Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational and Applied Mathematics 206 (2007) 306–321
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

Compact finite difference method for American option pricing

Jichao Zhao*, Matt Davison, Robert M. Corless

Department of Applied Mathematics, University of Western Ontario, Middlesex College, London, Ont., Canada N6A 5B7

Received 14 April 2005; received in revised form 11 July 2006

Abstract

A compact finite difference method is designed to obtain quick and accurate solutions to partial differential equation problems. The problem of pricing an American option can be cast as a partial differential equation. Using the compact finite difference method this problem can be recast as an ordinary differential equation initial value problem. The complicating factor for American options is the existence of an optimal exercise boundary which is jointly determined with the value of the option. In this article we develop three ways of combining compact finite difference methods for American option price on a single asset with methods for dealing with this optimal exercise boundary. *Compact finite difference method one* uses the implicit condition that solutions of the transformed partial differential equation be nonnegative to detect the optimal exercise value. This method is very fast and accurate even when the spatial step size h is large ($h \geqslant 0.1$). *Compact difference method two* must solve an algebraic nonlinear equation obtained by Pantazopoulos (1998) at every time step. This method can obtain second order accuracy for space x and requires a moderate amount of time comparable with that required by the Crank Nicolson projected successive over relaxation method. *Compact finite difference method three* refines the free boundary value by a method developed by Barone-Adesi and Lugano [The saga of the American put, 2003], and this method can obtain high accuracy for space x. The last two of these three methods are convergent, moreover all the three methods work for both short term and long term options. Through comparison with existing popular methods by numerical experiments, our work shows that compact finite difference methods provide an exciting new tool for American option pricing.

MSC: 65; 92; 91

Keywords: Compact finite difference method; Free boundary value; American option pricing; Optimal exercise boundary; Black–Scholes equation

1. Introduction

Financial securities (options, futures and forward contracts) have become essential tools for corporations and investors over the past few decades. Options can be used, for example, to hedge assets and portfolios in order to control the risk due to the movement in stock prices.

The simplest financial option is the *European option* which gives the holder of the option the right to buy or sell an asset at a prescribed price (the *Exercise price E*) and a prescribed date, the *Exercise date T* (in years). If the option is to buy the asset it is a *Call option c*, if to sell the asset it is a *Put option p*. From the definition of the European option, we see that the holder of option has the right without obligation to transact, so the option has some positive value.

* Corresponding author. Tel.: +1 519 4331967.
E-mail addresses: jzhao29@uwo.ca (J. Zhao), mdavison@uwo.ca (M. Davison), rcorless@uwo.ca (R.M. Corless).

0377-0427/$ - see front matter
doi:10.1016/j.cam.2006.07.006

That gives rise to the option pricing problem. We suppose that stock price S satisfies a geometric Brownian motion (GBM) given by the stochastic differential equation (SDE):

$$\mathrm{d}S(t) = (\mu - D)S(t)\,\mathrm{d}t + \sigma S(t)\,\mathrm{d}Z(t),$$

where μ is the risk-free interest rate, Z is a standard Wiener process, D is the dividend yield of the asset and σ stands for the volatility in return. In addition we need to specify the risk free interest rate r. The value of a European put option $p(S,t)$ is a function of underlying asset price S and time t, and satisfies the celebrated Black–Scholes partial differential equation (PDE):

$$p_t + \frac{\sigma^2}{2}S^2 p_{SS} + (r - D)Sp_S - rp = 0, \quad S > 0, \ \ t \in (0, T], \tag{1.1}$$

its final condition is

$$p(S,T) = \max(E - S, 0), \quad S > 0, \tag{1.2}$$

and boundary conditions are as follows:

$$p(S,t) \sim E\mathrm{e}^{-r(T-t)} \quad \text{as } S \to 0^+, \tag{1.3}$$

$$\lim_{S\to\infty} p(S,t) = 0 \quad \text{as } S \to +\infty. \tag{1.4}$$

Here T is the duration (in years) of the option contract, and E is the exercise price. The exact, explicit solution of the European put option problem (1.1)–(1.4) is well known [39]. In this paper we just deal with put option pricing as call option prices can be obtained in a similar way.

Most options traded around the world are *American options* which, unlike European options, can be exercised not just at expiry but at any time during the life of the option. American options are more complicated to price. We know that American put option $P(S,t)$ must depend on underlying asset price S and time t. Its value is also determined from the Black–Scholes equation, but with a different set of boundary conditions:

$$P_t + \frac{\sigma^2}{2}S^2 P_{SS} + (r - D)SP_S - rP = 0, \quad S \in [S^*(t), \infty), \ \ t \in (0, T], \tag{1.5}$$

and the final condition is

$$P(S,T) = \max(E - S, 0), \quad S \in [S^*(T), \infty), \tag{1.6}$$

where $S^*(t)$ stands for the free boundary value (the optimal exercise boundary) at the time t and it satisfies

$$S^*(T) = \max\left(E, \frac{rE}{D}\right), \tag{1.7}$$

$$P(S^*(t), t) = E - S^*(t), \quad P_S(S^*(t), t) = -1, \quad t \in (0, T], \tag{1.8}$$

and boundary conditions are as follows:

$$P(S,t) \sim E\mathrm{e}^{-r(T-t)} \quad \text{as } S \to 0^+, \tag{1.9}$$

$$\lim_{S\to\infty} P(S,t) = 0 \quad \text{when } S \to +\infty. \tag{1.10}$$

For an American put option, we know that

$$P_{\text{complete}} = \begin{cases} P(S,t), & S \in [S^*(t), +\infty), \\ \max(E - S, 0), & S \in [0, S^*(t)). \end{cases} \tag{1.11}$$

After 30 years of study, five dominant methods for dealing with American options have emerged. Brennan and Schwartz [7], and Courtadon [11] used finite difference methods for option valuation. Cox et al. [12] first gave the binomial tree method for option pricing. Since then many other versions of binomial parameters have been proposed in the literature, like Jarrow and Rudd [21] and Hull and White [24], Boyle [6] gave the trinomial model for option pricing which is similar to the binomial method, but approaches an accurate value faster than its binomial counterpart due to the use of a three-pronged path. Geske and Johnson [17], MacMillan [28] and Barone-Adesi and Whaley [4,2,3], developed an accurate analytical approximation method. Kim [25], Jacka [20] and Carr et al. [9] provided integral formulas which express the value of American option is the sum of corresponding European option and integral function of free boundary, then use recursive numerical algorithm to solve for optimal exercise boundary and option price. More recently, Longstaff and Schwartz [27] adapted Monte Carlo simulation methods using least squares techniques to solve American option pricing problem and obtained very good results. There are also many other methods for American option pricing problem, like the method of lines by Meyer and Van der Hoek [29] and Carr and Faguet [8].

In this article, we will give "compact" finite difference methods, which are high-order finite difference schemes, for American option price. Compact finite difference methods trace their origin to the work of Cowell and Crommelin in 1907, Stormer in 1909 or Numerov in 1922, see Refs. [5,18,19,31]. Sometimes, they are also called Padé, Hermitian or Mehrstellen ("Mehrstellenverfahren") methods. In recent years, these methods have generated renewed interest and a variety of techniques have been developed [1,15,40,35,36,16,26,46,10]. Many scholars have applied compact finite difference methods to various applications [37,42,22,32,31,16,41,46,45,23,44,43].

