Numerical Methods and Volatility Models for Valuing Cliquet Options

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

Revised: February 14, 2006
First Version: September 13, 2004

Abstract

Several numerical issues for valuing cliquet options using PDE methods are investigated. The use of a running sum of returns formulation is compared to an average return formulation. Methods for grid construction, interpolation of jump conditions, and application of boundary conditions are compared. The effect of various volatility modelling assumptions on the value of cliquet options is also studied. Numerical results are reported for jump diffusion models, calibrated volatility surface models, and uncertain volatility models.

Keywords: Cliquet options, jump diffusion, interpolation, boundary conditions, volatility models

AMS Classification: 65M12, 65M60, 91B28

Acknowledgment: This work was supported by the Natural Sciences and Engineering Research Council of Canada, RBC Financial Group, and a subcontract with Cornell University, Theory & Simulation Science & Engineering Center, under contract 39221 from TG Information Network Co. Ltd.

1 Introduction

Cliquet options are financial derivative contracts which provide a guaranteed minimum annual return in exchange for capping the maximum return earned each year over the life of the contract. Recent turmoil in financial markets has led to a demand for products that reduce downside risk while still offering upside potential.[1] For example, pension plans have been looking at attaching guarantees to their products that are linked to equity returns. Some plans, such as those described

*Equity Trading Lab, Morgan Stanley, 1585 Broadway Ave, 9th Floor, New York, NY 10036, (e-mail: Heath.Windcliff@morganstanley.com).

†School of Computer Science, University of Waterloo, 200 University Ave West, Waterloo ON, Canada N2L 3G1 (e-mail: paforsyt@elora.uwaterloo.ca).

‡Centre for Advanced Studies in Finance, University of Waterloo, 200 University Ave West, Waterloo ON, Canada N2L 3G1 (e-mail: kvetzal@watarts.uwaterloo.ca).

[1]See "Selling Pessimism", *The Economist*, March 8, 2003, for related discussion.

by Walliser (2003), limit the upside returns in order to reduce the costs associated with providing the guarantee. These products are essentially cliquet contracts.

Wilmott (2002) illustrated that cliquet options are sensitive to the model assumed for the underlying asset dynamics. In this paper we will explore a variety of modelling alternatives, including:

- a (finite activity) jump diffusion model (Merton, 1976);
- a state-dependent volatility surface (i.e. a model in which a local volatility function is calibrated to observed market prices of traded options, as described, for example, in Coleman et al. (1999)); and
- a nonlinear uncertain volatility model (Avellaneda et al., 1995; Lyons, 1995).[2]

We find that even if a time-dependent and state-dependent local volatility model is calibrated to prices of plain vanilla options generated from a jump diffusion model with constant parameters, there is no guarantee that the values of cliquet options calculated using the volatility surface will be close to cliquet values obtained using the jump diffusion model. This result is consistent with other studies, see e.g. Hirsa et al. (2003); Schoutens et al. (2004), which have demonstrated that a model calibrated to vanilla options does not necessarily price exotics correctly.

However, some practitioners are aware that simply using the local volatility surface calibrated to vanillas does not correctly capture the dynamics of the skew. Practitioners often attempt to make up for the known deficiencies of a calibrated local volatility surface by forcing the surface to be homogeneous of degree zero in price and strike. In addition, the surface is often *rolled forward* in time. It appears that previous studies have not taken into account these typical corrections.

Assuming that the true market process is a jump diffusion, we show that these two common corrections can result in much less error for the price of a cliquet option (at the initial value of the underlying asset). This provides some justification for common industry practice. However, the error is small only when the underlying is at the initial price. Consequently, the deltas computed using the local volatility surface are considerably in error.

Cliquets are discretely observed path-dependent contracts. As such, they can be conveniently valued by solving a set of one dimensional PDEs embedded in a higher dimensional space. These one dimensional PDEs exchange information through no-arbitrage jump conditions on observation dates. Wilmott (1998) has recommended this approach as a general framework and it has successfully been used to implement models for Parisian options (Vetzal and Forsyth, 1999), Asian options (Zvan et al., 1999), shout options (Windcliff et al., 2001), volatility swaps and options (Windcliff, Forsyth, and Vetzal, Windcliff et al.), and many others.

An important focus of this paper is to develop effective numerical methods for valuing cliquet options for all of the above models. Assuming we have effective methods for solving each one dimensional problem, there are still difficulties in solving the full cliquet problem. In particular, we will:

- show how the use of scaled grids for each one dimensional problem dramatically improves the convergence;

[2]Note that this was suggested by Wilmott (2002), who observed that the volatility risk for cliquet options is typically underestimated.

- investigate the effects of interpolation methods used to enforce the jump conditions arising from the state variable updating rules;
- show how to specify the boundary conditions at large and small values of the asset price; and
- look at the effects of using a finite computational domain on the data needed to enforce the jump conditions.

We will also compare the performances of a formulation that utilizes a running sum of returns with one that uses the running average of returns.

We emphasize that although the numerical techniques illustrated in this paper will be studied in conjunction with cliquet options, many of these methods also apply to other path-dependent options. For example, the interpolation and grid construction techniques described in this paper can also be used to dramatically improve the performance of algorithms for valuing discretely observed floating strike lookback options.

2 Formulation for Jump Diffusion and Uncertain Volatility Models

Let S represent the price of the underlying asset. The potential paths followed by S can be modelled by a stochastic differential equation given by

$$\frac{dS}{S} = (\xi - \lambda\kappa)dt + \sigma dz + (\eta - 1)dq, \tag{2.1}$$

where ξ is the drift rate, dq is an independent Poisson process with mean arrival time λ (i.e. $dq = 1$ with probability $\lambda\, dt$ and $dq = 0$ with probability $1 - \lambda\, dt$), $(\eta - 1)$ is an impulse function producing a jump from S to $S\eta$, κ is the expected value of $(\eta - 1)$, σ is the volatility (of the continuous part of the process), and dz is the increment of a standard Gauss-Wiener process.

Let $V(S, t)$ be the value of a European-style contract that depends on the underlying asset value S and time t. Following standard arguments (see, e.g. Merton, 1976; Wilmott, 1998; Andersen and Andreasen, 2000), the following backward partial integro differential equation (PIDE) for the value of $V(S, \tau)$ is obtained

$$V_\tau = \frac{\sigma(\Gamma, S, t)^2 S^2}{2} V_{SS} + (r - \lambda\kappa) S V_S - rV + \left(\lambda \int_0^\infty V(S\eta) g(\eta) d\eta - \lambda V \right), \tag{2.2}$$

where $\tau = T - t$, T is the maturity date of the contract, $\Gamma = V_{SS}$, r is the risk free rate of interest, and $g(\eta)$ is the probability density function of the jump amplitude η. In this paper, we will follow Merton (1976) and assume that η is lognormally distributed with mean μ and standard deviation γ, so that $\kappa = \exp(\mu + \gamma^2/2) - 1$. Specifically

$$g(\eta) = \frac{e^{\left(-\frac{(\log(\eta) - \mu)^2}{2\gamma^2} \right)}}{\sqrt{2\pi}\gamma\eta}\ . \tag{2.3}$$

In equation (2.2) we have allowed the volatility to be a function of $\Gamma = V_{SS}$, as well as the underlying asset price, S, and time, t. In an uncertain volatility model (Avellaneda et al., 1995; Lyons, 1995), it is assumed that

$$\sigma_{\min} \leq \sigma \leq \sigma_{\max} \tag{2.4}$$

but is otherwise uncertain. The worst case value for an investor with a long position in the option is determined from the solution to equation (2.2) with $\sigma(\Gamma)$ given by

$$\sigma(\Gamma)^2 = \begin{cases} \sigma_{\max}^2 & \text{if } \Gamma < 0 \\ \sigma_{\min}^2 & \text{if } \Gamma > 0 \end{cases}. \tag{2.5}$$

Conversely, the best case value for an investor with a long position is determined from the solution to equation (2.2) with $\sigma(\Gamma)$ given by

$$\sigma(\Gamma)^2 = \begin{cases} \sigma_{\max}^2 & \text{if } \Gamma > 0 \\ \sigma_{\min}^2 & \text{if } \Gamma < 0 \end{cases}. \tag{2.6}$$

The worst case value for an investor with a short position (a negative payoff) in the option is given by the negative of the solution to equations (2.2) and (2.6). Consequently, as discussed in Forsyth and Vetzal (2001), the worst-best case long values can be thought of as corresponding to the bid-ask prices for the option if buyers and sellers value it assuming worst case scenarios from their own perspectives.

3 Local Volatility Models

In this section, we postulate a synthetic market where the price process is a jump diffusion with constant parameters, as in equation (2.1). We then take the point of view of a practitioner who attempts to fit observed vanilla option prices using a local volatility model. We first develop some analytic results which provide some insight into the form of the fitted local volatility model. Then, we generate two local volatility surfaces which will be used in the subsequent numerical tests.

3.1 Overview

It is common practice to fit a local volatility model to vanilla option prices, and then use this local volatility surface to price exotic options. In this section, we will determine some properties of a local volatility model (no jumps) which has been calibrated to prices in a synthetic market where the price process is a constant volatility jump diffusion model.

Suppose that the asset in the synthetic market follows

$$\frac{dS}{S} = (\xi - \lambda\kappa)dt + \sigma_J dz + (\eta - 1)dq, \tag{3.1}$$

where we assume that for simplicity that $\sigma_J = const.$, and that the jump size distribution $g(\eta)$ is independent of (S, t), and given by equation (2.3).

If we assume the process (3.1), then the value of an option V given by (2.2) can be written in the form

$$V_\tau = \frac{\sigma_J^2 S^2}{2} V_{SS} + rSV_S - rV + \lambda \int_0^\infty \Big([V(S\eta) - V(S)] - (\eta - 1) S V_S \Big) g(\eta)\, d\eta. \tag{3.2}$$

Now, suppose the real world process follows the jump diffusion (3.1), but a practitioner assumes that stock prices evolve according to

$$\frac{dS}{S} = \nu dt + \sigma_L(S,t) dz, \tag{3.3}$$

where ν is the drift term and σ_L is the local volatility. Let W be the price of an option obtained assuming process (3.3). W satisfies

$$W_\tau = \frac{\sigma_L(S,t)^2 S^2}{2} W_{SS} + rSW_S - rW. \tag{3.4}$$

Typically, the local volatility is determined by calibration to a set of vanilla option prices for fixed (S,t), with varying strikes and maturities (K,T). Let $V(S,t;K,T)$ be the price of a vanilla call valued under process (3.1). The price can be regarded as a function of (S,t) with (K,T) fixed or as a function of (K,T) with (S,t) fixed. Using the latter perspective, we will let $S = S^*, t = t^*$ to emphasize that we regard (S^*, t^*) as fixed. Andersen and Andreasen (2000) show that the forward PIDE for a European call option is

$$\begin{aligned} V_T = {} & \frac{\sigma_J^2 K^2}{2} V_{KK} - rKV_K \\ & + \lambda \int_0^\infty \Big[\eta V(S^*, t^*; K/\eta, T) - \eta V(S^*, t^*; K, T) + (\eta - 1) K V_K \Big] g(\eta) d\eta. \end{aligned} \tag{3.5}$$

The boundary conditions for equation (3.5) are

$$\begin{aligned} V(S^*, t^*; 0, T) &= S^* \\ V(S^*, t^*; K \to \infty, T) &= 0 \\ V(S^*, t^*; K, t^*) &= \max(S^* - K, 0). \end{aligned} \tag{3.6}$$

Let $W(S^*, t^*; K, T)$ be the price of a European call obtained assuming process (3.3). As shown in Dupire (1994), the forward PDE for W satisfies

$$W_T = \frac{\sigma_L(K,T)^2 K^2}{2} W_{KK} - rKW_K. \tag{3.7}$$

We will first consider the solution of the forward equation (3.7) on the finite domain $\Omega_f = [K_{\min}, K_{\max}]$. We will assume as well that $S^* \in \Omega_f$. We will also consider expiry times T bounded away from t^*, i.e. $T \in [T_{\min}, T_{\max}], T_{\min} > t^*$. After carrying out an analysis in the finite domain $\Omega_f \times [T_{\min}, T_{\max}]$, we will take limits as $T_{\min} \to t^*$, $K_{\min} \to 0$, and $K_{\max} \to \infty$.

