# Application of the Fast Gauss Transform to Option Pricing

Mark Broadie • Yusaku Yamamoto
*Graduate School of Business, Columbia University, 3022 Broadway, New York, New York, 10027-6902*
*Central Research Laboratory, Hitachi, Ltd., Tokyo, Japan*
*mnb2@columbia.edu • marula@crl.hitachi.co.jp*

In many of the numerical methods for pricing American options based on the dynamic programming approach, the most computationally intensive part can be formulated as the summation of Gaussians. Though this operation usually requires $O(NN')$ work when there are $N'$ summations to compute and the number of terms appearing in each summation is $N$, we can reduce the amount of work to $O(N+N')$ by using a technique called the fast Gauss transform. In this paper, we apply this technique to the multinomial method and the stochastic mesh method, and show by numerical experiments how it can speed up these methods dramatically, both for the Black-Scholes model and Merton's lognormal jump-diffusion model. We also propose extensions of the fast Gauss transform method to models with non-Gaussian densities.
(*Option Pricing; American Options; Fast Gauss Transform; Jump-Diffusion Model*)

## 1. Introduction

Many options traded in the market have American features, and it is therefore optimal to exercise before maturity. The rational price of such options can be calculated as a discounted expectation value under the risk-neutral measure (Duffie 1996) of the payoff under the optimal (adapted) exercise strategy, that is

$$Q_0(S_0) = \sup_{\tau} e^{-r\tau} E_0[h_\tau(S_\tau)], \tag{1}$$

where $S_t$ is the stock price at time $t$, $h_t(S_t)$ is the payoff from exercise at time $t$, and $\tau$ is a stopping time. However, unlike European options, there are no explicit formulas for the option price $Q_0(S_0)$, except for some special cases such as the perpetual American option, and one has to resort to numerical methods for pricing.

Many of the numerical methods for American option pricing use a dynamic programming approach. In this approach, we discretize the time, and starting from the option value at maturity $Q_T(S_T) = h_T(S_T)$, compute the option value at each time step by working backwards. Specifically, the option price at time $t$ and with asset price $S_t$ is computed as the maximum of the immediate exercise value and the continuation value as follows:

$$Q_t(S_t) = \max\{h_t(S_t),\, e^{-r\Delta t} E_t[Q_{t+1}(S_{t+1})]\}. \tag{2}$$

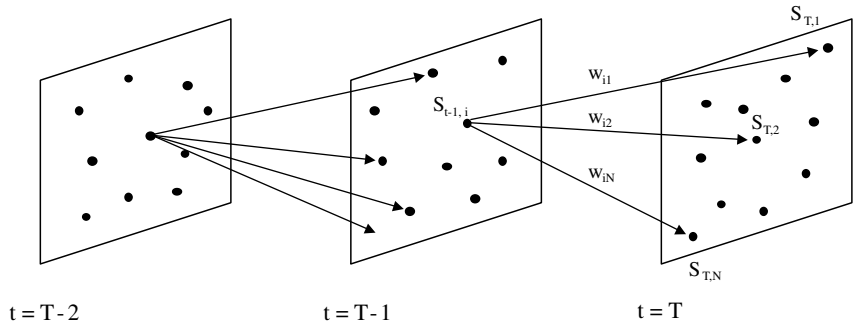
In the actual algorithms, we also discretize the space as $\{S_{t,1}, S_{t,2}, \ldots, S_{t,N'}\}$, as shown in Figure 1, and approximate the continuation value at point $(t, S_{t,i})$ as follows:

$$E[Q_{t+1}(S_{t+1})|S_{t,i}] \cong \sum_{j=1}^{N} w_{ij} Q_{t+1,j}, \quad i = 1, 2, \ldots, N'. \tag{3}$$

Here, $Q_{t+1,j}$ denotes the option value at time $t+1$ and asset price $S_{t+1,j}$, and $w_{ij}$ is the weight of $Q_{t+1,j}$ used in the evaluation of the continuation value at $S_{t,i}$. Some of the numerical procedures that can be cast into this framework are binomial and multinomial methods, explicit finite difference methods, and the stochastic mesh method.

0025-1909/03/4908/1071$05.00
1526-5501 electronic ISSN

**Figure 1 Calculation of the Continuation Value**




Apparently, (3) seems to require $O(NN')$ computation for each time step, and in fact, in the algorithms listed above, most of the computational effort is spent to evaluate the expectation value through (3). However, in some cases it can be shown that the matrix $(w_{ij})$ has a special property that enables much faster evaluation of (3). Such a situation arises, for example, when the underlying assets follow the (multidimensional) geometric Brownian motion process. In this case, as we will show in the following sections, the weight matrix $(w_{ij})$ can be written as a Gaussian function:

$$w_{ij} = c \exp\left\{ \frac{\|\mathbf{x}_i - \mathbf{y}_j\|^2}{\delta} \right\}, \tag{4}$$

where $\mathbf{x}_i$ and $\mathbf{y}_j$ are state variables at time $t$ and $t+1$, respectively, and $c$ and $\delta$ are constants. Then, the sums in the right-hand side of (3) can be evaluated in $O(N+N')$ time using a method called the fast Gauss transform (FGT), instead of $O(NN')$ time needed for direct evaluation. This technique can also be extended to deal with the lognormal jump-diffusion model, and more generally, models for which the weight matrix $(w_{ij})$ has a rapidly converging expansion with respect to the state variables $\mathbf{x}_i$ and $\mathbf{y}_j$. In this paper, we use multinomial type methods and the stochastic mesh method as examples and show how the FGT can improve the efficiency of these methods under the Black-Scholes or lognormal jump-diffusion models. We also give a brief sketch on how our method can be generalized to treat Kou's double-exponential jump-diffusion model (Kou 2002) and stochastic volatility models at the end of this paper.

There are other approaches to the computation of the continuation value. When the state variables are defined on a regular mesh and the weight $w_{ij}$ depends only on the difference $\mathbf{x}_i - \mathbf{y}_j$, as in (4), the computation of the continuation value can be regarded as discrete convolution. Reiner (2000) exploits this fact to compute (3) using the fast Fourier transform (FFT). Van Steenkiste and Foresi (1999) consider a situation where the underlying asset prices follow an affine jump-diffusion process, and using the fact that the conditional characteristic function is given explicitly, propose an FFT method to compute the continuation value. While these methods apply to a wider class of processes, our approach is more efficient when applicable, because it requires only $O(N)$ work when the number of mesh points at each time step is $N$, instead of $O(N \log N)$ work required by the FFT-based methods. Unlike the FFT approach, our method does not require that the state variables be defined on a regular mesh. This may be particularly useful for models that have time-varying parameters and in pricing options with barriers.

Other approaches use the Monte Carlo method to compute the continuation value; a recent comparsion of methods is given in Fu et al. (2001). The stochastic mesh (Broadie and Glasserman 1997) is a convergent method which can estimate both lower and upper bounds on the true price. We show in this paper how the FGT can be used to speed the computation of the continuation value in this method. More recently, Longstaff and Schwartz (2001) propose another Monte Carlo-based approach that approximates the continuation value function by regression. Their method is very similar to the stochastic mesh

method, where regression weights are used instead of likelihood ratio weights for estimating the continuation value. Their method has been successfully applied to many problems and has proven to be especially useful for computing lower bounds for option prices in some high-dimensional problems. For low-dimensional problems (e.g., three assets or less) where standard multinomial methods are applicable, they are typically orders of magnitude faster than simulation methods. Likewise, our method offers significant improvement over standard multinomial implementations.

This paper is organized as follows. In §2, we describe the algorithm of the multinomial methods for the Black-Scholes model and formulate the computation of the continuation value as summations of Gaussians. In §3, we give the basic idea of the FGT and demonstrate how these summations can be calculated in $O(N+N')$ work. This technique is applied to the multinomial method for Merton's log-normal jump-diffusion model and the stochastic mesh method for the multifactor Black-Scholes model in §§4 and 5, respectively. Results of numerical experiments can be found in §6. Section 7 treats extensions of the FGT to non-Gaussian densities. Concluding remarks are given in the final section.

## 2. Multinomial Methods

### 2.1. Construction of the Multinomial Lattice

In this section, we consider the risk-neutralized Black-Scholes model where the asset price $S_t$ follows the geometric Brownian motion process

$$dS_t = rdt + \sigma dW_t, \tag{5}$$

where $W_t$ is a Wiener process. This can also be written in integral form as

$$S_t = S_0 \exp\left\{\left(r - \frac{1}{2}\sigma^2\right)t + \sigma W_t\right\}. \tag{6}$$

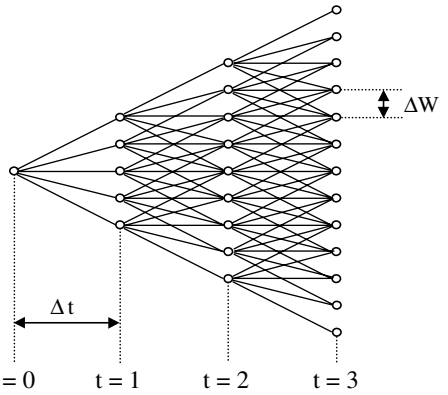
Though we restrict ourselves to a single asset case here, extension of the method to multiasset cases is quite straightforward.