The idea of standard compact finite difference schemes is to use a linear combination of the values of a function at three points (or some other small number) to approximate a linear combination of the values of derivatives of the same function at the same three points (or some other small number) with a high accuracy. A standard compact finite difference formula of a univariate function for second derivatives is given in the following formula:

$$a_{-1} f_{k-1} + a_0 f_k + a_1 f_{k+1} = b_{-1} f''_{k-1} + b_0 f''_k + b_1 f''_{k+1}, \tag{1.12}$$

where a_{-1}, a_0, a_1, b_{-1}, b_0, and b_1 are constant, the values of a function and its derivative are denoted by $f_i = f(x_i)$ and $f''_i = f''(x_i)$, respectively, here $i = k-1, k, k+1$. To yield a fourth order accuracy, we choose

$$\begin{aligned} &a_{-1} = \frac{12}{h^2}, \quad a_0 = -\frac{24}{h^2}, \quad a_1 = \frac{12}{h^2}, \\ &b_{-1} = 1, \quad b_0 = 10, \quad b_1 = 1. \end{aligned} \tag{1.13}$$

So a standard fourth-order compact finite difference formula of a univariate function for second derivatives is given in the following formula:

$$\frac{12}{h^2}(f_{k-1} - 2f_k + f_{k+1}) = f''_{k-1} + 10 f''_k + f''_{k+1}. \tag{1.14}$$

In the following sections we will show this scheme is quite simple and really works. The compact finite difference method is used to convert the Black–Scholes PDE to an ordinary differential equation (ODE). The resulting ODE problem can be solved using excellent built-in ODE solvers in many software packages such as Matlab and Maple. We use three different ways from [30,2] to deal with optimal exercise boundary in this paper. Then through comparison with the existing popular methods described above, we find that compact finite difference method can, under some conditions, be superior.

In the next section, we will demonstrate how to use compact finite difference methods on the European put option pricing problem. In Section 3, we adapt compact finite difference methods for American option pricing problem. Numerical results and comparisons with existing methods are given in Section 4. Conclusions are drawn in Section 5.

2. Compact finite difference method

In this part, we will show how to apply the compact finite difference method on the European put option problem (1.1)–(1.4). It is well known that European option pricing problems (1.1)–(1.4) have an explicit solution, here we only show the idea of how to apply compact finite difference methods to solve them numerically since European option pricing problems are easily understood. In the next section, we will adapt these schemes to American case. Refs. [13,14] give a compact finite difference method to solve the nonlinear Black–Scholes equation, however, their methods are hard to extend to American option pricing problems.

Before discussing the compact finite difference method, we introduce the following transformation [30]:

$$\tau = \sigma^2 (T - t)/2, \tag{2.1}$$

$$x = \ln(S/E) + (k_2 - 1)\tau, \tag{2.2}$$

$$u(x, \tau) = \mathrm{e}^{k_1 \tau}(p(S, t) + S - E)/E \tag{2.3}$$

to simplify the European put option problem, where $k_1 = 2r/\sigma^2$ and $k_2 = 2(r - D)/\sigma^2$. With the aid of this transformation, we can rewrite the European put option pricing problem (1.1)–(1.4) in the following simple form:

$$u_\tau = u_{xx} + g(x, \tau), \tag{2.4}$$

where $x \in (-\infty, +\infty)$, $\tau \in (0, (\sigma^2/2)\, T)$ and

$$g(x, \tau) = \mathrm{e}^{k_1 \tau}((k_1 - k_2)\mathrm{e}^{x - (k_2 - 1)\tau} - k_1).$$

This problem has initial and boundary conditions as follows:

$$u(x, 0) = \max(\mathrm{e}^x - 1, 0), \quad x \in (-\infty, +\infty), \tag{2.5}$$

$$\lim_{x \to +\infty} u(x, \tau) = \mathrm{e}^{k_1 \tau}(\mathrm{e}^{x - (k_2 - 1)\tau} - 1), \tag{2.6}$$

$$\lim_{x \to -\infty} u(x, \tau) = 1 + \mathrm{e}^{k_1 \tau}(\mathrm{e}^{x - (k_2 - 1)\tau} - 1). \tag{2.7}$$

To solve the above problem, we need to truncate space x into a finite domain. Standard probabilistic arguments may be employed to imply that using -2 and 2 as the lower and upper bounds is adequate, and numerical experiments confirm these hypotheses.

Next we explain the compact finite difference method and how to apply this method to the European case. As in the method of lines, we discretize only space, $x_i = ih - 2$ and $u_i = u(x_i)$, where $i = 0, 1, \ldots, (N + 1)$, $x_0 = -2$ and $x_{N+1} = 2$. We can derive the second derivative of $u(\tau)$ with respect to x as follows:

$$\begin{aligned} c_0 u_1''(\tau) + u_2''(\tau) = \frac{1}{12h^2}&((10c_0 - 1)u_0(\tau) - (15c_0 - 1)u_1(\tau) - 2(2c_0 + 15)u_2(\tau) \\ &+ 2(7c_0 + 8)u_3(\tau) - (6c_0 + 1)u_4(\tau) + c_0 u_5(\tau)), \end{aligned} \tag{2.8}$$

where c_0 is a parameter to be decided below.

$$u_{i-1}''(\tau) + 10u_i''(\tau) + u_{i+1}''(\tau) = \frac{12}{h^2}(u_{i-1}(\tau) - 2u_i(\tau) + u_{i+1}(\tau)), \tag{2.9}$$

and Eq. (2.9) is true for $i = 2, \ldots, (N - 1)$. When $i = N$, we have

$$\begin{aligned} u_{N-1}''(\tau) + 10u_N''(\tau) = \frac{1}{12h^2}&(10u_{N-4}(\tau) - 61u_{N-3}(\tau) + 156u_{N-2}(\tau) \\ &- 70u_{N-1}(\tau) - 134u_N(\tau) + 99u_{N+1}(\tau)). \end{aligned} \tag{2.10}$$

All formulae are $\mathrm{O}(h^4)$, and can be obtained using Taylor expansions. Eqs. (2.8)–(2.10) can be written in the following compact matrix form:

$$AU''(\tau) = MU(\tau) + H(\tau), \tag{2.11}$$

where

$$A := \begin{pmatrix} c_0 & 1 & 0 & \cdots & 0 \\ 1 & 10 & 1 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & 1 & 10 & 1 \\ 0 & \cdots & 0 & 1 & 10 \end{pmatrix}_{N\times N} \quad \text{and} \quad U'' := \begin{pmatrix} u_1''(\tau) \\ u_2''(\tau) \\ \vdots \\ u_{N-1}''(\tau) \\ u_N''(\tau) \end{pmatrix},$$

$$M := \frac{12}{h^2}\begin{pmatrix} \frac{(1-15c_0)}{144} & -\frac{2(2c_0+15)}{144} & \frac{2(7c_0+8)}{144} & -\frac{(6c_0+1)}{144} & \frac{c_0}{144} & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & & \vdots \\ 0 & 1 & -2 & 1 & \ddots & \ddots & \ddots & 0 \\ \vdots & & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & \frac{10}{144} & \frac{-61}{144} & \frac{156}{144} & \frac{-70}{144} & \frac{-134}{144} \end{pmatrix},$$

$$H := \frac{1}{12h^2}\begin{pmatrix} (10c_0-1)u_0(\tau) \\ 0 \\ \vdots \\ 0 \\ 99u_{N+1}(\tau) \end{pmatrix}_{N\times N} \quad \text{and} \quad U := \begin{pmatrix} u_1(\tau) \\ u_2(\tau) \\ \vdots \\ u_{N-1}(\tau) \\ u_N(\tau) \end{pmatrix}.$$