The calibration problem can then be stated as follows. Given $V(S^*, t^*; K, T)$ satisfying initial conditions (3.6) in $0 \le K \le \infty$, $t^* \le T \le T_{\max}$, determine $\sigma_L(S^*, t^*; K, T)$ such that $W = V$ in

$\Omega_f \times [T_{\min}, T_{\max}]$. In other words, determine the local volatility function such that the market prices for vanilla calls V are matched by the calibrated prices W at a specific (S^*, t^*), for all strikes and maturities in $\Omega_f \times [T_{\min}, T_{\max}]$. Note that we have emphasized that the solution to the calibration problem $\sigma_L = \sigma_L(S^*, t^*; K, T)$ is in general valid only for a specific (S^*, t^*).

For future reference, at this point we gather together some conditions on the solutions for V, W:

Conditions 3.1 (Conditions on V) *We assume the following conditions for V, the solution to equation (3.5) in the domain $0 \le K \le \infty, t^* \le T \le T_{\max}$:*

- *Initial conditions ($T = t$) and boundary conditions are given by equation (3.6).*
- *$\sigma_J^2 > 0$ in equation (3.5).*
- *V has bounded and continuous derivatives of up to first order in T and second order in K in $\Omega_f \times [T_{\min}, T_{\max}]$ (i.e. V is $C^{1,2}$).*
- *$V_{KK} > 0$ in $\Omega_f \times [T_{\min}, T_{\max}]$.*

Conditions 3.2 (Conditions on W) *We assume the following conditions for W, the solution to equation (3.7) in the domain $\Omega_f \times [T_{\min}, T_{\max}]$:*

- *Given a solution V to equation (3.5), initial conditions and boundary conditions for W are*

$$\begin{aligned} W(S^*, t^*; K, T_{\min}) &= V(S^*, t^*; K, T_{\min}) \\ W(S^*, t^*; K_{\max}, T) &= V(S^*, t^*; K_{\max}, T) \\ W(S^*, t^*; K_{\min}, T) &= V(S^*, t^*; K_{\min}, T). \end{aligned} \tag{3.8}$$

- *$\sigma_L^2 > 0$ in $\Omega_f \times [T_{\min}, T_{\max}]$.*
- *W has bounded and continuous derivatives of up to first order in T and second order in K (i.e. W is $C^{1,2}$).*

Remark 3.1 *Clearly, as $T \to t^*$, then the initial condition (3.6) implies that $V_{KK} = 0$ for $K \neq S$. However, for any $T_{\min} > t^*$, $V_{KK} > 0$ in $\Omega_f \times [T_{\min}, T_{\max}]$.*

Let $E = W - V$ and subtract equation (3.5) from equation (3.7) to obtain

$$\begin{aligned} \mathcal{L}E &= f(S^*, t^*; K, T) \\ f(S^*, t^*; K, T) &= -\left(\frac{\sigma_J^2 - \sigma_L^2}{2}\right) K^2 V_{KK} \\ &\quad - \lambda \int_0^\infty \Big(\eta V(S^*, t^*; K/\eta, T) - \eta V(S^*, t^*; K, T) + (\eta - 1) K V_K\Big) g(\eta) d\eta, \end{aligned} \tag{3.9}$$

where

$$\mathcal{L}E \equiv E_T - \left[\frac{\sigma_L^2 K^2}{2} E_{KK} - rKE_K\right]. \tag{3.10}$$

From equation (3.8) we have

$$\begin{aligned} E(K,T) &= 0, \quad K \in \partial\Omega_f, \quad T \in [T_{\min}, T_{\max}], \\ E(K,T_{\min}) &= 0, \quad K \in \Omega_f. \end{aligned} \tag{3.11}$$

Consequently, the calibration problem may be restated as: find $\sigma_L(S^*,t^*;K,T)$ such that $E=0$ in $\Omega_f \times [T_{\min}, T_{\max}]$.

If $\sigma_L^2 > 0$ in equation (3.10), then the Green's function (Roach, 1982; Garroni and Menaldi, 1992) of $\mathcal{L}$ is the solution to

$$\begin{aligned} \mathcal{L}G &= \delta(K-K')\delta(T-T') \\ G(K,T) &= 0, \quad K \in \partial\Omega_f \\ G(K,T_{\min}) &= 0, \quad K \in \Omega_f. \end{aligned} \tag{3.12}$$

The solution to equation (3.9) can then be written as

$$E(K,T) = \int_{\Omega_f} \int_{T_{\min}}^{T} G(K,T,K',T') f(S^*,t^*;K',T')\, dT'\, dK'. \tag{3.13}$$

Detailed conditions for the existence of a Green's function for non-self adjoint operators of the type (3.10) are given in Garroni and Menaldi (1992). Briefly, if the PDE is non-degenerate, and has bounded coefficients, then existence of the Green's function can be proven. Note that since we have restricted the domain to Ω_f, these conditions are satisfied. (As a point of interest, reference (Garroni and Menaldi, 1992) also discusses the existence of Green's functions for PIDEs of the type (2.2). It appears to us that the work in Garroni and Menaldi (1992) deserves to be better known.)

Lemma 3.1 (Condition on $f(S^*,t^*;K,T)$) *If $f(S^*,t^*;K,T)$ is a continuous function and $\sigma_L^2 > 0$, then $E(K,T) \equiv 0$ in $\Omega_f \times [T_{\min}, T_{\max}]$ if and only if $f(S^*,t^*;K,T) \equiv 0$ in $\Omega_f \times [T_{\min}, T_{\max}]$.*

Proof. If $f(S^*,t^*;K,T) = 0$ in $\Omega_f \times [T_{\min}, T_{\max}]$, then from equation (3.13) we have immediately that $E = 0$ in $\Omega_f \times [T_{\min}, T_{\max}]$. Conversely if $E = 0$, then from equation (3.9) we have that $f(S^*,t^*;K,T) = 0$ in $\Omega_f \times [T_{\min}, T_{\max}]$. □

Proposition 3.1 (Existence of σ_L) *Given a solution V to equation (3.5) for $0 \le K \le \infty$, $t^* \le T \le T_{\max}$ such that Condition (3.1) holds, then there exists a unique, positive, and bounded local volatility function σ_L such that the solution W of equation (3.7) yields the same prices ($V = W$) for given (S^*,t^*) in the domain $\Omega_f \times [T_{\min}, T_{\max}]$.*

Proof. From Condition (3.1) we have that V is $C^{1,2}$ and hence $f(S^*,t^*;K,T)$ is a continuous function. Since $V_{KK} > 0$, a Taylor series argument shows that $\eta V(S^*,t^*;K/\eta,T) - \eta V(S^*,t^*;K,\eta) + (\eta-1)KV_K \ge 0$ ($\eta \ge 0$), and we have that σ_L given by

$$\begin{aligned} \sigma_L(S^*,t^*;K,T)^2 &= \sigma_J^2 \\ &+ \frac{\lambda \int_0^\infty \Big(\eta V(S^*,t^*;K/\eta,T) - \eta V(S^*,t^*;K,T) + (\eta-1)KV_K\Big) g(\eta) d\eta}{\frac{K^2 V_{KK}}{2}} \end{aligned} \tag{3.14}$$

is positive if $\sigma_J^2 > 0$, and bounded. Thus a solution W to equation (3.7) exists with σ_L given by equation (3.14). Hence W satisfies Condition (3.2), and $E = W - V$ satisfies equation (3.9) in $\Omega_f \times [T_{\min}, T_{\max}]$. Equation (3.9) implies that if σ_L is given by equation (3.14) then $f = 0$ in $\Omega_f \times [T_{\min}, T_{\max}]$, and so from Lemma 3.1, $V = W$ in $\Omega_f \times [T_{\min}, T_{\max}]$. Lemma 3.1 also shows that $V - W = 0$ if and only if $f \equiv 0$ in $\Omega_f \times [T_{\min}, T_{\max}]$, and thus σ_L given by equation (3.14) is unique. □

Remark 3.2 (Homogeneity of V) *As noted by Merton (1973), if the underlying stock return distribution is independent of S, call/put option values are homogeneous of degree one in S and K, e.g. if $\sigma_J^2 = const.$ and $g(\eta)$ is independent of S, then the solution to equation (3.5) with initial condition $V(S^*, t^*; K, T = t^*) = \max(S^* - K, 0)$ is such that*

$$V(\rho S^*, t^*; \rho K, T) = \rho V(S^*, t^*; K, T). \tag{3.15}$$

Remark 3.3 (Bounded σ_L) *Note that Proposition 3.1 requires that $V_{KK} > 0$. For vanilla options with convex payoffs, the solution to equation (3.5) is such that $V_{KK} > 0$ in $\Omega_f \times [T_{\min}, T_{\max}]$. Clearly, the usual call payoffs have $V_{KK} = 0$ away from the strike at $T = t^*$, and for $T > t^*$, $V_{KK} \to 0$ as $K \to 0, \infty$. For short term options the denominator of*

$$\frac{\lambda \int_0^\infty \Big(\eta V(S^*, t^*; K/\eta, T) - \eta V(S^*, t^*; K, T) + (\eta - 1) K V_K \Big) g(\eta) d\eta}{\frac{K^2 V_{KK}}{2}} \tag{3.16}$$

will tend to zero faster than the numerator (which is a non-local term) as $K \to 0, \infty$. Hence the local volatility will become unbounded as $K_{\min} \to 0$, and $K_{\max} \to \infty$, $T \to t^*$. This means that we cannot expect to solve the calibration problem over the entire domain $t^* \le T \le T_{max}$, $0 \le K \le \infty$, but only over a subset of this domain.

Now, if we calibrate $\sigma_L(S^*, t^*; K, T)$ to the prices of vanilla options for fixed (S^*, t^*) at various values of (K, T), then we can re-label $K = S, T = t$ to obtain $\sigma_L(S^*, t^*; S, t)$. However, we have only matched the prices at a fixed value of (S^*, t^*). There is no guarantee that the calibrated σ_L found using equation (3.4) will match the values of exotic options derived from a jump diffusion model using equation (3.2), especially for path-dependent options such as cliquets. For related discussion, see Hirsa et al. (2003); Schoutens et al. (2004).

Lemma 3.2 (Homogeneity of σ_L) *For vanilla options, denote the value of the option as a function of (K, T) for fixed (S^*, t^*) by $V(S^*, t^*; K, T)$. Also, denote the local volatility function by $\sigma_L(S^*, t^*; K, T)$. If*

$$\begin{aligned} \sigma_J(\rho S^*, t^*; \rho K, T) &= \sigma_J(S^*, t^*; K, T) \\ V(\rho S^*, t^*; \rho K, T) &= \rho V(S^*, t^*; K, T) \ , \end{aligned} \tag{3.17}$$

then $\sigma_L(\rho S^*, t^*, \rho K, T) = \sigma_L(S^*, t^*, K, T)$.

