

RESEARCH

Open Access



# A new kind of parallel finite difference method for the quanto option pricing model

Xiaozhong Yang, Lifei Wu\* and Yuying Shi

\*Correspondence:  
wulf@ncepu.edu.cn  
School of Mathematics and Physics,  
North China Electric Power  
University, Beijing, 102206, China

## Abstract

The quanto option pricing model is an important financial derivatives pricing model; it is a two-dimensional Black-Scholes (B-S) equation with a mixed derivative term. The research of its numerical solutions has theoretical value and practical application significance. An alternating band Crank-Nicolson (ABdC-N) difference scheme for solving the quanto options pricing model was constructed. It is constituted of the classical implicit scheme, the explicit scheme and the Crank-Nicolson scheme, it has the following advantages: parallelism, high precision, and unconditional stability. Numerical experiments and theoretical analysis all show that ABdC-N scheme can be used to solve the quanto options pricing problems effectively.

**Keywords:** quanto options pricing model; alternating band Crank-Nicolson (ABdC-N) scheme; stability; parallel computing; numerical experiments

## 1 Introduction

The multi-asset options pricing model (the multi-dimensional Black-Scholes equation) is a famous financial mathematics basic model; its numerical solutions had played a significant role in promoting a lot of financial derivatives pricing methods. Therefore, the numerical solutions have attracted more and more attention from applied mathematicians and economists. With the rapid development of multi-core and cluster technology, parallel algorithms have become one of the mainstream technologies improving the numerical calculation efficiency. The research of parallel numerical difference methods for solving a multi-asset options pricing problem has basic scientific significance. This is so because the option pricing has higher time requirements from the need of practical application. Therefore, over the past 20 years an efficient numerical solution of the multi-asset options pricing model has been the focus of academic research [1].

A quanto option is a kind of multi-asset option, the two-dimensional Black-Scholes (2D B-S) equation of the quanto option is [1–3]

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2} \left[ S_1^2 \sigma_1^2 \frac{\partial^2 V}{\partial S_1^2} + 2\rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} \right] \\ - (r_1 - q_1) S_1 \frac{\partial V}{\partial S_1} - (r_2 - q_2) \frac{\partial V}{\partial S_2} + r_1 V = 0, \\ q_1 = r_1 - r_2 + q + \sigma_1\sigma_2\rho, \quad q_2 = r_2. \end{aligned} \tag{1}$$

Here, we assume an American investor buys a Nikkei index call option.  $V$  is the price of the quanto option (dollar),  $S_1$  is the price of foreign risk asset (yen),  $S_2$  is the exchange rate of the foreign currency against the domestic one (dollar),  $r_1$  is the domestic interest rate without risk,  $r_2$  is the foreign rate without risk,  $\sigma_1$  is the volatility of  $S_1$ ,  $\sigma_2$  is the volatility of  $S_2$ ,  $\rho$  is the correlation coefficient, and  $q$  is the dividend.

This equation has the analytical solutions:

$$\begin{aligned}
 V(S_1, S_1, t) &= \frac{e^{-r_1(T-t)}}{2\pi(T-t)\sigma_1\sigma_2\sqrt{1-\rho^2}} \\
 &\quad \times \int_0^\infty \int_0^\infty \frac{(\eta_1 - K)^+}{\eta_1} \exp\left[\frac{\sigma_1^2\alpha_2^2 - 2\rho\sigma_1\sigma_2\alpha_1\alpha_2 + \sigma_2^2\alpha_1^2}{2\sigma_1^2\sigma_2^2(1-\rho^2)(T-t)}\right] d\eta_1 d\eta_2, \\
 \alpha_1 &= \ln \frac{S_1}{\eta_1} + \left(r_2 - q - \rho\sigma_1\sigma_2 - \frac{\alpha_1^2}{2}\right)(T-t), \\
 \alpha_2 &= \ln \frac{S_2}{\eta_2} + \left(r_1 - r_2 - \frac{\alpha_2^2}{2}\right)(T-t).
 \end{aligned}$$

Here,  $K$  is the strike price of the options (yen),  $T$  is the due date of the options (year).

The analytical solution is very complex, difficult to quickly solve, so numerical solutions were usually used to compute option pricing models in the real financial market, for example, the Monte-Carlo method, the binary tree method, and the finite difference method, *etc.* [1, 2]. The considered computing speed and accuracy, and the finite difference method was usually used in the real financial market.

In recent years, the study of finite difference methods for solving the dual currency option pricing model has made a lot of progress. An implicit scheme for the multi-asset option pricing model had been made by Gilli *et al.* (2002), but a calculation of this scheme was relatively complex; one needed to solve algebra equations which contained a large tridiagonal block matrix [4]. Khaliq *et al.* (2008) had given a kind of difference method for solving the 2D B-S equation [5]; the method needs to use the penalty function approach. So the method was difficult on using parallel computing on a computer. Yang and Zhou (2011) had put forward a rapid AOS difference method of quanto option pricing model [6], but the calculation accuracy of the method is not ideal, because the error of its mixed derivative term is not ideal. In addition, most of those schemes had applied serial calculation. When the computing grid points or dimension of equation required is large, a higher order algebraic equation  $Ax = b$  should be solved. The efficiency of the calculation process is not ideal, and it is difficult to meet the requirements of the options as regards time.

