Applied Mathematics and Computation 251 (2015) 363–377

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Pricing European and American options by radial basis point interpolation

Jamal Amani Rad [a], Kourosh Parand [a], Luca Vincenzo Ballestra [b,*]

[a] Department of Computer Sciences, Faculty of Mathematical Sciences, Shahid Beheshti University, Evin, P.O. Box 198396-3113, Tehran, Iran
[b] Dipartimento di Economia, Seconda Università di Napoli, Corso Gran Priorato di Malta, 81043 Capua, Italy

ARTICLE INFO

Keywords:
Radial basis point interpolation
Meshfree method
Option pricing
Black–Scholes
Projected successive overrelaxation
Penalty method

ABSTRACT

We propose the use of the meshfree radial basis point interpolation (RBPI) to solve the Black–Scholes model for European and American options. The RBPI meshfree method offers several advantages over the more conventional radial basis function approximation, nevertheless it has never been applied to option pricing, at least to the very best of our knowledge. In this paper the RBPI is combined with several numerical techniques, namely: an exponential change of variables, which allows us to approximate the option prices on their whole spatial domain, a mesh refinement algorithm, which turns out to be very suitable for dealing with the non-smooth options' payoff, and an implicit Euler Richardson extrapolated scheme, which provides a satisfactory level of time accuracy. Finally, in order to solve the free boundary problem that arises in the case of American options three different approaches are used and compared: the projected successive overrelaxation method (PSOR), the Bermudan approximation, and the penalty approach. Numerical experiments are presented which demonstrate the computational efficiency of the RBPI and the effectiveness of the various techniques employed.

1. Introduction

Over the last thirty years, financial derivatives have raised increasing popularity in the markets. In particular, large volumes of options are traded everyday all over the world and it is therefore of great importance to give a correct valuation of these instruments.

Options are contracts that give to the holder the right to buy (call) or to sell (put) an asset (underlying) at a previously agreed price (strike price) on or before a given expiration date (maturity). The majority of options can be grouped in two categories: European options, which can be exercised only at maturity, and American options, which can be exercised not only at maturity but also at any time prior to maturity.

Options are priced using mathematical models that are often challenging to solve. In particular, the famous Black–Scholes model [1] yields explicit pricing formulae for some kinds of European options, including vanilla call and put, but for American options closed-form solutions are not available, and numerical approximations are needed. To this aim, the most common approaches are the finite difference/finite element/finite volume methods (see, e.g., [2–18]) and the binomial/

* Corresponding author.
E-mail addresses: j.amanirad@gmail.com, j_amanirad@sbu.ac.ir (J.A. Rad), k_parand@sbu.ac.ir (K. Parand), lucavincenzo.ballestra@unina2.it (L.V. Ballestra).

http://dx.doi.org/10.1016/j.amc.2014.11.016
0096-3003/

trinomial trees (see, e.g., [19–22]), nevertheless some authors have also proposed the use of meshfree algorithms based on radial basis functions [23–28] and on quasi radial basis functions [29].

Meshfree methods are a very powerful tool for solving partial differential equations as they yield spectral accuracy (see [30,31]), are very easy to implement (mesh generation is avoided), and can be applied with any (also very irregular) node distribution. In the technical literature several meshfree discretization techniques have been proposed and for a comprehensive description of them the interested reader is referred to [32].

One of the most common meshfree approaches is the so-called point interpolation method (PIM), see, e.g., [32], which is often based on two different types of interpolations: Polynomial basis point interpolation (PBPI) [33,34], and radial basis point interpolation (RBPI) [35,36]. In particular, the PBPI is one of the earliest interpolation schemes with the so-called Kronecker property, which allows one to easily impose essential boundary and initial (or final) conditions. Unfortunately, the PBPI has the disadvantage of requiring special care in choosing the polynomial basis (see [35]). In particular, if an inappropriate basis is employed, then the resulting system of linear equations can be seriously ill conditioned.

The RBPI, originally proposed by Liu et al. [32,35], employs a basis of both polynomials and radial functions (see, e.g., [37,38]). Such an approach retains the Kronecker property, but is more stable than the PBPI and also more flexible for arbitrary node distributions. RBPI methods have also been employed, for example, in [39,40].

Another meshfree discretization scheme has been recently developed by Krige [41]. This technique, which is referred to as Kriging interpolation, is a sort of generalized linear regression that aims at finding the best approximation in a statistical sense. More precisely, the interpolation error is required to be null in the mean and minimum in the variance. For a detailed description of such an approach the reader is referred to [42–45]. Here we simply do mention that, as shown in [42], the Kriging method can be made identical to the PIM by properly choosing the statistical measure (semivariogram) used to compute the variance of the interpolation error.

To the best of our knowledge, the PIM has not yet been used in mathematical finance. Therefore, it appears to be interesting to extend such a numerical technique also to option valuation, which is done in the present manuscript. In particular, we develop a RBPI algorithm for pricing both European and American options under the Black–Scholes model. Note that, with respect to the RBF meshfree methods employed in [23–28], the RBPI offers the following advantages: First of all, it incorporates polynomial terms in the basis, which are useful to reduce the ill-conditioning of the resulting linear systems. Second, it possesses the Kronecker property, so that essential boundary and initial (or final) conditions can be easily imposed. Third, the RBPI can be reinterpreted as a Kriging method, and thus its interpolation error can be considered optimal also from a statistical point of view.

In addition, in this paper the RBPI is used in conjunction with a suitable change of variables, which allows us to approximate the option price on its whole semi-infinite spatial domain. This is a remarkable difference with previous methods that replace the original domain with a finite one, or introduce unknown finite boundaries and prescribe artificial conditions there. Furthermore, by exploiting the great flexibility of the RBPI to any (also very irregular) node distribution, a local mesh refinement strategy is employed which allows us to easily and effectively handle the non-smoothness of the options' payoff (which is not differentiable at the strike price).

As far as the time discretization is concerned, we use the implicit Euler method, which is unconditionally stable [46] and allows us to smooth the discontinuities of the options' payoffs (see, e.g., [47]). Such an approach is only first-order accurate, however a second-order time discretization is obtained by performing a Richardson extrapolation procedure with halved time step.

Finally, in order to solve the free boundary problem that arises in the case of American options, three different approaches are used and compared: the so-called projected successive overrelaxation (PSOR) method (see, for example, [48–50]), a time discretization scheme which amounts to replacing the price of the American option with prices of Bermudan options (see [8,9,22,24,51]), and the penalty approach (see, for example, [6,10,15,52,53]).

Numerical experiments are presented showing that the proposed approach is very efficient from the computational standpoint. In particular, the prices of both European and American options can be computed with an error (in both the maximum norm and the root mean square norm) of order 10^{-4} or 10^{-5} in few hundredth of a second. Moreover, the Bermudan approximation reveals to be the most efficient of the three algorithms used to deal with the early exercise opportunity, whereas the penalty method turns out to be the less accurate and fast (see SubSection 5).

We remark that the main contribution of this manuscript is to show that the RBPI, which, to the best of our knowledge, has never been applied to problems in mathematical finance, can yield accurate and fast approximations of European and American option prices. In particular, in order to efficiently handle the several peculiarities of the problems considered, the RBPI is coupled with other numerical procedures, namely an implicit Euler Richardson extrapolated time stepping, a local mesh refinement, a change of variables that allows us to cope with the unboundedness of the price domain, and a suitable method to take into account the early exercise opportunity. Note that none of these techniques, if considered separately, is new, nevertheless the combination of all of them has never been employed for option pricing. Furthermore, in the present manuscript we also test and compare three of the main approaches to deal with the early exercise feature typical of American options (the Bermudan approximation, the PSOR and the penalty method), which has not yet been done in the context of meshless approximations.

The paper is organized as follows: In Section 2 a detailed description of the Black–Scholes model for European and American options is provided. Section 3 is devoted to presenting the RBPI approach. The application of such a numerical technique to the option pricing problems considered is shown in Section 4. The numerical results obtained are presented

and discussed in Section 5. Finally, in Section 6 the conclusions are drawn and some possible directions for future research are briefly outlined.

2. The Black–Scholes model for European and American options

For the sake of simplicity, from now we restrict our attention to options of put type, but the case of call options can be treated in perfect analogy.

Let us consider a put option with maturity T and strike price E on an underlying asset s that follows (under the risk-neutral measure) the stochastic differential equation (geometric Brownian motion):

$$ds = rsdt + \sigma sdW, \tag{2.1}$$

where r and σ are the interest rate and the volatility, respectively. Moreover let $V(s,t)$ denote the option price, and let us define the Black–Scholes operator:

$$\mathcal{L}V(s,t) = -\frac{\partial}{\partial t}V(s,t) - \frac{\sigma^2}{2}s^2\frac{\partial^2}{\partial s^2}V(s,t) - rs\frac{\partial}{\partial s}V(s,t) + rV(s,t). \tag{2.2}$$

2.1. European option

The option price $V(s,t)$ satisfies, for $s \in (0,+\infty)$ and $t \in [0,T)$, the following partial differential problem:

$$\mathcal{L}V(s,t) = 0, \tag{2.3}$$

with final condition:

$$V(s,T) = \varsigma(s) \tag{2.4}$$

and boundary conditions:

$$V(0,t) = E\exp(-r(T-t)), \quad \lim_{s\to+\infty} V(s,t) = 0, \tag{2.5}$$

where ς is the so-called option's payoff:

$$\varsigma(s) = \max(E-s,0), \tag{2.6}$$

which is clearly not differentiable at $s = E$.

