Forward deterministic pricing of options using Gaussian radial basis functions

Jamal Amani Rad[a], Josef Höök[b], Elisabeth Larsson[b,*], Lina von Sydow[b]

[a] *Dept. of Computer Sciences, Faculty of Mathematical Sciences and Dept. of Cognitive Modeling, Institute for Cognitive and Brain Sciences, Shahid Beheshti University, G. C. Tehran, Iran*

[b] *Dept. of Information Technology, Uppsala University, Box 337, SE-751 05 Uppsala, Sweden*

Abstract

The price of a fixed-term option is the expected value of the payoff at the time of maturity. When not analytically available, the option price is computed using stochastic or deterministic numerical methods. The most common approach when using deterministic methods is to solve a backward partial differential equation (PDE) such as the Black-Scholes equation for the option value. The problem can alternatively be formulated based on a forward PDE for the probability of the asset value at the time of maturity. This enables simultaneous pricing of several contracts with different payoffs written on the same underlying asset. The main drawback is that the initial condition is a (non-smooth) Dirac function. We show that by using an analytical expansion of the solution for the first part of the time interval, and applying a high-order accurate radial basis function (RBF) approximation in space, we can derive a competitive forward pricing method. We evaluate the proposed method on European call options and barrier options, and show that even for just one payoff it is more efficient than solving the corresponding backward PDE.

Keywords: option pricing, Fokker-Planck equation, radial basis function, Dirac delta function, Kolmogorov forward equation
2000 MSC: 65M70, 91G60

*Corresponding author
Email addresses: j_amanirad@sbu.ac.ir (Jamal Amani Rad), josef.hook@it.uu.se (Josef Höök), elisabeth.larsson@it.uu.se (Elisabeth Larsson), lina.von.sydow@it.uu.se (Lina von Sydow)

1. Introduction

Trading of options is taking place daily in the financial market all over the world. Current prices must be available at each time, indicating that pricing methods need to be highly efficient. Pricing is important not only for trading of options, but also for hedging and calibration purposes.

Different models are considered for the underlying asset(s) as well as different solution strategies, leading to a rich set of combinations to explore. In this paper our focus is on the solution strategies. The objective of the paper is to show that a combination that has mostly been overlooked so far, deterministic methods applied to the forward partial differential equation (PDE) for the dynamics, is computationally efficient. In [1] a set of benchmark problems for a number of different models with state-of-the-art numerical methods to solve them are evaluated. We use these results for comparisons.

As a model problem where we have analytical solutions to compare with, we consider a Black-Scholes market with a deterministic bond and a stochastic asset with price-dynamics B_t and S_t respectively, given by

$$\begin{aligned} dB_t &= rB_t dt, \\ dS_t &= \mu S_t dt + \sigma S_t dW_t, \end{aligned} \tag{1}$$

where r is the risk-free interest rate, and μ and σ are the drift and the volatility of the asset. For an option issued on S_t from (1) that at the time of maturity T pays $\phi(S_T, K)$, where ϕ is called the pay-off function and K is the strike price of the option, the value u_0 of the option today is given by

$$u_0 = e^{-rT}\mathbb{E}^Q(\phi(S_T, K)), \tag{2}$$

where $\mathbb{E}^Q$ denotes the expected value under the risk-neutral measure Q.

From this basic formulation, different mathematical models for the option pricing problem can be derived, that in turn are appropriate for different types of numerical solution methods.

The risk-neutral expectation (2) together with the dynamics (1) leads to the stochastic differential equation

$$\begin{aligned} S_t &= rS_t dt + \sigma S_t dW_t, \\ S_0 &= s_0 \end{aligned} \tag{3}$$

for the asset price, where s_0 is the asset value today. This formulation is mainly used in Monte Carlo methods [2]. A European option value is

approximated as the discounted mean value of a large number of simulated trajectories $S_t(j)$, $j = 1, \ldots, M$, as

$$\tilde{u}_0 = e^{-rT} \frac{1}{M} \sum_{j=1}^{M} (\phi(S_T(j), K)). \tag{4}$$

Monte Carlo methods are flexible with respect to the underlying model and the option contract. The basic Monte Carlo method has a convergence rate of $\mathcal{O}(M^{-1/2})$. By applying and combining variance reduction techniques, quasi random sampling, and multi level Monte Carlo approaches [3, 4], the convergence rate can be improved. For high-dimensional problems, Monte Carlo simulation is the standard method. However, for problems in low dimensions, when the option is issued on one or only a few assets, deterministic methods are more computationally efficient [1].

In order to use a deterministic numerical method, we need a different mathematical model. The most well known deterministic model for option pricing is the Black-Scholes backward PDE for the option value $u(s, t)$

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 u}{\partial s^2} + rs\frac{\partial u}{\partial s} - ru = 0, \ s \in \mathbb{R}_+, \ t > 0, \\ u(s, T) = \phi(s, K). \end{aligned} \tag{5}$$

independently derived by Black and Scholes [5] and Merton [6]. The solution value $u(s_0, 0)$ is equivalent to the option value u_0 defined in (2). Solving Equation (5), provides the value of the particular option with pay-off function $\phi(s, K)$. However, we get the value of u for all values of $s > 0$, which in some sense is unnecessary. We know the value s_0 of s today, and the values of u for $s \neq s_0$ are not useful to us, except for the potential computation of hedging parameters, using values of u in the vicinity of s_0. For the basic case with constant volatility and simple pay-off functions it is possible to rescale (5) to obtain multiple option values, but this is no longer true for more general problem-settings. Common methods in relation to (5) are, e.g., finite difference methods [7, 8, 9, 10, 11] and RBF approximation [12, 13, 14, 15, 16].

There is also a forward deterministic counterpart to (3), which is the adjoint of the Black-Scholes equation (5). The probability $p(s, t)$ that the asset takes the value s at time t when the asset follows the dynamics in (1),

is described by the Fokker-Planck (or forward Kolmogorov) equation

$$\begin{aligned} \frac{\partial p}{\partial t} - \frac{1}{2}\frac{\partial\left(\sigma^2 s^2 p\right)}{\partial s^2} + \frac{\partial\left(rsp\right)}{\partial s} = 0,\ s \in \mathbb{R}_+,\ t > 0, \\ p(s,0) = \delta(s_0 - s), \end{aligned} \tag{6}$$

where $\delta(x)$ is the Dirac delta function. The forward PDE (6) is commonly used for maximum likelihood parameter estimation from discretely observed market data [17]. The solution $p(s,t)$ describes the transition probability density between two observed states. Stochastic and deterministic simulation approaches to approximate transition densities are compared in [18], and solution methods based on RBF approximation are derived in [19, 20]. In the recent paper [21], the forward PDE is solved using a finite volume method to calibrate stochastic local volatility models.

To use (6) for option pricing, we integrate the pay-off function $\phi(s,K)$ against the probability density $p(s,T)$ and discount the result as

$$u_0(K,T) = e^{-rT}\int_{s\in R_+} p(s,T)\phi(s,K)ds. \tag{7}$$

This means that we solve the Fokker-Planck equation (6) once and can then use (7) to price options with different pay-off functions $\phi(s,K)$. This is useful from a practical point of view as it is common to price many contracts simultaneously for the same underlying diffusion model.

