

# <sup>1</sup> FracDimPy: A Comprehensive Python Package for Fractal Dimension Calculation and Multifractal Analysis

<sup>4</sup> **Zhile Han**  <sup>1</sup>, **Cong Lu** <sup>1</sup>, and **Shouxin Wang**  <sup>1</sup>

<sup>5</sup> <sup>1</sup> 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Published: unpublished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 <sup>18</sup> Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))<sub>19</sub>

## <sup>7</sup> Summary

<sup>8</sup> 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.

## <sup>18</sup> Statement of Need

<sup>20</sup> The analysis of fractal patterns has become increasingly critical across scientific research, yet researchers face significant challenges when accessing appropriate computational tools. Existing <sup>21</sup> fractal analysis software often suffers from several limitations: fragmented implementations <sup>22</sup> across multiple packages, restrictive licensing, steep learning curves, or limited methodological <sup>23</sup> coverage. Commercial alternatives can be prohibitively expensive, while many academic tools <sup>24</sup> lack comprehensive documentation or maintenance.

<sup>25</sup> FracDimPy addresses these challenges by providing a unified, open-source solution that <sup>26</sup> combines multiple fractal analysis methodologies within a single, well-documented package. <sup>27</sup> While existing tools like Nolds, Fathon, and FracLab provide excellent implementations of <sup>28</sup> specific methods (INRIA, 2020; Russo & S. Carlo, 2019; Scholz, 2019), FracDimPy offers <sup>29</sup> a comprehensive suite that integrates both monofractal and multifractal analysis methods <sup>30</sup> with extensive visualization capabilities. The software includes eight different monofractal <sup>31</sup> analysis methods (Hurst exponent, box-counting, information dimension, correlation dimension, <sup>32</sup> structural function, variogram, sandbox, and detrended fluctuation analysis), comprehensive <sup>33</sup> multifractal analysis with partition function and spectrum calculation, and fractal generation <sup>34</sup> tools for testing and educational purposes.

<sup>35</sup> The package fills a critical gap in the Python scientific computing ecosystem by offering fractal <sup>36</sup> analysis capabilities that integrate seamlessly with popular libraries like NumPy, SciPy, and <sup>37</sup> Matplotlib (Harris et al., 2020; Hunter, 2007; Virtanen et al., 2020). This integration enables <sup>38</sup> researchers to incorporate fractal analysis into existing data analysis workflows without learning <sup>39</sup> entirely new software environments. FracDimPy has already been applied across diverse fields <sup>40</sup> including geoscience (terrain analysis, seismic data), materials science (surface roughness,

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

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

## <sup>48</sup> Methods

### <sup>49</sup> Monofractal Analysis

<sup>50</sup> FracDimPy implements diverse monofractal methods:

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

### <sup>59</sup> Multifractal Analysis

<sup>60</sup> Advanced multifractal techniques include:

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

### <sup>65</sup> Fractal Generation

<sup>66</sup> The generator module creates synthetic fractals with known dimensions:

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

## <sup>70</sup> Validation and Applications

### <sup>71</sup> Algorithm Validation

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

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

<sup>77</sup> Our implementations recover these theoretical values with relative errors less than 2%.

## 78 Case Study 1: NMR Data Analysis

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

## 82 Case Study 2: Surface Roughness Analysis

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

## 85 Performance

86 FracDimPy is optimized for computational efficiency:  
87 - Vectorized operations using NumPy -  
88 Optional parallel processing for large datasets - Memory-efficient algorithms for high-dimensional  
89 data  
Benchmarks show linear scaling with data size up to  $10^6$  points.

## 90 Acknowledgements

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

## 99 References

- 100 Foroutan-pour, K., Dutilleul, P., & Smith, D. L. (1999). Advances in the implementation  
101 of the box counting method of fractal dimension estimation. *Applied Mathematics and  
102 Computation*, 105(2-3), 195–210.
- 103 Grassberger, P., & Procaccia, I. (1983). Measuring the strangeness of strange attractors.  
104 *Physica D: Nonlinear Phenomena*, 9(1-2), 189–208.
- 105 Harris, C. R., Millman, K. J., Walt, S. J. van der, & others. (2020). Array programming with  
106 NumPy. *Nature*, 585, 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- 107 Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science &  
108 Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- 109 Hurst, H. E. (1951). Long-term storage capacity of reservoirs. *Transactions of the American  
110 Society of Civil Engineers*, 116, 770–808.
- 111 INRIA. (2020). *FracLab: A fractal analysis toolbox for MATLAB*. <https://fraclab.saclay.inria.fr/>
- 112 Kantelhardt, J. W., Zschiegner, S. A., Koscielny-Bunde, E., Havlin, S., Bunde, A., & Stanley,  
113 H. E. (2002). Multifractal detrended fluctuation analysis of nonstationary time series.  
114 *Physica A: Statistical Mechanics and Its Applications*, 316(1-4), 87–114.
- 115 Liebovitch, L. S., & Toth, T. (1989). Fast algorithm to determine fractal dimensions by box  
116 counting. *Physics Letters A*, 141(8-9), 386–390.

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

DRAFT