OPERATIONS RESEARCH
Vol. 59, No. 5, September–October 2011, pp. 1233–1245
ISSN 0030-364X | EISSN 1526-5463 | 11 | 5905 | 1233

http://dx.doi.org/10.1287/opre.1110.0962
© 2011 INFORMS

Exact Simulation of Point Processes with Stochastic Intensities

K. Giesecke
Department of Management Science and Engineering, Stanford University, Stanford, California 94305, giesecke@stanford.edu

H. Kakavand
The Perot Group, hossenink@gmail.com

M. Mousavi
Department of Management Science and Engineering, Stanford University, Stanford, California 94305, mousavi@stanford.edu

Point processes with stochastic arrival intensities are ubiquitous in many areas, including finance, insurance, reliability, health care, and queuing. They can be simulated from a Poisson process by time scaling with the cumulative intensity. The paths of the cumulative intensity are often generated with a discretization method. However, discretization introduces bias into the simulation results. The magnitude of the bias is difficult to quantify. This paper develops a sampling method that eliminates the need to discretize the cumulative intensity. The method is based on a projection argument and leads to unbiased simulation estimators. It is exemplified for a point process whose intensity is a function of a jump-diffusion process and the point process itself. In this setting, the method facilitates the exact sampling of both the point process and the driving jump-diffusion process. Numerical experiments demonstrate the effectiveness of the method.

Subject classifications: point process; intensity projection; filtering; exact sampling.
Area of review: Simulation.
History: Received March 2010; revisions received June 2010, September 2010; accepted September 2010.

1. Introduction

Stochastic point processes are prominent in many areas. In finance and economics, they describe the arrival of events such as corporate bankruptcies, mergers and acquisitions, or security trades on an exchange. In insurance, they represent claim arrivals. In reliability, they model failures of system components. In queuing, they record the arrivals or departures of customers. In health care, they represent incidences of diseases. In seismology, they model the timing of earthquakes.

Monte Carlo simulation is an important computational tool to address point process applications. At its center is a method for generating the event times. The choice of method is governed by the properties of the point process intensity, which represents the conditional event arrival rate and which is often modulated by risk factors that follow stochastic processes on their own. The time-scaling method has the widest scope. It is based on a result of Meyer (1971), which implies that under mild conditions, any counting process can be transformed into a standard Poisson process by a change of time that is given by the counting process compensator, or cumulative intensity. Thus, the event times can be generated by rescaling Poisson arrivals with the compensator. Barring special cases, the continuous-time path of the compensator must be approximated on a discrete-time grid. However, the approximation of a continuous-time process by a discrete-time process introduces bias into the simulation estimator. The bias is undesirable for several reasons. First, because the size of the bias is often unknown, it is hard to obtain valid confidence intervals. Second, a very fine time discretization may be required to reduce the bias to an acceptable level. Even more computational effort may be required to verify that the bias is sufficiently small. Finally, both the number of time steps and the number of trials need to be increased together to decrease the total error of the simulation estimator, but the optimal allocation of resources is difficult to specify in advance.

In this article, we develop an *exact* simulation method that eliminates the need to discretize the compensator. The method exploits a projection argument and is based on a change of the filtration that describes the information flow in the point process model. Rather than sampling the point process in the reference filtration associated with the given point process model, we project the point process onto its own filtration and then sample it in this coarser filtration. By changing the filtration, the point process acquires an intensity with much simpler dynamics: the projected intensity, which is the conditional expectation of the intensity in the reference filtration, is deterministic between event times. This property facilitates the exact sampling of the

INFORMS holds copyright to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

point process by sequential thinning or the inverse transform scheme. The projection method extends the reach of these classical schemes beyond their traditional domain of application.

We exemplify the projection method for a point process whose arrival intensity is a function of a one-dimensional jump-diffusion process and the point process itself. This formulation covers a broad family of self-exciting point processes with applications in finance, insurance, queuing, health care, and other areas. Point process filtering arguments are used to devise a recursive scheme for the computation of the conditional distribution of the driving jump-diffusion process given the point process path. The conditional distribution leads to the projected intensity, and also allows us to generate exact samples of the jump-diffusion state given the path of the point process. We therefore obtain an unbiased estimator of the expectation of a function of both a point process path and a skeleton of the driving jump-diffusion process. Numerical experiments demonstrate the effectiveness of our method relative to a conventional discretization scheme for the diffusive component of the driving process and a time-scaling scheme for the jump component. The projection method achieves the optimal square-root convergence, whereas the convergence of the discretization scheme is much slower. In addition, our method generates the estimator with the smallest error, for any given computational budget.

Our jump-diffusion scheme extends a scheme for a jump-diffusion process with deterministic jump intensity described by Glasserman (2004, §3.5). If the intensity is deterministic, then the jump times can be generated independently of the diffusion component. A skeleton of the diffusion component can be generated by the exact scheme of Beskos and Roberts (2005); see also Chen (2009). Our scheme is also related to the exact scheme developed by Casella and Roberts (2011) for a one-dimensional jump diffusion with state-dependent jump intensity that is almost surely bounded from above. The boundedness property allows Casella and Roberts (2011) to generate the jump times by state-dependent thinning as in Glasserman and Merener (2003). The projection method does not require the intensity to be bounded. However, its practical implementation requires more structure on the coefficients of the jump-diffusion process.

Giesecke et al. (2010) develop an exact scheme for a vector of indicator—i.e., one-jump—point processes with stochastic intensities. They construct an auxiliary Markov chain whose value at a fixed time has the same distribution as the value of the indicator point process at that time, and then estimate the point process expectation of interest by sampling the mimicking chain. The approach of Giesecke et al. (2010) targets a multivariate setting with correlated component point processes, where the quantity of interest is an expectation of a function of the value of the vector point process at a fixed time. The approach proposed in this paper targets a one-dimensional setting where the quantity of interest is an expectation of a function of the *path* of the point process, and possibly a skeleton of the driving state process.

The remainder of this paper is organized as follows. Section 2 formulates the problem, outlines conventional simulation approaches, and motivates the projection method. Section 3 develops the projection method. Section 4 exemplifies the method for a point process whose arrival intensity is driven by a jump-diffusion process. Section 5 provides numerical results. Section 6 concludes. An electronic companion to this paper is available as part of the online version that can be found at http://or.journal.informs.org/.

2. Point Process Simulation

2.1. Point Process

Fix a complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and a right-continuous and complete information filtration $\mathbb{F} = (\mathcal{F}_t)_{t\geqslant 0}$. Consider a sequence $(T_n)_{n\geqslant 0}$ of stopping times that is strictly increasing to ∞ and for which $T_0 = 0$, almost surely. These stopping times represent the ordered arrival times of events such as bankruptcies, equipment failures, claim arrivals, or disease incidences. Let N be the counting process given by

$$N_t = \sum_{n\geqslant 1} 1_{\{T_n \leqslant t\}}. \tag{1}$$

The Doob-Meyer theorem guarantees the existence of a nondecreasing process A starting at 0 such that $M = N - A$ is a local martingale. The compensator A is uniquely defined up to indistinguishability and governs the distribution of N. We assume that there is a nonnegative, integrable intensity process λ that satisfies

$$A_t = \int_0^t \lambda_s \, ds \tag{2}$$

almost surely, for every t. The process λ is the density of the random measure associated with A relative to Lebesgue measure. It represents the mean arrival rate of events relative to $\mathbb{F}$ in the sense that $\mathbb{E}(N_{t+\Delta} - N_t \mid \mathcal{F}_t) \approx \lambda_t \Delta$ for "small" $\Delta > 0$.

Let $(\ell_n)_{n\geqslant 0}$ be a sequence of real-valued, $\mathcal{F}_{T_n}$-measurable random variables with $\ell_0 = 0$. The "mark" ℓ_n encodes additional information that is revealed at the event time T_n. For example, if the T_n model insurance claim arrivals, then the ℓ_n could describe the claim sizes. In portfolio credit risk applications, the T_n represent default times and the ℓ_n could model the loss due to default. Define the point process L by

$$L_t = \sum_{n=0}^{N_t} \ell_n = \sum_{n\geqslant 0} \ell_n 1_{\{T_n \leqslant t\}}. \tag{3}$$

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

2.2. Conventional Simulation Schemes

Our goal is to develop an unbiased simulation estimator of the expectation

$$\mathbb{E}(f((L_s)_{s\leqslant t})) \tag{4}$$

for suitable functions f and fixed $t > 0$. In portfolio credit risk applications, for example, the expectation (4) could represent the present value of a derivative security that pays at t a specified function of the total default loss in a reference portfolio of defaultable assets such as loans or bonds. To estimate (4) by simulation, we need to generate a trajectory of L over $[0, t]$, i.e., the pairs $(T_n, \ell_n)_{n\leqslant N_t}$. We focus on the generation of the T_n.

If the intensity λ_t is a deterministic function of time t, then N is a nonhomogeneous Poisson process and the arrival times can be generated by the inverse method from the interarrival time distribution, the order statistics property of the Poisson process, or the thinning scheme of Lewis and Shedler (1979). Although fundamental, deterministic intensity models are often too simplistic. Many applications require state-dependent intensities. Here, λ follows an $\mathbb{F}$-adapted stochastic process. The value λ_t may depend on the path $(L_s)_{s\leqslant t}$, i.e., past arrival times and marks $(T_n, \ell_n)_{n\leqslant N_t}$, and other sources of randomness recorded by the sigma field $\mathcal{F}_t$. In the special case where λ is adapted to the filtration generated by L, i.e., if λ_t is a function of $(T_n, \ell_n)_{n\leqslant N_t}$ only, the interarrival intensity is deterministic, and the interarrival times can be generated sequentially by the inverse method from the conditional distribution of the interarrival time, or, preferably, thinning. Examples include the birth and Hawkes processes, see Ogata (1981).

