

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

**7,100**

Open access books available

**189,000**

International authors and editors

**205M**

Downloads

**154**

Countries delivered to

**TOP 1%**

most cited scientists

**12.2%**

Contributors from top 500 universities



**WEB OF SCIENCE™**

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.

For more information visit [www.intechopen.com](http://www.intechopen.com)



## Chapter

# GeostatsPy: Open-Source Geostatistics in Python

*Michael J. Pyrcz*

## Abstract

GSLIB: Geostatistical Library by Clayton V. Deutsch and Andre G. Journel is the original spatial data analytics, geostatistics, open-source library built in FORTRAN with a function-based implementation to maximize workflow construction ease and flexibility. From simple toy problems for education to complicated subsurface model workflows, GSLIB is up to the task. Yet, it is difficult to teach the next generation with FORTRAN executables and PostScript visualizations. While there are a variety of efforts to add geostatistical methods to Python, I failed to find a package to meet my pedagogical needs in the modern Python language. I was compelled to reimplement GSLIB, function-by-function, often the nights before the associated lectures were given, to support my students. For reliability, I committed to rely only on the most common Python packages, such as NumPy, Pandas, SciPy and Numba. Yes, I took shortcuts, the methods are generally only available for 2D and there are missed opportunities to leverage existing code and to further accelerate for faster run times. The good news, it's an open-source project, so if you see an opportunity to contribute you are most welcome. Participating in this project further expanded my respect for the vision and contributions of the original authors, Professors Clayton V. Deutsch and Andre G. Journel.

**Keywords:** geostatistics, spatial data analytics, uncertainty modeling, spatial prediction, open source

## 1. Introduction

Geostatistics is an applied branch of statistics developed to support spatial, often subsurface, characterization and modeling through the integration of the spatial context and domain expertise, spatial continuity, data and model scale, and uncertainty. The fundamental geostatistical methods and workflows for characterization and uncertainty modeling through spatial estimation and simulation paradigms have been critical for optimum management of spatial resources, including agriculture, water, minerals and hydrocarbons. In the digital revolution with enhanced focus on data analytics and machine learning, interest in geostatistics has expanded as a tool for feature engineering and data integration aspects of machine learning workflows [1, 2] and with various hybrid methods that merge geostatistics and machine learning to build and diagnose models [3, 4].

Access to open-source, well-maintained codes for geostatistical methods is critical to support educating the next generation of scientists and engineers, along with supporting

professional development and rapid prototyping for efficient innovation cycles. GSLIB: Geostatistical Library provided the first well-documented compilation of geostatistics codes [5]. Through these codes, generations of ‘geostatisticians’ learned the necessary tradecraft through experiential learning backed by an accessible treatment of theory and expertise presented clearly and concisely in the user guide. The function and subfunction structure of the library results in approachable codes that support ‘look at the code’ learning, the words that I often heard during my Ph.D. with Professor Clayton Deutsch. A simple standardized data file (GeoEAS) and parameter file (also ASCII text) provide a fully plug and play, editable approach for characterization and modeling workflow construction. Those who use GSLIB are only limited by their own imagination.

The first class that I taught at The University of Texas at Austin in the fall of 2017 was based on GSLIB. Many students struggled with the various issues of utilizing the FORTRAN executables and PostScript graphics outputs on modern computers. It was apparent that my students needed geostatistical functionality, the GSLIB vision, in a modern programming language, and development and prototyping environment. I attempted to substitute various R and Python packages but I ran into various issues. Codes that were not maintained broke the morning of lectures due to updates with the package dependencies, packages that departed from the basic function approach were difficult to learn and apply and detracted from teaching the fundamentals. Other codes were not open-source and prevented students from having access or seeing the code.

