

## A Numerical Study of RBFs-DQ Method for Multi-Asset Option Pricing Problems

Leila Khodayari and Mojtaba Ranjbar

**ABSTRACT:** In this paper, we propose a numerical scheme to solve multi-dimensional Black-Scholes equation using the global radial basis functions-based differential quadrature (RBFs-DQ) method. Before applying the method, it is needed to remove mixed derivatives from the Black-Scholes equation by making an appropriate change of variables. Then, any spatial derivatives are approximated by a linear weighted sum of all the function values in the whole physical domain. In the RBFs-DQ method the weighting coefficients are computed by RBFs. The method is very easy to implement and the non-singularity is ensured. The proposed method combines the advantages of the conventional DQ method and the RBFs. It also remains mesh-free feature of RBFs.

**Key Words:** Radial basis functions, multi-dimensional Black-Scholes equation, differential quadrature, European option

### Contents

|                                                                             |           |
|-----------------------------------------------------------------------------|-----------|
| <b>1 Introduction</b>                                                       | <b>1</b>  |
| <b>2 RBFs-based DQ method</b>                                               | <b>3</b>  |
| 2.1 Radial basis functions . . . . .                                        | 3         |
| 2.2 DQ method for partial derivative approximation in multi-dimension       | 4         |
| 2.3 RBFs-DQ approximation . . . . .                                         | 5         |
| <b>3 Global RBFs-DQ method for multi-dimensional Black-Scholes equation</b> | <b>7</b>  |
| <b>4 Stability of the method</b>                                            | <b>8</b>  |
| <b>5 Numerical computation of two asset European option</b>                 | <b>9</b>  |
| <b>6 Conclusion</b>                                                         | <b>12</b> |

### 1. Introduction

In the past several decades, financial derivative products have become increasingly important in the world of finance. Options as a kind of important financial derivatives have a wide range of applications that one of the major concerns in financial markets is determining the value of options. An option contract is an agreement between a buying party (the holder) and a selling party (the underwriter). The holder of the option contract has no obligation to use his option

---

2000 *Mathematics Subject Classification:* 91G20, 65M06, 65M12

contract, whereas the underwriter is obliged to agree with the holder uses the option contract. Since an investor of options can set the strike price of underlying assets in advance, these options are powerful tools for hedging risk in financial market, therefore, the trading volume of options are increasing all over the world. Options occur in many forms. Examples are vanilla options, barrier options, digital options and multi-asset options. In this paper, in order to implement the method, we consider options where the payoff depends on multi underlying assets. Such option pricing can be modelled by higher dimensional generalization of the original black-Scholes equation which Black and Scholes proposed an explicit formula for evaluating European call options without dividends [1] and extended by [2]. By assuming that the asset price is risk-neutral, Black and Scholes showed that the European call options value satisfies a lognormal diffusion type partial differential equation (PDE) which is known now as the Black-Scholes equation. If we introduce the notation,  $C(s_1, s_2, \dots, s_d, t)$  to represent the option value at time  $t$  and stock prices  $s_1, s_2, \dots, s_d$ , then we get the following linear parabolic PDE

$$\frac{\partial C}{\partial t} + \frac{1}{2} \sum_{i,j=1}^d \rho_{ij} \sigma_i \sigma_j s_i s_j \frac{\partial^2 C}{\partial s_i \partial s_j} + \sum_{i=1}^d r s_i \frac{\partial C}{\partial s_i} - rC = 0, \quad (1)$$

known as the Black-Scholes equation for multi-asset option problems. Where  $s_i, \sigma_i, \rho_{ij}$ , and  $r$  are respectively,  $i$ -th asset price, volatility of the  $i$ -th asset price, correlation between the prices of  $i$ -th and  $j$ -th assets, and risk free interest rate. Since then, methods for option pricing have been discovered and improved by many scholars. Details can be found in reference [3] which is a good review of valuation models and applications to the option pricing. According to [4], there have been numerous attempts to find the analytic form solutions of the Black-Scholes equation for various derivative products in financial world. However, it is not easy to get an analytic form solution for most of financial derivatives because of the complexity of the financial product itself and the system of the financial market. Therefore, in finance, numerical methods such as Finite Difference Method (FDM), Finite Element Method (FEM) and Monte Carlo (MC) simulation techniques for pricing derivatives problems, cf. [5,6,7,8,9,10,11,12], have been used. As has been mentioned in [13], different numerical methods can be applied to price multi-variate derivatives. Higher dimensional generalizations of lattice binomial methods can be used, cf. [14], where European options based on three underlying assets are solved numerically. Stable higher order methods for the Black and Scholes equation have been introduced by [15] and [16], mesh-free methods based on RBFs may also reduce the computational efforts significantly, see [17]. In [4] Jo and Kim combined the operator splitting method with parallel computation technique to solve the multi-dimensional Black-Scholes equations. [13] used Penalty methods for the numerical solution of American multi-asset option problems. [18] proposed explicit Runge-Kutta methods for multi-asset American options in 2014.