About the parallel finite difference scheme, Evans and Abdullah (1983) had put forward an alternating group explicit (AGE) scheme based on the Saul'yev asymmetric format [7]. The AGE scheme not only was good to keep the stability of numerical calculation, but it also has good parallel properties. Then Zhang (1991) had established a variety of alternating segment explicit-implicit (ASE-I) schemes and alternating segment Crank-Nicolson (ASC-N) schemes [8], and they had got some research results which contained stability and the parallelism. Now, the research of this method has been extended to solve many development equations. For example, Wang (2006) had given a kind of alternating segment difference scheme with intrinsic parallelism for the KdV equation [9]. Sheng *et al.* (2007) had constructed two kinds of difference formats with intrinsic parallelism for a linear parabolic equation [10], and they proved that the format is unconditionally stable

and we have second-order convergence. Yuan (2007) has put forward a parallel difference scheme with second-order accuracy and unconditional stability for a nonlinear parabolic system [11].

For the quanto option pricing model (2D B-S equation), we used the classical implicit scheme, the explicit scheme, and the Crank-Nicolson scheme, constructed a parallel difference scheme-alternating band Crank-Nicolson (ABdC-N) scheme, which is unconditionally stable, and which is close to second-order accuracy. Numerical experiments show that the method is effective.

## 2 ABdC-N difference scheme

### 2.1 Initial-boundary value condition of 2D B-S equation

In order to solve the equation of the quanto option pricing model, the initial condition and boundary condition meeting (1) will be given in this section. In theory, the solving area of this equation is

$$\{(S_1, S_2, t) \mid 0 < S_1 < \infty, 0 < S_2 < \infty, t \in [0, T]\}.$$

But in the actual transaction, the price of the underlying asset will not always appear to be zero or infinity. Therefore, the financial institution provides a small enough value  $S_{\min}$  ( $S_{\min} > 0$ ) as the lower bound and a large enough value  $S_{\max}$  ( $S_{\max} > 0$ ) as the upper bound for it. Then the pricing problem can be solved in the bounded area

$$\Omega = \{(S_1, S_2, t) \mid S_{1\min} < S_1 < S_{1\max}, S_{2\min} < S_2 < S_{2\max}, t \in [0, T]\}.$$

Assume that the foreign option is the call option, then construct the initial and boundary conditions for (1). For the reason that the option pricing is a backward problem, the initial condition is

$$V(S_1, S_2, T) = S_2 \cdot \max\{S_1 - K, 0\}.$$

The boundary conditions are

$$\begin{aligned} V(S_{1\min}, S_2, t) &= 0, & V(S_{1\max}, S_2, t) &= 0, \\ V(S_1, S_{2\min}, t) &= 0, & V(S_1, S_{2\max}, t) &= 0. \end{aligned}$$

In order to solve (1), we can substitute its variable as follows [2, 12]:

$$x = \ln S_1, \quad y = \ln S_2, \quad \tau = T - t.$$

Then this pricing model will be transformed into the initial-boundary value problem of a partial differential equation with constant coefficients:

$$\begin{aligned} \frac{\partial V}{\partial \tau} - \frac{1}{2} \left[ \sigma_1^2 \frac{\partial^2 V}{\partial x^2} + 2\rho\sigma_1\sigma_2 \frac{\partial^2 V}{\partial x \partial y} + \sigma_2^2 \frac{\partial^2 V}{\partial y^2} \right] \\ - \left( r_1 - q_1 - \frac{\sigma_1^2}{2} \right) \frac{\partial V}{\partial x} - \left( r_2 - q_2 - \frac{\sigma_2^2}{2} \right) \frac{\partial V}{\partial y} + r_1 V = 0, \end{aligned}$$

$$q_1 = r_1 - r_2 + q + \sigma_1\sigma_2\rho, \quad q_2 = r_2. \tag{2}$$

Initial condition:

$$V(x, y, 0) = e^y \cdot \max\{e^x - K, 0\}.$$

Boundary conditions:

$$\begin{aligned} V(\ln S_{1\min}, y, \tau) &= 0, & V(\ln S_{1\max}, y, \tau) &= 0, \\ V(x, \ln S_{2\min}, \tau) &= 0, & V(x, \ln S_{2\max}, \tau) &= 0. \end{aligned}$$

### 2.2 Construction of the ABdC-N scheme

Let  $\Delta x, \Delta y, \Delta \tau$  be the steps of  $x, y$ , and  $\tau$ , respectively. Here,  $\Delta x = (\ln S_{1\max} - \ln S_{1\min})/m$ ,  $\Delta y = (\ln S_{2\max} - \ln S_{2\min})/n$ ,  $\Delta \tau = T/nt$ ,  $m, n, nt$  are positive integers.  $x_i = \ln S_{1\min} + i\Delta x$ ,  $y_j = \ln S_{2\min} + j\Delta y$ ,  $\tau_k = k\Delta \tau$ ,  $i = 0, 1, \dots, m, j = 0, 1, \dots, n, k = 0, 1, \dots, nt$ . For convenience, let  $h = \Delta x = \Delta y$ . We use  $V_{ij}^k$  to denote the solution of (2) at point  $(x_i, y_j, \tau_k)$ .

We introduce the following notation:

$$\begin{aligned} \delta_x V_{ij}^k &= (V_{i+1,j}^k - V_{i-1,j}^k)/2h, \\ \delta_x^2 V_{ij}^k &= (V_{i+1,j}^k - 2V_{ij}^k + V_{i-1,j}^k)/h^2, \\ \delta_y V_{ij}^k &= (V_{i,j+1}^k - V_{i,j-1}^k)/2h, \\ \delta_y^2 V_{ij}^k &= (V_{i,j+1}^k - 2V_{ij}^k + V_{i,j-1}^k)/h^2, \\ RV_{ij}^k &= (V_{i+1,j+1}^k - V_{i-1,j+1}^k - V_{i+1,j-1}^k + V_{i-1,j-1}^k)/4h^2, \\ \Delta_\tau V_{ij}^k &= (V_{ij}^{k+1} - V_{ij}^k)/\Delta \tau. \end{aligned}$$

