On a Parallel Time-domain Method for the Nonlinear Black-Scholes Equation

Choi-Hong Lai[1], Diane Crane[2], and Alan Davies[2]

[1] School of Computing and Mathematical Sciences, University of Greenwich, Old Royal Naval College, Greenwich, London SE10 9LS, UK.C.H.Lai@gre.ac.uk
[2] Department of Physics, Astronomy, and Mathematics, University of Hertfordshire, Hatfield Campus, Herts AL10 9AB, UK.
{D.Crann,A.J.Davies}@herts.ac.uk

1 Introduction

A parallel time-domain algorithm is described of the time-dependent nonlinear Black-Scholes equation, which may be used to build financial analysis tools to help traders making rapid and systematic evaluation of buy/sell contracts. The algorithm is particularly suitable for problems that do not require fine details at each intermediate time step, and hence the method applies well for the present problem.

The method relies on a Laplace transform technique applied to the Black-Scholes equation and generates subproblems that can be executed in a parallel/distributed computing environment. These subproblems are thus solved independently without subproblem communication. Early studies of the scalability of the algorithm for linear Black-Scholes equation may be found in [Cra96] and [CDL98]. This paper extends the previous work to nonlinear Black-Scholes equation. Two linearization methods, one based on the updating of nonlinear coefficients within an iterative loop and the other based on a Newton's method. A numerical inverse [Ste70, Wid46] of the approximate solution is used to retrieve the final solution of the nonlinear Black-Scholes equation. Numerical tests are performed to demonstrate the viability of the algorithm. Efficiency of the algorithm is also studied.

This paper concludes with a discussion on an extension of the present Laplace transform technique to a parallel time-domain algorithm in order to obtain detials of physical quantities at intermediate finer time steps.

2 A Nonlinear Black-Scholes Model

Let $v(S,t)$ denote the value of an option, where S is the current value of the underlying asset and t is the time. The value of the option relates to the

current value of the underlying asset via the Black-Scholes equation:

$$\frac{\partial v}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 v}{\partial S^2} + rS - rv = 0 \in \Omega^+ \times [T, 0) \tag{1}$$

where $\Omega^+ = \{S : S \geq 0\}$. The stochastic background of the equation is not discussed in this paper, and readers who are interested should consult [Wil93].

Only European options are considered in this paper. This means that the holder of the option may execute at expiry a prescribed asset, known as the underlying asset, for a prescribed amount, known as the strike price. There are two different types of option, namely the call option and the put option. At expiry, the holder of the call option has the right to buy the underlying asset and the holder of the put option has the right to sell the underlying asset. For a European put option with strike price k and expiry date T, it is sensible to impose the boundary condition $v(0,t) = ke^{-r(T-t)},\ v(L,t) = 0$, where L is usually a large value. At expiry, if $S < k$ then one should exercise the call option, i.e. handing over an amount k to obtain an asset with S. However, if $S > k$ at expiry, then one should not exercise the option because of the loss $k - S$. Therefore the final condition $v(S,T) = \max\{k - S, 0\}$ needs to be imposed. The solution v for $t < T$ is required.

Since (1) is a backward equation, it needs to be transformed to a forward equation by using $\tau = T - t$, which leads to,

$$\frac{\partial V}{\partial \tau} = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS - rV \in \Omega^+ \times (0, T] \tag{2}$$

subject to initial condition $V(S,0) = \max\{k - S, 0\}$ and boundary conditions $V(0,\tau) = ke^{-r\tau},\ V(L,\tau) = 0$. A field method, such as the finite volume method, is of more interest for two reasons. First, there are many examples in multi-factor model such that a reduction of the time dependent or nonlinear coefficient to a constant coefficient heat is impossible. Hence analytic form of solutions cannot be found. Second, financial modelling typically requires large number of simulations and solutions at intermediate time steps are usually not of interest. Efficiency of the numerical algorithm is very important in order to make evaluation and decision before the agreement of a contract is reached. Ideally one would like to use an algorithm which can be completely distributed onto a number of processors with only minimal communications between processors.