In the general case, λ is modulated by additional random factors that follow stochastic processes on their own. In this case, λ is not adapted to the filtration generated by L and does not evolve deterministically between arrivals. Then, the inverse method is feasible only if the inverse of the conditional distribution function of an interarrival time can be evaluated and the intensity value at an arrival time can be simulated. The thinning method applies only if the interarrival intensity can be bounded from above and the values of the intensity at the candidate times can be simulated.

In many model specifications of interest, the interarrival intensity is not bounded, or it is difficult to sample the values of the intensity at an arrival time. A simulation method with wider scope exploits a time-change result for point processes due to Meyer (1971). Suppose the compensator A increases to ∞ almost surely. Meyer proved that N is a standard Poisson process under a change of time defined by A, relative to the time-changed filtration. Thus, for a sequence $(\mathcal{E}_n)$ of standard exponential random variables that are mutually independent, the arrival time

$$T_n \stackrel{d}{=} \inf\left\{t\colon \int_0^t \lambda_s\, ds \geqslant \mathcal{E}_1 + \cdots + \mathcal{E}_n\right\}. \tag{5}$$

This provides a recipe for simulating N: draw $\mathcal{E}_1$ from a standard exponential distribution, generate a path of the integral $\int_0^t \lambda_s\, ds$, and record its hitting time to $\mathcal{E}_1$. Then draw $\mathcal{E}_2$, continue to generate a path of the integral $\int_0^t \lambda_s\, ds$, and record the hitting time to $\mathcal{E}_1 + \mathcal{E}_2$. Continue until the simulation horizon is reached.

Although this time-scaling scheme is widely applicable, it often leads to biased simulation estimators of (4). This is because it is usually not possible to generate the complete path of the continuous-time stochastic process $\int_0^t \lambda_s\, ds$. Often, the path must be approximated on a discrete-time grid. Moreover, it may not be possible to draw exact samples of the values of $\int_0^t \lambda_s\, ds$ at the grid points, because the required conditional distribution may not be known or computationally tractable. This forces one to approximate the values of $\int_0^t \lambda_s\, ds$ on the discrete-time grid by first approximating the continuous-time intensity process λ on the grid, and then integrating the discretized values. If the intensity values cannot be sampled exactly, then the stochastic differential equation (SDE) that describes the intensity dynamics must be discretized by the Euler or other schemes, introducing another source of bias.

Due to the multiple layers of approximations in the time-scaling scheme, it is difficult to quantify the magnitude of the discretization bias in the estimator of (4). This, in turn, makes it hard to obtain valid confidence intervals. The bias can be reduced by increasing the number of discretization time steps, but this also increases the computational cost of a replication. Reducing the bias to an acceptable level may require a prohibitively large computational effort. Even more computational effort may be required to verify that the bias is small enough. Finally, the optimal allocation of resources between the number of time steps and the number of trials is difficult to specify.

2.3. Projection Method: Illustrative Example

We propose a method for sampling N that eliminates the need to discretize the process $\int_0^t \lambda_s\, ds$. This method can lead to unbiased simulation estimators of (4). It does not require the intensity to be bounded almost surely. Before explaining this method in full generality, we discuss an elementary example to illustrate the main idea.

Let X be a random variable and g be a positive, integrable function. Suppose the intensity takes the form $\lambda_t = g(X)$. That is, the intensity is modulated by the variable X, which could represent the state of the random environment. The point process N is a mixed Poisson process: conditional on a realization of X, N is a Poisson process with (conditionally deterministic) rate $g(X)$. Thus, one could simulate N by first drawing a sample $X(\omega)$ from the distribution of X, and then generating Poisson arrivals according to the rate $g(X(\omega))$. This is feasible whenever the values of X can be sampled. If these samples are exact, then N can also be sampled exactly.

The projection method seeks to avoid the sampling of X altogether: exact sampling of X may not be feasible or

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

may be too costly. The idea is to smooth out the randomness associated with the intensity $\lambda_t = g(X)$, which comes from X, *before* the arrivals of N are generated. We cannot, however, simply generate arrivals from a Poisson process with rate $\mathbb{E}(g(X))$. We must generate the arrival times (T_n) of N sequentially and account for the path of N as we proceed. More precisely, in order to generate T_1, we sample the first jump time of a (time-inhomogenous) Poisson process with rate $h_0(t) = \mathbb{E}(g(X) \mid N_t = 0)$. This can be done by the inverse transform method, or thinning. We obtain an exact sample of T_1 if the projection $h_0(t)$ can be computed exactly. In order to generate the waiting time $T_2 - T_1$ given T_1, we sample the first jump time of a Poisson process started at T_1 with rate $h_1(t) = \mathbb{E}(g(X) \mid T_1, N_t = 1)$ for $t \geqslant T_1$, and so on.

To demonstrate the correctness of this scheme, we express the distribution of T_1 in terms of the function $h_0(t)$. A similar argument applies to the other times. Because N is conditionally Poisson, $\mathbb{P}(T_1 > t \mid X) = \mathbb{P}(N_t = 0 \mid X) = \exp(-g(X)t)$, so by Bayes' formula and iterated expectations,

$$h_0(t) = \frac{\mathbb{E}(g(X)1_{\{N_t=0\}})}{\mathbb{P}(N_t = 0)} = \frac{\mathbb{E}(g(X)\exp(-g(X)t))}{\mathbb{E}(\exp(-g(X)t))} = \frac{-\partial_t \mathbb{P}(T_1 > t)}{\mathbb{P}(T_1 > t)}. \tag{6}$$

Thus, $h_0(t)$ is the hazard function of the random time T_1. Integrating, we get

$$\mathbb{P}(T_1 > t) = \exp\left(-\int_0^t h_0(s)\,ds\right), \tag{7}$$

showing that T_1 is equal in distribution to the first jump time of a Poisson process with rate function $h_0(t)$. The projection method suggests drawing T_1 as this jump time by applying the inverse scheme to (7) or the thinning scheme to $h_0(t)$.

3. Projection Method

This section explains the projection method in the general setting of §2.1. In the abstract, the method is based on a change of filtration. Instead of simulating L based on the given $\mathbb{F}$-intensity model λ, we *project* L onto a subfiltration of $\mathbb{F}$, and then sample the arrival times (T_n) based on the intensity relative to the subfiltration. Because the subfiltration is coarser than $\mathbb{F}$, the subfiltration intensity has simpler dynamics than the $\mathbb{F}$-intensity. We choose the subfiltration such that the subfiltration intensity is deterministic between arrivals. Then we can apply the inverse or thinning scheme relative to the subfiltration to sample the interarrival times. This can often be done exactly.

We project L onto its own right-continuous and complete filtration $\mathbb{G} = (\mathcal{G}_t)_{t\geqslant 0}$ generated by the sigma fields $\sigma(L_s\colon s \leqslant t)$. This is the smallest subfiltration of $\mathbb{F}$ that is compatible with L. Since the counting process N has an intensity λ relative to $\mathbb{F}$, it also has an intensity relative to $\mathbb{G}$. In analogy to the definition of the $\mathbb{F}$-intensity λ, the $\mathbb{G}$-intensity is a $\mathbb{G}$-adapted process h such that N has compensator $\int_0^t h_s\,ds$ relative to $\mathbb{G}$. That is, the process defined by $N_t - \int_0^t h_s\,ds$ is a $\mathbb{G}$-local martingale.

The $\mathbb{G}$-intensity h is determined by the $\mathbb{F}$-intensity λ and the subfiltration $\mathbb{G}$. It is given by the optional projection of λ onto the subfiltration $\mathbb{G}$. The optional projection is a $\mathbb{G}$-adapted process that is unique up to indistinguishability, see Dellacherie and Meyer (1982, Ch. IV, pp. 43–44). It satisfies

$$h_t = \mathbb{E}(\lambda_t \mid \mathcal{G}_t) \tag{8}$$

almost surely, for every t. Formula (8) indicates that if λ is $\mathbb{G}$-adapted as for Poisson, birth, Hawkes, and time-inhomogeneous Markov point processes, then $h = \lambda$. If λ is not a priori $\mathbb{G}$-adapted, as in the application cases we have in mind, then h is nontrivial.

Due to the special structure of the subfiltration $\mathbb{G}$, the projected intensity h evolves deterministically between events, and jumps at event times. Intuitively, the sources of randomness influencing λ beyond the arrival times T_n and jump sizes ℓ_n are smoothed out by the projection. More precisely, h takes the form

$$h_t = \sum_{n\geqslant 0} h_n(t) 1_{\{N_t=n\}} \tag{9}$$

almost surely, where h_n is the $\mathcal{G}_{T_n}$-measurable function defined by

$$h_n(t) = \frac{\mathbb{E}(\lambda_t 1_{\{N_t=n\}} \mid \mathcal{G}_{T_n})}{\mathbb{P}(N_t = n \mid \mathcal{G}_{T_n})} \tag{10}$$

for $t \geqslant T_n$ and $n \geqslant 0$. This is because on the set $\{N_t = n\}$, the sigma field $\mathcal{G}_t$ consists of $\mathcal{G}_{T_n}$ and the information that $N_t = n$. Now formula (9) is a consequence of Bayes' rule. The following observation explains the significance of the functions h_n.

