# Shout Options: A Framework For Pricing Contracts Which Can Be Modified By The Investor

H. Windcliff[*], P.A. Forsyth[†], and K.R. Vetzal[‡]

University of Waterloo
Waterloo, Ontario
Canada N2L 3G1[§]

November 23, 1999

**Abstract**

A shout option may be broadly defined as a financial contract which can be modified by its holder according to specified rules. In a simple example, the holder could have the right to set the strike of an option equal to the current value of the underlying asset. In such a case, the holder effectively has the right to select when to take ownership of an at-the-money option. More generally, the holder could have multiple rights along these lines, in some cases with a limit placed on the number of rights which may be exercised within a given time period (e.g. four times per year). The value of these types of contracts can be estimated by solving a system of interdependent linear complementarity problems. This paper describes a general framework for the valuation of complex types of shout options. Numerical issues related to interpolation and choice of timestepping method are considered in detail. Some illustrative examples are provided.

**Keywords:** PDE option pricing, shout options, linear complementarity, interpolation

**Running Title:** Shout options

---

[*]Department of Applied Mathematics, `hawindcliff@elora.math.uwaterloo.ca`

[†]Department of Computer Science, `paforsyt@elora.math.uwaterloo.ca`

[‡]Centre for Advanced Studies in Finance, `kvetzal@watarts.uwaterloo.ca`

[§]**Acknowledgment:** This work was supported by the Natural Sciences and Engineering Research Council of Canada, the Social Sciences and Humanities Research Council of Canada, and the Communications and Information Technology Office of Ontario. In addition, we would like to thank Rational Software for access to the Purify software development tool under the SEED program.

# 1 Introduction

An option is a financial contract which gives its holder the right to buy or sell an underlying asset for a specified price (called the strike) during a specified time period. Standard options come in two flavours: puts, which provide the right to sell, and calls, which provide the right to buy. An additional distinction is between European options and American options: the former can only be exercised at the maturity date of the option, whereas the latter may be exercised any time during the life of the contract.

In recent years, the complexity of financial contracts has increased dramatically. A feature which is sometimes included in such contracts is the ability of the holder to modify the contract according to specified rules. Contracts with this property are termed shout options. A common example is a protective floor index. In this case, the holder has the ability to set a protective floor which "locks in" a minimum amount which will be received at the maturity of the contract. The valuation and hedging of these contracts is more complex than for standard options because there is an element of uncertainty in the investor's actions. This paper describes a method for valuing complex contracts incorporating this feature of *investor modification*. Similar to an American option, the fair market value of a shout option can be determined from the solution of a system of linear complementarity problems, which can be formulated as partial differential equations (PDE) with a nonlinear penalty term [21].

Prior academic research in this area has not been extensive. Shout options have been described in the context of basic calls and puts [17] and resettable floors [7]. Both of these papers describe the valuation using simple explicit type methods (otherwise known as lattice or tree methods in the finance literature). In [7] the authors work under standard assumptions assuming constant volatility of the underlying asset price, and deal exclusively with shout options which reset the strike price to the current value of the underlying asset. These simplifications allow the authors to reduce the dimensionality of the numerical problem.

More complicated shout options are embedded in products such as "segregated funds" sold by Canadian life insurance companies. These products are typically a guarantee on the value of a mutual fund. Investors are often permitted to reset the level of the guarantee (i.e. "shout") multiple times, up to some limit within a time period (e.g. four times per year). It is also worth noting that some energy derivative contracts have recently included a feature called a swing option [14], which is similar in many respects to a complex shout option.

In this work, we will employ a numerical PDE approach for valuing shout options. The standard assumption that the volatility of the underlying asset is constant has been subjected to increasing scrutiny recently (see [2] for an overview and some empirical evidence). Using a fully numerical approach enables us to study the effect of deviations from this common assumption such as constant elasticity of variance (CEV) models [9] and implied volatility surfaces [1, 8].

Numerical schemes frequently used in the finance industry are not well-suited for valuation of these contracts. At present, standard Monte Carlo approximation techniques, which work forward in time, cannot effectively handle the optimization component of these contracts. However, a backward dynamic programming approach based on advanced Monte Carlo algorithms [6] for evaluating the Green's functions integrals has been applied to shout options [4]. This assumes that the Green's function is easily computed, and that continuous early exercise is well approximated by a limited number of exercise opportunities at specified discrete points in time. Shout options are often long term, and as a result, explicit type methods as described in [17, 7] are very expensive, owing to timestep limitations arising from stability considerations.

The objective of this article is to develop a robust method that can be used to value a wide variety of shout contracts while making minimal assumptions about the underlying stochastic process followed by the risky asset. Specifically, we do not assume that volatility has any particular functional form. We focus on the following concerns:

- It is computationally necessary to use a finite computational domain, although the actual pricing problem is posed on an infinite region. We will address the issue of the approximation of boundary conditions imposed by artificially truncating the solution domain.

- The shout feature requires interpolation of the numerical PDE solution. Various methods of interpolation and grid construction will be devised to minimize this source of error.

- Since these contracts are typically long term it is beneficial to use a high order timestepping algorithm. Here we compare the performance of several timestepping methods, including a second order BDF multi-step method.

# 2 Mathematical Model

In order to be precise, we define a shout option as follows:

**Definition:** *A* ***shout option*** *is a contract defined by the following objects:*

- *An underlying asset price process $S$ upon which the derivative security is written.*

- *A maturity time $T$ for the contract.*

- *A payoff function $g(S, K, U)$ which determines the payment made to the holder of the security at maturity which is a function of the asset level $S$ and a parameter called the strike, $K$, which can be changed (at the discretion of the holder) during the life of the contract.*

- *A maximum number of times $U_{\max}$ which the holder of the security can shout during a given time period, thereby resetting the strike $K$. We will use the discrete variable $U = 0, \ldots, U_{\max}$ to count the number of shouts used at any point in time.*

- *A function $\mathcal{F}(S, K, U, t)$ which determines how the strike $K$ is set upon shouting. In the case of a simple shout option where the strike is reset to the current asset level, this function would be defined as $K^* = S$.*

- *A shout dividend function $\mathcal{D}(S, K, U, t)$ which represents payments generated by the option upon shouting. Cases where $\mathcal{D}(S, K, U, t) < 0$ can be thought of as a fee charged for shouting.*

With this definition we see that a shout option is a contract which gives the holder the opportunity to exchange one contract for another (presumably better) contract of the same form but with less flexibility. Essentially we are faced with the task of valuing an American-style security where the effective payment received from exercising (i.e. shouting) is determined by *the value of the contract received*, plus any additional payments (positive or negative) specified by the function $\mathcal{D}(S, K, U, t)$.

We assume the existence of a risk free money market account and that the underlying asset price $S$ satisfies the stochastic differential equation

$$dS = \mu(S,t)S dt + \sigma(S,t)S dz, \tag{1}$$

where $dz$ is the increment of a Wiener process, $\mu(S,t)$ is the drift rate (which turns out to be irrelevant in this option valuation context), and $\sigma(S,t)$ is the volatility. Note that in the standard Black-Scholes setting, $\sigma(S,t)$ is a constant. This is an important special case and in this paper we will denote this constant as $\sigma_{BS}$. Returning to the general specification (1), following standard arguments (e.g. [19]) we can model the value of the shout option as a set of interdependent linear complementarity problems

$$\frac{\partial V}{\partial t} + r(t)S\frac{\partial V}{\partial S} + \frac{1}{2}\sigma(S,t)^2 S^2 \frac{\partial^2 V}{\partial S^2} - r(t)V \leq 0 \tag{2}$$

$$V^* \leq V \tag{3}$$

where one of (2-3) holds with equality. In (2-3), $r(t)$ is the risk free rate of return, $V = V(S, K, U, t)$, and we have defined $V^*$ (the value of the contract the holder receives upon shouting) as

$$V^*(S,K,U,t) = \begin{cases} V(S, \mathcal{F}(S,K,U,t), U+1, t) + \mathcal{D}(S,K,U,t) & \text{if } U+1 \leq U_{\max}, \\ -\infty & \text{otherwise.} \end{cases} \tag{4}$$

It is convenient to reformulate the linear complementarity problems (2-3) as PDEs with a nonlinear penalty term $q(V, V^*)$. The penalty function $q(V, V^*)$ is defined [21] so that $V$ satisfies (2-3) where one of (2-3) holds with equality,

$$\frac{\partial V}{\partial t} + r(t)S\frac{\partial V}{\partial S} + \frac{1}{2}\sigma(S,t)^2 S^2 \frac{\partial^2 V}{\partial S^2} - r(t)V + q(V, V^*) = 0. \tag{5}$$

At contract expiry $t = T$ we impose the terminal condition

$$V(S, K, U, T) = g(S, K, U). \tag{6}$$

For simplicity, we will restrict attention in this work to put options, where the payoff is given by

$$g(S, K, U) = \max(K - S, 0). \tag{7}$$

We emphasize that all of the methods discussed in this paper can be used with arbitrary $g(S, K, U)$.

The constraint (3) ensures that for any exercise policy adopted by the holder the writer can liquidate his current hedging strategy and have a sufficient amount to start a replicating strategy to "copy" the new security which the holder receives upon shouting. This also shows that the holder cannot increase a shout option's value merely by shouting (although he can decrease it by *not* shouting), provided that the option value is computed under the assumption that the holder will make optimal decisions regarding exercise.

Shout option contracts are often sold with the feature that the holder is allowed to shout up to a maximum of $U_{\max}$ times during a given time interval, typically one year (e.g. segregated funds). At the end of this time interval the discrete variable $U$ which counts the number of shouts used is reset to zero. If $t_i$ is a shout counter reset time, then no arbitrage considerations require that the condition

$$V(S, K, U, t_i^-) = V(S, K, 0, t_i^+) \tag{8}$$

be satisfied for all $U$ where $t_i^-$ and $t_i^+$ are the times the instant before and after the reset time $t_i$.

We can gain insight regarding the mechanics of a shout option using a geometric interpretation. The discrete variable $U$ represents the number of shouts used since the inception of the contract. Therefore, planes of $U = const.$ represent contracts which have the same number of shouts expended (see Figure 1). In each plane $U = const.$, the line $K = const.$ represents a particular setting of the strike. If the holder of the security shouts (and $U + 1 \leq U_{max}$), he receives a security defined on the next higher plane, $U^* = U + 1$, with a strike $K^* = \mathcal{F}(S, K, U, t)$. We can see that, upon shouting, information is collected from the plane one level above the current position according to the "look up function" $\mathcal{F}$. It is important to realize that the value is only reset to $V^*$ in the case where $V^* \geq V$. In effect, $V^*$ is a minimal value of the option, similar to a simple American constraint.

