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

Mesh-free 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.

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,

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

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 discretization 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. One way to introduce locality is to employ a partition of unity framework, which was proposed by Babuška and Melenk in 1997 [22]. A partition based formulation is also well suited for parallel implementation. Some work on parallisation for localised RBF methods has been done, see

for example [23, 24, 25]. The ill-conditioning can be addressed by, for example, the RBF-QR technique [26, 27, 28].

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

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

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 [29] derived general conditions for when to impose boundary conditions for parabolic PDEs with degenerate diffusion operators (see also the Feller condition [30]). 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 [31]. 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 [32] 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 [33], 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

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

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

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

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 [33], 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 homogenous 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

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

   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,

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

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

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

   Applying the Cauchy-Schwarz inequality we obtain

$$Q_1 \leq 2\left(\int_0^{x^*} \left(\frac{e\,(Dx - rK)}{V_a + \eta + e - q}\right)^2 dx\right)^{1/2} \left(\int_0^{x^*} (V - V_a)^2\, dx\right)^{1/2}. \tag{3.26}$$

   Since $V_a \geq q$, $V \geq q$, we have that $|V - V_a| \leq |V - q|$ and we get

$$Q_1 \leq 2\left(\int_0^{x^*} \left(\frac{e\,(Dx - rK)}{V_a + \eta + e - q}\right)^2 dx\right)^{1/2} \left(\int_0^{x^*} (V - q)^2\, dx\right)^{1/2}. \tag{3.27}$$

   Noting that $V_a + \eta - q = V - q \geq 0$ and $e \geq 0$ and using the fact that $(a+b)^2 \geq a^2 + b^2$ if $a \geq 0$, $b \geq 0$ we obtain that

$$[(V_a + \eta - q) + e]^2 \geq (V_a + \eta - q)^2 + e^2 \geq (V_a + \eta - q)^2 = (V - q)^2. \tag{3.28}$$

   Extending the integration region to Ω_E we have

$$Q_1 \leq 2||e(Dx - rK)||\frac{1}{||V - q||}||V - q||, \tag{3.29}$$

   simplifying we obtain

$$Q_1 \leq 2e||(Dx - rK)|| \leq 2e||Dx - rK||_\infty \leq 2e(Dx_\infty - rK). \tag{3.30}$$

(b) After the free boundary we have

$$Q_2 = -2\int_{x^*}^{x_\infty} F\eta\,dx - 2\int_{x^*}^{x_\infty} \frac{e\,(rK - Dx)}{V_a + \eta + e - q}\,\eta\,dx = \\ -2\int_{x^*}^{x_\infty} (Dx - rK)\eta\,dx + 2\int_{x^*}^{x_\infty} \frac{e\,(Dx - rK)}{V_a + \eta + e - q}\,\eta\,dx. \tag{3.31}$$

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

$$Q_2 = -2\int_{x^*}^{x_\infty} |(Dx - rK)||\eta|\,dx + 2\int_{x^*}^{x_\infty} \left|\frac{e\,(Dx - rK)}{V_a + \eta + e - q}\right| |\eta|\,dx. \tag{3.32}$$

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

$$\frac{e}{V_a + \eta + e - q} \leq 1. \tag{3.33}$$

Applying this fact to (3.32) we obtain that

$$Q_2 \leq -2\int_{x^*}^{x_\infty} |(Dx - rK)||\eta|\,dx + 2\int_{x^*}^{x_\infty} |(Dx - rK)||\eta|\,dx = 0. \tag{3.34}$$

Thus, summing up all the terms we get an estimate

$$\frac{d}{dt}||\eta||^2 \leq (\sigma^2 - 3r + D)||\eta||^2 + 2(Dx_\infty - rK)e, \tag{3.35}$$

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

$$||\eta||^2 \leq \begin{cases} \frac{\mu e}{\nu}(\exp(\nu t) - 1), & \text{if } \nu \neq 0, \\ 2\mu e t, & \text{if } \nu = 0, \end{cases} \tag{3.36}$$

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 [33] would be needed.