The question is then why deterministic forward pricing methods are not widely used. In [22] a forward pricing method was implemented for a two-dimensional basket option using mesh adaptive FEM for the spatial discretization. The resulting method was not competitive in terms of computational time due to the effect of the Dirac initial condition. In order to capture the initial development of the probability density accurately enough to reach the desired error tolerance in the final solution, adaptive remeshing had to be performed several times.

In order to preserve the property of only solving one PDE for several contracts, while avoiding the troublesome Dirac initial condition, there is a third way to derive a deterministic mathematical model. In [23] it is shown that for a European call option under the Black-Scholes model, integration of (6) twice with respect to s and using that $\frac{\partial^2(s-K)_+}{\partial s^2} = \delta(s-K)$ gives the

Dupire equation

$$\begin{aligned} \frac{\partial u}{\partial T} - \frac{1}{2}\sigma^2 K^2 \frac{\partial^2 u}{\partial K^2} + rK\frac{\partial u}{\partial K} = 0, K \in R_+,\ T > 0, \\ u(K,0) = (s_0 - K)_+. \end{aligned} \tag{8}$$

In this case, given the asset value today, s_0, we get the option values for all values of $K > 0$ and $T > 0$. This means that by solving one PDE in forward time we obtain many option prices. A drawback with this approach is that a new PDE needs to be derived for each type of payoff. In, e.g., the articles [22, 24, 25, 26, 27] Dupire-like equations are derived for some other types of options. In practice, the Dupire equation is used more often for the purpose of calibrating volatility surfaces, than for pricing options.

Methods that do not fall into the previous categories are binomial tree methods, where a discrete version of both the forward and backward problem is solved, and Fourier methods that use the characteristic function of the underlying dynamics and can avoid the solution of any differential equation. Fourier methods were shown to be highly efficient in [1].

In this paper, we show that a deterministic forward pricing method where RBF approximation is applied to (6) can be a competitive approach if special care is taken to deal with the initial condition. As a pilot case, we have implemented the method for options on one underlying asset following Black-Scholes dynamics. The success for this case shows that it can be worthwhile to extend this to higher-dimensional pricing problems as well as to other types of dynamics. A simple extension that can be applied without additional computational effort is to consider variable interest rate and volatility such that (3) becomes

$$\begin{aligned} dS_t &= \mu(S_t, t)dt + \sigma(S_t, t)dW_t, \\ S_0 &= s_0. \end{aligned} \tag{9}$$

In the related work [28], the forward pricing approach is employed for pricing options where the underlying asset follows a CGMY-process. The idea to use this approach in [28] originated with the present paper.

The outline of this paper is as follows. In Section 2, the mathematical model is derived in more detail. In the following Sections, 3 and 4, we describe the numerical treatment of the initial condition and the time discretization, while Section 5 is devoted to the description of the RBF-method for the spatial approximation. Section 6 shows how to compute the expected payoff using analytical integration, while Section 7 presents the results from our numerical experiments. Finally, conclusions are drawn in Section 8.

2. Mathematical model

In this paper we consider options issued on assets that follow the dynamics in (1). The different types of options that we use as examples are European call options of vanilla type with payoff

$$\phi(s, K) = (s - K)_+, \tag{10}$$

and barrier up–and–out options with payoff

$$\phi(s, K) = \begin{cases} (s - K)_+, & 0 \leq s < B, \\ 0, & s \geq B. \end{cases} \tag{11}$$

where B is the barrier above which the option becomes worthless. However, much of the methodology described in the paper can be applied generally for options of European type. If we for instance consider binary options that have payoff functions

$$\phi(s, K) = \begin{cases} A, & s > K, \\ A/2, & s = K, \\ 0, & s < K, \end{cases} \tag{12}$$

the probability density function $p(s, t)$ is the same as for the European call option and hence the solution strategy is identical for this option type. We refer to the option defined by (10) as a vanilla option and the option defined by (11) as a barrier option.

For the probability density function that solves (6), it holds that

$$p(0, t) = 0, \quad t \in [0, T]. \tag{13}$$

For the vanilla option, it further holds that $\lim_{s \longrightarrow \infty} p(s, t) = 0$. We truncate the domain and solve for p in $s \in [0, D]$, and prescribe boundary conditions at $\partial\Omega$ defined by $s = 0$ and $s = D$. For the vanilla option the probability density function has the analytical solution

$$p(s, t) = \frac{1}{\sqrt{2\pi t}\sigma s} \exp\left(-\frac{(\log(s/s_0) - (r - \sigma^2/2)t)^2}{2\sigma^2 t} \right). \tag{14}$$

To choose a suitable D, we solve the equation $p(s = D, T) = \epsilon$, which gives

$$D = s_0 \exp\left[(r - \frac{3}{2}\sigma^2)T + \sigma\sqrt{2\sigma^2 T^2 - 2T(rT + \log(\epsilon s_0 \sqrt{2\pi T}))} \right], \tag{15}$$

where ϵ is a chosen tolerance.

The PDE in (6) may be written as a continuity equation,

$$\frac{\partial p}{\partial t} + \frac{\partial J}{\partial s} = 0 \tag{16}$$

where

$$J = rsp - \frac{1}{2}\frac{\partial}{\partial s}\left(\sigma^2 s^2 p\right). \tag{17}$$

To derive a natural boundary condition at $s = D$, we interpret (16) and (17) as Kirchoff's law, where J is a probability current [29]. To preserve the total probability density in Ω it is sufficient to require that the probability current vanishes at the boundary $\partial\Omega$, which is trivially true at $s = 0$. Thus we prescribe

$$J = 0, \ s \in \partial\Omega. \tag{18}$$

For the barrier option, the option value is non-zero if the underlying asset did not hit the barrier during the time interval $[0, T]$. Otherwise, the option is worthless. Since the probability of hitting the barrier is positive, the probability density mass that contributes to the option value decreases over time. This is reflected by a homogeneous Dirichlet boundary condition at $s = B$ in this case

$$p(B, t) = 0, \quad t \in [0, T]. \tag{19}$$

Altogether, this gives us the following initial-boundary value problem to solve

$$\frac{\partial p}{\partial t} + \mathcal{L}p = 0, \qquad s \in \Omega, \qquad t \in (0, T], \tag{20}$$

$$\mathcal{L}^b p = 0, \qquad s \in \partial\Omega, \qquad t \in (0, T], \tag{21}$$

$$p(s, 0) = \delta(s - s_0), \qquad s \in \Omega, \tag{22}$$

where

$$\mathcal{L}p(s, t) = \frac{\partial p}{\partial t} - \frac{1}{2}\frac{\partial\left(\sigma^2 s^2 p\right)}{\partial s^2} + \frac{\partial\left(rsp\right)}{\partial s}, \tag{23}$$

$$\mathcal{L}^b p(0, t) = p(0, t), \tag{24}$$

$$\mathcal{L}^b p(D, t) = -rsp(D, t) + \frac{1}{2}\frac{\partial}{\partial s}(\sigma^2 s^2 p(D, t)) \qquad \text{(vanilla option)}, \tag{25}$$

$$\mathcal{L}^b p(B, t) = p(B, t) \qquad \text{(barrier option)}. \tag{26}$$

In the following sections we describe how to solve Problem (20)–(22) numerically.

