International Journal of Theoretical and Applied Finance
Vol. 16, No. 8 (2013) 1350046 (25 pages)

DOI: 10.1142/S0219024913500465

NUMERICAL SCHEMES FOR OPTION PRICING IN REGIME-SWITCHING JUMP DIFFUSION MODELS

IONUT FLORESCU

Financial Engineering, School of Systems and Enterprises
Stevens Institute of Technology, Castle Point on Hudson
Hoboken, NJ 07030
ifloresc@stevens.edu

RUIHUA LIU

Department of Mathematics
University of Dayton, 300 College Park
Dayton, OH 45469-2316
rliu01@udayton.edu

MARIA CRISTINA MARIANI* and GRANVILLE SEWELL†

Department of Mathematical Sciences
The University of Texas at El Paso
Bell Hall 124, El Paso, TX 79968-0514
**mcmariani@utep.edu*
†*sewell@utep.edu*

Received 26 August 2013
Accepted 11 November 2013
Published 18 December 2013

In this paper, we present algorithms to solve a complex system of partial integro-differential equations (PIDE's) of parabolic type. The system is motivated by applications in finance where the solution of the system gives the price of European options in a regime-switching jump diffusion model. The new algorithms are based on theoretical analysis in Florescu *et al.* (2012) where the proof of convergence of the algorithms is carried out. The problems are also solved using a more traditional approach, where the integral terms (but not the derivative terms) are treated explicitly. Another contribution of this work details a novel type of jump distribution. Empirical evidence suggests that this type of distribution may be more appropriate to model jumps as it makes them more clearly distinguishable from the signal variability.

Keywords: Numerical algorithms; system of partial integro-differential equations; regime-switching jump diffusion; option pricing; implicit and explicit finite element methods.

1. Introduction

In this work, we present numerical algorithms to solve a complex system of partial integro-differential equations (PIDE's) of parabolic type. Our study is mainly

motivated by applications in mathematical finance. It is well-known that the problem of pricing financial derivatives often leads to solving partial differential and/or integral equations. Due to the needed consideration of stochastic volatility, stochastic interest rate, and discontinuous jumps in the mathematical models describing the dynamics of the underlying asset prices, the complexity of these equations has increased significantly in recent years. In general, analytical solutions are not available for these problems and thus in practical applications numerical methods are employed.

The new algorithms presented in the present paper are based on the theoretical work in [12] where we study a general system of PIDE's modeling the option pricing problem in a regime-switching jump diffusion model. We prove the existence of solution to the PIDE system via a method of upper and lower solutions. We construct monotonic sequences of solutions started from the lower solution whose limit is a strong solution of the PIDE system. This constructive proof suggests a numerical scheme allowing to compute approximating option prices. This scheme is different from the existing literature, thus prompting the current study.

Pioneered by [24], jump diffusion models have been a main topic of study in financial mathematics. The rational for including a jump component in a diffusion model is due to the fact that the driving Brownian motion is a continuous process, which means that the process has difficulties fitting the market data with large fluctuations. Thus, the necessity of taking into account large market movements, as well as a great amount of information arriving suddenly (i.e. a jump) has led to the study of jump diffusion models. On the other hand, aiming to include the influence of macroeconomic factors on the behavior of individual asset prices, considerable attention has been drawn to regime-switching models in recent years. See [15, 23]. In this regime-switching setting, asset prices are modeled by several stochastic differential equations coupled by a finite-state Markov chain, which represents various randomly changing economical factors. Mathematically, the regime-switching models generalize traditional models in such a way that the coefficients in the models depend on the respective state of the Markov chain. This type of asset models results in a system of coupled PDE's (or PIDE's) for the problem of pricing options. To model both jump and regime-switching in an appropriate way, a regime-switching jump diffusion model is posed in [12]. This model is numerically solved in the present work.

Option pricing using regime-switching jump diffusion models has been studied by several authors in the literature. Elliott *et al.* in [11] employ a generalized regime-switching Esscher transform to determine an equivalent martingale measure that is used for pricing options. They also derive a system of coupled partial-differential-integral equations satisfied by the European option prices. Yuen and Yang in [34] develop a trinomial tree method to calculate approximate option prices. Siu *et al.* in [30] study the problem of valuing participating life insurance products under a generalized jump diffusion model with a Markov-switching compensator. They develop a numerical algorithm based on the Monte-Carlo simulations. Recently, [26]

proposes a pricing method based on Fourier Transform for the Merton jump diffusion with regime-switching where the jump sizes are assumed to follow a lognormal distribution. In [4], the authors develop algorithms for pricing double barrier options in a regime-switching hyper-exponential jump-diffusion model. This last model uses a mixture of any number of exponential distributions for jumps (generalizing Kou's log double exponential model which is a mixture of just two distributions). The approach in this latter paper is based on a combination of Carr's randomization and the Wiener–Hopf factorization.

Along another line, there exists a rich literature on numerical schemes for solving the PIDEs for pricing options in jump-diffusion models. See [1–3, 6–10, 14, 20, 21, 27, 28, 31, 32], among others. However, these papers treat a single PIDE arising from using a jump-diffusion model without considering the possible change of regime. On the other hand, numerical solutions have been developed for systems of PDEs for options in regime-switching models without inclusion of jumps. See [5, 16–18], among others.

The numerical schemes presented in the current paper, based on a constructive proof of an existence theorem, are very different from the above-mentioned methods. To our best knowledge, the present paper is the first attempt on numerical solutions of the system of PIDEs using a regime-switching jump diffusion model. The algorithms developed herein may be particularized to work with models in any one of the papers mentioned by simply choosing the appropriate jump density $g(y)$. Moreover, we numerically examine a new class of jump distributions, namely a lognormal mixture. This class of distributions seems to be more useful for modeling jumps since it is easy to understand the jump characteristics as well as making the jumps more clearly distinguishable from the signal variability. We refer the reader to Sec. 4.1 for a detailed discussion of this distribution and its characteristics.

The algorithms described in this paper may be implemented using any PDE solver. In our work we used PDE2D, a general purpose partial differential equation solver available from Rogue Wave, Inc. (www.roguewave.com/pde2d or www.pde2d.com), which has been used to solve many mathematical finance applications [33]. PDE2D is able to solve linear or non-linear, steady-state, time-dependent and eigenvalue problems, in 1D intervals, general 2D regions (with curved boundaries), and a wide range of simple 3D regions. It has a sophisticated GUI interface which makes it extremely easy to use. PDE2D can solve 1D, 2D or 3D problems similar to the problems presented, using a collocation finite element method with cubic Hermite basis functions to discretize the spatial derivatives and implicit finite difference methods to discretize the time derivatives. Without the integral term, a problem such as the system resulting from a regime-switching model is a straightforward PDE2D application, and can be solved with little user effort. The integral term in the current problem resulting from the jumps, required special efforts which are detailed in the paper.

The paper is organized as follows. Section 2 presents the mathematical model of regime-switching jump diffusion as well as the system of PIDE's needed to be solved

in order to calculate European type option prices. Section 3 presents two types of algorithms for numerically solving the PIDE system. The type 1 algorithms (two of them) are new to the PIDE system considered in this paper and are described in detail. For the purpose of comparison, we also present what we call a type 2 algorithm which is based on a more traditional approach, in which the integral terms are treated explicitly in time, while the derivative terms are treated implicitly. In Sec. 4, we exemplify these algorithms using numerical examples. We first give a discussion about the jump distributions used. We follow with three different examples. The algorithms are implemented and tested for these examples. Numerical results are reported and compared. Section 5 provides further remarks and concludes the paper.

2. Problem Formulation

In this section, we present the regime-switching jump diffusion model used for asset price evolution and the system of PIDE's for pricing European options.

2.1. *Model description*

We assume as given a complete probability space $(\Omega, \mathcal{F}, \mathbf{P})$, endowed with a filtration $\{\mathcal{F}_t\}_t$ on which all the stochastic processes introduced in this paper are defined, respectively adapted to. Here, $\mathbf{P}$ is the real-world probability measure. In this space let B_t be an 1D standard Brownian motion and α_t be a continuous-time Markov chain taking values in the set $\mathcal{M} := \{1, \ldots, m\}$. Let the intensity matrix (or the generator) $Q = (q_{ij})_{m\times m}$ of α_t be given. In this context, the elements q_{ij}, $i, j = 1, \ldots, m$ of the generator satisfy:

(I) $q_{ij} \geq 0$ if $i \neq j$;
(II) $q_{ii} = -\sum_{j\neq i} q_{ij}$ for each $i = 1, \ldots, m$.

We assume that the Brownian motion B_t and the Markov chain α_t are independent. The process α_t governs the times when the regime is switched as well as the values of regime.

