

## Time Domain Decomposition for European Options in Financial Modelling

Diane Crann, Alan J. Davies, Choi-Hong Lai, and Swee H. Leong

### 1. Introduction

Finance is one of the fastest growing areas in modern applied mathematics with real world applications. The interest of this branch of applied mathematics is best described by an example involving shares. Shareholders of a company receive dividends which come from the profit made by the company. The proceeds of the company, once it is taken over or wound up, will also be distributed to shareholders. Therefore shares have a value that reflects the views of investors about the likely dividend payments and capital growth of the company. Obviously such value will be quantified by the share price on stock exchanges. Therefore financial modelling serves to understand the correlations between asset and movements of buy/sell in order to reduce risk. Such activities depend on financial analysis tools being available to the trader with which he can make rapid and systematic evaluation of buy/sell contracts. There are other financial activities and it is not an intention of this paper to discuss all of these activities. The main concern of this paper is to propose a parallel algorithm for the numerical solution of an European option.

This paper is organised as follows. First, a brief introduction is given of a simple mathematical model for European options and possible numerical schemes of solving such mathematical model. Second, Laplace transform is applied to the mathematical model which leads to a set of parametric equations where solutions of different parameteric equations may be found concurrently. Numerical inverse Laplace transform is done by means of an inversion algorithm developed by Stehfast [4]. The scalability of the algorithm in a distributed environment is demonstrated. Third, a performance analysis of the present algorithm is compared with a spatial domain decomposition developed particularly for time-dependent heat equation. Finally, a number of issues are discussed and future work suggested.

### 2. European Options

One simple and interesting financial model known as the European Option has two types of contracts available namely, call options and put options. The holder

---

1991 *Mathematics Subject Classification*. Primary 65D99; Secondary 65M06.

*Key words and phrases*. Time Domain Decomposition, Financial Applications.

Scalability tests in this paper were performed by the first two authors.

of a call option has the right, at a prescribed time known as the expiry date, to purchase a prescribed asset for a prescribed amount usually known as the strike price. While the other party of the contract must sell the asset if the holder chooses to buy it. On the other hand, the holder of a put option has the right, at the expiry date, to sell the prescribed asset at the strike price. While the other party of the contract must buy the asset if the holder chooses to sell it [6]. This section only examines a European call option. The stochastic background of the equation is not discussed in this paper and readers should consult [6].

Let  $v(x, t)$  denotes the value of an option where  $x$  is the current value of the underlying asset and  $t$  is the time. The value of the option depends on  $\sigma(t)$ ,  $E$ ,  $T$  and  $r(t)$  which are, respectively, known as the volatility of the underlying asset, the strike price, the expiry time and the interest rate. The Black-Scholes analysis for one independent variable leads to the famous Black-Scholes equation [6],

$$(1) \quad \frac{\partial v}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 v}{\partial x^2} + rx \frac{\partial v}{\partial x} - rv = 0 \in \Omega^+$$

where  $\Omega^+ = \{x : x \geq 0\}$ . In order to describe a European call option, boundary conditions and final conditions are required. Since the call option is worthless at  $x = 0$  even if there is a long time to reach expiry, therefore it is sensible to have  $v(0, t) = 0$ . Since the asset price increases without bound, therefore it becomes likely that the option will be exercised and the magnitude of the strike price becomes less important. Therefore it is sensible to have  $v(s, t) \sim s$  as  $s \rightarrow \infty$ . At expiry, if  $x > E$  then one should exercise the call option, i.e. to hand over an amount  $E$  to obtain an asset with  $x$ . However, if  $x < E$  at expiry, one should not exercise the call option. Since the expiry date is in the future, the final condition  $v(x, T) = \max(x - E, 0)$  must be imposed. The solution  $v$  for  $t < T$  is required.

The financial interpretation of the above model is as follows. First, the difference between the return on an option portfolio, which involves the first two terms, and the return on a bank deposit, which involves the last two terms, should be zero for a European option. Second, the only parameter that affects the option in a stochastic way is the volatility  $\sigma(t)$  which measures the standard deviation of the returns.

