Received: 26 September 2021 | Revised: 8 December 2021 | Accepted: 13 December 2021

DOI: 10.1002/sta4.446

ORIGINAL ARTICLE

WILEY

Adapting conditional simulation using circulant embedding for irregularly spaced spatial data

Maggie D. Bailey | **Soutir Bandyopadhyay** | **Douglas Nychka**

Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, Colorado, 80401, USA

Correspondence
Soutir Bandyopadhyay, Department of Applied Mathematics and Statistics, Colorado School of Mines, 1500 Illinois St., Golden, CO 80401, USA.
Email: sbandyopadhyay@mines.edu

Funding information
Colorado School of Mines Faculty Development Funds; National Science Foundation, Grant/Award Number: DMS-1811384

Computing an ensemble of random fields using conditional simulation is an ideal method for retrieving accurate estimates of a field conditioned on available data and for quantifying the uncertainty of these realizations. Methods for generating random realizations, however, are computationally demanding, especially when the estimates are conditioned on numerous observed data and for large domains. In this article, a *new*, *approximate* conditional simulation approach is applied that builds on *circulant embedding* (CE), a fast method for simulating stationary Gaussian processes. The standard CE is restricted to simulating stationary Gaussian processes (possibly anisotropic) on regularly spaced grids. In this work, we explore two possible algorithms, namely, local Kriging and approximate grid embedding, that extend CE for irregularly spaced data points. We establish the accuracy of these methods to be suitable for practical inference and the speedup in computation allows for generating conditional fields close to an interactive time frame. The methods are motivated by the U.S. Geological Survey's software *ShakeMap*, which provides near real-time maps of shaking intensity after the occurrence of a significant earthquake. An example for the 2019 event in Ridgecrest, California, is used to illustrate our method.

KEYWORDS
circulant embedding, conditional simulation, earthquakes

1 | INTRODUCTION

An important contribution of spatial statistics is not only being able to make predictions at unobserved locations but to also attach a measure of uncertainty to these predictions. Conditional simulation is a Monte Carlo-based method that is ideal for quantifying prediction uncertainty at new locations especially when the predictions comprise a large, fine grid of points representing a surface. Besides providing Monte Carlo estimates of prediction standard errors, it can also be applied to nonlinear features in a spatial prediction such as the uncertainty in contour lines and level sets. A major computational bottleneck in conditional simulation, however, is the initial simulation of the *unconditional* spatial process at a large set of observed and unobserved locations. Using exact methods limits the analysis on a laptop and in R to a combined number of locations on the order of tens of thousands. This constraint is easily exceeded for larger spatial data sets and prediction to modest sized spatial grids. Here, we exploit the circulant embedding (CE) method (Wood & Chan, 1994; Chan & Wood, 1999) because of its efficiency and make some additions for irregular locations. The focus of our work is on geophysical problems where the conditional simulation is typically required on a fine grid and where computing resources are modest.

The value of CE lies in the fact that it is both a *fast* and *exact* method for simulating stationary Gaussian random fields on a regular grid. For example, on an $m \times m$ lattice, CE uses 40 m^2 $\log_2$ 2 m floating point operations (FLOPS) compared to 6 m^5 FLOPS using a Cholesky decomposition. This reduced operation count translates to significant speedup in simulation time and reduction in memory. If observation locations are a subset of the grid, then generating realizations from the conditional distribution is a straight forward computation by pairing CE unconditional

simulation with efficient implementation of Kriging predictions. However, a major restriction of CE is that it only simulates the process on a regular grid of locations. For data sets where the observation locations are irregular and cannot be registered to a prediction grid, it is difficult to simulate the process simultaneously at locations both on a grid and at observation locations. This follows because the efficiency of CE cannot be exploited in this case. We study two simple local algorithms that use CE and extend conditional simulation to irregular, off-grid locations. These extensions produce accurate simulations of the prediction error and take a small fraction of computational time relative to other parts of the conditional simulation algorithm. Although at its core these are simple modifications, to our knowledge, these algorithms have not been identified in previous work and implemented in open source data analysis environments such as R.

This work is motivated by a practical problem encountered by the United States Geological Survey (USGS) in using earthquake measurements. Ground motion sensors, used to measure earthquakes, are typically irregularly placed in a spatial domain. Based on these data, estimates of median ground motion are produced by the USGS software package, *ShakeMap* (Worden et al., 2020), in conjunction with observed seismic data (Verros et al., 2017). Generally, ShakeMap-based estimates are then used as input fields in additional models that quantify fatalities and economic losses from significant seismic events. The statistical problem is to extrapolate ShakeMap estimates to a fine spatial field to address losses over a complete spatial region and to quantify the uncertainty in these estimates. In particular, several hundred conditional simulations of the field are required to compute a Monte Carlo estimate of the prediction standard error.

Throughout this work, the two new methods will be referred to as *local Kriging* and *approximate grid embedding*, and we evaluate them by the accuracy in approximating the exact prediction variances over a range of spatial data models. There are several strategies in the literature related to this problem. To work with irregular grids, a block circulant embedding method has been proposed (Park & Tretyakov, 2015). This method focuses specifically on data that have a block circulant structure, however, and is less applicable to the ShakeMap software. A second approach is currently used in the R package fields and implemented as the function `sim.mKrig.approx`. Here, linear interpolation uses the four nearest grid points to simulate an observation (Furrer et al., 2009). While this strategy is very fast, it does not preserve the covariance structure of the spatial data and does not adjust for interpolation error, and its accuracy is untested.

An elegant solution to simulating unconditional and conditional Gaussian spatial fields is to use algorithms based on sequentially conditioning on subsets of the locations, termed *Vecchia* approximations (VA) (Katzfuss et al., 2020). This is a comprehensive and efficient framework for approximating Gaussian process computations. Although an alternative for simulation, coding a VA is more complex, and it is fundamentally a sequential algorithm that can be more difficult to implement in parallel. Under some circumstances, our approach fits into the Vecchia framework. For our local Kriging method, if the neighbourhoods of grid points do not overlap, then we are following a Vecchia algorithm where the final conditioning set is the entire field on the regular grid generated by CE. When neighbourhoods overlap, our method departs from the standard Vecchia setup, and we study the impact of this difference. Similar spatial process models exist such as Nearest Neighbour Gaussian Processes (NNPG) (Datta et al., 2016; Finley et al., 2019). NNPG is also a fast inferential algorithm; however, simulations are done sequentially. While the NNPG algorithm produces the same conditional distribution regardless of order and utilizes nearest neighbours via directed edges, the local Kriging algorithm proposed here is simpler in that it requires no order specification. In either case, our approach benefits from being able to simulate large and exact unconditional fields using CE that moderate the effect of computational shortcuts in simulating at the off grid observation locations. Some timing results for VA and NNGP as a comparison to local Kriging are given in Section 3.6.

