

¹ SolarWindPy: A Heliophysics Data Analysis Tool Set

² B. L. Alterman 

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Published: unpublished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](#)).¹⁶

³ Summary

⁴ The region of space within the Sun's envelope of influence is called the heliosphere. The field
⁵ of heliophysics starts in the solar interior and extends out to the very local interstellar medium,
⁶ just beyond the heliosphere. The solar wind is a stream of charged particles that continuously
⁷ flows away from the Sun, carrying mass, energy, and momentum along with an embedded
⁸ magnetic field. In short, it mediates the interaction of the Sun with the heliosphere and this is
⁹ a feature shared by stars and their astrospheres more broadly. Changes in the solar wind are
¹⁰ one source of space weather, which is a critical threat to our technological infrastructure on
¹¹ Earth and in space. SolarWindPy provides a unified framework for analyzing the solar wind
¹² and related space weather data, filling the gap between packages targeting astronomy, remote
¹³ observations of the Sun, and general timeseries analysis of spacecraft based data. The package
¹⁴ is available via PyPI¹ and conda-forge² and can be installed using pip install solarwindpy
¹⁵ or conda install -c conda-forge solarwindpy.¹⁶

Statement of Need

¹⁷ There is a growing ecosystem of python libraries to enable astrophysics, solar physics, plasma
¹⁸ physics, and space physics. The table below cites key examples. Notably, there are several
¹⁹ packages that support different elements of space physics, including magnetospheric data
²⁰ analysis (Pysat), integration of magnetospheric observations (SpacePy), and the retrieval
²¹ and analysis of heliophysics timeseries data (pySpedas and PyTplot). Tools for the dedicated
²² analysis of solar wind observations are noticeably absent. SolarWindPy fills this gap by
²³ providing a unified framework for analyzing solar wind observations in combination with
²⁴ relevant information about the spacecraft from which the observations were made.

Library	Purpose	Citation
AstroPy	Astronomical observations.	Astropy Collaboration et al. (2022)
SunPy	Remote sensing observations of the Sun.	Barnes et al. (2020)
PlasmaPy	Theoretical plasma physics.	(PlasmaPy Community, 2025)
SpacePy	Timeseries analysis and magnetospheric modeling.	Morley et al. (n.d.)
Pysat	Magnetospheric mission data analysis.	Stoneback et al. (2023)
pySpedas	Retrieval and plotting of heliophysics timeseries.	(Grimes et al., 2022)
PyTplot	Timeseries and spectrograph data visualization.	(Harter & MAVENSDC Team, 2019)

¹<https://pypi.org/project/solarwindpy/>

²<https://anaconda.org/conda-forge/solarwindpy>

25 The SolarWindPy framework utilizes a pythonic, class-based architecture that combines ion
26 and magnetic field objects into a single, unified plasma. It is designed for both experienced
27 researchers and to provide an intuitive scaffold for students learning to analyze spacecraft
28 data. SolarWindPy's primary functionality (core, fitfunctions, plotting, instabilities, and
29 solar_activity submodules) was written by the author and developed or utilized in support
30 of multiple publications B. L. Altermann, Rivera, Lepri, & Raines (2025). The transformation
31 from thesis research code to a production package deployable via PyPI and conda-forge was
32 accomplished using AI-assisted development with specialized quality assurance infrastructure
33 for the supporting infrastructure (test suites, documentation, and deployment workflows), while
34 the core scientific functionality remains human-authored.

35 The package builds on well-established libraries including NumPy van der Walt et al. (2011),
36 SciPy (Virtanen et al., 2020), Matplotlib (Hunter, 2007), and Pandas Mckinney (2013) to
37 ensure that the dependencies are stable. The plotting functionality retains the mapping
38 between timeseries and aggregated observations to enable researchers to easily extract subsets
39 of their observations for detailed analysis. The plot labeling functionality maps the quantities
40 plotted to their file names, improving the mapping from the user's analysis to the saved
41 output. The non-linear fitting libraries (utilizing scipy optimize) are designed for multi-step
42 fitting in which the user performs nested regression of one variable on parameters derived
43 from fitting other quantities. Submodules for the analysis of magnetohydrodynamic turbulence
44 parameters and kinetic instabilities are also provided. The solar_activity submodule provides
45 the user with seamless access to solar activity indicators provided by the LASP Interactive
46 Solar IRradiance Datacenter (LISIRD) (Leise et al., 2019) and the Solar Information Data
47 Center (SIDC) at the Royal Observatory of Belgium (?). This tool enables easy comparison
48 of solar wind parameters across different phases of the solar cycle and different solar cycles,
49 which is an essential component of solar wind data analysis. SolarWindPy currently stores data
50 in pandas DataFrames and Timeseries objects. However, there is a clear separation between
51 the two libraries such that future development could transition to using more nuanced and
52 scientifically-targeted data structures, for example those provided by xarray (Hoyer & Hamman,
53 2017), SunPy, or AstroPy.