Proof. This follows directly from substituting the conditions (3.17) into equation (3.14). □

Remark 3.4 (Significance of Lemma 3.2.) *In particular, if $\sigma_J = const.$ in equation (3.2), then the local volatility function $\sigma_L(S^*, t^*; S, t) = \sigma_L(\rho S^*, t^*; \rho S, t)$. If $S^* = K$, then $\sigma_L(K, t^*; S, t) = \sigma_L(K^*, t^*; K^*(S/K), t)$ is* "sticky delta"*, i.e. a function only of S/K for vanilla options.*

Remark 3.5 (Jump size η concentrated near unity) *Suppose that $g(\eta) \sim \delta(\eta - 1)$, i.e. the jump size distribution is concentrated near $\eta = 1$, then*

$$\begin{aligned} V(K/\eta) &= V(K) + \frac{K}{\eta} V_K(K)(1-\eta) + \frac{V_{KK}(K)K^2}{2}\frac{(1-\eta)^2}{\eta^2} + O[(1-\eta)^3] \\ &\simeq V(K) + \frac{K}{\eta} V_K(K)(1-\eta) + \frac{V_{KK}(K)K^2}{2}(1-\eta)^2, \end{aligned} \tag{3.18}$$

so that

$$\int_0^\infty \Big(\eta V(S^*, t^*; K/\eta, T) - \eta V(S^*, t^*; K, T) + (\eta - 1) K V_K \Big)\, g(\eta)\, d\eta \simeq \frac{V_{KK}K^2}{2} \int_0^\infty (1-\eta)^2\, g(\eta)\, d\eta. \tag{3.19}$$

Substituting equation (3.19) into equation (3.14) gives

$$\sigma_L^2 \simeq \sigma_J^2 + \lambda \int_0^\infty (1-\eta)^2\, g(\eta)\, d\eta. \tag{3.20}$$

This result is simply a formal statement of the well-known concept that small jumps are indistinguishable from diffusion (Cont and Tankov, 2004). In this unusual situation, we can expect that σ_L calibrated from vanilla prices can be used to value exotics with little error (in this special case, σ_L has no dependence on (S^*, t^*)).

3.2 Volatility Surface Based on the Analytic Expression

In order to provide a realistic volatility surface for our numerical tests, we will assume that the synthetic market process is given by equation (3.1), with the data in Table 1. We define the current time $t^* = 0$ in this and subsequent sections. For future reference, this table also shows the constant Black-Scholes implied volatilities which match the prices of at-the-money vanilla call options with maturities of $T = .25$ and $T = 5$. The parameters in Table 1 are similar those reported by Andersen and Andreasen (2000), which were obtained by calibration to S&P 500 index option data.

We will construct a deterministic local volatility surface (with no jumps) which is consistent with the observed market prices (from our synthetic market). We use the expression for the local volatility surface given by equation (3.14). The exact analytical solution for European options under a Merton jump diffusion model (Merton, 1976) is used to compute the prices and derivatives in equation (3.14). The non-trivial integral in equation (3.14) was computed by converting the integral to a convolution form and then using an FFT. We could also, of course, use the method in Dupire (1994).

Even with an analytical solution, the local volatility becomes unbounded as $T \to t^* = 0$ for $K \neq S^*$, and for finite T, with $K \to 0, \infty$ (recall Remark 3.3). In particular, the numerical computation becomes very ill-posed for $T <$ one month, and for $K > 1.5\ S^*$ and $K < .5\ S^*$ (the

Parameter	Value
S^*	100
σ_J	.20
r	.05
λ	.10
μ	-.90
γ	.45
$\sigma_{BS}, T = 0.25$	.2359
$\sigma_{BS}, T = 5.0$	.3167

TABLE 1: *Parameters for the jump diffusion model, with price process given in equation (3.1). The jump size distribution is given in equation (2.3). Also shown is σ_{BS}, the constant volatility that, if used in a Black-Scholes model with no jumps, reproduces the jump diffusion model price of an at-the-money vanilla call with the specified maturity ($T = .25, 5$).*

initial asset price $S^* = 100$). We set a maximum value for the local volatility of 3.0, to avoid problems with unbounded values from equation (3.14).

We generate a dense grid of $\sigma(S,t)$ for $S \in [50, 150]$, and $t \in [1/12, 5.0]$ with S spacing of 1.00 and t spacing of 1 month. Any other data needed is obtained by linear interpolation, for values of (S,t) within the grid. For values outside the grid, we use the nearest grid value.

This grid of $\sigma(S,t)$ values would have to be considered as a very good estimate of the volatility surface which matches the synthetic market prices (in practice, we would not have data for far out of the money options with short maturities). The resulting local volatility surface is shown in Figure 1.

Table 2 shows the synthetic market prices of the vanilla calls, at various strikes and times, compared with the prices obtained using the volatility surface shown in Figure 1, computed using equation (3.14). The fit is quite good, considering that the actual surface which matches all the prices would become unbounded as $T \to 0$, and for $K \gg S^*$ or $K \ll S^*$. Recall that we limit the maximum value of $\sigma \leq 3.0$, and we have used a constant surface for $t \leq$ month.

3.3 Volatility Surface Obtained by Calibration

Of course, in general we would not know the precise representation of the real world process. A more realistic example of fitting a local volatility surface would be the algorithm in Coleman et al. (1999). To summarize, this method uses a set of specified knot locations, and then attempts to determine the best $\mathcal{L}_2$ fit to the data using a spline interpolant.

Synthetic market prices were generated for vanilla puts and calls at strikes $\{70, 80, 90, 100, 110, 120, 130\}$, at monthly intervals from $[0, 1.0]$ and yearly intervals from $[1.0, 5.0]$. These prices were computed using the exact European prices under a Merton jump diffusion model, with the data in Table 1. In order to avoid unbounded values of the local volatility (i.e. Remark 3.3), as well as improve the fit to the data, the maximum value of the local volatility was constrained to be $\sigma = 1.0$.

The knot locations of the spline representation of the local volatility surface were specified to be at $S = \{70, 100, 130\}$ and at times $t = \{0.0, 0.5, 1.0, 3.0, 5.0\}$, giving a total of fifteen parameters. Since the number of knot locations for the spline is relatively small, this has the effect of regularizing

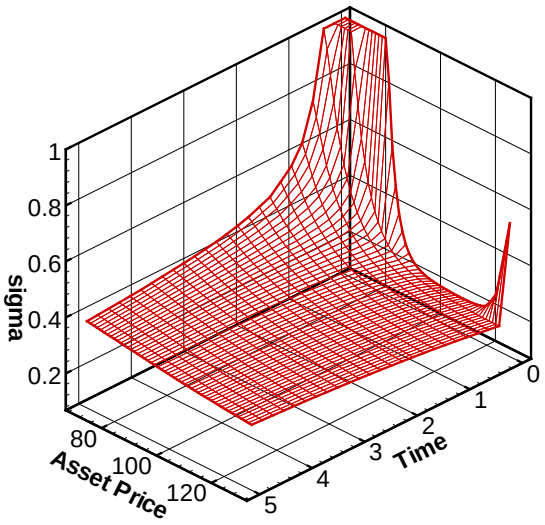


FIGURE 1: *This volatility surface was obtained by using equation (3.14) with the exact analytical solution for a European option under a Merton jump diffusion (parameters are given in Table 1). The plot is truncated at a maximum value of* $\sigma = 1.0$. *The actual surface used had a maximum value of* $\sigma = 3.0$.

the surface, as described in Coleman et al. (1999). The resulting volatility surface is shown in Figure 2.

Table 3 shows the error between the prices computed using the surface in Figure 2 and the European call prices. Note that in this case, there is more error for $T \leq 1$ and $S^* = 100$ compared to the surface used in Table 2.

4 Cliquet Contracts

4.1 Contract Description

As noted earlier, cliquet options have become popular because these contracts provide protection against downside risk while retaining significant upside potential. This is achieved by offering a combination of floors and caps on returns on the underlying asset.

Let $S(t_i)$ be the value of the underlying asset at observation time t_i. There are a total of N_{obs} observation times over the life of the contract. Define the return during the period $[t_{i-1}, t_i]$ to be

$$R_i = \frac{S(t_i) - S(t_{i-1})}{S(t_{i-1})}. \tag{4.1}$$

The payoff of a cliquet is

$$\text{Payoff} \;=\; \text{Notional} \;\times \max\left(F_g, \min\left(C_g, \left[\sum_{i=1}^{N_{obs}} \max\left\{F_l, \min\left\{C_l, R_i\right\}\right\}\right]\right)\right) \tag{4.2}$$

	$K = 90$		$K = 100$		$K = 110$	
Expiry time	Vol. Surf.	Synthetic	Vol. Surf.	Synthetic	Vol. Surf.	Synthetic
1/12	10.67	10.80	2.77	2.75	0.20	0.18
1/6	11.64	11.71	4.14	4.15	0.79	0.77
1/4	12.56	12.62	5.30	5.32	1.50	1.49
1/2	15.18	15.18	8.26	8.27	3.77	3.77
1	19.69	19.59	13.09	13.07	8.12	8.12
2	26.83	26.72	20.85	20.78	15.78	15.74
3	32.67	32.53	27.18	27.07	22.33	22.23
4	37.64	37.47	32.59	32.43	28.01	27.87
5	41.99	41.79	37.31	37.12	33.03	32.84

TABLE 2: *Comparison of vanilla call synthetic market prices (jump diffusion model) and the volatility surface model computed using equation (3.14). Input parameters for the jump diffusion model are given in Table 1. The local volatility surface is shown in Figure 1. The computed prices are accurate to the number of digits shown.*

where C_l, F_l are local caps and floors placed on the individual returns, and C_g, F_g are a global cap and floor respectively.

Note that a modification to equation (4.2) is

$$\text{Payoff} \;=\; \text{Notional} \times \max\left(F_g, \hat{R} + \min\left(0, \left[\sum_{i=1}^{N_{obs}} \max\left\{F_l, \min\left\{0, R_i\right\}\right\}\right]\right)\right) \tag{4.3}$$

where $\hat{R}$ is a specified nominal return. If $F_l < 0$, then equation (4.3) is the payoff of a *reverse cliquet*. Such contracts give the holder a higher nominal return $\hat{R}$ in exchange for bearing some downside market risk.

Since we are solving the PIDE (2.2) backwards in time from the maturity date to the valuation date, we need to maintain additional state variables as one would in a dynamic programming context. There are two obvious approaches that we now discuss and compare.

4.2 Formulation

4.2.1 Running Sum Formulation

In this case, we introduce two new state variables: P, corresponding to the asset price at the previous observation, and Z which is defined below. The value of the option is then given by $V = V(S, t; P, Z)$. Assume that $t_k < t < t_{k+1}$. Let

$$Z(t_k < t < t_{k+1}) = \sum_{i=1}^{k} \max\left(F_l, \min\left(C_l, R_i\right)\right), \tag{4.4}$$

where $Z(t < t_1) \equiv 0$. Consequently, the payoff (4.2) at $t = T$ becomes

$$\text{Payoff} \;=\; \text{Notional} \;\times \max\left(F_g, \min\left(C_g, Z\right)\right). \tag{4.5}$$

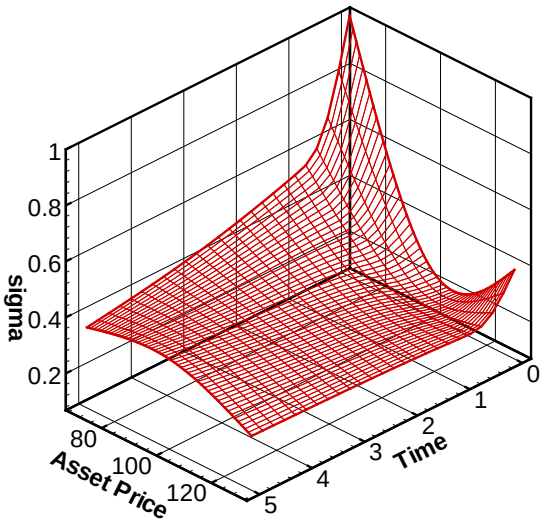


FIGURE 2: *This volatility surface was obtained by using a least squares fit (Coleman et al., 1999) to the exact European call and put prices (under a Merton jump diffusion). The Merton model parameters are given in Table 1. The actual surface used had a maximum value of* $\sigma = 1.0$.

