

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](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Published: unpublished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](https://creativecommons.org/licenses/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.

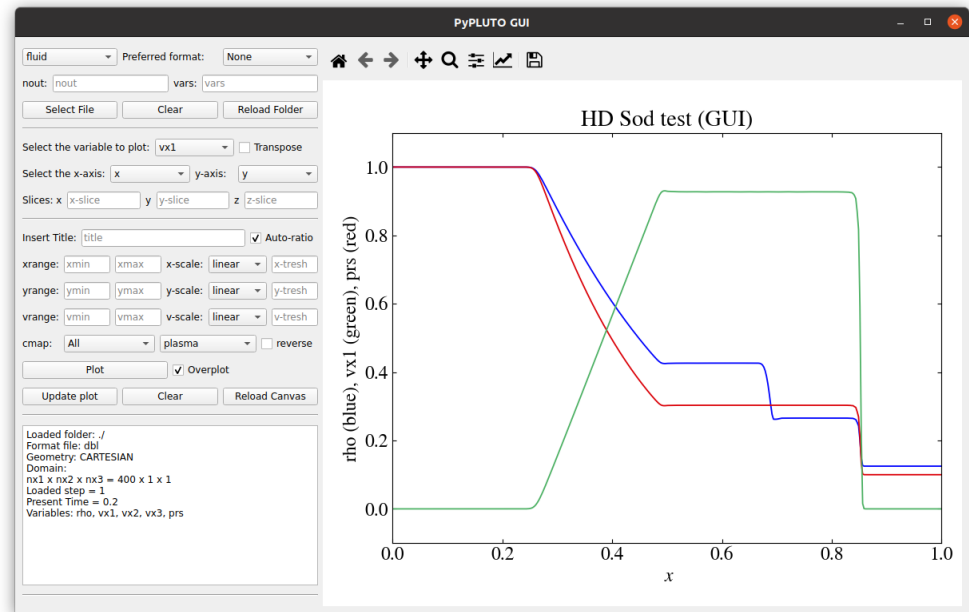


Figure 1: Interactive visualization of shock tube test results (i.e., density, pressure, and velocity profiles) with the GUI.

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 for Python 3.10 or later with the additions of external packages like NumPy (Harris et al., 2020), Matplotlib (Hunter, 2007), SciPy (Virtanen et al., 2020), pandas (team, 2020), h5py (Collette, 2013), and PyQt6. 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 simulates the interaction of a planet embedded in a disk ([Mignone, Flock, et al.,
78 2012](#)) and represents an ideal scenario for understanding the formation and evolution of
79 planetary systems. In particular, forming spiral density waves and disk gaps represents some
80 key observational signatures of planet formation and planet-disk interaction ([Melon Fuksman
81 et al., 2021](#); [Muley et al., 2024](#)). In the left panel of Fig. 2, we show an adaptation of Figure
82 10 of ([Mignone, Flock, et al., 2012](#)), featuring two separate zoom-ins around the planet's
83 location.

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

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

91 Particles Accelerated near an X-point

92 This test problem examines particle acceleration near an X-type magnetic reconnection region
93 ([Puzzoni et al., 2021](#)). In the last decades, magnetic reconnection ([Bugli et al., 2025](#); [Mattia
94 et al., 2023](#)) has proven to be a key physical process to explain the population of non-thermal
95 particles in solar flares, relativistic outflows, and neutron star magnetospheres. This sort of test
96 provides valuable insights into particle acceleration mechanisms in high-energy astrophysical
97 environments by enabling the investigation of particle trajectories and energy distribution near
98 the X-point.

99 In the right panel of Fig. 2, we show an adaptation of the top panel of Figures 13-14 from
100 ([Mignone et al., 2018](#)). The main plot displays the distribution of test particles, color-coded by
101 their velocity magnitudes, with magnetic field lines overlaid as solid and dashed lines. The inset
102 panel shows the energy spectrum at the initial ($t = 0$, in blue) and final ($t = 100$, in red) time.
103 In this scenario, the absence of a guide field ($\vec{E} \cdot \vec{B} = 0$) results in a symmetric distribution

104 along the y-axis from the combined effects of the gradient, curvature, and $\vec{E} \times \vec{B}$ drifts in the
105 vicinity of the X-point, where the electric field is the strongest. This plot provides a clear visual
106 representation of particle motion and energy changes, demonstrating how PyPLUTO can be
107 used to investigate complex processes such as particle acceleration in astrophysical sources.

