

# Ergodicity Library: Integrated stochastic-process simulation, ergodicity diagnostics, and agent-based experimentation in Python

Ihor Kendiukhov<sup>1¶</sup>

<sup>1</sup> University of Tübingen, Germany ¶ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: ↗

Submitted: 10 February 2026

Published: unpublished

## License

Authors of papers retain copyright and release the work under a

Creative Commons Attribution 4.0

International License ([CC BY 4.0](#))

## Summary

`ergodicity_library` is an open-source Python package for research workflows centered on stochastic dynamics, especially settings where time-average behavior and ensemble-average behavior can diverge (Peters, 2019; Peters & Gell-Mann, 2016). The package integrates three layers that are often handled separately in ad hoc scripts: process simulation, diagnostics/analysis tools, and agent-based decision experiments.

In practice, many projects require repeated transitions between model definition, trajectory generation, fitting/diagnostic checks, and decision evaluation. The core contribution of this software is to make these transitions operationally coherent in one environment, reducing glue-code overhead and improving reproducibility (Kendiukhov, 2024, 2025).

The package supports both baseline and advanced use-cases: Brownian and related processes, heavy-tailed families (including Lévy-stable variants), multiplicative growth settings, memory-dependent dynamics, and selected stochastic field simulations. This functionality is designed for researchers who need to compare assumptions quickly and inspect consequences with consistent interfaces.

## Statement of need

Research questions in quantitative finance, economics, physics-inspired modeling, and computational social science frequently involve nontrivial stochastic assumptions. Typical workflows are fragmented: one code path for simulation, another for diagnostics, another for fitting, and often a separate implementation for decision logic. This fragmentation increases implementation time and makes it harder to compare model variants under consistent conditions.

The specific need addressed by `ergodicity_library` is an integrated software framework for studies where process assumptions, transient diagnostics, and downstream decisions should be tested together. The intended users include applied researchers, graduate-level users, and research software practitioners who need to:

- simulate heterogeneous process families using a shared API,
- compare additive versus multiplicative dynamics,
- evaluate Gaussian versus heavy-tailed assumptions,
- inspect transient/preasymptotic behavior before asymptotic claims,
- couple stochastic dynamics to utility/portfolio/agent rules.

This package is not framed as a new mathematical theory. It is a research software contribution that operationalizes existing theory and methods into reusable computational workflows.

## State of the field

The scientific Python ecosystem provides high-quality numerical building blocks, notably NumPy, SciPy, SymPy, and Matplotlib (Harris et al., 2020; Hunter, 2007; Meurer et al., 2017; Virtanen et al., 2020). These tools are foundational and intentionally general.

For stochastic simulation specifically, packages such as stochastic, sdeint, and sdepy provide useful focused capabilities (sdeint developers, 2026; sdepy developers, 2026; stochastic developers, 2026). They are valuable for generating paths or integrating SDEs, but generally do not aim to unify process abstraction, ergodicity-oriented diagnostics, fitting workflows, and agent-based experimentation under one research workflow model.

ergodicity\_library was developed as a standalone package rather than a thin wrapper because its primary use-case depends on this cross-layer integration. The project targets a gap between low-level numerical primitives and narrowly scoped simulation utilities: comparative stochastic research where model choice, diagnostics, and decisions must be iterated jointly.

## Software design

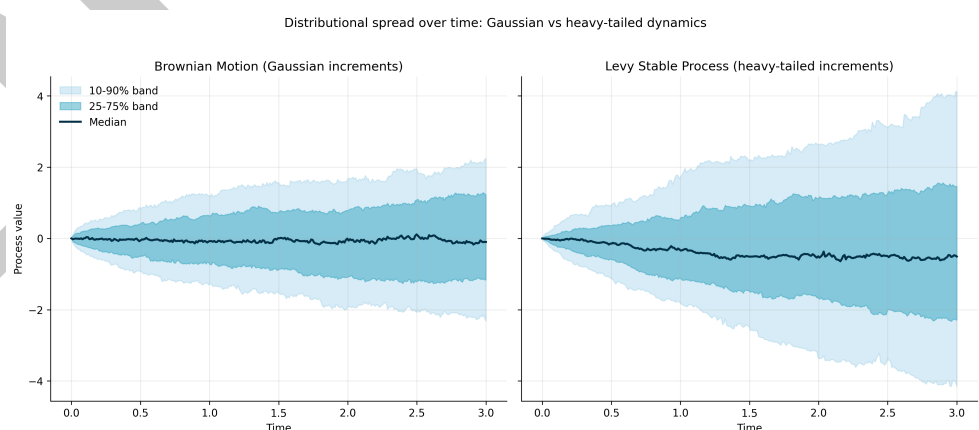
The software is organized into three primary modules:

- **Processes:** Ito and non-Ito process classes, including heavy-tailed, multiplicative, and memory-dependent variants.
- **Tools:** symbolic/numerical helpers for fitting, diagnostics, preasymptotic analysis, automation, and selected partial stochastic differential equation experiments.
- **Agents:** utility-based decision logic, portfolio/pool experiments, and optimization-oriented workflows.