4. Radial basis function methods

RBF methods are mesh-free 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 Euclidian 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 [34] 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 [35] 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 [36]

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

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 center point.

5. Time discretization and space approximations

When we solve the option pricing problem numerically, we collocate the different RBF approximations in space as described below 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 differential scheme (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 non-linear 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 [37, 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 [38] 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.

The boundary conditions are enforced at each new time level through

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

This leads to a linear system for each time step of the form

$$\begin{pmatrix} E_I - \beta_0 \mathcal{L}_{II} & -\beta_0 \mathcal{L}_{IB} \\ 0 & E_B \end{pmatrix} \begin{pmatrix} V_I^n \\ V_B^n \end{pmatrix} = \begin{pmatrix} f_I^n \\ f_B^n \end{pmatrix}, \tag{5.5}$$

where

$$f_I^n = \beta_1^n V_I^{n-1} - \beta_2^n V_I^{n-2} - \beta_0^n P(V_I^{n-1}). \tag{5.6}$$

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

$$\Delta t \leq \frac{Ce}{\left| rK - \sum_{i=1}^{d} \alpha_i D_i x_{i,\infty} \right|}, \tag{5.7}$$

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 $\Delta t \leq \frac{e}{rK}$, which can be found in [7] for the case when finite differences are used to price an American put without dividends. Condition (5.7) 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 (5.7) 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)

$$V_\varepsilon^n(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j^n \phi(\varepsilon\|\mathbf{x} - \mathbf{x}_j\|). \tag{5.8}$$

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

$$A\bar{\lambda}^n = \bar{v}^n, \tag{5.9}$$

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

$$\bar{\lambda}^n = A^{-1}\bar{v}^n. \tag{5.10}$$

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

$$\frac{\partial \bar{v}^n}{\partial x_k} = A^{(k)}\bar{\lambda}^n = A^{(k)}A^{-1}\bar{v}^n, \quad \frac{\partial^2 \bar{v}^n}{\partial x_k \partial x_m} = A^{(km)}\bar{\lambda}^n = A^{(km)}A^{-1}\bar{v}^n, \tag{5.11}$$

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,

$$L\bar{v}^n = \left[\frac{1}{2}\sum_{k,m=1}^{d} \Sigma_{km} x_k x_m A^{(km)} + \sum_{k=1}^{d}(r - D_k)x_k A^{(k)} - rA\right] A^{-1}\bar{v}^n, \tag{5.12}$$

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

$$P(v_j^n) = \frac{e\left(rK - \sum_{k=1}^{d} \alpha_k D_k x_k\right)}{v_j^n + e - q}. \tag{5.13}$$

These expressions are then used for populating the blocks in the system of form (5.5).

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

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

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

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

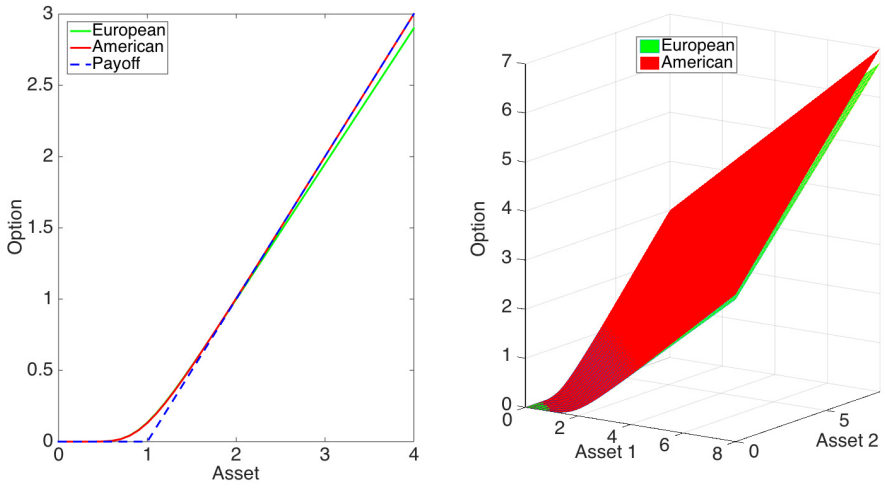

Figure 1: Left: Price of an option on one underlying dividend paying asset. Right: Price of a basket option on two underlying dividend paying assets.

