CORSSA

Theme V - Models and Techniques for Analyzing Seismicity

Stochastic simulation of earthquake catalogs

Jiancang Zhuang [1] · **Sarah Touati**[2]

1. Institute of Statistical Mathematics
2. School of GeoSciences, University of Edinburgh

How to cite this article:

Zhuang, J. & S. Touati (2015), Stochastic simulation of earthquake catalogs, Community Online Resource for Statistical Seismicity Analysis, doi:10.5078/corssa-43806322. Available at http://www.corssa.org

Document Information:
Issue date: 19 March 2015 Version: 1.0

Contents

Abstract Starting from basic simulation procedures for random variables, this article presents the theories and techniques related to the simulation of general point process models that are specified by the conditional intensity. In particular, we focus on the simulation of point process models for quantifying the characteristics of the occurrence processes of earthquakes, including the Poisson model (homogeneous or nonhomogeneous), the recurrence (renewal) models, the stress release model and the ETAS models.

1 Introduction and motivation

This article will describe the procedure of simulation: that is, creating a 'synthetic' earthquake catalog (or parts of the data for such a catalog) based on some model of how the earthquake process works. A simulation provides a concrete example of the abstract process that the model is describing, which may be designed and calibrated to represent a particular real catalog, or may be a theoretical model set up for some other purpose. This example of the process—the synthetic catalog—is referred to as a 'realisation' of the model.

In abstract terms, a real earthquake catalog may be considered a realisation of the underlying stochastic processes governing earthquake occurrence. We do not know for certain what those are, but we build models to represent our reasoned hypotheses, knowing that there are fundamental limitations to their ability to reflect real processes, but knowing that they can still be useful in explaining earthquake patterns and making predictions despite these limitations. The process of going from a data set to the model parameter values that best explain it is referred to as 'inference.' Simulation is conceptually the opposite of that process: it takes a parameterised model and produces a data set (or several different data sets) consistent with that model.

Simulation has become a widely used tool for research, as computing power has increased to allow extensive simulations to be carried out. In many cases having a realisation—or many realisations—of a model makes it considerably easier to analyze the properties of the model, such as the uncertainty in an event rate predicted by the model, or the dependence of some quantity on the model parameters. Rather than only treating the model analytically as a set of equations, running simulations allows us to explore the model "experimentally," and, provided the simulation is done correctly, it can quickly and easily answer questions that an analytical treatment alone cannot.

Simulation has many uses; perhaps the most obvious application is in attempting to reproduce the earthquake occurrence pattern of a particular catalog: synthetic versions of a catalog can be compared with the real one visually (through plotting graphs) and mathematically (through calculation of various properties), to assess how well the model describes the data. Another important application is in understanding the properties of the model itself, and their sensitivity to input parameter values; or testing the effectiveness of techniques for estimating those parameters—the benefit of testing on a synthetic catalog being that you know what the true parameter values are. Both of these applications can involve 'bootstrapping': running lots of simulations (that differ only in terms of each being a different realisation of the same random process) to determine some average properties as well as confidence limits for those properties.

You may also glean new insights into your catalog through looking at a synthetic one, assuming those insights are applicable back to the real data. For example, when simulating a model such as ETAS (see, e.g., *Zhuang et al.* 2012, for the definition and properties of the ETAS model) where events can trigger other events (aftershocks), it is possible to keep track of the 'family' structure of the data set (in the sense of 'parent' events triggering 'offspring' aftershocks)—something we can never be certain of in real data. This allowed, for example, the discovery of the origin of the bimodality (two peaks) in the distribution of inter-event times: as shown in *Touati et al.* (2009), the two peaks arise from (1) pairs of events that both belong to the same aftershock sequence, and (2) pairs of events that are not related to each other. Depending on the relative sizes of these two contributions, a range of shapes for the overall distribution can be observed, but the underlying reason for these shapes is quite simple and was discovered by looking at synthetic catalogs.

A more important usage of simulation is forecasting. Suppose that you have constructed a reasonably good model for describing the occurrence process of earthquakes. With the observation of earthquakes and other related geophysical or geochemical observations, you can use the model to simulate the seismicity in a certain future period, for example, tomorrow or next month. Such a simulation can be done many times, and the probability that a large earthquake will occur in a given space-time window can be approximated by the proportion of simulations with such an event happening.

1.1 Starting point

You will perhaps have read some of the Theme V CORSSA articles on seismicity models, or otherwise be familiar with some of these models. You want to simulate some of these models, perhaps by using available software, and you need (or wish to have) a fuller understanding of how they work, from basic principles. Or perhaps you want to understand the principles of simulation in order to apply these methods to simulating a model of your own design.

1.2 How the article is structured

Section 2 introduces a couple of common basic methods for sampling from probability distributions, which are the building blocks for the methods in the rest of the article. Section 3 shows how these methods apply to temporal point processes, a class of models that encompasses a large number of the seismicity occurrence models in the literature. The remaining sections apply the principles learnt to various aspects of seismicity modelling: section 4 discusses simulation of common magnitude and moment distributions; section 5 applies the point process simulation methods to models such as the Epidemic-Type Aftershock Sequences (ETAS) and the Stress Release models (combining point process simulation with independent magnitude sampling to produce synthetic earthquake catalogs); section 6 deals with models in which the spatial aspect is dealt with by the partitioning of space into interacting sub-regions; and section 7 looks at models that include full spatial coordinates for events, including a challenging case where the probability distributions for the magnitude, location and occurrence time are not independent. Section 9 gives a brief look at some of the applications for the results of simulations, and section 10 notes some existing software that readers may wish to access and use.

1.3 A note on the use of symbols

We use the following rules of thumb for our choice of symbols in the text:

- Upper case letters are usually used to represent particular (definite) values of a random variable, while the corresponding lower case letters represent possible values of those variables
- x and y are used for generic variables
- x_1, x_2 and x_3 are also used for spatial coordinates
- r and R are used to represent distance (the variable and variate, respectively)
- R is also used in some instances to denote a particular region by a single symbol
- t is used to represent an instance in time, whereas τ is used to represent a time interval
- v is used as a dummy integration variable
- U represents a uniform random variate
- m is used for earthquake magnitude, whereas M is used for seismic moment

2 Basic simulation: simulating random variables

The essence of simulating from a stochastic model—that is, a model containing some randomness—is *sampling* from the probability distributions that describe that randomness. This gives a *realisation* of the random process. In this section we describe the two key methods for performing the sampling: the inversion method and the rejection method.

2.1 The inversion method

The inversion method, or 'inverse transform sampling,' is a common method usually used for simulating a continuous random variable. It involves firstly obtaining the cumulative distribution for the variable to be sampled: this is defined as the probability of that variable being less than or equal to a given value, and so it starts from 0 at the variable's lowest possible value and increases to 1 at the maximum possible value. It is obtained by integrating the probability density function (pdf) with respect to the variable you want to sample. Secondly, this cumulative distribution is rearranged to give an expression for the variable of interest in terms of its probability (rather than the other way around), and thirdly, the variable can then be sampled by inserting uniform random values between 0 and 1 into this expression in place of the cumulative probability.

More formally:

Algorithm 1 *Simulate a continuous random variable using the inversion method*

Suppose that X has a monotonically increasing cumulative distribution function $F(x)$.

Step 1. Obtain a random variate U, uniformly distributed in $[0, 1]$.

Step 2. Let $X = F^{-1}(U)$, where $F^{-1}(\cdot)$ is the inverse function corresponding to $F(\cdot)$, i.e., $F(X) = U$.

A proof of this is detailed in Appendix A.

The above algorithm can be modified to simulate *discrete* random variables as follows:

Algorithm 2 *Simulate a discrete random variable*

Suppose X is a random variate, with probability mass function $p(x_i)$, $i = 1, 2 \cdots$.

Step 1. Obtain a random variate U, uniformly distributed in $[0, 1]$.

Step 2. Let $X = x_j$, where j satisfies $\sum_{i=1}^{j} p_i(x_i) \leqslant U < \sum_{i=1}^{j+1} p(x_i)$.

Simple example: In a Poisson process, the time intervals between the events are all independent, and follow an exponential distribution (see *Zhuang et al.* 2012 and references therein). Simulating a Poisson process is therefore a matter of repeatedly drawing the time interval τ to the next event from an exponential distribution:

$$f(\tau) = \lambda \, \mathrm{e}^{-\lambda\tau} \tag{1}$$

where λ is the event rate of the Poisson process, and τ can only be positive. Conveniently, the technical name for this single parameter λ of an exponential distribution is the 'rate' parameter.

The cumulative distribution function is obtained by integration:

$$F(\tau) = \int_0^\tau f(v)\,\mathrm{d}v = 1 - \mathrm{e}^{-\lambda\tau} \tag{2}$$

We obtain the inverse, $F^{-1}(\cdot) = \tau$, by simply rearranging:

$$\tau = -\frac{1}{\lambda}\log[1 - F(\tau)].$$

Thus, to generate a random variate T from this exponential distribution with a rate parameter λ, simply let $T = -\frac{1}{\lambda}\log(1-U)$, where U is uniformly distributed in $[0,1]$ — This is equivalent to $T = -\frac{1}{\lambda}\log U$, since, if U is uniformly distributed in [0,1], so is $1-U$.

The above simple example serves to illustrate the procedure; in reality most software packages will have a convenient function to sample randomly from an exponential distribution, requiring you only to pass in the parameter λ and not to go through all the steps above for this simple case. In R, this function is `rexp`, and, in Matlab, `exprnd`. But the underlying code is usually an implementation of the inversion method as outlined here. When you need to sample non-standard distributions, you may need to implement the inversion method from scratch, following steps as shown above. The following longer example is a case such as this.