Similarly, let $P(t_k < t < t_{k+1}) = S(t_k)$ denote the value of the asset at the most recent observation. If t_k^-, t_k^+ are the times the instant before and after the k^{th} observation, then, following Wilmott (1998), no-arbitrage considerations lead to the following jump conditions:

$$\begin{aligned} R &= \frac{S - P^-}{P^-} \\ R^* &= \max\left(F_l, \min\left(C_l, R\right)\right) \\ Z^+ &= Z^- + R^* \\ P^+ &= S \\ V(S, t^-; P^-, Z^-) &= V(S, t^+; P^+, Z^+), \end{aligned} \tag{4.6}$$

where $P^+ = P(t_k^+)$, $P^- = P(t_k^-)$, $Z^+ = Z(t_k^+)$, and $Z^- = Z(t_k^-)$.

4.2.2 Average Formulation

It was demonstrated by Zvan et al. (1999) that the use of the arithmetic average for the additional state variable was superior to a running sum formulation in terms of numerical performance for Asian options. Consequently, we will also investigate an alternative formulation which uses average (capped and floored) returns. Again the value of the option is given by $V = V(S, t; P, Z)$ where P is the previous asset price as defined in the running sum context. However, in this case the state

	$K = 90$		$K = 100$		$K = 110$	
Expiry time	Vol. Surf.	Synthetic	Vol. Surf.	Synthetic	Vol. Surf.	Synthetic
1/12	10.81	10.80	3.01	2.75	0.21	0.18
1/6	11.91	11.71	4.43	4.15	0.82	0.77
1/4	12.92	12.62	5.60	5.32	1.55	1.49
1/2	15.49	15.18	8.45	8.27	3.78	3.77
1	19.69	19.59	13.11	13.07	8.04	8.12
2	26.82	26.72	20.89	20.78	15.81	15.74
3	32.64	32.53	27.21	27.07	22.34	22.23
4	37.57	37.47	32.57	32.43	27.96	27.87
5	41.91	41.79	37.28	37.12	32.94	32.84

TABLE 3: *Comparison of vanilla call synthetic market prices (jump diffusion model) and the calibrated volatility surface model (using a low order spline fit (Coleman et al., 1999)). Input parameters for the jump diffusion model are given in Table 1. The calibrated local volatility surface is shown in Figure 2. The computed prices are accurate to the number of digits shown.*

variable Z is defined as

$$Z_{avg}(t_k < t < t_{k+1}) = \frac{1}{k}\sum_{i=1}^{k} \max\left(F_l, \min\left(C_l, R_i\right)\right), \tag{4.7}$$

where $Z_{avg}(t < t_1) \equiv 0$. In this case payoff (4.2) becomes

$$\text{Payoff} \;=\; \text{Notional} \;\times \max\left(F_g, \min\left(C_g, N_{obs} \times Z_{avg}\right)\right). \tag{4.8}$$

When using the average formulation the jump conditions are given by

$$\begin{aligned} R &= \frac{S - P^-}{P^-} \\ R^* &= \max\left(F_l, \min\left(C_l, R\right)\right) \\ Z^+_{avg} &= Z^-_{avg} + \frac{(R^* - Z^-_{avg})}{k} \\ P^+ &= S \\ V(S, t^-; P^-, Z^-_{avg}) &= V(S, t^+; P^+, Z^+_{avg}). \end{aligned} \tag{4.9}$$

An advantage of the average formulation is that the possible range of values of $Z = Z_{avg}$ is limited to $\min(0, F_l) \le Z \le C_l$, independent of N_{obs}. In the case of the running sum formulation, the Z values are bounded by $\min(0, N_{obs} \times F_l) \le Z \le N_{obs} \times C_l$, so that the grid is dependent on the number of observations. Notice that we are interested in the solution $V(S^*, t = 0; P = S^*, Z = 0)$, where S^* is the current value of the underlying asset. As we solve backwards in time, the solution for some of the large Z values cannot affect the solution at $V(S^*, t = 0; P = S^*, Z = 0)$ in the running sum formulation. Consequently, some of the nodes in the Z-direction are wasted unless the grid is dynamically reconstructed after each observation.

5 Numerical Solution

5.1 Discretization

Note that the PIDE (2.2) is independent of the new state variables (P, Z). Consequently, we can discretize the state variables as

$$\{P_1, \ldots, P_j, \ldots, P_{j_{\max}}\} \quad \text{and} \quad \{Z_1, \ldots, Z_k, \ldots, Z_{k_{\max}}\}.$$

For each discrete value of (P_j, Z_k), we can solve the one dimensional PIDE (2.2) at times between the observation dates. To move the solution across an observation date, we use the jump conditions (4.6) or (4.9). Notice that the jump conditions (4.6) and (4.9) are undefined if $P = 0$. Therefore, it is important to discretize P so that $P_1 > 0$. Note that for geometric Brownian motion with lognormally distributed jumps, a stock price of $S = 0$ is unattainable in finite time.

For fixed (P_j, Z_k), each one-dimensional PDE (2.2) is a function of (S, t) only. Our numerical experiments utilize Crank-Nicolson timestepping with the modification suggested by Rannacher (1984). Other details of the discretization can be found in Pooley et al. (2003). In particular, we employ the iterative method described in that paper for the nonlinear uncertain volatility models. In situations where a jump diffusion model was used, the discrete algebraic equations are solved using a fixed point iteration combined with an FFT evaluation of the integral term in the PIDE (2.2). This is described in detail in d'Halluin et al. (2005, 2004). The tolerances for all iterative methods (within each timestep) were set to guarantee that the error in the solution of the discretized equations did not affect the first six significant digits of the solution.

5.2 Similarity Reduction

As discussed by Forsyth et al. (2002), it is generally necessary to carry out an interpolation operation to approximate the jump conditions at observation dates. Denote the possible dependence of σ on P in equation (2.2) by $\sigma = \sigma(S, t; P)$ (dropping possible dependence on Γ). This interpolation can be avoided if we assume that

$$\sigma(S, t; P) = \sigma(\rho S, t; \rho P). \tag{5.1}$$

If equation (5.1) holds, the payoff is given by either (4.5) or (4.8), and the jump conditions are given by either (4.6) or (4.9), then from equation (2.2) we have that V is homogeneous of degree zero in (S, P):

$$V(S, t; P, Z) = V\left(\rho S, t; \rho P, Z\right). \tag{5.2}$$

Setting $\rho = P^*/P$ gives

$$V(S, t; P, Z) = V\left(\frac{S}{P} \times P^*, t; P^*, Z\right), \tag{5.3}$$

which implies that we need only solve for one reference value $P = P^*$. This effectively reduces the dimensionality of the problem from three to two. As long as the node $S = P^*$ is in the S grid, no interpolation is required (in the S direction) to satisfy the jump conditions (4.6) or (4.9). In the following, we will refer to assumption (5.1), which then implies equation (5.3), as the similarity reduction.

The assumption (5.1) seems somewhat peculiar, but has a modelling rationale which we will discuss in a later section.

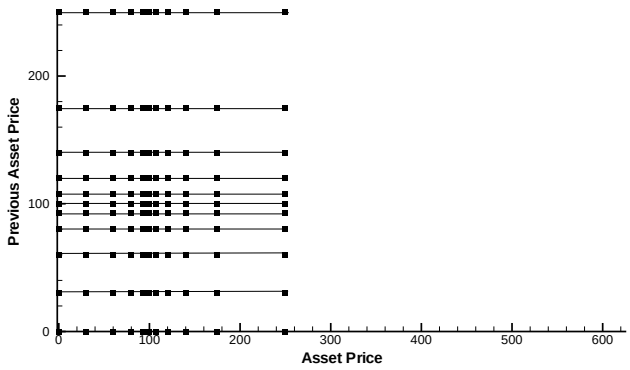


FIGURE 3: *Repeated grid, constructed using the same asset price S grid for each discrete setting of the variable P (the asset price at the preceding observation time).*

5.3 Mesh Construction

With regard to the mesh for the Z variable, there are no particularly noteworthy issues. We simply use a uniformly spaced grid. However, some issues arise in the construction of the P grid (for cases where no similarity reduction is available). Suppose that we use an S grid with $\mathcal{S}_g = \{S_1, \ldots, S_i, \ldots, S_{i_{\max}}\}$ and a P grid, $\mathcal{P}_g = \{P_1, \ldots, P_j, \ldots, P_{j_{\max}}\}$, with $\mathcal{P}_g = \mathcal{S}_g$ (i.e. a Cartesian product $S \times P$ grid, with the same node spacing in the S and P directions). In this case, no interpolation in the S or P directions is required during the application of the state variable updating rule

$$P^+ = S. \tag{5.4}$$

A illustration of a set of grids constructed in this fashion is shown in Figure 3. We refer to this grid as a *repeated grid* in the following. We emphasize that a major advantage of a repeated grid, for pricing cliquet options, is that no interpolation error is introduced in the (S, P) plane at each observation date.

In Windcliff et al. (2001), it is shown that this type of grid results in poor convergence for shout options. Normally, we choose a fine node spacing near the initial asset price $S = S^*$, since this is the region of most interest. However, since the nodes $P = S$ for all values of S are required during the application of the jump condition (5.4), these values may have poor accuracy in areas where the S node spacing is large. It is therefore desirable to have a fine node spacing in the S direction for all nodes near the diagonal of the (S, P) grid.

Suppose we have a prototype S grid constructed with a fine node spacing near $S = S^*$. Denote this set of nodes by $\mathcal{S}_g = \{S_1, \ldots, S_i, \ldots, S_{i_{\max}}\}$. We also assume that the grid has been constructed so that the point S^* is contained in the discretization. In other words, there is an index i^* such that $S_{i^*} = S^*$ and $S_{i^*} \in \mathcal{S}_g$. Let $\mathcal{S}_g^j = \{S_1^j \ldots, S_i^j, \ldots, S_{imax}^j\}$ represent the S nodes corresponding to the discrete value P_j. The following algorithm is used to construct $\mathcal{S}_g^j$, with $P_j \in \mathcal{S}_g^j$:

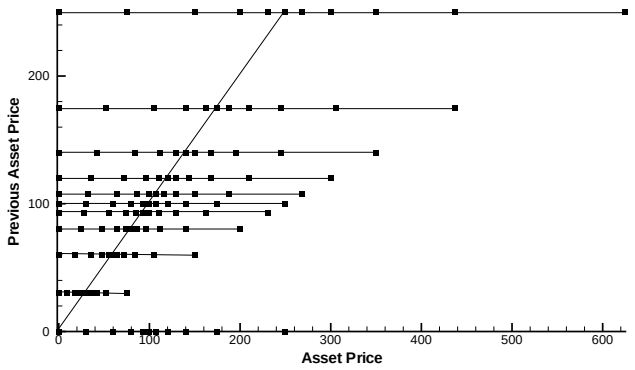


FIGURE 4: *Scaled grid, constructed using algorithm (5.5).*

Scaled Grid Construction

$$
\begin{aligned}
&\text{Set } P_j = S_j; \quad j = 1, \ldots, i_{\max}; \quad S_j \in S_g \\
&\text{Set } j_{\max} = i_{\max} \\
&\text{For } j = 1, \ldots, j_{\max} \\
&\qquad S_i^j = S_i P_j / S^* \quad i = 1, \ldots, i_{\max} \\
&\text{EndFor}
\end{aligned}
\tag{5.5}
$$

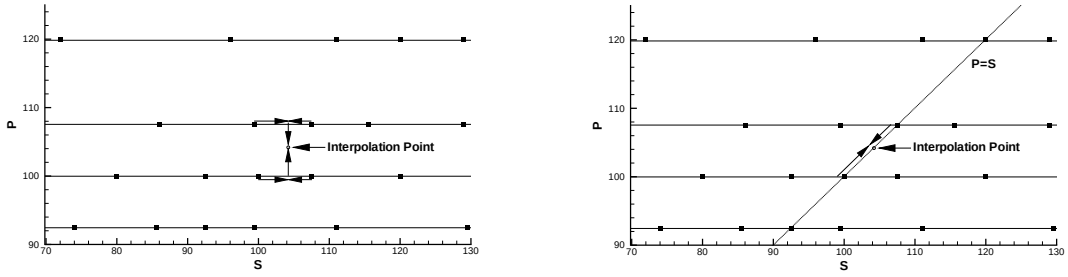
Since there is an i^* such that $S_{i^*} = S^*$, $S_{i^*}^j = P_j$. In other words, for each line of constant P_j, there is a node on the diagonal $S_i^j = P_j$, as depicted in Figure 4.

