The Journal of Open Source Software

PyTurbo_SF: An Adaptive Bootstrap Framework for Efficient Structure Function Analysis in Turbulent Flows

Adam Ayouche [1¶], **Baylor Fox-Kemper** [1], **Momme Hell** [2], **Brodie Pearson** [3], **and Cassidy Wagner** [3]

1 Department of Earth, Environmental, and Planetary Sciences (DEEPS), Brown University, Providence, RI, USA **2** Woods Hole Oceanographic Institution, Woods Hole, MA, USA **3** Department of Earth, Environmental, and Planetary Sciences (DEEPS), Brown University, Providence, RI, USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Gabriele Bozzola

Reviewers:

- @NoraLoose
- @einaraz

Submitted: 01 October 2025
Published: unpublished

Summary

Structure functions are fundamental diagnostic tools in turbulence research that quantify spatial correlations between field differences at varying separation distances, revealing energy cascade characteristics and scaling laws (Frisch, 1995; Pope, 2000). PyTurbo_SF is a comprehensive Python package providing efficient, statistically rigorous structure function calculations for 1D, 2D, and 3D turbulent datasets through an innovative adaptive bootstrap framework.

The package addresses computational bottlenecks through power-of-2 spacing strategies, adaptive convergence monitoring, and memory-optimized algorithms. PyTurbo_SF delivers robust uncertainty quantification through bootstrap resampling and supports diverse (conditional) structure function types including longitudinal, transverse, scalar, advective, and energy flux functions (Pearson et al., 2021, 2024).

Applications span oceanographic time series and satellite measurements to high-resolution simulations, enabling consistent methodology across scales from laboratory to planetary systems.

Statement of need

Contemporary turbulence research relies on massive datasets from satellite missions, autonomous platforms, and high-resolution simulations. Traditional structure function calculations face severe limitations: computational intractability for large datasets, absence of uncertainty quantification, manual parameter tuning, and limited function types.

Existing tools address only subsets of these challenges. fastSF provides parallelized implementations but lacks advanced function types and uncertainty quantification (Sadhukhan et al., 2021). MATLAB toolkits are environmentally limited and lack comprehensive statistical frameworks (Fuchs et al., 2022). Alternative approaches like FlowSieve cannot provide the scale-by-scale information structure functions uniquely deliver (Storer & Aluie, 2023).

There is a growing need for tools to analyze emerging datasets from NASA's SWOT satellite and next-generation atmospheric simulations generating terabyte-scale outputs. Recent advances in structure function theory, particularly advective structure functions (Pearson et al., 2021) and spectral flux estimation (Pearson et al., 2024), require frameworks handling both traditional and novel function types with statistical rigor.

PyTurbo_SF fills this gap by providing the first comprehensive, statistically robust framework that scales from small observational datasets to massive simulation outputs while delivering quantified uncertainties essential for scientific interpretation.

Software functionality

PyTurbo_SF implements the complete mathematical framework for structure function analysis, supporting functions of the form $S_n(r) = \langle|\phi(\mathbf{x}+\mathbf{r}) - \phi(\mathbf{x})|^n\rangle_{\mathbf{x}}$ where ϕ represents arbitrary field variables (velocity, scalars, derived quantities), $\mathbf{r}$ is the separation vector, n is the order, and $\langle\cdot\rangle_{\mathbf{x}}$ denotes spatial averaging. The package supports traditional functions (longitudinal, transverse, scalar), cross functions, and advective structure functions enabling direct energy flux quantification (Pearson et al., 2021).

The core algorithmic breakthrough is adaptive bootstrap methodology increasing computational efficiency and statistical reliability. The algorithm employs power-of-2 spacings optimizing memory access patterns while providing optimal scale separation. Adaptive convergence monitoring dynamically allocates computational resources, eliminating manual parameter tuning while guaranteeing robust uncertainty estimates.

