

American Put Option: Richardson's Extrapolation And A Posteriori Error Estimator For A Front-Fixing Finite Difference Scheme*

Riccardo Fazio[†]

Received 1 October 2019

Abstract

In the market for financial derivatives, the most important problem is the so-called *option valuation problem* or in a few words: the problem of computing a fair price for a given option. Analytical solutions of American options problems are seldom available, but such derivatives of financial markets can be priced by numerical methods.

For the numerical solution of the American option valuation problem, we provide a script written in MATLAB implementing an explicit finite difference scheme. Our main contribute is the definition of a posteriori error estimator for the American options pricing which is based on Richardson's extrapolation theory. This error estimator allows us to find a suitable grid where the computed solutions, both the option price field variable and the free boundary position, satisfies a prefixed error tolerance.

1 Introduction

In the market for financial derivatives, the most important problem is the so-called *option valuation problem* or in a few words: the problem of computing a fair price for a given option. An American call (put) option is a contract written on an underlying asset and gives the holder the right to buy (sell) the asset for a pre-specified price or strike price on or before a pre-specified date also indicated as maturity. Unlike European options where the holder can exercise the option only on the maturity date, the possibility of early exercise makes the pricing of American options a problem in stochastic optimization. While a closed-form solution for the price of European options is derived in the celebrated work by Black and Scholes [3] and by Merton [27], there exists no analogous result for American options. The reason can be explained as follows: while the governing differential equation is still the one obtained by Black and Scholes [3], McKean [21] and Merton [26] show that the price of an American option satisfies boundary conditions governed by a boundary that is not known a priori and needs to be computed as part of the solution itself. Such problems are called free-boundary problems. In particular, the American call option problem is a free boundary problem defined on a finite interval. On the other hand, the American put option problem is a free boundary problem defined on a semi-infinite interval so that it is a non-linear problem complicated by a boundary condition at infinity. Unfortunately, the vast majority of listed options in financial markets are American-style. Nonetheless, much progress has been made in developing approaches which approximate the value of such options. These approaches bifurcate into either numerical or analytic approximations.

Within analytical approximations, MacMillan [25] and Barone-Adesi and Whaley [2] defined a quadratic approximation for American put option. These methods are not convergent and have trouble pricing long-maturity options accurately. To correct this problem, Ju and Zhong [22] develop an approximation based on the method proposed in Barone-Adesi and Whaley. While this improved method prices long-maturity options more accurately, it is still not convergent. Johnson [20] used an interpolation scheme to price the American put option, Geske and Johnson [17] derived a valuation formula expressed in terms of a series

*Mathematics Subject Classifications: 65M06.

[†]Department of Mathematics, Computer Science, Physical Sciences and Earth Sciences, University of Messina, Viale F. Stagno D'Alcontres 31, 98166 Messina, Italy. e-mail: rfazio@unime.it home-page: <http://mat521.unime.it/fazio>

of compound-option functions for the same reason, Bunch and Johnson [7] propose a modified two-point Geske-Johnson approach. Carr and Faguet [8] view put options as the limit of a sequence of perpetual option values which are subject to default risk, and use this view to deriving approximations for the price of an option. More recently, Zhu [35] derives a semi-closed form solution for the price of the option as a Taylor series expansion consisting of infinite terms, but requiring thirty terms for an accurate option price.

Within numerical approximations, the most popular methods for pricing American option can be classified to lattice method, Monte Carlo simulation, and finite difference method. The lattice method is conceptually simple and widely implemented for evaluating American options. This method was first introduced by Cox et al. [11], and its convergence was proved by Amin and Khanna [1]. Fu [15, 16] applied Monte Carlo method along with gradient-based optimization techniques to price American style options. The application of a finite difference method to price American options was initiated by Brennan and Schwartz [4, 5] and Schwartz [32]. Jaillet et al. [19] proved the finite difference method convergence. A front-fixing finite difference method was proposed by Wu and Kwok (1997), Nielsen et al. [28], and Company et al. [9] to compute option prices. The front-fixing method utilizes a change of variables to transform the free boundary problem into a nonlinear problem on a fixed domain. Nielsen et al. [28] also propose a penalty method to price American put options, where the unknown boundary is removed by adding a penalty term, again leading to a nonlinear problem posed on a fixed domain. Finally, comparison of different numerical methods are reported by Broadie and Detemple [6] and by Geske and Shastri [18].

In this paper, we list a script written in MATLAB implementing a finite difference scheme for the numerical solution of the American option models of financial markets. We implemented both the method defined by Company et al. [9] and the method developed by Nielsen et al. [28] and found that the first method implementation is more efficient than the second method. Our main contribute is the definition of a posteriori error estimator for the American options pricing which is based on Richardson's extrapolation theory. This error estimator allows us to find a suitable grid where the computed solution, both the option price field variable and the free boundary position, satisfies a prefixed error tolerance.

