

PyPLUTO: a Data Analysis Python Package for the PLUTO Code

Giancarlo Mattia^{1,2}, Daniele Crocco³, David Melon Fuksman¹,
Matteo Bugli^{3,4,5,6}, Vittoria Berta^{3,6}, Eleonora Puzzone⁷, Andrea
Mignone^{3,6}, and Bhargav Vaidya⁸

¹ Max Planck Institut für Astronomie, Königstuhl 17, Heidelberg, 69117, Germany ² INFN, Sezione di Firenze, Via G. Sansone 1, Sesto Fiorentino (FI), 50019, Italy ³ Dipartimento di Fisica, Università di Torino, Via P. Giuria 1, Torino, 10125, Italy ⁴ Institut d'Astrophysique de Paris, UMR 7095, CNRS & Sorbonne Université, 98bis boulevard Arago, 75014 Paris, France ⁵ Université Paris-Saclay, Université Paris Cité, CEA, CNRS, AIM, Gif-sur-Yvette, 91191, France ⁶ INFN, Sezione di Torino, Via P. Giuria 1, Torino, 10125, Italy ⁷ Observatoire de la Côte d'Azur, Laboratoire Lagrange, Bd de l'Observatoire, CS 34229, 06304 Nice cedex 4, France ⁸ Department of Astronomy, Astrophysics and Space Engineering, Indian Institute of Technology, Khandwa Road, Simrol, Indore, 453552, India

DOI: 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Published: unpublished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

In recent years, numerical simulations have become indispensable for addressing complex astrophysical problems. The so-called magnetohydrodynamics (MHD) framework represents a key tool for investigating the dynamical evolution of astrophysical plasmas. This formalism consists of a set of partial differential equations (Chiuderi & Velli, 2015) that enforce the conservation of mass, momentum, and energy, along with Maxwell's equations for the evolution of the electromagnetic fields. Due to the high nonlinearity of the MHD equations (regardless of their specifications, e.g., classical/relativistic or ideal/resistive), a general analytical solution is not possible, making numerical approaches crucial. Numerical simulations usually produce large sets of data files, and their scientific analysis relies on dedicated software tools designed for data visualization (Ahrens et al., 2005; Childs et al., 2012). However, to encompass all code output features, specialized tools focusing on the numerical code may represent a more versatile and integrated solution. Here, we present PyPLUTO, a Python package tailored for efficient loading, manipulation, and visualization of outputs produced with the PLUTO code (Mignone et al., 2007; Mignone, Zanni, et al., 2012). PyPLUTO uses memory mapping to optimize data loading and provides general data manipulation and visualization routines. PyPLUTO also supports the particle modules of the PLUTO code, enabling users to load and visualize particles, such as cosmic rays (Mignone et al., 2018), Lagrangian (Vaidya et al., 2018), or dust (Mignone et al., 2019) particles, from hybrid simulations. A dedicated Graphical User Interface (GUI, shown in Fig. 1) simplifies the generation of single-subplot figures, making PyPLUTO a powerful yet user-friendly toolkit for astrophysical data analysis.