We use a Cox process (a specialized non-homogeneous Poisson process) N_t with regime-dependent intensity λ_{α_t} to model the random jump times. Thus, if the current regime is $\alpha_t = i$, then the time until the next jump is given by an exponential random variable with mean $1/\lambda_i$. Here, N_t counts the total number of jumps in the asset price up to time t. Let the jump sizes be given by a sequence of iid random variables Y_k, $k = 1, 2, \ldots$, with probability density $g(y)$. Assume that Y_k, $k = 1, 2, \ldots$, are independent of B_t and α_t.

We consider a risky asset whose price S_t under the real-world probability measure $\mathbf{P}$ is governed by the regime-switching jump diffusion:

$$\frac{dS_t}{S_{t-}} = \mu_{\alpha_t} dt + \sigma_{\alpha_t} dB_t + dJ_t, \quad t \geq 0, \tag{2.1}$$

where S_{t-} denotes the asset price right before the time t, μ_{α_t} and σ_{α_t} are the expected rate of return and the volatility rate, respectively; J_t is the jump component given by

$$J_t = \sum_{k=1}^{N_t} (Y_k - 1). \tag{2.2}$$

$Y_k - 1$ represents the percentage by which the asset price changes at the kth-jump. Note that, in between regime-switching times the asset process follows a regular jump diffusion with constant coefficients. However, the coefficients are switching as governed by the corresponding state of the Markov chain. Also note that, in the model setting (2.1) the volatility is modeled as a finite-state Markov chain σ_{α_t}. This may be considered as a discrete approximation of a continuous-time diffusion model for the stochastic volatility (e.g., the Heston's model). We refer the reader to [22] for further discussions.

2.2. *The system of PIDE's for European options*

Given that the asset price process follows the regime-switching jump diffusion model (2.1), we look into the problem of option pricing. To this end let r_{α_t} denote the risk-free interest rate corresponding to the state α_t of the Markov chain.

To simplify exposition, we consider European options written on a single asset S_t with maturity $T < \infty$. The system of PIDE's for options written on multiple correlated assets can be derived in a similar manner. In Sec. 4.4, we give an example where the option is written on two assets, each following a regime-switching jump diffusion process. The derivation is entirely similar. Let $V_i(S,t)$ denote the option value at time t, when the asset price $S_t = S$ and the regime $\alpha_t = i$ (assuming that the regime α_t is observable). Under these assumptions we change the time variable from t to $T-t$ and rewrite the PIDE's in forward form. This step is not necessary but most PDE solvers are constructed for forward rather than backward equations. We present the algorithms in forward form for convenience of implementation. Assuming that the value functions $V_i(S,t)$, $i = 1, \ldots, m$, satisfy appropriated regularity conditions, for example $V_i(S,t) \in C^2(\Pi, \mathbb{R}) \cap C(\overline{\Pi}, \mathbb{R}) \cap L^1(\Pi, \mathbb{R})$ where Π denotes the domain of S, they verify the following system of PIDE's [12, 15, 25]:

$$\begin{aligned} &\frac{1}{2}\sigma_i^2 S^2 \frac{\partial^2 V_i}{\partial S^2} + (r_i - \lambda_i \kappa) S \frac{\partial V_i}{\partial S} - (r_i + \lambda_i - q_{ii}) V_i - \frac{\partial V_i}{\partial t} \\ &\quad = -\lambda_i \int V_i(Sy, t) g(y) dy - \sum_{j \neq i} q_{ij} V_j, \end{aligned} \tag{2.3}$$

where $\kappa = E[Y-1] = \int (y-1) g(y) dy$. Here, Y denotes the jump size random variable. The boundary conditions are expressed at $t = 0$ and when $S \to 0$ and $S \to \infty$.

To determine the option prices corresponding to different regimes, the system (2.3) needs to be solved for various boundary conditions which depend on the

particular option being priced. To our best knowledge, there exists no known analytical solution to such complicated system of PIDE's. In [12], we analyze a general version of such a system of PIDE's. In that work we proved that, under suitable regularity conditions on the parameters as well as boundary conditions, the system admits a solution. Our proof of the main theorem [12, Theorem 1] was using the method of upper and lower solutions. We construct monotonic sequences of approximating solutions whose limit is a strong solution of the PIDE system. This constructive proof provides a natural numerical scheme to computing approximating solutions of the system. As we shall describe in the next section, this scheme is different from the existing literature for the PIDE system we consider in this paper.

Remark 2.1. In this paper, the market regime α_t is assumed to be observable and hence the option price can be identified with the regime information. However, in practice, information of α_t is usually hidden in noisy market data and not directly observable. When the modulating Markov chain is unobservable, the volatility term in (2.1) is unobservable too. The volatility term may be identified by using the predictable quadratic variation of the observed share price process. Furthermore, if the modulating Markov chain is hidden, it is necessary to develop non-linear filtering methods (for example, the Wonham filter) for estimating the conditional probabilities of α_t given the stock process. It can be used in connection with our numerical algorithms to determine the option prices. We leave this as an interesting project for future research.

3. Description of Algorithms

In this section, we present two different types of algorithms. The algorithms of type 1 (we call them Algorithms 1a and 1b) are suggested by the constructive proof presented in [12, Theorem 1]. These algorithms work if upper/lower solutions of the problem exist (see the definitions and further discussions in [12]). They may be considered as implicit type algorithms since the integral terms are evaluated at the same time point as the derivative terms, not at a previous time point. These implicit algorithms can start from either a lower solution or an upper solution at the first step. In the numerical examples presented in Sec. 4, the algorithms are used to calculate option prices which are always non-negative, therefore the zero solution $V_i(S,t) = 0 \forall\, i, S, t$ can be chosen as an initial lower solution. In our implementation, the first iteration always starts from this particular lower solution. The algorithms would converge faster by choosing a better (closer to true value) lower (or upper) solution. The similar ideas in literature of which we are aware are [10] and [3]. These algorithms are designed for the simple jump diffusion model (without regime switching or stochastic volatility) and they may be viewed as a local fixed point algorithm [10] and a global fixed point iteration algorithm [3], both different from the algorithms proposed herein. In fact the closest idea is our previous work [13] and [14] developed in the context of PIDE's obtained for stochastic volatility and

jumps. The algorithms presented next extend this work to regime-switching models as well as jumps.

The algorithm of type 2 (called Algorithm 2) is a more traditional algorithm, in which the integral terms are treated explicitly in time, though the derivative terms are treated implicitly. Specifically, this algorithm propagates in time from the initial condition at time zero using a classical finite element scheme. Indeed variants of this algorithm for simple jump-diffusion processes have been analyzed in [8] using a so-called implicit-explicit method (the implicit part refers to dealing with the jump component using a "compensated vanishing viscosity" approximating solution).

We present the algorithms for a general PIDE system with $d \geq 1$ underlying asset variables $S = (S_1, \ldots, S_d)$. This general system is treated in [12] which includes both the single asset (2.3) and two assets (4.5) as special cases. Let the differential operator $\mathcal{L}_i$ be defined by

$$\mathcal{L}_i V_i = \sum_{j=1}^{d} \sum_{k=1}^{d} a^i_{jk}(S,t) \frac{\partial V_i}{\partial S_j \partial S_k} + \sum_{j=1}^{d} b^i_j(S,t) \frac{\partial V_i}{\partial S_j} + c^i(S,t) V_i, \quad i = 1, \ldots, m, \tag{3.1}$$

and the integral operator $\mathcal{G}_i$ by

$$\mathcal{G}_i(t, V_i) = -\lambda_i \int V_i(S_1 y_1, \ldots, S_d y_d, t) g(y_1, \ldots, y_d) dy_1 \cdots dy_d, \tag{3.2}$$

where the coefficients a^i_{jk}, b^i_j and c^i, $i \in \{1, \ldots, m\}$; $j, k \in \{1, \ldots, d\}$ satisfy appropriate conditions [12], and $g(y_1, \ldots, y_d)$ denotes the joint probability density of the d-dimensional jump size random variables.

Using these notations we rewrite the problem with general boundary conditions on the general domain $\Omega \subseteq \mathbb{R}^d$ as:

$$\begin{cases} \mathcal{L}_i V_i - \dfrac{\partial V_i}{\partial t} = \mathcal{G}_i(t, V_i) - \displaystyle\sum_{j \neq i} q_{ij} V_j & \text{in } \Omega \times (0,T) \\ V_i(S,0) = V_{i,0}(S) & \text{on } \Omega \times \{0\} \\ V_i(S,t) = h_i(S,t) & \text{on } \partial\Omega \times (0,T) \end{cases} \quad \text{for } i = 1, \ldots, m, \tag{3.3}$$

where $V_{i,0}$ and h_i specify the boundary conditions and must satisfy the compatibility condition

$$h_i(S,0) = V_{i,0}(S), \quad \text{for any } S \in \partial\Omega,\ i = 1, \ldots, m. \tag{3.4}$$

The algorithms presented next provide numerical procedures for solving the system (3.3). For notational concision, we may use $V = \{V_i, 1 \leq i \leq m\}$ for the collection of m solution functions to (3.3).