2 American Put Option

Let us suppose that at time t the price of a given underlying asset is S . We consider here the following mathematical model for the value $P = P(S, \tau)$ of an American put option to sell the asset:

$$\begin{aligned} \frac{\partial P}{\partial \tau} &= \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} + rS \frac{\partial P}{\partial S} - rP \quad \text{on } 0 \leq \tau \leq T, \quad S^*(\tau) < S < \infty, \\ P(S, 0) &= \max(E - S, 0), \quad S^*(0) = E, \\ \lim_{S \rightarrow \infty} P(S, \tau) &= 0, \\ P(S^*(\tau), \tau) &= E - S^*(\tau), \quad \frac{\partial P}{\partial S}(S^*(\tau), \tau) = -1, \\ P(S, \tau) &= E - S, \quad 0 \leq S < S^*(\tau) \end{aligned} \tag{1}$$

where $\tau = T - t$ denotes the time to maturity T , $S^*(\tau)$ is a free boundary, that is the unknown early exercise boundary, σ , r and E are given constant parameters representing the volatility of the underlying asset, the interest rate and the exercise price of the option, respectively.

To fix the free boundary we apply the dimensionless new variables

$$x = \ln \frac{S}{S_f(\tau)}, \quad S_f = \frac{S^*(\tau)}{E}, \quad p(x, \tau) = \frac{P(xS_f(\tau), \tau)}{E}, \tag{2}$$

see Wu and Kwok [33]. From the variables transformation defined by (2) follows that $S_f(\tau)$ is mapped on the fixed line $x = 0$, $0 \leq p(x, \tau) \leq 1$ and $0 \leq S_f(\tau) \leq 1$. By using (2), the put option problem (1) can be

rewritten as follows

$$\frac{\partial p}{\partial \tau} = \frac{1}{2}\sigma^2 \frac{\partial^2 p}{\partial x^2} + \left(r - \frac{\sigma^2}{2}\right) \frac{\partial p}{\partial x} - rp + \frac{1}{S_f(\tau)} \frac{dS_f}{d\tau}(\tau) \frac{\partial p}{\partial x}, \quad (3)$$

$$p(x, 0) = 0 \quad \text{for } 0 \leq x, \quad S_f(0) = 1, \quad (4)$$

$$\lim_{x \rightarrow \infty} p(x, \tau) = 0, \quad (5)$$

$$p(0, \tau) = 1 - S_f(\tau), \quad \frac{\partial p}{\partial x}(0, \tau) = -S_f(\tau), \quad (6)$$

that has to be solved on the domain defined by $0 \leq \tau \leq T$ and $0 < x < \infty$.

3 An Explicit Finite Difference Scheme

To solve the problem (3-6) numerically, we introduce a truncated boundary x_∞ , which is a suitable large value where it would be convenient to impose the asymptotic boundary condition. In other words, we replace the asymptotic boundary condition (5) with the side condition

$$p(x_\infty, \tau) = 0. \quad (7)$$

For the choice of x_∞ and the accuracy of the related numerical solution, we can refer to the study by Kangro and Nicolaidis [23]. On the other hand, the boundary condition at infinity can be enforced exactly by using a non-standard finite difference scheme, and this has been shown for the numerical solution of the so-called perpetual American put option in [12].

Next, by setting an integer J and a positive value μ , we define the step-sizes

$$\Delta x = \frac{x_\infty}{J}, \quad \Delta t = \mu \Delta x^2, \quad (8)$$

the integer N

$$N = \left\lceil \frac{T}{\Delta t} \right\rceil, \quad (9)$$

where $\lceil \cdot \rceil : \mathbb{R}^+ \rightarrow \mathbb{N}$ is the *ceiling* function which maps a real number to the least integer that is greater than or equal to that number. Therefore, μ is the grid-ratio

$$\mu = \frac{\Delta t}{\Delta x^2}. \quad (10)$$

So that, within the finite domain, we can introduce a mesh of grid-points

$$x_j = j\Delta x, \quad t_n = n\Delta t, \quad (11)$$

for $j = 0, 1, \dots, J$ and $n = 0, 1, \dots, N$.

We would like to define a numerical scheme that allows us to compute the grid values

$$p_j^n \approx p(x_j, t_n), \quad (12)$$

for $j = 0, 1, \dots, J$ and $n = 0, 1, \dots, N - 1, N$ and the free boundary values

$$S_f^n \approx S_f(t_n), \quad (13)$$

for $n = 0, 1, \dots, N - 1, N$. To this end let us consider the explicit finite difference scheme

$$\begin{aligned} \frac{p_j^{n+1} - p_j^n}{\Delta t} &= \frac{1}{2}\sigma^2 \frac{p_{j-1}^n - 2p_j^n + p_{j+1}^n}{(\Delta x)^2} + \\ &+ \left(r - \frac{\sigma^2}{2}\right) \frac{p_{j+1}^n - p_{j-1}^n}{2\Delta x} + \frac{S_f^{n+1} - S_f^n}{\Delta t S_f^n} \frac{p_{j+1}^n - p_{j-1}^n}{2\Delta x} - rp_j^n \end{aligned} \quad (14)$$

for $j = 1, 2, \dots, J-1$ and $n = 0, 1, \dots, N-1$. For our specific problem p_j^n and S_f^n are given and our goal is to compute p_j^{n+1} and S_f^{n+1} . If we apply some simple algebra, then we can rewrite the explicit finite difference scheme as follows

