

# *Pricing European and Discretely Monitored Exotic Options under the Lévy Process Framework*

**Dale Olivier Roberts**  
**Alexander Novikov**

In this article we consider both European and discretely monitored exotic options (Bermudan and discrete barrier) in a market where the underlying asset follows a geometric Lévy process. First, we briefly introduce this extended framework. Then, using the variance gamma model, we show how to price European options and demonstrate the application of the recursive quadrature method to Bermudan and discrete barrier options.

## ■ Introduction

It is well known that the classic Black–Scholes framework cannot capture a number of financial market phenomena such as the leptokurtic property found in empirical distributions of asset returns. A number of new models have been proposed, such as stochastic volatility which incorporates a random volatility and generalisations of the classic framework whereby the price process contains a jump component (i.e., the price follows a Lévy process).

First, we demonstrate how to price European options when the stock price follows the variance gamma (VG) process and then present a functional programming implementation of the quadrature method for discretely monitored options where the stock price is modelled by geometric Brownian motion.

## ■ The Lévy Process Price Model

### □ Lévy Processes

#### **Definition**

A stochastic process  $X_t$  on  $(\Omega, \mathcal{F}, P)$  such that  $X_0 = 0$  is called a Lévy process if it has the following properties.

- **Stationary increments:**  
The distribution (or law) of the increment  $X_{t+\Delta} - X_t$  is independent of the time  $t$ .
- **Independent increments:**  
The increments of the process  $X_{t_1}, X_{t_2} - X_{t_1}, \dots, X_{t_k} - X_{t_{k-1}}$  are independent for all times  $0 \leq t_1 < t_2 < \dots < t_k$ .
- **Stochastically continuous paths:**  
For all  $\epsilon > 0$ , we have  $\lim_{\Delta \rightarrow 0} \mathbb{P}\{|X_{t+\Delta} - X_t| \geq \epsilon\} = 0$ .

#### **Infinite Divisibility**

The most distinctive property of Lévy processes is that of stationary increments which implies the probability distribution of an increment of length  $\Delta$  is the same as the distribution of an increment of length  $n\Delta$  (the sum of  $n$  increments). This is called infinite divisibility.

This may also be expressed in terms of characteristic functions: consider a probability measure  $\mu$  on  $\mathbb{R}$ , and its characteristic function  $\mathcal{F}\mu(z) = \int_{\mathbb{R}} e^{izx} d\mu$ . The distribution is called infinitely divisible if for any positive integer  $k$ , there exists a probability measure  $\mu_k$  with characteristic function  $\mathcal{F}\mu_k$  such that  $\mathcal{F}\mu = (\mathcal{F}\mu_k)^k$ .

This property places a restriction on the distributions which may be used for the random variables  $X_t$ , but a number of nice distributions with this property exist (e.g., the Student's  $t$ -distribution, the log normal distribution, the gamma distribution, the Poisson distribution, and the VG process).

### □ Stock Price Model

We consider a market which consists of a riskless bond whose price follows the deterministic process  $B_t = \exp(rt)$ , and a nondividend paying stock  $S_t$  with price process

$$S_t = S_0 \exp(L_t), \quad (1)$$

where  $\{L_t\}_{t \geq 0}$  is a Lévy process under an appropriate risk-neutral (martingale) measure  $Q$ . According to the Lévy–Khinchin theorem, the Lévy process  $\{L_t\}_{t \geq 0}$  has the decomposition

$$L_t = \mu t + \sigma W_t + Y_t, \quad (2)$$

where  $W_t$  is a standard Wiener process and  $Y_t$  is a jump Lévy process that is independent of  $W_t$ ;  $\mu$  and  $\sigma$  are parameters.

The choice of the particular Lévy process used determines the uniqueness of this measure. If the measure  $Q$  is not unique, this leads to the notion of an incom-

plete market. It is well known that  $Q$  is unique only for two special cases: (a) there is no jump component  $Y_t$  in (2), or (b) the parameter  $\sigma = 0$  and  $Y_t$  has only a fixed size jump (i.e.,  $Y_t$  is like a Poisson process). The choice of measure  $Q$  is usually provided by use of a utility function.