3. Approximation of the probability density function for small times using Hermite polynomials

A major challenge when solving (20)–(22) is the approximation of the initial condition (22) due to the singularity of the Dirac function. To avoid this problem, we approximate the probability density function at a small positive time t_0 using the strategy introduced in [30] by Aït-Sahalia. The idea in [30] is to approximate the density function $p_S(s, t_0)$ associated with

$$\begin{aligned} dS_t &= \mu_S(S_t)dt + \sigma_S(S_t)dW_t, \\ S_0 &= s_0, \end{aligned} \tag{27}$$

using orthogonal Hermite polynomials. In order to ensure convergence of the approximation, the original process (27) is transformed into a process

$$Y \equiv \gamma(S) = \int_0^S \frac{1}{\sigma_S(u)} du,$$

with dynamics

$$\begin{aligned} dY_t &= \mu_Y(Y_t)dt + dW_t, \\ \mu_Y(y) &= \frac{\mu_S(\gamma^{-1}(y))}{\sigma_S(\gamma^{-1}(y))} - \frac{1}{2}\frac{\partial \sigma_S}{\partial s}(\gamma^{-1}(y)). \end{aligned}$$

Next, a second transformation is introduced

$$Z \equiv \frac{Y - y_0}{\sqrt{t_0}},$$

where for a fixed t_0, Z is close to a $\mathcal{N}(0, 1)$-variable. Using this to approximate $p_Z(z, t_0)$ it is possible to find

$$p_Y(y, t_0) = \frac{1}{\sqrt{t_0}} p_Z(\frac{y - y_0}{\sqrt{t_0}}, t_0)$$

and finally

$$p_S(s, t_0) = \frac{1}{\sigma_S(s)} p_Y(\gamma(s), t_0).$$

This gives us (for details, see [30])

$$p_S(s, t_0) = \frac{1}{\sigma_S(s)\sqrt{t_0}} f\left(\frac{y - y_0}{\sqrt{t_0}}\right) \exp\left(\int_{y_0}^{y} \mu_Y(\omega)d\omega\right) \sum_{\ell=0}^{L} c_\ell(y) \frac{t_0^\ell}{\ell!}, \tag{28}$$

where

$$c_0(y) = 1,$$
$$c_\ell(y) = \ell\,(y-y_0)^{-\ell} \int_{y_0}^{y} (\omega - y_0)^{\ell-1} \left(\lambda_Y(\omega) c_{\ell-1}(\omega) + \tfrac{1}{2} \tfrac{\partial^2 c_{\ell-1}(\omega)}{\partial \omega^2} \right) d\omega, \qquad \ell \geq 1,$$

$y = \gamma(s)$, $y_0 = \gamma(s_0)$, and finally

$$\eta_Y(y) = -\tfrac{1}{2}\mu_Y^2(y) - \tfrac{1}{2}\tfrac{\partial \mu_Y(y)}{\partial y},$$

$$f(z) = \exp(-z^2/2)/\sqrt{2\pi},$$

$$\lambda_Y(y) = -\tfrac{1}{2}\mu_Y^2(y) - \tfrac{1}{2}\tfrac{\partial}{\partial y}\mu_Y(y).$$

Using this approach for S in (1), we have

$$\begin{aligned}
y = \gamma(s) &= \int_0^s \frac{1}{\sigma x} dx = \frac{\log(s)}{\sigma}, \\
\mu_Y(y) &= \frac{r}{\sigma} - \frac{1}{2}\sigma, \\
\lambda_Y(y) &= -\frac{1}{2}(\frac{r}{\sigma} - \frac{1}{2}\sigma)^2, \\
c_1(y) &= -\frac{1}{2}(\frac{r}{\sigma} - \frac{1}{2}\sigma)^2, \\
c_2(y) &= \frac{1}{4}(\frac{r}{\sigma} - \frac{1}{2}\sigma)^4, \\
c_l(y) &= \frac{(-1)^\ell}{2^\ell}(\frac{r}{\sigma} - \frac{1}{2}\sigma)^{2\ell}.
\end{aligned}$$

Hence, we can get an approximate solution to (6) at time t_0 by taking the first three terms in (28) to get

$$\begin{aligned}
\tilde{p}(s, t_0) = \frac{1}{\sigma s \sqrt{2\pi t_0}} &\left(1 - \frac{1}{2}\left(\frac{r}{\sigma} - \frac{\sigma}{2}\right)^2 t_0 + \frac{1}{8}\left(\frac{r}{\sigma} - \frac{\sigma}{2}\right)^4 t_0^2 \right) \times \\
\times \exp &\left(-\frac{1}{2t_0}\left(\frac{\log(s/s_0)}{\sigma}\right)^2 + \left(\frac{r}{\sigma} - \frac{\sigma}{2}\right)\left(\frac{\log(s/s_0)}{\sigma}\right) \right).
\end{aligned} \tag{29}$$

The properties of this approximation are discussed in detail in [30]. Here we can note that the error in the approximation $e(t_0) = \max_s(\tilde{p}(s, t_0) - p(s, t_0))$ can be estimated through the first neglected term in the expansion (28). By

forming the term involving c_3 and maximizing over the argument of the exponential function we get

$$e(t_0) \approx \frac{t_0^{2.5}}{48\sigma s_0\sqrt{2\pi}} \left(\frac{r}{\sigma} - \frac{\sigma}{2}\right)^6 \exp\left(-\sigma t_0(\frac{r}{\sigma} - \frac{\sigma}{2})\right), \tag{30}$$

for the Black-Scholes model. The error estimate has been numerically validated and it accurately reflects the true error values. The error as a function of the volatility σ vanishes for $\sigma = \sqrt{2r}$ and increases with the distance from that point both for larger and smaller values of the volatility. For the barrier option, the density approximation above is valid until the time when the probability mass at the barrier location B becomes non-negligible. In both cases, the approximation error grows with the size of the time step t_0.

We use the approximation (29) as a new initial condition for the system (20)–(22) such that we instead solve

$$\frac{\partial p}{\partial t} + \mathcal{L}p = 0, \qquad s \in \Omega, \qquad t \in (t_0, T], \tag{31}$$

$$\mathcal{L}^b p = 0, \qquad s \in \partial\Omega, \qquad t \in (t_0, T], \tag{32}$$

$$p(s, t_0) = \tilde{p}(s, t_0), \qquad s \in \Omega. \tag{33}$$

There is a trade-off between the error (30) introduced by the approximation (29) that grows with t_0 and the numerical error in interpolating $\tilde{p}(s, t_0)$ that decreases with t_0 due to the increasing smoothness of the initial condition. In Section 6 we establish numerically how large the first time-step should be.