More complex example: The following probability density is often used in the simulation of the positions (X, Y) of the direct aftershocks by an ancestor event at (x_0, y_0) (see, e.g, *Console et al.* (2003)):

$$f_{XY}(x,y) = \frac{q-1}{\pi}\frac{d^{2(q-1)}}{[(x-x_0)^2+(y-y_0)^2+d^2]^q}, \quad q > 1, \tag{3}$$

where d and q are the constant parameters. To simplify the simulation, we use the polar coordinate instead, through the following transformation:

$$\begin{cases} X = x_0 + R\,\cos\varTheta \\ Y = y_0 + R\,\sin\varTheta. \end{cases} \tag{4}$$

The joint p.d.f. of R and $\varTheta$ is

$$\begin{aligned} f_{R\varTheta}(r,\theta) &= f_{XY}(x_0 + r\,\cos\theta, y_0 + r\,\sin\theta)\left|\begin{matrix} \frac{\partial X}{\partial R} & \frac{\partial X}{\partial \varTheta} \\ \frac{\partial Y}{\partial R} & \frac{\partial Y}{\partial \varTheta} \end{matrix}\right|_{R=r,\,\varTheta=\theta} \\ &= \frac{(q-1)\,d^{2(q-1)}}{\pi}\frac{r}{r^2+d^2}. \end{aligned}$$

where the appearance of the Jacobian $|\cdot|$ is due to coordinate transformation. $f_{R\Theta}(r,\theta)$ can be split into the product of two independent probability densities,

$$f_R(r) = (q-1)\, d^{2(q-1)} \frac{2r}{r^2+d^2}, \quad r \geq 0, \tag{5}$$

and

$$f_\Theta(\theta) = \frac{1}{2\pi}, \quad \theta \in [0, 2\pi].$$

In the latter, θ does not appear on the right-hand side at all, meaning that the aftershock distribution is isotropic around the triggering event. The form of $F_R(r)$ implies that the probability of aftershocks falls away with increasing distance r, as a power law.

To obtain the cumulative probability function of R, integrate (5) with respect to r:

$$\begin{aligned} F_R(r) &= (q-1) d^{2(q-1)} \int_0^r \frac{2u}{(u^2+d^2)^q}\, \mathrm{d}u \\ &= 1 - \left(1 + \frac{r^2}{d^2}\right)^{1-q} \end{aligned} \tag{6}$$

The next step is rearranging equation $1 - F_R(R) = U$ to get an expression for R:

$$R = d\sqrt{U^{\frac{1}{1-q}} - 1} \tag{7}$$

where U is a uniform-random number between 0 and 1. Θ can be generated as uniform random numbers in $[0,\ 2\pi]$. Using transform (4), we can obtain the required position of the aftershock.

2.2 The rejection method

The rejection method is useful for simulating a random variable, say x, which has a complicated density function $f(x)$ making it difficult to obtain the cumulative distribution by integration. Instead of generating X directly, we generate another random variate Y which has a simpler density function $g(x)$ such that $f(x) < B\, g(x)$, where B is a positive real number greater than 1. ($g(\cdot)$ may be sampled using the inversion method above.) Then X can be generated as follows:

Algorithm 3 *Simulate random variables using the rejection method*

```
Step 1. Generate a random variate $Y$ from density $g(\cdot)$ and a random variate $U$
    uniformly distributed in $[0,\ 1]$.
Step 2. If $U < \frac{f(Y)}{Bg(Y)}$, accept $X = Y$; otherwise, back to Step 1.
```

Thus we sample from $g(x)$, but insert the sampled values into $f(x)$ as a 'check.' With the method outlined above, at values of x where $f(x)$ is much smaller than $g(x)$, the probability of acceptance of variates generated from $g(x)$ is suitably decreased, such that the set of accepted variates is eventually distributed according to $f(x)$. A formal proof of this is detailed in Appendix B.

An advantage of the rejection method is that, even if we only know $f(x) = Cq(x)$, where $q(\cdot)$ is a known nonnegative function and C is an unknown positive constant, we can still simulate a random variate with a p.d.f. of $f(\cdot)$. This method is extremely useful for complex and high-dimensional cases, where the normalising factors for p.d.f.s are difficult to obtain. The following example deals with just such a case where $f(x) \propto g(x)$.

Illustrative example: Suppose we want to simulate a 3D random variable uniformly distributed in the unit ball within $\mathbb{R}^3$. The density function of the target random variable is:

$$f(x_1, x_2, x_3) = \begin{cases} 3/(4\pi), & x_1^2 + x_2^2 + x_3^2 \leq 1; \\ 0, & \text{otherwise.} \end{cases}$$

In words: the density function has a constant value $(3/(4\pi))$ within a sphere of radius 1 centred on the origin, and is 0 everywhere outside of this sphere.

To simulate from this distribution, we can first generate three 1D random variates, X_1, X_2 and X_3, all uniformly distributed in $[-1, 1]$. These samples effectively come from a distribution $g(x_1, x_2, x_3)$ that takes the constant value 1/8 when x_1, x_2, and x_3 are all in [-1,1], and 0 otherwise. This is also a uniform distribution, but is simpler than $f(\cdot)$ because the conditions on the values of x_1, x_2, and x_3 are expressed as three independent constraints, and thus these variables can be sampled separately, unlike $f(\cdot)$ which has one condition expressed in terms of a combination of the variables. It represents the smallest possible cube surrounding the unit sphere we want to sample from. It is clear from the outset that we should accept the sample if it lies within the sphere and reject it otherwise, i.e., accept it if $X_1^2 + X_2^2 + X_3^2 \leq 1$. This straightforward approach would solve the problem; however, we pursue the steps of the rejection method explicitly here for illustration purposes.

The value of $g(x)$ within the sphere $(= 1/8)$ is lower than that of $f(x)$ $(= 3/(4\pi))$, so we need to choose a value for the factor B that will make $B \times g(x)$ greater than $3/(4\pi)$—say, 3. We can then accept $X = (X_1, X_2, X_3)$ as a sample from $f(x)$ if $U < f(X_1, X_2, X_3)/[3 \times g(X_1, X_2, X_3)]$, having generated a random variate U uniformly in [0, 1]. This condition will always be false where $f(x) = 0$, i.e., outside the sphere, and thus points outside the sphere will always be rejected. Points inside the sphere will be accepted with probability $\frac{3}{4\pi}/(B \times \frac{1}{8})$ (with $B = 3$).

This simple example shows the effect of the value of B: the higher this value is, the more points will be rejected. In this particular example, we can see that it is not necessary to reject *any* of the points generated inside the sphere, as the probabilities $f(\cdot)$ and $g(\cdot)$ are both uniform. Indeed if we set B to be as low as $3/(4\pi) \times 8$, the probability of acceptance reaches 1 within the sphere, and we will no longer reject any points unnecessarily. (At this point, the generation of U becomes unnecessary altogether since $f(X_1, X_2, X_3)/[B \times g(X_1, X_2, X_3)]$ can only be 0 or 1—the acceptance or rejection of a given sample is now completely deterministic.) More generally, though, if $g(\cdot)$ is not proportional to $f(\cdot)$, it will not be possible to achieve this situation, but by choosing the lowest B we can find such that $B \times g(x)$ is always greater than $f(x)$, we will maximise the efficiency of the sampling for the given $g(x)$.

Another key observation here is that the choice of the 'envelope' density, $g(\cdot)$, is critical: if the shape of $g(\cdot)$ differs too much from $f(\cdot)$, then a large B will be required to ensure $f(x) < B\, g(x)$ for all x, and thus the overall fraction of samples being rejected will be high. An efficient choice for $g(\cdot)$ is not always easy to find, especially when $f(\cdot)$ is highly concentrated in a certain region or has a spike at some location. This problem can be solved using an adaptive extension, i.e., using a set of piecewise envelope distributions.

2.3 Alternatives

The inversion and rejection methods are two standard methods for generating random variates, but they are not always the easiest or quickest. For example, to generate a beta distribution with an integer degree of freedom (m, n), i.e., $f(x) = x^{m-1}(1-x)^{n-1}/\mathrm{B}(m, n)$, where $\mathrm{B}(m, n)$ is the beta function, it is better to use the connection between the beta distribution and the order statistics of the uniform distribution:

Algorithm 4 *Generate a random number from the Beta distribution*

```
Step 1. Generate $m + n - 1$ random variates $U_1, U_2, \cdots, U_{m+n-1}$.
Step 2. Sort $U_1, U_2, \cdots, U_{m+n-1}$ in ascending order, and denote the new sequence
    by $U_{(1)}, U_{(2)}, \cdots, U_{(m+n-1)}$.
step 3. Let $X = U_{(m)}$.
```

The specific simulation methods for many other basic probability distributions can be found in standard textbooks on basic probability theory, e.g., *Ross* (2003).

Having covered the two basic techniques for simulation now, the next section extends these methods into the realm of temporal point process models—a cornerstone of earthquake occurrence modelling.

3 More advanced simulations: simulating temporal point processes

The Poisson process, seen in the simple example of section 2.1, is the simplest of temporal point processes. In general, point processes are specified by their *conditional intensity* function $\lambda(t)$, which can be thought of as analagous to the rate parameter λ of the exponential distribution that describes time intervals between events in the Poisson process. The conditional intensity also plays the same role as the hazard function for estimating the risk of the next event.

In this section, we discuss simulation of temporal point process models, building on the methods introduced in the previous section.

3.1 Using the inversion method

The simple example in section 2.1 shows how to use the inversion method for simulating a Poisson process with rate λ through generating the waiting times between successive events. The inversion method can also be used for more general temporal point processes.

