Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

SIMULATING NONSTATIONARY SPATIO-TEMPORAL POISSON PROCESSES USING THE INVERSION METHOD

Haoting Zhang
Zeyu Zheng

Department of Industrial Engineering and Operations Research
University of California, Berkeley
Berkeley, CA 94720, USA

ABSTRACT

We study the problem of simulating a class of nonstationary spatio-temporal Poisson processes. The Poisson intensity function is non-stationary and piecewise linear in both the time dimension and the spatial location dimensions. We propose an exact simulation algorithm based on the inversion method. This simulation algorithm adopts three advantages. First, the entire procedure involves only closed-form computation with no need for numerical integration or numerical inversion of any function. Each step in the algorithm only requires exact arithmetic operations. Second, the proposed algorithm is sample efficient, especially compared to the thinning method when the maximum intensity value is much larger than the minimum intensity value. Third, the algorithm generates arrivals sequentially, one at a time in ascending order, so that they can be conveniently fed into real-time or online decision-making tools.

1 INTRODUCTION

A spatio-temporal point process is a random point field that models both temporal and spatial dispersions of points. Each point represents the arrival of an event or an entity at a specific time and location. The arrival time belongs to a time interval $[0,T]$ (for example, an operational period); and the arrival location rests within a set of spatial locations $S \subset \mathbb{R}^2$ (for example, a region of interest). For any sub-space $S_1 \subset S$ and time sub-interval $[t_1,t_1'] \subset [0,T]$, $N(S_1 \times [t_1,t_1'])$ records the total number of arrivals whose arrival location and arrival time fall into $S_1 \times [t_1,t_1']$. A classical and widely used point process model is a Poisson process. The non-stationary spatio-temporal structure is characterized by a non-negative Poisson intensity function $\lambda(s,t)$ where s denotes location and t denotes time. For a spatio-temporal Poisson process, given any $S_1 \subset S$ and $0 \le t_1 < t_1' \le T$, $N(S_1 \times [t_1,t_1'])$ is Poisson distributed with mean parameter

$$\mathbb{E}\left(N\left(S_1 \times \left[t_1,t_1'\right]\right)\right) = \int_{s\in S_1} \int_{t\in\left[t_1,t_1'\right]} \lambda(s,t)\, ds\, dt.$$

Moreover, if $S_1 \times [t_1,t_1'], S_2 \times [t_2,t_2'], \ldots, S_m \times [t_m,t_m']$ are m disjoint subsets of $S \times [0,T]$, the arrival counts attributed to each subset $N(S_1 \times [t_1,t_1']), N(S_2 \times [t_2,t_2']), \ldots, N(S_m \times [t_m,t_m'])$ are independent.

The non-stationary spatio-temporal Poisson process (NSSTPP) is a popular model in applications with time-varying arrival streams with heterogeneous geographical information. The Poisson assumption can be justified when the point process is used to model arrivals that originate from many independent sources. One theoretic support of the Poisson assumption is given by the Palm-Khintchine theorem, describing that the superposition of a large number of independent point processes with small volumes can approximately be viewed as a Poisson process (Heyman and Sobel 2004). Therefore, spatio-temporal Poisson processes are especially suitable to model arrivals of entities if each one of them arrives at the system independent of the others. The calling for a taxi or a ride-sharing service, for example, could be regarded as a Poisson

process as passengers largely make their own decisions on when and where to use the service. The NSSTPP also serves as a baseline model to add additional realistic features, such as a self-exciting feature and an event-driven feature, which are not captured by the Poisson assumption. Our work makes the Poisson process assumption and focuses primarily on the modeling and simulation of NSSTPPs.

In order to build a spatio-temporal Poisson point process model with time-varying and location-varying intensity, we start by partitioning the space and the time interval into disjoint sub-spaces and sub-intervals. The resolution of the partition typically depends on the application objective and the sensitivity to model granularity. Once the partition is done, the simplest non-stationary intensity function is then given by assigning the intensity value to be constant for a given sub-space over a given sub-interval of time. Such a model is called a piecewise constant spatio-temporal intensity model. However, such an intensity model seems aesthetically unreasonable, given its implausible discontinuities, and is likely to model poorly arrivals that are close to the sub-interval boundaries and at locations that are close to sub-space boundaries. A piecewise linear spatio-temporal intensity is a natural next step to consider that strikes a balance between model granularity and model complexity. Specifically, for any given time t, the intensity function is linear over each sub-space, and for any given location s, the intensity is linear over each time sub-interval. The continuity of the intensity function, if desired, can be easily ensured in the time dimension, by connecting the intensity value over boundaries of time sub-intervals (see Zheng and Glynn (2017)). However, the continuity is not as straightforward to ensure over the boundaries of sub-space. In fact, not every type of each sub-space partition allows a successful construction of a continuous piecewise linear intensity. In this paper, we propose a specific triangulation partition of space. That is, we consider a smallest rectangle that covers S and then divide the space into rectangular sub-regions based on the partition of the x-axis and y-axis, and each rectangle is further partitioned into two adjacent right triangles sharing the same hypotenuse, denoted as lower and upper right triangles. Given the triangulation partition of space, a spatio-temporal piecewise linear intensity that is continuous in location can be fully determined by the intensity value on the triangle vertices.

With a given partition of space and time and the piecewise linear intensity in hand, we propose an efficient simulation algorithm based on the inversion method. The algorithm is designed by first sequentially generating arrival times based on the integrated intensity over space. Then, conditional on each arrival time generated, the location of that arrival is sampled according to the conditional probability density function (p.d.f.) in space. The inversion method is used in both steps — simulating arrival times and simulating arrival locations. The function inversions used in the algorithm all adopt closed-form representations that require no numerical integration or numerical inversion of functions. We describe the algorithm details and implement the algorithm with two simple numerical examples.