Performance optimization enables analysis of previously intractable datasets. Memory-efficient structures maintain peak usage at 2-5× base dataset size. Parallel processing provides near-linear scaling with available cores. Benchmark testing demonstrates O(NM log N log M) complexity for 2D data, enabling analysis of datasets with millions of grid points.

The package provides three main interfaces: bin_sf_1d() for time series, bin_sf_2d() for surface fields, and bin_sf_3d() for volumetric data. All functions automatically optimize computational strategies while supporting both isotropic and directional analysis.

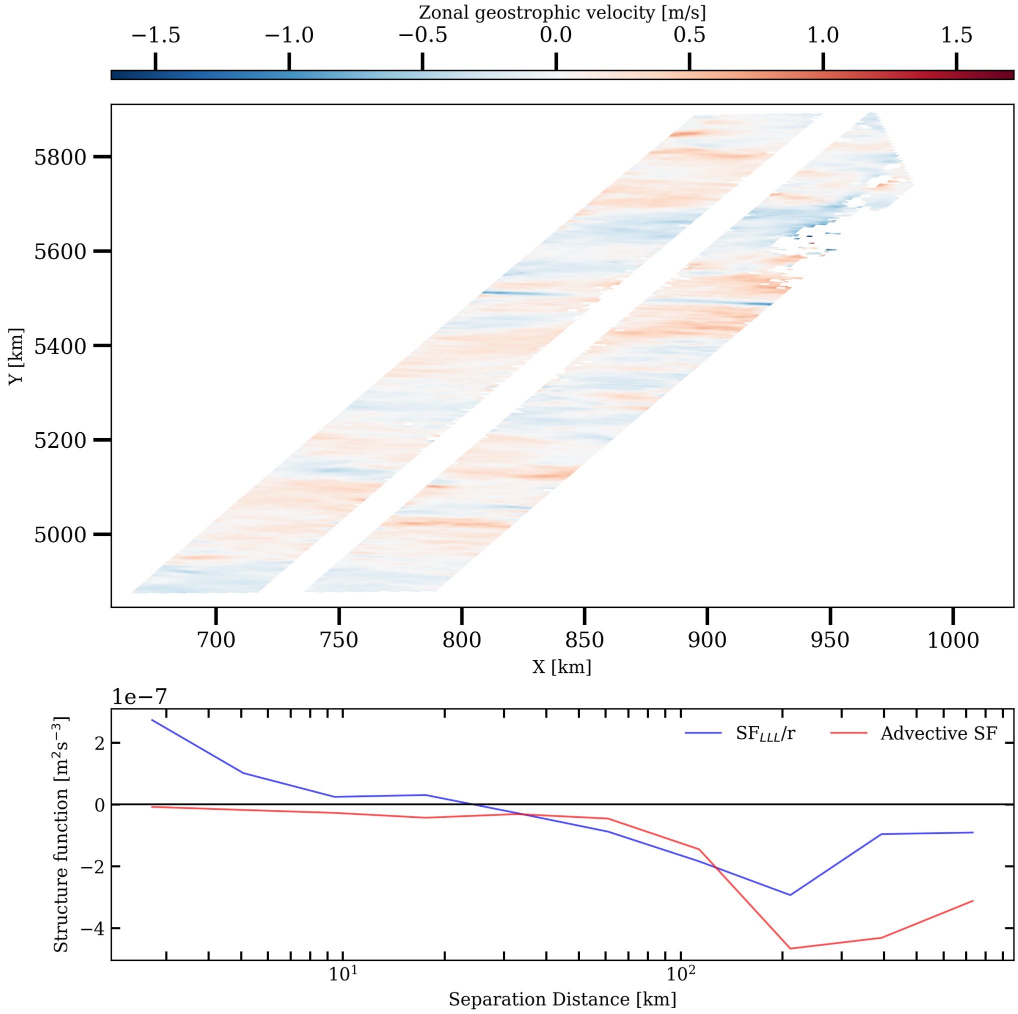

Figure 1: Satellite altimetry analysis showing second-order longitudinal structure functions calculated from SWOT data in the Gulf Stream region. Bootstrap error bars demonstrate statistical rigor while revealing energy cascade signatures in ocean surface turbulence. Results show the characteristic $r^{2/3}$ scaling predicted by geostrophic turbulence theory.