Recall the definitions of the cumulative probability distribution function, the probability density function, the survival function and the hazard function corresponding to the waiting time to the next event (from a certain time t) discussed in the CORSSA article by *Zhuang et al.* (2012), which are, respectively,

$$F_t(\tau) = \Pr\{\text{from } t \text{ onwards, waiting time to next event} \le \tau\ \}, \tag{8}$$

$$f_t(\tau)\,\mathrm{d}\tau = \Pr\{\text{from } t \text{ onwards, waiting time is between } \tau \text{ and } \tau + \mathrm{d}\tau\ \}, \tag{9}$$

$$\begin{aligned} S_t(\tau) &= \Pr\{\text{from } t \text{ onwards, waiting time} > \tau\ \} \\ &= \Pr\{\text{no event occurs between } t \text{ and } t+\tau\} \\ &= 1 - F_t(\tau), \end{aligned} \tag{10}$$

and

$$\begin{aligned} h_t(\tau)\,\mathrm{d}\tau = \Pr\{&\text{next event occurs between } t+\tau \text{ and } t+\tau+\mathrm{d}\tau \\ &\mid \text{it does not occur between } t \text{ and } t+\tau.\}. \end{aligned} \tag{11}$$

If any of F_t, S_t and f_t is given, the waiting time from t to the occurrence of the next event in the process could be simulated in the same way as was shown for the Poisson process. When only the conditional intensity $\lambda(t)$ of the model is provided, the simulation could be done by converting the conditional intensity to the probability distribution function F_t. Consider the equivalence between the conditional intensity and the hazard function, i.e., $\lambda(t+\tau) = h_t(\tau),\ \tau \geqslant 0$, and the relation between the hazard function and the probability distribution function

$$h_t(\tau) = -\frac{\mathrm{d}}{\mathrm{d}\tau}[\log(1 - F_t(\tau))]. \tag{12}$$

Then the distribution of waiting times could be solved:

$$F_t(\tau) = 1 - \exp\left[-\int_t^{t+\tau} \lambda(v)\,\mathrm{d}v\right].$$

The above equation enables us to use the inversion method to simulate temporal point processes. Compare with equation (2): the two are very similar, except that in this case, the rate is not a constant parameter but a more complex function of time that must be integrated.

Algorithm 5 *The inversion method for simulating a temporal point process*

Suppose that we are simulating a temporal point process N described by a conditional intensity $\lambda(t)$ in the time interval $[t_{start}, t_{end}]$ for all $t \in [t_{start}, t_{end}]$.

```
Step 1. Set $t = t_{start}$, and $i = 1$.
Step 2. Generate a random variate $U$ uniformly distributed in $[0, 1]$.
Step 3. Find $t'$ by rearranging $U = \exp\left[-\int_t^{t'} \lambda(v)\,\mathrm{d}v\right]$.
Step 4. If $x < t_{end}$, set $t_{i+1} = t'$, $t = t'$, and increment $i$ by 1; return to step 2. ($t$
    refers to the current time, whereas $t_i$ refers to the time of event $i$.)
Step 5. Output $\{t_1, t_2, \cdots, t_n\}$, where $t_n$ is the last generated event before $t_{end}$.
```

In most cases, t' in Step 3 may be hard or even impossible to solve analytically. To avoid this issue, the following thinning method is introduced.

3.2 Using the thinning method

It is also possible to apply an analogue of the rejection method (section 2.2) to temporal point processes. This is known as the thinning method. It works in a similar fashion to the simulation of a Poisson process, by generating waiting times sampled from an exponential distribution, as in the simple example in section 2.1 but with an extra 'rejection' step. For a point process the conditional intensity $\lambda(t)$—the equivalent of the constant rate parameter of the Poisson process—is now a function of time and of the history of the process, and so its value varies from moment to moment.

Algorithm 6 *The thinning method for simulating temporal point process (Ogata 1981)*

Suppose that we are simulating a temporal point process N described by a conditional intensity $\lambda(t)$ in the time interval $[t_{start}, t_{end}]$. Suppose we can find a positive number B such that $\lambda(t) \leqslant B$ for all t between t_{start} and the occurrence time of the next event.

Step 1. Set $t = t_{start}$, and $i = 0$.
Step 2. Simulate a waiting time, τ, according to a Poisson process of rate B, i.e., $\tau \sim Exp(B)$.
Step 3. Generate a random variate $U \sim Uniform(0, 1)$.
Step 4. If $U \leq \lambda(t + \tau)/B$, set $t_{i+1} = t + \tau$, $t = t + \tau$ and $i = i + 1$. Otherwise, only set set $t = t + \tau$.
Step 5. Return to step 2 until t exceeds t_{end}.
Step 6. Return $t_1, t_2, \cdots, t_n$.

In Step 2, in the case of a model with decreasing intensity after an event (such as ETAS, see, e.g., *Zhuang et al.* 2012 or Section 5.4 in this article, for the explanation of this model), B can be the conditional intensity at time t, $\lambda(t)$.

The actual distribution requiring to be sampled from at each time t is locally an exponential, but its rate, the conditional intensity, is non-stationary in time. Given that what we are sampling is time intervals, the non-stationarity cannot be ignored, and we cannot sample it directly but must instead sample from an envelope distribution and apply a rejection, with probability as calculated above. A formal proof of this algorithm is given in Appendix C.

The above thinning method thus has some resemblance to the rejection method (section 2.2) with an exponential 'envelope' distribution; however, there are some significant differences. First of all, the rejection method is used for simulating individual random variates, while the thinning method is used for simulating a process, i.e., a random finite subset of points on the target time interval, where the points in the subset can be either independent of or dependent on each other. When using the rejection method, once a sample is rejected, the simulation is started over from the beginning again. In the thinning method, the sampling of the next event starts from the location of the last event generated, no matter whether it is rejected or accepted.

The algorithm above is suitable for models built around the concept of aftershock triggering, such as ETAS (which implements the Omori law). The conditional intensity is decreasing or bounded between two neighbouring events, so it is quite easy to find a suitable value of B in step 2: we can use the value of the conditional intensity at the current event, knowing that the intensity will only decrease with time until another event occurs. We don't need to know in advance when the next event will occur. Having established a provisional time t_{i+1} for the next event by sampling from the envelope distribution, the conditional intensity at t_{i+1} is checked;

the new event is then either created or not created, with a probability that is the ratio of the later conditional intensity to the earlier one.

But for models in which $\lambda(t)$ increases boundlessly between events (such as SRM, see, e.g., *Zhuang et al.* 2012 or Section 5.3 for definition of this model), this algorithm is not easy to apply. The maximum value that the conditional intensity reaches before the next event depends on the precise timing of the next event, which is unknown in advance. However, if an upper bound for $\lambda(t)$ can be determined within some precise time interval $\varDelta$ (not necessarily the interval to the next event), this can be taken advantage of, as in the following algorithm.

Algorithm 7 *The modified thinning method (Daley and Vere-Jones 2003)*

Suppose that we simulate a temporal point process N equipped with a conditional intensity $\lambda(t)$ in the time interval $[t_{start}, t_{end}]$.

Step 1. Set $t = t_{start}$, and $i = 0$.
Step 2. Choose a positive number $\varDelta \le t_{end} - t$, and find a positive B such that $\lambda(t) < B$ on $[t, t + \varDelta]$ under the condition that no event occurs in $[t, t + \varDelta]$.
Step 3. Simulate a waiting time, τ, according to a Poisson process of rate B, i.e., $\tau \sim Exp(B)$.
Step 4. Generate a random variate $U \sim Uniform(0, 1)$.
Step 5. If $U \le \lambda(t + \tau)/B$ and $\tau < \varDelta$, set $t_{i+1} = t + \tau$, $t = t + \tau$ and $i = i + 1$. Otherwise, set $t = t + \min(\tau, \varDelta)$—i.e., reject the event and move on in time anyway.
Step 6. Return to step 2 if t does not exceed t_{end}.
Step 7. Return $\{t_1, t_2, \cdots, t_n\}$, where n is sequential number of the last event.

In Step 2, if $\lambda(t)$ is an increasing function, the value of $\varDelta$ could be chosen in a way such that $\varDelta \approx 1/\lambda(t) \sim 10/\lambda(t)$.

Having covered the essential simulation techniques in these last two sections, the remainder of the article demonstrates their application to specific earthquake-related models. New techniques are introduced along the way as needed, that are either built on the above methods or are simpler techniques worth knowing about for special cases.

4 Simulation of earthquake magnitudes, moments, and energies

The distribution of earthquake magnitudes is usually implemented as one of the following: (a) the Gutenberg–Richter (G–R) magnitude-frequency relation, (b) the tapered exponential distribution, and (c) an empirical distribution. This section briefly explains how to simulate earthquake magnitudes according to the above three cases.

4.1 Simulation based on the G-R magnitude frequency relation

The G-R magnitude frequency relation states

$$\log_{10} N(\geqslant m) = a + b\,m \tag{13}$$

for the earthquakes from a particular region in a given time period, where $\log_{10} N(\geqslant m)$ is the logarithm of the number of events of magnitude greater than or equal to m, a and b are constants, and b is the so-called G–R b-value. This empirical law can be translated into probability language: the magnitude distribution of these earthquakes has an exponential probability density function

$$f(m) = \beta\,\mathrm{e}^{-\beta(m-m_0)}; \quad m \geqslant m_0 \tag{14}$$

if we only look at the events of magnitude greater than m_0. The value β is linked with the G–R b-value by

$$\beta = b \ln 10. \tag{15}$$

The a parameter in Gutenberg–Richter scales the curve to reflect absolute rates of events; this information is not contained in the p.d.f. for a given event's magnitude (equation (14)), which is concerned only with capturing the relative rates of events of different sizes.