Proposition 3.1. *Assume N has $\mathbb{F}$-intensity λ such that for all t*

$$\mathbb{E}\left(\int_0^t \lambda_s\,ds\right) < \infty. \tag{11}$$

Let h be the optional projection of λ onto the right-continous and complete filtration $\mathbb{G}$ generated by L and let h_n be the function satisfying (9)–(10). *Then, for $t \geqslant T_n$,*

$$\mathbb{P}(T_{n+1} > t \mid \mathcal{G}_{T_n}) = \exp\left(-\int_{T_n}^t h_n(s)\,ds\right). \tag{12}$$

Proof. First observe that

$$1_{\{N_t>n\}} = \sum_{0<s\leqslant t} \Delta 1_{\{N_s>n\}} = \int_0^t 1_{\{N_{s-}=n\}}\,dN_s \tag{13}$$

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

where $\Delta V_t = V_t - V_{t-}$ is the jump of a right-continuous left-limit process V at t and $V_{t-} = \lim_{s \uparrow t} V_s$. With the integrability hypothesis (11) on λ, the local $\mathbb{F}$-martingale $U = N - \int_0^{\cdot} \lambda_s \, ds$ is a true $\mathbb{F}$-martingale, and so is $H = \int_0^{\cdot} 1_{\{N_{s-}=n\}} \, dU_s$ because the integrand is bounded and predictable. Using this property, taking $\mathcal{F}_{T_n}$-conditional expectation on both sides of Equation (13) gives, for $t \geqslant T_n$,

$$\begin{aligned}\mathbb{P}(N_t > n \mid \mathcal{F}_{T_n}) &= \mathbb{E}\left(\int_{T_n}^t 1_{\{N_{s-}=n\}} (dU_s + \lambda_s \, ds) \,\middle|\, \mathcal{F}_{T_n} \right) \\ &= \mathbb{E}\left(\int_{T_n}^t 1_{\{N_s=n\}} \lambda_s \, ds \,\middle|\, \mathcal{F}_{T_n} \right).\end{aligned}$$

Then, by iterated expectations and Fubini's theorem, we get

$$\begin{aligned}\mathbb{P}(N_t > n \mid \mathcal{G}_{T_n}) &= \int_{T_n}^t \mathbb{E}(\lambda_s 1_{\{N_s=n\}} \mid \mathcal{G}_{T_n}) \, ds \\ &= \int_{T_n}^t h_n(s) \mathbb{P}(N_s = n \mid \mathcal{G}_{T_n}) \, ds,\end{aligned}$$

where the second equality follows from Definition (10). Now

$$\begin{aligned}-\partial_t \mathbb{P}(N_t = n \mid \mathcal{G}_{T_n}) &= \partial_t \mathbb{P}(N_t > n \mid \mathcal{G}_{T_n}) \\ &= h_n(t) \mathbb{P}(N_t = n \mid \mathcal{G}_{T_n}),\end{aligned}$$

which gives formula (12), noting that $\mathbb{P}(T_{n+1} > t \mid \mathcal{G}_{T_n}) = \mathbb{P}(N_t = n \mid \mathcal{G}_{T_n})$. □

Formula (12) expresses the conditional survival function of T_{n+1} given $\mathcal{G}_{T_n}$ in terms of the $\mathcal{G}_{T_n}$-measurable function $h_n(t)$ that represents the projected intensity on the event $\{N_t = n\}$. The formula demonstrates that given $\mathcal{G}_{T_n}$, the waiting time to next event date T_{n+1} is equal in distribution to the first jump time of an inhomogeneous $\mathbb{G}$-Poisson process started at T_n with intensity given by h_n.

The event times and marks can be generated sequentially in the filtration $\mathbb{G}$, based on the functions h_n. This requires a technical condition on the marks ℓ_n. Assume that the law of ℓ_n depends at most on the events in the sigma-field $\mathcal{G}_{T_n-}$, which is generated by the variables $(T_k, \ell_k)_{k<n}$ and T_n, see Brémaud (1980, Ch. III, Theorem T2). We start at $T_0 = 0$ and generate T_1 based on the function h_0. Next we generate the mark ℓ_1. Given the pair (T_1, ℓ_1), we then calculate h_1, draw T_2, and then subsequently ℓ_2. The procedure is continued until the simulation horizon is reached.

The inverse transform method can be used to generate a sample of T_{n+1} from (12) given $\mathcal{G}_{T_n}$. This, however, requires us to evaluate the inverse to (12), which may be computationally expensive, and which may require numerical approximations. Alternatively, we can use the thinning scheme of Lewis and Shedler (1979) to generate a sample of T_{n+1} given $\mathcal{G}_{T_n}$. This scheme requires us to evaluate $h_n(t)$ only at candidate arrival times t generated from a dominating process. The algorithm is detailed below. For clarity in the exposition, we do not include a termination condition.

ALGORITHM 3.2. *To generate the event time T_{n+1} given $(T_k, \ell_k)_{k \leqslant n}$,*

(1) *Initialize $t = T_n$.*

(2) *Find functions B_t^n and C_t^n such that $h_n(t+s) \leqslant B_t^n$ for $0 \leqslant s \leqslant C_t^n$.*

(3) *Draw $\mathcal{E} \sim \text{Exp}(B_t^n)$. If*

- *$\mathcal{E} > C_t^n$, then set $t = t + C_t^n$ and go to Step* (2).
- *$\mathcal{E} \leqslant C_t^n$, then draw $U \sim U(0,1)$. If $UB_t^n \leqslant h_n(t+\mathcal{E})$, then set $T_{n+1} = t + \mathcal{E}$ and stop. Else set $t = t + \mathcal{E}$ and go to Step* (2).

The specific properties of the function h_n at hand will determine the choice of the bound B_t^n and the interval length C_t^n for which it is valid. If, for example, h_n is increasing, then B_t^n can be taken as the interval end point $h_n(t + C_t^n)$ for some C_t^n. The optimal value of C_t^n depends on the slope of h_n: intuitively, the steeper h_n, the smaller C_t^n should be. If h_n is decreasing, one can take $B_t^n = h_n(t)$ and C_t^n equal to the simulation horizon. An adaptive choice of the bound may improve efficiency. For example, the bound may be reset at each candidate time generated. This is especially meaningful if h_n is decreasing: at each candidate time, the bound for generating the next time can be set to the value of h_n at the candidate time, which is computed in any case for the acceptance test.

The thinning scheme requires that h_n is piecewise bounded from above. Note that this requirement is much weaker than the boundedness of the $\mathbb{F}$-intensity λ between arrivals. The boundedness of λ is required for the application of the thinning scheme in the reference filtration $\mathbb{F}$. This property often fails when λ is governed by an SDE. Moreover, the application of the thinning scheme to the projected intensity h eliminates the need to generate samples of the intensity λ at a candidate time, which can be expensive when λ is driven by an SDE. After projection, the thinning scheme requires only the evaluation of deterministic functions at a finite set of points.

The projection Algorithm 3.2 facilitates the sampling of the point process L whenever the projected intensity h can be calculated for the given intensity λ. The associated estimator of (4) is theoretically unbiased. If h can be computed exactly, then the estimator is also unbiased in the practical implementation.

4. Point Process Driven by a Jump Diffusion

The projection method can potentially be applied to a point process with arbitrary stochastic intensity dynamics. Its practical implementation requires the calculation of the projected intensity h for the intensity model λ at hand. This section develops a recursive scheme for the efficient calculation of h for an important family of point processes. It focuses on a point process L whose arrival intensity λ_t is a time-dependent function of the value at t of a jump-diffusion process and L_t. When applied to this setting, the projection method also facilitates the sampling of the

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

driving jump-diffusion process. This, in turn, leads to an estimator of the expectation of a function of both a point process path and a skeleton of the driving jump-diffusion process.

4.1. Specification

Suppose that X is a real-valued Markov process that solves the SDE

$$dX_t = \mu(X_t)\,dt + \sigma(X_t)dW_t + \delta dL_t, \tag{14}$$

where X_0 has some fixed distribution on $\mathbb{R}$ such that $\mathbb{E}(X_0^2) < \infty$, W is a standard Brownian motion relative to the filtration $\mathbb{F}$, μ, and σ are suitable real-valued functions that specify the drift and variance of X, respectively, $\delta \geqslant 0$ is a sensitivity parameter, and $L = \ell_0 + \cdots + \ell_N$. The jump size ℓ_n is measurable with respect to $\mathcal{G}_{T_n^-}$ and takes values in a discrete subset of $\mathbb{R}$. It is drawn from a distribution that may depend on T_n. The intensity λ is given by a nonnegative function Λ on $\mathbb{R}_+ \times \mathbb{R} \times \mathbb{R}_+$:

$$\lambda_t = \Lambda(t, X_t, L_t). \tag{15}$$

The drift, variance, and intensity functions, and the jump-size distribution are assumed to satisfy conditions guaranteeing that the solution to the SDE (14) exists in a strong sense and is unique in a weak sense. Such conditions are stated in Ceci and Gerardi (2006, Appendix A). They involve, in particular, the requirement that $\mathbb{E}(\int_0^t \Lambda(s, X_s, L_s)\,ds)$ is finite for all t, which is also a prerequisite for Proposition 3.1.