$$p_j^{n+1} = ap_{j-1}^n + bp_j^n + cp_{j+1}^n + \frac{S_f^{n+1} - S_f^n}{\Delta t S_f^n} \frac{p_{j+1}^n - p_{j-1}^n}{2\Delta x}, \quad (15)$$

for $j = 2, 3, \dots, J-1$ and $n = 0, 1, \dots, N-1$, where

$$\begin{aligned} a &= \frac{\mu}{2} \left[\sigma^2 - \left(r - \frac{\sigma^2}{2} \right) \Delta x \right], \\ b &= 1 - \mu\sigma^2 - r\Delta t, \\ c &= \frac{\mu}{2} \left[\sigma^2 - \left(r + \frac{\sigma^2}{2} \right) \Delta x \right]. \end{aligned} \quad (16)$$

Now, we have to take into account the side conditions. From the two initial conditions (4), we obtain

$$p_j^0 = 0, \quad S_f^0 = 1, \quad (17)$$

for $j = 0, 1, \dots, J$. From the boundary conditions (7), we get

$$p_j^n = 0, \quad (18)$$

for $n = 0, 1, \dots, N$. From the two boundary conditions (6), using a central finite difference formula, we derive

$$\frac{p_1^n - p_{-1}^n}{2\Delta x} = -S_f^n, \quad p_0^n = 1 - S_f^n, \quad (19)$$

where $x_{-1} = -\Delta x$ is a fictitious point out of the computational domain. Moreover, by considering the governing differential equation (3) at $x_0 = 0$, $\tau > 0$ and taking into account the side conditions (6) one gets a new boundary condition:

$$\frac{\sigma^2}{2} \frac{\partial^2 p}{\partial x^2}(0^+, \tau) + \frac{\sigma^2}{2} S_f(\tau) - r = 0, \quad (20)$$

see Wu and Kwok [33], Zhang and Zhu [34] or Kwok [24, p. 341], and its central finite difference discretization

$$\frac{\sigma^2}{2} \frac{p_{-1}^n - 2p_0^n + p_1^n}{\Delta x^2} + \frac{\sigma^2}{2} S_f^n - r = 0. \quad (21)$$

Now, we can eliminate the value of p_{-1}^n from equations (19) and (21) to get

$$p_1^n = 1 + \frac{r\Delta x^2}{\sigma^2} - \left(1 + \Delta x + \frac{\Delta x^2}{2} \right) S_f^n. \quad (22)$$

If we evaluate the numerical scheme (15) for $j = 1$ and take into account (22) for $n = n + 1$, then the free boundary S_f^{n+1} can be defined by

$$S_f^{n+1} = d^n S_f^n. \quad (23)$$

for $n = 0, 1, \dots, N-1$, where

$$d^n = \frac{1 + \frac{r\Delta x^2}{\sigma^2} - \left(ap_0^n + bp_1^n + cp_2^n - \frac{p_2^n - p_0^n}{2\Delta x} \right)}{\frac{p_2^n - p_0^n}{2\Delta x} + \left(1 + \Delta x + \frac{\Delta x^2}{2} \right) S_f^n}. \quad (24)$$

Taking into account all the formulae defined above we get the following algorithm:

1. Input σ, r, E, T, J, μ and x_∞ ;
2. Define the grid $(x_j, \Delta x, t_n, \Delta t)$ according to equations (8) and (11);
3. for $j = 0, 1, 2, \dots, J$ do $p_j^0 = 0$ end, set $S_f^0 = 1$;
4. compute a, b and c according to (16);
5. for $n = 0, 1, \dots, N - 1$, compute d^n according to (24), compute S_f^{n+1} according to (23) and apply the free boundary conditions

$$p_0^{n+1} = 1 - S_f^{n+1},$$

$$p_1^{n+1} = 1 + \frac{r\Delta x^2}{\sigma^2} - \left(1 + \Delta x + \frac{\Delta x^2}{2}\right) S_f^{n+1},$$

compute

$$a^n = a - \frac{S_f^{n+1} - S_f^n}{2\Delta x S_f^n}, \quad c^n = c - \frac{S_f^{n+1} - S_f^n}{2\Delta x S_f^n}$$

and for $j = 2, 4, \dots, J - 1$ compute the values of p_j^{n+1} according to

$$p_j^{n+1} = a^n p_{j-1}^n + b p_j^n + c^n p_{j+1}^n.$$

end, set $p_j^n = 0$, end.

The implementation of this algorithm in MATLAB has been used to get the numerical results reported below. The script file is listed in the Appendix.

4 Positivity, Monotonicity and Stability

In this section, we recall the theoretical results that make the explicit difference scheme suitable for doing numerical studies of the American put option. For the sake of simplicity, let us define the numerical solution vector at time level t_n as $p^n = (p_0^n, p_1^n, \dots, p_J^n)^T$.

Lemma 1 (Company et al. [9]) *If Δt and Δx verify the two inequalities*

$$\Delta x \leq \frac{\sigma^2}{|r - \sigma^2/2|} \Delta t, \quad r \neq \sigma^2/2, \quad (25)$$

$$\Delta t \leq \frac{\Delta x^2}{\sigma^2 + r\Delta x^2}. \quad (26)$$

then the coefficients a, b and c are non-negative. If $r = \sigma^2/2$, then the non-negativity of these coefficients is verified under the condition (26).