We have already seen how to sample from an exponential distribution in the simple example of section 2.1, for the inter-event times in a Poisson process. If we change variables to $m' = m - m_0$, simulating the magnitude with the inversion method follows the exact same procedure as that example. The cumulative function is $F(m') = 1 - \mathrm{e}^{-\beta m'}$; generate a uniform random variate U in $[0, 1]$ and then let $m' = -\frac{1}{\beta} \log U$, and finally, convert back to $m = m' + m_0$.

An exponential law for the magnitude distribution corresponds to an inverse power law (also called the Pareto distribution) for the earthquake energy, moment, or Benioff strain, i.e.,

$$\Pr\{\text{seismic moment} > M\} = CM^{-\alpha}, \quad \text{for } \alpha > 0,\ M > 0. \tag{16}$$

Simulation of such a distribution can be similarly done through the inversion method.

A shortcoming of using the p.d.f. in equation (14) is that the maximum expected magnitude increases to infinity when the total number n of samples increases to infinity. This is obviously unphysical (although in finite simulations of magnitudes it may not be a problem). To avoid this eventually unbounded maximum magnitude, some researchers simply truncate the distribution at some maximum magnitude m_M:

$$f(m) = \begin{cases} \frac{\beta\,\mathrm{e}^{-\beta(m-m_0)}}{1-\mathrm{e}^{-\beta(m_M-m_0)}}, & m_0 \leqslant m \leqslant m_M; \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

To simulate with this truncated distribution, we would repeat the same steps as for the non-truncated distribution and then only accept the sample if it is less than m_M.

4.2 Simulating from the tapered Pareto (Kagan) distribution

In reality, the frequency-magnitude relation is likely to have a more gradual roll-off at large magnitude. *Kagan and Schoenberg* (2001) suggested the tapered Pareto distribution for describing earthquake moments, which takes the form

$$\Pr\{\text{seismic moment} > M\} = \left(\frac{M}{M_0}\right)^k \exp\left(\frac{M_0 - M}{M_c}\right) \tag{18}$$

with a density function

$$f(M) = \left(\frac{k-1}{M} + \frac{1}{M_c}\right)\left(\frac{M}{M_0}\right)^k \exp\left(\frac{M_0 - M}{M_c}\right), \tag{19}$$

where M_0 is the energy corresponding to the threshold magnitude m_0, and M_c is the corner moment.

This is essentially a power-law distribution in the seismic moment (corresponding to the usual exponential in the magnitude), with an exponential roll-off at large M. This distribution could be simulated using the inversion or rejection methods. However, *Vere-Jones et al.* (2001) suggested a simpler method, taking advantage of the order statistics between exponential and Pareto random variables:

Algorithm 8 *Simulate the Kagan distribution*

```
Step 1. Simulate $X$ and $Y$ from an exponential distribution with rate $1/M_c$ and a
   Pareto distribution with a survival function $y^k/M_0^k$, respectively.
Step 2. $\min\{X, Y\}$ is the required random variate.
```

In terms of magnitudes, the Kagan distribution for seismic moment corresponds to a tapered exponential distribution

$$\Pr\{\text{magnitude} > m\} = \exp\left\{-\beta(m - m_0) - \gamma\, \mathrm{e}^{\delta(m-m_c)}\right\}, \quad \text{for } m > m_0, \tag{20}$$

where γ and δ are positive constants. Such a distribution can also be simulated by using the order statistics between an exponential and a double exponential random variables, i.e., the smaller one between two random variables from distributions whose survival functions are $\exp\left[-\beta(m - m_0)\right]$ and $\exp\left[-\gamma\, \mathrm{e}^{\delta(m-m_c)}\right]$.

4.3 Simulating magnitudes by random resampling

Simulation from the magnitude distributions above requires that the parameters of a given distribution must already be estimated or known. When the parameters from a certain region are unknown and we have the earthquake catalog from this region, we can use a simple resampling method:

```
Algorithm 9 Resample magnitudes from existing catalog
    Suppose that the magnitudes in the catalog are $\{m_i : i = 1, 2, \cdots, N\}$.
  Step 1. Generate a random variate, $U$, uniformly distributed in $(0, N)$.
  Step 2. Return $m_{[U+1]}$ as the required magnitude, where $[x]$ represents the largest
    integer less than $x$.
```

5 Simulation of specific temporal earthquake models

5.1 Poisson processes

The humble Poisson process can describe a declustered catalog, or perhaps a record of the very largest events worldwide, and its simulation was covered as an example in section 2.1. However, we include here an alternative method for simulating a Poisson process, for the sake of completeness:

```
Algorithm 10 Alternative method for a stationary Poisson process
    Suppose that a stationary Poisson process with rate $\lambda$ is to be simulated within
the time interval $[t_{start}, t_{end}]$, $t_{start} < t_{end}$.
  Step 1. Generate a Poisson random variate $N$ with a mean of $\lambda(t_{end} - t_{start})$, to
    represent the total number of events occurring in $[t_{start}, t_{end}]$;
  Step 2. Generate the times of the $N$ events, uniformly distributed in $[t_{start}, t_{end}]$.
```

From the viewpoint of computing, this method is not so efficient, since simulating a Poisson random variable takes a much longer time than simulating an exponential random variable. However, most high level computing languages have provided interfaces for generating Poisson random variables, for example, `poissrnd` in Matlab and `rpois` in R.

This method can also be combined with rejection for simulating a non-stationary Poisson process (which might be used to represent the background process after declustering of a non-stationary catalog, for example due to volcanic or fluid-driven processes):

```
Algorithm 11 simulate a non-stationary Poisson process using the rejection/thinning
method
    Suppose that the rate of the process is $\lambda(t)$ and that this rate has an upper bound
of $B$ (a positive, real number).
```

Step 1. Generate N according to a Poisson distribution of mean $B\,(t_{end} - t_{start})$.
Step 2. Generate N events t_1, t_2, $\cdots$, t_N uniformly over $[t_{start},\, t_{end}]$.
Step 3. For each $i = 1, 2, \cdots, N$, independently generate a uniform random variate U_i, and accept t_i if $U_i \leqslant \lambda(t_i)/B$; otherwise, reject t_i.

Thus, some events are rejected, and the acceptance level depends on the value of the rate at the time of each event, $\lambda(t_i)$. The accepted events form a non-stationary Poisson process with rate $\lambda(t)$. This can also suffer from inefficiency: especially when the values of $\lambda(t)$ are generally quite small outside of some particular periods where $\lambda(t)$ is large, and most generated events in the envelope process will be rejected. In this case, a better approach may be to divide the whole target interval into many small intervals, find the upper bounds of $\lambda(t)$ on each interval, and then apply the above algorithm to each interval.

The Omori-Utsu law for aftershock occurrence (or Reasenberg–Jones model),

$$\lambda(t) = \frac{K}{(t+c)^p}, \tag{21}$$

where t is the time since the mainshock, and K, c and p are constants (e.g., *Zhuang et al.* 2012 for details), can be simulated using the above algorithm. The algorithm can be also easily generalized to spatiotemporal Poisson processes, and can be implemented in parallel computing.

5.2 Renewal models

A renewal model can be thought of as a generalisation of the Poisson process, but with inter-event times that have a specified distribution—not necessarily exponential. Given the p.d.f. of inter-event times, say $f(\tau)$, simulating a renewal model is quite straightforward as for the Poisson process: simulate the waiting times (independently), and then set the event times accordingly. In other words, first simulate a group of random variates from $f(\tau)$, T_1, T_2, $\cdots$, T_n, and then let the occurrence times in the simulated process be $\{t_1, t_2, \cdots t_n\}$ with $t_1 = T_1$, $t_2 = t_1 + T_2$, $\cdots$, $t_n = t_{n-1} + T_n$.

The only tricky thing is that the whole process usually does not start at the occurrence time of an event. Thus, the first time interval should be simulated according to the distribution of the forward occurrence times (see *Zhuang et al.* 2012), whose p.d.f. is

$$w_f(\tau) = \frac{1 - F(\tau)}{\int_0^\infty v\, f(v)\, \mathrm{d}v}, \tag{22}$$

provided that a finite mean of the waiting times exists.

5.3 Stress release models

The Stress Release Model is based on the idea that earthquakes occur when elastic stress accumulates to a critical level, and takes time to build up again afterwards. The conditional intensity of a Stress Release Model takes the form

$$\lambda(t) = \exp\left[a + b\,t - c\,S(t)\right], \tag{23}$$

(e.g. *Zhuang et al.* 2012) where

$$S(t) = \sum_{i:\,t_i<t} 10^{0.75 m_i}$$

represents the total Benioff strain released by previous earthquakes, and a, b, and c are constants.

The survival function (see equation (10)) for the distribution of waiting times to the next event from t is

$$S_t(\tau) = \exp\left[-\int_t^{t+\tau} \mathrm{e}^{a+b\,v-c\,S(v)}\,\mathrm{d}v\right] = \exp\left[-\frac{\mathrm{e}^{a-c\,S(t)}}{b}\left(\mathrm{e}^{b(t+\tau)} - \mathrm{e}^{bt}\right)\right].$$

Because $S_t(\cdot) = 1 - F_t(\cdot)$, replacing $F_t(\cdot)$ with a uniform random number between 0 and 1 is equivalent to replacing $S_t(\cdot)$ with such a random number. Thus, to simulate the waiting time using the inversion method, let $S_t(\tau) = U$, where U is a random variate uniformly distributed in $[0, 1]$. Then

$$\tau = \frac{1}{b}\log\left[\frac{-b\log U}{\mathrm{e}^{a-c\,S(t)}} + \mathrm{e}^{b\,t}\right] - t \tag{24}$$

The magnitudes of simulated earthquakes could be generated by using the methods in section 4.

