A distributed Laplace transform algorithm for European options

A. J. Davies\textsuperscript{1}, M. E. Honnor\textsuperscript{1}, C.-H. Lai\textsuperscript{2}, A. K. Parrott\textsuperscript{2} & S. Rout\textsuperscript{2}
\textsuperscript{1}Department of Physics, Astronomy and Mathematics, University of Hertfordshire, UK
\textsuperscript{2}School of Computing and Mathematical Sciences, University of Greenwich, UK

Abstract

A distributed algorithm is developed to solve the Black-Scholes equation in the hedging of portfolios. The Black-Scholes equation is parabolic in time and such problems have been shown to be ideally suited to the use of the Laplace transform rather than a finite difference time-stepping method. The application of the Laplace transform to the Black-Scholes equation gives rise to an elliptic problem and any suitable solver may be used in the transform space. We shall use the finite volume method. The numerical inversion of the Laplace transform is effected by Stehfest’s method which requires a solution for each transform parameter. Since these solutions are obtained independently the method is an excellent candidate for an implementation in a distributed computing environment.

Keywords: option pricing, Black-Scholes equation, Laplace transform, nonlinear volatility, finite volume method.

1 Introduction

Financial modelling in the area of option pricing involves the understanding of hedge assets and portfolios in order to control the risk due to movements in share prices. Such activities depend on financial analysis tools being available to traders with which they can make rapid and systematic evaluations of buy/sell contracts. In turn, analysis tools rely on fast numerical algorithms for the solution of financial mathematical models. There are many different financial activities apart from shares buy/sell activities. However it is not the intention of
this paper to discuss various financial activities. The main aim of this paper is to propose and discuss a distributed algorithm for the numerical solution for European options. Both linear and non-linear cases are considered.

The algorithm is based on the concept of the Laplace transform and its numerical inverse. First, a Laplace transform is applied to the linear Black-Scholes equation, which leads to a set of mutually independent linear ordinary differential equations. The set of differential equations may then be solved concurrently in a distributed environment. The scalability of the algorithm has been studied theoretically in [1] and [2]. This paper provides numerical tests to demonstrate the effectiveness of the algorithm for financial analysis. Time dependent functions for volatility and interest rates are also discussed. Second, an extension is given of the algorithm for non-linear Black-Scholes equations where the volatility is a function of the option value. The Laplace transform is applied to a linearisation of the non-linear Black-Scholes equation. A set of mutually independent linear ordinary differential equations is obtained, and these equations may be solved in a distributed computing environment. The numerical inverse Laplace transform is then obtained within an outer iteration loop. The convergence behaviour of the algorithm is discussed. The algorithm relies on fast computation of the numerical inverse Laplace transform. The main goal of this paper is to demonstrate the effectiveness of using an inverse Laplace transform in applications to linear and nonlinear Black-Scholes equations. This paper will also examine the various computational issues of such a numerical inverse in terms of distributed computing.

2 The 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 two stochastic parameters, namely the volatility \( \sigma \) and the interest rate \( r \), of the Black-Scholes equation:

\[
\frac{\partial v}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 v}{\partial S^2} + rS \frac{\partial v}{\partial S} - rv = 0, \quad (S,t) \in \Omega^+ \times [T,0)
\] (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 a text such as Wilmot et al. [3].

In this paper, attention is paid to European options, which mean 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 options, 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 conditions:
\( v(0,t) = k \exp[-r(T-t)], \quad v(L,t) = 0 \)

where \( L \) is usually a large value. At expiry, if \( S < k \) then the call option should be exercised, i.e. the handing over of an amount \( k \) to obtain an asset with value \( S \).

However, if \( S > k \) at expiry, then the option should not be exercised 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.

The financial interpretation of the above model is as follows. 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. Note that within a given short period, it is possible to assume the interest rate to be a constant rather than a stochastic parameter.

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

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

subject to the initial condition:

\( V(S,0) = \max\{k - S, 0\} \)

and the boundary condition:

\( V(0,\tau) = k \exp[-r\tau], \quad V(L,\tau) = 0 \).

Analytic solutions may be derived if a change of variable is made where the Black-Scholes model is converted to a time-dependent heat conduction equation with constant coefficients [3]. However a field method, such as the finite volume method, 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 forms of solution cannot be found. Second, financial modelling typically requires large numbers 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.