Very often, over a short period of time the interest rate, r, is fixed while the volatility, σ, is varying. The volatility may be a function of the transaction costs [BarSon98], the second derivative of the option value [ParAve94], or, in some cases, the solution of a nonlinear initial value problem [BarSon98]. In order to develop the nonlinear solver in this section, the volatility $\sigma = \sigma_0\sqrt{1 + a}$ proposed in [BoyVor73] is used, where a is the proportional transaction cost scaled by σ_0 and the transaction time. Very often the transaction cost is related to the option value and follows a Gaussian distribution. In order to

demonstrate the time-domain parallel algorithm for nonlinear problems, a sine function is used in the subsequent tests to produce the effect of a pulse-like distribution instead of a Gaussian distribution, i.e. $a = \sin(\frac{V\pi}{k})$ where k is the strike price.

3 Reference Solutions Using a Temporal Integration

The forward Black-Scholes equation given by (2) is written as

$$\frac{\partial V}{\partial \tau} = A(V)\frac{\partial^2 V}{\partial S^2} + rS - rV = 0 \in \Omega^+ \times (0, T] \tag{3}$$

where $A(V) = \frac{1}{2}\sigma(V)^2 S^2$. In order to obtain a reference solution for (3) a linearisation method combined with a temporal integration may be applied. The coefficient A is computed by using an approximation $\bar{V}$, which is updated in every step of a nonlinear iterative update process. Each step of the nonlinear iterative update process involves a numerical solution to the equation

$$\frac{\partial V}{\partial \tau} = A(\bar{V})\frac{\partial^2 V}{\partial S^2} + rS - rV = 0 \in \Omega^+ \times (t_i, t_{i+1}] \tag{4}$$

defined in the time interval $\tau \in (t_i, t_{i+1}]$. Let $V^{(n)}(S, t_{i+1})$ and $V^{(n)}(S, t_i)$ be the numerical solutions of (3) at $\tau = t_{i+1}$ and $\tau = t_i$ respectively. The nonlinear iterative update process to obtain the numerical solution $V^{(n)}(S, t_{i+1})$, using $V^{(n)}(S, t_i)$ as the initial approximation to $\bar{V}$, is described in the algorithm below.

Algorithm R: Obtain a reference solution for (3).

do i = 0,1,2,...
$t_i = i\delta\tau$;
Initial approximation:- $V^{(0)}(S, t_{i+1}) := V^{(n)}(S, t_i)$; $k := 0$;
Iterate
$k := k + 1$;
$\bar{V} := V^{(k-1)}(S, t_{i+1})$;
Compute $A(\bar{V})$;
$V^{(k)}(S, t_{i+1})$:= Apply Euler's method to (4);
Until $\|V^{(k)}(S, t_{i+1}) - V^{(k-1)}(S, t_{i+1})\| < \epsilon$
$n := k$;
end-do

4 The Parallel Time-Domain Method

Let

$$l(V) = \int_0^\infty e^{-\lambda\tau} V(S,\tau) d\tau = U(\lambda; S)$$

be the Laplace transform of the function $V(S.\tau)$. Application of the Laplace transform [Wid46] to (4), now being defined in $\Omega^+ \times (T_i, T_{i+1}]$, leads to

$$A(\bar{V})\frac{d^2U}{dS^2} + rs\frac{dU}{dS} - (r+\lambda)U = -V(S,T_i) \in \Omega^+ \tag{5}$$

where $U = U(\lambda; S)$ defined in the Laplace space. Here $\lambda \in \{\lambda_j\}$ is a finite set of transformation parameter defined by

$$\lambda_j = j\frac{\ln 2}{T_{i+1} - T_i}.\ j = 1, 2, ..., m \tag{6}$$

where m is required to be chosen as an even number [Ste70]. Therefore the problem defined in (4) is converted to m independent parametric boundary value problems as described by (5), and these problems may be distributed and solved independently in a distributed environment.

