Computers and Mathematics with Applications 75 (2018) 2259–2270

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

High order method for Black–Scholes PDE☆

Jinhao Hu, Siqing Gan *

School of Mathematics and Statistics, Central South University, Changsha, Hunan 410083, China

ARTICLE INFO

Article history:
Received 20 July 2017
Received in revised form 28 October 2017
Accepted 3 December 2017
Available online 27 December 2017

Keywords:
Option pricing
Black–Scholes Formula
Compact difference scheme
Backward differentiation formula
Grid refinement method

ABSTRACT

In this paper, the Black–Scholes PDE is solved numerically by using the high order numerical method. Fourth-order central scheme and fourth-order compact scheme in space are performed, respectively. The comparison of these two kinds of difference schemes shows that under the same computational accuracy, the compact scheme has simpler stencil, less computation and higher efficiency. The fourth-order backward differentiation formula (BDF4) in time is then applied. However, the overall convergence order of the scheme is less than $\mathcal{O}(h^4 + \delta^4)$. The reason is, in option pricing, terminal conditions (also called pay-off function) is not able to be differentiated at the strike price and this problem will spread to the initial time, causing a second-order convergence solution. To tackle this problem, in this paper, the grid refinement method is performed, as a result, the overall rate of convergence could revert to fourth-order. The numerical experiments show that the method in this paper has high precision and high efficiency, thus it can be used as a practical guide for option pricing in financial markets.

1. Introduction

When solving Black–Scholes PDE, the Crank–Nicolson difference method is a popular approach. In their book [1], Tavella and Randall applied second-order central finite difference method in space, Crank–Nicolson difference method in time, and therefore obtained a numerical method with global second-order convergence. Brian, McCartin, and Labadie [2] suggested that the Crandall–Douglas method can be used to achieve fourth-order accuracy. However, the pollution effect in the vicinity of the strike price is not considered in their paper, so the overall convergence is still only second-order. Oosterlee, Leentvaar and Huang [3] applied fourth-order central finite difference method in space and utilized fourth-order backward differentiation formula (BDF4) in time. Moreover, the grid refinement method in numerical analysis is introduced to eliminate the pollution effect. With coordinate transformation, the overall convergence order is raised to fourth-order. Nevertheless, in their paper, the central finite difference stencil is so large that it decreases computing efficiency. Tangman, Gopaul and Bhuruth [4] introduced the compact finite difference method into this problem. They used fourth-order Jain difference scheme in space, making the original five-diagonal matrix into a three-diagonal matrix, so that greatly improves the computing efficiency. Whereas the Jain difference scheme [5] and Spotz difference scheme [6] mentioned in their paper are too complicated. Shukla and Zhong [7] proposed that the polynomial interpolation method can be applied to the determination of coefficients of compact finite difference schemes, which significantly simplifies the derivation of compact finite difference schemes. Patel and Mehra [8] firstly utilized the polynomial interpolation method to derive the compact scheme for option pricing. However, the desired convergence rate is not obtained in [8]. The reason is probably that, neither did they deal with the pollution effect near the strike price, nor did they correct the errors about the compact difference

☆ This work is supported by NSF of China (Nos. 11571373, 11671405, 91630312).

* Corresponding author.
E-mail addresses: 0803130219@csu.edu.cn (J. Hu), sqgan@csu.edu.cn (S. Gan).

https://doi.org/10.1016/j.camwa.2017.12.002
0898-1221/

Table 1
Notations.

Symbol	Meaning	Explanation
S	Price of stock	The price of the corresponding stock in the contract
X	Strike price	The prescribed fixed price to buy/sell the stock
T	Mature date	The time an option can be executed, here T is expressed in terms of the relative number, i.e., If the mature date is 100 days, then $T = \frac{100}{365}$
r	Risk-free interest rate	The risk-free interest rate must be in the form of continuous compound interest, the formula for the conversion of the interest rate to the annual interest rate r_0 is $r = \ln(1 + r_0)$
V	Value	The value of the option

formula in [7]. In recent years, nonlinear Black–Scholes models have been used to build transaction costs, market liquidity, or volatility uncertainty into the celebrated Black–Scholes concept. Guo and Wang [9,10] proposed an unconditionally stable method for nonlinear Black–Scholes equation with transaction costs and the nonlinear Black–Scholes equation in illiquid markets. Edeki, Owoloko, and Ugbebor [11] modified Black–Scholes Model under the situation that the volatility is a non-constant function.

In this paper a highly accurate and efficient numerical method to solve Black–Scholes PDE is presented. It could be summarized as follows.

(a) A fourth-order compact scheme is employed to discrete Black–Scholes PDE in space. Polynomial interpolation is used to derive high-order compact schemes. Compared to the conventional method of undetermined coefficients, the polynomial interpolation is more efficient in finding coefficients of the scheme. BDF4 is then applied to discrete the resulting ordinary differential equations, which is expected to match the precision both in space and time.

(b) Because the initial condition (i.e., the pay-off function of the option) is non-differentiable at the strike price, the overall convergence order of the method mentioned above is less than $\mathcal{O}(h^4+\delta^4)$). To deal with this issue, we use the grid refinement method to distribute more points in the vicinity of the bad point (cusp or singularity), so that the value of the derivative around the non-differentiable point can be approximated by a smaller step size. A coordinate transformation is used to implement the grid refinement.

This paper is organized as follows. In Section 2, some basic knowledge for option pricing is presented. In Sections 3 and 4, different kinds of finite difference methods are given. In Section 5, the grid refinement theory is introduced to solve the pollution effect in option pricing.

2. Options and Black–Scholes formula

2.1. The definition of options

An option, in finance, is a contract that gives its owner the right (but not the obligation) to buy or sell a prescribed amount of particular asset from the writer of the option for a prescribed fixed price (called the strike price) on or before the certain date (called maturity date). For various purposes, there are many kinds of options, such as, vanilla options (European call or put option, American call or put option), Asian option, Bermudan option, exotic option, look-back option, barrier option, etc. Options that can be exercised only on the maturity date are called European option, while options that can be exercised at any time up to the maturity date are called American option. If the option is to buy the asset it is a call option, if to sell the asset it is a put option.

