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

Victor Shcherbakov and Elisabeth Larsson

Abstract

Mesh-free methods based on radial basis function (RBF) approximation are 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 (PU). RBF-PU methods allow for significant sparsification of the linear system and lower the computational effort. In this work we apply a global RBF method as well as an RBF-PU method to problems in option pricing. We consider one- and two-dimensional vanilla options. In order to price American options we employ a penalty approach. The RBF-PU method is competitive with already existing methods and the results are promising for extension to higher-dimensional problems.

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 [2]. 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, 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 [10]. 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 one of the standard methods, such as projected successive over-relaxation (PSOR) [24]. The drawback of this method is that it is relatively slow. Another method, that is used in industry, is the operator splitting (OS) method [11]. It is fast and effective for one-dimensional problems. In this paper we are going to focus on a penalty approach [17], which, by adding a small penalty term, allows for removing the free boundary and solving the problem on a fixed domain.

There are various numerical methods, which are used in option pricing for industry as well as in academia. Perhaps the most popular methods are Monte-Carlo (MC) methods [8] and finite difference (FD) methods [24]. 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.

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 [21, 22]. 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 [20] in one and two dimensions.

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 [1].

A partition based formulation is also well suited for a parallel implementation. The ill-conditioning can be addressed by the RBF-QR technique [6, 7, 15].

In this paper we consider the problem of pricing dividend paying vanilla basket call options. In order to solve the problem we use RBF and RBF partition of unity (RBF-PU) methods and employ the RBF-QR technique to deal with the ill-conditioning. We show that RBF based methods provide a good alternative to already existing methods. All comparisons of the solutions are made against the standard FD solution for European options and 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 and RBF-PU methods, as well as the RBF-QR technique. 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

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

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

where 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, D_i is the continuous dividend yield paid out by the ith asset, σ is the volatility matrix, $\Sigma = [\sigma\sigma^*]$, 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. The domain $\Omega_{\{E,A\}}$ is defined below. The subscripts E and A indicate European and American options, respectively.

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

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

2.1 The European case

In the case of the European option, $\Omega_E = \mathbb{R}^d_+$, and equation (2.1) is subject to the following boundary conditions. The near-field boundary can be seen as the single point $\mathbf{x} = 0$, and there we have

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

An asymptotic solution can be defined for large values of $\mathbf{x}$

$$V(\mathbf{x}, t) \to \sum_{i=1}^{d} \alpha_i x_i e^{-D_i t} - K e^{-rt}, \quad ||\mathbf{x}|| \to \infty. \tag{2.6}$$

For computational purposes, the infinite domain needs to be truncated for large $\|\mathbf{x}\|$. The asymptotic solution (2.6) is used as a boundary condition at the truncation (far-field) boundary. We do not impose any boundary conditions on the boundaries of the type $\Gamma_i = \{\mathbf{x} \,|\, \mathbf{x} \in \Omega_E, \mathbf{x} \neq 0, x_i = 0\}$, since it was shown in [12] that as long as the growth at infinity is restricted, the problem is well posed without boundary conditions.

2.2 The American case

In the case of the American option, Ω_A is a subdomain of $\mathbb{R}^d_+$, which falls inside the free early exercise boundary $\Gamma(\mathbf{x}, t)$. Thus, equation (2.1) is subject to the following boundary conditions. For the same reason as for the European option [12] the near-field boundary can be represented by just the point $\mathbf{x} = 0$, and we enforce the following condition there:

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

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 Penalty method

Using a penalty method is a well-known technique 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 [3]. In relation to option pricing, the penalty method was introduced by Zvan et al. in [28], where a penalty approach for a Black-Scholes model with stochastic volatility for American options is discussed. Then, Nielsen et al. [17] proposed a new form of the penalty term for American put options, which has subsequently been used by several authors, combined with finite differences [18] and radial basis functions [5, 22].