54 AI-Assisted Development Workflow

55 SolarWindPy's evolution from thesis research code (B. L. Altermann et al., 2018; Benjamin L.
56 Altermann, 2019; B. L. Altermann & Kasper, 2019) to a production software package required
57 comprehensive testing, documentation, and deployment infrastructure. To be explicit about
58 the scope of AI assistance: the core scientific modules (core/, fitfunctions/, plotting/,
59 instabilities/, solar_activity/) containing the physics algorithms and analysis methods
60 were developed by the author without AI assistance and represent the scholarly contribution of
61 this work, validated through eight peer-reviewed publications B. L. Altermann, Rivera, Lepri,
62 & Raines (2025). AI-assisted development was used exclusively for supporting infrastructure:
63 test suites, continuous integration pipelines, package deployment workflows, and completion of
64 docstring documentation.

65 This was accomplished using Claude Code (Anthropic, 2024) with custom AI development
66 infrastructure designed for scientific computing quality assurance.

67 The implementation includes specialized domain-specific agents and automated validation
68 workflows using pre-commit hooks for physics validation, test execution, and coverage monitoring.
69 This systematic approach enabled rapid development of test suites for modules outside
70 the original core implementation, completion of documentation including missing docstrings,
71 and creation of continuous integration and deployment pipelines for PyPI, conda-forge, and
72 ReadTheDocs. The current agent system contains 7 specialized agents with an extensible
73 architecture designed for integration with Claude Code's skills system. The infrastructure
74 incorporates git commit integration, GitHub Issues planning workflows, and comprehensive
75 audit trails to ensure traceability of all AI-generated modifications, establishing an infrastructure

76 for trustworthy AI-assisted scientific software.

77 The project targets 95% test coverage, with core physics and plasma functionality currently
78 achieving comprehensive coverage (95%), while tests for advanced features such as fitfunctions
79 and plotting capabilities remain in active development, bringing overall coverage to 78%. All
80 code generated or modified by AI in the supporting infrastructure (representing the test suites,
81 CI/CD pipelines, and packaging tooling) undergoes expert review to ensure correctness, while
82 the scientific algorithms themselves remain entirely human-authored as evidenced by their
83 multi-year publication history. The complete AI-assisted development infrastructure, including
84 agent specifications, validation hooks, and workflow automation, is publicly available in the
85 .claude/ directory of the repository.

86 References

87 (Astropy Collaboration et al., 2013) (Astropy Collaboration et al., 2018) (Astropy Collaboration
88 et al., 2022) (The SunPy Community et al., 2020) (Mumford & others, 2020) (Barnes et al.,
89 2020) (PlasmaPy Community, 2025) (Niehof et al., 2022) (Morley et al., n.d.) (Stoneback
90 et al., 2021) (Stoneback et al., 2018) (Stoneback et al., 2023) (Grimes et al., 2022) (Harter
91 & MAVENSDC Team, 2019) (B. L. Alerman et al., 2018) (B. L. Alerman, 2025) (B. L.
92 Alerman & D'Amicis, 2025b) (B. L. Alerman & D'Amicis, 2025a) (B. L. Alerman & Kasper,
93 2019) (Benjamin L. Alerman et al., 2021) (B. L. Alerman, Rivera, Lepri, Raines, & D'Amicis,
94 2025) (B. L. Alerman, Rivera, Lepri, & Raines, 2025) (Harris et al., 2020) (van der Walt et al.,
95 2011) (Virtanen et al., 2020) (Hunter, 2007) (McKinney, 2010) (McKinney, 2011) (McKinney,
96 2013) (Leise et al., 2019) (?) (Vanlommel et al., 2005) (Anthropic, 2024) (Hoyer & Hamman,
97 2017)

98 Acknowledgements

99 The author thanks L. Woodham and R. D'Amicis for discussions about Alfvénic turbulence
100 and calculating the Elsasser variables. In line with the transition to AI-augmented software
101 development, Claude code (Anthropic, 2024) was used in writing this paper.

102 Alerman, Benjamin L. (2019). *The significance of proton beams in the multiscale solar wind*
103 [PhD thesis]. University of Michigan.