A classical difference scheme of (2) is as follows:

$$\begin{aligned} \Delta_\tau V_{ij}^k &= \left(\frac{1}{2}(\sigma_1^2 \delta_x^2 + \sigma_2^2 \delta_y^2 + 2\rho\sigma_1\sigma_2 R)\right) + \left(r_1 - q_1 - \frac{\sigma_1^2}{2}\right) \delta_x \\ &+ \left(r_2 - q_2 - \frac{\sigma_2^2}{2}\right) \delta_y - r \left(\theta V_{ij}^k + (1 - \theta)V_{ij}^{k+1}\right). \end{aligned}$$

The above scheme can be written as

$$\begin{aligned} &\theta(-a_1 V_{i-1,j}^{k+1} - a_2 V_{i,j-1}^{k+1} - b_1 V_{i+1,j}^{k+1} - b_2 V_{i,j+1}^{k+1} - cRV_{ij}^{k+1}) + (1 + \theta\alpha)V_{ij}^{k+1} \\ &= (1 - \theta)(-a_1 V_{i-1,j}^k - a_2 V_{i,j-1}^k - b_1 V_{i+1,j}^k - b_2 V_{i,j+1}^k - cRV_{ij}^k) + (1 + (1 - \theta)\alpha)V_{ij}^k. \end{aligned} \tag{3}$$

Here  $a_1 = \frac{\sigma_1^2 \Delta \tau}{2h^2} - \frac{\Delta \tau}{2h}(r_1 - q_1 - \frac{\sigma_1^2}{2})$ ,  $a_2 = \frac{\sigma_2^2 \Delta \tau}{2h^2} - \frac{\Delta \tau}{2h}(r_2 - q_2 - \frac{\sigma_2^2}{2})$ ,  $b_1 = \frac{\sigma_1^2 \Delta \tau}{2h^2} + \frac{\Delta \tau}{2h}(r_1 - q_1 - \frac{\sigma_1^2}{2})$ ,  $b_2 = \frac{\sigma_2^2 \Delta \tau}{2h^2} + \frac{\Delta \tau}{2h}(r_2 - q_2 - \frac{\sigma_2^2}{2})$ ,  $c = \rho\sigma_1\sigma_2 \frac{\Delta \tau}{2h^2}$ ,  $\alpha = \frac{\Delta \tau}{h^2}(\sigma_1^2 + \sigma_2^2) + r\Delta \tau$ ,  $RV_{ij}^k = V_{i+1,j+1}^k - V_{i+1,j-1}^k - V_{i-1,j+1}^k + V_{i-1,j-1}^k$ .

Here, (3) is called the universal difference scheme ( $\theta$ -scheme).

As is well known, when  $\theta = 0$ , (3) is a classical explicit scheme, which has a natural parallelism, but its stability condition is more demanding. When  $\theta = 1$ , (3) is the classical

implicit scheme, which has unconditional stability. When  $\theta = 0.5$ , (3) is a classical Crank-Nicolson scheme, which is of second-order accuracy and has unconditional stability. But the implicit scheme's and the C-N scheme's computing times are longer.

The design of ABdC-N is as follows.

Assume the value of the  $k$ th time layer  $V_{ij}^k$  ( $i = 1, 2, \dots, m - 1$ ) is known, the value of the  $k + 1$ th time layer  $V_{ij}^{k+1}$  waits for calculating. Assume  $s$  is a positive integer and  $1 < 2s \leq m - 1$ ,  $I_l, l = 1, 2, \dots, 2s$  are  $2s$  positive integers, which meet  $1 \leq I_1 < I_2 < \dots < I_{2s} \leq m - 1$ .

When  $k$  is an even number, at point  $x_{ij}$  ( $i = I_1, I_3, \dots, I_{2s-1}$ ), we apply the classical explicit scheme ( $\theta = 0$ ) to calculate  $V_{ij}^{k+1}$ ; at point  $x_{ij}$  ( $i = I_2, I_4, \dots, I_{2s}$ ), we apply the classical implicit scheme ( $\theta = 1$ ) to calculate  $V_{ij}^{k+1}$ . At the remaining points, we apply the classical C-N scheme ( $\theta = 0.5$ ) to calculate  $V_{ij}^{k+1}$ . When  $k$  is an odd number, at point  $x_{ij}$  ( $i = I_2, I_4, \dots, I_{2s}$ ), we apply the implicit scheme ( $\theta = 0$ ) to calculate  $V_{ij}^{k+1}$ ; at point  $x_{ij}$  ( $i = I_1, I_3, \dots, I_{2s-1}$ ), we apply the explicit scheme ( $\theta = 1$ ) to calculate  $V_{ij}^{k+1}$ . At the remaining points, we apply the classical C-N scheme ( $\theta = 0.5$ ) to calculate  $V_{ij}^{k+1}$ .

The matrix form of the ABdC-N scheme is

$$\begin{cases} (E + D_1G)V^{k+1} = (E - D_2G)V^k + g^k/2, \\ (E + D_2G)V^{k+2} = (E - D_1G)V^{k+1} + g^{k+1}/2, \end{cases} \quad k = 0, 2, \dots, \tag{4}$$

$$G = \begin{pmatrix} A_{m-1} & C_{m-1} & & \\ B_{m-1} & A_{m-1} & C_{m-1} & \\ & \ddots & \ddots & \ddots \\ & & B_{m-1} & A_{m-1} \end{pmatrix}_{(m-1)^2 \times (m-1)^2},$$

$$A_{m-1} = \begin{pmatrix} 2\alpha & -2b_2 & & \\ 2a_2 & 2\alpha & -2b_2 & \\ & \ddots & \ddots & \ddots \\ & & 2a_2 & 2\alpha \end{pmatrix}_{(m-1) \times (m-1)},$$