3 The distributed algorithm

It should be noted that in a field method for time dependent partial differential equations, the time step length is often restricted by a stability criterion and by the truncation errors in the discretised approximation of the time derivatives. On the other hand, concurrent computation of all time steps is almost impossible. Hence a distributed algorithm does not apply in this sense. Since the solutions at intermediate steps of equation (2) are usually not of interest, it is possible to apply the Laplace transform [4] to equation (2) and to reduce it to a number of mutually independent boundary-value problems. Let
be the Laplace transform of the function \( V(S, \tau) \), then the Laplace transform of equation (2) leads to

\[
\frac{1}{2} \sigma^2 S^2 \frac{d^2 U}{dS^2} + rS \frac{dU}{dS} - (r + \lambda)U = -V(S, 0), \quad S \in \Omega^+
\]  

subject to the boundary condition

\[
U(\lambda; 0) = \frac{k}{\lambda + r}, \quad U(\lambda; L) = 0
\]

where \( \lambda \in \{\lambda_j\} \) which is a finite set of transformation parameters defined by

\[
\lambda_j = j \frac{\ln 2}{T}, \quad j = 1, 2, \ldots, m
\]  

where \( m \) is required to be chosen to be an even number [5]. Therefore, the original problem (2) is converted to \( m \) independent parametric boundary-value problems as described by equation (3), and these problems may be distributed and solved independently in a distributed environment which consists of a number of processors linked by a network. From experience, the value of \( m \) is usually a small even number not larger than ten [1], [6]. Numerical experiments in sections 4 and 5 also confirm such experience.

In order to retrieve \( V(S, T) \), the approximate inverse Laplace transform due to Stehfest [5] is given by

\[
V(S, T) \approx \frac{\ln 2}{T} \sum_{j=1}^{m/2} w_j U(\lambda_j; S).
\]  

Here the weights are given by

\[
w_j = (-1)^{m/2 + j} \sum_{k=\lfloor 1 + j/2 \rfloor/2}^{\min(j, m/2)} \frac{k^{m/2}(2k)!}{(m/2 - k)!k!(k-1)!(j-k)!(2k-j)!}.
\]

This approximate inverse Laplace transform is not necessarily the most accurate. The authors select the Stehfest method in view of their previous experience with the method for linear problems for diffusion problems [2] and we wish to investigate the application of the inverse method to option pricing problems. We note that Davies and Martin [7] discuss a variety of numerical inversion methods and conclude that, for parabolic problems of the diffusion-type, the Stehfest method is as good as any other in terms of accuracy, efficiency and ease of application.

For time varying \( \sigma(t) \) and \( r(t) \), it is possible to make a suitable coordinate transformation to the Black-Scholes equation in order to obtain a time
independent diffusion-like equation [3]. Hence the above Laplace transform method may still be applied.

4 Examples with linear volatility

An example of a European put option is given in this section. The spatial domain is chosen to be $\Omega^+ = \{ S : 0 \leq S \leq 320 \}$, the strike price being $k = 100$ and the expiry date being $T = 0.25$ (three months). The two parameters $\sigma$ and $r$ are chosen to be 0.4 and 0.5 respectively, throughout the simulation period. A second order finite volume method is applied to each parametric equation as given by (3). The mesh size is chosen to be $h = 320 / 2^9$.

As a comparison the forward Black-Scholes equation as given by equation (2) is solved by means of an Euler time-marching scheme along the temporal axis with time step-length being $1/365$, i.e. 1 day, in conjunction with the above finite volume scheme applied along the spatial axis $S$. The discretisation leads to a tri-diagonal system of equations at each time-step, which may then be solved by a direct method. The solution obtained in this case is denoted as $V_{TI}$. Then the Laplace transformed set of equations are solved, sequentially in the same computational environment, with different values of $m$. An approximation to $V(S,T)$ corresponds to each value of $m$ is found by using the inverse Laplace transform as described in section 3 and is denoted as $V_{IL}$. Discrepancies between solutions, $\|V_{TI} - V_{IL}\|_2$, are recorded and shown in Table 1 for comparisons. Timings were obtained on a Sun Ultra-5 workstation using a Fortran90 program which implements the above two methods.