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 [13], 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 non-zero part of the payoff function,

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

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

$$\frac{\partial V}{\partial t} - \mathcal{L}V + P(V) = 0, \quad \mathbf{x} \in \Omega_E,\, t \in [0, T]\,, \tag{3.2}$$

where Ω_E is the same domain as for the European option, since the free boundary has been removed. The equation is subject to the following initial and boundary conditions

$$V(\mathbf{x}, 0) = \Phi(\mathbf{x}), \quad \mathbf{x} \in \Omega_E, \tag{3.3}$$
$$V(0, t) = 0, \qquad t \in [0, T]\,, \tag{3.4}$$
$$V(\mathbf{x}, t) = \Phi(\mathbf{x}), \quad \|\mathbf{x}\| \to \infty,\, t \in [0, T]\,. \tag{3.5}$$

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 has to be greater than or equal to the value of the payoff. We need to design the penalty term in such a way that it is negligible when we are *"far"* from the payoff, and it increases in magnitude as we approach the payoff function, to prevent the solution from falling below the payoff. The current form of the penalty implies this property. When $V(x,t) > q(x)$, the penalty is roughly of size e, and it increases towards rK as $V(x,t) \to q(x)$.

Consider the one-dimensional case. The equation takes the form

$$\frac{\partial V}{\partial t} = \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V}{\partial x^2} + (r-D)x\frac{\partial V}{\partial x} - rV - \frac{e\,(rK - dx)}{V + e - q}. \tag{3.6}$$

We assume that the solution V is close to the payoff function, i.e., $V \approx x - K$. Now, for some x we have that $V = x - K + \delta$, for some $\delta > 0$, inserting this representation for V into the right part of (3.6), we obtain

$$\frac{\partial V}{\partial t} = -Dx\delta + rK\delta - r\delta^2 + rKe - re\delta - erK,$$

and if $V = x - K - \delta$, then

$$\frac{\partial V}{\partial t} = Dx\delta - rK\delta - r\delta^2 + rKe + re\delta - erK.$$

We use the fact that $Dx > rK$ when x is above the free boundary [13] and take into account that δ and e are small, we end up with

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

This shows that if the solution is positively perturbed (above the payoff), the time derivative becomes negative, and the solution starts decreasing, and *vice versa* for negative perturbations. This meta-proof confirms that our choice of the form for the penalty term is consistent with attraction to the payoff for large x.

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 [14] for elliptic PDEs, and applied to option pricing problems in [22]. 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 [23] 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 [26]

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

The elements 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 [9, 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 [16] 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 is in line with the condition $\Delta t \leq \frac{e}{rK}$, which can be found in [17] 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.

Further down 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 $R_i : \mathcal{I} \to \mathcal{I}$ is a permutation projection 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

$$\frac{\partial \bar{v}^n}{\partial x_k} = \sum_{i=1}^{M} R_i \left[W_i^{(k)} A_i + W_i A_i^{(k)} \right] \bar{\lambda}^{i,n} = \sum_{i=1}^{M} R_i \left[W_i^{(k)} A_i + W_i A_i^{(k)} \right] A_i^{-1} \bar{v}^{i,n},$$

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

$$\sum_{i=1}^{M} R_i \left[W_i^{(km)} A_i + W_i^{(k)} A_i^{(m)} + W_i^{(m)} A_i^{(k)} + W_i A_i^{(km)} \right] A_i^{-1} \bar{v}^{i,n},$$

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,

$$L\bar{v}^n = \sum_{i=1}^{M} R_i \left[\frac{1}{2} \sum_{k,m=1}^{d} \Sigma_{km} x_k x_m \left(W_i^{(km)} A_i + W_i^{(k)} A_i^{(m)} + W_i^{(m)} A_i^{(k)} + W_i A_i^{(km)} \right) + \right.$$

$$\left. + \sum_{k=1}^{d} (r - D_k) x_k \left(W_i^{(k)} A_i + W_i A_i^{(k)} \right) - r W_i A_i \right] A_i^{-1} \bar{v}^{i,n},$$

and

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

6 Numerical results

In order to solve the option pricing problems numerically, we truncate the domain where the problem is defined. A common choice for call options is to truncate at $x_{i,\infty} = 4dK$ in each direction, where d is the dimension of the problem and K is the strike price. Therefore we will carry out numerical experiments on $\Omega = [0, 4dK]^d$.

For the numerical experiments we use the semi-implicit discretisation described in the previous section. The type of basis functions we select is multiquadric. We use the following set of parameters: $K = 1$, $T = 1$, $r = 0.1$, $\sigma = 0.3$, $D_i = 0.05$ for one underlying asset and $\sigma = \begin{pmatrix} 0.3 & 0.05 \\ 0.05 & 0.3 \end{pmatrix}$ for two underlying assets.

