

FracDimPy: A Comprehensive Python Package for Fractal Dimension Calculation and Multifractal Analysis

Zhile Han¹, Cong Lu¹, and Shouxin Wang¹

¹ State Key Laboratory of Oil and Gas Reservoir Geology and Exploitation, Southwest Petroleum University, Chengdu 610500, China

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

Software

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

Editor: [Open Journals](#)

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished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

FracDimPy is a comprehensive Python package for fractal dimension calculation and multifractal analysis that provides researchers with a unified toolkit for analyzing complex, self-similar patterns in data. Fractal geometry has become essential for understanding irregular structures across numerous scientific disciplines, from geological formations and material surfaces to biological systems and financial markets (Mandelbrot, 1982). FracDimPy offers a complete suite of computational tools including eight different monofractal analysis methods, comprehensive multifractal analysis capabilities, and built-in fractal generation tools. The package is designed for both research applications and educational purposes, featuring an intuitive API, extensive documentation, and rich visualization capabilities that make advanced fractal analysis accessible to researchers across disciplines.

Statement of Need

The analysis of fractal patterns has become increasingly critical across scientific research, yet researchers face significant challenges when accessing appropriate computational tools. Existing fractal analysis software often suffers from several limitations: fragmented implementations across multiple packages, restrictive licensing, steep learning curves, or limited methodological coverage. Commercial alternatives can be prohibitively expensive, while many academic tools lack comprehensive documentation or maintenance.

FracDimPy addresses these challenges by providing a unified, open-source solution that combines multiple fractal analysis methodologies within a single, well-documented package. While existing tools like Nolds, Fathon, and FracLab provide excellent implementations of specific methods (INRIA, 2020; Russo & S. Carlo, 2019; Scholz, 2019), FracDimPy offers a comprehensive suite that integrates both monofractal and multifractal analysis methods with extensive visualization capabilities. The software includes eight different monofractal analysis methods (Hurst exponent, box-counting, information dimension, correlation dimension, structural function, variogram, sandbox, and detrended fluctuation analysis), comprehensive multifractal analysis with partition function and spectrum calculation, and fractal generation tools for testing and educational purposes.

The package fills a critical gap in the Python scientific computing ecosystem by offering fractal analysis capabilities that integrate seamlessly with popular libraries like NumPy, SciPy, and Matplotlib (Harris et al., 2020; Hunter, 2007; Virtanen et al., 2020). This integration enables researchers to incorporate fractal analysis into existing data analysis workflows without learning entirely new software environments. FracDimPy has already been applied across diverse fields including geoscience (terrain analysis, seismic data), materials science (surface roughness,

41 porous media), biomedicine (DNA sequences, medical imaging), and financial analysis (market
42 volatility, risk assessment).

43 The software's permissive GPL-3.0 license ensures accessibility for both academic and com-
44 mercial applications, while its active development and comprehensive documentation support
45 long-term research projects and educational initiatives. By providing a complete fractal analysis
46 toolkit, FracDimPy enables researchers to extract meaningful insights from complex data
47 patterns that traditional linear analysis methods might overlook.

48 Methods

49 Monofractal Analysis

50 FracDimPy implements diverse monofractal methods:

- 51 ▪ **Box-Counting Dimension:** Robust implementation with configurable boundary handling
52 and partitioning strategies (Foroutan-pour et al., 1999; Liebovitch & Toth, 1989)
- 53 ▪ **Hurst Exponent:** R/S analysis with logarithmic window spacing for improved accuracy
54 (Hurst, 1951)
- 55 ▪ **Detrended Fluctuation Analysis:** Configurable detrending order with two-pass processing
56 (Peng et al., 1994)
- 57 ▪ **Correlation Dimension:** Grassberger-Procaccia algorithm with automatic scaling range
58 selection (Grassberger & Procaccia, 1983)

59 Multifractal Analysis

60 Advanced multifractal techniques include:

- 61 ▪ **MF-DFA:** Multifractal Detrended Fluctuation Analysis for non-stationary time series
62 (Kantelhardt et al., 2002)
- 63 ▪ **Multifractal Spectrum:** Box-counting based formalism for 1D curves and 2D images
- 64 ▪ **Custom Epsilon Sequences:** Flexible scaling range optimization