Note the above formula we derive is always true for any value of parameter c_0. By carefully choosing the c_0 (we choose $c_0 = 5+2\sqrt{6}$ in [10], the other choice $c_0 = 5-2\sqrt{6}$ leads to an unstable recurrence), we can ensure that matrix A factors exactly into $A = L_0U_0 = LDL^{\mathrm{T}}$, where

$$L_0 := \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ k & 1 & 0 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & k & 1 & 0 \\ 0 & \cdots & 0 & k & 1 \end{pmatrix}, \quad U_0 := \begin{pmatrix} c_0 & 1 & 0 & \cdots & 0 \\ 0 & c_0 & 1 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & 0 & c_0 & 1 \\ 0 & \cdots & 0 & 0 & c_0 \end{pmatrix},$$

with $k = 1/c_0$, and $D = \mathrm{diag}(c_0, c_0, \ldots, c_0)$. Then the solution (use $u(\tau)$ to express $u''(\tau)$) to $AU'' = b(b = MU + H)$ can be solved very efficiently and accurately at a cost of just $\mathrm{O}(n)$ flops:

$$AU'' = (L_0U_0)U'' = L_0(U_0U'') = b, \tag{2.12}$$

by first solving for the vector Y such that

$$L_0Y = b, \tag{2.13}$$

i.e.,

$$y_1 = b_1, \quad y_i = b_i - ky_{i-1}, \quad i = 2, 3, \ldots, N, \tag{2.14}$$

and then solving

$$U_0 U'' = Y, \tag{2.15}$$

i.e.,

$$u_N'' = k y_N, \quad u_i'' = k(-u_{i+1}'' + y_i), \quad i = N-1, N-2, \ldots, 1, \tag{2.16}$$

where

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{pmatrix}.$$

We can obtain the values of U'' very efficiently in term of U by using LU decomposition of the matrix A. This avoids floating-point divisions in the recurrence equation solutions, and thus can be expected to be faster. We can always pre-factor the matrix A into L_0 and U_0 as we have done here, we need not even store matrices L_0, U_0 and A. We see that errors decrease in the above recurrences by the fact that $|L_{0,i,i-1}| = k < 1$, $i = 2, 3, \ldots, N$, and $|U_{0,i,i}| = c_0 > 1$, $i = 1, 2, \ldots, N$.

From the above, we can see that the compact finite difference method for Eq. (2.11) can obtain a high order accuracy, while it only uses the time complexity of N to solve the matrix equations.

We explain the procedure to solve the European option pricing problem with the compact finite difference method:

1. Set initial conditions: $u(0)$ by Eq. (2.5).
2. Suppose we know the values of $u(\tau)$ at time step m, and want to compute for the time step $m+1$. We solve the system of linear equations $AU''(\tau^m) = b(\tau^m)$ and $b(\tau^m) = MU(\tau^m) + H(\tau^m)$ by compact finite difference method, say $U''(\tau^m) = \Phi(U^m)$ at each time step. Vector $H(\tau^m)$ is known since the boundary conditions (2.6) and (2.7) on truncated bounds $[-2, 2]$ are known. Then the matrix form of Eq. (2.4) can be rewritten

   $$U_\tau = \Phi(U^m) + G(\tau^m), \tag{2.17}$$

   here

   $$G(\tau^m) = \begin{pmatrix} g(x_1, \tau^m) \\ g(x_2, \tau^m) \\ \vdots \\ g(x_{N-1}, \tau^m) \\ g(x_N, \tau^m) \end{pmatrix}.$$

   The above is just an ODE with the initial condition U^m. We solve problem (2.17) to get U^{m+1}. Because we observe that the ODE is stiff when τ is very small, we implement the algorithm using the Matlab's powerful ode15s solver [33,34]. ode15s is a variable order solver based on numerical differentiation formulae, especially to solve a stiff differential-algebraic problem. In our implementation, we set RelTol $= 1e-6$, AbsTol $= 1e-6$, MaxStep $= 1e-5$ and stats $=$ off for the ode15s solver in Matlab 7.0.
3. Repeat Step 2 until $\tau^{m+1} = (\sigma^2/2)T$, then recast $u(x, \tau)$ as option price $p(S, t)$ by the transformations (2.1)–(2.3).

The above scheme is unconditionally stable and of high order accuracy $O(h^4)$. In [38], boundary value methods and fourth order compact difference scheme are combined to solve the heat equations. The authors of [38] get the same fourth order compact finite difference formula as we do. They call it "high order compact boundary value method".

3. Compact finite difference method for American option pricing

American option pricing problems (1.5)–(1.10) contain a complicated partial differential equation which makes hard to apply compact finite difference schemes directly. While Pantazopoulos [30] introduced transformations which make the partial differential equation into a simple heat equation. By using his transformations (2.1)–(2.3) as above and adding a new transformation equation for the optimal free boundary

$$X^* = \ln(S^*/E) + (k_2 - 1)\tau, \tag{3.1}$$

Eqs. (1.5)–(1.10) are transformed into the following equations:

$$u_\tau = u_{xx} + g(x, \tau), \tag{3.2}$$

where $x \in (X^*(\tau), +\infty)$, $\tau \in (0, (\sigma^2/2)\, T]$ and

$$g(x, \tau) = \mathrm{e}^{k_1 \tau}((k_1 - k_2)\mathrm{e}^{x-(k_2-1)\tau} - k_1),$$

with initial and boundary conditions:

$$u(x, 0) = \max(\mathrm{e}^x - 1, 0), \tag{3.3}$$

$$X^*(0) = \min(0, \ln(r/D)), \tag{3.4}$$

$$u(X^*(\tau), \tau) = 0, \tag{3.5}$$

$$u_x(X^*(\tau), \tau) = 0, \tag{3.6}$$

$$\lim_{x\to+\infty} u(x, \tau) = \mathrm{e}^{k_1\tau}(\mathrm{e}^{x-(k_2-1)\tau} - 1). \tag{3.7}$$

To solve the American option pricing PDE, we need to decide the free boundary value $X^*(\tau)$ at each time step τ. Depending on how we compute the location of this free boundary, we develop three compact finite difference methods. *Compact finite difference method one* uses an implicit condition that the solutions of transformed PDE are nonnegative to detect the optimal exercise value. This method is very fast and obtains first order accuracy for space x when h is large ($h \geqslant 0.1$). *Compact finite difference method two* needs to solve an algebraic nonlinear equation [30] at every time step. This method can obtain second order accuracy for space x and consumes decent time, so it is comparable with Crank Nicolson projected successive over relaxation (SOR) method. *Compact finite difference method three* refines the free boundary value based on *compact difference method two* by a method developed by Barone-Adesi and Lugano [4,2,3], then use the compact finite difference method. This method is also accurate and is easily parallelized. We all call the ode15s for the three compact finite difference methods in our implementation.

We know that the original American option pricing problem (1.5)–(1.10) has the payoff function $\max(E - S, 0)$, and the fact

$$P(S, t) \geqslant \max(E - S, 0), \quad S > 0, \quad t \in (0, T], \tag{3.8}$$

if $P(S, t) = \max(E - S, 0)$, which means $S < S^*$ and if $P(S, t) > \max(E - S, 0)$, which means $S \geqslant S^*$. The transformations (2.1)–(2.3) can be used to obtain the new payoff function:

$$u(x, \tau) \geqslant \frac{\mathrm{e}^{k_1\tau}}{E} \max(\mathrm{e}^{x-(k_2-1)\tau} - 1, 0), \tag{3.9}$$

where $x \in (-\infty, +\infty)$, $\tau \in (0, (\sigma^2/2)T]$. The next three compact finite difference methods all use this condition to update the new values of $u(x, \tau)$.