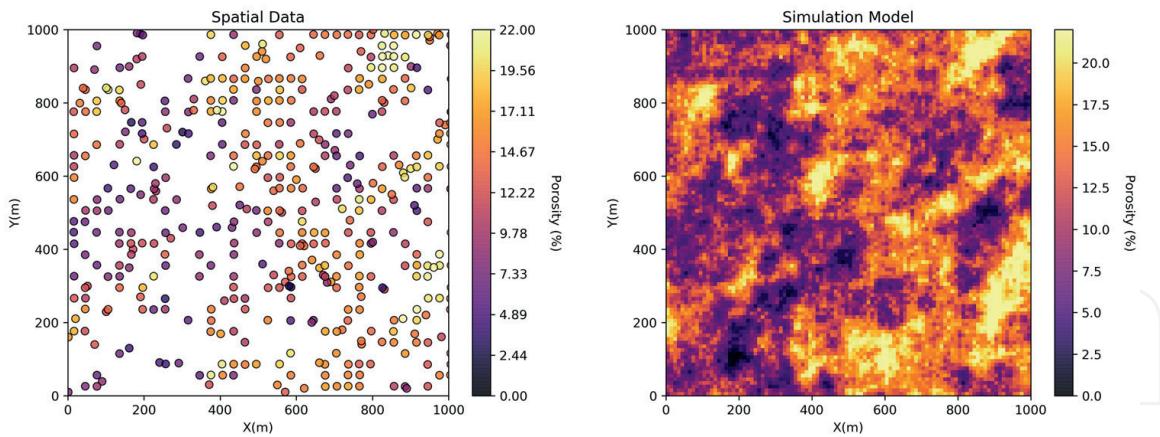
Given all of this, I decided to develop the GeostatsPy package [6]. Function-by-function I translated and adapted GSLIB to Python often the evening before the algorithm was needed in class. The students are my alpha testers and I have benefited from their unlimited ability to find issues in codes by exploring the combinatorial space of unexpected inputs and breaking things. Some students jumped in and have assisted with the open-source project (see the Acknowledgments). There have been many opportunities to use standard Python packages like SciPy [7], NumPy [8], Pandas [9] and matplotlib [10] for critical functionality to avoid the case of reinventing the wheel in GSLIB, and to optimize and simplify the code.

Urgency has motivated some shortcuts. For example, while the Numba package [11] is applied for code acceleration and could be applied more extensively along with improved utilization of available packages for code optimization. I decided to reduce most functions to 2D, a decision based on the prioritization of education and ease of visualization. In many ways, I feel GeostatsPy is not yet worthy of the greatness of the GSLIB legacy, but many classes with hundreds of students have validated that GeostatsPy has met its goal to make geostatistics accessible in a modern platform. There is a lot more work to do and if you see opportunity to improve GeostatsPy, here’s an invitation to participate in this open-source project.

In the next sections, I demonstrate functionality of GeostatsPy, including data visualization, data preparation, estimation, simulation and model summarization. There is also functionality to support hybrid Python and FORTRAN workflows that continue to utilize GSLIB executables in Python workflows. I also provide information on a large library of well-documented demonstration workflows and interactive workflow to support education.

## **2. Data visualization**

The ability to visualize our data is an essential prerequisite for any geostatistical study. The matplotlib package provides unlimited ability to draft effective figures,

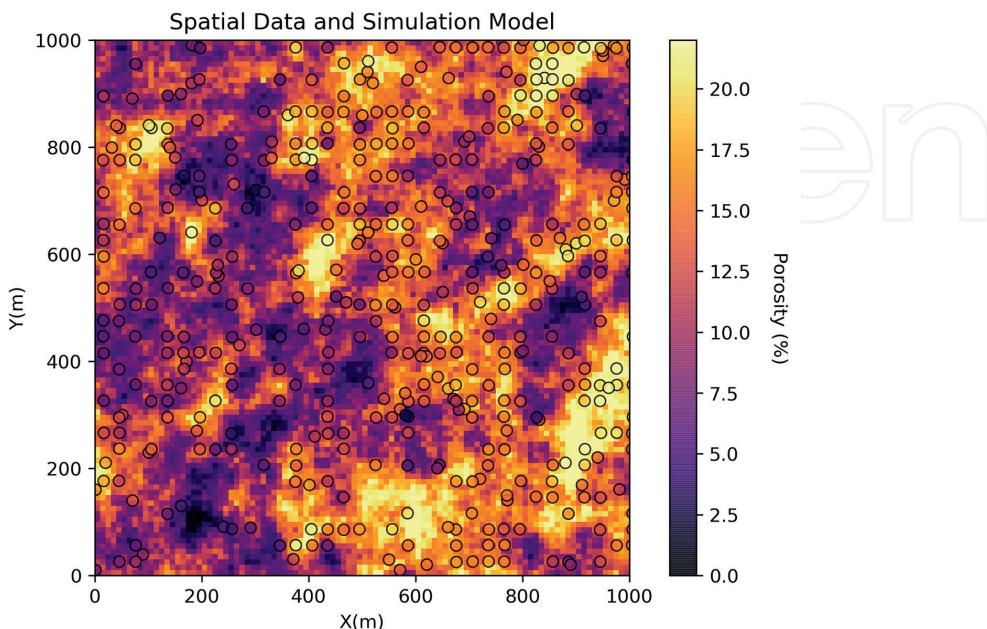


**Figure 1.**  
*Example spatial data and model plots, including location map (locmap) (left) and pixel plot (pixelplt) (right).*