Figure 1 displays typical solutions for European and American options on one and two underlying assets.

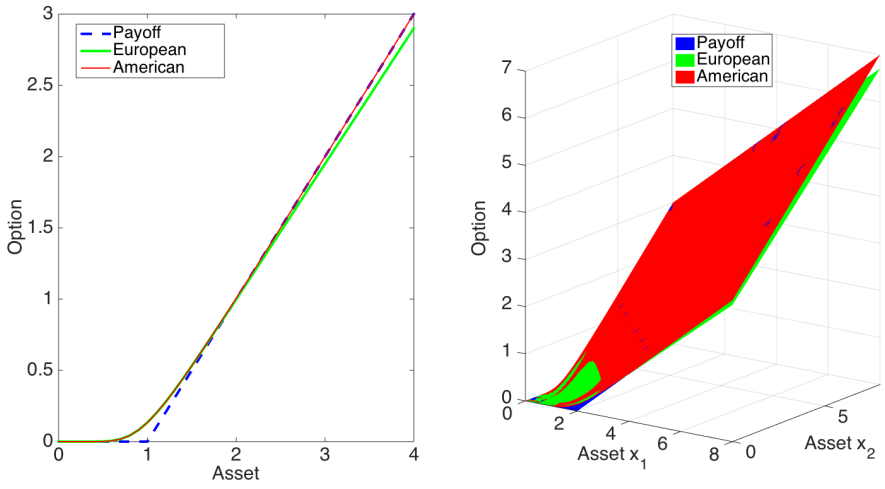

Figure 1: Left: Price of an option on 1 underlying asset. Right: Price of a basket option on 2 underlying assets.

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 too 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.

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 and RBF-PU methods were derived in [22] based on the results in [21]. 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 unknown.

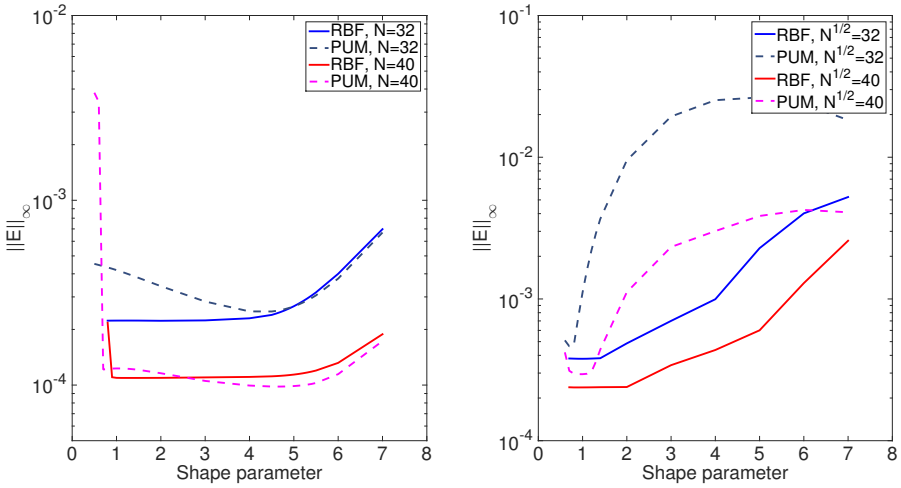

Figure 2: Left: Error in the price of the European option on 1 underlying asset against the shape parameter of RBFs. Right: Error in the price of the European option on 2 underlying assets against the shape parameter of RBFs.

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 the RBF-PUM, and $\varepsilon = 1$ for the American option on two assets with both methods.

6.2 Refinement RBF-PUM

For the global RBF method exponential convergence in space with respect to the number of nodes can be expected [21, 22]. For the RBF-PU method 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 [22] 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 d is the dimension of Ω, 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.