3.1. Compact finite difference method one

We develop *compact finite difference method one* for American option pricing problem based on the fact that $u(x, \tau)$ is always positive since the right side of the inequality (3.9) is always non-negative. If $u(x, \tau)$ is negative, it is an indication that this option should be exercised. We use this fact to implicitly detect optimal exercise values during the implementation.

Combining the implicit identification of the free boundary values and the compact finite difference method, we add the following steps as steps 3 and 4 after steps 1 and 2 in Section 2 to solve the American option pricing problem:

3. Detect the location of free boundary value: record the last location of the solution u^{m+1}, say at the ith point u_i^{m+1} such that $u_i^{m+1} \leqslant 0$. Because u^{m+1} is increasing in x, we know $u_k^{m+1} > 0, \forall k > i$. Reset the values $u_k^{m+1} = 0$, $k = 1, \ldots, i$. Then save the free boundary value for time step $m+1$, if $u_i^{m+1} = 0$, then save x_i as the free boundary value; otherwise hold $u_i^{m+1} < 0$, and $u_{i+1}^{m+1} > 0$, we use the zero point of the unique linear equation through points (x_i, u_i^{m+1}) and (x_{i+1}, u_{i+1}^{m+1}) to approximate the free boundary value. Note we use the monotonicity [30] of u_i when we locate free boundary values.

4. Repeat steps 2 and 3 until $\tau^{m+1} = (\sigma^2/2)T$, then back transform $u(x, \tau)$ to option price $P(S, t)$ by the transformations (2.1)–(2.3).

From the algorithm above, we see *compact finite difference method one* uses an implicit condition to detect the optimal exercise value and that this method is very fast. The free boundary values we get by this method can obtain first order accuracy, and the accuracy of option prices is low for stock prices near the free boundary values. This method does not converge for the propagation of errors.

3.2. Compact finite difference method two

From Section 3.1, we see that to improve the accuracy of option price, we need to know the optimal exercise values more accurately. To meet this end, we use a method called explicit front tracking method from [30]. This method to decide free boundary value needs to solve the follow nonlinear equation at every time step:

$$\Phi(p, \tau) = u_{N^m} + \frac{(p_m h)^2}{2} g(D^- + (N^m + p_m)h, \tau) = 0, \tag{3.10}$$

where D^- and D^+ are the lower bound and upper bound of truncation interval, respectively. The free boundary value $X^*(\tau^{m+1}) = D^- + (N^m + p_m)h$ and the number of space step $N^{m+1} = \text{floor}(D^+ - X^*(\tau^{m+1}))$, $m = 1, 2, \ldots$. This nonlinear equation has second order accuracy for space x.

With this we are ready to define the algorithm for *compact finite difference method two* as follows:

1. Set initial conditions: $u(0)$, $X^*(0)$, and $N^0 = \text{floor}(D^+ - X^*(0))$.
2. Suppose we know the values of $u(\tau^m)$, N^m and $X^*(\tau^m)$ at time step m, and want to compute for the time step $m+1$. Solve the nonlinear equation (3.10) by Newton method to get the p_m, then we get the free boundary value for $m+1$ time step by $X^*(\tau^{m+1}) = D^- + (N^m + p_m)h$. Update $N^{m+1} = \text{floor}(D^+ - X^*(\tau^{m+1}))$ and $u_i(\tau^{m+1}) = 0$, $1 \leqslant i \leqslant N^{m+1}$.
3. We just need to solve the subsystem of linear equations (3.2) at time step $m+1$ by compact finite difference method since we already know the values $u_i(\tau^{m+1}) = 0$ when $1 \leqslant i \leqslant N^{m+1}$ in step 2. In the last part of step 3, all vectors contain only the last $(N - N^{m+1})$ entries of original vectors, but to simplify the notation, we still use the same names for them. Solve $AU''(\tau^m) = b(\tau^m)$ and $b(\tau^m) = MU(\tau^m) + H(\tau^m)$, say $U''(\tau^m) = \Phi(U^m)$ at each time step. Then the matrix form of Eq. (3.2) can be rewritten
$$U_\tau = \Phi(U^m) + G(\tau^m), \tag{3.11}$$
again we get an ODE with the initial condition U^m, then we solve the problem (3.11) to get the last $(N - N^{m+1})$ entries of U^{m+1}.
4. Same with step 4 of the algorithm in Section 3.1.

3.3. Compact finite difference method three

The free boundary values obtained from *compact finite difference method two* are still not accurate enough. So we need to find a way to get optimal exercise value more accurately.

Table 1
American put option price ($T = 1$)

x	Stock price S	European option price	Binomial method	Trinomial method	Crank Nicolson PSOR	LS Monte Carlo	Integral equation method	Analytic approx. method	Compact method 1	Compact method 2	Compact method 3	True values
$r = 0.04$ $D = 0.02$												
−0.3	75.9572	24.3973	25.33949	25.32663	25.3265	25.43516	25.3722	25.4509	25.10042	25.32570	25.32739	25.329862
−0.2	83.9457	18.9060	19.49101	19.49863	19.4918	19.61486	19.5288	19.6617	19.34597	19.49193	19.49383	19.496910
−0.1	92.7743	13.9057	14.27957	14.26916	14.2561	14.39809	14.2838	14.4477	14.16375	14.25707	14.25914	14.262648
0.0	102.5315	9.6391	9.87092	9.85271	9.83652	9.96446	9.85631	10.0278	9.78167	9.83789	9.84000	9.843537
0.1	113.3148	6.2552	6.35580	6.37390	6.35927	6.46336	6.3727	6.53401	6.32881	6.36044	6.36241	6.365579
0.2	125.2323	3.7773	3.84473	3.83792	3.82849	3.91709	3.83686	3.97728	3.81244	3.82898	3.83064	3.833369
0.3	138.4031	2.1112	2.14801	2.13808	2.13483	2.17167	2.13937	2.25467	2.12653	2.13452	2.13578	2.137839
$r = 0.02$ $D = 0.04$												
−0.3	79.0571	24.89017	24.88553	24.90369	24.8932	25.50118	24.896	24.9164	24.89236	24.89269	24.89269	24.895250
−0.2	87.3716	19.28794	19.29966	19.29948	19.2861	19.75011	19.2903	19.3055	19.28640	19.28651	19.28648	19.290010
−0.1	96.5605	14.18661	4.21335	14.19473	14.1819	14.53676	14.1875	14.1983	14.18321	14.18322	14.18315	14.187428
0.0	106.716	9.83381	9.82141	9.83571	9.82785	10.13364	9.83412	9.84164	9.82971	9.82968	9.82958	9.834102
0.1	117.939	6.38155	6.39237	6.37552	6.3758	6.61808	6.38165	6.38679	6.37752	6.37748	6.37740	6.381553
0.2	130.343	3.85361	3.86812	3.84058	3.84916	3.97671	3.85364	3.85711	3.85015	3.85014	3.85011	3.853583
0.3	144.051	2.15388	2.13629	2.13713	2.15117	2.21886	2.15389	2.15622	2.15125	2.15126	2.15129	2.153912

Note the parameters for American options: $E = 100$, $\sigma = 0.3$, $\tau = T \cdot \sigma^2/2$. The binomial method is based on time step $\Delta t = 0.01$. The trinomial method is based on time step $\Delta t = 0.01$. The Crank Nicolson projected SOR method is based on space step $h = 0.02$. The least square Monte Carlo method is based on 100, 000 sample paths and time step $\Delta t = 0.005$. Integral method and analytical approximations are based on time step $\Delta t = 0.02$. *Compact finite difference method one and two* are based on space step $h = 0.02$. *Compact finite difference method three* is based on $h = 0.02$ space step for option price and time step $\Delta t = 0.0005$ for the free boundary values using the method of Barone-Adesi and Lugano [2]. The true option values are based on trinomial method using time step $\Delta t = 0.00005$.