In a multinomial method, we approximate the Wiener process $W_t$ by a discrete random variable $W_{t_i}$ which is defined at discrete times $t_i = i\Delta t$, and takes discrete values $W_j = j\Delta W$. Following Alford and Webber (2001), we require that if $W_{t_i}$ has the value $W_j$ at time $t_i$, its value at time $t_{i+1}$ takes the values in the set $\{W_{j+k} | k = -b, \ldots, b\}$ for some constant $b$. Then, each node of the lattice has $l = 2b+1$ branches and there are $2bi+1$ nodes at time $t_i$, as shown in Figure 2. This is called a multinomial lattice of branching order $l$. We associate each of the branches with a branching probability $p_i$. Let the option value at grid point $(i, j)$ be $Q_{i,j}$. Then, because the approximation $W_{t_i}$ to $W_t$ readily leads to an approximation $S_{t_i}$ to $S_t$, we can compute the continuation value of (3) by

$$E[Q_{t_{i+1}} | S_{i,j}] = \sum_{k=-b}^{b} p_k Q_{i+1, j+k}. \tag{7}$$

To attain a high order of convergence in the multinomial methods, one has to choose the probabilities $p_i$ so that $W_{t_i}$ becomes a good approximation of $W_t$. One of the guidelines for this is given by Heston and Zhou (2000). They show that if the first $q$ moments of $W_{t_i}$ match those of $W_t$, the multinomial method with $M$ time steps has a convergence rate of $O(M^{\frac{q-1}{2}})$ under the condition that the option payoff is $2q$ times differentiable. One can easily match all the odd moments (which are zero for $W_t$) by putting $p_{-k} = p_k$ for $k = 1, \ldots, b$. To match the first $b$ even moments and to

**Figure 2** **A Pentanomial Lattice** ($b = 2$)



ensure that the sum of $p_i$s is 1, one has to solve the following linear equation:

$$\begin{pmatrix} 1 & 2 & 2 & \dots & 2 \\ 0 & 2\cdot\Delta W^2 & 2\cdot(2\Delta W)^2 & \dots & 2\cdot(b\Delta W)^2 \\ 0 & 2\cdot\Delta W^4 & 2\cdot(2\Delta W)^4 & \dots & 2\cdot(b\Delta W)^4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 2\cdot\Delta W^{2b} & 2\cdot(2\Delta W)^{2b} & \dots & 2\cdot(b\Delta W)^{2b} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_b \end{pmatrix} = \begin{pmatrix} 1 \\ \Delta t \\ 3\Delta t^2 \\ \vdots \\ \dfrac{(2b)!}{2^b b!}(\Delta t)^b \end{pmatrix}. \tag{8}$$

Alford and Webber (2001) solve this equation numerically and confirm that the resulting methods for $b = 3, 7, 11, 15$, and 19 have much higher rate of convergence for European options than the binomial methods.

### 2.2. The Continuation Value as the Sum of Gaussians

Here, we consider a multinomial method with a large value of $b$ and show how it can be cast into a form to take advantage of the FGT.

First, let

$$p(x) = \frac{1}{\sqrt{2\pi\Delta t}} \exp\left(-\frac{x^2}{2\Delta t}\right). \tag{9}$$

Then, from the definition of Riemann integral, we have

$$\lim_{\Delta W \to 0} \sum_{k=-\infty}^{+\infty} (k\Delta W)^{2n} p(k\Delta W)\Delta W = \int_{-\infty}^{\infty} x^{2n} p(x)dx = \frac{(2n)!}{2^n n!}(\Delta t)^n. \tag{10}$$

Therefore, for sufficiently small $\Delta W$,

$$\sum_{k=-\infty}^{+\infty} (k\Delta W)^{2n} p(k\Delta W)\Delta W \cong \frac{(2n)!}{2^n n!}(\Delta t)^n. \tag{11}$$

By truncating the infinite sum, we have

$$\sum_{k=-b}^{b} (k\Delta W)^{2n} p(k\Delta W)\Delta W \cong \frac{(2n)!}{2^n n!}(\Delta t)^n. \tag{12}$$

This shows that by putting

$$p_k = p(k\Delta W)\Delta W, \tag{13}$$

Equation (8) is approximately satisfied. Note that the error introduced by approximation (11) is $O(e^{-c/\Delta W})$, because this is an approximation of the integration of an analytical function over the entire real axis by a trapezoidal rule (Sloan and Joe 1994). Also, the error introduced by (12) decreases exponentially as the number of branches increases. Thus, we can expect the $p_k$ given by (13) is a very accurate approximate solution of (8) when $b$ is large.

From (7) and (13), the continuation value can be computed as

$$E[Q_{t_{i+1}}|S_{i,j}] = \frac{\Delta W}{\sqrt{2\pi\Delta t}} \sum_{k=-b}^{b} Q_{i+1,j+k} \cdot \exp\left\{-\frac{1}{2\Delta t}(W_j - W_{j+k})^2\right\}. \tag{14}$$

This has the form of sum of Gaussians and can be computed efficiently by the FGT which we will introduce in the next section.

## 3. The Fast Gauss Transform

### 3.1. The Basic Idea

In this section, we describe the basic idea of the FGT introduced by Greengard and Strain (1991), Strain (1991), and Greengard and Sun (1998), and show how it can compute the sum in the right-hand side of (3) in $O(N + N')$ time. Though we only detail the one-dimensional case here, the algorithm can be extended to higher dimensional cases. For details of the extension, as well as for more comprehensive description of the algorithm including error analysis, consult Greengard and Strain (1991) and Baxter and Roussos (2002).

Suppose that we want to calculate the sums

$$G(x_i) = \sum_{j=1}^{N} q_j \exp\left\{\frac{(x_i - y_j)^2}{\delta}\right\}, \quad i = 1, 2, \ldots, N'. \quad (15)$$

This is called the Gauss transform of $\{q_j\}_{j=1}^N$ with respect to the point sets $\{x_i\}_{i=1}^{N'}$ and $\{y_j\}_{j=1}^N$. Apparently, it needs $O(NN')$ work to evaluate these sums based on the above definition.

The basis of the FGT is the following expansion of the Gaussian in terms of Hermite functions:

$$e^{-(x-y)^2} = \sum_{\alpha=0}^{\infty} \frac{y^\alpha}{\alpha!} h_\alpha(x), \quad (16)$$

where the Hermite function $h_n(x)$ is defined by

$$h_\alpha(x) = (-1)^\alpha \left(\frac{d}{dx}\right)^\alpha e^{-x^2}. \quad (17)$$

It is known that this expansion converges very quickly and truncation at $n = 8$ is sufficient to achieve a relative error of $10^{-8}$ when $|y| < 1/2$.

For the FGT, we use a shifted and scaled version of this expansion, namely,

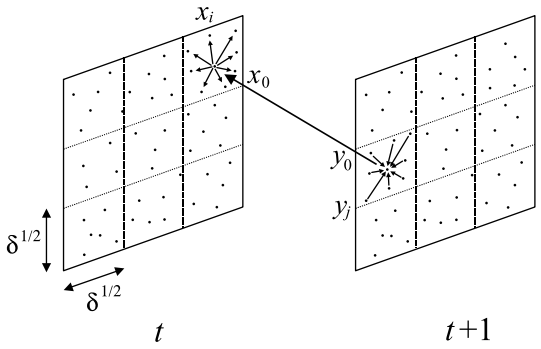
$$e^{-(x_i - y_j)^2/\delta} = \sum_{\alpha=0}^{\infty} \frac{1}{\alpha!} \left(\frac{y_j - y_0}{\sqrt{\delta}}\right)^\alpha h_\alpha\left(\frac{x_i - y_0}{\sqrt{\delta}}\right). \quad (18)$$

The Hermite function appearing in the right-hand side of this expression can further be expanded, and we finally obtain

$$e^{-(x_i - y_j)^2/\delta} = \sum_{\beta=0}^{\infty} \sum_{\alpha=0}^{\infty} \frac{1}{\beta!} \frac{1}{\alpha!} \left(\frac{y_j - y_0}{\sqrt{\delta}}\right)^\alpha \cdot h_{\alpha+\beta}\left(\frac{x_0 - y_0}{\sqrt{\delta}}\right)\left(\frac{x_0 - x_i}{\sqrt{\delta}}\right)^\beta. \quad (19)$$

We first consider a special case where all the target points $\{x_i\}$ are in a box (or an interval in the one-dimensional case) with center $x_0$ and side length $\sqrt{\delta}$ and all the source points $\{y_j\}$ are in another box with center $y_0$ and side length $\sqrt{\delta}$, as shown in Figure 3. Then, the above expansion converges quickly by truncating the sums in (19) at some integer $\alpha_{\max}$ and

**Figure 3 The Source and Target Boxes**




the sum can be written as

$$\begin{aligned} G(x_i) &\cong \sum_{j=1}^{N} q_j \sum_{\beta=0}^{\alpha_{\max}} \sum_{\alpha=0}^{\alpha_{\max}} \frac{1}{\beta!} \frac{1}{\alpha!} \left(\frac{y_j - y_0}{\sqrt{\delta}}\right)^\alpha \\ &\quad \cdot h_{\alpha+\beta}\left(\frac{x_0 - y_0}{\sqrt{\delta}}\right)\left(\frac{x_0 - x_i}{\sqrt{\delta}}\right)^\beta \\ &= \sum_{\beta=0}^{\alpha_{\max}} \frac{1}{\beta!} \left(\frac{x_0 - x_i}{\sqrt{\delta}}\right)^\beta \\ &\quad \cdot \left\{\sum_{\alpha=0}^{\alpha_{\max}} h_{\alpha+\beta}\left(\frac{x_0 - y_0}{\sqrt{\delta}}\right)\left\{\frac{1}{\alpha!} \sum_{j=1}^{N} q_j \left(\frac{y_j - y_0}{\sqrt{\delta}}\right)^\alpha\right\}\right\}. \end{aligned} \quad (20)$$

This expression shows that the computation of $G(x_i)$ can be divided into three steps:

*Step 1.* Compute

$$A_\alpha \equiv \frac{1}{\alpha!} \sum_{j=1}^{N} q_j \left(\frac{y_j - y_0}{\sqrt{\delta}}\right)^\alpha \quad \text{for } \alpha = 0, \ldots, \alpha_{\max}.$$

*Step 2.* Compute

$$B_\beta \equiv \sum_{\alpha=0}^{\alpha_{\max}} A_\alpha h_{\alpha+\beta}\left(\frac{x_0 - y_0}{\sqrt{\delta}}\right) \quad \text{for } \beta = 0, \ldots, \alpha_{\max}.$$

*Step 3.* Compute

$$G(x_i) = \sum_{\beta=0}^{\alpha_{\max}} B_\beta \frac{1}{\beta!} \left(\frac{x_0 - x_i}{\sqrt{\delta}}\right)^\beta \quad \text{for } i = 1, \ldots, N'.$$

When $\alpha_{\max}$ is fixed, Steps 1 and 3 require $O(N)$ and $O(N')$ computational effort, respectively, while Step 2 can be done in a constant time that does not depend either on $N$ or $N'$.