An exact analytical solution to the problem (2.3)–(2.5), i.e. the famous Black–Scholes formula, is available.

2.2. American option

The option price $V(s,t)$ satisfies, for $s \in [0,+\infty)$ and $t \in [0,T)$, the following partial differential problem:

$$\mathcal{L}V(s,t) = 0, \quad s > B(t), \tag{2.7}$$

$$V(s,t) = E - s, \quad 0 \leqslant s < B(t), \tag{2.8}$$

$$\left.\frac{\partial V(s,t)}{\partial s}\right|_{s=B(t)} = -1, \tag{2.9}$$

$$V(B(t),t) = E - B(t) \tag{2.10}$$

with final condition:

$$V(s,T) = \varsigma(s) \tag{2.11}$$

and boundary condition:

$$\lim_{s\to+\infty} V(s,t) = 0, \tag{2.12}$$

where $B(t)$ denotes the so-called exercise boundary, which is unknown and is implicitly defined by (2.7)–(2.12). The above free-boundary partial differential problem does not have an exact closed-form solution, and thus some numerical approximation is required.

Problem (2.7)–(2.12) can be reformulated as a linear complementarity problem:

$$\mathcal{L}V(s,t) \geqslant 0, \tag{2.13}$$

$$V(s,t) - \varsigma(s) \geqslant 0, \tag{2.14}$$

$$(\mathcal{L}V(s,t)) \cdot (V(s,t) - \varsigma(s)) = 0, \tag{2.15}$$

which holds for $s \in (0, +\infty)$ and $t \in [0, T)$, with final condition:

$$V(s,T) = \varsigma(s) \tag{2.16}$$

and boundary conditions:

$$V(0,t) = E, \quad \lim_{s \to +\infty} V(s,t) = 0. \tag{2.17}$$

Let us recall that the function ς is the option's payoff given by (2.6). Problem (2.13)–(2.17) can be solved using a penalty approach [10,52], which amounts to computing $V(s,t)$ as the solution of the following problem:

$$\mathcal{L}V(s,t) - \frac{C\varepsilon}{V(s,t) + \varepsilon - E + s} = 0, \tag{2.18}$$

$$V(s,T) = \varsigma(s), \tag{2.19}$$

$$V(0,t) = E, \quad \lim_{s \to +\infty} V(s,t) = 0. \tag{2.20}$$

Note that in (2.18) ε is a (small) positive constant and $C \geqslant rE$ is a constant (both ε and C will be chosen in Section 5).

3. Methodology

3.1. Point interpolation method (PIM)

Let $u : \mathbb{R} \to \mathbb{R}$ be a generic function. According to the PIM, the value of u at any (given) point $x \in \mathbb{R}$ is approximated by interpolation at $n+1$ scattered nodes $x_0, x_1, \ldots, x_n$ (centers). Various different PIM approaches can be obtained depending on the functions used to interpolate u. In this paper we focus our attention onto the so-called radial basis point interpolation method (RBPI), which employs a combination of polynomials and radial basis functions.

3.1.1. Radial basis point interpolation (RBPI) method

The function that interpolates u, which we denote by u_{RBPI}, is obtained as follows:

$$u_{RBPI}(x) = \sum_{i=0}^{n} R_i(x) a_i + \sum_{j=0}^{m} P_j(x) b_j, \tag{3.1}$$

where $P_0, P_1, \ldots, P_m$ denote the first $m+1$ monomials in ascending order (i.e. $P_0 = 1,\ P_1 = x, \ldots, P_m = x^m$) and $R_0, R_1, \ldots, R_n$ are $n+1$ radial functions centered at $x_0, x_1, \ldots, x_n$, respectively. Moreover $a_0, a_1, \ldots, a_n, b_0, b_1, \ldots, b_m$ are $n+m+2$ real coefficients that have to be determined.

As far as the radial basis functions $R_0, R_1, \ldots, R_n$ are concerned, several choices are possible, such as the so-called multiquadrics, inverse multiquadrics, Gaussian RBFs or thin plate splines (see, for example, [54]). In particular, the multiquadrics, the inverse multiquadrics and the Gaussian RBFs contain a free shape parameter on which the performances of the RBF approximation strongly depend. Precisely, values of the shape parameters that yield a high spatial resolution (i.e. a high level of accuracy) also lead to severely ill-conditioned linear systems. Therefore, one has to find a value of the shape parameter such that a high level of accuracy is obtained and at the same time the numerical approximation does not blow up (due to ill-conditioning problems). Now, in the technical literature, various approaches for selecting the RBF shape parameter have been proposed, see, e.g., [55,36,24,56–64]. These algorithms, which are often based on rules of thumbs or on semi-analytical relations, can yield satisfactory results in some circumstances. Nevertheless, to the best of our knowledge, a method to choose the RBF shape parameter which is rigorously established and proven to perform well in the general case is still lacking.

Therefore, in the present work we decide to use the thin plate splines, as they do not involve any free shape parameter. In particular, we employ the very popular thin plate splines of second order, which are as follows:

$$R_i(x) = (x - x_i)^4 \log(|x - x_i|), \quad i = 0, 1, \ldots, n. \tag{3.2}$$

Note that the monomials $P_0, P_1, \ldots, P_m$ are not always employed (if $b_i = 0,\ i = 0, 1, \ldots, m$, pure RBF approximation is obtained). However, in the case of thin plate splines, their presence guarantees that the resulting interpolation matrix, i.e. the matrix G used below, is non-singular [65]. In the present work, both the constant and the linear monomials are used to augment the RBFs (i.e. we set $m = 1$).

By requiring that the function u_{RBPI} interpolate u at $x_0, x_1, \ldots, x_n$, we obtain a set of $n+1$ equations in the $n+m+2$ unknown coefficients $a_0, a_1, \ldots, a_n, b_0, b_1, \ldots, b_m$:

$$\sum_{i=0}^{n} R_i(x_k) a_i + \sum_{j=0}^{m} P_j(x_k) b_j = u_k, \quad k = 0, 1, \ldots, n. \tag{3.3}$$

Moreover, in order to uniquely determine u_{RBPI}, we also impose:

$$\sum_{i=0}^{n} P_j(x_i)a_i = 0, \quad j = 0, 1, \ldots, m. \tag{3.4}$$

That is we have the following system of linear equations:

$$\mathbf{G}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix} = \begin{bmatrix}\mathbf{U}\\ \mathbf{0}\end{bmatrix},$$

where

$$\mathbf{U} = [\,u_0 \quad u_1 \quad \ldots \quad u_n\,]^T = [\,u(x_0) \quad u(x_1) \quad \ldots \quad u(x_n)\,]^T, \tag{3.5}$$

$$\mathbf{G} = \begin{bmatrix}\mathbf{R} & \mathbf{P}\\ \mathbf{P}^T & \mathbf{0}\end{bmatrix},$$

$$\mathbf{R} = \begin{bmatrix} R_0(x_0) & R_1(x_0) & \ldots & R_n(x_0)\\ R_0(x_1) & R_1(x_1) & \ldots & R_n(x_1)\\ \vdots & \vdots & \ddots & \vdots\\ R_0(x_n) & R_1(x_n) & \ldots & R_n(x_n)\end{bmatrix},$$

$$\mathbf{P} = \begin{bmatrix} P_0(x_0) & P_1(x_0) & \ldots & P_m(x_0)\\ P_0(x_1) & P_1(x_1) & \ldots & P_m(x_1)\\ \vdots & \vdots & \ddots & \vdots\\ P_0(x_n) & P_1(x_n) & \ldots & P_m(x_n)\end{bmatrix},$$

$$\mathbf{a} = [a_0 \quad a_1 \quad \ldots \quad a_m]^T, \tag{3.6}$$

$$\mathbf{b} = [b_0 \quad b_1 \quad \ldots \quad b_m]^T, \tag{3.7}$$

As already mentioned, the matrix $\mathbf{R}$ is non-singular, so that

$$\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix} = \mathbf{G}^{-1}\begin{bmatrix}\mathbf{U}\\ \mathbf{0}\end{bmatrix}.$$

Accordingly, (3.1) can be rewritten as

$$u_{RBPI}(x) = [\mathbf{R}^T(x) \quad \mathbf{P}^T(x)]\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix},$$

or, equivalently,

$$u_{RBPI}(x) = [\mathbf{R}^T(x) \quad \mathbf{P}^T(x)]\mathbf{G}^{-1}\begin{bmatrix}\mathbf{U}\\ \mathbf{0}\end{bmatrix}. \tag{3.8}$$

Let us define the vector of shape functions:

$$\mathbf{\Phi}(x) = [\varphi_0(x) \quad \varphi_1(x) \quad \ldots \quad \varphi_n(x)],$$

where

$$\varphi_k(x) = \sum_{i=0}^{n} R_i(x)\mathbf{G}^{-1}_{i+1,k+1} + \sum_{j=0}^{m} P_j(x)\mathbf{G}^{-1}_{n+j+2,k+1}, \quad k = 0, 1, \ldots, n \tag{3.9}$$

and $\mathbf{G}^{-1}_{i,k}$ is the (i,k) element of the matrix $\mathbf{G}^{-1}$.