### Black-Scholes Model

Setting  $L_t$  to be a Wiener process, we find ourselves in the classic Black-Scholes framework where the bond price is as before and the stock price process follows

$$dS_t = \mu_t S_t dt + \sigma S_t dW_t, \quad (3)$$

where  $\mu_t$  is a deterministic function of  $t$ . The stochastic differential equation (SDE) in (3) has a unique solution

$$S_t = S_0 \exp\left(\int_0^t \mu_s ds + t \frac{\sigma^2}{2} + \sigma W_t\right), \quad (4)$$

and under the risk-neutral measure  $Q$  to obtain a martingale we must have  $\mu_t = r_t$ , or simply  $\mu_t = r$  if we assume a constant risk-free rate  $r$ .

### Variance Gamma Model

Although the Black-Scholes model has become the de facto standard in the finance industry, it is well known that the fair prices it produces do not reflect what often occurs in the market for options which are deeply in- or out-of-the-money, as was shown by Rubinstein in 1985 [1], and Madan, Carr, and Chang [2].

The VG process introduces the notion that market information comes at random time intervals. This concept is modelled by a Wiener process with constant drift evaluated at a random time change given by a gamma process which leads to a pure jump process. The VG model has three parameters that allow us to control volatility, kurtosis and skewness and therefore provide a way to calibrate the model to the prices found in the market. Pricing under the VG framework was first proposed by Madan and Seneta in 1990 [3] and was extended in 1991 [4], 1998 [2], and 2003 [5].

Under the VG framework, the log stock price is defined in terms of a Wiener process with drift  $\theta$  and volatility  $\sigma$

$$B(t, \theta, \sigma) = \theta t + \sigma W(t), \quad (5)$$

where the time  $t$  follows a gamma process  $T(t, \nu) \sim \gamma(t, 1, \nu)$  with mean rate 1 per unit of time and variance  $\nu$  which results in the pure jump process that has an infinite number of jumps in any interval of time:

$$X(t, \theta, \sigma, \nu) = B(T(t, \nu), \theta, \sigma), \quad (6)$$

which may be calibrated by three parameters:  $\sigma$ ,  $\theta$ , and  $\nu$ . Under an equivalent martingale measure, the mean rate of return of the stock is the continuously compounded interest rate  $r$ , and the price then evolves as

$$S_t = S_0 \exp(r t + X(t, \theta, \sigma, \nu) + \omega t),$$

where  $\omega = \log(1 - \theta \nu - \frac{1}{2} \sigma^2 \nu) / \nu$  is a compensator to ensure that we have a martingale.

Madan and Seneta [3] showed the characteristic function to be

$$\Phi(u) = \mathbb{E}[\exp(i u X_t)] = \left( \frac{1}{1 - i \theta \nu u + (\sigma^2 \frac{\nu}{2}) u^2} \right)^{\frac{t}{\nu}}, \quad (7)$$

and the density  $b(z)$  for the log price relative to  $z = \log(S_t / S_0)$  to be written in terms of the modified Bessel function of the second kind  $K_n(z)$  as

$$b(z) = \frac{(2 e^{\frac{\theta \nu}{\sigma^2}})}{\nu^{\frac{t}{\nu}} \sqrt{2 \pi} \sigma \Gamma(\frac{t}{\nu})} \left( \frac{x^2}{\theta^2 + \frac{2 \sigma^2}{\nu}} \right)^{\frac{t}{2 \nu} - \frac{1}{4}} K_{\frac{t}{\nu} - \frac{1}{2}} \left( \frac{\sqrt{x^2 (\theta^2 + \frac{2 \sigma^2}{\nu})}}{\sigma^2} \right), \quad (8)$$

where  $x = z - r t - \omega t$ .