These two inequalities follow from positivity preserving considerations related to the explicit finite difference scheme. The theoretical framework is similar to the one used by Friedrichs [14] to study positivity preserving finite difference schemes for the advection equation, see also Fazio and Jannelli [13]. Moreover, the coefficients of p_i^{n+1} , for $i = j - 1, j, j + 1$, in our difference scheme (15) might be regarded, in the limit for $\Delta t \rightarrow 0$, as probability values. In fact, they sum up to $1 - r\Delta t$ and by imposing their positivity we get the two inequalities (25)-(26).

Theorem 1 (Company et al. [9]) *Let $\{p_j^n, S_f^n\}$ be the computed numerical solution and d^n be defined by equations (24), then under hypothesis of the Lemma 1, for sufficiently small values of Δx , we have:*

- $\{S_f^n\}$ is positive and non-increasing monotone for $n = 0, 1, \dots, N$;

- the vectors p^n have positive components for $n = 0, 1, \dots, N$;
- the vectors p^n are non-increasing monotone with respect to the j for each fixed $n = 0, 1, \dots, N$.

As far as the stability of the explicit finite difference scheme is concerned we can introduce the definition: the numerical scheme is said to be $\|\cdot\|_\infty$ -stable if, for every mesh in the computational domain $[0, x_\infty] \times [0, T]$, there exists a positive constant C such that

$$\|p^n\|_\infty \leq C \quad \text{for } n = 0, 1, \dots, N, \quad (27)$$

where C is independent on Δt , Δx and n , see Company et al. [10].

Theorem 2 (Company et al. [9]) *Under the hypothesis of Theorem 1 the explicit finite difference scheme for the fixed domain problem (3-6) is $\|\cdot\|_\infty$ -stable.*

5 A Posteriori Error Estimator

For a scalar U of interest, either a value of the solution p_j^n or a free boundary value S_j^n , the numerical error e can be defined by

$$e = u - U, \quad (28)$$

where u is the exact, usually unknown, value. Of course, we have several different sources of errors: discretization, round-off, and so on. Discretization errors are due to our replacement of a continuous problem with a discrete one and these errors can be reduced by reducing the discretization parameters, enlarging the value of N in our case. Round-off errors are due to the utilization of floating-point arithmetic to implement the algorithms available to solve the discrete problem. This kind of error can be reduced by using higher precision arithmetic, double or, when available, fourth precision. Several other kinds of errors can be considered but are all beyond the scope of this study. However, when the numerical error is caused prevalently by the discretization error and in the case of smooth enough solutions the discretization error can be decomposed into a sum of powers of the inverse of N

$$u = U_N + C_0 \left(\frac{1}{N}\right)^{p_0} + C_1 \left(\frac{1}{N}\right)^{p_1} + C_2 \left(\frac{1}{N}\right)^{p_2} + \dots, \quad (29)$$

where C_0, C_1, C_2, \dots are coefficient that depend on u and its derivatives, but are independent on N , and p_0, p_1, p_2, \dots are the true orders of the discretization error, see Schneider and Marchi [31] and the references quoted therein. Each p_k is usually a positive integer with $p_0 < p_1 < p_2 < \dots$ and all together they constitute an arithmetic progression of ratio $p_1 - p_0$. The value of p_0 is called the asymptotic order or the order of accuracy of the method or of the numerical value U_N . By replacing into equation (29) $N = N_g$ and $N = N_{g+1}$ and subtracting, to the second obtained equation the first times $(1/q)^{p_0}$, $q = N_{g+1}/N_g$, we get the first extrapolation formula

$$u \approx U_{g+1} + \frac{U_{g+1} - U_g}{q^{p_0} - 1}, \quad (30)$$

that has a leading order of accuracy equal to p_1 . This type of extrapolation is due to Richardson [29, 30]. Taking into account equation (30) we can conclude that the error estimator by a first Richardson's extrapolation is given by

$$e_r = \frac{U_{g+1} - U_g}{q^{p_0} - 1}, \quad (31)$$

where p_0 is the order of the numerical method used to compute the numerical solutions. Hence, (31) gives the real value of the numerical solution error without knowledge of the exact solution. In comparison with (31) a safer error estimator can be defined by

$$e_s = U_{g+1} - U_g. \quad (32)$$

Of course, p_0 can be estimated with the formula

$$p_0 \approx \frac{\log(|U_g - u|) - \log(|U_{g+1} - u|)}{\log(q)}, \tag{33}$$

where u is again the exact solution (or, if the exact solution is unknown, a reference solution computed with a suitable large value of N), and both u and U_{g+1} are evaluated at the same grid-points of U_g .