$$B_{m-1} = \begin{pmatrix} 2a_1 & 2c & & \\ -2c & 2a_1 & 2c & \\ & \ddots & \ddots & \ddots \\ & & -2c & 2a_1 \end{pmatrix}_{(m-1) \times (m-1)},$$

$$C_{m-1} = \begin{pmatrix} 2b_1 & 2c & & \\ -2c & 2b_1 & 2c & \\ & \ddots & \ddots & \ddots \\ & & -2c & 2b_1 \end{pmatrix}_{(m-1) \times (m-1)},$$

$$g^k = \begin{pmatrix} g_1 \\ \vdots \\ g_{m-1} \end{pmatrix}_{(m-1) \times 1}, \quad g_1 = \begin{pmatrix} a_2(V_{1,0}^{k+1} + V_{1,0}^k) + \rho_1(V_{1,1}^k) + c(V_{2,0}^{k+1} + V_{2,0}^k) \\ \vdots \\ \rho_1(V_{1,j}^k) \\ \vdots \\ b_2(V_{1,m}^{k+1} + V_{1,m}^k) + \rho_1(V_{1,m-1}^k) + c(V_{3,m}^{k+1} + V_{3,m}^k) \end{pmatrix}_{(m-1) \times 1},$$

$$\begin{aligned}
 g_{m-1} &= \begin{pmatrix} a_2(V_{m-1,0}^{k+1} + V_{m-1,0}^k) + \rho_1(V_{m-1,1}^k) + c(V_{m-2,0}^{k+1} + V_{m-2,0}^k) \\ \vdots \\ \rho_2(V_{m-1,j}^k) \\ \vdots \\ b_2(V_{m-1,m}^{k+1} + V_{m-1,m}^k) + \rho_1(V_{m-1,m-1}^k) + c(V_{m-2,m}^{k+1} + V_{m-2,m}^k) \end{pmatrix}_{(m-1) \times 1}, \\
 g_i &= \begin{pmatrix} a_2(V_{i,0}^{k+1} + V_{i,0}^k) + c(V_{i-1,0}^{k+1} + V_{i-1,0}^k - V_{i+1,0}^{k+1} - V_{i+1,0}^k) \\ 0 \\ \vdots \\ 0 \\ b_2(V_{i,m}^{k+1} + V_{i,m}^k) + c(V_{i-1,m}^{k+1} + V_{i-1,m}^k - V_{i+1,m}^{k+1} - V_{i+1,m}^k) \end{pmatrix}_{(m-1) \times 1}, \\
 \rho_1(V_{1,j}^k) &= a_1(V_{0,j}^{k+1} + V_{0,j}^k) + c(V_{0,j-1}^{k+1} + V_{0,j-1}^k - V_{0,j+1}^{k+1} - V_{0,j+1}^k), \\
 \rho_2(V_{m-1,j}^k) &= b_1(V_{m,j}^{k+1} + V_{m,j}^k) + c(V_{m,j-1}^{k+1} + V_{m,j-1}^k - V_{m,j+1}^{k+1} - V_{m,j+1}^k).
 \end{aligned}$$

$D_1$  and  $D_2$  are  $(m - 1)^2$ th-order diagonal matrices and they meet  $D_1 + D_2 = E$ ,  $E$  is unit matrix.  $D_1 = (\theta_{i,j})$ ,

$$\theta_{i,j} = \begin{cases} 0, & i = I_{2l}, 0 < l \leq s, \\ 1, & i = I_{2l-1}, 0 < l \leq s, \\ 0.5, & \text{elsewhere.} \end{cases}$$

### 3 Existence and uniqueness of the ABdC-N scheme solution

Assume the value  $V_{ij}^k$  of the  $k$ th time layer is known, the value  $V_{ij}^{k+1}$  of the  $k + 1$ th time layer waits for calculating. From the ABdC-N scheme (4), the matrix equation for calculating the value of the  $k + 1$ th time layer is

$$(E + D_1G)V^{k+1} = (E - D_2G)V^k + g^k/2. \tag{5}$$

The coefficient matrix is  $E + D_1G$ . From the expression of  $G$ , we can see that  $G$  is a diagonally dominant matrix. So  $E + D_1G$  is also a diagonally dominant matrix. In other words,  $E + D_1G$  is a nonsingular matrix. Therefore, formally (5) has a unique solution. Similarly, applying the ABdC-N scheme (4) to calculate the value of the  $k + 2$ th time layer, the coefficient matrix  $E + D_2G$  of this matrix equation is also a nonsingular matrix. Therefore, this matrix equation of the  $k + 2$ th time layer has a unique solution.

Based on the above analysis, we will get the following theorem.

**Theorem 1** *The ABdC-N scheme (4) for solving the quanto option pricing model is uniquely solvable.*

### 4 Stability and convergence of the ABdC-N scheme

The stability and convergence of the ABdC-N scheme for solving the quanto option pricing model will be analyzed in this section. The growth matrix of the ABdC-N scheme (4) is

$$M = (E + D_2G)^{-1}(E - D_1G)(E + D_1G)^{-1}(E - D_2G).$$

For discussing the stability of the ABdC-N scheme, we need to introduce the Kellogg lemma [13].