The specification (14)–(15) generates a broad class of point processes with features that are important in many application areas. For example, a point process in this class is self-exciting, because it has an intensity λ that responds to events. In our specification, this occurs because L influences the state X, which drives λ. The process L also directly influences λ through the function (15). Thus, there is also a dependence structure between λ and the jump sizes ℓ_n. Point processes with these features play an important role in finance. They can describe the arrival of security orders in an order book as in Bowsher (2007), Cont et al. (2010), and others. Here, the self-exciting property models the clustering of order arrivals. They can also describe the timing of corporate defaults as in Arnsdorf and Halperin (2008), Ding et al. (2009), and others. Here, the process X represents the state of the economy, and the self-exciting property models the contagious impact of a default event on the surviving firms. If the ℓ_n model the losses due to default, then there is a dependence structure between default and loss rates, consistent with empirical observation. In yet another setting, the state process X could model the price of the stock of a company that defaults at the first jump of N with an intensity λ that is a function of the stock price, as in Carr and Linetsky (2006).

In the special case that the coefficient functions $\mu(x)$ and $\sigma^2(x)$ are affine in x and the intensity function $\Lambda(t, x, l)$ is affine in (x, l), the process (X, L) is a two-dimensional affine jump diffusion. In this case the transform of (X, L) can be obtained from the results in Duffie et al. (2000). The projection method extends beyond this affine jump-diffusion case and allows us to treat specifications of (X, L) whose transforms are not analytically tractable.[1] Further, the estimators generated by the projection method apply to functionals of a complete point process path and a skeleton of X. These functionals are often difficult to treat with analytical methods.

4.2. Intensity Projection

The projection method requires the calculation of the intensity h of N with respect to the filtration $\mathbb{G}$ generated by L. This is a filtering problem for X given observations of L. From Equations (8) and (15), because L is $\mathbb{G}$-adapted,

$$h_t = \mathbb{E}(\Lambda(t, X_t, L_t) \mid \mathcal{G}_t) = \int_{\mathbb{R}} \Lambda(t, x, L_t)\,\pi_t(dx) \tag{16}$$

almost surely, where $\pi_t(dx)$ is the conditional distribution of X_t given $\mathcal{G}_t$. That distribution is encoded in the transform

$$\begin{aligned} M_t(z, u) &= \mathbb{E}(\exp(-zX_t - uX_t^2) \mid \mathcal{G}_t) \\ &= \int_{\mathbb{R}} \exp(-zx - ux^2)\,\pi_t(dx), \end{aligned} \tag{17}$$

which is defined for all those $z, u \in \mathbb{R}$ for which the expectation is finite. We first provide a recursive scheme for calculating $M_t(z, u)$, and then show how this scheme leads to recursive, semianalytical formulas for h_t for several cases of interest. These formulas then facilitate the application of the projection Algorithm 3.2.

Owing to the structure of the filtration $\mathbb{G}$, the transform $M_t(z, u)$ evolves deterministically between events and is updated at events. In our setting, the observation process L and the state process X have common jumps, and the point process filtering results in Kliemann et al. (1990) imply that $M_t(z, u)$ is the weakly unique solution to the Kushner-Stratonovich equation that describes the time evolution of the filter $\mathbb{E}(f(t, X_t) \mid \mathcal{G}_t)$ for suitable functions $f(t, x)$. In our case, $f(t, x) = \exp(-zx - ux^2)$. The Kushner-Stratonovich equation can be solved piecewise. We obtain a characterization of the filter for the interval $[T_n, T_{n+1})$ and an updating map for the event time T_{n+1}. We examine these cases separately, specializing in the results of Kliemann et al. (1990) and especially Ceci and Gerardi (2006), who provide a more explicit discussion. The analysis leads to a recursive scheme for calculating $M_t(z, u)$.

Let Y be a (weakly unique) solution to the SDE

$$Y_t = x + \int_s^t \mu(Y_u)\,du + \int_s^t \sigma(Y_u)\,dW_u \tag{18}$$

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

for any initial condition (s, x), where the coefficient functions μ and σ are those of the SDE (14). For $s \leqslant t$, real x, v, u and nonnegative l, let

$$\varphi_t(s,x,v,u,l) = \mathbb{E}\left[\exp\left(-\int_s^t \Lambda(w, Y_w, l)\,dw\right)e^{-vY_t-uY_t^2}\,\middle|\, Y_s = x\right] \quad (19)$$

whenever the expectation is finite. Lemmas 3.1 and 3.2 and Proposition 3.2 of Ceci and Gerardi (2006) imply that

$$M_t(z,u) = \frac{\rho_t^n(z,u)}{\rho_t^n(0,0)} \quad \text{on } \{N_t = n\} \quad (20)$$

where for any v and u

$$\rho_t^n(v,u) = \int_{\mathbb{R}} \varphi_t(T_n, x, v, u, L_{T_n})\pi_{T_n}(dx). \quad (21)$$

The filter $M_t(z,u)$ jumps at event times. Proposition 3.1 of Ceci and Gerardi (2006) implies that at time T_{n+1}, the filter is given by

$$M_{T_{n+1}}(z,u) = J_{n+1}(z,u)\frac{\mathbb{E}(\lambda_{T_{n+1}^-}\exp[-(z+2u\delta\ell_{n+1})X_{T_{n+1}^-} - uX_{T_{n+1}^-}^2]\,|\,\mathcal{G}_{T_{n+1}^-})}{\mathbb{E}(\lambda_{T_{n+1}^-}\,|\,\mathcal{G}_{T_{n+1}^-})}, \quad (22)$$

where for any z and u

$$J_{n+1}(z,u) = \exp(-z\delta\ell_{n+1} - u\delta^2\ell_{n+1}^2). \quad (23)$$

It is important to note that the transform (22) at the jump time T_{n+1} is governed by the transform (20), which is valid for times $t \in [T_n, T_{n+1})$. This leads to a *recursive scheme* for calculating $M_t(z,u)$, starting with $M_0(z,u)$. Below we show how this scheme can be used to obtain recursive, semianalytical formulas for the projected intensity h in two model specifications of interest.

4.2.1. Explicit Formulas: Affine Specification. Suppose the drift and variance functions of the state process X satisfy:

$$\mu(x) = K_0 + K_1 x, \quad \sigma(x)^2 = H_0 + H_1 x \quad (24)$$

for constant coefficients (K_i, H_i). In this case, the auxiliary process Y is an affine diffusion process. Suppose further that the intensity function $\Lambda(t,x,l)$ is affine in x,

$$\Lambda(t,x,l) = \Lambda_0(t,l) + \Lambda_1(t,l)x \quad (25)$$

for nonnegative functions $\Lambda_i(t,l)$ on $\mathbb{R}_+ \times \mathbb{R}$ such that $\mathbb{E}(\int_0^t \Lambda(s, X_s, L_s)\,ds) < \infty$. Proposition 1 in Duffie et al. (2000) gives conditions such that

$$\varphi_t(s,x,v,0,l) = \exp(a(s,t,v,l) - b(s,t,v,l)x) \quad (26)$$

where the coefficient functions satisfy the ordinary differential equations (ODEs)

$$\partial_s b(s,t,v,l) = -\Lambda_1(s,l) - K_1 b(s,t,v,l) + \tfrac{1}{2}H_1 b(s,t,v,l)^2 \quad (27)$$

$$\partial_s a(s,t,v,l) = \Lambda_0(s,l) + K_0 b(s,t,v,l) - \tfrac{1}{2}H_0 b(s,t,v,l)^2 \quad (28)$$

with boundary conditions

$$b(t,t,v,l) = v \text{ and } a(t,t,v,l) = 0.$$

These ODEs can be solved explicitly for certain choices of the coefficient parameters in (24) and (25). Analytically solvable cases include the case where each $\Lambda_i(t,l)$ is independent of t, combined with either $H_0 = 0$ or $H_1 = 0$. If analytical solutions are not available, a numerical scheme such as Runge-Kutta quickly yields numerical solutions.

The specification (24)–(25) generates a jump-diffusion state process X with affine drift and variance functions. The arrival intensity of the jumps in X is affine in X but has arbitrary dependence on L. This specification nests the class of affine jump-diffusion processes analyzed by Duffie et al. (2000). Affine jump-diffusion processes are analytically tractable but require that the arrival intensity function $\Lambda(t,x,l)$ has affine dependence on (x,l). The projection method allows us to treat applications based on the more-general specification (24)–(25) by Monte Carlo simulation.