3.1. *Algorithm 1a*

Algorithm 1a follows the procedure used to construct the convergent sequences in [12]. We have to make sure that there exist a lower solution $\alpha()$ and an upper

solution $\beta()$ of the problem and that the operators $\mathcal{L}_i$ and $\mathcal{G}_i$ satisfy the hypotheses of the theorem. As noted before, in financial applications when we price options, $\alpha() = 0$ is always a lower solution of the problem.

Algorithm 1a is outlined in the following.

(1) We start with $V^0 = \alpha$, the lower solution.
(2) For every $n \geq 0$, we find an approximated solution V^{n+1} of the problem

$$\begin{cases} \mathcal{L}_i V_i^{n+1} - \dfrac{\partial V_i^{n+1}}{\partial t} = \mathcal{G}_i(t, V_i^n) - \displaystyle\sum_{j \neq i} q_{ij} V_j^n & \text{in } \Omega \times (0,T) \\ V_i^{n+1}(S,0) = V_{i,0}(S) & \text{on } \Omega \times \{0\} \\ V_i^{n+1}(S,t) = h_i(S,t) & \text{on } \partial\Omega \times (0,T) \end{cases} \quad i = 1, \ldots, m. \tag{3.5}$$

The m PDE's in this system cannot be solved analytically. Therefore, we implement a finite element scheme to numerically solve them.

- We take a $d+1$ dimensional grid[1] in state variables $t \in [0,T]$, and $S_k \in [0, S_{max}^k]$, $k \in \{1, \ldots, d\}$. The upper bounds $S_{\max}^k$ are suitably chosen large numbers. This grid is kept the same for all iterations.
- We approximate the integrals in $\mathcal{G}_i(t, V_i^n)$ using a midpoint rule, with the required values of $V_i^n(S,t)$ interpolated from the values saved at all the points on the grid, using quadratic interpolation in the components of S and linear interpolation in t.
- Since the equations are iteratively solved forward in time, the boundary conditions are set at $t = 0$ and we solve the resulting PDE's starting from time $t = 0$ by using a PDE solver — we use PDE2D [29] in our experiments.
- The result is V^{n+1} calculated at all grid points and to be used in the next iteration.

(3) The sequence $\{V^n, n \geq 1\}$ converges to the solution of the main system (3.3). Thus, the algorithm stopping criteria is that the maximum difference between two consecutive iterations at all points on the grid is less than a pre-specified error tolerance.

We note that in this algorithm at every iteration we solve a separate PDE for each i. This is due to the fact that the coupling term $\sum_{j \neq i} q_{ij} V_j^n$ uses the previously found solution so in fact this term is known when solving for the variables V^{n+1}. This treatment for the coupling term is particularly suitable for parallel computation, using m processors, one for each PDE. If this is done, each processor must access the solutions saved on the previous iteration by the other processors, but otherwise there is no communication between processors, because (unlike Algorithms 1b and 2) the processor solving equation i does not need to know the solutions on

[1]In the numerical examples we present later $d \leq 2$.

the current iteration being calculated by the other processors. So, there is almost no communication between processors, which means excellent parallel performance would be expected. In fact, we have now implemented Algorithm 1a in parallel and applied it to Example 2 of Sec. 4.3 (call option, Case A) which involves $m = 4$ equations, and thus four processors are used, and the computer time decreased from 34:20 to 3:12, with identical results (the time for the original version does not agree with Table 5 because this was run on a different, multi-processor, system). The reason the observed speed-up is even greater than the number of processors used is that our original implementation solved the four equations as a coupled system, for convenience, even though for Algorithm 1a the m equations are uncoupled: solving the four equations sequentially, on a single processor, as four separate uncoupled equations, decreased the time from 34:20 to 11:49. Then when this new program was parallelized, a further decrease in time resulted, to 3:12.

3.2. *Algorithm 1b*

Algorithm 1b is a slight modification of Algorithm 1a. Based on the previous observation we hope to speed up the solver by treating the coupling terms as unknown variables. However, this will lead to solving a system of PDE's as opposed to solving separate PDE's as in Algorithm 1a. Specifically, the equations in (3.5) are replaced with the following system of PDE's:

$$\begin{cases} \mathcal{L}_i V_i^{n+1} - \dfrac{\partial V_i^{n+1}}{\partial t} + \displaystyle\sum_{j \neq i} q_{ij} V_j^{n+1} = \mathcal{G}_i(t, V_i^n) & \text{in } \Omega \times (0,T) \\ V_i^{n+1}(S,0) = V_{i,0}(S) & \text{on } \Omega \times \{0\} \\ V_i^{n+1}(S,t) = h_i(S,t) & \text{on } \partial\Omega \times (0,T) \end{cases} \quad i = 1, \ldots, m. \tag{3.6}$$

This system of PDE's can be numerically solved using a finite element scheme as in Algorithm 1a in a very similar manner. The numerical experiments we perform seem to indicate that the approximating sequences $\{V^n, n \geq 1\}$ generated by Algorithm 1b converge in a fewer number of iterations than that generated by Algorithm 1a. This is expected since the coupling term of the system uses the current iteration instead of the previous iteration solution. In terms of computational time, in our implementations Algorithm 1b is also faster than Algorithm 1a, but as mentioned in the previous section, it is possible to speed up Algorithm 1a substantially by solving the uncoupled PDEs of the system separately.

3.3. *Algorithm 2*

Algorithm 2 treats the integral terms explicitly, though it treats the derivative terms implicitly. This algorithm does not use a discretization of the system (3.5) or (3.6), instead it works directly with the original system (3.3). The algorithm works

forward in time from $t = 0$ and provides an approximation solution at $t = T$. Recall that we changed the time from backward to forward so at $t = T$ we in fact obtain the time-zero option values. Specifically:

- We start with the initial condition at time $t_0 = 0$. That is, set $V_i(S, 0) = V_{i,0}(S)$.
- A PDE solver (in the numerical implementation we used PDE2D [29], which uses a finite element method, but implicit finite difference discretizations for the time derivatives) is used to solve (3.3) forward from t_n to t_{n+1}. The integral terms $\mathcal{G}_i(t, V_i(t))$ are evaluated at time t_n, i.e. $\mathcal{G}_i(t_n, V_i(t_n))$ for which the integrals are approximated using a midpoint rule. The needed values of V_i at t_n are approximated using quadratic interpolation of all the values saved at the S grid points at time t_n.
- When $t = T$ we stop. The approximation solution is obtained for all grid points for S.

3.4. *Comments about the algorithms proposed*

The current work attempts to answer the following questions via numerical experiments.

- Is Algorithm 2 (for which we do not have a proof of convergence) giving the same solution as the two algorithms of type 1 (which we know converge to a solution)?
- Is Algorithm 2 significantly faster than the two algorithms of type 1?

About the first point, we note that only Algorithm 1a is proven to converge to the solution of the main problem (3.3). Algorithm 1b contains an easy modification to speed things up and we have no doubt that it could be proven to converge to the true solution. Algorithm 2 is a natural way to find solutions. It may be possible to extend the analysis of [8] to the system of PIDEs and establish the convergence of the implicit-explicit algorithm. The difficulties may come from the integral term and the coupling term due to regime-switching. This will be studied in the future. Our experience with a similar algorithm from a previous paper [14] (and experience with explicit methods in general) suggests that it may go unstable if Δt is larger than some threshold which may decrease with increasing λ_i values, though with the values of λ_i used in these examples we never saw any evidence of instability even with large Δt values. However, studying the behavior of the resulting solution and comparing with the results provided by the other two algorithms would provide ample indication whether or not the algorithm is producing an accurate solution.

We further note that in the theoretical work [12] we only prove an existence result concerning the solution of the Eq. (3.3). If the three algorithms provide similar solutions to the numerical examples we believe this would indicate that a uniqueness study of the solution may be fruitful.

Concerning the speed of the algorithms we note that in our previous applied work [14] where we worked with a single PIDE resulting from a stochastic volatility model with jumps, an algorithm of type 1 was much slower than an algorithm of

type 2. In the current paper, we are solving a system of PIDE's. As we shall see, the numerical examples seem to indicate that algorithms of type 1 may be practical as well, for some problems.

