Solar-MACH: An open-source tool to analyze solar magnetic connection configurations

Jan Gieseler¹*, Nina Dresing¹, Christian Palmroos¹, Johan L. Freiherr von Forstner²,³, Daniel J. Price⁴, Rami Vainio¹, Athanasios Kouloumvakos⁵, Laura Rodríguez-García⁶, Domenico Trotta⁷, Vincent Génot⁸, Arnaud Masson⁹, Markus Roth¹⁰ and Astrid Veronig¹¹

¹Space Research Laboratory, Department of Physics and Astronomy, University of Turku, Turku, Finland, ²Institute of Experimental and Applied Physics, Kiel University, Kiel, Germany, ³Now at Paradox Cat GmbH, München, Germany, ⁴Department of Physics, University of Helsinki, Helsinki, Finland, ⁵Applied Physics Laboratory, The Johns Hopkins University, Laurel, MD, United States, ⁶Space Research Group, Universidad de Alcalá, Alcalá de Henares, Spain, ⁷The Blackett Laboratory, Department of Physics, Imperial College London, London, United Kingdom, ⁸Institut de Recherche en Astrophysique et Planétologie, CNRS, CNES, Université Paul Sabatier, Toulouse, France, ⁹ESAC Science Data Centre, European Space Agency, Madrid, Spain, ¹⁰Thüringer Landessternwarte, Tautenburg, Germany, ¹¹Kanzelhöhe Observatory for Solar and Atmospheric Research, Institute of Physics, University of Graz, Graz, Austria

The Solar MAgnetic Connection HAUS¹ tool (Solar-MACH) is an open-source tool completely written in Python that derives and visualizes the spatial configuration and solar magnetic connection of different observers (i.e., spacecraft or planets) in the heliosphere at different times. For doing this, the magnetic connection in the interplanetary space is obtained by the classic Parker Heliospheric Magnetic Field (HMF). In close vicinity of the Sun, a Potential Field Source Surface (PFSS) model can be applied to connect the HMF to the solar photosphere. Solar-MACH is especially aimed at providing publication-ready figures for the analyses of Solar Energetic Particle events (SEPs) or solar transients such as Coronal Mass Ejections (CMEs). It is provided as an installable Python package (listed on PyPI and conda-forge), but also as a web tool at solar-mach.github.io that completely runs in any web browser and requires neither Python knowledge nor installation. The development of Solar-MACH is open to everyone and takes place on GitHub, where the source code is publicly available under the BSD 3-Clause License. Established Python libraries like sunpy and pfsspy are utilized to obtain functionalities when possible. In this article, the Python code of Solar-MACH is explained, and its functionality is demonstrated using real science examples. In addition, we introduce the overarching SERPENTINE project, the umbrella under which the recent development took place.
1 Introduction

The Solar energetic particle analysis platform for the inner heliosphere (SERPENTINE, 2021–2024) is a 42-month long project funded through the H2020-SPACE-2020 call of the European Union’s Horizon 2020 framework programme. The project addresses several outstanding questions on the origin of solar energetic particle (SEP) events and provides an advanced data analysis and visualization platform that will benefit the whole heliophysics community. It utilizes the most recent European and US missions, i.e., Solar Orbiter (Müller et al., 2020), Parker Solar Probe (Fox et al., 2016) and BepiColombo (Benkhoff et al., 2021). These observations are complemented with supporting data from several current missions near Earth’s orbit as well as ground-based radio imaging and spectroscopic observations by the European Low Frequency Array (LOFAR; van Haarlem et al., 2013).

SEP events are large and sporadic outbursts of charged particle radiation from the solar corona that are related to solar eruptions such as flares and coronal mass ejections (CMEs; e.g., Reames, 1999). They can be classified as impulsive and gradual events, based on their duration and the duration of the related solar X-ray flare. Impulsive SEP events are associated with impulsive flares and narrow CMEs, and they are enriched in electrons, ³He isotope and heavy ions. Gradual SEP events are associated with gradual solar X-ray flares and fast and wide CMEs, and their abundances resemble nominal coronal abundances (Reames, 2013; Desai and Giacalone, 2016). Gradual events are usually broader in their helio-longitudinal extent than impulsive events and their intensities are also typically larger, making them the main concern of spacecraft operations and crews (Vainio et al., 2009).

