



# GStatSim V1.0: a Python package for geostatistical interpolation and conditional simulation

Emma J. MacKie<sup>1</sup>, Michael Field<sup>1</sup>, Lijing Wang<sup>2,3</sup>, Zhen Yin<sup>2</sup>, Nathan Schoedl<sup>4,5,6</sup>, Matthew Hibbs<sup>4</sup>, and Allan Zhang<sup>6</sup>

<sup>1</sup>Department of Geological Sciences, University of Florida, Gainesville, FL 32611, USA

<sup>2</sup>Department of Earth and Planetary Sciences, Stanford University, Stanford, CA 94305, USA

<sup>3</sup>Earth and Environmental Sciences Area, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

<sup>4</sup>Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611, USA

<sup>5</sup>Department of Mathematics, University of Florida, Gainesville, FL 32611, USA

<sup>6</sup>Department of Statistics, University of Florida, Gainesville, FL 32611, USA

**Correspondence:** Emma J. MacKie (emackie@ufl.edu)

Received: 6 November 2022 – Discussion started: 18 November 2022

Revised: 4 May 2023 – Accepted: 31 May 2023 – Published: 6 July 2023

**Abstract.** The interpolation of geospatial phenomena is a common problem in Earth science applications that can be addressed with geostatistics, where spatial correlations are used to constrain interpolations. In certain applications, it can be particularly useful to perform geostatistical simulation, which is used to generate multiple non-unique realizations that reproduce the variability in measurements and are constrained by observations. Despite the broad utility of this approach, there are few open-access geostatistical simulation software applications. To address this accessibility issue, we present GStatSim, a Python package for performing geostatistical interpolation and simulation. GStatSim is distinct from previous geostatistical tools in that it emphasizes accessibility for non-experts, geostatistical simulation, and applicability to remote sensing data sets. It includes tools for performing non-stationary simulations and interpolations with secondary constraints. This package is accompanied by a Jupyter Book with user tutorials and background information on different interpolation methods. These resources are intended to significantly lower the technological barrier to using geostatistics and encourage the use of geostatistics in a wider range of applications. We demonstrate the different functionalities of this tool for the interpolation of subglacial topography measurements in Greenland.

## 1 Introduction

The interpolation of geological and geophysical observations is a common problem in the geosciences with applications in mineral exploration (Journel and Huijbregts, 1976; Emery and Maleki, 2019), oil reservoir modeling (Kelkar and Perez, 2002; Pyrcz and Deutsch, 2014), groundwater hydrology (Kitanidis, 1997; Feyen and Caers, 2006), soil sciences (Goovaerts, 1999; Lark, 2012; Xiong et al., 2015), climate modeling (Costa and Soares, 2009), natural hazard prediction (Youngman and Stephenson, 2016), and glaciology (MacKie et al., 2020; Zuo et al., 2020). The field of geostatistics emerged in the 1950s with the development of kriging, a method for optimizing spatial interpolation, which was originally used to estimate gold ore grades (Krig, 1951; Cressie, 1990). This theory was expanded by Matheron (1963), who formalized the variogram as a measure of spatial variance. Since then, a number of methods have evolved to include a variety of interpolators and descriptors of spatial statistics (e.g., Cressie and Hawkins, 1980; Solow, 1986; Goovaerts, 1998; Strebelle, 2002).

In contrast to deterministic methods such as kriging, which are designed to minimize estimation variance, geostatistical simulation is designed to reproduce the variability in measurements (Deutsch and Journel, 1992). Geostatistical simulation involves the generation of many stochastic realizations of a Gaussian random field that retains the spatial statistics of observations. These realizations are conditioned to observed

data, meaning they exactly match observations. The ensemble of realizations quantifies uncertainty. This approach is often used to account for subsurface heterogeneity when modeling petroleum reservoirs (Pyrz and Deutsch, 2014), performing geophysical inversions (Nunes et al., 2012; Shamsipour et al., 2010; Volkova and Merkulov, 2019), and modeling groundwater hydrology (Feyen and Caers, 2006).

Despite this wide range of geological applications, many existing geostatistical modeling software applications are primarily intended for use in oil and mineral exploration applications. For example, T-PROGS (Carle, 1999) is intended for use with borehole data, and the Leapfrog geological modeling software is used almost exclusively in industry. The Geostatistics Software Library (GSLIB; Deutsch and Journel, 1992) test cases are predominantly devoted to modeling porosity and permeability for oil reservoir modeling. While these tools were primarily tested on point measurements such as borehole data, many scientific applications now rely on large-scale airborne geophysical data sets, which typically consist of cross-cutting line surveys that are densely sampled in the along-track direction with large gaps in between survey lines. This data configuration can prove challenging for geostatistical interpolation. Furthermore, geophysical surveys, and remote sensing methods in particular, can generate extremely large volumes of data. This can be problematic for certain software, such as the Stanford Geostatistical Modeling Software (SGeMS; Remy, 2005) which becomes slow to operate and sometimes crashes when users upload large files. As such, many geostatistical software applications are not directly suited to the scale and nature of certain geophysical data interpolation problems.

The commercial nature of many geostatistical applications and software further restricts the use of geostatistics in academic and educational settings. Many geostatistical modeling software applications are locked behind a pay-wall. For example, Leapfrog and T-PROGS are proprietary and cost thousands of dollars per year (e.g., GMS, 2021). Leapfrog and SGeMS only work on Windows operating systems. GSLIB is written in FORTRAN, which has a limited user base compared to Python. These interoperability issues make it difficult to integrate geostatistical simulation with existing scientific workflows. Furthermore, many of these software applications are intended for use by trained experts and require the purchase of a companion textbook to learn the documentation (e.g., Deutsch and Journel, 1992; Remy et al., 2009). These accessibility issues create a barrier to producing open-access, reproducible workflows and fail to meet modern accessibility standards such as the FAIR principles (Findable, Accessible, Interoperable, Reusable; Wilkinson et al., 2016). Moreover, the restricted availability of geostatistical software limits its use in educational settings. As such, developments in open-access, user-friendly tools are critically needed for advancing the use of geostatistical simulation in open science and education.

Recently, a growing number of open-source geostatistical modeling tools have become available. This includes the *gstat* package in R (Pebesma, 2004) and the Python packages *PyKrig* (Murphy, 2014), *GeostatsPy* (Pyrz et al., 2021), *SciKit-GStat* (Mälicke, 2022), and *GSTools* (Müller et al., 2022). *PyKrig* provides tools for performing kriging interpolation. *SciKit-GStat* offers variogram modeling functions. *GeostatsPy* is a Python wrapper of some of the *GSLIB* functions. *GSTools* provides a number of useful functions including tools for time-series analysis, data transformations, and interpolation. While these resources are important steps towards providing the scientific community with accessible geostatistical software and have significantly expanded the repertoire of geostatistical functions in Python, additional tools are needed for performing geostatistical simulation. In particular, existing Python tools are limited in their ability to accommodate non-stationary or variations in spatial statistics, incorporate secondary constraints, and perform interpolations with large conditioning data sets.

To address the aforementioned needs, we present *GStatSim*, a Python package for performing geostatistical interpolations and simulations. *GStatSim* enables the user to perform a variety of deterministic and stochastic interpolations in Python including various versions of kriging and sequential Gaussian simulation (SGS). *GStatSim* is intended to complement existing Python and geostatistical frameworks. While *GStatSim* does not include variogram modeling tools, it is designed to be compatible with existing variogram modeling packages such as *SciKit-GStat*, a robust variogram modeling toolkit with extensive documentation. In contrast to previous tools, *GStatSim* emphasizes interpolation, particularly conditional simulation. We provide new tools for accommodating non-stationarity or variability in spatial statistics. This package is intended to strike a balance between flexibility and ease of use. The tools are sufficiently deconstructed to allow for a variety of interpolation workflows while also limiting the number of parameter selections that must be made. In contrast to previous software applications that are tested on point data sets, *GStatSim* was constructed with geophysical line data sets specifically in mind.

In addition to providing tools for scientific analyses, *GStatSim* is also intended to be an educational resource. We developed a Jupyter Book that contains a series of tutorials for reproducing the figures in this paper. These tutorials provide intuition on the different algorithms and guidance on parameter selection. This is intended to make *GStatSim* accessible to non-experts and reduce the amount of time needed to develop models.

In this paper, we demonstrate the different functionalities of *GStatSim* (Tables 1 and 2) and compare it to existing packages such as *GSTools*. We also provide a brief description of the underlying theory behind each function. The interpolation functions are demonstrated for the interpolation

of ice-sheet bed elevation data measured by airborne ice-penetrating radar. Our goal is not to recommend a specific interpolation method for this case study; rather, we describe the implementation considerations and illustrate the advantages and limitations of each interpolation method. We provide an overview of the available user support and documentation. Finally, we discuss possible future directions for further enabling the widespread utilization of geostatistical simulation in academic disciplines.

## 2 Data

We apply the `GStatSim` functions to the interpolation of ice-penetrating radar measurements of subglacial topography in Greenland. Subglacial topography is a critical parameter in ice-sheet models (Parizek et al., 2013; Seroussi et al., 2017). For instance, modeled ice flow behavior and subglacial water routing models can be highly sensitive to the method used to interpolate measurements (MacKie et al., 2021; Wernecke et al., 2022; Law et al., 2023). We use bed elevation measurements from the Center for the Remote Sensing of Ice Sheets (CReSIS, 2022) over a  $150\text{ km} \times 150\text{ km}$  region in northwest Greenland (Fig. 1). This location was chosen for the variability in line spacing (10–50 km), abundant crossover points, irregular survey orientations, and the presence of a large-scale trend. These conditions pose a challenge to interpolation, making this data set a good case study for testing the rigor of the `GStatSim` algorithms. For the multivariate applications we use ice surface elevation data from ArcticDEM (Porter et al., 2018) as a secondary source of information for constraining interpolations. We use the 500 m resolution version. The surface and bed elevation coordinates are transformed from geographic coordinates to polar stereographic coordinates using the National Snow and Ice Data Center (NSIDC) coordinate transformation tool (Torrence, 2019) so that Euclidean distances can be determined.

## 3 Interpolation with existing geostatistical tools

It was our original intent to test existing Python interpolation functions from `PyKrig`, `GSTools`, and `GeostatsPy` on the aforementioned bed elevation data set in order to establish a baseline level of performance for comparison with `GStatSim`. However, the `PyKrig` and `GSTools` kriging and conditional simulation routines crash when applied to our data set due to the high random access memory (RAM) consumption. We are also unable to produce a simulation result using `GeostatsPy` due to a “singular matrix” error, which generally indicates that there is some sort of numerical instability. We discuss these limitations in more detail in Sect. 7. Only the `GeostatsPy` kriging function is successfully applied to this data set. The `GeostatsPy` kriging implementation and results are described in Sect. 4.3.3.

## 4 GStatSim functions and implementation

`GStatSim` can be used with Python 3.0–3.10 and relies on commonly used packages such as `scipy` (Virtanen et al., 2020), `numpy` (Harris et al., 2020), `pandas` (McKinney, 2010), and `scikit-learn` (Pedregosa et al., 2011). This ensures that `GStatSim` is easy to use in different computing environments. The source code is written in accordance with PEP8 standards (Van Rossum et al., 2001), making it interpretable and easy to modify.

### 4.1 Data initialization

Prior to performing any geostatistical analysis, we fit the bed elevation data to a 1 km resolution grid using the `GStatSim` `grid_data` function which averages values within each grid cell. We do this because the interpolation examples shown in this paper use a 1 km resolution. In this way, the spatial statistics of the data will be compatible with the interpolation grid. This gridded data set has 3650 data points. In order to satisfy the Gaussian assumptions of our interpolation functions, we convert our data to a standard Gaussian distribution using the `scikit-learn` `QuantileTransformer` function. This step converts the data to a standard Gaussian distribution (mean = 0, variance = 1), shown in Fig. 2. It is not strictly necessary for data to be Gaussian when using `GStatSim`. However, the interpolation results will be more robust if a Gaussian distribution is used.