**Lemma 1** *If  $\rho > 0$  and  $C + C^T$  is a non-negative (or positive) matrix, then  $(\rho E + C)^{-1}$  exists, and  $\|(\rho E - C)(\rho E + C)^{-1}\|_2 \leq 1$ .*

**Lemma 2**  *$D_1G, D_2G$  in the growth matrix of the ABdC-N scheme for solving the quanto option pricing model are non-negative matrices.*

*Proof* If  $D_1G, D_2G$  meet the requirement that  $D_1G + (D_1G)^T, D_2G + (D_2G)^T$  are non-negative matrices, and Lemma 2 is correct. Therefore, we only need to prove  $D_1G + (D_1G)^T, D_2G + (D_2G)^T$  are non-negative matrices. We have

$$\begin{aligned}
 D_1G + (D_1G)^T &= D_1(G + G^T) \\
 &= D_1 \begin{pmatrix} 2A_{m-1} & -2(a_1 + b_1)E & & & \\ -2(a_1 + b_1)E & 2A_{m-1} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & -2(a_1 + b_1)E & 2A_{m-1} \end{pmatrix}, \\
 A_{m-1} &= 2 \begin{pmatrix} \alpha & -a_2 - b_2 & & & \\ -a_2 - b_2 & \alpha & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & -a_2 - b_2 & \alpha \end{pmatrix}.
 \end{aligned}$$

Form the definition of  $a_1, b_1, a_2, b_2, \alpha$ , it is obvious that  $D_1G + (D_1G)^T$  is a diagonally dominant matrix and the diagonal elements of  $D_1G + (D_1G)^T$  are non-negative real numbers. In other words,  $D_1G + (D_1G)^T$  is a non-negative matrix. Similarly, we see that  $D_2G + (D_2G)^T$  is also a non-negative matrix. Therefore,  $D_1G$  and  $D_2G$  are non-negative matrices.  $\square$

Note

$$M_1 = (E + D_2G)T(E + D_2G)^{-1} = (E - D_1G)(E + D_1G)^{-1}(E - D_2G)(E + D_2G)^{-1}.$$

From Lemma 2, we know that  $A_1G$  and  $A_2G$  are non-negative matrices. We can apply Lemma 1 to get the following inequality easily:

$$\|(E - D_iG)(E + D_iG)^{-1}\|_2 \leq 1, \quad i = 1, 2.$$

Then we get  $\rho(M) = \rho(M_1) \leq \|T_1\|_2 \leq 1$ . Therefore, we get the following theorem.

**Theorem 2** *The ABdC-N scheme (4) for solving the quanto option pricing model is unconditionally stable.*

Due to the Lax theorem [14], we can get a corollary.

**Corollary 1** *The ABdC-N scheme (4) for solving the quanto option pricing model is convergent.*

### 5 Accuracy of the ABdC-N scheme

First, the accuracy of universal difference scheme (3) for solving the quanto option pricing model will be analyzed in this section. The universal difference scheme will be expanded as the Taylor series at the point  $(x_i, y_j, \tau_k)$ . Then we can get the truncation error  $T_1(\Delta\tau, h)$ ,

$$\begin{aligned}
 T_1(\Delta\tau, h) &= -\frac{\partial V}{\partial \tau} - \frac{\Delta\tau}{2} \frac{\partial^2 V}{\partial \tau^2} + \frac{1}{2} \left\{ \sigma_1^2 \left( \frac{\partial^2 V}{\partial x^2} + \frac{h^2}{12} \frac{\partial^4 V}{\partial x^4} \right) + \sigma_2^2 \left( \frac{\partial^2 V}{\partial y^2} + \frac{h^2}{12} \frac{\partial^4 V}{\partial y^4} \right) \right. \\
 &\quad + 2\rho\sigma_1\sigma_2 \left( \frac{\partial^2 V}{\partial x \partial y} + \frac{h^3}{3} \frac{\partial^4 V}{\partial x \partial y^3} + \frac{h^3}{3} \frac{\partial^4 V}{\partial x^3 \partial y} \right) \\
 &\quad \left. + \theta \Delta\tau \left( 2\rho\sigma_1\sigma_2 \frac{\partial^3 V}{\partial \tau \partial x \partial y} + \sigma_1^2 \frac{\partial^3 V}{\partial \tau \partial x^2} + \sigma_2^2 \frac{\partial^3 V}{\partial \tau \partial y^2} \right) \right\} \\
 &\quad + \left( r_1 - q_1 - \frac{\sigma_1^2}{2} \right) \left( \frac{\partial V}{\partial x} + \theta \Delta\tau \frac{\partial^2 V}{\partial \tau \partial x} + \frac{h^3}{3} \frac{\partial^3 V}{\partial x^3} \right) \\
 &\quad + \left( r_2 - q_2 - \frac{\sigma_2^2}{2} \right) \left( \frac{\partial V}{\partial y} + \theta \Delta\tau \frac{\partial^2 V}{\partial \tau \partial y} + \frac{h^3}{3} \frac{\partial^3 V}{\partial y^3} \right) \\
 &\quad - r \left( V + \theta \Delta \frac{\partial V}{\partial \tau} \right) + O(\Delta\tau^2 + h^2). \tag{6}
 \end{aligned}$$

When  $\theta = 0.5$ , the truncation error is  $O(\Delta\tau^2, h^2)$ , the universal difference scheme (3) is the Crank-Nicolson scheme at this time. The truncation error is of second order in time and space. When  $\theta = 0, 1$ , the truncation is of first order in time and second order of space.

From the construction of the ABdC-N scheme, we take inside points without interior boundary points as ‘interior point’. Because the ABdC-N scheme (4) is applied to the C-N scheme at the interior point, the truncation error of the interior point is of second order.

The truncation error of interior boundary points will be analyzed in the following. The ABdC-N scheme (4) alternatively applies the classical explicit scheme and the implicit scheme at the interior boundary points.