It would appear to be a formidable task to solve the three dimensional, time-dependent nonlinear PDE (5) which is constrained by the value of other similarly complex equations. However, this PDE can be simplified by exploiting the structure of the dependence on the variables $U$ and $K$. This will be discussed in detail below, but it is worth noting at this juncture that the problem simplifies under the Black-Scholes assumption that $\sigma(S, t) = \sigma_{BS}$ for a wide variety of contract specifications. In this case, the PDE (5) can be seen to admit the similarity solution

$$V(S, K, U, t) = K\hat{V}(\zeta, U, t) \tag{9}$$

where $\zeta = S/K$. For the special contract design where

$$\begin{aligned} g(S, K, U) &= K\hat{g}(\zeta) \\ \mathcal{F}(S, K, U, t) &= K\hat{F}(\zeta, U, t) \\ \mathcal{D}(S, K, U, t) &= K\hat{D}(\zeta, U, t), \end{aligned}$$

the penalty function can be written as $q = K\hat{q}(\hat{V}(\zeta, U, t), \hat{V}(\zeta^*, U + 1, t))$. We can see that (9) satisfies (5,6). This class of contracts includes the important cases of call/put payoffs where the holder is allowed to reset the strike according to $K^* = cS$ where $c$ is some constant.

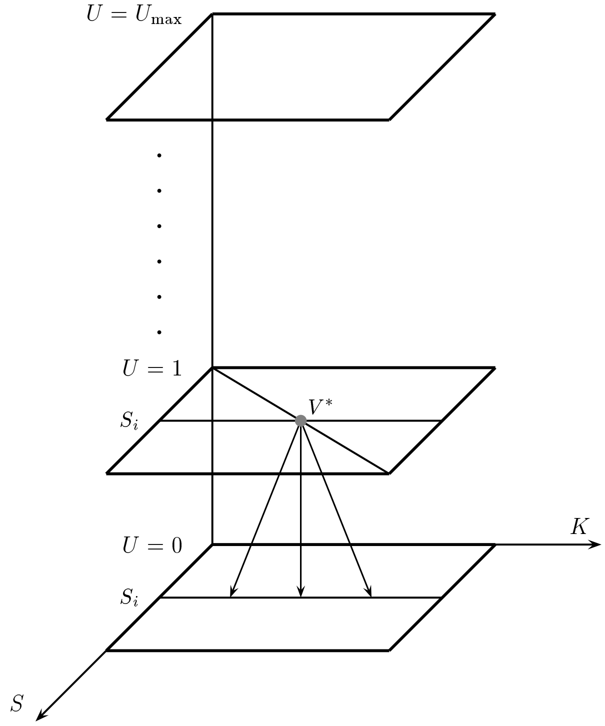



FIGURE 1: *The flow of information in a standard shout option with $\mathcal{F}(S, K, U, t) = S$. The value along the line $S = S_i$ is compared with the value $V^* = V(S_i, S_i, U+1, t)$ to determine whether or not shouting is optimal. Planes of $U =$ const. represent contracts which have the same number of shouts expended while lines of $K =$ const. represent specific settings of the strike $K$.*

This also admits dividends/penalties paid upon shouting which are differences of the current strike and the asset level, i.e. $\mathcal{D}(S, K, U, t) = S - K$ as suggested in [17]. This similarity solution will be used frequently below both because it is a very important case in its own right and because it provides a convenient way of testing the solution for the full three dimensional problem.

# 3 Solving The Pricing Equation

We are ultimately interested in the value of a shout option with a specific initial strike $K = K_0$ with no shouts used $U = 0$ at the time of sale of the security, for specified values of the price of the underlying asset. Observe that equation (5) modelling the value, $V$, for a fixed $U$ and $K$ is not defined until we have determined the constraint

$$V^*(S, K, U, t) = V(S, \mathcal{F}(S, K, U, t), U + 1, t) + \mathcal{D}(S, K, U, t),$$

the value of the security received upon shouting. This requires that we know the solution $V$ on the plane $U + 1$ with $K$ ranging over all possible settings of the strike under the function $\mathcal{F}$. However, once we have determined $V^*$ and specified the terminal condition $g(S, K, U)$ the differential equation (5) is independent of the variables $K$ and $U$.

Of course it would be prohibitively expensive to store the complete solution through time for $V^*$. Using a timestepping numerical algorithm it is only necessary to know the solution on the $U+1$ plane during the current timestep. Consequently, if we imagine stepping through time, then at each timestep we should evaluate the solutions recursively in the order $U = U_{\max}, \ldots, 0$.

### 3.1 Discretization Techniques

Equation (5) can be put into the more familiar form of the convection-diffusion equation if we define $\tau = T - t$

$$V_\tau = \frac{1}{2}\sigma(S,\tau)^2 S^2 V_{SS} + r(\tau) S V_S - r(\tau) V + q(V, V^*). \tag{10}$$

Consider the solution of equation (10) for fixed values of $(K, U)$. To avoid notational clutter, we drop the dependence on $(K, U)$ in the following. Let $V(S_i, \tau) = V_i$ be the discrete solution to equation (10) at asset value $S_i$, and time (going backwards) $\tau$. Applying a finite volume method [22] gives

$$A_i \left(\frac{dV_i}{d\tau}\right) = \left(\sum_{j\in\eta_i} \gamma_{ij}\left(V_j - V_i\right) + \sum_{j\in\eta_i} \vec{L_{ij}} \cdot \mathbf{V}_i V_{ij+1/2} - A_i r V_i\right) + A_i q(V_i, V_i^*) \tag{11}$$

where:

$$\begin{aligned}
A_i &= \left(S_{i+1} - S_{i-1}\right)/2 \\
\eta_i &= \{i-1, i+1\} \\
\gamma_{ij}^n &= \frac{\sigma_i^2 S_i^2}{2|S_j - S_i|} \\
V_{ij+1/2} &= \text{value of } V \text{ at the face between nodes } i \text{ and } j \\
\mathbf{V}_i &= (-r S_i)\hat{\mathbf{i}} \\
\vec{L_{ij}} &= \begin{cases} -\hat{\mathbf{i}} & \text{if } j = i+1 \\ +\hat{\mathbf{i}} & \text{if } j = i-1 \end{cases} \\
\hat{\mathbf{i}} &= \text{unit vector in the positive } S \text{ direction .}
\end{aligned} \tag{12}$$

Standard central weighting uses $V_{ij+1/2} = (V_i + V_j)/2$, but in some cases this may cause problems if the PDE becomes convection dominated. Other possibilities include the use of a flux limiter [22]. We remark that the algorithm used in this work detects if the PDE is convection dominated and switches to a limiter. However, for the test cases presented here the limiter was rarely used because there was sufficient diffusion. Defining

$$\mathcal{L} V_i = \left(\sum_{j\in\eta_i} \gamma_{ij}\left(V_j - V_i\right) + \sum_{j\in\eta_i} \vec{L_{ij}} \cdot \mathbf{V}_i V_{ij+1/2} - A_i r V_i\right), \tag{13}$$

equation (11) can be written

$$A_i \left( \frac{dV_i}{d\tau} \right) = \mathcal{L}V_i + A_i q(V_i, V_i^*). \tag{14}$$

A detailed description of the discrete form for the penalty term $q(V_i, V_i^*)$ is provided in [21], and will not be repeated here.

We will consider three timestepping methods: fully implicit (backward Euler), central weighting (Crank-Nicolson), and a second order backward differencing method (BDF). We can write the backward Euler and Crank-Nicolson algorithms as

$$A_i \left[ \frac{V_i^{n+1} - V_i^n}{\Delta\tau^{n+1}} \right] = \theta \mathcal{L} V_i^{n+1} + (1-\theta)\mathcal{L}V_i^n + A_i q(V_i^{n+1}, (V_i^*)^{n+1}) \tag{15}$$

where $\theta = 1$ for backward Euler and $\theta = \frac{1}{2}$ for Crank-Nicolson, $V_i^n = V(S_i, \tau_n)$, and $\Delta\tau^{n+1} = \tau^{n+1} - \tau^n$. The second order BDF scheme is obtained by using the following approximation for the time derivative [3]:

$$A_i \left[ \left( 1 + \frac{\Delta\tau^{n+1}}{\Delta\tau^n} \right) \frac{(V_i^{n+1} - V_i^n)}{\Delta\tau^{n+1}} - \left( \frac{\Delta\tau^{n+1}}{\Delta\tau^n} \right) \frac{V_i^{n+1} - V_i^{n-1}}{\Delta\tau^{n+1} + \Delta\tau^n} \right] = \mathcal{L}V_i^{n+1} + A_i q(V_i^{n+1}, (V_i^*)^{n+1}) \tag{16}$$

where $\Delta\tau^n = \tau^n - \tau^{n-1}$. The BDF algorithm is a multi-step method which requires the solution at the previous two timesteps. We use a fully implicit step to start the algorithm.

It is important to exploit the solution behaviour when choosing a time discretization. Note that the payoff function $g(S, K, U)$ is generally not smooth in $S$. A typical case is the put payoff, $g(S, K, U) = \max(K - S, 0)$, which has a discontinuous partial derivative with respect to $S$ at $S = K$. Although this discontinuity is smoothed by the parabolic differential operator, we still often require many small timesteps near the expiry time to capture this behaviour. As the time to expiry increases the solution becomes smoother and we are able to take larger timesteps. Since many of the contracts which we are valuing are long term, it is important to use a variable timestep selection mechanism to take advantage of this computational performance gain.