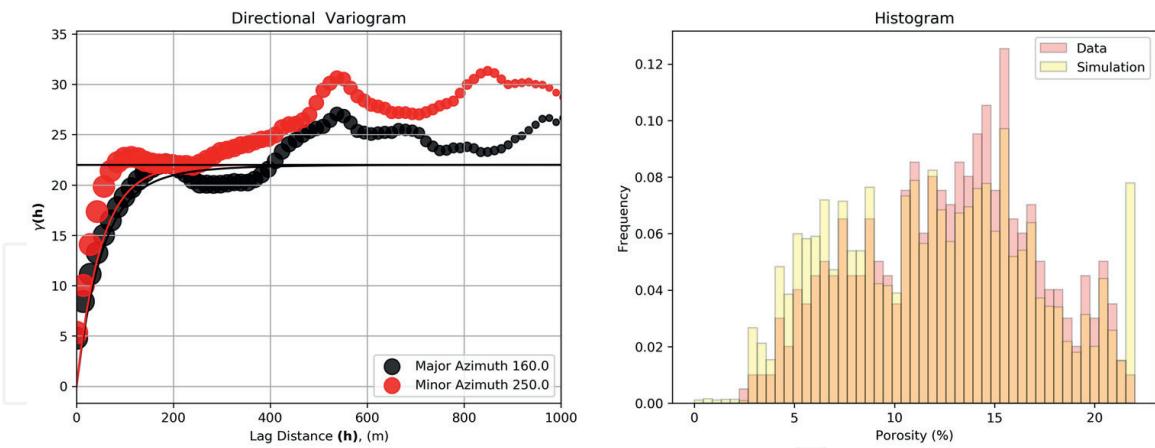
and to publish them in any required file format and resolution. Yet, the length of code and details required to build figures may be a barrier for some busy students and working professionals. GSLIB provided simple, robust, canned functions for common plots, such as histograms (histplt), location plots (locplot), pixel plots (pixelplt) and variogram plots (varplot). They all use standard input arguments like trimming limits, color bar range and labels for fast adoption and compact workflows.

GeostatsPy provides reimplementations of these basic plotting functions with the standard GSLIB arguments as a bridge to those accustomed with GSLIB and to support a compact and readable code. For example, the pixel plot function includes the input parameters' array, X and Y extents, cell size, feature extents, title, X and Y labels like the GSLIB function. Examples of a location map (locmap) and pixel plot (pixelplt) are shown in **Figure 1**.

It is convenient to superimpose location maps on pixel plots, for example to compare an interpolated spatial model to the spatial data. A combined location map



**Figure 2.**  
*Example of combined spatial data and model plot, including location map superimposed on a pixel plot (locpix).*



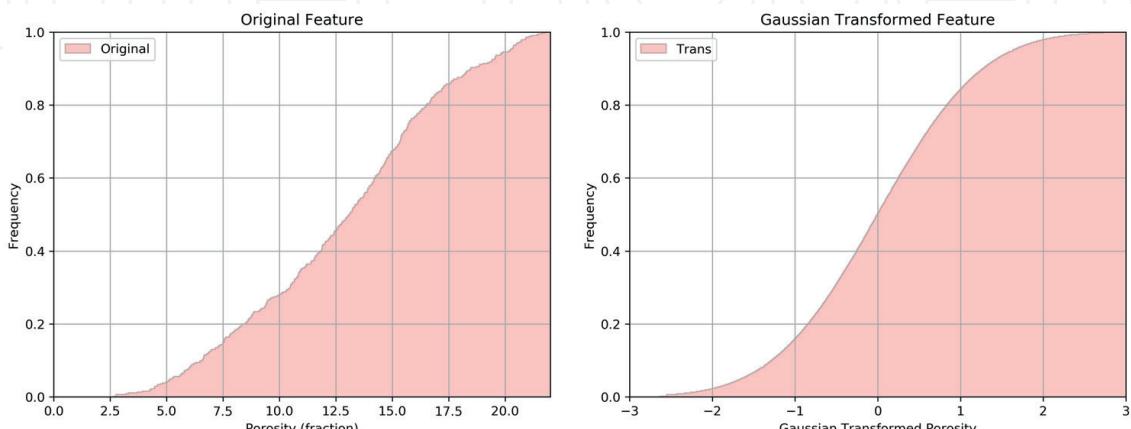
**Figure 3.**  
An example histogram and variogram plot with experimental and model variograms shown.

and pixel plot is shown in **Figure 2**. An example histogram (histplt) and experimental variogram and variogram model plot (vargplot) are shown in **Figure 3**. It is generally possible to superimpose plots with consistent axes. For example, in **Figure 3**, two histograms are combined to compare the distributions from the input data and a simulated realization (same data and model shown in **Figure 2**).