For the 1D Examples 1 and 2, a non-uniform finite element mesh of 100 cubic elements was used to discretize S, with smaller elements close to $S = K$, where the solutions vary more rapidly. Experiments showed that making this mesh finer did not significantly improve the accuracy, with 100 elements the time discretization was the primary source of error. For the 2D Example 3, a non-uniform 30 by 30 mesh was used to discretize S_1 and S_2. In every test, the same finite element mesh was used for all three algorithms, and the same backward Euler implicit finite difference scheme was used to discretize the time derivatives. Thus, the only significant difference between Algorithms 1 and 2 was the way the integral terms were handled: in Algorithm 2 the solution values used to calculate the integrals came from the previous time step, in Algorithm 1 they came from the current time, but the previous iteration.

4. Numerical Examples

We discuss the various types of distributions used for jumps and then present three examples solved in this work. This is an important discussion; the jumps produce the integral term in the PIDE equations and that term is the part which makes solving the system (3.3) difficult.

4.1. *A discussion about the jump distributions used*

Recall that Y is a random variable with density $g(y)$ representing the percentage change in the stock price after a jump. More precisely, $Y - 1$ is the percentage by which the price drops or rises after such a jump. For example, if $Y - 1 = 0.5$ or $Y = 1.5$ and the price before the jump was $S_1 = \$30$, then the price after the jump becomes $S_2 = 1.5S_1 = \$45$. It is easy to see that jump values of Y close to one do not in fact change the price process significantly and are undistinguishable from the regular observed price fluctuations caused by sampling the process at discrete times.

The first density $g(y)$ for Y attempted in literature was the lognormal distribution proposed by Merton in his classical jump diffusion paper [24]. A random variable is lognormal if and only if its logarithm has a normal distribution. There are two reasons why this particular distribution was chosen.

(1) Since Y is lognormal it may be thought of as $Y = e^Z$ where Z is a normal random variable. Thus, Y can never be negative. This is necessary — recall that the stock price becomes SY after a jump with size Y. Furthermore, zero is an absorbing state for the stock process (i.e. once the equity price reaches zero the company defaults) and no positive shock can take the process out of this state.

(2) Perhaps more interesting from the computational perspective, the underlying stock process distribution remains lognormal with this jump distribution and the author was able to obtain analytical formulae for European type options in this case.

The drawback is that using the lognormal distribution for the jumps means that most of the jumps are small (the jump distribution is centered at one). Thus, few jumps have a reasonable probability of actually being observed. Furthermore, the interesting mathematical property of the stock distribution remaining lognormal means that the return process ($R_t = \log(S_t/S_{t-\Delta t})$) lacks the observed leptokurtic (fat tails) property of the observed data. This was noticed by [19] who proposed to model the jump distribution $g(y)$ using a log double exponential distribution. The resulting returns are leptokurtic and Kou preserves the economical aspects of the stock process. However, despite the increased probability of large jumps, much of the distributional mass of the jump is still concentrated around one, thus there is still a significant chance of producing small jumps, undetectable in practice. This is due to the shape of the log double exponential distribution as may be observed in Fig. 1(b).

To keep the nice leptokurtic and economic features while avoiding this issue of small jumps we are also exemplifying an alternative density, one which we call the lognormal mixture density. This density uses the same idea of transforming a probability density on $\mathbb{R}$ through e^Z into a density on $(0, \infty)$. This density has a smaller amount of mass around one. The name is quite natural as it extends the notion of lognormal to include mixture of normals. This distribution is a natural extension of the single lognormal used by Merton. A lognormal mixture of two normals is displayed in Fig. 1(d). This distribution can be quite naturally generalized to the possibility of modeling moderate and high jumps by having a log mixture of more than two normals.

In what follows the two cases labeled Case A and Case B present the two jump distributions and the specific parameters used in the numerical experiments.

Case A. *Using a log double exponential density.* The logarithm of the jump size Y is a non-symmetric double exponential distribution (a generalization of the Laplace distribution). Specifically, we shall use the following density for Y:

$$g(y) = \begin{cases} p\alpha_1 y^{-\alpha_1 - 1} & \text{if } y > 1 \\ (1-p)\alpha_2 y^{\alpha_2 - 1} & \text{if } 0 < y \leq 1 \end{cases}, \quad \alpha_1, \alpha_2 > 0.$$

This particular jump distribution was proposed initially by [19] (in its Laplace form with $p = 1/2$ and $\alpha_1 = \alpha_2$). In the current study, we use a more general form. As parameter values, we use $\alpha_1 = 3.0465$, $\alpha_2 = 3.0775$, and $p = 0.3445$. These are the values used in [10] and later in [31]. Figures 1(a) and 1(b) present the double exponential and the log double exponential densities respectively.

With this choice of parameters the expected value of the jumps is $E[Y] = 1.007576$ and the variance is $\text{Var}(Y) = 0.3849779$. The corresponding value for

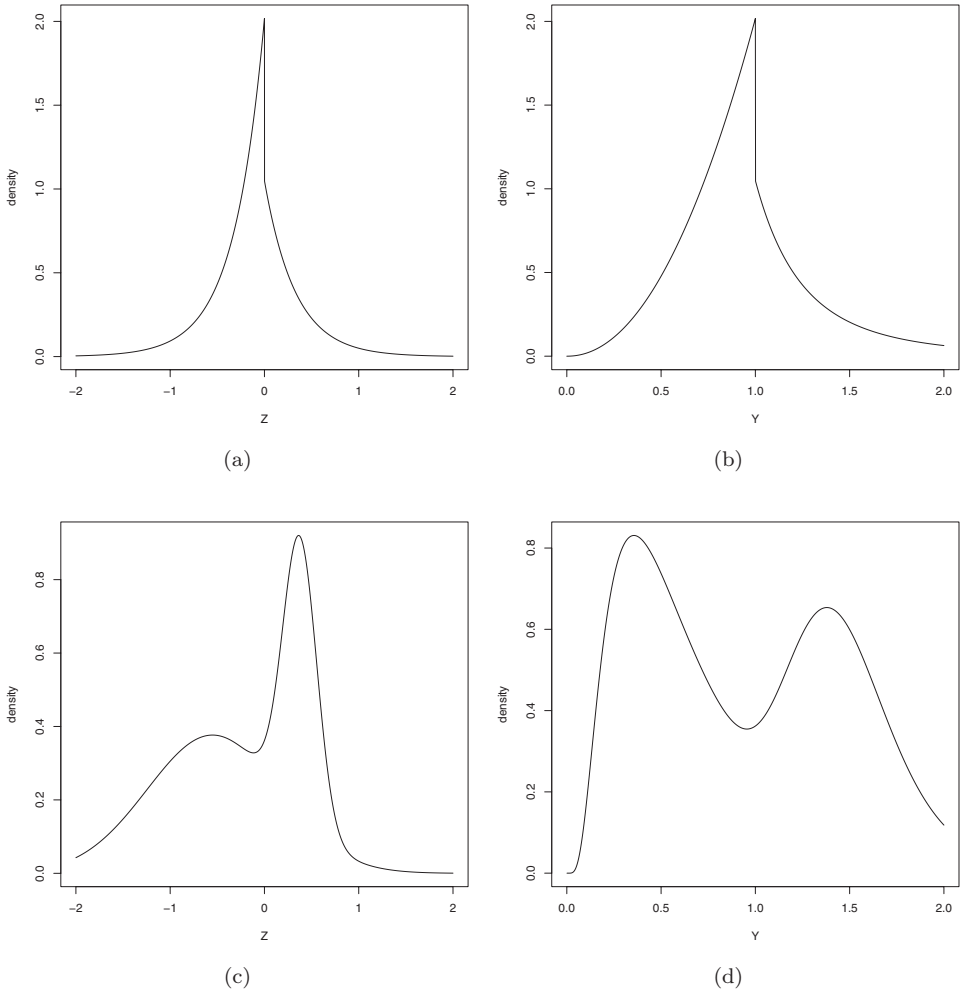


Fig. 1. A depiction of jump distributions. (a) Case A: Double exponential density. (b) Case A: The corresponding log double exponential density. (c) Case B: A normal mixture density and (d) Case B: The corresponding lognormal mixture density.

$\kappa = \mathbf{E}[Y] - 1$ in (2.3) is 0.007576. Furthermore, by having separate distributions for the up and down jumps we may easily calculate that a jump is positive with probability $p = 0.3445$ and that the expected values of the up and down jumps are 1.488639 and 0.7547517 respectively.

We want to remark that despite the clear interpretation of positive and negative jumps it is pretty clear from Fig. 1(b) and also from calculations that many realized jumps will be small. For example, the probability that the value of Y is between 0.9 and 1.1 is 0.268. Of course, these values may be changed using suitable parameters but due to the shape of the density making probability of values close to one small

automatically increases the probability of huge unrealistic jumps. For example, with the particular parameters above, the chance of a jump Y more than size 2 (a jump that doubles the value of the stock price) is 4.17%, that is, one in about 24 shocks will double the price. Further, the chance of a crippling jump of $Y < 0.4$ (stock price drops more than 60%) is 3.91%, again we believe unreasonably high.