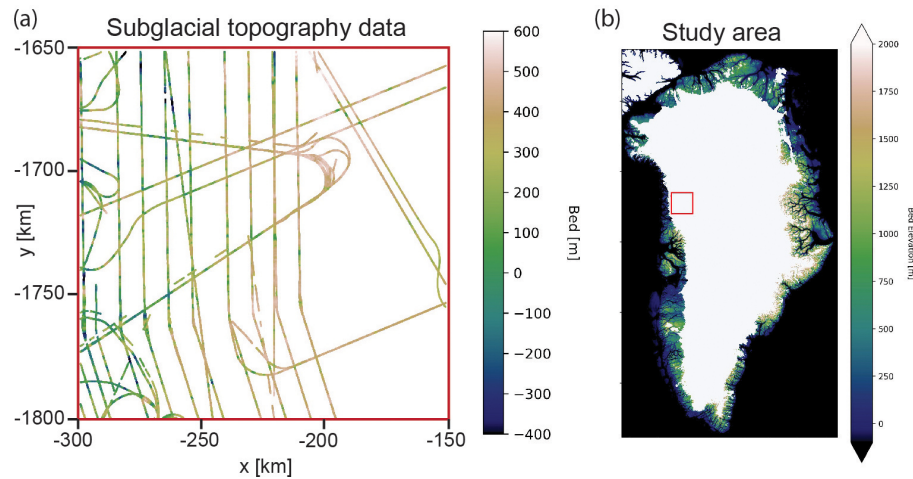
### 4.2 Variogram calculation and modeling with SciKit-GStat

The interpolation functions in `GStatSim` rely on the variogram, which describes spatial statistics by quantifying the covariance between data points as a function of their separation distance or lag (Matheron, 1963). The experimental variogram  $\gamma(h)$  is defined as the averaged squared differences of data points separated by a lag distance  $h$ :

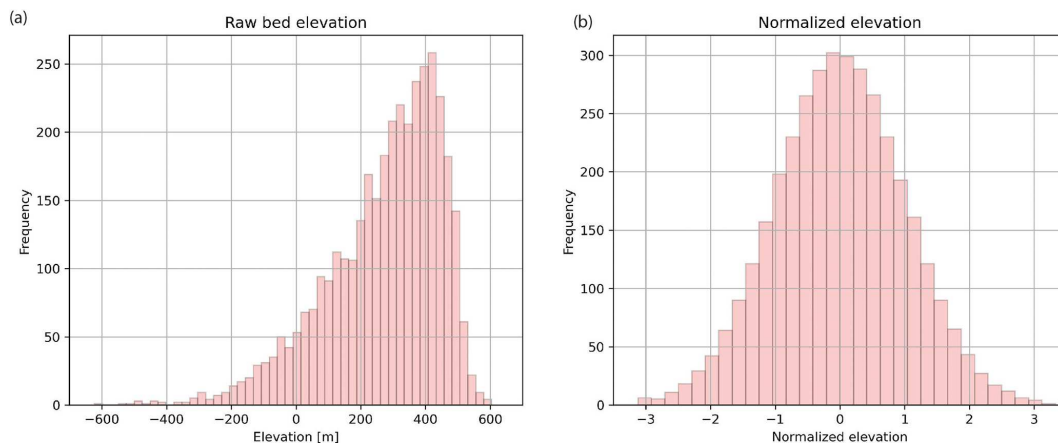
$$\gamma(h) = \frac{1}{2N(h)} \sum_{\alpha=1}^N (Z(u_{\alpha}) - Z(u_{\alpha} + h))^2, \quad (1)$$

where  $Z$  is a variable,  $u$  is a spatial location, and  $N(h)$  is the number of pairs of data points for a lag  $h$ . Typically, the variance increases with lag distance until a threshold is reached where measurements are no longer spatially correlated. This threshold is called the *sill*, which is, in theory, equal to 1 for normalized data sets. The lag distance where the variogram reaches the sill is called the *range*.

`GStatSim` does not include variogram estimation tools as there are already existing resources (e.g., `SciKit-GStat`). Our interpolation functions are designed to be integrated with outside variogram modeling tools. We perform our variogram analysis with the `SciKit-GStat` functions, which offer robust tools



**Figure 1.** Ice-penetrating radar topography data for test region. The coordinates are shown in a polar stereographic projection.



**Figure 2.** (a) Original data distribution. (b) Transformed data with a standard Gaussian distribution.

for variogram estimation and modeling. We compute both the isotropic and anisotropic variograms. The anisotropic variogram computes the variogram for different angles from horizontal. Figure 3 shows that the data do not contain significant anisotropy. While the anisotropic variograms do exhibit some disagreement at lags greater than  $\sim 35$  km, these differences can be attributed to regional trends in the topography. Additionally, the first part of the variogram with small lags is more important than the tail distributions when it comes to calculating the kriging weights used for interpolation. As such, only the isotropic variogram is used for the variogram modeling analysis.

In order to use the variogram for interpolation, a variogram model must be fit to the experimental variogram. GStatSim can currently accommodate exponential, Gaussian, and spherical variogram model types. The exponential model is as follows:

$$\gamma_{\text{exp}}(h) = \begin{cases} 0 & \text{if } h = 0 \\ b + (C_0 - b) \left[ 1 - e^{-\frac{3h}{a}} \right] & \text{if } h > 0 \end{cases}, \quad (2)$$

where  $a$  is the range parameter,  $C_0$  is the sill, and  $b$  is the nugget. The nugget describes the variance at near-zero lags, which is often the result of measurement error. In most cases, the nugget is zero. The Gaussian model is

$$\gamma_{\text{Gauss}}(h) = \begin{cases} 0 & \text{if } h = 0 \\ b + (C_0 - b) \left[ 1 - e^{-\frac{3h^2}{a^2}} \right] & \text{if } h > 0 \end{cases}, \quad (3)$$

and the spherical model is

$$\gamma_{\text{sph}}(h) = \begin{cases} 0 & \text{if } h = 0 \\ b + (C_0 - b) \left( 1.5 \cdot \frac{h}{a} - 0.5 \cdot \frac{h^3}{a^3} \right) & \text{if } 0 < h \leq a \\ C_0 & \text{if } h > a \end{cases}. \quad (4)$$

See Chiles and Delfiner (2009) for a detailed description of these models. In practice, variogram model parameters are often selected manually, though SciKit-GStat also offers automatic variogram modeling tools. Using these tools,



**Table 1.** GStatSim interpolation functions.

Function	Description
skrige	Simple kriging estimator
okrige	Ordinary kriging estimator
skrige_sgs	Sequential Gaussian simulation using simple kriging
okrige_sgs	Sequential Gaussian simulation using ordinary kriging
cluster_sgs	Sequential Gaussian simulation using simple kriging, where different variograms are defined in different regions
cokrige_mm1	Co-kriging under Markov model 1 (MM1) assumptions, uses secondary data as a constraint
cosim_mm1	Co-simulation under MM1 assumptions, uses secondary data as a constraint

**Table 2.** GStatSim accessory functions.

Function	Description
grid_data	Fits conditioning data to a grid of some specified resolution
prediction_grid	Makes an array of coordinates to be interpolated based on bounding coordinates and resolution
rbf_trend	Estimates the large-scale trend of the conditioning data using a smoothed radial basis function
adaptive_partitioning	Breaks conditioning data into blocks based on the density of measurements
find_colocated	Finds the co-located primary and secondary data points for performing co-kriging and co-simulation

we fit each of these model types to the experimental variogram, shown in Fig. 3a. A visual inspection of Fig. 3a shows that the exponential variogram model produces the best fit. As such, we use this model type for the topography interpolations shown in this paper. The exponential model has a nugget of 0, a range of 31.9 km, and a sill of 0.7. Geostatistical interpolations can be sensitive to the methods and parameters used to compute and model variograms. As such, we recommend referring to the SciKit-GStat documentation for best practices and considerations when conducting variogram analysis. The effect of different variogram models on interpolations is described in Sect. 4.5.

### 4.3 Kriging

Kriging is used to produce deterministic interpolations of a spatial variable, where the goal is to minimize estimation variance (e.g., Matheron, 1963; Cressie, 1990). For a spatial variable  $Z$ , kriging models the residuals  $Z^*$  from a mean  $m$ :

$$Z^*(u_\alpha) = Z(u_\alpha) - m(u_\alpha). \quad (5)$$

In cases where the mean is constant,  $m$  is treated as a global mean. Each interpolated value  $Z^*$  at a location  $u$  is the weighted sum of neighboring measurements:

$$Z^*(u_0) = \sum_{\alpha}^N \lambda_{\alpha} Z(u_{\alpha}), \quad (6)$$

where  $\lambda_{\alpha}$  are the weights on the  $N$  data points. These weights account for the variability in the measurements, their proximity to each other and the node being estimated, and the redundancy between nearby measurements. Specifically, the weights are determined by solving the kriging system of

equations which, for a system with three data points, would look like

$$\begin{bmatrix} C(u_1 - u_1) & C(u_1 - u_2) & C(u_1 - u_3) \\ C(u_2 - u_1) & C(u_2 - u_2) & C(u_2 - u_3) \\ C(u_3 - u_1) & C(u_3 - u_2) & C(u_3 - u_3) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} C(u_0 - u_1) \\ C(u_0 - u_2) \\ C(u_0 - u_3) \end{bmatrix}, \quad (7)$$

which can be abbreviated to

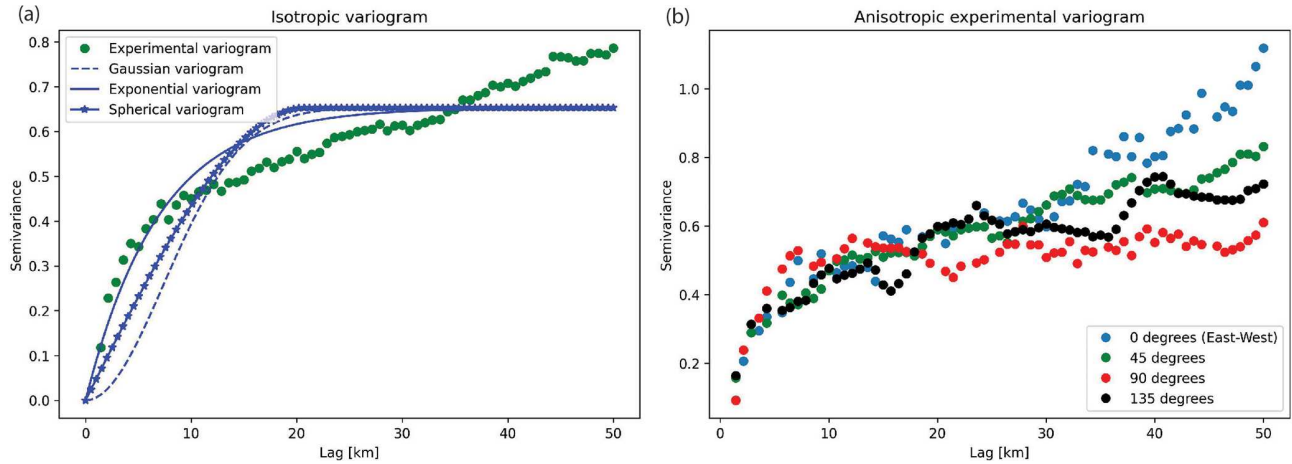
$$\mathbf{C}\boldsymbol{\lambda} = \mathbf{c}, \quad (8)$$

where  $\mathbf{C}$  describes the covariance between pairs of data points, and  $\mathbf{c}$  describes the covariance between each data point and the location that is being estimated,  $u_0$ . Note that the covariance is linked to the variogram:

$$C(h) = C(0) - \gamma(h), \quad (9)$$

where  $C(0)$  is the variance of the data. We use the modeled variogram parameters determined in Sect. 4.2.