Barone-Adesi and Lugano [4,2,3] proposed a method to get a remarkably accurate free boundary by solving the system of the following equations:

$$A = -p + E - S^*, \tag{3.12}$$

$$\gamma = -N(d)S^*/A, \tag{3.13}$$

$$\frac{\sigma^2}{2}\gamma(\gamma - 1) - r - (r - D)\gamma - F = 0, \tag{3.14}$$

with $d = (\ln(S/E) + (r - D + \sigma^2/2)\tau)/(\sigma\sqrt{\tau})$, $F = (\partial p/\partial t)/A$, and $p(S, t)$ is the European put option price with the same parameters with American put. After obtaining the values of A, S^*, and γ from above, then obtain option price by

$$P(S, t) = A(t)(S/S^*)^\gamma \quad \text{for } S \geqslant S^*. \tag{3.15}$$

Using (3.12)–(3.14) can yield accurate free boundary values, but option prices using the formula (3.15) are not very accurate for the approximation in Eqs. (3.12)–(3.15) deteriorates quickly moving away from free boundary values (see our numerical experiments in the next section). So in this method, we combine method (3.12)–(3.14) and the compact finite difference method to obtain a new accurate method for American option pricing. The algorithm for *compact finite difference method three* is almost same as that of compact finite difference method two, except that we use the free boundary values from Eqs. (3.12)–(3.14), instead of Eq. (3.10) in step 2 of the algorithm in Section 3.2.

Nonlinear equation (3.10) depends on values of option price at the last time step, while Eqs. (3.12)–(3.14) are independent of this, which means that we can compute free boundary values in advance or use parallel computing in *compact finite difference method three* to save time.

4. Comparisons of results

In this section, we compare compact finite difference methods with the existing popular methods in option pricing. We first focus on the accuracy issue for space x, not only for option prices, but also for the free boundary values, then look at computational time.

The binomial method we use is from Cox et al. [12], and trinomial method is from Hull's book [24]. For the finite difference method, we use the Crank Nicolson scheme and projected SOR algorithm [39] to obtain second order accuracy for space x. We also implement the integral method of Kim [25], the analytical approximations of Barone-Adesi and Whaley [4,2,3] and the least square Monte Carlo simulation method of Longstaff and Schwartz [27]. Binomial, trinomial and integral equation method all converge, but only with first order of accuracy for space x. Least square Monte Carlo method converges very slowly and the accuracy is only $1/\sqrt{n}$, where n is the number of sample paths. We must point out we are only interested in the accuracy for space x instead of time, so for all methods we choose time step Δt is quite small comparing with space step h so that space errors dominate computational errors.

To compare these methods, we choose the option prices obtained by the trinomial method with $\Delta t = 0.00005$ as the benchmark since we know this method converges. Tables 1 and 2 are the computational results from these various methods.

Table 1 is for the short term option $T = 1$. From the computational results of the table, we can tell that the results by *our compact finite difference method two* and *compact finite difference method three* are closer to "correct" option values than other methods. Table 2 is for the long term option $T = 6$. Our three compact finite difference methods all still work very well, while the Crank Nicolson projected SOR method fails for this case.

We know that Crank Nicolson projected SOR method has second order accuracy for space x and convergence. We determine the accuracy of our compact finite difference method by comparing with the Crank Nicolson projected SOR method and compact finite difference methods since other methods, for example the binomial tree method, are hard

Table 2
American put option price ($T = 6$)

x	Stock price S	European option price	Binomial method	Trinomial method	Crank* Nicolson PSOR	LS* Monte Carlo	Integral equation method	Analytic approx. method	Compact method 1	Compact method 2	Compact method 3	True values
$r = 0.04$ $D = 0.02$												
−0.4	77.8800	26.28780	30.76587	30.68724	46.3654	32.2179	31.2276	31.3954	30.17890	30.75338	30.74339	30.75650
−0.3	86.0708	23.39922	26.97866	26.97421	37.7104	28.5394	27.4241	27.7339	26.52391	27.00252	26.99423	27.00561
−0.2	95.1229	20.61936	23.53280	23.49592	30.7142	25.0031	23.8659	24.298	23.10647	23.50196	23.49518	23.50498
−0.1	105.127	17.97918	20.29196	20.25361	25.0334	21.6594	20.5713	21.1012	19.94109	20.26464	20.25919	20.26757
0.0	116.183	15.50560	17.26721	17.26848	20.3951	18.6213	17.5538	18.1551	17.03865	17.30079	17.29647	17.30356
0.1	128.402	13.22031	14.65062	14.56766	16.5866	15.7984	14.8218	15.4689	14.40649	14.61680	14.61344	14.61936
0.2	141.907	11.13906	12.24709	12.20728	13.4445	13.2624	12.3777	13.0473	12.04753	12.21455	12.21199	12.21686
0.3	156.831	9.27125	10.05998	10.09738	10.8441	10.9940	10.2188	10.8907	9.95974	10.09095	10.08905	10.09297
$r = 0.02$ $D = 0.04$												
−0.6	81.0584	36.3671	37.03809	37.00901	79.2291	46.20171	37.1093	37.559	36.86963	37.00108	36.99784	37.00241
−0.5	89.5834	32.9808	33.49474	33.44970	63.8027	41.91451	33.5653	34.0223	33.37438	33.47467	33.47208	33.47601
−0.4	99.005	29.6394	29.98593	29.97654	51.4668	37.66625	30.0946	30.5495	29.94423	30.01987	30.01780	30.02114
−0.3	109.417	26.3825	26.71774	26.67299	41.602	33.26973	26.733	27.1778	26.61586	26.67222	26.67059	26.67336
−0.2	120.925	23.2483	23.49694	23.48207	33.6944	29.33670	23.5148	23.9432	23.42466	23.46615	23.46487	23.46748
−0.1	133.643	20.2715	20.37046	20.42939	27.3282	25.57870	20.4718	20.8787	20.40314	20.43332	20.43231	20.43479
−0.0	147.698	17.4825	17.63880	17.54145	22.1737	22.05123	17.6312	18.0132	17.57949	17.60117	17.60037	17.60271
0.1	163.232	14.9059	15.02061	14.95917	17.9743	18.80441	15.0148	15.3696	14.97627	14.99167	14.99103	14.99321

Note the parameters for American options: $E = 100$, $\sigma = 0.3$, $\tau = T \cdot \sigma^2/2$. The binomial method is based on time step $\Delta t = 0.06$. The trinomial method is based on time step $\Delta t = 0.06$. The Crank Nicolson projected SOR method is based on space step $h = 0.02$. The least square Monte Carlo method is based on 100, 000 sample paths and time step $\Delta t = 0.03$. Integral method is based on time step $\Delta t = 0.12$. Analytical approximations is based on time step $\Delta t = 0.12$. *Compact finite difference method one and two* are based on space step $h = 0.02$. *Compact finite difference method three* is based on space step $h = 0.02$ for option price and time step $\Delta t = 0.012$. for the free boundary values using the method of Barone-Adesi and Lugano [2]. The true option values are based on trinomial method when taken time step $\Delta t = 0.0003$.