5.4 Interpolation

If $\mathcal{S}_g^j$ is constructed using algorithm (5.5), then interpolation is required to satisfy the state variable updating rule (5.4). An obvious method is to linearly interpolate along the S axis and then along the P axis, which we refer to as xy interpolation in the following. If we omit the dependence of V on the variables Z and t for brevity, then xy interpolation is defined as:

$$
\begin{aligned}
V(S,P) &= V^S(P_{low}) + \frac{V^S(P_{high}) - V^S(P_{low})}{P_{high} - P_{low}} \quad (P - P_{low}) \\
V^S(P) &= V(S_{low}, P) + \frac{V(S_{high}, P) - V(S_{low}, P)}{S_{high} - S_{low}} \quad (S - S_{low}) \\
&S_{low} \le S \le S_{high} \\
&P_{low} \le P \le P_{high}.
\end{aligned}
\tag{5.6}
$$

In Windcliff et al. (2001) it is argued that diagonal interpolation (along the diagonal of the grid shown in Figure 4) is more suited to capturing the non-smooth payoff of a shout option. Diagonal



(a) *xy interpolation.* (b) *Diagonal interpolation along $P = S$.*

FIGURE 5: *Different interpolation strategies.*

interpolation is defined as

$$V(S, P = S) = V(S = P_{low}, P_{low}) + \frac{V(P_{high}, P_{high}) - V(P_{low}, P_{low})}{P_{high} - P_{low}} \quad (S - P_{low})$$
$$P_{low} \leq S \leq P_{high}. \tag{5.7}$$

Unlike xy interpolation, this method is exact if a similarity reduction is valid. These two approaches (xy and diagonal) are illustrated in Figure 5.

5.5 Boundary Conditions

As previously discussed, away from observation dates, we need to solve a set of one dimensional PIDEs of the form (2.2) for each discrete value of (P, Z). Consequently, boundary conditions must be specified. Normally, a finite computational domain $[S_{\min}, S_{\max}]$ is specified for a one dimensional Black-Scholes equation. Usually, $S_{\max}$ is selected to be a large value, and the boundary condition $V_{SS} = 0$ is specified at $S = S_{\max}$ as suggested by a variety of authors, including Tavella and Randall (2000) and Wilmott (1998). The reason for this is that many contracts (including cliquets) are asymptotically linear as $S \to \infty$. For a discussion of the stability issues surrounding this boundary condition, see Windcliff et al. (2004). In the following, we will specify $V_{SS} = 0$ at $S = S_{\max}$.

Since the PDE degenerates to an ODE at $S = 0$, which is easily implemented numerically, usually $S_{\min} = 0$. However, as noted in Section 5.1, the return $R_i = (S(t_i) - S(t_{i-1}))/S(t_{i-1})$ becomes undefined when $S(t_{i-1}) = P = 0$. Since the jump conditions require that $P^+ = S$, having a node at $S = 0$ causes difficulty. Consequently, we should view the solution to the cliquet valuation problem as being embedded in the computational domain $S_{\min} \leq (S, P) \leq S_{\max}$, with $S_{\min} > 0$, and $S_{\max} < \infty$. We seek the solution in the limit as $S_{\min} \to 0$ and $S_{\max} \to \infty$. From the nature of the cliquet payoff, it is reasonable to impose the boundary condition $V_{SS} = 0$ at $S = S_{\min}$ as well. Under normal market parameters, setting $V_{SS} = 0$ at the lower boundary results in a first order hyperbolic equation with outgoing characteristic, which contrasts with the more delicate situation studied in Windcliff et al. (2004) at $S = S_{\max}$. In our numerical tests, we will select a value of $S_{\min}$ on a coarse grid, and then reduce $S_{\min}$ as the grid is refined, so as to ensure the correct limiting behavior.

5.6 Effect of Finite Computational Domain on the Jump Conditions

If the scaled one dimensional grids are constructed as shown in Figure 4, then there will be situations where $S_i^j > P_{j_{\max}}$ or $S_i^j < P_{j_{\min}}$. In these cases, our computational domain does not have sufficient data to allow interpolation of the state variable updating rule $P^+ = S$. If this happens, we assume that this data can be approximated by assuming that the similarity reduction (5.3) is locally valid. From equation (5.3), this means that

$$\begin{aligned} V(S_i^j, t; P = S_i^j, Z) &\simeq V(P_{j_{\max}}, t; P_{j_{\max}}, Z); \qquad S_i^j > P_{j_{\max}} \\ &\simeq V(P_{j_{\min}}, t; P_{j_{\min}}, Z); \qquad S_i^j < P_{j_{\min}}. \end{aligned} \tag{5.8}$$

We will refer to (5.8) as a *similarity extrapolant*. In situations where there is a similarity reduction, this extrapolation scheme is exact. Using a local volatility surface would invalidate the use of the similarity reduction. However, typically the volatility function is assumed to be constant outside of some range of asset values near the current asset price. Consider the boundary for large asset values. Since the state variable updating rules only query values of S near $S = P_{j_{\max}}$, the effect of far-field errors introduced by the approximation of constant volatility can be made arbitrarily small, as demonstrated by Kangro and Nicolaides (2000).

5.7 Properties of the Discrete Equations

Suppose we solve a full three dimensional cliquet problem, with variables (S, P, Z). Consider the special case where a similarity reduction (Section 5.2) is valid. In this case, it seems natural to require that our grid construction/discretization method is discretely homogeneous of degree zero in (S, P), and that there should be no interpolation error incurred in the (S, P) plane after applying the jump conditions. A grid construction/discretization method satisfying these properties should also be useful when solving problems where a similarity reduction is not valid.

Let

$$U_{ijk}^n = U(S_i^j, P_j, Z_k, \tau^n) \tag{5.9}$$

be the discrete solution to the cliquet pricing problem. Note that we have allowed the grid $\mathcal{S}_g^j$ to depend on P_j. Let U_{jk}^n be the vector of discrete solution values for grid $\mathcal{S}_g^j$, i.e.

$$(U_{jk}^n)_i \quad = \quad U_{ijk}^n \ . \tag{5.10}$$

Since equation (2.2) contains no derivatives w.r.t. (P, Z), then, given U_{ij}^n, we can solve for U_{ij}^{n+1} for each (jk) independently. If a fully implicit ($\theta = 1$) or Crank-Nicolson ($\theta = 1/2$) timestepping method is used, and equation (2.2) is discretized as in d'Halluin et al. (2005, 2004), then we have that

$$(I + \theta M_j) U_{jk}^{n+1} \quad = \quad (I - (1-\theta) M_j) U_{jk}^n \ , \tag{5.11}$$

where $M_j = M(\{S_i^j\}, P_j)$ is the matrix form of the discretization operator (for a given grid $\mathcal{S}_g^j$). Note that since equation (2.2) is independent of Z, then M_j has no k dependence.

We first gather some conditions which are required for a similarity reduction (Section 5.2) to be valid.

Conditions 5.1 (Conditions for a Similarity Reduction) *The following conditions are required in order to use the similarity reduction method described in Section 5.2.*

- *The payoff of the cliquet is homogeneous of degree zero in (S, P), i.e. for any scalar $\lambda > 0$*

$$V(\lambda S, \lambda P, Z, \tau = 0) \quad = \quad V(S, P, Z, \tau = 0) \ . \tag{5.12}$$

- *The discrete form of the PIDE operator (2.2) is homogeneous of degree zero in (S, P), i.e. for any scalar $\lambda > 0$*

$$M(\{\lambda S_i^j\}, \lambda P_j) \quad = \quad M(\{S_i^j\}, P_j) \tag{5.13}$$

- *The jump conditions are given as in (4.6) and (4.9).*

Remark 5.1 (Homogeneity Property of the Discrete Operator) *Property (5.13) holds if either $\sigma = const.$ in equation (2.2), or σ satisfies condition (5.1); and, in addition, boundary conditions $V_{SS} = 0$ are imposed at $S = S_{min}, S_{max}$, and the discretization method in d'Halluin et al. (2005, 2004) is used.*

We also gather some conditions on the grid construction, discretization and jump condition enforcement that we wish to impose.

Conditions 5.2 (Grid Construction/Discretization Properties) *We assume that the grid is constructed with the following conditions*

- *The mesh is constructed using the scaled grids as described in Algorithm 5.5.*
- *Diagonal interpolation (5.7) is used where required to enforce the jump conditions. The similarity extrapolant (5.8) is used if missing data is required.*
- *The boundary condition $V_{SS} = 0$ is imposed for each grid at $S = S_{min}^j$, $S = S_{max}^j$.*

We can now state an interesting property of grid construction and discretization methods which satisfy Conditions 5.2.

Property 5.1 (Grid Construction/Discretization Property) *Provided that the similarity reduction Conditions 5.1 are satisfied, and the grid is constructed satisfying Conditions 5.2 then*

$$U_{ijk}^n \quad = \quad U_{ij^*k}^n \quad ; \quad \forall i, j, k, n \ , \tag{5.14}$$

where j^* denotes the index such that $P_{j^*} = S^*$. Equation (5.14) then implies that

$$\begin{aligned} U(\lambda S_i^j, \lambda P_j, Z_k, \tau^n) \quad &= \quad U(S_i^j, P_j, Z_k, \tau^n) \\ &\qquad \lambda = \frac{P_l}{P_j} \ , \end{aligned} \tag{5.15}$$

which can be interpreted as a discrete homogeneity property. In addition, there is no interpolation error introduced in the (S, P) planes upon applying jump conditions (4.6) or (4.9).

Proof. Suppose that the scaled grid for S_i^j is constructed as in Algorithm 5.5. Since the payoff is homogeneous of degree zero in (S, P) (condition (5.12)), then at timestep $n = 0$, we have

$$\begin{aligned} U_{ijk}^0 &= U(S_i^j, P_j, Z_k, \tau = 0) \\ &= U(\frac{P_j}{S^*} S_i, \frac{P_j S^*}{S^*}, Z_k, \tau = 0) \\ &= U(S_i, S^*, Z_k, \tau = 0) \\ &= U_{ij^*k} \ , \end{aligned} \tag{5.16}$$

Suppose that at timestep n, we have that

$$U_{ijk}^n = U_{ij^*k}^n \ . \tag{5.17}$$

From condition (5.13) we have that

$$\begin{aligned} M_j &= M\left(\left\{\frac{P_j}{S^*} S_i\right\}, \frac{P_j S^*}{S^*}\right) \\ &= M(\{S_i\}, S^*) \\ &= M_{j^*} \ , \end{aligned} \tag{5.18}$$

It then follows from equation (5.11), using equations (5.17) and (5.18) that

$$\begin{aligned} U_{jk}^{n+1} &= (I + \theta M_j)^{-1}(I - (1-\theta)M_j) U_{jk}^n \\ &= (I + \theta M_{j^*})^{-1}(I - (1-\theta)M_{j^*}) U_{j^*k}^n \\ &= U_{j^*k}^{n+1} \ . \end{aligned} \tag{5.19}$$

Consequently, since $U_{jk}^0 = U_{j^*k}^0$, then $U_{jk}^n = U_{j^*k}^n$ for all steps between applications of the jump conditions.