2 LITERATURE REVIEW

Poisson point processes are widely used to model the times at which arrivals enter a system. Common examples include calls to a service center, clicks on a website, and the stream of photons from an optical modulator. When the locations of arrivals are taken into consideration, especially in the areas of transportation and the sharing economy, a spatio-temporal Poisson process is then required to construct the model. Other application areas of Poisson process models include astronomy (Babu and Feigelson 1996), biology (Othmer et al. 1988), ecology (Achcar et al. 2011) and wireless networks (Haenggi 2012). A Poisson process with non-stationary intensities can appropriately capture the time-of-day and location-of-area effects, which may be first-order features that impact system performances and decisions for a wide range of applications. Different forms of intensities can be modeled to capture the non-stationarity. In Zhou et al. (2015), for example, time-varying Gaussian mixture intensity models are modeled to predict the ambulance demand. Other non-Poisson features such as the self-exciting behavior can be added on top of the standard Poisson process models, which are popular in application areas such as seismology (Ogata and Zhuang 2006) and criminology (Mohler et al. 2011), where a Poisson process is often regarded as a background random field.

The algorithms of simulating Poisson processes can be generally classified into two categories: the method of inversion and the method of thinning (see Pasupathy (2011) for a review). The thinning method (see Lewis and Shedler (1979a) for reference) first generates a Poisson process with a simple intensity function (a constant function, for example) that dominates the true underlying intensity function. Then, each of the generated arrivals with time and location information is accepted with probability proportional to the ratio of the associated intensity value divided by the dominating constant intensity value. The method of thinning is particularly attractive and easy to implement when the intensity value is not changing much over time and space, or if there is a simple majorizing function that dominates the underlying intensity function. However, when the intensity exhibits heavy fluctuations in time and location, the thinning method can be sample inefficient in the sense that a large proportion of the generated arrivals can be rejected. Using a closely majorizing function helps mitigate this inefficiency. In fact, a widely used majorizing function is a piecewise linear intensity function. In those cases, the use of the thinning method is based on first efficiently simulating a Poisson process with a piecewise linear intensity, which is exactly the focus of our paper.

The use of the inversion method (see Çinlar (2013)), alternatively, can be sample efficient where no generated samples are rejected or wasted. However, the inversion method requires the cumulative intensity function and its inversion to be tractable. When the Poisson process only has a temporal structure, popular intensity models that are suitable for the inversion method include piecewise constant intensities (Bratley et al. 1987; Harrod and Kelton 2006), piecewise linear intensities (Klein and Roberts 1984) and a handful of other specific intensity models (Lewis and Shedler 1976; Lewis and Shedler 1979b; Chen and Schmeiser 1992). Numerical integration and numerical inversion are needed for more complicated intensity models. When the Poisson process involves both temporal and spatial structure, it is less clear which models are suitable for the use of inversion method, and, in particular, how to efficiently perform the inversion method.

Combinations of the inversion method and the thinning method are often jointly used to enhance efficiency. When the Poisson process only has a temporal structure, a piecewise constant or piecewise linear majorizing function is usually chosen to dominate the underlying intensity function. Then the inversion method is first used to simulate a Poisson process whose intensity is given by the majorizing function. Then the thinning method is used to accept or reject arrivals to generate a point process with the desired intensity. See Liu et al. (2019), Lee et al. (1991), Bratley et al. (1987), Ross (2013) for reference.

When the Poisson process has both temporal and spatial structure, Saltzman et al. (2012) discuss a simulation algorithm suitable for general multi-dimensional Poisson processes. Their idea is to first project the Poisson process onto a lower dimension subspace. The use of projection allows the simulation procedure to be split into two independent and simpler parts. First, the projected Poisson process (in the lower dimension) is simulated according to an integrated intensity function that integrates out the other dimensions. Then, the conditional p.d.f. given the arrival time is exploited to determine the coordinates of arrivals in the other dimensions. Our algorithm uses the same philosophy by first simulating in the time dimension and then in the space dimension. Given the piecewise linear intensity model in time and location, both the integrated intensity function and the conditional p.d.f. are also piecewise linear. This feature allows the use of the inversion method in both parts. In particular, our algorithm does not require any numerical integration and numerical inversion of functions. All computation is closed-form and involves only arithmetic operations.

3 METHOD

3.1 Model Description

We consider the time horizon of interest as $[0,T]$ and the space of interest as S. With a given partition of space and time, we propose a continuous piecewise linear intensity function on $S \times [0,T]$, such that the intensity is linear over each sub-interval of time for a given location and is linear over each sub-space at a given time. The continuity is ensured at the time sub-interval boundaries and the location sub-space boundaries. For the

proper definition of continuity of the intensity function on the sub-space boundaries, the partition of space is via triangulation. That is, each sub-space is a right triangle. Two adjacent right triangles share a hypotenuse and two vertices. Without loss of generality, we take the view that the full space $S \subset \mathbb{R}^2$ is a rectangle. (If the real full space S is not a rectangle, we can always find a set of rectangles to cover S, and then constrain the intensity to be zero on the excessive space.) Consider the following specific triangulation of S induced by the two-dimensional grids. Set $X = (x_0, x_1, \ldots, x_K)$ and $Y = (y_0, y_1, \ldots, y_J)$ as two vectors with ascending real elements. The full space rectangle S has its four vertices given by $(x_0, y_0), (x_0, y_J), (x_K, y_J), (x_K, y_0)$. Each subspace is a right triangle with vertices $(x_k, y_j), (x_{k+1}, y_j), (x_k, y_{j+1})$ or $(x_k, y_j), (x_{k-1}, y_j), (x_k, y_{j-1})$ for all the feasible j, k's. There are in total $2JK$ right triangles partitioning S. Assume further that the time interval is partitioned as $0 = s_0 < s_1 < s_2 < \cdots < s_p = T$. Denote the intensity value at vertex (x_k, y_j, s_i) as $z_{k,j,i}$ for $0 \le k \le K, 0 \le j \le J$, and $0 \le i \le p$. Next, we define a piecewise linear intensity function that can be fully specified by the intensity values $\{z_{k,j,i}\}$. The intensity $\lambda(x,y,t)$ for any given $(x,y) \in S$, $0 \le t \le T$ is fully determined by the $z_{k,j,i}$'s and is an affine function of the $\{z_{k,j,i} : \text{ all } k, j, i\}$. Note that for any (x,y,t) there must exist j, k, i such that