4. Discretization in time using BDF-2

We divide the time-interval $[t_0, T]$ into M time-steps of length $k_m = t_m - t_{m-1}$, $m = 1, \ldots, M$, and let the approximate solution at the discrete times t_m be denoted by

$$p^m(s) \approx p(s, t_m).$$

The BDF-2 implicit time-stepping method [31] uses a backward Euler step for the first time interval, and a two-step approximation for the remaining time intervals. The method is given by

$$\begin{aligned} p^m(s) - \beta_0^m \mathcal{L}p^m(s) &= f^m, && s \in \Omega, && m = 1, \ldots, M \\ \mathcal{L}^b p^m(s) &= 0, && s \in \partial\Omega, && m = 1, \ldots, M, \\ p^0(s) &= \tilde{p}(s, t_0), && s \in \Omega \cup \partial\Omega, \end{aligned} \tag{34}$$

where

$$\begin{aligned} f^m &= p^{m-1} && s \in \Omega, \quad m = 1, \\ f^m &= \beta_1^m p^{m-1}(s) - \beta_2^m p^{m-2}(s), && s \in \Omega, \quad m = 2, \ldots, M, \end{aligned}$$

The coefficients are defined by $\beta_0^1 = k_1$, and

$$\beta_0^m = k_m \frac{1+\omega_m}{1+2\omega_m}, \ \beta_1^m = \frac{(1+\omega_m)^2}{1+2\omega_m}, \ \beta_2^m = \frac{\omega_m^2}{1+2\omega_m}, \quad m = 2, \ldots, M,$$

where $\omega_m = k_m/k_{m-1}$. By choosing ω_m such that $\beta_0^m = \beta_0^1 = k_1$, we obtain the same operator in the left hand sides of (34) for all time-steps, which is computationally efficient. In [32] a recursion formula is derived for this purpose.

5. Radial basis function approximations in space

We solve (34) using approximation with global RBFs in space, see [33] for more information on RBF approximation methods. Such methods are meshfree, and approximations are defined on scattered node sets. A standard RBF approximation $\tilde{p}^m(s)$ to $p^m(s)$ in (34) is given by

$$\tilde{p}^m(s) = \sum_{j=1}^{N} \lambda_j^m \varphi(\|s - s_j^c\|), \tag{35}$$

where $\|\cdot\|$ is the Euclidean norm, $\lambda_j^m \in \mathbb{R}$ for $j = 1, \ldots, N$, $m = 1, \ldots, M$, and $S_c = \{s_j^c\}_{j=1}^N$ is the set of center nodes for the real-valued RBFs $\varphi(r)$. In this paper we use Gaussian RBFs $\varphi(r) = e^{-\varepsilon^2 r^2}$, where the shape parameter ε governs the flatness of the RBFs.

A common approach to determine the coefficients λ_j^m in RBF approximation methods is collocation at the set of node points S_c. However, in this paper we use a least squares formulation for the approximation which separates center nodes and evaluation points [34]. We define two sets of evaluation points, a set of interior points $S_e = \{s_j^e\}_{j=1}^{N_e}$ where we enforce the Fokker-Planck equation using least squares, and a set of boundary points $S_b = \{s_i^b\}_{i=1}^{N_b}$, $N_b + N_e > N$, where we enforce the boundary conditions exactly.

We collect the unknown coefficients λ_j^m at each time level into the two vectors $\bar{\lambda}^m = (\lambda_1^m, \ldots, \lambda_{N-N_b}^m)^T$ and $\bar{\kappa}^m = (\lambda_{N-N_b+1}^m, \ldots, \lambda_N^m)^T$ and define the corresponding subsets of center nodes $S_c^\lambda = \{s_j^c\}_{j=1}^{N-N_b}$, and $S_c^\kappa =$

$\{s_j^c\}_{j=N-N_b+1}^{N}$. By also defining the solution vectors $\bar{p}_e^m = \tilde{p}^m(S_e)$, $\bar{p}_b^m = \tilde{p}^m(S_b)$, where $\tilde{p}^m(S_e) = (\tilde{p}^m(s_1^e), \ldots, \tilde{p}^m(s_{N_e}^e))^T$ and similarly for $\bar{p}_b^m$ we get the following relations

$$\begin{aligned} \bar{p}_e^m &= \varphi(S_e, S_c^\lambda)\bar{\lambda}^m + \varphi(S_e, S_c^\kappa)\bar{\kappa}^m, \\ \bar{p}_b^m &= \varphi(S_b, S_c^\lambda)\bar{\lambda}^m + \varphi(S_b, S_c^\kappa)\bar{\kappa}^m \end{aligned} \tag{36}$$

where $\varphi(S_e, S_c^\lambda)$ is the $N_e \times (N - N_b)$ matrix with elements $\varphi(\|s_k^e - s_j^c\|)$, $k = 1, \ldots, N_e$, $j = 1, \ldots, N - N_b$, and similarly for the other combinations. Combining (34) and (36) results in the following over-determined system of equations for the RBF coefficients at each time step:

$$\begin{pmatrix} \varphi(S_e, S_c^\lambda) - \beta_0 \mathcal{L}\varphi(S_e, S_c^\lambda) & \varphi(S_e, S_c^\kappa) - \beta_0 \mathcal{L}\varphi(S_e, S_c^\kappa) \\ \mathcal{L}^b\varphi(S_b, S_c^\lambda) & \mathcal{L}^b\varphi(S_b, S_c^\kappa) \end{pmatrix} \begin{pmatrix} \bar{\lambda}^m \\ \bar{\kappa}^m \end{pmatrix} = \begin{pmatrix} \bar{f}^m \\ 0 \end{pmatrix}, \tag{37}$$

$m = 1, \ldots, M$, where $\mathcal{L}\varphi(S_e, S_c^\lambda)$ is the $N_e \times (N - N_b)$ matrix with elements $\mathcal{L}\varphi(\|s_k^e - s_j^c\|)$, $k = 1, \ldots, N_e$, $j = 1, \ldots, N - N_b$, and similarly for the other matrices with operators. To simplify the notation in the description of the solution algorithm, we denote the four matrix blocks by A_{ij}, $i, j = 0, 1$.

In order to enforce the boundary conditions exactly, we eliminate $\bar{\kappa}$ from the first block row in (37) to obtain

$$\begin{pmatrix} C & 0 \\ A_{10} & A_{11} \end{pmatrix} \begin{pmatrix} \bar{\lambda}^m \\ \bar{\kappa}^m \end{pmatrix} = \begin{pmatrix} \bar{f}^m \\ 0 \end{pmatrix}, \quad m = 1, \ldots, M, \tag{38}$$

where

$$C = A_{00} - A_{01}A_{11}^{-1}A_{10}.$$

For the collocation of the boundary conditions we use $S_b = S_c^\kappa$, i.e., the points where we evaluate the boundary conditions coincide with the center points. This gives that A_{11} (which is a square matrix) is symmetric and non-singular for standard choices of RBFs including Gaussians [35].

We summarize the above in the following algorithm to solve (37) with exact enforcement of the boundary conditions and a linear least squares solution in the interior:

1. Solve $C\bar{\lambda}^m = f^m$ in the least squares sense.
2. Solve $A_{11}\bar{\kappa}^m = -A_{10}\bar{\lambda}^m$.

Due to the specific choice of time-step to use in (34), the coefficient matrices C and A_{11} are the same for all time-steps and can be factorized once prior to the time-stepping. The factorization step consists of the following computations:

1. Factorize $A_{11} = LU$.

2. Form C using the factorization of A_{11}.

3. Factorize $C = QR$.

6. Computation of option prices

By the numerical procedure defined in Sections 3, 4 and 5, we can compute an approximation to the probability density at the time of maturity $\tilde{p}^M(s) \approx p(s,T)$. Using this density in Equation (7), we can compute the option price for a general pay-off function $\phi(s,K)$ by numerical integration. However, for a large class of pay-off functions, this integral can be computed analytically when the density is approximated by Gaussian RBFs.