In this work, the global RBFs-DQ method is utilized for the numerical solution of multi-asset option pricing problems. As printed out in [19], the classic form of the differential quadrature (DQ) method was introduced by [20,21] for the numer-

ical solution of PDEs. The basic idea of the DQ method is that any derivative at a mesh point can be approximated by a weighted linear sum of all the function values along a mesh line. The key procedure in the DQ method is the determination of weighting coefficients for any order derivative discretization. As shown by [22], when the solution of a PDE is approximated by a high order polynomial, the weighting coefficients can be computed by a simple algebraic formulation or by a recurrence relationship. Later, [23] also showed that when the solution of the PDE is approximated by a Fourier series expansion, the weighting coefficients of the first and second-order derivatives can be computed explicitly by algebraic formulations. The details of the polynomial-based and Fourier series expansion-based DQ methods can be found in the book of [24]. The DQ method has been extensively applied in engineering for the rapid and accurate solution of various linear and non-linear differential equations [22,23,24,25,26,27,28,29,30]. In general, the polynomial-based and Fourier series expansion-based DQ methods can achieve very accurate results by using a considerable small number of grid points. It is a robust and efficient technique, but the dependency of the method on a mesh leads to complications. To overcome this difficult, using advantages of the RBF, [19] proposed other set of meshless methods which is named the RBFs-DQ method, which combines the mesh-free nature of RBFs with the derivative approximation of differential quadrature (DQ) method. It seems that the multi-dimensional polynomial approximation as the test function may not be a good choice in the DQ approximation. As has been shown in [19], RBFs, which have truly meshless property and insensitivity to high dimension, could be a good choice in the DQ approximation. The advantages of RBF-DQ method as combination of DQ approximation and RBFs provide an efficient discretization method, which is a derivative approximation approach and is mesh-free. In this method, the RBFs are taken as the test functions in the DQ approximation to compute the weighting coefficients. Moreover, the method not only inherits the advantages of the DQ method such as high accuracy and efficient computation, but also owns the merits of RBFs such as mesh-free feature and easy extension to high dimension.

The organization of this paper is as follows. In Section 2 we describe RBFs-DQ method for partial derivatives approximation. In Section 3 we apply the method on multi-dimensional option pricing problems by applying Theorem 3.1. Stability of the method to solve multi-asset option pricing problems is studied in Section 4. In the following Section 5, the numerical example for two-asset option pricing with this method is given. Finally, Section 6 is dedicated to a brief conclusion. Note that we have computed the numerical results by Matlab programming.

## 2. RBFs-based DQ method

In this section, we will show in detail the global RBFs-DQ method. In the following, we will show the details step by step.

**2.1. RADIAL BASIS FUNCTIONS.** In the past decades, the interpolation theory of RBF has undergone intensive research, and nowadays RBFs play an increasingly important role in the field of reconstructing functions from multivariate scattered

data. In general, the interpolation theory of RBF can be described as follows: if an unknown function  $f(X)$  is only known at a finite set of centres  $X_i, i = 1, \dots, N$ , the approximation of a function  $f(X)$  can be written as a linear combination of  $N$ , RBFs

$$f(X) \cong \sum_{j=1}^N \lambda_j \varphi(X, X_j) + \mathbb{P}(X), \quad X \in \Omega \subset \mathbb{R}^d, \quad (2)$$

where  $N$  is the number of data points,  $X = (x_1, x_2, \dots, x_d)$ ,  $d$  is the dimension of the problem,  $\lambda$ 's are coefficients to be determined and  $\varphi$  is the RBF. Also, Eq. (2) can be written without the additional polynomial  $\mathbb{P}$ . The success of RBF interpolant is due to its excellent performance. Based on numerical experiments, [31] gave a comprehensive review on the interpolation methods for scattered data. From the numerical tests, Franke found that RBFs performed better than other tested methods regarding accuracy, stability, efficiency, memory requirement, and simplicity of implementation. Among the tested RBFs, multi-quadrics (MQs henceforth) yields the most accurate results. MQ-RBFs can be written as

$$\varphi(X, X_j) = \varphi(r_j) = \sqrt{r_j^2 + c_j^2}, \quad (3)$$

where  $r_j = |X - X_j|$  is the usual Euclidean distance and  $c_j$  is a shape parameter (for generality, it is written with a variable shape parameter  $c_j$ ) and given by the practitioners. It is noted that the shape parameter plays a very important role in the MQ formulation. Its value determines the fundamental shape of the basis function. These effects were initially observed for scattered data interpolation, but we will see that they also occur in the numerical solution of the PDEs. Thus, the problem of how to select a good value for the parameter  $c$  appears in front of us. Several methods for selecting  $c$  for the MQ interpolants in the two-dimensional case were suggested in the literature. [32] used  $c = 0.815d$  where  $d = 1/N \sum_{i=1}^N d_i$  and  $d_i$  is the distance between the  $i$  th node and its neighboring node. [31] replaced  $d$  by  $D/\sqrt{N}$  where  $D$  is the diameter of the minimal circle enclosing all supporting points and suggested to use  $1.25D/\sqrt{N}$ . Up to now, optimization of shape parameter and its distribution are still under research. If  $\mathbb{P}_q^d$  denotes the space of  $d$ -variate polynomials of order not exceeding  $q$ , and letting the polynomials  $P_1, \dots, P_m$  be the basis of  $\mathbb{P}_q^d$  in  $\mathbb{R}^d$ , then the polynomial  $\mathbb{P}(X)$  in Eq. (2), is usually written in the following form

$$\mathbb{P}(X) = \sum_{i=1}^m \xi_i P_i(X), \quad (4)$$

where  $m = \frac{(q-1+d)!}{(d!(q-1)!)}.$  To determine the coefficients  $(\lambda_1, \lambda_2, \dots, \lambda_N)$  and  $(\xi_1, \xi_2, \dots, \xi_m)$ , extra  $m$  equations are required in addition to the  $N$  equations resulting from the collocating Eq. (2) at the  $N$  points. This is ensured by the  $m$  conditions for Eq. (2),