## □ Simulating Variance Gamma Price Paths

The random variables of the underlying jump process  $X(t, \theta, \sigma, \nu)$  may be generated by first drawing a random variable from the gamma process for the time parameter  $t$  and then one from the standard normal distribution denoted  $n$ , and then our random variable  $x$  from  $X(t, \theta, \sigma, \nu)$  is  $x = \theta t + \sigma \sqrt{t} n$ .

```

In[1]:= << "Statistics`ContinuousDistributions`";
distGamma = GammaDistribution[ $\frac{tee}{nu}$ , nu];
distNormal = NormalDistribution(0, 1);
X(t_,  $\theta$ _,  $\sigma$ _,  $\nu$ _) := Block[{T, N, tee = t, nu =  $\nu$ ,  $\omega$ },
  T = Random[distGamma]; N = Random[distNormal];  $N \sqrt{T} \sigma + \theta T$ 

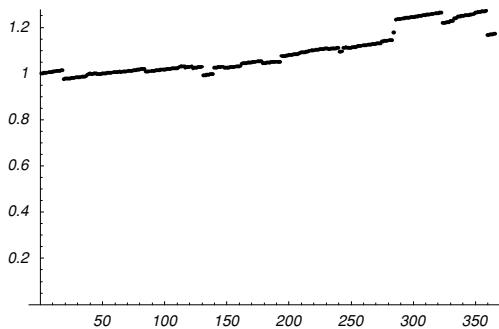
```

A path of a VG process may be simulated by taking a discrete approximation of the time dimension. By plotting a simulated path we can clearly see its random jump behaviour.

```

In[5]:= Block[{n = 365, r = 0.10,  $\sigma$  = 0.12,  $\nu$  = 0.20,  $\theta$  = -0.14,  $\Delta$ ,  $\omega$ },
   $\Delta = 1/n$ ;  $\omega = \text{Log}[1 - \theta \nu - \nu \sigma^2 / 2] / \nu$ ;
  ListPlot[FoldList[#1  $e^{r \Delta + \omega \Delta + \#2}$  &, 1, Array[X[ $\Delta$ ,  $\theta$ ,  $\sigma$ ,  $\nu$ ] &, {n}]],
  PlotRange → {0, Automatic}];

```



It should be noted that even though the price process looks continuous over some regions, it is actually composed of many very small jumps with sudden larger jumps. The stochastic continuity condition of the Lévy process means that for any given time  $t$ , the probability of seeing a jump at  $t$  is zero. The discontinuities of the path must occur at random times; this excludes a process with jumps at predetermined (nonrandom) times.

## ■ Pricing European Options with Variance Gamma

To introduce the notion of pricing under the VG process, we start by pricing a simple European option where the payoff is only a function of the price at expiry. In the case of a European put with strike price  $K$ , we have the payoff

$$f(S_T) = \max(0, K - S_T), \quad (9)$$

where  $T$  is the time of expiry,  $K$  is the strike price of the option, and  $S_T$  is the stock price at time  $T$ . The arbitrage-free price  $V_t$  of the option at time  $t = 0$  is the present value of the expectation, with respect to the risk-neutral martingale measure  $Q$ , of the option payoff

$$V_0 = D E_Q[\max(0, K - S_T)], \quad (10)$$

where  $D = \exp(-rT)$  is the discounting factor.

## □ Monte-Carlo Simulation

VG options may easily be priced using a Monte-Carlo simulation. To derive the expected value for a European option as in (10), we only need to simulate a large number of outcomes for the stock price at expiry and then take the average overall outcomes. The option price is then the present value of this average.

First, we define a function to give a 95% confidence interval for the price given a list of price outcomes.

```
In[6]:= ConfidenceInterval[list_] :=
Module[{σ = StandardDeviation[list], n = Length[list], μ = Mean[list]},
Interval[{μ -  $\frac{1.96 \sigma}{\sqrt{n}}$ , μ +  $\frac{1.96 \sigma}{\sqrt{n}}$ }]}
```

Then we define the option payoff function.

```
In[7]:= Λ[S_, K_] := Max[0, K - S];
```