### 3. Data preparation and analysis

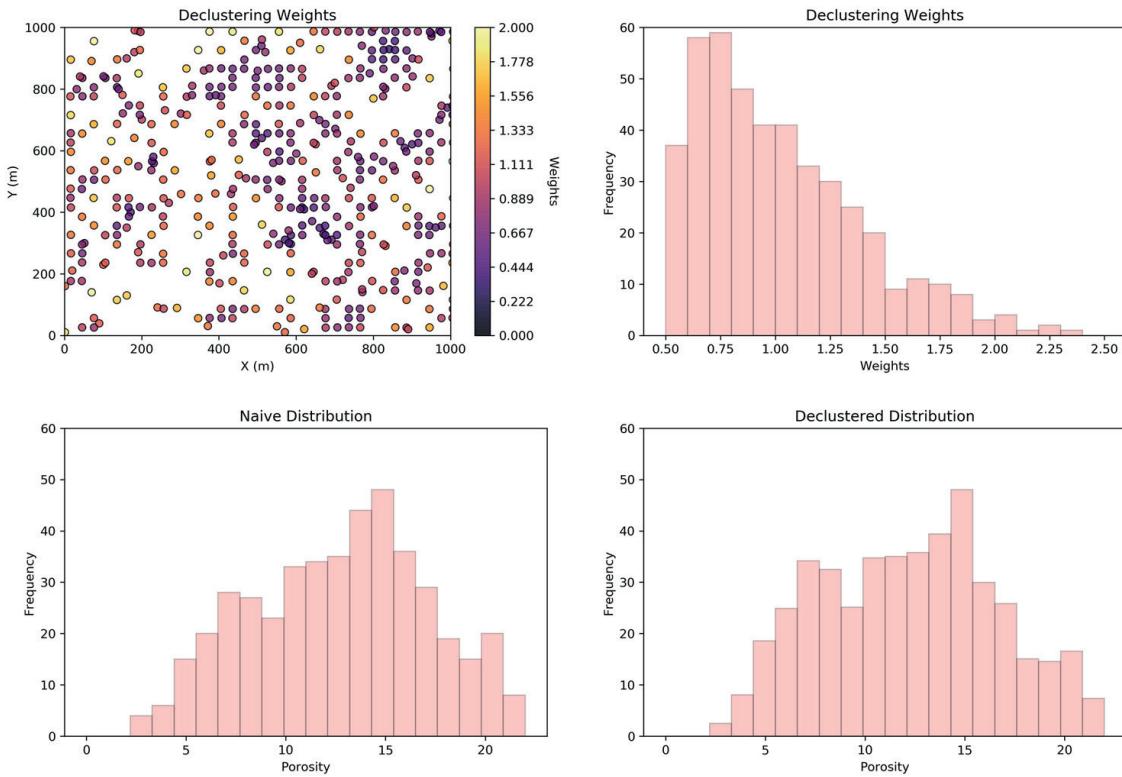
There are various data preparation functions available in GSLIB and commonly applied in geostatistical workflows, such as distribution transformations and data declustering. These have been implemented in GeostatsPy with GSLIB consistent arguments.

An example of Gaussian transformation, a quantile-quantile transformation to a standard normal distribution, known as Gaussian anamorphosis, is shown in **Figure 4**. An example of cell-based declustering is shown in **Figure 5** with the weights location map and weights histogram for model quality control, along with the naïve and declustered (weighted) distributions. Further diagnostics are possible as the

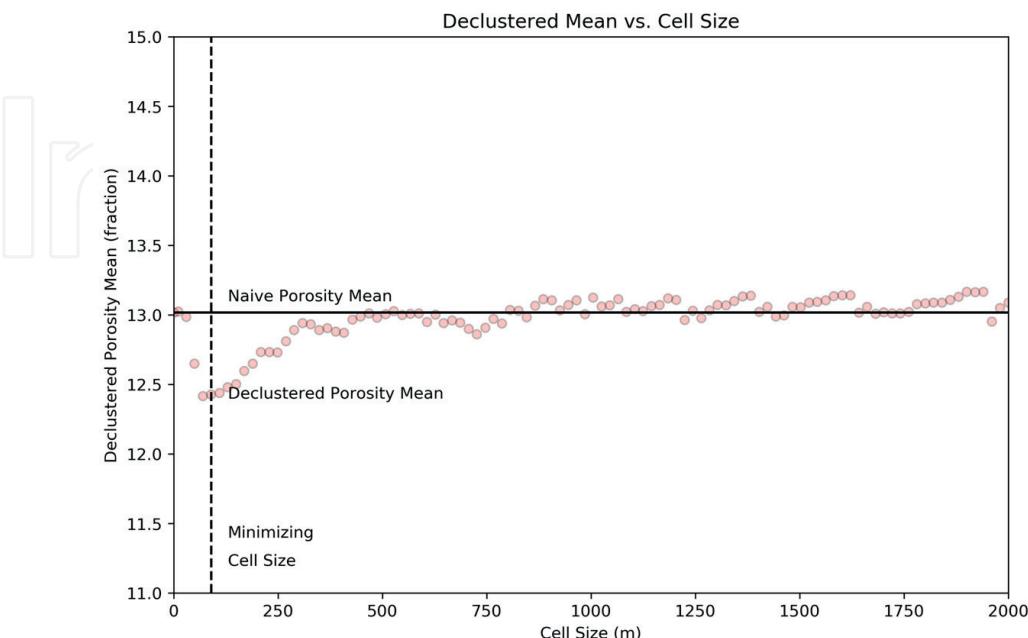


**Figure 4.**  
An example of Gaussian transformation with the original feature (left) and Gaussian transformed (right) cumulative distribution functions.