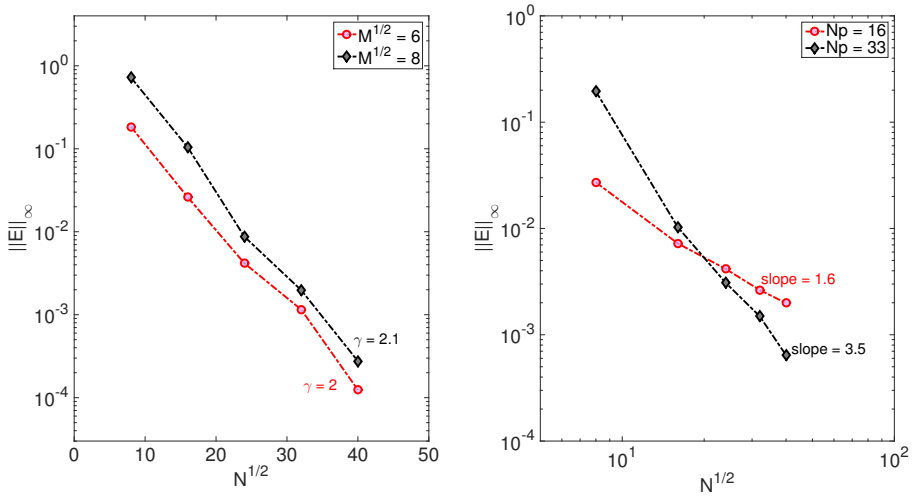

Figure 3: Left: Error in the price of the European option on 2 underlying assets against the problem size with respect to the number of partitions. Right: Error in the price of the European option on 2 underlying assets against the problem size with respect to the number of points per partition.

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 figure3 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

In the case of the RBF-PU method 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 $M = 4$.

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.

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 have any 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.

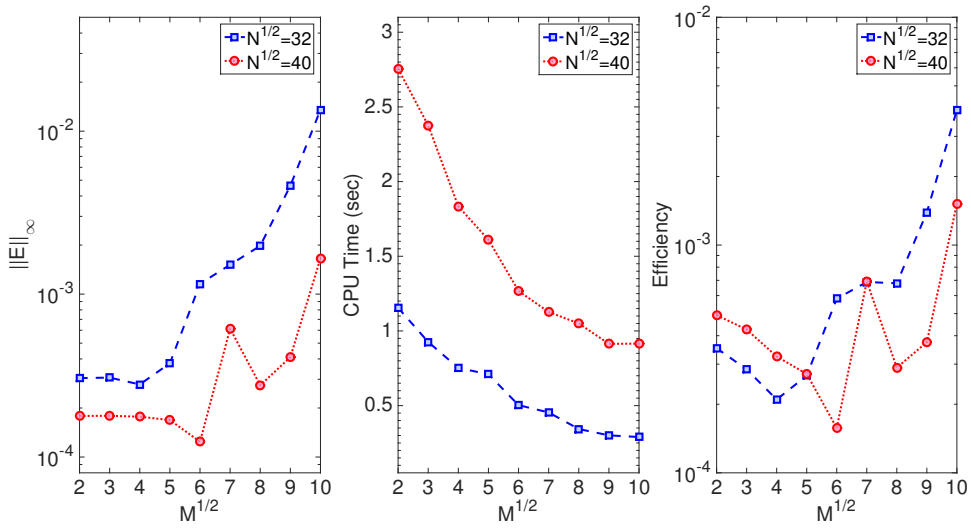

Figure 4: Left: Error in the price of the European option on 2 underlying assets against the number of partitions in one spatial dimension. Center: Computational time against the number of partitions in one spatial dimension. Right: Efficiency computed as product between the error and CPU time.

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 RBF and RBF-PU methods 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 grid to compare with.

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 the RBF-PUM.

For the European option pricing problem, the initial condition is only C^0. Hence exponential approximation accuracy at the initial time is not possible as this re-