Using (3.9) relations (3.8) are rewritten in the more compact form:

$$u_{RBPI}(x) = \mathbf{\Phi}(x)\mathbf{U}, \tag{3.10}$$

or, equivalently,

$$u_{RBPI}(x) = \sum_{i=0}^{n} u_i\phi_i(x). \tag{3.11}$$

It can be easily shown that the shape functions (3.9) satisfy the so-called Kronecker property, that is

$$\varphi_i(x_j) = \delta_{ij}, \tag{3.12}$$

where δ_{ij} is the well-known Kronecker symbol, so that essential boundary and final conditions such as those considered in Section 2 (e.g., (2.4), (2.5), (2.16), (2.17), (2.20), (2.21)) can be easily imposed. Note also that the derivatives of u_{RBPI} (of any order) with respect to x are easily obtained by direct differentiation in (3.11).

4. Numerical implementation of the proposed methods

Let us show how to apply the RBPI described above to the option pricing problems considered in Section 2.

4.1. Spatial change of variables

For both the European and the American options the underlying asset price s can take any value in $[0,+\infty)$. In this paper, in order to numerically handle the unboundedness of the s-domain, we employ the following change of variables:

$$x(s)=1-\exp\left(-\frac{s}{L}\right),\quad s(x)=-L\log(1-x), \tag{4.1}$$

where the positive constant parameter L is the characteristic length of the mapping and will be chosen below.

Trivially, we have $x(0)=0$, $\lim_{s\to+\infty}x(s)=1$ and for $0<x<1$

$$\frac{dx}{ds}=\frac{1-x}{L}>0.$$

The exponential change of variables (4.1) transforms the s-domain $[0,+\infty)$ into the x-domain [0,1), so that we can easily choose a finite number of equally spaced RBPI centers in [0,1] (which would not be possible in $[0,+\infty)$). We define

$$U(x,t)=V(s(x),t), \tag{4.2}$$

$$\tilde{\mathcal{L}}U(x,t)=-\frac{\partial}{\partial t}U(x,t)+\mathcal{A}(x)\frac{\partial^2}{\partial x^2}U(x,t)+\mathcal{B}(x)\frac{\partial}{\partial x}U(x,t)+rU(x,t), \tag{4.3}$$

where

$$\mathcal{A}(x)=-\frac{\sigma^2}{2}(1-x)^2\log^2(1-x),$$
$$\mathcal{B}(x)=-\frac{\sigma^2}{2}(x-1)\log^2(1-x)+r(1-x)\log(1-x).$$

Using the change of variables (4.1)–(4.3) the European option problem (2.3)–(2.5) is rewritten as follows:

$$\begin{cases}\tilde{\mathcal{L}}U(x,t)=0,\\ U(x,T)=\tilde{\varsigma}(x),\\ U(0,t)=E\exp\left(-r(T-t)\right),\quad U(1,t)=0,\end{cases} \tag{4.4}$$

where

$$\tilde{\varsigma}(x)=\max(E+L\log(1-x),0). \tag{4.5}$$

Accordingly, the American option problems (2.13)–(2.17) and (2.18)–(2.21) are rewritten as follows:

$$\begin{cases}\tilde{\mathcal{L}}U(x,t)\geqslant 0,\\ U(x,t)-\tilde{\varsigma}(x)\geqslant 0,\\ \left(\tilde{\mathcal{L}}U(x,t)\right)\cdot\left(U(x,t)-\tilde{\varsigma}(x)\right)=0,\\ U(x,T)=\tilde{\varsigma}(x),\\ U(0,t)=E,\quad U(1,t)=0,\end{cases} \tag{4.6}$$

$$\begin{cases}\tilde{\mathcal{L}}U(x,t)-\dfrac{C\varepsilon}{U(x,t)+\varepsilon-E-L\log(1-x)}=0,\\ U(x,T)=\tilde{\varsigma}(x),\\ U(0,t)=E,\quad U(1,t)=0,\end{cases} \tag{4.7}$$

respectively, where

$$\tilde{B}(t)=1-\exp\left(-\frac{B(t)}{L}\right). \tag{4.8}$$

4.2. Time discretization of the Black–Scholes operator

First of all, we discretize the Black–Scholes operator (4.3) in time. In $[0,T]$ let us consider $M+1$ times $t_0, t_1, \ldots, t_M$, such that $t_k = k\Delta t,\ \Delta t = \frac{T}{M}$. Then, we set $U^k(x) = U(x, k\Delta t)$, $k = 0, 1, \ldots, M$. Let us consider the following θ-weighted scheme:

$$\tilde{\mathcal{L}}U^k(x) = -\frac{1}{\Delta t}(U^{k+1}(x) - U^k(x)) + \theta\mathcal{A}(x)U^{k+1}_{xx}(x) + \theta\mathcal{B}(x)U^{k+1}_{x}(x) + \theta rU^{k+1}(x) \tag{4.9}$$

$$+(1-\theta)\mathcal{A}(x)U^{k}_{xx}(x) + (1-\theta)\mathcal{B}(x)U^{k}_{x}(x) + (1-\theta)rU^{k}(x). \tag{4.10}$$

The popular implicit Euler and Crank–Nicolson schemes are obtained by choosing $\theta = 0$ and $\theta = 1/2$, respectively. Now, in [47] it is shown that the Crank–Nicolson scheme fails to achieve its usual second-order accuracy, due to the non-smoothness of the options' payoff. Therefore, in this work, following a common approach, we use the implicit Euler scheme, which is unconditionally stable [46] and allows us to smooth the discontinuities of the options' payoffs (see, e.g., [47]). That is we apply (4.9) with $\theta = 0$:

$$\tilde{\mathcal{L}}U^k(x) = -\frac{1}{\Delta t}(U^{k+1}(x) - U^k(x)) + \mathcal{A}(x)U^{k}_{xx}(x) + \mathcal{B}(x)U^{k}_{x}(x) + rU^{k}(x). \tag{4.11}$$

Then, the approximation (4.11), which is only first-order accurate, is improved by Richardson extrapolation. In particular, we manage to obtain second-order accuracy by extrapolation of two solutions computed using M and $2M$ time steps (see, for example, [8,9,24]).

In the following, for the sake of brevity, we will restrict our attention to first stage of the Richardson extrapolation procedure, where M time steps are employed, and the fact that the partial differential problems considered are also solved with $2M$ time steps will be understood.

4.3. RBPI discretization

In the interval $[0,1]$ let us consider a set of $n+1$ equally spaced points $x_i = i\Delta x,\ i = 0, 1, \ldots, n,\ \Delta x = \frac{1}{n}$. Then based on (3.11) we employ the following approximation U^k_{RBPI} of U^k:

$$U^k_{RBPI}(x) = \sum_{j=0}^{n} \lambda^k_j \varphi_j(x), \quad k = 0, 1, \ldots, M, \tag{4.12}$$

where $\varphi_j\ \ j = 0, 1, \ldots, n$, are the PIM shape functions, which are computed using (3.9) and the coefficients $\lambda^k_j,\ j = 0, 1, \ldots, n,\ k = 0, 1, \ldots, M$ are still to be determined.

Now we are in the position to select the length scale parameter L needed in (4.1). First of all, we note that relations (4.1) map the equally spaced nodes $x_0, x_1, \ldots, x_n$ in the x-domain [0,1] to nodes $s_0, s_1, \ldots, s_n$ in the s-domain $[0, +\infty)$, where $s_i = s(x_i),\ i = 0, 1, \ldots, n$ and $s(\cdot)$ is given by the second of (4.1) (to simplify things we adopt the convention that $s_n = +\infty$). Now, for the sake of computational efficiency, we would like that almost all the centers $s_0, s_1, \ldots, s_n$ be located in the interval $[0, 2E]$, where the option prices take values significantly different from zero (see [50,66]). To this aim, we note that the change of variables (4.1) transforms the strike price E to the point $1 - \exp\left(-\frac{E}{L}\right)$ in the x-domain [0,1]. Therefore, we choose the parameter L such that the seventy percent of the RBPI centers $x_0, x_1, \ldots, x_n$ lay in the interval $\left[0, 1 - \exp\left(-\frac{E}{L}\right)\right]$. By doing that, we clearly have that the seventy percent of the centers $s_0, s_1, \ldots, s_n$ lay in the interval $[0, E]$. However, thanks to the continuity of the function $s(\cdot)$, we also obtain that a significant fraction of centers are located in the interval $[E, 2E]$ (see Fig. 1). As a result, almost all of the RBPI centers $s_0, s_1, \ldots, s_n$ are placed in the interval $[0, 2E]$ (but, clearly, we will also have some small fraction of centers in the interval $[2E, +\infty)$, so that the option price is actually approximated on its whole spatial domain).