The Stress Release Model could also be simulated using the modified thinning algorithm as follows: Suppose that at the starting time t, the conditional intensity is $\lambda(t)$. As discussed in Section 3.2, if $\lambda(t)$ is an increasing function between two neighbouring events, $\varDelta$ could be chosen between $2/\lambda(t)$ and $10/\lambda(10)$ in order not to get $\varDelta$ too small or B too big. Here we can simply set $\varDelta = 2/\lambda(t)$ and $B = \lambda(t) * \mathrm{e}^{2b/\lambda(t)}$, and then follow the modified thinning method (section 3.2).

5.4 Temporal ETAS models

The ETAS model (*Zhuang et al.* 2012) has a conditional intensity of the form

$$\lambda(t) = \mu + \sum_{i:\,t_i<t} \kappa(m_i)\, g(t - t_i) \tag{25}$$

where

$$\kappa(m) = A\,\mathrm{e}^{\alpha(m-m_0)}, \quad m \geqslant m_0,$$

$$g(t) = \frac{p-1}{c}\left(1+\frac{t}{c}\right)^{-p},$$

m_0 being the magnitude threshold.

Simulation of ETAS was loosely described in words in section 3.2 in introducing the thinning method. More formally, the process is as follows:

Algorithm 12 *The thinning algorithm applied to the ETAS model*

Suppose that we have observations $\{(t_i, m_i) : i = 1, 2, \cdots, K\}$ up to time t_{start} and we want to simulate the process forward into the time interval $[t_{start},\, t_{end}]$.

Step 1. Set $t = t_{start}$, $k = K$.
Step 2. Let $B = \lambda(t)$.
Step 3. From time t, simulate the waiting time to the occurrence of the next event, say τ, according to a Poisson process with rate B. This can be done by simulating from an exponential distribution using the inversion method, as in the simple example of section 2.1.
Step 4. If $t + \tau > t_{end}$, go to Step 8.
Step 5. Generate a uniform random variate U in $[0,\, 1]$.
Step 6. Set $t_{k+1} = t + \tau$, generate the magnitude m_{k+1} according to the methods given in section 4, and let $k = k + 1$, if $U < \lambda(t+\tau)/B$; otherwise, reject the sample.
Step 7. Set $t = t + \tau$ regardless and go to Step 2.
Step 8. Append newly accepted events $\{(t_i, m_i) : i = K+1, K+2, \cdots\}$ to the catalog.

An implementation of this method is contained in the PtProcess package of SSLib, written in R (*Harte* 2012).

There are two equivalent ways of looking at the ETAS model. Thus far, we have treated it as a single point process, with the probability of an event at each point in time reflected in the conditional intensity—which contains a component of ‘background’ and a component of ‘triggered’ seismicity contributed by all past events in the history of the process. The ETAS model is also a branching model; it is mathematically equivalent to consider each event as belonging to one of two processes, the background and the aftershocks. Each background event results in a cascade of triggering—direct aftershocks, aftershocks of those aftershocks, and so on. Thus it is possible to simulate the background events as a Poisson process, and then recursively simulate the aftershocks resulting from each of these background events in turn. Finally, all events are combined and put into the correct temporal order of occurrence.

Algorithm 13 *Simulating the ETAS model as a branching process (see, e.g., Zhuang et al. 2004; Zhuang 2006, 2011)*

Suppose that we have observations $\{(t_i, m_i) : i = 1, 2, \cdots, K\}$ up to time t_{start} and we want to simulate the process forward into the time interval $[t_{start}, t_{end}]$.

Step 1. Generate the background catalog as a stationary Poisson process with the estimated background intensity μ in $[t_{start}, t_{end}]$, together with original observations $\{(t_i, m_i) : i = 1, 2, \cdots, K\}$ before t_{start}, recorded as Generation 0, namely $G^{(0)}$.

Step 2. Set $\ell = 0$.

Step 3. For each event i, namely (t_i, m_i), in the catalog $G^{(\ell)}$, simulate its $N_i^{(\ell)}$ offspring, namely, $O_i^{(\ell)} = \{(t_k^{(i)}, m_k^{(i)}) : i = 1, 2, \cdots, N_i^{(\ell)}\}$, where $N_i^{(\ell)}$ is a Poisson random variable with a mean of $\kappa(m_i)$, and $t_k^{(i)}$ is generated from the probability density $g(t - t_i)$, and $m_k^{(i)}$ is generated according the methods given in Section 4.

Step 4. Set $G^{(\ell+1)} = \cup_{i \in G^{(\ell)}} O_i^{(\ell)}$.

Step 5. Remove all the events whose occurrence times after t_{end} from $G^{(\ell+1)}$.

Step 6. If $G^{(\ell+1)}$ is not empty, let $\ell = \ell + 1$ and go to Step 3; else return $\cup_{j=0^\ell} G^{(\ell+1)}$
.

This has a number of advantages: it is computationally more efficient (and thus quicker), as the sum over past events is only done within a single aftershock cascade for each simulated event, not over the whole history as in the first method. It also allows events to be labelled to indicate their position in the branching process, in such a way that the 'family tree' information of background event and multiple generations of aftershocks is preserved and available for analysis—information that is not available in real earthquake catalogs (although it can be stochastically reconstructed or estimated).

An R code implementation of this method is given in the accompanying software. See Section 8 below for further detail on this code.

When simulating the temporal ETAS model, there are two kinds of boundary problem: the temporal boundary and the magnitude boundary. The magnitude boundary is caused by the neglecting of the triggering effect from the events smaller than the magnitude threshold. It is not only a problem for simulation, but also for parameter estimation and other statistical inference. Here we refer to Wang et al. (2010) and Werner (2007) for details. The temporal boundary problem is: if we simulate an ETAS model from a starting time and the observation history before this starting time is empty, the whole process needs a certain time to come into equilibrium. This problem could be partially solved by simulating a longer catalog than required, and then neglecting the very early part of the simulated catalog.

Temporal models for earthquake occurrence are at the heart of statistical seismology. In this section we have looked at how to simulate some of the main ones. One further thing to be aware of is that for models with temporal interactions between the events, such as SRM and ETAS model, starting a simulation without any event history is an artificial temporal boundary condition, as any time window should contain aftershocks of events occurrence before the start of the window. This is a practical issue for simulation and is discussed more fully (with suggested solution) in Section 8.

In the next two sections, we consider models that include some information on the spatial dimensions of the earthquake process. Section 6 discusses models that incorporate the spatial aspect in a relatively simple way, by assigning earthquakes to one of a number of 'regions' and attempting to model interactions between these regions. Section 7 looks at models that include a more complete description of the spatial characteristics of the process, by assigning spatial coordinates to each event and considering interactions at the level of the events themselves.

6 Simulation of multi-region models

Multi-region models such as the linked Stress Release Model partition the study region, giving each part its own conditional intensity, and consider interactions between the sub-regions such as stress transfer from earthquakes. Suppose the observed earthquakes are denoted by $\{(t_i, R_i, m_i) : i = 1, 2, \cdots, N\}$, where R_i is an index for the subregion where the ith event falls, taking values from 1 to Z, Z being the total number of subregions, and N is the total number of events. The conditional intensity of the linked Stress Release Model for the jth subregion is written as

$$\lambda_j(t) = a_j \, \exp\left[b\,t - \sum_{k=1}^{Z} c_{jk}\, S_k(t) \right], \qquad j = 1, Z \tag{26}$$

where Z is the number of subregions, a and b are constants, c_{jk} is a coupling term, and $S_k(t)$ is the stress released by earthquakes from subregion k, defined by

$$S_k(t) = \sum_{i=1}^{N} 10^{0.75 m_i}\, I(R_i = k)\, I(t_i < t), \tag{27}$$

the logical function $I(\cdot)$ taking value 1 if the specified condition is true and 0 otherwise. Equation (26) above is similar to equation (23) for the Stress Release Model, but with the stress release term here being a sum over all other subregions. We refer to *Zhuang et al.* (in preparation) for details of the linked Stress Release Model.

Another typical multi-region model is Hawkes' mutally exciting process, which adopts a conditional intensity of the form

$$\lambda_j(t) = \mu_j + \sum_{k=1}^{Z} \sum_{i:\, t_i<t} g_{R_i}^{(k)}(t - t_i). \tag{28}$$

Here we refer to the CORSSA article of Zhuang et al. (2014) for more details.

Both the linked Stress Release Model and the mutually and self-exciting Hawkes process can be simulated using the following algorithm.

Algorithm 14 *Simulate the linked Stress Release Model and Hawkes' mutally exciting process*

Suppose that the observed earthquakes up to time t are $\{(t_i, R_i) : i = 1, 2, \cdots, N\}$.

```
Step 1. From $t$, generate next events in each subregion independently according to
    the simulation of the Stress Release Model given in Section 5.3.
Step 2. Add the earliest event, say $(t_l, R_l)$, to the catalog.
Step 3. Set $t = t_l$ and go back to Step 1 until the terminating condition is reached
    (for example, enough events are generated, or a preset ending time is reached).
```

7 Simulation of spatiotemporal models

We now look at models that incorporate a full spatial characterisation by assigning more specific coordinates to events, beginning with the nonhomogeneous and nonstationary spatiotemporal Poisson model and ending with the ETAS model. The simulation of the nonhomogeneous Poisson model is often involved in simulation of other spatiotemporal models. The spatiotemporal ETAS model is constructed from temporal ETAS in such a way that the temporal branching structure remains unchanged, and the assignment of locations to simulated events is done independently by sampling from distributions that describe the locations of background events and of aftershocks (in relation to the parent event).