Here are our parameters for the option and the market.

```
In[8]:= K = 100.00; S0 = 100.00; T = 1.00; r = 0.10; σ = 0.12; ν = 0.20; θ = -0.14;
```

$$\omega = \text{Log}[1 - \theta \nu - \nu \sigma^2 / 2] / \nu;$$

Finally, we proceed with the simulation.

```

In[10]:= mcResult = Module[{outcomes, payoffs, n = 10000},
    outcomes = Array[S0  $e^{r T + \omega T + X[T, \theta, \sigma, \nu]}$  &, {n}];
    payoffs = ( $e^{-r T}$   $\wedge$  [#1, K] &) /@ outcomes;
    ConfidenceInterval[payoffs]] // Timing

Out[10]:= {1.94 Second, Interval[{1.68328, 1.86021}]}

```

This example may be extended to path-dependant options by simulating a discrete approximation of the price process path (as performed earlier), calculating the payoff for each path, taking the average and discounting.

## □ Numerical Integration

For European options, we may alternatively compute the expectation numerically using numerical integration by integrating the payoff of the price process against the density of the normal distribution and the density of the gamma distribution.

```

In[11]:= N[z_] = PDF[NormalDistribution[0, 1], z];
    G[g_] = PDF[GammaDistribution[T /  $\nu$ ,  $\nu$ ], g];

f[g_, z_] :=  $\wedge$ [S0 Exp[ $r T + \omega T + g \theta + \sqrt{g} z \sigma$ ], K];

quadResult =
     $e^{-r T}$  NIntegrate[f[g, z] N[z] G[g], {z, - $\infty$ ,  $\infty$ }, {g, 0,  $\infty$ }] // Timing

Out[14]:= {0.72 Second, 1.85377}

```

This is within the 95% confidence interval found by our Monte-Carlo approach.

```

In[15]:= IntervalMemberQ[Last[mcResult], Last[quadResult]]

Out[15]:= True

```

## ■ Pricing Discretely Monitored Options

Discretely monitored options have payoffs that are triggered by events occurring on discrete times before expiry (e.g., Bermudan options, barrier options, and lookback options). We shall limit ourselves to the cases of Bermudan and discrete barrier options.

### □ Bermudan Options

A Bermudan option is a variation of the American option whereby the early exercise dates are restricted to a finite number throughout the life of the option. This gives the holder of a Bermudan option more rights than holding a European equivalent and less than the American equivalent. Thus from an economic point of view, it should be obvious that the risk-neutral price of a Bermudan is bounded above by the American and below by the European. Although uncom-

mon in equity and foreign exchange markets, it is often found with an underlying fixed income. For example, a Bermudan swaption can be exercised only on the dates when swap payments are exchanged. By letting the number of exercise dates go to infinity, we may approximate the value of an American option by a Bermudan option.

## □ Discrete Barrier Options

A discrete barrier option is monitored at discrete dates before maturity and is either knocked in (comes into existence) or knocked out (is terminated) if the spot price is across the barrier at the time it is monitored. As there is a positive probability of the spot price crossing (or not crossing), barrier options are generally cheaper than ‘vanilla’ equivalents. Analytical pricing formulas are known but assume continuous monitoring of the barrier; however, this may not reflect an accurate price. In the real world, barrier options are typically monitored at discrete times (e.g., at the close of the market). This should not be neglected as the frequency of monitoring has a strong effect on an option’s price.

There are six characteristics of a barrier option that define how it should be priced: the barrier could be above or below the initial value of spot (up or down), the barrier could knock in or knock out the option and the option could be a call or a put. This leads to eight barrier options types.

## □ The Recursive Quadrature Approach

### **Introduction to the Method**

Quadrature is a useful tool for the probability theorist as it allows numerically calculating the expectations in a natural manner without the need to repose the problem in terms of a differential equation or a lattice.

Discretely monitored options may be priced by first identifying the times where a certain condition must hold and then formulating the expectation of the option in a recursive manner such that the expectation of each discrete time step is a function of the expectation of the previous step. This technique easily applies to a range of path-dependant options such as discrete barrier, American, and Bermudan options.