The primary reason for the broad spatial extent of some gradual events is still unresolved. It could be due to a broad source, like a global coronal shock driven by a CME (e.g., Lario et al., 2016), due to efficient particle transport processes across the heliospheric magnetic field, or as a result of both mechanisms (e.g., Dresing et al., 2012; Rodriguez-García et al., 2021). The main objective of SERPENTINE is to pinpoint the primary causes of large gradual and widespread SEP events. To address this objective, SERPENTINE will answer the following open science questions:

Q1: What are the primary causes for widespread SEP events observed in the heliosphere?

Q2: What are the shock acceleration mechanisms responsible for accelerating ions from thermal/suprathermal energies to near-relativistic energies in the corona and in the interplanetary medium?

Q3: What is the role of shocks in electron acceleration in large gradual and widespread events? How does it relate to ion acceleration and what is its importance relative to flare acceleration?

To reach these goals and also to broaden the impact of the project, SERPENTINE will develop and release to the community a platform of tools for analyzing SEP events. Furthermore, the tools may also be useful for the broader heliophoric community looking at different aspects of solar activity or solar wind phenomena. Part of this platform will be a JupyterHub server that provides free access to the tools developed by the SERPENTINE project as Jupyter Notebooks, without requiring any installations beyond a web browser. This manuscript and the accompanying papers (Kouloumvakos et al., 2022; Palmroos et al., 2022; Trotta et al., 2022) will present the first batch of these tools to the heliophysics community.

Because SEPs are measured in situ as enhancements of the energetic particle fluxes, the presence of multiple, well-separated observers is indispensable to study widespread SEP events (e.g., Dresing et al., 2014; Richardson et al., 2014). The new space missions in combination with established spacecraft form a fleet that is ideal for this purpose, as it covers varying heliocentric distances and large longitudinal ranges around the Sun. The different orbits of the multiple spacecraft, in combination with varying source regions at the Sun, constantly form new constellations, building the baseline for in-depth SEP event analyses. The first released tool of SERPENTINE, Solar-MACH, provides the user with a quick overview of these specific constellations for a selected time, as shown in the example of Figure 1. “Solar-MACH” is an abbreviation for Solar MAgnetic Connection HAUS, with HAUS standing for the Heliophysics Archives USer group at ESA. “MACH” is intended to be pronounced like the Mach number.

2 Method

The main functionality of Solar-MACH is to provide the user with a polar plot showing a Sun-centered top view of the heliographic equatorial plane with the constellation of different observers for a given time, as shown in Figure 1. Additional information can be added, such as indicating the magnetic connection of each observer to the Sun as given by an idealized
FIGURE 1
Solar-MACH plot for the time of the ground-level enhancement (GLE) event on 28 October 2021. Numbered symbols indicate the observers’ locations and the spiral lines corresponding HMF lines connecting them to the Sun. Radial distance is provided in astronomical units (AU), and the angular information is given in Carrington longitude. The arrow points out the freely choosable location of a “reference” (e.g., a solar flare) at the Sun, and the dashed spiral line indicates a corresponding HMF line originating at that position.

classic Parker heliospheric magnetic field line, whose curvature is dependent on the radial solar wind speed. This and other options are described in Section 2.1.

All software operations are performed within Python 3 (e.g., Van Rossum and Drake, 2009); the corresponding package solarmach is described in Section 2.2. In addition, a web tool is provided that requires neither Python knowledge nor installation. This web tool and its technical background are presented in Section 2.4.