So, we choose L such that

$$1 - \exp\left(-\frac{E}{L}\right) = w\Delta x, \tag{4.13}$$

where w is set equal to $\frac{7}{10\Delta x}$. Note that L is readily obtained from (4.13):

$$L = -\frac{E}{\log(1 - w\Delta x)}. \tag{4.14}$$

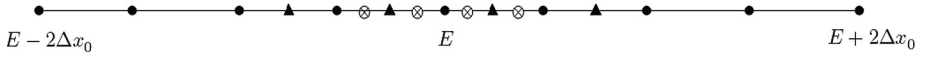

Fig. 1. Local mesh refinement.

Remark 1. The numerical method proposed in this paper approximates the option price on its whole semi-infinite spatial domain. This is a remarkable difference with previous approaches that replace the original domain with a finite one, or introduce unknown finite boundaries and prescribe artificial conditions there.

4.4. Local mesh refinement

The options' payoffs considered in this paper are non-smooth functions, in particular their derivatives are discontinuous at the strike price. Therefore, for the sake of computational efficiency, a refined mesh is used in the neighborhood of the strike price, i.e. according to (4.13), in the neighborhood of $w\Delta x$. Precisely, we apply a local mesh refinement technique similar to that employed in [67,68], which is conveniently described by means of the following pseudocode:

```
begin
  Set Δx_0 ← Δx and k ← 1
  repeat for a desired number of times
    (1) Δx_k ← Δx_{k-1}/2
    (2) Consider the subdomain Ω_k = (wΔx − 2Δx_{k-1}, wΔx + 2Δx_{k-1})
    (3) In the interior of Ω_k add 7 equally spaced nodes
    for i = 1, 2, ..., 7
      x_{i,k} = wΔx − 2Δx_{k-1} + iΔx_k,
    end
    k ← k + 1
  end repeat
end
```

The RBPI centers obtained after performing the first three iterations of this algorithm are illustrated in Fig. 1. For the sake of simplicity, from now, with abuse of notation, $x_0, x_1, \ldots, x_n$ will denote all the RBPI centers employed (including those obtained by mesh refinement), and not only the equally space ones defined previously.

4.5. European option

Let us consider problem (4.4). Following a standard procedure (see [69]), we evaluate these equations at $t = t_k,\ k = 0, 1, \ldots, M$, and substitute the Black–Scholes operator (4.3) with the time discretized operator (4.11). Then, we substitute $U^k(x)$ with $U^k_{RBPI}(x)$ (given by (4.12)) and evaluate the equations obtained at $x = x_i,\ i = 0, 1, \ldots, n$.

We end up with the following systems of linear equations:

$$\mathbf{B}\boldsymbol{\Lambda}^k = \mathbf{A}\boldsymbol{\Lambda}^{k+1} + \mathbf{H}^k, \tag{4.15}$$

to be recursively solved for $k = M-1, M-2, \ldots, 0$, starting from

$$\boldsymbol{\Lambda}^M = \boldsymbol{\Pi}, \tag{4.16}$$

where

$$\boldsymbol{\Lambda}^k = [\,\lambda_0^k \quad \lambda_1^k \quad \ldots \quad \lambda_n^k\,]^T, \tag{4.17}$$

$$\mathbf{A} = \frac{1}{\Delta t}\boldsymbol{\Phi}_1, \tag{4.18}$$

$$\mathbf{B} = \mathbf{M}\nabla^2\boldsymbol{\Phi}_1 + \mathbf{N}\nabla\boldsymbol{\Phi}_1 + \left(r + \frac{1}{\Delta t}\right)\boldsymbol{\Phi}_1 + \boldsymbol{\Phi}_2, \tag{4.19}$$

$$\mathbf{H}^k = [\,U^k(0) \quad 0 \quad 0 \quad \ldots \quad U^k(1)\,]^T, \tag{4.20}$$

$$\mathbf{M} = \mathrm{Diag}(0, \mathcal{A}(x_1), \mathcal{A}(x_2), \ldots, \mathcal{A}(x_{n-1}), 0),$$

$$\mathbf{N} = \mathrm{Diag}(0, \mathcal{B}(x_1), \mathcal{B}(x_2), \ldots, \mathcal{B}(x_{n-1}), 0),$$

$$\boldsymbol{\Phi}_1 = [\,\mathbf{0}_{(n+1)\times 1} \quad \psi_1 \quad \psi_2 \quad \ldots \quad \psi_{n-1} \quad \mathbf{0}_{(n+1)\times 1}\,]^T = \mathbf{I}_{(n+1)\times(n+1)} - \boldsymbol{\Phi}_2, \tag{4.21}$$

$$\boldsymbol{\Phi}_2 = [\,\psi_0 \quad \mathbf{0}_{(n+1)\times(n-1)} \quad \psi_n\,]^T = \mathrm{Diag}(1, \overbrace{0, 0, \ldots, 0, 0}^{n-1}, 1),$$

$$\psi_i = [\,\varphi_0(x_i) \quad \varphi_1(x_i) \quad \ldots \quad \varphi_n(x_i)\,]^T = [\,\delta_{i0} \quad \delta_{i1} \quad \ldots \quad \delta_{in}\,]^T,$$

$$\boldsymbol{\Pi} = [\,\tilde{\varsigma}(x_0) \quad \tilde{\varsigma}(x_1) \quad \ldots \quad \tilde{\varsigma}(x_n)\,]^T, \tag{4.22}$$

In (4.19), the symbols ∇ and ∇^2 mean that the functions $\varphi_0, \varphi_1, \ldots \varphi_n$ in $\mathbf{\Phi}_1$ has to be differentiated one and two times, respectively, with respect to their argument. Moreover, in (4.20), according to the boundary conditions given by the third of relations (4.4), we have $U^k(0) = E\exp(-r(T-t_k))$ and $U^k(1) = 0,\ k = 0, 1, \ldots, M-1$.

4.6. American option

The American option price is computed using three different algorithms. Two of them (Algorithm 1 and Algorithm 2) are obtained by applying two different discretization approaches to the linear complementarity problem (4.6), whereas the third one stems from the numerical approximation of problem (4.7).

4.6.1. Algorithm 1

We consider problem (4.6) and we apply to it the same numerical procedure used in the previous subsection. That is we evaluate (4.6) at $t = t_k,\ k = 0, 1, \ldots, M$, and substitute the Black–Scholes operator (4.3) with (4.11). Then, we substitute $U^k(x)$ with $U^k_{RBPI}(x)$ (given by (4.12)) and evaluate the equations obtained at $x = x_i,\ i = 0, 1, \ldots, n$. This yields the following systems:

$$\begin{cases} \mathbf{B}\mathbf{\Lambda}^k \geqslant \mathbf{A}\mathbf{\Lambda}^{k+1} + \mathbf{H}^k, \\ \mathbf{I}_{(n+1)\times(n+1)}\mathbf{\Lambda}^k \geqslant \mathbf{\Pi}, \\ \left[\mathbf{B}\mathbf{\Lambda}^k - \mathbf{A}\mathbf{\Lambda}^{k+1} - \mathbf{H}^k\right]^T \cdot \left[\mathbf{I}_{(n+1)\times(n+1)}\mathbf{\Lambda}^k - \mathbf{\Pi}\right] = 0, \end{cases} \tag{4.23}$$

to be recursively solved for $k = M-1, M-2, \ldots, 0$, starting from

$$\mathbf{\Lambda}^M = \mathbf{\Pi}, \tag{4.24}$$

where $\mathbf{\Lambda}^k$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{H}^k$ and $\mathbf{\Pi}$ are defined as in (4.17)–(4.20), (4.22), respectively. Note that, according to (4.20), $\mathbf{H}^k$ involves $U^k(0)$ and $U^k(1)$ which, by taking into account the boundary conditions given by the fifth of relations (4.6), are computed as follows: $U^k(0) = E,\ U^k(1) = 0,\ k = 0, 1, \ldots, M-1$.