As will be shown, multi-step methods (such as a second order BDF) perform poorly on this type of problem. This is because a history of the solution at previous time levels is a poor indicator of the solution at the new time level if the solution becomes constrained by the minimal value $V^*$. As well, it is necessary to use the same timestep for each of the one (spatial) dimensional problems on all planes $U = U_{max}, \ldots, 0$. This is required in order to avoid time interpolation of $V^*$ (which is unlikely to have a smooth time behaviour). Therefore we must know, before we begin a timestep, the value of the global timestep which will be used on all planes for all one dimensional problems. To this end, we use a very simple inexpensive timestep selector, which is a modified form of the method in [15]. This method uses only information from the current timestep to predict a suitable timestep adjustment for the next timestep. The method employs a relative change criterion. Specifically, given an initial timestep $\Delta\tau^{n+1}$, then a new timestep $\Delta\tau^{n+2}$ is selected so that

$$\Delta\tau^{n+2} = \left( \min_{U,K,S} \left[ \frac{\texttt{dnorm}}{\frac{|V(S,U,K,\tau^n + \Delta\tau^{n+1}) - V(S,U,K,\tau^n)|}{\max(M, |V(S,U,K,\tau^n + \Delta\tau^{n+1})|, |V(S,U,K,\tau^n)|)}} \right] \right) \Delta\tau^{n+1} \tag{17}$$

where `dnorm` is a target relative change (during the timestep) specified by the user. The scale $M$ is selected so that the timestep selector does not take an excessive number of timesteps in regions where the value is small. For values $V < M$, the target relative change criterion is replaced with a target absolute change. Since we are valuing contracts with a wide range of strikes, we use the scale $M = \max(.01K, 10^{-2})$.

## 3.2 Meshing Considerations: Constructing A Solution Domain

The original problem is posed on an unbounded domain $(S, K) \in [0, \infty) \times [0, \infty)$. However, computationally it is necessary to approximate this with a truncated finite region $(S, K) \in [0, S_{max}] \times [0, K_{max}]$ where $S_{max}$, $K_{max}$ are suitably large numbers. We will discuss a methodology for choosing $S_{max}$, $K_{max}$ appropriately.

In conventional option valuation applications, the boundary conditions which are applied at $S \to \infty$ are constructed under the asymptotic assumption that $S \gg K$. For example, the appropriate boundary condition for a standard put option is given by $V(S \gg K) = 0$. This is because as $S \to \infty$ it becomes very unlikely that the asset price will fall below the (finite, fixed) strike $K$, rendering the contract worthless. In the context of shout options, if we construct a rectangular computational domain then we require $S_{max} \gg K_{max}$ in order to accurately apply these asymptotic boundary conditions.

Consider the rectangular computational domain shown in Figure 2, with $S_{max} \gg K_{max}$. In general, we are unable to determine the constraint, $V^*$, in cases where the strike is reset to $K^* > K_{max}$ upon shouting. For example, if we take the case of a standard shout floor where the strike is set according to $K^* = S$ upon shouting, we will not have the necessary data within the computational domain to determine $V^*$ for $S > K_{max}$. An exception occurs when there exists a similarity solution, but as we are interested in valuing a general class of shout options we develop a methodology which can be adapted to more complex securities and modelling situations.

To avoid an *ad hoc* procedure for estimating $V^*$ for $S > K_{max}$, we will modify the contract so that there is no uncertainty about computing $V^*$. Consider an alternative contract where the holder is not allowed to reset to a higher strike level than $K_{max}$. In other words, if the original contract specifies

$$V^* = V(S, K = S, U + 1, t) \tag{18}$$

for $U + 1 \leq U_{max}$, then we modify (18) so that $V^*$ is given by

$$V^* = V(S, K = \min(S, K_{max}), U + 1, t) \tag{19}$$

for $U + 1 \leq U_{max}$. This contract will only require information which is available within our computational domain. Furthermore, this contract becomes equivalent to the original

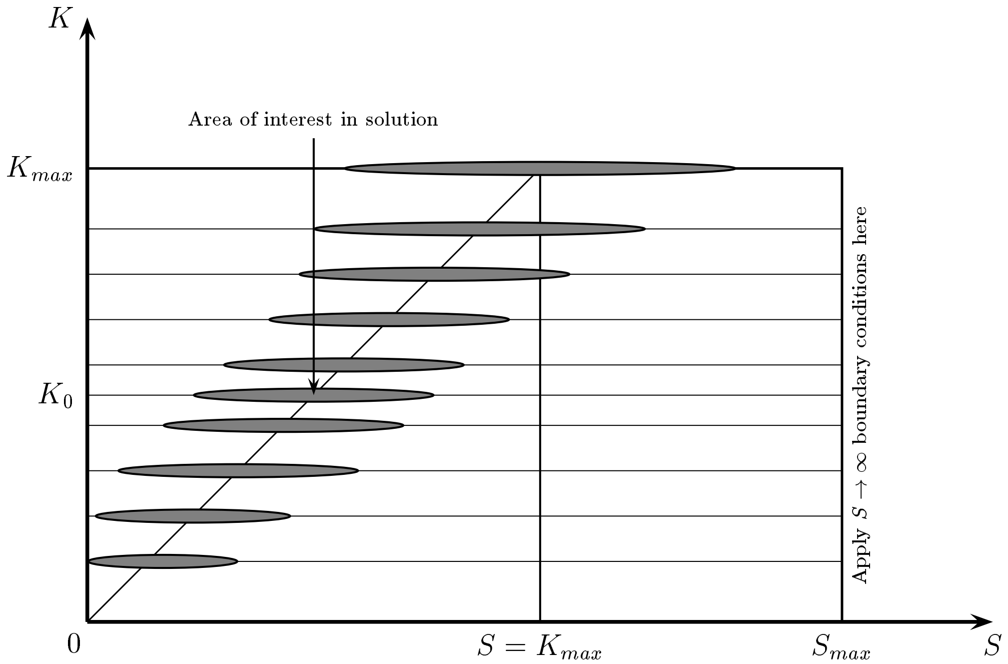



FIGURE 2: *A rectangular computational domain for the numerical solution of a shout option sold with an initial strike $K_0$ for fixed $U = const.$ The shaded areas represent refined regions in the S-grid.*

specification as $K_{max} \to \infty$. We need to understand how large $K_{max}$ should be to effectively recover the original problem specification. Interestingly enough it is sometimes the case that the solution in the region of interest is completely unaffected by this approximation.

For clarity, consider the case of a shout put option with a single exercise opportunity, so that $U \in \{0, 1\}$. Notice that the selection of $K_{max}$ does not depend upon any of the problems on the top plane $U_{max} = 1$ since on this plane there are no more shout opportunities. The contracts on this plane given by lines of $K = const.$ are essentially self-contained standard options. We denote the optimal exercise boundary on the plane $U = 0$ for a particular strike setting $K$ by $S^*_K(t)$. The holder would optimally shout and lock in the higher protective floor at time $t$ if $S \geq S^*_K(t)$. It is important to note that the exercise boundary need not be finite, i.e. there can be situations where it is not optimal to shout regardless of how high $S$ becomes. In the exercise region, $S \geq S^*_K(t)$, the solution value is determined by the algebraic constraint $V^*$ (the value of the contract received upon shouting). In the continuation region, $S < S^*_K(t)$, the solution satisfies the PDE

$$\frac{\partial V}{\partial t} + r(t)S\frac{\partial V}{\partial S} + \frac{1}{2}\sigma(S,t)^2 S^2 \frac{\partial^2 V}{\partial S^2} - r(t)V = 0$$

subject to the free boundary condition

$$V(S^*_K(t), K, U = 0, t) = V(S^*_K(t), \mathcal{F}(S^*_K(t), K, U = 0, t), U = 1, t)$$

and the "smooth-pasting" condition [11]

$$\frac{\partial}{\partial S}V(S^*_K(t), K, U = 0, t) = \frac{\partial}{\partial S}V(S^*_K(t), \mathcal{F}(S^*_K(t), K, U = 0, t), U = 1, t).$$

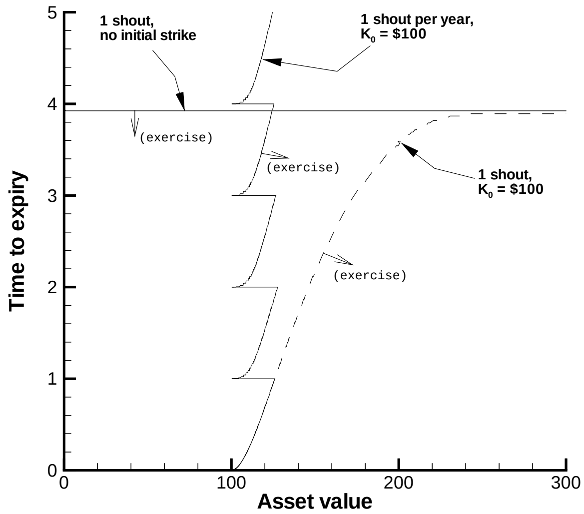



FIGURE 3: *Comparison of the optimal exercise boundary of various contracts. Shown are contracts with no initial strike and with* $K_0 = \$100$ *which have a single shout opportunity during the life of the contract. At early times in the contract (large times to expiry), these contracts have an unbounded optimal exercise boundary. Also shown is a shout option with* $K_0 = \$100$ *which permits one shout opportunity per year and has a bounded exercise boundary. The arrows on the contours indicate the region in which it is optimal to shout. Parameters used:* $\sigma_{BS} = .25$, $r(t) = .06$, $T = 5$ *years.*

As long as the exercise boundary $S^*_K(t)$ is contained within our computational domain for all $t \in [0, T]$, the value in the continuation region will be unaffected by the truncation of the $K$ axis.

Often the exercise boundary $S^*_K(t)$ can become unbounded during the life of the contract (see Figure 3). In [7], the authors develop a formula for the optimal exercise time for contracts which have no initial strike settings under constant volatility assumptions. The holder of a contract with an initial strike setting will wait at least as long before shouting. This is clear financially since the holder of a contract with an initial strike setting does not have to shout to protect the asset levels $S \leq K_0$ and thus can wait longer if $S$ is in this region. Mathematically, the value of the contract with the initial strike setting is never less than the value of the contract with no initial strike setting (because of the larger terminal condition due to the payoff at expiry). These contracts are constrained by the same $V^*$ since upon shouting the holder receives the same contract in either case. Thus optimal exercise for the holder of a contract with an initial strike implies optimal exercise for the holder of the contract with no initial strike setting. Since there can be "waiting times" where it is not optimal to shout regardless of the asset level $S$, for the no initial strike contract, the contract with initial strike setting $K_0$ must also have a waiting region where it is not optimal for the holder to shout.