Noting that

$$\rho_t^n(v,0) = \exp(a(T_n,t,v,L_{T_n}))\int_{\mathbb{R}}\exp(-b(T_n,t,v,L_{T_n})x)\pi_{T_n}(dx) = \exp(a(T_n,t,v,L_{T_n}))M_{T_n}(b(T_n,t,v,L_{T_n}),0),$$

from formula (20) we find that

$$M_t(z,0) = \frac{\exp(a(T_n,t,z,L_{T_n}))M_{T_n}(b(T_n,t,z,L_{T_n}),0)}{\exp(a(T_n,t,0,L_{T_n}))M_{T_n}(b(T_n,t,0,L_{T_n}),0)} \quad (29)$$

on the set $\{N_t = n\}$. Thus, we obtain a recursive scheme for the calculation of the functions $h_n(t)$ that define the projected intensity h between events:

- At $T_0 = 0$, compute $M_0(z,0) = \mathbb{E}(\exp(-zX_0))$.
- For $n = 0, 1, 2, \ldots$ and $t \in [T_n, T_{n+1})$, compute

$$h_n(t) = \Lambda_0(t,L_t) - \Lambda_1(t,L_t)\partial_z M_t(z,0)|_{z=0} = \Lambda_0(t,L_t) - \Lambda_1(t,L_t) \cdot \frac{\partial_z \exp(a(T_n,t,z,L_{T_n}))M_{T_n}(b(T_n,t,z,L_{T_n}),0)|_{z=0}}{\exp(a(T_n,t,0,L_{T_n}))M_{T_n}(b(T_n,t,0,L_{T_n}),0)}.$$

- Recursion, at T_{n+1} compute

$$M_{T_{n+1}}(z,0) = J_{n+1}(z,0)\frac{\Lambda_0(T_{n+1}^-,L_{T_n})M_{T_{n+1}^-}(z,0) - \Lambda_1(T_{n+1}^-,L_{T_n})\partial_z M_{T_{n+1}^-}(z,0)}{\Lambda_0(T_{n+1}^-,L_{T_n}) - \Lambda_1(T_{n+1}^-,L_{T_n})\partial_z M_{T_{n+1}^-}(z,0)|_{z=0}}$$

where $M_{T_{n+1}^-}(z,0) = \lim_{t\uparrow T_{n+1}} M_t(z,0)$ is obtained from (29).

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

This recursive scheme for the calculation of h_n facilitates the simulation of a point process L specified by (14)–(15) and (24)–(25) via the projection Algorithm 3.2. The simulation estimator is unbiased if h_n can be computed exactly, i.e., without the use of numerical approximations. This is often possible if the ODEs (27)–(28) admit closed-form solutions. We shall discuss this issue further in §4.2.3 below.

4.2.2. Explicit Formulas: Quadratic Specification. Suppose the drift and variance functions of the state X satisfy:

$$\mu(x) = K_0 + K_1 x, \quad \sigma(x)^2 = H_0 \tag{30}$$

for constant coefficients K_i and H_0. Then, the auxiliary process Y is an Ornstein-Uhlenbeck process. Moreover, assume that $\Lambda(t, x, l)$ has quadratic dependence on the state x,

$$\Lambda(t, x, l) = \Lambda_0(l) + \Lambda_1(l)x + \Lambda_2(l)x^2 \tag{31}$$

for functions $\Lambda_i(l)$ on $\mathbb{R}_+$ such that $\mathbb{E}(\int_0^t \Lambda(s, X_s, L_s)\, ds) < \infty$. Leippold and Wu (2002) and Chen et al. (2004) provide conditions guaranteeing that

$$\varphi_t(s, x, v, u, l) = \exp(a(s, t, v, u, l) - b(s, t, v, u, l)x - c(s, t, v, u, l)x^2), \tag{32}$$

where the coefficient functions satisfy the ODEs

$$\partial_s c(s, t, v, u, l) = -\Lambda_2(l) - 2K_1 c(s, t, v, u, l) + 2H_0 c(s, t, v, u, l)^2 \tag{33}$$

$$\partial_s b(s, t, v, u, l) = -\Lambda_1(l) - 2K_0 c(s, t, v, u, l) - K_1 b(s, t, v, u, l) + 2H_0 b(s, t, v, u, l) c(s, t, v, u, l) \tag{34}$$

$$\partial_s a(s, t, v, u, l) = \Lambda_0(l) + K_0 b(s, t, v, u, l) + H_0 c(s, t, v, u, l) - \tfrac{1}{2} H_0 b(s, t, v, u, l)^2 \tag{35}$$

with boundary conditions

$$c(t, t, v, u, l) = u, \quad b(t, t, v, u, l) = v, \text{ and}$$
$$a(t, t, v, u, l) = 0.$$

Chen et al. (2004) provide closed-form solutions to these ODEs.

The specification (30)–(31) generates a jump-diffusion state process X with affine drift and variance functions. The arrival intensity of the jumps in X is quadratic in X but has arbitrary dependence on L. This specification extends the class of Ornstein-Uhlenbeck diffusion processes, whose transforms are analyzed by Leippold and Wu (2002) and Chen et al. (2004), to include a state-dependent jump term. The projection method allows us to treat applications based on these jump-diffusion processes by simulation.

We proceed as in §4.2.1 to obtain

$$\rho_t^n(v, u) = \exp(a(T_n, t, v, u, L_{T_n}))M_{T_n}(b(T_n, t, v, u, L_{T_n}), c(T_n, t, v, u, L_{T_n})).$$

From formula (20), we then find that

$$M_t(z, u) = \frac{\exp(a(T_n, t, z, u, L_{T_n}))M_{T_n}(b(T_n, t, z, u, L_{T_n}), c(T_n, t, z, u, L_{T_n}))}{\exp(a(T_n, t, 0, 0, L_{T_n}))M_{T_n}(b(T_n, t, 0, 0, L_{T_n}), c(T_n, t, 0, 0, L_{T_n}))} \tag{36}$$

on the set $\{N_t = n\}$. Thus, we obtain a recursive scheme for the calculation of the functions $h_n(t)$ that define the projected intensity h between events:

- At $T_0 = 0$, compute $M_0(z, u) = \mathbb{E}(\exp(-zX_0 - uX_0^2))$.
- For $n = 0, 1, 2, \ldots$ and $t \in [T_n, T_{n+1})$, compute

$$h_n(t) = \Lambda_0(L_t) - \Lambda_1(L_t)\partial_z M_t(z, 0)|_{z=0} - \Lambda_2(L_t)\partial_u M_t(0, u)|_{u=0}.$$

- Recursion, at T_{n+1} compute

$$M_{T_{n+1}}(z, u) = \frac{J_{n+1}(z, u)A_{T_{n+1}^-}(z, u)}{\Lambda_0(L_{T_n}) - \Lambda_1(L_{T_n})\partial_z M_{T_{n+1}^-}(z, 0)|_{z=0} - \Lambda_2(L_{T_n})\partial_u M_{T_{n+1}^-}(0, u)|_{u=0}}$$

where

$$A_{T_{n+1}^-}(z, u) = \Lambda_0(L_{T_n})M_{T_{n+1}^-}(z + 2u\delta\ell_{n+1}, u) - \Lambda_1(L_{T_n})\partial_v M_{T_{n+1}^-}(v, u)|_{v=z+2u\delta\ell_{n+1}} - \Lambda_2(L_{T_n})\partial_w M_{T_{n+1}^-}(z + 2u\delta\ell_{n+1}, w)|_{w=u}$$

and where $M_{T_{n+1}^-}(z, u)$ is obtained from (36).

This recursive scheme for the calculation of h_n facilitates the simulation of a point process L defined by (14)–(15) and (30)–(31) via the projection Algorithm 3.2. Because the ODEs (33)–(35) admit closed-form solutions, the calculation of h_n is analytical.

4.2.3. The Role of Numerical Approximations. Although Algorithm 3.2 is theoretically exact, its practical implementation generates an unbiased simulation estimator only if h_n can be computed exactly, i.e., without the use of numerical approximations. In the context of the affine and quadratic formulations discussed above, there are two potential sources of approximation bias. First, if the ODEs governing the transform (19) do not admit closed-form solutions, then the numerical solution may introduce bias. This is not an issue for the quadratic formulation, but may be an issue for some affine specifications. If the derivatives of (20) appearing in the recursive calculation

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

of h_n are taken numerically rather than analytically, then one has another source of bias. This is relevant when the ODEs governing (19) are solved numerically, because one is then forced to numerically approximate the derivatives. The magnitude of the total approximation bias depends on the quality of the numerical approximations, the specific parameter values, and the form of the expectation (4) being estimated.

To evaluate the bias and the additional computational cost associated with the numerical approximations, in §5 we compare an estimator based on a numerical approximation of h_n with an estimator based on the exact evaluation of h_n. For the affine example specification and example functional (4) we consider, the total approximation bias is negligible in that it does not degrade the convergence of the method. Although the numerical approximations increase the computational costs, the associated projection scheme is still much faster than a conventional discretization scheme.

4.3. Sampling the Jump-Diffusion State

In the context of the jump-diffusion formulation of this section, the projection method can also be used to estimate expectations of functionals of the point process and the driving jump diffusion. In the abstract, one can obtain the conditional distribution $\pi_t(dx)$ of X_t given $\mathcal{G}_t$ by inverting the Laplace transform $M_t(z,0)$. One can draw from this distribution using the inverse or other methods to get exact samples of the value X_t of the jump-diffusion process (14) given the associated point process trajectory $(L_s)_{s\leqslant t}$. This in turn leads to an estimator of $\mathbb{E}(g((L_s)_{s\leqslant t},(X_s)_{s\in S}))$ for suitable functions g, where S is a discrete set of times in the interval $[0,t]$. In practice, a numerical inversion of $M_t(z,0)$ is often unnecessary, and the implementation is much simpler. To see this, suppose $g((L_s)_{s\leqslant t},(X_s)_{s\in S})$ takes the form $g_1((L_s)_{s\leqslant t})g_2(X_t)$. Then,

$$\mathbb{E}(g((L_s)_{s\leqslant t},(X_s)_{s\in S}))=\mathbb{E}(g_1((L_s)_{s\leqslant t})\mathbb{E}(g_2(X_t)\mid\mathcal{G}_t)) \quad (37)$$

by iterated expectations, because L is $\mathbb{G}$-adapted. For example, if g_2 is polynomial with coefficients p_k, then the estimator of (37) is

$$g_1((L_s)_{s\leqslant t})\sum_k p_k(-1)^k\partial_z^k M_t(z,0)|_{z=0}. \quad (38)$$

Note that M and its first derivative are also required for the calculation of the projected intensity and, hence, the sampling of L. Assuming L can be sampled exactly (see §4.2.3), the estimator (38) will be unbiased if M_t and its partial derivatives can be evaluated exactly. For an example, see §5 below.