<table>
<thead>
<tr>
<th>$m$</th>
<th>time, $V_{IL}$</th>
<th>$|V_{TI} - V_{IL}|_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.006</td>
<td>1.0767</td>
</tr>
<tr>
<td>4</td>
<td>0.009</td>
<td>0.0812</td>
</tr>
<tr>
<td>6</td>
<td>0.014</td>
<td>0.0111</td>
</tr>
<tr>
<td>8</td>
<td>0.017</td>
<td>0.0037</td>
</tr>
<tr>
<td>10</td>
<td>0.018</td>
<td>0.0032</td>
</tr>
<tr>
<td>12</td>
<td>0.021</td>
<td>0.0032</td>
</tr>
<tr>
<td>14</td>
<td>0.028</td>
<td>0.0032</td>
</tr>
<tr>
<td>16</td>
<td>0.028</td>
<td>0.0032</td>
</tr>
<tr>
<td>18</td>
<td>0.035</td>
<td>0.0032</td>
</tr>
</tbody>
</table>

The timing of each run observed in this example consists of two parts. First, then time for the second order finite volume solver, second, the overheads due to the computation of inverse Laplace transforms. Dividing the timings for the
inverse Laplace algorithm gives a crude estimation on the distributed processing time. Suppose there are as many processors available as the value of $m$, the scalability of the algorithm can still be easily observed from the sequential timings recorded in Table 1 using the above crude estimation of the corresponding distributed timings. The discrepancy $\|V_{II} - V_{II}\|$ approaches an asymptotic value of 0.0032 when $m \geq 8$. Therefore it is not necessary to take $m$ very much larger than ten. This result confirms the previous tests on a linear heat conduction problem [2].

5 Solving non-linear models

Very often, over a short period of time the interest rate, $r$, is fixed while the volatility, $\sigma$, varies. The volatility may be a function of the transaction costs [8], the second derivative of the option value [9], or, in some cases, the solution of a nonlinear initial-value problem [6]. In order to develop the nonlinear solver in this section we use the volatility proposed by Boyle and Vorst [8]:

$$\sigma = \sigma_0 \sqrt{1 + A}$$

where $A$ is the proportional transaction cost scaled by $\sigma_0$ and the transaction time. Here the authors adopted a heuristic approach in which the transaction cost is related to the option value and follows a Gaussian distribution. In order to demonstrate the inverse Laplace transform technique for nonlinear problems, a sine function is used to produce the effect of a pulse like distribution instead of a Gaussian distribution. Therefore the proportional transaction cost, $A$, may be replaced by a function of the option value such as

$$A = \sin \left( \frac{V(S, \tau)}{\max \{V(S, \tau)\}} \pi \right)$$

This volatility is used in the subsequent numerical tests.

The forward Black-Scholes model as given in equation (2) may be re-written as

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

where $A(V) = \frac{1}{2} \sigma(V)^2 S^2$. Two linearisation techniques can be applied to equation (6).

First, the coefficient $A$ is frozen at the given approximation, say, $\bar{V}$, during each step of the nonlinear iteration. A Laplace transform can then be applied to equation (6), which leads to

$$A(\bar{V}) \frac{d^2 U}{dS^2} + rS \frac{dU}{dS} - (r + \lambda)U = -V(S, 0)$$

where $U = U(V)$. Therefore the algorithm for solving the Black-Scholes equation with non-linear coefficients as in equation (6) may be written as:
Algorithm: Frozen coefficient.
Initial approximation: $ V := \bar{V} $;
Iterate
$ \bar{V} := V $; \{Store for comparison\}
Compute $ A(\bar{V}) $;
Parallel for $ j := 1 $ to $ m $)
Solve (7) for $ l(V(\lambda_j; S)) $;
End Parallel for
Compute inverse Laplace: $ V(S, T) $;
Until $ \| V - \bar{V} \| < \varepsilon $.

In order to solve equation (7), we can employ the same finite volume technique described above.

Second, Newton’s method may be applied to equation (2) which leads to
\[
\begin{equation}
\left( A'(V) \frac{\partial^2 V}{\partial S^2} + A(V) \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - r \right) e
\end{equation}
\]
where $ V + e $ gives a new approximation. A Laplace transform may then be applied to equation (8), which leads to
\[
\begin{align*}
\left( A'(V) \frac{\partial^2 V}{\partial S^2} + A(V) \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - r \right) l(e)
\end{align*}
\]
\[
= -l(V) - V(S, 0) - \left( A(V) \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV \right)
\]

Similar to the above algorithm for a fixed coefficient approximation for $ A $, we have the algorithm:

Algorithm: Newton’s method.
Initial approximation: $ V := \bar{V} $;
Iterate
$ \bar{V} := V $; \{Store for comparison\}
Compute $ A(\bar{V}) $; Compute $ A'(\bar{V}) $;
Compute $ l(\bar{V}) $ and the remaining terms in the r.h.s. of equation (9);
Parallel for $ j := 1 $ to $ m $)
Solve (9) for $ l(e(\lambda_j; S)) $;
End Parallel for
Compute inverse Laplace: $ e(S, T) $;
\[ V := V_0 + \varepsilon \]
Until \[ \|e\| < \varepsilon \]