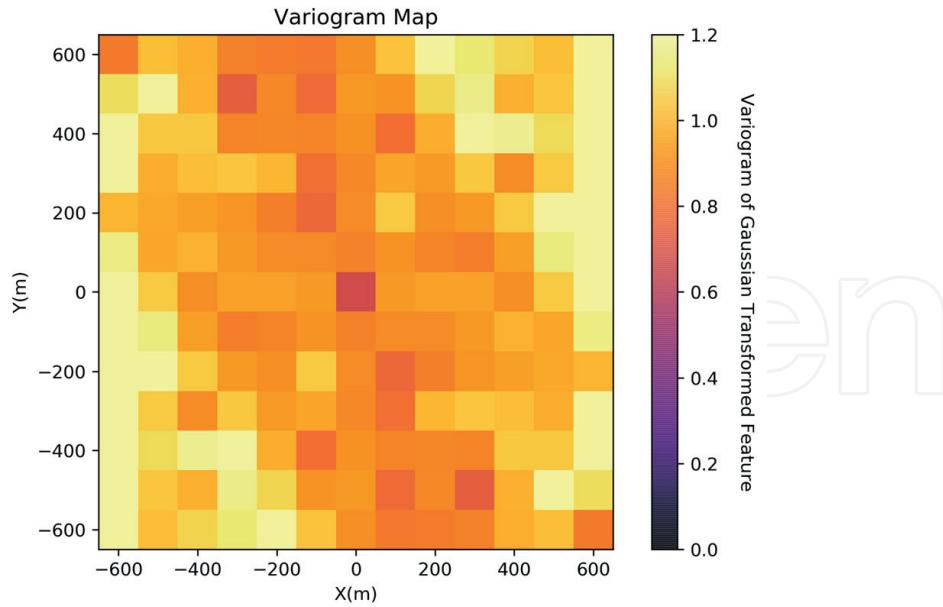
GeostatsPy declustering function returns the 1D ndarrays with the cell sizes and associated declustered means. With matplotlib, it is straightforward to produce the standard plot from geostatistics textbooks (**Figure 6**) [12].



**Figure 5.**  
*Location map of the cell-based declustering weights (top left), histogram of declustering weights (top right), naïve data distribution (bottom left) and weighted data distribution (bottom right).*



**Figure 6.**  
*Plot of declustering mean versus cell size for determination of optimum cell size and quality control of declustering weights.*



**Figure 7.**  
Variogram map of the Gaussian transformed feature. See **Figure 1** for the original feature location map and **Figure 3** for the directional variograms.

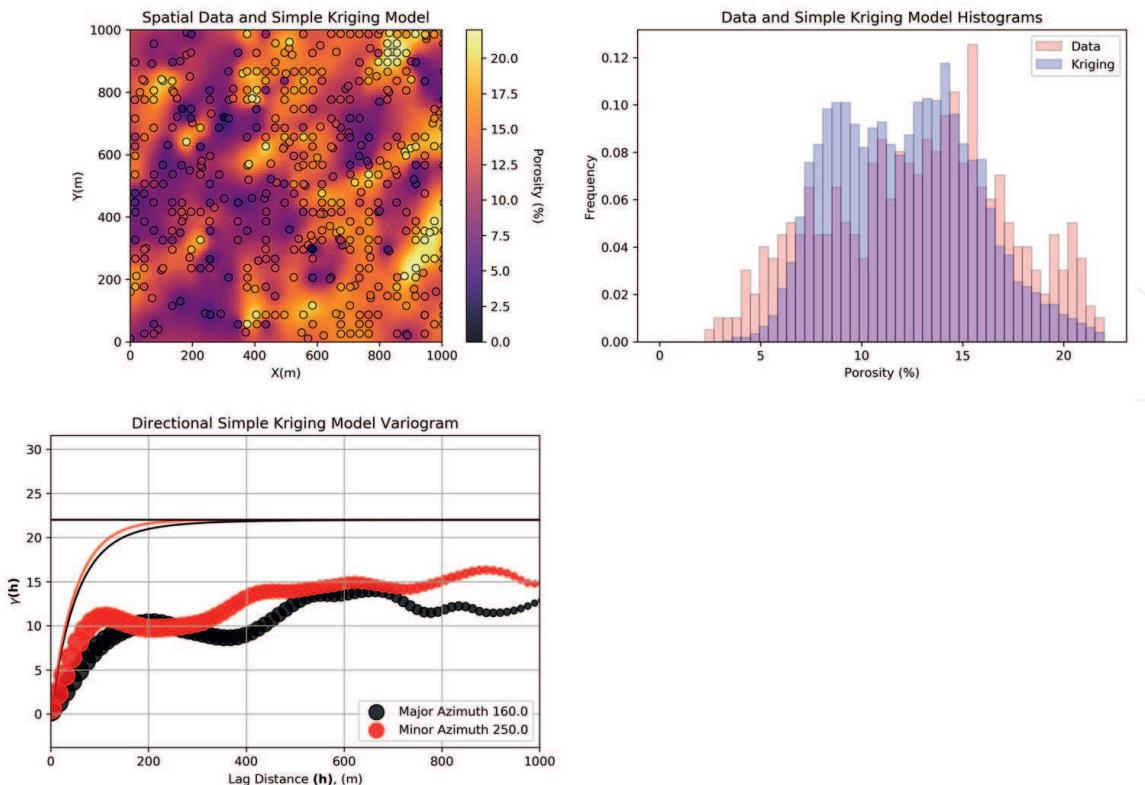
Functions for spatial continuity characterization and modeling, including variogram calculation for irregular spatial data (`gamv`) and gridded spatial data (`gam`), variogram maps (`vmap`), and variogram model visualization (`vmodel`), are available in GeostatsPy. Also, a convenience class `vario` is available to store and pass variogram model parameters to variogram visualization, and spatial estimation and simulation functions. See **Figure 3** for an example of experimental direction variogram and associated 2D variogram model and see **Figure 7** for the associated variogram map.