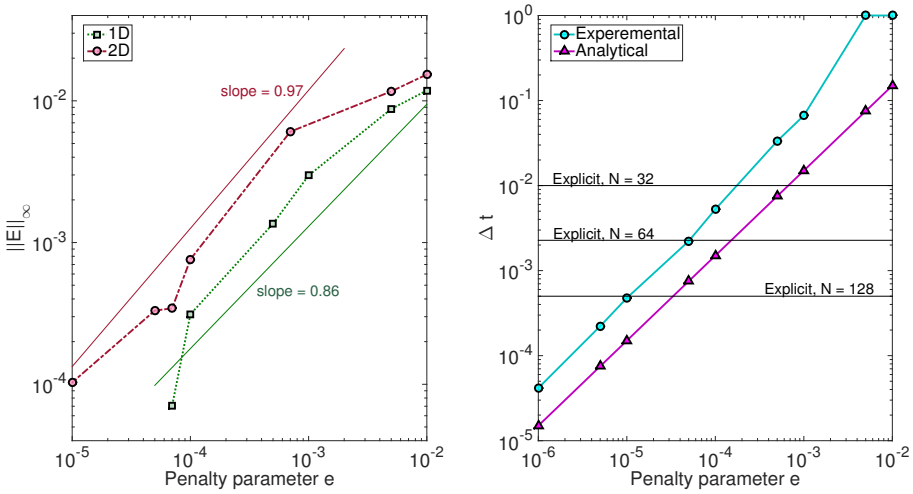

Figure 5: Left: Error measured in l_∞-norm against the penalty parameter size for the one asset and two asset cases. Right: Stable time step size for different sizes of the penalty parameter. Analytical - obtained from inequality (5.7) with $C = 3/2$, Experimental - experimentally obtained maximal time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

quires smoothness of the solution [21]. However, due to the smoothing properties of parabolic problems, the solution can be approximated with high accuracy at larger times [22]. It has been proved in [25], 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 RBF and RBF-PU methods 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 RBF and RBF-PU methods 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 [15]. 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 higher precision in the result is required. More details about RBF-QR can be found in [7, 6, 15].

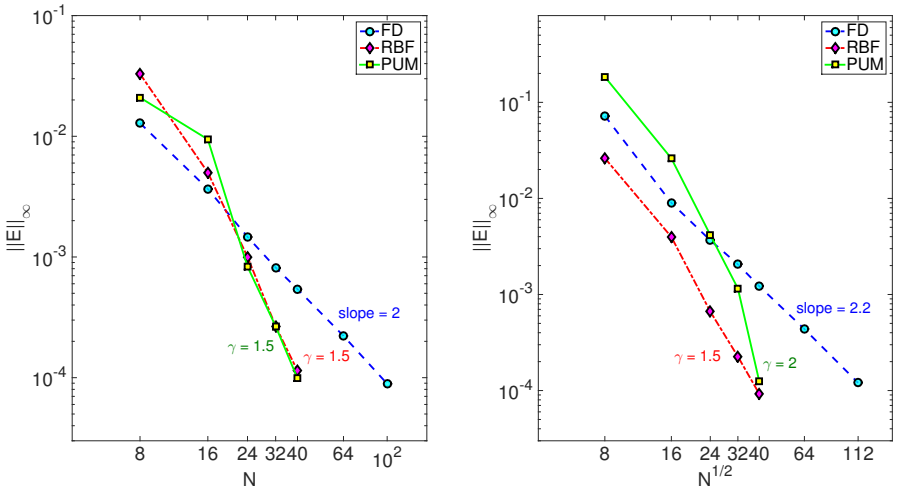

Figure 6: Left: Error convergence in l_∞-norm for a European option on 1 underlying asset. Right: Error convergence in l_∞-norm for a basket European option on 2 underlying asset.

6.6 Convergence study: American option

Here we study the convergence rates of the RBF and RBF-PU 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 [11] on a fine enough grid. Note that the OS method approximates the original PDE, and therefore the error introduced by the penalty term is not present.

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 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}.