Another difference with our work compared to that of using a VA is our focus on conditional simulation and the Monte Carlo approximation of the prediction standard error. We study how well our methods work in conditional inference for spatial problems and observation densities similar to the Shakemap application, and we believe this is an important metric for judging our algorithms. As a practical example, we consider sensor observations from an earthquake near Ridgecrest, CA, in 2019 and work with more than 600 spatial locations and a prediction grid of more than 250 K points. The goal is to handle this size problem in an interactive data analysis environment, such as R, and on a standard laptop computer and generate accurate prediction intervals.

In the next section, we detail the relevant computational algorithms. This is followed in Section 3 with a derivation of the misspecification of the prediction standard error under our proposed algorithms. Also, numerical results are reported for the approximation error and timing results for a spatial context relevant to USGS data. Section 4 applies the method to the Ridgecrest, CA, seismic event, and Section 5 closes with some discussion of future directions.

2 | EXTENDING CIRCULANT EMBEDDING SIMULATION

In this work, we assume a stationary, mean-zero spatial process $y(\cdot)$ with finite variance σ^2 and an associated stationary covariance function, $C(\cdot)$. For example, y may represent a particular shaking intensity measurement such as log peak ground acceleration. We assume that the spatial process is observed with error at n spatial locations $\{\mathbf{s}_i, i = 1,...,n\}$ and can be represented as the standard additive geostatistical model:

$$z_i = \mu(\mathbf{s}_i) + y(\mathbf{s}_i) + \varepsilon_i, \text{for} i = 1,...n.$$

In the above equation, z_i is the observation at location $\mathbf{s}_i$, $\mu(\cdot)$ is the mean function and ε_i's are mean zero, Gaussian random variables with variance ϕ_i^2, often referred to as the *nugget effect*. It is assumed the ε is uncorrelated with y. The mean function typically takes the form of a linear model, $\mu(\mathbf{s}_i) = \mathbf{x}(\mathbf{s}_i)^T\boldsymbol{\beta}$, where $\mathbf{x}(\mathbf{s}) = [x_1(\mathbf{s}),...,x_n(\mathbf{s})]^T$ and the parameters $\boldsymbol{\beta}$ are estimated using generalized least-squares (see Cressie, 1993). To streamline the presentation, it is assumed that $\mu(\mathbf{s}_i)$ is zero. The extension to include a linear mean function is straight forward and is mentioned in the discussion. Finally, we note that here "measurement error" is a generic component that can include microscale variation in the spatial process or essentially any component of the observation that is not predictable from other spatial locations.

Given $\mathbf{z}$, we are interested in predicting $y(\mathbf{s})$ on the regularly spaced prediction grid $\mathcal{S}^g = \{\mathbf{s}_1^g, \mathbf{s}_2^g, ..., \mathbf{s}_M^g\}$. Specifically in two dimensions, for an $m_1 \times m_2$ grid $M = m_1 m_2$. In this work, it will be convenient to use notation $y(A) = \{y(\boldsymbol{\ell}) : \boldsymbol{\ell} \in A\}$ to denote the *vector* of y variables over the set $A = \{\boldsymbol{\ell}_1, ..., \boldsymbol{\ell}_k\}$, and the order in this vector is fixed. Thus, the computational goal is to simulate values for $y(\mathcal{S}^g)$ conditioned on a spatial process observed with error, $\mathbf{z} = [z_1, ..., z_n]^T$. The conditional simulation algorithm has the following steps assuming a Gaussian process for y and normally distributed error:

1. Compute the spatial prediction of the process at the prediction grid based on the actual spatial data. Denote this by $\hat{\mathbf{y}}$.
2. Simulate a spatial process y at the union of locations $S = \mathcal{S}^g \cup \mathcal{S}$.
3. Form the synthetic data $z_i^* = y(\mathbf{s}_i) + \varepsilon_i^*$, for $\mathbf{s}_i \in \mathcal{S}$ and ε_i^* are independent $N(0, \phi_i^2)$.
4. Based on $\mathbf{z}^* = \{z_i^* : \mathbf{s}_i \in \mathcal{S}\}$, compute the spatial predictions at $\mathcal{S}^g$ resulting in $\tilde{\mathbf{y}}$.
5. Obtain the conditional simulation as $\mathbf{v} = \hat{\mathbf{y}} + (y(\mathcal{S}^g) - \tilde{\mathbf{y}})$.

The resulting vector $\mathbf{v}$ has the correct covariance structure associated with the Kriging prediction uncertainty, and multiple realizations of $\mathbf{v}$ can be interpreted as all being equally plausible representations of $y(\mathcal{S}^g)$ consistent with the observations. In more mathematical terms, $\mathbf{v}$ is a draw from the conditional distribution of $y(\mathcal{S}^g)$ *given* $\mathbf{z}$ under the assumption of a Gaussian process, known covariance and normally distributed measurement error, and we identify this explicitly in (3). It is worth mentioning that in Step 3, the nugget variance is just τ^2 (i.e., $\tau^2 \equiv \phi_i^2$) in the exact implementation of this algorithm. However, in our approximations, this variance will be inflated to reflect the additional variance of the approximations.

Note that the resulting vector $\mathbf{v}$ is a draw from the conditional distribution assuming that all covariance parameters are known but will still have more variability than just the spatial prediction, that is, the conditional mean.

2.1 | Local Kriging

The method of *local Kriging* approximates Steps 3 and 4 of the conditional simulation algorithm entailing an unconditional simulation of the spatial process at the full set of locations $S = \mathcal{S}^g \cup \mathcal{S}$. The approximation made by local Kriging starts with an unconditional simulation over the grid $\mathcal{S}^g$ using CE and results in the random vector $y(\mathcal{S}^g)$ of length M. These results are used to generate off-grid observations, $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_n\}$. Note that if this operation was done using the full conditional distribution of $y(\mathcal{S})$ given $y(\mathcal{S}^g)$, then this would be exact. The difficulty, of course, is that it becomes computationally expensive for even moderate size grids. Therefore, the goal is to use a limited subset of $y(\mathcal{S}^g)$ to generate the process at off-grid locations. We define a grid box as being formed from four adjacent grid points. Denote the order neighbourhood as n_p where this is the number of grid boxes containing a particular point in each dimension. This equates to $2n_p$ grid points closest to an off-grid observation in one dimension and $(2n_p)^2$ grid points in two dimensions (see Figure 1). With this neighbourhood scheme, it is useful to expand the range of grid points larger than any observation location by at least n_p grid units. By adding this margin one avoids edge effects; all neighbourhoods have the same size.