If the exercise boundary becomes unbounded, then it is not possible to perform the above construction which exactly retains the original solution. To handle these cases we rely on

| | $S_{max}$ | | | |
|---|---|---|---|---|
| | 250 | 500 | 1000 | 2000 |
| Model | Option Value at $S = \$100$ | | | |
| Black-Scholes ($\sigma_{BS} = .25$) | 8.7981 | 8.8024 | 8.8024 | 8.8024 |
| CEV ($\alpha = 3.0$) | 8.8202 | 8.8660 | 8.8664 | 8.8664 |

TABLE 1: *The effect of truncating the asset price computational domain on standard at-the-money ($S = \$100$) European put options. For these test cases we use the same $S$ discretization for all problems with 200 nodes for $0 \leq S \leq 250$ and extend it to the desired computational domain, $0 \leq S \leq S_{max}$. Crank-Nicolson timestepping was used with 200 timesteps. See equation (21) for a definition of the CEV model. Note that we are not attempting to converge to the exact solution, rather we are studying the effect of approximating the boundary condition at infinity. Parameters used: $\sigma_{BS} = .25$, $r = .06$, $K_0 = \$100$, $S_0 = \$100$, $T = 5$ years.*

the fact that the Green's function of equation (10) decays rapidly so that the effects of the approximate boundary condition should be small in the region of interest if $K_{max}$ is suitably large. The boundary condition which is applied on planes with shout opportunities remaining ($U \neq U_{max}$) is given by

$$V(S \to \infty, K, U, t) = \begin{cases} V^* & \text{if it is optimal to shout} \\ V_t = 0 & \text{otherwise.} \end{cases} \tag{20}$$

It is not practical to determine $K_{max}$ *a priori* since this requires knowledge of the behaviour of the optimal exercise boundary. In practice, we simply carry out an initial computation with an estimated value of $K_{max}$, and then repeat the computation with an increased $K_{max}$ until we observe no effect on the results to the desired accuracy. This $K_{max}$ is then used to determine $S_{max}$.

Table 1 explores the effect of truncating the $S$ domain when valuing standard European put options. For the Black-Scholes case, setting $S_{max}$ at around five times the strike appears to suffice. Also included in the table is a CEV model [9] with $\alpha = 3.0$. We follow [16, 5] by specifying the CEV volatility function as:

$$\sigma(S, t) = \sigma_{BS} S_0^{1-\alpha/2} S^{\alpha/2-1} \tag{21}$$

This specification is used to facilitate comparisons between the CEV and Black-Scholes models by ensuring that both models have the same diffusion at $S_0$. Note that the CEV model contains the Black-Scholes model as the particular case $\alpha = 2$. The CEV model has a very large diffusion coefficient for high asset values when $\alpha > 2$, implying that the effects of domain truncation are more intrusive into the region of interest. From Table 1 we observe that we may need to set $S_{max}$ in the neighbourhood of ten times the strike in the CEV case.

Based on the results in Table 1, we adopt the conservative approach of choosing $S_{max} = \max(K_{max}^2/K_0, 100K_{max})$ when valuing shout put options. Since we are using an unstructured discretization we are able to extend the computational domain without hindering performance by adding several large cells at the end of the domain. In Table 2 we study the

| Model | $K_{max}$ | | | |
|---|---|---|---|---|
| Black-Scholes ($\sigma_{BS} = .25$) | 250 | 500 | 1000 | Similarity |
| Option value at $S = \$100$ | 14.776 | 14.776 | 14.776 | 14.776 |
| Option value at $S = \$200$ | 17.973 | 18.069 | 18.069 | 18.069 |
| | $K_{max}$ | | | |
| CEV ($\alpha = 3.0$) | 15,000 | 50,000 | 150,000 | 500,000 |
| Option value at $S = \$100$ | 17.553 | 17.563 | 17.569 | 17.574 |
| Option value at $S = \$200$ | 41.702 | 42.171 | 42.667 | 42.692 |

TABLE 2: *The effect of modifying the contract specification according to (19). The value of a standard shout put option with a single exercise opportunity with various maximum strike settings, $0 \leq K \leq K_{max}$. In the Black-Scholes setting we are able to use a similarity solution (see equation (9)) which allows us to determine $V^*$ for any strike setting $K^*$. For these test cases we use the same $K$ discretization for all problems with 200 nodes and extend it to the desired computational domain, $0 \leq K \leq K_{max}$. For $S$ direction, we use the computational domain, $0 \leq S \leq \max(K_{max}^2/K_0, 100K_{max})$ for non-similarity solutions and $0 \leq S \leq 100K_0$ for similarity solutions. Both of these contracts have unbounded optimal exercise boundaries. Crank-Nicolson timestepping was used with 200 timesteps. The CEV model is described in equation (21). Note that we are not attempting to converge to the exact solution, rather we are studying the effect of our approximation of the original contract. Parameters used: $\sigma_{BS} = .25$, $r = .06$, $K_0 = \$100$, $S_0 = \$100$, $T = 5$ years.*

numerical effects of truncating the solution domain of a shout put option which resets the strike according to $K^* = S$ upon shouting. The contract is modified as given in equation (19). In the case of the Black-Scholes model we are able to compare our answers with a similarity solution which does not require such a domain truncation. The effect of the domain truncation is minimal for reasonable asset levels $S$. Even in the more diffusive CEV case, the associated errors can still be reduced to an acceptable level by selecting $K_{max}$ sufficiently large.

## 3.3 Implementation: An Object-Oriented Approach

By construction, this framework for the valuation of shout options lends itself naturally to an object-oriented implementation. An object-oriented approach allows easy extensibility and quick, robust implementation once a suitable library of objects has been defined.

As we have seen, a shout option can be considered to be a collection of American style options, each associated with a particular strike setting $K$, and number of shouts used $U$. Each of these individual problems are independent except through the application of the constraint $V \geq V^*$. It is natural to define an object which represents the solution for a particular $U$, $K$ and associate with it methods to solve the approximate differential equation (11). A collection of these objects is then controlled by a *super-class* which is responsible for setting the appropriate algebraic constraints for the individual problems.

In addition, this implementation makes it feasible to exploit the parallelism inherent in the valuation algorithm. Notice that each of the problems for individual $K = const.$ are independent within a plane of $U = const.$ We could therefore achieve a theoretical increase in performance of $\approx n_k$ (where $n_k$ is the number of nodes in the $K$ discretization), given a sufficient number of processors. In this case the processor time to value a shout option would be $C_{shout} \approx U_{max} C_{standard}$, where $C_{standard}$ is the cost to value one of the building block standard contracts. Thus a parallel architecture can make the general solution of a shout option using more complex modelling assumptions (such as a volatility surface) no more expensive than a Black-Scholes model solution using a similarity reduction. We remark that typically these contracts are designed so that $U_{max} < 10$. As $U_{max} \to \infty$ this option would become a lookback put option, which can be valued more efficiently using other methods [18].

Finally, as mentioned in the introduction, we are interested in developing a framework for valuing complex types of contracts which allow investor modification. For example, many contracts are presently sold with the feature that the maturity date of the contract is extended when the holder shouts. In such cases, the holder can be thought of as setting the tuple $(K, T_c)$ upon shouting, where $K$ is the strike setting and $T_c$ is the current contract maturity. Using object-oriented techniques allows us to extend our approach to handle the additional complexity of these contracts in a straightforward fashion.

# 4 Error Analysis

In order to gain understanding as to how error is introduced into the approximate solution we develop the following simple heuristic model. For any fixed $U$ and $K$, the error can be decomposed into two parts: a "local" discretization error, due to the fact that we are solving each problem for fixed $U$, $K$ on a finite grid with finite timesteps; and an "external" error due to forcing the algebraic constraints (i.e. determining and enforcing the condition $V \geq V^*$). Recalling the dependence of $V^*$ on the solution on the plane $U + 1$, we can think of this latter error as composed of interpolation errors from determining $V^*$, errors introduced by the penalty formulation of applying the constraint, and errors in the approximate solution for $U+1$. It can be shown that the relative errors introduced by the penalty formulation are $O(1/\texttt{Large})$, where `Large` is a parameter used to force the constraint. This source of error will not be considered further here (see [21] for a more detailed discussion). We define the global solution error $\varepsilon_U$ as

$$\varepsilon_U(t) = \max_{S,K} |V_e(S,K,U,t) - V(S,K,U,t)| \tag{22}$$

where $V_e$ and $V$ respectively are the exact and approximate solutions of (5,6). The cumulative effect of these interpolation and discretization errors can be bounded by the backwards recurrence inequality for the global solution error

$$\begin{aligned} &\varepsilon_U = \text{discret. error}_U \quad \text{for } U = U_{max} \\ &\varepsilon_U \leq \text{discret. error}_U + \text{interp. errors}_U + \varepsilon_{U+1} \quad \text{for } U = U_{max-1}, \ldots, 0. \end{aligned}$$

Since we are interested in the solution $V(S_0, K_0, 0, 0)$, then the error in the computed solution at this point $\varepsilon_0$ is bounded by

$$\varepsilon_0 \leq U_{max} \left[\text{discret. error}^* + \text{interp. errors}^*\right] + \text{discret. error}^*, \tag{23}$$

where

$$\text{discret. error}^* = \max_{0 \le U \le U_{max}} \text{discret. error}_U$$
$$\text{interp. error}^* = \max_{0 \le U < U_{max}} \text{interp. error}_U.$$

Since we are using a second order space and time discretization, the global discretization error for a solution along a line $U = const.$, $K = const.$ will be given by [13]:

$$\text{discretization error} = O\left((\Delta S)^2\right) + O\left((\Delta t)^2\right)$$

where $\Delta S$ is the maximum grid spacing in the $S$ direction.

During each timestep as the PDE is solved the value $V$ will be compared with $V^*$. If we introduce an interpolation error of $O\left((\Delta S)^{h_1} + (\Delta K)^{h_2}\right)$ for each determination of $V^*$ (where $\Delta K$ is the maximum grid spacing in the $K$ discretization and $h_1, h_2$ represent the order of the interpolation method in the $S$ and $K$ directions respectively), then the total cumulative interpolation error after $O(1/\Delta t)$ timesteps is given by:

$$\text{cumulative interpolation error} = O\left(\frac{(\Delta S)^{h_1} + (\Delta K)^{h_2}}{\Delta t}\right).$$

Using these results, our model for the error in the approximate solution gives:

$$\varepsilon_0 \le U_{max}\left[O\left((\Delta S)^2\right) + O\left((\Delta t)^2\right) + O\left(\frac{(\Delta S)^{h_1} + (\Delta K)^{h_2}}{\Delta t}\right)\right] + O\left((\Delta S)^2\right) + O\left((\Delta t)^2\right). \tag{24}$$

If the contract is designed to allow the holder $U_{max}$ shout opportunities per year then the no arbitrage considerations (8) imply that the value on the plane $U_{max}$ is set equal to the value on plane $U = 0$ at each reset time $t_i$. In this case, the cumulative error is bounded by (24) with $U_{max}$ replaced with $U_{total} = U_{max} \times N$, $N$ being the number of times the shout counter variable is reset to zero.