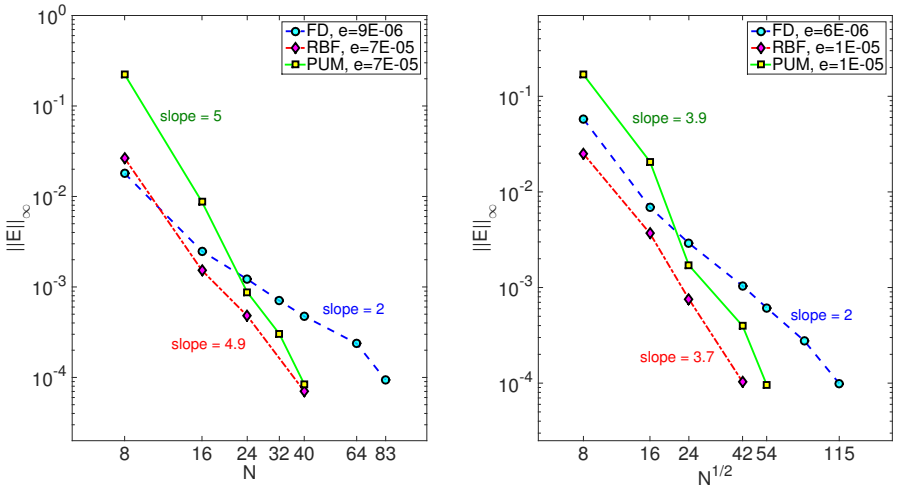

Figure 7: Left: Error convergence in l_∞-norm for an American option on 1 underlying asset. Right: Error convergence in l_∞-norm for a basket American option on 2 underlying asset. All the three methods use the penalty approach.

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 instead 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. Figure 8 shows computational times needed to achieve a certain level of accuracy for the FD, RBF and RBF-PU method for pricing a basket European option. We can see that in order to get to error level 10^{-4}, the RBF-PU method requires approximately 5 times less time than the standard FD method, while the global RBF method needs the same time even though the number of computational nodes is almost 8 times less. This clearly reflects the advantage of the partition of unity technique over the global method. It essentially reduces the computational time while remaining highly accurate.

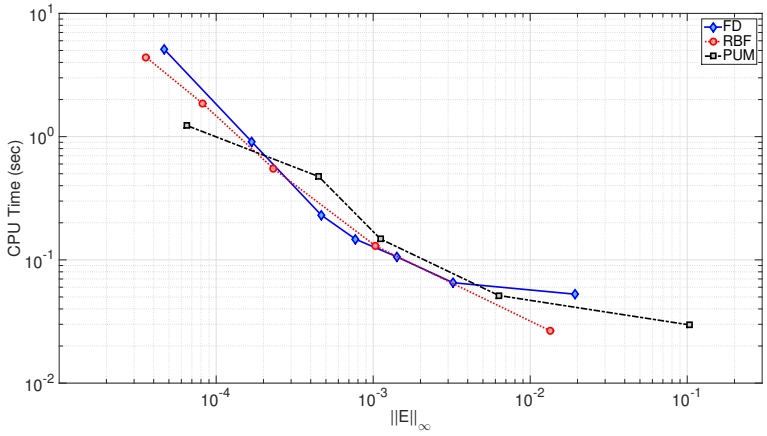

Figure 8: CPU time against error for the FD, RBF and RBF-PU methods for a basket European option.

We should notice that for the experiment we measured only the time of solving the linear system. Preparation procedures for RBF-PUM, such as computation of local matrices and assembling them into a big matrix, may take a little while, but this part can easily be parallelised. Therefore we do not take it into account. The number time steps used was the same in each case.

7 Summary

RBF methods provide an alternative to already existing methods for solving problems in financial applications. The RBF-PU method allows to overcome the high computational cost associated with the global RBF method, while maintaining high accuracy. The RBF-PU method allows to reach a given level of accuracy with less computational effort than the standard FD method. One way to reduce the computational time even more is to use the geometrical flexibility of RBF method. For example, for the European option the problem can be solved on a triangular domain instead of a square domain, thus, halving the problem size.

The fact that RBF methods are mesh-free allows 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. However, if lower errors are required, a different evaluation method, such as for example the RBF-QR method, is needed. Then convergence can be maintained almost down to machine precision [6, 14].

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 the desirable level.

References

[1] I. Babuška and J. M. Melenk, *The partition of unity method*, International Journal of Numerical Methods in Engineering, Vol. 40, Issue 4 (1997), pp. 727-758.

[2] F. Black and M. Scholes, *The Pricing of Options and Corporate Liabilities*, Journal of Political Economy, Vol. 81, No. 3 (1973), pp. 637-654.

[3] R. Courant, *Variational methods for the solution of problems of equilibrium and vibrations*, Bulletin of the American Mathematical Society, Vol. 49, No. 1 (1943), pp. 1-23.

[4] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, World Scientific Publishers, 2007.