In the general case, we divide the space into boxes of side length $\sqrt{\delta}$ and apply the above method to each of the possible pairs of a source box and a target box. Let $J$ and $I$ denote the source box and the target box, respectively, and $y_J$ and $x_I$ denote their centers. The algorithm can be written as follows:

1. Compute

$$A_{\alpha,J} \equiv \frac{1}{\alpha!} \sum_{y_j \in J} q_j \left( \frac{y_j - y_J}{\sqrt{\delta}} \right)^{\alpha}$$

for $\alpha = 0, \ldots, \alpha_{\max}$ and for each source box $J$.

2. Compute

$$B_{\beta,I} \equiv \sum_{J} \sum_{\alpha=0}^{\alpha_{\max}} A_{\alpha,J} h_{\alpha+\beta} \left( \frac{x_I - y_J}{\sqrt{\delta}} \right)$$

for $\beta = 0, \ldots, \alpha_{\max}$ and for each target box $I$.

3. Compute

$$G(x_i) = \sum_{\beta=0}^{\alpha_{\max}} B_{\beta,I} \frac{1}{\beta!} \left( \frac{x_I - x_i}{\sqrt{\delta}} \right)^{\beta} \quad \text{for } i = 1, \ldots, N'.$$

Here $I$ is the target box $x_i$ belongs to.

Because each $x_i$ and $y_j$ belong to only one box, the total work for Steps 1 and 3 is still $O(N)$ and $O(N')$, respectively, while Step 2 needs work proportional to $O(N_{\text{box}}^2)$, where $N_{\text{box}}$ is the number of the boxes.

### 3.2. Implementation Details

In the actual algorithm, we can improve the efficiency of the above method by adopting some modifications, as we will state below.

First, because the Gaussian function $e^{-(x-y)^2}$ decreases very rapidly when the distance between $x$ and $y$ becomes large, the interaction between distant boxes in Step 2 can be omitted. It is shown in Greengard and Strain (1991) that considering only $2n+1$ nearest boxes with $n = 8$ is sufficient for double precision accuracy.

Second, if both the source and target boxes contain only a small number of points, it may be faster to evaluate the Gaussian directly than to use the Hermite expansion. Or, if the target box contains only a few points, it may be faster to expand the Gaussian only with respect to $y_j$ and use (18) to evaluate $G(x_i)$. Greengard and Strain (1991) recommend setting some threshold for the number of points in the box and using different evaluation methods according to whether the number of points in the source and target boxes is above or below the threshold.

Finally, it is possible to use alternative basis functions to expand the Gaussian. Greengard and Sun (1998) propose the use of an expansion formula based on the Fourier transform of the Gaussian, instead of the Hermite expansion, and show that it can reduce the work required in Step 2 drastically. This choice seems preferable when the number of points is moderate and the computation in Step 2 occupies a considerable part of the total work.

### 3.3. Application to Multinomial Methods

The algorithm described above is readily applicable to the computation of the continuation value (14) in the multinomial methods. This makes it possible to increase the branching order $l = 2b+1$ and attain a higher order of convergence without much increasing the computational work. When the number of time steps is $M$, the average number of nodes at each time step is $O(bM)$ and the total computational work is $O(bM^2)$.

The resulting algorithm is especially useful for Bermudan options, i.e., a variant of the American option for which the exercise opportunity is limited to $d$ discrete time points during the life of the option. This is because the method can approximate the underlying asset price process accurately even if the time step is large, thanks to the large number of branches. As a result, we can omit the time steps between the exercise dates and reduce the computational work to $O(bd^2)$, which is linear in the parameter controlling the accuracy. In contrast, in the binomial method, one has to increase the number of time steps $M$ to get a convergent result even if the number of exercise dates is fixed. This results in a computational work of $O(M^2)$, which is quadratic in the parameter controlling the accuracy.

## 4. Applications to Merton's Model

### 4.1. Multinomial Methods for the Lognormal Jump-Diffusion Model

We next extend our method to deal with the lognormal jump-diffusion model introduced by Merton

(1992). In this model, the asset price follows the equation

$$S_{t+\Delta t} = S_t \exp\left\{\left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t} z_0 + \sum_{i=1}^{N_t^P(\Delta t)} (\delta z_i - \kappa)\right\}, \tag{21}$$

where $N_t^P(\Delta t)$ is the number of jumps between time $t$ and $t+\Delta t$, which follows a Poisson process with intensity $\lambda$, and $z_i$ $(i = 0, 1, \ldots)$ are independent and follow the standard normal distribution $N(0,1)$. $\kappa$ and $\delta$ are constants that determine the mean and the standard deviation of the jumps, respectively. In this model, the market becomes incomplete because of the existence of jumps and therefore the standard option pricing argument based on the replicating portfolio is no longer valid. However, under the assumption that jump risk is diversifiable, Merton (1992) shows that the price can be written as

$$Q_0(S_0) = e^{-rT} E_0[h_T(S_T)]. \tag{22}$$

Because this has the same form as the formula in the complete market case, the incompleteness of the market causes little difficulty from the computational point of view.

To apply our method to this model, we introduce a change of variables and work with

$$x_t = \log S_t - \left(r - \frac{1}{2}\sigma^2\right) t, \tag{23}$$

which satisfies the equation

$$x_{t+\Delta t} = x_t + \sigma\sqrt{\Delta t} z_0 + \sum_{i=1}^{N_t^P(\Delta t)} (\delta z_i - \kappa). \tag{24}$$

If we fix the number of jumps between time $t$ and $t+\Delta t$ to $n$, we have

$$\begin{aligned} x_{t+\Delta t} &= x_t + \sigma\sqrt{\Delta t} z_0 + \sum_{i=1}^{n} (\delta z_i - \kappa) \\ &\sim N(x_t - n\kappa, \sigma^2\Delta t + n\delta^2), \end{aligned} \tag{25}$$

because the sum of Gaussian random variables is again a Gaussian random variable. We write the conditional probability density function of $x_{t+\Delta t}$ as

$$\begin{aligned} p^{(n)}(x_{t+\Delta t}|x_t) &\equiv p(x_{t+\Delta t}|x_t, N_t^P(\Delta t) = n) \\ &= \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left\{-\frac{1}{2\sigma_n^2}(x_{t+\Delta t} - x_t - \mu_n)^2\right\}, \end{aligned} \tag{26}$$

where

$$\sigma_n^2 = \sigma^2\Delta t + n\delta^2, \tag{27}$$

$$\mu_n = -n\kappa. \tag{28}$$

Now we approximate the stochastic process $x_t$ by a discrete random variable $x_{t_i}$ which is defined at discrete times $t_i = i\Delta t$ and takes discrete values $x_j = j\Delta x$. Then, as we have shown in the previous subsection, the expectation value of the option price conditioned on the current asset price and $N_t^P(\Delta t) = n$ can be approximated as

$$E[Q_{t_{i+1}}|S_{i,j}, N_t^P(\Delta t) = n] \cong \Delta x \sum_{k=-b}^{b} Q_{i+1,\,j+k} p^{(n)}(x_{j+k}|x_j). \tag{29}$$

Finally, the continuation value is calculated by

$$\begin{aligned} E[Q_{t_{i+1}}|S_{i,j}] &= \sum_{n=0}^{\infty} \Pr(N_t^P(\Delta t) = n) E[Q_{t_{i+1}}|S_{i,j}, N_t^P(\Delta t) = n] \\ &= \sum_{n=0}^{\infty} e^{-\lambda\Delta t} \frac{(\lambda\Delta t)^n}{n!} E[Q_{t_{i+1}}|S_{i,j}, N_t^P(\Delta t) = n], \end{aligned} \tag{30}$$

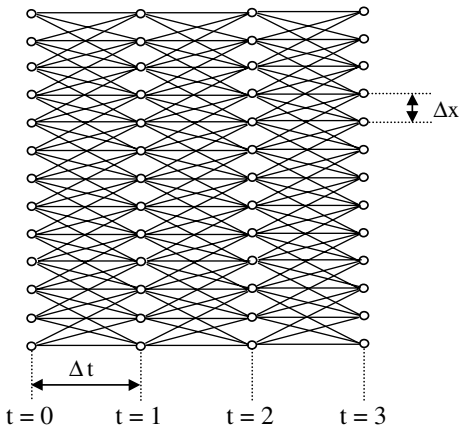
where for numerical computations the infinite sum can be truncated at a sufficiently large value of $n$, say $N_{\text{jump}}$.

Note that in the jump-diffusion case, unlike in the Black-Scholes model, the variances of $p^{(n)}(x_{t+\Delta t}|x_t)$ $(n = 1, 2, \ldots)$ remain finite even when $\Delta t \to 0$ because of the existence of jumps. It is therefore mandatory to use a value of $b$ large enough to approximate $p^{(n)}(x_{t+\Delta t}|x_t)$'s accurately. Hence, computing the sums of (29) by a direct method is computationally expensive. Note also that it is inefficient to use a multinomial lattice as shown in Figure 2, because the number of nodes increases rapidly with the number of time steps. Instead, we use a modified lattice as shown in Figure 4, in which the number of nodes remains a constant throughout the time.

### 4.2. Improving the Efficiency by the FGT

Now we can apply the FGT to compute the sums in the right-hand side of (29) efficiently. When the number of nodes at each time step is $N$ and the number of

**Figure 4** **A Modified Pentanomial Lattice** ($b = 2$)




time steps is $M$, the resulting algorithm has computational work of $O(MNN_{\text{jump}})$, regardless of the value of $b$. Thus, we can safely use the maximum value of $b$, which corresponds to using all the nodes at time $t_{i+1}$ to compute the expectation values at each node at time $t_i$.