Let $\mathcal{S}_i^g \subset \mathcal{S}^g$ be the subset of grid points for the n_p order neighbourhood surrounding $\mathbf{s}_i$ and $y(\mathcal{S}_i^g)$ be the subvector of the simulated process on the grid. The off-grid value, $y(\mathbf{s}_i)$, is generated as a draw from the conditional distribution of $y(\mathbf{s}_i)$ given $y(\mathcal{S}_i^g)$. Two important features figure into this approximation. The first is that only the nearest neighbouring grid points are used as the conditioning set. We assert that this is not a serious issue due to the screening effect for spatial prediction. The second assumption is that for any pair of observations i and j, we assume that $y(\mathbf{s}_i)$ and $y(\mathbf{s}_j)$ are independent conditional on $y(\mathcal{S}_i^g) \cup y(\mathcal{S}_j^g)$. This is a strong assumption but can also be justified by the screening effect combined with a fine conditioning grid and it separates observation locations into different grid boxes. More discussion about this assumption will be given Section 3.5. Note that the assumption of conditional independence allows for parallel simulation of the process at the observation locations given the grid values and is different than a sequential algorithm using Vecchia approximations to the covariance matrices. The simulated observations with this local Kriging method can still exhibit strong correlations, however, based on the correlations among the conditioning grid values.

To summarize, the local Kriging method proposes the following computations to approximate Steps 2–4 in the conditional simulation algorithm.

(a) Simulate the process on the regular grid, $\mathcal{S}^g$, using CE.
(b) Using an order neighbourhood of size n_p, perform univariate sampling of $y(\mathbf{s}_i)$ conditional on $y(\mathcal{S}_i^g)$.

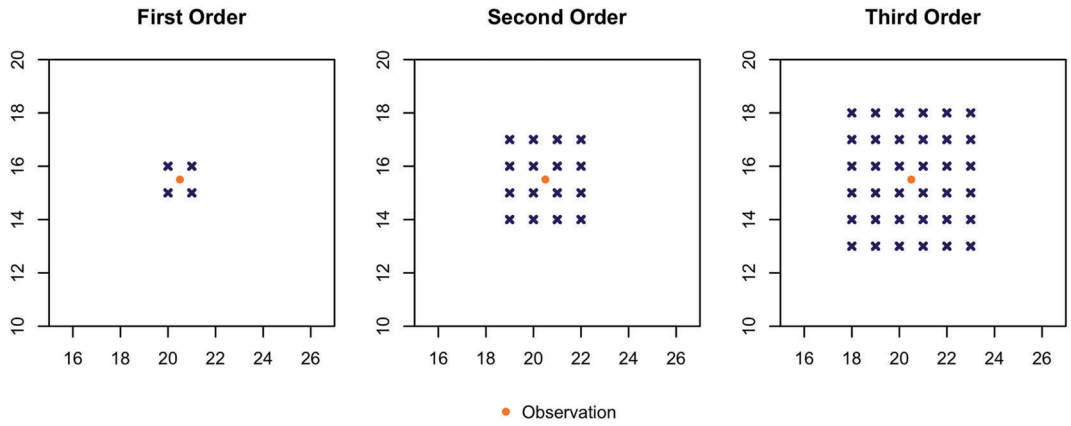

FIGURE 1 Illustration of the neighbourhood order $n_p = 1, 2, 3$ used in the local Kriging approximation

(c) Use a nugget variance of $\phi_i^2 = \tau^2 + \gamma_i$ in generating synthetic observations, where γ_i represents the conditional variance in generating synthetic observations at off-grid locations.

(d) Use τ^2 alone for the nugget variance in the spatial prediction.

This is the algorithm studied in the next section; however, a slightly more complicated version has some practical benefit and little extra computational overhead. If more than one observation location is in a grid box, then the conditional sampling in (b) can be done for the multivariate vector of observations in the grid box. This is distinct from the method proposed in Stroud et al. (2017), where simulating observations was done independently regardless of proximity to neighbouring grid boxes with observations. The local Kriging algorithm in this paper does not require that there is only one point per grid box and therefore does not require the grid to be extremely fine to accommodate this. Additionally, the method proposed by Stroud et al. (2017) utilizes all grid points from the conditional simulation at $\mathcal{S}^g$ to conditionally simulate at the observation locations. We find that using a limited sized neighbourhood rather than the full simulation suffices due to some screening. Some more details about this are given in Section 3.6 and can avoid working with very fine grids just to allocate all the observations into separate grid boxes. The conditional sampling for observations in *separate* grid boxes is still done independently. Finally, step (c) above is important in this proposed algorithm as it provides necessary variance adjustment to the conditional simulation and avoids potential underestimation of the conditional variance.

2.2 | Approximate grid embedding

A second method to adapt CE for use with off-grid points will be termed *approximate grid embedding*. In this method, the information in the spatial observations is shifted to the closest grid locations. In this way, there are no longer any off-grid locations and CE alone can be used in Step 2 of the conditional simulation process. The simplest way to shift the data is to identify the closest grid point and then just use that as the surrogate location. We make this operation more robust by a spatial prediction from the full set of observations to the n_p order neighbourhood for each observation location. This is illustrated by referring again to Figure 1. In addition, the nugget variance, nominally τ^2, is inflated at each grid point observation by the error variance from this prediction. The net result is a surrogate spatial data set that attempts to retain the information from the original, irregular locations but is now registered to locations on a regular grid. This transformation of the data and the locations is only done once for each choice of spatial model, and then, the standard conditional simulation algorithm listed above can be used without modification.

A benefit to this approximation is its simplicity, and in the case of very fine grids and a sparse set of observation locations, one would expect this approach to be very effective. However, in general, artificially increasing the number of observations adds to the computation and grids that are close to the resolution of the observation spacings may introduce biases. Here, the highlighted grid points can be thought of as new "observations," which will be used in place of the original data locations and with some additional error included in the observational model. Another disadvantage is that the redistribution of the observations to the grid points does not correspond to a specific approximation to the conditional distribution from the off-grid points (Cressie, 1993) and can result in biases.

3 | EVALUATION OF APPROXIMATION OF PREDICTION ERRORS

For the purposes of this work, the exact standard error for Kriging and approximate standard errors for local and approximate grid embedding have closed forms. These exact expressions avoid the additional complication of a simulation study to assess the accuracy of the approximations.

They also give some insight into how the approximations are made and capture the additional prediction variance, that is, step (c) in the local Kriging algorithm, which ensures a more accurate conditional variance.

The following matrices will be used to derive the explicit formulas in this section. Throughout, the subscripts 1 and 2 will refer, respectively, to the grid and observation subsets. First we have the full covariances and cross-covariance matrices for the grid points and observations:

$$\underset{(M\times M)}{K_{11}} = \mathrm{Cov}(y(\mathcal{S}^g), y(\mathcal{S}^g)) = C(\mathcal{S}^g, \mathcal{S}^g), \qquad \underset{(M\times n)}{K_{12}} = \mathrm{Cov}(y(\mathcal{S}^g), y(\mathcal{S})) = C(\mathcal{S}^g, \mathcal{S}), \; \underset{(n\times M)}{K_{21}} = K_{12}^T$$
$$\text{and} \qquad \underset{(n\times n)}{K_{22}} = \mathrm{Cov}(y(\mathcal{S}), y(\mathcal{S})) = C(\mathcal{S}, \mathcal{S}). \tag{1}$$

Also, the conditional distribution for the multivariate normal follows the well-known expression

$$[y(\mathcal{S}^g)|\mathbf{z}] \sim MN(K_{12}(K_{22} + \tau^2 I)^{-1}\mathbf{z}, K_{11} - K_{12}(K_{22} + \tau^2 I)^{-1}K_{21}),$$

and the conditional simulation algorithm is an efficient way to sample from this distribution. Moreover, the diagonal elements for the conditional covariance are the appropriate prediction variances. We study whether these variances are accurately approximated by the local Kriging and approximate grid embedding methods.