Table 1 explains the symbol used in this paper:

2.2. Black–Scholes formula

In 1973, Black and Scholes derived the well-known Black–Scholes Formula based on the next eight assumptions [12]:

(1) The asset price S follows geometric Brownian motion, i.e. S satisfies the following stochastic differential equation:

$$dS = \mu S dt + \sigma S dW,$$

where μ is the drift rate, σ is the volatility and dW is the increment of a standard Brownian motion.

(2) The risk-free interest rate r is a constant during the expiration date.

(3) Investors can borrow and lend at risk-free rates.

(4) The stock does not pay dividends during the expiration date.

(5) The stock market is frictionless, i.e., there is no tax or transaction cost, and all securities are divisible.

(6) There is no risk-free arbitrage opportunity in the market.

(7) Securities trading is continuous.

(8) The option is a European option, which is non-executable until the expiration date.

Based on the eight assumptions above, the Black–Scholes PDE can be written as:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0. \tag{2.1}$$

According to the definition of option and a little knowledge of stocks, it is not difficult to deduce the boundary condition and the terminal condition of Black–Scholes PDE. For European call options, we have

$$V(0,t)=0, \tag{2.2}$$

$$\lim_{S\to+\infty} V(S,t)=S. \tag{2.3}$$

On the contrary, for European put option, we have

$$V(0,t)=Xe^{-r(T-t)}, \tag{2.4}$$

$$\lim_{S\to+\infty} V(S,t)=0. \tag{2.5}$$

By the definition of the European option, it is clear that at expiry date T, the value of European option $V(S,T)$ (also called as pay-off function) is given by

$$V(S,T)=\begin{cases}\max(S-X,0), & \text{for an European call option,}\\ \max(X-S,0), & \text{for an European put option.}\end{cases} \tag{2.6}$$

Unlike many other PDEs, the solution $V(S,t)$ of the Black–Scholes PDE (2.1) with the above final condition (2.6) can be written as

$$V(S,t)=\begin{cases}SN(d_1)-XN(d_2)e^{-r(T-t)}, & \text{for an European call option,}\\ XN(-d_2)e^{-r(T-t)}-SN(-d_1), & \text{for an European put option,}\end{cases} \tag{2.7}$$

where

$$N(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{y^2}{2}}\,dy, \tag{2.8}$$

$$d_1=\frac{\log(\frac{S}{X})+(r+\frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}, \tag{2.9}$$

$$d_2=\frac{\log(\frac{S}{X})+(r-\frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}. \tag{2.10}$$

In this paper, we get the numerical solution by using finite difference method. Meanwhile, we take the theoretical solution, which is calculated by formula (2.7), as the benchmark. Comparing numerical solution and theoretical solution, the calculation error is obtained.

3. Space discretization

3.1. Fourth-order central finite difference

Considering a set of $N-1$ points, the standard fourth-order central finite difference formula is

$$\frac{\partial V_i}{\partial S}=\frac{-V_{i+2}+8V_{i+1}-8V_{i-1}+V_{i-2}}{12h},\quad 2\le i\le N-2, \tag{3.1}$$

$$\frac{\partial^2 V_i}{\partial S^2}=\frac{-V_{i+2}+16V_{i+1}-30V_i+16V_{i-1}-V_{i-2}}{12h^2},\quad 2\le i\le N-2. \tag{3.2}$$

The formula above cannot provide approximation for the first point V_1 and the last point V_{N-1}. To get the approximation formula on them, we can use backward difference method or unilateral derivative method. The result can be written as follows.

$$\frac{\partial V_1}{\partial S}=\frac{-3V_0-10V_1+18V_2-6V_3+V_4}{12h}, \tag{3.3}$$

$$\frac{\partial^2 V_1}{\partial S^2}=\frac{10V_0-15V_1-4V_2+14V_3-6V_4+V_5}{12h^2}. \tag{3.4}$$

In the above formula, V_0 represents the value of options when the value of its corresponding stock is 0. For European put options, according to Section 2, we have $V(0,t)=Xe^{-r(T-t)}$. The same approach can be applied on the last point and we can obtain the approximation formula on the last point:

$$\frac{\partial V_{N-1}}{\partial S}=\frac{-3V_N-10V_{N-1}+18V_{N-2}-6V_{N-3}+V_{N-4}}{12h}, \tag{3.5}$$

$$\frac{\partial^2 V_{N-1}}{\partial S^2}=\frac{10V_N-15V_{N-1}-4V_{N-2}+14V_{N-3}-6V_{N-4}+V_{N-5}}{12h^2}. \tag{3.6}$$

In the above formula, V_N represents the value of options when the value of its corresponding stock reaches its highest point, that is, $S_{\max}$. For European put options, according to Section 2, we have $\lim_{S\to+\infty} V(S,t)=0$.

If we write the difference formula above in the form of matrices, we have

$$\frac{\partial V}{\partial S} = AV + b_1, \tag{3.7}$$

$$\frac{\partial^2 V}{\partial S^2} = BV + b_2, \tag{3.8}$$

where $V = (V_1 V_2 \cdots V_{N-1})^T$,

$$A = \begin{bmatrix} \frac{-10}{12h} & \frac{18}{12h} & \frac{-6}{12h} & \frac{1}{12h} & & \\ \frac{-8}{12h} & 0 & \frac{8}{12h} & \frac{-1}{12h} & & \\ \frac{1}{12h} & \frac{-8}{12h} & 0 & \frac{8}{12h} & \ddots & \\ & & \ddots & \ddots & \ddots & \\ & & \frac{1}{12h} & \ddots & 0 & \frac{8}{12h} \\ & & \frac{1}{12h} & \frac{-6}{12h} & \frac{18}{12h} & \frac{-10}{12h} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{-15}{12h^2} & \frac{-4}{12h^2} & \frac{14}{12h^2} & \frac{-6}{12h^2} & \frac{1}{12h^2} & \\ \frac{16}{12h^2} & \frac{-30}{12h^2} & \frac{16}{12h^2} & \frac{-1}{12h^2} & & \\ \frac{-1}{12h^2} & \frac{16}{12h^2} & \frac{-30}{12h^2} & \frac{16}{12h^2} & \ddots & \\ & & \ddots & \ddots & \ddots & \\ & & \frac{-1}{12h^2} & \ddots & \frac{-30}{12h^2} & \frac{16}{12h^2} \\ & \frac{1}{12h^2} & \frac{-6}{12h^2} & \frac{14}{12h^2} & \frac{-4}{12h^2} & \frac{-15}{12h^2} \end{bmatrix}, \tag{3.9}$$

