Exact Simulation of Quadratic Intensity Models[*]

Yan Qu[†]
Beijing University of Posts and Telecommunications

Angelos Dassios[‡]
London School of Economics

Anxin Liu[§] Hongbiao Zhao[¶]
Shanghai University of Finance and Economics

20th September 2024

Abstract

We develop efficient algorithms of exact simulation for quadratic stochastic intensity models which have become increasingly popular for modelling events arrivals, especially in economics, finance and insurance. They have huge potential to be applied to many other areas such as operations management, queueing science, biostatistics and epidemiology. Our algorithms are developed by the principle of exact distributional decomposition, which lies in a *fully analytical* expression for the joint Laplace transform of quadratic process and its integral newly derived in this paper. They do not involve any numerical Laplace inversion, and have been validated by extensive numerical experiments, and substantially outperform all existing alternatives in the literature. Moreover, our algorithms are extendable to multidimensional point processes, and beyond Cox processes to additionally incorporate two-sided random jumps with arbitrarily distributed sizes in the intensity for capturing self-exciting and self-correcting effects in event arrivals. Applications to portfolio loss modelling are provided to demonstrate the applicability and flexibility of our algorithms.

Keywords: Exact simulation; Monte Carlo simulation; Stochastic intensity model; Quadratic intensity model; Point process; Doubly stochastic Poisson process; Cox process; Self-exciting process; Portfolio credit risk

Mathematics Subject Classification (2010): Primary: 60G55; Secondary: 60H35 · 65C05 · 60G17

[*] An edited version will be published by *INFORMS Journal on Computing*.

[†] School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing 100876, China. Email: quyan0418@bupt.edu.cn

[‡] Department of Statistics, London School of Economics, Houghton Street, London WC2A 2AE, United Kingdom. Email: a.dassios@lse.ac.uk

[§] School of Statistics and Management, Shanghai University of Finance and Economics, No. 777 Guoding Road, Shanghai 200433, China. Email: liu.anxin@163.sufe.edu.cn

[¶] *Corresponding author*, School of Statistics and Management, Shanghai University of Finance and Economics, No. 777 Guoding Road, Shanghai 200433, China; Shanghai Institute of International Finance and Economics, No. 777 Guoding Road, Shanghai 200433, China. Email: h.zhao1@lse.ac.uk

1 Introduction

Highly analytically tractable models are treasures in the toolbox for quantitative analysis in finance. In particular, they can lead to analytical formulas for fast evaluating financial products, for efficiently calibrating the real-time data of market prices of liquidly-traded vanilla financial derivatives (such as European options), and then for evaluating more exotic and less liquidly-traded derivatives often with the aid of efficient simulations. Researchers for decades are fascinated in searching for these models. A notable class are *affine models* which include the classical Ornstein-Uhlenbeck (OU) and Cox-Ingersoll-Ross (CIR) processes as special cases. At the early stage, *affine term structure models* have gained significant attention in the finance literature, mainly due to their analytical tractability and statistical flexibility (Cuchiero et al., 2010). Affine models have been later extended broadly to stochastic models for price movements and stochastic volatilities, and more recently for point (jump) processes with stochastic intensities.

Indeed, affine models have been widely adopted for modelling the dynamics of economic variables, see e.g. Duffie and Kan (1996), Duffie et al. (2000, 2003) in continuous time, and Gourieroux et al. (2006), Le et al. (2010), Joslin et al. (2011) in discrete time. They are highly tractable, and many quantities of interest, such as bond prices, default probabilities and option prices, have analytical forms. However, a key limitation of classical affine models is that they cannot simultaneously allow for *negative* correlation and *positivity* among state variables, see e.g. Duffie and Liu (2001).

Alternatively, *quadratic* class serves as a very strong competitor to the affine class, and becomes increasingly popular recently for financial modelling, especially for nonnegative variables such as asset prices, interest rates and intensity processes of event arrivals. For example, they are adopted for modelling asset prices by Leippold and Wu (2002), and for exchanges rates by Leippold and Wu (2007). They are intensively used for interest rates and form a framework called *quadratic term structure models*, see e.g. Jamshidian (1996), Duffie and Liu (2001), Ahn et al. (2002), Leippold and Wu (2003), Chen et al. (2004) in continuous time, and Ang et al. (2011), Campbell et al. (2017) in discrete time. The typical finding is that, they outperform the classical affine term structure models in better capturing the correlation and heteroscedasticity of yields (Dai and Singleton, 2000; Duffie and Liu, 2001; Ahn et al., 2002; Li and Zhao, 2006), particularly for short-term government bond yields in near zero-rate environments (Kim and Singleton, 2012; Andreasen and Meldrum, 2013).

More recently, quadratic class is adopted for stochastic *intensity* processes of event arrivals (e.g.

price jumps, corporate defaults, bankruptcies, crises, disasters), and hence forms a framework of *quadratic intensity models*. For example, the jump-arrival intensity for *unscheduled* FOMC (Federal Open Market Committee) announcements is quadratic in Piazzesi (2001, 2005). Similarly, in order to capture the jump clustering of large movements and potential market crashes in the S&P 500 index, Santa-Clara and Yan (2010) used a quadratic model for the intensity process of jump arrivals. As pointed out by Singleton (2009, p.369), stochastic intensity models for default risk have largely paralleled with the literature on stochastic interest rate models. Naturally, quadratic intensity models, as an important class of *point processes*, are also suitable for modelling default arrivals and for pricing credit products (such as defaultable bonds, credit default swaps and collateralised debt obligations), see e.g. Duffie and Liu (2001) in continuous time, and Doshi et al. (2013, 2017, 2018), Choi et al. (2020) in discrete time. For example, Duffie and Liu (2001) evaluated corporate bond prices, and found that the quadratic model can offer a more flexible correlation structure for capturing the *negative* instantaneous correlation between *positive* default intensities and interest rates, which is impossible for standard affine models such as the popular CIR intensity models (Duffie and Singleton, 1999). Alternatively, Singleton (2009, §14.2.2) and Doshi et al. (2013, 2017, 2018) used discrete-time quadratic intensity models for pricing credit default swaps. Moreover, Gourieroux and Monfort (2008) applied them to French human mortality in insurance. In fact, similar quadratic intensity models were developed for survival analysis in biostatistics and epidemiology earlier than the economic literature, see e.g. Woodbury and Manton (1977) and Aalen and Gjessing (2004).

Evidently from voluminous existing literature, quadratic intensity models are important. Due to vast events of different types occurring randomly in the real world, their potential applications to modelling these event arrivals (e.g. job losses, trade orders, unscheduled announcements, news reports, information flows, online posts, website visits, particulate emissions, traffic flows, queues, customer arrivals, crimes, reliability, COVID-19 infections) in different disciplines and interdisciplines of both natural science and social science may be even larger in the future. In the paper, we only offer applications in finance to facilitate our illustration. As an obvious example of potential applications beyond finance, they may be also feasible for modelling call arrivals in telephone call centers with *time-varying arrival rates*, which have been extensively studied in the literature of operations management, see e.g. Whitt (1999), Gans et al. (2003), Avramidis et al. (2004), Brown et al. (2005), Cezik and L'Ecuyer (2008), Kim and Whitt (2014) and Ata and Peng (2020).

Efficient algorithms of Monte Carlo simulation for stochastic models would be generically useful, and play important roles in their practical implementations, such as simulation-based statistical inference, optimisation and empirical studies, see e.g. Hong and Nelson (2009), L'Ecuyer (2012) and Fu (2015). In the financial industry, for instance, simulations are widely adopted for risk management (e.g. stress tests) and asset pricing (e.g. option pricing), see e.g. Glasserman (2003), Asmussen and Glynn (2007), Hong and Liu (2009) and Gordy and Juneja (2010). Traditional algorithms for simulating continuous-time stochastic processes are mainly based on *time discretisation* such as Euler scheme. Although straightforward for implementations, they introduce estimation *bias* which is hard to be quantified and measured. In particular, an algorithm for *exact* simulation is highly desirable, since it has the primary advantage of generating sample paths according to the law of underlying process exactly without bias. More precisely, exact simulation means that the joint distribution of simulated values coincides with the joint distribution of the continuous-time process on any simulation time grid (Glasserman, 2003, p.79). Therefore, the problems of designing exact simulation for stochastic processes have attracted substantial attention in the academic community, see e.g. Beskos and Roberts (2005), Ahdida and Alfonsi (2013), Chen and Huang (2013), Giesecke and Smelov (2013) for diffusion and jump-diffusion processes, Broadie and Kaya (2006), Cai et al. (2017), Kang et al. (2017), Li and Wu (2019) for stochastic-volatility models, and Giesecke et al. (2011a,b), Dassios and Zhao (2011, 2013, 2017) for point processes, see also a recent survey by Glynn (2016).

Exact simulation for the quadratic intensity model was first proposed in Giesecke et al. (2011a, §4.2.2) as *projection scheme*, which is the only *exact* algorithm in the existing literature. It is theoretically correct, but it is difficult to implement even a small number of event realisations using a normal computer in practice. As pointed out by themselves in their paper, the key drawback of projection scheme is that it involves massive recursive calculations of projected intensities, and these recursions cumulate a huge number of analytic terms in symbolic format stored in a computer, which would be extremely time-consuming and memory-consuming. Therefore, to speed up they also proposed an approximation scheme which, however, is not exact any more.

In our paper, we adopt a completely different approach, and develop the first *practically implementable* exact simulation scheme for quadratic intensity models. It is based on *exact distributional decomposition* without any numerical inversion of cumulative distribution function or Fourier transform. Quite elegantly, interarrival times and the intensity levels at event-arrival times can be sequentially decomposed into simpler and more easily simulatable random variables. Moreover, our algorithms are applicable to more general quadratic intensity models beyond the canonical version in Giesecke et al. (2011a). For example, *two-sided* random jumps with arbitrarily distributed jump sizes are also allowed.

Another theoretical contribution of this paper is that, we obtain a *fully analytical* expression for the *joint* Laplace transform of a quadratic process and its integral (as given later by Theorem 3.1),[1] an analogically important result as the counterpart for the classical CIR process.[2] This fully analytical expression is crucial, since it further inspires us to *exactly* identify and decompose the underlying distributions, which is the key to our *exact decomposition* approach for simulation design.

The paper is organised as follows. Section 2 introduces the quadratic intensity model formally. Section 3 derives key distributional properties, which lead to our new algorithm for exact simulation in Section 4. Section 5 outlines some further extensions of quadratic intensity models with additional self-exciting and self-correcting effects, and multidimensional point processes, with the associated simulation algorithms given by Appendix P. Section 6 provides extensive numerical experiments for validating our algorithms, performance comparisons with other algorithms, and an application to loss modelling for a credit portfolio. Section 7 draws a brief conclusion and suggests potential topics for future research. All the proofs are deferred to the appendices.The associated codes for replicating numerical results in our paper can be found in Qu et al. (2024).

2 Quadratic Intensity Models

Let us first introduce a simple canonical version of quadratic (stochastic) intensity models similarly as adopted by e.g. Santa-Clara and Yan (2010) and Giesecke et al. (2011a). More comprehensive model extensions are given in Section 5.

Definition 2.1 (Quadratic OU Process)**.** *Stochastic process λ_t is a quadratic OU process such that*

$$\lambda_t \; = \; X_t^2, \qquad t \geq 0, \tag{2.1}$$

where

- *X_t is the state process following a general mean-reverting Ornstein-Uhlenbeck (OU) process*

[1] To the best of our knowledge, the parameters of this joint Laplace transform in the previous literature were just left within *unsolved* (or partially solved, or numerically solved) ordinary differential equations (ODEs) as in Giesecke et al. (2011a) or in a semi-analytical integral form as Chen et al. (2004, §4.4) Recently, Li and Wu (2019, p.770) derived the conditional Laplace transform of quadratic process, and developed simulation algorithms for the quadratic stochastic volatility model. They also suggested using the *joint* Laplace transform already given in Chen et al. (2004) to alteratively obtain the conditional Laplace transform of quadratic process.

[2] The *fully analytical* expression for the *joint* Laplace transform of CIR process and its integral can be found in e.g. Glasserman (2003, Equation 3.76) and Lamberton and Lapeyre (2008, Proposition 6.2.4).

in continuous time on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$,[3] *i.e.,*

$$\mathrm{d}X_t \;=\; -\delta(X_t - \mu)\mathrm{d}t + \sigma \mathrm{d}W_t, \qquad t \geq 0, \tag{2.2}$$

- $\delta \in \mathbb{R}^+$ *is the mean-reverting rate (or mean-reverting speed),*
- $\mu \in \mathbb{R}$ *is the mean-reverting level,*
- $\sigma \in \mathbb{R}^+$ *is the volatility parameter,*
- W_t *is a standard Brownian motion.*

It is trivial to alternatively define a quadratic OU process seemingly more general than the canonical version (2.1) by parameterising as $\lambda_t = (aX_t + b)^2$ for any $a, b \in \mathbb{R}$. However, it is essentially equivalent by replacing μ by $(a\mu + b)$ and σ by $a\sigma$ in (2.2) for the original canonical version.

Definition 2.2 (Quadratic Intensity Model). *A quadratic intensity model is a point process*

$$N_t \;:=\; \sum_{i \geq 1} \mathbb{1}_{\{T_i \leq t\}}, \qquad N_0 = 0,$$

where T_i's are ordered arrival times with the quadratic intensity λ_t specified by Definition 2.1.

Notice that, a quadratic intensity model is *non-affine* in general, and cannot be simply considered as a subclass of affine processes (Duffie et al., 2003). Since by Itô's lemma, we have

$$\begin{aligned}
\mathrm{d}\lambda_t \;&=\; \Big(- 2\delta X_t^2 + 2\mu\delta X_t + \sigma^2 \Big)\mathrm{d}t \;+\; 2\sigma X_t \mathrm{d}W_t \\
&=\; \Big(-2\delta\lambda_t + 2\mu\delta\sqrt{\lambda_t}\mathsf{sign}(X_t) + \sigma^2 \Big)\,\mathrm{d}t \;+\; 2\sigma\sqrt{\lambda_t}\mathrm{d}\tilde{W}_t,
\end{aligned}$$

where $\tilde{W}_t$ is a standard Brownian motion defined by $\mathrm{d}\tilde{W}_t := \mathsf{sign}(X_t)\mathrm{d}W_t = \frac{X_t}{|X_t|}\mathrm{d}W_t$, see also e.g. Chen et al. (2004, p.522). In particular, for the *centered* case with $\mu = 0$, the intensity process reduces to a special CIR process,

$$\mathrm{d}\lambda_t \;=\; \Big(\sigma^2 - 2\delta\lambda_t\Big)\,\mathrm{d}t + 2\sigma\sqrt{\lambda_t}\mathrm{d}\tilde{W}_t.$$

However, it does not satisfy the Feller condition, and zero is accessible. So, the quadratic model for $\mu \neq 0$ in general cannot be reduced to a CIR process, as also pointed out by Jamshidian (1996,

[3] Notice that it is more general than the original Ornstein-Uhlenbeck (OU) process (Uhlenbeck and Ornstein, 1930), since it includes both *central* ($\mu = 0$) and *noncentral* ($\mu \neq 0$) OU processes. It is a classical model of Vasicek (1977) for describing the evolution of interest rates in finance.

p.103). One key advantage of quadratic models over the CIR counterparts is that, they guarantee non-negative values without any additional conditions, such as the Feller condition and initial conditions (Lamberton and Lapeyre, 2008, §6.2.2).

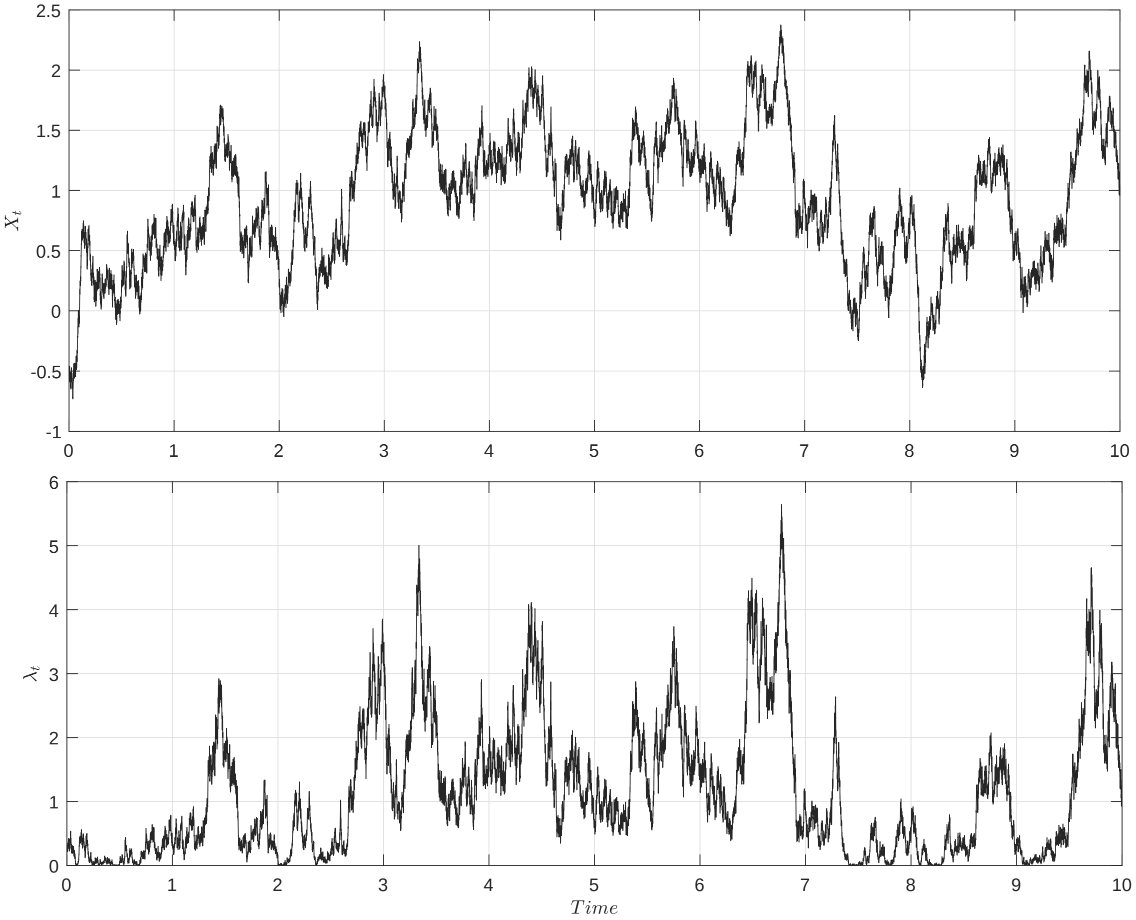

Figure 1: Simulated paths of state process X_t and quadratic intensity $\lambda_t = X_t^2$ under the parameter setting $(X_0, \mu, \delta, \sigma) = (-0.5, 0.8, 1.1, 1.3)$.

Given the value of $X_t \in \mathbb{R}$ at time t, the solution to the SDE (2.2) at time $t + s$ for any $s > 0$ is

$$X_{t+s} \;=\; X_t e^{-\delta s} + \mu\left(1 - e^{-\delta s}\right) + \sigma \int_0^s e^{-\delta(s-u)} \mathrm{d}W_u, \tag{2.3}$$

with a normal marginal distribution $\mathsf{N}\left(\nu, \varsigma^2\right)$, i.e.,

$$X_{t+s} \mid X_t \;\overset{\mathcal{D}}{=}\; \nu + \varsigma\epsilon, \qquad \epsilon \sim \mathsf{N}(0,1), \tag{2.4}$$

where

$$\nu \;:=\; X_t e^{-\delta s} + \mu\left(1 - e^{-\delta s}\right), \qquad \varsigma \;:=\; \sigma\sqrt{\frac{1 - e^{-2\delta s}}{2\delta}}. \tag{2.5}$$

For the purpose of illustration, simulated sample paths of state process X_t and the associ-

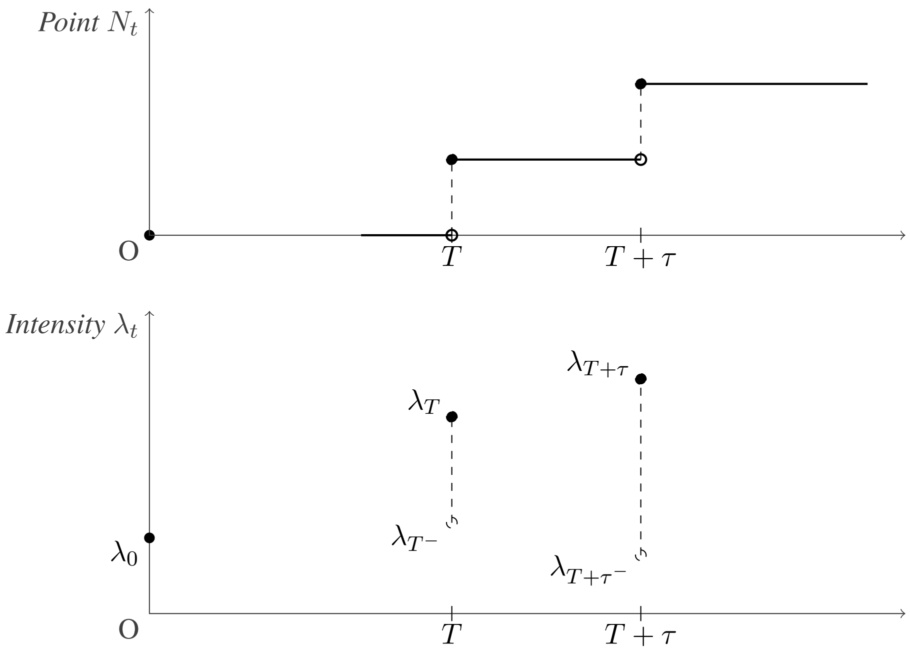

Figure 2: Illustration for exactly simulating a *path* of point process N_t and a *skeleton* of intensity process λ_t.

ated quadratic intensity λ_t within the period $[0, 10]$ under the parameter setting $(X_0, \mu, \delta, \sigma) = (-0.5, 0.8, 1.1, 1.3)$ are jointly plotted in Figure 1. As we can see from (2.5), the underlying quadratic intensity process λ_t at any (regularly-spaced or irregularly-spaced) time points can be exactly simulated very easily due to the marginal distribution of *squared normal*.[4] For applications, another advantage of quadratic specification is that, by construction, no parametric restriction is required to ensure the strict positivity of intensity processes.[5]

Let's first very briefly introduce key steps for our exact simulation before diving into theoretic and algorithmic details. Based on our theoretical results newly derived in Section 3, an entire path of a point process with quadratic intensity can be exactly simulated in a recursive manner as explained in Section 4. Figure 2 illustrates that, conditional on the *current* event-arrival time $t = T$ and the associated positions of the intensity (quadratic) process and point process, the *next* event-arrival can be exactly simulated via the following core procedures:

1. Conditional on the *current* event-arrival time $t = T$, the state process X_T and intensity $\lambda_T = X_T^2$, exactly simulate the (next) *interarrival time* τ;

2. With this simulated interarrival time τ, exactly simulate the *pre-event*[6] intensity level $\lambda_{T+\tau^-}$ right before the next arrival time $T + \tau$;

[4] However, the exact simulation for point process N_t is nontrivial, which is the main problem we have tackled in this paper.

[5] For example, the Feller condition is required to ensure the positivity of a CIR process.

[6] Here, "*pre-event*" means prior to an event (or equivalently a jump in the point process N_t for counting the number of event arrivals).

3. With this simulated pre-event intensity level $\lambda_{T+\tau^-}$, exactly simulate the pre-event position of state process $X_{T+\tau^-}$;

4. Add one unit in the point process at the *next* event-arrival time $T+\tau$.

By implementing these fours steps recursively, a full path of point process can be exactly simulated.