7.1 Inhomogeneous Poisson model

Let us consider the most terrible case of a Poisson model with a intensity function $\lambda(t, x, m)$ on a region R, where x denotes a location in the 2D or 3D space. It can be simulated by sequential conditioning by generating each component one by one, but we can use the rejection method to simulate for the most general cases.

Algorithm 15 *The thinning method applied to the nonhomogeneous Poisson model*

```
Step 1. Choose a density function $\pi(t, x, m)$, which is easier to simulate than
    $\lambda(t, x, M)$, such that $C\pi(t, x, m) > \lambda(t, x, M)$ on a regular region $R'$ that
    envelopes the region of interest $R$, where $C$ is a constant.
Step 2. Let $\Omega = \int_{\mathcal{M}} \mathrm{d}m \int_{R'} \mathrm{d}x \int_0^{t_{end}} \pi(t, x, M) dt$.
Step 3. Generate a Poisson random variate $K$ with mean $\Omega$.
Step 4. Generate $K$ random vectors $(t_i, x_i, M_i)$ by simulating the density function
    $\pi(t, x, m)$.
```

Step 5. Generate K random numbers, $U_1, U_2, \cdots, U_K$, uniformly distributed in $[0, 1]$.

Step 6. For each i, accept (t_i, x_i, M_i) if $CU_i < \lambda(t_i, x_i, M_i)/\pi(t_i, x_i, M_i)$; otherwise, reject (t_i, x_i, M_i).

Step 7. Sort all the accepted (t_i, x_i, M_i) by t_i into ascending order.

The choice of $\pi(t, x, M)$, which is usually called the envelope function, is an essential technique in using the rejection method. The most simple choice is to set $\pi(t, x, M)$ to a constant function. A slightly more complex choice is a density function with the time, the location and the magnitude to be independent of each other, i.e., to set $\pi(t, x, M) = \pi_t(t)\pi_x(x)\pi_M(M)$, such that we generate each coordinate independently.

The choice of the covering region R' usually is made in such a way that the integral in Step 2 can be easily obtained. For example, in the 2D case, when $\pi_{x,y}(x, y) = \mathrm{e}^{ax+by}$ and R is a complex polygon, R' could be chosen as a rectangle of the form of $[a_1, a_2] \times [b_1, b_2]$ that covers R.

7.2 The space-time ETAS model

The space-time ETAS model has a conditional intensity in the form of

$$\lambda(t, x, y, m) = s(m) \left[\mu(x, y) + \sum_{i:t_i<t} \kappa(m_i)\, g(t - t_i)\, f(x - x_i, y - y_i; m_i) \right], \qquad (29)$$

where $s(m)$ is the magnitude probability density for all the events, $\mu(x, y)$ is the background rate, $\kappa(m) = A\, e^{\alpha(m-m_0)}$, $m \geq m_0$, is the expectation of the number of direct offspring (productivity) from an event of magnitude m, $g(t) = \frac{p-1}{c}\left(1 + \frac{t}{c}\right)^{-p}$, $t > 0$, is the p.d.f. of the length of the time interval between a direct offspring and its parent, and $f(x, y; m)$ is the p.d.f. of the relative locations between a parent of magnitude m and its direct offspring.

The simulation of a space-time ETAS model can be done using the branching algorithm as described in section 5.4 for temporal ETAS, adding the spatial components in by simulating them independently, using the inversion method for example. It requires a spatial background distribution for the initial background events, and an aftershock kernel—a p.d.f. describing the placement of aftershocks around a parent event. The algorithm is outlined as below.

Algorithm 16 *Simulating the space-time ETAS model as a branching process (see, e.g., Zhuang et al. 2004; Zhuang 2006, 2011)*

Suppose that we have observations $\{(t_i, x_i, y_i, m_i) : i = 1, 2, \cdots, K\}$ up to time t_{start} and we want to simulate the process forward into the time interval $[t_{start}, t_{end}]$ in a spatial region R.

Step 1. Generate the background catalog as a Poisson process with the background intensity $\mu(t, x, y)$ *in* $[t_{start}, t_{end}] \times R$ *and magnitude of each event from* $s(m)$*, together with original observations* $\{(t_i, x_i, y_i, m_i) : i = 1, 2, \cdots, K\}$ *before* t_{start}*, recorded as Generation 0, namely* $G^{(0)}$*.*

Step 2. Set $\ell = 0$*.*

Step 3. For each event i, namely (t_i, x_i, y_i, m_i)*, in the catalog* $G^{(\ell)}$*, simulate its* $N_i^{(\ell)}$ *offspring, namely,* $O_i^{(\ell)} = \{(t_k^{(i)}, x_k^{(i)}, y_k^{(i)}, m_k^{(i)}) : i = 1, 2, \cdots, N_i^{(\ell)}\}$*, where* $N_i^{(\ell)}$ *is a Poisson random variable with a mean of* $\kappa(m_i)$*, and* $t_k^{(i)}$*,* $(x_k^{(i)}, y_k^{(i)})$*, and* $m_k^{(i)}$ *are generated from the probability densities* $g(t - t_i)$*,* $f(x, y; m_i)$*, and* $s(m)$*, respectively.*

Step 4. Set $G^{(\ell+1)} = \cup_{i \in G^{(\ell)}} O_i^{(\ell)}$*.*

Step 5. Remove all the events whose occurrence times after t_{end} *from* $G^{(\ell+1)}$*.*

Step 6. If $G^{(\ell+1)}$ *is not empty, let* $\ell = \ell + 1$ *and go to Step 3; else return all the events from* $\cup_{j=0}^{\ell} G^{(\ell+1)}$ *that are in R, i.e, return* $\{(t_\iota, x_\iota, y_\iota, m_\iota) : (t_\iota, x_\iota, y_\iota, m_\iota) \in \cup_{j=0}^{\ell} G^{(\ell+1)}$ *and* $(x_\iota,\ y_\iota) \in R\}$ *.*

An R code for simulating spatiotemporal ETAS in this way is given in the accompanying software, which uses a spatially uniform background over a circular or rectangular target area.

As noted for the temporal ETAS model, there can be practical issues associated with the boundary problem; in space-time ETAS, the spatial boundary is also important to consider. If no background events (or indeed aftershocks) are allowed to occur outside the region of interest, the aftershocks that would have been generated by such events *within* the target region will be missing; thus there will be an artificially lower density of events near the boundary, similar to the low level of events in the temporal run-in period. This could be partially solved by starting the simulation within a much bigger spatial window than the target one and then trimming all the events falling out of the target region. The estimation of the extra distance needed at the boundaries can be done in a similar way to that for the temporal run-in period estimation: in equation (7), if $F(R)$ is set to 0.95, this gives the expected radius for the circle containing 95% of the aftershocks—a kind of spatial equivalent of the effective sequence duration. The accompanying software uses twice this distance as a ‘buffer.’ It is also possible to wrap sequences spatially (a periodic boundary condition), as was explained in terms of temporal wrapping for temporal ETAS.

Challenges arise when models include some coupling between components. There is also the issue of sampling from a complicated region shape or spatial background distribution. We explore these challenges now by looking at an inhomogeneous Poisson model that includes these features.

Again, for models with temporal interactions between the events, starting a simulation without any event history is an artificial temporal boundary condition that should usually be avoided. For models with spatial interactions between events, there is a similar issue of applying spatial boundaries in a realistic way. Please see section 8 below for a discussion of this.

8 Practical issues: Boundary problems

Of course, there are many issues while carrying out simulations of seismicity models in practice. The most frequent issue is the boundary problem. Boundary problems are of three types: (a) temporal boundaries, (b) spatial boundaries, and (c) magnitude boundaries.

Magnitude boundaries were touched upon in section 4, where it was noted that a lower magnitude bound is needed for most p.d.f.s describing magnitude (such as the exponential Gutenberg–Richter one), and in some cases an upper bound (sharp or tapered) is also needed to prevent unphysically large events being simulated. In this section we briefly discuss the importance of handling temporal and spatial boundaries carefully in simulations.

Hard temporal boundaries can lead to artifacts in simulations of models which have some interation between events (and thus some time-dependence), such as SRM or ETAS. Using the simulation of temporal ETAS model as an example, if there is no 'history' before the start of the simulation, that is, past events from which aftershocks can be triggered within the target time window of the simulation, the event rate will start off artificially low and can take some time to reach a stationary state. Initially, the conditional intensity takes the value of μ; it then gradually increases as more events are added to the catalog and contribute through the triggering term. Assuming the branching ratio is less than 1, the event rate will reach a finite long-term average value. The ratio of spontaneous to triggered events in the catalog also converges towards its long-term value as the simulation progresses and more aftershocks are added. Figure 1 shows an example of the effect of this on a simulation: note the low number of events near the beginning. Usually we are only interested in a simulation that is representative of the long-term stationary state, and so basing an analysis on a simulation that contains this non-stationary run-in period can be misleading.

One solution to this issue is to allow for the run-in period and remove this from the final result. The period to be removed could be, for example, the average time required for 95% of aftershocks—direct and indirect—to have occurred following a given parent event: this is $T \approx c(1-k)^{-\frac{1}{\theta}}$, where c is one of the Omori parameters, as is $p = 1 + \theta$, and k is the fraction of aftershocks to have occurred (e.g. 0.95). (This can be derived from $\int_0^T g(v)\,\mathrm{d}v = k$, where $g(t)$ is defined in Equation (25).)