$$b_1 = \left(\frac{-3V_0}{12h} \quad \frac{V_0}{12h} \quad 0 \quad \cdots \quad \frac{-V_N}{12h} \quad \frac{-3V_N}{12h} \right)^T, b_2 = \left(\frac{10V_0}{12h^2} \quad \frac{-V_0}{12h^2} \quad 0 \quad \cdots \quad \frac{-V_N}{12h^2} \quad \frac{10V_N}{12h^2} \right)^T. \tag{3.10}$$

Now we can rewrite the Black–Scholes PDE in the form of linear system

$$\frac{dV}{dt} = -PV - Q, \tag{3.11}$$

where $P = \frac{1}{2}\sigma^2 S^2 B + rSA - rI$, $Q = \frac{1}{2}\sigma^2 S^2 b_2 + rSb_1$ and I is an identity matrix.

3.2. Fourth-order compact finite difference method

From the difference formula in Section 3.1, we can clearly see that the fourth-order central difference method requires the function value on five points nearby at least. When it is written in the form of matrices, the matrices A and B in formula (3.9) are both five-diagonal matrices and the stencil is quite huge. Therefore, a natural idea is to approximate the partial derivative value in a simpler way. Actually, it is the basic idea of high order compact finite difference method, i.e., when approximate the partial derivative value, we not only use function values on nearby points, but also use the derivative values on these points.

Once we have decided our scheme, the only problem is to determine the coefficients. In the past, the method of undetermined coefficient is mainly used, but this method is of low efficiency. In 2005, Shukla and Zhong [7] put forward that polynomial interpolation could be used in determining the coefficients of compact finite difference method, which can greatly improve computing efficiency. We can give a brief presentation about the polynomial interpolation as follows.

For approximating first-order partial derivatives, we consider a set of n points I_n on which values of the function and its first derivative have been specified and another set of m points I_m on which only function values have been specified. The independent variable representing the points is x_i, i being the index of the node and the function values are given by $u_i = u(x_i)$ and the first derivative is given by $u_i' = u'(x_i)$. Then a polynomial $u(x)$ of degree $\leq 2n+m-1$ that assumes the values $u_i = u(x_i)$, $i \in I_n \cup I_m$ and $u_i' = u'(x_i)$, $i \in I_n$ is of the form

$$u(x) = \sum_{i\in I_n} u_i p_i(x) + \sum_{i\in I_n} u_i' q_i(x) + \sum_{i\in I_m} u_i r_i(x), \tag{3.12}$$

where the polynomials $p_i(x)$, $q_i(x)$ and $r_i(x)$ satisfy the following conditions:

$$p_i(x_j) = \delta_{ij}, \quad \forall i \in I_n, \forall j \in I_n \cup I_m, \quad p_i'(x_j) = 0, \quad \forall i \in I_n, \forall j \in I_n, \tag{3.13}$$

$$q_i(x_j) = 0, \quad \forall i \in I_n, \forall j \in I_n \cup I_m, \quad q_i'(x_j) = \delta_{ij}, \quad \forall i \in I_n, \forall j \in I_n, \tag{3.14}$$

$$r_i(x_j) = \delta_{ij}, \quad \forall i \in I_m, \forall j \in I_n \cup I_m, \quad r_i'(x_j) = 0, \quad \forall i \in I_m, \forall j \in I_n, \tag{3.15}$$

Table 2
Fourth-order compact schemes for first derivative.

Index	I_n, I_m	Difference formula ($x_i = x_1 + h(i-1)$)
1	$\{3,4\}, \{1,2\}$	$u'_1 + 3u'_2 = -\frac{17}{6h}u_1 + \frac{3}{2h}u_2 + \frac{3}{2h}u_3 - \frac{1}{6h}u_4$
$2, 3, \ldots, N-1$	$\{i-1, i+1\}, \{i\}$	$\frac{1}{4}u'_{i-1} + u'_i + \frac{1}{4}u'_{i+1} = \frac{3}{4h}(u_{i+1} - u_{i-1})$
N	$\{N-2, N-3\}, \{N, N-1\}$	$u'_N + 3u'_{N-1} = \frac{17}{6h}u_N - \frac{3}{2h}u_{N-1} - \frac{3}{2h}u_{N-2} + \frac{1}{6h}u_{N-3}$

Table 3
Fourth-order compact schemes for second derivative.