It is possible to further reduce the computational work by modifying the algorithm of the FGT. Using the truncated version of (19), we can expand $p^{(n)}(x_{j+k}|x_k)$ as follows:

$$p^{(n)}(x_{j+k}|x_k) = \frac{1}{\sqrt{2\pi}\sigma_n} \sum_{\beta=0}^{\alpha_{\max}} \sum_{\alpha=0}^{\alpha_{\max}} \frac{1}{\beta!}\frac{1}{\alpha!}\left(\frac{x_{j+k}-x''}{\sqrt{2}\sigma}\right)^{\alpha} \cdot \left(\frac{\sigma}{\sigma_n}\right)^{\alpha+\beta} h_{\alpha+\beta}\left(\frac{x'-x''+\mu_n}{\sqrt{2}\sigma_n}\right) \cdot \left(\frac{x'-x_j}{\sqrt{2}\sigma}\right)^{\beta}, \tag{31}$$

where $x'$ and $x''$ are the centers of boxes containing $x_j$ and $x_{j+k}$, respectively. By putting this expansion into the right-hand side of (29) and substituting the result into (30), $E[Q_{t_{i+1}}|S_{i,j}]$ can be written as

$$\Delta x \sum_{k=-b}^{b} Q_{i+1,j+k} \sum_{\beta=0}^{\alpha_{\max}} \sum_{\alpha=0}^{\alpha_{\max}} \frac{1}{\beta!}\frac{1}{\alpha!}\left(\frac{x_{j+k}-x''}{\sqrt{2}\sigma}\right)^{\alpha} \cdot \left\{ \sum_{n=1}^{N_{\text{jump}}} e^{-\lambda t}\frac{(\lambda t)^n}{n!}\frac{1}{\sqrt{2\pi}\sigma_n}\left(\frac{\sigma}{\sigma_n}\right)^{\alpha+\beta} h_{\alpha+\beta}\left(\frac{x'-x''+\mu_n}{\sqrt{2}\sigma_n}\right)\right\} \cdot \left(\frac{x'-x_j}{\sqrt{2}\sigma}\right)^{\beta}. \tag{32}$$

Comparing this expression with (20), we know that we can construct an algorithm similar to the FGT by replacing the Hermite function with a weighted sum of shifted and scaled Hermite functions. Specifically, we have only to replace the formula to compute $B_\beta$ (see §3.1) with the following:

$$B_\beta = \sum_{\beta=0}^{\alpha_{\max}} A_\alpha \left\{ \sum_{n=1}^{N_{\text{jump}}} e^{-\lambda t}\frac{(\lambda t)^n}{n!}\frac{1}{\sqrt{2\pi}\sigma_n}\left(\frac{\sigma}{\sigma_n}\right)^{\alpha+\beta} \cdot h_{\alpha+\beta}\left(\frac{x'-x''+\mu_n}{\sqrt{2}\sigma_n}\right)\right\}. \tag{33}$$

This will enable us to compute $E[Q_{t_{i+1}}|S_{i,j}]$ with only one FGT instead of $N_{\text{jump}}$ transforms and reduce the computational work from $O(MNN_{\text{jump}})$ to $O(MN)$.

In implementing this algorithm, it is important to note that the box size must be determined from the smallest $\sigma_n$, that is $\sigma_0 = \sigma\Delta t$, to ensure the convergence of expansion (31). On the other hand, the number of boxes interacting with a given box must be determined from the cut-off radius of the Gaussian with the largest $\sigma_n$, that is $\sigma_{N_{\text{jump}}}$. As a result, the number of the interacting boxes becomes larger compared with the ordinary FGT. However, this will have little effect on the efficiency of the algorithm, because the computational work for the box-box interaction is a constant that depends neither on the number of source nor target points.

The method we have introduced in this section is especially suited to pricing Bermudan options, as is the case of the multinomial methods for the Black-Scholes model, and it requires only $O(dN)$ work when the number of exercise dates is $d$. As we shall see in §7.1, the method can also be extended to the double-exponential jump-diffusion model proposed by Kou (2002).

### 4.3. Amin's Algorithm

An alternative way to compute the option price under the lognormal jump-diffusion model is an algorithm proposed by Amin (1993). In this algorithm, the time step $\Delta t$ is taken to be small enough to be able to neglect the possibility of multiple jumps between $t$

and $t+\Delta t$. The jump probabilities are approximated as follows:

$$\begin{aligned} \Pr(N_t^P(\Delta t)=0) &= 1-\lambda\Delta t, \\ \Pr(N_t^P(\Delta t)=1) &= \lambda\Delta t. \end{aligned} \tag{34}$$

Then, using a multinomial lattice as shown in Figure 4, the expectation value (29) for $n=0$ is computed by a binomial-type method, with nonzero branching probabilities only for $k=+1$ (single-up) and $k=-1$ (single-down). The expectation value for $n=1$ is computed with a sufficiently large value of $b$ to represent the jump distribution. Amin (1993) showed that by setting the discretization step in the $x$-direction to $\Delta x=\sigma\sqrt{\Delta t}$ and using all the nodes at time $t_{i+1}$ to compute the expectation values for $n=1$ at each node at time $t_i$, the European and American option price calculated by this algorithm converges to the true price when $\Delta t$ approaches 0. When the number of time step is $M$, this algorithm requires computational work of $O(M^2)$ per time step, or of $O(M^3)$ for the entire procedure.

We can apply the FGT to this algorithm to compute the expectation values for $n=1$ in $O(M)$ work, thereby reducing the total work to $O(M^2)$. In the numerical results which follow in §6.2, we label this method FGT I. It is also possible to reduce the work to $O(M^2)$ by selecting a small number of nodes and compute the expectation values for $n=1$ by direct evaluation using only these nodes (Amin 1993). However, the accuracy of this algorithm is lower than that of the original one, and according to Amin (1993), the difference is around one cent when the option price is ten dollars and 30 selected nodes are used. In contrast, our approach based on the FGT can attain computational work of $O(M^2)$ without sacrificing the accuracy of the calculated prices.

# 5. The Stochastic Mesh Method

## 5.1. The Basic Algorithm

As another example of DP-based American option pricing methods that can be sped up by the FGT, we take up the stochastic mesh method proposed by Broadie and Glasserman (1997). This method is based on Monte Carlo simulation and computes the expectation value in (3) using randomly distributed sample points at time $t+1$. Unlike many other pricing algorithms based on Monte Carlo methods, it has the advantage that it can estimate the upper bound on the American option price, and has successfully been applied to the pricing of high-dimensional American options. Also, several extensions to improve the convergence and parallel implementations have been studied.

Assume that there are $n$ assets and the dynamics of the price vector $S_t$ is given in the form of the conditional probability density function

$$\Pr(S_{t+\Delta t}|S_t)=f_t^S(S_{t+\Delta t}|S_t). \tag{35}$$

In an implementation of the stochastic mesh method using the average mesh density function (for details see Broadie and Glasserman 1997), we first discretize the time and generate $b$ independent sample paths of the asset prices, as in the usual Monte Carlo approach. Let the vector of prices on path $i$ and time step $t$ be $S_{t,i}$. Then, the option value at maturity ($t=T$) for each path is given by

$$Q_T(S_{T,i})=h_T(S_{T,i}), \tag{36}$$

where $h_t$ is the payoff function. Next, we calculate the continuation value on each path at time $t=T-1$ by estimating the conditional expectation value $E[Q_T(S_T)|S_{T-1,i}]$, using the sample points at $t=T$. However, this cannot be done by taking a simple average, because one needs a set of sample points that follow the conditional distribution function $f_{T-1}^S(S_T|S_{T-1,i})$ to calculate $E[Q_T(S_T)|S_{T-1,i}]$, whereas the actual sample points follow the distribution

$$g_T^S(S_T)=\frac{1}{b}\sum_{i=1}^{b} f_{T-1}^S(S_T|S_{T-1,i}), \tag{37}$$

from the construction of the sample paths.

To overcome this difficulty, the stochastic mesh method uses the following relationship:

$$\begin{aligned} E[Q_t(S_t)|S_{t-1,i}] &= \int Q_t(u) f_{t-1}^S(u|S_{t-1,i})\,du \\ &= \int Q_t(u)\frac{f_{t-1}^S(u|S_{t-1,i})}{g_t^S(u)} g_t^S(u)\,du \\ &= E\left[ Q_t(S_{t,j})\frac{f_{t-1}^S(S_{t,j}|S_{t-1,i})}{g_t^S(S_{t,j})} \right]. \end{aligned} \tag{38}$$

The last expression is an expectation value under the density function $g^S_{(t-1)}$ and can be calculated using the sample points at $t=T$ as

$$E\left[Q_t(S_{t,j})\frac{f^S_{t-1}(S_{t,j}|S_{t-1,i})}{g^S_t(S_{t,j})}\right]=\sum_{j=1}^{b}Q_{t,j}w_{t,ij}, \tag{39}$$

where $(w_{t,ij})$ is a weight matrix defined by

$$\begin{aligned} w_{t,ij} &= \frac{f^S_{t-1}(S_{t,j}|S_{t-1,i})}{g^S_t(S_{t,j})} \\ &= \frac{f^S_{t-1}(S_{t,j}|S_{t-1,i})}{\sum_{i'=1}^{b} f^S_{t-1}(S_{t,j}|S_{t-1,i'})}. \end{aligned} \tag{40}$$

Apparently, the formation of and multiplication by the weight matrix needs $O(b^2)$ computation, and this is the most time-consuming part in the stochastic mesh method.