GStatSim contains functions for both simple and ordinary kriging. With simple kriging, it is assumed that the mean is a constant, known parameter defined by the mean of the data. For data that have undergone a normal score transformation, the mean is 0. For ordinary kriging, the mean is estimated implicitly within a search neighborhood around the coordinate being estimated, and simple kriging is applied to the residuals. This makes ordinary kriging more robust to spatial trends. The ordinary kriging weight estimation pro-



**Figure 3.** (a) Isotropic experimental variogram and variogram model. (b) Anisotropic variogram.

cedure is modified from Eq. (7) to

$$\begin{bmatrix} C(u_1 - u_1) & C(u_1 - u_2) & C(u_1 - u_3) & 1 \\ C(u_2 - u_1) & C(u_2 - u_2) & C(u_2 - u_3) & 1 \\ C(u_3 - u_1) & C(u_3 - u_2) & C(u_3 - u_3) & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \mu \end{bmatrix} = \begin{bmatrix} C(u_0 - u_1) \\ C(u_0 - u_2) \\ C(u_0 - u_3) \\ 1 \end{bmatrix}, \quad (10)$$

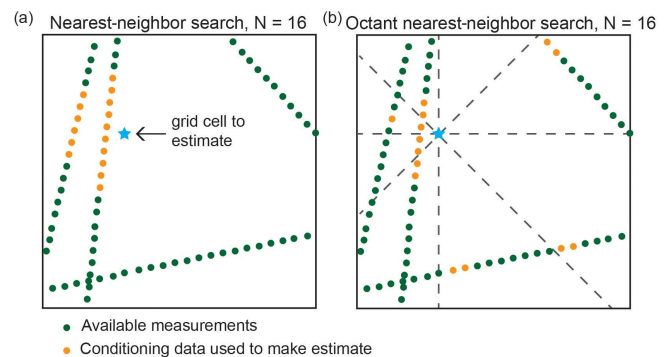
where  $\mu$  is a Lagrange multiplier. The covariance function is also used to compute the uncertainty, or variance, at each location. The variance of an estimate at location  $u_0$ ,  $\sigma_E^2(u_0)$  is defined as

$$\sigma_E^2(u_0) = C(0) - \sum_{\alpha}^N \lambda_{\alpha} C(u_0 - u_{\alpha}). \quad (11)$$

The ability of kriging to quantify uncertainty is a major advantage of this method over other deterministic interpolation methods such as spline interpolation.

#### 4.3.1 Nearest-neighbor octant search

In practice, the covariance matrices in Eqs. (7) and (10),  $\mathbf{C}$  and  $\mathbf{c}$ , used to determine kriging weights typically do not use every value in the conditioning data. Instead, it is conventional to use the  $N$  nearest neighbors, where  $N$  is the number of conditioning nodes (e.g., Pyrcz et al., 2021). This significantly reduces the size of the covariance matrix and ensures that estimates are made using only relevant information. However, the nearest-neighbor approach can be problematic when the conditioning data are unevenly spatially distributed. In the ice-penetrating radar example, measurements are sampled densely along flight lines but are entirely absent in between lines. This means that the  $N$  nearest neighbors of a coordinate next to a flight line could all be on



**Figure 4.** When measurements are gathered along line surveys, a simple nearest-neighbor search (a) may result in estimates being skewed towards values on one side. The octant nearest-neighbor search (b) used in GStatSim ensures that the conditioning data surround the coordinate being estimated, which minimizes this bias in interpolations.

one side, yielding lopsided conditioning data (Fig. 4). This lopsidedness would produce skewed estimates. To avoid this bias, our interpolation algorithms use a nearest-neighbor octant search (e.g., Pebesma, 2004), where the  $N$  neighbors are divided amongst the eight different octants of the coordinate plane (Fig. 4). This ensures that the conditioning data include values on all sides of the node being estimated. The ability of the nearest-neighbor octant search to produce a balanced conditioning data set is important for producing high-quality estimates, particularly for remote sensing data sets where the measurement density is often non-uniform. We note that the primary drawback of this approach is that it increases the algorithm runtime.

### 4.3.2 Covariance matrix inversion

According to Eq. (8), solving for the kriging weights  $\lambda$  requires the inversion of the covariance matrix  $\mathbf{C}$ :

$$\lambda = \mathbf{C}^{-1}\mathbf{c}. \quad (12)$$

However,  $\mathbf{C}$  is not always a positive definite matrix, meaning that  $\mathbf{C}^{-1}$  cannot be computed. This is especially common for complex scenarios such as ice-penetrating radar data sets which often contain overlapping and conflicting measurements. The matrix singularity issue can be resolved by computing the pseudo-inverse,  $\mathbf{C}^+$ , instead. However the pseudo-inverse requires a time-consuming singular value decomposition with a runtime that scales by  $O(n^3)$  with data input size, making it an impractical choice for applications with large data sets. For improved performance, we use the `scipy.linalg.lstsq` function, which computes the least-squares solution to the equation  $\mathbf{A}\mathbf{x} = \mathbf{b}$ . Solving for the weights with `lstsq` instead of computing the matrix inversion directly lends greater numerical stability to the interpolation functions. Therefore, all functions in GStatSim involving covariance matrix inversions are implemented using `lstsq`.

### 4.3.3 Kriging implementation and results

We implement simple kriging and ordinary kriging using the GStatSim `skrige` and `okrige` functions, respectively. The input parameters for these functions are the conditioning data, search radius, number of conditioning points, variogram parameters, and a list of coordinates that describe the prediction grid. We use a search radius of 50 km and 100 conditioning points. For convenience, GStatSim includes a `prediction_grid` function that automatically generates a list of coordinates defined by the minimum and maximum coordinate extents and the grid cell resolution. After GStatSim interpolation is applied, the kriging estimates are back-transformed using the `scikit-learn` `QuantileTransformer` function so that the original data distribution is recovered. For comparison, we also perform simple kriging using the `GeostatsPy` `kb2d` function.

The GStatSim kriging results (Fig. 5) show that the right side of the ordinary kriging estimate has a slightly higher ( $\sim$  tens of meters) elevation than the simple kriging estimate. This is because ordinary kriging does not assume that the mean is constant. The GStatSim and `GeostatsPy` simple kriging estimates are visually similar but differ by tens of meters throughout the domain. The black arrows in Fig. 5c and f highlight a visible seam in the `GeostatsPy` interpolation. This seam is located along grid cells that are equidistant from conditioning lines and may be the result of using a simple nearest-neighbor search (Fig. 4).

To investigate the spatial statistics of kriging interpolation, we compute the variogram for the GStatSim simple

kriging estimate. The resulting variogram, shown in Fig. 6a, has substantially lower variances than the experimental variogram derived from the data. This difference highlights an important limitation of kriging interpolation: the variogram is not reproduced, making the interpolated values smoother than observations.

### 4.4 Sequential Gaussian simulation

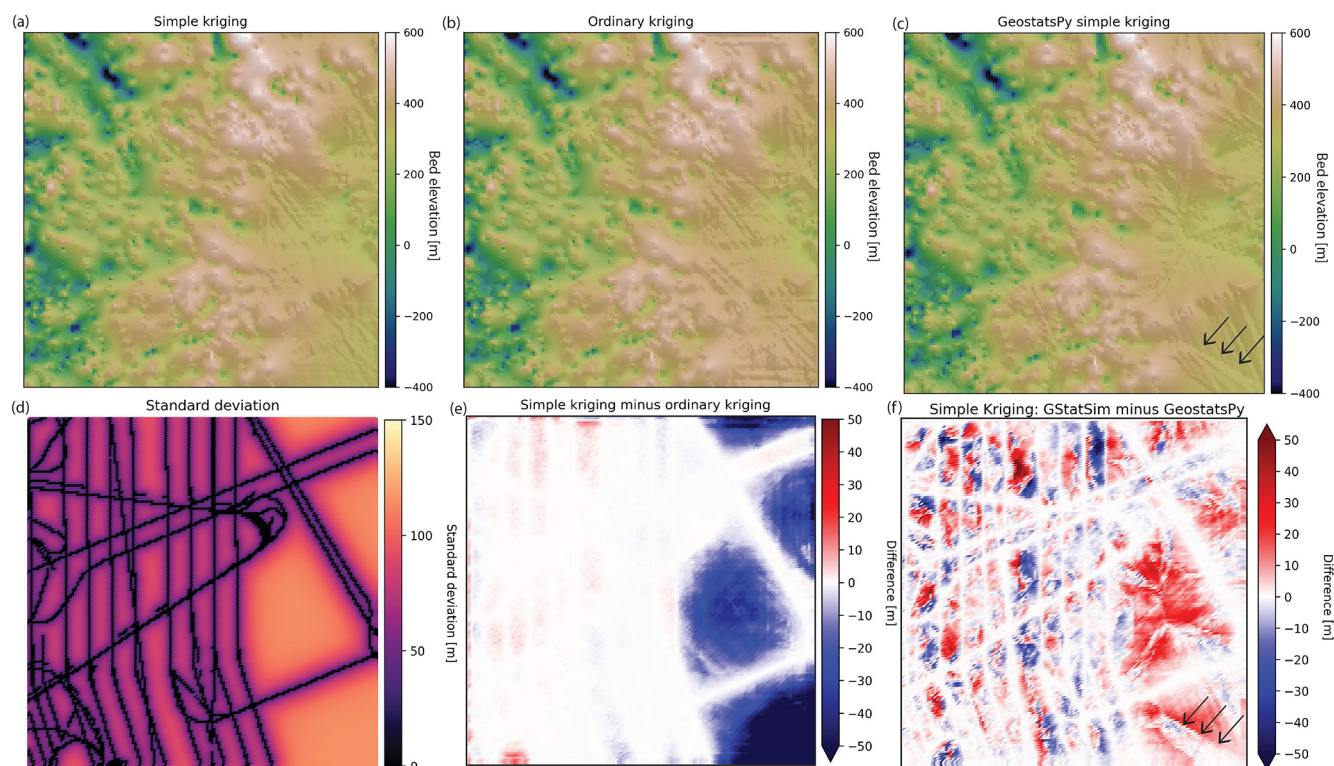
In contrast to kriging, sequential Gaussian simulation (SGS) is designed to preserve the variance of observations. SGS is the stochastic alternative to kriging. In contrast to kriging where the objective is to optimize local accuracy, SGS is used to generate multiple realizations that reproduce the variogram statistics (e.g., Deutsch and Journel, 1992). The implementation of SGS, described in Fig. 7, is the following:

1. Define a random simulation order in which to visit each node.
2. At each node  $u$ , use kriging to estimate the mean  $Z^*(u)$  and variance  $\sigma_E^2(u)$ .
3. Randomly sample from the Gaussian distribution defined by  $Z^*(u)$  and  $\sigma_E^2(u)$ . This becomes the simulated value at that node.
4. Proceed to the next node and repeat steps 2 and 3 until all nodes have been simulated.

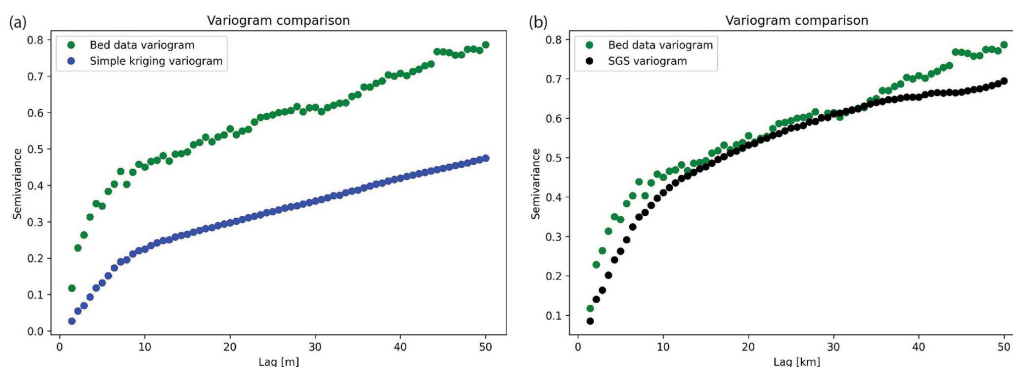
GStatSim has two different SGS functions, `okrige_sgs` and `skrige_sgs`, which use ordinary kriging and simple kriging for step 2, respectively. The results are unlikely to differ significantly between these two methods, but `okrige_sgs` may be more robust when large-scale trends are present. The input parameters are the same as those used for the `skrige` and `okrige` functions. We simulate four realizations using `okrige_sgs` using the same input parameters as in Sect. 4.3.3. To evaluate the ability of SGS to reproduce the spatial statistics of observations, we compute the variogram for realization no. 1 (Fig. 6). The variogram of the simulated topography matches the variogram of observations. The realizations are shown in Fig. 8. Each realization matches the conditioning data and represents an equally probable solution.