3 Distributional Properties

In this section, we investigate distributional properties of quadratic intensity model, and provide the theoretical foundation for designing our exact simulation algorithms later in Section 4. Without loss of generality, we focus on a single recursion of a point process by resetting the *current time* t as the currently-realised event-arrival time, i.e., the *current arrival time* t. The (next) *interarrival time* between the current arrival time t and the *next arrival time* is denoted by τ. The *next arrival time* is then $t+\tau$.

3.1 Joint Laplace Transform of Intensity Process and Its Integral

The conditional joint Laplace transform of state process X_t and its quadratic process X_t^2 can be derived analytically in Proposition 3.1 as below.

Proposition 3.1. *The joint bivariate Laplace transform of $\left(X_{t+s}, X_{t+s}^2\right)$ conditional on X_t at the current arrival time t for any $s>0$ is given by*

$$\mathbb{E}\left[e^{-uX_{t+s}}e^{-wX_{t+s}^2}\,\Big|\,X_t\right] = \frac{1}{\sqrt{2\varsigma^2 w+1}}\exp\left(\frac{\varsigma^2}{2(2\varsigma^2 w+1)}\left(u-\frac{\nu}{\varsigma^2}\right)^2-\frac{\nu^2}{2\varsigma^2}\right), \tag{3.1}$$

where $u\in\mathbb{R}$, $w>-\dfrac{1}{2\varsigma^2}$, and ν, ς are specified by (2.5). The Laplace transform of X_{t+s}^2 conditional on X_t is given by

$$\mathbb{E}\left[e^{-wX_{t+s}^2}\,\Big|\,X_t\right] = \frac{1}{\sqrt{2\varsigma^2 w+1}}\exp\left(-\frac{\nu^2 w}{2\varsigma^2 w+1}\right). \tag{3.2}$$

For notational simplicity, let us denote the *cumulative intensity process* (i.e., *compensator* of point process N_t) by

$$\Lambda_t := \int_0^t \lambda_s \mathrm{d}s.$$

Based on $\left(X_t, X_t^2, \Lambda_t\right)$, we can construct a martingale M_t in Lemma 3.1 as below which will be later used for carrying out a change of measure.

Lemma 3.1. *For* $w > -\frac{\delta}{2\sigma^2}$*, we have a* $\mathbb{P}$*-martingale*

$$M_t \;:=\; e^{-w\Lambda_t}e^{-\kappa_1 X_t}e^{-\kappa_2 X_t^2}e^{\kappa_0 t}, \qquad t \geq 0, \tag{3.3}$$

where

$$\kappa_2 \;:=\; \frac{-\delta + \sqrt{\delta^2 + 2\sigma^2 w}}{2\sigma^2}, \quad \kappa_1 \;:=\; \frac{2\mu\delta\kappa_2}{\delta + 2\sigma^2\kappa_2}, \quad \kappa_0 \;:=\; \mu\delta\kappa_1 - \frac{1}{2}\sigma^2\left(\kappa_1^2 - 2\kappa_2\right). \tag{3.4}$$

Since M_t is a strictly positive martingale, an equivalent martingale probability measure $\tilde{\mathbb{P}} \sim \mathbb{P}$ can be defined via the Radon-Nikodŷm derivative

$$\frac{\mathrm{d}\tilde{\mathbb{P}}}{\mathrm{d}\mathbb{P}} \;=\; \frac{M_t}{M_u}, \qquad 0 \leq u \leq t. \tag{3.5}$$

Under this new measure $\tilde{\mathbb{P}}$, the parameter setting of process X_t is transformed from (μ, δ, σ) to $\left(\tilde{\mu}, \tilde{\delta}, \tilde{\sigma}\right)$ via (3.6) in Proposition 3.2 as below.

Proposition 3.2. *The parameter setting* (μ, δ, σ) *of process* X_t *under* $\mathbb{P}$ *is transformed to the new parameter setting* $\left(\tilde{\mu}, \tilde{\delta}, \tilde{\sigma}\right)$ *of process* X_t *under* $\tilde{\mathbb{P}}$ *according to*

$$\tilde{\delta} \;:=\; \sqrt{\delta^2 + 2\sigma^2 w}, \qquad \tilde{\mu} \;:=\; \frac{\delta^2}{\delta^2 + 2\sigma^2 w}\mu, \qquad \tilde{\sigma} \;:=\; \sigma, \tag{3.6}$$

for $w > -\frac{\delta}{2\sigma^2}$.

Based on the joint Laplace transform of $\left(X_t, X_t^2\right)$ in Proposition 3.1 and change of measure in Proposition 3.2, now we can analytically derive the conditional joint Laplace transform of (λ_t, Λ_t) in Theorem 3.1 as below, the most important theoretical result in this paper.

Theorem 3.1. *The joint bivariate Laplace transform of* $(\lambda_{t+s}, \Lambda_{t+s} - \Lambda_t)$ *conditional on* X_t *at the current arrival time t for any* $s > 0$ *is given by*

$$\begin{aligned} &\mathbb{E}\left[e^{-u\lambda_{t+s}}e^{-w(\Lambda_{t+s}-\Lambda_t)}\;\middle|\; X_t\right] \\ =\;& \frac{e^{-\kappa_1 X_t}e^{-\kappa_2 X_t^2}e^{-\kappa_0 s}}{\sqrt{2\tilde{\varsigma}^2 u + (1 - 2\tilde{\varsigma}^2\kappa_2)}} \exp\left(\frac{\tilde{\varsigma}^2}{2\left[2\tilde{\varsigma}^2 u + (1 - 2\tilde{\varsigma}^2\kappa_2)\right]}\left(\kappa_1 + \frac{\tilde{\nu}}{\tilde{\varsigma}^2}\right)^2 - \frac{\tilde{\nu}^2}{2\tilde{\varsigma}^2}\right), \end{aligned} \tag{3.7}$$

where $w > -\frac{\delta}{2\sigma^2}$, $u > \kappa_2 - \frac{1}{2\varsigma^2}$,

$$\tilde{\nu} \;:=\; X_t e^{-\tilde{\delta}s} + \tilde{\mu}\left(1 - e^{-\tilde{\delta}s}\right), \qquad \tilde{\varsigma} \;:=\; \tilde{\sigma}\sqrt{\frac{1 - e^{-2\tilde{\delta}s}}{2\tilde{\delta}}}, \tag{3.8}$$

and $\left(\tilde{\mu}, \tilde{\delta}, \tilde{\sigma}\right)$, $(\kappa_2, \kappa_1, \kappa_0)$ *are given by* (3.6) *and* (3.4)*, respectively.*

Joint Laplace transforms, similar to (3.7), are also found in e.g. Chen et al. (2004) and Giesecke et al. (2011a). However, the parameters of their results are not explicitly given but just left within an unsolved (or partially solved, or numerically solved) ODE system or in a semi-analytical integral form. To the best of our knowledge, no *fully analytical* expression as ours (3.7) exists in the literature. It is mainly due to the fact that the traditional ODE approach adopted in the literature involves massive calculations, whereas our new approach avoids those calculations via a *change of measure* and eventually leads to a much more concise expression. More importantly, we will shortly see that, this fully analytic expression as purposely organised in the form (3.7) is *crucial* for us to discover an *exact* distributional decomposition that ultimately leads to our new algorithm for exact simulation.

3.2 Interarrival Time

Theorem 3.2. *The probability density function (PDF) of interarrival time τ conditional on X_t at the current arrival time t is given by*

$$f_{\tau|X_t}(s \mid X_t) \;=\; -P(s)Q(s)\Big(p(s)+q(s)\Big), \qquad s>0, \tag{3.9}$$

where

$$P(s) \;:=\; \frac{e^{-\kappa_1 X_t} e^{-\kappa_2 X_t^2} e^{-\kappa_0 s}}{\sqrt{1-2\tilde{\varsigma}^2\kappa_2}}, \tag{3.10}$$

$$Q(s) \;:=\; \exp\left(\frac{\tilde{\varsigma}^2\kappa_1^2+2\tilde{\nu}\kappa_1+2\tilde{\nu}^2\kappa_2}{2\left(1-2\tilde{\varsigma}^2\kappa_2\right)}\right), \tag{3.11}$$

and

$$p(s) \;:=\; -\left(\mu\tilde{\mu}+\frac{\tilde{\varsigma}^2}{1-2\tilde{\varsigma}^2\kappa_2}\right), \tag{3.12}$$

$$q(s) \;:=\; -\frac{\tilde{\delta}\left(\kappa_1+2\tilde{\nu}\kappa_2\right)}{\left(1-2\tilde{\varsigma}^2\kappa_2\right)^2}\left[\left(\tilde{\nu}-\tilde{\mu}\right)\left(1-2\tilde{\varsigma}^2\kappa_2\right)-\frac{\tilde{\sigma}^2 e^{-2\tilde{\delta}s}}{2\tilde{\delta}}\left(\kappa_1+2\tilde{\nu}\kappa_2\right)\right] \tag{3.13}$$

are the (partial) derivatives of $\ln P(s)$ *and* $\ln Q(s)$*, respectively, with* $(\tilde{\nu},\tilde{\varsigma})$ *specified in* (3.8)*, parameters* $\left(\tilde{\mu},\tilde{\delta},\tilde{\sigma}\right)$ *under the new measure* $\tilde{\mathbb{P}}$ *specified in* (3.6) *by setting* $w=1$*, i.e.*

$$\tilde{\delta} \;:=\; \sqrt{\delta^2+2\sigma^2}, \qquad \tilde{\mu} \;:=\; \frac{\delta^2}{\tilde{\delta}^2}\mu, \qquad \tilde{\sigma} \;:=\; \sigma, \tag{3.14}$$

and $(\kappa_2,\kappa_1,\kappa_0)$ *specified in (3.4) by setting* $w=1$*, i.e.,*

$$\kappa_2 \;:=\; \frac{\tilde{\delta}-\delta}{2\tilde{\sigma}^2} \;\equiv\; \frac{1}{\tilde{\delta}+\delta}, \quad \kappa_1 \;:=\; \frac{2\mu\delta\kappa_2}{\tilde{\delta}} \;\equiv\; \frac{2\tilde{\mu}\tilde{\delta}\kappa_2}{\delta}, \quad \kappa_0 \;:=\; \mu\tilde{\mu}+\tilde{\sigma}^2\kappa_2. \tag{3.15}$$

3.3 Pre-Event Intensity and State

Theorem 3.3. *Conditional on X_t at the current arrival time t and interarrival time $\tau = s$, the Laplace transform of pre-event intensity level λ_{t+s^-} is given by*

$$\mathbb{E}\left[e^{-u\lambda_{t+\tau^-}}\middle|\tau = s, X_t\right] = \frac{\Psi'(u)}{\Psi'(0)}, \tag{3.16}$$

where

$$\Psi(u) := \frac{1}{\sqrt{2\tilde{\varsigma}^2 u + (1 - 2\tilde{\varsigma}^2\kappa_2)}} \exp\left(\frac{\tilde{\varsigma}^2}{2\big[2\tilde{\varsigma}^2 u + (1 - 2\tilde{\varsigma}^2\kappa_2)\big]}\left(\kappa_1 + \frac{\tilde{\nu}}{\tilde{\varsigma}^2}\right)^2\right), \tag{3.17}$$

where $(\tilde{\nu}, \tilde{\varsigma})$ are specified in (3.8) with $(\tilde{\mu}, \tilde{\delta}, \tilde{\sigma})$ in (3.14) and $(\kappa_2, \kappa_1, \kappa_0)$ in (3.15).

Given the pre-event intensity level λ_{t+s^-}, the associated position X_{t+s^-} of state process could be either positive or negative. Proposition 3.3 below gives the probability of the sign of X_{t+s^-} when λ_{t+s^-} and X_t are known.

Proposition 3.3. *Conditional on X_t at the current arrival time t and the pre-event intensity level $\lambda_{t+s^-} = \lambda \in \mathbb{R}^+$, we have*

$$\begin{aligned}
\mathbb{P}\left(X_{t+s^-} = \sqrt{\lambda}\,\middle|\,\lambda_{t+s^-} = \lambda, X_t\right) &= \frac{e^{-\frac{(\sqrt{\lambda}-\nu)^2}{2\varsigma^2}}}{e^{-\frac{(\sqrt{\lambda}-\nu)^2}{2\varsigma^2}} + e^{-\frac{(\sqrt{\lambda}+\nu)^2}{2\varsigma^2}}}, \\
\mathbb{P}\left(X_{t+s^-} = -\sqrt{\lambda}\,\middle|\,\lambda_{t+s^-} = \lambda, X_t\right) &= \frac{e^{-\frac{(\sqrt{\lambda}+\nu)^2}{2\varsigma^2}}}{e^{-\frac{(\sqrt{\lambda}-\nu)^2}{2\varsigma^2}} + e^{-\frac{(\sqrt{\lambda}+\nu)^2}{2\varsigma^2}}},
\end{aligned}$$

where ν and ς are specified by (2.5).

4 Exact Simulation Algorithms

Based on the distributional properties derived in Section 3, exact simulation algorithms for a point process with quadratic intensity can be developed in this section. More precisely, we can develop exact simulation algorithms for the interarrival time τ, the pre-event intensity level $\lambda_{t+\tau^-}$ and the pre-event state position $X_{t+\tau^-}$ based on Theorem 3.2, Theorem 3.3 and Proposition 3.3, respectively.

4.1 Exact Simulation of Interarrival Time

We offer two algorithms for exactly simulating the interarrival time τ based on Theorem 3.2 and integrate them into Algorithm 4.1:

Algorithm 4.2 and 4.3: When the parameter setting $(X_t, \mu, \delta, \sigma)$ of state process satisfies certain conditions, we discover that the interarrival time τ can be expressed simply as the minimum of two random variables, V^* and V_t^*, both of which can be exactly simulated via Algorithm 4.2 and Algorithm 4.3, respectively. Then, we obtain the exact simulation scheme for interarrival time τ in Algorithm 4.1.

Algorithm 4.4: When the conditions for parameters in Algorithm 4.1 do not hold, we switch to Algorithm 4.4 for exactly simulating the interarrival time τ *in general*.

These algorithms are provided as below.

Algorithm 4.1 Exact Simulation for Interarrival Time τ

```
$a^* \leftarrow -\frac{2\tilde{\mu}\tilde{\sigma}^2\kappa_2 X_t}{\delta}$
$b^* \leftarrow X_t\,(X_t - 2\tilde{\mu})$
$c^* \leftarrow \frac{\tilde{\mu}X_t}{\delta\kappa_2}$
        ▷ with $\left(\tilde{\mu}, \tilde{\delta}, \tilde{\sigma}\right)$ and $(\kappa_2, \kappa_1, \kappa_0)$ specified respectively in (3.14) and (3.15)
Condition 1 $\leftarrow \left(\kappa_1 X_t + \kappa_2 X_t^2 > 0\right)$
Condition 2a $\leftarrow \left(a^* > 0\right)$ && $\left((b^*)^2 - 4a^*c^* < 0\right)$
Condition 2b $\leftarrow \left(a^* > 0\right)$ && $\left(-\frac{b^*}{2a^*} > 1\right)$ && $\left(a^* + b^* + c^* > 0\right)$
Condition 2c $\leftarrow \left(a^* > 0\right)$ && $\left(-\frac{b^*}{2a^*} < 0\right)$ && $\left(c^* > 0\right)$
Condition 2d $\leftarrow \left(a^* < 0\right)$ && $\left(c^* > 0\right)$ && $\left(a^* + b^* + c^* > 0\right)$
Condition 2 $\leftarrow$ Condition 2a || Condition 2b || Condition 2c || Condition 2d
if (Condition 1 && Condition 2) then
    sample $V^*$ via Algorithm 4.2
    sample $V_t^*$ via Algorithm 4.3
    $\tau \leftarrow \min\{V^*, V_t^*\}$
else
    sample $\tau$ via Algorithm 4.4
end if
return $\tau$
```

Algorithm 4.2 Exact Simulation for Random Variable V^*

```
1: $R_{V^*}(\lambda, s) \leftarrow \frac{e^{\lambda s} f_{V^*}(s)}{\lambda}$
                                   $\triangleright$ with the PDF $f_{V^*}(s)$ given analytically by (J.4)
2: $\lambda^* \leftarrow \underset{\lambda \in \mathbb{R}^+}{\arg\min} \left( \max_{s \in \mathbb{R}^+} R_{V^*}(\lambda, s) \right)$
3: $C_{V^*} \leftarrow \max_{s \in \mathbb{R}^+} R_{V^*}(\lambda^*, s)$
                                   $\triangleright$ by numerical optimisation
4: repeat
5:     sample $V^* \sim \mathsf{Exp}(\lambda^*)$
6:     sample $U \sim \mathsf{U}(0,1)$
7: until $\left( U \leq \frac{R_{V^*}(\lambda^*, V^*)}{C_{V^*}} \right)$
8: return $V^*$
```

Note that, V^* is independent of X_t, which can be seen from its CDF (J.3) or its PDF (J.4). The associated optimisation results λ^* and C_{V^*} depend only on parameters (μ, δ, σ) and will not change with respect to X_t. Therefore, for a given parameter setting, the definition of R_{V^*} in Step 1 and maximisation in Step 2 and 3 of Algorithm 4.2 need to be numerically solved *only once* prior to the implementation of *all* simulations for V^*. In fact, this maximisation optimises the Accept/Reject (A/R) scheme[7] for *exact* simulation.

Algorithm 4.3 Exact Simulation for Random Variable V_t^*

```
1: sample $U \sim \mathsf{U}[0,1]$
2: if $\left( U \leq e^{-\kappa_1 X_t - \kappa_2 X_t^2} \right)$ then
3:     $V_t^* \leftarrow +\infty$
4: else
5:     $Z \leftarrow \ln U + \kappa_1 X_t + \kappa_2 X_t^2$
6:     $a \leftarrow \kappa_2 X_t (X_t - 2\tilde{\mu}) - \frac{\tilde{\sigma}^2 \kappa_2}{\tilde{\delta}} Z$
7:     $b \leftarrow \frac{2\tilde{\mu} X_t}{\delta}$
8:     $c \leftarrow -\frac{Z}{2\tilde{\delta}\kappa_2}$
9:     $V_t^* \leftarrow -\frac{1}{\tilde{\delta}} \ln \left( \frac{-b+\sqrt{b^2-4ac}}{2a} \right)$
10: end if
11: return $V_t^*$
```

We carry out numerical tests to verify Algorithm 4.1 based on 10^5 replications under parameter settings $(X_t, \mu, \delta, \sigma) = (0.5, 0.8, 1.1, 1.3)$, and $(-0.5, -0.8, 1.1, 1.3)$ which satisfy $\mathbf{Condition\ 1}$ and any of $\mathbf{Condition\ 2a}-\mathbf{2d}$. The comparisons for the *empirical* CDF and PDF with the asso-

[7] For introducing the Accept/Reject (A/R) method in general, see e.g. Glasserman (2003, §2.2.2) and Asmussen and Glynn (2007, §II.2b).

ciated *true* (theoretical) CDF (3.9) and PDF (G.2) are presented in Figure 3, respectively. We can see that empirical CDFs and PDFs are well fitted to their true counterparts.

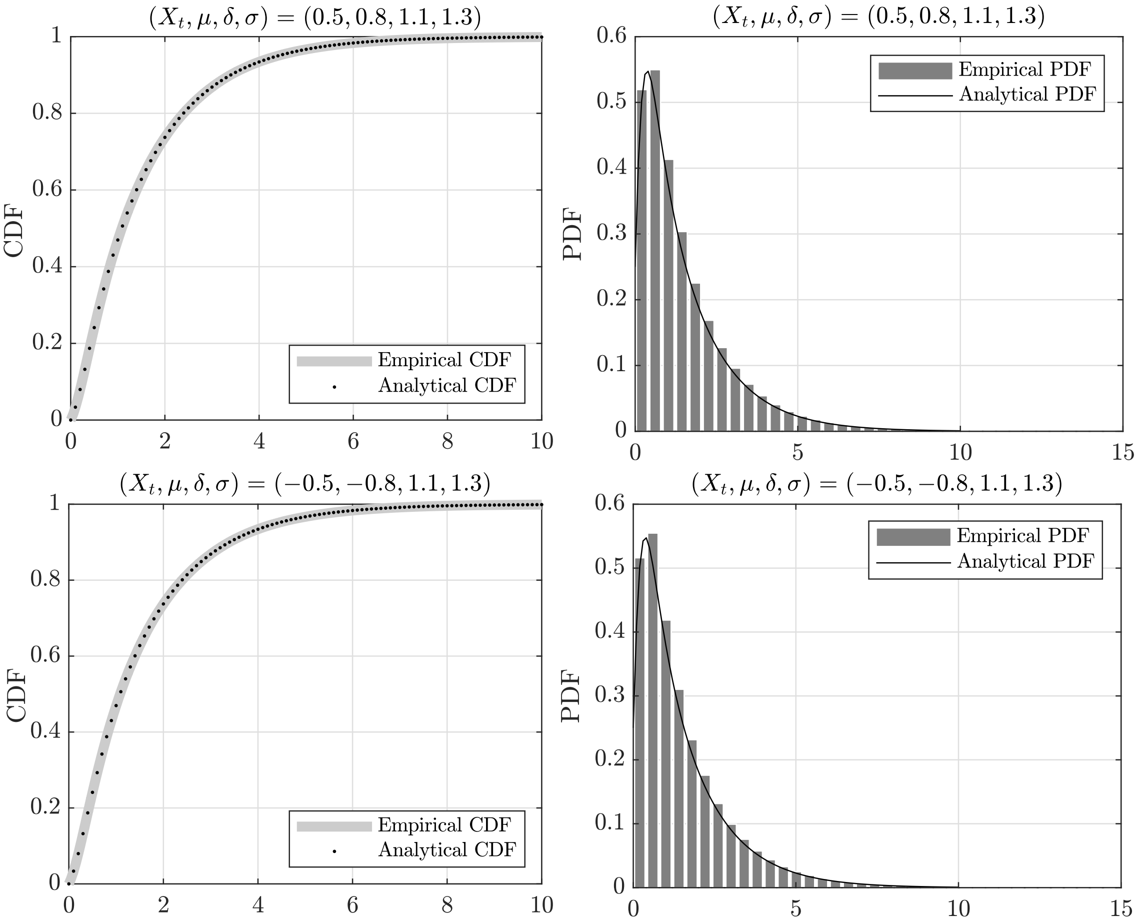

Figure 3: Comparison of the *empirical* CDF and PDF for Algorithm 4.1 based on 10^5 replications with the *true* (theoretical) CDF (G.2) and PDF (3.9) of interarrival time τ under parameter settings $(X_t, \mu, \delta, \sigma) = (0.5, 0.8, 1.1, 1.3)$, and $(-0.5, -0.8, 1.1, 1.3)$, respectively.

Alternatively, we could implement Algorithm 4.4 below to exactly simulate the interarrival time in general.