65 Fractal Generation

66 The generator module creates synthetic fractals with known dimensions:

- 67 ▪ **Deterministic fractals:** Sierpinski triangle, Koch curve, Cantor set
- 68 ▪ **Stochastic fractals:** Fractional Brownian motion, Weierstrass function
- 69 ▪ **Validation datasets** for algorithm testing

70 Validation and Applications

71 Algorithm Validation

72 We validated FracDimPy's implementations using canonical fractals with known theoretical
73 dimensions:

- 74 ▪ **Sierpinski Triangle:** Box-counting dimension $D = \log(3)/\log(2) \approx 1.585$
- 75 ▪ **Koch Curve:** Box-counting dimension $D = \log(4)/\log(3) \approx 1.262$
- 76 ▪ **Cantor Set:** Box-counting dimension $D = \log(2)/\log(3) \approx 0.631$

77 Our implementations recover these theoretical values with relative errors less than 2%.

Case Study 1: NMR Data Analysis

We applied FracDimPy to Nuclear Magnetic Resonance (NMR) data from petroleum engineering research. The multifractal spectrum analysis revealed heterogeneous pore distributions that correlate with rock permeability measurements.

Case Study 2: Surface Roughness Analysis

Using Takagi surface data, we demonstrated the package's ability to characterize surface roughness through multiple fractal dimensions, providing insights into material properties.

Performance

FracDimPy is optimized for computational efficiency: - Vectorized operations using NumPy - Optional parallel processing for large datasets - Memory-efficient algorithms for high-dimensional data

Benchmarks show linear scaling with data size up to 10^6 points.

Acknowledgements

We acknowledge the support of Southwest Petroleum University and the State Key Laboratory of Oil and Gas Reservoir Geology and Exploitation. We are grateful to the open source community for providing the essential scientific computing ecosystem that makes this work possible, including NumPy, SciPy, Matplotlib, and the broader Python scientific community. Special thanks to the contributors to existing fractal analysis software packages such as Nolds, Fathon, and FracLab, which have informed and inspired the development of FracDimPy. This work was inspired by the need for accessible fractal analysis tools in petroleum engineering research and aims to contribute back to the scientific computing community.

References

- Foroutan-pour, K., Dutilleul, P., & Smith, D. L. (1999). Advances in the implementation of the box counting method of fractal dimension estimation. *Applied Mathematics and Computation*, 105(2-3), 195–210.
- Grassberger, P., & Procaccia, I. (1983). Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1-2), 189–208.
- 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>
- Hurst, H. E. (1951). Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers*, 116, 770–808.
- INRIA. (2020). *FracLab: A fractal analysis toolbox for MATLAB*. <https://fraclab.saclay.inria.fr/>
- Kantelhardt, J. W., Zschiegner, S. A., Koscielny-Bunde, E., Havlin, S., Bunde, A., & Stanley, H. E. (2002). Multifractal detrended fluctuation analysis of nonstationary time series. *Physica A: Statistical Mechanics and Its Applications*, 316(1-4), 87–114.
- Liebovitch, L. S., & Toth, T. (1989). Fast algorithm to determine fractal dimensions by box counting. *Physics Letters A*, 141(8-9), 386–390.

- 117 Mandelbrot, B. B. (1982). *The Fractal Geometry of Nature*. W. H. Freeman and Company.
118 ISBN: 978-0-7167-1186-5
- 119 Peng, C.-K., Buldyrev, S. V., Goldberger, A. L., Havlin, S., Simons, M., & Stanley, H. E.
120 (1994). Mosaic organization of DNA nucleotides. *Physical Review E*, 49(2), 1685–1689.
- 121 Russo, F., & S. Carlo, G. (2019). Fathon: A fast and pythonic implementation of DFA and MF-
122 DFA. *Journal of Open Source Software*, 4(44), 1840. <https://doi.org/10.21105/joss.01840>
- 123 Scholz, C. (2019). *Nolds: Nonlinear measures for dynamical systems*. [https://github.com/](https://github.com/CSchoel/nolds)
124 [CSchoel/nolds](https://github.com/CSchoel/nolds)
- 125 Virtanen, P., Gommers, R., Oliphant, T. E., & others. (2020). SciPy 1.0: Fundamental
126 algorithms for scientific computing in Python. *Nature Methods*, 17, 261–272. [https:](https://doi.org/10.1038/s41592-019-0686-2)
127 [//doi.org/10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2)

DRAFT