$$\sum_{j=1}^N \lambda_j P_i(X_j) = 0, i = 1, \dots, m. \quad (5)$$

## 2.2. DQ METHOD FOR PARTIAL DERIVATIVE APPROXIMATION IN MULTI-DIMENSION.

In this section we explain the method to approximate the spatial derivative in two-dimension. However, the method can be generalized for multi-dimensional cases. It is noted that the basic idea of the DQ method is that any derivative can be approximated by a linear weighted sum of function values at some mesh points. We can keep this idea but release the choice of function values along a mesh line in the conventional DQ approximation. In other words, for a two-dimensional problem, any spatial derivative is approximated by a linear weighted sum of all the function values in the whole two-dimensional domain. In this approximation, a mesh point in the two-dimensional domain is represented by one index,  $k$ , while in the conventional DQ approximation, the mesh point is represented by two indexes  $i, j$ . If the mesh is structured, it is easy to establish the relationship between  $i, j$  and  $k$ . For the example,  $k$  can be written as  $k = (i - 1)N_2 + j, i = 1, 2, \dots, N_1; j = 1, 2, \dots, N_2$ . Clearly, when  $i$  is changed from 1 to  $N_1$ , and  $j$  is changed from 1 to  $N_2$  and  $k$  is changed from 1 to  $N = N_1 \times N_2$ . The DQ approximation for the  $n$ -th order derivative with respect to  $x, \frac{\partial^n f}{\partial x^n}$ , and the  $m$ -th order derivative with respect to  $y, \frac{\partial^m f}{\partial y^m}$  at  $(x_k, y_k)$  can be written as

$$\frac{\partial^n f}{\partial x^n}(x_k, y_k) = \sum_{s=1}^N w_{k,s}^{(n)_x} f(x_s, y_s), \quad (6)$$

$$\frac{\partial^m f}{\partial y^m}(x_k, y_k) = \sum_{s=1}^N w_{k,s}^{(m)_y} f(x_s, y_s). \quad (7)$$

[19] employed this technique and made a novel and effective algorithm for the use of RBFs to solve the PDEs. Instead of using polynomials for determining coefficients, they applied RBFs as test functions.

In the following subsection, we will show that the weighting coefficients in Eq. (6) and Eq. (7) and can be determined by the function approximation of RBFs and the analysis of a linear vector space.

**2.3. RBFs-DQ APPROXIMATION.** In this subsection, we will use the MQ-RBFs as test functions to determine the weighting coefficients in the DQ approximation of derivatives for a two-dimensional problem. However, the method can be easily extended to the multi-dimensional case as it is dimension-independent, and other RBFs can also be used as test functions.

Suppose that the solution of a PDE is continuous, which can be approximated by MQ-RBFs in Eq. (3), and only a constant is included in the polynomial term  $\mathbb{P}(x, y)$ . So, for a two-dimensional case, the function approximation can be written as

$$f(x, y) = \sum_{j=1}^N \lambda_j \varphi(|(x, y) - (x_j, y_j)|) + \mu. \quad (8)$$

Eq. (8) has  $(N + 1)$  unknown coefficients and can only be applied at  $N$  nodes. So, we need an additional equation to close the system. To make the problem

be well-posed, this additional equation can be made by Eq. (5) ( the sum of the expansion coefficients to be zero). As a result, we have

$$\sum_{j=1}^N \lambda_j = 0 \implies \lambda_i = - \sum_{j=1, j \neq i}^N \lambda_j. \quad (9)$$

Substituting Eq. (9) into Eq. (8) gives

$$f(x, y) = \sum_{j \neq i, j, i=1}^N \lambda_j (\varphi(|(x, y) - (x_j, y_j)|) - \varphi(|(x, y) - (x_i, y_i)|)) + \mu. \quad (10)$$

The number of unknowns in Eq. (8) is reduced to  $N$ . As no confusion rises,  $\mu$  can be replaced by  $\lambda_i$  and Eq. (10) can be written as

$$f(x, y) = \sum_{j \neq i, j, i=1}^N \lambda_j (\varphi(|(x, y) - (x_j, y_j)|) - \varphi(|(x, y) - (x_i, y_i)|)) + \lambda_i \quad (11)$$

By setting  $g_j(x, y) = \varphi(|(x, y) - (x_j, y_j)|) - \varphi(|(x, y) - (x_i, y_i)|)$ ,  $j = 1, \dots, i-1, i+1, \dots, N$ , Eq. (11) can be further written as

$$f(x, y) = \sum_{j \neq i=1}^N \lambda_j g_j(x, y) + \lambda_i. \quad (12)$$

The form of Eq. (12) constructs an  $N$ -dimensional linear vector space  $\mathbf{V}^N$ . A set of base functions in  $\mathbf{V}^N$  can be taken as  $q_i = 1, q_j = g_j(x, y) = \varphi(|(x, y) - (x_j, y_j)|) - \varphi(|(x, y) - (x_i, y_i)|)$ ,  $j = 1, \dots, i-1, i+1, \dots, N$ . From the concept of linear independence, the bases of a vector space can be considered as linearly independent subset that spans the entire space. From the property of a linear vector space, if all the base functions satisfy the linear Eq. (6) and Eq. (7), so does any function in the space  $\mathbf{V}^N$  represented by Eq. (12). There is an interesting feature. From Eq. (12), while all the base functions are given, the function  $f(x, y)$  is still unknown since the coefficients  $\lambda_i$  are unknown. However, when all the base functions satisfy Eq. (6) and Eq. (7), we can guarantee that  $f(x, y)$  also satisfies Eq. (6) and Eq. (7). In other words, we can guarantee that the solution of a PDE approximated by the RBF satisfies Eq. (6) and Eq. (7). Thus, when the weighting coefficients of DQ approximation are determined by all the base functions, they can be used to discretize the derivatives in a PDE. That is the essence of the RBFs-DQ method. Using the idea of DQ method, the weighting coefficients of the  $n$ -th order partial derivatives can be determined by substituting all the base functions  $q_1, q_2, \dots, q_N$ , into Eq. (6), as