From (7) and the fact that

$$p(s,T) \approx \tilde{p}^M(s) = \sum_{j=1}^{N} \lambda_j^M e^{-\varepsilon^2(s-s_j)^2}, \tag{39}$$

we get

$$u_0(K,T) \approx e^{-rT} \sum_{j=1}^{N} \lambda_j^M \int_0^\infty e^{-\varepsilon^2(s-s_j)^2} \phi(s,K) ds. \tag{40}$$

Using the properties of the normal distribution (Gaussian) density, we have that

$$\int_a^b e^{-\varepsilon^2(s-s_j)^2} = \left[\frac{\sqrt{\pi}}{2\varepsilon} \mathrm{erf}(\varepsilon(s-s_j)) \right]_a^b.$$

From the definitions (10), and (11) of the pay-off functions for the options that we consider, we see that we need

$$\int_K^\infty e^{-\varepsilon^2(s-s_j)^2} ds = \left[\frac{\sqrt{\pi}}{2\varepsilon} \mathrm{erf}(\varepsilon(s-s_j)) \right]_K^\infty,$$

$$\int_K^\infty s\ e^{-\varepsilon^2(s-s_j)^2} ds = \left[-\frac{1}{2\varepsilon^2} e^{-\varepsilon^2(s-s_j)^2} + \frac{\sqrt{\pi} s_j}{2\varepsilon} \mathrm{erf}(\varepsilon(s-s_j)) \right]_K^\infty,$$

to obtain the vanilla option prices

$$u_0(K,T) = e^{-rT} \sum_{j=1}^{N} \lambda_j^M \left[\tfrac{\sqrt{\pi}}{2\varepsilon} (s_j - K) (1 - \mathrm{erf}(\varepsilon(K-s_j))) + \tfrac{1}{2\varepsilon^2} e^{-\varepsilon^2(K-s_j)^2} \right], \tag{41}$$

and the barrier option prices

$$u_0(K,T) = e^{-rT}\sum_{j=1}^{N}\lambda_j^M \left[-\tfrac{1}{2\varepsilon^2}(e^{-\varepsilon^2(B-s_j)^2} - e^{-\varepsilon^2(K-s_j)^2}) + \right.$$
$$\left. + \tfrac{\sqrt{\pi}}{2\varepsilon}(x_i - K)\left(\operatorname{erf}(\varepsilon(B-x_i)) - \operatorname{erf}(\varepsilon(K-x_i))\right)\right]. \tag{42}$$

Similarly, if we are interested in pricing a binary option with pay-off function (12) we get

$$u_0(K,T) = A\, e^{-rT}\sum_{j=1}^{N}\lambda_j^M \left[\frac{\sqrt{\pi}}{2\varepsilon}(1 - \operatorname{erf}(\varepsilon(K-s_j)))\right]. \tag{43}$$

Due to the nature of the solution for $p(s,t)$ using the RBF method with Gaussian basis functions (39), analytical expressions like (41), (42), and (43) can be derived for many option types.

7. Numerical results

The aim of this paper is to investigate whether forward deterministic pricing can be competitive compared with other pricing methods, and to understand how to apply the method to a given problem in the best possible way. For the evaluation of the method, we use one-dimensional problems with analytical solution. If the question regarding competitiveness is answered affirmatively, we can use the understanding gained about the method to develop forward pricing for multi-asset or multi-factor options as well as options following other types of dynamics in future work.

The method is implemented in MATLAB and the numerical experiments are performed using a laptop with an Intel® Core™2 Duo CPU T9550, 2.66 GHz, and 4 GB RAM. The two test cases used are the vanilla option and the up-and-out barrier option with $B = 1.5$.

We start by illustrating the properties of the two test problems. The probability densities at the time of maturity $p(s,T)$ for an initial asset value $s_0 = 1$ for the vanilla option and the barrier option are shown in Figure 1. The two densities are similar, but due to the barrier $B = 1.5$, the distribution for the barrier option goes directly to zero, while the unconstrained density continues smoothly.

Figure 2 shows the option values as a function of K for both options. The prices are also similar, but the barrier price is lower as expected.

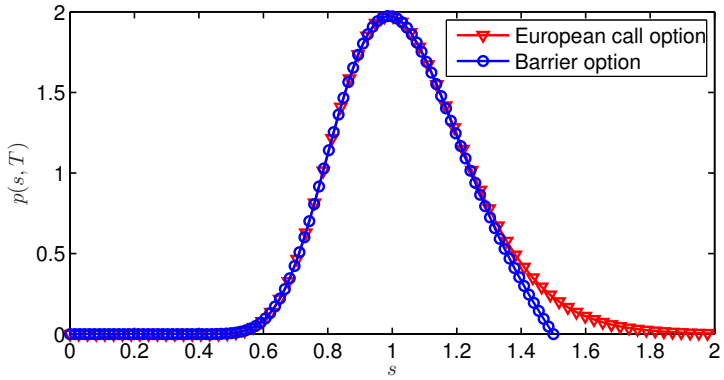

Figure 1: Probability density function $p(s,T)$ for a vanilla option and barrier option respectively. The parameters used are $\sigma = 0.2$, $s_0 = 1.0$, $r = 0.05$, $T = 1.0$, and $t_0 = 0.01$. For the barrier option the barrier $B = 1.5$.

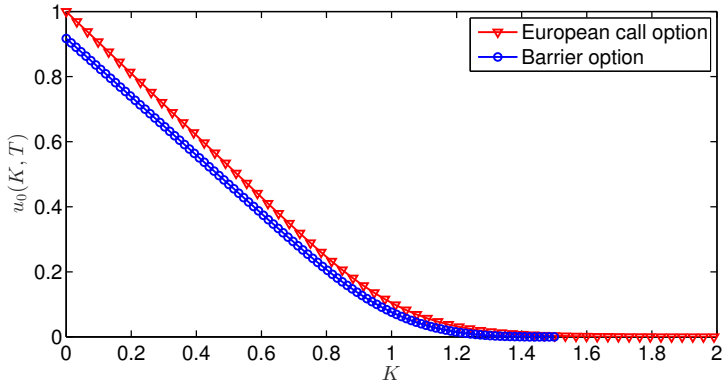

Figure 2: Value of vanilla option and barrier option as a function of strike price K. The parameters used are the same as in Figure 1, $\sigma = 0.2$, $s_0 = 1.0$, $r = 0.05$, $T = 1.0$, $t_0 = 0.01$, and $B = 1.5$.

The method described in the previous subsections has the parameters t_0 for the length of the Aït-Sahalia step, ε for the shape of the RBFs, h for the distance between RBFs, and M for the number of time steps. Each parameter has an influence on accuracy and computational cost. We will go through one parameter at a time and see how they can be chosen in relation to the problem parameters. The accuracy is measured in terms of the errors in the computed option values. We define the pointwise and maximum norm errors as

$$e_u(K) = |u(K,T) - u_{exact}(K,T)|,$$

$$E_u = \max_K e_u(K),$$

where $K \in [0, D]$ for the vanilla option and $K \in [0, B]$ for the barrier option. The functions u_{exact} and u denote the analytical option price using Black-Scholes formula (see Appendix A), and its approximation obtained using our proposed method, respectively.