Since (1) is a backward equation, one can transform it to a forward equation by using  $\tau = T - t$  and it leads to,

$$(2) \quad \frac{\partial v}{\partial \tau} = \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 v}{\partial x^2} + rx \frac{\partial v}{\partial x} - rv \in \Omega^+$$

subject to boundary conditions  $v(0, \tau) = 0$  and  $v(s, \tau) \sim s$  as  $s \rightarrow \infty$  and initial conditions  $v(x, 0) = \max(x - E, 0)$ .

An analytic solution may be derived if a change of variable is made where the Black-Scholes equation is converted to a time-dependent heat conduction equation with constant coefficients [6]. However a field method, such as finite volume methods, is of more interest for two reasons. First, there are many examples in multi-factor models such that a reduction of the time dependent coefficient to a constant coefficient heat equation is impossible. Hence analytic form of solutions cannot be found. Second, the computational environment at Greenwich is based on the finite volume code PHYSICA [1] which is the main research and development code for multi-physics work. The code has capability of solving unsteady diffusion, convection and radiation type of equations. Financial modelling typically requires large number of simulations and hence computing resources and efficiency of

algorithms are very important in order to make evaluation and decision before the agreement of a contact. With the present day high performance computing and/or distributed computing, parallel algorithms offer efficient numerical solutions to the equation given by (2).

### 3. Time Domain Decomposition

For time varying  $\sigma(t)$  and  $r(t)$ , it is possible to make suitable coordinate transformation to the Black-Scholes equation in order to obtain a time independent like heat equation [6]. Hence the method describes in this section may then be applied. Here, a method is described which focuses on time independent coefficients  $\sigma$  and  $r$ . Taking Laplace transform of (2) and taking integration by parts to the left-hand-side of the transformed equation, one obtains the parametric equation

$$(3) \quad \frac{1}{2}\sigma^2x^2\frac{d^2u}{dx^2} + rx\frac{du}{dx} - (r + \lambda_j)u = v(x, 0) \in \Omega^+$$

subject to boundary conditions  $u(0; \lambda_j) = 0$  and  $u(s; \lambda_j) = \frac{s}{\lambda_j}$ . Here  $u(x, \lambda_j)$  is the Laplace transform of  $v(x, t)$  and  $\lambda_j$  is a discrete set of transformation parameters defined by

$$(4) \quad \lambda_j = j\frac{\ln 2}{\tau}, \quad j = 1, 2, \dots, m$$

where  $m$  is required to be chosen as an even number [5]. An approximate inverse Laplace transform [4] may be used to retrieve  $v(x, t)$  according to

$$(5) \quad v(x, \tau) \approx \frac{\ln 2}{\tau} \sum_{j=1}^m w_j u(x, \lambda_j)$$

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 weight factor. Each of the above  $m$  parametric equations may be rewritten as

$$(6) \quad \frac{d}{dx}\left(\frac{1}{2}\sigma^2x^2\frac{du}{dx} + \frac{d}{dx}((r - \sigma^2)xu) - (2r - \sigma^2 + \lambda_j)u = v(x, 0)\right)$$

The computational domain has a uniform mesh and finite volume method is applied to (6) which leads to

$$(7) \quad \int_S \left(\frac{1}{2}\sigma^2x^2\frac{du}{dx} + (r - \sigma^2)xu\right) ds - \int_{\Omega} (2r - \sigma^2 + \lambda_j)u d\Omega = \int_{\Omega} v(x, 0) d\Omega$$

The resulting system of linear equations is solved in a local area network which consists of  $P$  workstations. There are two possible implementations as follow.

First, the solution at a particular time  $\tau$  is being sought. For the case when  $m = P$ , one would expect ideal load balancing. The total computing time,  $t_{A_1}$ , using the present scheme can be estimated as  $t_A = t_1 + t_a/P$  where  $t_1$  is the computing time for solving one parametric equation and  $t_a$  is the corresponding computing time for numerical inverse Laplace transforms given by (5) and is assumed to be equally spreaded across the  $P$  workstations. For the case when  $m > P$ , one would expect just a slight out of load balance for the reason that  $m$  is

possibly not an integral multiple of  $P$ . It is possible to estimate the total computing time as

$$(8) \quad t_{A_1} = \lceil \frac{m}{P} \rceil t_1 + t_a/P$$

The case  $m < P$  is not of interest because the active workstations become a subset of the local area network.