The next distribution proposed aims to control these types of unexpected probabilities.

Case B. *Using a log mixture of normals.* In this case, the logarithm of the variable Y (controlling the jump size) is a mixture of two normal variables. Specifically, the density used is:

$$g(y) = p\frac{1}{y\sqrt{2\pi\sigma_1^2}}e^{-\frac{(\log y-\mu_1)^2}{2\sigma_1^2}} + (1-p)\frac{1}{y\sqrt{2\pi\sigma_2^2}}e^{-\frac{(\log y-\mu_2)^2}{2\sigma_2^2}}.$$

To model up and down jumps we chose $\mu_1 > 0$ and $\mu_2 < 0$. We may think about a jump modeled by such distributions as being with probability p a jump up drawn from the lognormal distribution with parameters μ_1 and σ_1 or, with probability $1-p$, a jump down drawn from the lognormal distribution with parameters μ_2 and σ_2.

In all the examples presented next we are using the parameter values: $\mu_1 = 0.3753$, $\sigma_1 = 0.18$, $\mu_2 = -0.5503$, $\sigma_2 = 0.6944$, and $p = 0.3445$. We plot the normal mixture density and the corresponding lognormal mixture density in Figs. 1(c) and 1(d). These parameter values are chosen so that the expected values and variances of the jumps up and down, and overall are approximatively matching the values of the log double exponential density in Case A. Specifically, the expectation and variance of jump sizes are 0.9907409 and 0.3849779 while the means and variances of the up and down jump sizes are matched closely. The parameter κ is $\kappa = -0.0092591$. It is the matching of these parameters that makes the shape of the density look crooked in Fig. 1(d). Despite this we observe that most jumps will be away from $Y = 1$, much more so than in the previous case.

4.2. *Example 1. A model with two regimes*

We first consider the case of two regimes ($m = 2$). The generator of the Markov chain α_t is $Q = \begin{pmatrix} -q_{12} & q_{12} \\ q_{21} & -q_{21} \end{pmatrix}$, where $q_{12} > 0$ is the switching rate from regime 1 to regime 2 and $q_{21} > 0$ is the switching rate from regime 2 to regime 1. To be specific, the pair of coupled PIDE's resulted from the general system (2.3) is:

$$\frac{1}{2}\sigma_1^2 S^2\frac{\partial^2 V_1}{\partial S^2} + (r_1 - \lambda_1\kappa)S\frac{\partial V_1}{\partial S} - (r_1 + \lambda_1 + q_{12})V_1 - \frac{\partial V_1}{\partial t}$$
$$= -\lambda_1\int_0^\infty V_1(Sy,t)g(y)dy - q_{12}V_2,$$

$$\frac{1}{2}\sigma_2^2 S^2 \frac{\partial^2 V_2}{\partial S^2} + (r_2 - \lambda_2\kappa) S \frac{\partial V_2}{\partial S} - (r_2 + \lambda_2 + q_{21}) V_2 - \frac{\partial V_2}{\partial t}$$
$$= -\lambda_2 \int_0^\infty V_2(Sy, t) g(y) dy - q_{21} V_1. \tag{4.1}$$

In the numerical computations, we use the following parameter values. The Markov chain transition rates are: $q_{12} = q_{21} = 0.5$, the volatilities and risk free interest rates corresponding to the two regimes are: $\sigma_1 = 0.15$, $\sigma_2 = 0.25$, $r_1 = r_2 = 0.05$. For the frequency of the jump component in the two regimes we use $\lambda_1 = 5$, $\lambda_2 = 2$ which correspond to expected five and two jumps/year respectively. For the jump distribution $g(y)$ we use Cases A and B presented in the previous section. The constant κ calculated earlier is used.

4.2.1. *Comment*

The parameter values for the regime-switching component (without the jumps) are used in [5] to exemplify a finite difference method, and in [22] to exemplify a binomial tree method to calculate the European call option prices for a number of different initial S_0 values. We refer to [22, Tables 1 and 2] for published results. These published numbers provide a natural way to check the regime-switching part of the algorithms by setting the λ_i parameters equal to zero. For the jump component in Case A, we use the parameter values in [9, 31], thus we may verify the implementation for the jump diffusion part as well by comparing with [31, Table 3]. These results are obtained by setting the q_{ij} values equal to zero and the volatilities to be the same for both regimes (so only one PIDE needs to be solved). In all cases we get very good agreement with the previously published values for these reduced models.

We use the algorithms presented in Sec. 3 to approximate the prices of European type options that have maturity $T = 1$ (year) and exercise price $K = 100$. The corresponding boundary conditions are:

- For call option:

$$V_1(S, 0) = V_2(S, 0) = \max(S - K, 0); \quad \forall\, S \in [0, \infty),$$
$$V_1(0, t) = V_2(0, t) = 0; \quad \forall\, t \in [0, T],$$
$$\frac{\partial V_1}{\partial S}(S_{\max}, t) = \frac{\partial V_2}{\partial S}(S_{\max}, t) = 1; \quad \forall\, t \in [0, T].$$

- For put option:

$$V_1(S, 0) = V_2(S, 0) = \max(K - S, 0); \quad \forall\, S \in [0, \infty),$$
$$\frac{\partial V_1}{\partial S}(0, t) = \frac{\partial V_2}{\partial S}(0, t) = -1; \quad \forall\, t \in [0, T],$$
$$\frac{\partial V_1}{\partial S}(S_{\max}, t) = \frac{\partial V_2}{\partial S}(S_{\max}, t) = 0; \quad \forall\, t \in [0, T].$$

Table 1. Call option values for Example 1 obtained using the three algorithms with different jump distributions.

Jump	Algorithm	V_1					V_2				
		92.00	96.00	100.00	104.00	108.00	92.00	96.00	100.00	104.00	108.00
Case A	1a	32.38	34.97	37.64	40.39	43.21	24.57	26.99	29.51	32.15	34.89
	1b	32.38	34.97	37.65	40.39	43.21	24.57	26.99	29.52	32.15	34.89
	2	32.37	34.97	37.64	40.39	43.21	24.57	26.98	29.51	32.15	34.88
Case B	1a	43.24	46.32	49.44	52.60	55.80	32.84	35.72	38.67	41.70	44.78
	1b	43.24	46.31	49.43	52.60	55.80	32.84	35.72	38.67	41.70	44.78
	2	43.24	46.31	49.43	52.60	55.80	32.84	35.72	38.67	41.69	44.78

Table 2. Put option values for Example 1 obtained using the three algorithms with different jump distributions.

Jump	Algorithm	V_1					V_2				
		92.00	96.00	100.00	104.00	108.00	92.00	96.00	100.00	104.00	108.00
Case A	1a	35.51	34.13	32.82	31.58	30.42	27.71	26.13	24.67	23.32	22.07
	1b	35.52	34.13	32.82	31.59	30.42	27.71	26.13	24.67	23.32	22.07
	2	35.51	34.13	32.82	31.58	30.42	27.70	26.13	24.67	23.32	22.06
Case B	1a	46.54	45.63	44.76	43.93	43.15	36.07	34.96	33.92	32.95	32.04
	1b	46.55	45.63	44.76	43.94	43.15	36.07	34.96	33.92	32.95	32.04
	2	46.55	45.63	44.76	43.93	43.15	36.07	34.96	33.92	32.95	32.04

4.2.2. *Results*

Tables 1 and 2 present the numerical results obtained with the three algorithms. The numbers represent the final option values at the initial time ($t = 0$) in a region in S around the exercise price K.

We can clearly see that once rounded to the nearest decimal the solutions agree for the S values presented. Algorithm 2 needs a smaller time step (20 times smaller, $\Delta t = 0.00005$) to produce values which are close to the accuracy produced by the Algorithms 1a and 1b, and this approximately compensates for the fact that 1a,1b need to solve the problem from $t = 0$ to $t = 1$ several times, so the run times for the three algorithms are comparable, as seen in Table 3.