#### 4. Spatial estimation

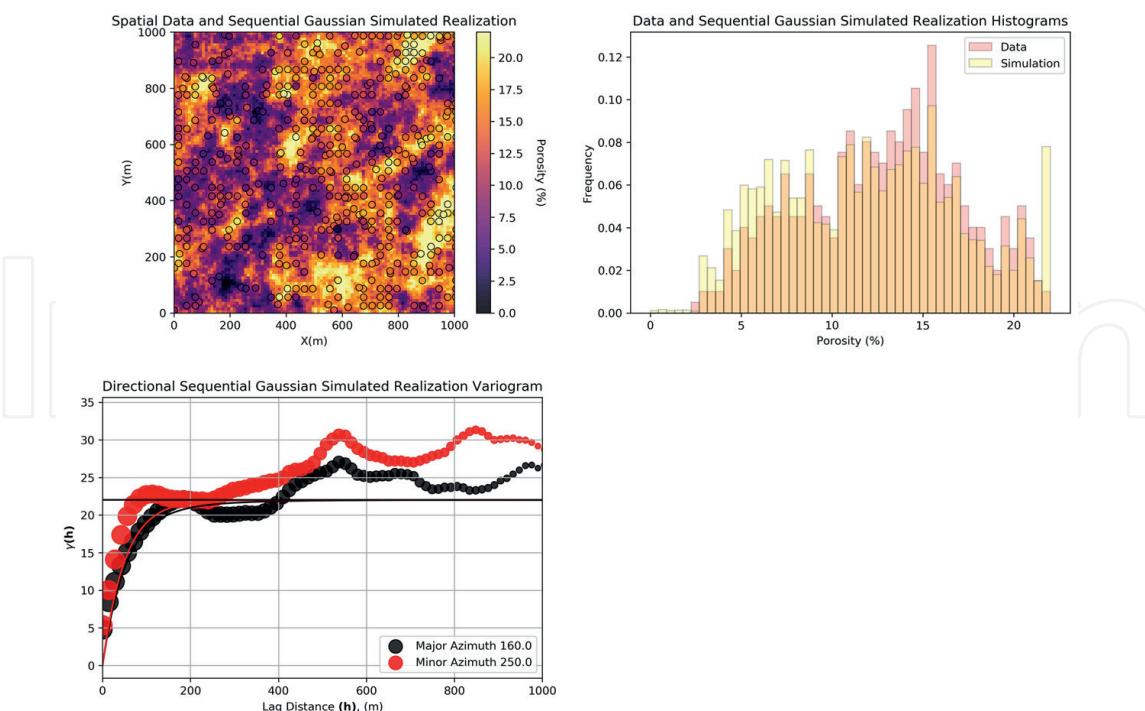
Simple, ordinary and indicator kriging are available along with the ability to integrate trend models, nonstationary means or proportions. An example of simple kriging with the spatial data from **Figure 1** and variogram model from **Figure 3** is shown in **Figure 8**. Two components of minimum acceptance tests from Leuangthong [13] are included to demonstrate model checking.