$$\frac{\partial^n q_i(x_i, y_i)}{\partial x^n} = \sum_{k=1}^N w_{i,k}^{(n)x} = 0, \quad (13)$$

$$\frac{\partial^n q_j(x_i, y_i)}{\partial x^n} = \sum_{k=1}^N w_{i,k}^{(n)x} q_j(x_k, y_k), \quad j = 1, 2, \dots, N, j \neq i. \quad (14)$$

For the given  $i$  equation system (13) and (14) has  $N$  unknowns with  $N$  equations. The matrix form for the weighting coefficients can be written as

$$\mathbf{q}_i^{(n)x} = [\mathbf{Q}] \mathbf{W}_i^{(n)x}, \quad (15)$$

where

$$\mathbf{q}_i^{(n)x} = [0, \frac{\partial^n q_1(x_i, y_i)}{\partial x^n}, \dots, \frac{\partial^n q_{i-1}(x_i, y_i)}{\partial x^n}, \frac{\partial^n q_{i+1}(x_i, y_i)}{\partial x^n}, \dots, \frac{\partial^n q_N(x_i, y_i)}{\partial x^n}]^T,$$

$$[\mathbf{Q}] = \begin{bmatrix} 1 & \cdots & 1 \\ q_1(x_1, y_1) & \cdots & q_1(x_N, y_N) \\ \vdots & \ddots & \vdots \\ q_{i-1}(x_1, y_1) & \cdots & q_{i-1}(x_N, y_N) \\ q_{i+1}(x_1, y_1) & \cdots & q_{i+1}(x_N, y_N) \\ \vdots & \ddots & \vdots \\ q_N(x_1, y_1) & \cdots & q_N(x_N, y_N) \end{bmatrix},$$

and

$$\mathbf{W}_i^{(n)x} = [w_{i,1}^{(n)}, w_{i,2}^{(n)}, \dots, w_{i,N}^{(n)}]^T.$$

Clearly, there exists a unique solution only if the collocation matrix  $[\mathbf{Q}]$  is non-singular. The non-singularity of the collocation matrix  $[\mathbf{Q}]$  depends on the properties of used RBFs. [33] proved that matrix  $[\mathbf{Q}]$  is conditionally positive definite for MQ-RBFs. This fact can not guarantee the non-singularity of matrix  $[\mathbf{Q}]$ . [34] showed that cases of singularity are quite rare, and not serious objection to a valuable numerical technique. Therefore, the coefficient vector  $\mathbf{W}_i^{(n)x}$  can be obtained by

$$\mathbf{W}_i^{(n)x} = [\mathbf{Q}]^{-1} \mathbf{q}_i^{(n)x}. \quad (16)$$

Then, the coefficient vector  $\mathbf{W}_i^{(n)x}$  can be used to approximate the  $n$ -th-order derivative in the  $x$  direction for any unknown smooth function at node  $i$ . In a similar manner, all the base functions are substituted into Eq. (7) to approximate the  $m$ -th-order derivative in the  $y$  direction for any unknown smooth function at node  $i$ . For MQ-RBFs,  $q_j(x, y)$  can be written as

$$q_j(x, y) = \sqrt{(x - x_j)^2 + (y - y_j)^2 + c_j^2} - \sqrt{(x - x_i)^2 + (y - y_i)^2 + c_i^2}, \quad (17)$$

where second- and higher-order derivatives of  $q_j(x, y)$  can also be obtained by differentiating Eq. (17). One of the most attractive properties in the above method is that the weighting coefficients are only related to the test functions and the position of the collocation points.

### 3. Global RBFs-DQ method for multi-dimensional Black-Scholes equation

In this section, first we will introduce the variable transformation  $S_i = e^{x_i}$  and  $C(s_1, \dots, s_d, t) = V(x_1, \dots, x_d, t)$ . Under this change of variables Eq. (1) becomes the parabolic PDE with constant coefficients as

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sum_{i,j=1}^d \rho_{ij} \sigma_i \sigma_j \frac{\partial^2 V}{\partial x_i \partial x_j} + \sum_{i=1}^d \left(r - \frac{1}{2} \sigma_i^2\right) \frac{\partial V}{\partial x_i} - rV = 0, \quad (18)$$

as Eq. (18) satisfies the conditions of the following theorem, we can remove the crossing terms

$$\rho_{ij} \sigma_i \sigma_j \frac{\partial^2 V}{\partial x_i \partial x_j}, i \neq j,$$

then we can apply the RBF-DQ method to option pricing.