$$x \in [x_{k-1}, x_k], y \in [y_{j-1}, y_j], t \in [s_{i-1}, s_i].$$

If $y < y_{j-1} + (y_j - y_{j-1}) \frac{x_k - x}{x_k - x_{k-1}}$, (x,y) lies in the lower right triangular sub-space in the rectangle $[x_{k-1}, x_k] \times [y_{j-1}, y_j]$, with triangle vertices $(x_{k-1}, y_{j-1}), (x_k, y_{j-1}), (x_{k-1}, y_j)$. Otherwise, (x,y) lies in the upper right triangle with vertices $(x_k, y_j), (x_k, y_{j-1}), (x_{k-1}, y_j)$. The intensity of an arbitrary point (x,y) at t is determined based on the coplanarity. We present the detailed derivation process in the supplement document. Define

$$f_{k'',j'',k',j'}(x,y) \triangleq \begin{vmatrix} x - x_{k''} & y - y_{j''} \\ x - x_{k'} & y - y_{j'} \end{vmatrix}.$$

Then, if (x,y) is contained in the lower right triangular sub-space, the intensity value is given by

$$\begin{aligned} &\lambda(x,y,t) \\ =&\frac{f_{k,j-1,k-1,j}(x,y)\lambda(x_{k-1}, y_{j-1}, t) + f_{k-1,j,k-1,j-1}(x,y)\lambda(x_k, y_{j-1}, t) + f_{k-1,j-1,k,j-1}(x,y)\lambda(x_{k-1}, y_j, t)}{f_{k,j-1,k-1,j}(x,y) + f_{k-1,j,k-1,j-1}(x,y) + f_{k-1,j-1,k,j-1}(x,y)}. \end{aligned} \tag{1}$$

Otherwise, if (x,y) is contained in the upper right triangle, the intensity value is given by

$$\begin{aligned} &\lambda(x,y,t) \\ =&\frac{f_{k,j-1,k,j}(x,y)\lambda(x_{k-1}, y_j, t) + f_{k,j,k-1,j}(x,y)\lambda(x_k, y_{j-1}, t) + f_{k-1,j,k,j-1}(x,y)\lambda(x_k, y_j, t)}{f_{k,j-1,k,j}(x,y) + f_{k,j,k-1,j}(x,y) + f_{k-1,j,k,j-1}(x,y)}. \end{aligned} \tag{2}$$

Note that in (1) and (2), the intensity value at (x,y,t) is specified as a linear interpolation of intensity values at the points $\{(x_k, y_j, t)\}$ where the $\{(x_k, y_j)\}$ are vertices of some triangle sub-space. Then, for any $0 \le k' \le K, 0 \le j' \le J$ and $t \in [s_{i-1}, s_i]$,

$$\lambda(x_{k'}, y_{j'}, t) = z_{k',j',i-1} + \frac{t - s_{i-1}}{s_i - s_{i-1}} \left(z_{k',j',i} - z_{k',j',i-1}\right).$$

Therefore, we have now established that $\lambda(x,y,t)$ is a closed-form linear function of the intensities at all the sub-interval and sub-space vertices, namely $\{z_{k,j,i} : \text{ all } k, j, i\}$.

3.2 Simulation Procedure

In this section, we present the simulation procedure of our model. Due to the page limit, we omit some of the detailed derivations as well as closed-form representations for coefficients of Equation (A.1), Equation (A.2) and Equation (A.3). These details and formulas can be found in the supplement document.

In order to simulate the spatio-temporal Poisson processes with a piecewise linear intensity function that varies in both time and location, we first simulate the time epoch of each arrival, immediately followed by the corresponding spatial location before the simulation of the next arrival. This approach is referred to as *projection* (Saltzman et al. 2012). Two results for high-dimensional Poisson processes in Proposition 1 and 2 provide theory support, whose proofs could be referred in Saltzman et al. (2012). See also Proposition 4.3.1 and Proposition 4.10.1 of Resnick (1992) for further understandings of these propositions of Poisson processes.

Proposition 1 Let N be a d-dimensional Poisson process defined on $(a_1,b_1]\times\cdots\times(a_d,b_d]$ with intensity function $\lambda(z_1,z_2,\ldots,z_{d-1},t)$. Let N_1 be a sorted process that records the last coordinate of the realization of N, then N_1 is a Poisson process supported on $(a_d,b_d]$ with intensity function

$$m_1(t)=\int_{z\in(a_1,b_1]\times\cdots\times(a_{d-1},b_{d-1}]}\lambda(z,t)\,dz.$$

Proposition 2 Conditional on the last coordinate of an arrival epoch being t, the distribution for the other coordinates has the probability density function given by

$$\lambda(z_1,z_2,\ldots,z_{d-1},t)/m_1(t).$$

In our spatio-temporal setting, we treat the "last coordinate" described in Proposition 1 and Proposition 2 as time, and "the other coordinates" as the two coordinates indicating the spatial location of an arrival. The first step is to sequentially generate the time coordinates of arrivals based on the inversion method. The inversion method is based on the integrated intensity function as illustrated by Proposition 1 and the associated cumulative integrated intensity. Specifically, the integrated intensity function denoted by $m(t)$ is obtained by integrating out the space dimensions of the underlying intensity,

$$m(t)=\iint_S\lambda(x,y,t)\mathrm{d}x\mathrm{d}y=\sum_{l=1}^{p}(\mathfrak{a}_{0,l}+\mathfrak{a}_{1,l}t)\,\mathbb{I}\{t\in(s_{l-1},s_l]\}\ \text{ for each } t\in[0,T] \tag{A.1}$$

where $\mathfrak{a}_{0,l}$'s and $\mathfrak{a}_{1,l}$'s are closed-form functions of $X=(x_0,x_1,\ldots,x_K)$, $Y=(y_0,y_1,\ldots,y_J)$ and the $\{z_{k,j,i}:\text{ all } k,j,i\}$. The detailed proof and the closed-form representation for the coefficients of Equation (A.1) are in the supplement document.

Proposition 3 The integrated intensity function $m(t)$ is piecewise linear in time for $t\in[0,T]$, as is shown in Equation (A.1).