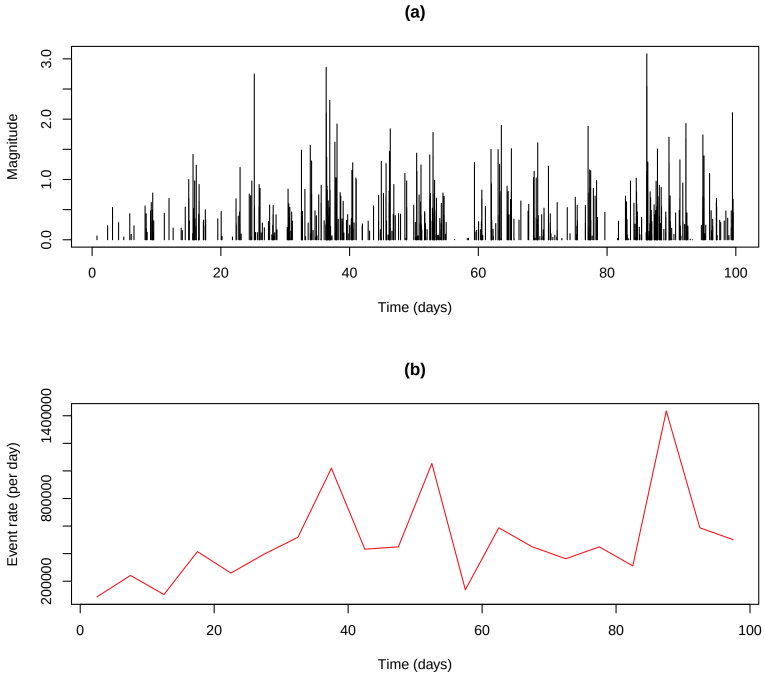

Fig. 1 A simulation of the ETAS model from time 0 to 100 days, without any history prior to time 0. Plot (a) shows the events; (b) shows the average event rate (per day) in a five-day moving window throughout the simulation. (The background rate used was $\mu = 1$ event per day, and the aftershocks governed by a branching ratio of $n = 0.96$.)

However be aware that this can sometimes be quite large. The problem is especially acute for simulations with high μ, as the high temporal density of events in such a case might mean that the computational time per unit of simulation time is very large.

Another option is to simulate each Omori sequence for a fixed duration (such as the duration for 95% of the aftershocks to have occurred, as above), even if they overshoot the end of the catalog, and then wrap aftershocks occurring after the end time back to the start (as many times as necessary). These aftershocks will now be considered to be offspring of events occurring before the start of the simulation. This removes the need to discard a run-in period; however, in order to avoid excessive wrapping, the simulated catalog still needs to be of a reasonable length (and truncated at the end if necessary). This is the technique used in the accompanying software.

For some particular types of temporal Hawkes models, a theoretical solution, which is called "perfect simulation," was found by *Møller and Rasmussen* (2005).

For spatial models with interactions between events, there is a spatial boundary problem of a similar nature: if no background events (or indeed aftershocks) are allowed to occur outside the region of interest, the aftershocks that would have been generated by such events *within* the target region will be missing; thus there will be an artificially lower density of events near the boundary. This could perhaps be mitigated by starting the simulation within a much bigger spatial window than the target one, and then at the end, trimming all the events falling out of the target region. The estimation of the extra distance needed at the boundaries can be done in a similar way to that for the temporal run-in period estimation: in equation (7), if $F(R)$ is set to 0.95, this gives the expected radius for the circle containing 95% of the aftershocks—a kind of spatial equivalent of the effective sequence duration. The accompanying software uses twice this distance as a 'buffer.' It is also possible to wrap sequences spatially (a periodic boundary condition), as described above in terms of temporal wrapping.

9 Usages of simulations

9.1 Visualizing models

As mentioned in section 1, one of the usages of simulation is visualizing the model. Once a statistical model is formulated, one can use explorative data analysis techniques (such as those introduced in Theme III of CORSSA) to have a visual impression of how the characteristics of the simulated catalog change with the model parameters.

The following example is taken from *Harte and Vere-Jones* (2005). Consider a renewal model with a gamma p.d.f. of waiting times:

$$f(\tau, \kappa, \theta) = \tau^{\kappa-1} \frac{e^{-\tau/\theta}}{\theta^\kappa \, \Gamma(\kappa)}, \tag{30}$$

where κ is the shape parameter, θ is the scale parameter, and $\Gamma(\cdot)$ represents Euler's gamma function defined by $\Gamma(x) = \int_0^\infty v^{x-1} \mathrm{e}^{-v} \mathrm{d}v$. It is obvious that we can always choose a proper time unit for τ such that $\theta = 1$. Figure 2 gives the simulation results for different κ: We can see that, when $\kappa < 1$, the process is clustered; when $\kappa > 1$, the process becomes more regularly spaced; when $\kappa = 1$, the process is neither clustered nor regularly spaced, but corresponds to the case of complete randomness, i.e., the Poisson process.

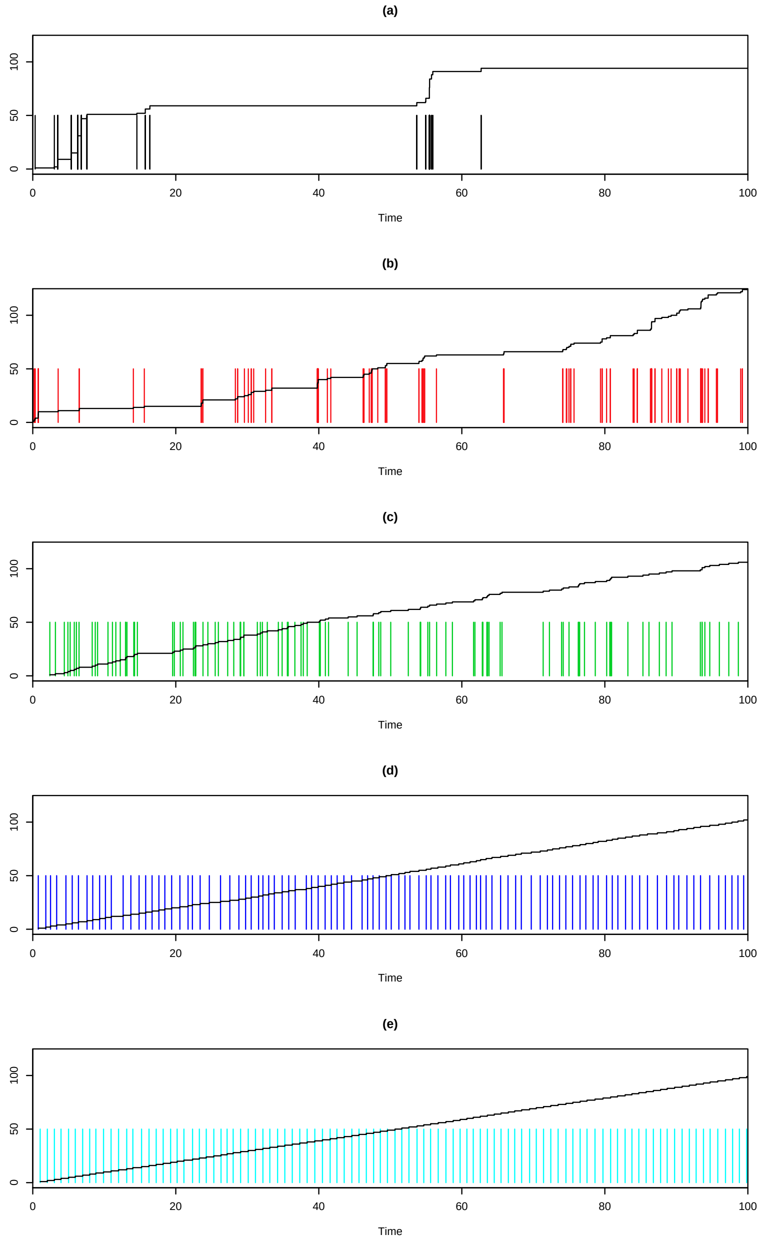

Fig. 2 Cumulative frequencies of occurrences in simulations of the gamma renewal process for (a) $\kappa = 0.04$, (b) $\kappa = 0.2$, (c) $\kappa = 1$, (d) $\kappa = 10$, and (e) $\kappa = 100$. The average occurrence is set 1 by adjusting the value of θ. The vertical bars represent the occurrence times of each event.

9.2 Probability earthquake forecasting

Suppose we are given the observation of earthquakes and precursors to a time t, and try to predict the occurrence of at least one event in a given time interval $[t, t+dt]$ within a particular space-time window with a formulated model (*Vere-Jones* 1998). Then time T^* to the next event from the time t is

$$\Pr\{T^* > \tau\} = \exp\left[-\int_0^\tau \lambda(t+v)dv\right]. \tag{31}$$

Algorithm 17 *Forecasting based on simulations*

Step 1. Fit the model to the observation data up to the given time t to get the model parameters by using, e.g., maximum likelihood estimation.

Step 2. Simulate a time τ to the next occurring event by using the $\lambda(t)$, or simulate the time to the next occurring event and its location and magnitude by using the $\lambda(t, x, M)$ if the conditional intensity has a space-time-magnitude form. The simulation method generally depends on the form of the conditional intensity function.

Step 3. Add the new events to the history, and return to step (2), replacing t by $t+\tau$.

Step 4. Continue until the end of the time interval for forecasting.

Step 5. Repeat the above simulation in Steps (2) to (5) many times; the proportion containing at least one event in the target prediction space-time-magnitude window can be determined and used as an estimate of the required forecast probability.

There are many references on forecasting seismicity using simulations, for example, *Vere-Jones* (1998), *Helmstetter et al.* (2006) and *Zhuang* (2011).