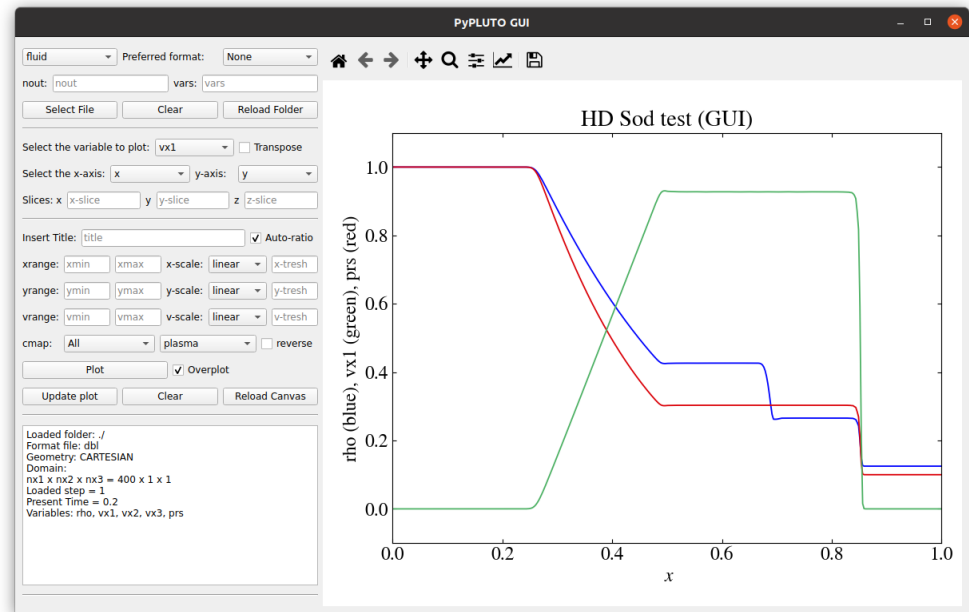


Figure 1: Interactive visualization of shock tube test results (i.e., density, pressure, and velocity profiles) with the GUI.

Statement of Need

The PLUTO code (Mignone et al., 2007) is a widely used, freely distributed computational fluid dynamics code designed to solve the classical and (special) relativistic MHD equations in different geometries and spatial dimensions. The original code is written in C (while the upcoming GPU version provides a complete C++ rewrite) and it contains several numerical methods adaptable to different contexts. Data post-processing is a crucial step in analyzing the results of any numerical simulation. Other packages addressing related needs (e.g., plutoplot) provide valuable functionality for working with PLUTO data, including loading and visualization. However, they may not support all data formats or offer integration for data manipulation and advanced plotting tasks. In this work, we present PyPLUTO, a Python package designed to efficiently load, manipulate, and visualize the output from the PLUTO code. The initial version of PyPLUTO (written by Bhargav Vaidya) is available here. PyPLUTO has since been completely rewritten and is now maintained at a new repository. The package retains its core strengths while offering user-friendly methods for generating publication-quality plots with high customization. In addition to its enhanced flexibility, PyPLUTO offers strong computational efficiency, enabling the rapid handling of large datasets typical of state-of-the-art numerical simulations. Through this balance between customization, performance, and ease of use, PyPLUTO represents a key tool to effectively communicate scientific results while minimizing the effort required for post-processing.

Main Features

PyPLUTO is a package written in Python (version 3.10) with the additions of NumPy (Harris et al., 2020), Matplotlib (Hunter, 2007), SciPy (Virtanen et al., 2020), pandas (team, 2020), h5py (Collette, 2013), and PyQt6 (although the last two are optional). The package, which can be installed through pip, primarily consists of three main classes:

59 ▪ The Load class loads and manipulates the PLUTO output files containing fluid-related
60 quantities.

61 ▪ The LoadPart class loads and manipulates the PLUTO output files containing particle-related
62 quantities.

63 ▪ The Image class produces and handles the graphical windows and the plotting procedures.

64 Additionally, a separate PyPLUTOApp class launches a GUI able to load and plot 1D and 2D
65 data in a single set of axes. PyPLUTO has been implemented to be supported by Windows,
66 MacOS, and Linux, through both standard scripts and more interactive tools (e.g., IPython or
67 Jupyter). The style guidelines follow the [PEP8](#) conventions for Python code, enforced through
68 the Black package ([Langa & Black, 2020](#)), and focus on clarity and code readability. Finally,
69 by leveraging the capabilities of the [sphinx package](#), PyPLUTO features extensive docstrings,
70 providing a helpful reference for both users and developers.

71 Benchmark Examples

72 PyPLUTO provides a set of benchmarks that are immediately accessible after the package
73 is installed. These consist of test problems that can be applied to relevant astrophysical
74 applications and showcase the full range of PyPLUTO's features. Here we report two examples
75 demonstrating the package's capabilities.