Let $(U_{jk}^{n+1})^- = (U_{j^*k}^{n+1})^-$ be the discrete solution the instant before application of the jump conditions (going backwards in time), and $(U_{jk}^{n+1})^+$ be the solution the instant after application of the jump conditions (backwards in time). If the jump conditions (4.6) or (4.9) are specified, and if diagonal interpolation (5.7) is used, along with the similarity extrapolant (5.8), then it is easy to see that

$$(U_{jk}^{n+1})^+ = (U_{j^*k}^{n+1})^+ \ , \tag{5.20}$$

hence properties (5.14) and (5.15) hold after application of the jump conditions. Note that the jump conditions (4.6) or (4.9) require evaluation of (noting equation (5.15))

$$(U(S_i^j, S_i^j, Z, \tau^{n+1})^- = (U(S^*, S^*, Z, \tau^{n+1})^- \tag{5.21}$$

hence from equation (5.7), there is no interpolation error in the (S, P) plane with diagonal interpolation. □

Remark 5.2 (Significance of Property 5.1) *If the Conditions 5.1 for a similarity reduction are rigorously satisfied, and the grid is constructed satisfying Conditions 5.2, then the solution vector is discretely homogeneous of degree zero in* (S, P)*, as in equation (5.15). Furthermore, application of the jump conditions does not generate any interpolation error in the* (S, P) *plane. In this case, we need only solve for a single value of* $P = S^*$ *in each* (S, P) *plane, i.e. there is no need to solve a full three dimensional problem. However, in cases where the similarity reduction is not valid, we expect that a grid satisfying Conditions 5.2 will still be desirable. For example, if* $\sigma = \sigma(S, t)$*, then in general a similarity reduction cannot be used. However, any interpolation error introduced by the diagonal interpolant (on the grid satisfying Conditions 5.2) will be a result of deviations from a constant volatility, which we expect to be small in regions where the local volatility function is smooth.*

6 Numerical Tests: Methods

6.1 Comparison of Running Sum and Average Formulations

As a first test we compare the convergence of the running sum and average formulations as the grid size is refined and number of timesteps is increased. Details of the contract used in these tests are provided in Table 4. Note that the contract is the same as that studied in Wilmott (2002). For these tests we use a constant volatility model without jumps. Input parameters are presented Table 5.

The results of the convergence tests are shown in Table 6. Since $\sigma_J = const.$, we can use the similarity reduction to reduce this to a two dimensional PDE. A series of tests was carried out where at each refinement level new nodes were inserted between each pair of nodes in the coarser grid, a new node was added between 0 and $S_{\min}$ from the previous grid, and the timestep size was reduced by a factor of two. Contrary to the results found by Zvan et al. (1999) for Asian options, the running sum formulation seems to be converging faster than the average formulation. In all subsequent tests, we will use the running sum formulation.

Note that we use second order methods to discretize each one dimensional PIDE as described in d'Halluin et al. (2005, 2004). The jump conditions are also imposed using linear interpolation in the Z direction, which would also have quadratic error for smooth solutions (assuming a finite number of observations). However, application of the jump conditions (4.9) results in a non-smoothness

Parameter	Value
Observation times	1.0, 2.0, 3.0, 4.0, 5.0
T	5.0
Notional	1.0
C_l	0.08
F_l	0.0
C_g	∞
F_g	0.16

TABLE 4: *Cliquet contract details.*

Parameter	Value
S^*	100
σ_J	0.20
r	0.03
λ	0.0

TABLE 5: *Parameters for the constant volatility case without jumps.*

Nodes	Timesteps	Value	Change	Ratio
Running Sum Formulation				
31×13	40	0.174467312		
62×25	80	0.174230223	.00023709	
124×49	160	0.174099404	.00013082	1.8
248×97	320	0.174066778	.00003262	4.0
496×193	640	0.174060717	.00000606	5.4
Average Formulation				
31×13	40	0.174807390		
62×25	80	0.174368430	.00043896	
124×49	160	0.174207223	.00016121	2.7
248×97	320	0.174110595	.00009663	1.7
496×193	640	0.174089486	.00002111	4.6

TABLE 6: *Value of a cliquet option using the running sum and average formulations. Contract details are provided in Table 4. Parameters are given in Table 5. Nodes refers to the number of nodes in the S and Z directions respectively. A similarity reduction is used, so no grid is needed in the P direction. At each refinement level, new nodes are inserted between each pair of grid nodes on the coarser grid, a new node is added between $S = 0$ and the first coarser grid node, and the timestep size is halved. Change refers to the change in numerical value from one level of refinement to the next. Ratio refers to the ratio of changes between successive refinements. An asymptotic ratio of four indicates quadratic convergence.*

of the solution in the Z direction, due to the local caps and floors. This non-smoothness can be expected to cause the convergence rate to be somewhat erratic. We can see in Table 6 that the ratio of changes departs somewhat from the ideal asymptotic value of four which would be observed for exact quadratic convergence.

6.2 Effect of Grid and Interpolation

Using the local volatility surface shown in Figure 1, obtained using equation (3.14), and the contract outlined in Table 4, a series of convergence tests was carried out. These are shown in Table 7. In this case no similarity reduction is possible since the volatility surface is a general function of S and t and does not satisfy equation (5.1). As before, a series of refined grids was constructed where on each refinement the timestep size was halved, new nodes were inserted between each coarse grid node, and a new node was inserted in the S grid in $(0, S_{\min})$.

The results in Table 7 indicate that using a Cartesian product grid (*Repeated Grid*) that uses

the same node spacing in the (S, P) directions results in very poor convergence even though there is no interpolation error in the (S, P) directions when applying the jump conditions. Clearly, the use of the scaled grid/discretization method satisfying Conditions 5.2 is very effective. Hence, we will use these methods in the following.

Nodes	Timesteps	Scaled Grids (diagonal interpolation)	Scaled Grids (xy interpolation)	Repeated Grid
$35 \times 35 \times 13$	40	.167847	.169728	.148230
$70 \times 70 \times 25$	80	.167229	.167837	.159672
$140 \times 140 \times 49$	160	.167046	.167211	.164720

TABLE 7: *Value of a cliquet option with a volatility surface (Figure 1 and equation (3.14)). Contract details are provided in Table 4. Nodes refers to the number of nodes in the S, P, and Z directions respectively. Scaled grids refers to the S-grid construction method (5.5), shown in Figure 4, and satisfying Conditions 5.2. Diagonal interpolation refers to interpolation method (5.7), shown in Figure 5(b), whereas xy interpolation refers to the interpolation method (5.6), shown in Figure 5(a). Repeated grid refers to a Cartesian product grid with the same node spacing in the S and P directions (Figure 3).*

6.3 Effect of Boundary at $S = S_{max}$

Table 8 shows the effect of choosing different values of S_{max}. The results were obtained using the volatility surface in Figure 1, with grids/discretization satisfying Conditions 5.2. The initial coarse grid with $35 \times 35 \times 13$ nodes used $S^j_{max} = 18 \times P_j$. The initial grid with $31 \times 31 \times 13$ nodes used $S^j_{max} = 4 \times P_j$. As we can see, there is no effect on the solution (to six figures) of imposing the boundary condition $V_{SS} = 0$ at these values of S_{max}. Note that this is because we use both scaled grids (as shown in Figure 4), and the similarity extrapolant (5.8) where necessary to impose jump conditions. In the following, we use $S^j_{max} = 18 \times P_j$.

Nodes	Timesteps	Scaled Grids	Nodes	Timesteps	Scaled Grids
$35 \times 35 \times 13$	40	.167847	$31 \times 31 \times 13$	40	.167847
$70 \times 70 \times 25$	80	.167229	$62 \times 62 \times 25$	40	.167229
$140 \times 140 \times 49$	160	.167046	$124 \times 124 \times 25$	160	.167046

TABLE 8: *Value of a cliquet option with a volatility surface (Figure 1 and equation (3.14)). Contract details are provided in Table 4. Nodes refers to the number of nodes in the S, P, and Z directions respectively. Scaled grids refers to the S-grid construction method (5.5), shown in Figure 4. Diagonal interpolation refers to interpolation method (5.7), shown in Figure 5(b). The base grid $35 \times 35 \times 13$ used $S^j_{max} = 18 \times P_j$. The reduced base grid $31 \times 31 \times 13$ used $S^j_{max} = 4 \times P_j$.*

7 Numerical Tests: Modelling Assumptions

7.1 Volatility Surface Based on Analytic Expression

There is increasing evidence that jump processes provide reasonable explanations of volatility smiles and skews. If the real stock price process is a jump diffusion, then the common approach of fitting a volatility surface to vanilla option prices can, as we will demonstrate, result in a significant error when pricing exotic options such as cliquet contracts. However, as we shall see, it is standard practice in industry to use various methods to correct for these deficiencies. These corrections do appear to reduce the pricing errors dramatically, but only near $S = S^*$, the initial asset price.

We assume that the synthetic market dynamics are given by the parameters in Table 1 and the contract details are contained in Table 4. We use the local volatility surface constructed using equation (3.14), as described in Section 3.2. The surface is shown in Figure 1.

The resulting value of a cliquet option is given in Table 9. At each refinement level, new nodes are inserted between each two coarse grid nodes, and the timestep is halved. As we increase the refinement level, the solution will converge to the correct value. The column "Vol. Surf." of this table provides corresponding results if we value this same option without using a jump diffusion model but instead using a volatility surface which was fit to our synthetic vanilla market data (arising from a jump model). Note that these are quite different from the correct values, at least in our synthetic market. This is also illustrated in Figure 6.

Refinement Level	Jump Diffusion Const. Vol.	Vol. Surf. (Fig 1)	Vol. Surf. Sim. Red.	Vol. Surf. Sim. Red. Rebased t	Const. Vol. $\sigma = .2359$	Const. Vol. $\sigma = .3167$
0	.177303	.167847	.170017	.178142	.163874	.160253
1	.177515	.167229	.169491	.177942	.163415	.159533
2	.177540	.167046	.169358	.177909	.163259	.159339

TABLE 9: *Value of a cliquet option under jump diffusion with constant volatility (parameters are given in Table 1) and various other volatility models. The volatility surface is computed using the method described in Section 3.2, using equation (3.14), ans shown in Figure 1. Contract details are provided in Table 4. Volatility surface shown in Figure 1, computed using equation (3.14). Vol. Surf. refers to a volatility surface model with no jumps. Vol. Surf. Sim. Red. refers to the assumption of equation (5.1), and no jumps. Vol. Surf. Sim. Red. Rebased t refers to assumption (7.5), and no jumps. Const. Vol. models have no jumps (values of σ are provided in Table 1). At each refinement level, new nodes are inserted between each coarse grid node, and the timestep is halved.*

Figure 6 compares the true synthetic market price ($t^* = 0$) compared with the local volatility approach. This Figure shows the value of $V(S, P = S^*, Z = 0, t = 0)$. Note that the minimum value of this contract, independent of the underlying model and initial asset price, is about .1246, so that the error, relative to this minimum value, is quite large near the current price $S^* = 100$.

However, practitioners are well aware of the problems with using a calibrated local volatility surface to price exotics. If we calibrate a local volatility surface at $t = t^*$ with $S^* = P$, and subsequently assume that equation (5.1) holds, then a modelling decision has been made to assume that the volatility surface is a function of S/P. We can think of this as assuming that the observed

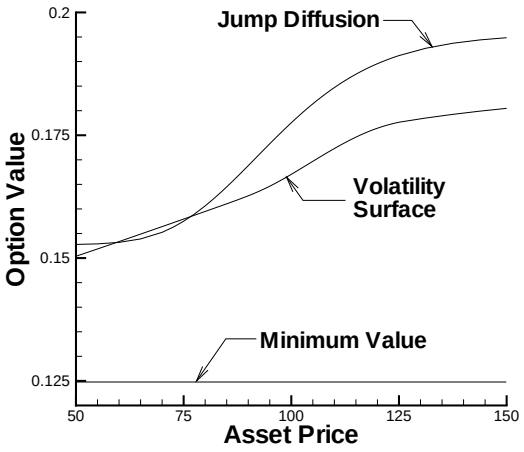


FIGURE 6: *Cliquet option value. Comparison of jump diffusion model with constant volatility (parameters are given in Table 1) with a volatility surface (Figure 1). The surface is computed using the method described in Section 3.2.*

volatility skew will always be "centered" in some sense around the current asset level. This can be justified on the basis of Lemma 3.2 in Section 3. If we calibrate a local volatility model (assuming process (3.3)) to prices generated where the process actually follows (3.1), then in the case of $\sigma_J = const.$, Lemma 3.2 states

$$\sigma_L(\rho S^*, t^*; \rho S, t) = \sigma_L(S^*, t^*; S, t), \tag{7.1}$$

where σ_L is the local volatility, determined by calibration as described in Section 3. If we let $S^* = P$ in equation (7.1), then

$$\begin{aligned} \sigma_L(\rho P, t^*; \rho S, t) &= \sigma_L(P, t^*; S, t) \\ &= \sigma_L\left(S^*, t^*; \frac{S}{P} \times S^*, t\right). \end{aligned} \tag{7.2}$$

Therefore, if $\sigma(S, t; P) = \sigma_L(P, t^*; S, t)$, then equation (5.1) holds.