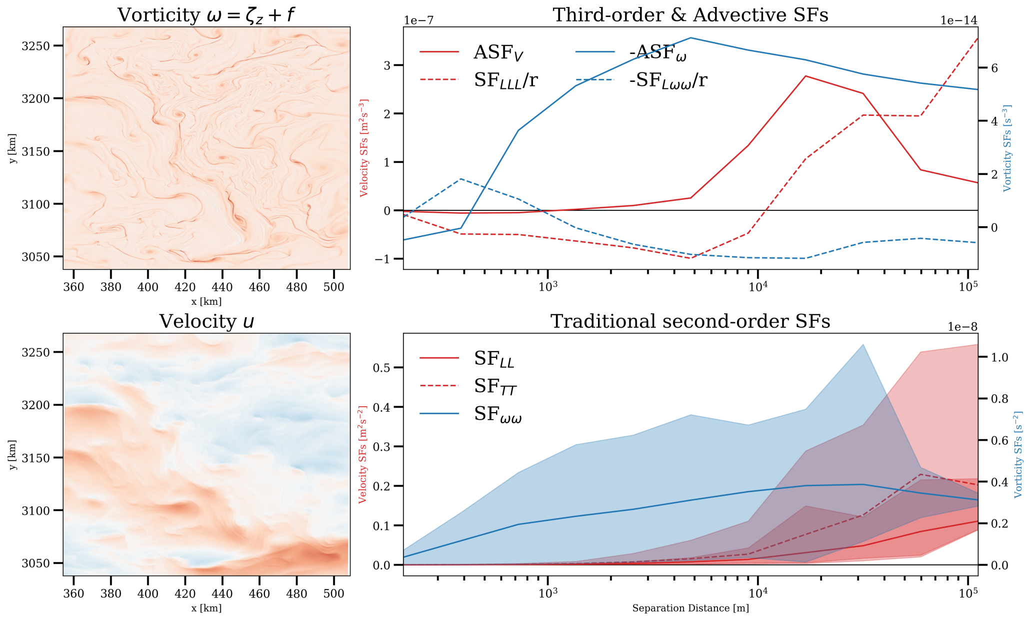

Figure 2: Regional ocean model analysis using CROCO simulation data showing energy transfer mechanisms through combined velocity-scalar structure functions. The adaptive bootstrap algorithm efficiently handles the large 2D spatial dataset while providing robust uncertainty quantification for energy flux estimates.

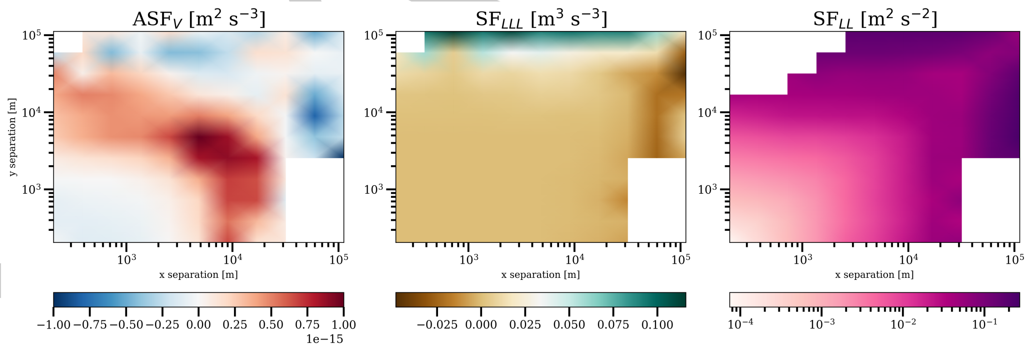

Figure 3: Regional ocean model analysis using CROCO simulation data showing energy transfer mechanisms through combined 2D velocity-scalar structure functions.