Algorithm 4.4 Exact Simulation for Interarrival Time τ (Directly through PDF)

```
1: $R_\tau(s) \;\leftarrow\; -\frac{e^{\kappa_0 s}}{\kappa_0} P(s)Q(s)\Big(p(s)+q(s)\Big)$
        ▷ with $P(s)$, $Q(s)$, $p(s)$ and $q(s)$ provided in (3.10), (3.11), (3.12) and (3.13) respectively and $\kappa_0$ specified in (3.15)
2: $C_M \;\leftarrow\; \max_{0<s<+\infty}\{R_\tau(s)\}$
        ▷ by numerical optimisation
3: $C_0 \;\leftarrow\; \frac{1}{\kappa_0}\left(\tilde{\delta}\left(\kappa_1+2\kappa_2 X_t\right)\Big[\left(X_t-\tilde{\mu}\right)-\frac{\tilde{\sigma}^2}{2\tilde{\delta}}\left(\kappa_1+2\kappa_2 X_t\right)\Big]+\mu\tilde{\mu}\right)$
4: $C_\infty \;\leftarrow\; \sqrt{2\tilde{\delta}\kappa_2}e^{-\kappa_1 X_t-\kappa_2 X_t^2}e^{\frac{\tilde{\mu}^2\tilde{\delta}\kappa_2^2}{\delta^2}\left(\tilde{\delta}+\delta\right)\left(\tilde{\delta}+2\delta\right)}$
        ▷ with $\left(\tilde{\mu},\tilde{\delta},\tilde{\sigma}\right)$ specified in (3.14) and $\left(\kappa_2,\kappa_1,\kappa_0\right)$ specified in (3.15)
5: $C_\tau \;\leftarrow\; \max\{C_0, C_M, C_\infty\}$
6: repeat
7:     sample $\tau \sim \mathsf{Exp}\left(\kappa_0\right)$
8:     sample $U \sim \mathsf{U}[0,1]$
9: until $\left(U \le \dfrac{R_\tau(\tau)}{C_\tau}\right)$
10: return $\tau$
```

For a given parameter setting, the maximisation[8] in Step 2 of Algorithm 4.4 needs to be numerically solved *only once* prior to the implementation of *all* A/R simulations. For example, for a given parameter setting $(X_t,\mu,\delta,\sigma)=(0.5,0.8,1.1,1.3)$, we can numerically find C_M as illustrated in Figure 4. To demonstrate the accuracy of Algorithm 4.4 for simulating the interarrival time τ, we carry out numerical validations by comparing the *empirical* CDF and PDF (via Algo-

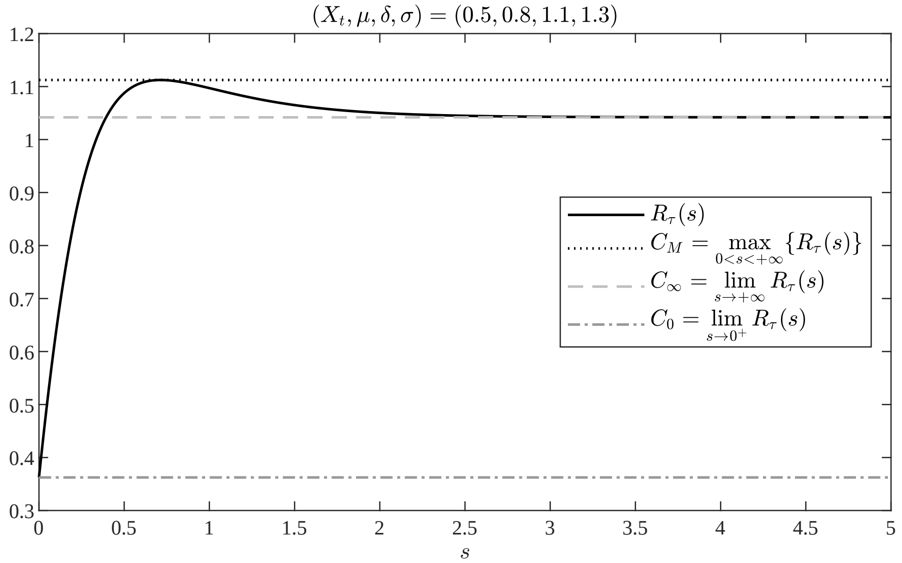
$(X_t,\mu,\delta,\sigma)=(0.5,0.8,1.1,1.3)$

Figure 4: Numerical maximisation in Step 2 of Algorithm 4.4.

[8] The algorithm for numerical maximisation adopted here is Brent (1973) which is directly available in MatLab (function `fminbnd`) or R (function `optimize`).

rithm 4.4) with the *true* CDF (3.9) and PDF (G.2) as presented in Figure 5. Overall, we observe that Algorithm 4.4 can achieve a high level of accuracy.

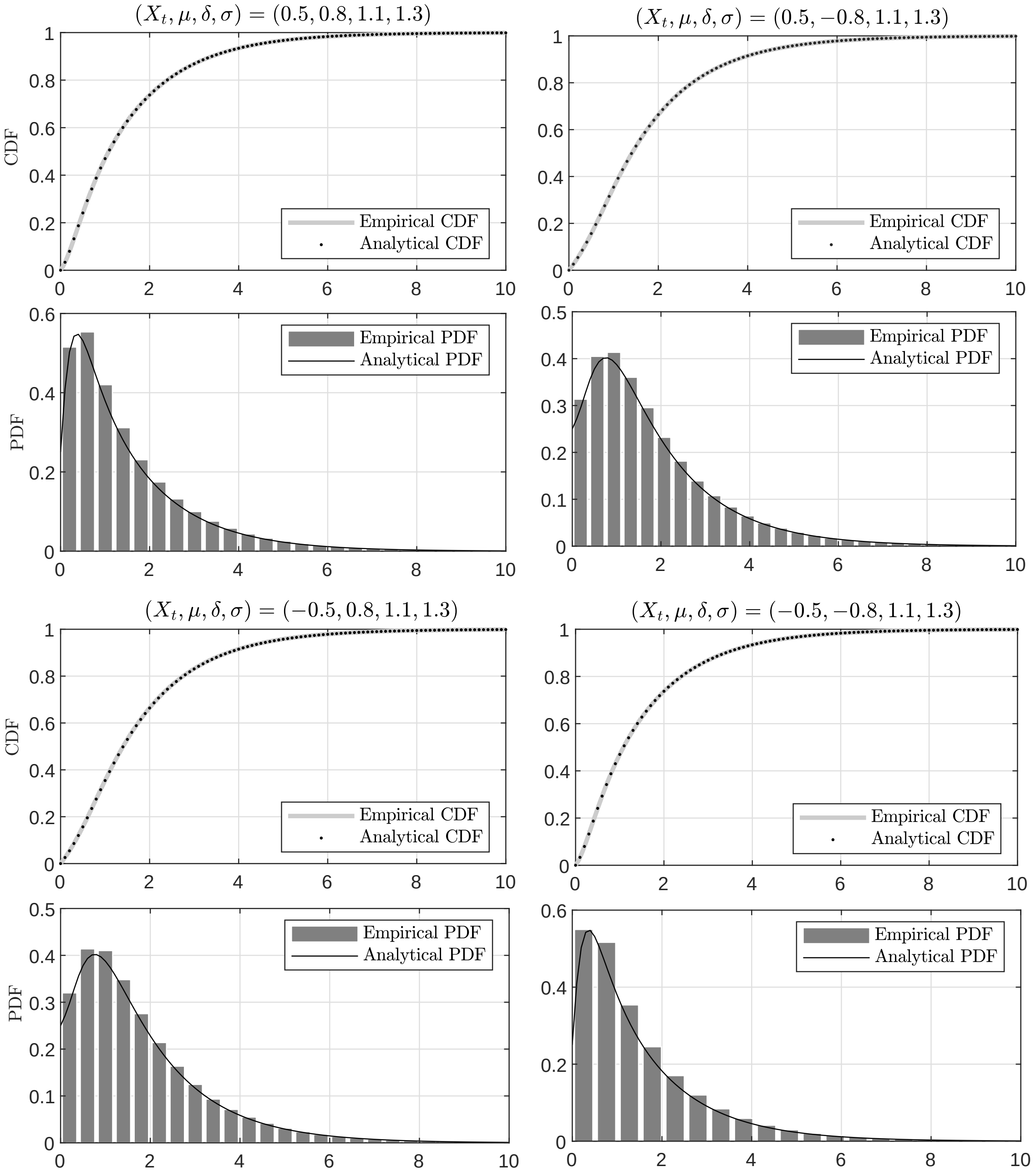

Figure 5: Comparisons between the *empirical* CDF/PDF of interarrival time τ based on Algorithm 4.4 with 10^5 replications and the associated *true* CDF (3.9) and PDF (G.2) under parameter settings $(X_t, \mu, \delta, \sigma) = (0.5, 0.8, 1.1, 1.3)$, $(0.5, -0.8, 1.1, 1.3)$, $(-0.5, 0.8, 1.1, 1.3)$ and $(-0.5, -0.8, 1.1, 1.3)$, respectively.

4.2 Exact Simulation of Pre-Event Intensity and State

Conditional on the state position X_t at the current arrival time t and the realisation of interarrival time τ (via Algorithm 4.1), we can develop an exact scheme for simulating the (next) pre-event intensity level $\lambda_{t+\tau^-}$ and state position $X_{t+\tau^-}$ in Algorithm 4.5.

Algorithm 4.5 Exact Simulation of Pre-Event Intensity and State

Conditional on the state position X_t at the current arrival time t and the realisation of interarrival time τ, the (next) pre-event intensity level $\lambda_{t+\tau^-}$ and state position $X_{t+\tau^-}$ can be exactly simulated via the following steps:

```
$\tilde{\nu} \leftarrow X_t e^{-\tilde{\delta}\tau} + \tilde{\mu}\left(1 - e^{-\tilde{\delta}\tau}\right)$
$\tilde{\varsigma} \leftarrow \tilde{\sigma}\sqrt{\frac{1-e^{-2\tilde{\delta}\tau}}{2\tilde{\delta}}}$
                                        ▷ with $\left(\tilde{\mu}, \tilde{\delta}, \tilde{\sigma}\right)$ specified in (3.14)
$A \leftarrow \frac{\tilde{\varsigma}^2}{2}\left(\kappa_1 + \frac{\tilde{\nu}}{\tilde{\varsigma}^2}\right)^2$
$B \leftarrow 2\tilde{\varsigma}^2$
$C \leftarrow 1 - 2\tilde{\varsigma}^2\kappa_2$
                                        ▷ with $(\kappa_2, \kappa_1)$ specified in (3.15)
$p_\lambda \leftarrow \frac{\frac{A}{C}}{\frac{A}{C}+\frac{1}{2}}$
sample $B_\lambda \sim \mathsf{Bernoulli}\,(p_\lambda)$
sample $J \sim \mathsf{Poisson}\left(\frac{A}{C}\right)$
                                        ▷ with the probability mass function (PMF)
    $\mathbb{P}(J = j) = \frac{\left(\frac{A}{C}\right)^j}{j!} e^{-\frac{A}{C}}, \qquad j = 0, 1, \cdots$
sample $\lambda_{t+\tau^-} \sim \mathsf{Gamma}\left(J + \frac{3}{2} + B_\lambda, \frac{C}{B}\right)$
                                        ▷ with the PDF
    $\mathbb{P}\left(\lambda_{t+\tau^-} = x\right) = \frac{\left(\frac{C}{B}\right)^{J+\frac{3}{2}+B_\lambda}}{\Gamma\left(J + \frac{1}{2} + B_\lambda\right)} x^{J+\frac{3}{2}+B_\lambda} e^{-\frac{C}{B}x}, \qquad x > 0$
return $\lambda_{t+\tau^-}$
```

Conditional on the pre-event intensity level $\lambda_{t+\tau^-}$, the pre-event state position $X_{t+\tau^-}$ can be simulated via the following procedures:

```
$\nu \leftarrow X_t e^{-\delta\tau} + \mu\left(1 - e^{-\delta\tau}\right)$
$\varsigma \leftarrow \sigma\sqrt{\frac{1-e^{-2\delta\tau}}{2\delta}}$
$X \leftarrow \sqrt{\lambda_{t+\tau^-}}$
$p_X \leftarrow \frac{e^{-\frac{(X-\nu)^2}{2\varsigma^2}}}{e^{-\frac{(X-\nu)^2}{2\varsigma^2}} + e^{-\frac{(X+\nu)^2}{2\varsigma^2}}}$
sample $B_X \sim \mathsf{Bernoulli}\,(p_X)$
$X_{t+\tau^-} \leftarrow (2B_X - 1)\,X$
return $X_{t+\tau^-}$
```

We can numerically verify our exact scheme for simulating the pre-event intensity level by comparing the CDF and PDF estimated by our Algorithm 4.5 and by the direct numerical inversion of Laplace transform (3.16) of pre-event intensity level.[9] The associated plots of CDFs and PDFs under different parameter settings are illustrated in Figure 6, where we can observe that simulation-estimated CDFs and PDFs are well fitted to their direct numerical inversions.

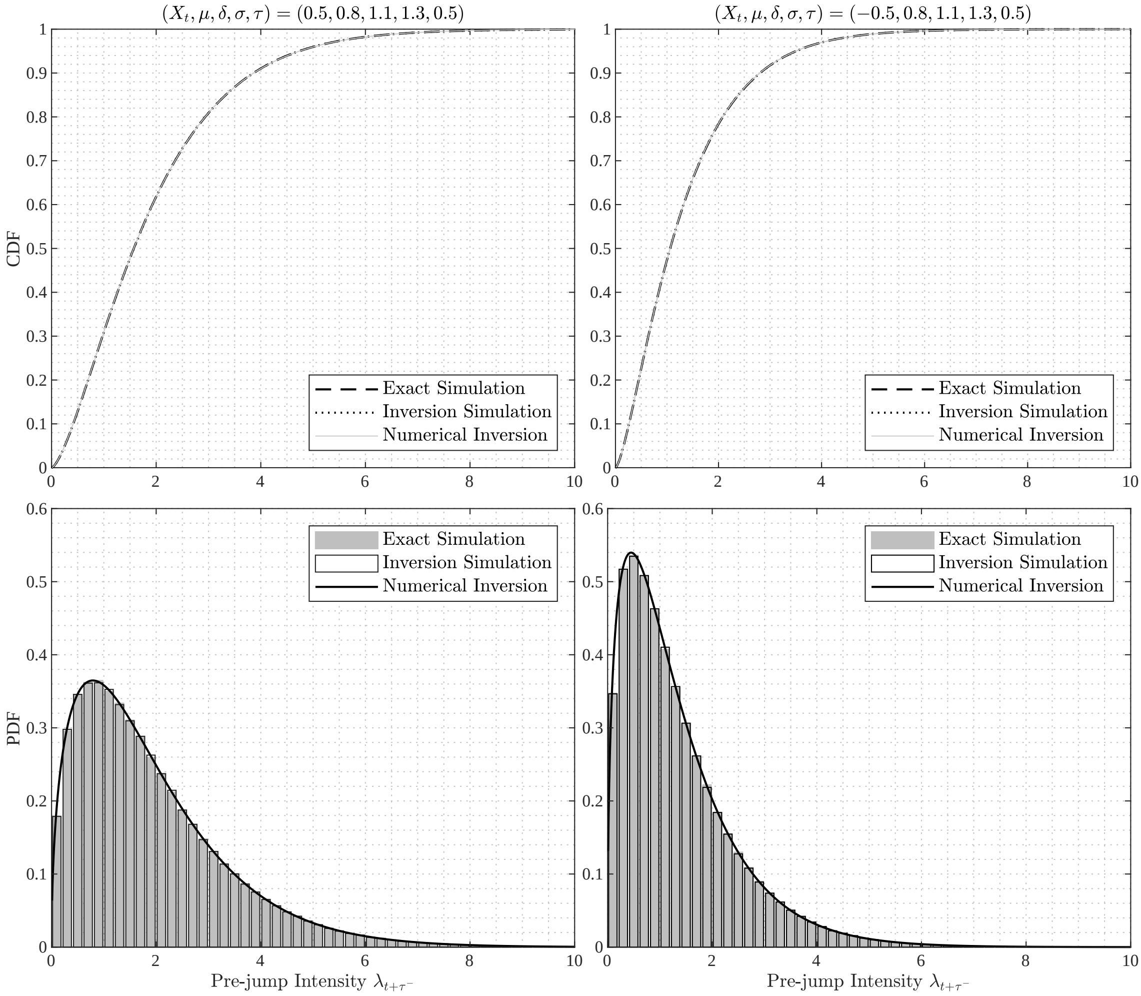

Figure 6: Comparison of the distribution functions of the pre-event intensity level $\lambda_{t+\tau^-}$ estimated by three methods, 1) our exact simulation of Algorithm 4.5 based on 10^5 replications, 2) simulation of numerical Laplace inversion using Ridout (2009) based on 10^5 replications, 3) direct numerical inversion of Laplace transform (3.16) without simulation, under parameter settings $(X_t, \mu, \delta, \sigma, \tau) = (0.5, 0.8, 1.1, 1.3, 0.5)$ and $(-0.5, 0.8, 1.1, 1.3, 0.5)$, respectively.

Alternatively, the pre-event intensity level can be simulated via numerically inverting Laplace transform (3.16). Note that, however, it would not practically exact, since numerical Laplace inversion involves truncation-based approximations which additionally induce truncation errors and

[9] A variety of methods are available for numerically inverting Laplace transform with high accuracy, such as Gaver (1966), Stehfest (1970), Abate and Whitt (1992, 1995, 2006) and Abate et al. (2000). Here, we adopt the *Euler algorithm* of Abate and Whitt (2006, §5) and Abate et al. (2000, §8), see more details in Appendix O.

discretisation errors.[10] We provide numerical comparisons[11] in CPU time (in seconds) between our exact simulation (Algorithm 4.5) and numerical Laplace inversion for simulating the pre-event (pre-jump) intensity level $\lambda_{t+\tau-}$ under parameter settings $(X_0, \mu, \delta, \sigma, t) = (0.5, 0.8, 1.1, 1.3, 0.5)$ and $(-0.5, 0.8, 1.1, 1.3, 1)$, respectively, in Table 1. The associated plots of simulation-estimated CDFs and PDFs under different parameter settings are also illustrated in Figure 6. Here, a recently-developed simulation algorithm by Ridout (2009) for simulating a general random variable via numerical Laplace inversion is directly adopted. Ridout (2009) shows that his algorithm is more efficient than alternatives, such as Devroye (1981). However, the associated code[12] is written in a loop-based format, which cannot be easily vectorised. In terms of simulating the pre-event intensity level $\lambda_{t+\tau-}$ as one random variable defined by the conditional Laplace transform (3.16) in particular, it is straightforward to vectorise our Algorithm 4.5, making our algorithm much faster than the loop-based algorithm of Ridout (2009) for numerically inverting Laplace transform (3.16) as evident in Table 1. If we also adopt the loop-based format for our Algorithm 4.5 for a fairer comparison, our Algorithm 4.5 is still about 5 times faster than the Laplace inversion.

Table 1: Numerical comparisons in CPU time (in seconds) between our exact simulation (Algorithm 4.5) and numerical Laplace inversion for simulating the pre-event (pre-jump) intensity level $\lambda_{t+\tau-}$ under parameter settings $(X_0, \mu, \delta, \sigma, t) = (0.5, 0.8, 1.1, 1.3, 0.5)$ and $(-0.5, 0.8, 1.1, 1.3, 1)$, respectively.

Parameters	$(X_0, \mu, \sigma, \delta, t) = (0.5, 0.8, 1.1, 1.3, 0.5)$				$(X_0, \mu, \sigma, \delta, t) = (-0.5, 0.8, 1.1, 1.3, 1)$			
Paths	15,625	62,500	250,000	1,000,000	15,625	62,500	250,000	1,000,000
Inversion (Ridout, 2009)	1.50	4.73	17.90	66.06	1.54	5.02	22.87	83.98
Exact (Loop)	0.24	0.94	3.32	13.77	0.28	0.94	4.42	16.94
Exact (Vectorised)	0.01	0.02	0.07	0.27	0.01	0.02	0.07	0.26

4.3 Exact Simulation of Quadratic Intensity Models

By integrating Algorithms 4.1 and 4.5, it is then straightforward to exactly simulate the entire path of a point process with quadratic intensity recursively as summarised by Algorithm 4.6. For a point process N_t with quadratic intensity $\lambda_t = X_t^2$ and initial value $X_0 \in \mathbb{R}$, a path of N_t can be simulated exactly via the following steps recursively based on the current arrival time $t = 0$.

[10] This *generic* problem of numerical Laplace inversion for *exact* simulation was also pointed out in the literature, e.g. Li and Wu (2019, p.771).

[11] The experiments are conducted by R on a laptop PC with an Intel Core i7-6500U CPU@2.50GHz processor, 8.00GB of RAM, Windows 10 Home, and 64-bit operating system.

[12] The R code by Ridout (2009) is publicly available: https://www.kent.ac.uk/smsas/personal/msr/rlaptrans.html.

Algorithm 4.6

```
$T \leftarrow 0$
$N_T \leftarrow 0$
$X_T \leftarrow X_0$
repeat
    sample $\tau$ via Algorithm 4.1
    sample $\lambda_{T+\tau^-}$ and $X_{T+\tau^-}$ via Algorithm 4.5
    $T \leftarrow T+\tau$
    $X_T \leftarrow X_{T+\tau^-}$
    $N_T \leftarrow N_T+1$
until $\big(T > t\big)$
$N_t \leftarrow N_T - 1$
return $N_t$
```

For simulating the interarrival time τ in general, Algorithm 4.2 and 4.3 are more efficient than Algorithm 4.4, since with the aid of parameter combination we could strategically avoid the numerical optimisation within the recursive loop in Algorithm 4.6 with Algorithm 4.1 for simulating point processes. The details of numerical tests and comparisons will be carried out later in Section 6.

5 Extensions

Once the canonical model is found *exactly simulatable* via Algorithm 4.6, it is then easy to make further extensions to more comprehensive quadratic intensity models. In this section, we present the generality of our scheme, by further developing algorithms for exactly simulating *extended* models with quadratic intensity, such as exogenous and endogenous jumps in intensity processes, and multidimensional point processes with contemporaneous jumps (cojumps). These extended models are also new to the literature. Our newly developed algorithms here could potentially facilitate and broaden their applications in practice. The associated simulation algorithms are outlined in Appendix P, and applications to portfolio loss modelling are provided in Section 6.3.

5.1 Jump-Quadratic Intensity Models

The canonical quadratic intensity model of Definition 2.2 studied in previous sections has no jump component in the intensity, and it belongs to the big family of *doubly stochastic Poisson processes* or *Cox processes* (Cox, 1955). By definition, these point arrivals are independent conditional on the realisations of state variables. This key property makes the associated models more analytically tractable. However, it may be still too restrictive to capture more complex dependence structures in the real world, such as "contagious" defaults which have been documented recently by e.g. Das et al. (2007), Duffie et al. (2007, 2009), Benzoni et al. (2015) and Azizpour et al. (2018).

In this section, we extend the canonical quadratic intensity model beyond the classical Cox family by adding jumps in the intensity process, in particular, *endogenous* jumps with *self-exciting* and *self-correcting* effects, where the associated distributional functions (such as the PMF of N_t) in analytical forms are no longer available and efficient simulation algorithms are more desirable.[13] As specified in Definition 5.1 below, these jumps are endogenous, since they occur *simultaneously* in the state process X_t, intensity process λ_t and counting process N_t. These extra two types of endogenous jumps with arbitrary sizes would equip additional flexibilities for modelling event arrivals with endogenous impacts such as contagion and stress release.[14]

Definition 5.1 (Quadratic Intensity Model with Self-Exciting and Self-Correcting Effects)**.** *A quadratic intensity model with self-exciting and self-correcting effects is a point process*

$$N_t = \sum_{i\geq 1} \mathbb{1}_{\{T_i \leq t\}},$$

with stochastic intensity $\lambda_t = X_t^2$ *and*

$$X_t = \underbrace{X_0 e^{-\delta t} + \mu\left(1-e^{-\delta t}\right) + \sigma \int_0^t e^{-\delta(t-u)} \mathrm{d}W_u}_{\textit{Gaussian OU Diffusion}} + \underbrace{\sum_{0\leq T_i<t} Y_i e^{-\delta(t-T_i)}}_{\textit{Two-Sided Jumps}}, \quad t \geq 0, \tag{5.1}$$

where $\{Y_i\}_{i=1,2,\cdots}$ *are jump sizes in state process* X_t*, a sequence of random variables with an arbitrary* $\mathcal{F}_{T_i^-}$*-measurable CDF* $G(y), y \in \mathbb{R}$*, occurring at the associated ordered arrival times* $\{T_i\}_{i=1,2,\cdots}$*, respectively.*

Note that, jump sizes being $\mathcal{F}_{T_i^-}$-measurable means that, the functional form of the CDF $G(y)$ is revealed *just before* the arrival time T_i. This distribution could have a highly general dependency structure $G(y) = G(y \mid \cdot)$. For example, it could depend on the initial state X_0, the past history of state process at or just before the jump arrival times $\{T_k\}_{k=1,2,\cdots,i}$, all past jump sizes

[13] In particular, self-correcting and stress-release point processes are even less analytically tractable than self-exciting point processes, see e.g. Isham and Westcott (1979), Ogata and Vere-Jones (1984), Vere-Jones and Ogata (1984) and Lee et al. (2022).