*The method fails to get accurate solution for this case.

Table 3
When $h = 0.2$

x		$h = 0.2$		$\triangle\tau = 0.0001$		
	Stock price S	Crank Nicolson	Compact method 1	Compact method 2	Compact method 3	True values
−0.2	83.9457	19.0604	19.13811	19.26822	19.51910	19.496910
0	102.5315	9.12163	9.35873	9.474996	9.60421	9.843537
0.2	125.2323	3.39863	3.47309	3.46295	3.50414	3.833369
RMS	–	0.081	0.065	0.042	0.013	–

Table 4
When $h = 0.1$

x		$h = 0.1$		$\triangle\tau = 0.0001$		
	Stock price S	Crank Nicolson	Compact method 1	Compact method 2	Compact method 3	True values
−0.3	75.9572	25.2462	25.13063	25.24257	25.26206	25.329862
−0.2	83.9457	19.3749	19.30428	19.39379	19.42878	19.496910
−0.1	92.7743	14.1062	14.07499	14.14775	14.19061	14.262648
0	102.5315	9.67344	9.67773	9.72538	9.76840	9.843537
0.1	113.3148	6.21132	6.23089	6.25590	6.29326	6.365579
0.2	125.2323	3.7184	3.73115	3.74193	3.77034	3.833369
0.3	138.4031	2.07034	2.06590	2.06956	2.08854	2.137839
RMS	–	0.030	0.063	0.029	0.021	–

Table 5
When $h = 0.05$

x		$h = 0.05$		$\triangle\tau = 0.0001$		
	Stock price S	Crank Nicolson	Compact method 1	Compact method 2	Compact method 3	True values
−0.3	75.9572	25.3092	25.1291	25.30609	25.31373	25.329862
−0.2	83.9457	19.4659	19.3296	19.46919	19.47754	19.496910
−0.1	92.7743	14.2229	14.1408	14.23126	14.24079	14.262648
0	102.5315	9.80054	9.75972	9.81129	9.82128	9.843537
0.1	113.3148	6.32658	6.30826	6.33576	6.34505	6.365579
0.2	125.2323	3.80388	3.79488	3.80841	3.81609	3.833369
0.3	138.4031	2.12009	2.11319	2.11915	2.12480	2.137839
RMS	–	0.0074	0.062	0.0079	0.0053	–

to compare with compact finite difference methods for the issue of efficiency. To measure the error, we use root mean squared (RMS) relative error. The RMS error is defined by

$$\mathrm{RMS} = \sqrt{\frac{1}{m}\sum_{k=1}^{m}\left(\frac{\tilde{P}_k - P_k}{P_k}\right)^2},$$

and $\tilde{P}_k$ is the estimated option price we want to compare and P_k is the option price which we take as "accurate" (we use the values obtained by the trinomial method with time step $\Delta t = 0.00005$).

We use the same parameters as used in Table 1, and obtain Tables 3–6:

Table 6
When $h = 0.01$

x	Stock price S	$h = 0.01$ Crank Nicolson	Compact method 1	$\triangle\tau = 0.0001$ Compact method 2	Compact method 3	True values
−0.3	75.9572	25.329	25.1119	25.32869	25.32673	25.329862
−0.2	83.9457	19.4956	19.348	19.49545	19.49362	19.496910
−0.1	92.7743	14.2609	14.1665	14.26101	14.25934	14.262648
0	102.5315	9.84172	9.78475	9.84192	9.84046	9.843537
0.1	113.3148	6.36402	6.33174	6.36420	6.36296	6.365579
0.2	125.2323	3.83209	3.81494	3.83215	3.83116	3.833369
0.3	138.4031	2.13702	2.12844	2.13692	2.13619	2.137839
RMS	–	0.00031	0.066	0.00039	0.001	–

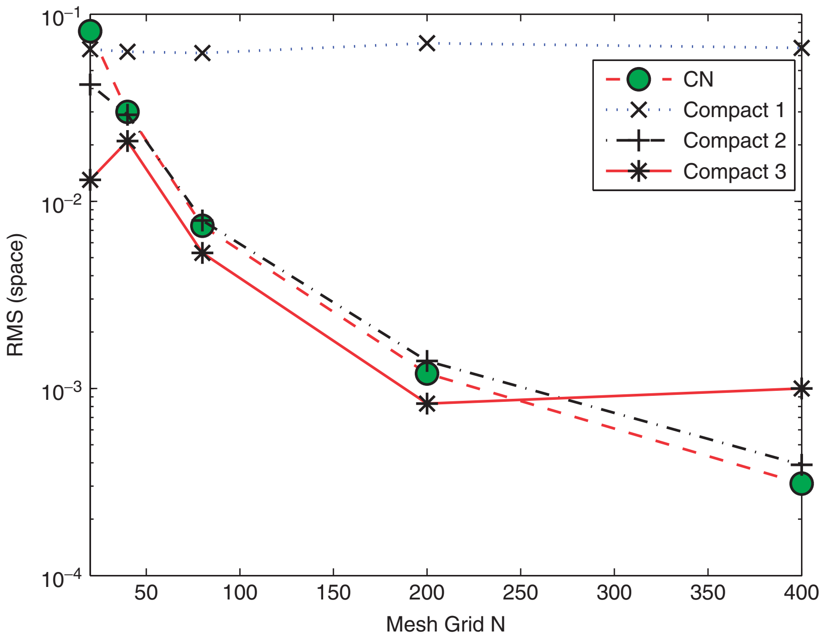

Fig. 1. Computational errors with varied mesh grid N.

From the above results or Fig. 1, we see that with the *compact finite difference method one* can obtain quite good results when space step $h = 4/N$ is not too small ($h \geqslant 0.1$), and N is the mesh grid. While h is too small, say $h = 0.01$, this method does not converge, which can be explained by the poor free boundary values obtained by the implicit method. The *compact finite difference method two* obtains second order accuracy, so it is comparable with Crank Nicolson projected SOR method. The *compact finite difference method three* is more accurate ($O(h^4)$) than the Crank Nicolson projected SOR and *compact finite difference method two* when $h \geqslant 0.02$. The free boundary values we use in the algorithm of the *compact finite difference method three* can only obtain accuracy of $\frac{1}{5000}$ when $h = 0.01$, for the option prices obtained by the *compact finite difference method three* are not more accurate than it.