We must be careful to control the cumulative interpolation error as $\Delta t \to 0$. We will investigate the following three techniques:

- **Method 1:** As mentioned earlier (see equation (9) and accompanying discussion), a similarity solution is available for some cases under the Black-Scholes assumption of constant volatility. For example, if $\mathcal{F}(S, K, U, t) = S$ then $V^*(S, K, U, t) = K\hat{V}(\zeta^* = 1, U+1, t)$. As long as the discretization for $\hat{V}$ contains the node $\zeta = 1$, no interpolation is required. In [7] the authors describe an explicit type algorithm which essentially uses this similarity solution to determine $V^*$ using

$$V^*(S, K, U, t) = \frac{\mathcal{F}(S, K, U, t)}{K_0} V\left(\frac{K_0 S}{\mathcal{F}(S, K, U, t)}, K_0, U + 1, t\right) \tag{25}$$

  where $K_0 \neq 0$. The authors ensure that the node $S = K_0$ is in the $S$ discretization, thereby eliminating interpolation error.

- **Method 2:** If the function $\mathcal{F}$ is autonomous, i.e. $\mathcal{F}(S,K,U,t) = \mathcal{F}(S,K,U)$, we can guarantee that the interpolation error is exactly zero by choosing our discretization of the variable $K_{U+1}$ on the $U+1$ plane so that $V^* = V(S, \mathcal{F}(S,K,U,t), U+1, t)$ is known for each of the discrete values of $S_U, K_U$ on the $U$ plane. For the example of a shout option where the strike is set to the current asset level, $K^* = S$, one could achieve this by using the same grid spacings in the $S$ and $K$ directions. Unfortunately this simple solution has limited usefulness in practice. In order to control the local discretization error it is desirable to use a different grid in the $S$ direction for each value of $K$. The $K$ discretization on the $U+1$ plane is the union of the required strike settings for each line of $K = const.$ on the $U$ plane. Therefore, in practice, we are forced to use the same $S$ discretization for each line $K = const.$ on the $U$ plane in order to obtain a reasonable number of nodes on the $U+1$ plane. This results in a higher truncation error in the solution for $K$ values for which the $S$ discretization has not been specifically tailored.

- **Method 3:** We can facilitate convergence by choosing $\Delta S = C_1 \Delta t$ and $\Delta K = C_2 \Delta t$ for some constants $C_1, C_2$ and using an interpolation scheme which has order $h_1, h_2 \geq 2$. This suggests an optimal choice of $\Delta S = \Delta K = C\Delta t$ with $h_1 = h_2 = 3$ in order to cause the errors introduced in the approximate solution to tend to zero at the same rate asymptotically as $\Delta t \to 0$. Note that this assumes that the solution has bounded second derivatives along the interpolating directions. This technique of controlling the interpolation error has the drawback that as $\Delta t \to 0$ we will require an increasingly fine discretization in the $S$ and $K$ variables. There is still freedom to select the interpolation method within the unstructured mesh for $S$ and $K$. Many common contracts have solution surfaces which are not approximated well by coordinate axis aligned interpolation (see Figure 4). In cases where the strike setting upon shouting is independent of the current strike level, we can interpolate along the curve $K^* = \mathcal{F}(S,U,t)$ for a fixed $U$, $t$. Often the solution is smoother along this curve, allowing more accurate interpolation.

## 4.1 Numerical Study Of Error In The Approximate Solution

We can study a simple contract which admits a similarity solution to gain intuition regarding the effects of interpolation and truncation errors on our approximate solution. We will use a shout put option with an initial strike $K_0 = \$100$ which allows the holder to reset the strike according to $K^* = S$.

When performing convergence testing we define four refinement levels. These are constructed so that each increasing level contains double the number of nodes in the $S$ direction. In addition, the parameters controlling the automatic timestep selector are set so that the number of timesteps is approximately doubled. These parameters are given in Table 3. As shown in equation (17), the parameter `dnorm` specifies a target relative change during the timestep. The parameter `dtime` is the initial timestep and the timestep used after each reset time $t_i$. The parameter `dtimax` specifies the maximum allowable timestep.

As noted previously, in order to determine a suitable computational domain we perform an initial computation with an estimated value of $K_{max}$. We then repeat the computation with an increased $K_{max}$ until we observe no effect on the results to the desired accuracy.

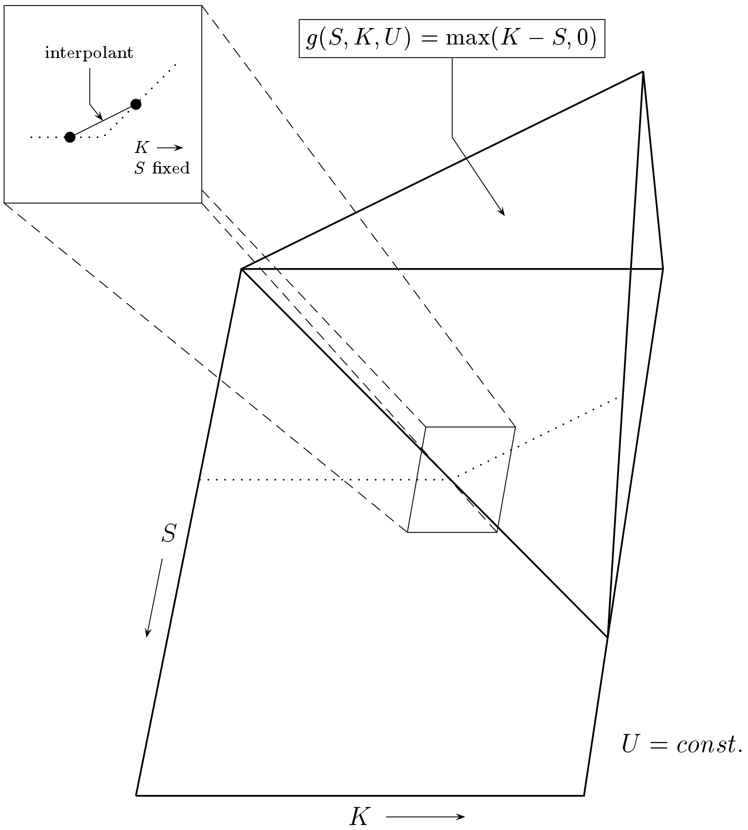



FIGURE 4: *Interpolation of a shout put option with payoff $g(S, K, U) = \max(K - S, 0)$ using coordinate axis aligned interpolation. The determination of $V^* = V(S, K^* = S, U + 1, t)$ requires us to interpolate the diagonal $K = S$. This is not approximated well by linear or quadratic interpolants for fixed $S$ due to the discontinuous derivative at the point of interest.*

Following our discussion above, we take $S_{max} = \max(K_{max}^2/K_0, 100K_{max})$ as an upper bound for our $S$ discretization.

The discrete system of equations (11) are solved using a Newton iteration. The solution at a particular time level, $t$, has been deemed to have converged if

$$\max_{S \in [0, S_{max}]} \frac{|V_{U,K}^{(k+1)}(S,t) - V_{U,K}^{(k)}(S,t)|}{\max(M, |V_{U,K}(S,t)|)} \leq \texttt{ctol}$$

where $V_{U,K}^{(k)}(S,t)$ is the solution after the $k^{th}$ Newton iteration and $M$ is a scale constant used to switch the relative change criterion to an absolute change criterion for $|V_{U,K}| \leq M$.

Scaling is particularly important for these problems since we are valuing contracts with strike settings $K$ varying over several orders of magnitude. Throughout this paper we use $M = \max(.01K, 10^{-2})$. As suggested in [21] we set `Large` $= 1/$`ctol` where `Large` is the penalty parameter used to force the constraint. Table 4 gives the value of a shout put option at $S = \$100$ for various choices of convergence tolerance. *Level 3* (see Table 3) grid/timestep parameters were used. Table 4 shows that the magnitude of `ctol` is an estimate of the number of significant figures in the approximate solution. For the remainder of this paper we specify

| | Asset nodes | `dnorm` | `dtime` | `dtimax` |
|---|---|---|---|---|
| Level 1 | 100 | .10 | 1.e-4 | .10 |
| Level 2 | 200 | .05 | 5.e-5 | .05 |
| Level 3 | 400 | .025 | 2.5e-5 | .025 |
| Level 4 | 800 | .0125 | 1.25e-5 | .0125 |

TABLE 3: *Refinement levels used for convergence tests. The solver parameters* `dnorm`*,* `dtime`*, and* `dtimax` *represent the target relative change per timestep, the initial timestep size, and the maximum timestep size respectively.*

| | `ctol` | | | |
|---|---|---|---|---|
| | $10^{-3}$ | $10^{-4}$ | $10^{-5}$ | $10^{-6}$ |
| Option value at $S = \$100$ | 24.3606 | 24.3726 | 24.3734 | 24.3734 |
| Iterations | 15224 | 22927 | 27952 | 28339 |
| Timesteps | 5052 | 5052 | 5052 | 5052 |
| Iterations per timestep | 3.01 | 4.54 | 5.53 | 5.61 |

TABLE 4: *Effect of Newton convergence tolerance,* `ctol`*, on similarity solution for the value of a shout put option with 5 exercise opportunities and no shout counter resets. Parameters used: $\sigma_{BS} = .25$, $r = .06$, $K_0 = \$100$, $T = 5$ years. Crank-Nicolson timestepping used. Compare with corresponding case ($T = 5$ years) Level 3 in Table 5.*

`ctol` $= 10^{-5}$.

Many of the examples studied in this paper admit a similarity solution. An approximate solution obtained using a similarity reduction can be regarded as an optimal approximate answer under the required simplified modelling assumptions. As mentioned previously, for the similarity solution one can create an unstructured mesh which admits no interpolation error (see (25)). Further, the discretization can be tailored for the specific strike setting resulting in low discretization error. Table 5 shows the value of a shout put option at various initial asset prices and times until maturity. This table illustrates convergence and timing results for a simple contract having a similarity solution.