[14] Here, "stress release" means a type of events that the occurrence of past events *inhibits* the occurrence of future events, see e.g. Isham and Westcott (1979), Ogata and Vere-Jones (1984), Vere-Jones and Ogata (1984) and Lee et al. (2022). Conceptually, "stress release" is just opposite to "contagion" which means a type of events that the occurrence of past events *accelerates* the occurrence of future events.

$\{Y_k\}_{k=1,2,\cdots,i-1}$, the cumulated number of jumps N_t, and so on. As long as we can record these above information, we can express $G(y)$ as the form,

$$G(y) \;=\; G\left(y \mid T_1, T_2, \cdots, T_i, X_0, X_{T_1^-}, \cdots, X_{T_i^-}, X_{T_1}, \cdots, X_{T_{i-1}}, Y_1, Y_2, \cdots, Y_{i-1}\right). \tag{5.2}$$

Since $y \in \mathbb{R}$, the jumps in the intensity process λ_t of this extended model with Definition 5.1 are allowed to be *two-sided*. If the realised jump in the intensity is *upward*, it is *self-exciting* similarly as Hawkes (1971); if the realised jump in the intensity is *downward*, it is *self-correcting* similarly as Isham and Westcott (1979). There is a great flexibility in modelling jump sizes $\{Y_i\}_{i=1,2,\cdots}$. Notice that, these jump sizes can be either positive or negative, fixed or random with an *arbitrary* $\mathcal{F}_{T_i^-}$-measurable distribution, or even more general to be functions of any previous information such as the functional form (5.2). Of course, when they are all equal to zero, the model reduces to the canonical version of Definition 2.2.

Since endogenous jumps only occur at the associated arrival times of a point process, we can extend Algorithm 4.6 to Algorithm P.1 by developing an exact simulation scheme for a quadratic intensity model with self-exciting and self-correcting effects.

We could further incorporate *exogenous* jumps with *externally-exciting* and *externally-correcting* effects into state process X_t, i.e.,

$$\begin{aligned} X_t \;=\; & \underbrace{X_0 e^{-\delta t} \;+\; \mu\left(1-e^{-\delta t}\right) \;+\; \sigma \int_0^t e^{-\delta(t-u)} \mathrm{d}W_u}_{\text{Gaussian OU Diffusion}} \\ & + \underbrace{\sum_{0 \le T_i < t} Y_i e^{-\delta(t-T_i)}}_{\text{Self-Exciting/Correcting Jumps}} \;+\; \underbrace{\sum_{0 \le S_k < t} Z_k e^{-\delta(t-S_k)}}_{\text{Externally-Exciting/Correcting Jumps}}, \end{aligned} \tag{5.3}$$

where $\{S_k\}_{k=1,2,\dots}$ are the arrival times of externally-exciting or externally-correcting jumps following a Poisson process of constant rate $\varrho > 0$, and $\{Z_k\}_{k=1,2,\dots}$ are the associated jump sizes with an arbitrary $\mathcal{F}_{S_k^-}$-measurable CDF $G_Z(z), z \in \mathbb{R}$. These additional jumps are exogenous, since they are arriving independently from point process N_t. By adjusting Algorithm P.1, we still can exactly simulating this extension via Algorithm P.2.

5.2 Multidimensional Cojump-Quadratic Intensity Models

Multidimensional point processes with quadratic intensities and *contemporaneous* jumps (or *cojumps*) can be also exactly simulated by further adjusting our original algorithms for univariate point processes. These cojumps in intensity processes could introduce self/mutually exciting and correcting effects into a dynamic system, leading to a much more complicated dependency among point processes in continuous time. More precisely, the multidimensional cojump-quadratic intensity model is defined as a D-dimensional point process $\left\{N_t^{(\jmath)}\right\}_{\jmath=1,2,\cdots,D}$, where the $\jmath^{\text{th}}$-component point process $N_t^{(\jmath)} \equiv \left\{T_i^{(\jmath)}\right\}_{i=1,2,\cdots}$ has its intensity process specified as $\lambda_t^{(\jmath)} = \left(X_t^{(\jmath)}\right)^2$ with the state process

$$X_t^{(\jmath)} = \underbrace{X_0^{(\jmath)} e^{-\delta_\jmath t} + \mu_\jmath\left(1 - e^{-\delta_\jmath t}\right) + \sigma_\jmath \int_0^t e^{-\delta_\jmath (t-u)} \mathrm{d}W_u^{(\jmath)}}_{\text{Gaussian OU Diffusion}} + \underbrace{\sum_{\ell=1}^{D} \sum_{0 \leq T_i^{(\ell)} < t} Y_i^{(\jmath,\ell)} e^{-\delta_\jmath\left(t - T_i^{(\ell)}\right)}}_{\text{Cojumps}}, \tag{5.4}$$

where $\left\{W_t^{(\jmath)}\right\}_{\jmath=1,2,\cdots,D}$ are mutually independent Brownian motions, $\left\{Y_i^{(\jmath,\ell)}\right\}_{\jmath=\ell}$ are sizes of self-exciting or self-correcting jumps occurring at time $T_i^{(\jmath)}$, and $\left\{Y_i^{(\jmath,\ell)}\right\}_{\jmath\neq\ell}$ are sizes of *mutual* jumps occurring at time $T_i^{(\ell)}$. In general, these *cojump* sizes $\left\{Y_i^{(\jmath,\ell)}\right\}_{\jmath,\ell=1,2,\cdots,D}$ follow an arbitrary $\mathcal{F}_{T_i^{(\ell)}-}$-measurable joint CDF $\mathbf{G}(\mathbf{y})$ for a vector $\mathbf{y} \in \mathbb{R}^D$. Depending on the directions of impacts to intensities, these mutual jumps could be *mutually-exciting*, *mutually-correcting* or even *mutually-reversing*. This flexility now can simultaneously allow for negative correlation and positivity among state variables, which cannot be captured by classical affine models as also mentioned at the introduction of this paper. By adjusting Algorithm P.1, we still can exactly simulating this multidimensional extension via Algorithm P.3. Similarly, we can further extend to simulate a multidimensional version with additional exogenous jumps in intensity processes by adjusting Algorithm P.2.

6 Numerical Experiments and Applications

In this section, we illustrate the performance of our exact scheme through numerical experiments. They are conducted by R or MatLab on a laptop PC with an Intel Core i7-6500U CPU@2.50GHz processor, 8.00GB of RAM, Windows 10 Home, and 64-bit operating system. The computing time is measured by the *elapsed CPU time* in seconds. The *true* (theoretical) value of conditional expectation is provided by (B.2), which is used for numerically validating and testing our algorithms. The errors from the associated true values are reported by three standard measures:

1. *Error* $=$ estimated value $-$ true value;
2. *Relative error (error %)* $= \dfrac{\text{estimated value} - \text{true value}}{\text{true value}}$;
3. *Root mean square error* (RMSE) $= \sqrt{\text{bias}^2 + \text{SE}^2}$, where the SE is the standard error of simulation output, and the bias is the difference between the expectation of estimator and the

associated true value. The bias is zero for exact scheme.[15]

6.1 Numerical Validation

First of all, we carry out the conventional convergence analysis for Algorithm 4.6 under four different parameter settings for $(X_0, \mu, \delta, \sigma, t)$ to demonstrate the accuracy and efficiency of our algorithms. In particular, we simulate in a fixed period $[0, t]$ under

- Case I: Algorithm 4.6 by applying Algorithm 4.1 in Step 5;
- Case II: Algorithm 4.6 by applying only Algorithm 4.4 (without Algorithm 4.2 and 4.3) in Step 5.

The simulated sample paths of point processes within a long period $[0, 500]$ and the associated histograms are plotted in Figure 7. The (log-log) plots for the RMSE against the CPU time are presented in Figure 8, and the associated numerical results are reported in Table 2. It is evident that our exact scheme can achieve a high level of accuracy and efficiency. We also observe that implementing Algorithm 4.6 under Case I is more efficient than the one under Case II. This is because, under certain combinations of $(X_t, \mu, \delta, \sigma)$ such that $\mathbf{Condition\ 1}$ and any of $\mathbf{Condition\ 2a{-}2d}$ in Algorithm 4.1 are satisfied, we could apply our decomposition scheme to sample the interarrival time τ without any numerical optimisation within the loops for simulating point processes, and thereby speed up the entire simulation in Algorithm 4.6.

6.2 Numerical Comparisons

In this section, we establish comparisons of numerical performance between our *exact scheme* with the classical *discretisation scheme* as well as the exact algorithm of *projection scheme* (Giesecke et al., 2011a), respectively.

Discretisation Scheme Conventionally, the discretisation scheme for simulating a point process is implemented via the *time-scaling method* (Meyer, 1971), see also e.g. Lando (1998), Das et al. (2007) and Lando and Nielsen (2010): Given the current arrival time $t = T$, the (next) interarrival time τ satisfies

$$\tau \overset{\mathcal{D}}{=} \inf\left\{ s \geq 0 : \int_T^{T+s} \lambda_u \mathrm{d}u \geq \mathcal{E} \right\}, \tag{6.1}$$

where $\mathcal{E}$ follows an exponential distribution of unit parameter, i.e., $\mathcal{E} \sim \mathsf{Exp}(1)$. The integral term $\int_T^{T+s} \lambda_u \mathrm{d}u$ is approximately by discretising the intensity process λ_t within the period $[T, T+s]$ using the Euler scheme. Therefore, the associated point process can be *approximately* simulated via

[15] Here, we adopt these error measures and the definition of RMSE in Giesecke et al. (2011a,b) and Giesecke and Smelov (2013) for consistent comparisons.

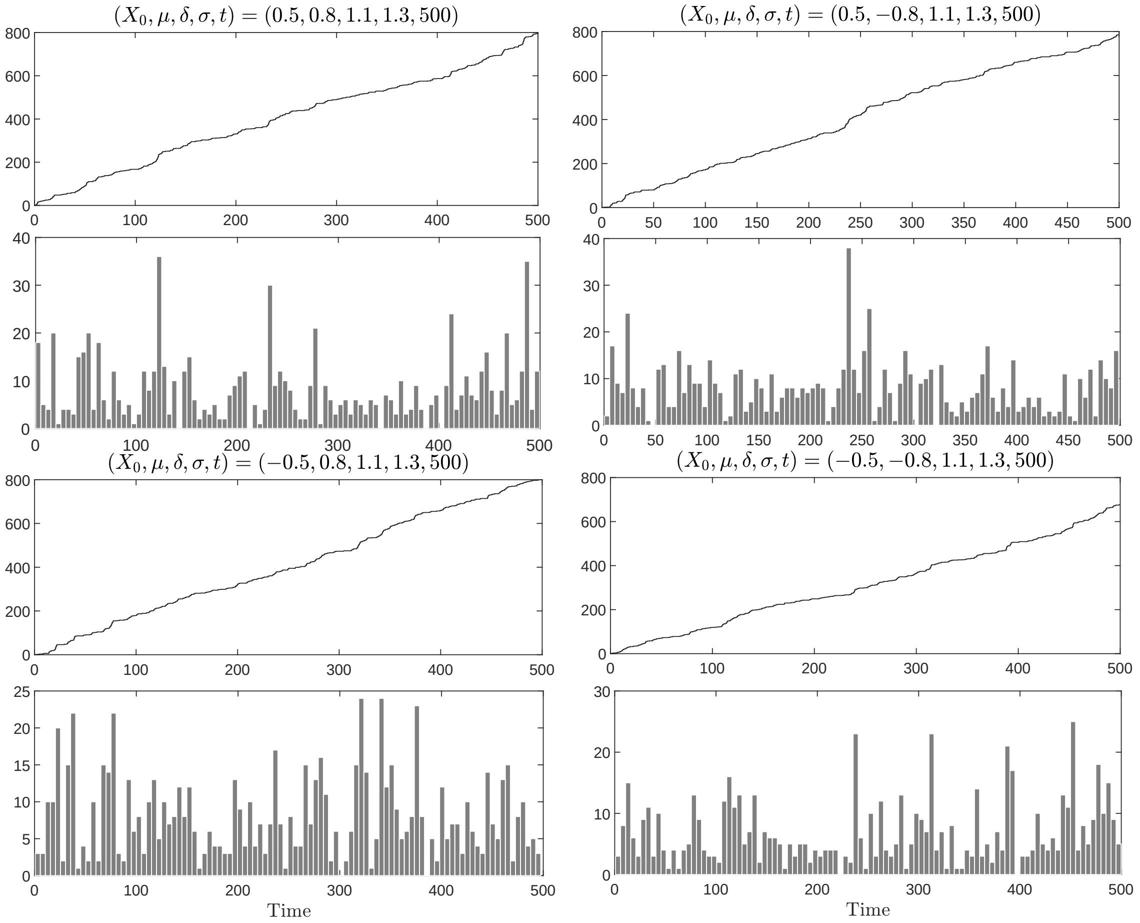

Figure 7: Simulated paths and histograms of point processes under parameter settings $(X_0, \mu, \delta, \sigma, t) = (0.5, 0.8, 1.1, 1.3, 0.5)$, $(0.5, -0.8, 1.1, 1.3, 0.5)$, $(-0.5, 0.8, 1.1, 1.3, 1)$ and $(-0.5, -0.8, 1.1, 1.3, 1)$, respectively.

(6.1) recursively. Based on the principle of optimal allocation for computation budget proposed by Duffie and Glynn (1995), we set the number of time-discretisation grids to be the square root of the sample-path number. The numerical comparison between our exact scheme (Algorithm 4.6 under Case I) and discretisation scheme under the parameter setting $(X_0, \mu, \delta, \sigma) = (0.5, 0.8, 1.1, 1.3)$ with $t = 0.5, 1, 2, 3$ are reported in Table 3. Figure 9 shows the comparison of the convergence of the RMSE against the CPU time graphically. We can see that the estimators obtained via our scheme have lower RMSE than the discretisation-based estimators, especially for a longer horizon. This indicates that our scheme outperforms the discretisation scheme in terms of convergence rate.

Projection Scheme Projection scheme proposed by Giesecke et al. (2011a) is theoretically based on a change of filtration such that the projected intensity simply has a time-deterministic dynamics which can be exactly computed. Then, the traditional inverse scheme or thinning scheme could be applied to exactly simulate point processes. Given the specification of projection scheme in Giesecke et al. (2011a, §5.2), we carry out a comparison between our exact scheme and their pro-

Table 2: Comparisons between Algorithm 4.6 under Case I and Case II under parameter settings $(X_0, \mu, \delta, \sigma, t) = (0.5, 0.8, 1.1, 1.3, 0.5)$, $(0.5, -0.8, 1.1, 1.3, 0.5)$, $(-0.5, 0.8, 1.1, 1.3, 1)$ and $(-0.5, -0.8, 1.1, 1.3, 1)$, respectively.

Paths	True	Simulation	Error	Error%	RMSE	CPU time	Simulation	Error	Error%	RMSE	CPU time
		$(X_0, \mu, \sigma, \delta, t) = (0.5, 0.8, 1.1, 1.3, 0.5)$									
				Case I					*Case II*		
15,625	0.3138	0.3110	-0.0028	-0.89%	0.0051	0.65	0.3158	0.0020	0.64%	0.00508	1.57
62,500	0.3138	0.3136	-0.0002	-0.07%	0.0025	4.74	0.3148	-0.0009	-0.29%	0.00254	12.37
250,000	0.3138	0.3139	0.0001	0.03%	0.0013	16.58	0.3143	0.0005	0.16%	0.00128	34.89
1,000,000	0.3138	0.3139	0.0001	0.02%	0.0006	74.80	0.3139	0.0001	0.02%	0.00064	176.97
		$(X_0, \mu, \sigma, \delta, t) = (0.5, -0.8, 1.1, 1.3, 0.5)$									
				Case I					*Case II*		
15,625	0.1837	0.1804	-0.0033	-1.81%	0.0037	0.64	0.1824	-0.0013	-0.69%	0.0037	1.76
62,500	0.1837	0.1843	0.0006	0.33%	0.0019	2.52	0.1827	-0.0009	-0.51%	0.0019	7.13
250,000	0.1837	0.1835	-0.0002	-0.09%	0.0009	16.09	0.1835	-0.0002	-0.09%	0.0009	29.66
1,000,000	0.1837	0.1837	0.0000	0.01%	0.0005	41.93	0.1837	0.0000	0.03%	0.0005	137.53
		$(X_0, \mu, \sigma, \delta, t) = (-0.5, 0.8, 1.1, 1.3, 1)$									
				Case I					*Case II*		
15,625	0.5193	0.5223	0.0030	0.58%	0.0069	0.77	0.5224	0.0031	0.59%	0.0069	3.24
62,500	0.5193	0.5177	-0.0016	-0.31%	0.0034	3.13	0.5214	0.0022	0.42%	0.0034	12.64
250,000	0.5193	0.5206	0.0013	0.25%	0.0017	13.14	0.5181	-0.0011	-0.22%	0.0017	70.03
1,000,000	0.5193	0.5189	-0.0004	-0.08%	0.0009	48.03	0.5194	0.0001	0.02%	0.0009	183.63
		$(X_0, \mu, \sigma, \delta, t) = (-0.5, -0.8, 1.1, 1.3, 1)$									
				Case I					*Case II*		
15,625	0.8430	0.8378	-0.0052	-0.62%	0.0096	0.85	0.8381	-0.0048	-0.57%	0.0097	3.00
62,500	0.8430	0.8510	0.0081	0.96%	0.0049	3.41	0.8453	0.0023	0.27%	0.0049	10.34
250,000	0.8430	0.8434	0.0005	0.06%	0.0024	13.52	0.8436	0.0006	0.07%	0.0024	48.09
1,000,000	0.8430	0.8430	0.0000	0.00%	0.0012	56.08	0.8433	0.0004	0.04%	0.0012	191.48

jection scheme under the parameter setting $(X_0, \mu, \delta, \sigma) = (0.5, 0, 1.1, 1.3)$[16] with $t = 0.5$ and 1, respectively. The numerical results in details are reported in Table 4, and the convergence plots are presented in Figure 10. We can see that both schemes achieve high levels of accuracy but our Algorithm 4.6 is much faster. In particular, our scheme is more efficient when simulating sample paths for a longer horizon. This is due to the fact that, the projection scheme involves massive recursive numerical calculations for the projected intensity, and these recursions would accumulate a huge number of analytic terms in the projected intensity dynamics, which would be both very time-consuming and memory-consuming, especially when horizon t or number N_t becomes larger. Therefore, the projection scheme is practically hard to be maintained *truly exact* for simulating the scenarios of many jumps occurring within a given horizon. That is why they alternatively introduced an *approximation* algorithm for speeding up. Nevertheless, such an approximate scheme inevitably generates biases as pointed out by Giesecke et al. (2011a, §4.2.3). For $\mu \neq 0$, the outperformance of our scheme would become even more substantial, since the recursive calculation of the projected intensity further involves numerical integration to calculate the coefficient functions,

[16] The projection scheme proposed by Giesecke et al. (2011a) involves mathematically tedious calculations and is very time-consuming once the intensity process becomes more complicated, as the authors mentioned themselves. Therefore, we directly adopt their original MatLab codes for the numerical examples of projection scheme under the same parameter setting to be consistently comparable to our exact scheme. The parameter setting above is equivalent to $(X_0, K_0, K_1, H_0, H_1) = (0.5, 1.69, -2.2, 0, 6.76)$ in Equation (39) of Giesecke et al. (2011a), a special case of quadratic OU process by setting $\mu = 0$.

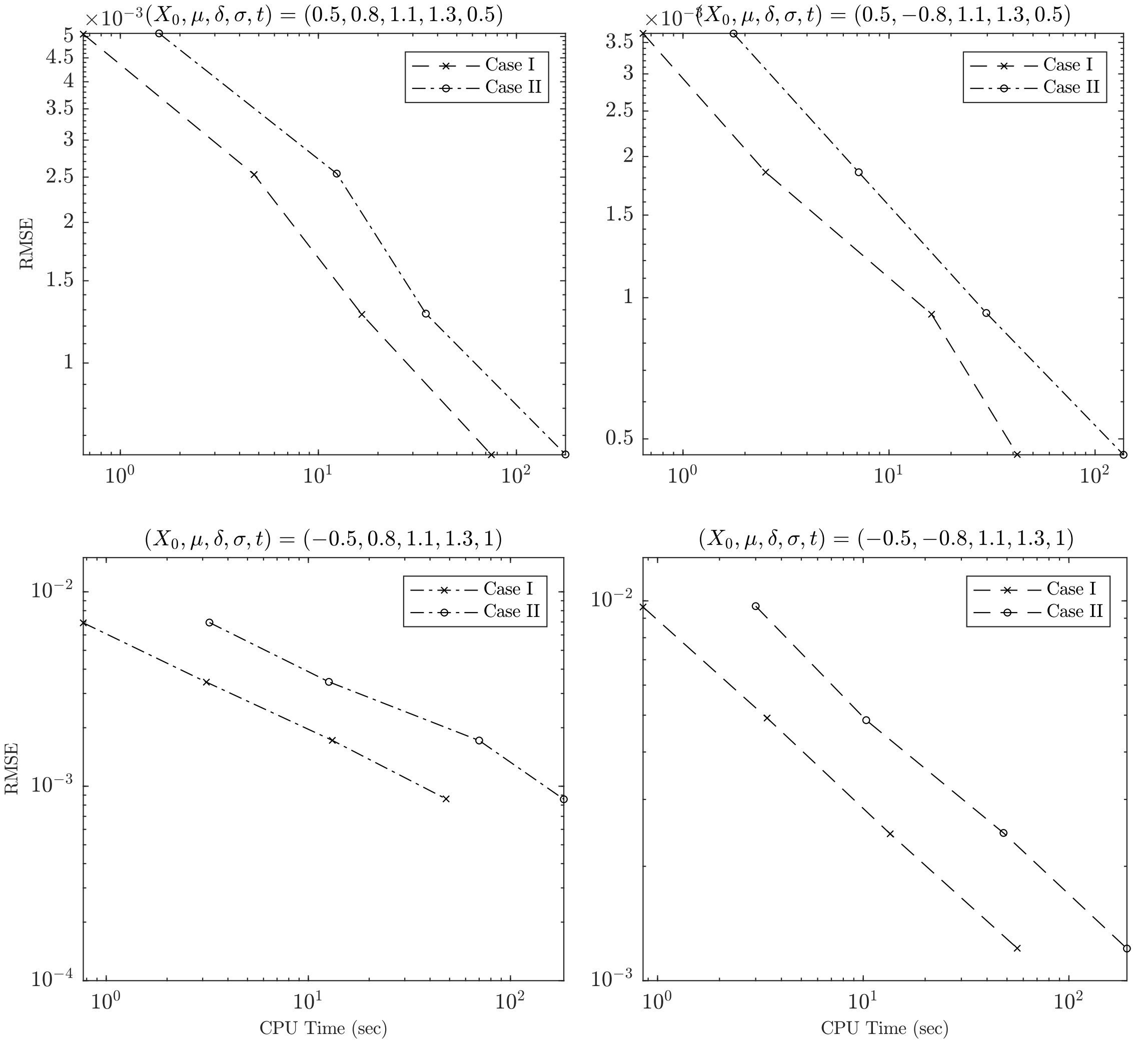

Figure 8: Convergence analysis via log-log plots of RMSE v.s. CPU time for Case I and Case II under parameter settings $(X_0, \mu, \delta, \sigma, t) = (0.5, 0.8, 1.1, 1.3, 0.5)$, $(0.5, -0.8, 1.1, 1.3, 0.5)$, $(-0.5, 0.8, 1.1, 1.3, 1)$ and $(-0.5, -0.8, 1.1, 1.3, 1)$, respectively.

which requires more computing time.