### 5.2. Improvement of Efficiency by the FGT

Prior to applying the FGT to the stochastic mesh method, we first investigate the effect of changing the variables in the calculation. Suppose we change the variable from $S_t$ to $y_t$, which is again an $n$-dimensional vector. Then, the conditional probability density function of $y_t$ is related to that of $S_t$ by

$$f^S_{t-1}(S_t|S_{t-1})=f^y_{t-1}(y_t|y_{t-1})\left|\frac{\partial y_t}{\partial S_t}\right|, \tag{41}$$

where $\left|\partial y_t/\partial S_t\right|$ is the Jacobian matrix. By substituting this into the expression of $w_{t,ij}$, we have

$$\begin{aligned} w_{ij,t} &= \frac{f^S_{t-1}(S_{t,j}|S_{t-1,i})}{\sum_{i'=1}^{b} f^S_{t-1}(S_{t,j}|S_{t-1,i'})} \\ &= \frac{f^y_{t-1}(y_{t,j}|y_{t-1,i})\left|\partial y_t/\partial S_t\right|_{S_t=S_{t,j}}}{\sum_{i'=1}^{b} f^y_{t-1}(y_{t,j}|y_{t-1,i'})\left|\partial y_t/\partial S_t\right|_{S_t=S_{t,j}}} \\ &= \frac{f^y_{t-1}(y_{t,j}|y_{t-1,i})}{\sum_{i'=1}^{b} f^y_{t-1}(y_{t,j}|y_{t-1,i'})}. \end{aligned} \tag{42}$$

Thus, we have established that the weight matrix in the stochastic mesh method (using the average mesh density function) is invariant under change of variables. So we can choose any convenient representation of the state variables to compute the weight matrix and the sum in (39).

We now turn our attention to the multiasset Black-Scholes model:

$$\begin{aligned} dS_m &= r_m S_m dt+\sigma_m S_m dW_m, \\ dW_m dW_l &= \rho_{ml} dt \quad (m\neq l), \end{aligned} \tag{43}$$

where $S_m$ is the price of the $m$th asset, and $dW_m$ is a Wiener process. By changing the variables from $S_m$ to

$$x_m=\log S_m-\left(r_m-\frac{1}{2}\sigma_m^2\right)t, \tag{44}$$

we get a vector variable $x=(x_1,\dots,x_n)^t$ which satisfies

$$\begin{aligned} dx\,dx^t &= \Sigma R\Sigma\, dt, \\ \Sigma &= \mathrm{diag}(\sigma_1,\dots,\sigma_n), \\ R &= (\rho_{ij}). \end{aligned} \tag{45}$$

By further introducing a new change of variables using the Cholesky decomposition $R=LL^t$ as

$$y=L^{-1}\Sigma^{-1}x, \tag{46}$$

we finally have the vector $y$ of $n$ independent Wiener process which satisfies

$$dy\,dy^t=I\,dt. \tag{47}$$

For this new variable, the conditional probability density function can be seen to be

$$f^y_{t-1}(y_1,\dots,y_n)=\prod_{m=1}^{n}\frac{1}{\sqrt{2\pi\Delta t}}\exp\left\{-\frac{1}{2\Delta t}(y_{t-1,m}-y_m)^2\right\}, \tag{48}$$

and multiplication by the weight matrix can be done using the FGT. It is also straightforward to extend this approach to Merton's lognormal jump-diffusion model by using the technique introduced in §4.

## 6. Numerical Experiments

### 6.1. Multinomial Methods for the Black-Scholes Model

We implemented the multinomial method using the FGT for European and Bermudan options under the Black-Scholes model and compared its efficiency with that of the binomial method by Cox et al. (see, e.g.,

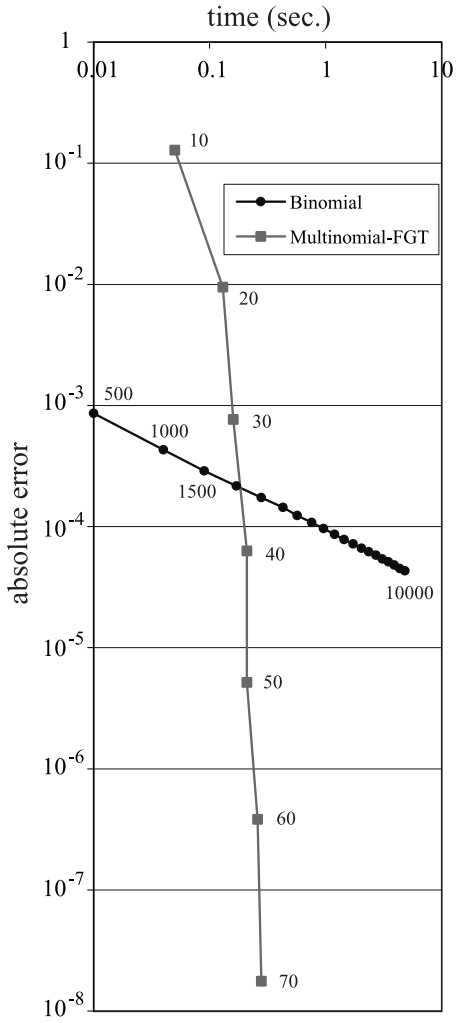
Kwok 1998). In all of the cases below, we used a closed-form solution by the Black-Scholes formula for the continuation value at the penultimate time step, as suggested in Broadie and Detemple (1996). In the case of European options, this corresponds to replacing the actual payoff function at maturity with an analytical payoff function at the penultimate time step, and is effective in removing the oscillatory convergence characteristic of the binomial and multinomial type methods.

All the experiments were done on a 266MHz Pentium II PC with Red-Hat Linux using gnu C++ compiler. Throughout this section, we denote the initial asset price, the strike price, and the option price at $t=0$ by $S_0$, $K$, and $Q_0$, respectively. We also denote the riskless annual interest rate, dividend rate, and volatility by $r$, $q$, and $\sigma$, respectively. Finally, we use $T$ (in years) for option maturity and $d$ for the number of exercise dates.

**6.1.1. European Option on a Single Asset.** We show the results for a European call option in Figure 5. Here, $S_0=K=100$, $r=0.03$, $q=0.07$, $\sigma=0.20$, and $T=0.5$. The exact price by the Black-Scholes formula is 4.57776128. We calculated the option value using a binomial method with time steps $M$ from 500 to 10,000 (in increments of 500) and a multinomial method with $2b+1$ branches, where $b=10$ to 150 (in increments of 10). The values of these parameters are also shown in the graph. In the multinomial method, the number of time steps was fixed to 10, because it can guarantee convergence without increasing the number of time steps when the number of branches is increased.

In Figure 5, the vertical axis and the horizontal axis represent the error in the calculated option price and the computational time, respectively, both in a log scale. As can be clearly seen from the graph, the steepness of the binomial result is $-1/2$, implying that the error decreases as $O(M^{-1})$. On the other hand, the graph of the multinomial method is nearly vertical, which confirms the very high order of convergence theoretically guaranteed when the option payoff function is analytical. Note that the computational time of the multinomial method and the binomial method cross around error $\sim 10^{-4}$, and the former becomes faster when a higher accuracy is needed.

**Figure 5 European Call Option Price Under the Black-Scholes Model**



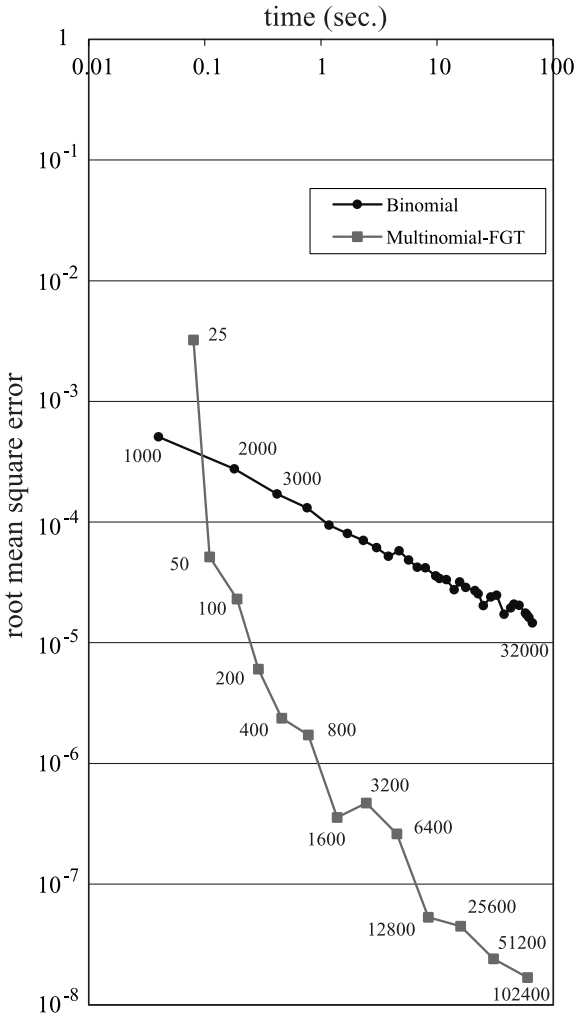

**6.1.2. Bermudan Option on a Single Asset.** Next, we calculated the prices of Bermudan call options. The parameters are the same as in the case of the European option described above, except that the strike price $K$ is varied from 90 to 110 (in increments of 5). The number of exercise dates is $d=10$. We adopted the prices calculated by the multinomial method with $b=409{,}600$ as reference prices against which to compute the error. These values are listed in Table 1 and they agree with the values calculated by the Binomial Black-Scholes Richardson extrapolation (BBSR) method (Broadie and Detemple 1996) with 5,00,000 time steps to at least six digits after the decimal point. The time steps of the binomial

**Table 1 Bermudan Call Option Price Under the Black-Scholes Model**

| $K$ | 90 | 95 | 100 | 105 | 110 |
|---|---|---|---|---|---|
| $Q_0$ | 10.73001013 | 7.32288562 | 4.75727741 | 2.94105489 | 1.73255637 |

method is from 1,000 to 32,000 (in increments of 1,000), and $b$ for the multinomial method is set to $25 \cdot 2^k$ ($k = 0, 1, \ldots, 12$). The number of time steps in the multinomial method was set equal to the number of exercise dates. The result is shown in Figure 6. Here, the vertical axis represents the root mean square error of the five calculated prices.