Index	I_n, I_m	Difference formula ($x_i = x_1 + h(i-1)$)
1	$\{3,4,5\}, \{1,2\}$	$u''_1 + 10u''_2 = \frac{145}{12h^2}u_1 - \frac{76}{3h^2}u_2 + \frac{29}{2h^2}u_3 - \frac{4}{3h^2}u_4 + \frac{1}{12h^2}u_5$
$2, 3, \ldots, N-1$	$\{i-1, i+1\}, \{i\}$	$\frac{1}{10}u''_{i-1} + u''_i + \frac{1}{10}u''_{i+1} = \frac{6}{5h^2}(u_{i+1} - u_{i-1}) - \frac{12}{5h^2}u_i$
N	$\{N-2, N-3, N-4\}, \{N, N-1\}$	$u''_N + 10u''_{N-1} = \frac{145}{12h^2}u_N - \frac{76}{3h^2}u_{N-1} + \frac{29}{2h^2}u_{N-2} - \frac{4}{3h^2}u_{N-3} + \frac{1}{12h^2}u_{N-4}$

where δ_{ij} is the Kronecker delta. The conditions (3.13), (3.14) and (3.15) suggest following form of the polynomials

$$p_i(x) = \frac{\prod_m(x)}{\prod_m(x_i)}\left(l_i^n(x)\right)^2\left[1 - \left[2l_i'^n(x_i) + \frac{\prod'_m(x_i)}{\prod_m(x_i)}\right](x - x_i)\right], \forall i \in I_n, \tag{3.16}$$

where $l_i^n(x)$ is the Lagrange polynomials on I_n and $\prod_m(x)$ are defined as $\prod_m(x) = \prod_{j\in I_m}(x - x_j)$.

A similar analysis for $q_i(x)$ and $r_i(x)$ gives

$$q_i(x) = \frac{(x - x_i)\prod_m(x)}{\prod_m(x_i)}(l_i^n(x))^2, \ \forall i \in I_n, \tag{3.17}$$

$$r_i(x) = \left[\frac{\prod_n(x)}{\prod_n(x_i)}\right]^2 l_i^m(x), \ \forall i \in I_m, \tag{3.18}$$

where $l_i^m(x)$ is the Lagrange polynomials on I_m and $\prod_n(x) = \prod_{j\in I_n}(x - x_j)$.

The same method can also be applied to derive the approximation for second-order partial derivatives.

Using proper choice of sets of points, I_m and I_n, arbitrary order scheme can be constructed. Also if I_m has m points and I_n has n points then the order of the scheme will be $2n + m - 1$. For example, fourth-order accurate first derivative tridiagonal compact schemes for interior and boundary points for a uniform grid with the distribution of nodes given by $x_i = x_1 + h(i-1)$, $i = 1, 2, \ldots, N$ is presented in Table 2 along with the choice of sets I_n and I_m needed to derive them.

It is worthwhile to point out that the fourth-order compact schemes for second derivative on the first point and the last point in the original paper [7] are wrong, it is easy to verify that the formula in the original paper is of third-order rather than fourth-order. The formulas in Table 3 have been already adjusted.

If we use D to denote the first derivative operator and D^2 to denote the second derivative operator, the results in Tables 2 and 3 can be used in the form of matrices.

$$D = F^{-1}G, \tag{3.19}$$

$$D^2 = U^{-1}W, \tag{3.20}$$

where matrices F, G, U, W are

$$F = \begin{bmatrix} 1 & 3 & 0 & & & \\ \frac{1}{4} & 1 & \frac{1}{4} & & & \\ & \frac{1}{4} & 1 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & 1 & \frac{1}{4} \\ & & & & 3 & 1 \end{bmatrix}, G = \begin{bmatrix} \frac{-17}{6h} & \frac{3}{2h} & \frac{3}{2h} & \frac{-1}{6h} & & & \\ \frac{-3}{4h} & 0 & \frac{3}{4h} & & & & \\ & \frac{-3}{4h} & 0 & \frac{3}{4h} & & & \\ & & \frac{-3}{4h} & 0 & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & 0 & \frac{3}{4h} \\ & & & \frac{1}{6h} & \frac{-3}{2h} & \frac{-3}{2h} & \frac{17}{6h} \end{bmatrix}, \tag{3.21}$$

$$U=\begin{bmatrix} 1 & 10 & 0 & & & \\ \frac{1}{10} & 1 & \frac{1}{10} & & & \\ & \frac{1}{10} & 1 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & 1 & \frac{1}{10} \\ & & & & 10 & 1 \end{bmatrix}, W=\begin{bmatrix} \frac{145}{12h^2} & \frac{-76}{3h^2} & \frac{29}{2h^2} & \frac{-4}{3h^2} & \frac{1}{12h^2} & & & \\ \frac{6}{5h^2} & \frac{-12}{5h^2} & \frac{6}{5h^2} & & & & & \\ & \frac{6}{5h^2} & \frac{-12}{5h^2} & \frac{6}{5h^2} & & & & \\ & & \frac{6}{5h^2} & \frac{-12}{5h^2} & \frac{6}{5h^2} & & & \\ & & & \frac{6}{5h^2} & \frac{-12}{5h^2} & \ddots & & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & \ddots & \frac{-12}{5h^2} & \frac{6}{5h^2} \\ & & & \frac{1}{12h^2} & \frac{-4}{3h^2} & \frac{29}{2h^2} & \frac{-76}{3h^2} & \frac{145}{12h^2} \end{bmatrix}. \tag{3.22}$$

According to compact finite difference formula, the original Black–Scholes PDE can be rewritten in the form of matrices as follows.

$$\frac{\partial V}{\partial t}=-LV, \tag{3.23}$$

where $L=\frac{1}{2}\sigma^2S^2D^2+rSD-rI$, I is an identical matrix.

Comparing the fourth-order central finite difference formula in Section 3.1 and the fourth-order compact finite difference formula in Section 3.2, we can easily find that under the same accuracy, compact finite difference formula has at least two advantages. Firstly, it has simpler computing stencil and secondly, there is no need to deal with the boundary conditions.

4. Time discretization

At the beginning of this section, we firstly make a simple transformation of time, letting $\tau=T-t$ and substituting t into the original Black–Scholes PDE, so that we can convert the original terminal condition to the initial condition. Considering that the general nature of the symbol, in the following we still use t to represent τ. The transformed Black–Scholes PDE is

$$\frac{\partial V}{\partial t}=\frac{1}{2}\sigma^2S^2\frac{\partial^2 V}{\partial S^2}+rS\frac{\partial V}{\partial S}-rV. \tag{4.1}$$

After the transformation, for the fourth-order compact finite difference scheme, the matrix form is

$$\frac{\partial V}{\partial t}=LV. \tag{4.2}$$

Now we can apply our difference method in time. From now on, we use $V^{(j)}$ to denote the value of the option when time is at the jth step, that is, $t=j\delta$.