76 Disk-planet Interaction

77 This test problem examines particle acceleration near an X-type magnetic reconnection region
78 ([Puzzoni et al., 2021](#)). In the last decades, magnetic reconnection ([Bugli et al., 2024](#); [Mattia
79 et al., 2023](#)) has proven to be a key physical process to explain the population of non-thermal
80 particles in solar flares, relativistic outflows, and neutron star magnetospheres. This sort of test
81 provides valuable insights into particle acceleration mechanisms in high-energy astrophysical
82 environments by enabling the investigation of particle trajectories and energy distribution near
83 the X-point. This test simulates the interaction of a planet embedded in a disk ([Mignone,
84 Flock, et al., 2012](#)) and represents an ideal scenario for understanding the formation and
85 evolution of planetary systems. In particular, forming spiral density waves and disk gaps
86 represents some key observational signatures of planet formation and planet-disk interaction
87 ([Melon Fuksman et al., 2021](#); [Muley et al., 2024](#)). In the left panel of Fig. 2, we show an
88 adaptation of Figure 10 of ([Mignone, Flock, et al., 2012](#)), featuring two separate zoom-ins
89 around the planet's location.

- 90 ▪ The first zoom (upper-right subplot) shows an enlarged view of the density distribution
91 using the same color map and logarithmic scale as the global plot.
- 92 ▪ The second zoom (lower-left subplot) highlights the changes in toroidal velocity due to the
93 planet's presence by employing a different color map (to enhance the sign change) and a linear
94 color scale.

95 These zoomed views offer deeper insights into the physical processes at play and demonstrate
96 the utility of PyPLUTO for analyzing complex astrophysical systems.

97 Particles Accelerated near an X-point

98 This test problem examines particle acceleration near an X-type magnetic reconnection region
99 ([Puzzoni et al., 2021](#)). In the last decades, magnetic reconnection ([Bugli et al., 2024](#); [Mattia
100 et al., 2023](#)) has proven to be a key physical process to explain the population of non-thermal
101 particles in solar flares, relativistic outflows, and neutron star magnetospheres. This sort of test
102 provides valuable insights into particle acceleration mechanisms in high-energy astrophysical
103 environments by enabling the investigation of particle trajectories and energy distribution near
104 the X-point.

In the right panel of Fig. 2, we show an adaptation of the top panel of Figures 13-14 from (Mignone et al., 2018). The main plot displays the distribution of test particles, color-coded by their velocity magnitudes, with magnetic field lines overlaid as solid and dashed lines. The inset panel shows the energy spectrum at the initial ($t = 0$, in blue) and final ($t = 100$, in red) time. In this scenario, the absence of a guide field ($\vec{E} \cdot \vec{B} = 0$) results in a symmetric distribution along the y -axis from the combined effects of the gradient, curvature, and $\vec{E} \times \vec{B}$ drifts in the vicinity of the X-point, where the electric field is the strongest. This plot provides a clear visual representation of particle motion and energy changes, demonstrating how PyPLUTO can be used to investigate complex processes such as particle acceleration in astrophysical sources.

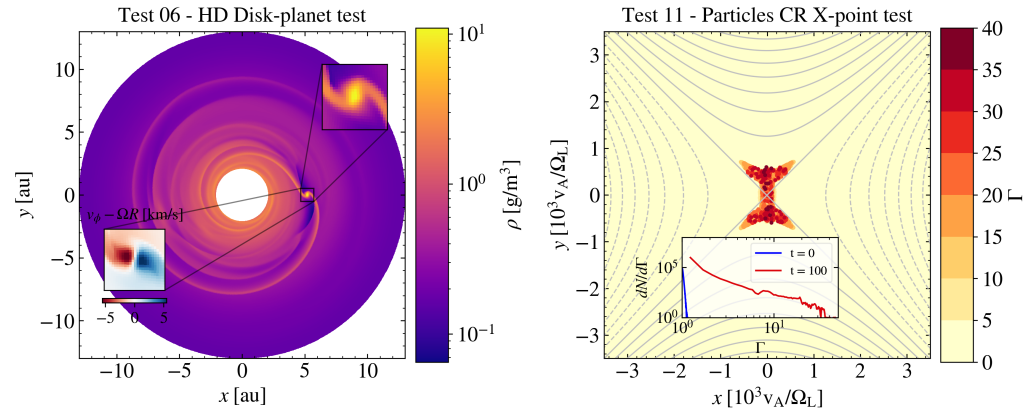


Figure 2: Left panel: example of inset zooms of the planet region of the disk-planet test problem. The main plot and the right zoom show the density on a logarithmic scale, while the left zoom highlights the toroidal velocity on a linear scale. Right panel: example of an X-point region with magnetic field lines overlaid (as contour lines of the vector potential, solid lines). The main plot shows the test-particle distribution, color-coded by velocity magnitudes, while the inset plot displays the particle energy spectrum at the beginning (in blue) and end (in red) of the simulation.