Following a rather common approach (see, e.g., [50,49]), problem (4.23) and (4.24) is solved using the projected successive overrelaxation (PSOR) method, which is conveniently described by means of the pseudocode:

```
begin
  Set v ← 0
  Set Λ^{k,v} ← Λ^{k+1}
  Do
    b^k = AΛ^{k+1} + H^k,
    for i = 1, 2, ..., n + 1
      Γ_i^{v+1} = (1/B_ii) (b_i^k − Σ_{j=1, j>i}^{n+1} B_ij Λ_i^{k,v} − Σ_{j=1, j<i}^{n+1} B_ij Λ_i^{k,v+1}),
      Λ_i^{k,v+1} = max (Λ_i^{k,v} + ω(Γ_i^{v+1} − Λ_i^{k,v}), Π_i),
    end
    v ← v + 1,
  while ‖Λ^{k,v+1} − Λ^{k,v}‖_2 < tolerance;  tolerance = 10^{−6}
  Λ^k = Λ^{k,v+1},
end
```

In the above pseudocode ω denotes a constant parameter, which, by employing a standard rule [46], is selected as follows:

$$\omega = \frac{2}{1+\sqrt{1-\rho^2}}, \tag{4.25}$$

where ρ is the spectral radius of the matrix $\mathbf{D}^{-1}(\mathbf{B}-\mathbf{D})$, and $\mathbf{D}$ is the diagonal matrix whose diagonal is equal to the diagonal of $\mathbf{B}$. In this paper, using the Gershgorin theorem [46], ρ is estimated as follows:

$$\rho = \max_{i=1,2,\ldots,n+1} \frac{1}{\mathbf{B}_{ii}} \sum_{j=1\, j\neq i}^{n+1} |\mathbf{B}_{ij}|. \tag{4.26}$$

4.6.2. Algorithm 2

The second Algorithm we use to compute the American option price stems from a time discretization of problem (4.6) which is rather easy to implement (see [8,9,22,24,51]). In particular, we approximate the price of the American option with

the price of a Bermudan option, that is an option that can be exercised not on the whole time interval $[0,T]$, but only at the dates $t_0,t_1,\ldots,t_M$.

More precisely, we assume that in each time interval $(t_k,t_{k+1}),\ k=0,1,\ldots,M-1$, the first of relations (4.6) holds true with the equality sign, together with its boundary conditions given by the fifth of (4.6). That is we consider the problems

$$\begin{cases} \tilde{\mathcal{L}}U(x,t)=0, \\ U(0,t)=E, \quad U(1,t)=0, \end{cases} \tag{4.27}$$

which hold in the time intervals $(t_0,t_1),(t_1,t_2),\ldots,(t_{M-1},t_M)$. By doing that also the third of relations (4.6) is automatically satisfied in every time interval (t_k,t_{k+1}), $k=0,1,\ldots,M-1$. Moreover, the second of relations (4.6) is enforced only at times $t_0,t_1,\ldots,t_{M-1}$, by setting

$$U(x,t_k)=\max\left(\lim_{t\to t_k^+}U(t,x),\tilde{\varsigma}(x)\right),\quad k=0,1,\ldots,M-1. \tag{4.28}$$

Note that the function $U(\cdot,t_k)$ computed according to (4.28) is used as the final condition for the problem (4.27) that holds in the time interval $(t_{k-1},t_k),\ k=1,2,\ldots,M-1$. Instead, the final condition for the problem (4.27) that holds in the time interval (t_{M-1},t_M), according to the fourth of relations (4.6), is prescribed as follows:

$$U(x,t_M)=\varsigma(x). \tag{4.29}$$

That is, in summary, problems (4.27) are recursively solved for $k=M-1,M-2,\ldots,0$, starting from the condition (4.29), and at each time $t_{M-1},t_{M-2},\ldots,t_0$ the American constraint (4.28) is imposed.

It can be shown that the error due to such a time discretization decays like $O(\Delta t)$ as Δt tends to zero (see [51]). Nevertheless, in our work the first-order component of the error is suppressed by the Richardson extrapolation procedure employed in SubSection (4.2), and second-order accuracy is achieved (see [22]).

Therefore, Algorithm 2 proceeds as follows: First of all problem (4.27) is approximated in time using (4.11). Then, in the equations obtained, as well as in (4.28), (4.29), we substitute $U^k(x)$ with $U^k_{RBPI}(x)$ given by (4.12). Finally, we evaluate all the resulting equations at $x=x_i,\ i=0,1,\ldots,n$. This yields:

$$\begin{cases} \mathbf{B}\boldsymbol{\Xi}^k=\mathbf{A}\boldsymbol{\Lambda}^{k+1}+\mathbf{H}^k, \\ \boldsymbol{\Lambda}^k=\max\left(\boldsymbol{\Xi}^k,\boldsymbol{\Pi}\right), \end{cases}$$

to be recursively solved for $k=M-1,M-2,\ldots,0$, starting from

$$\boldsymbol{\Lambda}^M=\boldsymbol{\Pi}, \tag{4.30}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{H}^k$ and $\boldsymbol{\Pi}$ are defined as in (4.18)–(4.20), (4.22), respectively.

Let us observe that the first of relations (4.30) comes from the discretization of (4.27), whereas the second of (4.30) is obtained from (4.28). Note also that in (4.20) $\mathbf{H}^k$ is a function of $U^k(0)$ and $U^k(1)$ which, according to the second of (4.27), are computed as follows: $U^k(0)=E,\ U^k(1)=0,\ k=0,1,\ldots,M-1$.

4.6.3. Algorithm 3

Algorithm 3 is directly based on solving problem (4.7), which is done as follows: First of all we evaluate (4.7)at $t=t_k,\ k=0,1,\ldots,M$, and substitute the Black–Scholes operator (4.3) with (4.11). Then, we substitute $U^k(x)$ with $U^k_{RBPI}(x)$ (given by (4.12)), and evaluate the equations obtained at $x=x_i,\ i=0,1,\ldots,n$. This yields the systems of equations:

$$\mathbf{B}\boldsymbol{\Lambda}^k=\mathbf{A}\boldsymbol{\Lambda}^{k+1}+C\varepsilon\times\left(\mathbf{O}./(\boldsymbol{\Lambda}^k+(\varepsilon-E)\mathbf{O}+\mathbf{X})\right)+\mathbf{H}^k, \tag{4.31}$$

to be recursively solved for $k=M-1,M-2,\ldots,0$, starting from

$$\boldsymbol{\Lambda}^M=\boldsymbol{\Pi}, \tag{4.32}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{H}^k$, $\boldsymbol{\Pi}$ are defined as in (4.18)–(4.20), (4.22), respectively, and

$$\begin{aligned} &\mathbf{O}_{(n+1)\times 1}=[1 \quad 1 \quad \ldots \quad 1]^T, \\ &\mathbf{X}_{(n+1)\times 1}=[0 \quad -L\log(1-x_1) \quad -L\log(1-x_2) \quad \ldots \quad -L\log(1-x_{n-1}) \quad 0]^T. \end{aligned} \tag{4.33}$$

In Eq. (4.31) the symbol $./$ means componentwise division of two vectors. The above system of equations is nonlinear and thus is solved using an inner iteration cycle. Precisely, for each (fixed) value of k, $k=0,1,\ldots,M-1$, we set $\boldsymbol{\Lambda}^{k,0}=\boldsymbol{\Lambda}^{k+1}$ and compute $\boldsymbol{\Lambda}^{k,1}$, $\boldsymbol{\Lambda}^{k,2}$, $\boldsymbol{\Lambda}^{k,3},\ldots$, by recursively solving the following system of linear equations:

$$\mathbf{B}\boldsymbol{\Lambda}^{k,l}=\mathbf{A}\boldsymbol{\Lambda}^{k+1}+C\varepsilon\times\left(\mathbf{O}./(\boldsymbol{\Lambda}^{k,l-1}+(\varepsilon-E)\mathbf{O}+\mathbf{X})\right)+\mathbf{H}^k. \tag{4.34}$$

This inner cycle is stopped when

$$||\Lambda^{k,l} - \Lambda^{k,l-1}||_\infty \leqslant \varepsilon_k, \tag{4.35}$$

where $\varepsilon_k = 10^{-6}$. If condition (4.35) is satisfied, then we set $\Lambda^k = \Lambda^{k,l}$ and go ahead to the next time level.

Remark 2. Both the numerical method developed in Subsection 4.5 and Algorithm 2 require solving at every time step a system of linear equations (systems (4.15) and (4.30), respectively). Also, when using Algorithm 3 one has to solve a system of linear equations at every inner iteration (4.34). Now, the matrices associated to such systems are full and often ill-conditioned (see [35,69]), but are not very large, as the RBPI can be applied with a relatively small number of centers (say $n \leqslant 200$). Therefore, the aforementioned linear systems are solved using the LU factorization method with partial pivoting, which is particularly suitable for handling full and ill-conditioned matrices of relatively small dimension (see [46,70]). Moreover, as the matrices to be inverted are the same for every time step (and in the case of Algorithm 3 also for every inner iteration (4.34)), the LU factorization can be performed only once at the beginning of the numerical simulation, and thus at each time step (or at each inner iteration (4.34)) the corresponding linear system is efficiently solved by forward and backward recursion [46,70].