**Figure 6 Bermudan Call Option Price Under the Black-Scholes Model**




In this case, again, the multinomial method is more efficient than the binomial method when the required accuracy is higher than $10^{-4}$. However, the convergence speed of the multinomial method is slower than in the European option case. This seems to be because the payoff function of the Bermudan option is not analytical, i.e., it has a discontinuity in the second- and higher-order derivatives at the exercise boundary (see, e.g., Kwok 1998).

### 6.2. Merton Results

We also implemented the numerical methods for pricing the options under Merton's lognormal jump-diffusion model which we described in §4. The methods we take up are as follows:

- Amin's original algorithm.
- Amin's algorithm with the FGT (FGT I): The conditional expectation values corresponding to a single jump between two time steps are calculated using the FGT.
- Multinomial method with the FGT (FGT II): for European and Bermudan options only. The option prices are computed only on the exercise dates and the possibility of multiple jumps between two time steps is taken into account. This is the algorithm described in §4.2.

We use the parameters $S_0 = 40$, $K = 30$ to 50 (in increments of 5), $r = 0.08$, $q = 0$, $\sigma = \sqrt{0.05}$, and $T = 1.0$. The jump intensity and variance of the jumps are $\lambda = 5.0$/year and $\delta = \sqrt{0.05}$, respectively, and the mean of the jump size is set to $-(1/2)\delta^2$. These are the values used in Amin (1993). We computed the price of European options, American options, and Bermudan options with ten exercise dates. As reference prices for the European options, we used the prices computed by Merton's formula. For the American options, we adopted the prices obtained by extrapolating the results from Amin's algorithm. For the Bermudan options, we used the values computed by FGT II with $N = 204{,}801$, where $N$ is the number of nodes at each time step. These prices, which we denote by $Q_0^E$, $Q_0^A$, and $Q_0^B$, respectively, are listed in Table 2.

The numerical results for the European, American, and Bermudan options are shown in Figures 7, 8, and 9, respectively. The number of nodes $N$ at each time step is also shown in the graph. $N$ is connected with

**Table 2 Put Option Prices Under Merton's Model**

| $K$ | 30 | 35 | 40 | 45 | 50 |
|---|---|---|---|---|---|
| $Q_0^E$ | 2.62113699 | 4.41159556 | 6.69595339 | 9.42219161 | 12.52384675 |
| $Q_0^A$ | 2.71690583 | 4.59897152 | 7.02293320 | 9.94610561 | 13.30914732 |
| $Q_0^B$ | 2.70636133 | 4.58094453 | 6.99527475 | 9.90671277 | 13.25586096 |

the number of time steps $M$ by $N=2M+1$ in the case of Amin's original algorithm and FGT I. For Amin's algorithm and FGT I, it can be clearly seen that the errors decrease as $T_c^{-1/3}$ and $T_c^{-1/2}$, respectively, where $T_c$ is the computational time. This is in agreement with our observation in §4.3 that the computational work of Amin's algorithm can be reduced from $M^3$ to $M^2$ without sacrificing the accuracy by using the FGT. For Bermudan options, the efficiency can further be improved by omitting the intermediate time steps using the algorithm FGT II, resulting in a convergence rate of $T_c^{-1}$. From these results, we can conclude that the FGT is especially useful in computing the option price under the jump-diffusion models.

**Figure 7 European Put Option Price Under Merton's Model**

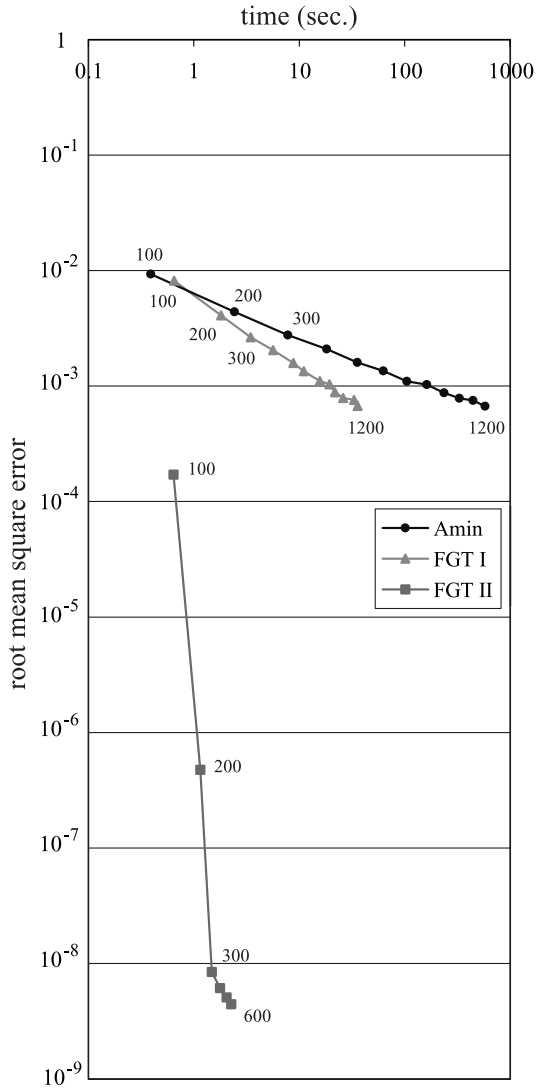
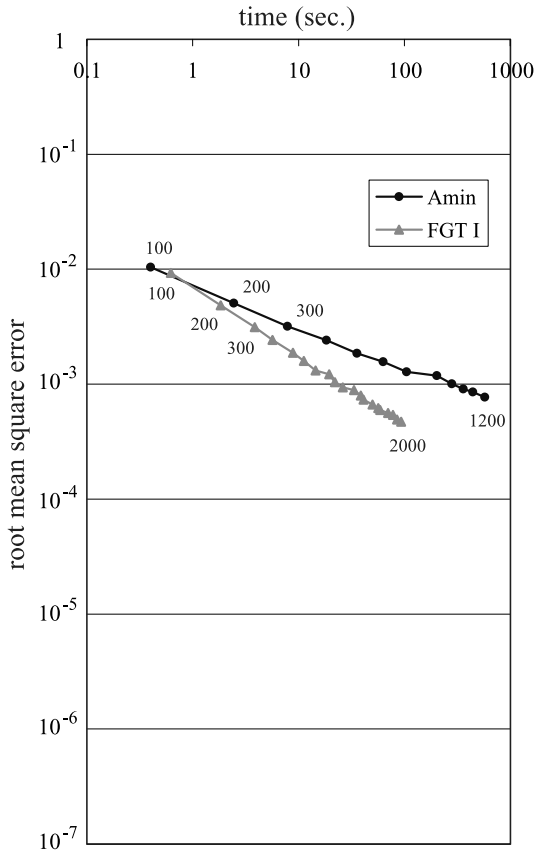


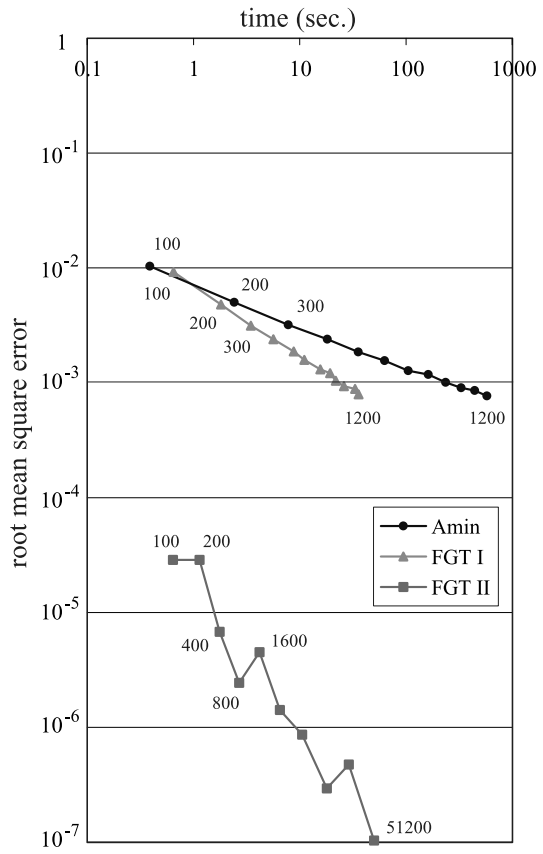

**Figure 8 American Put Option Price Under Merton's Model**




### 6.3. Stochastic Mesh Method Results

We also implemented the stochastic mesh method for pricing Bermudan options with and without the FGT and compared the speed and accuracy of the two implementations. The options we chose are Bermudan call options on one, two, and three underlying assets under the Black-Scholes model. The environment of the experiments is the same as described in the previous subsection and we use the same notation.

**Figure 9 Bermudan Put Option Price Under Merton's Model**




The options we used in our experiment are as follows:

- A Bermudan call option on a single underlying asset, with $S_0=K=100$, $r=0.05$, $q=0.10$, $\sigma=0.20$, $T=3.0$, and $d=2$.
- A Bermudan max call option on two underlying assets, with $S_0=K=100$, $r=0.05$, $q=0.10$, $\sigma=0.20$, $T=1.0$, and $d=3$. The correlation between the two option is $\rho=0.3$.
- A Bermudan max call option on three underlying assets, with $S_0=K=100$, $r=0.05$, $q=0.10$, $\sigma=0.20$, and $T=1.0$. The correlation between the two option is $\rho=0.3$.

The parameter values used in Option 1 are the same as those used in Andersen and Broadie (2001), and the option price given there is 7.18. The values in Case 2 are the same as in Fu et al. (2001), and the price given there is 9.390.