Ongoing research using PyPLUTO

Research applicable with PyPLUTO includes the development of numerical algorithms (Berta et al., 2024; Mattia & Mignone, 2022; Melon Fuksman et al., 2025) and numerical simulations of astrophysical objects, such as jets (Costa et al., 2025; Mattia et al., 2023, 2024; Mattia & Fendt, 2022; Sciacaluga et al., 2025), star clusters (Härer et al., 2025), and protoplanetary disks (Melon Fuksman et al., 2024a, 2024b), as well as physical processes, such as particle acceleration (Wang et al., 2024) and magnetic reconnection (Bugli et al., 2024).

Conclusion and Future Perspectives

The PyPLUTO package is designed as a powerful yet flexible tool to facilitate the data analysis and visualization of the output from PLUTO simulations, focusing on user-friendliness while allowing the necessary customization to produce publication-quality figures. To overcome current limitations and further enhance the package's capabilities, particular focus will be devoted to:

- introducing specific routines for rendering 3D data to provide users with tools for visualizing volumetric data;
- supporting interactive visualization and comparison of multiple simulation outputs, allowing the users to track temporal evolution directly with the GUI;

131 ▪ expanding the graphical interface to support particle data, including dynamic visualization
132 of particle distributions and trajectories.

133 Alongside these improvements, PyPLUTO development will focus on encompassing the latest
134 features of the PLUTO code, such as new Adaptive Mesh Refinement strategies and extensions
135 to more general metric tensors. PyPLUTO is a public package that can be downloaded
136 alongside the [CPU and GPU versions of the PLUTO code](#). Regular updates will be released with
137 improvements and bug fixes. Additionally, a [repository](#) containing the PyPLUTO development
138 versions will be available for users who wish to exploit the code's latest features in advance.

139 Acknowledgments

140 The authors thank the reviewers for improving this work with valuable comments and suggestions.
141 G. Mattia thanks L. Del Zanna and M. Flock for the discussions on data visualization and the
142 Data Science Department of the Max Planck Institute for Astronomy for helping with Python
143 and Matplotlib. The authors thank Simeon Doetsch for their insights on memory mapping
144 techniques and Deniss Stepanovs and Antoine Strugarek for contributing to previous PyPLUTO
145 versions throughout the years. The authors thank Agnese Costa, Alberto Sciacaluga, Alessio
146 Suriano, Asmita Bhandare, Dhruv Muley, Dipanjan Mukherjee, Jieshuang Wang, Jacksen
147 Narvaez, Lucia Haerer, Prakruti Sudarshan, Stefano Truzzi, and Stella Boula for testing the
148 module while it was under development. M. Bugli acknowledges the support of the French
149 Agence Nationale de la Recherche (ANR), under grant ANR-24-ERCS-0006 (project BlackJET).
150 This project has received funding from the European Union's Horizon Europe research and
151 innovation programme under the Marie Skłodowska-Curie grant agreement No 101064953
152 (GR-PLUTO), and from the European High Performance Computing Joint Undertaking (JU)
153 and Belgium, Czech Republic, France, Germany, Greece, Italy, Norway, and Spain under grant
154 agreement No 101093441 (SPACE).