5. Numerical results

The numerical experiments are performed on a PC Laptop Intel(R) Core(TM)2 Duo CPU T9550 2.66 GHz 4 GB RAM and the software programs are developed and run under Matlab R2010b, 64-bit. Following the notation employed in Section 3, let U and U_{RBPI} respectively denote the option price (either European or American) and its approximation obtained using the RBPI method developed in the previous section. The error on U_{RBPI} at the current time ($t = 0$) is measured using both the discrete maximum norm:

$$\text{MaxError} = \max_{i=0,1,\ldots,n} |U_{RBPI}(x_i, 0) - U(x_i, 0)| \tag{5.1}$$

and the mean square norm:

$$\text{RMSError} = \frac{1}{n+1}\sqrt{\sum_{i=0}^{n}(U_{RBPI}(x_i, 0) - U(x_i, 0))^2}. \tag{5.2}$$

Note that in the case of the American option the exact value of U is not available. Therefore, in (5.1), (5.2) we use instead a very accurate approximation of it, which is obtained using the Algorithm 2 (described in SubSection 4.6.2) with a very large number of centers and time steps (precisely we set $n = 300$ and $M = 1000$).

5.1. Test case 1

First of all, the proposed RBPI method is applied to a European put option. In particular, we consider the same test case reported in [29,66], where the option and model parameters are chosen as follows: $E = 10$, $T = 0.5$ (years), $r = 0.05$, $\sigma = 0.2$. The number of time discretization steps is set equal to fifty ($M = 50$). As we have experimentally checked, this choice is such that in all the simulations performed the error due to the time discretization is negligible with respect to the error due to the RBPI discretization (note that in the present work we are mainly concerned with the RBPI spatial approximation).

The error on the option price and the computer times are shown in Table 1. Note that these results are obtained by applying the mesh refinement algorithm described in SubSection 4.4 with three levels of refinement.

Looking at Table 1 we can see that the RBPI is very accurate and fast. In fact, for example, the option price can be computed with an error of order 10^{-5} (in both the mean square norm and the maximum norm) in only 0.11 s. Furthermore, very satisfactory levels of accuracy (errors of order 10^{-3} or 10^{-4}) are achieved using a relatively small number of RBPI centers ($n = 50$ or $n = 75$).

In Fig. 2 we present the spatial distribution of the RBPI centers. In particular, in this plot we consider 101 centers ($n = 100$) and we show the first 100 of them, i.e. $s_0, s_1, \ldots, s_{99}$ (s_{100} being at infinity). As we can see, the change of variables

Table 1
Test case 1, efficiency of the RBPI.

n	RMSError	maxError	CPU time (s)
25	1.33×10^{-2}	2.37×10^{-2}	0.01412
50	5.08×10^{-3}	6.56×10^{-3}	0.01783
75	9.53×10^{-4}	1.33×10^{-3}	0.02290
100	1.17×10^{-4}	4.82×10^{-4}	0.03024
150	7.83×10^{-5}	1.15×10^{-4}	0.04363
200	3.28×10^{-5}	9.51×10^{-5}	0.11112

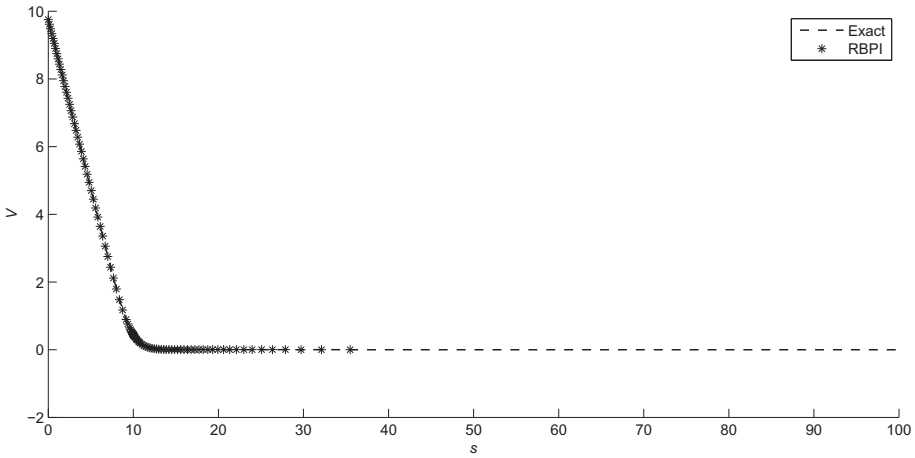

Fig. 2. Test case 1, RBPI center distribution, $n = 100$.

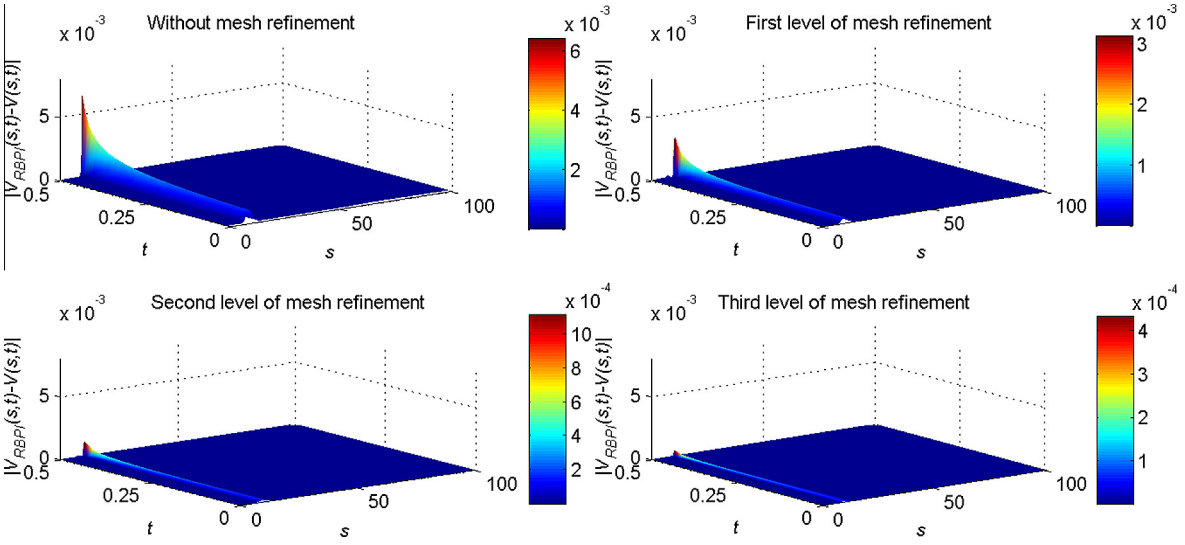

Fig. 3. Test case 1, mesh refinement ($n = 100$).

(4.1) and the mesh refinement technique described in SubSection 4.4 allow us to concentrate almost all of the centers in the price interval $[0, 2E]$. However, there is still a relatively small fraction of centers in the interval $[2E, +\infty)$.

Finally, in Fig. 3 we show the effect of the local mesh refinement described in SubSection 4.4, which turns out to very suitable for handling the non-smooth option's payoff. In fact, as we can observe, only three levels of mesh refinement allow us to reduce the error of a factor equal to some tens.

5.2. Test case 2

Let us consider the case of an American put option. In particular, the option and model parameters are chosen as in [19,29,71]: $E = 100,\ T = 1,\ r = 0.1,\ \sigma = 0.3$.

As done in Test case 1 we set $M = 100$ (which again is such that the error due to the time discretization is negligible with respect to the error due to the RBPI discretization). Moreover, the mesh refinement algorithm described in SubSection 4.4 is applied with three levels of refinement.

Finally, it remains to choose the parameter C and ε required by Algorithm 3 (see (4.34)). Following a common procedure (see, e.g., [52,23]), we set $C = rE$. Furthermore, we choose $\varepsilon = 10^{-5}$, which, by numerical experiments, roughly minimizes the error of the computed solutions.

Table 2
Test case 2, efficiency of the RBPI.

n	Algorithm 1			Algorithm 2			Algorithm 3		
	RMSError	MaxError	CPU time (s)	RMSError	MaxError	CPU time (s)	RMSError	MaxError	CPU time (s)
50	5.46×10^{-4}	7.12×10^{-4}	0.1775	2.08×10^{-3}	5.33×10^{-3}	0.0365	4.89×10^{-3}	7.41×10^{-3}	0.1122
60	2.89×10^{-4}	5.37×10^{-4}	0.2190	8.97×10^{-4}	2.71×10^{-3}	0.0375	9.45×10^{-4}	3.24×10^{-3}	0.1126
70	9.02×10^{-5}	3.74×10^{-4}	0.2269	7.76×10^{-4}	9.10×10^{-4}	0.0421	7.92×10^{-4}	9.12×10^{-4}	0.1234
80	8.15×10^{-5}	1.60×10^{-4}	0.2533	4.63×10^{-4}	7.01×10^{-4}	0.0500	5.44×10^{-4}	8.11×10^{-4}	0.1541
90	6.11×10^{-5}	8.35×10^{-5}	0.3698	9.51×10^{-5}	3.34×10^{-4}	0.0600	1.70×10^{-4}	5.27×10^{-4}	0.1837
100	3.47×10^{-5}	6.02×10^{-5}	0.3937	7.66×10^{-5}	1.70×10^{-4}	0.0677	9.51×10^{-5}	3.91×10^{-4}	0.2001

Table 3
Value of optimal exercise boundary in American put option by using $\Delta t = 0.01$.