A central design choice is a common process abstraction that allows users to swap process families without rewriting surrounding diagnostics and experiment code. This improves comparability across assumptions and lowers the cost of model stress testing.

Figure-based examples below illustrate the workflow span.

### Heavy-tailed spread geometry



**Figure 1:** Quantile fan charts for Brownian and Lévy-stable ensembles. The heavy-tailed model shows broader and less regular spread dynamics over the same horizon.

Figure 1 shows why Gaussian assumptions can understate dispersion in heavy-tailed settings: quantile bands widen asymmetrically and nonlinearly.

## 67 Multiplicative heavy-tailed growth diagnostics

Growth Rate vs Time and Number of Instances for GeometricLevyProcess with parameters {'alpha': 1.5, 'beta': 0, 'loc': 0.01, 'scale': 0.1}

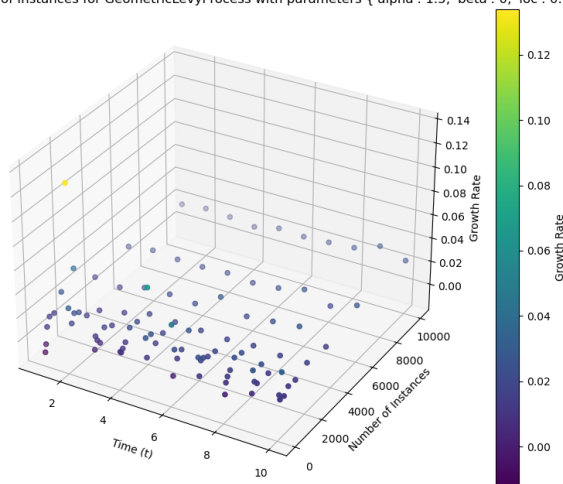


Figure 2: Geometric Lévy growth-rate map as a function of time horizon and number of instances.

68 Figure 2 demonstrates finite-sample variability of estimated growth rates under multiplicative  
69 heavy-tailed dynamics. This is directly relevant when practitioners compare expected outcomes  
70 to path-typical behavior.

## 71 Memory-dependent process dynamics

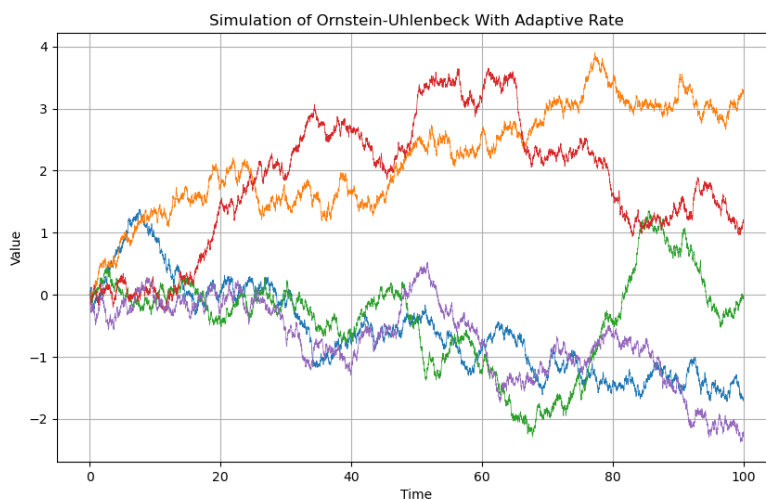
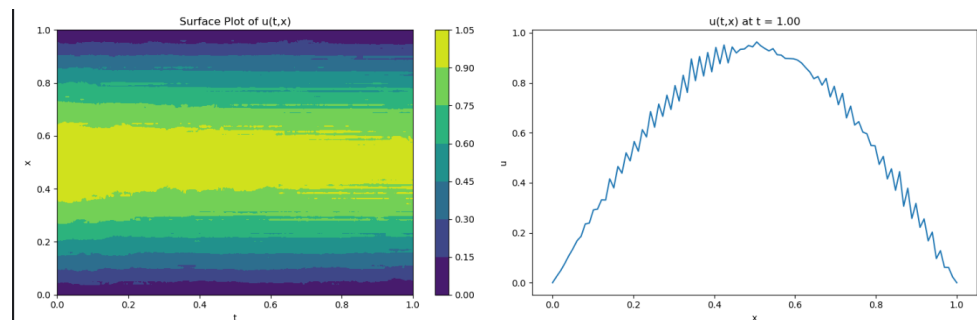


Figure 3: Adaptive-rate Ornstein–Uhlenbeck trajectories (book-derived example output).