### 4.1.1 Timestepping Methods And Considerations

It is important to utilize an efficient yet robust timestepping algorithm because many contracts of practical interest are long term. We will consider fully implicit (backward Euler), central weighting (Crank-Nicolson), and a second order backward differencing method (BDF).

All of the timestepping algorithms used in this numerical study are unconditionally stable, implying that there are no timestep limitations placed on the solution due to stability considerations. The BDF and Crank-Nicolson methods are second order accurate while the backward Euler algorithm is first order accurate. Furthermore, the BDF algorithm is strongly A-stable [3], meaning that errors introduced are always damped regardless of timestep size. This suggests that the BDF method could be the preferred algorithm.

To compare the algorithms we study a contract which admits a similarity solution so that

| | | Level 1 ($n = 100$) | Level 2 ($n = 200$) | Level 3 ($n = 400$) | Level 4 ($n = 800$) |
|---|---|---|---|---|---|
| Maturity | $S$ | Option Value | | | |
| $T = 5$ years | \$80 | 23.562 | 23.584 | 23.586 | 23.587 |
| | \$100 | 24.354 | 24.370 | 24.373 | 24.374 |
| | \$120 | 27.637 | 27.656 | 27.659 | 27.659 |
| | Timesteps | 108 | 213 | 421 | 837 |
| | cpu (sec) | 2.88 | 12.4 | 58 | 259 |
| Maturity | $S$ | Option Value | | | |
| $T = 10$ years | \$80 | 24.452 | 24.477 | 24.491 | 24.492 |
| | \$100 | 27.245 | 27.275 | 27.279 | 27.280 |
| | \$120 | 31.137 | 31.170 | 31.174 | 31.175 |
| | Timesteps | 158 | 313 | 621 | 1237 |
| | cpu (sec) | 3.86 | 16.9 | 78.3 | 404 |
| Maturity | $S$ | Option Value | | | |
| $T = 20$ years | \$80 | 23.253 | 23.299 | 23.306 | 23.307 |
| | \$100 | 27.414 | 27.459 | 27.465 | 27.466 |
| | \$120 | 31.920 | 31.968 | 31.974 | 31.975 |
| | Timesteps | 258 | 513 | 1021 | 2037 |
| | cpu (sec) | 5.39 | 23.4 | 109.6 | 529 |

TABLE 5: *Convergence of similarity solution for the value of a shout put option with five exercise opportunities (no shout counter resets). Parameters used: $\sigma_{BS} = .25$, $r = .06$, $K_0 = \$100$. $n$ is the number of asset nodes in the discretization. Notice that the longer term options with maturities of ten or twenty years are not much more expensive computationally than the shorter term option with a maturity of five years. This is because the solution becomes smoother as it evolves, allowing the automatic timestep selector to take larger timesteps. Crank-Nicolson timestepping was used. CPU times were obtained using a Pentium II 400 PC.*

we can eliminate interpolation error. This is important because in order to isolate the time truncation error we will be increasing the number of timesteps while maintaining the same spatial discretization, which precludes *Method 3* for controlling the interpolation effects.

Table 6 presents a comparison of the different time discretization techniques. We observe the usual relationship between the backward Euler and Crank-Nicolson algorithms. The slow convergence of the first order backward Euler algorithm compared with the second order Crank-Nicolson scheme illustrates the practical significance of using a higher order method. The multi-step BDF algorithm performs poorly, although it appears to be converging in the limit as $\Delta t \to 0$. As argued previously, this failure is a direct consequence of the multi-step technique. The BDF scheme assumes time domain smoothness over the previous two timesteps. This smoothness is disrupted by the enforcing of the algebraic constraints due to the exercise opportunities (notice that the performance of the BDF method is worst at higher asset levels, corresponding to the exercise region). For the remainder of this paper we use the second order Crank-Nicolson scheme in the numerical examples.

| | | dnorm | | |
|---|---|---|---|---|
| | | .10 | .05 | .025 |
| | $S$ | Option Value | | |
| Crank-Nicolson | \$80 | 19.109 | 19.110 | 19.111 |
| | \$100 | 14.616 | 14.598 | 14.594 |
| | \$120 | 14.929 | 14.930 | 14.932 |
| | Timesteps | 67 | 133 | 267 |
| | $S$ | Option Value | | |
| Backward Euler | \$80 | 19.067 | 19.089 | 19.100 |
| | \$100 | 14.453 | 14.523 | 14.558 |
| | \$120 | 14.800 | 14.864 | 14.897 |
| | Timesteps | 64 | 131 | 266 |
| | $S$ | Option Value | | |
| BDF | \$80 | 19.672 | 19.434 | 19.220 |
| | \$100 | 16.213 | 15.377 | 14.750 |
| | \$120 | 17.296 | 15.774 | 14.980 |
| | Timesteps | 67 | 134 | 269 |

TABLE 6: *Timestepping algorithm comparison for shout put option with one exercise opportunity per year (two total). Parameters used: $\sigma_{BS} = .25$, $r = .06$, $K_0 = \$100$, $T = 2$ years. Solution obtained using similarity transformation with asset grid having $n = 200$ nodes. As in Table 3, we also reduce the initial timestep size,* `dtime`*, and the maximum timestep size,* `dtimax`*, as we decrease* `dnorm`*. Notice the poor performance of the multistep second order BDF timestepping scheme and the slow first order convergence of the backward Euler algorithm compared with Crank-Nicolson.*

#### 4.1.2 Effect Of Number Of Shout Opportunities

To explore the effect of the number of shouts on the accuracy of the solution we again use the similarity solution (see Table 7). In this case the only errors are due to the discretization and in determining $V^*$ due to error in the solution on the $U+1$ plane. Again the parameters for the automatic timestep selector were adjusted so that the number of timesteps was approximately equal to the number of nodes in the asset grid. For comparison purposes, we also include the results given in [7] for this same option. The results in [7] were obtained using a similarity reduction and an explicit method (more commonly known in finance as a lattice or tree method). The error in the approximate solution increases with the number of shout opportunities. As expected, the error is bounded above by linear growth in the number of shout opportunities. Contracts where the holder is allowed a fixed number of reset opportunities per year are of great practical importance in the context of segregated funds. In this particular case the error introduced is approximately the same as the error in a shout contract with the same total number of shout opportunities. Notice also that these contracts are slightly cheaper than their unrestricted counterparts since the shout opportunities are confined to one year windows.

| | No shout counter resets | | | | Shout counter reset yearly | | |
|---|---|---|---|---|---|---|---|
| Number of shouts | 1 | 2 | 5 | 10 | 1/yr. (5 total) | 2/yr. (10 total) | 4/yr. (20 total) |
| This work | | | | | | | |
| Asset nodes | Option Value at $S = \$100$ | | | | | | |
| 100 | 5.5648 | 7.7144 | 11.278 | 13.793 | 9.9412 | 12.399 | 14.463 |
| 200 | 5.5697 | 7.7215 | 11.291 | 13.814 | 9.9524 | 12.415 | 14.485 |
| 400 | 5.5709 | 7.7233 | 11.294 | 13.820 | 9.9551 | 12.419 | 14.490 |
| 800 | 5.5712 | 7.7238 | 11.295 | 13.821 | 9.9557 | 12.419 | 14.491 |
| Coarse grid error | .0064 | .0094 | .017 | .028 | .0145 | .020 | .028 |
| Results in [7] | | | | | | | |
| Timesteps | Option Value at $S = \$100$ | | | | | | |
| 100 | 5.59 | 7.74 | 11.22 | 13.53 | N/A | N/A | N/A |
| 250 | 5.58 | 7.73 | 11.27 | 13.71 | | | |
| 500 | 5.58 | 7.73 | 11.28 | 13.77 | | | |
| 1000 | 5.57 | 7.73 | 11.29 | 13.80 | | | |
| 5000 | 5.57 | 7.72 | 11.29 | 13.82 | | | |
| Coarse grid error | .02 | .02 | .07 | .29 | | | |

TABLE 7: *Effect of number of shouts on solution of an at-the-money shout put option. Parameters used: $\sigma_{BS} = .20$, $r = .10$, $K_0 = \$100$, $T = 5$ years. This work: similarity solution using asset grid nodes $= n$, timesteps $\approx n$ (see text). Results in [7]: timesteps $= n$, asset grid nodes $= 2n - 1$. The error given is for the coarsest grid and was estimated using the converged answer at the finest level as an approximation for the exact solution.*

## 4.2 Full Three Dimensional Valuation

So far we have concentrated on contracts and modelling assumptions which admit a similarity solution. Although these are important special cases, we also wish to compute the solution for alternative underlying asset price models (e.g. CEV models [9] or volatility surfaces [1, 8]). In such cases we must solve the full three dimensional time dependent numerical problem. We are then faced with two additional possible sources of error: i) interpolation of the numerical PDE solution to obtain the constraint; and ii) meshing complications arising from locally refined meshes near $S = K$. For present purposes, we will often study a model which has a similarity solution to facilitate comparisons.

### 4.2.1 The Importance Of Specifically Tailored $S$ Discretizations

Recall that using *Method 2* we are able to eliminate all interpolation error by using the same $S$ discretization for each strike setting $K$ and then using this same discretization in the $K$ direction. As a result, our approximate solution contains only discretization error. The drawback of this method is that the $S$ discretization has been tailored for a particular strike

| | Level 1 ($n = 100$) | Level 2 ($n = 200$) | Level 3 ($n = 400$) | Level 4 ($n = 800$) |
|---|---|---|---|---|
| Discretization Method | Option Value at $S = \$100$ | | | |
| Method 2 | 13.756 | 14.226 | 14.407 | – |
| $U_{max}$ optimized | 14.051 | 14.342 | 14.447 | – |
| Similarity | 14.463 | 14.485 | 14.490 | 14.491 |

TABLE 8: *The importance of using specifically tailored $S$ discretizations for each strike setting $K$. Here we eliminate the interpolation error using Method 2 which utilizes the same $S$ discretization for each strike setting $K$. Using a specifically tailored discretization for the plane $U_{max}$ results in a significant improvement. Notice the slow convergence compared with the similarity solution. The contract allows four shout opportunities per year. Parameters used: $\sigma_{BS} = .20$, $r = .10$, $K_0 = \$100$, $T = 5$ years, $K_{max} = 10,000$. Solutions for the full three dimensional problem were not computed for Level 4 due to computational memory constraints.*

setting (in this case $K_0$, the initial strike setting). Consequently, we may not expect good performance for strike settings $K \neq K_0$. In order to study the effect of the discretization errors introduced for $K \neq K_0$, we will solve a problem which admits a similarity solution using a full three dimensional grid. This is a useful comparison since the similarity solution also requires no interpolation and contains only discretization error from the grid $K = K_0$.