104 Alerman, B. L. (2025). Characterizing the Impact of Alfvén Wave Forcing in Interplanetary
105 Space on the Distribution of Near-Earth Solar Wind Speeds. *The Astrophysical Journal*,
106 984(2), L64. <https://doi.org/10.3847/2041-8213/add0a6>

107 Alerman, B. L., & D'Amicis, R. (2025a). On the Regulation of the Solar Wind Helium
108 Abundance by the Hydrogen Compressibility. *Astrophysical Journal Letters* (Accepted).

109 Alerman, B. L., & D'Amicis, R. (2025b). Cross Helicity and the Helium Abundance as
110 an In Situ Metric of Solar Wind Acceleration. *The Astrophysical Journal*, 982(2), L40.
111 <https://doi.org/10.3847/2041-8213/adb48e>

112 Alerman, B. L., & Kasper, J. C. (2019). Helium Variation across Two Solar Cycles Reveals a
113 Speed-dependent Phase Lag. *The Astrophysical Journal*, 879(1), L6. <https://doi.org/10.3847/2041-8213/ab2391>

115 Alerman, Benjamin L., Kasper, J. C., Leamon, R. J., & McIntosh, S. W. (2021). Solar
116 Wind Helium Abundance Heralds Solar Cycle Onset. *Solar Physics*, 296(4), 67. <https://doi.org/10.1007/s11207-021-01801-9>

118 Alerman, B. L., Kasper, J. C., Stevens, M. L., & Koval, A. (2018). A Comparison of Alpha
119 Particle and Proton Beam Differential Flows in Collisionally Young Solar Wind. *The
120 Astrophysical Journal*, 864(2), 112. <https://doi.org/10.3847/1538-4357/aad23f>

- 121 Alterman, B. L., Rivera, Y. J., Lepri, S. T., & Raines, J. M. (2025). The transition from slow
122 to fast wind as observed in composition observations. *Astronomy and Astrophysics*, 694,
123 A265. <https://doi.org/10.1051/0004-6361/202451550>
- 124 Alterman, B. L., Rivera, Y. J., Lepri, S. T., Raines, J. M., & D'Amicis, R. (2025). The
125 Evolution of Heavy Ion Abundances with Solar Activity. *arXiv e-Prints*, arXiv:2504.18092.
126 <https://doi.org/10.48550/arXiv.2504.18092>
- 127 Anthropic. (2024). *Claude code: An agentic coding tool*. <https://github.com/anthropics/clause-code>
- 128
- 129 Astropy Collaboration, Price-Whelan, A. M., & others. (2018). The Astropy Project: Building
130 an Open-science Project and Status of the v2.0 Core Package.
131 *Aj*, 156(3), 123. <https://doi.org/10.3847/1538-3881/aabc4f>
- 132 Astropy Collaboration, Price-Whelan, A. M., & others. (2022). The Astropy Project: Sustaining
133 and Growing a Community-oriented Open-source Project and the Latest Major Release
134 (v5.0) of the Core Package.
135 *Apj*, 935(2), 167. <https://doi.org/10.3847/1538-4357/ac7c74>
- 136 Astropy Collaboration, Robitaille, T. P., & others. (2013). Astropy: A community Python
137 package for astronomy.
138 *Aap*, 558, A33. <https://doi.org/10.1051/0004-6361/201322068>
- 139 Barnes, W. T., Bobra, M. G., Christe, S. D., Freij, N., Hayes, L. A., Ireland, J., Mumford, S.,
140 Perez-Suarez, D., Ryan, D. F., Shih, A. Y., Chanda, P., Glogowski, K., Hewett, R., Hughtt, V. K.,
141 Hill, A., Hiware, K., Inglis, A., Kirk, M. S. F., Konge, S., ... Dang, T. K. (2020). The
142 SunPy Project: Open Source Development and Status of the Version 1.0 Core Package.
143 *The Astrophysical Journal*, 890(1), 68. <https://doi.org/10.3847/1538-4357/ab4f7a>
- 144 Grimes, E. W., Harter, B., Hatzigeorgiu, N., Drozdov, A., Lewis, J. W., Angelopoulos, V.,
145 Cao, X., Chu, X., Hori, T., Matsuda, S., Jun, C.-W., Nakamura, S., Kitahara, M.,
146 Segawa, T., Miyoshi, Y., & Le Contel, O. (2022). The space physics environment data
147 analysis system in python. *Frontiers in Astronomy and Space Sciences*, 9, 1020815.
148 <https://doi.org/10.3389/fspas.2022.1020815>
- 149 Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau,
150 D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van
151 Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ...
152 Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362.
153 <https://doi.org/10.1038/s41586-020-2649-2>
- 154 Harter, B., & MAVENSDC Team. (2019). PyTplot: A Python version of the IDL tplot libraries.
155 In *Github repository*. GitHub. <https://github.com/MAVENSDC/PyTplot>
- 156 Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled arrays and datasets in Python. *Journal
157 of Open Research Software*, 5(1). <https://doi.org/10.5334/jors.148>
- 158 Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science &
159 Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- 160 Leise, H., Baltzer, T., Wilson, A., Lindholm, D., Snow, M., Woodraska, D., Béland, S., Cod-
161 dington, O., & C., P. (2019). LASP Interactive Solar IRradiance Datacenter (LISIRD). *EGU
162 General Assembly Conference Abstracts*. <https://ui.adsabs.harvard.edu/abs/2019EGUGA.2112479L>
- 164 Mckinney, W. (2010). Data Structures for Statistical Computing in Python. In S. van der Walt
165 & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 51–56).
- 166 McKinney, W. (2011). Pandas: A Foundational Python Library for Data Analysis and Statistics.
167 *Python for High Performance and Scientific Computing*, 1–9.