### 4.5 Interpolation with different variogram models

GStatSim can perform interpolations using the exponential, Gaussian, or spherical variogram models, as described in Sect. 4.2. In this section, we demonstrate how interpolations differ when using different variogram model types. We generate a synthetic conditioning data set of 100 points using the unconditional Gaussian field tools in `GSTools` (Fig. 9a). We use a variance of 2 and length scale of 8. While the true variogram for the synthetic data is Gaussian, we perform interpolations using different variogram models for demonstration purposes. We compute the experimental variogram and



**Figure 5.** (a) Simple kriging and (b) ordinary kriging interpolations using GStatSim. (c) Simple kriging interpolation using GeostatsPy. (d) Standard deviation for the simple kriging estimate from panel (a). (e) Difference between simple and ordinary kriging. (f) Difference between GStatSim and GeostatsPy simple kriging. The black arrows highlight an artifact in the GeostatsPy interpolation that is likely caused by using a simple nearest-neighbor search (Fig. 4).



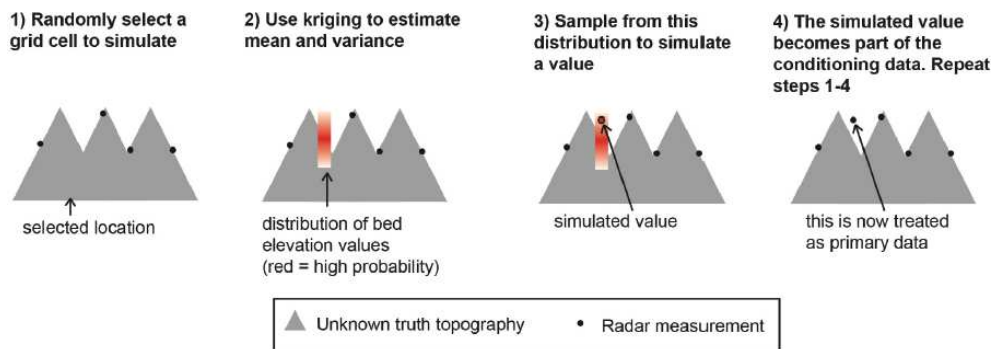
**Figure 6.** (a) Variogram for simple kriging estimate from Fig. 5a. (b) Variogram for SGS realization no. 1.

fit Gaussian, exponential, and spherical variogram models (Fig. 9b). Then we perform both simple kriging interpolation and SGS with each variogram model. The results, shown in Fig. 9, show that the Gaussian model produces the SGS realization with the smoothest spatial structure and lowest kriging variance. The exponential model produces the roughest SGS topography and highest kriging variance, and the spherical model falls in the middle. These different model types allow GStatSim to accommodate different types of spatial structures.

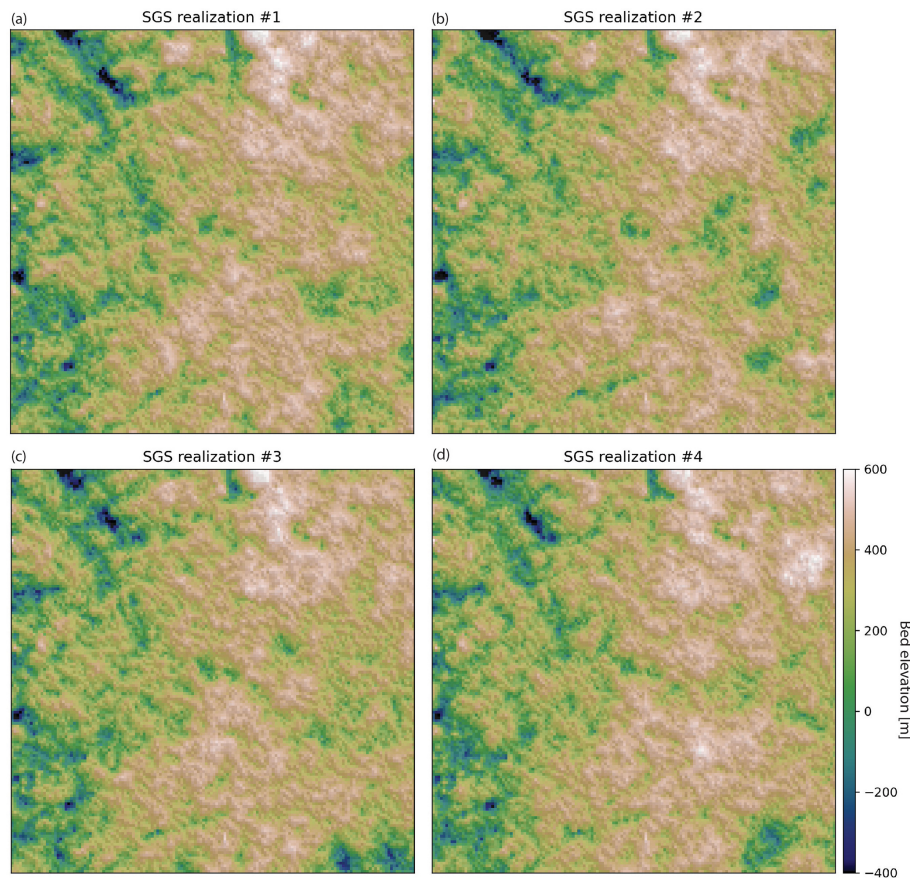
#### 4.6 Interpolation with anisotropy

Geologic phenomena frequently exhibit anisotropy in their spatial statistics. For example, preferential erosion can cause subglacial topography to be smoother in the direction that is parallel to ice flow (MacKie et al., 2021). The GStatSim interpolation functions can account for this anisotropy by accepting major and minor range parameters. The major range is the range of the variogram in the smoothest (major) direction, and the minor range is the range of the variogram in the





**Figure 7.** Schematic of SGS. This figure is modified from MacKie et al. (2021), which was published under a Creative Commons Attribution license (<https://creativecommons.org/licenses/by/4.0/>, last access: 1 May 2023).

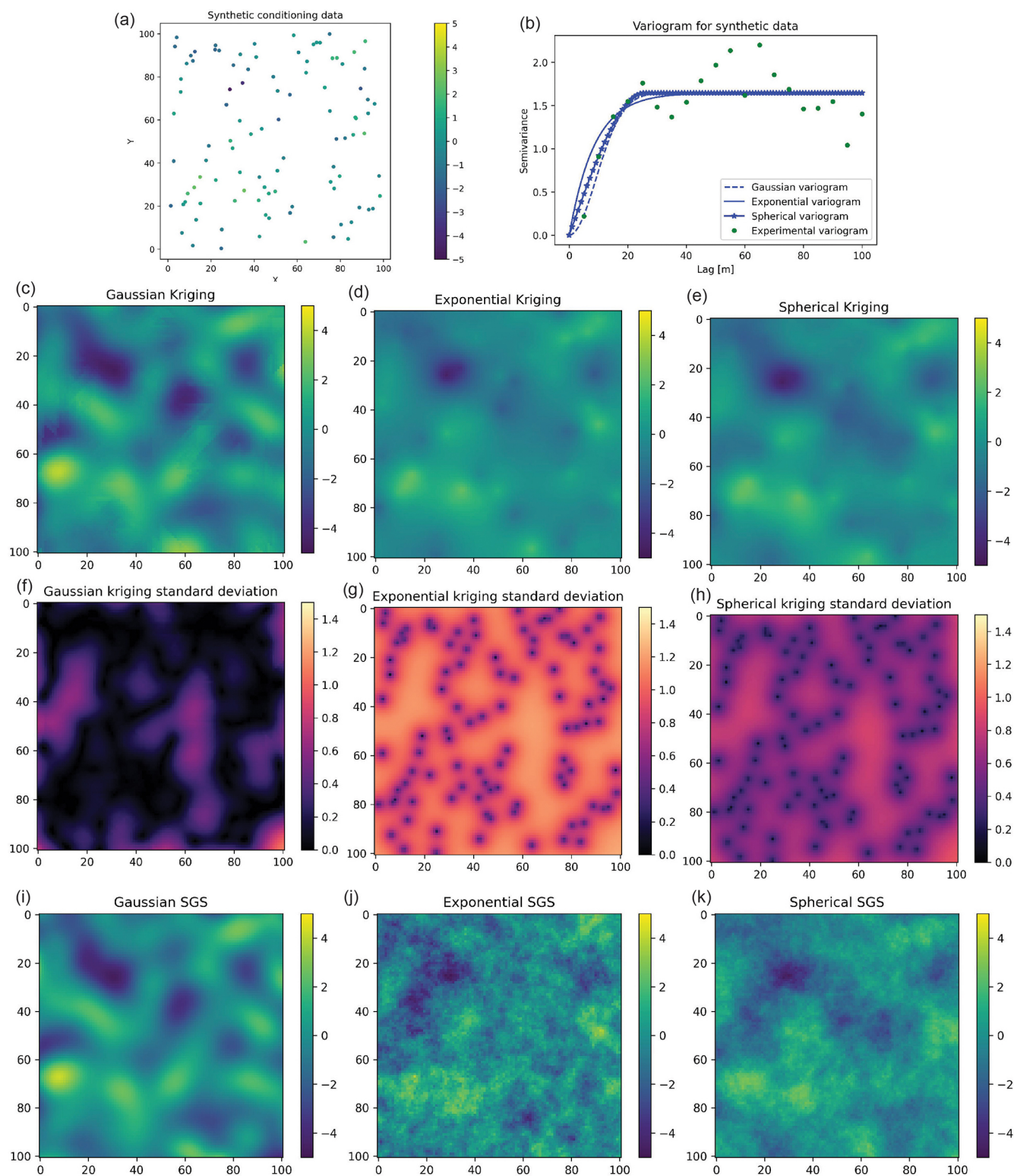


**Figure 8.** (a–d) SGS realizations generated with `okrige_SGS`.

minor direction, which is orthogonal to the major direction. Not all pairs of data points are oriented along the major or minor orientations, so the range is treated as the radius of a rotated ellipse where the major and minor ranges are the lengths of each axis, and the degree of rotation is determined by the angle of anisotropy. The major angle orientation is defined as the angle from zero (horizontal) of the major direction. In practice, these parameters can be determined by

modeling the variogram in multiple directions and determining the major angle through visual inspection.

To demonstrate the implementation of interpolation with anisotropy, we perform simple kriging and SGS with anisotropy for angles of 0 and 60°. We use the modeled variogram parameters from previous sections (nugget = 0, range = 31.9 km, sill = 0.7). We use 31.9 km as the minor range and set the major range to be 15 km greater than



**Figure 9.** Interpolation comparison using different variogram models. Synthetic conditioning data (a), variograms (b), kriging interpolation (c–e), kriging uncertainty (f–h), and SGS (i–k).



the minor range for exaggerated effect. These interpolations (Fig. 10) have clearly visible anisotropy.

#### 4.7 Mean non-stationarity

Data sets often contain variations in spatial statistics, a condition known as non-stationarity. Mean non-stationarity occurs in the presence of a large-scale trend or systematic changes across a study area. While ordinary kriging can accommodate variations in the mean, to some extent, it does not extrapolate well (Journel and Rossi, 1989). For this reason, it is sometimes convenient to detrend the data prior to variogram modeling and interpolation. This approach is known as kriging (or SGS) with a trend or universal kriging. Detrending has the additional advantage of producing residual data that more closely represent a random Gaussian process, which makes the variogram modeling more robust. In contrast, data with a strong trend can yield an experimental variogram that never reaches an obvious sill, making it difficult to fit a variogram model.