The two classical schemes approximate the analytical solution from either side, respectively. It had been proved that the numerical solution of the classical explicit scheme  $\theta = 0$  is greater than the analytical solution, and the numerical solution of the classical implicit scheme  $\theta = 1$  is less than the analytical solution. Therefore, alternatively applying them can improve the calculation accuracy. For example, the truncation error of the ‘Explicit-Implicit scheme’ or the ‘Implicit-Explicit scheme’ is of second order in time and space, and unconditionally stable. In the ABdC-N scheme (4) for solving the quanto option pricing model is alternatively applied the classical explicit and the implicit scheme at the interior boundary point; then the truncation error of interior boundary point also can achieve a second order.

**Theorem 3** *The truncation error of the ABdC-N scheme (4) for solving the quanto option pricing model is  $O(\Delta\tau^2 + h^2)$ .*

In order to make have the ABdC-N scheme a better parallelism, usually we take  $s$  constant. Then we can use  $(s + 1)$  CPU for parallel computing. Every CPU applies the Thomas method to solve a tridiagonal equation. Usually, we take  $I_{2l} - I_{2l-1}$  and  $I_{2l+1} - I_{2l}$  ( $0 < l \leq s$ ) constant for different  $l$ .

### 6 Numerical experiments

Numerical experiments will be done in Matlab 2008a, based on Intel core i5-2400 CPU@3.10GHz. The comparison is between the ABdC-N scheme and the classical C-N scheme, referring to computing accuracy and computing time.

**Example** We consider an American investor buying a Nikkei index call option. Assuming the current price of Nikkei is 20,000 yen, the dividend rate of the Nikkei is 0.03, the volatility of the Nikkei is 0.2, the exchange rate of Japanese yen against dollar is 0.01, the volatility of the exchange rate is 0.1, the correlation coefficient is 0.2, the risk-free rates of American and Japan are 0.08 and 0.04, respectively. The strike price of an option is 15,000 yen. We consider the deadline of the option to be one year (12 months), and the final exchange rate is the spot exchange rate.

In terms of the computation accuracy, from Table 1, we can see that the accuracy of numerical solution of the ABdC-N scheme is high, and is close to the classical C-N scheme. By comparing numerically the ABdC-N scheme and the C-N scheme from Table 1, the relative error of the numerical solution of ABdC-N is 2.49%, and the relative error of the classical C-N scheme is 4.09%. The numerical result of the ABdC-N scheme is closer to the analytical solution than the classical C-N scheme.

Next we will analyze the root mean square error (RSME) of the ABdC-N scheme and C-N scheme. The solutions of the difference scheme are denoted as  $\bar{u}_{i,j}$ . The analytical solution is denoted as  $u_{i,j}$ . The definition of RSME is as follows:

$$RSME = \sqrt{\frac{\sum_{i=1}^M \sum_{j=1}^N (u_{i,j} - \bar{u}_{i,j})^2}{M * N}}$$

The ratio of RSME (RRSME) is defined as

$$RRSME = \frac{RSME}{h}$$

From Tables 2 and 3, the RSMEs of the ABdC-N scheme and the C-N scheme are becoming smaller and smaller with mesh grid refinement. The RSME of the ABdC-N scheme is smaller than the classical C-N scheme's. This shows that ABdC-N scheme is better than the classical C-N scheme in terms of computation accuracy. The RRSMEs of the ABdC-N

**Table 1 Comparison of analytical and numerical solution**

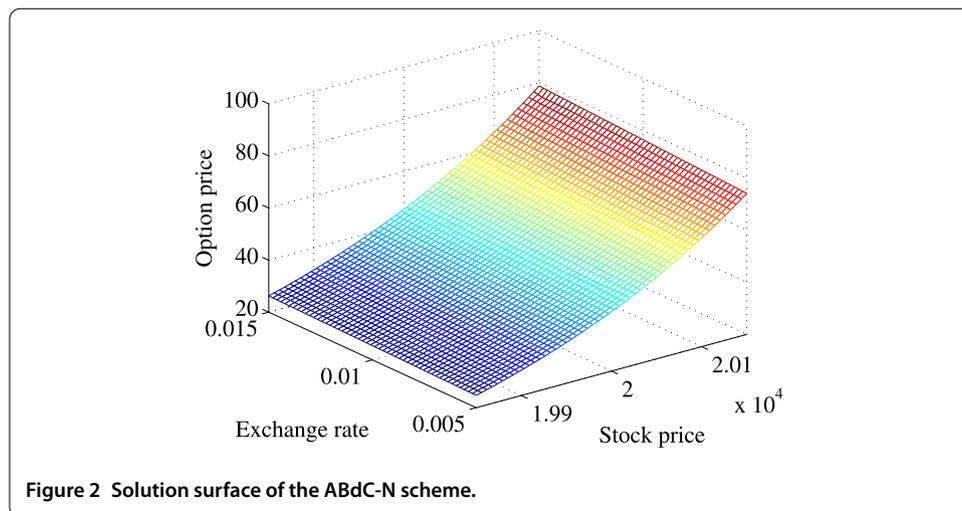
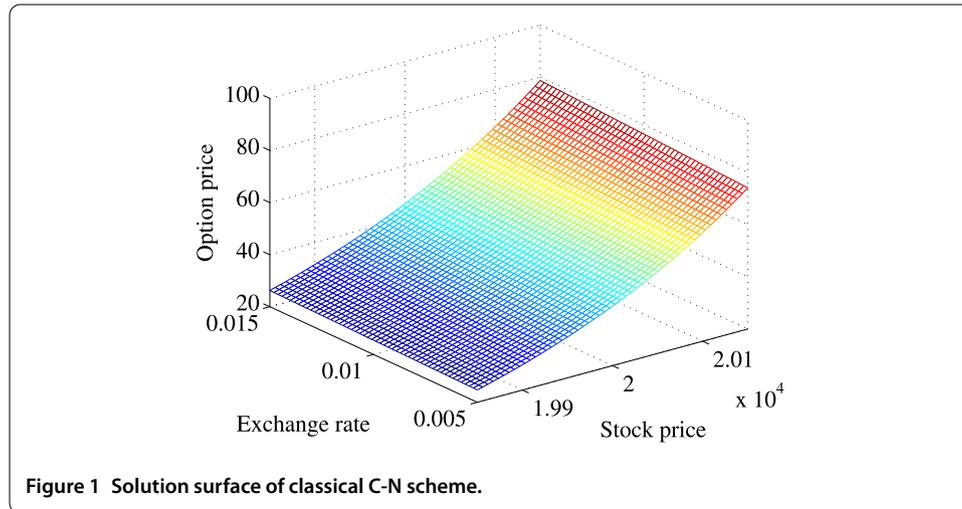
Scheme	12 months	Relative error	CPU time
Analytical solution	53.521809	-	-
Classical C-N scheme	51.331487	4.09%	158.157s
ABdC-N scheme	52.284465	2.49%	18.037s