In order to retrieve $V(S, T_{i+1})$, the approximate inverse Laplace transform due to Stehfest [Ste70] given by

$$V(S,T_{i+1}) \approx \frac{\ln 2}{T_{i+1} - T_i} \sum_{j=1}^{m} w_j U(\lambda_j; S) \tag{7}$$

where

$$w_j = (-1)^{m/2+j} \sum_{k=(1+j)/2}^{\min(j,m/2)} \frac{k^{m/2}(2k)!}{(m/2-k)!k!(k-1)!(j-k)!(2k-j)!}$$

is known as the weighting factor, is used. The authors select Stehfest method because of previous experience with the method used for linear problems [Cra96, CDL98] and wish to investigate the application of the inverse method to nonlinear problems.

A nonlinear iterative update process is required to update $\bar{V}$ and to obtain the numerical solution $V^{(n)}(S, T_{i+1})$, using $V^{(n)}(S, T_i)$ as the initial approximation to $\bar{V}$, and is described in the algorithm below.

Algorithm P1: Parallel algorithm 1 for (3).

do i = 0,1,2,...

$T_i = i\Delta\tau$;

Initial approximation:- $V^{(0)}(S, T_{i+1}) := V^{(n)}(S, T_i)$; $k := 0$;

Iterate

$k := k + 1$; $\bar{V} := V^{(k-1)}(S, T_{i+1})$; Compute $A(\bar{V})$;

Parallel for $j := 1$ to $m(i)$

Solve (5) for $U(\lambda_j; S)$;

End parallel for

Compute $V^{(k)}(S, T_{i+1})$ using inverse Laplace transform (7);
Until $\|V^{(k)}(S, T_{i+1}) - V^{(k-1)}(S, T_{i+1})\| < \epsilon$
$n := k$;
end-do

Here $m(i)$ is the number of transformation parameters and $T_i = \Delta\tau$. In order to solve (5) for $U(\lambda_j; S)$, one can employ the finite volume technique as the one used in Section 3. In essence the actual implementation does not require different values of $m(i)$ for many problems, and the results shown in this paper use the same number of transformation parameters, denoted as $\bar{m}$, for different values of i during the outer iteration loop. Note that in this case $\Delta\tau$ can be chosen to be much greater than $\delta\tau$ because the fine details of $V(S, \tau)$ at each time step of a temporal integration is not required in the present example.

5 Newton's Linearisation

Alternatively, a small perturbation may be applied to (2), defined in the time interval $\tau \in (T_i, T_{i+1}]$, which leads to

$$\{\frac{\partial}{\partial\tau} - (A\prime(V)\frac{\partial^2 V}{\partial S^2} + A(V)\frac{\partial^2}{\partial S^2} + rS\frac{\partial}{\partial S} - r)\}\delta V$$

$$= -\{\frac{\partial V}{\partial\tau} - (A(V)\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV)\} \tag{8}$$

where δV is a small incremental change of V. Application of the Laplace transform to (8), defined in the interval $\tau \in (T_i, T_{i+1}]$, results to

$$l(\delta V) - \delta V(S, T_i) - (A\prime(V)\frac{\partial^2 V}{\partial S^2} + A(V)\frac{\partial^2}{\partial S^2} + rS\frac{\partial}{\partial S} - r)\}l(\delta V)$$

$$= -l(V) - V(S, T_i) - (A(V)\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV)\} \tag{9}$$

The method requires the numerical solution $l(\delta V^{(n)}(S, T_{i+1}))$ using $V^{(n)}(S, T_i)$ as the initial approximation to $V^{(0)}(S, T_{i+1})$ and is described in the algorithm below.