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.

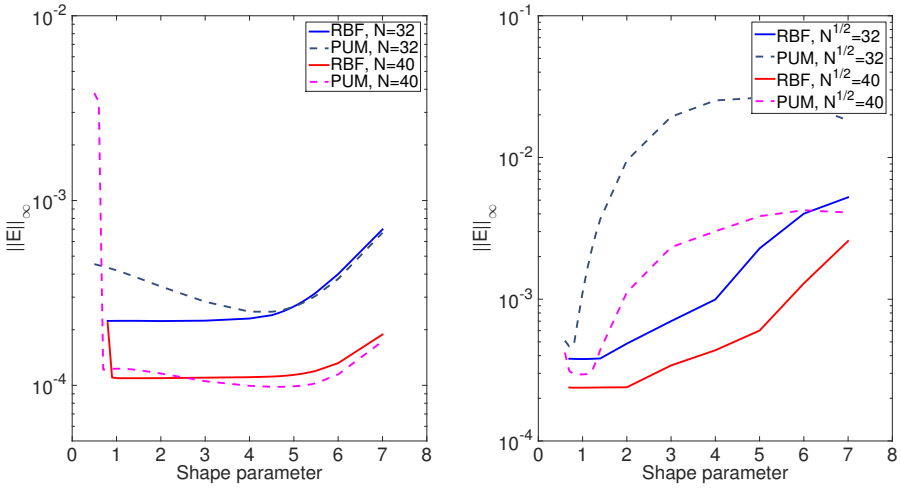

Figure 2: Left: Error in the price of the European option on one underlying asset against the shape parameter ε. Right: Error in the price of the European option on two underlying assets against the shape parameter ε. RBF denotes the global RBF method and PUM denotes RBF–PUM.

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 [20, 12]. 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 centers, 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.

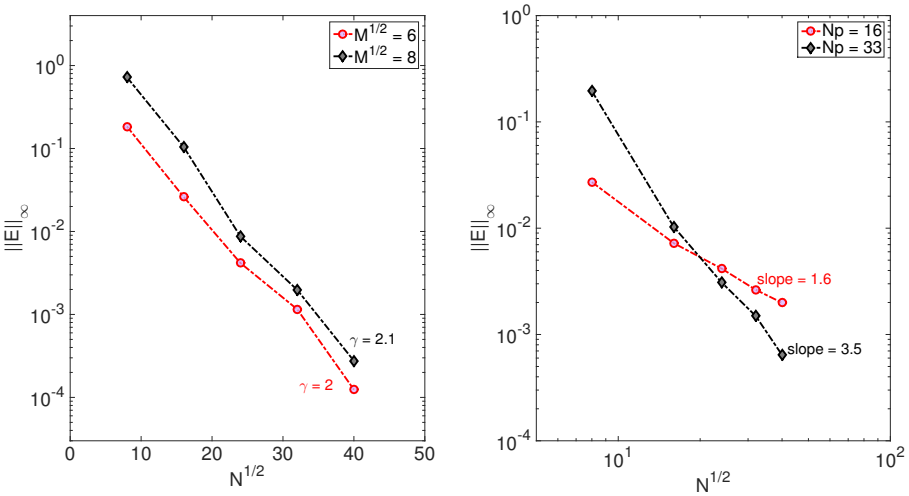

Figure 3: Left: Error in the price of the European option on two underlying assets against the problem size with respect to the number of partitions. Right: Error in the price of the European option on two underlying assets against the problem size with respect to the number of points per partition.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

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.