**Table 2 Error analysis of classical C-N scheme**

Grid	CPU time	RMSE	RRMSE
20 × 20	0.836s	0.1450	2.900
30 × 30	7.259s	0.0999	2.997
40 × 40	38.07s	0.0762	3.048
50 × 50	158.15s	0.0616	3.080

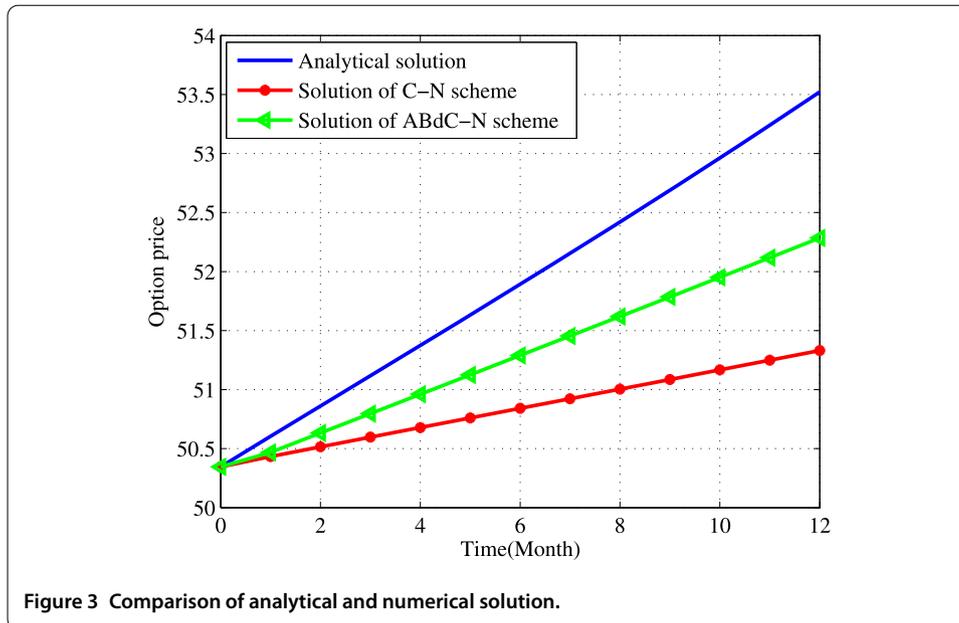
**Table 3 Error analysis of the ABdC-N scheme**

Grid	CPU time	RMSE	RRMSE
20 × 20	1.410s	0.0795	1.590
30 × 30	2.095s	0.0547	1.641
40 × 40	5.649s	0.0417	1.668
50 × 50	18.307s	0.0336	1.680



scheme are near 1.6 for each grid. The RRSMEs of the C-N scheme are near 3.0 for each grid. This shows that the ABdC-N scheme and the C-N scheme have better stability.

Comparing Figure 1 and Figure 2, the solution surface of the ABdC-N scheme is consistent with the C-N scheme, and the two solution surfaces are very smooth. From Figure 3, we see that the ABdC-N scheme is better than the classical C-N scheme in terms of the computation accuracy. The result is consistent with the view which has been put forward by Zhang *et al.* [13]. His option is that the numerical computing result of the alternating segment or the band parallel method usually is better than the original method without splitting processing in terms of computing accuracy.



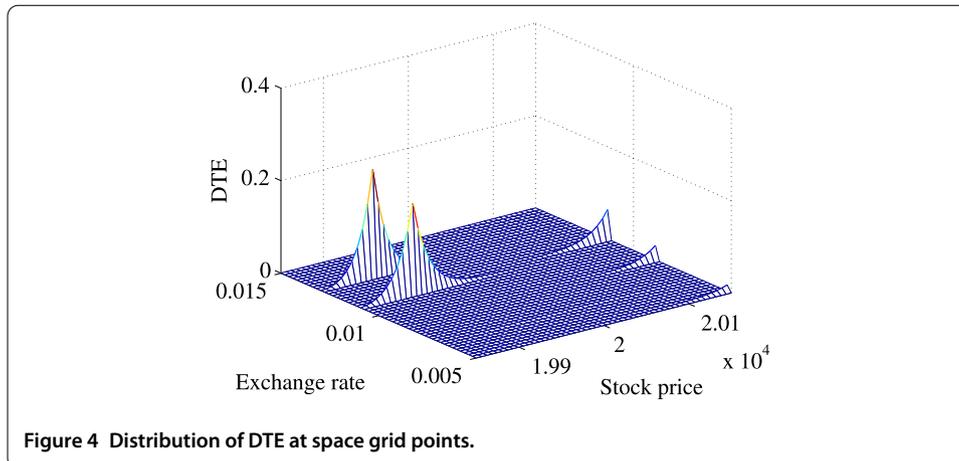
In terms of computation time, from Tables 1, 2, and 3, the computing time of the ABdC-N scheme of the quanto option has a big advantage compared with the classical C-N scheme (except grid  $20 \times 20$ ). When the number of grid points is smaller, the impact of the data communication on the cycle can greatly reduce the computation efficiency. So when the grid is  $20 \times 20$ , the CPU time of the C-N scheme is smaller than the ABdC-N scheme. The computing time (CPU time) of the ABdC-N scheme is 28.86%, 14.83%, 11.57% of the C-N scheme for grids  $30 \times 30$ ,  $40 \times 40$ ,  $50 \times 50$ , respectively. When the number of grid points is larger, the advantages of parallel computing of the ABdC-N scheme are obviously superior. The computation time of the ABdC-N scheme can save about 88% compared with the classical C-N scheme for grid  $50 \times 50$ . Therefore, the ABdC-N scheme can be more effective to solve option pricing problems than the classical C-N scheme.