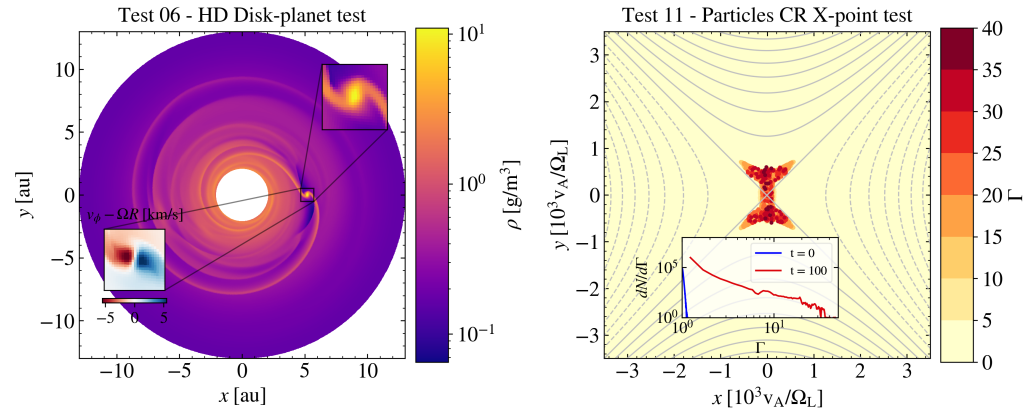


Figure 2: Left panel: example of inset zooms of the planet region of the disk-planet test problem. The main plot and the right zoom show the density on a logarithmic scale, while the left zoom highlights the toroidal velocity on a linear scale. Right panel: example of an X-point region with magnetic 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., 2025).

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;
- expanding the graphical interface to support particle data, including dynamic visualization of particle distributions and trajectories.

Alongside these improvements, PyPLUTO development will focus on encompassing the latest features of the PLUTO code, such as new Adaptive Mesh Refinement strategies and extensions to more general metric tensors. PyPLUTO is a public package that can be downloaded

130 alongside the [CPU and GPU versions of the PLUTO code](#). Regular updates will be released with
 131 improvements and bug fixes. Additionally, a [repository](#) containing the PyPLUTO development
 132 versions will be available for users who wish to exploit the code's latest features in advance.

133 Acknowledgments

134 The authors thank the reviewers for improving this work with valuable comments and suggestions.
 135 G. Mattia thanks L. Del Zanna and M. Flock for the discussions on data visualization and the
 136 Data Science Department of the Max Planck Institute for Astronomy for helping with Python
 137 and Matplotlib.

138 The authors thank Simeon Doetsch for their insights on memory mapping techniques and
 139 Deniss Stepanovs and Antoine Strugarek for contributing to previous PyPLUTO versions
 140 throughout the years. The authors thank Agnese Costa, Alberto Sciacaluga, Alessio Suriano,
 141 Asmita Bhandare, Dhruv Muley, Dipanjan Mukherjee, Jieshuang Wang, Jackson Narvaez, Lucia
 142 Haerer, Prakruti Sudarshan, Stefano Truzzi, and Stella Boula for testing the module while it
 143 was under development. M. Bugli acknowledges the support of the French Agence Nationale
 144 de la Recherche (ANR), under grant ANR-24-ERCS-0006 (project BlackJET). This project
 145 has received funding from the European Union's Horizon Europe research and innovation
 146 programme under the Marie Skłodowska-Curie grant agreement No 101064953 (GR-PLUTO),
 147 and from the European High Performance Computing Joint Undertaking (JU) and Belgium,
 148 Czech Republic, France, Germany, Greece, Italy, Norway, and Spain under grant agreement No
 149 101093441 (SPACE).