We shall present an implementation of the method proposed by Huang, Subrahmanyam, and Yu [6], Sullivan [7] and Andricopoulos, Widdicks, Duck, and Newton [8] who pose the value of the option at each step  $i$  in terms of the risk-neutral expectation of the step  $i + 1$  which gives

$$V_i = D_i E_Q[V_{i+1}],$$

where  $D_i = \exp(-r(t_{i+1} - t_j))$  is the discounting factor between time steps, and  $V_i$  is the value of the option at step  $i$ . At the terminal step, we have  $V_N = f$  where  $f$  is the payoff of the option. It can be noted that this method allows time steps to be nonequidistant, though in the following implementation we will take time steps of equal length to simplify our exposition.

## □ Application to a Bermudan Put

Before pricing our Bermudan put option we must first set some parameters for the contract, the stock and the market:  $M$  is the time to expiry of the option in years,  $K$  is the strike price,  $r$  is the risk-free rate,  $\sigma$  is the volatility of the underlying stock and  $S$  is the current price of the stock.

*In[16]:*  $M = 0.3333; K = 40.00; r = 0.0488; \sigma = 0.30; S = 40.00;$

We also introduce the parameter  $\lambda$  which represents the number of standard deviations away from the boundary. Modifying both  $\lambda$  and the accuracy goal of the numerical integration allows tuning of the accuracy and speed of this method as needed.

*In[17]:*  $\lambda = 10;$   
**SetOptions[NIntegrate, AccuracyGoal  $\rightarrow$  4];**

We price this option under the Black–Scholes framework, so we define the conditional PDF of the risk-neutral distribution with respect to the previous price  $x$  and the CDF of the standard normal, noting that we transform the prices so that  $y = \log(S_{i+1} / K)$  and  $x = \log(S_i / K)$  where  $S_i$  is the price at time step  $i$ .

*In[19]:*  $uz = (r - \sigma^2 / 2) \Delta; sz = \sigma \sqrt{\Delta};$   
*In[20]:* **<< "Statistics`ContinuousDistributions`";**  
 $\Psi(x\_)=CDF(NormalDistribution(0, 1), x);$   
 $\Phi(y\_ , x\_)=PDF(NormalDistribution(uz + x, sz), y);$

The risk-neutral expectation of the value is broken into two integrals at the implicit boundary  $b$ . In the case of a put option, below the boundary we have the Black–Scholes analytic solution.

*In[23]:*  $d2[x\_ , y\_]= (Log[y / x] - uz) / sz;$   
 $d1[x\_ , y\_]= d2[x, y] - sz;$   
*In[25]:* **belowBoundary[b\_Real, x\_Real] :=**  
 $K e^{-r \Delta} \Psi[d2[K e^x, K e^b]] - K e^x \Psi[d1[K e^x, K e^b]]$

The upper integral takes a function approximation of the previous step (working backwards) and computes the expectation numerically. Using function approximation allows us to not indulge in a recursive calculation at each step.

*In[26]:* **aboveBoundary[func\_, b\_Real, ymax\_Real, x\_Real] :=**  
 $e^{-r \Delta} NIntegrate[func[y] \Phi[y, x], \{y, b, ymax\}]$

Thus the value at each step is the sum of these integrals.

*In[27]:* **putValue[func\_, b\_Real, x\_Real] :=**  
 $\text{belowBoundary}[b, x] + \text{aboveBoundary}[func, b, q, x]$

The difficulty of Bermudan and American options is the implicit or moving boundary; at each step we must numerically identify the price where we are indifferent to holding the option or exercising the option. Again, as finding this point requires a number of iterations of the value function, function approximation simplifies this greatly.