Algorithm P2: Parallel algorithm 2 for (3).
do i = 0,1,2,...
$T_i = i\Delta\tau$;
Initial approximation:- $V^{(0)}(S, T_{i+1}) := V^{(n)}(S, t_i)$; $k := 0$;
Iterate
$k := k + 1$; $\bar{V} := V^{(k-1)}(S, T_{i+1})$;
Compute $A(\bar{V})$; Compute $A'(\bar{V})$; Compute $A'(\bar{V})\frac{\partial^2 \bar{V}}{\partial S^2}$;
Compute $-l(\bar{V}) - V(S, T_i) - (A(\bar{V})\frac{\partial^2 \bar{V}}{\partial S^2} + rS\frac{\partial \bar{V}}{\partial S} - r\bar{V})\}$;

Parallel for $j := 1$ to $m(i)$
Solve (9) for $l(\delta V^{(k)}(S, T_{i+1}))$;
End parallel for
Compute $\delta V^{(k)}(S, T_{i+1})$ using inverse Laplace transform (7);
$V^{(k)}(S, T_{i+1} := \bar{V} + \delta V^{(k)}(S, T_{i+1})$;
Until $\|\delta V^{(k)}(S, T_{i+1})\| < \epsilon$
$n := k$;
end-do

6 Numerical Examples

The problem of European put option is solved upto the expiry date $T = 0.25$ at the strike price $k = 100$. The volatility σ is chosen as the function described and the parameters σ_0 and r are chosen to be 0.4 and 0.5 respectively. A second order finite volume method is applied to each parametric equation as given by (5) or (9). The mesh size is chosen to be $h = 320/2^9$.

A sequential computational environment is used in the tests. The approximations to $V(S, T)$ obtained by means of algorithms P1 and P2 are denoted as V_{P1} and V_{P2} respectively. Using $\Delta\tau = \frac{T}{10}, \frac{T}{20}, \frac{T}{30}, \frac{T}{40}$, the number of outer iterations or time steps required for algorithms P1 and P2 are 10, 20, 40, and 80 respectively.

The above two parallel time-domain algorithms are compared with the reference solution obtained by means of algorithm R with $\delta\tau = 1/365$, i.e. 1 day, in conjunction with the second order finite volume scheme applied along the spatial axis S. The discretisation leads to a number of tri-diagonal systems of equations due to the linearisation step at every time step, which may be solved by a direct method. The numerical solution $V(S, T)$ obtained by this temporal integration is denoted as V_R. The stopping criterion used in the linearization step is chosen as $\epsilon = 10^{-5}$.

In order to examine the efficiency of the parallel time-domain algorithms, the computational work required for solving a tri-diagonal system of equations results from a chosen mesh size is counted as one work unit. The total sequential work unit is obtained by multiplying the total number of work unit to $\bar{m}$, and the total parallel work unit is simply the total work unit plus overheads due to the calculation of inverse Laplace transform and communication.

Discrepancies in solutions, i.e. $\|V_R - V_{P1}\|$ and $\|V_R - V_{P2}\|$ using various $\Delta\tau$, are presented in Fig. 1 and 2. In general the discrepancy levels off when $m \geq 8$, which suggests that the use of more terms in the inverse Laplace transform at a fixed value of $\Delta\tau$ has no effect on the accuracy. On the other hand smaller $\Delta\tau$ produces smaller discrepancy at the expense of requiring more work unit as recorded in Table 1. Furthermore the work unit required by using algorithm P2 is less than that of algorithm P1, and there is no sudden increase of work when $\bar{m} = 12$.

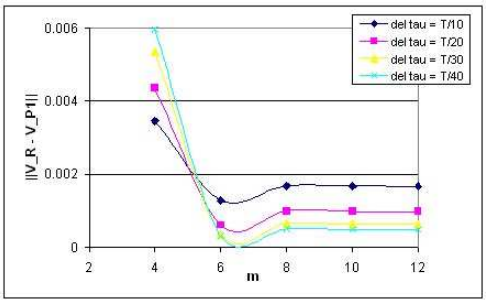

Fig. 1. Discrepancies of solutions: $\|V_R - V_{P1}\|$.