Recall that the inversion method heavily exploits the structure of the cumulative intensity function (Klein and Roberts 1984). Using Proposition 3, the cumulative intensity function adopts the form of a piecewise quadratic function, which allows closed-form inversion of the cumulative intensity function. The details will be further discussed in Section 4. The inversion method permits sequential generation of inter-arrival times between consecutive arrivals.

For $i=1,2,\ldots,$ upon the generation of the i-th arrival ordered in time, say, with arrival time t_i, the specific location is immediately sampled according to a conditional p.d.f. given t_i. In fact, from Proposition 2, the location of an arrival occurring at t_i is a random vector in space, independent of other arrivals, whose conditional p.d.f. is given by $p(x,y;t_i)=\lambda(x,y,t_i)/m(t_i)$.

Proposition 4 For any given t, $\lambda(x,y,t)/m(t)$ as a function of location (x,y) is piecewise linear in the space S. Specifically, if we denote $\mathscr{T}_{k,j}^{\text{lower}}$ and $\mathscr{T}_{k,j}^{\text{upper}}$ as the lower and upper right triangle respectively in

the sub-rectangle $[x_{k-1},x_k]\times[y_{j-1},y_j]$, the function is linear in each triangular sub-space and continuous on the sub-space boundaries. The closed-form representation for the coefficients of $p(x,y;t)$ are shown in the supplement document.

The second step is then to simulate the location random vector of an arrival conditional on its arrival time t. As this random vector (x,y) adopts a piecewise linear density function in space, the marginal cumulative distribution function (c.d.f.) of each spatial coordinate is given by a piecewise cubic function. A proof can be found in the supplement document. Although the inversion is harder than that of a piecewise quadratic function, the closed-form inversion is available for the piecewise cubic function, permitting the efficient use of the inversion method. In order to simulate the location random vector (x,y), the y-coordinate is first simulated followed by the simulation of the x-coordinate. To simulate the y-coordinate, the conditional marginal p.d.f. of y given t is needed,

$$h(y;t)=\int_{x_0}^{x_K}p(x,y;t)\mathrm{d}x=\sum_{l=1}^{J}\left(\mathfrak{c}_{0,l;t}+\mathfrak{c}_{1,l;t}y+\mathfrak{c}_{2,l;t}y^2\right)\mathbb{I}\{y\in(y_{l-1},y_l]\}\ \text{ for each } y\in[y_0,y_J] \tag{A.2}$$

which is a continuous piecewise quadratic function of y. The coefficients $\mathfrak{c}_{0,l;t}$, $\mathfrak{c}_{1,l;t}$ and $\mathfrak{c}_{2,l;t}$ are dependent on l and t. To use the inversion method to generate the y-coordinate, the cumulative distribution function, denoted by $H(\cdot;t)$, is needed. The c.d.f. $H(\cdot;t)$ is a continuous, piecewise cubic function, and the difficulty of the inversion method lies in solving a cubic equation and identifying which one of the three roots gives the valid y-coordinate quantity. The closed-form expression of the equation is determined by finding $j^*\in\{1,2,\ldots,J\}$ that satisfies $H(y_{j^*-1};t)<U\le H(y_{j^*};t)$, where $U\sim\text{Uniform}(0,1)$. Among the three roots, there exists one and only one real root that is the valid y-coordinate because of the monotone property of the c.d.f. Therefore, after solving the cubic equation when applying the inversion method, the appropriate y-coordinate, denoted as y^* is found by checking all three roots one after another to verify whether $y_{j^*-1}<y^*\le y_{j^*}$. The detailed description is in Algorithm 2.

Once the y-coordinate is simulated, the conditional distribution for the x-coordinate given t and y is computed by the inversion method to generate the x-coordinate. The density function for this conditional distribution is piecewise linear. The partition of sub-intervals is related to y. Suppose $y_{j^*-1}<y\le y_{j^*}$. In each initial sub-interval $[x_k,x_{k+1}]$, a new point $x'_k=\frac{y-y_{j^*-1}}{y_{j^*-1}-y_{j^*}}(x_k-x_{k-1})+x_k$ is inserted to compose a new partition of the x-dimension, denoted as $V=(v_0,v_1,\ldots,v_{2K})$, where $v_{2k}=x_k$ and $v_{2k-1}=x'_k$. The conditional p.d.f. of x given t and y is

$$q(x;y,t)=p(x,y;t)/h(y;t)=\sum_{l=1}^{2K}\left(\mathfrak{d}_{0,l;y,t}+\mathfrak{d}_{1,l;y,t}x\right)\mathbb{I}\{x\in(v_{l-1},v_l]\}\ \text{ for each } x\in[x_0,x_K] \tag{A.3}$$

where $\mathfrak{d}_{0,l;y,t}$'s and $\mathfrak{d}_{1,l;y,t}$'s are dependent on y, t and l. The last coordinate x^* is generated by the inversion method, with more details described in Algorithm 3 and supplement document.

In summary, the simulation procedure first sequentially generates arrivals in their time order. Upon the generation of each arrival time, the simulation procedure generates the two-dimensional location information one coordinate after the other. The inversion method with closed-form computations is used in the generation of all time and location coordinates.

4 ALGORITHM

In this section, implementation details for the simulation algorithm are described. Recall that the first step in the simulation procedure is to sequentially simulate the time epochs of arrivals according to the integrated intensity

$$m(t)=\iint_S\lambda(x,y,t)\mathrm{d}x\mathrm{d}y=\sum_{l=1}^{p}(\mathfrak{a}_{0,l}+\mathfrak{a}_{1,l}t)\mathbb{I}\{t\in(s_{l-1},s_l]\}\ \text{ for each } t\in[0,T],$$

which is a piecewise linear function in time. The inversion method is applied to sequentially generate the inter-arrival times. Since the intensity function is piecewise linear, the corresponding cumulative intensity function is piecewise quadratic.