Second, the solutions at  $P$  particular times  $\tau_k, k = 1, 2, \dots, P$  are being sought. In this situation, each workstation looks after the solutions of  $m$  parametric equations and the corresponding inverse Laplace transform at a particular time  $\tau_k$ . Hence the total computing time,  $t_{A_2}$ , may be estimated as

$$(9) \quad t_{A_2} = mt_1 + t_a$$

In order to check the scalability of the algorithm, a cell-centred finite volume scheme is applied to the constant coefficient heat equation,

$$(10) \quad \nabla^2 u = \frac{1}{k} \frac{\partial u}{\partial t} \in -1 < x < 1, -1 < y < 1$$

subject to unit boundary conditions along the whole boundary and zero initial condition. A uniform 16 x 16 grid where the set of discrete equation is solved by Gauss-Seidel iteration. Solutions at eight time values,  $\tau = 0.1, 0.2, 0.5, 1, 2, 4, 10$  and  $20$ , are sought. For the purpose of demonstration, a network of 4 T800 transputers were used as the hardware platform. The computing times for  $P = 1, 2$  and  $4$  are respectively 2537, 1309 and 634 seconds. The speed-up ratio for using two and four processors are thus 1.94 and 4 respectively. More results about this test problem can be found in [2] and experience shown in the paper suggests that  $m = 8$  provides sufficient accuracy for the model test. Note that the value of  $m$  determines the accuracy of the inverse Laplace transform and hence it depends on the mesh size or the number of grid point. In general,  $m = \sqrt{N}/2$  where  $N$  is the total number of grid points in a two-dimensional problem [2]. Note also that the scalability property as shown in this section also applies to eqn (6).

#### 4. A Comparison with Spatial Domain Decomposition

In order to find out the suitability of the proposed algorithm for a distributed computing environment, a comparison with a spatial domain decomposition method is examined in this section. The spatial domain decomposition method is similar to the one developed by Dawson et al [3] for unsteady heat conduction equation. The problem described in (2) is partitioned into  $P$  subdomains so that a coarse mesh of mesh size  $H = s/P$  is imposed with interior boundary of the subdomains being the same as the nodal points of the coarse mesh. In order to determine the interior boundary values of each of the subdomains, an explicit scheme derived from using a central difference method along the spatial axis and a forward difference method along the temporal axis is as follows,

$$(11) \quad v_i^n = \left(\frac{1}{2}\sigma^2 x_i^2 \frac{\Delta t}{H^2} - rx_i \frac{\Delta t}{2H}\right)v_{i-1}^{n-1} + \left(\frac{1}{2}\sigma^2 x_i^2 \frac{\Delta t}{H^2} + rx_i \frac{\Delta t}{2H}\right)v_{i+1}^{n-1} + (1 - \sigma^2 x_i^2 \frac{\Delta t}{H^2} - r\Delta t)v_i^{n-1}$$

Here subscripts  $i$  denote the mesh points on the coarse mesh and superscripts  $n$  denotes the time step at  $t = n\Delta t$ . The choice of  $\Delta t$  must satisfy the coarse grid restriction for an explicit scheme which is

$$(12) \quad \Delta t \leq \min_{x_i} \left( \frac{\sigma^2 x_i^2}{H^2} + r \right)^{-1} \leq (\sigma^2(P-1)^2 + r)^{-1}$$

Note here that  $\Delta t$  should be of the order of  $h^2$  where  $h$  is the grid size of the fine mesh. Therefore the total number of time steps involved in the present calculation is  $T(\sigma^2(P-1)^2 + r)$ .

It is reasonable to assume that the computing time for obtaining the solution at a new time step using the fine mesh and a finite difference scheme is the same as the computing time for obtaining the solution of a parametric equation given in (3). Therefore the computing time for marching one time step forward using the classical spatial domain decomposition with  $P$  subdomains on  $P$  processors is  $t_1/P$ . The total parallel computing time can be estimated as

$$(13) \quad t_B = T(\sigma^2(P-1)^2 + r) \left( \frac{t_1}{P} + \frac{t_b}{P} \right)$$

where  $t_b$  is the overheads for obtaining interior boundary conditions and is assumed to be equally spreaded across the  $P$  workstations.