At each step we must find the boundary of the previous calculated step, calculate the expectation, and create a new function approximation to pass along to the next step.

```
In[28]:= valueStep[data_] :=
  Block[{b},
    b = Interpolation[data];
    b = z /. FindRoot[b[z] = K (1 - e-z), {z, 0.0}];
    generateData[b, b]]
```

Our function approximation is created by sampling the value at evenly spaced points within  $\lambda$  standard deviations distance from the boundary.

```
In[29]:= Ys := Range[Log[S/K] - q, Log[S/K] + q, 2 q/N];
generateData[func_, b_] := {#, putValue[func, b, #]} & /@ Ys
```

To find our option value, we now simply step backwards through time to the present day which gives us a function approximation for a range of stock prices. The function takes two arguments: the first is the number of exercise dates and the second is the number of evenly spaced sampling points for each step. Our option value is equal to the value for our current stock price.

```
In[31]:= Off[InterpolatingFunction::"dmval"];
bermudanPut[nDates_, qPoints_] :=
  Block[{v0, b = Log[S/K], Δ, T, N, Y}, T = nDates; N = qPoints;
    Δ = M/T; q = λ σ √Δ; v0 = ({#1, belowBoundary(b, #1)} &)/@ Ys;
    Interpolation[Nest[valueStep, v0, T - 1]][0.0]]
```

```
In[33]:= bermudanPut[16, 32] // Timing
```

```
Out[33]:= {0.92 Second, 2.47801}
```

We can compare these results with the paper by Sullivan [7], where the number of points  $q = 32$ .

Exercise Dates	<i>Mathematica</i>	Sullivan	RMSE
16	2.47801	2.4775	0.0005
64	2.47288	2.4812	0.0083

**Table 1.** Comparison of results to Sullivan with  $q = 32$ .

## □ Application to a Discrete “Down-And-Out” Barrier Call

We shall now apply this quadrature technique to a discrete “down-and-out” barrier call option with parameters for time to maturity  $M$ , strike price  $K$ , risk-free rate  $r$ , stock volatility  $\sigma$ , and current price  $S$ .

```
In[34]:= M = 0.3333; K = 40.00; r = 0.0488; σ = 0.30; S = 40.00;
```

In the case of a discrete barrier option, we now also have a list of stock prices for which the option is knocked out, or in other words rendered useless. Again for simplicity, we restrict ourselves to the case where the times are equally spaced but it should be known that this method works equally well for arbitrary times. We start with one knock-out price, and append our strike price at expiry to the list.

```
In[35]:= B = {35.00}; AppendTo[B, K];
```

We now know how many equally spaced steps are needed to value this option, and we set our number of sample points to be N.

```
In[36]:= T = Length[B]; N = 32;
```

And, as before, we may tune the accuracy and speed as necessary.

```
In[37]:= λ = 10;
SetOptions[NIntegrate, AccuracyGoal → 4];
```

We now transform the boundaries and define the time step  $\Delta$ , and again define  $q$  as the price change from the boundary.

```
In[39]:= B = Log[B/K]; Δ = M/T; q = λ σ √Δ;
```

We define the conditional PDF of the stock price change under the transformation  $y = \log(S_{i+1}/K)$  and  $x = \log(S_i/K)$ .

```
In[40]:= uz = (r - σ² / 2) Δ; sz = σ √Δ;
```

```
In[41]:= << Statistics`ContinuousDistributions`
```

```
In[42]:= Φ[y_, x_] = PDF[NormalDistribution[uz + x, sz], y];
```

Discrete barrier options are somewhat simpler than Bermudan options as we know the location of the boundary and for the down-and-out call below the boundary the option is worth zero. This leaves us with only the upper part of the integral to calculate.

```
In[43]:= callValue[func_, b_Real, x_Real] :=
e-r Δ NIntegrate[func[y] Φ[y, x], {y, b, b + q}];
```

At each step we identify the upper and lower bounds of our price range and then generate a function approximation for the next step, and since we explicitly know the boundary points we no longer need to find them.

```
In[44]:= valueStep[data_, b_] :=
Block[{b},
b = Interpolation[data];
generateData[b, b]]
```

Again, our function approximation is created by sampling the value at evenly spaced points within  $\lambda$  standard deviations distance from the boundary.

```
In[45]:= Ys := Range[Log[S/K] - q, Log[S/K] + q, 2 q / N];
generateData[func_, b_] := {#, callValue[func, b, #]} & /@ Ys
```