#### 5. Spatial simulation

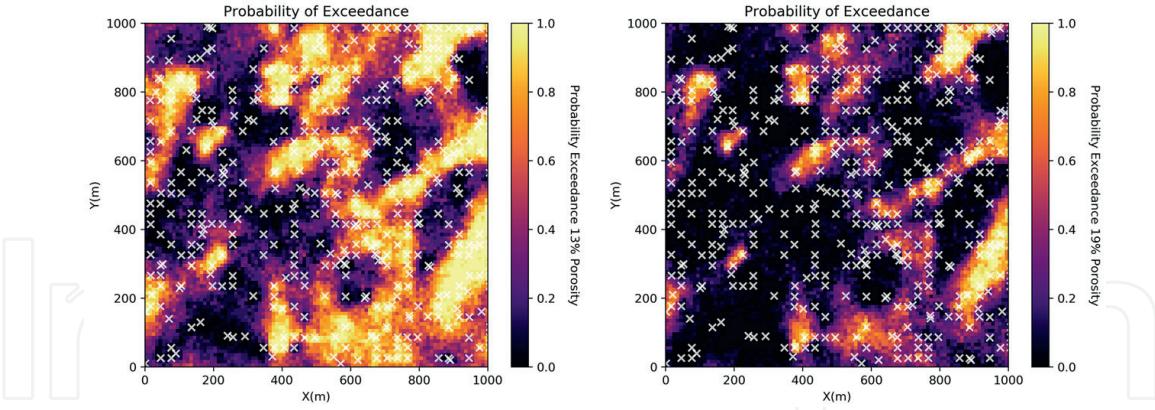
Sequential Gaussian and sequential indicator simulation are available along with the ability to integrate trend models, nonstationary means or proportions. An example of sequential Gaussian simulation with the spatial data from **Figure 1** and variogram model from **Figure 3** is shown in **Figure 8**. Once again, two components of minimum acceptance tests from Leuangthong [13] are included to demonstrate model checking (**Figure 9**).



**Figure 8.**  
*Simple kriging estimates with the spatial data (top left), histogram of original data and simple kriging estimates (top right) and variogram model and experimental variograms calculated over the estimates.*



**Figure 9.**  
*Sequential Gaussian simulated realization with the spatial data (top left), histogram of the original data and simulated realization (top right) and variogram model and experimental variograms calculated over the simulated realization.*

**Figure 10.**

Uncertainty model summarization with probability of exceedance calculated from 20 realizations of sequential Gaussian simulation based on two thresholds, 13% (left) and 19% (right). Well data are indicated with white 'x's.

## 6. Model summarization

Functions for model summarization are available to support uncertainty management. These functions produce summary models integrated over multiple realizations, including summaries of the local model expectation, conditional variance, percentiles and probability of exceedance. Twenty realizations are calculated and applied to calculate the probability of exceeding two porosity thresholds in **Figure 10**.

## 7. GSLIB compatibility

It may be beneficial to integration of GSLIB with hybrid Python and FORTRAN workflows to integrate the computationally efficient, robust and more complete functionality of GSLIB in the powerful Python ecosystem. To support these hybrid workflows, GeostatsPy includes functions to read and write GeoEAS file formats (the standard file format used by GSLIB) for gridded data for NumPy’s ndarray class and tabular data for Pandas’ DataFrame class. In addition, there is a set of wrappers that write GSLIB parameter files, execute the GSLIB FORTRAN executables and read the outputs back to Python workflows. Of course, more sophisticated code-based integration is possible, but these simple methods may be helpful.

## 8. Example workflows

To support the wide adoption, I have developed a comprehensive set of well-documented Python Jupyter Notebook files to demonstrate basic workflows for geostatistical analysis and modeling with GeostatsPy. All these workflows are available in my GeostatsPyDemos GitHub repository [14]. Some of the demonstration workflows for common modeling tasks are presented in **Table 1**.

## 9. Interactive workflows

To further support education, I have utilized GeostatsPy to develop a set of interactive workflows with ipywidgets from matplotlib. This includes interactive simple

Data distributions	GeostatsPy_datadistributions.ipynb
Data transformations	GeostatsPy_transformations.ipynb
Spatial data plotting	GeostatsPy_plottingdata.ipynb
Cell-based declustering	GeostatsPy_declustering.ipynb
Variogram calculations	GeostatsPy_variogram_calculation.ipynb
Variogram modeling	GeostatsPy_variogram_modeling.ipynb
Kriging	GeostatsPy_kriging.ipynb
Simulation	GeostatsPy_simulation.ipynb
Model summarization	GeostatsPy_simulation_postsim.ipynb

**Table 1.**

*A list of well-documented demonstration Python Jupyter Notebook workflows for the application of GeostatsPy for common spatial, subsurface modeling tasks.*

kriging with the ability to change the data locations and variogram model parameters in real-time and to graphically observe the impact on data kriging weights, and kriging estimates and estimation variance. In another demonstration, cell-based declustering is shown with the ability to stay interactive in real-time with the cell size while observing the impact on the data declustering weights. The interactive variogram calculation and modeling workflow is quite useful for teaching and learning variograms. These workflows are available in my PythonNumericalDemos GitHub repository [15] with Jupyter Notebook files with names ‘Interactive\_\*.ipynb’.