Table 8 shows that the use of the same discretization for all strike settings $K$ is ineffective. We can improve on this technique by noticing that the solution for the $U_{max}$ plane can use an unstructured grid in the $S$ discretization which has been optimized for each individual $K$ if we ensure that these discretizations include the diagonal $K = S$. We are able to implement this optimization on the top plane $U_{max}$ since this plane has no more shout opportunities and therefore does not place any meshing requirements on other planes of $U = const$. We require the diagonal $K = S$ to be contained in the discretization since this plane is required to supply information from its diagonal to the plane $U_{max} - 1$. From Table 8 we can see that this scheme provides substantial improvement. As mentioned previously, if this were to be repeated for planes other than $U = U_{max}$ the number of nodes in the $K$ discretization on the plane $U + 1$ would become impractically dense. As a result this extended scheme would become unsuitable for contracts with more than one shout opportunity.

### 4.2.2 Effect Of Interpolation Method

In order to examine the effect of the interpolation of $V^*$ we will again solve a problem which admits a similarity solution using a full three dimensional grid. This allows us to compare the various interpolation methods to a highly accurate solution obtained by using the similarity solution.

Attacking the problem using *Method 3* we can minimize the discretization errors and then deal with the interpolation effects. We achieve this by using unstructured meshes which are specifically tailored for each individual problem leaving us with an error dominated by interpolation effects (a typical grid is shown in Figure 2 where the shaded regions have small $S$ spacing). As mentioned, we use the same discretization in the $K$ direction as in the $S$

| | | Level 1 ($n = 100$) | Level 2 ($n = 200$) | Level 3 ($n = 400$) | Level 4 ($n = 800$) |
|---|---|---|---|---|---|
| | Interpolation Method | Option Value at $S = \$100$ | | | |
| Black-Scholes: $\sigma_{BS} = .20$ | Linear (axis) | 15.592 | 14.755 | 14.582 | – |
| | Quadratic (axis) | 14.825 | 14.537 | 14.501 | – |
| | Quadratic (diag) | 14.463 | 14.485 | 14.490 | – |
| | Similarity | 14.463 | 14.485 | 14.490 | 14.491 |
| | Interpolation Method | Option Value at $S = \$100$ | | | |
| CEV: $\alpha = 1.0$ | Linear (axis) | 11.233 | 10.437 | 10.288 | – |
| | Quadratic (axis) | 10.572 | 10.245 | 10.212 | – |
| | Quadratic (diag) | 10.168 | 10.196 | 10.203 | – |

TABLE 9: *Comparison of interpolation techniques on solution of a shout put option with four shout opportunities per year. Parameters used: $\sigma_{BS} = .20$, $r = .10$, $K_0 = \$100$, $S_0 = \$100$, $T = 5$ years, $K_{max} = 10,000$. Notice the slow convergence of the axis aligned interpolation methods. Solutions for the full three dimensional problem were not computed for Level 4 due to computational memory constraints. CEV model defined by equation (21).*

discretization for $K = K_0$. Since we are studying a contract which resets the strike to the current asset level, the constraint $V^*$ can be determined without interpolation for the line $K = K_0$ (provided that the diagonal $S = K$ is in the $S$ discretization for fixed $K$). In this way we have minimized discretization error and kept interpolation effects as distant as possible from $K = K_0$, the problem of interest. If there are multiple shout opportunities (including shout counter reset features), then interpolation errors can be introduced indirectly into the solution since the $K = K_0$ problem will be using solutions for $K \neq K_0$ which have not been constructed to be free of interpolation error. It is this source of error which we now study.

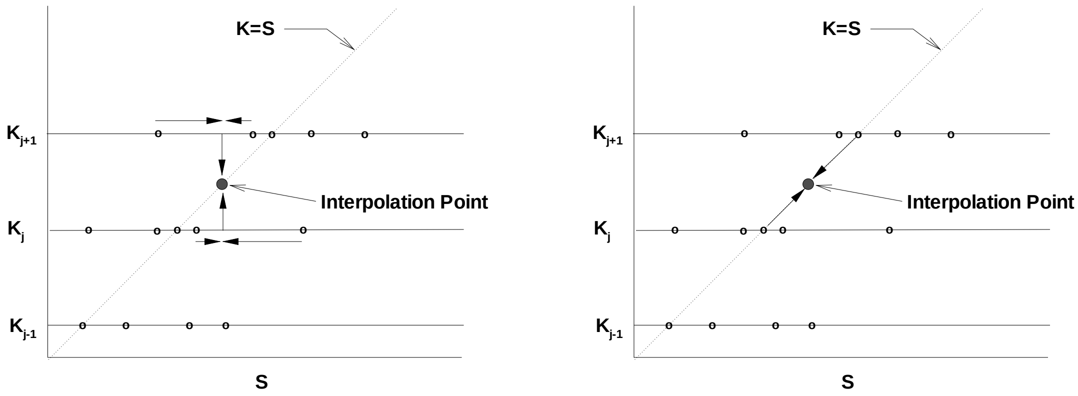
The most straightforward interpolation scheme which can be expected to converge is axis aligned linear interpolation. We are interpolating in the Cartesian plane $S \times K$. Axis aligned interpolation refers to interpolation methods which interpolate the point $(S, K)$ by interpolating along lines of $K = const.$ and then interpolating these values along lines of $S = const.$ In practice, the discretization in the $S$ direction is refined in the region where interpolation is expected (in our example at the strike). Therefore, $\Delta S \ll \Delta K$ and the majority of the interpolation error occurs in the $K$ interpolation. Omitting the dependence on $U$ and $t$ for clarity, linear interpolation is defined as

$$V(S, K) = V_S(K_{lo}) + \frac{V_S(K_{hi}) - V_S(K_{lo})}{K_{hi} - K_{lo}}(K - K_{lo}),$$

where

$$V_S(K) = V(S_{lo}, K) + \frac{V(S_{hi}, K) - V(S_{lo}, K)}{S_{hi} - S_{lo}}(S - S_{lo})$$

and $S_{lo} \leq S \leq S_{hi}$, $K_{lo} \leq K \leq K_{hi}$. As anticipated from Figure 4, linear interpolation severely overestimates the value of the contract (see Table 9).




(a) *Axis aligned interpolation.*

(b) *Diagonal interpolation along the strike setting curve $K = S$.*

FIGURE 5: *Different interpolation strategies.*

As a second attempt we can increase the order of the interpolation method while maintaining the axis aligned interpolation direction. Quadratic axis aligned interpolation is defined as

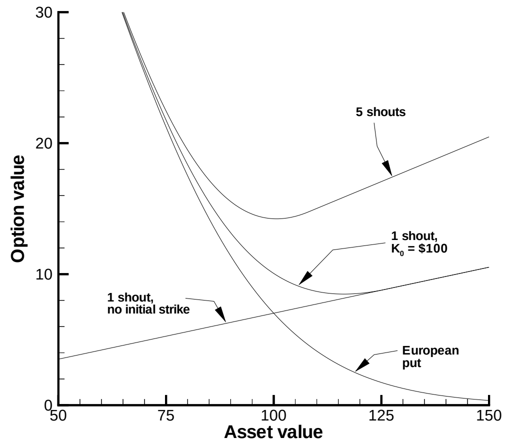
$$V(S, K) = c_{lo} V_S(K_{lo}) + c_{hi} V_S(K_{hi}) + c_{2up} V_S(K_{2up})$$

where the interpolation coefficients are derived from the usual Lagrange quadratic basis functions. The interpolation points are selected such that $K_{lo} \le K \le K_{hi}$. The third point $2up$ in the quadratic interpolation is selected to be the closest neighbouring node. The values $V_S(K)$ are defined analogously. The interpolated value is then limited so that it lies between the nearest neighbour values along the coordinate axes. Although quadratic axis aligned interpolation gives better results than linear axis aligned interpolation, the convergence is still slow (see Table 9).

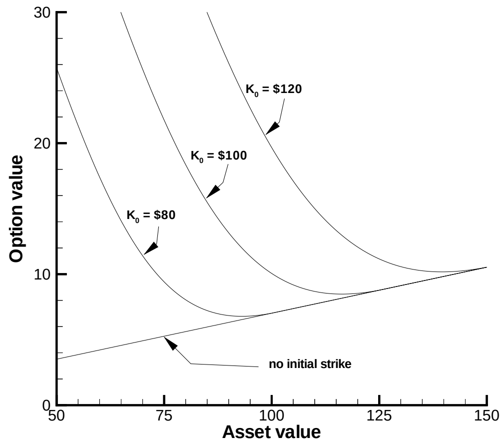
In cases where the strike setting upon shouting is independent of the current strike setting, i.e. $K^* = \mathcal{F}(S, U, t)$, we can define the curve $(S, K^*(S))$ for fixed $U$ and $t$. In the case $K^* = S$, the solution is smoother along this curve than it is along curves of constant $S$ or $K$ as used in the axis aligned interpolation. The two interpolation strategies are shown in Figure 5.

In fact, since this contract has a similarity solution, the value is linear along the diagonal $K = S$. Therefore, quadratic interpolation along the diagonal is exact as can be seen by the comparison between the quadratic diagonal interpolation and the similarity solution (which has no interpolation error) in Table 9. In the case of a CEV model (see equation (21)), quadratic interpolation along the diagonal is no longer exact. However, Table 9 shows that this interpolation method is still very effective compared with axis aligned interpolation. Consequently, in the following examples we will use quadratic (diagonal) interpolation for pricing options in the full three dimensional case, for $K^* = S$.