Within the above framework, in order to improve the numerical accuracy by using only a small number of grid-nodes, we can generalize (30) introducing the following repeated extrapolation formula

$$U_{g+1,k+1} = U_{g+1,k} + \frac{U_{g+1,k} - U_{g,k}}{q^{p_k} - 1}, \tag{34}$$

where $g \in \{0, 1, 2, \dots, G - 1\}$, $k \in \{0, 1, 2, \dots, G - 1\}$, $q = N_{g+1}/N_g$ is the grid refinement ratio, and p_k is the true order of the discretization error. The formula (34) is asymptotically exact in the limit as N_0 goes to infinity if we use uniform grids. We notice that to obtain each value of $U_{g+1,k+1}$ requires two computed solutions U in two adjacent grids, namely $g + 1$ and g at the extrapolation level k . For any g , the level $k = 0$ represents the numerical solution of U without any extrapolation. We recall that the theoretical orders of accuracy of the numerical values $U_{g,k}$, with $N = N_g$ and k extrapolations, verify the relation

$$p_k = p_0 + k(p_1 - p_0), \tag{35}$$

where this equation is valid for $k \in \{0, 1, 2, \dots, G - 1\}$.

6 Numerical Results

We introduce a numerical test for the finite difference schemes defined below. To this end, we consider the American put option problem (3-6) with the following parameters:

$$r = 0.1, \quad \sigma = 0.2, \quad E = T = 1. \tag{36}$$

First of all, we are interested to define a suitable value of the truncated boundary x_∞ . Then, we investigate numerically how the choice of the value of x_∞ influences the numerical solution. To take a simple approach, we can monitor the final free boundary computed values S_f^N . Therefore, in Table 1 we compare sample numerical results of the free boundary location S_f^N at $t = T$ obtained for different values of grid-steps, but always with $\mu = 20$, and by setting $x_\infty = 1$, $x_\infty = 2$ and $x_\infty = 4$. Motivated by the results reported in

x_∞	$N = 10$	$N = 20$	$N = 40$
1	0.867635453443530	0.865575022242718	0.864386633629751
2	0.867635453443530	0.865575022242718	0.864386633629751
4	0.867635453443530	0.865575022242718	0.864386633629751

Table 1: Free boundary location at $t = T$.

Table 1 we decided to use $x_\infty = 1$. This can be contrasted with the preliminary numerical results obtained by the MATLAB implementation of the explicit finite difference scheme developed by Nielsen et al. [28]. In fact, in that case we were forced to use $x_\infty = 4$.

Figure 1 shows an unstable computation. For the sake of clarity, in Figure 1 we display only the initial condition and the numerical solution at $t = T$, that is the last time iterate. As usual, the instability manifest itself with large oscillations between positive and negative values. Therefore, as far as our explicit finite difference scheme is concerned, we have to find a compromise between accuracy and stability. Of course, from now on, the chosen grid-spacings are defined in order to verify the stability inequalities (25)-(26).

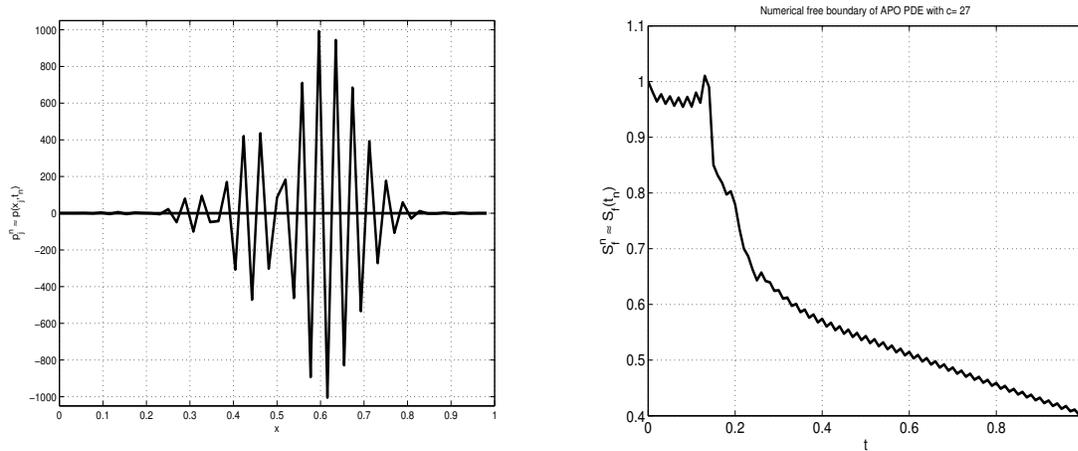


Figure 1: *Instability for $N = 100$, $\mu = 27$ and $x_\infty = 1$: on the left p_j^N versus x_j , on the right, S_f^n versus t_n .*

From the results listed in Table 1 we realize that, fixed a value of the truncated boundary x_∞ , the computed values of S_f^N for different values of the grid-steps are in agreement only for the first two decimal places. Then, we decided to improve the numerical accuracy by performing a mesh refinement. Moreover, we applied repeated Richardson's extrapolation to improve the numerical accuracy. Let us recall that the explicit difference scheme is first order in time and second order in space both for the field variable and the free boundary value, i.e. the truncation error is of the order $O(\Delta t^2) + O(\Delta t \Delta x^2)$. We will use this result below when we perform a mesh refinement keeping constant the grid-ratio, i.e. μ , so that we end up with second-order truncation error $T_j^n = O(\Delta t^2)$ in time and, therefore, the global error is first-order, that is the p_0 value defined above is equal to one. We remark that in our case the sequence of q^{pk} , for $k = 0, 1, \dots$, is given by 4, 16, 64, 256, 1024, \dots , that is $q = 4$ and $p_k = k + 1$, for $k = 0, 1, \dots$. In Table 2 we report sample numerical results for the benchmark value S_f^N . Within the same table, we report the results obtained by