155 References

- 156 Ahrens, J. P., Geveci, B., & Law, C. (2005). ParaView: An end-user tool for large-data
157 visualization. In *Visualization handbook* (pp. 717–731). Unknown Publisher. <https://doi.org/10.1016/B978-012387582-2/50038-1>
158
- 159 Berta, V., Mignone, A., Bugli, M., & Mattia, G. (2024). A 4th-order accurate finite volume
160 method for ideal classical and special relativistic MHD based on pointwise reconstructions.
161 *Journal of Computational Physics*, 499, 112701. <https://doi.org/10.1016/j.jcp.2023.112701>
- 162 Bugli, M., Lopresti, E. F., Figueiredo, E., Mignone, A., Cerutti, B., Mattia, G., Del Zanna,
163 L., Bodo, G., & Berta, V. (2024). Relativistic reconnection with effective resistivity: I.
164 Dynamics and reconnection rate. *arXiv e-Prints*, arXiv:2410.20924. <https://doi.org/10.48550/arXiv.2410.20924>
165
- 166 Childs, H., Brugger, E., Whitlock, B., Meredith, J., Ahern, S., Pugmire, D., Biagas, K.,
167 Miller, M., Harrison, C., Weber, G. H., Krishnan, H., Fogal, T., Sanderson, A., Garth,
168 C., Bethel, E. W., Camp, D., Rübel, O., Durant, M., Favre, J. M., & Navrátil, P.
169 (2012). VisIt: An end-user tool for visualizing and analyzing very large data. In *High*
170 *performance visualization—enabling extreme-scale scientific insight* (pp. 357–372). <https://doi.org/10.1201/b12985>
171
- 172 Chiuderi, C., & Velli, M. (2015). *Basics of Plasma Astrophysics*. <https://doi.org/10.1007/978-88-470-5280-2>
173
- 174 Collette, A. (2013). *Python and HDF5*. O'Reilly.
- 175 Costa, A., Bodo, G., Tavecchio, F., Rossi, P., Coppi, P., Sciacaluga, A., & Boula, S. (2025).
176 How do recollimation-induced instabilities shape the propagation of hydrodynamic relativistic

- 177 jets? *arXiv e-Prints*, arXiv:2503.18602. <https://doi.org/10.48550/arXiv.2503.18602>
- 178 Härer, L., Vieu, T., & Reville, B. (2025). Stellar-wind feedback and magnetic fields around
179 young compact star clusters: 3D magnetohydrodynamics simulations. *698*, A6. <https://doi.org/10.1051/0004-6361/202554057>
- 180
- 181 Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau,
182 D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van
183 Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ...
184 Oliphant, T. E. (2020). Array programming with NumPy. *585*(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- 185
- 186 Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science and*
187 *Engineering*, *9*(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- 188 Langa, Ł., & Black, contributors to. (2020). *Black: The uncompromising python code*
189 *formatter*. <https://black.readthedocs.io/en/stable/>
- 190 Mattia, G., Del Zanna, L., Bugli, M., Pavan, A., Ciolfi, R., Bodo, G., & Mignone, A.
191 (2023). Resistive relativistic MHD simulations of astrophysical jets. *679*, A49. <https://doi.org/10.1051/0004-6361/202347126>
- 192
- 193 Mattia, G., Del Zanna, L., Pavan, A., & Ciolfi, R. (2024). Magnetic dissipation in short
194 gamma-ray-burst jets: I. Resistive relativistic MHD evolution in a model environment. *691*,
195 A105. <https://doi.org/10.1051/0004-6361/202451528>
- 196 Mattia, G., & Fendt, C. (2022). Jets from Accretion Disk Dynamos: Consistent Quenching
197 Modes for Dynamo and Resistivity. *935*(1), 22. <https://doi.org/10.3847/1538-4357/ac7d59>
- 198
- 199 Mattia, G., & Mignone, A. (2022). A comparison of approximate non-linear Riemann solvers
200 for Relativistic MHD. *510*(1), 481–499. <https://doi.org/10.1093/mnras/stab3373>
- 201 Melon Fuksman, D., Flock, M., & Klahr, H. (2024a). Vertical shear instability in two-moment
202 radiation-hydrodynamical simulations of irradiated protoplanetary disks. I. Angular momen-
203 tum transport and turbulent heating. *682*, A139. <https://doi.org/10.1051/0004-6361/202346554>
- 204
- 205 Melon Fuksman, D., Flock, M., & Klahr, H. (2024b). Vertical shear instability in two-moment
206 radiation-hydrodynamical simulations of irradiated protoplanetary disks. II. Secondary insta-
207 bilities and stability regions. *682*, A140. <https://doi.org/10.1051/0004-6361/202346555>
- 208 Melon Fuksman, D., Flock, M., Klahr, H., Mattia, G., & Muley, D. (2025). Multidimensional
209 half-moment multigroup radiative transfer. Improving moment-based thermal models of
210 circumstellar disks. *arXiv e-Prints*, arXiv:2504.13999. <https://doi.org/10.48550/arXiv.2504.13999>
- 211
- 212 Melon Fuksman, D., Klahr, H., Flock, M., & Mignone, A. (2021). A Two-moment Radiation
213 Hydrodynamics Scheme Applicable to Simulations of Planet Formation in Circumstellar
214 Disks. *906*(2), 78. <https://doi.org/10.3847/1538-4357/abc879>
- 215 Mignone, A., Bodo, G., Massaglia, S., Matsakos, T., Tesileanu, O., Zanni, C., & Ferrari, A.
216 (2007). PLUTO: A Numerical Code for Computational Astrophysics. *170*(1), 228–242.
217 <https://doi.org/10.1086/513316>
- 218 Mignone, A., Bodo, G., Vaidya, B., & Mattia, G. (2018). A Particle Module for the PLUTO
219 Code. I. An Implementation of the MHD-PIC Equations. *859*(1), 13. <https://doi.org/10.3847/1538-4357/aabccd>
- 220
- 221 Mignone, A., Flock, M., Stute, M., Kolb, S. M., & Muscianisi, G. (2012). A conservative
222 orbital advection scheme for simulations of magnetized shear flows with the PLUTO code.
223 *545*, A152. <https://doi.org/10.1051/0004-6361/201219557>