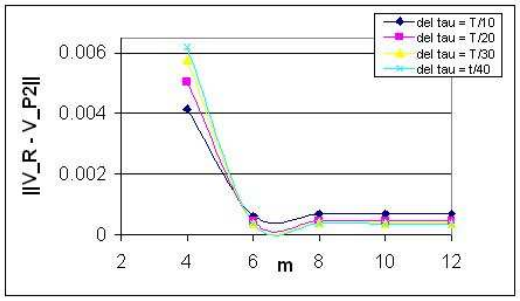

Fig. 2. Discrepancies of solutions: $\|V_R - V_{P2}\|$.

Table 1. Work units comparison (V_R requires 246 work units).

$\bar{m}$ / $\Delta\tau$	4	6	8	10	12
	Algorithm P1				
$9.125\delta\tau$	58	58	58	58	180
$4.5626\delta\tau$	103	103	103	103	123
$2.28125\delta\tau$	177	177	177	177	186
$1.140625\delta\tau$	326	326	326	326	327
	Algorithm P2				
$9.125\delta\tau$	43	43	43	43	43
$4.5626\delta\tau$	83	70	71	71	71
$2.28125\delta\tau$	134	126	126	126	126
$1.140625\delta\tau$	249	245	220	214	213

7 Conclusions

Two linearisation methods were used in conjunction with the Laplace transform method for non-linear Black-Scholes models. Work unit counts of the numerical experiments suggest that the present technique has advantages in solving nonlinear option pricing problems using parallel or distributed computing environment. One such advantage is the use of a larger time step, i.e. $\Delta\tau$, when the fine details at intermediate time steps of the time interval (T_i, T_{i+1}) are not required. Parallelisation is introduced by solving in parallel a number of parametric problems, each of which defines in the interval (T_i, T_{i+1}), i=1,2,..., $T/\delta\tau$, in the Laplace space. Note that as $\Delta\tau$ approaches $\delta\tau$ the tranform into Laplace space does not show advantages as can be seen from the results in Table 1. Therefore fine details on a fine time step should not be computed by means of Laplace transform method. Instead fine details within the time interval (T_i, T_{i+1}), for all values of i, may be obtained in parallel using a temporal integration method. Effectively the present algorithm provides initial conditions for every interval (T_i, T_{i+1}), i=1,2,..., $T/\delta\tau$. As a result fine details of the time interval (T_i, T_{i+1}) are decoupled from other time intervals and may be obtained independently with a smaller time-step, say $\delta\tau$.

References

[BarSon98] Barles, G. and Soner, H.M.: Option pricing with transaction costs and a nonlinear Black-Scholes equation. Finance Stochast, **2**, 369–397 (1998)

[BoyVor73] Boyle, P. and Vorst, T.: Option replication in discrete time with transaction costs. Journal of Finance, **47**, 271–293 (1973)

[Cra96] Crann, D.: The Laplace transform: Numerical inversion for computational methods. Technical Report No. 21, University of Hertfordshire, UK (1996)

[CDL98] Crann, D., Davies, A.J., Lai, C.-H., and Leong, S.W.: Time domain decomposition for European options in financial modelling. In: Mandel, J., Farhat, C., and Cai, X.-C. (eds) Proceedings of the 10th International Conference on Domain Decomposition Methods. American Mathematical Society (1998)

[ParAve94] Pars, A. and Avellaneda, M.: Dynamic hedging portfolios for derivative securities in the presence of large transaction costs. Appl. Math. Finance, **1**, 165–193 (1994)

[Ste70] Stehfest, H.: Numerical inversion of Laplace transforms. Comm ACM, **13**, 47–49 (1970)

[Wid46] Widder, D.V.: The Laplace Transform. Princeton University Press, Princeton (1946)

[Wil93] Wilmott, P., Howison, S. and Dewynne, J.: The Mathematics of Financial Derivatives. Press Syndicate of the University of Cambridge, New York (1993)