150 References

- 151 Ahrens, J. P., Geveci, B., & Law, C. (2005). ParaView: An end-user tool for large-data
 152 visualization. In *Visualization handbook* (pp. 717–731). Unknown Publisher. <https://doi.org/10.1016/B978-012387582-2/50038-1>
 153
- 154 Berta, V., Mignone, A., Bugli, M., & Mattia, G. (2024). A 4th-order accurate finite volume
 155 method for ideal classical and special relativistic MHD based on pointwise reconstructions.
 156 *Journal of Computational Physics*, 499, 112701. <https://doi.org/10.1016/j.jcp.2023.112701>
- 157 Bugli, M., Lopresti, E. F., Figueiredo, E., Mignone, A., Cerutti, B., Mattia, G., Del Zanna,
 158 L., Bodo, G., & Berta, V. (2025). Relativistic reconnection with effective resistivity:
 159 I. Dynamics and reconnection rate. *Astronomy and Astrophysics*, 693, A233. <https://doi.org/10.1051/0004-6361/202452277>
 160
- 161 Childs, H., Brugger, E., Whitlock, B., Meredith, J., Ahern, S., Pugmire, D., Biagas, K.,
 162 Miller, M., Harrison, C., Weber, G. H., Krishnan, H., Fogal, T., Sanderson, A., Garth,
 163 C., Bethel, E. W., Camp, D., Rübel, O., Durant, M., Favre, J. M., & Navrátil, P.
 164 (2012). VisIt: An end-user tool for visualizing and analyzing very large data. In *High*
 165 *performance visualization—enabling extreme-scale scientific insight* (pp. 357–372). <https://doi.org/10.1201/b12985>
 166
- 167 Chiuderi, C., & Velli, M. (2015). *Basics of Plasma Astrophysics*. <https://doi.org/10.1007/978-88-470-5280-2>
 168
- 169 Collette, A. (2013). *Python and HDF5*. O'Reilly.
- 170 Costa, A., Bodo, G., Tavecchio, F., Rossi, P., Coppi, P., Sciacaluga, A., & Boula, S. (2025).
 171 How do recollimation-induced instabilities shape the propagation of hydrodynamic relativistic
 172 jets? *arXiv e-Prints*, arXiv:2503.18602. <https://doi.org/10.48550/arXiv.2503.18602>
- 173 Härer, L., Vieu, T., & Reville, B. (2025). Stellar-wind feedback and magnetic fields around
 174 young compact star clusters: 3D magnetohydrodynamics simulations. *Astronomy and*

- 175 *Astrophysics*, 698, A6. <https://doi.org/10.1051/0004-6361/202554057>
- 176 Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau,
177 D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van
178 Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ...
179 Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362.
180 <https://doi.org/10.1038/s41586-020-2649-2>
- 181 Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science and*
182 *Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- 183 Langa, Ł., & Black, contributors to. (2020). *Black: The uncompromising python code*
184 *formatter*. <https://black.readthedocs.io/en/stable/>
- 185 Mattia, G., Del Zanna, L., Bugli, M., Pavan, A., Ciolfi, R., Bodo, G., & Mignone, A. (2023).
186 Resistive relativistic MHD simulations of astrophysical jets. *Astronomy and Astrophysics*,
187 679, A49. <https://doi.org/10.1051/0004-6361/202347126>
- 188 Mattia, G., Del Zanna, L., Pavan, A., & Ciolfi, R. (2024). Magnetic dissipation in short
189 gamma-ray-burst jets: I. Resistive relativistic MHD evolution in a model environment.
190 *Astronomy and Astrophysics*, 691, A105. <https://doi.org/10.1051/0004-6361/202451528>
- 191 Mattia, G., & Fendt, C. (2022). Jets from Accretion Disk Dynamos: Consistent Quenching
192 Modes for Dynamo and Resistivity. *The Astrophysical Journal*, 935(1), 22. <https://doi.org/10.3847/1538-4357/ac7d59>
- 193 <https://doi.org/10.3847/1538-4357/ac7d59>
- 194 Mattia, G., & Mignone, A. (2022). A comparison of approximate non-linear Riemann solvers
195 for Relativistic MHD. *Monthly Notices of the Royal Astronomical Society*, 510(1), 481–499.
196 <https://doi.org/10.1093/mnras/stab3373>
- 197 Melon Fuksman, D., Flock, M., & Klahr, H. (2024a). Vertical shear instability in two-
198 moment radiation-hydrodynamical simulations of irradiated protoplanetary disks. I. Angular
199 momentum transport and turbulent heating. *Astronomy and Astrophysics*, 682, A139.
200 <https://doi.org/10.1051/0004-6361/202346554>
- 201 Melon Fuksman, D., Flock, M., & Klahr, H. (2024b). Vertical shear instability in two-moment
202 radiation-hydrodynamical simulations of irradiated protoplanetary disks. II. Secondary
203 instabilities and stability regions. *Astronomy and Astrophysics*, 682, A140. <https://doi.org/10.1051/0004-6361/202346555>
- 204 <https://doi.org/10.1051/0004-6361/202346555>
- 205 Melon Fuksman, D., Flock, M., Klahr, H., Mattia, G., & Muley, D. (2025). Multidimensional
206 half-moment multigroup radiative transfer: Improving moment-based thermal models of
207 circumstellar disks. *Astronomy and Astrophysics*, 701, A97. [https://doi.org/10.1051/](https://doi.org/10.1051/0004-6361/202554994)
208 [0004-6361/202554994](https://doi.org/10.1051/0004-6361/202554994)
- 209 Melon Fuksman, D., Klahr, H., Flock, M., & Mignone, A. (2021). A Two-moment Radiation
210 Hydrodynamics Scheme Applicable to Simulations of Planet Formation in Circumstellar
211 Disks. *The Astrophysical Journal*, 906(2), 78. <https://doi.org/10.3847/1538-4357/abc879>
- 212 Mignone, A., Bodo, G., Massaglia, S., Matsakos, T., Tesileanu, O., Zanni, C., & Ferrari, A.
213 (2007). PLUTO: A Numerical Code for Computational Astrophysics. *The Astrophysical*
214 *Journal Supplement Series*, 170(1), 228–242. <https://doi.org/10.1086/513316>
- 215 Mignone, A., Bodo, G., Vaidya, B., & Mattia, G. (2018). A Particle Module for the PLUTO
216 Code. I. An Implementation of the MHD-PIC Equations. *The Astrophysical Journal*,
217 859(1), 13. <https://doi.org/10.3847/1538-4357/aabccd>
- 218 Mignone, A., Flock, M., Stute, M., Kolb, S. M., & Muscianisi, G. (2012). A conservative
219 orbital advection scheme for simulations of magnetized shear flows with the PLUTO code.
220 *Astronomy and Astrophysics*, 545, A152. <https://doi.org/10.1051/0004-6361/201219557>
- 221 Mignone, A., Flock, M., & Vaidya, B. (2019). A Particle Module for the PLUTO Code. III.