Here we demonstrate kriging and SGS with a trend (Fig. 11). This is achieved by the following:

1. Estimate the trend.
2. Subtract the trend from the data to obtain residual measurements.
3. Apply a normal score transformation to the residual data.
4. Conduct a variogram analysis on the transformed residual measurements.
5. Interpolate the residuals.
6. Back-transform the interpolation to recover the original data distribution.
7. Add the back-transformed interpolation to the trend.

While it is common practice to estimate the trend by fitting a linear or polynomial function to the data (e.g., Neven et al., 2021), determining higher-order polynomial models for complex landscapes can be difficult. Instead, we estimate the trend using a radial basis function (RBF), which is a function that changes with distance from a location, such as a thin-plate spline or Gaussian function. RBFs measure the distance between input values and a center point, assigns weights based on these distances, and combines these weights to produce an output value (Broomhead and Lowe, 1988). The RBF trend estimation was implemented using the `GStatSim rbf_trend` function, which relies on the `scipy` RBF implementation. `rbf_trend` uses a multi-quadratic kernel as its RBF, which is known for its high accuracy and flexibility (Franke, 1982; Sarra and Kansa, 2009). The `rbf_trend` function input parameters are the conditioning data, the grid cell resolution, and a smoothing factor.

The smoothing factor ensures that the trend is not overfitting to measurements. The smoothing parameter should be at least as large as the largest data gap. We use a smoothing factor of 100 km.

To evaluate the performance of the detrended SGS approach, we compute the experimental variogram for residual data (Fig. 11d) and the residual SGS (Fig. 11e). The variograms, shown in Fig. 12, show close agreement between the spatial statistics of the residual measurements and simulation. Note that both variograms reach a sill of 1 in contrast to the variogram for the data without detrending (Fig. 6). According to the theoretical definition of the variogram (Eq. 2), data with a standard Gaussian distribution should reach a sill of 1. As such, the detrended data can often be better suited to variogram-based interpolation.

#### 4.8 Variogram non-stationarity

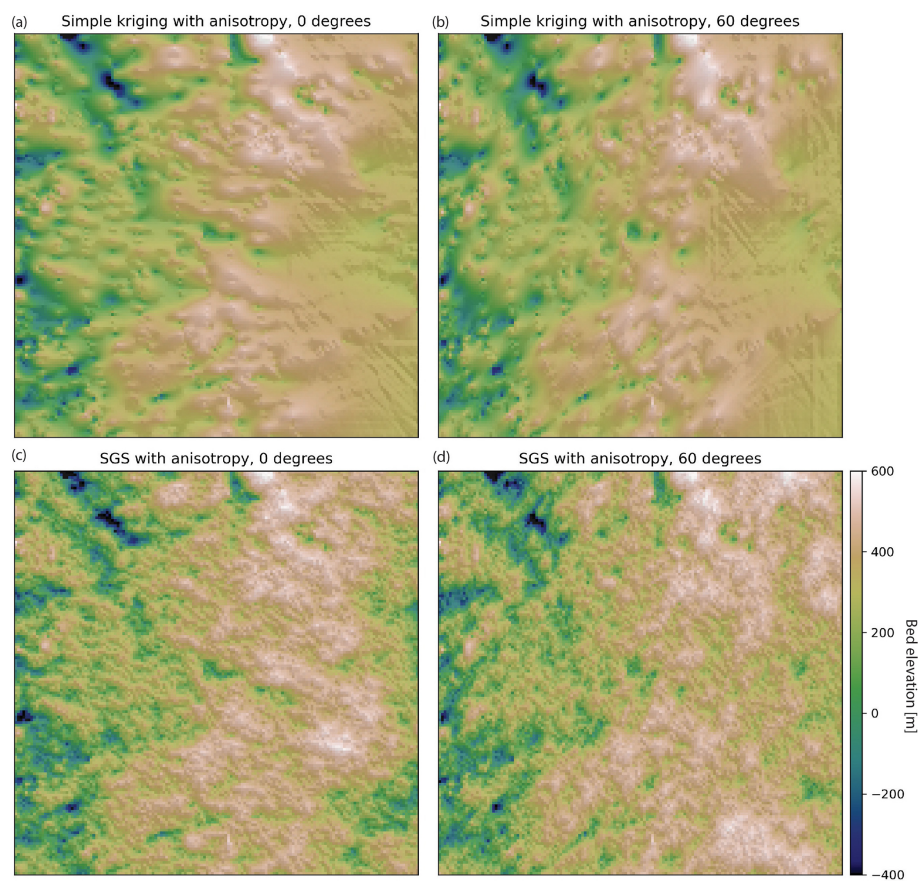
Variogram non-stationarity describes the condition when the variogram statistics vary throughout a domain. For example, the topographic roughness varies throughout Greenland (Cooper et al., 2019). To handle these cases, `GStatSim` includes a `cluster_sgs` function where the user can assign different variograms to different areas. It is implemented by partitioning the data into different clusters and then computing the variogram for each cluster. To simulate a grid cell,

1. choose a random grid cell;
2. find the nearest-neighbor value and look up the cluster number of the nearest-neighbor point;
3. using the variogram parameters associated with the cluster of the nearest point, simulate a value with SGS;
4. assign the nearest-neighbor cluster number to the simulated value;
5. repeat until each grid cell is simulated.

This process ensures that the boundaries between different clusters vary for each realization. `cluster_sgs` takes the same inputs as SGS, as well as the cluster numbers for each data point and a data frame with the variogram parameters for each cluster. Users can use any method they like to partition the data into different clusters. In the next two subsections, we outline two approaches.

##### 4.8.1 K-means clustering approach

One of the simplest ways to cluster data is with *k*-means clustering, where the data are partitioned into *k* clusters such that the average squared sum of distances from the center within each cluster is minimized. We apply *k*-means clustering to the conditioning data and determine three clusters based on the coordinates and bed elevation (Fig. 13a). The justification for using these parameters is that data points that are



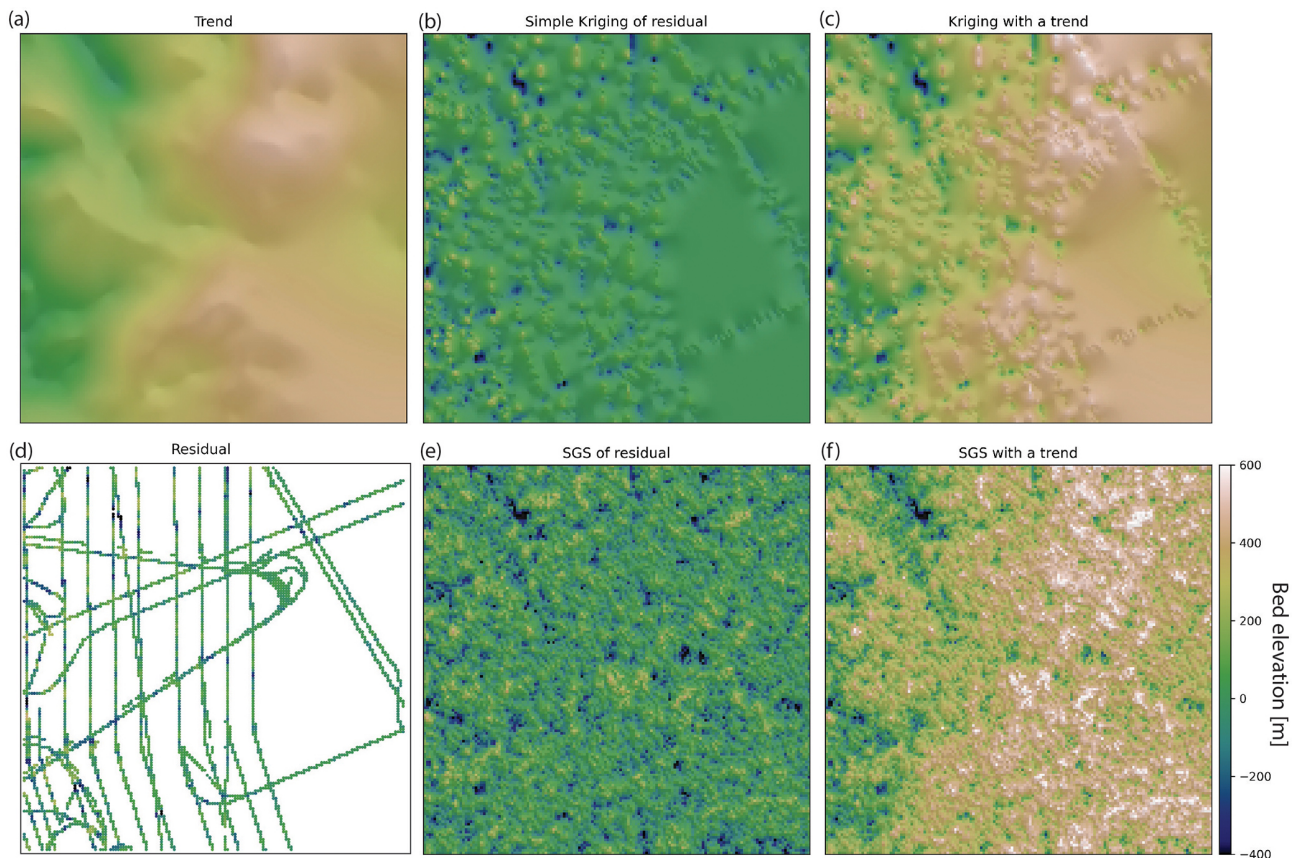
**Figure 10.** (a) Simple kriging with anisotropy at  $0^\circ$ . (b) Simple kriging with anisotropy at  $60^\circ$ . (c) SGS with anisotropy at  $0^\circ$ . (d) SGS with anisotropy at  $60^\circ$ .

geographically close to each other and have similar elevation values should have similar spatial statistics. We use a  $k$  of 3, which was chosen arbitrarily for demonstration purposes. In real applications, external information or domain expertise may be used to inform the clustering decision.

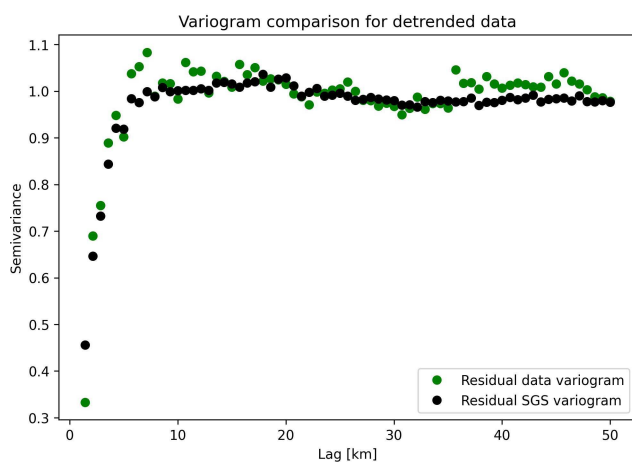
We apply the variogram analysis to the data in each cluster separately using the procedure described in Sect. 4.2. To make the differences between clusters more pronounced, we manually modify several of the variogram parameters to exaggerate the differences in spatial statistics for visualization purposes. Clusters 0 and 2 are assigned angles for anisotropy ( $45^\circ$  for Cluster 0 and  $90^\circ$  for Cluster 2). The minor ranges for Clusters 0 and 2 are assigned based on the automatic variogram modeling. The major ranges are set by adding 15 km to the minor ranges. Cluster 1 is given a sill of 0.6, and Clusters 0 and 2 are assigned sills equal to 1. The `cluster_sgs` simulation using these parameters produces visible differences in spatial statistics throughout the domain. For example, the anisotropy is clearly visible in the region defined by Cluster 0 (Fig. 13b). Despite the extreme differences in parameters used for each cluster, the cluster boundaries do not have artifacts or inconsistencies.