72 Figure 3 illustrates how adaptive-rate memory effects produce heterogeneous mean-reverting  
73 trajectories across realizations.

## 74 Stochastic field simulation support



**Figure 4:** SPDE visualization with space-time contour (left) and final-time slice (right), from documented examples.

75 **Figure 4** shows that the package is not limited to scalar path simulation; it also supports  
76 stochastic field-style diagnostics in space-time settings.

77 The project is accompanied by a long-form book with worked examples and code context  
78 ([Kendiukhov, 2024](#)). At the same time, maturity constraints are explicit: selected API surfaces  
79 remain incomplete, automated pytest-style coverage is currently sparse relative to breadth, and  
80 optional ML-heavy agent paths can be environment-sensitive.

## 81 Research impact statement

82 The current impact is primarily infrastructural and methodological. `ergodicity_library`  
83 reduces setup time for experiments that combine stochastic simulation, diagnostics, and  
84 decision logic in one workflow ([Kendiukhov, 2025](#)). The package already enables reproducible  
85 demonstrations for heavy-tailed spread behavior, multiplicative growth diagnostics, adaptive-  
86 memory dynamics, and stochastic field visualization ([Kendiukhov, 2024](#)).

87 At this stage, the strongest evidence is implemented breadth plus reproducible artifacts rather  
88 than large-scale adoption metrics. The software is publicly available under an OSI-approved  
89 license and can serve as a base for domain-specific extensions.

90 A realistic near-term impact path is improved reproducibility and faster iteration in cross-method  
91 stochastic studies, where users can move from hypothesis definition to comparative diagnostics  
92 without rebuilding experiment scaffolding repeatedly.

## 93 AI usage disclosure

94 Generative AI assistance was used in preparing JOSS submission materials (manuscript  
95 drafting/editing, submission packaging, and figure-selection workflow scripting). The tools  
96 used were GPT-5-class coding assistants in an interactive development workflow.

97 All AI-assisted outputs were reviewed, edited, and validated by the human author. The author  
98 made the core scientific and software decisions and remains responsible for accuracy, originality,  
99 licensing compliance, and final submitted content.

## 100 Acknowledgements

101 No external funding is declared for this submission. The author acknowledges the open scientific  
102 Python ecosystem that this project builds upon ([Harris et al., 2020](#); [Hunter, 2007](#); [Meurer et al., 2017](#); [Virtanen et al., 2020](#)).

## References

- Harris, C. R., Millman, K. J., Walt, S. J. van der, & others. (2020). Array programming with NumPy. *Nature*, 585, 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- Kendiukhov, I. (2024). *Ergodicity library: An introduction*. Self-published. ISBN: 978-3-00-080360-4
- Kendiukhov, I. (2025). *Ergodicity\_library*. GitHub repository. [https://github.com/Kendiukhov/ergodicity\\_library/](https://github.com/Kendiukhov/ergodicity_library/)
- Meurer, A., Smith, C. P., Paprocki, M., & others. (2017). SymPy: Symbolic computing in python. *PeerJ Computer Science*, 3, e103. <https://doi.org/10.7717/peerj-cs.103>
- Peters, O. (2019). The ergodicity problem in economics. *Nature Physics*, 15, 1216–1221. <https://doi.org/10.1038/s41567-019-0732-0>
- Peters, O., & Gell-Mann, M. (2016). Evaluating gambles using dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(2), 023103. <https://doi.org/10.1063/1.4940236>
- sdeint developers. (2026). *Sdeint: Numerical integration of stochastic differential equations (SDE)*. PyPI project page. <https://pypi.org/project/sdeint/>
- sdepy developers. (2026). *Sdepy: Numerical integration of itô stochastic differential equations*. PyPI project page. <https://pypi.org/project/sdepy/>
- stochastic developers. (2026). *Stochastic: Generate realizations of stochastic processes*. PyPI project page. <https://pypi.org/project/stochastic/>
- Virtanen, P., Gommers, R., Oliphant, T. E., & others. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>