10 Software

Most simple processes can be simulated with several lines of `R` or `Matlab` commands. `R` codes for the simulation of complex models, such as the Stress Release Model and its multi-region version, can be found in SSLIB. Alternative codes for simulating temporal and spatio-temporal ETAS are given in the accompanying software. R is a free statistical computing language developed by *R Development Core Team* (2012).

The SASeis-2006 also provides the implementation of the simulation of the temporal ETAS model and Hawkes' models in Fortran.

11 Acknowledgments

We thank Dr. Mark Naylor for serving as the editor of this article and Dr. Maximilian J. Werner for his constructive and helpful comments.

Appendix A Proof of the inversion method

If the inversion algorithm in section 2.1 is correct, then $F^{-1}(U)$ has the cumulative distribution $F(x)$, as required of the variable we want to sample. Thus we need to show that

$$\Pr\{F^{-1}(U) \leqslant x\} = F(x) \tag{32}$$

The left-hand side is equivalent to $\Pr\{U \leqslant F(x)\}$, since $F(\cdot)$ is a monotonically increasing function and can therefore be applied to both sides of the inequality without changing the probability. Also, $U = F(X)$ belongs to a uniform distribution, implying

$$\Pr\{U \leqslant F(x)\} = F(x), \tag{33}$$

as required.

Appendix B Proof of the rejection method

The random variate X generated by the rejection algorithm in section 2.2 has a density $f(\cdot)$ since

$$\begin{aligned}\Pr\{Y \text{ is accepted}\} &= \int \Pr\{Y \in (x, x + \mathrm{d}x) \text{ and } Y \text{ is accepted}\} \\ &= \int \Pr\{Y \text{ is accepted} \mid Y \in (x, x + \mathrm{d}x)\} \Pr\{Y \in (x, x + \mathrm{d}x)\} \\ &= \int \frac{f(x)}{B\, g(x)}\, g(x)\, \mathrm{d}x = \frac{1}{B} \end{aligned} \tag{34}$$

and so

$$\begin{aligned}\Pr\{Y \in (x, x + \mathrm{d}x) \mid Y \text{ is accepted}\} &= \frac{\Pr\{Y \in (x, x + \mathrm{d}x) \text{ and } Y \text{ is accepted}\}}{\Pr\{Y \text{ is accepted}\}} \\ &= \frac{g(x)\,\mathrm{d}x \times \frac{f(x)}{B\,g(x)}}{\frac{1}{B}} = f(x)\,\mathrm{d}x. \end{aligned} \tag{35}$$

This last statement says that the probability that an accepted Y value is in the range $[x, x + \mathrm{d}x]$ is equal to $f(x)\,\mathrm{d}x$. In other words, $f(x)$ is the distribution of the accepted Y values.

Appendix C Proof of the thinning method

To prove that the events generated and accepted by the thinning algorithm in section 3.2 form a point process with conditional intensity $\lambda(t)$, we need only prove that, from time t, the survival function of the waiting time τ to the next accepted event is

$$S_t(\tau) = \Pr\{\text{From time } t, \text{ no event is accepted before time } t+\tau\} = \exp\left[-\int_t^{t+\tau} \lambda(v)\mathrm{d}v\right].$$

The envelope process is a Poisson process with rate B, and so the number of events simulated in the envelope process within $[t, t+\tau]$ is a Poisson random variate K with a mean of $B\tau$:

$$\Pr\{K = k\} = \frac{[B\tau]^k}{k!}\mathrm{e}^{-B\tau}, \quad k = 0, 1, 2, \cdots.$$

When no event from the target process occurs in $[t, t+\tau]$, all of these K events in the envelope process are rejected. For each of these K events, the probability that it is rejected can be evaluated by

$$\begin{aligned}
&\Pr\{\text{An envelope-process event in } [t, t+\tau] \text{ is rejected}\}\\
&= \int_t^{t+\tau} \Pr\{\text{This envelope-process event occurs within } [v, v+\mathrm{d}v], \text{ and it is rejected}\}\,\mathrm{d}v\\
&= \int_t^{t+\tau} \frac{1}{\tau}\left[1 - \frac{\lambda(v)}{B}\right]\mathrm{d}v = 1 - \frac{1}{B\tau}\int_t^{t+\tau} \lambda(v)\,\mathrm{d}v
\end{aligned}$$

In the above equation, $\frac{1}{\tau}$ is the probability density for the location (in time) of any event in the envelope process, given that it does occur within $[t, t+\tau]$, and $1 - \lambda(v)/B$ is the probability that an event from the envelope process at location v

is rejected. Thus,

$$\begin{aligned}
S_t(\tau) &= \Pr\{\text{From time } t, \text{ no event is accepted before time } t+\tau\} \\
&= \sum_{k=0}^{\infty} \Pr\left\{\begin{array}{l}\text{there are } k \text{ events from the envelope process} \\ \text{falling in } [t, t+\tau]\text{, and all of them are rejected}\end{array}\right\} \\
&= \sum_{k=0}^{\infty} \Pr\{K=k\} \times \Pr\left\{\begin{array}{l}\text{An envelope-process event occurs} \\ \text{within } [t, t+\tau]\text{, and it is rejected}\end{array}\right\}^k \\
&= \sum_{k=0}^{\infty} \frac{[B\tau]^k}{k!} \mathrm{e}^{-B\tau} \left[1 - \frac{1}{B\tau}\int_t^{t+\tau} \lambda(v)\,\mathrm{d}v\right]^k \\
&= \mathrm{e}^{-B\tau} \sum_{k=0}^{\infty} \frac{\left[B\tau - \int_t^{t+\tau}\lambda(v)\,\mathrm{d}v\right]^k}{k!} \\
&= \exp\left[-\int_t^{t+\tau} \lambda(v)\,\mathrm{d}v\right]
\end{aligned}$$

In the last step, the power series expansion of the exponential function, $\mathrm{e}^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$, is used.

References

Console, R., M. Murru, and A. M. Lombardi (2003), Refining earthquake clustering models, *Journal of Geophysical Research*, *108*(B10), 2468, doi:10.1029/2002JB002130. 7

Daley, D. D., and D. Vere-Jones (2003), *An Introduction to Theory of Point Processes – Volume 1: Elementrary Theory and Methods (2nd Edition)*, Springer, New York, NY. 14

Harte, D. (2012), *PtProcess: Time Dependent Point Process Modelling*, R package version 3.3-1. 20

Harte, D., and D. Vere-Jones (2005), The entropy score and its uses in earthquake forecasting, *Pure and Applied Geophysics*, *162*(6), 1229–1253, doi:10.1007/s00024-004-2667-2. 28

Helmstetter, A., Y. Y. Kagan, and D. D. Jackson (2006), Comparison of short-term and time-independent earthquake forecast models for Southern California, *Bulletin of the Seismogological Society of America*, *96*(1), 90–106, doi:10.1785/0120050067. 30

Kagan, Y. Y., and F. Schoenberg (2001), Estimation of the upper cutoff parameter for the tapered pareto distribution, *Journal of Applied Probability*, *38A*, 158–175. 16

Møller, J., and J. G. Rasmussen (2005), Perfect simulation of hawkes processes, *Advances in applied probability*, pp. 629–646. 27

Ogata, Y. (1981), On Lewis' simulation method for point processes, *IEEE Transactions on Information Theory*, *IT-27*(1), 23–31. 12

R Development Core Team (2012), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0. 30

Ross, S. M. (2003), *Introduction to Probability Models, Eighth Edition*, 773 pp., Academic Press. 10

Touati, S., M. Naylor, and I. G. Main (2009), Origin and Nonuniversality of the Earthquake Interevent Time Distribution, *Physical Review Letters*, *102*(16). 4

Vere-Jones, D. (1998), Probability and information gain for earthquake forecasting, *Computational Seismology*, *30*, 248–263. 30

Vere-Jones, D., R. Robinson, and W. Yang (2001), Remarks on the accelerated moment release model: problems of model formulation, simulation and estimation, *Geophysical Journal International*, *144*(3), 517–531. 16

Wang, Q., D. D. Jackson, and Z. J. (2010), How does data censoring influence on estimates of earthquake clustering characteristics?, *accepted by to Geophysical Research Letters*. 21

Werner, M. J. (2007), On the fluctuations of seismicity and uncertainties in earthquake catalogs: Implications and methods for hypothesis testing, Ph.D. thesis, Univ. of Calif., Los Angeles. 21

Zhuang, J. (2006), Second-order residual analysis of spatiotemporal point processes and applications in model evaluation, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(4), 635–653, doi: 10.1111/j.1467-9868.2006.00559.x. 21, 24

Zhuang, J. (2011), Next-day earthquake forecasts by using the ETAS model, *Earth, Planet, and Space*, *63*, 207–216, doi:10.5047/eps.2010.12.010. 21, 24, 30

Zhuang, J., Y. Ogata, and D. Vere-Jones (2004), Analyzing earthquake clustering features by using stochastic reconstruction, *Journal of Geophysical Research*, *109*(3), B05,301, doi:10.1029/2003JB002879. 21, 24

Zhuang, J., D. Harte, S. Werner, M.J.and Hainzl, and S. Zhou (2012), Basic models of seismicity: temporal models, *Community Online Resource for Statistical Seismicity Analysis*, doi:10.5078/corssa-79905851. 4, 6, 11, 13, 14, 18, 19

Zhuang, J., M. Werner, S. Hainzl, D. Harte, and S. Zhou (in preparation), Basic models of seismicity: multi-regional models, *Community Online Resource for Statistical Seismicity Analysis*, doi:10.5078/corssa-77640714. 22