As target for the error tolerance we have set $E_u = 10^{-4}$, which is reasonable for practical applications. For the vanilla option, the size of the computational domain D is determined from (15) using the tolerance $\epsilon = 10^{-8}$. The number of least-squares evaluation points should be related to the number of RBFs $N = D/h + 1$. We have used $N_e = 3N$ evaluation points in all simulations. In experiments for parameters that influence the spatial approximation, we let the number of time steps $M = 500$, such that the error due to the time discretization is negligible compared to the spatial approximation errors.

In the first experiment, we investigate how to choose t_0. If we choose a small t_0, the initial condition is sharp, and we may need a small h (large N) to resolve it in the RBF approximation. The computational cost for initializing the RBF matrices is $\mathcal{O}(N^3)$ and the cost for each time step is $\mathcal{O}(N^2)$. Hence, in terms of computational cost we want to maximize t_0. On the other hand, the accuracy of the initial approximation decreases with increasing t_0. In Figure 3 $\log(E_u)$ is plotted against t_0. We can see that

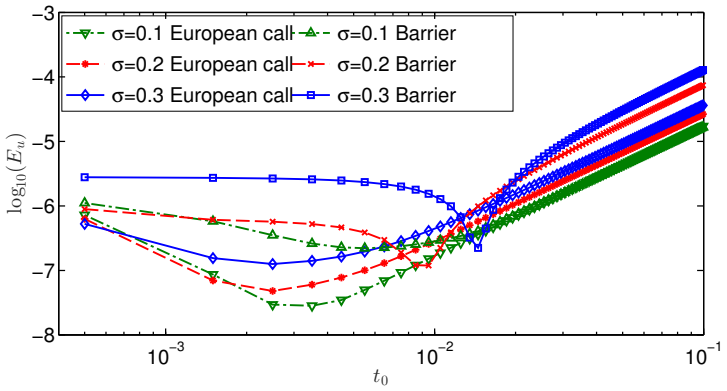

Figure 3: Error in the option price as a function of t_0 for different σ for both a vanilla option and a barrier option. The parameters used are $s_0 = 1.0$, $r = 0.05$, and $T = 1.0$.

when t_0 is as large as 0.1, we start getting an error in the option price that is larger than the target error. We conclude that a reasonable choice of t_0 is around 0.01 and use this in the subsequent experiments in this section.

In [36], our proposed method to advance the first time-step using (29) was compared with the method suggested in, e.g., [37, 38], where the Dirac initial condition was approximated using matching of moments. In the parameter range that we are studying in this paper, our proposed method was numerically shown to be superior. If we were to consider parameters that deviate largely from the ones studied here, we believe that by fine-tuning of t_0, our proposed method would still be the method of choice.

For a given h, the choice of shape parameter ε does not influence the computational cost. However, it can have a significant influence on the accuracy of the solution. When the shape parameter becomes too small, the RBF matrices become ill-conditioned and the numerical accuracy suffers, while for large shape parameters, the approximation accuracy is reduced leading to larger errors in the option values.

In Figures 4 and 5 we show contour plots of the errors in the option prices as a function of h and ε. In the same figures red lines that are fitted to the minimal error are displayed. It is clear that the lines follow the region of smallest error and that the error grows both for larger and smaller values of the shape parameter. The optimal shape parameter values are represented by $\varepsilon(h) = \frac{0.81}{h} + 0.15$ for the vanilla option and $\varepsilon(h) = \frac{0.35}{h} + 0.15$ for the barrier option.

Numerical experiments for the other problem parameters s_0, r, σ, T, and B show that these do not significantly affect the error profiles in Figures 4–5. The formulas for $\varepsilon(h)$ given above are used in the remainder of the paper.

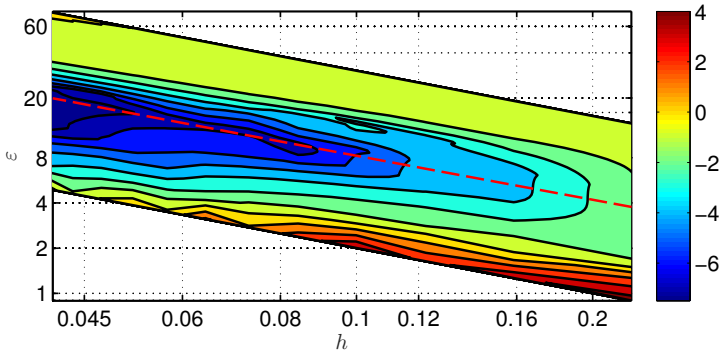

Figure 4: Error in the vanilla option price for different ε and h using $s_0 = 1.0$, $r = 0.05$, $\sigma = 0.2$, and $T = 1.0$. The color values represent $\log_{10}(E_u)$.

The next parameter to determine is h. As discussed already when choosing t_0, the computational cost grows rapidly with $N = D/h + 1$. We should therefore find the largest h that gives an error less or equal to the target

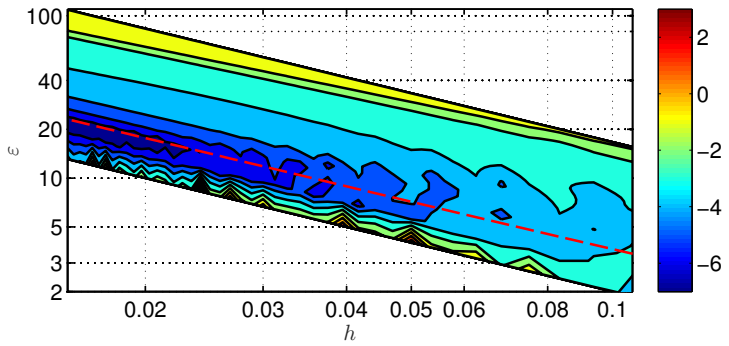

Figure 5: Error in the barrier option price for different ε and h using $s_0 = 1.0$, $r = 0.05$, $\sigma = 0.2$, $T = 1.0$, and $B = 1.5$. The color values represent $\log_{10}(E_u)$.

level 10^{-4}. The only problem parameter that showed a significant influence on the error in our experiments is σ, which is expected as the volatility influences the width of the probability density function. The optimal h for the given target error is plotted against σ in Figure 6. There is a linear dependence between σ and h. The fitted lines are $h = 0.55\sigma$ for the vanilla option and $h = 0.2\sigma$ for the barrier option. The size of h alone does not tell what the final computational cost is, it needs to be related to the size of the computational domain D, which also grows with σ. As is shown later in this section the net effect of σ, h and D is that the number of RBFs N is almost the same for the tested volatilities. This means that the computational cost is insensitive to the volatility.