The polynomial formulation is often relevant. An example from finance, developed further below, is $g_1((L_s)_{s\leqslant t})=1_{\{L_t\leqslant K\}}$ for $K\geqslant 0$ and $g_2(X_t)=X_t$. Consider a security paying X_t at t if the issuer survives to t, and 0 otherwise. When taken under a pricing measure, the expectation $\mathbb{E}(X_t 1_{\{L_t\leqslant 0\}})$ represents the (undiscounted) value of this security, assuming that the issuer defaults at intensity (15). See Duffie et al. (1996) for a general treatment of these types of valuation problems. Carr and Linetsky (2006) analyze a concrete application to derivatives written on defaultable stocks when the intensity is a decreasing function of the stock price.

5. Numerical Results

We illustrate the effectiveness and relative performance of the projection method through numerical experiments. We specialize into the affine formulation of §4.2.1.

5.1. Specification

Suppose the coefficient functions (24) of the jump-diffusion state process (14) satisfy

$$(K_0,K_1,H_0,H_1)=(\kappa c,-\kappa,0,\sigma^2) \quad (39)$$

for parameters $\kappa\geqslant 0$, $c>0$, and $\sigma\geqslant 0$ such that $2\kappa c\geqslant\sigma^2$. The initial value X_0 has a fixed distribution on $\mathbb{R}_+$ that is specified later. Then the state process X solves

$$dX_t=\kappa(c-X_t)dt+\sigma\sqrt{X_t}dW_t+\delta dL_t. \quad (40)$$

For the intensity function (25), we take

$$\Lambda(t,x,l)=x\log(\theta+l),\quad \theta>1. \quad (41)$$

The intensity is a nonlinear function of the point process, and therefore the vector (X,L) is not an affine jump-diffusion process in the sense of Duffie et al. (2000). The transform of (X,L) does not take a computationally tractable form.

Because the interarrival intensity is not bounded from above by a constant, (X,L) falls outside the scope of the exact scheme of Casella and Roberts (2011) and the discretization scheme of Glasserman and Merener (2003).

5.2. Projected Intensity

With the parameterization (39), the ODEs (27)–(28) take the convenient form

$$\partial_s b(s,t,v,l)=\kappa b(s,t,v,l)+\tfrac{1}{2}\sigma^2 b(s,t,v,l)^2-\log(\theta+l) \quad (42)$$

$$\partial_s a(s,t,v,l)=\kappa c b(s,t,v,l) \quad (43)$$

with boundary conditions $b(t,t,v,l)=v$ and $a(t,t,v,l)=0$. Let $\gamma=(\kappa^2+2\sigma^2\log(\theta+l))^{1/2}$ and $c(s,t)=\exp(\gamma(t-s))$. The solutions are

$$b(s,t,v,l)=\frac{v(\gamma+\kappa+(\gamma-\kappa)c(s,t))+2(c(s,t)-1)\log(\theta+l)}{v\sigma^2(c(s,t)-1)+\gamma-\kappa+(\gamma+\kappa)c(s,t)} \quad (44)$$

$$a(s,t,v,l)=\frac{2\kappa c}{\sigma^2}\log\frac{2\gamma\exp((t-s)(\gamma+\kappa)/2)}{v\sigma^2(c(s,t)-1)+\gamma-\kappa+(\gamma+\kappa)c(s,t)}. \quad (45)$$

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

The analysis in §4.2.1 leads to an explicit recursive scheme to compute the functions $h_n(t)$ required by the projection Algorithm 3.2:

- At $T_0 = 0$ we have $M_0(z, 0) = \mathbb{E}(\exp(-zX_0))$
- For $n = 0, 1, 2, \ldots$ and $t \in [T_n, T_{n+1})$:

$$h_n(t) = -\log(\theta + L_{T_n}) \cdot \frac{\partial_z \exp(a(T_n, t, z, L_{T_n}))M_{T_n}(b(T_n, t, z, L_{T_n}), 0)|_{z=0}}{\exp(a(T_n, t, 0, L_{T_n}))M_{T_n}(b(T_n, t, 0, L_{T_n}), 0)}$$

- Recursion, at T_{n+1} compute

$$M_{T_{n+1}}(z, 0) = \exp(-z\delta\ell_{n+1}) \cdot \frac{\partial_v \exp(a(T_n, T_{n+1}, v, L_{T_n}))M_{T_n}(b(T_n, T_{n+1}, v, L_{T_n}), 0)|_{v=z}}{\partial_v \exp(a(T_n, T_{n+1}, v, L_{T_n}))M_{T_n}(b(T_n, T_{n+1}, v, L_{T_n}), 0)|_{v=0}}$$

Given the closed-form solutions (44)–(45), the recursive calculation of $M_{T_{n+1}}(z, 0)$ and, hence, $h_n(t)$, is analytical. However, because the derivatives we need to take get more involved with every step, the computation of $h_n(t)$ may be time consuming and even inaccurate for large n. Proposition 1.1 in the online appendix develops an alternative polynomial representation of $M_{T_{n+1}}(z, 0)$ that facilitates the fast and exact computation of $h_n(t)$ for any n in the case that $\delta = 0$ in (40). For $\delta > 0$, this convenient representation holds as an approximation. Proposition 1.2 in the online appendix shows that the approximation error in $M_{T_{n+1}}(z, 0)$ is of order $O(\delta^2)$. If this approximation is used instead of the exact (but slower to evaluate) expression, then the implementation of the projection Algorithm 3.2 is not exact anymore. Below we show that the approximation provides a favorable trade-off, however: it does reduce the run time of the algorithm while not degrading its convergence.

Figure 1 shows sample paths of the $\mathbb{F}$-intensity λ and the point process L. The parameter values are

Figure 1. Sample path of (λ, L) under the model (40)–(41).

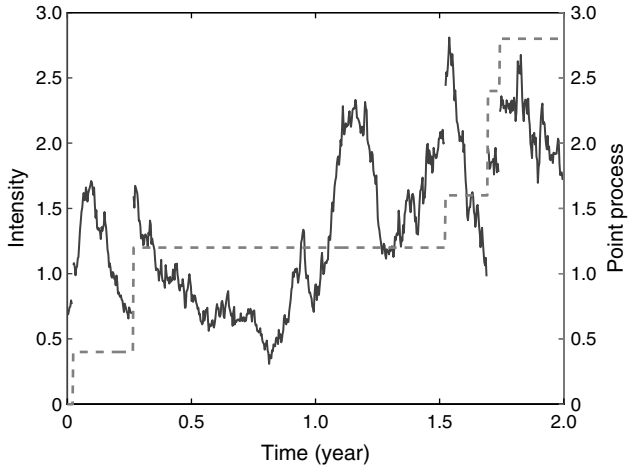
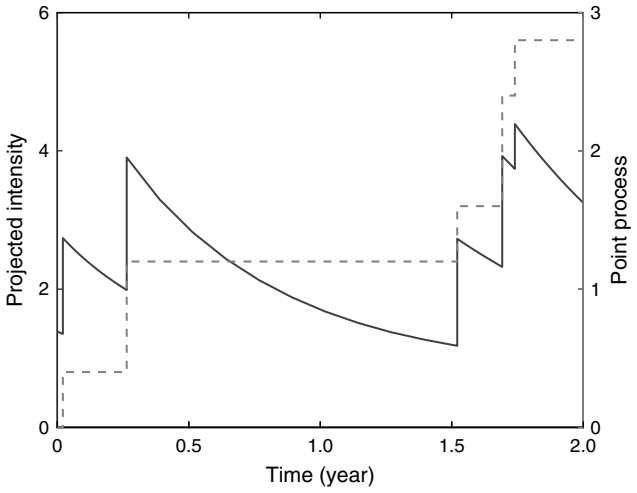


Figure 2. Sample path of (h, L) under the model (40)–(41).



$(\kappa, c, \sigma, \delta, \theta, X_0) = (1, 1, 1, 0.5, 2, 1)$, and the jump magnitudes ℓ_n are drawn from a uniform distribution ν over $\{0.4, 0.8\}$. The jumps in λ occur at the event times. Between events, λ evolves randomly according to a mean-reverting Feller diffusion. Figure 2 shows, for the path of L in Figure 1, the projected intensity h. As the $\mathbb{F}$-intensity λ, the projected intensity h jumps at the event times. However, unlike λ, h evolves deterministically between event times.