The two linearisation techniques are used in conjunction with the Laplace transform. As an example of the European put option the same problem described in section 4 is used. The volatility, \( \sigma \), is chosen as the Boyle and Vorst function described above where the parameters \( \sigma_0 \) and \( r \) are chosen to be 0.4 and 0.5, respectively, throughout the simulation period. A second order finite volume method is applied to each parametric equation as given by equations (7) or (9). The mesh size is chosen to be \( h = \frac{320}{2^m} \).

The Laplace transformed set of equations for each of the linearisation methods are solved, sequentially, in the same computational environment with different values of \( m \) and an approximation to \( V(S,T) \) corresponds to each value of \( m \) is found by using the inverse Laplace transform as described in section 3. The approximations obtained by means of the frozen coefficient and Newton’s methods are denoted as \( V_{IL}(F) \) and \( V_{IL}(N) \) respectively.

Table 2: Computational work and discrepancy comparisons. NB, the number of iterations for \( V_{IT} \) is 293.

<table>
<thead>
<tr>
<th>( m )</th>
<th>( V_{IL}(F) ) iterations</th>
<th>( |V_{IT} - V_{IL}(F)|_2 )</th>
<th>( V_{IL}(N) ) iterations</th>
<th>( |V_{IT} - V_{IL}(N)|_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9</td>
<td>0.1620</td>
<td>7</td>
<td>0.1560</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>0.0255</td>
<td>8</td>
<td>0.0141</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>0.0127</td>
<td>7</td>
<td>0.0046</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>0.0121</td>
<td>7</td>
<td>0.0045</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>0.0121</td>
<td>7</td>
<td>0.0045</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>0.0121</td>
<td>8</td>
<td>0.0045</td>
</tr>
<tr>
<td>14</td>
<td>large</td>
<td>0.0121</td>
<td>7</td>
<td>0.0045</td>
</tr>
</tbody>
</table>

The above two distributed algorithms are compared with the linearised forward Black-Scholes equation given in equation (8) solved by means of an Euler time-marching scheme along the temporal axis with time step-length being 1/365, i.e. one day, in conjunction with the second order finite volume scheme applied along the spatial axis \( S \). The discretisation leads to a number of tridiagonal systems of equations due to the linearisation step at every time step, which may be solved by a direct method. The solution obtained in this case is denoted as \( V_{IT} \). Discrepancies in solutions, \( \|V_{IT} - V_{IL}(F)\|_2 \) and \( \|V_{IT} - V_{IL}(N)\|_2 \), and the number of nonlinear iterations are recorded in Table 2 for comparisons. A F77 program was written, which implements the above two linearisation methods, and run on a COMPAQ laptop. Defining one work unit as the
computational work to solve a tri-diagonal equation, the total sequential work unit is obtained by multiplying the number of nonlinear iterations and $m$ and the total parallel work unit is simply the number of nonlinear iterations plus overheads.

The other feature of the linearisation is that the use of frozen coefficients has no advantage over Newton’s method. Finally, the discrepancies of the solutions, $\|V_{TI} - V_{IL}(F)\|_2$ and $\|V_{TI} - V_{IL}(N)\|_2$, as compared with a time-stepping method applied to the linearised system for the case $h = 320/2^9$ are recorded in Figure 1. It shows that the largest discrepancies occur near the strike price. Therefore it is a resolution problem near the strike price. On the other hand the discrepancies are of the order of $10^{-3}$ which are similar to the results for linear problems.

![Figure 1: Discrepancies of solutions for nonlinear problems.](image)

6 Conclusions

A distributed algorithm for solving European options model is discussed. Numerical examples are provided for a European put option with one spatial dimension variable $S$. Timings of the linear problems were obtained on a Sun Ultra-5 workstation show the advantages of the present Laplace transform approach for option pricing. A projection of the timing to a distributed computing environment shows the scalability of the algorithm. Two linearisation methods were used in conjunction with Laplace transform for nonlinear Black-Scholes models are discussed. Work counts are also presented. The computational work suggests that the inverse Laplace techniques have
advantages in solving nonlinear option pricing problems. Further investigation into other methods of inverse Laplace transform is currently being undertaken by the authors.

References