4.2.3. *Interpretation of the solution*

It is clear that all algorithms agree on the solution, in each case. However, since we analyze two jump distributions and the regime-switching model produces two

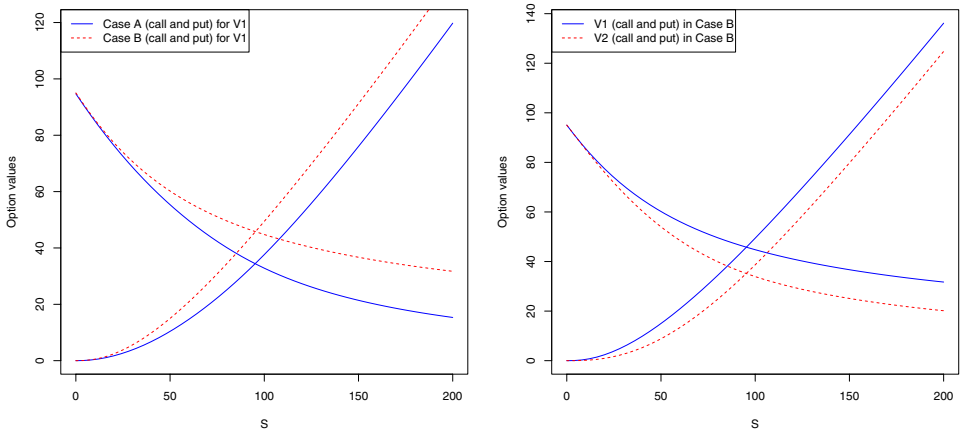
Table 3. Run-times (min:sec) to compare the algorithms convergence, with jump distribution as in Case A.

	Alg. 1a	Alg. 1b	Alg. 2
Call options	11:12	10:18	9:22
Put options	11:59	11:25	9:38

solutions automatically (one for each of the regimes), we may attempt to interpret the numbers obtained to perhaps indicate future research in the theoretical analysis. We note that more volatile stocks increase the price of options (both calls and puts). Jumps also produce variability in the stocks and they also increase the price of options.

We first note the clear difference between the solutions obtained for the two jump distributions (Fig. 2(a)). It is fairly clear that when a jump is modeled using the lognormal mixture (Case B) the resulting option price is considerably higher than the solution obtained in the log double exponential (Case A). This is also evidenced by looking at the difference in numbers between Cases A and B for V_1 (about \$11) and V_2 (about \$9). The only thing different is the jump distribution, but recall that V_1 has a higher jump frequency (average 5/year) versus V_2 (average 2/year). Thus, with the higher frequency, the jumps produced in Case B seem to be consistently larger than in the Case A. This is despite the fact that the log double exponential distribution in Case A can produce much larger jumps than in the Case B.

We would like to make a second observation, by comparing the jumps and the variability. The effect is not so evident but is worth noting. By design, regime 1 has a lower volatility (0.15) than regime 2 (0.25). Higher volatility typically leads to higher option values. However, as noted above, regime 1 has a higher incidence of jumps (both negative and positive). The values obtained are always higher for V_1 versus V_2 (Fig. 2(b)) and this would seem to indicate that having more frequent jumps has a more dramatic effect on the option prices than having an increased variability.



(a) (b)

Fig. 2. Differences between the regime solutions of the Example 1. (a) Difference between V_1 values for the two jump distributions and (b) difference between V_1 and V_2.

4.3. *Example 2. A model with four regimes*

In the second example, we consider a four-state ($m = 4$) Markov chain α_t with states

$$\mathcal{M} = \{1, 2, 3, 4\} \text{ and generator } Q = \begin{pmatrix} -1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & -1 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & -1 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & -1 \end{pmatrix}.$$

In this case, the market can be in any of the four regimes with equal probability.

The volatilities and interest rates are:

$$\sigma_1 = 0.9, \quad \sigma_2 = 0.5, \quad \sigma_3 = 0.7, \quad \sigma_4 = 0.2,$$
$$r_1 = 0.02, \quad r_2 = 0.1, \quad r_3 = 0.06, \quad r_4 = 0.15.$$

All options have maturity $T = 1$ (year) and exercise price $K = 100$.

The values for the jump rate are set to $\lambda_1 = 8$, $\lambda_2 = 2$, $\lambda_3 = 5$, $\lambda_4 = 1$. For the jump distribution, we use Cases A and B. The system of PIDE's is:

$$\frac{1}{2}\sigma_i^2 S^2 \frac{\partial^2 V_i}{\partial S^2} + (r_i - \lambda_i \kappa) S \frac{\partial V_i}{\partial S} - (r_i + \lambda_i - q_{ii}) V_i - \frac{\partial V_i}{\partial t}$$
$$= -\lambda_i \int_0^\infty V_i(Sy, t) g(y) dy - \sum_{j \neq i} q_{ij} V_j, \tag{4.2}$$

for $i = 1, 2, 3, 4$. The boundary conditions for the four value functions V_i, $i = 1, 2, 3, 4$ are the same as in Example 1.

4.3.1. *Results*

As in the case of the two-regime example the solutions obtained from the three algorithms are virtually indistinguishable. To save space Table 4 shows selected values obtained running the Algorithm 1b. To obtain similar precision, once again the Algorithm 2 needed a smaller time step (20 times smaller again), but only has

Table 4. Option values for Example 2 for each of the four regimes using either jump distribution.

Jump	Regime	Call values					Put values				
		92.00	96.00	100.00	104.00	108.00	92.00	96.00	100.00	104.00	108.00
Case A	V_1	45.56	48.54	51.57	54.64	57.75	48.89	47.89	46.94	46.03	45.16
	V_2	32.50	35.22	38.02	40.88	43.81	31.74	30.47	29.27	28.15	27.09
	V_3	39.78	42.64	45.54	48.50	51.51	41.04	39.91	38.84	37.81	36.84
	V_4	27.41	30.13	32.97	35.91	38.95	24.19	22.92	21.76	20.71	19.76
Case B	V_1	55.11	58.42	61.76	65.13	68.52	58.68	58.00	57.35	56.73	56.14
	V_2	39.60	42.61	45.68	48.81	51.98	38.94	37.96	37.04	36.17	35.34
	V_3	48.56	51.74	54.95	58.21	61.49	49.99	49.18	48.40	47.66	46.96
	V_4	34.05	37.08	40.18	43.37	46.62	30.91	29.94	29.05	28.24	27.50

Table 5. Run-times (min:sec) for Example 2, with jump distribution as in Case A.

	Alg. 1a	Alg. 1b	Alg. 2
Call options	49:48	43:11	25:57
Put options	56:56	46:05	25:48

to integrate the problem from $t = 0$ to $t = 1$ once, and it is somewhat faster, see Table 5.

4.3.2. *Interpretation of the solution*

We first note that as in the previous example the solutions for all regimes are universally higher when the jumps are modeled by the lognormal mixture distribution of Case B. This is clear from the values in Table 4 and from Fig. 3(a) (results are similar for all regimes). Second, in Figs. 3(b)–3(d) we see that the solutions for the four regimes are in the order $V_1 > V_3 > V_2 > V_4$ and in fact this makes perfect sense because this is the decreasing order of both volatility and jump intensity. Once again we may observe by comparing the numerical values as well as Figs. 3(c) and 3(d) that the differences between the regime solutions are higher when the jumps are modeled as in Case B.

4.4. *Example 3. A basket option written on two assets which follow regime-switching jump diffusions*

In this example, we illustrate how the algorithms may be used to price options written on multiple assets. We consider European type options written on two correlated assets in the regime-switching jump diffusion models. The asset prices follow the model:

$$\frac{dS_n(t)}{S_n(t-)} = \mu_n(\alpha_t)dt + \sigma_n(\alpha_t)dB_n(t) + dJ_n(t), \quad t \geq 0, \quad n = 1, 2, \tag{4.3}$$

where

$$J_n = \sum_{k=1}^{N_t} (Y_k^n - 1), \quad n = 1, 2, \tag{4.4}$$

and Y_k^n, $k = 1, 2, \ldots$, for each n is a sequence of iid random variables with density $g_n(y)$. The driving Brownian motions $B_1(t)$ and $B_2(t)$ are correlated with correlation coefficient ρ. Note that for this particular example we consider a special case for which the jumps occur simultaneously for both assets, but with independent jump size distributions. A more general model for which jumps for different assets are governed by different (correlated) stochastic processes) is interesting and deserves further care for future study.