[5] G. E. Fasshauer, A. Q. M. Khaliq and D. A. Voss, *Using mesh free approximation for multi-asset American option problems*, Journal of the Chinese Institute of Engineers, Vol. 27, No. 4 (2004), 563-571.

[6] B. Fornberg, E. Larsson and N. Flyer, *Stable computations with Gaussian radial basis functions*, SIAM Journal on Scientific Computing, Vol. 33, No. 2, pp. 869-892.

[7] B. Fornberg and C. Piret, *A stable algorithm for flat radial basis functions on a sphere*, SIAM Journal on Scientific Computing, Vol. 30, No. 1, pp. 60-80.

[8] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer, 2003.

[9] E. Hairer, S. P. Nørsett, G. Wanner, *Solving ordinary differential equations. I. Nonstiff problems.* 2nd edn, Springer Series in Computational Mathematics, 8, Springer-Verlag, Berlin, 1993.

[10] J. C. Hull, *Options, Futures and Other Derivatives*, 7th ed., Pearson Prentice Hall, 2008.

[11] S. Ikonen and J. Toivanen, *Operator Splitting Method for American option Pricing*, Applied Mathematics Letters, Vol. 17, Issue 7 (2004), pp. 809-814.

[12] S. Janson, J. Tysk, *Feynman-Kac formulas for Black-Scholes type operators*, Bulletin of the London Mathematical Society, Vol. 38, Issue 2, pp 269-282.

[13] V. A. Kholodnyi, *A nonlinear partial differential equation for American options in the entire domain of the state variable*, Nonlinear Analysis, Theory, Methods & Applications, Vol. 30, No. 8 (1997), pp. 5059-5070.

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

[15] E. Larsson, E. Lehto, A. Heryudono and B. Fornberg, *Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions*, SIAM Journal on Scientific Computing, Vol. 35, No. 4, pp. 2096-2119

[16] E. Larsson, K. Åhlander, A. Hall *Multi-dimensional option pricing using radial basis functions and the generalized Fourier transform*, Journal of Computational and Applied Mathematics, Vol. 222 (2008), pp. 175192.

[17] B. F. Nielsen, O. Skavhaug and A. Tveito, *Penalty and front-fixing methods for the numerical solution of American option problems*, Journal of Computational Finance, Vol. 5, No. 4 (2002), pp. 69-97.

[18] B. F. Nielsen, O. Skavhaug and A. Tveito, *Penalty methods for the numerical solution of American multi-asset option problems*, Journal of Computational and Applied Mathematics, Vol. 222, Issue 1 (2008), pp 3-16.

[19] J. Persson and L. von Sydow, *Pricing American option using a space-time adaptive finite difference method*, Mathematics and Computers in Simulation, Vol. 80, Issue 9 (2010), pp. 1922-1935.

[20] U. Pettersson, E. Larsson, G. Marcusson and J. Persson, *Improved radial basis function methods for multi-dimensional option pricing*, Journal of Computational and Applied Mathematics, Vol. 222, Issue 1 (2008), pp. 82-93.

[21] C. Rieger, B. Zwicknagl, *Sampling inequalities for infinitely smooth functions, with applications to interpolation and machine learning*, Advances in Computational Mathematics, Vol. 32, Issue 1, pp. 103-129.

[22] A. Safdari-Vaighani, A. Heryudono and E. Larsson, *A radial basis function partition of unity collocation method for convection-diffusion equations*, in print.

[23] D. Shepard, *A two dimensional interpolation function for irregularly space data*, Proceedings of the 23rd ACM national conference (1968), pp. 517-524.

[24] D. Tavella and C. Randall, *Pricing Financial Instruments: The Finite Difference Method*, John Wiley & Sons, 2000.

[25] V. Thomée, *Galerkin Finite Element Methods for Parabolic Problems*, 2nd ed., Springer 2006.

[26] H. Wendland, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*, Advances in Computational Mathematics, Vol. 4, Issue 1 (1995), pp 389-396.

[27] P. Wilmott, J. Dewynne and S. Howison, *Option pricing: Mathematical models and computations*, Oxford Financial, 1998.

[28] R. Zvan, P. A. Forsyth and K. R. Vetzal, *Penalty methods for American options with stochastic volatility*, Journal of Computational and Applied Mathematics, Vol. 91, Issue 2 (1998), pp. 199-218.