N	$U_{g,0}$	$U_{g,1}$	$U_{g,2}$	$U_{g,3}$	$U_{g,4}$
5	0.871621				
20	0.865575	0.863560			
80	0.863700	0.863075	0.863043		
320	0.863071	0.862861	0.862847	0.862844	
1280	0.862859	0.862788	0.862783	0.862782	0.862782
5120	0.862788	0.862764	0.862763	0.862762	0.862762

Table 2: Richardson's repeated extrapolations with $x_\infty = 1$ for the free boundary value S_f^N at $t = T$. The next extrapolated value is $U_{5,5} = 0.862762$.

repeated Richardson's extrapolations. From the results listed in Table 2 we can conclude that our benchmark value is $S_f^N \approx 0.862762$. This value can be compared with the values $S_f^N \approx 0.86222$ computed by Nielsen et al. [28] and $S_f^N \approx 0.8628$ found by Company et al. [9].

Next, we indicate how to use the error estimator defined by equation (31), or alternatively by equation (32). Let us assume that our goal is to solve the American put option problem with a given tolerance ϵ , where $0 < \epsilon \ll 1$. To this end we should solve the given problem twice, for two grids defined with given values of $J_g = J$ and $J_{g+1} = 2J$ of space intervals but for the same value of the grid-ratio μ . The corresponding time intervals N_g and N_{g+1} verify the relation $q = N_{g+1}/N_g$. Hence we can apply (component-wise) to p^n and to

S_f^n the error estimator formula (31), or (32). Then, we can verify whether

$$\|e_r(p^n)\|_\infty \leq \epsilon \quad |e_r(S_f^n)| \leq \epsilon \quad \text{for } n = 1, 2, \dots, N. \quad (37)$$

If the two inequalities (37) hold true, for $n = 1, 2, \dots, N$, then we can accept the numerical solution computed on the grid defined by J_{g+1} and N_{g+1} , otherwise we have to increase these two integers and repeat the calculations.

Figure 2 shows the error estimator results computed by setting $\epsilon = 0.001$. We fixed $\mu = 20$ and started with $J_0 = 5$ and $J_1 = 10$ repeating the computation by doubling the number of spatial grid-intervals if the required accuracy was not achieved. Our algorithm stopped when $J_7 = 640$ that for $\mu = 20$ means $N_7 = 20480$.

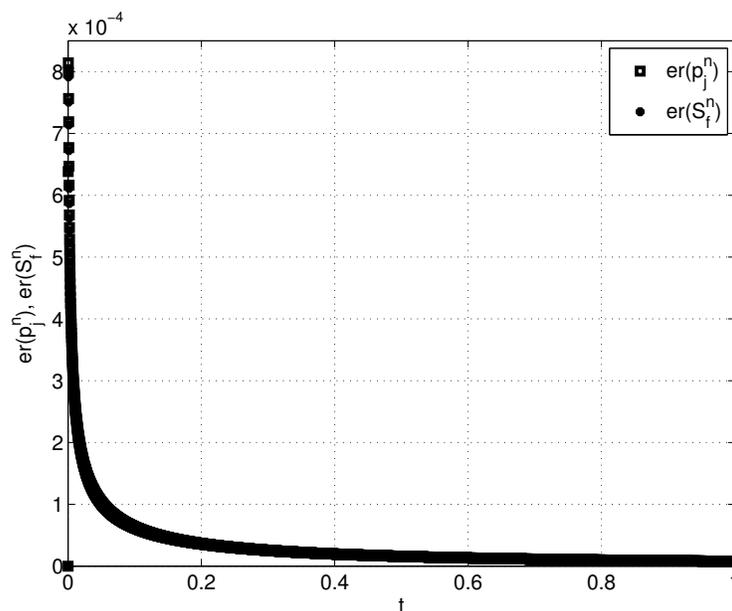


Figure 2: Numerical estimated errors $e_r(p^n)$ and $e_r(S_f^n)$ versus t_n .

For the sake of completeness, in Figure 3 we plot p_j^N versus x_j and S_f^n versus t_n , these results were obtained by the finite difference scheme with $N = 320$ and $\mu = 20$.

From the numerical results shown in Figure 2 we can easily realize that the greatest errors are found within a few time steps. This suggests that a better accuracy can be achieved, without the use of Richardson's extrapolation, by developing an adaptive version of the explicit finite difference scheme. This adaptive version should start with a small value of Δt and increase the time step-size approximatively soon after $T/5$.