Specifically, conditional on the $(i-1)$st arrival time t_{i-1}, the next arrival time t_i as a random variable satisfies $\int_{t_{i-1}}^{t_i} m(t)\,dt \overset{\mathscr{D}}{=} -\ln(1-U)$, where $\overset{\mathscr{D}}{=}$ denotes equality in distribution, and U represents a uniform random variable supported on $[0,1]$.

The subtlety of using the inversion method to simulate the next arrival time t_i is caused by the sub-interval boundaries. That is, t_i and t_{i-1} may be on different time sub-intervals, where the slopes of the linear intensities are different. As introduced in Klein and Roberts (1984), to resolve this subtlety, an auxiliary variable u_k is introduced to represent the distance between t_{i-1} and the endpoint of the present sub-interval s_k. The specific expression of u_k is shown in the Algorithm 1. If the realization of the uniform random variable U used in variate generation, denoted as u, satisfies $u \le u_k$, the next arrival time $t_i \le s_k$, while if $u > u_k$, $t_i > s_k$. For $t_i \le s_k$, t_i remains in the same sub-interval as t_{i-1} and is derived by solving a quadratic equation. Note that the parameters in the quadratic equation, $\mathfrak{a}_{1,l}$'s and $\mathfrak{a}_{0,l}$'s, have already been evaluated in $m(t)$, permitting a closed-form solution of the quadratic equation using the quadratic formula. The detailed discussion of this equation is in the supplement document. For $t_i > s_k$, t_i goes beyond the current sub-interval where t_{i-1} is in. Then, a new loop of the iterations begins by setting $t_{i-1} = s_k$ and $u = (u-u_k)/(1-u_k)$. In the new loop, after the auxiliary variable u_{k+1} is obtained, u and u_{k+1} are compared to decide whether t_i exceeds the right-side boundary of the new sub-interval. The algorithm details are shown in Algorithm 1 for the generation of t_i based on t_{i-1}. The algorithm stops if the generation of next arrival time exceeds the overall time horizon T.

Algorithm 1 Generating arrivals in the time dimension

Input: The partition of time period, $0 = s_0 < s_1 <, \ldots, < s_p = T$; The sequence of slopes and intercepts of $m(t)$, $\mathfrak{a}_{1,l}$'s and $\mathfrak{a}_{0,l}$'s; The epoch of current arrival, t_{i-1}; The current index of the sub-interval, k;

Output: The occurrence time of the next arrival, t_i;

1: Obtaining $u \sim \text{Uniform}(0,1)$;
2: Compute $u_k = 1 - \exp\left[-(\mathfrak{a}_{1,k}/2)\left(s_k^2 - t_{i-1}^2\right) - \mathfrak{a}_{0,k}\left(s_k - t_{i-1}\right)\right]$;
3: **if** $u \le u_k$ **then**
4: **if** $a_k \ne 0$ **then**
5: $t_i = \left(-\mathfrak{a}_{0,k} + \sqrt{\mathfrak{a}_{0,k}^2 + \mathfrak{a}_{1,k}^2 t_{i-1}^2 + 2\mathfrak{a}_{1,k}\mathfrak{a}_{0,k} t_{i-1} - 2\mathfrak{a}_{1,k}\ln(1-u)}\right)/\mathfrak{a}_{1,k}$
6: **else**
7: $t_i = t_{i-1} - \ln(1-u)/\mathfrak{a}_{0,k}$
8: **end if**
9: **else**
10: $u = (u-u_k)/(1-u_k)$
11: $t_{i-1} = s_k$
12: $k = k+1$
13: **end if**
14: Go back to step 2;
15: **return** t_i;

Conditional on the the arrival time t of any simulated arrival, recall that the location of this arrival is a random vector that has probability density given by $p(x,y;t) = \lambda(x,y,t)/m(t)$ at location (x,y). The algorithm first generates the y-coordinate, described in Algorithm 2.

Once the y-coordinate of the arrival is simulated, the algorithm then simulates the random variable representing the x-coordinate. The conditional density of this random variable given t and y adopts a piecewise linear form, with $2K$ pieces. The boundaries of these $2K$ pieces have closed-form representations.

Algorithm 2 Simulating the spatial coordinate y^* given the arrival time t

Input: The joint p.d.f. $p(x,y;t)$ of x and y conditional on t;

Output: The y-coordinate of the given epoch, y^*;

1: Obtain the piecewise quadratic probability density function of y,

$$h(y;t)=\int_{x_0}^{x_K} p(x,y;t)\mathrm{d}x=\sum_{l=1}^{J}\left(\mathfrak{c}_{0,l;t}+\mathfrak{c}_{1,l;t}y+\mathfrak{c}_{2,l;t}y^2\right)\mathbb{I}\{y\in(y_{l-1},y_l]\}\ \text{ for each } y\in[y_0,y_J];$$

2: $C_1=-(\frac{1}{3}\mathfrak{c}_{2,1;t}y_0^3+\frac{1}{2}\mathfrak{c}_{1,1;t}y_0^2+\mathfrak{c}_{0,1;t}y_0)$;

3: **for** each $l\in\{1,\ldots,J-1\}$ **do**

4: $\quad C_{l+1}=\frac{1}{3}(\mathfrak{c}_{2,l;t}-\mathfrak{c}_{2,l+1;t})y_l^3+\frac{1}{2}(\mathfrak{c}_{1,l;t}-\mathfrak{c}_{1,l+1;t})y_l^2+(\mathfrak{c}_{0,l;t}-\mathfrak{c}_{0,l+1;t})y_l+C_l$;

5: **end for**

6: Obtain $u\sim\text{Uniform}(0,1)$;

7: $j^*=1$;

8: **while** $u\geq\frac{1}{3}\mathfrak{c}_{2,j^*;t}y_{j^*}^3+\frac{1}{2}\mathfrak{c}_{1,j^*;t}y_{j^*}^2+\mathfrak{c}_{0,j^*;t}y_{j^*}+C_{j^*}$ **do**

9: $\quad j^*=j^*+1$;

10: **end while**

11: Solve the cubic equation:

$$\frac{1}{3}\mathfrak{c}_{2,j^*;t}y^3+\frac{1}{2}\mathfrak{c}_{1,j^*;t}y^2+\mathfrak{c}_{0,j^*;t}y+C_{j^*}-u=0$$

12: Determine the only root y^* that satisfies the requirement among the three roots by checking

$$y_{j^*-1}<y^*\leq y_{j^*}$$

13: **return** y^*;

Therefore, the simulation of the x-coordinate also adopts the use of the inversion method, based on a piecewise quadratic cumulative distribution function. The detailed algorithm to simulate the x-coordinate is described in Algorithm 3. Now with Algorithm 1, Algorithm 2 and Algorithm 3 in hand, the arrivals are simulated sequentially in time and the location of each arrival is immediately simulated before the simulation of next arrival. The algorithm for the complete simulation procedure is shown in the supplement document. We have written a Python code implementation that is available online, see here.