- 222 Dust. *The Astrophysical Journal Supplement Series*, 244(2), 38. [https://doi.org/10.3847/](https://doi.org/10.3847/1538-4365/ab4356)
 223 [1538-4365/ab4356](https://doi.org/10.3847/1538-4365/ab4356)
- 224 Mignone, A., Zanni, C., Tzeferacos, P., van Straalen, B., Colella, P., & Bodo, G. (2012). The
 225 PLUTO Code for Adaptive Mesh Computations in Astrophysical Fluid Dynamics. *The*
 226 *Astrophysical Journal Supplement Series*, 198(1), 7. [https://doi.org/10.1088/0067-0049/](https://doi.org/10.1088/0067-0049/198/1/7)
 227 [198/1/7](https://doi.org/10.1088/0067-0049/198/1/7)
- 228 Muley, D., Melon Fuksman, D., & Klahr, H. (2024). Three-temperature radiation hydrodynam-
 229 ics with PLUTO: Thermal and kinematic signatures of accreting protoplanets. *Astronomy*
 230 *and Astrophysics*, 687, A213. <https://doi.org/10.1051/0004-6361/202449739>
- 231 Puzzoni, E., Mignone, A., & Bodo, G. (2021). On the impact of the numerical method on
 232 magnetic reconnection and particle acceleration - I. The MHD case. *Monthly Notices of the*
 233 *Royal Astronomical Society*, 508(2), 2771–2783. <https://doi.org/10.1093/mnras/stab2813>
- 234 Sciacaluga, A., Costa, A., Tavecchio, F., Bodo, G., Coppi, P., & Boula, S. (2025). The
 235 polarization of the synchrotron radiation from a recollimated jet: Application to high-energy
 236 BL Lacs. *Astronomy and Astrophysics*, 699, A296. [https://doi.org/10.1051/0004-6361/](https://doi.org/10.1051/0004-6361/202554490)
 237 [202554490](https://doi.org/10.1051/0004-6361/202554490)
- 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. *The Astrophysical Journal*, 865(2), 144. [https://doi.org/](https://doi.org/10.3847/1538-4357/aadd17)
 243 [10.3847/1538-4357/aadd17](https://doi.org/10.3847/1538-4357/aadd17)
- 244 Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,
 245 Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M.,
 246 Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ...
 247 SciPy 1.0 Contributors. (2020). SciPy 1.0: fundamental algorithms for scientific computing
 248 in Python. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- 249 Wang, J.-S., Reville, B., Rieger, F. M., & Aharonian, F. A. (2024). Acceleration of Ultra-
 250 high-energy Cosmic Rays in the Kiloparsec-scale Jets of Nearby Radio Galaxies. *The*
 251 *Astrophysical Journal Letters*, 977(1), L20. <https://doi.org/10.3847/2041-8213/ad9589>