4.1. Explicit Euler method

The formula for the explicit Euler method is given by

$$\frac{V^{(j+1)}-V^{(j)}}{\delta}=LV^{(j)}, \tag{4.3}$$

which yields

$$V^{(j+1)}=(I+\delta L)V^{(j)}. \tag{4.4}$$

That is, $V^{(j+1)}$ can be recursively calculated by $V^{(j)}$, so this method is also called explicit difference method or forward difference method. According to the above equation we can see that the order of the global truncation error in time is $O(\delta)$. It can be proved that the explicit Euler method is conditionally stable.

4.2. Implicit Euler method

In order to obtain an unconditionally stable method, an implicit difference method can be constructed by using backward differentiation formula. For the above linear system, the formula for the implicit Euler method is given by

$$\frac{V^{(j)}-V^{(j-1)}}{\delta}=LV^{(j)}, \tag{4.5}$$

which yields

$$(I-\delta L)V^{(j)}=V^{(j-1)}. \tag{4.6}$$

That is, the solution of $V^{(j)}$ requires the solution of linear equations or the iterative method to solve the linear equation, so this method is also called an implicit difference method or a backward difference method. According to the above equation we can see that the order of the global truncation error of the method is also $O(\delta)$. It can be proved that the implicit Euler method is globally stable.

4.3. *Crank–Nicolson method*

The weakness of the explicit Euler method is that the order of the global truncation error in time is $O(\delta)$, which requires that the time step is much smaller than the spatial step size. Obviously, a suitable way is to make the global truncation error order in time $O(\delta^2)$. For this reason, an effective method is to average the forward difference method and the backward difference method. That is the basic idea of Crank–Nicolson method, this method can be written in the following form

$$\left(I-\frac{\delta L}{2}\right)V^{(j+1)}=\left(I+\frac{\delta L}{2}\right)V^{(j)}. \tag{4.7}$$

It can be proved that the Crank–Nicolson method is an unconditionally stable method with the convergence order $O(\delta^2)$.

4.4. *Backward differentiation formula*

The backward differentiation formula (BDF) is a family of implicit methods for the numerical integration of ordinary differential equations. They are linear multistep methods that, for a given function and time, approximate the derivative of that function using information from already computed times, thereby increasing the accuracy of the approximation. These methods are especially used for the solution of stiff differential equations. In actual use, the backward differentiation formula is one of the most efficient methods [13].

In Section 3, we use fourth-order compact finite difference method. To match the same accuracy, we apply BDF4 on time discretization with a view to achieve the global truncation error $O(\delta^4+h^4)$.

For a general linear system

$$\frac{du}{dt}=Au+b, \tag{4.8}$$

the form of BDF4 is

$$\left(\frac{25}{12}I-\delta A\right)u^{(j+1)}=4u^{(j)}-3u^{(j-1)}+\frac{4}{3}u^{(j-2)}-\frac{1}{4}u^{(j-3)}+\delta b^{(j+1)}. \tag{4.9}$$

From the formula above we can clearly see that, if we want to apply BDF4, we have to obtain the first four initial values. Thus, we need to make initialization first. A simple way is to use the Crank–Nicolson method twice to obtain the values of $u^{(1)}$ and $u^{(2)}$ from $u^{(0)}$ and then use the BDF3 to obtain the value of $u^{(3)}$, and finally the value of $u^{(4)}$ is obtained by BDF4 [3]. The general form of BDF3 is given by

$$\left(\frac{11}{6}I-\delta A\right)u^{(j+1)}=3u^{(j)}-\frac{3}{2}u^{(j-1)}+\frac{1}{3}u^{(j-2)}+\delta b^{(j+1)}. \tag{4.10}$$

In addition to BDF4, the fourth-order implicit Gauss–Legendre method and the fourth-order Padé approximation method also can be used to solve the problem in time [14]. The formula of fourth-order implicit Gauss–Legendre method is given by

$$y_l=u^{(j)}+\delta\sum_{m=1}^{2}p_{lm}(Ay_m+b),\ l=1,2, \tag{4.11}$$

$$u^{(j+1)}=u^{(j)}+\delta\sum_{m=1}^{2}w_m(Ay_m+b), \tag{4.12}$$

where

$$P=\begin{bmatrix}\frac{1}{4} & \frac{1}{4}-\frac{1}{6}\sqrt{3}\\ \frac{1}{4}+\frac{1}{6}\sqrt{3} & \frac{1}{4}\end{bmatrix},\ w=\begin{bmatrix}\frac{1}{2} & \frac{1}{2}\end{bmatrix}^T. \tag{4.13}$$

Since $p_{ij}(i\le j)$ is non-zero, the difference formula is implicit. To obtain $u^{(j+1)}$, we have to solve a system of 2 equations with 2 unknown vectors of size $N-1$.

Table 4
Applying fourth-order central finite difference method in space, Crank–Nicolson method and BDF4 in time.