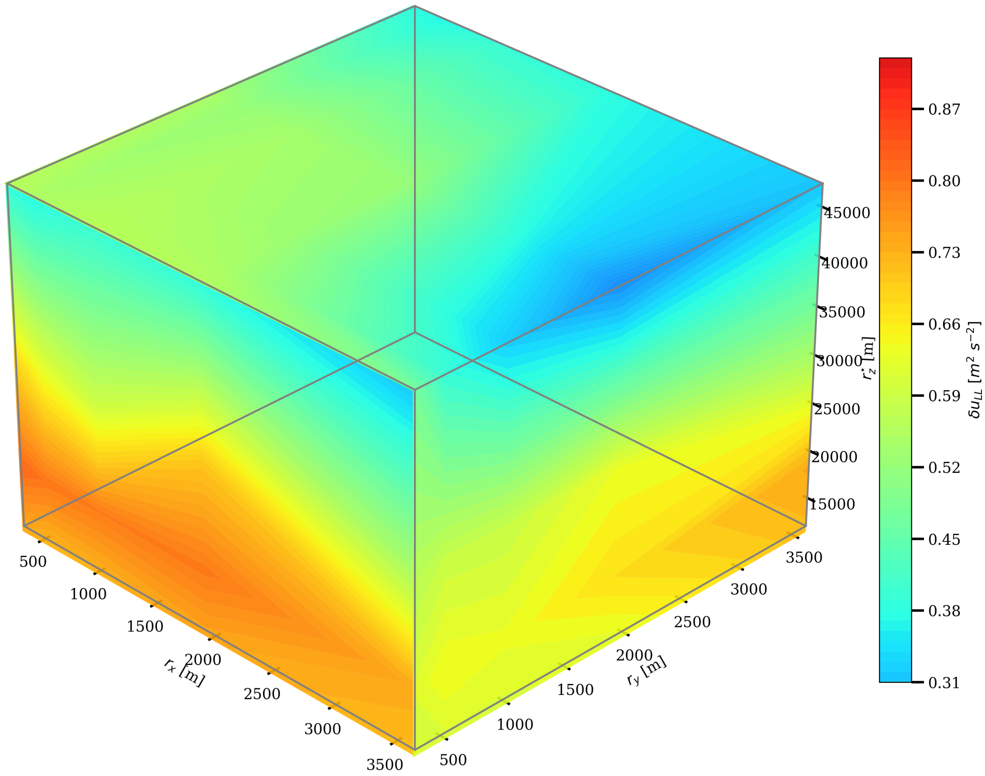

Figure 4: Atmospheric boundary layer turbulence analysis from DYCOMS large eddy simulation demonstrating 3D structure function capabilities. Volume-weighted binning accurately captures isotropic averaging in the 3D velocity field while bootstrap resampling provides statistical reliability for intermittency characterization.

Related work and scientific impact

PyTurbo_SF represents a significant advancement by uniquely combining comprehensive function types, adaptive bootstrap methodology, and optimized algorithms. While fastSF provides basic parallelized calculations (Sadhukhan et al., 2021) and MATLAB toolkits offer specific analyses (Fuchs et al., 2022), no existing software delivers the combination of statistical rigor, efficiency, and breadth required for contemporary turbulence research.

FluidSF (Wagner et al., 2025) is a related Python package for structure function calculations. While both packages support 1D/2D/3D data and core SF types (longitudinal, transverse, scalar, advective), they differ substantially in scope and methodology. Table 1 summarizes the key differences.

Table 1: Comparison of PyTurbo_SF and FluidSF.