Methods		$B(0)$
Algorithm 1	$n = 50$	76.2022
	$n = 60$	76.2254
	$n = 70$	76.2267
	$n = 80$	76.2362
	$n = 90$	76.2434
	$n = 100$	6.2483
Algorithm 2	$n = 50$	76.1903
	$n = 60$	76.1962
	$n = 70$	76.2070
	$n = 80$	76.2232
	$n = 90$	76.2304
	$n = 100$	76.2423
Algorithm 3	$n = 50$	76.1903
	$n = 60$	76.1820
	$n = 70$	76.1883
	$n = 80$	76.1962
	$n = 90$	76.2024
	$n = 100$	76.2233
"True" value		76.2491

The errors and the computer times obtained are shown in Table 2. As we can see, also in this case the RBPI is very accurate and fast. In fact, if, for example, Algorithm 1 is employed, then the price of the American option can be computed with an error of order 10^{-5} (in both the mean square norm and the maximum norm) in only 0.39 s. We observe that, given the number n of RBPI centers, Algorithm 1 is the most accurate of the three methods employed. However, Algorithm 2 is only $3 \div 5$ times less accurate than Algorithm 1, but is also $5 \div 6$ times faster than it. Therefore, if we think to measure the computational efficiency by the ratio between the error and the computer time, then on the overall Algorithm 2 turns out to be slightly more efficient than Algorithm 1. Finally, Algorithm 3 reveals to be approximately as accurate as Algorithm 2, but is about three times slower than it.

Finally, we want to compute the value of the exercise boundary (at the current time $t = 0$). To this aim, first of all we evaluate $\tilde{B}(0)$ by solving with the bisection method the equation

$$U(\tilde{B}(0), 0) = E + L\log(1 - \tilde{B}(0)) \tag{5.3}$$

and then we evaluate

$$B(0) = -L\log(1 - \tilde{B}(0)). \tag{5.4}$$

The results obtained are shown in Table 3 (again the "true" value of the free boundary is computed using Algorithm 2 with $n = 300$ and $M = 1000$). As we can see, the proposed RBPI method allows us to obtain a very efficient approximation of the free boundary. In fact, by using 101 centers, both Algorithm 1 and Algorithm 2 provide $B(0)$ with four correct decimal digits in 0.39 s and 0.067 s, respectively (see also Table 2 where computer times are reported). Moreover, these two algorithms yield a satisfactory approximation of the free boundary even if a relatively small number of centers are employed (Algorithm 1 and Algorithm 2 allow us to obtain three correct decimal digits with 51 centers and 71 centers respectively). Finally, as far as Algorithm 3 is concerned, it reveals to be slightly less accurate than Algorithm 1 and Algorithm 2.

6. Conclusions and future work

We have proposed a new meshfree RBPI method to price European and American options under the Black–Scholes model. The RBPI approach offers several advantages over the more conventional radial basis function approximation, nevertheless it

has never been used for option pricing, at least to the very best of our knowledge. In this paper the RBPI is combined with several numerical techniques: an exponential change of variables, which allows us to approximate the option prices on their whole spatial domain, a mesh refinement algorithm, which turns out to be very effective for dealing with the non-smooth options' payoffs, and an implicit Euler–Richardson extrapolated scheme, which provides a satisfactory level of time accuracy. Moreover, in order to solve the free boundary problem that arises in the case of American options, three different approaches are employed: the PSOR method, the Bermudan approximation, and the penalty approach. Numerical experiments are presented which demonstrate the computational efficiency of the RBPI and the effectiveness of the various techniques employed. In particular, the prices of both the European and the American options can be computed with an error of order 10^{-4} or 10^{-5} in only few hundredths of a second. Moreover, the PSOR reveals to be the most accurate of the three algorithms used to deal with the early exercise opportunity, nevertheless the Bermudan discretization approach turns out to be slightly more efficient than it if computer times are taken into account.

Finally, the proposed RBPI method is straightforward to implement and model independent, and thus could be applied to a large variety of financial problems also different from the ones considered in this manuscript. For example, in order to describe the option's underlying asset, instead of the geometric Brownian motion (2.1) one could use a model with price dependent volatility, such as the popular CEV model [72], or a model with jumps [73]. Moreover, one could extend the numerical method developed in the present paper to the valuation of Asian options, i.e. options whose payoff depends on the average of the underlying asset price on a given period of time. In fact, Asian option prices can be obtained by solving one or a set of partial differential equations (depending wether the underlying asset price is continuously or discretely monitored, see, e.g., [74–76]). These equations are very similar to the Black–Scholes Eq. (2.3), and thus the RBPI method proposed in the present paper can be applied.

References

[1] F. Black, M. Scholes, The pricing of options and corporate liabilities, J. Polit. Econ. 81 (1973) 637–659.
[2] M. Brennan, E. Schwartz, Finite difference methods and jump processes arising in the pricing of contingent claim: a synthesis, J. Finance Quant. Anal. 13 (1978) 461–474.
[3] C. Vázquez, An upwind numerical approach for an American and European option pricing model, Appl. Math. Comput. 97 (1998) 273–286.
[4] X. Wu, W. Kong, A highly accurate linearized method for free boundary problems, Comput. Math. Appl. 50 (2005) 1241–1250.
[5] A. Arciniega, E. Allen, Extrapolation of difference methods in option valuation, Appl. Math. Comput. 153 (2004) 165–186.
[6] M. Yousuf, A.Q.M. Khaliq, B. Kleefeld, The numerical approximation of nonlinear Black–Scholes model for exotic path-dependent American options with transaction cost, I. J. Comput. Math. 89 (2012) 1239–1254.
[7] R. Zvan, P.A. Forsyth, K.R. Vetzal, A general finite element approach for PDE option pricing models (Ph.D. thesis), University of Waterloo, Waterloo, 1998.
[8] L.V. Ballestra, C. Sgarra, The evaluation of American options in a stochastic volatility model with jumps: an efficient finite element approach, Comput. Math. Appl. 60 (2010) 1571–1590.
[9] L.V. Ballestra, L. Cecere, A numerical method to compute the volatility of the fractional Brownian motion implied by American options, Int. J. Appl. Math. 26 (2013) 203–220.
[10] P.A. Forsyth, K.R. Vetzal, Quadratic convergence for valuing American options using a penalty method, SIAM J. Sci. Comput. 23 (2002) 2095–2122.
[11] R. Zvan, P.A. Forsyth, K.R. Vetzal, A finite volume approach for contingent claims valuation, IMA J. Numer. Anal. 21 (2001) 703–731.
[12] S.J. Berridge, J.M. Schumacher, An irregular grid approach for pricing high-dimensional American options, J. Comput. Appl. Math. 222 (2008) 94–111.
[13] A.Q.M. Khaliq, D.A. Voss, S.H.K. Kazmi, Adaptive -methods for pricing American options, J. Comput. Appl. Math. 222 (2008) 210–227.
[14] A.Q.M. Khaliq, D.A. Voss, S.H.K. Kazmi, A fast high-order finite difference algorithm for pricing American options, J. Comput. Appl. Math. 222 (2008) 17–29.
[15] B.F. Nielsen, O. Skavhaug, A. Tveito, Penalty methods for the numerical solution of American multi-asset option problems, J. Comput. Appl. Math. 222 (2008) 3–16.
[16] J. Zhao, M. Davison, R.M. Corless, Compact finite difference method for American option pricing, J. Comput. Appl. Math. 206 (2007) 306–321.
[17] D. Tangman, A. Gopaul, M. Bhuruth, Numerical pricing of options using high-order compact finite difference schemes, J. Comput. Appl. Math. 218 (2008) 270–280.
[18] B. Hu, J. Liang, L. Jiang, Optimal convergence rate of the explicit finite difference scheme for American option valuation, J. Comput. Appl. Math. 230 (2009) 583–599.
[19] J.C. Cox, S.A. Ross, M. Rubinstein, Option pricing: a simplified approach, J. Finance Econ. 7 (1979) 229–263.
[20] M. Broadie, J. Detemple, American option valuation: new bounds, approximations, and a comparison of existing methods, Rev. Finance Stud. 9 (1996) 1211–1250.
[21] M. Gaudenzi, F. Pressacco, An efficient binomial method for pricing American put options, Decision Econ. Finance 4 (2003) 1–17.
[22] S.L. Chung, C.C. Chang, R.C. Stapleton, Richardson extrapolation techniques for the pricing of American-style options, J. Futures Markets 27 (2007) 791–817.
[23] A. Khaliq, G. Fasshauer, D. Voss, Using meshfree approximation for multi-asset American option problems, J. Chin. Inst. Eng. 27 (2004) 563–571.
[24] L.V. Ballestra, G. Pacelli, Pricing European and American options with two stochastic factors: a highly efficient radial basis function approach, J. Econ. Dyn. Cont. 37 (2013) 1142–1167.
[25] Y.C. Hon, X. Mao, A radial basis function method for solving options pricing models, Finance Eng. 8 (1999) 31–49.
[26] A. Golbabai, D. Ahmadian, M. Milev, Radial basis functions with application to finance: American put option under jump diffusion, Math. Comput. Modell. 55 (2012) 1354–1362.
[27] Z. Wu, Y.C. Hon, Convergence error estimate in solving free boundary diffusion problem by radial basis functions method, Eng. Anal. Bound. Elem. 27 (2003) 73–79.
[28] M.D. Marcozzi, S. Choi, C.S. Chen, On the use of boundary conditions for variational formulations arising in financial mathematics, Appl. Math. Comput. 124 (2003) 197–214.
[29] Y.C. Hon, A quasi-radial basis functions method for American options pricing, Comput. Math. Appl. 43 (2002) 513–524.
[30] B. Fornberg, N. Flyer, J.M. Russell, Comparisons between pseudospectral and radial basis function derivative approximations, IMA J. Numer. Anal. 30 (2010) 149–172.
[31] B. Fornberg, T. Dirscol, G. Wright, R. Charles, Observations on the behavior of radial basis function approximations near boundaries, Comput. Math. Appl. 43 (2002) 473–490.
[32] G. Liu, Y. Gu, An Introduction to Meshfree Methods and Their Programming, Springer, Netherlands, 2005.