Tables 3, 4, and 5 and Figures 10 and 11 show the results of our computation. For each case, we changed the number of sample paths from $b=1{,}000$ to 100,000. As can be seen from the tables, the FGT can speed up the stochastic mesh method drastically, making it more than 1,300 times faster in the one-dimensional case when $b=10{,}000$, and about 60 times faster in the two-dimensional case when $b=10{,}000$. The speedup is less striking in the three-dimensional case, but still the new implementation is nearly ten times faster when $b=30{,}000$ and more than 30 times faster when $b=100{,}000$. It is also apparent that the computational time is linear in the number of sample paths for the implementation with the FGT, while it is quadratic for the conventional implementation. As for the accuracy, we can say that the FGT is sufficiently accurate,

**Table 3 Bermudan Call Option on a Single Asset**

| Number of sample paths | Direct calculation | | Fast Gauss transform | |
|---|---|---|---|---|
| | Price | Time | Price | Time |
| 1,000 | 7.754283 | 4.04 s | 7.754283 | 0.03 s |
| 3,000 | 7.445551 | 36.37 s | 7.445551 | 0.11 s |
| 10,000 | 7.365219 | 412.34 s | 7.365219 | 0.31 s |
| 30,000 | 7.276099 | (Timer overflow) | 7.276099 | 0.96 s |
| 100,000 | — | — | 7.202183 | 3.38 s |

**Table 4 Bermudan Max Call Option on Two Underlying Assets**

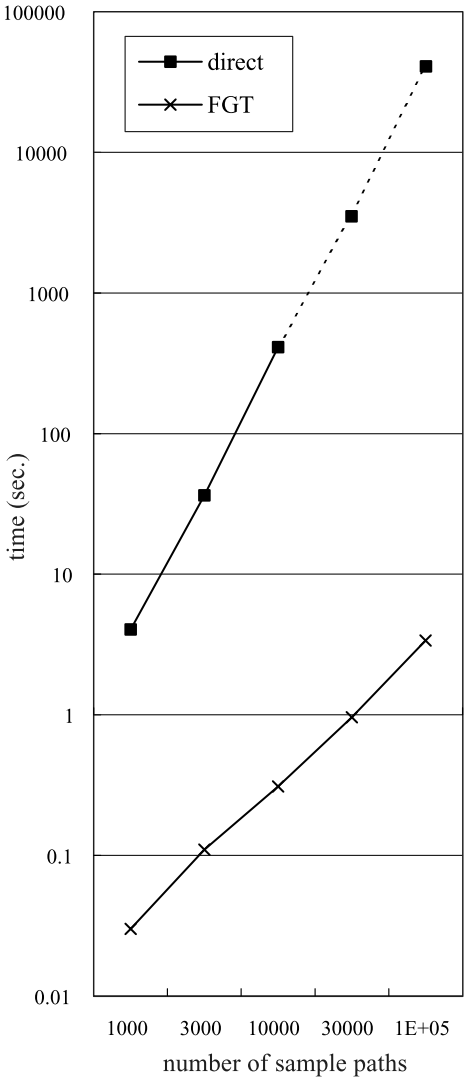
| Number of sample paths | Direct calculation | | Fast Gauss transform | |
|---|---|---|---|---|
| | Price | Time | Price | Time |
| 1,000 | 10.387124 | 7.07 s | 10.387124 | 2.95 s |
| 3,000 | 9.597424 | 63.11 s | 9.597424 | 5.95 s |
| 10,000 | 9.546258 | 706.21 s | 9.546258 | 11.96 s |
| 30,000 | — | — | 9.438231 | 21.96 s |
| 100,000 | — | — | 9.337766 | 49.26 s |

**Table 5 Bermudan Max Call Option on Three Underlying Assets**

| Number of sample paths | Direct calculation | | Fast Gauss transform | |
|---|---|---|---|---|
| | Price | Time | Price | Time |
| 1,000 | 13.241150 | 7.21 s | 13.241150 | 109.33 s |
| 3,000 | 12.397652 | 65.01 s | 12.397652 | 161.85 s |
| 10,000 | 12.294868 | 733.77 s | 12.294868 | 329.62 s |
| 30,000 | 12.075991 | 7236.2 s | 12.075991 | 778.07 s |
| 100,000 | — | 73377 s* | 12.015780 | 2253.76 s |

*Estimated time.

**Figure 10 Bermudan Call Option on a Single Asset**




for the option prices calculated by the two implementations agree to at least six digits after the decimal point.

# 7. Extension to Non-Gaussian Densities

## 7.1. Extension to Kou's Double-Exponential Jump-Diffusion Model

Kou (2002) proposed a new jump-diffusion model in which the logarithm of the jump size has a double-exponential distribution instead of the normal distribution. In this model, the asset price at time $t+\Delta t$ can be written as

$$S_{t+\Delta t} = S_t \exp\left\{\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}z + \sum_{i=1}^{N_t^P(\Delta t)} X_i\right\}, \quad (49)$$

where $N_t^P(\Delta t)$ is the number of jumps between time $t$ and $t+\Delta t$, which follows a Poisson process with intensity $\lambda$, $z \sim N(0,1)$, and $X_i$ are independent variables that follow a double-exponential distribution:

$$f_X(x) = \frac{1}{2\eta} e^{-|x-\kappa|/\eta}, \quad 0 < \eta < 1. \quad (50)$$

This model has an advantage over the lognormal jump-diffusion model in that it can have both higher peak and heavier tails than normal distribution.

We first change the variable from $S_t$ to

$$u_t = \log S_t - \left(\mu - \frac{1}{2}\sigma^2\right)t, \quad (51)$$

as we did in §4.1. Then, $u_t$ satisfies the equation

$$u_{t+\Delta t} = u_t + \sigma\sqrt{\Delta t}z + \sum_{i=1}^{N_t^P(\Delta t)} X_i. \quad (52)$$

To apply our method, we need an explicit form of the conditional probability density function $f_u(u_{t+\Delta t}|u_t)$. As in the case of the lognormal jump-diffusion model, this can be written as a sum over the number of jumps between $t$ and $t+\Delta t$:

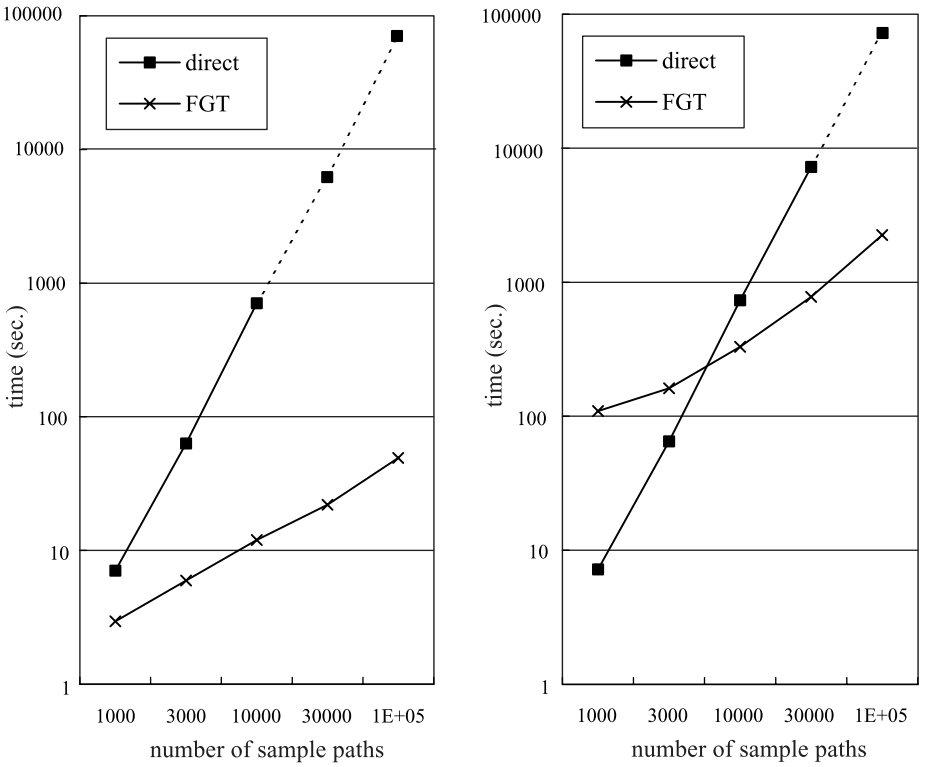
$$\begin{aligned} f_u(u_{t+\Delta t}|u_t) &= \sum_{n=0}^{\infty} \Pr(N_t^P(\Delta t) = n) f_u(u_{t+\Delta t}|u_t, N_t^P(\Delta t) = n) \\ &= \sum_{n=0}^{\infty} e^{-\lambda\Delta t} \frac{(\lambda\Delta t)^n}{n!} f_u^{(n)}(u_{t+\Delta t} - u_t), \end{aligned} \quad (53)$$

where $f_u^{(n)}(x)$ is a distribution function of the sum of $\sigma\sqrt{\Delta t}z$ and $n$ random variables $\{X_i\}_{i=1}^n$ that follow the double-exponential distribution (50).