3.1 | Local Kriging approximate conditional variance

To derive the approximate conditional variance for local Kriging, we first identify the weights to predict the spatial process on $\mathcal{S}$ based on the values at $\mathcal{S}^g$. In this approximation, it is helpful to define an $n \times M$ incidence matrix, $\mathcal{J}$, of 0 and 1 values such that $\mathcal{J}_{jk}$ is 1 for all n_p order neighbouring grid points of $\mathbf{s}_j$ and zero otherwise. It is also useful to define subsets of the full covariance matrices based on these neighbourhoods. Given a specific neighbourhood size, n_p, let $\mathcal{S}_j^g$ be the grid locations comprising the neighbourhood of $\mathbf{s}_j$, and we have the covariance (sub)matrices

$$K_{22}^i = \mathrm{Var}(y(\mathbf{s}_i)) \;, \; K_{11}^i = \mathrm{Cov}(y(\mathcal{S}_i^g), y(\mathcal{S}_i^g)), \text{ and } K_{21}^i = \mathrm{Cov}(y_i, y(\mathcal{S}_i^g)).$$

We have the conditional mean and variances for the j^{th} observation given by

$$\hat{y}_j = K_{21}^i (K_{11}^i)^{-1} y(\mathcal{S}_i^g) = K_{21}^i (K_{11}^i)^{-1} \mathcal{J} y(\mathcal{S}^g) = W_1^i y(\mathcal{S}^g),$$

where W_1^i is a vector of weights that map the grid values to the conditional mean for the observation. Also, we have the conditional variance based on the local values

$$\gamma_i = K_{22}^i - K_{21}^i (K_{11}^i)^{-1} (K_{12}^i)^T.$$

Given these individual estimates, let W_1 denote the $n \times M$ matrix resulting from stacking $\{W_1^i\}$ as row vectors and γ a vector representing $\{\gamma_i\}$. With this notation, the synthetic observation vector at Step 3 in the conditional algorithm is approximated as

$$\mathbf{z}^* = \mathbf{y}^* + \varepsilon^*,$$

where $\mathbf{y}^* = W_1 y(\mathcal{S}^g)$ and ε^* are independent and normally distributed with variance $\phi_i^2 = \gamma_i + \tau^2$.

The *weights* for predicting the grid points from these synthetic observations (Step 4) should only incorporate the actual nugget variance for the original spatial data and can be written as

$$W_2 = K_{12}^T (K_{22} + \tau I)^{-1},$$

where K_{12} and K_{22} are defined as in Equation 3. We can now define the composition of these weights in the form of the matrix Λ:

$$\underset{M\times M}{\Lambda} = W_2 W_1.$$

Finally, we can define the predicted values on the grid from Step 4 of the conditional simulation as $\hat{\mathbf{y}}^{S*} = \Lambda y(\mathcal{S}^g) + W_2\varepsilon^*$ and $\mathbf{y}^{Sg} = y(\mathcal{S}^g)$ as the "true" process from CE. We can write

$$\hat{\mathbf{y}}^{S*} - \mathbf{y}^{Sg} = \Lambda y(\mathcal{S}^g) - y(\mathcal{S}^g) + W_2(\varepsilon_{W_1} + \varepsilon_\tau) = (\Lambda - I)y(\mathcal{S}^g) + W_2(\varepsilon^*), \tag{2}$$

and from this, we derive a formula for $\mathrm{Var}(\hat{\mathbf{y}}^{S*} - \mathbf{y}^{Sg})$ assuming $y(\mathcal{S}^g)$ and ε^* are independent. Note that this computation builds in the approximation that observations in separate grid boxes are assumed to be conditionally independent given the neighbouring grid points.

3.2 | Grid embedding approximate conditional variance

The first step of this method is to translate the off-grid observations into a surrogate problem with observations at grid locations that are nearest neighbours to the observation locations. We will assume there are $n_{\mathcal{N}}$ total number of surrogate locations. In the case of no overlapping neighbourhoods and in 2 dimensions $n_{\mathcal{N}} = n(2n_p)^2$, the number of observations multiplied by the neighbourhood size. Accordingly, let $\mathcal{J}_{\mathcal{N}}$ be an $n_{\mathcal{N}} \times M$ incidence matrix, consisting of 0 and 1 values that maps the full grid locations into just the surrogate observations on the grid. The weight function mapping observations to this subset of the grid is given by

$$W_1^{\mathcal{N}} = \mathcal{J}_{\mathcal{N}} K_{21}(K_{11} + \tau^2 I)^{-1}, \tag{3}$$

and the covariance matrix for the prediction errors associated with this mapping is the submatrix

$$\mathcal{J}_{\mathcal{N}}(K_{11} - K_{12}K_{22}^{-1}K_{21})\mathcal{J}_{\mathcal{N}}^T,$$

Lastly, let $\sigma^2_{\mathcal{N}}$ be the diagonal elements of this matrix. The synthetic observations in Step 3 of the conditional simulation follow

$$\mathbf{z}^* = \mathcal{J}_{\mathcal{N}} y(\mathcal{S}^g) + \varepsilon^*,$$

where ε^* is independent and normally distributed with variances $\sigma^2_{\mathcal{N}} + \tau^2$.

The uncertainty estimated by Monte Carlo from the approximate conditional simulation is just the conditional variances from the surrogate model, that is,

$$K_{22} - K_{22}\mathcal{J}_{\mathcal{N}}(K_{\mathcal{N}} + \tau^2 I + \Sigma_{\mathcal{N}})^{-1}\mathcal{J}_{\mathcal{N}}^T K_{22},$$

and $\Sigma_{\mathcal{N}}$ is the diagonal matrix with entries $\sigma^2_{\mathcal{N}}$.

3.3 | Analysis of the approximations to the prediction variance

In this work, covariance functions from the Matérn covariance family are considered:

$$C_\nu(d) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{d}{\theta}\right)^\nu K_\nu \left(\sqrt{2\nu}\frac{d}{\theta}\right),$$

where Γ is the gamma function, K_ν the modified Bessel function of the second kind, ν the smoothness parameter and θ the range or scale parameter. Two cases for ν are considered: (i) exponential covariance with $\nu = 1/2$ and (ii) Matérn covariance with $\nu = 3/2$. These ν values have been chosen for their closed form representation in the Matérn covariance family, as the 1/2 integer cases do not require evaluating a Bessel function.