7 Appendix

Here we list the basic algorithm written in MATLAB. The script file is called `AP0efds.m` and it can be easily modified to apply our a posteriori error estimator defined by equation (31).

```
% AP0efds.m
% solve APO equation on [0,xinf] by forward in time
% and central finite difference in space
% J+1 space steps in x, N+1 time steps in t
% mi = k/h^2, where k is time step, h is space step
clear all; help AP0efds
r = .1; % parameters
```

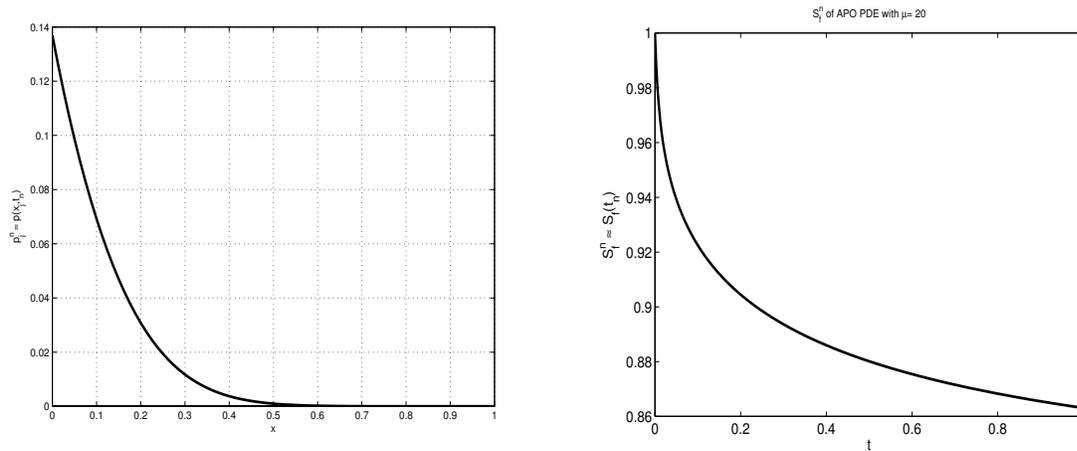


Figure 3: Numerical results: on the left p_j^N versus x_j , on the right, S_f^n versus t_n .

```

T = 1;                % Nielsen et al. 2002
sigma = .2;
E = 1;
xinf = 1;            % truncated boundary
J = 80;
mi = 20
h = xinf/J
k = mi*h^2
N = ceil(T/k)
x = 0:h:xinf;
w = zeros(J+1,1);   % set initial conditions
s(1) = 1;
t(1)=0;
A = .5*mi*(sigma^2-(r-.5*sigma^2)*h);
B = 1-mi*sigma^2-r*k;
C = .5*mi*(sigma^2+(r-.5*sigma^2)*h);
A1 = 1+r*h^2/sigma^2;
B1 = 1+h+.5*h^2;
for n=1:N           %time loop
    dp1 = .5*(w(3,1)-w(1,1))/h;
    D = (A1-(A*w(1,1)+B*w(2,1)+C*w(3,1)-dp1))/(dp1+B1*s(n));
    s(n+1) = D*s(n);
    if (s(n+1)<0 | s(n+1)>1), break, end
    w(1,2) = 1-s(n+1);
    w(2,2) = A1-B1*s(n+1);
    AM = A-.5*(s(n+1)-s(n))/(h*s(n));
    CM = C+.5*(s(n+1)-s(n))/(h*s(n));
    for j = 3:J     %space loop
        w(j,2) = AM*w(j-1,1)+B*w(j,1)+CM*w(j+1,1);
    end
    w(J+1,2) = 0;
    w(:,1) = w(:,2);
    t(n+1) = t(n)+k;

```

```

end
plot(t,s,'LineWidth',2.5);
axis([0 T .86 1]); grid
xlabel('t')
ylabel('S_f^n \approx S_f(t_n)')
title(['S_f^n of APO PDE with \mu= ',num2str(mi)])

```

Of course, our algorithm can be made more complex than this essential one. For instance, as far as the stability conditions (25)-(26) are concerned we can insert, after parameters and grid definitions, the script

```

Stab1 =sigma^2/(abs(r-.5*sigma^2))-h
Stab2 =h^2/(sigma^2+r*h^2)- k
if (Stab1<0 | Stab2<0), break, end

```

Moreover, in the implementation of the error estimator, we can stop the computation as soon as one of the inequalities (37) is violated. This can be done by using the script

```

er(n+1) = abs((w2(1,1)-w(1,1))/3);
    for j = 2:J+1
        em = (w2(2*j-1,1)-w(j,1))/3;
        er(n+1) = max(er(n+1),abs(em));
    end
ers(n+1) = abs((s2(n+1)-s(n+1))/3);
if (er(n+1)>epsilon | ers(n+1)>epsilon), break, end