We can see our *compact finite difference method two and three* converge rapidly, but the error order of compact finite difference methods cannot always obtain $O(h^4)$. The reason is that the free boundary values obtained by our three compact finite difference methods are not accurate enough. To address this point, we can compare the free boundary values by *compact finite difference method one and two* with other methods, the integral method of Kim [25], and the analytic approximations of Barone-Adesi and Elliott [2] (*compact finite difference method three* uses this method to

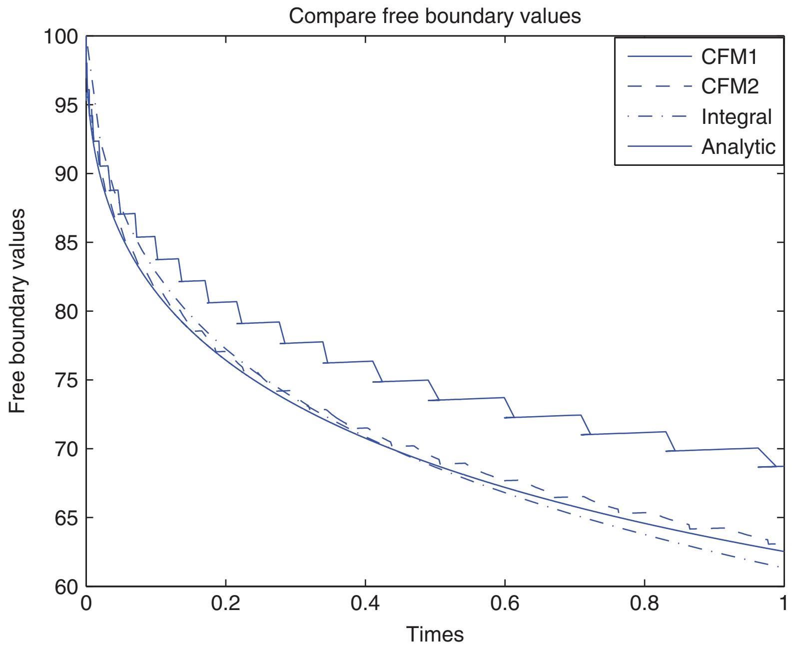

Fig. 2. Free boundary values for American put option.

approximate free boundary values). The "true" values for the free boundaries are based on analytical approximations method when time steps $n = 5000$, other methods are based on $n = 50$. Then we get Fig. 2.

From Fig. 2, we can tell that the accuracy of *compact finite difference method one* for the free boundary values is poor. The integral method of Kim [25] can only get first order accuracy. *Compact finite difference method two* can obtain second order accuracy for free boundary value and this method converges from our experiments. To yield higher accuracy of option prices when step size $h = 0.01$, say $O(h^4)$ for space x, we can use more time steps (larger n) to obtain free boundary values by the method of analytical approximations method, then use compact finite difference method to obtain option prices, i.e., the *compact finite difference method three*.

Speed and accuracy are the two most important issues we should keep in mind when we are talking about option pricing problem. Speed and accuracy indicators are plotted in Figs. 3 and 4. On the two figures, it is shown that *compact finite difference method one* always use fewer times than other methods, and this method can obtain higher accuracy even when space step is large. *Compact finite difference method two* and Crank Nicolson projected SOR method are comparable for the issues of time and accuracy. If we do not count in the times used to compute free boundary values by the method of analytic approximation method, *compact finite difference method three* uses the time close to that spent by the *compact finite difference method one*. From Figs. 3 and 4 we can see that computational time is not monotone for some case, for ode15s is a variable-order multistep solver for stiff problems.

5. Conclusions

It seems that the accuracy and speed of our compact finite difference methods depend heavily on the method we use to obtain the free boundary values. The *compact finite difference method one* can rapidly obtain high accuracy even when h is not too small ($h \geqslant 0.1$). While h is too small, say $h = 0.01$, this method fails to converge. The *compact finite difference method two* can obtain second order accuracy, and works for both short term and long term options, while the Crank Nicolson projected SOR method fails for the long term case. The *compact finite difference method three* is more accurate than Crank Nicolson projected SOR method, but at the cost of more computational time on the free boundary values.

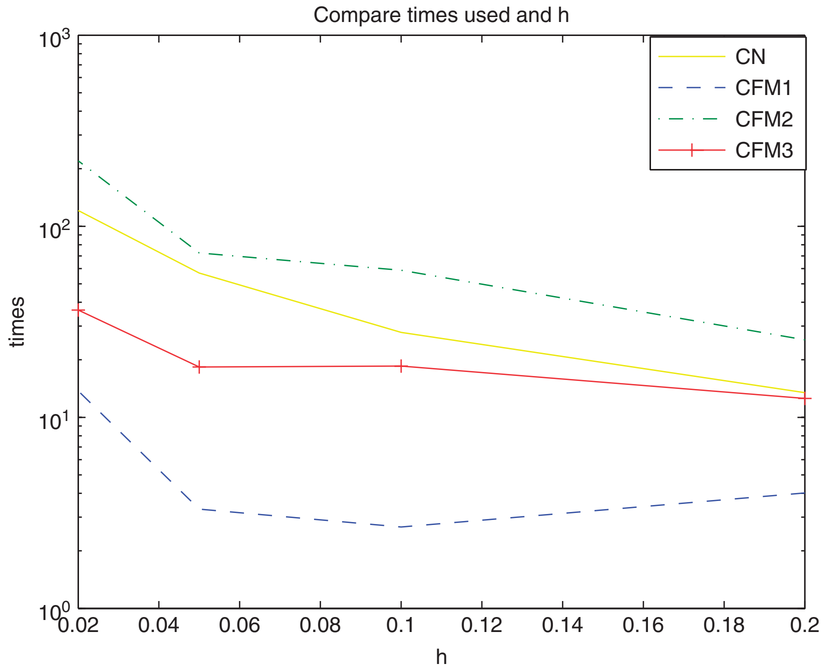

Fig. 3. Time versus space's length h. Note: in this graph, we do not incorporate the times used by method of Barone-Adesi and Elliott [2] to compute free boundary values for the *compact finite difference method three* due to the large expense of this calculation when $n = 5000$.

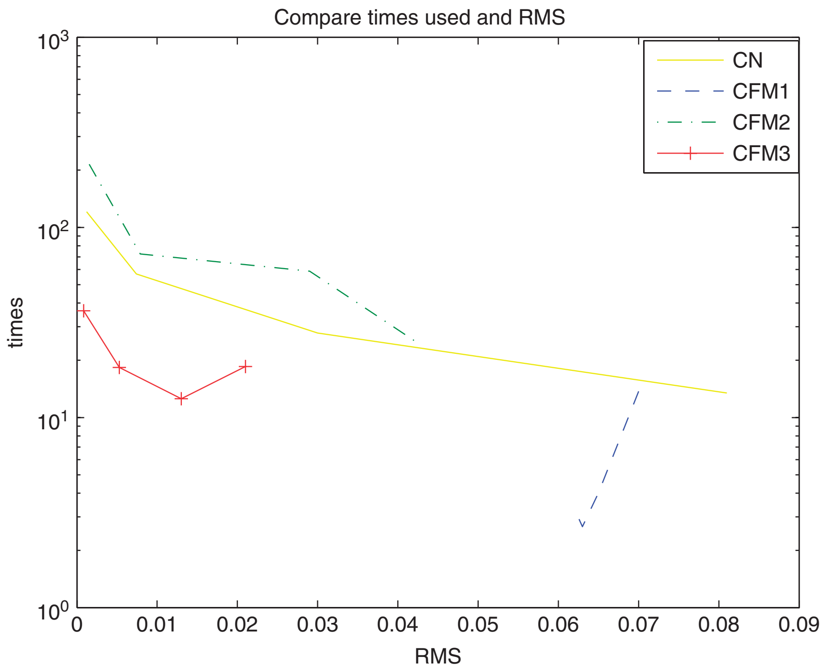

Fig. 4. Relative RMS versus time. Note: in this graph, we do not count in the times used by method of Barone-Adesi and Elliott [2] to compute free boundary values for the *compact finite difference method three*.

Acknowledgments

We thank the anonymous referees for their very useful comments and discussions on this paper. Davison and Corless thank NSERC and MITACS for financial support.