To value the option we step through each time step and find the value of the expectation with respect to the previous step, ensuring that below the barrier the option is worth zero.

```
In[47]:= v0 = {#, Max[0, K (Exp[#] - 1)]} & /@ Ys;
          downOutCallResult =
            Interpolation[Fold[valueStep, v0, Reverse[B]]][0.0] // Timing
```

```
Out[48]= {0.13 Second, 3.03237}
```

We may verify this result using a Monte-Carlo simulation.

```
In[49]:= DownOutCallMC[S_, K_, M_, σ_, r_, Bs_List, m_] :=
          Block[{Ps, Ss, s, dt, n, Path, rv, μ, Σ},
            n = Length[Bs] + 1; dt = M / n;
            Ss := S Drop[FoldList[#1 Exp[(r - σ² / 2) dt + σ √dt #2] &,
              1, RandomArray[NormalDistribution[0, 1], n], 1];
            Ps = Table[s = Ss; If[Min[Drop[s, -1] - Bs] < 0.0, 0.0,
              Exp[-r M] Max[Last[s] - K, 0.0]], {m}];
            ConfidenceInterval[Ps]
          ]
```

```
In[50]:= downOutCallInterval =
          DownOutCallMC[S, K, M, σ, r, {35}, 100000] // Timing
```

```
Out[50]= {23.88 Second, Interval[{3.02577, 3.084}]}
```

```
In[51]:= IntervalMemberQ[Last[downOutCallInterval], Last[downOutCallResult]]
```

```
Out[51]= True
```

## ■ Conclusion

In this article we have shown how to quickly price European options under the variance gamma process and have implemented the recursive quadrature technique, a powerful method that is often forgotten in the literature on option pricing and lacking the needed working examples to allow a quick implementation by industry practitioners.

## ■ References

- [1] M. Rubinstein, "Nonparametric Tests of Alternative Option Pricing Models Using All Reported Trades and Quotes on the 30 Most Active CBOE Option Classes from August 23, 1976 through August 31, 1978," *Journal of Finance*, **40**(2), 1985 pp. 455–480.
- [2] D. B. Madan, P. P. Carr, and E. C. Chang, "The Variance Gamma Process and Option Pricing," *European Finance Review*, **2**, 1998 pp. 79–105.
- [3] D. B. Madan and E. Seneta, "The Variance Gamma (V.G.) Model for Share Market Returns," *Journal of Business*, **63**(4), 1990 pp. 511–524.

- [4] D. B. Madan and F. Milne, "Option Pricing with VG Martingale Components," *Mathematical Finance*, **1**(4), 1991 pp. 39–56.
- [5] P. Carr, H. Geman, D. B. Madan, and M. Yor, "Stochastic Volatility for Lévy Processes," *Mathematical Finance*, **13**(3), 2003 pp. 345–382.
- [6] J.-Z. Huang, M. G. Subrahmanyam, and G. G. Yu, "Pricing and Hedging American Options: A Recursive Integration Method," *The Review of Financial Studies*, **9**(1), 1996 pp. 277–300.
- [7] M. A. Sullivan, "Valuing American Put Options Using Gaussian Quadrature," *The Review of Financial Studies*, **13**(1), 2000 pp. 75–94.
- [8] A. D. Andricopoulos, M. Widdicks, P. W. Duck, and D. P. Newton, "Universal Option Valuation Using Quadrature Methods," *Journal of Financial Economics*, **67**(3), 2003 pp. 447–471.

## About the Authors

Dale Olivier Roberts is a Ph.D. student in the School of Mathematics and Statistics at the University of New South Wales, Australia.

Alexander Novikov is a professor of probability in the Department of Mathematical Sciences at the University of Technology, Sydney (UTS), Australia and Senior Researcher at Steklov Mathematical Institute, Moscow.

### Dale Olivier Roberts

*School of Mathematics and Statistics  
University of New South Wales  
Sydney, NSW 2052, Australia  
droberts@maths.unsw.edu.au*

### Alexander Novikov

*Department of Mathematical Sciences  
University of Technology, Sydney  
P. O. Box 123  
Broadway NSW 2007, Australia  
alex.novikov@uts.edu.au*