6.3. Number of partitions

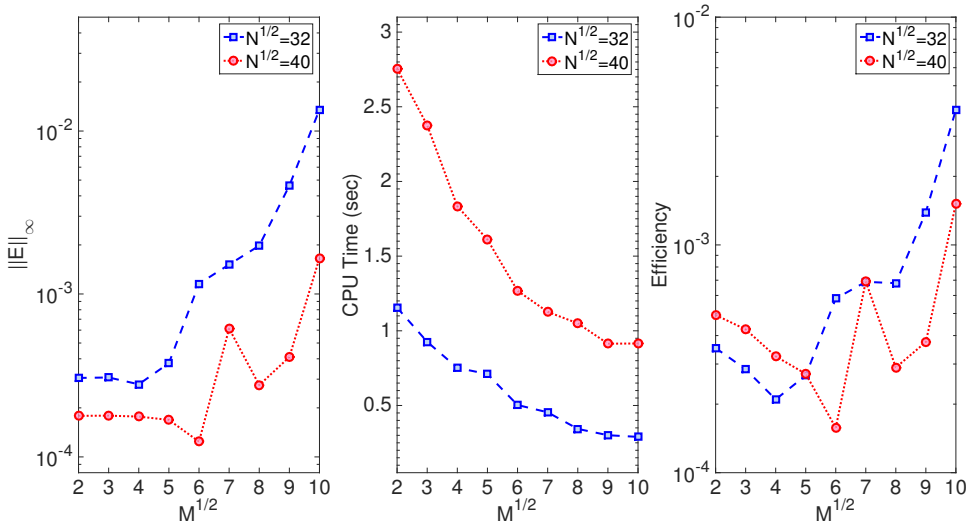

Figure 4: Left: Error in the price of the European option on 2 underlying assets against the number of partitions in one spatial dimension. Center: Computational time against the number of partitions in one spatial dimension. Right: Efficiency computed as product between the error and CPU time. Shape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

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 center, 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. Figure 5 confirms our expectations. The dependence is roughly linear in both the one-dimensional and two-dimensional case.

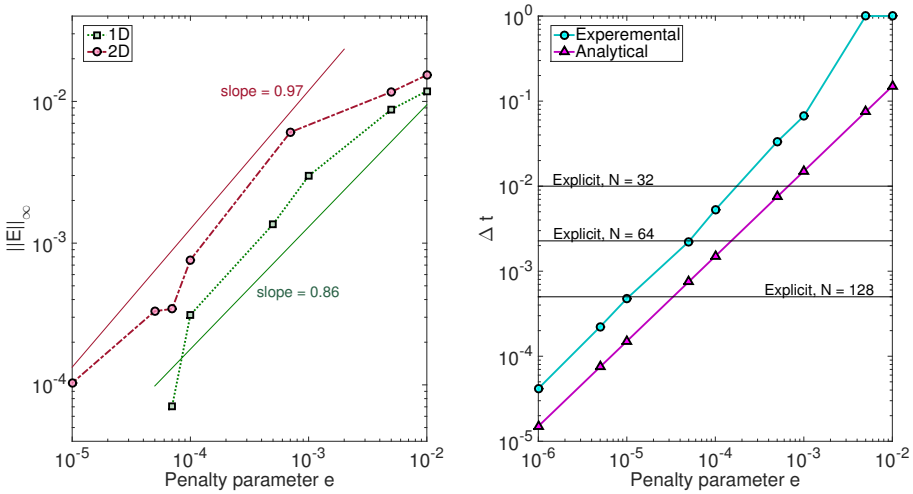

Figure 5: Left: Error measured in the region $\mathcal{U}$ against the penalty parameter size for the one asset and two asset cases. Right: Stable time step size for different sizes of the penalty parameter. Analytical – obtained from inequality (5.7) with $C = 3/2$, Experimental – experimentally obtained maximal time step for which stability holds. The three black lines show the time step size required for stability with the fully explicit scheme.