**Theorem 3.1** *Consider the second-order equation*

$$\sum_{i,j=1}^n a_{ij} u_{x_i x_j} + \sum_{i=1}^n a_i u_{x_i} + a_0 u = 0,$$

assuming that the mixed partial derivatives are equal, we may as well assume that  $a_{ij} = a_{ji}$ , by making an appropriate change of variables, we can write the top-order term  $\sum_{i,j=1}^n a_{ij} u_{x_i x_j}$  as  $\sum_{k=1}^n d_k u_{x_k x_k}$ , where the coefficients  $d_k$  are the eigenvalues of the  $n \times n$  matrix  $A = (a_{ij})$ .

Therefore, the multi-dimensional Black-Scholes equation transformed into

$$-\frac{\partial U}{\partial t} = \sum_{i=1}^d \left\{ a_i \frac{\partial U}{\partial \xi_i} + b_i \frac{\partial^2 U}{\partial \xi_i^2} \right\} - rU \equiv \mathcal{D}U, \quad (19)$$

The domain is discretized by taking  $N$  knots according to the method used in previous section and we would like to discrete the Eq. (19) with respect to time. We introduce a weight  $\theta$  and apply the  $\theta$  implicit scheme to the problem as

$$-\frac{\partial U}{\partial t}|_{(t_n, \chi_k)} \cong \frac{\tilde{U}_k^n - \tilde{U}_k^{n-1}}{\Delta t} = (1 - \theta) \mathcal{D}\tilde{U}|_{(t_n, \chi_k)} + \theta \mathcal{D}\tilde{U}|_{(t_{n-1}, \chi_k)} \quad (20)$$

Denote  $\tilde{U}|_{(t_n, \chi_k)} = \tilde{U}_k^n$ , where represents the approximation of the function value  $U$  at knot  $k$ ,  $\chi_k = (\xi_1^k, \dots, \xi_d^k)$ , and the time  $t_n$ . By applying the RBF-DQ method, Eq. (19) can be rewritten in a discrete form as follows:

$$\begin{aligned} (1 + r(1 - \theta)\Delta t) \tilde{U}_k^n + (1 - \theta)\Delta t \left\{ - \sum_{i=1}^d \left\{ a_i \sum_{j=1}^N w_{kj}^{(1)\xi_i} \tilde{U}_j^n + b_i \sum_{j=1}^N w_{kj}^{(2)\xi_i} \tilde{U}_j^n \right\} \right\} = \\ (1 - r\theta\Delta t) \tilde{U}_k^{n-1} - \theta\Delta t \left\{ - \sum_{i=1}^d \left\{ a_i \sum_{j=1}^N w_{kj}^{(1)\xi_i} \tilde{U}_j^{n-1} + b_i \sum_{j=1}^N w_{kj}^{(2)\xi_i} \tilde{U}_j^{n-1} \right\} \right\}, \end{aligned}$$

where  $k = 1, 2, \dots, N$  and  $w_{kj}^{(1)\varepsilon_i}$  and  $w_{kj}^{(1)\varepsilon_i}$  represents the computed weighting coefficients in the DQ approximation.

#### 4. Stability of the method

In this section, we study the stability of the implicit finite difference method described above. Let us assume that  $\mathbf{U}$  be exact and  $\tilde{\mathbf{U}}$  is the numerical solution of equation (20). Note that the discrete equations in the last section can be rewritten as the following form

$$[\mathbf{I} + (1 - \theta)\Delta t \mathbf{D}] \tilde{\mathbf{U}}_n = [\mathbf{I} - \theta \Delta t \mathbf{D}] \tilde{\mathbf{U}}_{n-1}. \quad (21)$$

So we can write Eq. (21) as

$$\tilde{\mathbf{U}}_n = [\mathbf{I} + (1 - \theta)\Delta t \mathbf{D}]^{-1} [\mathbf{I} - \theta \Delta t \mathbf{D}] \tilde{\mathbf{U}}_{n-1} = \mathbf{E} \tilde{\mathbf{U}}_{n-1}, \quad (22)$$

where

$$\mathbf{D} = r - \sum_{i=1}^d \{a_i \{[\mathbf{Q}]^{-1} [\mathbf{Q}]_{\xi_i^{(1)}}\}^T + b_i \{[\mathbf{Q}]^{-1} [\mathbf{Q}]_{\xi_i^{(2)}}\}^T\}$$

is  $N \times N$  matrix determined by discretized form of Eq. (19) and

$$[\mathbf{Q}]_{\xi_i^{(j)}} = [\mathbf{q}_1^{(j)\varepsilon_i}, \mathbf{q}_2^{(j)\varepsilon_i}, \dots, \mathbf{q}_N^{(j)\varepsilon_i}], j = 1, 2.$$

The scheme for initial value problem is stable if and only if there exists a positive constant  $C$  independent of the mesh spacing and initial data such that

$$\|\tilde{\mathbf{U}}_n\| \leq C \|\tilde{\mathbf{U}}_0\|, n \rightarrow \infty, \Delta t, \Delta \chi \rightarrow 0,$$

If  $\mathbf{E}$  depended on  $n$  we would get a product of the operators at each time level. Taking a norm

$$\|\tilde{\mathbf{U}}_n\| = \|\mathbf{E}^n \tilde{\mathbf{U}}_0\| \leq \|\mathbf{E}^n\| \|\tilde{\mathbf{U}}_0\|, n \rightarrow \infty, \Delta t, \Delta \chi \rightarrow 0,$$

Therefore, the numerical scheme is stable if and only if there exists positive constant  $C$  such that

$$\|\mathbf{E}^n\| \leq C, n \rightarrow \infty, \Delta t, \Delta \chi \rightarrow 0,$$

Since  $\rho(\mathbf{E})$  as the spectral radius of  $\mathbf{E}$  provides a lower bound to any matrix norm, for the scheme to remain stable, we should have  $\rho(\mathbf{E}) \leq 1$  or equivalently we can say that

$$\left| \frac{1 - \theta \Delta t \eta_{\mathbf{D}}}{1 + (1 - \theta) \Delta t \eta_{\mathbf{D}}} \right| \leq 1, \quad (23)$$

which holds for  $\eta_{\mathbf{D}}$ , eigenvalues of the matrix  $\mathbf{D}$ , located in the right half plane. Inequality (23) also shows that stability of the scheme, in the case of RBFs with shape parameter like MQ, depends upon shape parameter.