However, our scheme is designed only for exactly simulating this specifical class of quadratic-intensity point processes. The key advantage of projection scheme (Giesecke et al., 2011a) as well as discretisation scheme over ours is their generality with broader applicability beyond the quadratic-intensity point processes.

6.3 Portfolio Loss Modelling

Efficient algorithms for simulating portfolio loss processes are extensively discussed in the literature, see e.g. Glasserman and Li (2005) and Giesecke and Kim (2011a,b). In this section, we offer some numerical examples on risk assessments for large portfolios.

Table 3: Comparisons between our *exact scheme* (Algorithm 4.6) for Case I, and *discretisation scheme* under parameter setting $(X_0, \mu, \delta, \sigma) = (0.5, 0.8, 1.1, 1.3)$ for $t = 0.5, 1, 2$ and 3, respectively.

	Paths	Grid	True	Simulation	Bias	SE	RMSE	CPU Time
Discretisation	15,625	125	0.3138	0.3082	-0.0056	0.0050	0.0075	1.67
$t = 0.5$	62,500	250	0.3138	0.3128	-0.0011	0.0025	0.0027	5.91
	250,000	500	0.3138	0.3132	-0.0007	0.0013	0.0015	24.45
	1,000,000	1000	0.3138	0.3137	-0.0002	0.0006	0.0007	116.52
Exact	15,625		0.3138	0.3119	0	0.0050	0.0050	0.73
$t = 0.5$	62,500		0.3138	0.3183	0	0.0026	0.0026	3.86
	250,000		0.3138	0.3138	0	0.0013	0.0013	13.84
	1,000,000		0.3138	0.3138	0	0.0006	0.0006	61.81
Discretisation	15,625	125	0.8430	0.8337	-0.0092	0.0097	0.0134	1.91
$t = 1$	62,500	250	0.8430	0.8392	-0.0038	0.0049	0.0062	6.31
	250,000	500	0.8430	0.8412	-0.0019	0.0024	0.0031	26.89
	1,000,000	1000	0.8430	0.8411	-0.0018	0.0012	0.0022	121.47
Exact	15,625		0.8430	0.8392	0	0.0096	0.0096	3.53
$t = 1$	62,500		0.8430	0.8415	0	0.0049	0.0049	5.56
	250,000		0.8430	0.8428	0	0.0024	0.0024	20.17
	1,000,000		0.8430	0.8431	0	0.0012	0.0012	82.25
Discretisation	15,625	125	2.1239	2.1110	-0.0128	0.0184	0.0224	1.94
$t = 2$	62,500	250	2.1239	2.1139	-0.0100	0.0092	0.0136	6.25
	250,000	500	2.1239	2.1185	-0.0054	0.0046	0.0071	26.58
	1,000,000	1000	2.1239	2.1185	-0.0054	0.0023	0.0059	116.61
Exact	15,625		2.1239	2.1233	0	0.0187	0.0187	4.97
$t = 2$	62,500		2.1239	2.1241	0	0.0093	0.0093	9.63
	250,000		2.1239	2.1301	0	0.0047	0.0047	37.50
	1,000,000		2.1239	2.1238	0	0.0023	0.0023	153.81
Discretisation	15,625	125	3.4964	3.4535	-0.0429	0.0254	0.0499	1.95
$t = 3$	62,500	250	3.4964	3.4570	-0.0395	0.0128	0.0415	5.59
	250,000	500	3.4964	3.4778	-0.0186	0.0065	0.0197	25.67
	1,000,000	1000	3.4964	3.4805	-0.0159	0.0032	0.0162	122.94
Exact	15,625		3.4964	3.4944	0	0.0261	0.0261	4.89
$t = 3$	62,500		3.4964	3.5051	0	0.0131	0.0131	13.97
	250,000		3.4964	3.5025	0	0.0066	0.0066	55.08
	1,000,000		3.4964	3.4968	0	0.0033	0.0033	220.44

Aggregated Loss Distribution We have a portfolio of investments, and the arrival of losses is modelled by a self-exciting and self-correcting point process $N_t \equiv \{T_i\}_{i=1,2,\cdots}$ with intensity λ_t in Definition 5.1 with $N_0 = 0$. The aggregated loss process of this large portfolio by time t is

$$L_t = \sum_{i \geq 1} L_i \mathbb{1}_{\{T_i \leq t\}},$$

where L_i is the individual loss that occurs at arrival time $T_i, i = 1, 2, \cdots$.

Our exact scheme in Algorithm P.1 for point processes with quadratic intensities offers a greater flexibility in risk assessments of large portfolio investments, especially in the choices for each loss size L_i and jump size Y_i. For instance, simulated sample paths of state process X_t and the associated intensity λ_t with jump sizes $\{Y_i\}_{i=1,2,\cdots}$ following a standard normal distribution $\mathsf{N}(0, 1)$ are presented in Figure 11. In Figure 12, we illustrate a simulated path of a self-exciting and self-correcting point process with quadratic intensity and $N_0 = 0$ under the parameter setting

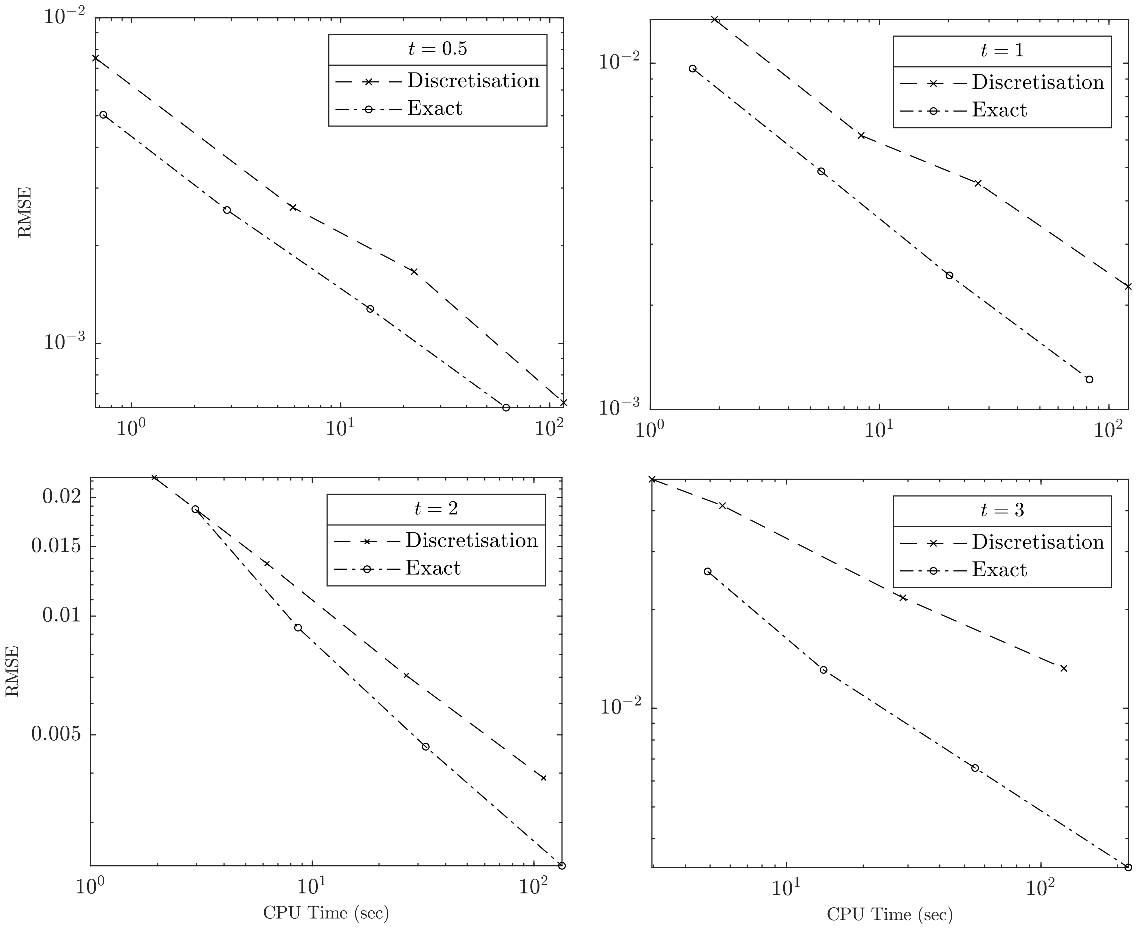

Figure 9: Comparison of convergence via log-log plots of RMSE v.s. CPU time between our *exact scheme* (Algorithm 4.6) under Case I, and *discretisation scheme* under parameter setting $(X_0, \mu, \delta, \sigma) = (0.25, 0.8, 1.1, 1.3)$ for $t = 0.5, 1, 2$ and 3, respectively.

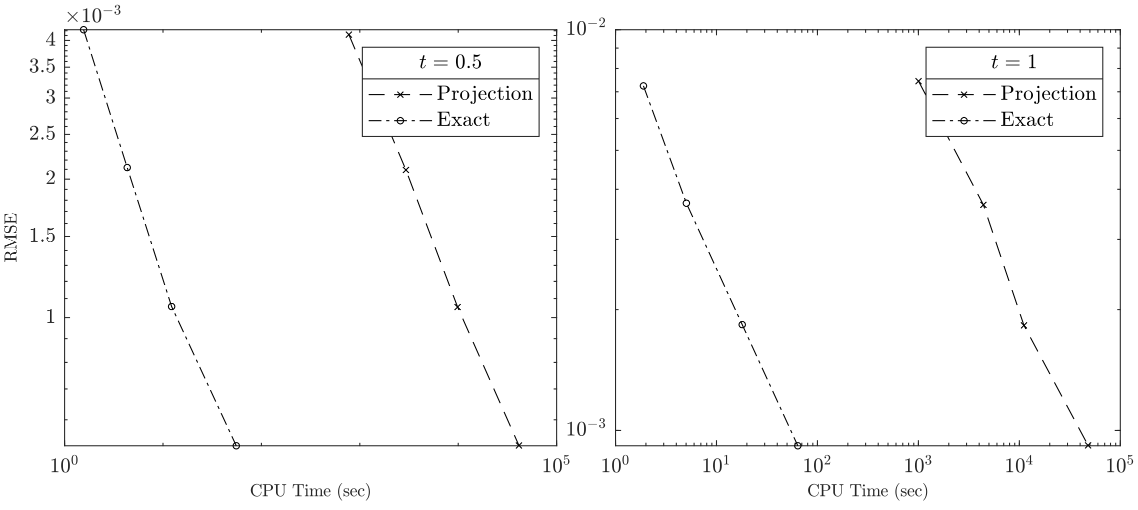

Figure 10: Comparison of convergence via log-log plots of RMSE v.s. CPU time between our *exact scheme* (Algorithm 4.6) under Case I, and *projection scheme* (Giesecke et al., 2011a) under parameter setting $(X_0, \mu, \delta, \sigma) = (0.5, 0, 1.1, 1.3)$ for $t = 0.5$ and 1, respectively.

$(X_0, \mu, \sigma, \delta, t) = (0.5, 0.8, 1.1, 1.3, 500)$ and $Y_i \sim \mathsf{N}(0, 1)$ for jump sizes. Comparing with Figure 7 for the canonical version, more large spikes are now observed in histograms mainly due to the

Table 4: Comparisons between our *exact scheme* (Algorithm 4.6) under Case I, and *projection scheme* (Giesecke et al., 2011a) under parameter setting $(X_0, \mu, \delta, \sigma) = (0.5, 0, 1.1, 1.3)$ for $t = 0.5$ and 1, respectively.

Paths	True	Simulation	Error	Error%	RMSE	CPU time	Simulation	Error	Error%	RMSE	CPU time
$t = 0.5$		*Projection*					*Exact*				
15,625	0.2270	0.2193	-0.0077	-3.39%	0.0041	774.33	0.2248	-0.0021	-0.94%	0.0042	1.56
62,500	0.2270	0.2277	0.0007	0.32%	0.0021	2942.45	0.2275	0.0005	0.22%	0.0021	4.33
250,000	0.2270	0.2261	-0.0009	-0.38%	0.0011	9857.22	0.2272	0.0002	0.10%	0.0011	12.23
1,000,000	0.2270	0.2269	-0.0001	-0.03%	0.0005	41400.33	0.2270	0.0000	0.01%	0.0005	55.53
$t = 1$		*Projection*					*Exact*				
15,625	0.5587	0.5671	0.0084	1.50%	0.0074	1003.37	0.5565	-0.0023	-0.41%	0.0072	1.89
62,500	0.5587	0.5533	-0.0054	-0.97%	0.0037	4404.86	0.5569	-0.0019	-0.34%	0.0037	5.03
250,000	0.5587	0.5555	-0.0032	-0.58%	0.0018	11107.05	0.5584	-0.0003	-0.06%	0.0018	17.98
1,000,000	0.5587	0.5595	0.0008	0.14%	0.0009	47960.32	0.5585	-0.0002	-0.04%	0.0009	63.90

additionally-introduced self-exciting and self-correcting effects.

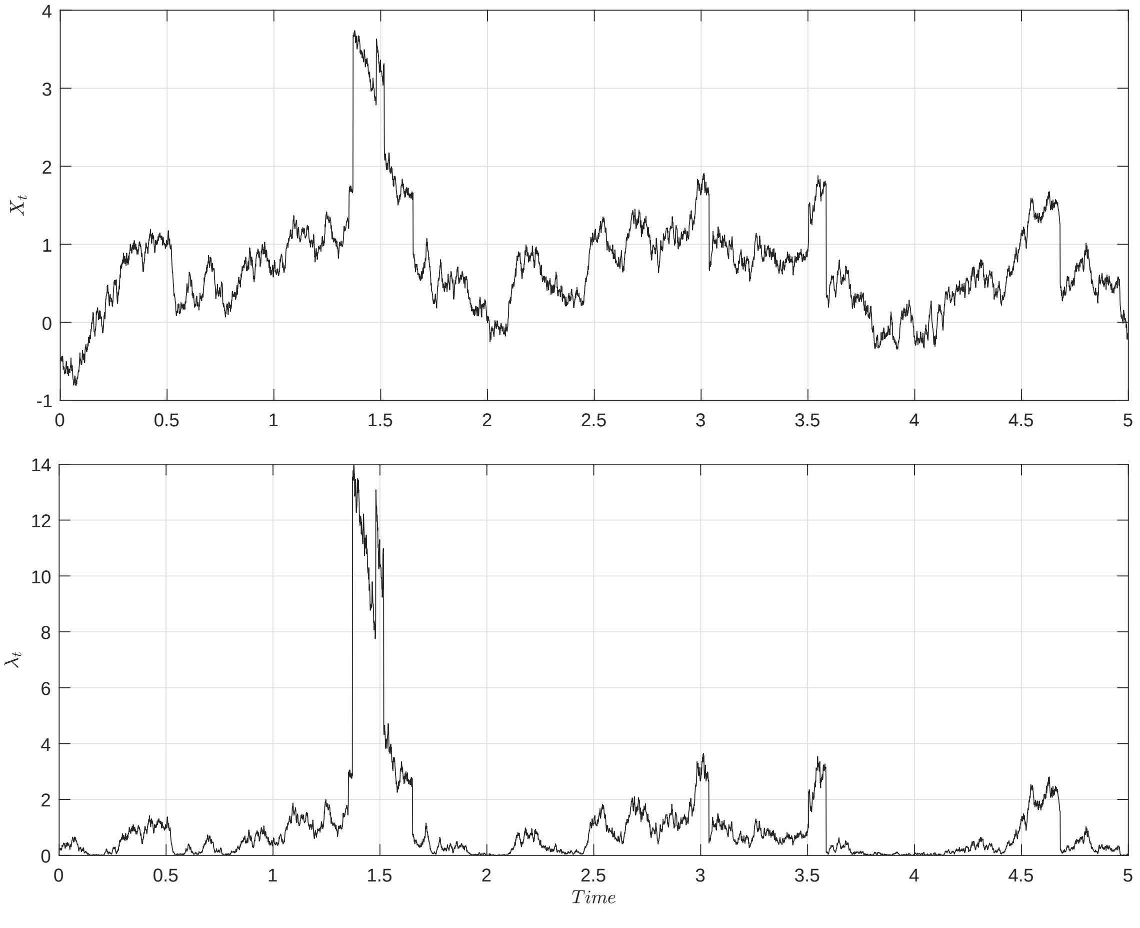

Figure 11: Simulated paths of state process X_t with self-exciting and self-correcting jumps and quadratic intensity λ_t with parameter setting $(X_0, \mu, \delta, \sigma) = (-0.5, 0.8, 1.1, 1.3)$ and $Y_i \sim \mathsf{N}(0, 1)$.

Various quantities about the aggregated loss process can be estimated by simulations. We can also obtain the Monte Carlo estimate of the CDF for the aggregated loss process at time t, i.e., $\mathbb{P}(L_t \leq l)$. Figure 13 illustrates how the loss distribution $\mathbb{P}(L_t \leq l)$ varies with different settings for L_i, Y_i and t under the parameter setting $(X_0, \mu, \delta, \sigma) = (0.5, 0.8, 1.1, 1.3)$ with 10^5 replications

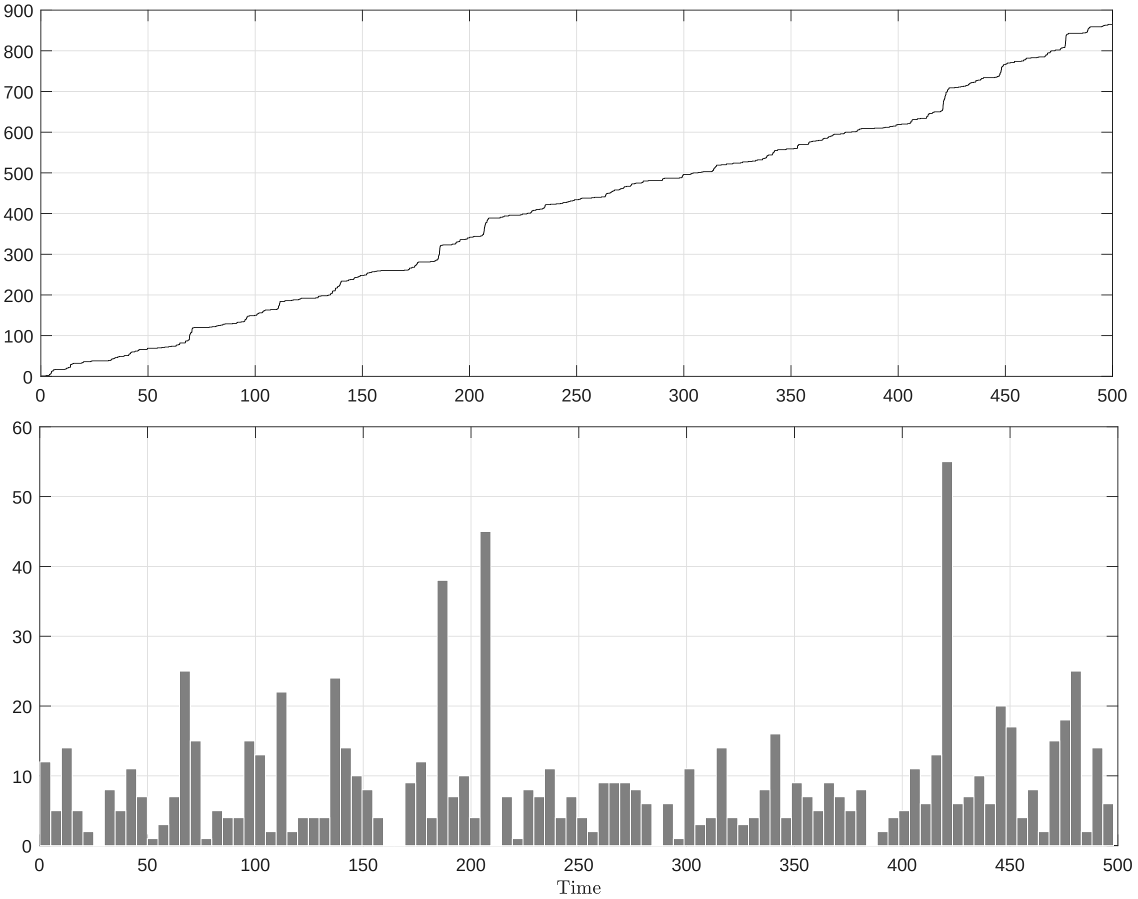

Figure 12: Simulated paths and histograms of self-exciting and self-correcting point process N_t under parameter setting $(X_0, \mu, \delta, \sigma, t) = (0.5, 0.8, 1.1, 1.3, 500)$ and $Y_i \sim \mathsf{N}(0, 1)$.

for $Y_i = 0$ and $Y_i \sim \mathsf{N}(0, 1)$, respectively. The individual loss L_i is assumed to be fixed, uniformly distributed on $[0, 1]$ and exponentially distributed, respectively, all with expected loss 0.5. We can see that, the tail in the CDF of aggregated loss process becomes much heavier if self-exciting jumps are incorporated into the loss process.

Probability Mass Function The probability mass functions (PMF) have many important applications, e.g. credit portfolio analysis and pricing, see Choi et al. (2020) for more details. If there is no any jump in the intensity process, i.e., $Y_i = 0$ for any i, the *empirical* PMF of aggregated number of losses at time t, $\mathbb{P}(N_t = n), n = 0, 1, \cdots$, estimated by 10^7 replications against the associated *true* values[17] under parameter settings, Case 1,2,3,4: $(X_0, \mu, \delta, \sigma, t) = (0.5, 0.8, 1.1, 1.3, 1)$, $(-0.5, 0.8, 1.1, 1.3, 1)$, $(0.5, 0.8, 1.1, 1.3, 0.5)$ and $(-0.5, 0.8, 1.1, 1.3, 0.5)$ are plotted in Figure

[17] For these *special cases* without jump in the intensity, since they reduce to Cox processes, the true values of PMFs in theory can be calculated by

$$\mathbb{P}(N_t = n) \;=\; \mathbb{E}\left[\frac{\Lambda_t^n}{n!} e^{-\Lambda_t}\right] \;=\; \frac{1}{n!}\mathbb{E}\left[\Lambda_t^n e^{-\Lambda_t}\right] \;=\; \frac{(-1)^n}{n!}\frac{\partial^n}{\partial w^n}\mathbb{E}\left[e^{-w\Lambda_t}\right]\Bigg|_{w=1}, \qquad n = 0, 1, \cdots,$$

where $\mathbb{E}\left[e^{-w\Lambda_t}\right]$ is the Laplace transform of cumulative intensity Λ_t and can be calculated by setting $u = 0$ and replacing (t, s) with $(0, t)$ in Theorem 3.1.