- 224 Mignone, A., Flock, M., & Vaidya, B. (2019). A Particle Module for the PLUTO Code. III.
225 Dust. 244(2), 38. <https://doi.org/10.3847/1538-4365/ab4356>
- 226 Mignone, A., Zanni, C., Tzeferacos, P., van Straalen, B., Colella, P., & Bodo, G. (2012). The
227 PLUTO Code for Adaptive Mesh Computations in Astrophysical Fluid Dynamics. 198(1),
228 7. <https://doi.org/10.1088/0067-0049/198/1/7>
- 229 Muley, D., Melon Fuksman, D., & Klahr, H. (2024). Three-temperature radiation hydrody-
230 namics with PLUTO: Thermal and kinematic signatures of accreting protoplanets. 687,
231 A213. <https://doi.org/10.1051/0004-6361/202449739>
- 232 Puzzeni, E., Mignone, A., & Bodo, G. (2021). On the impact of the numerical method on
233 magnetic reconnection and particle acceleration - I. The MHD case. 508(2), 2771–2783.
234 <https://doi.org/10.1093/mnras/stab2813>
- 235 Sciacaluga, A., Costa, A., Tavecchio, F., Bodo, G., Coppi, P., & Boula, S. (2025). The
236 polarization of the synchrotron radiation from a recollimated jet: Application to high-energy
237 BL Lacs. 699, A296. <https://doi.org/10.1051/0004-6361/20254490>
- 238 team, T. pandas development. (2020). *Pandas-dev/pandas: pandas* (latest). Zenodo.
239 <https://doi.org/10.5281/zenodo.3509134>
- 240 Vaidya, B., Mignone, A., Bodo, G., Rossi, P., & Massaglia, S. (2018). A Particle Module
241 for the PLUTO Code. II. Hybrid Framework for Modeling Nonthermal Emission from
242 Relativistic Magnetized Flows. 865(2), 144. <https://doi.org/10.3847/1538-4357/aadd17>
- 243 Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,
244 Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M.,
245 Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ...
246 SciPy 1.0 Contributors. (2020). SciPy 1.0: fundamental algorithms for scientific computing
247 in Python. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- 248 Wang, J.-S., Reville, B., Rieger, F. M., & Aharonian, F. A. (2024). Acceleration of Ultra-high-
249 energy Cosmic Rays in the Kiloparsec-scale Jets of Nearby Radio Galaxies. 977(1), L20.
250 <https://doi.org/10.3847/2041-8213/ad9589>