Another common assumption made by practitioners is to rebase the time for the evaluation of the volatility surface in order to fix the forward starting volatility skew dynamics. To explain, if we calibrate the surface initially at time t^*, then we will typically observe a heavy skew in the implied volatilities of short dated options maturing near t^*. For longer dated options, expiring at $t_i \gg t^*$, the volatility surface is much flatter. When hedging cliquet positions, we may want to take static positions in market traded options at observation dates in order to reduce model risk. If we re-calibrate the volatility surface at time t_i, usually we will find that the new surface now has a heavy skew for options maturing close to t_i, which are now short dated options. If $t_{k-1} \le t \le t_k$, where t_k are observation dates, then it can be postulated that

$$\sigma(S, t; P) = \sigma_L(P, t_{k-1}; S, t - t_{k-1}). \tag{7.3}$$

In our case since $\sigma_J = const.$, then

$$\sigma_L(P, t_{k-1}; S, t - t_{k-1}) = \sigma_L(P, t^*; S, t - t_{k-1}). \tag{7.4}$$

Combining assumptions (5.1), (7.1), and (7.4) together gives

$$\begin{aligned} \sigma(S, t; P) &= \sigma(\rho S, t; \rho P) \\ &= \sigma_L(\rho P, t^*; \rho S, t - t_{k-1}) \\ &= \sigma_L\left(S^*, t^*; \frac{S}{P} \times S^*, t - t_{k-1}\right) \end{aligned} \tag{7.5}$$

where σ_L is the volatility surface calibrated to prices at t_{k-1}, and we have assumed that the calibration is carried out at stock price $S^* = P$ at time t_{k-1}. Again, these modelling assumptions are often used by practitioners to mitigate the skew of the volatility surface and its flattening out, as one looks farther ahead in time.

Values for the cliquet option under assumption (7.1) or (7.5) are also given in Table 9 (columns headed "Vol. Surf. Sim. Red." and "Vol. Surf. Sim. Red. Rebased t"). In the latter case, the agreement between the volatility surface and the true synthetic market price is excellent. Note that either of the assumptions (7.1) or (7.5) allow us to use the similarity reduction method described in Section 5.2.

Finally, we also value this cliquet contract using constant volatility models (no jumps). The resulting values are poor approximations (see Table 9). The volatility values are obtained by calibration to a single at-the-money call (priced in our synthetic jump diffusion market) at two different maturities.

Although Table 9 suggests that the common practitioner adjustments to compensate for the deficiencies of a calibrated local volatility surface perform very well, this is a bit misleading. Figure 7 gives a plot of a comparison of the synthetic market price, and the prices which result from the local volatility with the additional assumptions (7.1) and (7.5). We can see that, under the similarity reduction, rebased time approximation (assumption (7.5)), the price agreement is only good near $S = S^*$, and deviates substantially as we move away from $S^* = 100$.

This problem can be seen more clearly by plotting the option delta. Figure 8 shows the option delta for the case of the jump diffusion model and the volatility surface model constructed using equation (3.14), and shown in Figure 1. We use the similarity reduction and rebased time assumption (7.5). Even though the values for the cliquet option are comparable close to $S^* = 100$ (see Figure 7), the deltas are significantly different.[3]

7.2 Volatility Surface Obtained by Calibration

In Table 10, we show the cliquet prices computed using the surface in Figure 2. We remind the reader that this surface was computed using the method in Section 3.3, which calibrates to vanilla prices using a least squares fit to a spline representation of the surface (Coleman et al., 1999).

Once again we see that the fitted local volatility surface cliquet prices are considerably in error, compared to the exact synthetic market price. However, the use of the rebased time and forced

[3]Note that in the context of a jump diffusion model, the hedging portfolio should include additional options to minimize the jump risk (Andersen and Andreasen, 2000).

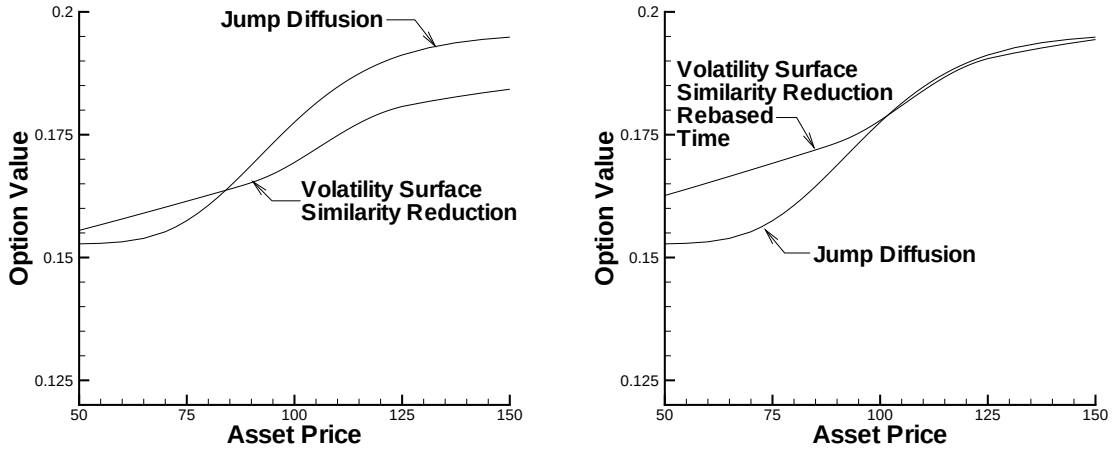


FIGURE 7: *Cliquet option value. Comparison of the jump diffusion model with constant volatility (parameters are given in Table 1) and the volatility surface approximation. Left panel: volatility surface with a similarity reduction (equations (7.1). Right panel: volatility surface with a similarity reduction, rebased time (equations (7.1) and (7.5)). The volatility surface is computed using the method described in Section 3.2, using equation (3.14). The surface is shown in Figure 1.*

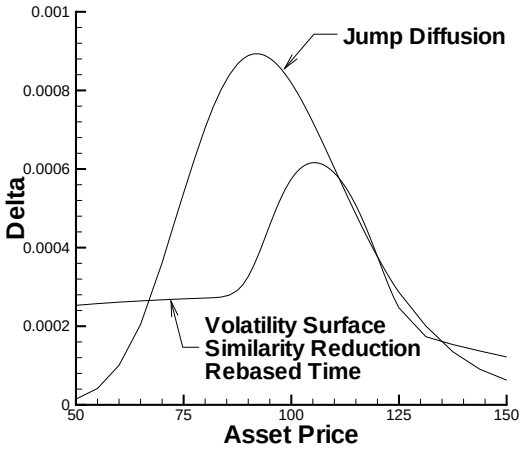


FIGURE 8: *Cliquet option delta. Comparison of jump diffusion model with constant volatility (parameters are given in Table 1) with a calibrated volatility surface with a similarity reduction (equations (7.1) and (7.5)). The volatility surface is computed with the method described in Section 3.2, and using equation (3.14). The surface is shown in Figure 1.*

homogenization of the volatility surface (as in equation (7.5)) considerably reduces the error. The error is, naturally, not as small as is obtained using the expression (3.14), as shown in Table 9.

Refinement Level	Jump Diffusion Const. Vol.	Vol. Surf. (Fig 2)	Vol. Surf. Sim. Red.	Vol. Surf. Sim. Red. Rebased t
0	.177303	.169180	.172563	.181226
1	.177515	.168433	.171966	.180802
2	.177540	.168193	.171866	.180569

TABLE 10: *Value of a cliquet option under jump diffusion with constant volatility (parameters are given in Table 1) and various other volatility models. The volatility surface was computed using a spline fit to the European prices (Coleman et al., 1999), as described in Section 3.3. The surface is shown in Figure 2. Contract details are provided in Table 4. Vol. Surf. refers to a volatility surface model with no jumps. Vol. Surf. Sim. Red. refers to the assumption of equation (5.1), and no jumps. Vol. Surf. Sim. Red. Rebased t refers to assumption (7.5), and no jumps. Const. Vol. models have no jumps (values of σ are provided in Table 1). At each refinement level, new nodes are inserted between each coarse grid node, and the timestep is halved. Compare with Table 9.*

Figure 9 verifies that the use of the calibrated local volatility model (Figure 2), with the additional assumptions (7.1) and (7.5), yields good agreement with the synthetic market price near $S = S^*$, but poor agreement elsewhere.

7.3 Uncertain Volatility

As pointed out by Wilmott (2002), valuing cliquets with extreme values of constant volatility does not really capture the risk of mis-specification of volatility. Wilmott suggests using an uncertain volatility model to quantify the effect of volatility risk. Table 11 provides the best-worst case prices of a cliquet option in an uncertain volatility model, where $\sigma_{\min}$ and $\sigma_{\max}$ are taken from Table 1. Table 11 reveals a large spread between the two cases.

Refinement Level	Worst Case Long	Best Case Long
0	.150548	.175657
1	.149929	.175183
2	.149745	.175049

TABLE 11: *Value of a cliquet option using an uncertain volatility model (no jumps) with $r = .05$, $\sigma_{\min} = .2359$, and $\sigma_{\max} = .3167$. Contract details are provided in Table 4.*

Figure 10(a) shows the jump diffusion price and the best-worst case uncertain volatility prices. Surprisingly, the uncertain volatility best case value is quite close to the jump diffusion value, but we suspect that this is will not be true in general. Note that the best case price is not always above the jump diffusion price. This is because the best case price is guaranteed to be above the constant volatility price only for pure diffusion models. This is illustrated in Figure 10(b).