Scheme	Grid	Error	Order
Crank–Nicolson	10×10	0.0952	–
	20×20	0.0158	2.4547
	40×40	0.0042	1.9396
BDF4	10×10	0.0951	–
	20×20	0.0157	2.4547
	40×40	0.0041	1.9569

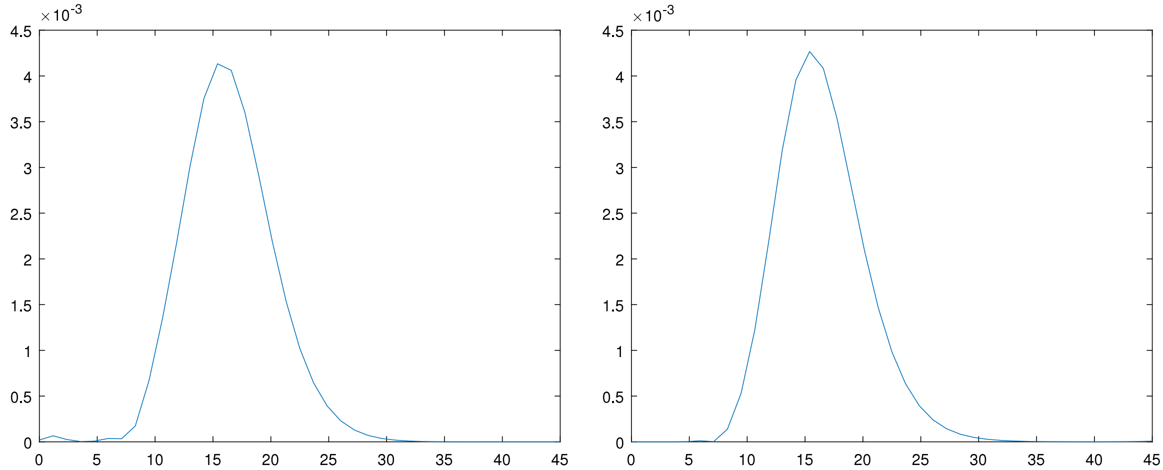

Fig. 1. Errors of the numerical results obtained from different difference methods in space and the BDF4 in time.

The general form of the fourth-order Padé approximation method is given by

$$\begin{aligned} u^{(j+1)} = u^{(j)} &+ \frac{1}{2}\delta\left(Au^{(j)} + b^{(j)} + Au^{(j+1)} + b^{(j+1)}\right) \\ &+ \frac{1}{12}\delta^2\left(A\left(Au^{(j)} + b^{(j)}\right) + \frac{db^{(j)}}{dt}\right) \\ &- \frac{1}{12}\delta^2\left(A\left(Au^{(j+1)} + b^{(j+1)}\right) + \frac{db^{(j+1)}}{dt}\right). \end{aligned} \tag{4.14}$$

From the general form of the implicit Gauss–Legendre method and the Padé approximation method, we can see that these two methods are both single-step methods, but they are of multi-stages, their forms, actually, are more complex, and require a lot of calculation to prepare for each step. For the implicit Gauss–Legendre method, the matrix equation requires inverse matrix calculation, which can consume a lot of CPU time. For the Padé approximation method, not only the matrix inversion operation, but also a large number of matrix multiplication operations. Therefore, although the backward difference scheme is a multi-step method, its format is simple. In the option pricing problem, it is more appropriate to choose the fourth-order backward difference scheme for the time discretization.

Tables 4 and 5 show the numerical results obtained by using different difference methods for the European put option with the parameters $S_{\max} = 45, S_{\min} = 0, \sigma = 0.3, X = 15, r = 0.02$ and $T = 0.5$.

Tables 4 and 5 provide very interesting results. The Crank–Nicolson method is a second-order accuracy method in theory and the results in Tables 4 and 5 show it indeed. However, when BDF4 is used in time, whether the fourth-order center finite difference method or the fourth-order compact finite difference method is used in space, the final order of convergence is still second-order and does not rise to the theoretical fourth-order accuracy. To analyze the cause of this problem, it is helpful to plot the error between the numerical solution and the theoretical solution. To get the numerical solution, we firstly make a 40×40 equidistant segmentation in space and time under the parameters $S_{\max} = 45, S_{\min} = 0, \sigma = 0.3, X = 15, r = 0.02, T = 0.5$. Then we apply BDF4 in time, fourth-order central finite difference method and fourth-order compact finite difference method in space, respectively. Finally, we get the theoretical solution by formula (2.7). The error between the numerical solution and the theoretical solution is plotted as follows (The left graph in Fig. 1 shows applying fourth-order central finite difference in space, and the right graph shows the fourth-order compact finite difference.).

As can be seen from Fig. 1, the error results are better on both sides, but suddenly jump around the strike price and reach the maximum value at the strike price. According to the finite difference method theory, the precision is derived from

Table 5
Applying fourth-order compact finite difference method in space, Crank–Nicolson method and BDF4 in time.

Scheme	Grid	Error	Order
Crank–Nicolson	20 × 20	0.0173	–
	40 × 40	0.0043	2.0058
	80 × 80	0.0011	1.9771
BDF4	10 × 10	0.0172	–
	20 × 20	0.0042	2.0237
	40 × 40	0.0011	1.9540

the precision of the derivative approximation, which is derived from the Taylor expansion formula. However, in the option pricing problem, the initial condition (i.e., the pay-off function of the option) is non-differentiable at the strike price. In other words, the fourth-order precision method results in a second-order precision is caused by the initial condition at the strike price. To tackle this problem, a simple idea is to abandon the concept of the classic "isometric" and place more nodes near the strike price so as to give a more accurate approximation of derivatives on it. By doing this, we could achieve the purpose of improving the overall accuracy. This is exactly the basic idea of grid refinement theory.

5. Grid refinement

5.1. Grid refinement theory

The results in Section 4 raise an interesting question, that is, even though we apply fourth-order methods both in space and time, the results show only second-order accuracy. The reason is that in the option pricing problem, the initial condition (i.e., the pay-off function of the option) is non-differentiable at the strike price. To solve this problem, we can apply the grid refinement method. In numerical analysis, the grid refinement method around the cusp or singularity is usually able to effectively improve the overall accuracy. The principle of this method is to distribute more points in the vicinity of the bad point (cusp or singularity), so that the value of the derivative around the non-differentiable point can be approximated by a smaller step size. This can be done by adaptive grid refinement for some regions, based on an error indicator, or by a coordinate transformation, which results in an a-priori grid stretching. The latter is computationally more efficient if the region of interest for refinement is known beforehand [14]. Also, Adaptive grid refinement requires extra computations during the solution process to estimate the error. Thus, in this problem, a coordinate transformation is our best choice. After the coordinate transformation, the equidistant discretization in the new coordinate system can be mapped into the non-equidistant discretization in the original coordinate system, so as to achieve the purpose of distributing more points near the non-differentiable point.