5.3. Estimators

Fix a horizon $T > 0$ and a level $K \geqslant 0$. We consider the expectation $\mathbb{E}(X_T 1_{\{L_T \leqslant K\}})$. As explained in §4.3, this expectation arises in finance as the (undiscounted) value of a derivative security issued by an entity subject to default. The estimator is

$$-\partial_z M_T(z, 0)|_{z=0} 1_{\{L_T \leqslant K\}}. \tag{46}$$

We first generate a sample of L_T. If that sample is less than K, then we evaluate

$$\frac{\partial_z \exp(a(T_{N_T}, T, z, L_T))M_{T_{N_T}}(b(T_{N_T}, T, z, L_T), 0)|_{z=0}}{\exp(a(T_{N_T}, T, 0, L_T))M_{T_{N_T}}(b(T_{N_T}, T, 0, L_T), 0)}.$$

This expression is equal to $\partial_z M_T(z, 0)|_{z=0}$, almost surely.

We contrast the projection estimator (46) with an estimator generated by a conventional discretization scheme for X and the time-scaling method for L. As explained in §2.2, the time-scaling method requires paths of the continuous-time process $\int_0^t \lambda_s ds$, where $\lambda_t = X_t \log(\theta + L_t)$. We must discretize the time interval and simulate the integral process dynamics on this discrete-time grid. Because X and hence λ follows a Feller diffusion between arrivals, the conditional distribution of the integral of λ between arrivals is known through its transform. Therefore, exact sampling of the values of the integral on the grid is theoretically feasible, but computationally expensive. Hence, we first

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

simulate the intensity dynamics on the discrete-time grid, and then approximate the integral by the corresponding Riemann sums. To generate the values of λ, we generate the values of X between arrivals by exact sampling from the noncentral chi-squared distribution that governs the transition of X between arrivals.[2]

To evaluate an estimator, we consider its root mean square error (RMSE), given by the square root of the sum of the squared bias and the squared standard error. The standard error (SE) is estimated as the sample standard deviation of the simulation output divided by the square root of the number of trials. The bias is given by the difference between the expectation of the estimator and the true value of $\mathbb{E}(X_T 1_{\{L_T \leqslant K\}})$. The bias is estimated using a large number of trials to estimate the expectation of the estimator and then taking the difference with the true value. The true value is estimated using the exact projection method with a very large number of trials.

5.4. Results: Permanent Event Impact

We estimate $\mathbb{E}(X_1 1_{\{L_1 \leqslant 1\}})$. The parameter values are $(\kappa, c, \sigma, \delta, \theta) = (1, 1, 1, 0, 2)$, the initial value X_0 has a gamma distribution with parameters $(2, 1)$, and the jump magnitudes ℓ_n are drawn from a uniform distribution ν over $\{0.4, 0.8\}$. Note that although we have taken $\delta = 0$, the point process is self-exciting because the intensity (41) is a function of the point process itself. In this formulation, the impact of an event on the intensity is permanent. In §5.5, we consider an alternative self-exciting formulation with $\delta > 0$, in which the impact of an event fades away with time.

At these parameter values, the projected intensity h_n is decreasing (see Figure 2). This suggests an adaptive rule for setting the intensity bound B^n in the implementation of Algorithm 3.2. The first candidate time for T_{n+1} is generated using $h_n(T_n)$ as a bound. If that time is rejected, we generate the next candidate time using the value of h_n at the first candidate time as a bound. That value is computed in any case for the acceptance test. We proceed according to this rule until a candidate time is accepted.

The implementation of Algorithm 3.2 is exact given the closed-form solutions (44)–(45). Thus, the corresponding estimator (46) is unbiased. To get a sense of the magnitude of the bias of (46) caused by numerical approximations in the implementation (see §4.2.3 for a discussion), we also evaluate (46) based on a numerical solution to the ODEs (42)–(43) using the Runge-Kutta scheme, and a finite-difference scheme to estimate the derivatives required to recursively compute $h_n(t)$.

Table 1 reports the simulation results. The bias column is estimated using 800,000 trials. The true value of $\mathbb{E}(X_1 1_{\{L_1 \leqslant 1\}})$ is estimated using the exact implementation of the projection method with 2 million trials. The exact implementation is based on the representation in Proposition 1.1 in the online appendix. The number of time steps in the discretization method is set equal to the square-root of the number of simulation trials.[3] The simulations were performed on a server with an Intel Core Duo 3.16 GHz processor and 4 GB RAM. The codes were written in MATLAB Version 7.9.0.529 (R2009b).

We see that the projection estimators have lower variance than the discretization-based estimator. Figure 3 shows the convergence of the RMS errors graphically. The projection method, in both its exact implementation and its implementation using numerical approximations, achieves the optimal square-root convergence: the error of the estimator decreases at the rate $O(1/\sqrt{t})$, where t is the computational budget. The bias introduced by the numerical approximations is negligible. The convergence rate of the discretization scheme is slower, indicating that the discretization bias is significant. The projection method also outperforms the discretization scheme in terms of absolute errors, with the exact implementation performing better than the implementation using numerical approximations: the exact implementation generates the smallest RMS error for a given computational budget.

5.5. Results: Transient Event Impact

We estimate $\mathbb{E}(X_1 1_{\{L_1 \leqslant 1\}})$ in a setting with nonzero δ, in which the self-exciting behavior takes a richer form. When

Table 1. Simulation results under the model (40)–(41) for $\mathbb{E}(X_1 1_{\{L_1 \leqslant 1\}})$.

Method	Trials	Steps	Est. value	Bias	SE	RMSE	Time (sec)
Projection	5,000	N/A	0.7717	0	0.0077	0.0077	6.22
	10,000	N/A	0.7750	0	0.0054	0.0054	12.25
	50,000	N/A	0.7748	0	0.0024	0.0024	65.48
	100,000	N/A	0.7726	0	0.0017	0.0017	135.38
Projection (approximate)	5,000	N/A	0.7626	−0.0004	0.0078	0.0078	173.13
	10,000	N/A	0.7808	−0.0004	0.0054	0.0054	335.71
	50,000	N/A	0.7667	−0.0004	0.0024	0.0025	1,727.70
	100,000	N/A	0.7731	−0.0004	0.0017	0.0018	3,486.92
Discretization time scaling	5,000	71	0.7631	0.0049	0.0114	0.0125	182.67
	10,000	100	0.7915	0.0020	0.0084	0.0087	532.87
	50,000	224	0.7733	0.0017	0.0037	0.0041	6,921.62
	100,000	316	0.7724	0.0005	0.0026	0.0027	21,383.54

Notes. The parameter values are $(\kappa, c, \sigma, \delta, \theta) = (1, 1, 1, 0, 2)$, the initial value X_0 has a gamma distribution with parameters $(2, 1)$, and the jump magnitudes ℓ_n are drawn from a uniform distribution over $\{0.4, 0.8\}$. The true value is given by 0.7723.

Figure 3. Convergence of the RMSEs of the methods shown in Table 1.

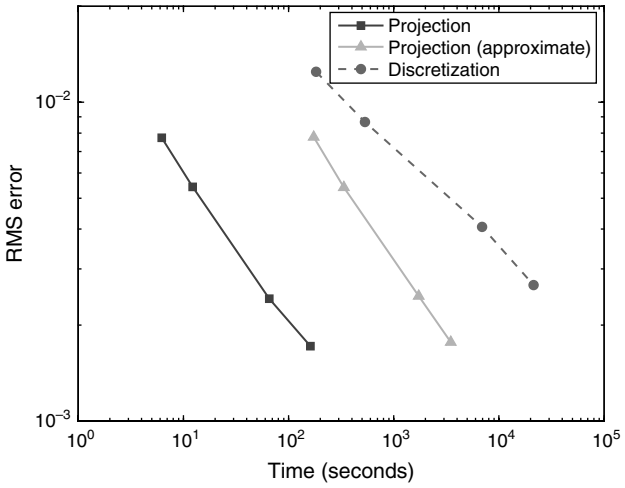
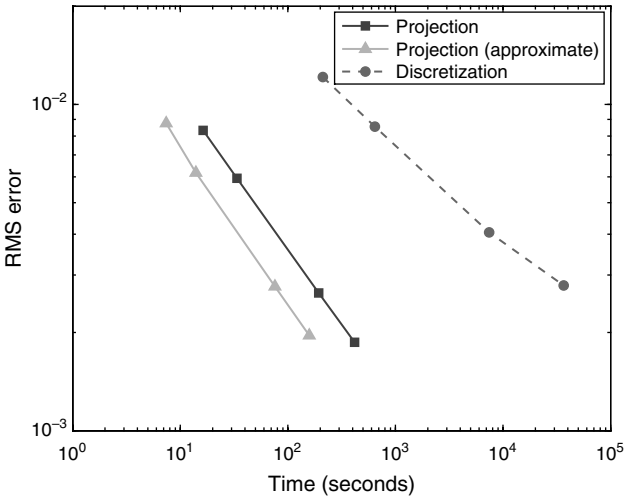


Figure 4. Convergence of the RMSEs of the methods shown in Table 2.



$\delta > 0$, the intensity responds to an event with impact decaying exponentially with time at rate κ. To avoid overlap with the formulation considered previously, in which an event has a permanent impact on the intensity, we take $\Lambda(t, x, l) = x$ instead of (41). Thus, the response of the intensity to an event comes from the response of the state X rather than the dependence of $\Lambda(t, x, l)$ on l.