#### 4.8.2 Adaptive partitioning

The  $k$ -means clustering approach described above provides a simple approach to partitioning the data, but the decision boundaries can seem arbitrary and result in clusters with widely varying sample counts. To overcome these challenges we present a recursive implementation of a divisive, density-based clustering approach we call adaptive partitioning. A similar approach was used in Yin et al. (2022). The algorithm begins by treating the data set as a single cluster and quartering this cluster into four rectangles of equal area. For each subsequent cluster, the areal extent of the cluster continues to be quartered until the number of data samples contained within the cluster is below the maximum number of samples allowed or the minimum cluster size has been reached. GStatSim implements this as a fully recursive function named `adaptive_partitioning` with primary parameters `max_points`, which controls the maximum points allowed in a cluster, and `min_length`, which controls the minimum side length of a cluster. It may take tuning of these two parameters to produce a desirable partitioning, including adjusting `max_points` to create clusters of adequate sizes given the measurement density; `min_length`



**Figure 11.** (a) Trend estimate. (b) Simple kriging of detrended data. (c) The simple kriging estimate from panel (b) added to the trend from panel (a). (d) The residual bed elevation data after the trend is subtracted from the data. (e) SGS applied to the detrended data. (f) The SGS interpolation from panel (e) added to the trend from panel (a).



**Figure 12.** Variogram for detrended data from Fig. 11d and simulation of trend from Fig. 11e. These variograms have a sill of 1, which is the theoretical sill for standard Gaussian data.

should be greater than the expected variogram range to ensure that there are sufficient data to accurately model the variogram parameters. Figure 14a shows the clusters result-

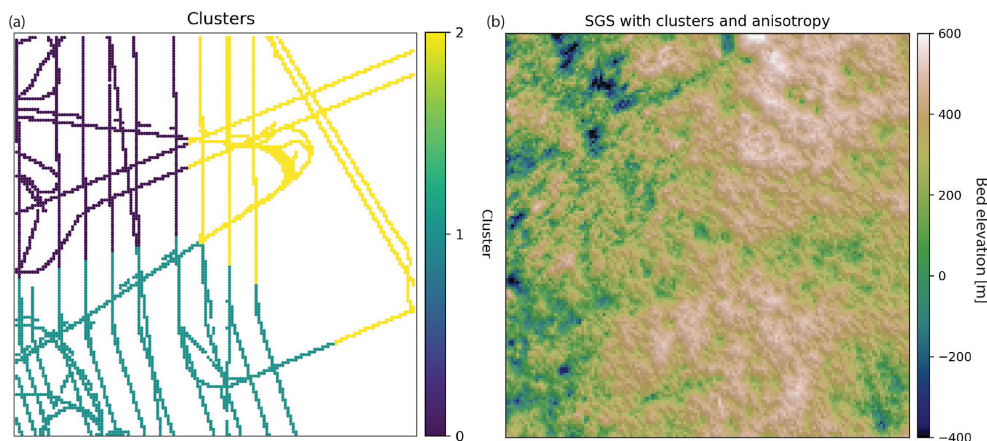
ing from `adaptive_partitioning` with `max_points` of 800 and `min_length` of 25 km.

The variogram parameters for each cluster are modeled automatically and then modified to show exaggerated differences between clusters. Cluster 1 is modified to have a sill of 0.6. Clusters 6 and 12 are given 90 and 45° directions of anisotropy, respectively, with major ranges that are determined by adding 15 km to the modeled range. These regional differences in variogram parameters are apparent in the resulting simulation in Fig. 14.

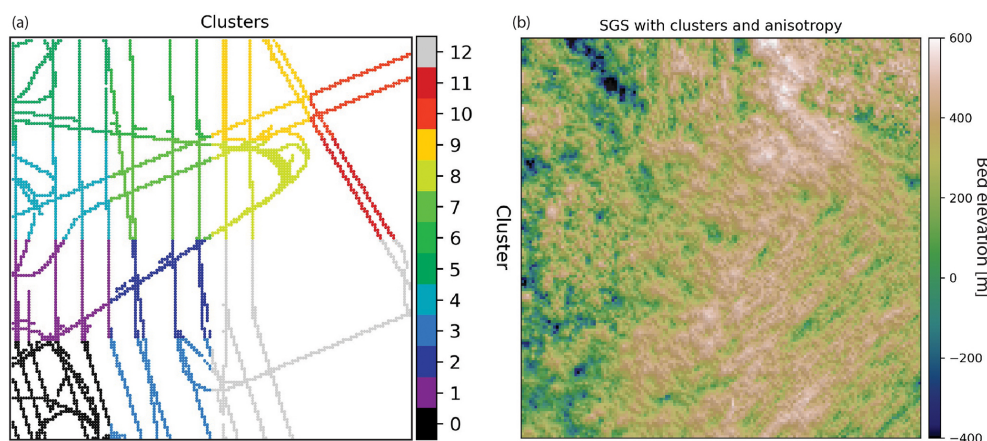
#### 4.9 Co-kriging and co-simulation

Secondary information can be integrated into interpolations via co-kriging and co-simulation, where a well-sampled secondary variable is used to constrain the interpolation of a sparsely sampled primary variable. For example, if the ice surface elevation (Fig. 15a) is found to be correlated with the subglacial bed elevation, then the ice surface can be used to improve the interpolation. We note that the ice surface and subglacial topography are unlikely to be linearly related (Ng et al., 2018); this example is merely intended to demonstrate the usage of our co-kriging and co-simulation tools. In the-





**Figure 13.** (a) *K*-means clustering of conditioning data. (b) SGS using different variogram parameters for each cluster.



**Figure 14.** (a) Clusters derived from density-based adaptive partitioning function. (b) SGS using a different variogram in each cluster.

ory, co-kriging and co-simulation are implemented by modeling the variogram for each variable and a cross-variogram that describes the covariation between the variables. In practice, the full co-kriging system of equations is difficult to solve, so approximations are used instead (Almeida and Journel, 1994; Journel, 1999).

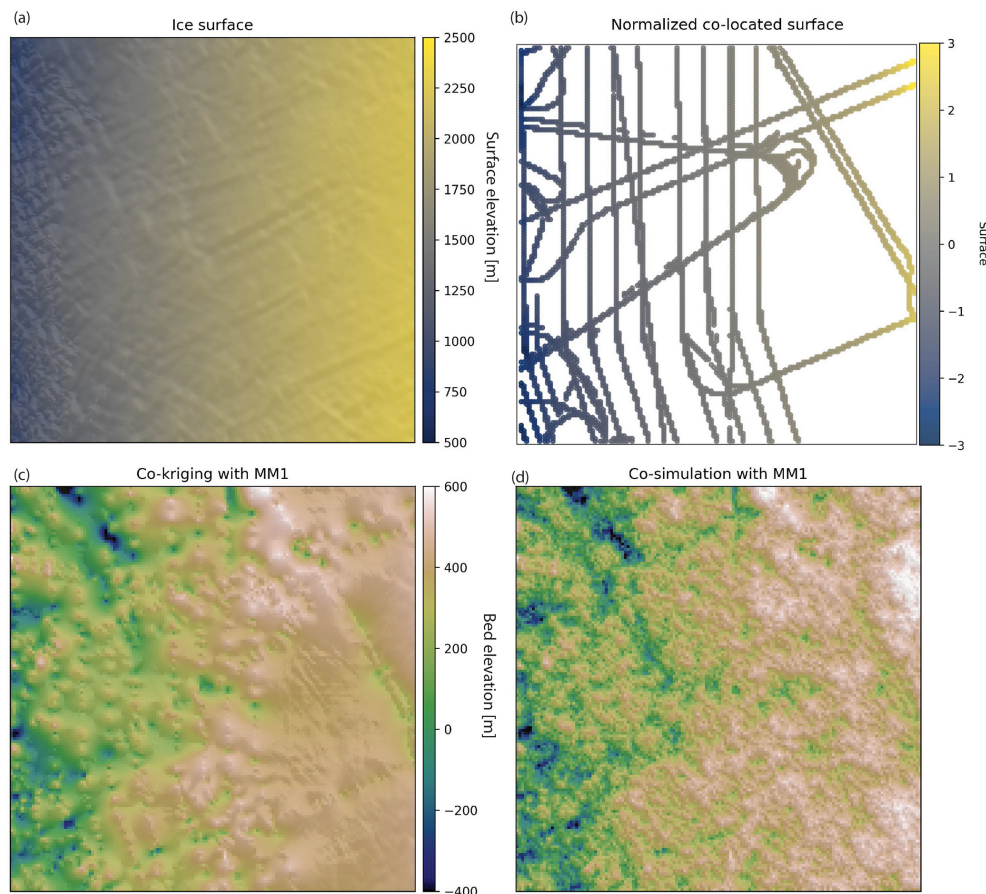
While there are several different ways to implement co-kriging (Journel, 1999), GStatSim only includes functions for co-located co-kriging under Markov model 1 (MM1) assumptions, as described by Almeida and Journel (1994). This version was chosen for its simplicity; expert geostatistical knowledge is not needed to use this algorithm. Co-located co-kriging means that an estimate is made using only the secondary data point that is *co-located* with, or at the same location as, the grid cell being estimated. This significantly reduces the number of conditioning data points that are used to estimate each grid cell. Co-kriging with MM1 approximates the cross-variogram through a Markov assumption of conditional independence. It is assumed that a secondary data variable  $Z_2$  at a location  $u$  is conditionally independent of the

primary variable  $Z_1$  at a location of  $u'$ , given  $Z_1(u)$ .  $u'$  refers to a spatial location that is not  $u$ . This statement of conditional independence is written as

$$\begin{aligned} E(Z_2(u)|Z_1(u) = z_1, Z_1(u') = z_1(u')) \\ = E(Z_2(u)|Z_1(u) = z_1), \end{aligned} \quad (13)$$

where  $E$  refers to the expected value. In the geostatistical community, this Markov assumption is referred to as Markov model 1 (Almeida and Journel, 1994). See Journel (1999) and Shmaryan and Journel (1999) for other co-kriging models. In the case where the ice surface elevation is used as a secondary constraint, MM1 means that the expected value of the ice surface at a location  $u$ , given knowledge of the bed elevation at location  $u$ , is independent of the bed elevation value at other locations. Under MM1 assumptions, the cross-correlogram describing the covariation between the primary and secondary variables is calculated as

$$\rho_{12}(h) = \rho_{12}(0)\rho_1(h), \quad (14)$$



**Figure 15.** (a) Ice surface elevation data. (b) Normalized ice surface elevation measurements that are co-located with radar measurements. (c) Co-kriging interpolation. (d) Co-simulation.

where  $h$  is the lag distance and  $\rho$  is the correlogram, calculated by

$$\rho(h) = 1 - \gamma(h), \quad (15)$$

assuming that  $\gamma(h)$  is the normal scores variogram. While the variogram measures spatial covariance, the correlogram measures spatial correlation. This means that  $\rho_{12}(0)$ , the correlogram at a lag distance of zero, is simply the correlation coefficient between the primary and secondary variables. Because the cross-covariance between the primary and secondary data relies only on the correlation coefficient and primary variogram, the secondary variogram is not needed. This means that co-kriging and co-simulation with MM1 only require the primary variogram and the correlation coefficient to determine the co-kriging weights. The simple co-located co-kriging estimate  $Z_1^*$  at a location  $u_0$  is defined as

$$Z_1^*(u_0) = \sum_{\alpha}^N \lambda_{\alpha} Z_1(u_{\alpha}) + \lambda_2 Z_2(u_0), \quad (16)$$

where  $\lambda_{\alpha}$  are the weights on the  $N$  primary data points,  $Z_2(u_0)$  is the co-located secondary datum, and  $\lambda_2$  is the

weight for the secondary datum. This means that the co-kriging estimate of a bed elevation value is the weighted sum of bed elevation data points within a specified search radius and the co-located ice surface elevation datum. The variance is computed using Eq. (11). Co-SGS is implemented in the same way as SGS, except that Eq. (16) is used instead of Eq. (6).