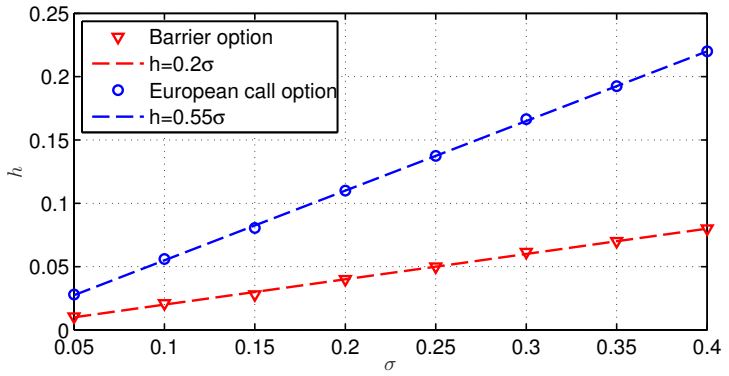

Figure 6: The spatial discretization parameter h needed to obtain an error in the option value that is less than 10^{-4} for different σ together with fitted linear functions $h(\sigma)$ for both a vanilla option and a barrier option. The parameters used are $s_0 = 1.0$, $r = 0.05$, $T = 1.0$, and $B = 1.5$ for the barrier option.

We end this section by comparing our proposed method with the corresponding least-squares RBF method [1] to solve the Black-Scholes equation (5) for a vanilla option. For the forward pricing method, the parameters are chosen according to the formulas derived in this section. For the Black-Scholes solver, we tune the parameters to achieve the target accuracy of 10^{-4}. We use the same relation for the least squares evaluation points $N_e = 3N$ as for the forward method. The computational domain is given by $s \in [0, 3K]$. The value of M is for both methods manually determined to be the smallest value that does not influence the final error in the solution.

The results of the comparison are reported In Table 1. We display the log of the error, $\ell_E = \log_{10}(E_u)$, to show that all the results are close to the target accuracy. For the forward pricing method, both the time to compute the option price for one pay-off function, T_1, and the time to compute the option price for 100 pay-off functions, T_{100}, when the underlying asset is following the same dynamics is presented.

Forward method with $s_0 = 0.9$								B-S with $s_0 = 0.9$			
σ	ℓ_E	D	h	N	M	T_1	T_{100}	ℓ_E	N	M	T_1
0.1	-4.1	2.05	0.055	38	10	0.039	0.063	-4.2	60	10	0.145
0.2	-4.1	3.54	0.110	32	10	0.031	0.052	-3.7	60	10	0.145
0.3	-4.0	5.94	0.165	36	10	0.034	0.057	-3.7	60	50	0.173
Forward method with $s_0 = 1.0$								B-S with $s_0 = 1.0$			
σ	ℓ_E	D	h	N	M	T_1	T_{100}	ℓ_E	N	M	T_1
0.1	-4.0	1.69	0.055	32	10	0.031	0.052	-4.1	60	10	0.145
0.2	-4.0	2.92	0.110	27	10	0.028	0.044	-4.0	60	10	0.145
0.3	-3.9	4.91	0.165	30	10	0.031	0.051	-3.9	60	10	0.145
Forward method with $s_0 = 1.1$								B-S $s_0 = 1.1$			
σ	ℓ_E	D	h	N	M	T_1	T_{100}	ℓ_E	N	M	T_1
0.1	-4.0	1.87	0.055	35	10	0.032	0.055	-4.1	60	10	0.145
0.2	-4.0	3.23	0.110	31	10	0.031	0.052	-3.6	60	10	0.145
0.3	-3.9	5.43	0.165	33	10	0.032	0.053	-3.6	60	50	0.173

Table 1: The logarithm of the error in the computed option value, ℓ_E together with the computational time to obtain the solution for one contract T_1 and 100 contracts T_{100} for $s_0 = 0.9$, 1.0, and 1.1 and $\sigma = 0.1$, 0.2, and 0.3. The other parameters used are $r = 0.05$, $K = 1$, and $T = 1.0$.

The comparison shows that the forward pricing method is 3.5–5.5 times faster than solving the Black-Scholes equation with a corresponding method. For the Black-Scholes results, $T_{100} = 100T_1$, and if we instead compute the time retios for pricing 100 contracts, forward pricing is 230–330 times faster than solving the Black-Scholes PDE for each contract.

Going back to the comparative study [1], we find that the fastest adaptive

finite difference method was 4.5 times faster than the least squares RBF-based Black-Scholes solver RBF-LSML, which is the method used in the comparison with the forward RBF method introduced here. This means that the new method is as fast as the finite difference method for one-dimensional problems.

8. Conclusions

Pricing based on the forward equations clearly offers an advantage with respect to pricing of multiple contracts, as the cost for integrating the payoff against the probability density is negligible compared with the computations to approximate the density. However, the numerical experiments show that even for pricing a single option, our proposed method is faster than the corresponding backward pricing method.

A key to this success is the analytical advancement of the initial Dirac function in time such that the PDE solution can start from a smooth initial condition, together with the application of a numerical method that is highly accurate for smooth solutions.

By using approximation with infinitely smooth Gaussian RBFs, we can achieve not only the tolerance of 10^{-4} used in the paper, but also higher accuracies, efficiently. The particular choice of Gaussians furthermore allows us to express the integration of simple payoffs against the density in terms of known functions, and we do not need to employ numerical integration schemes.

We have made a structured investigation of how to choose the method parameters and found simple relationships that provide an understanding of how errors and parameters interact. These can be used to achieve a desired tolerance in the option prices.

Further work on how to perform the analytical approximation for multi-factor dynamics is needed to extend the method to higher dimensions, and given the promising results in one dimension, this is an interesting direction for future research.

[1] L. von Sydow, L. Josef Höök, E. Larsson, E. Lindström, S. Milovanović, J. Persson, V. Shcherbakov, Y. Shpolyanskiy, S. Sirén, J. Toivanen, et al., Benchop–the benchmarking project in option pricing, International Journal of Computer Mathematics 92 (12) (2015) 2361–2379. doi:10.1080/00207160.2015.1072172.

[2] P. Glasserman, Monte Carlo methods in financial engineering, Vol. 53 of Applications of Mathematics (New York), Springer-Verlag, New York, 2004, stochastic Modelling and Applied Probability.

[3] M. B. Giles, Multilevel Monte Carlo path simulation, Oper. Res. 56 (3) (2008) 607–617. doi:10.1287/opre.1070.0496.

[4] M. B. Giles, D. J. Higham, X. Mao, Analysing multi-level Monte Carlo for options with non-globally Lipschitz payoff, Finance Stoch. 13 (3) (2009) 403–413. doi:10.1007/s00780-009-0092-1.

[5] F. Black, M. Scholes, The pricing of options and corporate liabilities, J. Polit. Econ. 81 (1973) 637–654.
URL http://www.jstor.org/stable/1831029

[6] R. C. Merton, Theory of rational option pricing, Bell J. Econom. Man. Sci. 4 (1973) 141–183.
URL https://www.jstor.org/stable/3003143

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

[8] J. Persson, L. von Sydow, Pricing European multi-asset options using a space-time adaptive FD-method, Computing and Visualization in Science 10 (4) (2007) 173–183. doi:10.1007/s00791-007-0072-y.