Feature	PyTurbo_SF	FluidSF
Structure function order	Arbitrary	2nd and 3rd only
Uncertainty quantification	Adaptive bootstrap with convergence	None
Parallel processing	Yes (joblib)	No

Feature	PyTurbo_SF	FluidSF
Conditional structure functions	Yes	No
Cross-term SF types	Extensive (longitudinal-transverse, longitudinal-scalar, transverse-scalar, scalar-scalar)	Limited (velocity-scalar blend)
3D transverse decomposition	Full (ij, ik, jk planes)	Single
Spectral flux via Bessel transform	Yes	No
Isotropic averaging	Exact spherical/polar binning	Simplified
Output format	xarray Dataset with metadata	NumPy arrays

PyTurbo_SF's primary contributions are: (1) rigorous uncertainty quantification through adaptive bootstrap resampling with automatic convergence monitoring, (2) support for arbitrary-order structure functions essential for intermittency analysis, and (3) computational efficiency through parallelization and power-of-2 spacing strategies enabling analysis of large datasets.

The package enables application of recent theoretical developments, particularly advective structure functions providing direct energy flux measurements (Pearson et al., 2021) and spectral flux estimation methodologies (Pearson et al., 2024). These reveal energy pathways traditional approaches cannot capture, offering insights into cascade mechanisms in ocean and atmospheric turbulence.

Scientific applications demonstrate transformative impact across domains. PyTurbo_SF enables analysis of satellite altimetry data for characterizing surface turbulence and large eddy simulation data for understanding boundary layer dynamics. The consistent methodology enables comparative studies previously impossible due to software limitations.

The adaptive bootstrap framework addresses a fundamental challenge: quantifying uncertainties in structure function estimates. PyTurbo_SF's principled uncertainty quantification enables robust statistical comparisons and hypothesis testing, elevating scientific standards.

Acknowledgements

This software package is based upon work supported by the US Department of Energy grant DE-SC0024572.

Any opinions, findings, and conclusions or recommendations expressed in this package are those of the authors and do not necessarily reflect the views of the US Department of Energy.

References

Frisch, U. (1995). *Turbulence: The legacy of A.N. Kolmogorov*. Cambridge University Press. https://doi.org/10.1017/CBO9781139170666

Fuchs, A., Kharche, S., Patil, A., Friedrich, J., Wächter, M., & Peinke, J. (2022). An open source package to perform basic and advanced statistical analysis of turbulence data and other complex systems. *Physics of Fluids*, *34*(10), 101801. https://doi.org/10.1063/5.0107974

Pearson, B., Pearson, J., & Fox-Kemper, B. (2021). Advective structure functions in anisotropic two-dimensional turbulence. *Journal of Fluid Mechanics*, *916*, A49. https://doi.org/10.1017/jfm.2021.247

Pearson, B., Wagner, C., Fox-Kemper, B., & Samelson, R. (2024). Estimating spectral fluxes in quasi-two-dimensional flows with advective structure functions and Bessel functions. *Journal of Fluid Mechanics*.

Pope, S. B. (2000). *Turbulent flows*. Cambridge University Press. https://doi.org/10.1017/CBO9780511840531

Sadhukhan, S., Bhattacharya, S., & Verma, M. K. (2021). fastSF: A parallel code for computing the structure functions of turbulence. *Journal of Open Source Software*, *6*(57), 2185. https://doi.org/10.21105/joss.02185

Storer, B. A., & Aluie, H. (2023). FlowSieve: A coarse-graining utility for geophysical flows on the sphere. *Journal of Open Source Software*, *8*(84), 4277. https://doi.org/10.21105/joss.04277

Wagner, C. M., Lee, A., & Pearson, B. (2025). FluidSF: A Python package for calculating turbulent flow statistics. *Journal of Open Source Software*, *10*(114), 7873. https://doi.org/10.21105/joss.07873