To implement co-kriging and co-SGS, we use the variogram model parameters from Sect. 4.2 and compute the correlation coefficient,  $\rho_{12}(0)$ , between the bed elevation measurements and co-located ice surface elevation. For convenience, we developed a `find_colocated` function that extracts the co-located data points from the secondary variable (Fig. 15b).  $\rho_{12}(0)$  was determined to be 0.6. The interpolations are performed using the `GStatSim` `cokrige_mm1` and `cosim_mm1` functions. These functions require the same input parameters as `skrige`, `okrige`, `skrige_sfs`, and `okrige_sfs` with the addition of  $\rho_{12}(0)$  and the secondary data set.

The co-kriging and co-SGS results are shown in Fig. 15. Compared to the previous kriging and simulation examples, these interpolations produce visibly higher-elevation topog-

raphy on the right side of the region. This is due to the correlation with the ice surface. To evaluate the simulation performance, we compute the correlation coefficient between the ice surface elevation (Fig. 15a) and co-simulated bed elevation (Fig. 15d). This correlation coefficient is found to be 0.7, which is greater than the correlation coefficient of the co-located data, 0.6. This inflated correlation between the primary and secondary data is a known issue with MM1 (Shmeryan and Journel, 1999). Co-simulation with MM1 is also known to produce interpolation artifacts (MacKie et al., 2021) and artificially high variance (Journel, 1999). This is because the MM1 assumption screens out the influence of the primary data on the secondary data, except at the location  $u_0$ , which can lead to the underestimation of the redundancy between the primary and secondary variables. For this reason, MM1 is most effective in situations where the primary variable is smoother than the secondary variable, which is not the case in our example. As such, it is important to check the statistics (variogram and correlation) of simulations produced with `cosim_mm1` to ensure that this method is used appropriately.

## 5 Tutorials and documentation

Each of the previous interpolation examples can be reproduced in tutorials found in the GStatSim GitHub page (<https://github.com/GatorGlaciology/GStatSim>, last access: 16 May 2023), Jupyter Book (<https://gatorglaciology.github.io/gstatsimbook/intro.html>, last access: 16 May 2023), or Zenodo repository (<https://doi.org/10.5281/zenodo.7274640>, MacKie et al., 2022). These tutorials can be downloaded as Jupyter Notebook files and are intended to enhance educational pathways to geostatistics.

## 6 Installation, licensing, and redistribution

GStatSim is added to the Python Package Index, commonly known as PyPi, and can be installed via the `pip install gstatsim` command. It can also be installed directly from GitHub with `git clone` (<https://github.com/GatorGlaciology/GStatSim>, last access: 16 May 2023). This package has an MIT license making it free to use and redistribute without restriction.

## 7 Discussion

Many existing geostatistical software applications are not open-source or directly suited to large-scale remote sensing problems. For example, we found that `GSTools` was unable to perform conditional interpolations for our topography case study due to the large RAM requirements, making it an impractical choice for applications with large conditioning data sets. Similarly, the `GeostatsPy` simulation crashed

due to a numerical instability. While it is certainly possible that these issues can be attributed to user error, they highlight the need for an open-source geostatistical package that is specifically tested on large-scale line surveys.

The GStatSim package provides an accessible, robust alternative to existing software. The functions in this package are efficient and effective at large scales. Additionally, the octant search can mitigate interpolation bias when line data are used. GStatSim relies on existing packages (e.g., `SciKit-GStat`) where possible and can easily be integrated with existing Python tools and scientific analyses. The tutorials and documentation provide ample scaffolding for users without prior geostatistical experience. This will encourage the utilization of geostatistical simulation in new scientific domains and enhance educational pathways in geostatistics.

The bed topography interpolation examples show many different ways to interpolate the same data set, each with different assumptions, advantages, and limitations. We do not make a recommendation for using one interpolation method over others. In practice, the optimal interpolation method or combination of methods will depend on the structure of the data and the scientific objectives for a specific problem. For example, kriging methods are best if local accuracy is desired, while SGS should be used if the goal is to preserve the heterogeneity of a given geological parameter. The `cluster_sgs` function is advantageous for enabling variations in the spatial statistics of simulated phenomena. In practice, expert domain knowledge will be needed to select the best clustering approach and parameters. For instance, a known geological boundary may be used to delineate clusters. Many of these interpolation approaches can be combined. For example, `cluster_sgs`, `cokrige_mm1`, and `cosim_mm1` could be applied to detrended data.

Although the focus of this article is subglacial topography, the GStatSim routines could be applied to numerous geoscientific problems. Cryosphere applications could include the interpolation of englacial layers (MacGregor et al., 2015), marine sediment cores, and radiometric properties of subglacial conditions (e.g., Chu et al., 2021). Beyond glaciology, GStatSim could be used for downscaling, potential field data analysis (Volkova and Merkulov, 2019), and planetary applications such as improving the resolution of digital terrain models (Gwinner et al., 2009). The `cluster_sgs` function would be particularly useful for capturing regional changes in spatial statistics, and the nearest-neighbor octant search makes GStatSim well-suited for geostatistical modeling with geophysical line data. Furthermore, the ability of co-kriging and co-simulation to include secondary constraints provides a mechanism for synthesizing multiple geophysical and geological data sets. GStatSim could also be integrated with geophysical modeling routines such as `SimPEG` (Cockett et al., 2015) to perform inversions. Specifically, geostatistical simulation can be used in Markov chain



Monte Carlo frameworks for solving inverse problems (Haas and Dubrule, 1994; Oh and Kwon, 2001; Nunes et al., 2012).

Each of the example interpolations in this paper takes 2 to 3 min to run on a MacBook Pro with an Apple M1 chip. The run times increase exponentially with size and resolution improvement. Most of this computational time is attributed to the nearest-neighbor octant search. As such, geostatistical simulation can become cumbersome and time-intensive for large ensembles, depending on the simulation grid size, nearest-neighbor search radius, and number of conditioning points. There are many approaches for accelerating SGS (Dimitrakopoulos and Luo, 2004; Daly et al., 2010; Mariethoz, 2010; Nussbaumer et al., 2018), though these methods lack flexibility or require greater computational resources. Future work is needed to develop tools with improved performance while maintaining the flexibility and functionality. GStatSim could also be modified to include additional variogram models and interpolation methods such as indicator kriging (Solow, 1986).

## 8 Conclusions

Spatial interpolation is a common problem in the geosciences, though the availability of open-access software remains a key barrier to the widespread utilization of geostatistics. The GStatSim package addresses this barrier by making available Python tools and educational materials for implementing various interpolation methods. The tools here could be applied to a variety of geological and geophysical data sets for both research and educational purposes and can be integrated with existing Python tools and workflows. The inclusion of non-stationary SGS, co-kriging, co-simulation, and trend estimation functions make GStatSim particularly useful for applying geostatistics to complex phenomena.

*Code and data availability.* The source code, data, and tutorials are permanently archived in Zenodo (<https://doi.org/10.5281/zenodo.7274640>, MacKie et al., 2022). These materials can also be accessed from GitHub (<https://github.com/GatorGlaciology/GStatSim>, 16 May 2023), PyPi (<https://pypi.org/project/gstatsim/>, 16 May 2023), and our Jupyter Book (<https://gatorglaciology.github.io/gstatsimbook/intro.html>, 16 May 2023).

*Author contributions.* EJM is the primary developer of GStatSim and author of the manuscript. MF made major contributions to the package and writing. LW, ZY, NS, MH, and AZ contributed to the GStatSim software development.

*Competing interests.* The contact author has declared that none of the authors has any competing interests.

*Disclaimer.* Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

*Acknowledgements.* We would like to thank Eric Stubbs for providing technical support on the software development. We thank Fernando Pérez for his guidance on the direction of this repository. We thank Caroline Riggall, Leo Ramsey Watson, and Caleb Koresh for helping test the tools. We also thank Mirko Mälicke and Joe MacGregor whose reviews greatly improved this article.

*Financial support.* This work was sponsored by the Earth Sciences Information Partners (ESIP) Lab, the National Science Foundation (NSF) GeoSMART program, and the University of Florida Artificial Intelligence Initiative Center.

*Review statement.* This paper was edited by Deepak Subramani and reviewed by Joseph MacGregor and Mirko Mälicke.

## References

- Almeida, A. S. and Journel, A. G.: Joint simulation of multiple variables with a Markov-type coregionalization model, *Math. Geol.*, 26, 565–588, 1994.
- Broomhead, D. S. and Lowe, D.: Radial basis functions, multi-variable functional interpolation and adaptive networks, *Tech. rep.*, Royal Signals and Radar Establishment Malvern, United Kingdom, 1988.
- Carle, S. F.: T-PROGS: Transition probability geostatistical software, Version 2.1, Department of Land, Air and Water Resources, University of California, Davis, 1999.
- Chiles, J.-P. and Delfiner, P.: *Geostatistics: modeling spatial uncertainty*, Vol. 497, John Wiley & Sons, <https://doi.org/10.1007/s11004-012-9429-y>, 2009.
- Chu, W., Hilger, A. M., Culberg, R., Schroeder, D. M., Jordan, T. M., Seroussi, H., Young, D. A., Blankenship, D. D., and Vaughan, D. G.: Multisystem synthesis of radar sounding observations of the Amundsen Sea sector from the 2004–2005 field season, *J. Geophys. Res.-Earth*, 126, e2021JF006296, <https://doi.org/10.1029/2021JF006296>, 2021.
- Cockett, R., Kang, S., Heagy, L. J., Pidlisecky, A., and Oldenburg, D. W.: SimPEG: An open source framework for simulation and gradient based parameter estimation in geophysical applications, *Comput. Geosci.*, 85, 142–154, 2015.
- Cooper, M. A., Jordan, T. M., Schroeder, D. M., Siegert, M. J., Williams, C. N., and Bamber, J. L.: Subglacial roughness of the Greenland Ice Sheet: relationship with contemporary ice velocity and geology, *The Cryosphere*, 13, 3093–3115, <https://doi.org/10.5194/tc-13-3093-2019>, 2019.
- Costa, A. C. and Soares, A.: Homogenization of climate data: review and new perspectives using geostatistics, *Math. Geosci.*, 41, 291–305, 2009.
- CReSIS: CReSIS Radar Depth Sounder, Digital Media, <http://data.cresis.ku.edu/> (last access: 1 August 2022), 2022.
- Cressie, N.: The origins of kriging, *Math. Geol.*, 22, 239–252, 1990.