5 NUMERICAL EXAMPLE

In this section, we implement our simulation algorithm with a simple given piecewise linear intensity function as an illustration. Consider the space $S=[0.5,5]\times[1,6]$ and the time horizon $[0,3]$. Adopting the notation in Section 3, the partition of the space is given by $X=(x_0,x_1,x_2)=(0.5,2,5)$, $Y=(y_0,y_1,y_2)=(1,3,6)$. The partition of time is $t_0=0,t_1=1,t_2=3$. The intensity value $z_{k,j,i}$ at each vertex (x_k,y_j,s_i) is

$$z_{0,0,0}=5,z_{0,1,0}=1,z_{0,2,0}=1,z_{1,0,0}=2,z_{1,1,0}=30,z_{1,2,0}=2,z_{2,0,0}=0.2,z_{2,1,0}=0.3,z_{2,2,0}=0.1$$
$$z_{0,0,1}=1,z_{0,1,1}=3,z_{0,2,1}=12,z_{1,0,1}=1,z_{1,1,1}=1,z_{1,2,1}=0.1,z_{2,0,1}=1,z_{2,1,1}=3,z_{2,2,1}=2$$
$$z_{0,0,2}=20,z_{0,1,2}=1.2,z_{0,2,2}=1,z_{1,0,2}=1.8,z_{1,1,2}=9,z_{1,2,2}=1,z_{2,0,2}=2,z_{2,1,2}=100,z_{2,2,2}=1.$$

Once the intensity value at each vertex is given, the piecewise linear intensity function at any time and location is fully specified by Equation (1) and Equation (2). The simulation procedure is implemented to generate $N=1000$ independent realizations of the process, including a total number of arrivals at the level of 600000.

Algorithm 3 Simulating the spatial coordinate x^* given t and y

Input: The joint p.d.f. $p(x,y;t)$ of x and y conditional on t; The y-coordinate of the given arrival and the associated sub-interval, $y \in [y_{j^*-1}, y_{j^*}]$; The marginal p.d.f. $h(y;t)$ of y given t;

Output: The x-coordinate of the given arrival, x^*;

1: Obtain the new partition for the x-dimension based on y:

$$[x_0, x_1'], [x_1', x_1], [x_1, x_2'], \cdots, [x_{K-1}, x_K'], [x_K', x_K]$$

where $x_k' = \frac{y - y_{j^*-1}}{y_{j^*-1} - y_{j^*}}(x_k - x_{k-1}) + x_k$, and renaming the partition as $[v_0, v_1], \cdots, [v_{2K-1}, v_{2K}]$ for convenience;

2: Obtain the piecewise linear probability density function conditional on y:

$$q(x;y,t) = p(x,y;t)/h(y;t) = \sum_{l=1}^{2K} \left(\mathfrak{d}_{0,l;y,t} + \mathfrak{d}_{1,l;y,t} x\right) \mathbb{I}\{x \in (v_{l-1}, v_l]\} \text{ for each } x \in [x_0, x_K];$$

3: $D_1 = -(\frac{1}{2}\mathfrak{d}_{1,1;y,t} x_0^2 + \mathfrak{d}_{0,1;y,t} x_0)$

4: **for** each $l \in \{1, \ldots, 2K-1\}$ **do**

5: $\quad D_{l+1} = \frac{1}{2}(\mathfrak{d}_{1,l;y,t} - \mathfrak{d}_{1,l+1;y,t}) v_l^2 + (\mathfrak{d}_{0,l;y,t} - \mathfrak{d}_{0,l+1;y,t}) v_l + D_l$

6: **end for**

7: Obtain $u \sim \text{Uniform}(0,1)$;

8: $k^* = 1$

9: **while** $u \geq G(v_{k^*}) = \frac{1}{2}\mathfrak{d}_{1,k^*;y,t} v_{k^*}^2 + \mathfrak{d}_{0,k^*;y,t} v_{k^*} + D_{k^*}$ **do**

10: $\quad k^* = k^* + 1$;

11: **end while**

12: Solve the quadratic equation

$$\frac{1}{2}\mathfrak{d}_{1,k^*;y,t} x^2 + \mathfrak{d}_{0,k^*;y,t} x + D_{k^*} - u = 0;$$

13: Determine the only root x^* that satisfies the requirement among the two roots by checking

$$x_{k^*-1} < x^* \leq x_{k^*};$$

14: **return** x^*

In the second example, the space is $S = [1,5] \times [2,9]$ and the time interval is $[0,3]$. The partition of space is $X = (x_0, x_1, x_2) = (1,3,5), Y = (y_0, y_1, y_2) = (2,3,9)$ and the partition of time is $t_0 = 0, t_1 = 0.5, t_2 = 1, t_3 = 1.5, t_4 = 2, t_5 = 3$. The intensity value at each vertex (x_k, y_j, s_i), $z_{k,j,i}$ is