### 5. Numerical computation of two asset European option

For two-dimensional option pricing under Geometric Brownian motion framework, consider the PDE (1) with  $d = 2$  and the final time condition for European call is

$$C(s_1, s_2, T) = \max(\max(s_1, s_2) - E, 0), \quad (24)$$

where  $E, T$  are exercise price and maturity time. The following boundary conditions are imposed

$$C(s_{1\max}, s_2, T) = s_{1\max} - Ee^{-r(T-t)}, \quad (25)$$

$$C(s_1, s_{2\max}, T) = s_{2\max} - Ee^{-r(T-t)}, \quad (26)$$

where  $s_{1\max}, s_{2\max}$  are respectively maximum of  $s_1$ , maximum of  $s_2$ .

In this case, the variable transformation  $s_1 = e^x, s_2 = e^y$  and  $C(s_1, s_2, t) = V(x, y, t)$ . Under this change of variables the two-dimensional Eq. (1) in which  $d=2$  becomes

$$\frac{\partial V}{\partial t} + \rho\sigma_1\sigma_2 \frac{\partial^2 V}{\partial x \partial y} + \frac{1}{2}\sigma_1^2 \frac{\partial^2 V}{\partial x^2} + \frac{1}{2}\sigma_2^2 \frac{\partial^2 V}{\partial y^2} + (r - \frac{1}{2}\sigma_1^2) \frac{\partial V}{\partial x} + (r - \frac{1}{2}\sigma_2^2) \frac{\partial V}{\partial y} - rV = 0. \quad (27)$$

Now we can remove the crossing term  $\rho\sigma_1\sigma_2 \frac{\partial^2 V}{\partial x \partial y}$ . In this way, we apply characteristic line to remove mixed derivatives in Eq. (27) by considering its characteristic equation for the above PDE [35] as

$$\frac{dy}{dx} = \frac{\rho\sigma_1\sigma_2 + \sqrt{\sigma_1^2\sigma_2^2(\rho^2 - 1)}}{\sigma_1^2} = \frac{\sigma_2}{\sigma_1}(\rho + \sqrt{1 - \rho^2}i),$$

where  $i = \sqrt{-1}$ . Thus we observe the following relation between spatial variables  $x$  and  $y$ ,

$$\sigma_1 y - \sigma_2 \rho x - i\sigma_2 \sqrt{1 - \rho^2} x = c'.$$

Let  $\xi = \sigma_1 y - (\sigma_2 \rho)x, \eta = -\sigma_2 \sqrt{1 - \rho^2}x$ , and  $V(x, y, t) = U(\xi, \eta, t)$  then

$$\left\{ \begin{array}{l} \frac{\partial V}{\partial x} = \frac{\partial U}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial U}{\partial \eta} \frac{\partial \eta}{\partial x} = -\sigma_2 \rho \frac{\partial U}{\partial \xi} - \sigma_2 \sqrt{1 - \rho^2} \frac{\partial U}{\partial \eta} \\ \frac{\partial V}{\partial y} = \frac{\partial U}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial U}{\partial \eta} \frac{\partial \eta}{\partial y} = \sigma_1 \frac{\partial U}{\partial \xi}. \end{array} \right.$$

Further, we have

$$\left\{ \begin{array}{l} \frac{\partial^2 V}{\partial x^2} = \sigma_2^2 \rho^2 \frac{\partial^2 U}{\partial \xi^2} + 2\sigma_2^2 \rho \sqrt{1-\rho^2} \frac{\partial^2 U}{\partial \xi \partial \eta} + \sigma_2^2 (1-\rho^2) \frac{\partial^2 U}{\partial \eta^2} \\ \frac{\partial^2 V}{\partial x \partial y} = -\sigma_1 \sigma_2 \rho \frac{\partial^2 U}{\partial \xi^2} - \sigma_1 \sigma_2 \sqrt{1-\rho^2} \frac{\partial^2 U}{\partial \xi \partial \eta} \\ \frac{\partial^2 V}{\partial y^2} = \sigma_1^2 \frac{\partial^2 U}{\partial \xi^2}. \end{array} \right.$$

Hence,

$$\frac{1}{2} \sigma_1^2 \frac{\partial^2 V}{\partial x^2} + \rho \sigma_1 \sigma_2 \frac{\partial^2 V}{\partial x \partial y} + \frac{1}{2} \sigma_2^2 \frac{\partial^2 V}{\partial y^2} = \frac{1}{2} \sigma_1^2 \sigma_2^2 (1-\rho^2) \left( \frac{\partial^2 U}{\partial \xi^2} + \frac{\partial^2 U}{\partial \eta^2} \right),$$

and

$$\left( r - \frac{1}{2} \sigma_1^2 \right) \frac{\partial V}{\partial x} + \left( r - \frac{1}{2} \sigma_2^2 \right) \frac{\partial V}{\partial y} = - \left( r \sigma_2 \rho - \frac{1}{2} \sigma_1^2 \sigma_2 \rho - r \sigma_1 + \frac{1}{2} \sigma_2^2 \sigma_1 \right) \frac{\partial U}{\partial \xi} - \left( r \sigma_2 \sqrt{1-\rho^2} - \frac{1}{2} \sigma_1^2 \sigma_2 \sqrt{1-\rho^2} \right) \frac{\partial U}{\partial \eta}.$$