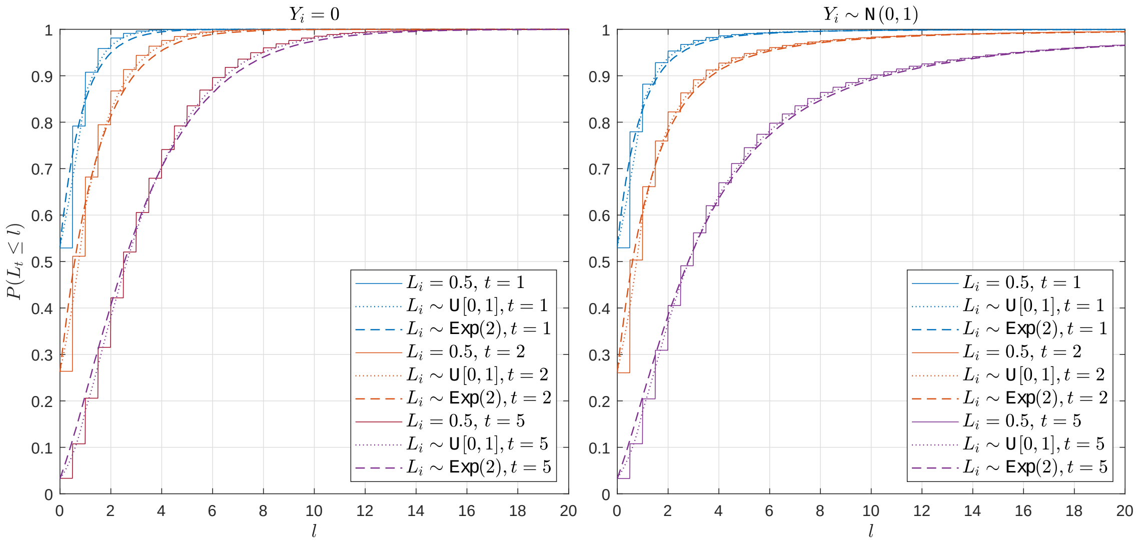

Figure 13: The estimated CDFs of aggregated portfolio loss process at time t, $\mathbb{P}(L_t \leq l)$, under the parameter setting $(X_0, \mu, \delta, \sigma) = (0.5, 0.8, 1.1, 1.3)$ with 10^5 replications for $Y_i = 0$ and $Y_i \sim \mathsf{N}(0,1)$, respectively.

14, respectively. The associated numerical results and computational cost measured by the CPU time (in seconds) for generating 10^7 replications are reported in Table 5. We can see that the estimated PMFs using our exact scheme are well fitted to the true values. Of course, our algorithms can be applied to exactly simulating more general point processes whose PMFs are not available in analytical forms. For example, the estimated PMFs by our exact simulation for additional jumps $Y_i \sim \mathsf{N}(0,1)$ and $Y_i \sim \mathsf{N}(0,2)$ in intensity processes are also reported in Table 5 and plotted in Figure 14, where we observe that these additional jumps make the tails of distributions heavier than the Cox counterparts ($Y_i = 0$) and could be more suited to loss modelling with contagion risk.

7 Concluding Remarks

Quadratic stochastic intensity models have become increasingly popular, especially in economics, finance and insurance, for modelling events arrivals. In this paper, we obtain a *fully analytical* expression for the crucial theoretical result (Theorem 3.1), i.e., the joint Laplace transform of quadratic process and its integral, which shows that the quadratic intensity model and the currently more popular CIR counterpart actually have a similar level of analytical tractability. The main contribution of this paper is that we develop new algorithms for exact simulation based on the principle of exact distributional decomposition. The algorithms are accurate and efficient, and have been numerically verified and tested by extensive numerical experiments. We also find that they substantially outperform all existing alternatives in the literature. Moreover, our exact scheme has the generality to be extended for more general point processes (beyond Cox processes) where

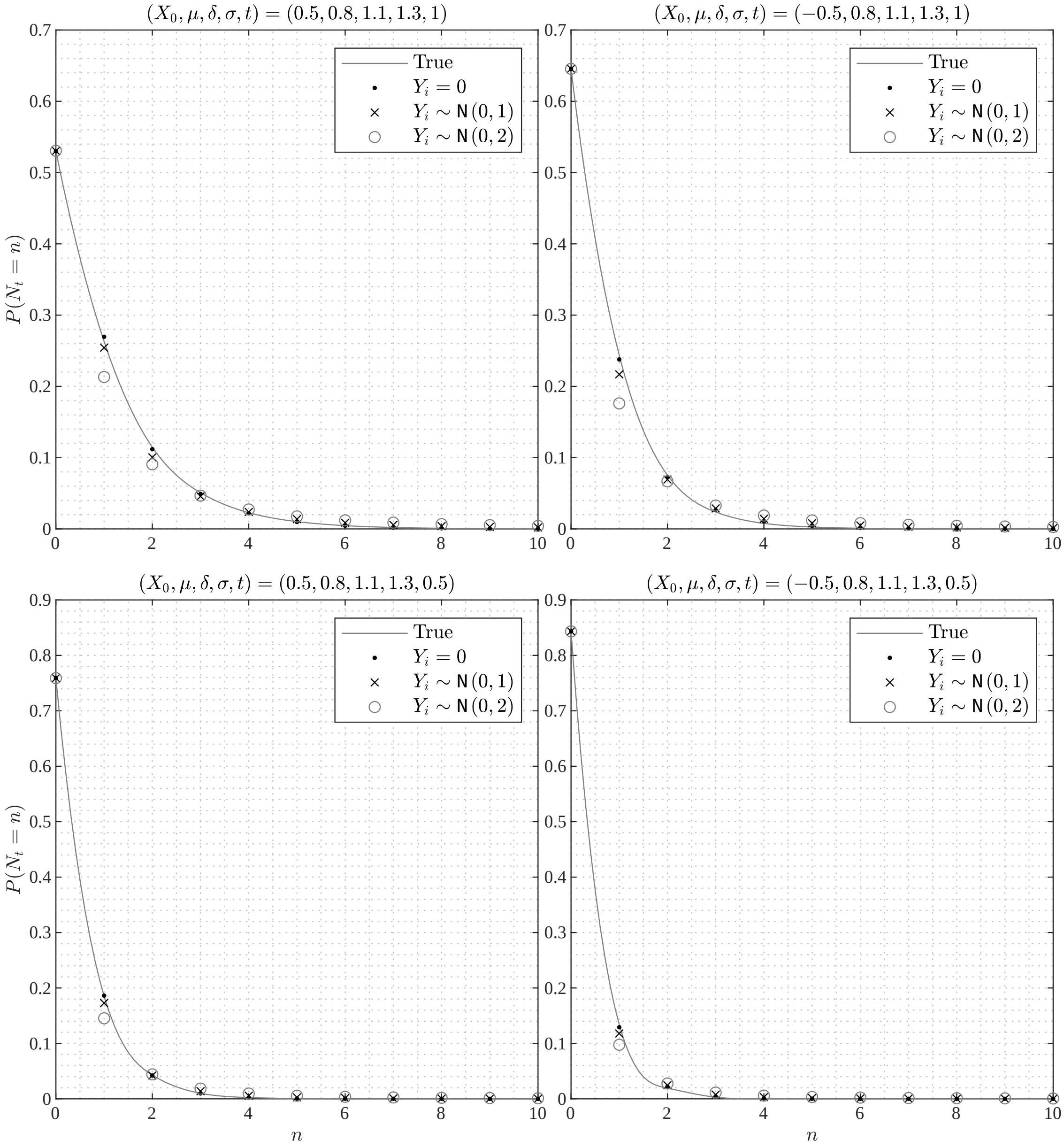

Figure 14: The *estimated* PMFs of aggregated number of losses at time t, $\mathbb{P}(N_t = n), n = 0, 1, \cdots$, via exact simulation with 10^7 replications for $Y_i = 0, Y_i \sim \mathsf{N}(0, 1), Y_i \sim \mathsf{N}(0, 2)$ against the *true* values for $Y_i = 0$ under four parameter settings, Case 1,2,3,4: $(X_0, \mu, \delta, \sigma, t) = (0.5, 0.8, 1.1, 1.3, 1)$, $(-0.5, 0.8, 1.1, 1.3, 1)$, $(0.5, 0.8, 1.1, 1.3, 0.5)$ and $(-0.5, 0.8, 1.1, 1.3, 0.5)$, respectively, with the associated numerical results reported in Table 5.

their PMFs are not available in analytical forms and traditional simulation approaches such as numerical Laplace transform are not applicable. For example, additional two-sided random jumps with arbitrarily distributed sizes in the intensity can be incorporated for capturing self-exciting and self-correcting effects in event arrivals, which form a new class of quadratic (stochastic) intensity models. We also generalise by introducing a new family of multidimensional point processes with cojumps in intensity processes, and offer the associated algorithms for exact simulation, which facilitate more practical applications such as portfolio credit risk modelling at a more granular level. The cojump sizes could be arbitrary, and simultaneously allow for negative correlation and posi-

Table 5: The *empirical* PMFs of aggregated number of losses at time t, $\mathbb{P}(N_t = n), n = 0, 1, \cdots$, estimated by 10^7 replications for $Y_i = 0, Y_i \sim \mathsf{N}(0, 1), Y_i \sim \mathsf{N}(0, 2)$ against the *true* values for $Y_i = 0$ under four parameter settings, Case 1,2,3,4: $(X_0, \mu, \delta, \sigma, t) = (0.5, 0.8, 1.1, 1.3, 1)$, $(-0.5, 0.8, 1.1, 1.3, 1)$, $(0.5, 0.8, 1.1, 1.3, 0.5)$ and $(-0.5, 0.8, 1.1, 1.3, 0.5)$, respectively.

$\mathbb{P}(N_t = n)$	*Case 1*				*Case 2*			
n	$Y_i = 0$ (True)	$Y_i = 0$	$Y_i \sim \mathsf{N}(0, 1)$	$Y_i \sim \mathsf{N}(0, 2)$	$Y_i = 0$ (True)	$Y_i = 0$	$Y_i \sim \mathsf{N}(0, 1)$	$Y_i \sim \mathsf{N}(0, 2)$
0	0.5304	0.5300	0.5303	0.5305	0.6455	0.6453	0.6456	0.6456
1	0.2629	0.2696	0.2543	0.2131	0.2435	0.2378	0.2169	0.1761
2	0.1146	0.1118	0.1004	0.0906	0.0753	0.0727	0.0693	0.0667
3	0.0508	0.0491	0.0457	0.0467	0.0237	0.0265	0.0288	0.0328
4	0.0228	0.0218	0.0238	0.0273	0.0078	0.0105	0.0142	0.0187
5	0.0102	0.0098	0.0137	0.0175	0.0027	0.0043	0.0080	0.0118
6	0.0046	0.0044	0.0086	0.0121	0.0009	0.0017	0.0048	0.0081
7	0.0020	0.0020	0.0057	0.0088	0.0003	0.0007	0.0031	0.0058
8	0.0009	0.0009	0.0039	0.0067	0.0001	0.0003	0.0021	0.0044
9	0.0004	0.0004	0.0028	0.0053	0.0000	0.0001	0.0015	0.0034
10	0.0002	0.0002	0.0021	0.0042	0.0000	0.0000	0.0011	0.0028
CPU Time		2,108	3,436	8,569		8,545	10,871	15,891
$\mathbb{P}(N_t = n)$	*Case 3*				*Case 4*			
n	$Y_i = 0$ (True)	$Y_i = 0$	$Y_i \sim \mathsf{N}(0, 1)$	$Y_i \sim \mathsf{N}(0, 2)$	$Y_i = 0$ (True)	$Y_i = 0$	$Y_i \sim \mathsf{N}(0, 1)$	$Y_i \sim \mathsf{N}(0, 2)$
0	0.7589	0.7589	0.7589	0.7589	0.8433	0.8433	0.8435	0.8433
1	0.1853	0.1862	0.1728	0.1453	0.1339	0.1291	0.1179	0.0975
2	0.0428	0.0421	0.0420	0.0442	0.0192	0.0220	0.0245	0.0277
3	0.0100	0.0098	0.0139	0.0185	0.0030	0.0044	0.0076	0.0114
4	0.0023	0.0023	0.0057	0.0096	0.0005	0.0009	0.0031	0.0058
5	0.0005	0.0005	0.0027	0.0057	0.0001	0.0002	0.0014	0.0034
6	0.0001	0.0001	0.0014	0.0037	0.0000	0.0000	0.0008	0.0022
7	0.0000	0.0000	0.0008	0.0025	0.0000	0.0000	0.0004	0.0015
8	0.0000	0.0000	0.0005	0.0018	0.0000	0.0000	0.0003	0.0011
9	0.0000	0.0000	0.0003	0.0014	0.0000	0.0000	0.0002	0.0008
10	0.0000	0.0000	0.0002	0.0011	0.0000	0.0000	0.0001	0.0006
CPU Time		1,039	1,505	3,633		6,755	7,341	9,374

tivity among state variables, which cannot be captured by classical affine models. Finally, we use portfolio risk management as an example to show the applicability and flexibility of our algorithms. Of course, the models and the associated simulation algorithms can be used for other applications far beyond. For instance, it could easily be extended to pricing financial derivatives, particularly path-dependent or multiple-name structured credit products similarly as Choi et al. (2020) where analytical pricing formulas are unavailable and efficient simulations are more desirable. In the paper, we only adopt financial applications of credit risk modelling to facilitate our illustration. However, the developed methods and algorithms would undoubtedly have a much wider application spectrum. These models have huge potential to be further applied to many other fields of both natural science and social science, for modelling events arrivals, such as job losses, trade orders, unscheduled announcements, news reports, information flows, online posts, website visits, particulate emissions, traffic flows, queues, customer arrivals, crimes, reliability and COVID-19 infections. For example, in the areas of queuing theory and operations management, they may be useful for modelling call arrivals in telephone call centers with stochastically time-varying arrival rates as motivated by Whitt (1999), Avramidis et al. (2004) and Ata and Peng (2020). The models with associated simulation algorithms might be extendable to multi-factor counterparts and even beyond the quadratic processes, such as polynomial processes (Cuchiero et al., 2012; Filipović and Larsson, 2016, 2020), which could be interesting and meaningful topics for future research.

Acknowledgments

The authors would like to thank the editors and referees for extremely helpful and constructive comments and suggestions, which have significantly improved our paper. They are particularly grateful to Prof. Mohammad Mousavi at the University of Pittsburgh, for sharing his detailed insights on numerically implementing the projection schemes with MatLab codes. The author Yan Qu would like to acknowledge the financial support of the research fund (#2022RC58) provided by the Beijing University of Posts and Telecommunications. The corresponding author Hongbiao Zhao would like to acknowledge the financial support from the *National Natural Science Foundation of China* (#71401147) and the research fund provided by the Innovative Research Team of *Shanghai University of Finance and Economics* (#2020110930) and *Shanghai Institute of International Finance and Economics*.

References

Aalen, O. O. and Gjessing, H. K. (2004). Survival models based on the Ornstein-Uhlenbeck process. *Lifetime Data Analysis*, 10(4):407–423.

Abate, J., Choudhury, G. L., and Whitt, W. (2000). An introduction to numerical transform inversion and its application to probability models. In Grassmann, W. K., editor, *Computational Probability*, pages 257–323. Springer, Dordrecht.

Abate, J. and Whitt, W. (1992). The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems*, 10(1):5–87.

Abate, J. and Whitt, W. (1995). Numerical inversion of Laplace transforms of probability distributions. *ORSA Journal on Computing*, 7(1):36–43.

Abate, J. and Whitt, W. (2006). A unified framework for numerically inverting Laplace transforms. *INFORMS Journal on Computing*, 18(4):408–421.

Ahdida, A. and Alfonsi, A. (2013). Exact and high-order discretization schemes for Wishart processes and their affine extensions. *Annals of Applied Probability*, 23(3):1025–1073.

Ahn, D.-H., Dittmar, R. F., and Gallant, A. R. (2002). Quadratic term structure models: Theory and evidence. *Review of Financial Studies*, 15(1):243–288.

Andreasen, M. and Meldrum, A. (2013). Likelihood inference in non-linear term structure models: The importance of the lower bound. Working Paper No. 481, Bank of England.

Ang, A., Boivin, J., Dong, S., and Loo-Kung, R. (2011). Monetary policy shifts and the term structure. *Review of Economic Studies*, 78(2):429–457.

Asmussen, S. and Glynn, P. W. (2007). *Stochastic Simulation: Algorithms and Analysis*. Springer, New York.

Ata, B. and Peng, X. (2020). An optimal callback policy for general arrival processes: A pathwise analysis. *Operations Research*, 68(2):327–347.

Avramidis, A., Deslauriers, A., and L'Ecuyer, P. (2004). Modeling daily arrivals to a telephone call center. *Management Science*, 50(7):896–908.

Azizpour, S., Giesecke, K., and Schwenkler, G. (2018). Exploring the sources of default clustering. *Journal of Financial Economics*, 129(1):154–183.

Benzoni, L., Collin-Dufresne, P., Goldstein, R. S., and Helwege, J. (2015). Modeling credit contagion via the updating of fragile beliefs. *Review of Financial Studies*, 28(7):1960–2008.

Beskos, A. and Roberts, G. O. (2005). Exact simulation of diffusions. *Annals of Applied Probability*, 15(4):2422–2444.

Brent, R. P. (1973). *Algorithms for Minimization without Derivatives*. Englewood Cliffs: Prentice Hall, New Jersey.

Broadie, M. and Kaya, Ö. (2006). Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations Research*, 54(2):217–231.

Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., and Zhao, L. (2005). Statistical analysis of a telephone call center: A queueing-science perspective. *Journal of the American Statistical Association*, 100(469):36–50.

Cai, N., Song, Y., and Chen, N. (2017). Exact simulation of the SABR model. *Operations Research*, 65(4):931–951.

Campbell, J. Y., Sunderam, A., and Viceira, L. M. (2017). Inflation bets or deflation hedges? The changing risks of nominal bonds. *Critical Finance Review*, 6(2):263–301.

Cezik, M. T. and L'Ecuyer, P. (2008). Staffing multiskill call centers via linear programming and simulation. *Management Science*, 54(2):310–323.

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

Chen, N. and Huang, Z. (2013). Localization and exact simulation of Brownian motion-driven stochastic differential equations. *Mathematics of Operations Research*, 38(3):591–616.

Choi, Y. S., Doshi, H., Jacobs, K., and Turnbull, S. M. (2020). Pricing structured products with economic covariates. *Journal of Financial Economics*, 135(3):754–773.

Cox, D. R. (1955). Some statistical methods connected with series of events. *Journal of the Royal Statistical Society. Series B (Methodological)*, 17(2):129–164.

Cuchiero, C., Keller-Ressel, M., and Teichmann, J. (2012). Polynomial processes and their applications to mathematical finance. *Finance and Stochastics*, 16:711–740.

Cuchiero, C., Teichmann, J., and Filipović, D. (2010). Affine models. In *Encyclopedia of Quantitative Finance*. John Wiley & Sons, New York.

Dai, Q. and Singleton, K. J. (2000). Specification analysis of affine term structure models. *Journal of Finance*, 55(5):1943–1978.

Das, S. R., Duffie, D., Kapadia, N., and Saita, L. (2007). Common failings: How corporate defaults are correlated. *Journal of Finance*, 62(1):93–117.

Dassios, A. and Zhao, H. (2011). A dynamic contagion process. *Advances in Applied Probability*, 43(3):814–846.

Dassios, A. and Zhao, H. (2013). Exact simulation of Hawkes process with exponentially decaying intensity. *Electronic Communications in Probability*, 18(62):1–13.

Dassios, A. and Zhao, H. (2017). Efficient simulation of clustering jumps with CIR intensity. *Operations Research*, 65(6):1494–1515.

Devroye, L. (1981). On the computer generation of random variables with a given characteristic function. *Computers & Mathematics with Applications*, 7(6):547–552.

Doshi, H., Elkamhi, R., and Ornthanalai, C. (2018). The term structure of expected recovery rates. *Journal of Financial and Quantitative Analysis*, 53(6):2619–2661.

Doshi, H., Ericsson, J., Jacobs, K., and Turnbull, S. M. (2013). Pricing credit default swaps with observable covariates. *Review of Financial Studies*, 26(8):2049–2094.

Doshi, H., Jacobs, K., and Zurita, V. (2017). Economic and financial determinants of credit risk premiums in the sovereign CDS market. *Review of Asset Pricing Studies*, 7(1):43–80.

Duffie, D., Eckner, A., Horel, G., and Saita, L. (2009). Frailty correlated default. *Journal of Finance*, 64(5):2089–2123.

Duffie, D., Filipovic, D., and Schachermayer, W. (2003). Affine processes and applications in finance. *Annals of Applied Probability*, 13(3):984–1053.

Duffie, D. and Glynn, P. (1995). Efficient Monte Carlo simulation of security prices. *Annals of Applied Probability*, 5(4):897–905.

Duffie, D. and Kan, R. (1996). A yield-factor model of interest rates. *Mathematical Finance*, 6(4):379–406.

Duffie, D. and Liu, J. (2001). Floating-fixed credit spreads. *Financial Analysts Journal*, 57(3):76–87.

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

Duffie, D., Saita, L., and Wang, K. (2007). Multi-period corporate default prediction with stochastic covariates. *Journal of Financial Economics*, 83(3):635–665.

Duffie, D. and Singleton, K. J. (1999). Modeling term structures of defaultable bonds. *Review of Financial Studies*, 12(4):687–720.

Filipović, D. and Larsson, M. (2016). Polynomial diffusions and applications in finance. *Finance and Stochastics*, 20(4):931–972.

Filipović, D. and Larsson, M. (2020). Polynomial jump-diffusion models. *Stochastic Systems*, 10(1):71–97.

Fu, M. C. (2015). *Handbook of Simulation Optimization*. Springer, New York.

Gans, N., Koole, G., and Mandelbaum, A. (2003). Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & Service Operations Management*, 5(2):79–141.

Gaver, J. D. P. (1966). Observing stochastic processes, and approximate transform inversion. *Operations Research*, 14(3):444–459.

Giesecke, K., Kakavand, H., and Mousavi, M. (2011a). Exact simulation of point processes with stochastic intensities. *Operations Research*, 59(5):1233–1245.

Giesecke, K. and Kim, B. (2011a). Risk analysis of Collateralized Debt Obligations. *Operations Research*, 59(1):32–49.

Giesecke, K. and Kim, B. (2011b). Systemic risk: What defaults are telling us. *Management Science*, 57(8):1387–1405.

Giesecke, K., Kim, B., and Zhu, S. (2011b). Monte Carlo algorithms for default timing problems. *Management Science*, 57(12):2115–2129.

Giesecke, K. and Smelov, D. (2013). Exact sampling of jump diffusions. *Operations Research*, 61(4):894–907.

Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering*. Springer, New York.

Glasserman, P. and Li, J. (2005). Importance sampling for portfolio credit risk. *Management Science*, 51(11):1643–1656.

Glynn, P. W. (2016). Exact simulation vs exact estimation. In Roeder, T., Frazier, P., Szechtman, R., Zhou, E., Huschka, T., and Chick, S., editors, *Proceedings of the 2016 Winter Simulation Conference*, pages 193–205. IEEE.

Gordy, M. B. and Juneja, S. (2010). Nested simulation in portfolio risk measurement. *Management Science*, 56(10):1833–1848.

Gourieroux, C. and Monfort, A. (2008). Quadratic stochastic intensity and prospective mortality tables. *Insurance: Mathematics and Economics*, 43(1):174–184.

Gourieroux, C., Monfort, A., and Polimenis, V. (2006). Affine models for credit risk analysis. *Journal of Financial Econometrics*, 4(3):494–530.

Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90.

Hong, L. J. and Liu, G. (2009). Simulating sensitivities of conditional value at risk. *Management Science*, 55(2):281–293.

Hong, L. J. and Nelson, B. L. (2009). A brief introduction to optimization via simulation. In *Proceedings of the 2009 Winter Simulation Conference*, pages 75–85. IEEE.