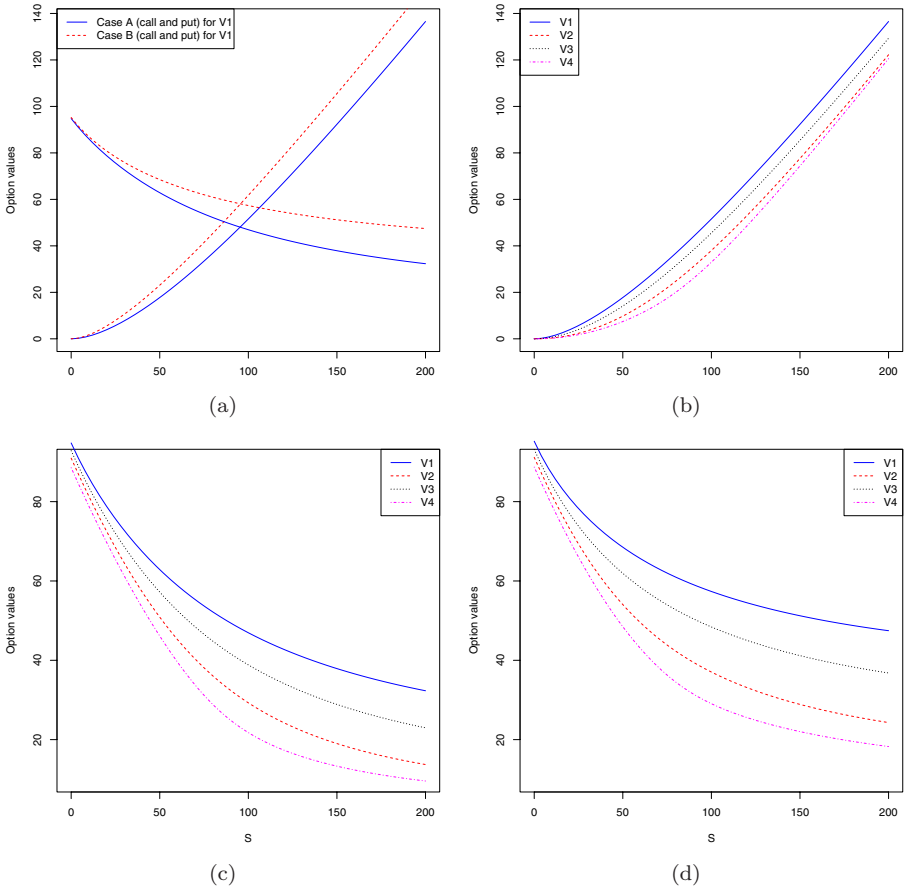


Fig. 3. Differences between the solutions for different regimes of Example 2. (a) Difference between V_1 for the two jump distributions, (b) values of call options for different regimes (Case A), (c) values of put options for different regimes (Case A), and (d) values of put options for different regimes (Case B).

Let $V_i(S_1, S_2, t)$ denote the value functions at t, where t is the time-to-maturity,[2] when the stock values are S_1 and S_2, and the regime $\alpha_t = i$. The V_i's, $i = 1, \ldots, m$ solve the PIDE system:

$$\frac{1}{2}\sigma_1^2(i)S_1^2\frac{\partial^2 V_i}{\partial S_1^2} + \frac{1}{2}\sigma_2^2(i)S_2^2\frac{\partial^2 V_i}{\partial S_2^2} + \rho\sigma_1(i)\sigma_2(i)S_1S_2\frac{\partial^2 V_i}{\partial S_1 \partial S_2} + (r_i - \lambda_i\kappa_1)S_1\frac{\partial V_i}{\partial S_1}$$
$$+ (r_i - \lambda_i\kappa_2)S_2\frac{\partial V_i}{\partial S_2} - r_iV_i - \frac{\partial V_i}{\partial t} + \lambda_i E[V_i(S_1Y_1, S_2Y_2, t) - V_i(S_1, S_2, t)]$$
$$+ \sum_{j\neq i} q_{ij}[V_j - V_i] = 0, \tag{4.5}$$

where $\kappa_n = E[Y_n - 1] = \int (y-1)g_n(y)dy$, $n = 1, 2$.

[2]So that we once again work with forward equations.

In this particular numerical example, we consider the call options on the sum S_1+S_2 (an example of the so-called basket options). The payoff is $\max(S_1+S_2-K,0)$ where K is the exercise price. We consider the two-regime case ($m = 2$). The boundary conditions are,

$$V_1(S_1,S_2,0) = V_2(S_1,S_2,0) = \max(S_1+S_2-K,0); \quad \forall\, S_1,S_2 \in [0,\infty),$$

$$V_1(S_1,0,t) = V_1(S_1,t), \quad V_2(S_1,0,t) = V_2(S_1,t), \quad \forall\, t \in [0,T], \quad S_1 \in [0,\infty),$$

$$V_1(0,S_2,t) = V_1(S_2,t), V_2(0,S_2,t) = V_2(S_2,t), \quad \forall\, t \in [0,T], \quad S_2 \in [0,\infty),$$

where $V_i(S_1,t)$, $i = 1,2$ are the solutions of the call option problem for asset S_1 only, and $V_i(S_2,t)$, $i = 1,2$ are the solutions of the call option problem for asset S_2 only,

$$\frac{\partial V_1(S_{1,\max},S_2,t)}{\partial S_1} = \frac{\partial V_2(S_{1,\max},S_2,t)}{\partial S_1} = 1; \quad \forall\, t \in [0,T], S_2 \in [0,\infty),$$

$$\frac{\partial V_1(S_1,S_{2,\max},t)}{\partial S_2} = \frac{\partial V_2(S_1,S_{2,\max},t)}{\partial S_2} = 1; \quad \forall\, t \in [0,T], S_1 \in [0,\infty).$$

Please note that if either S_1 or S_2 become zero then it stays there and the basket option becomes a regular option on a single asset. In the algorithmic implementation, at $S_1 = 0$ and $S_2 = 0$, we used the PDE2D boundary condition "none", which simply enforces the PDE rather than any boundary condition. Since at $S_2 = 0$, the PIDE (4.5) reduces to the call option for one variable S_1, we are indirectly enforcing the specified boundary condition; and similarly at $S_1 = 0$. However, using the boundary condition $\frac{\partial V_1}{\partial S_2} = \frac{\partial V_2}{\partial S_1} = 0$ at $S_2 = 0$ and similarly at $S_1 = 0$, gave similar result to "none".

We use the following parameter values in the numerical computation: the Markov chain transition rates $q_{12} = q_{21} = 0.5$, the interest rates $r_1 = r_2 = 0.05$, the volatilities for asset S_1: $\sigma_1(1) = 0.15$, $\sigma_1(2) = 0.25$, and for asset S_2: $\sigma_2(1) = 0.20$, $\sigma_2(2) = 0.35$, the correlation coefficient is $\rho = 0.5$. For the jump component, $\lambda_1 = 5$, $\lambda_2 = 2$. The resulting constants κ_n, $n = 1,2$ can be computed by $\kappa_n = \mathbf{E}[Y_n] - 1 = \int_0^\infty y g_n(y) dy - 1$, where the jump distribution $g_n(y)$ can be either Case A or Case B as discussed above. The option has maturity $T = 1$ (year) and exercise price $K = 100$.

4.4.1. *Results*

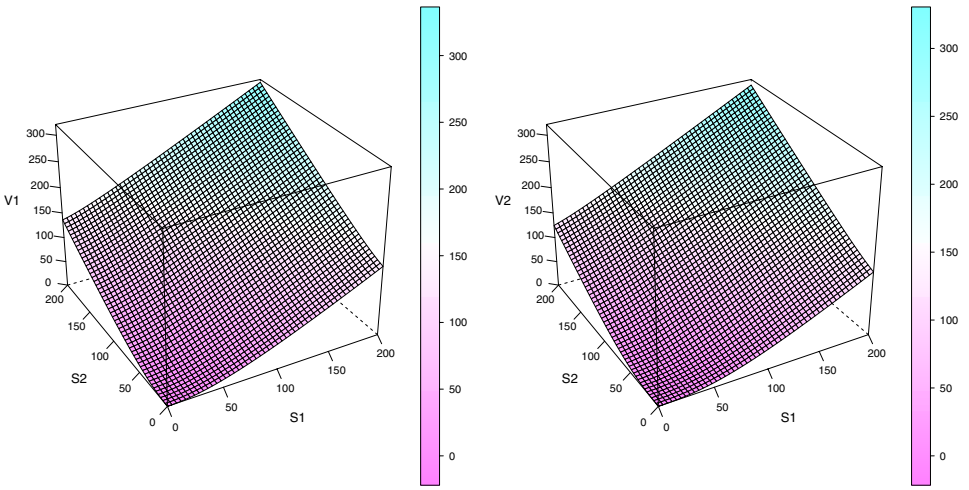
Since this example has not been attempted in literature (to our best knowledge) we present for future reference selected values in Table 6. We also display the option values for the two regimes in the jump distribution Case B in Fig. 4.

4.4.2. *Interpretation of results*

Once again the distribution choice makes a difference. It is remarkable that the boundary condition has so little effect on the solution of this example. Since the

Table 6. Example 3 option values for jump distributions in Cases A and B (Algorithm 1b).