Kou (2002) shows that sum of double-exponential random variables can be decomposed into a random sum of exponential random variables as follows:

$$\sum_{i=1}^{n} X_i - n\kappa = \begin{cases} \sum_{j=1}^{M(n)} \xi_j & \text{with probability } 1/2, \\ -\sum_{j=1}^{M(n)} \xi_j & \text{with probability } 1/2, \end{cases} \quad (54)$$

**Figure 11 Bermudan Max Call Option on Two Underlying Assets (Left Graph) and on Three Underlying Assets (Right Graph)**




where $\{\xi_j\}$ are independent random variables which follow an exponential distribution with mean $\eta$, and

$$\Pr(M(n)=m)=\frac{2^m}{2^{2n-1}}\binom{2n-m-1}{n-1}, \quad 1\le m\le n. \quad (55)$$

He also shows that if $X^{(m)}$ is a random variable such that

$$X^{(m)}=\begin{cases}\sum_{j=1}^{m}\xi_j & \text{with probability } p,\\ -\sum_{j=1}^{m}\xi_j & \text{with probability } 1-p,\end{cases} \quad (56)$$

and $Y\sim N(0,\sigma^2)$, then the probability density function of $X^{(m)}+Y$ is

$$f_{X^{(m)}+Y}(v)=\frac{\sigma^m}{\eta^m}\frac{e^{\sigma^2/(2\eta^2)}}{\sigma\sqrt{2\pi}}\left\{pe^{-v/\eta}Hh_{m-1}\left(-\frac{v\eta-\sigma^2}{\sigma\eta}\right)+(1-p)e^{v/\eta}Hh_{m-1}\left(\frac{v\eta+\sigma^2}{\sigma\eta}\right)\right\}. \quad (57)$$

Here, $Hh_m(x)$ is a $Hh$ function defined by

$$Hh_m(x)=\frac{1}{m!}\int_x^\infty (t-x)^m e^{-t^2/2}\,dt. \quad (58)$$

By combining these results, we can write the term $f_u(u_{t+\Delta t}|u_t)$ as

$$\begin{aligned}f_u(u_{t+\Delta t}|u_t)&=\sum_{n=0}^{\infty}e^{-\lambda\Delta t}\frac{(\lambda\Delta t)^n}{n!}f_u^{(n)}(u_{t+\Delta t}-u_t)\\&=\sum_{n=0}^{\infty}\sum_{m=1}^{n}e^{-\lambda\Delta t}\frac{(\lambda\Delta t)^n}{n!}\frac{2^m}{2^{2n-1}}\cdot\binom{2n-m-1}{n-1}f_{X^{(m)}+Y}(u_{t+\Delta t}-u_t-n\kappa).\end{aligned}$$

Because $f_u(u_{t+\Delta t}|u_t)$ is given as a sum of $f_{X^{(m)}+Y}(u_{t+\Delta t}-u_t-n\kappa)$, we only need to develop a fast algorithm for the latter. Moreover, because this consists of two terms as shown in (57), we only have to consider each of the terms, that is,

$$f^{(-)}(v)=e^{-v/\eta}Hh_{m-1}\left(-\frac{v\eta-\sigma^2}{\sigma\eta}\right) \quad (59)$$

and

$$f^{(+)}(v)=e^{v/\eta}Hh_{m-1}\left(\frac{v\eta+\sigma^2}{\sigma\eta}\right). \quad (60)$$

We consider an expansion of $f^{(+)}(x_i - y_j)$, as we did in (19). First, from the recurrence of the $Hh$ function

$$\frac{d}{dx} Hh_m(x) = -Hh_{m-1}(x), \quad m = 0, 1, 2, \ldots, \tag{61}$$

we can find a Taylor expansion of the $Hh$ function as we did in (18), where the Taylor coefficients are expressed again by $Hh$ functions (though $Hh_m$ functions are defined only for $m \geq -1$, we can naturally extend the definition for $m < -1$ by using the Hermite functions defined by (17)). Then, we can again expand the $Hh$ function in the sum using (61), obtaining a two-sided expansion of the $Hh$ function as we had in (19). Moreover, for the other factor in (60), $e^{v/\eta}$, we have an expansion

$$e^{x_i - y_j} = e^{x_i - x_0} e^{x_0 - y_0} e^{y_0 - y_j}. \tag{62}$$

In this way, $f^{(+)}(x_i - y_j)$ has the same form of expansion as (19). Based on this, we can construct a fast algorithm like the FGT.

The approach we took here to derive a fast algorithm can be applied to other jump-diffusion models as well, if the conditional probability density function $f_u(u_t + \Delta t | u_t)$ has a two-sided expansion as shown in (19). For example, if the state space is homogeneous, that is, $f_u(u_{t+\Delta t} | u_t)$ depends only on the difference $u_{t+\Delta t} - u_t$, the conditional probability density function can be expressed as a Fourier integral: $f_u(u_{t+\Delta t} | u_t) = \int g(k) e^{ik(u_{t+\Delta t} - u_t)} dk$. Then, the integral can be discretized to give a Fourier series, and the exponential function can be decomposed as in (62), leading to a two-sided expansion of the conditional probability density function.

While this method is quite general, its efficiency depends critically on whether the Fourier series converges sufficiently fast or not. It is therefore worthwhile to explore problem-specific expansions such as (19) and the expansion we have developed in this section. We are now developing such an expansion for other jump-diffusion models including the variance gamma model.

### 7.2. Extension to Stochastic Volatility Models

Another important class of models to which our method has potential application is the stochastic volatility model of Heston (1993) and other models in the affine jump-diffusion class treated in Duffie et al. (2000). In these models, the state space consists of two components, namely the log asset price $x_t$ and its volatility $v_t$. To apply our method, we need the conditional probability density function $f(x_{t+\Delta t}, v_{t+\Delta t} | x_t, v_t)$. Although this is typically not known in closed form, for affine jump-diffusion models, its Fourier transform with respect to $x_{t+\Delta t}$ and $v_{t+\Delta t}$ can be obtained in closed form. It is in principle possible to exploit this fact and derive a two-sided expansion of the conditional probability density function.

However, it is not so straightforward to implement this idea efficiently, because the two-dimensional space of $x_t$ and $v_t$ is not homogeneous, in contrast to the cases which we have dealt with so far. This means that the shape of the conditional probability density function (cpdf) differs much according to the value of $v_t$, and as a result, a large number of basis functions are necessary to represent the cpdf accurately for all values of $v_t$. We are now developing a method to reduce the number of the basis functions and improve the efficiency of our method for these models.

## 8. Conclusion

In this paper, we have shown that in many of the numerical methods for pricing American options, the most computationally intensive part can be formulated as summation of Gaussians. In particular, we demonstrated this for the multinomial method and the stochastic mesh method for the Black-Scholes model and the lognormal jump-diffusion model. We then introduced the idea of the FGT, and showed how it can reduce the order of computational work from $O(NN')$ to $O(N + N')$, where $N'$ is the number of summations and $N$ is the number of terms appearing in each summation.

The results of numerical experiments show that the multinomial method based on the FGT has a convergence rate much higher than that of the binomial method, and is more efficient than the latter when higher accuracy is needed. It was also shown that the stochastic mesh method can be accelerated by the FGT considerably, at least in one, two, and three dimensions.

Finally, we proposed an extension of the FGT to handle non-Gaussian densities, in particular, Kou's (2002) double-exponential jump-diffusion model. In future work, we plan to extend our approach to stochastic volatility models and other models in the affine jump-diffusion class.

## Acknowledgments

The authors are grateful to Steve Kou, Cedric Besnier, and an anonymous referee for many useful suggestions. This work was partially supported by NSF grant DMS-0074637.

## References

Alford, J., N. Webber. 2001. Very high order lattice methods for one factor models. Cass Business School, London, U.K., http://www.cass.city.ac.uk / facfin / facultypages / nwebber / research.html. Retrieved March 10, 2003.

Amin, K. 1993. Jump diffusion option valuation in discrete time. *J. Finance* **48**(5) 1833–1863.

Andersen, L., M. Broadie. 2001. Practical primal-dual simulation algorithms for pricing multidimensional American options. Working paper, Columbia University, New York.

Baxter, B., G. Roussos. 2002. A new error estimate of the fast Gauss transform. *SIAM J. Sci. Comput.* **24**(1) 257–259.

Broadie, M., J. Detemple. 1996. American option valuation: New bounds, approximations, and a comparison of existing methods. *Rev. Financial Stud.* **9**(4) 1211–1250.

——, P. Glasserman. 1997. A stochastic mesh method for pricing high-dimensional American options. Working paper, Columbia University, New York, *J. Comput. Finance*. Forthcoming.

Duffie, D. 1996. *Dynamic Asset Pricing Theory*, 3rd ed. Princeton University Press, Princeton, NJ.

——, J. Pan, K. Singleton. 2000. Transform analysis and asset pricing for affine jump diffusions. *Econometrica* **68** 1343–1376.

Florence, A. 2001. Computational multilinear algebra. Ph.D. thesis, Cornell University, Ithaca, NY.

Fu, M., S. Laprise, D. Madan, Y. Su, R. Wu. 2001. Pricing American options: A comparison of Monte Carlo approaches. *J. Comput. Finance* **4**(3) 39–88.

Greengard, L., J. Strain. 1991. The fast Gauss transform. *SIAM J. Sci. Statist. Comput.* **12**(1) 79–94.

——, X. Sun. 1998. A new version of the fast Gauss transform. *Documenta Math.* Extra Volume **ICM**(III) 575–584.

Heston, S. 1993. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financial Stud.* **6**(2) 327–343.

——, G. Zhou. 2000. On the rate of convergence of discrete-time contingent claims. *Math. Finance* **10**(1) 53–75.

Kou, S. 2002. A jump diffusion model for option pricing. *Management Sci.* **48**(8) 1086–1101.

Kwok, Y. K. 1998. *Mathematical Models of Financial Derivatives*. Springer, New York.

Lamberton, D., B. Lapeyre. 1996. *Introduction to Stochastic Calculus Applied to Finance*. Chapman & Hall/CRC, New York.

Longstaff, F., E. Schwartz. 2001. Valuing American options by simulation: A simple least-squares approach. *Rev. Financial Stud.* **14**(1) 113–147.

Merton, R. 1992. *Continuous-Time Finance*. Blackwell, New York.

Reiner, E. 2000. Convolution methods for exotic options. Presented at Columbia University, New York (March 22).

Sloan, I., S. Joe. 1994. *Lattice Methods for Multiple Integration*. Clarendon Press, Oxford, U.K.

Van Steenkiste, R., S. Foresi. 1999. Arrow-Debreu prices for affine models. Working paper, Salomon Smith Barney, New York.

Strain, J. 1991. The fast Gauss transform with variable scales. *SIAM J. Sci. Statist. Comput.* **12**(5) 1131–1139.

*Accepted by R. Jagannathan, former department editor; received May 21, 2002. This paper was with the authors 5 months for 2 revisions.*