[33] G. Liu, Y. Gu, A point interpolation method, in: Proceeding of 4th Asia–Pacific Conference on Computational Mechanics, Singapore (1999) 1009–1014.
[34] G. Liu, Y. Gu, A point interpolation method for two-dimensional solid, Int. J. Numer. Meth. Eng. 50 (2001) 937–951.
[35] J. Wang, G. Liu, A point interpolation meshless method based on radial basis functions, Int. J. Numer. Meth. Eng. 54 (2002) 1623–1648.
[36] J.G. Wang, G.R. Liu, On the optimal shape parameters of radial basis functions used for 2d meshless methods, Comput. Methods Appl. Mech. Eng. 191 (2002) 2611–2630.
[37] X. Li, J. Zhu, S. Zhang, A hybrid radial boundary node method based on radial basis point interpolation, Eng. Anal. Bound. Elem. 33 (2009) 1273–1283.
[38] G.R. Liu, K. Dai, K.M. Lim, Y. Gu, A radial point interpolation method for simulation of two-dimensional piezoelectric structures, Smart Mater. Struct. 12 (2003) 171–180.
[39] G. Liu, Y. Gu, A local radial point interpolation method (LRPIM) for free vibration analysis of 2-d solids, J. Sound Vib. 246 (2001) 29–46.
[40] Y. Gu, G. Liu, A boundary radial point interpolation method (BRPIM) for 2-d structural analyses, Struct. Eng. Mech. 15 (2002) 535–550.
[41] D. Krige, A review of the development of geostatistics in South Africa, Advanced Geostatistics in the Mining Industry, Holland, 1976, pp. 279–293.
[42] K.Y. Dai, G.R. Liu, K.M. Lim, Y.T. Gu, Comparison between the radial point interpolation and the Kriging interpolation used in meshfree methods, Comput. Mech. 32 (2003) 60–70.
[43] L. Gu, Moving Kriging interpolation and element-free Galerkin method, Int. J. Numer. Methods Eng. 56 (2003) 1–11.
[44] L. Chen, K.M. Liew, A local Petrov–Galerkin approach with moving Kriging interpolation for solving transient heat conduction problems, Comput. Mech. 47 (2011) 455–467.
[45] T.Q. Bui, M.N. Nguyen, A moving Kriging interpolation-based meshfree method for free vibration analysis of Kirchhoff plates, Comput. Struct. 89 (2011) 380–394.
[46] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer-Verlag, New York, 2000.
[47] D.M. Pooley, K.R. Vetzal, P.A. Forsyth, Convergence remedies for non-smooth payoffs in option pricing, J. Comput. Finance 6 (2003) 25–40.
[48] C.W. Cryer, The solution of a quadratic programming problem using systematic overrelaxation, SIAM J. Control 9 (1971) 385–392.
[49] R. Seydel, Tools for Computational Finance, Springer-Verlag, Berlin, 2002.
[50] S. Ikonen, J. Toivanen, Efficient numerical methods for pricing American options under stochastic volatility, Numer. Methods Partial Diff. Equ. 24 (2008) 813–825.
[51] S. Howison, A matched asymptotic expansions approach to continuity corrections for discretely sampled options, part 2: Bermudan options, Appl. Math. Finance 14 (2007) 91–104.
[52] B.F. Nielsen, O. Skavhaug, A. Tveito, Penalty and front-fixing methods for the numerical solution of American option problems, J. Comput. Finance 5 (2002) 69–97.
[53] A.Q.M. Khaliq, D.A. Voss, S.H.K. Kazmi, A linearly implicit predictor-corrector scheme for pricing American options using a penalty method approach, J. Bank Finance 30 (2006) 489–502.
[54] M.D. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge University Press, New York, 2004.
[55] S. Rippa, An algorithm for selecting a good parameter c in radial basis function interpolation, Adv. Comput. Math. 11 (1999) 193–210.
[56] L.V. Ballestra, G. Pacelli, A radial basis function approach to compute the first-passage probability density function in two-dimensional jump-diffusion models for financial and other applications, Eng. Anal. Bound. Elem. 36 (2012) 1546–1554.
[57] L.V. Ballestra, G. Pacelli, Computing the survival probability density function in jump-diffusion models: a new approach based on radial basis functions, Eng. Anal. Bound. Elem. 35 (2011) 1075–1084.
[58] R.E. Carlson, T.A. Foley, The parameter r^2 in multiquadric interpolation, Comput. Math. Appl. 21 (1991) 29–42.
[59] A.H.D. Cheng, M.A. Golberg, E.J. Kansa, Q. Zammito, Exponential convergence and H-c multiquadric collocation method for partial differential equations, Numer. Meth. Part. D.E. 19 (2003) 571–594.
[60] G. Fasshauer, J. Zhang, On choosing optimal shape parameters for RBF approximation, Numer. Algorithms 45 (2007) 346–368.
[61] R. Franke, Scattered data interpolation: test of some methods, Math. Comput. 38 (1982) 181–200.
[62] C. Franke, R. Schaback, Solving partial differential equations by collocation using radial basis functions, Appl. Math. Comput. 93 (1998) 73–82.
[63] C.S. Huang, C.F. Lee, A.H.D. Cheng, On the increasingly flat radial basis function and optimal shape parameter for the solution of elliptic PDEs, Eng. Anal. Bound. Elem. 34 (2010) 802–809.
[64] C.S. Huang, H.D. Yen, A.H.D. Cheng, Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method, Eng. Anal. Bound. Elem. 31 (2007) 614–623.
[65] H. Wendland, Scattered Data Approximation, Cambridge University Press, New York, 2005.
[66] P. Wilmott, S. Howison, J. Dewynne, The Mathematics of Financial Derivatives, Cambridge University Press, 1995.
[67] J. Zhang, H. Sun, J. Zhao, High order compact scheme with multigrid local mesh refinement procedure for convection diffusion problems, Comput. Methods Appl. Mech. Eng. 191 (2002) 4661–4674.
[68] S. Yang, S.T. Lee, H. Sun, Boundary value methods with the Crank–Nicolson preconditioner for pricing options in the jump-diffusion model, Int. J. Comput. Math. 88 (2011) 1730–1748.
[69] G. Liu, Meshfree Methods: Moving Beyond the Finite Element Method, Taylor and Francis/CRC Press, Boca Raton, 2009.
[70] B.P. Flannery, W.H. Press, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in Fortran 90: the Art of Parallel Scientific Computing, Cambridge University Press, New York, 1996.
[71] L. Wu, Y. Kwok, A front-fixing finite difference method for the valuation of American options, Finance Eng. 6 (1997) 83–97.
[72] Y.L. Hsu, T.I. Lin, C.F. Lee, Constant elasticity of variance (CEV) option pricing model: integration and detailed derivation, Math. Comput. Simul. 79 (2008) 60–71.
[73] A. Almendral, C.W. Oosterlee, Numerical valuation of options with jumps in the underlying, Appl. Numer. Math. 53 (2005) 1–18.
[74] L. Rogers, Z. Shi, The value of an Asian option, J. Appl. Probab. 32 (1995) 1077–1088.
[75] J. Vecer, A new PDE approach for pricing arithmetic average Asian options, J. Comput. Finance 4 (2001) 105–113.
[76] R. Zvan, P.A. Forsyth, K.R. Vetzal, Robust numerical methods for PDE models of Asian options, J. Comput. Finance 1 (1998) 39–78.