Isham, V. and Westcott, M. (1979). A self-correcting point process. *Stochastic Processes and their Applications*, 8(3):335–347.

Jamshidian, F. (1996). Bond, futures and option evaluation in the quadratic interest rate model. *Applied Mathematical Finance*, 3(2):93–115.

Joslin, S., Singleton, K. J., and Zhu, H. (2011). A new perspective on Gaussian dynamic term structure models. *Review of Financial Studies*, 24(3):926–970.

Kang, C., Kang, W., and Lee, J. M. (2017). Exact simulation of the Wishart multidimensional stochastic volatility model. *Operations Research*, 65(5):1190–1206.

Kim, D. H. and Singleton, K. J. (2012). Term structure models and the zero bound: An empirical investigation of Japanese yields. *Journal of Econometrics*, 170(1):32–49.

Kim, S.-H. and Whitt, W. (2014). Are call center and hospital arrivals well modeled by nonhomogeneous Poisson processes? *Manufacturing & Service Operations Management*, 16(3):464–480.

Lamberton, D. and Lapeyre, B. (2008). *Introduction to Stochastic Calculus Applied to Finance*. Chapman & Hall, Boca Raton, 2nd edition.

Lando, D. (1998). On Cox processes and credit risky securities. *Review of Derivatives Research*, 2(2-3):99–120.

Lando, D. and Nielsen, M. S. (2010). Correlation in corporate defaults: Contagion or conditional independence? *Journal of Financial Intermediation*, 19(3):355–372.

Le, A., Singleton, K. J., and Dai, Q. (2010). Discrete-time affine$^{\mathbb{Q}}$ term structure models with generalized market prices of risk. *Review of Financial Studies*, 23(5):2184–2227.

L'Ecuyer, P. (2012). Random number generation. In Gentle, J. E., Härdle, W. K., and Mori, Y., editors, *Handbook of Computational Statistics: Concepts and Methods*, chapter 3, pages 35–71. Springer, Berlin.

Lee, Y., Laub, P. J., Taimre, T., Zhao, H., and Zhuang, J. (2022). Exact simulation of extrinsic stress-release processes. *Journal of Applied Probability*, 59(1):105–117.

Leippold, M. and Wu, L. (2002). Asset pricing under the quadratic class. *Journal of Financial and Quantitative Analysis*, 37(2):271–295.

Leippold, M. and Wu, L. (2003). Design and estimation of quadratic term structure models. *Review of Finance*, 7(1):47–73.

Leippold, M. and Wu, L. (2007). Design and estimation of multi-currency quadratic models. *Review of Finance*, 11(2):167–207.

Li, C. and Wu, L. (2019). Exact simulation of the Ornstein–Uhlenbeck driven stochastic volatility model. *European Journal of Operational Research*, 275(2):768–779.

Li, H. and Zhao, F. (2006). Unspanned stochastic volatility: Evidence from hedging interest rate derivatives. *Journal of Finance*, 61(1):341–378.

Meyer, P.-A. (1971). Démonstration simplifiée d'un théorème de Knight. *Séminaire de Probabilités V Université de Strasbourg*, 191:191–195.

Ogata, Y. and Vere-Jones, D. (1984). Inference for earthquake models: A self-correcting model. *Stochastic Processes and their Applications*, 17(2):337–347.

Piazzesi, M. (2001). An econometric model of the yield curve with macroeconomic jump effects. Working Paper No. 8246 (April), NBER, Cambridge, MA.

Piazzesi, M. (2005). Bond yields and the Federal Reserve. *Journal of Political Economy*, 113(2):311–344.

Qu, Y., Dassios, A., Liu, A., and Zhao, H. (2024). Exact simulation of quadratic intensity models. Available for download at https://github.com/INFORMSJoC/2023.0323.

Ridout, M. S. (2009). Generating random numbers from a distribution specified by its Laplace transform. *Statistics and Computing*, 19(4):439–450.

Santa-Clara, P. and Yan, S. (2010). Crashes, volatility, and the equity premium: Lessons from S&P 500 options. *Review of Economics and Statistics*, 92(2):435–451.

Singleton, K. J. (2009). *Empirical Dynamic Asset Pricing: Model Specification and Econometric Assessment*. Princeton University Press, New Jersey.

Stehfest, H. (1970). Algorithm 368: Numerical inversion of Laplace transforms. *Communications of the ACM*, 13(1):47–49.

Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the Brownian motion. *Physical Review*, 36(5):823–841.

Vasicek, O. A. (1977). An equilibrium characterization of the term structure. *Journal of Financial Economics*, 5(2):177–188.

Vere-Jones, D. and Ogata, Y. (1984). On the moments of a self-correcting process. *Journal of Applied Probability*, 21(2):335–342.

Whitt, W. (1999). Dynamic staffing in a telephone call center aiming to immediately answer all calls. *Operations Research Letters*, 24(5):205–212.

Woodbury, M. A. and Manton, K. G. (1977). A random-walk model of human mortality and aging. *Theoretical Population Biology*, 11(1):37–48.

Appendices

A Joint Laplace Transform of $\left(\epsilon, \epsilon^2\right), \epsilon \sim \mathsf{N}(0,1)$

Proposition A.1. *The bivariate Laplace transform of* $\left(\epsilon, \epsilon^2\right)$ *for a standard normal random variable* $\epsilon \sim N(0,1)$ *is given by*

$$\mathbb{E}\left[e^{-u\epsilon}e^{-w\epsilon^2}\right] = \frac{1}{\sqrt{2w+1}}\exp\left(\frac{u^2}{2(2w+1)}\right), \tag{A.1}$$

where $u \in \mathbb{R}$ *and* $w > -\frac{1}{2}$.

Proof. For a standard normal random variable $\epsilon \sim \mathsf{N}(0,1)$, we have

$$\begin{aligned}
\mathbb{E}\left[e^{-u\epsilon}e^{-w\epsilon^2}\right] &= \int_{-\infty}^{\infty} e^{-us}e^{-ws^2}\frac{1}{\sqrt{2\pi}}e^{-\frac{s^2}{2}}\mathrm{d}s \\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\exp\left(-\frac{2w+1}{2}\left(s^2+\frac{2us}{2w+1}\right)\right)\mathrm{d}s \\
&= \frac{1}{\sqrt{2\pi}}\exp\left(\frac{u^2}{2(2w+1)}\right)\int_{-\infty}^{\infty}\exp\left(-\frac{2w+1}{2}\left(s+\frac{u}{2w+1}\right)^2\right)\mathrm{d}s \\
&= \frac{1}{\sqrt{2w+1}}\exp\left(\frac{u^2}{2(2w+1)}\right).
\end{aligned}$$

□

B Expectations of Intensity Process λ_t and Point Process N_t

The conditional expectations of intensity process and point process can be obtained analytically in simple forms as in Proposition B.1 below, which can be used for numerically validating and testing our algorithms in this paper.

Proposition B.1. *The expectation of intensity level* λ_{t+s} *conditional on* X_t *at time* t *is given by*

$$\mathbb{E}[\lambda_{t+s} \mid X_t] = \left[X_t e^{-\delta s} + \mu\left(1-e^{-\delta s}\right)\right]^2 + \frac{\sigma^2}{2\delta}\left(1-e^{-2\delta s}\right), \tag{B.1}$$

and the expectation of point number N_{t+s} *conditional on* N_t *and* X_t *is given by*

$$\begin{aligned}
\mathbb{E}\left[N_{t+s} \mid N_t, X_t\right] &= N_t + \left(\mu^2 + \frac{\sigma^2}{2\delta}\right)s \\
&\quad + \frac{2\mu}{\delta}\left(X_t-\mu\right)\left(1-e^{-\delta s}\right) + \frac{1}{2\delta}\left[\left(X_t-\mu\right)^2 - \frac{\sigma^2}{2\delta}\right]\left(1-e^{-2\delta s}\right).
\end{aligned} \tag{B.2}$$

Proof. Recalling (2.4), i.e.,

$$X_{t+s} \mid X_t \overset{\mathcal{D}}{=} \nu + \varsigma\epsilon, \qquad \epsilon \sim \mathsf{N}(0,1),$$

we have the expectation of λ_{t+s} conditional on X_t,

$$\mathbb{E}\left[\lambda_{t+s} \mid X_t\right] \;=\; \mathbb{E}\left[X_{t+s}^2 \,\middle|\, X_t\right] \;=\; \nu^2 + \varsigma^2,$$

where ν and ς are specified in (2.5). For the conditional expectation of point process, since

$$\mathbb{E}[N_{t+s} \mid N_t, X_t] \;=\; N_t + \mathbb{E}\left[\int_t^{t+s} \lambda_u \mathrm{d}u \,\middle|\, X_t\right] \;=\; N_t + \int_t^{t+s} \mathbb{E}\left[\lambda_u \mid X_t\right] \mathrm{d}u,$$

then (B.2) follows. □

C Proof for Proposition 3.1

Proof. Based on (2.4), we have the conditional joint bivariate Laplace transform of $\left(X_{t+s}, X_{t+s}^2\right)$,

$$\begin{aligned}\mathbb{E}\left[e^{-uX_{t+s}}e^{-wX_{t+s}^2} \,\middle|\, X_t\right] &= \mathbb{E}\left[e^{-u(\nu+\varsigma\epsilon)}e^{-w(\nu+\varsigma\epsilon)^2}\right] \\ &= e^{-\left(u\nu+w\nu^2\right)}\mathbb{E}\left[e^{-(u\varsigma+2w\nu\varsigma)\epsilon}e^{-w\varsigma^2\epsilon^2}\right], \qquad \epsilon \sim \mathsf{N}(0,1).\end{aligned}$$

According to the joint Laplace transform of $\left(\epsilon, \epsilon^2\right)$ in Proposition A.1 in Appendix A, we immediately have the result in (3.1). By setting $u = 0$ in (3.1), we have the univariate Laplace transform of X_t^2. □

D Proof for Lemma 3.1

Proof. The infinitesimal generator for the joint process (Λ_t, X_t, t) is given by

$$\mathcal{A}f(\Lambda, x, t) \;=\; \frac{\partial f}{\partial t} - \delta(x-\mu)\frac{\partial f}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2 f}{\partial x^2} + x^2\frac{\partial f}{\partial \Lambda}. \tag{D.1}$$

Consider an exponential martingale in form of

$$f(\Lambda, x, t) \;=\; e^{-w\Lambda}e^{-\kappa_2 x^2}e^{-\kappa_1 x}e^{\kappa_0 t}, \tag{D.2}$$

and set $\mathcal{A}f(\Lambda, x, t) = 0$. Then,

$$\kappa_0 + \delta(x-\mu)(2\kappa_2 x + \kappa_1) + \frac{1}{2}\sigma^2\left[(2\kappa_2 x + \kappa_1)^2 - 2\kappa_2\right] - wx^2 \;=\; 0, \tag{D.3}$$

which holds for any x. Therefore, we have three equations,

$$\begin{aligned}\kappa_0 - \mu\delta\kappa_1 + \frac{1}{2}\sigma^2\left(\kappa_1^2 - 2\kappa_2\right) &= 0, \\ \delta\kappa_1 - 2\mu\delta\kappa_2 + 2\sigma^2\kappa_1\kappa_2 &= 0, \\ 2\sigma^2\kappa_2^2 + 2\delta\kappa_2 - w &= 0,\end{aligned} \tag{D.4}$$

where $\kappa_0, \kappa_1, \kappa_2$ can be solved analytically as given by (3.4). The third equation has two solutions and we take the larger one. Hence, using $(\kappa_0, \kappa_1, \kappa_2)$ in (3.4) with $f(\Lambda, x, t)$ in (D.2), we find a martingale (3.3) by the basic property of infinitesimal generator. □

E Proof for Proposition 3.2

Proof. The infinitesimal generator $\tilde{\mathcal{A}}$ under the new measure $\tilde{\mathbb{P}}$ is defined as

$$\tilde{\mathcal{A}}\tilde{f} \ := \ \lim_{\Delta t \to 0^+} \frac{\tilde{\mathbb{E}}\left[\tilde{f}(t+\Delta t) \,\middle|\, \mathcal{F}_t\right] - \tilde{f}(t)}{\Delta t},$$

where $\tilde{\mathbb{E}}$ is denoted as the expectation under $\tilde{\mathbb{P}}$. Changing the measure from $\tilde{\mathbb{P}}$ to $\mathbb{P}$, we have

$$\tilde{\mathcal{A}}\tilde{f} \ = \ \lim_{\Delta t \to 0^+} \frac{\mathbb{E}\left[e^{g(t+\Delta t)-g(t)}\tilde{f}(t+\Delta t) \,\middle|\, \mathcal{F}_t\right] - \tilde{f}(t)}{\Delta t} \ = \ e^{-g}\mathcal{A}f, \tag{E.1}$$

where

$$f(\Lambda, x, t) \ = \ e^{g(\Lambda,x,t)}\tilde{f}(\Lambda, x, t), \qquad g(\Lambda, x, t) \ = \ -w\Lambda - \kappa_2 x^2 - \kappa_1 x + \kappa_0 t,$$

and $e^{g(\Lambda_t, X_t, t)}$ is the exponential ($\mathbb{P}$-)martingale (3.3). To find the parameter setting $\left(\tilde{\mu}, \tilde{\delta}, \tilde{\sigma}\right)$ under the new measure $\tilde{\mathbb{P}}$, we substitute (E.1) into the original generator (D.1), and the infinitesimal generator for (Λ_t, X_t, t) under $\tilde{\mathbb{P}}$ is then given by

$$\begin{aligned}
\tilde{\mathcal{A}}\tilde{f}(\Lambda, x, t) \ &= \ \left(\frac{\partial \tilde{f}}{\partial t} + \kappa_0 \tilde{f}\right) - \delta(x-\mu)\left[\frac{\partial \tilde{f}}{\partial x} - (2\kappa_2 x + \kappa_1)\tilde{f}\right] \\
&\quad + \frac{1}{2}\sigma^2\left\{\frac{\partial^2 \tilde{f}}{\partial x^2} - 2(2\kappa_2 x + \kappa_1)\frac{\partial \tilde{f}}{\partial x} - \left[(2\kappa_2 x + \kappa_1)^2 - 2\kappa_2\right]\tilde{f}\right\} + x^2\left(\frac{\partial \tilde{f}}{\partial \Lambda} - w\tilde{f}\right) \\
&= \ \frac{\partial \tilde{f}}{\partial t} - \left[\delta(x-\mu) + \sigma^2\left(2\kappa_2 x + \kappa_1\right)\right]\frac{\partial \tilde{f}}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2 \tilde{f}}{\partial x^2} + x^2\frac{\partial \tilde{f}}{\partial \Lambda} \\
&\quad + \left\{\kappa_0 + \delta(x-\mu)(2\kappa_2 x + \kappa_1) + \frac{1}{2}\sigma^2\left[(2\kappa_2 x + \kappa_1)^2 - 2\kappa_2\right] - wx^2\right\}\tilde{f} \\
&= \ \frac{\partial \tilde{f}}{\partial t} - \left[\delta(x-\mu) + \sigma^2\left(2\kappa_2 x + \kappa_1\right)\right]\frac{\partial \tilde{f}}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2 \tilde{f}}{\partial x^2} + x^2\frac{\partial \tilde{f}}{\partial \Lambda},
\end{aligned}$$

where the last equation infers from (D.3). Given (3.4), we then have

$$\tilde{\mathcal{A}}\tilde{f}(\Lambda, x, t) \ = \ \frac{\partial \tilde{f}}{\partial t} - \sqrt{\delta^2 + 2\sigma^2 w}\left(x - \frac{\delta^2}{\delta^2 + 2\sigma^2 w}\mu\right)\frac{\partial \tilde{f}}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2 \tilde{f}}{\partial x^2} + x^2\frac{\partial \tilde{f}}{\partial \Lambda}. \tag{E.2}$$

Comparing the current generator (E.2) with the original generator (D.1), the parameters under the new measure $\tilde{\mathbb{P}}$ can be uniquely identified as (3.6). □

F Proof for Theorem 3.1

Proof. Based on the change of measure with the Randon-Nikodým derivative (3.5), the joint bivariate Laplace transform of $(\lambda_{t+s}, \Lambda_{t+s} - \Lambda_t)$ under the original measure $\mathbb{P}$ can be elegantly expressed as the joint bivariate Laplace transform of $\left(X_{t+s}, X_{t+s}^2\right)$ under the new measure $\tilde{\mathbb{P}}$, more precisely,

$$
\begin{aligned}
& \mathbb{E}\left[e^{-u\lambda_{t+s}} e^{-w(\Lambda_{t+s}-\Lambda_t)} \,\middle|\, X_t\right] \\
= \quad & e^{w\Lambda_t} e^{-\kappa_0(t+s)} \mathbb{E}\left[e^{-w\Lambda_{t+s}} e^{-\kappa_1 X_{t+s}} e^{-\kappa_2 X_{t+s}^2} e^{\kappa_0(t+s)} \times e^{\kappa_1 X_{t+s}} e^{-(u-\kappa_2)X_{t+s}^2} \,\middle|\, X_t\right] \\
= \quad & M_t e^{w\Lambda_t} e^{-\kappa_0(t+s)} \mathbb{E}\left[\frac{M_{t+s}}{M_t} \times e^{\kappa_1 X_{t+s}} e^{-(u-\kappa_2)X_{t+s}^2} \,\middle|\, X_t\right] \\
= \quad & e^{-\kappa_1 X_t} e^{-\kappa_2 X_t^2} e^{-\kappa_0 s} \tilde{\mathbb{E}}\left[e^{\kappa_1 X_{t+s}} e^{-(u-\kappa_2)X_{t+s}^2} \,\middle|\, X_t\right].
\end{aligned}
$$

According to Proposition 3.1 and 3.2, the joint bivariate Laplace transform of $\left(X_{t+s}, X_{t+s}^2\right)$ under the new measure $\tilde{\mathbb{P}}$ is specified as

$$
\tilde{\mathbb{E}}\left[e^{\kappa_1 X_{t+s}} e^{-(u-\kappa_2)X_{t+s}^2} \,\middle|\, X_t\right] = \frac{1}{\sqrt{2\tilde{\varsigma}^2(u-\kappa_2)+1}} \exp\left(\frac{\tilde{\varsigma}^2}{2\left[2\tilde{\varsigma}^2(u-\kappa_2)+1\right]}\left(\kappa_1 + \frac{\tilde{\nu}}{\tilde{\varsigma}^2}\right)^2 - \frac{\tilde{\nu}^2}{2\tilde{\varsigma}^2}\right),
$$

where $\tilde{\nu}$ and $\tilde{\varsigma}$ are specified in (3.8). This leads to the result (3.7) for the joint bivariate Laplace transform of $(\lambda_{t+s}, \Lambda_{t+s})$ conditional on X_t under the original measure $\mathbb{P}$. □

G Proof for Theorem 3.2

Proof. By setting $u = 0$ and $w = 1$ in (3.7), we immediately obtain the conditional marginal tail distribution of s as

$$
\begin{aligned}
\mathbb{P}\left(\tau > s \mid X_t\right) &= \mathbb{E}\left[e^{-(\Lambda_{t+s}-\Lambda_t)} \,\middle|\, X_t\right] \\
&= \frac{e^{-\kappa_1 X_t} e^{-\kappa_2 X_t^2} e^{-\kappa_0 s}}{\sqrt{1-2\tilde{\varsigma}^2\kappa_2}} \exp\left(\frac{\tilde{\varsigma}^2}{2\left(1-2\tilde{\varsigma}^2\kappa_2\right)}\left(\kappa_1 + \frac{\tilde{\nu}}{\tilde{\varsigma}^2}\right)^2 - \frac{\tilde{\nu}^2}{2\tilde{\varsigma}^2}\right) \\
&= P(s)Q(s), \qquad \text{(G.1)}
\end{aligned}
$$

which follows from the doubly stochastic Poisson structure of the point process N_t. The cumulative distribution function (CDF) of τ conditional on X_t is then given by

$$
F_{\tau|X_t}\left(s \mid X_t\right) = 1 - \mathbb{P}\left(\tau > s \mid X_t\right) = 1 - P(s)Q(s), \qquad s > 0, \qquad \text{(G.2)}
$$

which implies the associated PDF of τ as

$$
f_{\tau|X_t}\left(s \mid X_t\right) = -\frac{\partial P(s)}{\partial s} \times Q(s) - P(s) \times \frac{\partial Q(s)}{\partial s} = -P(s)Q(s)\Big(p(s) + q(s)\Big).
$$

□

H Proof for Theorem 3.3

Proof. For the interarrival time τ conditional on X_t at the current arrival time t, we have

$$f_{\tau|X_t}(s \mid X_t)\,\mathrm{d}s \;=\; \mathbb{P}(\tau \in \mathrm{d}s \mid X_t) \;=\; \mathbb{E}\Big[\mathbb{1}_{\{\tau\in\mathrm{d}s\}}\,\Big|\,X_t\Big] \;=\; \mathbb{E}\Big[\lambda_{t+s^-}e^{-(\Lambda_{t+s}-\Lambda_t)}\,\Big|\,X_t\Big]\,\mathrm{d}s,$$

and

$$\mathbb{E}\Big[e^{-u\lambda_{t+\tau^-}}\mathbb{1}_{\{\tau\in\mathrm{d}s\}}\,\Big|\,X_t\Big] \;=\; \mathbb{E}\Big[e^{-u\lambda_{t+s^-}}\lambda_{t+s^-}e^{-(\Lambda_{t+s}-\Lambda_t)}\,\Big|\,X_t\Big]\,\mathrm{d}s.$$

Hence, further conditional on the interarrival time $\tau = s$, we have

$$\mathbb{E}\Big[e^{-u\lambda_{t+\tau^-}}\,\Big|\,\tau = s, X_t\Big] \;=\; \frac{\mathbb{E}\Big[e^{-u\lambda_{t+\tau^-}}\mathbb{1}_{\{\tau\in\mathrm{d}s\}}\,\Big|\,X_t\Big]}{\mathbb{E}\Big[\mathbb{1}_{\{\tau\in\mathrm{d}s\}}\,\Big|\,X_t\Big]} \;=\; \frac{\Psi'(u)}{\Psi'(0)}. \tag{H.1}$$

Both the numerator and denominator of (H.1) can be obtained analytically by differentiating the conditional joint Laplace transform of (λ_t, Λ_t) in (3.7) with $w = 1$, which immediately leads to (3.16). □

I Proof for Proposition 3.3

Proof. According to (2.3) and (2.4), we have the transition density of $X_{t+s^-} \mid X_t$ as

$$f_{X_{t+s^-}|X_t}(x \mid X_t) \;=\; \frac{1}{\sqrt{2\pi}\varsigma}e^{-\frac{(x-\nu)^2}{2\varsigma^2}}.$$