```

References

- [1] K. Amin and A. Khanna, Convergence of American option values from discrete to continuous-time financial models, *Math. Finance*, 4(1994), 289–304.
- [2] G. Barone-Adesi and R. E. Whaley, Efficient analytic approximation of American option values, *J. Finance*, 42(1987), 301–320.
- [3] F. Black and M. Scholes, The pricing of options and corporate liabilities, *J. Polit. Econ.*, 81(1973), 637–654.
- [4] M. Brennan and E. Schwartz, The valuation of American put options, *J. Finance*, 32(1977), 449–462.
- [5] M. Brennan and E. Schwartz, Finite difference methods and jump processes arising in the pricing of contingent claims: a synthesis, *J. Financ. Quant. Anal.*, 13(1978), 461–474.
- [6] M. Broadie and Detemple, American option valuation: new bounds, approximations, and a comparison of existing methods, *Rev. Financial Stud.*, 9(1996), 1211–1250
- [7] D. S. Bunch and H. E. Johnson, A simple and numerically efficient valuation method for American puts using a modified Geske-Johnson approach, *J. Finance*, 47(1992), 809–816.
- [8] P. Carr and D. Faguet, Valuing finite-lived options as perpetual, 1996, Cornell University Working Paper: 35 pages.
- [9] R. Company, V. N. Egorova, and L. Jódar, Solving American option pricing models by the front fixing method: Numerical analysis and computing, *Abstr. Appl. Anal.*, 2014, Art. ID 146745, 9 pp.
- [10] R. Company, L. Jódar, and J.-R. Pintos, A consistent stable numerical scheme for a nonlinear option pricing model in illiquid markets, *Math. Comput. Simulat.*, 82(2012), 1972–1985.

- [11] J. C. Cox, S. A. Ross, and M. Rubinstein, Option pricing: a simplified approach, *J. Finan. Econ.*, 7(1979), 229–263.
- [12] R. Fazio, Perpetual American put option: an error estimator for a non-standard finite difference scheme, accepted to appear *Appl. Math. E-Notes*.
- [13] R. Fazio and A. Jannelli, Second order positive schemes by means of flux limiters for the advection equation, *IAENG Inter. J. Appl. Math.*, 39(2009), 25–35.
- [14] K. O. Friedrichs, Symmetric hyperbolic differential equations, *Commun. Pure Appl. Math.*, 7(1954), 345–392.
- [15] M. C. Fu, Optimization using simulation: a review, *Annals of Operation Research*, 53(1994), 199–248.
- [16] M. C. Fu, A tutorial review of techniques for simulation optimization, 1994, *Proc. of the 1994 Winter Simulation Conference*, pp. 149–156.
- [17] R. Geske and H. E. Johnson, The American put options valued analytically, *Journal of Finance*, 39(1984), 1511–1524.
- [18] R. Geske and K. Shastri, Valuation by approximation: a comparison of alternative option valuation techniques, *J. Financ. Quant. Anal.*, 20(1985), 45–71.
- [19] P. Jaillet, D. Lamberton and B. Lapeyre, Variational inequalities and the pricing of American options, *Acta Appl. Math.*, 21(1990), 263–289.
- [20] H. E. Johnson, An analytic approximation for the American put price, *J. Financ. Quant. Anal.*, 18(1983), 141–148.
- [21] H. P. McKean, Appendix: A free boundary problem for the heat equation arising from a problem in mathematical economics, *Industrial Management Review*, 6(1965), 32–39.
- [22] N. Ju and R. Zhong, An approximate formula for pricing American options, *J. Derivatives*, 7(1983), 31–40.
- [23] R. Kangro and R. Nicolaidis, Far field boundary conditions for Black-Scholes equations, *SIAM J. Numer. Anal.*, 38(2000), 1357–1368.
- [24] Y.-K. Kwok, *Mathematical Models of Financial Derivatives*, Springer, Springer, 2nd edition, 2008.
- [25] L. W. MacMillan, An analytical approximation for the American put prices, *Ad. Futures Options Res.*, 1(1986), 119–139.
- [26] R. C. Merton, An intertemporal capital asset pricing model. *Econometrica*, 43(1973), 867–887.
- [27] R. C. Merton, The theory of rational option pricing, *Bell J. Econ. Manag. Sci.*, 4(1973), 141–183.
- [28] B. F. Nielsen, O. Skavhaug and A. Tveito, Penalty and front-fixing methods for the numerical solution of American option problems, *J. Comput. Finance*, 5(2002), 69–97.
- [29] L. F. Richardson, The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam, *Proc. R. Soc. London Ser. A*, 210(1910), 307–357.
- [30] L. F. Richardson and J. A. Gaunt, The deferred approach to the limit, *Proc. R. Soc. London Ser. A*, 226(1927), 299–349.

- [31] F. A. Schneider and C. H. Marchi, On the grid refinement ratio for one-dimensional advection problems with nonuniform grids, In Proceedings of COBEM 2005, 2005, 18th International Congress of Mechanical Engineering, 8 pages.
- [32] E. S. Schwartz, The valuation of warrants: implementing a new approach, *J. Financ. Econ.*, 4(1977), 79–93.
- [33] L. Wu and Y.-K. Kwok, A front-fixing method for the valuation of American option, *J. Financ. Eng.*, 6(1997), 83–97.
- [34] J. Zhang and S. P. Zhu, A hybrid finite difference method for valuing American puts, In S.I. Ao, L. Gelman, D.W.L. Hukins, A. Hunter, and A.M. Korsunsky, editors, Proceedings World Congress on Engineering 2009, held in London, 1-3 July, volume II, pages 1131–1136. IAENG, Kwun Tong – Hong Kong, 2009.
- [35] S. P. Zhu, An exact and explicit solution for the valuation of American put options, *Quant. Finance*, 6(2006), 229–242.