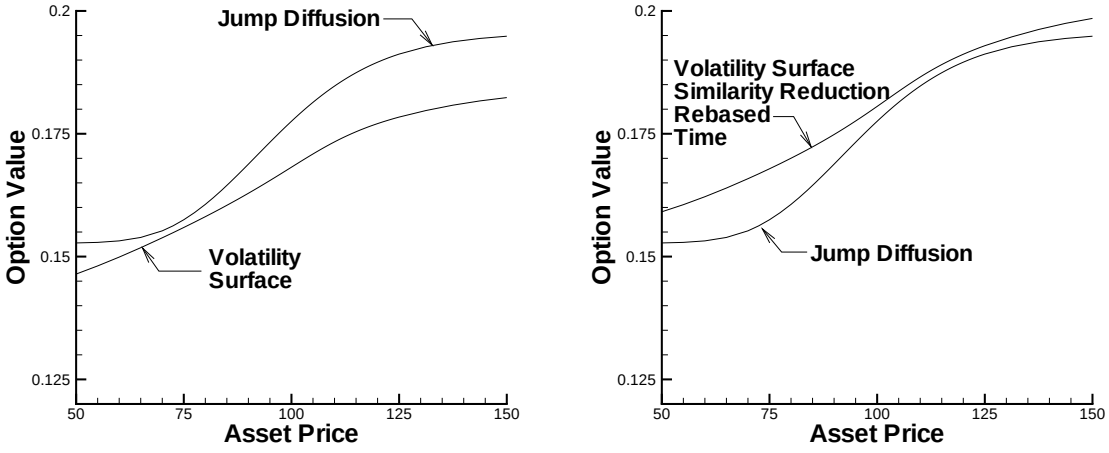
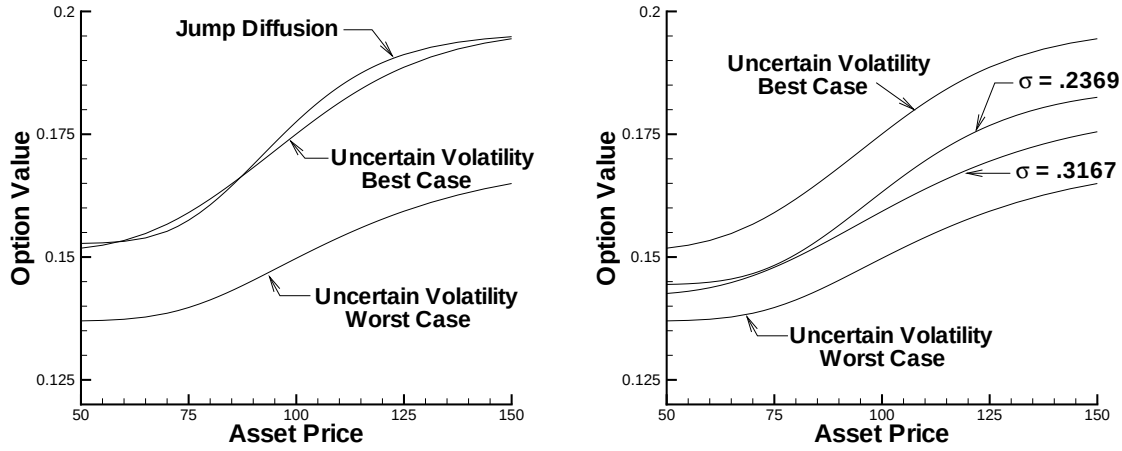


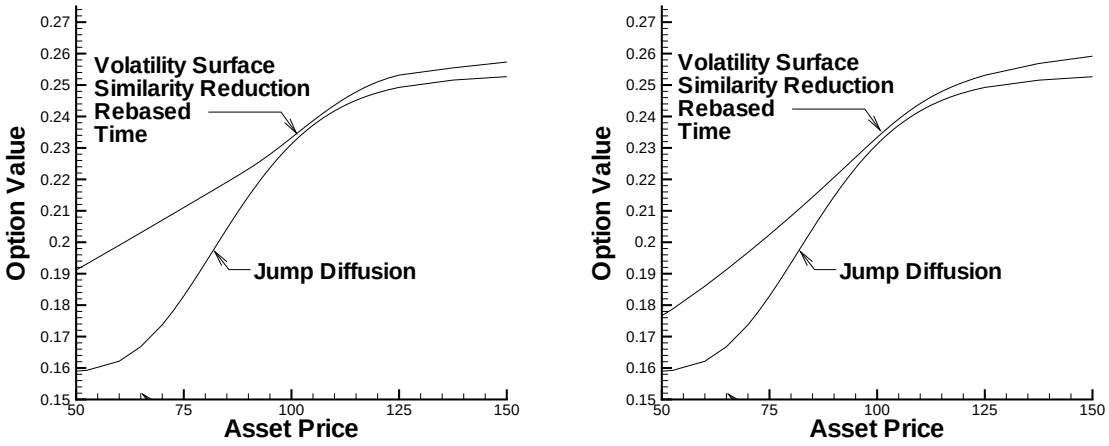
FIGURE 9: *Cliquet option value. Comparison of jump diffusion model with constant volatility (parameters are given in Table 1). The volatility surface is computed using the spline fit procedure described in Coleman et al. (1999). Left panel: pure volatility surface. Right panel: volatility surface with a similarity reduction, rebased time (equations (7.1) and (7.5)). Compare with Figure 7. Note that the local volatility in this case was obtained using the method in Section 3.3, and the surface is given in Figure 2.*



(a) *Cliquet option value. Comparison of jump diffusion model (constant volatility, parameters in Table 1) with an uncertain volatility model,* $\sigma_{\min} = .2359, \sigma_{\max} = .3167$.

(b) *Cliquet option value. Comparison of constant volatility model (no jumps) with an uncertain volatility model,* $\sigma_{\min} = .2359, \sigma_{\max} = .3167$.

FIGURE 10: *Uncertain volatility examples.*



(a) *Volatility surface computed using equation (3.14) as shown in Figure 1.*

(b) *Volatility surface computed using the method in Coleman et al. (1999), as shown in Figure 2.*

FIGURE 11: *Reverse cliquet option value ($S^* = 100$, $t = 0$). Comparison of jump diffusion model with constant volatility (parameters are given in Table 1), with a volatility surface, similarity reduction model, rebased time, as in equation (7.5). Contract details are provided in Table 12.*

7.4 Reverse Cliquet Example

As a final example, we consider a reverse cliquet, with the payoff given in equation (4.3). The contract details are given in Table 12.

Parameter	Value
S^*	100
Observation times	1.0, 2.0, 3.0, 4.0, 5.0
T	5.0
Notional	1
$\hat{R}$	.50
F_l	-.15
F_g	.10

TABLE 12: *Reverse cliquet contract details.*

Figure 11 compares the jump diffusion results for the reverse cliquet, and the volatility surfaces shown in Figure 1 and Figure 2, with the similarity reduction and rebased time modifications (7.5). As with the regular cliquet, there is good agreement between the synthetic market price and the local volatility model, as long as $S \sim S^*$. However, there are significant differences in the shape of the two curves. This indicates that the deltas computed with the volatility surface will be quite different from the deltas computed with the jump diffusion model.

8 Conclusions

Recent turmoil in financial markets has heightened investor awareness of the effect of volatility on portfolios. Cliquet options have become popular insurance against market volatility for large pension plans as well as retail investors. It is therefore of interest to have effective techniques for valuing and hedging these instruments.

The discretely observed cliquet valuation problem reduces to solving a set of one dimensional PDEs embedded in a two or three dimensional space. These one dimensional problems exchange information through jump conditions at each sampling date. With respect to numerical issues, our main results are:

- Unlike the case for discretely observed Asian options, the running sum formulation seems superior to the running average return formulation.
- The type of grid used and interpolation method employed for enforcement of the jump conditions at observation dates has a very large impact on the convergence of the solution. In particular, we recommend using a special scaling method for constructing each one dimensional grid, coupled with diagonal interpolation, and an extrapolation method for determining missing data at the extremes of the grid.

From a practical point of view, we also observe that cliquet options are sensitive to the volatility model assumed. If constant volatilities are used, the value of the cliquet option is insensitive to an extreme range of volatilities. On the other hand, if an uncertain volatility model is used, there is a large spread between best and worst case values.

Recently, jump diffusion models have been touted as better models of market dynamics than the commonly used volatility surface model. If we assume a synthetic market which is driven by a jump diffusion process, and then calibrate a volatility surface model (no jumps) to vanilla option prices generated in the synthetic market, there is a large discrepancy between the value obtained for a cliquet option using the calibrated volatility surface compared to the true value found using the jump diffusion model.

The limitations of volatility surfaces for modelling forward start type options are well known. Practitioners attempt to correct for these problems by

- forcing the local volatility function to homogeneous of degree zero in current price and price at last reset; and
- rolling the local volatility surface forward, i.e. using only the small time part of the surface, to avoid the flattening of the skew.

If the usual local volatility is modified by these two corrections, the price of the cliquet at the initial asset price is in very close agreement with the synthetic market price. However, if we move away from the initial asset price, the agreement deteriorates. The deltas from the corrected local volatility approach are quite different from the synthetic market deltas.

This result suggests that use of calibrated volatility surface models for cliquet options should be viewed with suspicion. If the corrected local volatility function is used, the prices are reasonably accurate, but only at a single point. An uncertain volatility model, as suggested in Wilmott (2002),

which generates a large spread between best and worst cases, at least signals to the hedger the real volatility risk involved in writing these options.

References

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

Avellaneda, M., A. Levy, and A. Parás (1995). Pricing and hedging derivative securities in markets with uncertain volatilities. *Applied Mathematical Finance* 2, 73–88.

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

Cont, R. and P. Tankov (2004). *Financial Modelling with Jump Processes*. Chapman & Hall/CRC.

d'Halluin, Y., P. Forsyth, and G. Labahn (2004). A penalty method for American options with jump diffusion processes. *Numerische Mathematik* 97, 321–352.

d'Halluin, Y., P. A. Forsyth, and K. R. Vetzal (2005). Robust numerical methods for contingent claims under jump diffusion processes. *IMA Journal of Numerical Analysis* 25, 87–112.

Dupire, B. (1994). Pricing with a smile. *Risk* 7 (January), 18–20.

Forsyth, P. A. and K. R. Vetzal (2001). Implicit solution of uncertain volatility/transaction cost option pricing models with discretely observed barriers. *Applied Numerical Mathematics* 36, 427–445.

Forsyth, P. A., K. R. Vetzal, and R. Zvan (2002). Convergence of lattice and PDE methods for valuing path dependent options using interpolation. *Review of Derivatives Research* 5, 273–314.

Garroni, M. G. and J. L. Menaldi (1992). *Green Functions for Second Order Parabolic Integro-differential Problems*. Number 275 in Pitman Research Notes in Mathematics. Longman Scientific and Technical, Harlow, Essex, UK.

Hirsa, A., G. Courtadon, and D. B. Madan (2003). The effect of model risk on the valuation of barrier options. *The Journal of Risk Finance* 4, 47–55.

Kangro, R. and R. Nicolaides (2000). Far field boundary conditions for Black-Scholes equations. *SIAM Journal on Numerical Analysis* 38, 1357–1368.

Lyons, T. (1995). Uncertain volatility and the risk free synthesis of derivatives. *Applied Mathematical Finance* 2, 117–133.

Merton, R. C. (1973). Theory of rational option pricing. *Bell Journal of Economics and Management Science* 4, 141–183.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics* 3, 125–144.

Pooley, D. M., P. A. Forsyth, and K. R. Vetzal (2003). Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis* 23, 241–267.

Rannacher, R. (1984). Finite element solution of diffusion problems with irregular data. *Numerische Mathematik* 43, 309–327.

Roach, G. E. (1982). *Green's Functions*. Cambridge University Press.

Schoutens, W., E. Simons, and J. Tistaert (2004). A perfect calibration: Now what? *Wilmott Magazine* March, 66–78.

Tavella, D. and C. Randall (2000). *Pricing Financial Instruments: The Finite Difference Method*. Wiley.

Vetzal, K. R. and P. A. Forsyth (1999). Discrete Parisian and delayed barrier options: A general numerical approach. *Advances in Futures and Options Research* 10, 1–16.

Walliser, J. (2003). Retirement guarantees in mandatory defined contribution systems. In O. S. Mitchell and K. Smetters (Eds.), *The Pension Challenge: Risk Transfers and Retirement Income Security*, Chapter 11, pp. 238–250. Oxford University Press.

Wilmott, P. (1998). *Derivatives: The Theory and Practice of Financial Engineering*. Wiley.

Wilmott, P. (2002). Cliquet options and volatility models. *Wilmott Magazine* December, 78–83.

Windcliff, H., P. A. Forsyth, and K. R. Vetzal. Pricing methods and hedging strategies for volatility derivatives. Journal of Banking and Finance, forthcoming.

Windcliff, H., P. A. Forsyth, and K. R. Vetzal (2001). Shout options: A framework for pricing contracts which can be modified by the investor. *Journal of Computational and Applied Mathematics* 134, 213–241.

Windcliff, H., P. A. Forsyth, and K. R. Vetzal (2004). Analysis of the stability of the linear boundary condition for the Black-Scholes equation. *Journal of Computational Finance* 8 (Fall), 65–92.

Zvan, R., P. A. Forsyth, and K. R. Vetzal (1999). Discrete Asian barrier options. *Journal of Computational Finance* 3 (Fall), 41–67.