Thus, the Black-Scholes equation in (27) transformed into

$$\frac{\partial U}{\partial t} + a \frac{\partial U}{\partial \xi} + b \frac{\partial U}{\partial \eta} + c \left( \frac{\partial^2 U}{\partial \xi^2} + \frac{\partial^2 U}{\partial \eta^2} \right) - rU = 0, \quad (28)$$

where

$$\begin{aligned} a &= -(r \sigma_2 \rho - \frac{1}{2} \sigma_1^2 \sigma_2 \rho - r \sigma_1 + \frac{1}{2} \sigma_2^2 \sigma_1), \\ b &= -(r \sigma_2 \sqrt{1-\rho^2} - \frac{1}{2} \sigma_1^2 \sigma_2 \sqrt{1-\rho^2}), \\ c &= \frac{1}{2} \sigma_1^2 \sigma_2^2 (1-\rho^2). \end{aligned}$$

where the mixed derivative term vanishes. Eq. (28) can be rewritten in a discrete form as follows:

$$\begin{aligned} (1 + r(1-\theta)\Delta t) \tilde{U}_k^n + (1-\theta)\Delta t \{ & -a \sum_{i=1}^N w_{ki}^{(1)\xi} \tilde{U}_i^n - b \sum_{i=1}^N \tilde{w}_{ki}^{(1)\eta} \tilde{U}_i^n - c \left( \sum_{i=1}^N w_{ik}^{(2)\xi} \tilde{U}_i^n + \sum_{i=1}^N \tilde{w}_{ki}^{(2)\eta} \tilde{U}_i^n \right) \} = \\ (1 - r\theta\Delta t) \tilde{U}_k^{n-1} - \theta\Delta t \{ & -a \sum_{i=1}^N w_{ki}^{(1)\xi} \tilde{U}_i^{n-1} - b \sum_{i=1}^N \tilde{w}_{ki}^{(1)\eta} \tilde{U}_i^{n-1} - c \left( \sum_{i=1}^N w_{ik}^{(2)\xi} \tilde{U}_i^{n-1} + \sum_{i=1}^N \tilde{w}_{ki}^{(2)\eta} \tilde{U}_i^{n-1} \right) \} \end{aligned}$$

where  $i = 1, 2, \dots, N$ . For illustration of the accuracy of the proposed method, we consider a European call option with two underlying assets with  $E = 10, r =$

Figure 1: Comparison of two-asset option pricing results by RBF-DQ method and Stulz method

0.05,  $\sigma_1 = 0.22$ ,  $\sigma_2 = 0.14$ ,  $\rho = 0.5$  and  $T = 0.5$  (year) with 100 time steps . Let  $s_1 \in [e^{-3.5}, e^4]$ ,  $s_2 \in [e^{-3.5}, e^4]$ , and  $\theta = 0.5$ . To study the behavior of the method, different structured mesh sizes are used for the point distribution in  $\xi$  and  $\eta$  direction with shape parameter  $c^2 = 1.25(D/\sqrt{N})$  as selected by Franke (1982), where  $D$  is the diameter of the minimal circle enclosing all supporting points. Fig. 1 shows that our scheme can provide reasonable approximations for Stulz method [36] which provides a closed form solution and is available in Matlab. The root-mean-square-error (RMSE) defined by the

$$RMSE = \frac{1}{N} \sqrt{\sum_{i=1}^N (U_{Stz.}^i - \tilde{U}_{num.}^i)^2},$$

where  $U_{Stz.}$  is the solution by Stulz Method,  $\tilde{U}_{num.}$  is solution by numerical approximations. As  $\rho(\mathbf{E})$ , the spectral radius of matrix  $\mathbf{E}$ , is equivalent or less than unity in each simulation, it guarantees the stability of the method.

## 6. Conclusion

A mesh-free RBFs-DQ method is presented and used for numerical solution of multi-dimensional Black-Scholes equation in this paper. In our approach, any spatial derivative is approximated by a linear weighted sum of all the function values in the whole physical domain. The weighting coefficients in the method are determined by RBF approximation and linear vector space analysis. The proposed method is similar to finite difference schemes in the sense of derivative approximation. Numerical results showed that our RBFs-DQ scheme is an efficient approach for solution of multi-asset option pricing.

## References

1. F. Black, M. Scholes, *The pricing of options and corporate liabilities*. J. of political economy, 637-654, (1973).
2. R. C. Merton, *Theory of rational option pricing*. The Bell J. economics and management science, 141-183, (1973).
3. M. Broadie , J. B. Detemple, *Anniversary article: Option pricing: Valuation models and applications*. Management Science 50(9), 1145-1177, (2004).
4. J. Jo , Y. Kim, *Comparison of numerical schems on multi-dimensional black-scholes equation*. Bulletin of the Korean mathematical society 50(6), 2035-2051, (2013).
5. P. Glasserman, *Monte Carlo methods in financial engineering*. Springer vol 53, (2004).
6. A. Ibanez , F. Zapatero, *Monte carlo valuation of american options through computation of the optimal exercise frontier*. J. Financial and Quantitative Analysis 39(02), 253-275, (2004).