It is natural to require  $t_{A_1} < t_B$  for any advantage of the proposed time domain decomposition scheme to be happened when comparing with the classical spatial domain decomposition, and hence one would require  $\lceil \frac{m}{P} \rceil t_1 < \frac{T(\sigma^2(P-1)^2 + r)}{P} t_1$ , i.e.

$$(14) \quad \lceil \frac{m}{P} \rceil < \frac{T(\sigma^2(P-1)^2 + r)}{P}$$

When  $m$  is an integral multiple of  $P$ , one obtains

$$(15) \quad m < T(\sigma^2(P-1)^2 + r)$$

In other words, one requires the number of parameters in inverse Laplace transforms to be smaller than the number of time steps involved in spatial domain decomposition. Note that the dominant term in the inequality is obviously  $T$ . Since typical values of  $m$  is usually much smaller than  $s/h$  for an acceptable accuracy of inverse Laplace transform [2], the inequality (15) is easily satisfied with typical ranges of  $0.05 < \sigma < 0.45$  and  $0.6 < r < 1.1$ . Therefore for a given value of  $P$ , the inequality offers only a very mild restriction. Hence the time domain decomposition method proposed in this paper has advantage over the classical spatial domain decomposition method for European call options.

From (15), we have  $mt_1 < T(\sigma^2(P-1)^2 + r) \frac{t_1}{P}$  which combines with (13) to give

$$(16) \quad \frac{t_{o_1}}{t_o} < \frac{T(\sigma^2(P-1)^2 + r)}{P}$$

The ratio governs the number of time levels  $\tau_k$  to be allowed in each workstation in order that the inequality  $t_{A_2} < t_B$  remains valid. Supposing each parametric equation has  $N$  grid points involved in the discretisation, the total number of floating point operations involved in (5) can be easily counted as  $2N + Nm$ . Also, supposing some of floating point operations in (11) can be done once for all, the total number of floating point operations for the update of interior boundary conditions

is counted as  $18P$ . Hence (16) become

$$(17) \quad m < \frac{18T}{N}(\sigma^2(P-1)^2 + r) - 2$$

It can be checked that the inequality is easily satisfied with the above typical ranges of  $\sigma$  and  $r$ .

## 5. Conclusions

A parallel algorithm based on Laplace transform of the time domain into a set of parametric equations is developed for European call option. Distributed computing may be applied to solve the parametric equations concurrently. An inverse Laplace transform based on Stehfast method is applied to retrieve the solution. The method is compared with classical spatial domain decomposition. A preliminary analysis shows that the proposed method has advantage over the spatial domain decomposition method.

## References

1. Centre for Numerical Modelling and Process Analysis - University of Greenwich, *Physica - user guide*, 1995, Beta Release 1.0.
2. AJ Davies, J Mushtaq, Radford LE, and Crann D, *The numerical Laplace transform solution method on a distributed memory architecture*, Applications of High Performance Computing in Engineering V (H Power and JJ Casares Long, eds.), Computational Mechanics Publications, 1997, pp. 245–254.
3. CN Dawson, Q Du, and TF Du Pont, *A finite difference domain decomposition algorithm for the numerical solution of the heat equation*, Math Comp **57** (1991), 63–67.
4. H Stehfast, *Numerical inversion of Laplace transforms*, COMM ACM **13** (1970), 47–49.
5. D V Widder, *The Laplace transform*, Princeton University Press, Princeton, 1946.
6. P Wilmott, S Howison, and J Dewynne, *The mathematics of financial derivatives*, Press Syndicate of the University of Cambridge, New York, 1995.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF HERTFORDSHIRE, HATFIELD, UK  
*E-mail address:* D.Crann@herts.ac.uk

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF HERTFORDSHIRE, HATFIELD, UK  
*E-mail address:* A.J.Davies@herts.ac.uk

SCHOOL OF COMPUTING & MATHEMATICAL SCIENCES, UNIVERSITY OF GREENWICH,  
 WELLINGTON STREET, WOOLWICH, LONDON SE18 6PF, UK  
*E-mail address:* C.H.Lai@greenwich.ac.uk

QUANTITATIVE RESEARCH AND TRADING GROUP, THE CHASE MANHATTAN BANK, 125 LONDON  
 WALL, LONDON EC2Y 5AJ, UK  
*E-mail address:* Swee.Leong1@chase.com