The coordinate transformation used in this paper is proposed by Tavella and Randall in 2000 [1], which can be written as follows:

$$y = \psi(S) = \frac{\sinh^{-1}(\xi(S - X)) - c_1}{c_2 - c_1}, \tag{5.1}$$

where $c_1 = \sinh^{-1}(\xi(S_{\min} - X))$, $c_2 = \sinh^{-1}(\xi(S_{\max} - X))$ are both constants and parameter ξ is called stretching factor. It is not difficult to verify that this transformation maps $S \in [S_{\min}, S_{\max}]$ on original coordinate system into $y \in [0, 1]$ on new coordinate system. The inverse transformation is

$$S = \psi^{-1}(y) = \varphi(y) = \frac{1}{\xi}\sinh(c_2 y + c_1(1 - y)) + X. \tag{5.2}$$

After the equidistant segmentation on [0, 1] on the new coordinate system, the distribution of corresponding points on the original coordinate system is shown in Fig. 2 (blue, red and green histograms represent the 80 × 80, 40 × 40 and 20 × 20 equidistant segmentation on the new coordinate system, respectively. The left figure shows the distribution where stretching factor is 1 and the right figure shows the situation where stretching factor is 12).

It can be seen from Fig. 2 that through transformation (5.2), there are a relatively large number of points near the refined grid (that is, strike price) while there are a relatively few points on both sides. In addition, the larger the stretching factor in the formula (5.2), the more the points will be distributed near the refined grid.

After the coordinate transformation, the coefficients also change and can be obtained by the chain rule

$$\frac{\partial V}{\partial S} = \frac{\partial V}{\partial y}\frac{dy}{dS} = \frac{\partial V}{\partial y}\left(\frac{dS}{dy}\right)^{-1} = \frac{1}{\varphi'(y)}\frac{\partial V}{\partial y}, \tag{5.3}$$

$$\frac{\partial^2 V}{\partial S^2} = \left(\frac{dS}{dy}\right)^{-1}\frac{\partial}{\partial y}\left(\left(\frac{dS}{dy}\right)^{-1}\frac{\partial V}{\partial y}\right) = \frac{1}{(\varphi'(y))^2}\frac{\partial^2 V}{\partial y^2} - \frac{\varphi''(y)}{(\varphi'(y))^3}\frac{\partial V}{\partial y}. \tag{5.4}$$

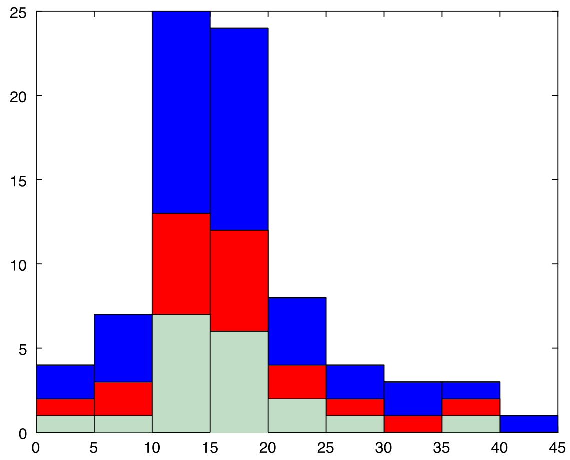

Fig. 2. The distribution of corresponding points on original coordinate system after the equidistant segmentation on the new coordinate system. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Applying fourth-order finite difference in space, BDF4 in time.

Scheme	Grid	Error	Order	Computing time
Fourth-order central finite difference method	20×20	0.0146	–	0.0374 s
	40×40	8.9287e−04	4.0435	0.0444 s
	80×80	6.0106e−05	3.8542	0.0474 s
Fourth-order compact finite difference method	20×20	0.0359	–	0.0308 s
	40×40	0.0024	3.8676	0.0329 s
	80×80	1.5299e−04	3.9607	0.0395 s

Table 7
Applying fourth-order finite difference in space, BDF4 in time.

Scheme	Grid	Error	Order	Computing time
Fourth-order central finite difference method	20×20	0.0185	–	0.0491 s
	40×40	0.0011	4.1010	0.0513 s
	80×80	9.3713e−05	3.4261	0.0585 s
Fourth-order compact Finite Difference Method	20×20	0.0127	–	0.0319 s
	40×40	6.4683e−04	4.4311	0.0369 s
	80×80	3.4865e−05	4.3073	0.0443 s

Thus the transformed Black–Scholes PDE is

$$\frac{\partial V}{\partial t} = \frac{1}{2}\sigma^2 \frac{\varphi(y)^2}{J(y)^2}\frac{\partial^2 V}{\partial y^2} + \left(r\frac{\varphi(y)}{J(y)} - \frac{1}{2}\sigma^2 \frac{\varphi(y)^2}{J(y)^3} H(y) \right) \frac{\partial V}{\partial y} - rV, \tag{5.5}$$

where $J(y) = \varphi'(y)$ and $H(y) = \varphi''(y)$.

5.2. Numerical examples

Consider the following examples. The numerical experiments are performed on Thinkpad X1 Carbon 2016 computer (CPU 2.60 GHz). The numerical results are listed in Tables 6–9 for Examples 1–4, respectively.

Example 1. $S_{\max} = 45, S_{\min} = 0, \sigma = 0.3, X = 15, r = 0.02, T = 0.5$, stretching factor $\xi = 12$.

Compared to Tables 4 and 5, it can be seen from Table 6 that, the overall accuracy is raised to fourth-order after grid refinement. In addition, Table 6 shows that the fourth-order compact scheme saves nearly 25% of the computing time.

Example 2. $S_{\max} = 75, S_{\min} = 0, \sigma = 0.3, X = 15, r = 0.02, T = 0.5$, stretching factor $\xi = 12$.

Example 3. $S_{\max} = 45, S_{\min} = 0, \sigma = 0.3, X = 15, r = 0.02, T = 0.5$, stretching factor $\xi = 9$.