$$z_{0,0,0} = 5, z_{0,1,0} = 1, z_{0,2,0} = 1, z_{1,0,0} = 2, z_{1,1,0} = 0, z_{1,2,0} = 2, z_{2,0,0} = 0.2, z_{2,1,0} = 0.3, z_{2,2,0} = 10$$
$$z_{0,0,1} = 1, z_{0,1,1} = 3, z_{0,2,1} = 60, z_{1,0,1} = 1, z_{1,1,1} = 1, z_{1,2,1} = 0.1, z_{2,0,1} = 1, z_{2,1,1} = 3, z_{2,2,1} = 2$$
$$z_{0,0,2} = 0, z_{0,1,2} = 1.2, z_{0,2,2} = 1, z_{1,0,2} = 1.8, z_{1,1,2} = 9, z_{1,2,2} = 1, z_{2,0,2} = 2, z_{2,1,2} = 1, z_{2,2,2} = 1$$
$$z_{0,0,3} = 5, z_{0,1,3} = 1, z_{0,2,3} = 1, z_{1,0,3} = 2, z_{1,1,3} = 30, z_{1,2,3} = 2, z_{2,0,3} = 2, z_{2,1,3} = 3, z_{2,2,3} = 0.1$$
$$z_{0,0,4} = 1, z_{0,1,4} = 3, z_{0,2,4} = 1, z_{1,0,4} = 1, z_{1,1,4} = 1, z_{1,2,4} = 0.1, z_{2,0,4} = 1, z_{2,1,4} = 3, z_{2,2,4} = 2$$
$$z_{0,0,5} = 200, z_{0,1,5} = 1.2, z_{0,2,5} = 1, z_{1,0,5} = 1.8, z_{1,1,5} = 9, z_{1,2,5} = 1, z_{2,0,5} = 2, z_{2,1,5} = 1, z_{2,2,5} = 1.$$

For the second experiment, $N = 1000$ independent realizations of the process are generated, including a total number of arrivals at the level of 400000.

Denote the cumulative intensity over time horizon $[0,T]$ as $\Lambda(x,y) = \int_0^T \lambda(x,y,t)\,dt$ and recall that the integrated intensity over space is given by $m(t) = \iint_S \lambda(x,y,t)\,dx\,dy$. For both experiments, we plot in Figure 1 and Figure 5 the true cumulative intensity $\Lambda(x,y)$ and in Figure 2 and Figure 6 the histogram generated by simulated samples. Specifically, this histogram records the spatial distribution of all simulated arrivals aggregated over time. We also plot in Figure 3 and Figure 7 the integrated intensity $m(t)$ and in Figure 4 and Figure 8 the histogram generated by simulated samples recording the temporal distribution of all simulated arrivals aggregating over space.

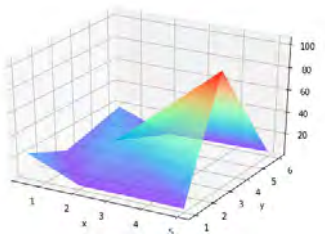

Figure 1: The plot of $\Lambda(x,y)$ for the first experiment in the space dimension.

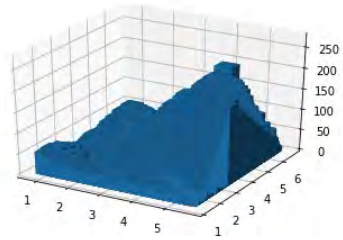

Figure 2: Histogram for the first experiment in the space dimension.

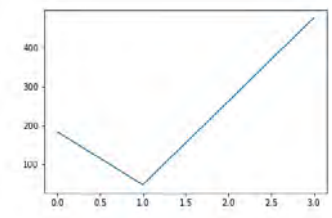

Figure 3: The plot of $m(t)$ for the first experiment in the time dimension.

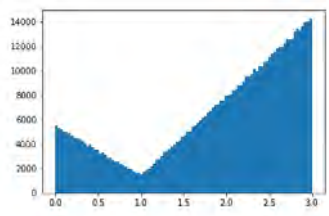

Figure 4: Histogram for the first experiment in the time dimension.

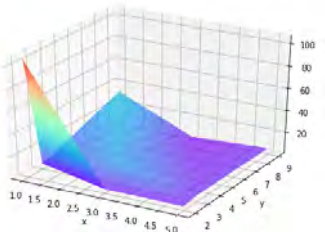

Figure 5: The plot of $\Lambda(x,y)$ for the second experiment in the space dimension.

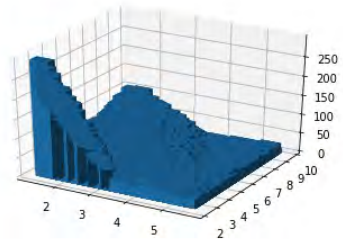

Figure 6: Histogram for the second experiment in the space dimension.

In comparison, we also implement a thinning method to generate the same arrival processes in both experiments by first generating a spatial-temporal Poisson process with a constant intensity function. The constant is equal to the maximum value of the given nonstationary intensity model. Then, each arrival is accepted with probability proportional to the intensity value at that arrival point.

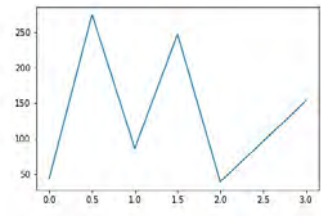

Figure 7: The plot of $m(t)$ for the second experiment in the time dimension.

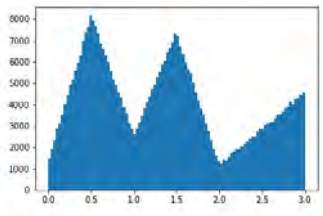

Figure 8: Histogram for the second experiment in the time dimension.

We record the computation time of both the inversion method (our approach) and the thinning method. All execution times are based on running Python on a 2.6 GHz Intel Core i5 MacBook Air. We find that the computer time per 10000 arrivals using the inversion method is around 3 seconds. The computer time per 10000 arrivals using the thinning method is around $0.2/r$ seconds in which r is the acceptance rate. In the first experiment, the acceptance rate r is 9.68%, and in the second experiment, r is around 2.6%. In conclusion, when the intensity function is relatively stationary in time and space, which indicates a close-to-one value of r, the thinning method performs better than the inversion method on efficiency. When the intensity function changes significantly in either time or location, indicating a small r, the inversion method (our approach) has advantage in terms of sample efficiency. In the application of taxi and ride-sharing services, for example, the arrival intensity of the peak hours can be significantly larger than the intensity of the off-peak hours. The arrival intensity at popular locations (airports, stadiums, etc.) can be much higher than the intensity at regular residence neighbors. Due to the significant difference between the peak and average intensities, the associated acceptance rate r could be even smaller than 0.1%. Our simulation algorithm may be more advantageous in these scenarios.