S1	S2	Case A		Case B	
		V1	V2	V1	V2
44	44	23.54	17.91	31.50	22.99
44	48	25.87	20.11	34.28	25.58
44	52	28.32	22.47	37.16	28.30
44	56	30.87	24.96	40.12	31.13
48	44	25.85	20.04	34.27	25.52
48	48	28.25	22.36	37.10	28.20
48	52	30.78	24.83	40.02	30.99
48	56	33.40	27.42	43.02	33.89
52	44	28.26	22.32	37.13	28.20
52	48	30.75	24.75	40.01	30.94
52	52	33.34	27.32	42.97	33.81
52	56	36.04	30.02	46.00	36.77
56	44	30.79	24.74	40.08	30.99
56	48	33.35	27.28	42.99	33.81
56	52	36.01	29.95	45.99	36.73
56	56	38.78	32.74	49.06	39.76



(a) V_1 values (b) V_2 values

Fig. 4. Solution for Case B (at $t = 0$) of Example 3.

Algorithms 1a and 1b agreed better with the Example 1 results along the $S_2 = 0$ boundary, this indicates that they may be a bit more accurate than the Algorithm 2 results. Algorithm 2 is somewhat faster, see Table 7, even though we again used a time step 20 times smaller ($\Delta t = 0.0005$) than that used by Algorithms 1a and 1b.

Table 7. Run-times (min:sec) for Example 3, with jump distribution as in Case A.

Alg. 1a	Alg. 1b	Alg. 2
339:31	288:15	219:53

5. Conclusions

In this paper, we present two Algorithms (1a and 1b in the paper) designed to numerically solve a system of PIDES. These algorithms are compared with a more traditional finite difference scheme (Algorithm 2 in the paper) which handles the integral term explicitly.

We generally find that Algorithm 2 seems to need a smaller time step than the other two algorithms to provide comparable accuracy.

With regard to the relative speeds of Algorithms 1a and 1b, both were implemented as coupled systems of PDE's, thus in all cases 1b converges in fewer iterations and since they both take the same time per iteration, 1b is always faster. However, as we have noted earlier, when Algorithm 1a is implemented to solve the decoupled PDE's separately, it takes considerably less time, and can be further speeded up by solving the equations simultaneously on different processors.

Another striking observation was that using as jump distribution the lognormal mixture distribution with parameters chosen to match moments of a more popular log-double exponential distribution, seemed to produce much more easily distinguishable jumps from random fluctuation, and this was reflected in the price of the options. Though we only introduced the lognormal mixture to provide comparison with the more popular density, it may be easily seen that this distribution has its own merits and we believe deserve a special study on its own. Today, there are two lines of research into models capable of producing the type of equity behavior observed in reality. Stochastic volatility models and Lévy models (the later are essentially continuous processes with a jump component). Although we did not treat the stochastic volatility using a continuous diffusion in the current paper, the numerical evidence we provide seems to indicate that jumps have a more significant effect than an increase in volatility on the behavior of the equity and the resulting option prices. Thus, we believe that the introduction of this new distribution is an important contribution to the literature besides the algorithms which are the main focus of this work.

References

[1] A. Almendral and C. W. Oosterlee, Numerical valuation of options with jumps in the underlying, *Appl. Numer. Math.* **53** (2005) 1–18.

[2] L. Andersen and J. Andreasen, Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing, *Review of Derivatives Research* **4** (2000) 231–262.

[3] E. Bayraktar and H. Xing, Pricing American options for jump diffusions by iterating optimal stopping problems for diffusions, *Math. Method. Oper. Res.* **70** (2009) 505–525.
[4] M. Boyarchenko and S. Boyarchenko, Double barrier options in regime-switching hyper-exponential jump-diffusion models, *Int. J. Theor. Appl. Finance* **14** (2011) 1005–1043.
[5] P. Boyle and T. Draviam, Pricing exotic options under regime switching, *Insur. Math. Econ.* **40** (2007) 267–282.
[6] M. Briani, R. Natalini and G. Russo, Implicit-explicit numerical schemes for jump-diffusion processes, *Calcolo* **44** (2007) 33–57.
[7] P. Carr and A. Mayo, On the numerical evaluation of options prices in jump diffusion processes, *Eur. J. Financ.* **13**(4) (2007) 353–372.
[8] R. Cont and E. Voltchkova, A finite difference scheme for option pricing in jump diffusion and exponential Lévy models, *SIAM J. Numer. Anal.* **43**(4) (2005) 1596–1626.
[9] Y. d'Halluin, P. A. Forsyth and G. Labahn, A penalty method for American options with jump diffusion processes, *Numer. Math.* **97** (2004) 321–352.
[10] Y. d'Halluin, P. Forsyth and K. R. Vetzal, Robust numerical methods for contingent claims under jump diffusion processes, *IMA J. Numer. Anal.* **25**(1) (2005) 87–112.
[11] R. Elliott, T. Siu, L. Chan and J. Lau, Pricing options under a generalized markov modulated jump diffusion model, *Stochastic Anal. Appl.* **25**(4) (2007) 821–843.
[12] I. Florescu, R. Liu and M. C. Mariani, Solutions to a partial integro-differential parabolic system arising in the pricing of financial options in regime-switching jump diffusion models, *Electron. J. Differ. Equ.* **2012**(231) (2012) 1–12.
[13] I. Florescu and M. Mariani, Solutions to an integro-differential parabolic problem arising on financial mathematics, *Electron. J. Differ. Equ.* **2010**(62) (2010) 1–10.
[14] I. Florescu, M. Mariani and G. Sewell, Numerical solutions to an integro-differential parabolic problem arising in the pricing of financial options in a Levy market, *Quant. Finance* (2011) 1–8.
[15] X. Guo, Information and option pricings, *Quant. Finance* **1** (2000) 38–44.
[16] Y. Huang, P. Forsyth and G. Labahn, Methods for pricing American options under regime switching, *SIAM J. Sci. Comput.* **33**(5) (2011) 2144–2168.
[17] A. Khaliq, B. Kleefeld and R. H. Liu, Solving complex pde systems for pricing American options with regime-switching by efficient exponential time differencing schemes, *Numer. Methods Partial Differential Equations* **29**(1) (2013) 320–336.
[18] A. Khaliq and R. H. Liu, New numerical scheme for pricing American option with regime-switching, *Int. J. Theor. Appl. Finance* **12**(3) (2009) 319–340.
[19] S. Kou, A jump-diffusion model for option pricing, *Manage. Sci.* **48** (2002) 1086–1101.
[20] Y. Kwon and Y. Lee, A second-order finite difference method for option pricing under jump-diffusion models, *SIAM J. Numer. Anal.* **49**(6) (2011a) 2598–2617.
[21] Y. Kwon and Y. Lee, A second-order tridiagonal method for American options under jump-diffusion models, *SIAM J. Sci. Comput.* **33**(4) (2011b) 1860–1872.
[22] R. Liu, Regime-switching recombining tree for option pricing, *Int. J. Theor. Appl. Financ.* **13** (2010) 479–499.
[23] R. Liu, Q. Zhang and G. Yin, Option pricing in a regime switching model using the Fast Fourier Transform, *Journal of Applied Mathametics and Stochastic Analysis* **2006** (2006) 1–22.
[24] R. C. Merton, Option pricing when underlying stock returns are discontinuous, *Journal of Financial Economics* **3**(12) (1976) 125–144.

[25] D. Nualart and W. Schoutens, Backward stochastic differential equations and feynman-kac formula for Lévy processes with applications in finance, *Bernoulli* **7**(5) (2001) 761–776.
[26] A. Ramponi, Fourier transform methods for regime-switching jump-diffusions and the pricing of forward starting options, *Working Paper*, Available online at arXiv:1105.4567v1.
[27] S. Salmi and J. Toivanen, An iterative method for pricing American options under jump-diffusion models, *Appl. Numer. Math.* **61** (2011) 821–831.
[28] S. Salmi and J. Toivanen, Comparison and survey of finite difference methods for pricing American options under finite activity jump-diffusion models, *Int. J. Comput. Math.* **89**(9) (2012) 1112–1134.
[29] G. Sewell, *The Numerical Solution of Ordinary and Partial Differential Equations*, 2nd edn. (John Wiley & Sons, NJ, 2005).
[30] T. K. Siu, J. W. Lau and H. Yang, Pricing participating products under a generalized jump-diffusion model, *Journal of Applied Mathematics and Stochastic Analysis* **2008** (2008) 1–30.
[31] J. Toivanen, Numerical valuation of European and American options under Kou's Jump-Diffusion Model, *SIAM J. Sci. Comput.* **30**(4) (2008) 1949–1970.
[32] J. Toivanen, A high-order front-tracking finite difference method for pricing American options under jump-diffusion models, *J. Comput. Finance* **13**(3) (2010) 61–79.
[33] J. Topper, *Financial Engineering with Finite Elements* (John Wiley & Sons, NJ, 2005).
[34] F. Yuen and H. Yang, Option pricing in a jump-diffusion model with regime-switching, *ASTIN Bulletin* **39**(2) (2009) 515–539.