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 (5.7). The purple line indicates the analytical time step limit obtained from (5.7) 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 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.

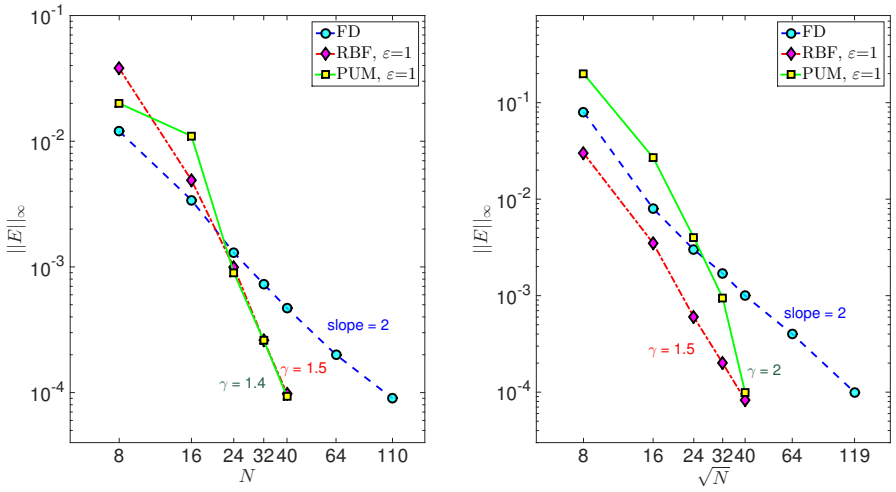

Figure 6: Left: Error convergence in l_∞-norm for a European option on one underlying asset. Right: Error convergence in l_∞-norm for a basket European option on two underlying asset.

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 [39], 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 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 [28]. 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 [26, 27, 28].

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 (5.7), 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.

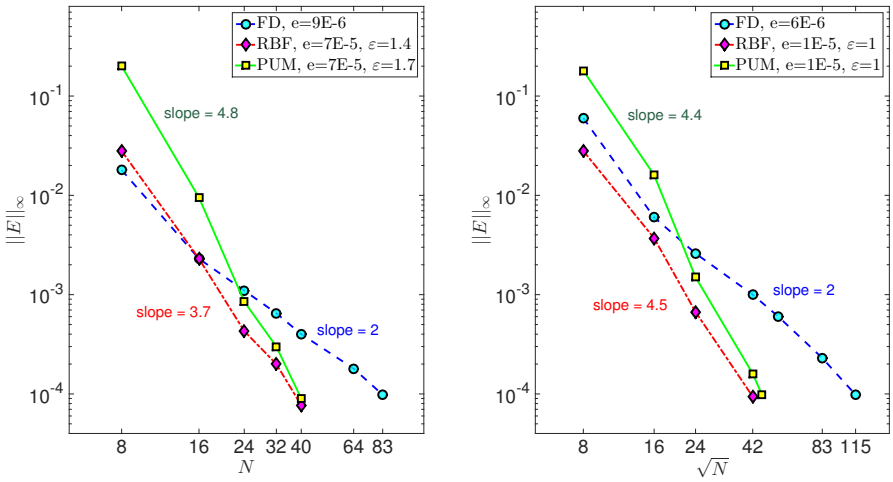

Figure 7: Left: Error convergence in l_∞-norm for an American option on one underlying asset. Right: Error convergence in l_∞-norm for a basket American option on two underlying asset. All the three methods use the penalty approach.

As we said before, we aim for error of the order 10^{-4} which is sufficient for financial 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 [40]. 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 mesh-free 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, doi:10.1086/260062.

[2] J. Hull, Options, Futures and Other Derivatives, Pearson Prentice Hall, 2009.