It is assumed that the variance of the stochastic process, also known as the "sill" of the process, $\mathrm{Var}(y(\mathbf{s}_i)) = \sigma^2 = 1$. While the sill is assumed to be constant, examining various values for τ^2 (or τ) will give a sense for how the method performs according to a diverse set of ratios between the sill, σ^2, and the nugget variance, τ^2. This relationship is formalized with the parameter $\lambda = \tau^2/\sigma^2$, called the smoothing parameter. Moreover,

$1/\lambda$ can be recognized as the more conventional signal-to-noise ratio common in signal processing. We consider three nugget values $\tau^2 = 0.1^2, 0.2^2, 0.3^2$.

Three range parameters are also considered for each case of smoothness so that θ could be a short, medium or longer range parameter relative to the domain. Specifically, the range parameters are set so that for a given smoothness, the correlation between two observations falls to 5% at a distance of 20, 45 or 70 units. These values are used for their applicability to earthquake IM ranges, as seen in Loth and Baker (2013). Thus, the numerical design has three factors comprising $18 = 2 \times 3 \times 3$ separate cases. We consider a two dimension, square spatial domain with extent $[0, 60] \times [0, 60]$.

The focus of this study is on the accuracy of the prediction standard errors, and we use relative per cent error as a measure of how well the approximations work against the exact formula. For this reason, a moderate grid size is used so that the exact standard errors can be readily computed. The locations of the observations are chosen so that both the x and y coordinates are uniformly distributed on the interval [0, 60]. One-hundred different random configurations are used in this experiment to ensure that the relative errors analysed are not due to chance from a single configuration. For the numerical analysis in this section, there are 35 observations on a grid size of 61×61. The ratio of the number of observations to the size of the grid reflects the density of observations and grid for the USGS application to ShakeMap data. It is also a case where a much finer prediction grid is considered relative to the spacing of the observations. Although this is a small number of observations, these results should be consistent for larger domains with the same relative observation and grid densities.

3.4 | Numerical results

For each combination of the smoothness parameter, range parameters and configuration of observation locations the exact prediction standard error ($SE_E(\mathbf{s})$), the standard error based on the local approximation ($SE_L(\mathbf{s})$) and the standard error based on the approximate grid embedding approximation ($SE_A(\mathbf{s})$) are computed for the $61^2 = 3721$ grid locations, and we find these errors for neighbour sizes $n_p = 1, 2, 3, 4$. In particular, the relative per cent errors are defined as

$$E_L(\mathbf{s}) = 100 \times (SE_L(\mathbf{s}) - SE_E(\mathbf{s}))/SE_E(\mathbf{s}), \text{and}, E_A(\mathbf{s}) = 100 \times (SE_A(\mathbf{s}) - SE_E(\mathbf{s}))/SE_E(\mathbf{s}).$$

The results are summarized in Figure 2, by boxplots of the 95th percentile (over the grid locations) of the errors for the 100 configurations of observations. The dashed red line represents the 1% relative error line. It is clear from this graph that the approximate grid embedding method

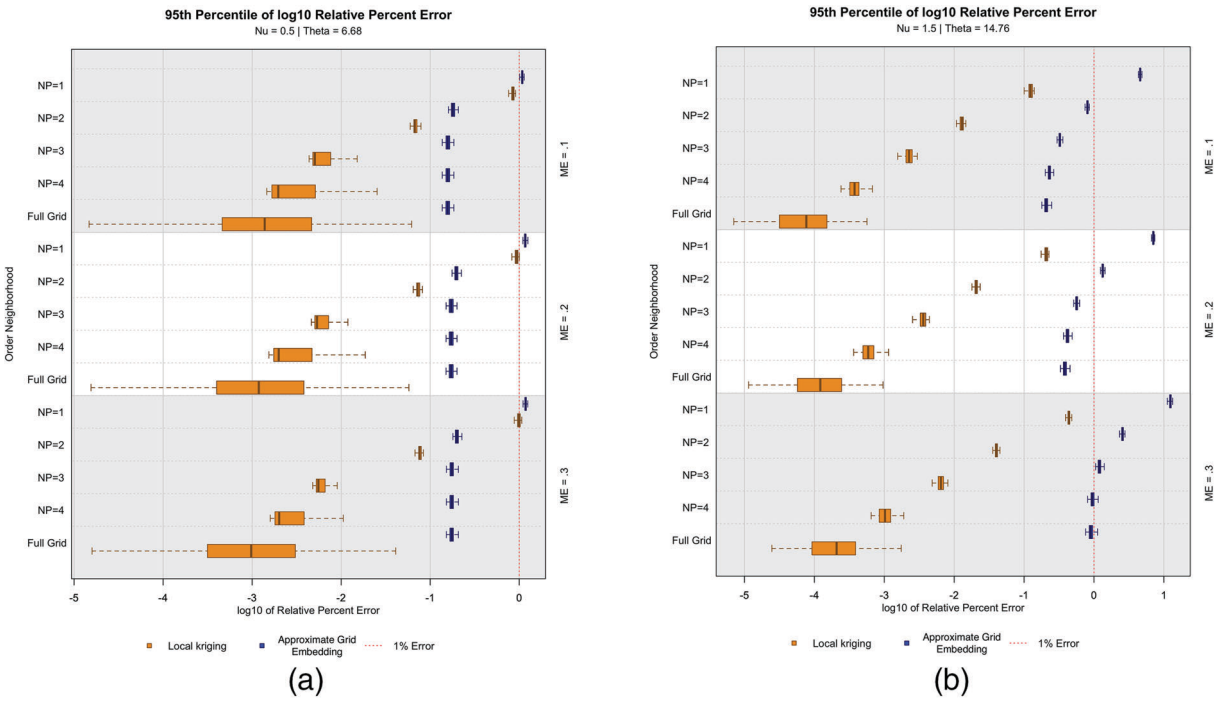

FIGURE 2 Relative per cent difference between local Kriging, approximate grid embedding and simple Kriging for $\nu = 0.5$ and $\theta = 6.68$

TABLE 1 Portion of standard errors resulting from local Kriging correct to three significant figures when compared to simple Kriging error for an exponential covariance function across 50 configurations of the grid

		$n_p = 1$		$n_p = 2$		$n_p = 3$		Full	
θ	τ	Mean	Sd	Mean	Sd	Mean	Sd	Mean	Sd
6.68	0.1	0.115	0.034	0.736	0.020	0.974	0.009	0.994	0.009
	0.2	0.117	0.035	0.746	0.020	0.974	0.012	0.993	0.012
	0.3	0.123	0.037	0.759	0.019	0.974	0.013	0.993	0.013
15.02	0.1	0.066	0.019	0.741	0.018	0.970	0.010	0.994	0.010
	0.2	0.070	0.021	0.756	0.018	0.970	0.013	0.993	0.013
	0.3	0.074	0.022	0.773	0.019	0.972	0.014	0.993	0.013
23.37	0.1	0.069	0.019	0.777	0.015	0.973	0.01	0.995	0.010
	0.2	0.075	0.022	0.791	0.017	0.973	0.013	0.994	0.012
	0.3	0.085	0.025	0.812	0.017	0.974	0.014	0.993	0 .013