The projected intensity h_n is decreasing also in this setting, so we choose the intensity bound required by Algorithm 3.2 adaptively as described in §5.4 above. The implementation is exact given the closed-form solutions (44)–(45), and the corresponding estimator (46) is unbiased. To assess the bias of (46) caused by the use of the polynomial representation in Proposition 1.1 in the online appendix, which is fast to compute but only approximate when $\delta > 0$, we also evaluate (46) using this representation. We are interested in whether we can reduce the run time of the algorithm without introducing a material bias.

Table 2 reports the simulation results and Figure 4 shows the convergence of the RMS errors. The parameter values are $(\kappa, c, \sigma, \delta) = (1, 1, 1, 0.5)$, the initial value X_0 has a gamma distribution with parameters $(2, 1)$, and the jump magnitudes ℓ_n are drawn from a uniform distribution ν over $\{0.4, 0.8\}$. The bias column is estimated using 800,000 trials. The true value of $\mathbb{E}(X_1 1_{\{L_1 \leqslant 1\}})$ is estimated using the exact implementation of the projection method, based on 2 million trials. We see that the projection method achieves the optimal square-root convergence rate even when its implementation is based on an approximation of h_n. The approximation reduces the run time of the algorithm, but does not introduce a meaningful bias into the simulation estimator. The discretization method has a slower convergence rate and requires more computation time than the projection method to achieve a given accuracy.

6. Conclusion

This paper develops a method for the exact simulation of point processes with stochastic intensities. The method is

Table 2. Simulation results under the model (40) with $\Lambda(t, x, l) = x$ for $\mathbb{E}(X_1 1_{\{L_1 \leqslant 1\}})$.

Method	Trials	Steps	Est. value	Bias	SE	RMSE	Time (sec)
Projection	5,000	N/A	0.6891	0	0.0083	0.0083	16.3281
	10,000	N/A	0.6842	0	0.0059	0.0059	33.7096
	50,000	N/A	0.6846	0	0.0026	0.0026	193.5535
	100,000	N/A	0.6807	0	0.0019	0.0019	417.3626
Projection (approximate)	5,000	N/A	0.6886	−0.0002	0.0088	0.0088	7.4107
	10,000	N/A	0.6772	−0.0002	0.0062	0.0062	13.9846
	50,000	N/A	0.6769	−0.0002	0.0028	0.0028	75.8327
	100,000	N/A	0.6793	−0.0002	0.0020	0.0020	158.3177
Discretization time scaling	5,000	71	0.6930	0.0027	0.0118	0.0121	211.8054
	10,000	100	0.6881	0.0019	0.0083	0.0086	643.6262
	50,000	224	0.6803	0.0017	0.0037	0.0041	7,433.5224
	100,000	316	0.6824	0.0009	0.0026	0.0028	36,662.6488

Notes. The parameter values are $(\kappa, c, \sigma, \delta) = (1, 1, 1, 0.5)$, the initial value X_0 has a gamma distribution with parameters $(2, 1)$, and the jump magnitudes ℓ_n are drawn from a uniform distribution over $\{0.4, 0.8\}$. The true value is given by 0.6802.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at http://journals.informs.org/.

based on a change of the filtration that describes the information flow in the point process model. Instead of simulating the point process in the reference filtration, we project the process onto its own filtration and then sample the process in this coarser filtration. In the coarser filtration, the point process intensity is deterministic between event times, and the arrivals can be generated using standard thinning or inverse transform schemes. The projection eliminates the need to discretize the stochastic intensity process in the reference filtration, and therefore avoids the discretization bias introduced by standard simulation schemes.

We exemplify the projection method for a point process with an intensity driven by a jump-diffusion process and the point process itself. In this setting, the projection method leads to estimators of expectations of functionals of the point process path and a skeleton of the jump-diffusion process. Numerical experiments demonstrate the effectiveness of the method and its advantages over conventional discretization schemes.

7. Electronic Companion

An electronic companion to this paper is available as part of the online version that can be found at http://or.journal.informs.org/.

Endnotes

1. Even if the transform of (X, L) is tractable, its numerical inversion can be challenging, especially if the parameters of the coefficient functions are not fixed in advance, and the inversion procedure must deal with a wide range of different parameter values. This situation regularly occurs in the context of parameter calibration or estimation problems, where an optimization loop requires the distribution of (X, L) for many different parameter sets. Here, simulation may be a good alternative.

2. An alternative would be to apply Euler discretization to the SDE (40). This, however, can result in negative values for X that must be treated by truncation or otherwise, generating another source of bias. Sampling from the transition law of X avoids this at the expense of greater computational effort.

3. It is not clear how to allocate the computational budget of the time-scaling method between the number of time steps and the number of replications. The square-root rule is adopted from Broadie and Kaya (2006). It is motivated by the results in Duffie and Glynn (1995), who show that for first-order methods, it is asymptotically optimal to increase the number of time steps proportional to the square root of the number of replications. However, the optimal constant of proportionality is not known.

Acknowledgments

An early version of this paper entitled "Simulating Point Processes by Intensity Projection" appeared in the *Proceedings of the 2008 Winter Simulation Conference*. The authors are grateful to Mizuho-DL Financial Technology for a grant that supported this work. They thank Baeho Kim for excellent research assistance, and Paul Glasserman, Sandeep Juneja, Jussi Keppo, Jeremy Staum, and participants at the 2008 Winter Simulation Conference, the 2009 Applied Probability Society Conference, the 2009 INFORMS Annual Meeting, and the 2009 Western Conference in Mathematical Finance for comments. The authors are also grateful to two anonymous referees, an associate editor, and Shane Henderson for insightful reviews.

References

Arnsdorf, M., I. Halperin. 2008. BSLP: Markovian bivariate spread-loss model for portfolio credit derivatives. *J. Comput. Finance* **12**(1) 77–100.

Beskos, A., G. Roberts. 2005. Exact simulation of diffusions. *Ann. Appl. Probab.* **15**(4) 2422–2444.

Bowsher, C. 2007. Modelling security market events in continuous time: Intensity based, multivariate point process models. *J. Econometrics* **141**(2) 876–912.

Brémaud, P. 1980. *Point Processes and Queues—Martingale Dynamics*. Springer-Verlag, New York.

Broadie, M., O. Kaya. 2006. Exact simulation of stochastic volatility and other affine jump diffusion processes. *Oper. Res.* **54**(2) 217–231.

Carr, P., V. Linetsky. 2006. A jump to default extended CEV model: An application of Bessel processes. *Finance Stochastics* **10**(3) 303–330.

Casella, B., G. Roberts. 2011. Exact simulation of jump-diffusion processes with Monte Carlo applications. *Methodology Comput. Appl. Probab.* **13**(3) 449–473.

Ceci, C., A. Gerardi. 2006. A model for high-frequency data under partial information: A filtering approach. *Internat. J. Theoret. Appl. Finance* **9**(4) 555–576.

Chen, L., D. Filipovic, H. V. Poor. 2004. Quadratic term structure models for risk-free and defaultable rates. *Math. Finance* **14**(4) 515–536.

Chen, N. 2009. Localization and exact simulation of Brownian motion driven stochastic differential equations. Working paper, Chinese University of Hong Kong, Hong Kong.

Cont, R., S. Stoikov, R. Talreja. 2010. A stochastic model for order book dynamics. *Oper. Res.* **58**(3) 549–563.

Dellacherie, C., P.-A. Meyer. 1982. *Probabilities and Potential*. North Holland, Amsterdam.

Ding, X., K. Giesecke, P. I. Tomecek. 2009. Time-changed birth processes and multiname credit derivatives. *Oper. Res.* **57**(4) 990–1005.

Duffie, D., P. Glynn. 1995. Efficient Monte Carlo estimation of security prices. *Ann. Appl. Probab.* **4**(5) 897–905.

Duffie, D., J. Pan, K. Singleton. 2000. Transform analysis and asset pricing for affine jump-diffusions. *Econometrica* **68**(6) 1343–1376.

Duffie, D., M. Schroder, C. Skiadas. 1996. Recursive valuation of defaultable securities and the timing of resolution of uncertainty. *Ann. Appl. Probab.* **6**(4) 1075–1090.

Giesecke, K., H. Kakavand, M. Mousavi, H. Takada. 2010. Exact and efficient simulation of correlated defaults. *SIAM J. Financial Math.* **1** 868–896.

Glasserman, P. 2004. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York.

Glasserman, P., N. Merener. 2003. Numerical solution of jump-diffusion LIBOR market models. *Finance Stochastics* **7**(1) 1–27.

Kliemann, W., G. Koch, F. Marchetti. 1990. On the unnormalized solution of the filtering problem with counting process observations. *IEEE Trans. Inform. Theory* **36**(6) 1415–1425.

Leippold, M., L. Wu. 2002. Asset pricing under the quadratic class. *J. Financial Quant. Anal.* **37**(2) 271–295.

Lewis, P., G. Shedler. 1979. Simulation of nonhomogeneous Poisson processes by thinning. *Naval Logist. Quart.* **26** 403–413.

Meyer, P.-A. 1971. Démonstration simplifée d'un théorème de Knight. *Séminaire de Probabilités V, Lecture Notes in Mathematics 191*. Springer-Verlag, Berlin, 191–195.

Ogata, Y. 1981. On Lewis' simulation method for point processes. *IEEE Trans. Inform. Theory* **27**(1) 23–31.