- 168 Mckinney, W. (2013). *Python for Data Analysis*. O'Reilly. <https://doi.org/10.1145/1985441.1985476>
- 170 Morley, S. K., Niehof, J. T., Welling, D. T., Larsen, B. A., Brunet, A., Engel, M. A., Gieseler,
171 J., Haidupek, J., Henderson, M., Hendry, A., Hirsch, M., Killick, P., Koller, J., Merrill,
172 A., Rastatter, L., Reimer, A., Shih, A. Y., & Stricklan, A. (n.d.). *SpacePy*. Zenodo.
173 <https://doi.org/10.5281/zenodo.325253>
- 174 Mumford, S. J., & others. (2020). SunPy: A python package for solar physics. *Journal of
175 Open Source Software*, 5(46), 1832. <https://doi.org/10.21105/joss.01832>
- 176 Niehof, J. T., Morley, S. K., Welling, D. T., & Larsen, B. A. (2022). The SpacePy space
177 science package at 12 years. *Frontiers in Astronomy and Space Sciences*, 9. <https://doi.org/10.3389/fspas.2022.1023612>
- 179 PlasmaPy Community. (2025). *PlasmaPy* (Version 2025.8.0). Zenodo. <https://doi.org/10.5281/zenodo.1674774>
- 181 Stoneback, R. A., Burrell, A. G., Klenzing, J., & Depew, M. D. (2018). PYSPAT: Python
182 Satellite Data Analysis Toolkit. *Journal of Geophysical Research: Space Physics*, 123(6),
183 5271–5283. <https://doi.org/10.1029/2018JA025297>
- 184 Stoneback, R. A., Burrell, A. G., Klenzing, J., & Smith, J. (2023). The pysat ecosystem. *Frontiers
185 in Astronomy and Space Science*, 10. <https://doi.org/10.3389/fspas.2023.1119775>
- 186 Stoneback, R. A., Klenzing, J. H., Burrell, A. G., Spence, C., Depew, M., Hargrave, N., Smith,
187 J., Bose, V. von, Pembroke, A., Iyer, G., & Luis, S. (2021). *Python satellite data analysis
188 toolkit (pysat) vX.y.z*. <https://doi.org/10.5281/zenodo.1199703>
- 189 The SunPy Community, Barnes, W. T., & others. (2020). The SunPy project: Open source
190 development and status of the version 1.0 core package. *The Astrophysical Journal*, 890,
191 68–68. <https://doi.org/10.3847/1538-4357/ab4f7a>
- 192 van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure
193 for Efficient Numerical Computation. *Computing in Science & Engineering*, 13(2), 22–30.
194 <https://doi.org/10.1109/MCSE.2011.37>
- 195 Vanlommel, P., Cugnon, P., Van Der Linden, R. A. M., Berghmans, D., & Clette, F. (2005).
196 The sidc: World data center for the sunspot index. *Solar Physics*, 224(1-2), 113–120.
197 <https://doi.org/10.1007/s11207-005-6504-2>
- 198 Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,
199 Burovski, E., Peterson, P., Weckesser, W., Bright, J., Van Der Walt, S. J., Brett, M.,
200 Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E.,
201 ... Vazquez-Baeza, Y. (2020). SciPy 1.0: Fundamental algorithms for scientific computing
202 in Python. *Nature Methods*, 17(3), 261–272. <https://doi.org/10.1038/s41592-019-0686-2>