Hence,

$$\begin{aligned}
\mathbb{P}\Big(X_{t+s^-} = \sqrt{\lambda}\,\Big|\,\lambda_{t+s^-} = \lambda, X_t\Big) &\;=\; \frac{f_{X_{t+s^-}|X_t}\big(\sqrt{\lambda}\big)}{f_{X_{t+s^-}|X_t}\big(\sqrt{\lambda}\big) + f_{X_{t+s^-}|X_t}\big(-\sqrt{\lambda}\big)},\\
\mathbb{P}\Big(X_{t+s^-} = -\sqrt{\lambda}\,\Big|\,\lambda_{t+s^-} = \lambda, X_t\Big) &\;=\; 1 - \mathbb{P}\Big(X_{t+s^-} = \sqrt{\lambda}\,\Big|\,\lambda_{t+s^-} = \lambda, X_t\Big),
\end{aligned}$$

which immediately leads to Proposition 3.3. □

J Proof for Algorithm 4.1

Proof. According to (G.1) in the proof for Theorem 3.2, the tail distribution function of interarrival time τ conditional on X_t can be expressed as

$$\mathbb{P}(\tau > s \mid X_t) \;=\; P(s)Q(s) \;=\; P_0(s)Q_0(s),$$

where $P(s)$ and $Q(s)$ are specified in (3.10) and (3.11), respectively,

$$P_0(s) \;:=\; \frac{e^{-\kappa_0 s}}{\sqrt{1-2\tilde{\varsigma}^2\kappa_2}}\exp\left(\frac{\tilde{\mu}^2\left(1-e^{-\tilde{\delta}s}\right)}{2\delta^2\left(1-2\tilde{\varsigma}^2\kappa_2\right)}\Big[\left(\tilde{\delta}+2\delta\right)+\left(\tilde{\delta}-2\delta\right)e^{-\tilde{\delta}s}\Big]\right) \tag{J.1}$$

is a constant function of X_t,[18] and

$$Q_0(s) \;:=\; \exp\left(-\kappa_1 X_t - \kappa_2 X_t^2 + \frac{X_t e^{-\tilde{\delta} s}}{1-2\tilde{\varsigma}^2\kappa_2}\left[\frac{2\tilde{\mu}}{\delta} + \kappa_2\left(X_t - 2\tilde{\mu}\right)e^{-\tilde{\delta} s}\right]\right). \tag{J.2}$$

If $\mathbf{Condition\ 1}$ and any of $\mathbf{Condition\ 2a{-}2d}$ are satisfied, then we have

$$\lim_{s\to 0^+} P_0(s) \;=\; \lim_{s\to 0^+} Q_0(s) \;=\; 1, \qquad \lim_{s\to+\infty} P_0(s) \;=\; 0, \qquad \lim_{s\to+\infty} Q_0(s) \;=\; e^{-\kappa_1 X_t - \kappa_2 X_t^2} \in (0,1),$$

and

$$\begin{aligned} \frac{\partial P_0(s)}{\partial s} \;&=\; -P_0(s)\left(\frac{\tilde{\varsigma}^2}{1-2\tilde{\varsigma}^2\kappa_2} + \frac{\tilde{\mu}^2\left(1-e^{-\tilde{\delta} s}\right)^2}{4\delta^2\left(1-2\tilde{\varsigma}^2\kappa_2\right)^2}\left[\tilde{\delta}\left(1+e^{-\tilde{\delta} s}\right) + \delta\left(1-e^{-\tilde{\delta} s}\right)\right]^2\right) \;<\; 0, \\ \frac{\partial Q_0(s)}{\partial s} \;&=\; -\frac{e^{-\tilde{\delta} s}Q_0(s)}{\left(1-2\tilde{\varsigma}^2\kappa_2\right)^2}\left(a^* e^{-2\tilde{\delta} s} + b^* e^{-\tilde{\delta} s} + c^*\right) \;<\; 0. \end{aligned}$$

Therefore, $P_0(s)$ and $Q_0(s)$ are both strictly decreasing functions on $(0,+\infty)$: $P_0(s)$ decreases from 1 to 0, and $Q_0(s)$ decreases from 1 to $e^{-\kappa_1 X_t - \kappa_2 X_t^2}$. We have a *well-defined* random variable V^* with the CDF

$$F_{V^*}(s) \;=\; 1 - \frac{e^{-\kappa_0 s}}{\sqrt{1-2\tilde{\varsigma}^2\kappa_2}}\exp\left(\frac{\tilde{\varsigma}^2}{2\left(1-2\tilde{\varsigma}^2\kappa_2\right)}\left[\kappa_1 + \frac{\tilde{\mu}\left(1-e^{-\tilde{\delta} s}\right)}{\tilde{\varsigma}^2}\right]^2 - \frac{\tilde{\mu}^2\left(1-e^{-\tilde{\delta} s}\right)^2}{2\tilde{\varsigma}^2}\right), \tag{J.3}$$

and the PDF

$$f_{V^*}(s) \;=\; P_0(s)\left(\frac{\tilde{\varsigma}^2}{1-2\tilde{\varsigma}^2\kappa_2} + \frac{\tilde{\mu}^2\left(1-e^{-\tilde{\delta} s}\right)^2}{4\delta^2\left(1-2\tilde{\varsigma}^2\kappa_2\right)^2}\left[\tilde{\delta}\left(1+e^{-\tilde{\delta} s}\right) + \delta\left(1-e^{-\tilde{\delta} s}\right)\right]^2\right), \tag{J.4}$$

for $s>0$. We have a *defective* random variable V_t^* with the CDF

$$F_{V_t^*}(s) \;=\; \begin{cases} 1 - \exp\left(-\kappa_1 X_t - \kappa_2 X_t^2 + \dfrac{X_t e^{-\tilde{\delta} s}}{1-2\tilde{\varsigma}^2\kappa_2}\left[\dfrac{2\tilde{\mu}}{\delta} + \kappa_2\left(X_t - 2\tilde{\mu}\right)e^{-\tilde{\delta} s}\right]\right), & 0<s<+\infty, \\ 1, & s=+\infty, \end{cases} \tag{J.5}$$

and the PDF

$$f_{V_t^*}(s) \;=\; \frac{e^{-\tilde{\delta} s}Q_0(s)}{\left(1-2\tilde{\varsigma}^2\kappa_2\right)^2}\left(a^* e^{-2\tilde{\delta} s} + b^* e^{-\tilde{\delta} s} + c^*\right), \qquad s>0. \tag{J.6}$$

Note that, V^* is independent of X_t since its CDF (J.3) is a constant function of X_t, and the CDF (J.5) of V_t^* indicates that

$$\lim_{s\to+\infty} F_{V_t^*}(s) \;=\; 1 - e^{-\kappa_1 X_t - \kappa_2 X_t^2} \;<\; 1 \;\neq\; F_{V_t^*}(+\infty),$$

and hence

$$\mathbb{P}\left(V_t^* = +\infty\right) \;=\; e^{-\kappa_1 X_t - \kappa_2 X_t^2}.$$

[18] We can derive $P_0(s)$ by setting $X_t = 0$ in $P(s)Q(s)$.

We can exactly simulate V^* and V_t^* via the *explicit* inverse transformation method in Algorithm 4.2 and the A/R scheme in Algorithm 4.3, respectively. Finally, the interarrival time τ conditional on X_t can be then exactly simulated by

$$\tau \overset{\mathcal{D}}{=} \min\{V^*, V_t^*\},$$

since

$$\mathbb{P}(\tau > s \mid X_t) = P_0(s)Q_0(s) = \mathbb{P}(V^* > s) \times \mathbb{P}(V_t^* > s).$$

□

K Proof for Algorithm 4.2

Proof. To sample V^* with the PDF $f_{V^*}(s)$ in (J.4), we choose an exponential *envelop* $\hat{V}$ with the PDF

$$g_{\hat{V}}(s) = \lambda e^{-\lambda s}, \qquad \lambda \in \mathbb{R}^+.$$

The ratio of two densities, $R_{V^*}(\lambda, s)$, is given as

$$R_{V^*}(\lambda, s) = \frac{f_{V^*}(s)}{g_{\hat{V}}(s)} = \frac{e^{\lambda s} f_{V^*}(s)}{\lambda} \leq R_{V^*}(\lambda, s^*),$$

such that

$$R_{V^*}(\lambda, s^*) = \max_{s \in \mathbb{R}^+} \{R_{V^*}(\lambda, s)\}.$$

Overall, the ratio $R_{V^*}(\lambda, s)$ is bounded by $R_{V^*}(\lambda, s^*)$, which is the associated A/R constant. The critical value λ^* can be obtained by minimising $R_{V^*}(\lambda, s^*) = \max_{s \in \mathbb{R}^+} \{R_{V^*}(\lambda, s)\}$ in order to reduce the expected number of simulated candidates to be rejected. This numerical maximisation is carried out *only once* before entering the loop for sampling the point process since V^* is independent of X_t. It induces almost no cost, but produces an *optimal* A/R scheme for *exact* simulation. □

L Proof for Algorithm 4.3

Proof. The *defective* random variable V_t^* can be exactly simulated via an *explicit* inverse transformation, since its CDF (J.5) can be inverted *explicitly* as

$$F_{V_t^*}^{-1}(u) = \begin{cases} -\dfrac{1}{\tilde{\delta}} \ln \left(\dfrac{-b + \sqrt{b^2 - 4\hat{c}\left(\hat{a} - \dfrac{\tilde{\sigma}^2 \kappa_2}{\tilde{\delta}}\left[\ln(1-u) + \hat{Z}\right]\right)\left[\ln(1-u) + \hat{Z}\right]}}{2\left(\hat{a} - \dfrac{\tilde{\sigma}^2 \kappa_2}{\tilde{\delta}}\left[\ln(1-u) + \hat{Z}\right]\right)} \right), & u \in \left[0,\ 1 - e^{-\hat{Z}}\right), \\ +\infty, & u \in \left[1 - e^{-\hat{Z}},\ 1\right], \end{cases} \tag{L.1}$$

where

$$\hat{a} := \kappa_2 X_t (X_t - 2\tilde{\mu}), \qquad \hat{c} := -\frac{1}{2\tilde{\delta}\kappa_2}, \qquad \hat{Z} := \kappa_1 X_t + \kappa_2 X_t^2.$$

Note that, $1-U$ is also uniform on $[0,1]$, i.e., $U \overset{\mathcal{D}}{=} 1-U$ for $U \sim \mathsf{U}[0,1]$. We can simplify (L.1) as

$$V_t^* \ \overset{\mathcal{D}}{=} \ F_{V_t^*}^{-1}(1-U) \ = \ \begin{cases} -\frac{1}{\tilde{\delta}} \ln\left(\frac{-b+\sqrt{b^2-4ac}}{2a}\right), & U \in \left(e^{-\hat{z}},\, 1\right], \\ +\infty, & U \in \left[0,\, e^{-\hat{z}}\right]. \end{cases}$$

□

M Proof for Algorithm 4.4

Proof. To sample the interarrival time τ conditional on X_t at the current arrival time t with the PDF $f_{\tau|X_t}(s \mid X_t)$ in (3.9), we choose an *envelop* of exponential distribution $\hat{\tau} \sim \mathsf{Exp}(\kappa_0)$ with the PDF

$$g_{\hat{\tau}}(s) \ = \ \kappa_0 e^{-\kappa_0 s}, \qquad s > 0.$$

The ratio of two PDFs is given by

$$R_\tau(s) \ = \ \frac{f_{\tau|X_t}(s \mid X_t)}{g_{\hat{\tau}}(s)} \ = \ -\frac{e^{\kappa_0 s}}{\kappa_0} P(s)Q(s)\Big(p(s)+q(s)\Big),$$

which is bounded, since it is continuous and

$$\lim_{s\to 0^+} R_\tau(s) \ = \ C_0 \ < \ +\infty, \qquad \lim_{s\to +\infty} R_\tau(s) \ = \ C_\infty \ < \ +\infty.$$

Denoting C_M as the internal maximum of $R_\tau(s)$ on $(0,+\infty)$, which can be obtained via numerical optimisation, we have

$$R_\tau(s) \ \leq \ C_\tau \ = \ \max\{C_0, C_M, C_\infty\}$$

for all $s > 0$, where C_τ is the associated A/R constant. The acceptance condition of A/R scheme for sampling τ is

$$U \ \leq \ \frac{R_\tau(\hat{\tau})}{C_\tau},$$

where $U \sim \mathsf{U}[0,1]$ and $\hat{\tau} \sim \mathsf{Exp}(\kappa_0)$. □

N Proof for Algorithm 4.5

Proof. According to Theorem 3.3, we can rewrite $\Psi(u)$ in (3.17) with $s=\tau$ in a more concise form as

$$\Psi(u) \ = \ g^{-\frac{1}{2}}(u)\exp\Big(Ag^{-1}(u)\Big) \ = \ \frac{1}{\sqrt{Bu+C}}\exp\left(\frac{A}{Bu+C}\right) \tag{N.1}$$

with $\Psi(0) = \frac{e^{\frac{A}{C}}}{\sqrt{C}}$, and

$$\Psi'(u) \ = \ -\frac{B}{(Bu+C)^{\frac{3}{2}}}\left(\frac{A}{Bu+C}+\frac{1}{2}\right)\exp\left(\frac{A}{Bu+C}\right), \qquad \Psi'(0) \ = \ -\frac{B\left(\frac{A}{C}+\frac{1}{2}\right)e^{\frac{A}{C}}}{C^{\frac{3}{2}}},$$

where

$$g(u) \;:=\; 2\tilde{\varsigma}^2 u + \left(1 - 2\tilde{\varsigma}^2 \kappa_2\right) \;\equiv\; Bu + C$$

with

$$g'(u) \;=\; 2\tilde{\varsigma}^2 \;=\; B, \qquad g(0) \;=\; 1 - 2\tilde{\varsigma}^2 \kappa_2 \;=\; C \;>\; 0.$$

Hence, the Laplace transform (3.16) of pre-event intensity level conditional on the realised interarrival time τ can be expressed as

$$\begin{aligned}
\mathbb{E}\left[e^{-u\lambda_{t+\tau^-}} \,\middle|\, \tau, X_t\right] &= \frac{\left(\frac{C}{Bu+C}\right)^{\frac{3}{2}} \left(\frac{A}{Bu+C} + \frac{1}{2}\right) e^{-\frac{A}{C}}}{\frac{A}{C} + \frac{1}{2}} \exp\left(\frac{A}{Bu+C}\right) \\
&= \frac{\left(\frac{C}{Bu+C}\right)^{\frac{3}{2}} \left(\frac{A}{Bu+C} + \frac{1}{2}\right) e^{-\frac{A}{C}}}{\frac{A}{C} + \frac{1}{2}} \sum_{j=0}^{\infty} \frac{1}{j!} \left(\frac{A}{Bu+C}\right)^j \\
&= \frac{\left(\frac{C}{Bu+C}\right)^{\frac{3}{2}} e^{-\frac{A}{C}}}{\frac{A}{C} + \frac{1}{2}} \left[\sum_{j=0}^{\infty} \frac{1}{j!} \left(\frac{A}{Bu+C}\right)^{j+1} + \frac{1}{2} \sum_{j=0}^{\infty} \frac{1}{j!} \left(\frac{A}{Bu+C}\right)^j\right] \\
&= \frac{\left(\frac{C}{Bu+C}\right)^{\frac{3}{2}} e^{-\frac{A}{C}}}{\frac{A}{C} + \frac{1}{2}} \left[\sum_{j=0}^{\infty} \frac{\left(\frac{A}{C}\right)^j}{j!} \left(\frac{C}{Bu+C}\right)^{j+1} + \frac{1}{2} \sum_{j=0}^{\infty} \frac{\left(\frac{A}{C}\right)^j}{j!} \left(\frac{C}{Bu+C}\right)^j\right] \\
&= \frac{\frac{A}{C}}{\frac{A}{C} + \frac{1}{2}} \sum_{j=0}^{\infty} \mathbb{P}(J=j) \left(\frac{\frac{C}{B}}{u + \frac{C}{B}}\right)^{j+\frac{5}{2}} + \frac{\frac{1}{2}}{\frac{A}{C} + \frac{1}{2}} \sum_{j=0}^{\infty} \mathbb{P}(J=j) \left(\frac{\frac{C}{B}}{u + \frac{C}{B}}\right)^{j+\frac{3}{2}} \\
&= p_\lambda \sum_{j=0}^{\infty} \mathbb{P}(J=j) \mathbb{E}\left[e^{-uG_1^{(j)}}\right] + (1 - p_\lambda) \sum_{j=0}^{\infty} \mathbb{P}(J=j) \mathbb{E}\left[e^{-uG_2^{(j)}}\right] \\
&= p_\lambda \mathbb{E}\left[e^{-uG_1}\right] + (1 - p_\lambda)\, \mathbb{E}\left[e^{-uG_2}\right],
\end{aligned}$$

where $G_1^{(j)} \sim \mathsf{Gamma}\left(j + \frac{5}{2}, \frac{C}{B}\right)$ and $G_2^{(j)} \sim \mathsf{Gamma}\left(j + \frac{3}{2}, \frac{C}{B}\right)$ for $j = 0, 1, \cdots$. Quite elegantly, the pre-event intensity level $\lambda_{t+\tau^-}$ conditional on X_t and τ can be *exactly decomposed* as

$$\lambda_{t+\tau^-} \,|\, \tau, X_t \;\stackrel{\mathcal{D}}{=}\; \begin{cases} G_1, & \text{with probability } p_\lambda, \\ G_2, & \text{with probability } 1 - p_\lambda. \end{cases}$$

Given the pre-event intensity level $\lambda_{t+\tau^-}$, we can back out the underlying pre-event position of state process, $X_{t+\tau^-}$, in order to sample the next interarrival time. Conditional on X_t, τ and $\lambda_{t+\tau^-}$, according to Proposition 3.3, we have

$$X_{t+\tau^-} \;\stackrel{\mathcal{D}}{=}\; B_X \sqrt{\lambda_{t+\tau^-}} - (1 - B_X) \sqrt{\lambda_{t+\tau^-}} \;=\; (2B_X - 1) \sqrt{\lambda_{t+\tau^-}}$$

where $B_X \sim \mathsf{Bernoulli}\,(p_X)$,

$$p_X \;=\; \frac{e^{-\frac{\left(\sqrt{\lambda_{t+\tau^-}} - \nu\right)^2}{2\varsigma^2}}}{e^{-\frac{\left(\sqrt{\lambda_{t+\tau^-}} - \nu\right)^2}{2\varsigma^2}} + e^{-\frac{\left(\sqrt{\lambda_{t+\tau^-}} + \nu\right)^2}{2\varsigma^2}}},$$

and (ν, ς) are specified in (2.5) by setting $s = \tau$. □

O Euler Algorithm of Abate and Whitt (2006, §5)

For the Laplace transform $\hat{f}$ of a given function f, i.e.,

$$\hat{f}(s) \;:=\; \int_0^\infty e^{-st} f(t) \mathrm{d}t,$$

the underlying function f can be approximated by

$$f(t) \;\approx\; f_n(t) \;:=\; \frac{1}{t} \sum_{k=0}^{n} \omega_k \hat{f}\left(\frac{\alpha_k}{t}\right), \qquad t > 0, \tag{O.1}$$

where *nodes* α_k and *weights* ω_k are (complex-valued) *tuning parameters* which depend on n but neither on the Laplace transform $\hat{f}$ nor the time argument t. For the *Euler algorithm* (Abate and Whitt, 2006, §5), the parameters in (O.1) are $n = 2M$ for a positive integer M, $\alpha_k = \beta_k$, $\omega_k = 10^{\frac{M}{3}} \eta_k$ and

$$\begin{aligned}
\beta_k &= \frac{\ln 10}{3} M + \pi i k, \qquad \eta_k \equiv (-1)^k \xi_k, \\
\xi_0 &= \frac{1}{2}, \qquad \xi_k \;=\; 1, \qquad 1 \le k \le M, \qquad \xi_{2M} \;=\; \frac{1}{2^M}, \\
\xi_{2M-k} &= \xi_{2M-k+1} + 2^{-M} \binom{M}{k}, \qquad 0 < k < M.
\end{aligned}$$

P Simulation Algorithms for Extensions in Section 5

Based on the current arrival time $t = 0$, a self-exciting and self-correcting point process N_t with quadratic intensity $\lambda_t = X_t^2$ and state X_t in (5.1) can be exactly simulated via Algorithm P.1.

Algorithm P.1

```
$T \leftarrow 0$
$N_T \leftarrow 0$
$X_T \leftarrow X_0$
**repeat**
    **sample** $\tau$ via Algorithm 4.1
    **sample** $\lambda_{T+\tau^-}$ and $X_{T+\tau^-}$ via Algorithm 4.5
    **sample** $Y$ with an arbitrary CDF $G(\cdot)$
    $T \leftarrow T + \tau$
    $X_T \leftarrow X_{T+\tau^-} + Y$
    $N_T \leftarrow N_T + 1$
**until** $\big(T > t\big)$
$N_t \leftarrow N_T - 1$
**return** $N_t$
```

Based on the current arrival time $t = 0$, a self-exciting/correcting and externally-exciting/correcting point process N_t with quadratic intensity $\lambda_t = X_t^2$ and state X_t in (5.3) can be exactly simulated via Algorithm P.2.

Algorithm P.2

```
$T \;\leftarrow\; 0$
$N_T \;\leftarrow\; 0$
$X_T \;\leftarrow\; X_0$
repeat
    sample $\tau_{\mathsf{en}}$ via Algorithm 4.1
                                                        ▷ by replacing $\tau$ with $\tau_{\mathsf{en}}$
    sample $\tau_{\mathsf{ex}} \sim \mathsf{Exp}(\varrho)$
    $\tau \;\leftarrow\; \min\{\tau_{\mathsf{en}}, \tau_{\mathsf{ex}}\}$
    sample $\lambda_{T+\tau^-}$ and $X_{T+\tau^-}$ via Algorithm 4.5
    if $\big(\tau = \tau_{\mathsf{en}}\big)$ then
        $N_T \;\leftarrow\; N_T + 1$
        sample $Y$ with an arbitrary CDF $G(\cdot)$
        $T \;\leftarrow\; T + \tau$
        $X_T \;\leftarrow\; X_{T+\tau^-} + Y$
    else
        sample $Z$ with an arbitrary CDF $G_Z(\cdot)$
        $T \;\leftarrow\; T + \tau$
        $X_T \;\leftarrow\; X_{T+\tau^-} + Z$
    end if
until $\big(T > t\big)$
$N_t \;\leftarrow\; N_T - 1$
return $N_t$
```

Note that, τ in Algorithm P.2 is now (re)interpreted as the interarrival time of jumps in the intensity process λ_t rather than jumps in the point process N_t, since exogenous and endogenous jumps do not arrive at the same time.

Based on the current arrival time $t = 0$, for a multidimensional point process $\left\{N_t^{(\jmath)}\right\}_{\jmath=1,2,\cdots,D}$, the $\jmath^{\text{th}}$-component point process $N_t^{(\jmath)}$ with quadratic intensity $\lambda_t^{(\jmath)} = \left(X_t^{(\jmath)}\right)^2$ and the state process $X_t^{(\jmath)}$ in (5.4) for each $\jmath = 1, 2, \cdots, D$ can be exactly simulated via Algorithm P.3.

Algorithm P.3

```
$T \leftarrow 0$
$N_T^{(j)} \leftarrow 0$
$X_T^{(j)} \leftarrow X_0^{(j)}$
**repeat**
    **sample** $\tau^{(\ell)}$ via Algorithm 4.1 for $\ell = 1, 2, \cdots, D$
        ▷ by replacing $\tau$ with $\tau^{(\ell)}$ using the associated parameters $(\mu_\ell, \delta_\ell, \sigma_\ell)$
    $\tau \leftarrow \min\left\{\tau^{(1)}, \cdots, \tau^{(D)}\right\}$
    **sample** $\lambda_{T+\tau^-}^{(j)}$ and $X_{T+\tau^-}^{(j)}$ via Algorithm 4.5
    **sample** a vector $(Y_1, \cdots, Y_D)$ with an arbitrary joint CDF $\mathbf{G}(\mathbf{y}), \mathbf{y} \in \mathbb{R}^D$
    **if** $\left(\tau = \tau^{(j)}\right)$ **then**
        $N_T^{(j)} \leftarrow N_T^{(j)} + 1$
    **end if**
    $T \leftarrow T + \tau$
    $X_T^{(\ell)} \leftarrow X_{T+\tau^-}^{(\ell)} + Y_\ell$ for $\ell = 1, 2, \cdots, D$
**until** $\left(T > t\right)$
$N_t^{(j)} \leftarrow N_T^{(j)} - 1$
**return** $N_t^{(j)}$
```