The computed results all appear to be consistent with quadratic convergence, as expected




(a) *Shout put options with one and five shout opportunities (no shout counter resets) and initial floor set at $K_0 = \$100$. Also shown are a European put option with strike $K = \$100$ and a shout put option with no initial strike setting.*




(b) *Shout put options with various initial strike settings. One shout opportunity.*

FIGURE 6: *Comparison of various shout options. Parameters used: $\sigma_{BS} = .25$, $r = .06$, and $T = 1$ year.*

from the error analysis. If we assume that $\Delta t = C_1 \Delta S = C_2 \Delta K$, where $C_1, C_2$ are constants, then the total work is

$$work \ = O\left(\frac{1}{(\Delta t)^3}\right).$$

Since the convergence rate is $O((\Delta t)^2)$, we have that

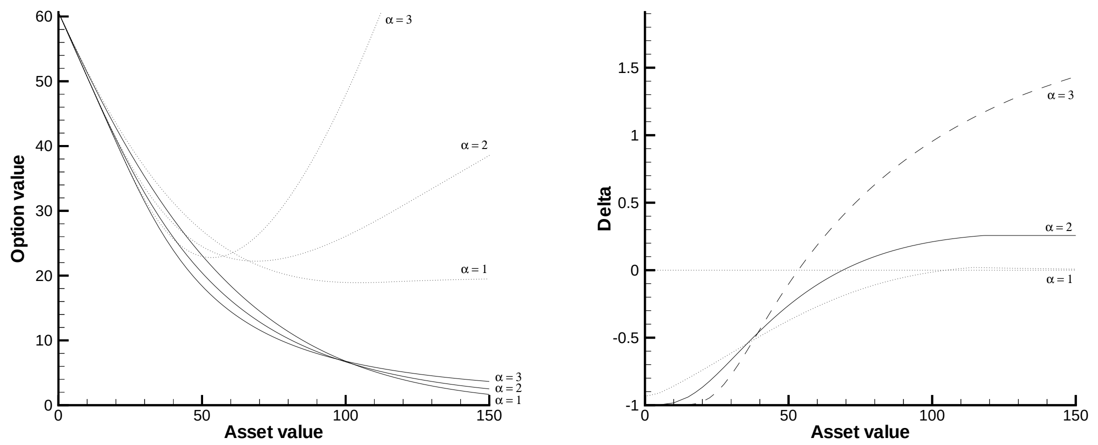
$$convergence \ = O\left(\frac{1}{(work)^{2/3}}\right).$$

# 5 Examples

In order to gain insight into the properties of shout options, we will consider a few instructive examples. This intuition will be helpful when faced with the task of understanding more complex financial instruments. Further financial results are provided in [20].

## 5.1 Limiting Asymptotic Behaviour

Consider a shout put option where the holder of the security has the ability to reset the strike price to the current asset level. Since the holder can always choose not to reset the strike from its initial setting, these contracts must always be worth at least as much as a European put option with the same initial strike. This is shown in Figure 6(a). As the underlying asset price $S \to \infty$ it becomes profitable for the holder of the shout put option




(a) *Values of shout put option (dotted) and standard European put option (solid).*

(b) *Delta* ($\frac{\partial V}{\partial S}$) *of shout put option.*

FIGURE 7: *The effects of various CEV process assumptions. The shout option has a single exercise opportunity. Parameters used:* $\sigma_{BS} = .25$, $r = .06$, $K_0 = \$100$, $S_0 = \$100$, $T = 5$ *years.*

to reset the strike to the higher asset level since the current strike setting is unlikely to be in-the-money at expiry. In Figure 6(b) we can see that as $S \to \infty$ the shout put options with various initial strike settings have *exactly* the same value since upon shouting holders will all receive the same contract independent of the current strike setting.

In this case we see that for asset levels, $S \approx \$150$, it is no more valuable to have a current strike setting of $K_0 = \$120$ than it is to have a strike of $K_0 = \$80$, or even no initial strike at all.

## 5.2 Effects of Nonconstant Volatility

Recent empirical evidence has led researchers to question the validity of the constant volatility Black-Scholes model for standard equity options [2]. The CEV model (see equation (21) ) provides a simple alternative. As noted in [10], various types of volatility "smiles" can be modelled by appropriate choices of the CEV parameter $\alpha$. In particular, there is a negative volatility skew if $\alpha < 2$, and a positive volatility skew if $\alpha > 2$. It is also worth noting that recent research [5, 10] has found that computed prices of some types of exotic options are very sensitive to the value of $\alpha$.

Figure 7 presents some results for shout options with values of $\alpha$ of 1, 2, and 3, as well as for a standard European put. The value of shout options is strongly affected by the volatility skew; especially for high $S$ values, i.e. in the exercise region, while the value of a European put is only weakly affected by changing $\alpha$. This can be explained by realizing that the at-the-money put option received upon shouting will be priced using a higher reference volatility for the positive volatility skew. The lower value for high asset values for the negative volatility skew model is similarly explained.

One of the most important results in option pricing theory is the ability for the writer of a derivative security to perfectly hedge (at least in theory) his risk exposure using a dynamic trading strategy. In the limit as trading is allowed continuously this can be achieved using the *delta hedging* strategy where the writer holds $\Delta = \partial V/\partial S$ in the risky asset and an appropriate amount to make the trading strategy self financing [12] in a risk free money market account. Figure 7(b) also shows the value of $\Delta$ for different values of asset price and $\alpha$. This figure shows the the hedging strategy for shout options is also dramatically different under the various volatility models. Although discrepancies due to deviation from the $\sigma(S,t) = const.$ model may be small when valuing standard options, care should be taken when using simple models to value and hedge more exotic contracts.

More realistic fits for the empirically observed or market implied volatility can be produced using a volatility surface which is a general function of the underlying asset $S$ and time $t$ [1, 8]. In the case of a simple shout option the contracts received upon shouting are vanilla options. The volatility surface can be thought of as a method to interpolate the values of traded options, increasing the consistency of the model with present market prices. We emphasize that there is no difficulty using a volatility surface in the algorithm presented here.

# 6 Conclusions

In this paper we have developed a method which can be used to value complex types of shout options. Such options may contain multiple exercise rights, reset features and/or complicated rules to determine the contract received upon shouting. Since the strategy followed by the holder in exercising these rights is unknown at the time of sale, the writer of the contract is faced with the task of super-replicating any exercise policy which could be adopted by the holder.

In order to value this contract, several numerical issues must be addressed:

- Since it is necessary to artificially truncate the solution domain, it is not obvious what boundary conditions should be imposed on the computational domain. It was shown that a modified problem could be solved, on the truncated domain, where all boundary data was known. As the size of the computational domain is increased, the modified problem becomes equivalent to the original problem.

- The numerical solution must be interpolated at each timestep in order to determine the algebraic constraint to be applied. We have discussed various approaches which can be applied to minimize this source of error. A quadratic interpolation along the strike-setting curve produced more rapid convergence compared to alternate methods.

- The long term nature of these contracts makes it necessary to use an efficient timestepping algorithm. Several common methods were compared. It is important to note that the second order multi-step BDF algorithm was unsuitable for this type of numerical problem because of the non-smoothness in the time domain.

The flexible object-oriented nature of the framework allows separation of the modelling and contract dependent aspects. Furthermore, the highly parallel nature of the valuation algorithm gives interesting possibilities on modern parallel computers. Using a general numerical PDE approach, we are able to explore the effects of incorporating various exercise features in contracts. Some of these more exotic contracts were shown to be very sensitive to departures from standard modelling assumptions.

# References


[1] L.B.G. Andersen and R. Brotherton-Ratcliffe. The equity option volatility smile: An implicit finite difference approach. *Journal of Computational Finance*, 1:5–37, Winter 1998.

[2] G. Bakshi, C. Cao, and Z. Chen. Empirical performance of alternative option pricing models. *Journal of Finance*, 52:2003–2049, 1997.

[3] J. Becker. A second order backward difference method with variable timesteps for a parabolic problem. *BIT*, 38:644–662, 1998.

[4] P. P. Boyle, A.W. Kolkiewicz, and K.S. Tan. Valuation of the reset option in segregated fund contracts using Quasi-Monte Carlo. Research Report 99-10, Institute of Insurance and Pension Research, University of Waterloo, 1999.

[5] P.P. Boyle and Y. Tian. Pricing lookback and barrier options under the CEV process. *Journal of Financial and Quantitative Analysis*, 34:241–264, 1999.

[6] M. Broadie and P. Glasserman. A stochastic mesh method for pricing high dimensional American options. *Proceedings of the Eighth Annual Derivative Securities Conference*, Boston, 1998.

[7] T.H.F. Cheuk and T.C.F. Vorst. Shout floors. *Net Exposure*, 2, November 1997.

[8] T.F. Coleman, Y. Li, and A. Verma. Reconstructing the unknown local volatility function. *Journal of Computational Finance*, 2(3):77–102, 1999.

[9] J.C. Cox. Notes on option pricing I: Constant elasticity of variance diffusions. Working paper, Stanford University (reprinted in *Journal of Portfolio Management* 22 (1996), 15–17), 1975.

[10] D. Davydov and V. Linetsky. Pricing options on one-dimensional diffusions: A unified approach. Working paper, University of Michigan, 1999.

[11] A.K. Dixit and R.S. Pindyck. *Investment Under Uncertainty.* Princeton University Press, 1994.

[12] D. Duffie. *Dynamic Asset Pricing Theory.* Princeton University Press, Princeton, New Jersey, Second edition, 1996.

[13] P.A. Forsyth, K.R. Vetzal, and R. Zvan. Convergence of lattice and PDE methods for pricing Asian options. *Proceedings of the Ninth Annual Derivative Securities Conference*, Boston, 1999.

[14] P. Jaillet, E.I. Ronn, and S. Tompaidas. Valuation of commodity-based "swing" options. *Proceedings of the Eighth Annual Derivative Securities Conference*, Boston, 1998.

[15] C. Johnson. *Numerical Solutions of Partial Differential Equations by the Finite Element Method.* Cambridge University Press, Cambridge, 1987.

[16] J.D. MacBeth and L.J. Merville. Tests of the Black-Scholes and Cox call option valuation models. *Journal of Finance*, 35:285–301, 1980.

[17] B. Thomas. Something to shout about. *RISK*, 6:56–58, May 1993.

[18] P. Wilmott. *Derivatives.* John Wiley and Sons, West Sussex, England, 1998.

[19] P. Wilmott, S. Howison, and J. Dewynne. *Option Pricing: Mathematical Models and Computation.* Oxford Financial Press, 1993.

[20] H. Windcliff, K.R. Vetzal, and P.A. Forsyth. Shout options: Pricing the reset feature in segregated funds. Northern Finance Association Annual Meeting, Calgary, 1999.

[21] R. Zvan, P.A. Forsyth, and K.R. Vetzal. Penalty methods for American options with stochastic volatility. *Journal of Computational and Applied Mathematics*, 91:119–218, 1998.

[22] R. Zvan, P.A. Forsyth, and K.R. Vetzal. Robust numerical methods for PDE models of Asian options. *Journal of Computational Finance*, 1(2):39–78, 1998.