In order to better compare the accuracy of the ABdC-N scheme with the classical C-N scheme, we will analyze the distribution of the difference total energy (DTE) at space grid points for grid  $50 \times 50$ . Taking the numerical solution  $\bar{u}$  of the classical C-N scheme as the control solution, we take the numerical solution  $u$  of the ABdC-N scheme as a perturbation solution. The definition of DTE is as follows [15]:

$$DTE(i, j) = \frac{1}{2} \sum_{k=1}^{nt} (\bar{u}_{i,j}^k - u_{i,j}^k)^2.$$

From Figure 4, the scope of DTE is  $0 \sim 0.25$ . In other words, the numerical solution of the ABdC-N is very close to the C-N scheme. The value of DTE is relatively large in three lines of the y axis (exchange rate). The three lines are inside the boundary points, the ABdC-N scheme alternatively applies the classical explicit scheme and the implicit scheme which are of first order accuracy. Therefore, the value of DTE is relatively large within the three lines.

Comprehensively considering the computing efficiency and the computing accuracy, the ABdC-N scheme can be more effective to solve the quanto option pricing problems.



## 7 Conclusion

An alternating band Crank-Nicolson (ABdC-N) difference scheme for solving the quanto options pricing model (2D B-S equation) has been constructed. The computing accuracy, stability, and convergence of the ABdC-N scheme have been analyzed. The result of the numerical experiments is consistent with the theoretical analysis. The ABdC-N scheme has an ideal computing accuracy and computing efficiency, and it can be more effective to solve the quanto options pricing problems.

### Competing interests

The authors declare that they have no competing interests.

### Authors' contributions

All authors contributed equally and significantly in writing this article. All authors read and approved the final manuscript.

### Acknowledgements

This work is sponsored by the project National Science Foundation of China (No. 11371135, 11271126), the Fundamental Research Funds for the Central Universities (Nos. 13QN30, 2014ZZD10).

Received: 16 June 2015 Accepted: 23 September 2015 Published online: 09 October 2015

### References

1. Kwork, YK: *Mathematical Models of Financial Derivatives*, 2nd edn. The World Book Publishing Company, Beijing (2011)
2. Jiang, LS: *Mathematical Modeling and Methods of Option Pricing*, 2nd edn. Higher Education Press, Beijing (2008) (in Chinese)
3. Li, YQ, Huang, LH: *Quanto Option and Delay Option Pricing Research*. Hunan University Press, Changsha (2011) (in Chinese)
4. Gilli, M, Kellezi, E, Pauletto, G: Solving finite difference schemes arising in trivariate option pricing. *J. Econ. Dyn. Control* **26**(9-10), 1499-1515 (2002)
5. Khaliq, AQM, Voss, DA, Kazmi, K: Adaptive  $\theta$ -methods for pricing American options. *J. Comput. Appl. Math.* **222**(1), 210-227 (2008)
6. Yang, XZ, Zhou, GX: A kind of accelerated AOS difference schemes for dual currency option pricing model. *Int. J. Inf. Syst. Sci.* **7**(2-3), 369-378 (2011)
7. Evans, DJ, Abdullah, ARB: Group explicit methods for parabolic equations. *J. Comput. Math.* **14**, 73-105 (1983)
8. Zhang, BL: Alternating segment explicit-implicit methods for diffusion equation. *J. Numer. Method Comput. Appl.* **4**, 245-253 (1991)
9. Wang, WQ: Difference schemes with intrinsic parallelism for the KdV equation. *Acta Math. Appl. Sin.* **29**(6), 995-1003 (2006) (in Chinese)
10. Sheng, ZQ, Yuan, GW, Hang, XD: Unconditional stability of parallel difference schemes with second order accuracy for parabolic equation. *Appl. Math. Comput.* **184**, 1015-1031 (2007)
11. Yuan, GW, Sheng, ZQ: The unconditional stability of parallel difference with second order convergence for nonlinear parabolic system. *J. Partial Differ. Equ.* **20**, 45-64 (2007)
12. Yang, XZ, Liu, YG, Wang, GH: A study on a new kind of universal difference schemes for solving black-Scholes equation. *Int. J. Inf. Syst. Sci.* **3**(2), 251-260 (2007)
13. Zhang, BL, Gu, TX, Mo, ZY: *Principles and Methods of Numerical Parallel Computation*. National Defence Industry Press, Beijing (1999) (in Chinese)

14. Zhang, SC: Finite Difference Numerical Calculation for Parabolic Equation with Boundary Condition. Science Press, Beijing (2010) (in Chinese)
15. Wu, LF, Yang, XZ, Zhang, F: A kind of difference method with intrinsic parallelism for nonlinear Leland equation. *J. Numer. Methods Comput. Appl.* **35**(1), 69-80 (2014) (in Chinese)

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](http://springeropen.com)

---