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

[4] S. Ikonen, J. Toivanen, Operator splitting methods for American option pricing, Appl. Math. Lett. 17 (7) (2004) 809–814, doi:10.1016/j.aml.2004.06.010.

[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, doi:10.1016/S0377-0427(98)00037-5.

[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, doi:10.1137/S1064827500382324.

[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, doi:10.1007/s00211-003-0511-8.

[9] G. E. Fasshauer, Meshfree approximation methods with MATLAB, vol. 6 of *Interdisciplinary Mathematical Sciences*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, doi:10.1142/6437, 2007.

[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, doi:10.1142/S0219024909005245.

[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, doi:10.1007/s10915-014-9935-9.

[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, doi:10.1016/j.amc.2014.11.016.

[14] Y. C. Hon, A quasi-radial basis functions method for American options pricing, Comput. Math. Appl. 43 (3-5) (2002) 513–524, doi:10.1016/S0898-1221(01)00302-9.

[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, doi:10.1016/S0955-7997(02)00083-8.

[17] P. Glasserman, Monte Carlo methods in financial engineering, Springer-Verlag, New York, doi:10.1007/978-0-387-21617-1, 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, doi:10.3905/jod.2012.19.4.029.

[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, doi:10.1137/080718061.

[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, doi:10.1007/s10444-008-9089-0.

[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, doi:10.1016/j.cam.2007.10.038.

[22] I. Babuška, J. M. Melenk, The partition of unity method, Internat. J. Numer. Methods Engrg. 40 (4) (1997) 727–758, doi:10.1002/(SICI)1097-0207(19970228)40:4¡727::AID-NME86¿3.0.CO;2-N.

[23] 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, doi:10.1016/j.jcp.2012.06.030.

[24] 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, doi:10.1016/j.compstruc.2013.11.016.

[25] M. Tillenius, E. Larsson, E. Lehto, N. Flyer, A Task Parallel Implementation of an RBF-generated Finite Difference Method for the Shallow Water Equations on the Sphere, Tech. Rep. 2014-011, Department of Information Technology, Uppsala University, 2014.

[26] B. Fornberg, C. Piret, A Stable Algorithm for Flat Radial Basis Functions on a Sphere, SIAM J. Sci. Comput. 30 (1) (2007) 60–80, doi:10.1137/060671991.

[27] B. Fornberg, E. Larsson, N. Flyer, Stable Computations with Gaussian Radial Basis Functions, SIAM J. Sci. Comput. 33 (2) (2011) 869–892, doi:10.1137/09076756X.

[28] 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, doi:10.1137/120899108.

[29] 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.

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

[31] R. Courant, Variational methods for the solution of problems of equilibrium and vibrations, Bull. Amer. Math. Soc. 49 (1943) 1–23, doi:10.1090/S0002-9904-1943-07818-4#sthash.xGwLOUMr.dpuf.

[32] 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, doi:10.1016/j.cam.2007.10.041.

[33] 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, doi:10.1016/S0362-546X(97)00207-1.

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

[35] 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, doi:10.1145/800186.810616, 1968.

[36] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. Comput. Math. 4 (4) (1995) 389–396, doi:10.1007/BF02123482.

[37] E. Hairer, S. Nørsett, G. Wanner, Solving Ordinary Differential Equations I. Nonstiff problems, Springer, Berlin, second edn., doi:10.1007/978-3-540-78862-1, 2000.

[38] 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, doi:10.1016/j.cam.2007.10.039.

[39] V. Thomée, Galerkin finite element methods for parabolic problems, vol. 25 of *Springer Series in Computational Mathematics*, Springer-Verlag, Berlin, second edn., doi:10.1007/3-540-33122-0, 2006.

[40] B. Fornberg, J. Zuev, The Runge phenomenon and spatially variable shape parameters in RBF interpolation, Comput. Math. Appl. 54 (3) (2007) 379–398, doi:10.1016/j.camwa.2007.01.028.