TABLE 2 Portion of standard errors resulting from approximate grid embedding correct to 3 significant figures when compared to simple Kriging error for an exponential covariance function across 50 configurations of the grid

		$n_p = 1$		$n_p = 2$		$n_p = 3$		Full	
θ	τ	Mean	SD	Mean	SD	Mean	SD	Mean	SD
6.68	0.1	0.228	0.036	0.896	0.003	0.917	0.003	0.918	0.003
	0.2	0.215	0.035	0.895	0.003	0.915	0.003	0.918	0.003
	0.3	0.211	0.036	0.896	0.004	0.916	0.003	0.919	0.003
15.02	0.1	0.006	0.024	0.840	0.006	0.901	0.003	0.904	0.004
	0.2	0.003	0.019	0.830	0.006	0.900	0.004	0.902	0.003
	0.3	0.074	0.016	0.773	0.007	0.972	0.004	0.993	0.004
23.37	0.1	0.019	0.015	0.785	0.009	0.889	0.004	0.894	0.004
	0.2	0.007	0.009	0.759	0.010	0.884	0.005	0.890	0.005
	0.3	0.004	0.006	0.738	0.009	0.882	0.004	0.890	0.004

Note: Values of θ are chosen so that the correlation between two observations falls to 5% at a distance of 20, 45 or 70 units, which has applicability to earthquake IM ranges, as seen in (Loth & Baker, 2013)

does not improve after an order neighbourhood of 3. On the other hand, $E_L(\mathbf{s})$ appears to decrease with n_p but exhibits more variability. Regardless of these differences, for both methods, the error quantiles tend to remain under 1%.

Another way to score the closeness of the approximations to the exact prediction variances is by quantifying their agreement to a specific number of significant digits. This comparison is motivated by a prediction confidence interval being computed to a practical accuracy. For example, the width is accurate to three digits or about .1% Table 1 shows the per cent of local standard errors that are correct to three significant figures when compared to the exact prediction standard error for an exponential covariance. Table 2 shows the same but for the approximate grid embedding method. Note that, even when using a full grid for local Kriging, there is still 1 configuration that is not correct to three significant figures. Table 2 shows that approximate grid embedding method does not do as well even when a full grid is used.

3.5 | Model misspecification

A key assumption in these approximate methods is that the off-grid observations are simulated separately and the prediction errors added to the off-grid simulated values are assumed to be independent from one another. This approach is very convenient for parallel computation and in many cases can be assumed due to the fine resolution of the simulated process on the grid by circulant embedding. In this section, we give some justification for assuming independence that we believe is tied to the screening effect for fine grids that separate observations. We consider a grid with unit spacing, second-order neighbours and two observations that are 1/2 and 1 unit apart. We position these observation pairs to give the largest conditional correlations, as seen in Figure 3. The conditional covariance is found for the conditional distribution of these two pairs for the Matérn

family of covariances. We vary the range parameter in the interval [0.2, 10] and the smoothness in the interval [0.5, 1.5] to obtain a maximum conditional correlation of 0.181 for one unit spacing (red circles in Figure 3) and 0.293 for half unit spacing (grey squares in Figure 3). These results suggest that it is important to have the grid separate observation (off-grid) locations; however, the screening effect greatly reduces the correlation for locations centred in adjacent grid boxes. As might be expected, these correlations increase somewhat as the smoothness of the Matérn family is increased.

3.6 | Fast computation

The basic off-grid prediction involves a small linear combination of neighbouring grid values for each off-grid location. This can be easily converted to a parallel operation by distributing the off-grid computation to multiple cores and tiling the grid locations to avoid passing the entire gridded field to each core. In this study, however, a serial approach was taken to assess how a simpler coding of the algorithm would perform. The basic idea is to rephrase the off-grid prediction in terms of a sparse matrix multiplication of the gridded field value and to leverage efficiency from the `spam` sparse matrix package in R.

Recall that the off-grid prediction involves a set of weights based on the nearest neighbour grid points. This can be divided up into two matrices. Recall K_{12}^{j} is the cross covariance matrix between the nearest neighbours and the off-grid location $\mathbf{s}_j$. We note that for regular grids finding nearest neighbours and their indices is particularly fast because it is equivalent to finding the nearest integer values to real numbers. K_{22}^{j} is the covariance matrix for the n_p order nearest neighbours. Since we are assuming a stationary covariance function, this matrix is the same for every observation location. By careful indexing, one can populate the rows of a *sparse* matrix version of W_1 using as rows $K_{12}^{j}(K_{22}^{j})^{-1}$ such that

$$\hat{y}(\mathcal{S}) = W_1 y(\mathcal{S}^g).$$

Here, the off-grid predictions, also the conditional mean values, are obtained by a single sparse matrix multiplication applied to the gridded field values unrolled as a vector. The advantage of this approach is that it exploits the efficient multiplication and indexing across large vectors.

This setup makes a few important assumptions. First, the same neighbourhood structure will be used around each off-grid point regardless of where the off-grid point is within a grid box. That is, a point that is directly in the middle of a grid box will have the same neighbourhood pattern as a point that is close to the corner of the grid box. This method also assumes the grid extends several grid points beyond all observations, and so padding the edge of the grid may be necessary if any observations lay too close to the edge of the domain. Finally, there is the assumption of stationarity so that the covariance matrix of nearest neighbours (K_{22}^{j}) can be inverted once and applied for all off-grid predictions. Although the nominal dimensions of a sparse W_1 can be large, there are only $(2n_p)^2$ nonzero entries per row, and so the multiplication is quite fast. Also note that for generating many realizations of the grid and off-grid points, W_1 needs to be created only once and stored. To generate the measurement/nugget errors, one can phrase this a multiplication of the measurement standard deviations by iid standard normal random variables. In the extension where a limited number of grid boxes have a small number of observations, this step can also be accomplished as sparse matrix multiplication. In the case the sparse matrix has diagonal elements ϕ_i for single observations in a grid box, for multiple observations in a grid box, there are diagonal blocks with the Cholesky decomposition of the conditional covariance matrix.