[9] P. Lötstedt, J. Persson, L. von Sydow, J. Tysk, Space–time adaptive finite difference method for European multi-asset options, Computers & Mathematics with Applications 53 (8) (2007) 1159–1180. doi:10.1080/00207160802140023.

[10] R. U. Seydel, Tools for computational finance, 4th Edition, Universitext, Springer-Verlag, Berlin, 2009.

[11] K. in 't Hout, R. Valkov, Numerical solution of a two-asset option valuation PDE by ADI finite difference discretization, AIP Conf. Proc. 1648 (2015) 850054. doi:10.1063/1.4912311.

[12] U. Pettersson, E. Larsson, G. Marcusson, J. Persson, Improved radial basis function methods for multi-dimensional option pricing, Journal of Computational and Applied Mathematics 222 (1) (2008) 82–93. doi:10.1016/j.cam.2007.10.038.

[13] L. V. Ballestra, G. Pacelli, Pricing European and American options with two stochastic factors: a highly efficient radial basis function approach, J. Econom. Dynam. Control 37 (6) (2013) 1142–1167. doi:10.1016/j.jedc.2013.01.013.

[14] J. A. Rad, K. Parand, L. V. Ballestra, Pricing European and American options by radial basis point interpolation, Appl. Math. Comput. 251 (2015) 363–377. doi:10.1016/j.amc.2014.11.016.

[15] V. Shcherbakov, E. Larsson, Radial basis function partition of unity methods for pricing vanilla basket options, Comput. Math. Appl. 71 (1) (2016) 185–200. doi:10.1016/j.camwa.2015.11.007.

[16] V. Shcherbakov, Radial basis function partition of unity operator splitting method for pricing multi-asset American options, BIT 56 (4) (2016) 1401–1423. doi:10.1007/s10543-016-0616-y.

[17] G. B. Durham, A. R. Gallant, Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes, J. Bus. Econ. Stat. 20 (3) (2002) 297–338. doi:10.1198/073500102288618397.

[18] B. Jensen, R. Poulsen, Transition densities of diffusion processes: Numerical comparison of approximation techniques, J. Deriv. 9 (4) (2002) 18–32. doi:10.3905/jod.2002.319183.

[19] 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 (9) (2011) 1075–1084. doi:10.1016/j.enganabound.2011.02.008.

[20] 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 (11) (2012) 1546–1554. doi:10.1016/j.enganabound.2012.04.011.

[21] M. Wyns, J. Du Toit, A finite volume–alternating direction implicit approach for the calibration of stochastic local volatility models, International Journal of Computer Mathematics (2017) 1–29. doi:10.1080/00207160.2017.1297805.

[22] A. Conze, N. Lantos, O. Pironneau, The forward Kolmogorov equation for two dimensional options, Chinese Annals of Mathematics.

[23] B. Dupire, Pricing with a smile, Risk 7 (1994) 18–20.

[24] O. Pironneau, Dupire-like identities for complex options, Comptes Rendus Mathematique 344 (2) (2007) 127–133. doi:10.1016/j.crma.2006.11.032.

[25] L. Andersen, J. Andreasen, Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing, Review of Derivatives Research 4 (3) (2000) 231–262. doi:10.1023/A:1011354913068.

[26] P. Carr, A. Hirsa, Why be backward? forward equations for American options, Risk 16 (1) (2003) 103–107.

[27] A. Bentata, R. Cont, Forward equations for option prices in semi-martingale models., Finance & Stochastics 19 (3) (2015) 617 – 651. doi:10.1007/s00780-015-0265-z.

[28] L. J. Höök, G. Ludvigsson, L. von Sydow, The Kolmogorov forward fractional partial differential equation for the cgmy-process with applications in option pricing, submitted for publication.

[29] H. Risken, The Fokker-Planck equation, 2nd Edition, Vol. 18 of Springer Series in Synergetics, Springer-Verlag, Berlin, 1989, methods of solution and applications. doi:10.1007/978-3-642-61544-3.

[30] Y. Aït-Sahalia, Maximum likelihood estimation of discretely sampled diffusions: A closed-form approximation approach, Econometrica 70 (1) (2002) 223–262. doi:10.1111/1468-0262.00274.

[31] E. Hairer, S. P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I (2Nd Revised. Ed.): Nonstiff Problems, Springer-Verlag New York, Inc., New York, NY, USA, 1993.

[32] E. Larsson, K. Åhlander, A. Hall, Multi-dimensional option pricing using radial basis functions and the generalized Fourier transform, J. Comput. Appl. Math. 222 (1) (2008) 175–192. doi:10.1016/j.cam.2007.10.039.

[33] G. E. Fasshauer, Meshfree approximation methods with MATLAB, Vol. 6, World Scientific, 2007.

[34] E. Larsson, S. M. Gomes, A least squares multi-level radial basis function method with applications in finance, manuscript in preparation (2017).

[35] I. J. Schoenberg, Metric spaces and completely monotone functions, Ann. of Math. (2) 39 (4) (1938) 811–841. doi:10.2307/1968466.

[36] H. Öhrn, A. Lindell, Discretization of the Dirac delta function for application in option pricing, Bachelor thesis, Report TVE 16 067, Dept. of Information Technology, Uppsala University (2016).
URL http://uu.diva-portal.org/smash/record.jsf?pid=diva2:942736

[37] J. Waldén, On the approximation of singular source terms in differential equations, Numerical Methods for Partial Differential Equations 15 (4) (1999) 503–520. doi:10.1002/(SICI)1098-2426(199907)15:4<503::AID-NUM6>3.0.CO;2-Q.

[38] A.-K. Tornberg, B. Engquist, Numerical approximations of singular source terms in differential equations, Journal of Computational Physics 200 (2) (2004) 462–488. doi:10.1016/j.jcp.2004.04.011.

[39] T. Björk, Arbitrage Theory in Continuous Time, 3rd Edition, Oxford University Press, New York, 2009.

Appendix A. Analytical formulas for the option prices

For the European call option, the closed form expression $u_{\text{exact}} = u^C$ for the option price is given by

$$u^C(t, s_0, K, T, r, \sigma^2) = s_0 \mathcal{N}(d^+(s_0/K, T-t)) - K e^{-r(T-t)} \mathcal{N}(d^-(s_0/K, T-t)), \tag{A.1}$$

where

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

$$d^-(x, y) = \frac{1}{\sigma\sqrt{y}} \left(\log(x) + \left(r - \frac{\sigma^2}{2} \right) y \right), \tag{A.3}$$

and $\mathcal{N}(x)$ is the cumulative distribution function for the standard normal distribution. For the barrier call up–and–out option, the closed form expression $u_{\text{exact}} = u^B$ for the option price is given by

$$\begin{aligned} u^B(t, s_0, K, T, r, \sigma^2) = {} & u^C(t, s_0, K, T, r, \sigma^2) \qquad - u^C(t, s_0, B, T, r, \sigma^2) \\ & - \left(\tfrac{B}{s_0} \right)^{\frac{2r}{\sigma^2} - 1} \ \big(\ u^C(t, B^2/s_0, K, T, r, \sigma^2) - u^C(t, B^2/s_0, B, T, r, \sigma^2) \ \big). \end{aligned} \tag{A.4}$$

For further details on the analytical expression for the barrier option, see [39, Theorem 18.12 p. 271].