## 10. Conclusions

GeostatsPy is an open-source Python package developed to support teaching, learning and prototyping with geostatistics in Python-based data science, data analytics and machine learning workflows. All are welcome to contribute to the open-source project.

## Acknowledgements

I would like to acknowledge the following individuals who have contributed to the development of GeostatsPy, Honggeun Jo, Wendi Liu, Anton Kupenko, Alex Gigliotti, Travis Salomaki and Javier Santos. I would also like to acknowledge the industry supporters of my industrial affiliates program, DIRECT, including Aramco Americas, BP, Chevron, JAPEX, JOGMEC, Shell, TotalEnergies, Equinor, ExxonMobil, Woodside Energy, Chord Energy, and Coterra Energy.

InTechOpen

InTechOpen

### **Author details**

Michael J. Pyrcz

The University of Texas at Austin, Austin, Texas, USA

\*Address all correspondence to: [mpyrcz@austin.utexas.edu](mailto:mpyrcz@austin.utexas.edu)

**InTechOpen**

---

© 2024 The Author(s). Licensee InTechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## References

- [1] Liu W, Pyrcz MJ. A spatial correlation-based anomaly detection method for subsurface modeling. *Mathematical Geosciences*. 2021;53(5):809-822. DOI: 10.1007/s11004-020-09892-z
- [2] Jo H, Pyrcz MJ. 2020, robust rule-based aggradational lobe reservoir models. *Natural Resources Research*. 2020;29(2):1193-1213. DOI: 10.1007/s11053-019-09482-9
- [3] Jo H, Pan W, Santos JE, Jung H, Pyrcz MJ. Machine learning assisted history matching for a deepwater lobe system. *Journal of Petroleum Science and Engineering*. 2021;207:109086. DOI: 10.1016/j.petrol.2021.109086
- [4] Pan W, Torre-Verdin C, Pyrcz MJ. Stochastic Pix2pix: A new machine learning method for geophysical and well conditioning of rule-based channel reservoir models. *Natural Resources Research*. 2021;30(2):1319-1345. DOI: 10.1007/s11053-020-09778-1
- [5] Deutsch CV, Journel AG. *GSLIB: Geostatistical Software Library and User's Guide*. 2nd ed. New York, NY, USA: Oxford University Press; 1997. 384 p. DOI: 10.1080/00401706.1995.10485913
- [6] Pyrcz MJ, Jo H, Kupenko A, Liu W, Gigliotti AE, Salomaki T, Santos J. *GeostatsPy Python Package*, PyPI, Python Package Index. 2021. Available from: <https://pypi.org/project/geostatspy/>
- [7] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D. SciPy 1.0 contributors SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*. 2020;17:261-272. DOI: 10.1038/s41592-019-0686-2
- [8] Harris CR, Millman KJ, van der Walt SJ. Array programming with NumPy. *Nature*. 2020;585:357-362. DOI: 10.1038/s41586-020-2649-2
- [9] McKinney W. Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference; 28 June–3 July 2010; Austin. SciPy; 2010. pp. 51-56. DOI: 10.25080/Majora-92bf1922-00a. Available from: <http://conference.scipy.org.s3-website-us-east-1.amazonaws.com/proceedings/scipy2010/index.html>
- [10] Hunter JD. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. 2007;9(3):90-95. DOI: 10.1109/MCSE.2007.55
- [11] Lam SK, Pitrou A, Seibert S. Numba: A LLVM-based Python JIT compiler. In: Proceedings of the 2<sup>nd</sup> Workshop on the LLVM Compiler Infrastructure in HPC; 15 November 2015; Austin. LLVM; 2015. New York, NY, USA: Association for Computing Machinery; 2015. pp. 1-6. DOI: 10.1145/2833157.2833162
- [12] Pyrcz MJ, Deutsch CV. *Geostatistical Reservoir Modeling*. 2nd ed. New York, NY, USA: Oxford University Press; 2004. 448 p. DOI: 10.1198/tech.2005.s339
- [13] Leuangthong O, McLennan JA, Deutsch CV. Minimum acceptance criteria for geostatistical realizations. *Natural Resources Research*. 2004;13:131-141. DOI: 10.1023/b:narr.0000046916.91703.bb
- [14] GeostatsPyDemos [Internet]. 2021. Available from: <https://github.com/GeostatsGuy/GeostatsPyDemos> [Accessed: 20 November 2021]
- [15] PythonNumericalDemos [Internet]. 2021. Available from: [www.github.com/GeostatsGuy/PythonNumericalDemos](http://www.github.com/GeostatsGuy/PythonNumericalDemos) [Accessed: 20 November 2021]