The Journal of Open Source Software

GaussianRandomFields.jl: A Julia package to generate and sample from Gaussian random fields

Pieterjan Robbe [1]

1 KU Leuven, Belgium

DOI: 10.21105/joss.05595

Software

- Review
- Repository
- Archive

Editor: Mehmet Hakan Satman

Reviewers:

- @ziyiyin97
- @shahmoradi

Submitted: 27 May 2023
Published: 31 August 2023

Summary

Random fields are used to represent spatially-varying uncertainty, and are commonly used as training data in uncertainty quantification and machine learning applications. GaussianRandomFields.jl is a Julia (Bezanson et al., 2017) software package to generate and sample from Gaussian random fields. It offers support for well-known covariance functions, such as Gaussian, exponential and Matérn covariances (Bishop & Nasrabadi, 2006; Chiles & Delfiner, 2012; Montero et al., 2015), as well as user-defined covariance structures defined on arbitrary domains. The package implements most common methods to generate samples from these random fields, including the Cholesky factorization, the Karhunen-Loève expansion, and the circulant embedding method (Lord et al., 2014). GaussianRandomFields.jl makes use of Plots.jl (Christ et al., 2023) to quickly visualize samples of the random fields.

Statement of need

Random fields are used by scientists to describe complex patterns and structures emerging in nature. They provide a statistical tool for describing a vast amount of different structures found in various applications such as electronics (Cui & Zhang, 2018), geostatistics (Pirot et al., 2015), machine learning (Stephenson & Chen, 2006) and cosmology (Chiang & Coles, 2000). Random fields can be viewed as an extension from random variables to random functions, in the sense that the random field takes random values at each point in the domain where it is defined. Gaussian random fields are particularly attractive, because they only require two parameters to be fully specified: a mean value and a covariance function. GaussianRandomFields.jl provides Julia implementations of Gaussian random fields with stationary separable and non-separable isotropic and anisotropic covariance functions. It has been used in a number of recent works, including (Blondeel et al., 2020), (Robbe et al., 2021) and (Wu et al., 2023).

Other packages for Gaussian random field generation are available in R (Schlather, 2022) and Python (Müller et al., 2022). GaussianRandomFields.jl offers a native Julia implementation. As such, it benefits from the performance advantage of Julia, see (Bezanson et al., 2017), and provides a convenient unified API for different covariance functions by leveraging multiple dispatch. A particular example are the covariance functions from KernelFunctions.jl, which can easily be linked to the Gaussian random field generators implemented in this package.

Usage

The full API of GaussianRandomFields.jl is described in detail in the documentation. We also provide a tutorial with various examples detailing how to define, sample from, and visualize Gaussian random fields. The following example is an excerpt from the tutorial. We refer to Figure 1 for an illustration.

```julia
using GaussianRandomFields, Plots

cov = CovarianceFunction(2, Exponential(.5))
pts = range(0, stop=1, length=1001)
grf = GaussianRandomField(cov, CirculantEmbedding(), pts, pts, minpadding=2001)
heatmap(grf)
```

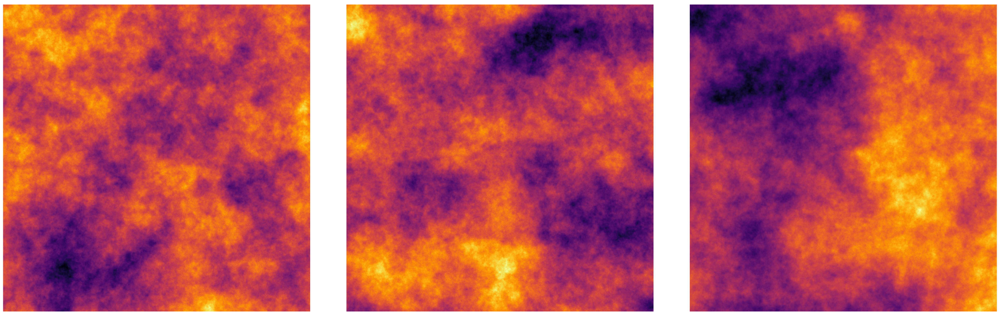

Figure 1: Three realizations of a two-dimensional Gaussian random field with exponential covariance function.

Acknowledgements

The author acknowledges the help of David Widmann implementing a dimension-independent variant of the circulant embedding method using `@generated` functions.

References

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4).

Blondeel, P., Robbe, P., Van hoorickx, C., François, S., Lombaert, G., & Vandewalle, S. (2020). P-refined multilevel quasi-Monte Carlo for Galerkin finite element methods with applications in civil engineering. *Algorithms*, *13*(5), 110. https://doi.org/10.3390/a13050110

Chiang, L.-Y., & Coles, P. (2000). Phase information and the evolution of cosmological density perturbations. *Monthly Notices of the Royal Astronomical Society*, *311*(4), 809–824. https://doi.org/10.1046/j.1365-8711.2000.03086.x

Chiles, J.-P., & Delfiner, P. (2012). *Geostatistics: Modeling spatial uncertainty* (Vol. 713). John Wiley & Sons. https://doi.org/10.1016/s0098-3004(00)00063-7

Christ, S., Schwabeneder, D., Rackauckas, C., Borregaard, M. K., & Breloff, T. (2023). *Plots.jl – A user extendable plotting API for the Julia programming language*. https://doi.org/10.5334/jors.431

Cui, C., & Zhang, Z. (2018). Uncertainty quantification of electronic and photonic ICs with non-Gaussian correlated process variations. *Proceedings of the International Conference on Computer-Aided Design*, 1–8. https://doi.org/10.1145/3240765.3240860

Lord, G. J., Powell, C. E., & Shardlow, T. (2014). *An introduction to computational stochastic PDEs* (Vol. 50). Cambridge University Press.

Montero, J.-M., Fernández-Avilés, G., & Mateu, J. (2015). *Spatial and spatio-temporal geostatistical modeling and kriging*. John Wiley & Sons.

Müller, S., Schüler, L., Zech, A., & Heße, F. (2022). `GSTools` v1.3: A toolbox for geostatistical modelling in Python. *Geoscientific Model Development*, *15*(7), 3161–3182. https://doi.org/10.5194/gmd-15-3161-2022

Pirot, G., Renard, P., Huber, E., Straubhaar, J., & Huggenberger, P. (2015). Influence of conceptual model uncertainty on contaminant transport forecasting in braided river aquifers. *Journal of Hydrology*, *531*, 124–141. https://doi.org/10.1016/j.jhydrol.2015.07.036

Robbe, P., Nuyens, D., & Vandewalle, S. (2021). Enhanced multi-index Monte Carlo by means of multiple semicoarsened multigrid for anisotropic diffusion problems. *Numerical Linear Algebra with Applications*, *28*(3), e2281. https://doi.org/10.1002/nla.2281

Schlather, M. (2022). RandomFields. In *CRAN repository*. CRAN. https://cran.r-project.org/web/packages/RandomFields/index.html

Stephenson, T. A., & Chen, T. (2006). Adaptive Markov random fields for example-based super-resolution of faces. *EURASIP Journal on Advances in Signal Processing*, *2006*, 1–11. https://doi.org/10.1155/asp/2006/31062

Wu, H., Greer, S. Y., & O'Malley, D. (2023). Physics-embedded inverse analysis with algorithmic differentiation for the Earth's subsurface. *Scientific Reports*, *13*(1), 718. https://doi.org/10.1038/s41598-022-26898-1