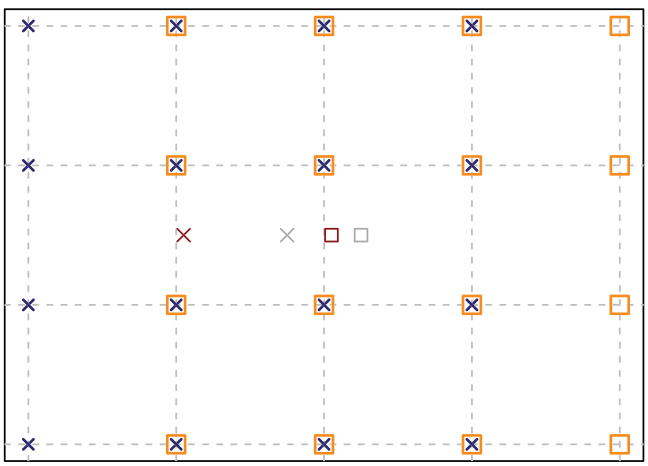

FIGURE 3 Location of grid and pairs of off-grid points for testing. Both sets of points have a second-order neighbourhood plotted where hollow orange squares correspond to the right red and grey symbols while the dark blue X's to the left red and grey symbols

This algorithm has been implemented in serial in the `fields` R package and has surprisingly fast performance. Our timing is done on a Macbook Pro laptop with 2.3 GHz Quad-Core Intel Core i5 using `R 4.1.0`, `fields 12.6` and `spam 2.7-0`. Although we expected the sparse methods to be efficient, the speedup over a conventional for loop in R was surprising. Creating W_1 and $VAR(\mathcal{S})$ took under 0.15 seconds for a 2,048 $\times$ 2,048 grid and for 6,400 off-grid observations, and $n_p = 4$. The simulation step 3 for the off-grid values took under 0.01 s.

To set these results in context, it is also useful to time our proposed conditional simulation algorithm. As above, we use a serial implementation based on easy to use and high level functions from the fields package, and we break up the timing into several key parts. The results are reported in Table 3 for $n_p = 4$. Here, $M = m^2$ is the total grid size, and n is the number of off-grid locations. The off-grid locations are drawn from a uniform distribution on the spatial domain. For these computations, we used an exponential covariance where the evaluation of the covariance matrix has been optimized somewhat using a lower level C subroutine called from R. Times are in seconds with `CESetup` being the time to setup computations for the circulant embedding, `OffSetup` setup time for the off-grid predictions, `CE` time for simulating a single realization on the grid, `OffGrid` time to simulate a realization of the off-grid values, `predict` the time to predict from the off-grid observations onto the regular grid and, finally, `draw` the total time to generate a single draw for this algorithm. Thus, generating 100 members from the conditional distribution on a 128 $\times$ 128 grid and for 400 observations will take on the order of about a minute (100 $\times$ 0.61 = 61 s). We did not time the exact computation because this would involve a Cholesky decomposition of the full covariance matrix of size 16,784 $\times$ 16,784. For the largest case, the time for a 100 member ensemble will be substantial, amounting to about 2 h. Note, however, that the time is dominated by how long it takes to compute the spatial process predictions from n locations to the $m \times m$ grid. The time in generating the unconditional field, the subject of this work, is negligible compared to the prediction step.

We compare this algorithm to the existing Vecchia approximation and related algorithms in the two R packages GpGp and spNNGP. The functions within the GpGp package are quick, accessible methods for conditional simulation of a spatial process. A short simulation study comparing the `cond_sim` function to our algorithm shows that our algorithm produces a distinct speedup for all grid sizes and the number of observations considered in Table 3. For example, on a 512 $\times$ 512 grid with 400 observations, we saw that a single draw from the conditional distribution takes 4.63 s while the GpGp function takes 52.85 s. Across all grids and observations considered, there is a median speed up of 3.4. It is important to note that this draw includes the prediction to the full grid which, as we saw in Table 3, contributes significantly to the total draw time. Timing results for the spNNGP are impressive, and we observe the conditional simulation for a 512 $\times$ 512 grid and 6,400 observations to be under 4 s.

Finally, we note an approximation to our approach that gives significant speedup, comparable to spNNGP. We have identified the prediction step as the bottleneck and approximate exact linear algebra with covariances restricted to the regular grid. This allows for a fast multiplication using the fast Fourier transform. We found this approximation is accurate, and for the 512 $\times$ 512 grid with 6,400 observations, we find a timing of 1.5 s for setup and under 0.5 s to generate a realization.

4 | APPLICATION TO RIDGECREST, CA, EARTHQUAKE

Earthquakes are one of the most devastating natural disasters and are the motivation for this work. After any significant earthquake, economic and fatality loss estimates are extremely important for local government officials to coordinate aid. Pinpointing the local variation in damage and loss in a timely manner is important. Therefore, to help prepare local governments, first responders, or utility companies, near real-time estimates for the intensity of shaking at specific locations are essential. Most of the seismic loss problems typically depend on the regional distribution of intensity measures (IM), rather than intensity only at a single site.

Quantifying ground-motion over a spatially distributed region requires information on the correlation between the ground-motion intensities at different sites during a single event (Loth & Baker, 2013). To this end, several correlation models have been developed to describe the decay in

TABLE 3 Timing results for the complete conditional simulation algorithm

m	n	CESetup	OffSetup	CE	OffGrid	predict	total
128	400	0.02	0.01	0.02	0.00	0.24	0.61
128	1,600	0.01	0.03	0.02	0.00	0.93	1.46
128	6,400	0.02	0.14	0.02	0.00	3.67	4.79
256	400	0.07	0.01	0.07	0.00	0.95	1.53
256	1,600	0.07	0.03	0.07	0.00	3.70	4.86
256	6,400	0.07	0.12	0.07	0.00	14.52	17.97
512	400	0.39	0.01	0.53	0.00	3.73	5.40
512	1,600	0.49	0.06	0.40	0.00	14.76	18.67
512	6,400	0.44	0.12	0.48	0.01	58.65	71.13

correlation from intensities at one site to another with increasing separation distance. However, for a large-scale application of ShakeMap, computing the spatially correlated field using classical spatial methods is generally memory intensive and computationally expensive. Therefore, it is necessary to improve the speed and accuracy of simulating IMs across an area affected by an earthquake.

To demonstrate local Kriging with observed shaking data, the algorithm is implemented using IMs observed during the 2019 Ridgecrest, CA earthquake. Here, the Ridgecrest earthquake is considered for the higher availability of station data. Observed peak ground acceleration (PGA) is normalized based on the ShakeMap Manual (Worden et al. 2018, 2020). In total, 637 observed values for PGA from 633 stations are used. For stations where there are multiple recordings of PGA, the mean across the observed values is taken resulting in 633 total observed values for the event. In order for the local Kriging to be fully implemented in this example, several updates to the simulation grid had to be made. The resolution of the grid was increased fourfold, and any stations that still remained in the same box after increasing the resolution were averaged over the same grid box resulting in a single observation per grid box. The final observation count was 623, and the final prediction grid has 275,394 locations. The observations are distributed throughout southern California as seen by the white circles in Figure 4. The left map in Figure 4 shows the final mean estimate across an ensemble of 50 realizations of the field, and the right map shows the estimated standard error of the realizations. The standard error is lowest near the observation values and increases as the distance to observations increases, as expected.