Example 4. $S_{\max} = 15, S_{\min} = 0, \sigma = 0.3, X = 5, r = 0.02, T = 0.5$, stretching factor $\xi = 9$.

Table 8
Applying fourth-order finite difference in space, BDF4 in time.

Scheme	Grid	Error	Order	Computing time
Fourth-order central finite difference method	20×20	0.0113	–	0.0388 s
	40×40	7.0128e−04	4.0141	0.0395 s
	80×80	9.0816e−05	2.7788	0.0548 s
Fourth-order compact finite difference method	20×20	0.0266	–	0.0296 s
	40×40	0.0014	4.3589	0.0386 s
	80×80	1.3194e−04	2.8535	0.0388 s

Table 9
Applying fourth-order finite difference in space, BDF4 in time.

Scheme	Grid	Error	Order	Computing time
Fourth-order central finite difference method	10×10	0.0152	–	0.0359 s
	20×20	0.0011	3.7173	0.0363 s
	40×40	8.1118e−05	3.6825	0.0389 s
Fourth-order compact finite difference method	10×10	0.0271	–	0.0305 s
	20×20	0.0021	3.5923	0.0308 s
	40×40	1.9735e−04	3.2621	0.0319 s

From the comparison of Examples 1–3, we can see that the value of $S_{\max}$ and stretching factor ξ affect the result of the final numerical solution. In addition, from Examples 1, 4 and a lot of numerical experiments, it is found that the option value can be accurate to cent after 40×40 equidistant segmentation, which is precise enough in option pricing because in the financial market, the price is only accurate to cent. In other words, the numerical method used in this paper only needs to use a small number of points to achieve the enough accuracy, thus it is very practical. Moreover, from Example 1 to Example 4, we can clearly see that it greatly reduces the computing time when applying compact finite difference method. Hence, it is not only practical, but also very efficient.

6. Conclusions

In this paper, the Black–Scholes PDE is numerically solved by using high-order numerical method for the European put option. Specifically, we use the fourth-order finite difference scheme and the fourth-order compact finite difference scheme in space, respectively. The comparison of the two difference schemes shows that the compact difference scheme has a smaller stencil and higher efficiency under the same calculation accuracy. Moreover, the compact scheme stencil will not be influenced by boundary conditions. Next, we use the BDF4 to discrete the resulting ordinary differential equation to match the precision both in space and time. However, in option pricing, there is a phenomenon that the final condition (the pay-off function of the option) cannot be differentiated at the strike price, which causes the problem that we can only get the second-order solution even if we apply the fourth-order method. To tackle this problem, we use the grid refinement method to distribute more points near strike price. By doing this, we use a smaller step size to approximate the derivative near strike price and greatly improve the overall accuracy.

In financial markets, the option price is only accurate to cent. It can be seen from Examples 1 and 4 in Section 5 that after 40×40 equidistant segmentation, the option value we calculate is precise enough. Furthermore, after examining a large number of numerical examples, it is found that the numerical method adopted in this paper only needs a small number of points to achieve the accuracy of practical application. Therefore, from the point of view of accuracy and efficiency, the numerical method adopted in this paper is very practical.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 11571373, 11671405, 91630312).

References

[1] D. Tavella, C. Randall, Pricing Financial Instruments: The Finite Difference Method, Wiley, New York, 2000.
[2] B.J. McCartin, S.M. Labadie, Accurate and efficient pricing of vanilla stock options via the Crandall-Douglas scheme, Appl. Math. Comput. 143 (2003) 39–60.
[3] C.W. Oosterlee, C.C.W. Leentvaar, X.Z. Huang, (2005) Accurate American Option Pricing by Grid Stretching and High Order Finite Differences equation, Working papers, DIAM, Delft University of Technology, the Netherlands.
[4] D.Y. Tangman, A. Gopaul, M. Bhuruth, Numerical pricing of options using high-order compact finite difference schemes, J. Comput. Appl. Math. 218 (2008) 270–280.
[5] M.K. Jain, R.K. Jain, R.K. Mohanty, A fourth-order difference method for the one-dimensional general quasilinear parabolic differential equation, Numer. Methods Partial Differential Equations 6 (1990) 311–319.
[6] W.F. Spotz, G.F. Carey, A high-order compact formulation for the 3D Poisson equation, Numer. Methods Partial Differential Equations 12 (1996) 235–243.

[7] R.K. Shukla, X. Zhong, Derivation of high-order compact finite difference schemes for non-uniform grid using polynomial interpolation, J. Comput. Phys. 204 (2005) 404–429.
[8] Kuldip Singh Patel, Mani Mehra, High-Order Compact Finite Difference Method for Black–Scholes PDE, in: P.N. Agrawal, et al. (Eds.), Mathematical Analysis and Its Applications, 2015, pp. 393–403.
[9] J. Guo, W. Wang, An unconditionally stable, positivity-preserving splitting scheme for nonlinear Black–Scholes equation with transaction costs, Sci. World J. 2014 (1) (2014) 525207. 2014, (2014-5-7).
[10] J. Guo, W. Wang, On the numerical solution of nonlinear option pricing equation in illiquid markets, Comput. Math. Appl. 69 (2) (2015) 117–133.
[11] S.O. Edeki, E.A. Owoloko, O.O. Ugbebor, The modified Black–Scholes model via constant elasticity of variance for stock options valuation, in: Progress in Applied Mathematics in Science & Engineering, AIP Publishing LLC, 2016, pp. 83–90.
[12] F. Black, M.S. Scholes, The Pricing of Options and Corporate Liabilities, J. Polit. Econ. 81 (1973) 637–654.
[13] E. Süli, Endre, D. Mayers, An Introduction To Numerical Analysis, Cambridge University Press, 2003.
[14] C.C.W. Leentvaar, Numerical Solution of the Black–Scholes Equation with a Small Number of Grid Points, Delfty University of Technology, 2003 (Master's thesis).