Larsson, E.: Radial basis function partition of unity methods for pricing vanilla basket options. Tech. Rep. 2015-001, Division of Scientific Computing, Department of Information Technology, Uppsala University (2015)
15. Shepard, D.: A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of the 1968 23rd ACM National Conference, ACM '68, pp. 517–524. ACM, New York, NY, USA (1968). DOI 10.1145/800186.810616
16. von Sydow, L., Höök, L.J., Lindström, E., Milovanović, S., Persson, J., Shcherbakov, V., Shpolyanskiy, Y., Samuel, S., Toivanen, J., Waldén, J., Wiktorsson, M., Levesley, J., Li, J., Oosterlee, C., Ruijter, M., Toropov, A., Zhao, Y.: BENCHOP — The BENCHmarking project in Option Pricing. Int. J. Comput. Math. **92** (2015). DOI 10.1080/00207160.2015.1072172
17. Wendland, H.: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Adv. Comput. Math. **4**(4), 389–396 (1995). DOI 10.1007/BF02123482

UPPSALA
UNIVERSITET

Department of Information Technology, Uppsala University, Sweden