- Cressie, N. and Hawkins, D. M.: Robust estimation of the variogram: I, *J. Int. Ass. Math. Geol.*, 12, 115–125, 1980.
- Daly, C., Quental, S., and Novak, D.: A faster, more accurate Gaussian simulation, in: *Proceedings of the Geocanada Conference*, Calgary, AB, Canada, 10–14, [https://www.searchanddiscovery.com/pdfz/abstracts/pdf/2014/90172cspg/abstracts/ndx\\_daly.pdf.html](https://www.searchanddiscovery.com/pdfz/abstracts/pdf/2014/90172cspg/abstracts/ndx_daly.pdf.html) (last access: 28 June 2023), 2010.
- Deutsch, C. V. and Journel, A. G.: *Geostatistical software library and user's guide*, Oxford University Press, 1–375, 1992.
- Dimitrakopoulos, R. and Luo, X.: Generalized sequential Gaussian simulation on group size  $v$  and screen-effect approximations for large field simulations, *Math. Geol.*, 36, 567–591, 2004.
- Emery, X. and Maleki, M.: Geostatistics in the presence of geological boundaries: Application to mineral resources modeling, *Ore Geol. Rev.*, 114, 103124, <https://doi.org/10.1016/j.oregeorev.2019.103124>, 2019.
- Feyen, L. and Caers, J.: Quantifying geological uncertainty for flow and transport modeling in multi-modal heterogeneous formations, *Adv. Water Resour.*, 29, 912–929, 2006.
- Franke, R.: Scattered data interpolation: tests of some methods, *Math. Comput.*, 38, 181–200, 1982.
- GMS: *GMS 10.6 Editions & Pricing*, <https://www.aquaveo.com/software/gms-pricing> (last access: 1 August 2022), 2021.
- Goovaerts, P.: Ordinary cokriging revisited, *Math. Geol.*, 30, 21–42, 1998.
- Goovaerts, P.: Geostatistics in soil science: state-of-the-art and perspectives, *Geoderma*, 89, 1–45, 1999.
- Gwinner, K., Scholten, F., Spiegel, M., Schmidt, R., Giese, B., Oberst, J., Heipke, C., Jaumann, R., and Neukum, G.: Derivation and validation of high-resolution digital terrain models from Mars Express HRSC data, *Photogramm. Eng. Rem. S.*, 75, 1127–1142, 2009.
- Haas, A. and Dubrule, O.: Geostatistical inversion—a sequential method of stochastic reservoir modelling constrained by seismic data, *First break*, 12, <https://doi.org/10.3997/1365-2397.1994034>, 1994.
- Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E.: Array programming with NumPy, *Nature*, 585, 357–362, 2020.
- Journel, A. G.: Markov models for cross-covariances, *Math. Geol.*, 31, 955–964, 1999.
- Journel, A. G. and Huijbregts, C. J.: *Mining geostatistics*, Academic Press, 600 pp., ISBN-10: 0123910501, 1976.
- Journel, A. G. and Rossi, M.: When do we need a trend model in kriging?, *Math. Geol.*, 21, 715–739, 1989.
- Kelkar, M. and Perez, G.: *Applied geostatistics for reservoir characterization*, Society of Petroleum Engineers, 274 pp., ISBN-10: 1555630952, 2002.
- Kitanidis, P. K.: *Introduction to geostatistics: applications in hydrogeology*, Cambridge University Press, 272 pp., ISBN: 9780521587471, 1997.
- Krige, D. G.: A statistical approach to some basic mine valuation problems on the Witwatersrand, *J. S. Afr. I. Min. Metall.*, 52, 119–139, 1951.
- Lark, R. M.: Towards soil geostatistics, *Spat. Stat.-Neth.*, 1, 92–99, 2012.
- Law, R., Christoffersen, P., MacKie, E., Cook, S., Haseloff, M., and Gagliardini, O.: Complex motion of Greenland Ice Sheet outlet glaciers with basal temperate ice, *Science Advances*, 9, eabq5180, <https://doi.org/10.1126/sciadv.abq5180>, 2023.
- MacGregor, J. A., Fahnestock, M. A., Catania, G. A., Paden, J. D., Prasad Gogineni, S., Young, S. K., Rybarski, S. C., Mabrey, A. N., Wagman, B. M., and Morlighem, M.: Radiostratigraphy and age structure of the Greenland Ice Sheet, *J. Geophys. Res.-Earth*, 120, 212–241, 2015.
- MacKie, E., Schroeder, D., Caers, J., Siegfried, M., and Scheidt, C.: Antarctic topographic realizations and geostatistical modeling used to map subglacial lakes, *J. Geophys. Res.-Earth*, 125, e2019JF005420, <https://doi.org/10.1029/2019JF005420>, 2020.
- MacKie, E., Field, M., Wang, L., Yin, Z., Schoedl, N., and Hibbs, M.: *GStatSim*, Version 1.0, Zenodo [code], <https://doi.org/10.5281/zenodo.7274640>, 2022.
- MacKie, E. J., Schroeder, D. M., Zuo, C., Yin, Z., and Caers, J.: Stochastic modeling of subglacial topography exposes uncertainty in water routing at Jakobshavn Glacier, *J. Glaciol.*, 67, 75–83, 2021.
- Mälicke, M.: SciKit-GStat 1.0: a SciPy-flavored geostatistical variogram estimation toolbox written in Python, *Geosci. Model Dev.*, 15, 2505–2532, <https://doi.org/10.5194/gmd-15-2505-2022>, 2022.
- Mariethoz, G.: A general parallelization strategy for random path based geostatistical simulation methods, *Comput. Geosci.*, 36, 953–958, 2010.
- Matheron, G.: Principles of geostatistics, *Econ. Geol.*, 58, 1246–1266, 1963.
- McKinney, W.: Data structures for statistical computing in python, in: *Proceedings of the 9th Python in Science Conference*, Austin, TX, 445, 51–56, 2010.
- Müller, S., Schüler, L., Zech, A., and Heße, F.: *GSTools v1.3: a toolbox for geostatistical modelling in Python*, *Geosci. Model Dev.*, 15, 3161–3182, <https://doi.org/10.5194/gmd-15-3161-2022>, 2022.
- Murphy, B. S.: PyKrig: development of a kriging toolkit for Python, in: *AGU fall meeting abstracts*, San Francisco, December 2014, H51K-0753, <https://geostat-framework.readthedocs.io/en/latest/> (last access: 29 June 2023), 2014.
- Neven, A., Dall'Alba, V., Juda, P., Straubhaar, J., and Renard, P.: Ice volume and basal topography estimation using geostatistical methods and ground-penetrating radar measurements: application to the Tsanfleuron and Scex Rouge glaciers, Swiss Alps, *The Cryosphere*, 15, 5169–5186, <https://doi.org/10.5194/tc-15-5169-2021>, 2021.
- Ng, F. S., Ignéczi, Á., Sole, A. J., and Livingstone, S. J.: Response of surface topography to basal variability along glacial flowlines, *J. Geophys. Res.-Earth*, 123, 2319–2340, 2018.
- Nunes, R., Soares, A., Schwedersky, G., Dillon, L., Guerreiro, L., Caetano, H., Maciel, C., and Leon, F.: Geostatistical inversion of prestack seismic data, in: *Ninth International Geostatistics Congress*, Oslo, Norway, June, 2012, 1–8, <http://geostats2012.nr.no/pdfs/1747134.pdf> (last access: 29 June 2023), 2012.
- Nussbaumer, R., Mariethoz, G., Gravey, M., Gloaguen, E., and Holliger, K.: Accelerating sequential gaussian simulation with a constant path, *Comput. Geosci.*, 112, 121–132, 2018.

- Oh, S.-H. and Kwon, B.-D.: Geostatistical approach to Bayesian inversion of geophysical data: Markov chain Monte Carlo method, *Earth Planets Space*, 53, 777–791, 2001.
- Parizek, B., Christianson, K., Anandakrishnan, S., Alley, R., Walker, R., Edwards, R., Wolfe, D., Bertini, G., Rinehart, S., Bindschadler, R., and Nowicki, S. M. J.: Dynamic (in) stability of Thwaites Glacier, West Antarctica, *J. Geophys. Res.-Earth*, 118, 638–655, 2013.
- Pebesma, E. J.: Multivariable geostatistics in S: the gstat package, *Comput. Geosci.*, 30, 683–691, 2004.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: Scikit-learn: Machine Learning in Python, *J. Mach. Learn. Res.*, 12, 2825–2830, 2011.
- Porter, C., Morin, P., Howat, I., Noh, M.-J., Bates, B., Peterman, K., Keese, S., Schlenk, M., Gardiner, J., Tomko, K., Willis, M., Kelleher, C., Cloutier, M., Husby, E., Foga, S., Nakamura, H., Platson, M., Wethington Jr., M., Williamson, C., Bauer, G., Enos, J., Arnold, G., Kramer, W., Becker, P., Doshi, A., D'Souza, C., Cummins, P., Laurier, F., and Bojesen, M.: ArcticDEM, Version 3, Harvard Dataverse [data set], <https://doi.org/10.7910/DVN/OHHUKH>, 2018.
- Pyrz, M., Jo, H., Kupaenko, A., Liu, W., Gigliotti, A., Salomaki, T., and Santos, J.: GeostatsPy, <https://github.com/GeostatsGuy/GeostatsPy> (last access: 1 June 2022), 2021.
- Pyrz, M. J. and Deutsch, C. V.: Geostatistical reservoir modeling, Oxford University Press, 384 pp., ISBN: 978-0199731442, 2014.
- Remy, N.: S-GeMS: the stanford geostatistical modeling software: a tool for new algorithms development, in: *Geostatistics Banff 2004*, edited by: Leuangthong, O. and Deutsch, C. V., Springer, 865–871, [https://doi.org/10.1007/978-1-4020-3610-1\\_89](https://doi.org/10.1007/978-1-4020-3610-1_89), 2005.
- Remy, N., Boucher, A., and Wu, J.: *Applied geostatistics with SGeMS: A user's guide*, Cambridge University Press, <https://doi.org/10.1017/CBO9781139150019>, 2009.
- Sarra, S. A. and Kansa, E. J.: Multiquadric radial basis function approximation methods for the numerical solution of partial differential equations, *Advances in Computational Mechanics*, 2, 220 pp., 2009.
- Seroussi, H., Nakayama, Y., Larour, E., Menemenlis, D., Morlighem, M., Rignot, E., and Khazendar, A.: Continued retreat of Thwaites Glacier, West Antarctica, controlled by bed topography and ocean circulation, *Geophys. Res. Lett.*, 44, 6191–6199, 2017.
- Shamsipour, P., Marcotte, D., Chouteau, M., and Keating, P.: 3D stochastic inversion of gravity data using cokriging and cosimulation, *Geophysics*, 75, I1–I10, 2010.
- Shmeryan, L. and Journel, A.: Two Markov models and their application, *Math. Geol.*, 31, 965–988, 1999.
- Solow, A. R.: Mapping by simple indicator kriging, *Math. Geol.*, 18, 335–352, 1986.
- Strebel, S.: Conditional simulation of complex geological structures using multiple-point statistics, *Math. Geol.*, 34, 1–21, 2002.
- Torrence, C.: DC Polar Stereographic Projection lon/lat conversion: polar\_convert, <https://github.com/nsidc/polarstereo-lonlat-convert-py/> (last access: 1 March 2023), 2019.
- Van Rossum, G., Warsaw, B., and Coghlan, N.: PEP 8–style guide for python code, Python.org, 1565, <https://peps.python.org/pep-0008/> (last access: 1 September 2022), 2001.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, I., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors: SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nat. Methods*, 17, 261–272, 2020.
- Volkova, A. and Merkulov, V.: Iterative Approach of Gravity and Magnetic Inversion through Geostatistics, in: *Petroleum Geostatistics 2019*, vol. 2019, European Association of Geoscientists & Engineers, 1–5, <https://doi.org/10.3997/2214-4609.201902257>, 2019.
- Wernecke, A., Edwards, T. L., Holden, P. B., Edwards, N. R., and Cornford, S. L.: Quantifying the impact of bedrock topography uncertainty in Pine Island Glacier projections for this century, *Geophys. Res. Lett.*, 49, e2021GL096589, <https://doi.org/10.1029/2021GL096589>, 2022.
- Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., et al.: The FAIR Guiding Principles for scientific data management and stewardship, *Scientific Data*, 3, 1–9, 2016.
- Xiong, X., Grunwald, S., Myers, D. B., Kim, J., Harris, W. G., and Bliznyuk, N.: Assessing uncertainty in soil organic carbon modeling across a highly heterogeneous landscape, *Geoderma*, 251, 105–116, 2015.
- Yin, Z., Zuo, C., MacKie, E. J., and Caers, J.: Mapping high-resolution basal topography of West Antarctica from radar data using non-stationary multiple-point geostatistics (MPS-BedMappingV1), *Geosci. Model Dev.*, 15, 1477–1497, <https://doi.org/10.5194/gmd-15-1477-2022>, 2022.
- Youngman, B. D. and Stephenson, D. B.: A geostatistical extreme-value framework for fast simulation of natural hazard events, *P. Roy. Soc. A-Math. Phys.*, 472, 20150855, <https://doi.org/10.1098/rspa.2015.0855>, 2016.
- Zuo, C., Yin, Z., Pan, Z., MacKie, E. J., and Caers, J.: A Tree-Based Direct Sampling Method for Stochastic Surface and Subsurface Hydrological Modeling, *Water Resour. Res.*, 56, e2019WR026130, <https://doi.org/10.1029/2019WR026130>, 2020.