7. S. Choi , M. D. Marcozzi, *A numerical approach to american currency option valuation*. J. Derivatives 9(2), 19-29, (2001).
8. S. Choi , M. Marcozzi, *The valuation of foreign currency options under stochastic interest rates*. Computers & Mathematics with Applications 46(5), 741-749, (2003).
9. N. Clarke , K. Parrott, *Multigrid for american option pricing with stochastic volatility*. Applied Mathematical Finance 6(3), 177-195, (1999).
10. P. Forsyth , K. Vetzal , R. Zvan, *A finite element approach to the pricing of discrete lookbacks with stochastic volatility*. Applied Mathematical Finance 6(2), 87-106, (1999).
11. P. Wilmott , J. Dewynne , S. Howison, *Option pricing: mathematical models and computation*. Oxford financial press, (1993)
12. Y. I. Zhu , Y. Sun, *The singularity-separating method for two-factor convertible bonds*. J. Computational Finance 3(1), 91-110, (1999).
13. B. F. Nielsen , O. Skavhaug , A. Tveito, *Penalty methods for the numerical solution of american multi-asset option problems*. J. Computational and Applied Mathematics 222(1), 3-16, (2008).
14. P. P. Boyle , J. Evnine , S. Gibbs, *Numerical evaluation of multivariate contingent claims*. Review of Financial Studies 2(2), 241-250, (1989).
15. D. A Voss , A. Q. M. Khaliq , S. Kazmi , H. He, *A fourth order l-stable method for the black-scholes model with barrier options*. In Computational Science and Its Applications ICCSA 2003, Springer, 199-207, (2003).
16. A. Khaliq , D. Voss , S. Kazmi, *A linearly implicit predictor-corrector scheme for pricing american options using a penalty method approach*. J. Banking & Finance 30(2), 489-502, (2006).
17. G. E. Fasshauer , A. Q. M. Khaliq , D. A. Voss, *Using meshfree approximation for multiasset american options*. J. the Chinese Institute of Engineers 27(4), 563-571, (2004).
18. J. Martin-Vaquero , A. Khaliq , B. Kleefeld, *Stabilized explicit runge-kutta methods for multi-asset american options*. Computers & Mathematics with Applications 67(6), 1293-1308, (2014).
19. C. Shu , H. Ding , K. Yeo, *Solution of partial differential equations by a global radial basis function-based differential quadrature method*. Engineering analysis with boundary elements 28(10), 1217-1226, (2004).
20. R. Bellman, J. Casti, *Differential quadrature and long-term integration*. J. Mathematical Analysis and Applications 34(2), 235-238, (1971).
21. R. Bellman , B. Kashef , J. Casti, *Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations*. J. Computational Physics 10(1), 40-52, (1972).
22. C. Shu , B. E. Richards, *Application of generalized differential quadrature to solve two-dimensional incompressible navier-stokes equations*. International J. Numerical Methods in Fluids 15(7), 791-798, (1992).
23. C. Shu , Y. Chew, *Fourier expansion-based differential quadrature and its application to helmholtz eigenvalue problems*. Communications in Numerical Methods in Engineering 13(8), 643-653, (1997).
24. C. Shu, *Differential quadrature and its application in engineering*. Springer (2000).
25. C. W. Bert , S. K. Jang , A. G. Striz, *Two new approximate methods for analyzing free vibration of structural components*. AIAA J. 26(5), 612-618, (1988).
26. A. Sherbourne , M. Pandey , *Differential quadrature method in the buckling analysis of beams and composite plates*. Computers & Structures 40(4), 903-913, (1991).
27. R. Gutierrez, P. Laura , R. Rossi, *The method of differential quadrature and its application to the approximate solution of ocean engineering problems*. Ocean Engineering 21(1), 57-66, (1994).

28. C.W. Bert , M. Malik, *The differential quadrature method for irregular domains and application to plate vibration*. International J. Mechanical Sciences 38(6), 589-606, (1996).
29. J. B. Han , K. Liew, *An eight-node curvilinear differential quadrature formulation for reissner/mindlin plates*. Computer Methods in Applied Mechanics and Engineering 141(3), 265-280, (1997).
30. C. Shu, W. Chen , H. Du, *Free vibration analysis of curvilinear quadrilateral plates by the differential quadrature method*. J. Computational Physics 163(2), 452-466, (2000).
31. R. Franke, *Scattered data interpolation: Tests of some methods*. Mathematics of computation 38(157), 181-200, (1982).
32. R. L. Hardy, *Multiquadric equations of topography and other irregular surfaces*. J. geophysical research 76(8), 1905-1915, (1971).
33. C. A. Micchelli, *Interpolation of scattered data: distance matrices and conditionally positive definite functions*. Springer (1984).
34. Y. Hon , R. Schaback, *On unsymmetric collocation by radial basis functions*. Applied Mathematics and Computation 119(2), 177-186, (2001).
35. J. Zhou , *Multi-asset option pricing with Levy process*. Ph. D. Thesis, National University of Singapore, (2009).
36. R. Stulz, *Options on the minimum or the maximum of two risky assets: analysis and applications*. J. Financial Economics 10(2), 161-185, (1982).

Mojtaba Ranjbar  
 Department of Mathematics,  
 Azarbaijan Shahid Madani University  
 Tabriz - Iran  
 m\_ranjbar@azaruniv.edu