References

[1] M. Ahmed, An exploration of compact finite difference methods for the numerical solution of PDE, Ph.D. Thesis, the University of Western Ontario, 1997.
[2] G. Barone-Adesi, R. Elliott, Approximations for the values of American options, Stochast. Anal. Appl. 9 (2) (1991).
[3] G. Barone-Adesi, U. Lugano, The saga of the American put, 2003.
[4] G. Barone-Adesi, R. Whaley, Efficient analytic approximation of American option values, J. Finance (1987) 301–320.
[5] E.K. Blum, Numerical Analysis and Computation: Theory and Practice, Addison-Wesley Publishing Co., Reading, MA, 1972.
[6] P. Boyle, A lattice framework for option pricing with two state variables, J. Financial Quant. Anal. 23 (1) (1988) 1–12.
[7] M.J. Brennan, E.S. Schwartz, Finite difference methods and jump processes arising in the pricing of contingent claims: a synthesis, J. Financial Quant. Anal. (1978) 461–474.
[8] P. Carr, D. Faguet, Fast accurate valuation of American options, Working Paper, Cornell University, 1994.
[9] P. Carr, R. Jarrow, R. Myneni, Alternative characterizations of American put options, J. Math. Finance 2 (1992) 87–106.
[10] R.M. Corless, J. Rokicki, J. Zhao, FINDIF: a routine for generation of finite difference formulae, share library package 1994, and upgraded to n dimensions for "iguana", 2006.
[11] G. Courtadon, A more accurate finite difference approximations for the valuation of options, J. Financial Quant. Anal. 17 (5) (1982) 697–703.
[12] J.C. Cox, S.A. Ross, M. Rubinstein, Option pricing: a simplified approach, J. Financial Economics 7 (1979) 229–263.
[13] B. Düring, M. Fournié, A. Jüngel, High order compact finite difference schemes for a nonlinear Black–Scholes equation, Int. J. Theor. Appl. Finance 6 (7) (2003) 767–789.
[14] B. Düring, M. Fournié, A. Jüngel, Convergence of a high-order compact finite difference schemes for a nonlinear Black–Scholes equation, Math. Model. Numer. Anal. 38 (2) (2004) 359–369.
[15] V. Druskin, S. Moshow, Three-point finite-difference schemes, Padé and the spectral Galerkin method. I. One-sided impedance approximation, Math. Comput. 71 (239) (2001) 995–1019.
[16] L. Gamet, F. Ducros, F. Nicoud, Compact finite difference schemes on non-uniform meshes, Application to direct numerical simulations of compressible flows, Int. J. Numer. Methods Fluids 29 (159) (1999).
[17] R. Geske, H. Johnson, The American put options valued analytically, J. Finance 39 (1984) 1511–1524.
[18] R.S. Hirsh, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique, J. Comput. Phys. 19 (1975) 90–109.
[19] R.S. Hirsh, Higher order approximations in fluid mechanics, Von Karman Institute for Fluid Dynamics Lecture Series, 1983–1984.
[20] S.D. Jacka, Optimal stopping and the American put, Math. Finance 1 (2) (1991) 1–14.
[21] R. Jarrow, A. Rudd, Tests of an approximate option-valuation formula, in: M. Brenner (Ed.), Option Pricing Theory and Applications, Lexington Books, Lexington, MA, 1983, pp. 81–100.
[22] Y. Jiang, J.M. Floryan, Finite-difference 4th-order compact scheme for the direct numerical simulation of instabilities of shear layers, Internat. J. Numer. Methods Fluids 48 (2005) 1259–1281.
[23] Y. Jin, J. Zhao, L. Ma, R.M. Corless, A new accurate algorithm for solving the partial differential equation in two dimensional cardiac tissue models, WSEAS Trans. Biol. Biomed. 3 (2) (2006) 63–68.
[24] C.H. John, Options, Futures, and Other Derivatives, fourth ed., Prentice-Hall Inc., Englewood Cliffs, NJ, 1999.
[25] I.J. Kim, The analytic valuation of American options, Rev. Financial Stud. 3 (1990) 547–572.
[26] K.K. Lele, Compact finite-difference schemes with spectral-like resolution, J. Comput. Phys. 103 (16) (1992).
[27] F.A. Longstaff, E.S. Schwartz, Valuing American options by simulation: a simple least-squares approach, Rev. Financial Stud. 14 (2001) 113–147.
[28] L.W. MacMillan, An analytical approximation for the American put prices, Adv. Futures Options Res. 1 (1986) 119–139.
[29] G.H. Meyer, J. Van der Hoek, The valuation of American options with the method of lines, Adv. Futures Options Res. 9 (1997) 265–286.
[30] K.N. Pantazopoulos, Numerical methods and software for the pricing of American financial derivatives, Ph.D. Thesis, Computer Science Department, Purdue University, 1998.
[31] M.F. Pettigrew, On compact finite difference schemes with applications to moving boundary problems, Ph.D. thesis, the University of Western Ontario, 1989.
[32] J. Rokicki, J.M. Floryan, A compact finite-difference method for the Navier–Stokes equations—its implementation on parallel computers, in: Proceedings of the XI Annual Conference on Fluid Mechanics, Warsaw, Poland, October 17–21, 1994, pp. 78–91.
[33] L.F. Shampine, M.W. Reichelt, The Matlab ODE suite, SIAM J. Sci. Comput., 2006, to appear.
[34] L.F. Shampine, M.W. Reichelt, The Matlab ODE suite, SISC 18 (1) (1997).
[35] S.E. Sherer, J.N. Scott, High-order compact finite-difference methods on general overset grids, J. Comput. Phys., 2006, to appear.
[36] R.K. Shukla, X.L. Zhong, Derivation of high-order compact finite difference schemes for non-uniform grid using polynomial interpolation, J. Comput. Phys. 205 (2004) 404–429.
[37] R. Smith, Optimal and near-optimal advection-diffusion finite-difference schemes 3. Black–Scholes equation, Proc. Roy. Soc. Lond. A 1999.
[38] H. Sun, J. Zhang, A high order compact boundary value method for solving one dimensional heat equations, Technical Report No. 333-02, University of Kentucky, 2002.

[39] P. Wilmott, J. Dewynne, S. Howison, Option Pricing: Mathematical Model and Computation, Oxford Financial Press, 1995.
[40] G.B. Wright, B. Fornberg, Scattered node compact finite difference-type formulas generated from radial basis functions, 2005, to appear.
[41] J. Zhang, An explicit fourth-order compact finite difference scheme for three dimensional convection-diffusion equation, Commun. Numer. Methods Eng. 14 (1998) 263–280.
[42] J. Zhao, R.M. Corless, Compact finite difference method for integro-differential equations, Appl. Math. Comput. 177 (2006) 271–288.
[43] J. Zhao, R.M. Corless, Compact finite difference method for high order integro-differential equations, Appl. Numer. Math., 2005, submitted for publication.
[44] J. Zhao, R.M. Corless, M. Davison, Financial applications of symbolically generated compact finite difference formulae, in: International Workshop on Symbolic-Numerical Computation Proceedings, Xi'an, China, 2005, pp. 220–234.
[45] J. Zhao, Y. Jin, L. Ma, R.M. Corless, A highly efficient and accurate algorithm for solving the partial differential equation in cardiac tissue models, in: Proceedings of the 2006 WSEAS International Conference on Mathematical Biology and Ecology, Miami, FL, USA, January 18–20, 2006, pp. 81–86.
[46] J. Zhao, T. Zhang, R.M. Corless, Convergence of the compact finite difference method for 2nd order elliptic equations, Appl. Math. Comput., 2006, accepted.