Timing of this algorithm was done on a MacBook Pro with 2.6 GHz 6-Core Intel Core i7 with 16GB of memory. Because of the increased resolution of the simulation grid, setting up the circulant matrix for simulation took a majority of the simulation time—approximately 42 s. However, within each run of the simulation, the unconditional simulation step took a median value of 28.6 s, while the off-grid prediction step only took .03 s. Finally, the prediction to the grid took approximately 6 s. Thus, it is clear that for larger grids where an increased resolution for the simulation grid is needed, the time for the circulant embedding setup may increase substantially. However, the off-grid prediction step remains an extremely efficient calculation due to the use of sparse linear algebra.

5 | DISCUSSION

We have established an approximate method for conditional simulation based on local Kriging that is accurate for data and conditional variance analysis and also fast enough to fit into an interactive data analysis session. A serial implementation is straightforward and easily incorporated into existing R packages for spatial data analysis. Moreover, faster prediction methods, including Vecchia-type approximations, can significantly speedup these computations. Based on the numerical results and timing, it is recommended that a fourth-order neighbourhood be used for local Kriging for both an exponential and Matérn with $\nu = 1.5$. Additionally, while both methods provide relative errors compared to exact Kriging that are within 1%, it is recommended that the local Kriging method be used. Sparse matrix methods provide significant speedup for the local Kriging method, and it is suggested that this be used over previous methods for generating ensemble members. A significant portion of time for the entire conditional simulation algorithm is dedicated to grid prediction. However, the methods developed here address reducing time for the unconditional simulation and prediction to the off-grid points, which was successful. Future work could focus on implementing fast grid-prediction methods to further improve the conditional simulation. For example, covariance functions such as the Wendland family have compact support that can be exploited for faster prediction. Some other approaches are to use fixed rank Kriging type predictions or develop approximations to the large covariance matrices needed for prediction using hierarchical matrix decompositions.

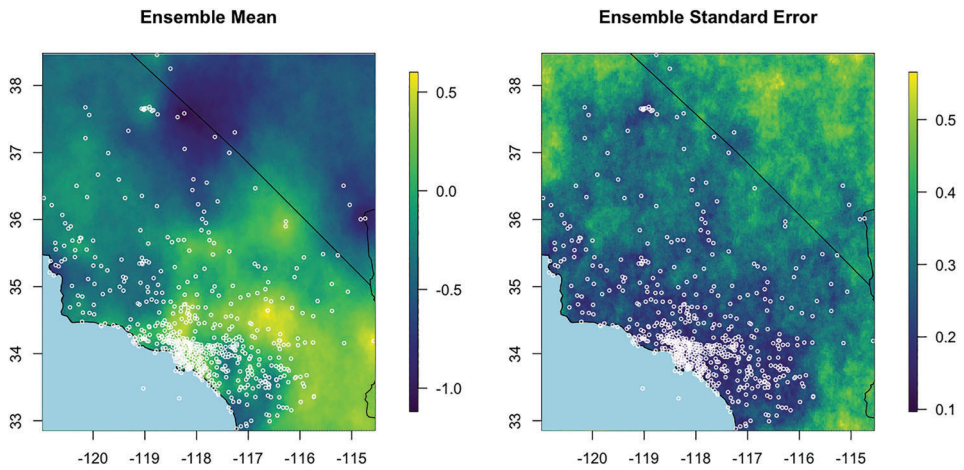

FIGURE 4 The mean field (left) across an ensemble of 50 for the Ridgecrest, CA earthquake using an order neighbourhood of 3. The standard error (right) shows lower error around observed values and increased error as distance from an observation increases

Future work could be to implement this method in three dimensions for use in atmospheric or subsurface modelling. A second area of further research is to extend this analysis to an anisotropic covariance function as CE can be used for anisotropic simulation. In particular, anisotropic processes could improve estimation where earthquake wave propagation is not uniform.

ACKNOWLEDGEMENTS

This research was supported in part by the Colorado School of Mines Faculty Development Funds and National Science Foundation award DMS-1811384.

ORCID

Soutir Bandyopadhyay https://orcid.org/0000-0003-2213-3333

Douglas Nychka https://orcid.org/0000-0003-1387-3356

REFERENCES

Chan, G., & Wood, A. T. A. (1999). Simulation of stationary Gaussian vector fields. *Statistics and Computing*, *9*, 265–268.

Cressie, N. A. C. (1993). *Statistics for Spatial Data*: Wiley. revised edition.

Datta, A., Banerjee, S., Finley, A. O., & Gelfand, A. E. (2016). Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, *111*(514), 800–812.

Finley, A. O., Datta, A., Cook, B. D., Morton, D. C., Andersen, H. E., & Banerjee, S. (2019). Efficient algorithms for Bayesian nearest neighbor Gaussian processes. *Journal of Computational and Graphical Statistics*, *28*(2), 401–414.

Furrer, R., Nychka, D., Sain, S., & Nychka, M. D. (2009). Package 'fields'. R Foundation for Statistical Computing, Vienna, Austria. http://www.idg.pl/mirrors/CRAN/web/packages/fields/fields.pdf (last accessed 22 December 2012).

Katzfuss, M., Guinness, J., Gong, W., & Zilber, D. (2020). Vecchia approximations of Gaussian-process predictions. *Journal of Agricultural, Biological and Environmental Statistics*, *25*(3), 383–414.

Loth, C., & Baker, J. W. (2013). A spatial cross-correlation model of spectral accelerations at multiple periods. *Earthquake Engineering & Structural Dynamics*, *42*(3), 397–417.

Park, M. H., & Tretyakov, M. V. (2015). A block circulant embedding method for simulation of stationary Gaussian random fields on block-regular grids. *International Journal for Uncertainty Quantification*, *5*(6), 527–544.

Stroud, J. R., Stein, M. L., & Lysen, S. (2017). Bayesian and maximum likelihood estimation for Gaussian processes on an incomplete lattice. *Journal of computational and Graphical Statistics*, *26*(1), 108–120.

Verros, S. A., Wald, D. J., Worden, C. B., Hearne, M., & Ganesh, M. (2017). Computing spatial correlation of ground motion intensities for ShakeMap. *Computers & Geosciences*, *99*, 145–154.

Wood A. T., & Chan G. (1994). Simulation of stationary gaussian processes in $[0, 1]^d$. *Journal of Computational and Graphical Statistics*, *3*(4), 409–432.

Worden, C. B., Thompson, E. M., Baker, J. W., Bradley, B. A., Luco, N., & Wald, D. J. (2018). Spatial and spectral interpolation of ground-motion intensity measure observations. *Bulletin of the Seismological Society of America*, *108*(2), 866–875.

Worden, C. B., Thompson, E. M., Hearne, M. G., & Wald, D. J. (2020). ShakeMap Manual. Technical Manual, users guide, and software guide Version.

How to cite this article: Bailey, M. D., Bandyopadhyay, S., & Nychka, D. (2022). Adapting conditional simulation using circulant embedding for irregularly spaced spatial data. *Stat*, *11*(1), e446. https://doi.org/10.1002/sta4.446