6 CONCLUSION AND FUTURE WORK

In this work, we propose an inversion-method based simulation procedure for a class of nonstationary spatio-temporal Poisson processes. When the Poisson intensity function is specified by a piecewise linear function in both space and time, the proposed simulation procedure involves only closed-form computation with no need for numerical integration of numerical inversion of any function, therefore rendering the efficiency. For future work, we are working on efficient estimation procedures and their statistical guarantees. Other potential future work include developing efficient space partition procedures, generalizing the two-dimensional space into a multi-dimensional space to incorporate additional features that are attached to each arrival, and extending the Poisson process model by adding non-Poisson features such as doubly stochasticity and self-exciting dynamics.

7 ACKNOWLEDGEMENT

We are grateful to the reviewers for their insightful suggestions and comments, which have significantly improved our paper.

REFERENCES

Achcar, J. A., E. Z. Martinez, A. D. P. d. Souza, V. M. Tachibana, and E. F. Flores. 2011. "Use of Poisson Spatiotemporal Regression Models for the Brazilian Amazon Forest: Malaria Count Data". *Revista da Sociedade Brasileira de Medicina Tropical* 44:749 – 754.

Babu, G. J., and E. D. Feigelson. 1996. "Spatial Point Processes in Astronomy". *Journal of Statistical Planning and Inference* 50(3):311 – 326.

Bratley, P., B. L. Fox, and L. E. Schrage. 1987. *A Guide to Simulation*. New York: Springer.

Çinlar, E. 2013. *Introduction to Stochastic Processes*. Mineola, New York: Dover.

Chen, H., and B. W. Schmeiser. 1992. "Simulation of Poisson Processes with Trigonometric Rates". In *Proceedings of the 1992 Winter Simulation Conference*, edited by J. J. Swain, D. Goldsman, R. Crain, and J. R. Wilson, 609–617. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Haenggi, M. 2012. *Stochastic Geometry for Wireless Networks*. Cambridge: Cambridge University Press.

Harrod, S., and W. D. Kelton. 2006. "Numerical Methods for Realizing Nonstationary Poisson Processes with Piecewise-constant Instantaneous-rate Functions". *Simulation* 82(no.3):147–157.

Heyman, D. P., and M. J. Sobel. 2004. *Stochastic Models in Operations Research: Stochastic Optimization*, Volume 2. New York: Dover.

Klein, R. W., and S. D. Roberts. 1984. "A Time-varying Poisson Arrival Process Generator". *Simulation* 43(no.4):193—-195.

Lee, S., J. R. Wilson, and M. M. Crawford. 1991. "Modeling and Simulation of a Nonhomogeneous Poisson Process Having Cyclic Behavior". *Communications in Statistics - Simulation and Computation* 20(2-3):777–809.

Lewis, P. A., and G. S. Shedler. 1976. "Simulation of Nonhomogeneous Poisson Processes with Log Linear Rate Function". *Biometrika* 63(3):501–505.

Lewis, P. A., and G. S. Shedler. 1979a. "Simulation of Nonhomogeneous Poisson Processes by Thinning". *Naval Research Logistics Quarterly* 26(3):403–413.

Lewis, P. A., and G. S. Shedler. 1979b. "Simulation of Nonhomogeneous Poisson Processes with Degree-two Exponential Polynomial Rate Function". *Operations Research* 27(5):1026–1040.

Liu, R., M. E. Kuhl, Y. Liu, and J. R. Wilson. 2019. "Modeling and Simulation of Nonstationary Non-Poisson Arrival Processes". *INFORMS Journal on Computing* 31(2):347–366.

Mohler, G. O., M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita. 2011. "Self-exciting Point Process Modeling of Crime". *Journal of the American Statistical Association* 106(493):100–108.

Ogata, Y., and J. Zhuang. 2006. "Space–time ETAS Models and an Improved Extension". *Tectonophysics* 413(1):13 – 23.

Othmer, H. G., S. R. Dunbar, and W. Alt. 1988. "Models of Dispersal in Biological Systems". *Journal of Mathematical Biology* 26(3):263–298.

Pasupathy, R. 2011. "Generating Nonhomogenous Poisson Processes". In *Wiley Encyclopedia of Operations Research and Management Science*. Hoboken: Wiley.

Resnick, S. I. 1992. *Adventures in Stochastic Processes*. Boston: Birkhäuser.

Ross, S. 2013. "Chapter 7 - The Discrete Event Simulation Approach". In *Simulation* (5th ed.)., edited by S. Ross, 111 – 134. Cambridge: Academic Press.

Saltzman, E. A., J. H. Drew, L. M. Leemis, and S. G. Henderson. 2012. "Simulating Multivariate Nonhomogeneous Poisson Processes Using Projections". *ACM Transactions on Modeling and Computer Simulation* 22(3):1–13.

Zheng, Z., and P. W. Glynn. 2017. "Fitting Continuous Piecewise Linear Poisson Intensities via Maximum Likelihood and Least Squares". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 1740–1749. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zhou, Z., D. S. Matteson, D. B. Woodard, S. G. Henderson, and A. C. Micheas. 2015. "A Spatio-temporal Point Process Model for Ambulance Demand". *Journal of the American Statistical Association* 110(509):6–15.

AUTHOR BIOGRAPHIES

HAOTING ZHANG is an M.S. student in the Department of Industrial Engineering & Operations Research at the University of California Berkeley. He has research interests in simulation, stochastic modeling, and data analytics. His email address is haoting_zhang@berkeley.edu.

ZEYU ZHENG is an assistant professor in the Department of Industrial Engineering & Operations Research at the University of California Berkeley. He received his Ph.D. in Management Science and Engineering, Ph.D. minor in Statistics and M.A. in economics from Stanford University, and a B.S. in Mathematics from Peking University. He has done research in simulation, stochastic modeling, data analytics, statistical learning, and over-the-counter financial markets. His email address is zyzheng@berkeley.edu.