2.1 General functionality

2.1.1 Ephemeris

After the user provides a specific date and time, the spatial coordinates of all requested observers are obtained dynamically from JPL Horizons, an online Solar System data and ephemeris computation service that is maintained by the Solar System Dynamics Group of the Jet Propulsion Laboratory (JPL). In addition to the web interface, JPL Horizons provides an API for programmatic control. This API is utilized through version 4.0.5 (Mumford et al., 2022) of the sunpy open-source software package (The SunPy Community et al., 2020), which itself uses Astropy’s (Astropy Collaboration et al., 2013; Astropy Collaboration et al., 2018; Astropy Collaboration et al., 2022) affiliated package astroquery (Ginsburg et al., 2021) for this functionality. The ephemeris information is returned per observer within Python as a SkyCoord object, which inherently provides coordinate system transformations. This allows easily supporting different heliographic coordinate systems. At the moment, the user can decide whether all operations should be carried out in Stonyhurst or Carrington coordinates (e.g., Thompson, 2006), with the latter being the default setting.

2.1.2 Magnetic connectivity

One important property for analyzing SEP events is the magnetic connection of each observer to the Sun. A simple approximation for this connection can be obtained by assuming an ideal Archimedean spiral configuration for the heliospheric field (Parker, 1958), which is defined in spherical coordinates by:

$$\psi(r) = \psi_0 + \frac{\omega(\theta)}{v_{sw}} \cdot (R - r) \cdot \cos \theta,$$

(1)

with heliographic longitude $\psi$ as a function of radial distance $r$, heliographic latitude $\theta$, differential solar rotation frequency $\omega(\theta)$, radial solar wind speed $v_{sw}$, as well as radial distance $R$ and longitude $\psi_0$ of the observer. The differential solar rotation frequency $\omega(\theta)$ takes into account that the Sun’s rotation speed varies with latitude. Here, we apply an empirical model by Poljančić Beljan et al. (2017) that describes $\omega(\theta)$ by $14.50 - 2.87 \sin^2 \theta$ (deg/day), which corresponds to

Solar-MACH
https://solar-mach.github.io
Example of using the Python package `solarmach` in a Jupyter Notebook for the SEP event on 9 October 2021.
14.50 (deg/day) = 1.678241 · 10⁻⁴ (deg/s) at the heliographic equator. For the two-dimensional visualization, all spherical coordinates are projected to the heliographic equatorial plane. The resulting field lines are shown in Figure 1 as solid spiral lines, with color-coding of the respective spacecraft. As the spatial coordinates are dynamically obtained for each spacecraft, the solar wind speed \( v_{sw} \) is the only free parameter that needs to be provided for each observer in units of km/s. By default, this is assumed to be 400 km/s. Automatically obtaining real measurements of the solar wind speed is planned as an option for the future. Applying Eq. 1 for \( r = R_\odot \) provides the magnetic footpoint coordinate for each spacecraft, that is, the heliographic longitude where the assumed magnetic field line originates at the Sun. This information is especially important in order to connect in situ measurements of SEPs with remote-sensing observations of the associated source regions at the Sun. We must point out, though, that this analysis is just an estimation that may deviate significantly from reality. Employing this tool for magnetic backmapping purposes comes with the known caveats of the simple assumptions of ballistic backmapping, which are a constant and totally radial solar wind speed as well as the absence of interplanetary structures (e.g., interplanetary CMEs) that could deform the ideal Parker field.

### 2.1.3 Reference location

To further assist this analysis, the location of a so-called “reference” at the Sun (and a corresponding \( v_{sw} \)) can be provided to the tool, for example, the position of a flare that is assumed to be related with the SEP event. This information is then used on the one hand to visually indicate the position of the reference (i.e., flare) in the plot with a black arrow (cf. Figure 1) and its corresponding heliospheric magnetic field line, which helps to quickly estimate which spacecraft are possible observers of particles originating at this flare. On the other hand, the separation angles between the position of the reference and the magnetic footpoints of the different spacecraft are calculated. This information can then be obtained for further use in a table that also contains all other coordinate parameters provided by JPL Horizons, as demonstrated in the bottom of Figure 2.

### 2.1.4 Plotting options

In addition to the functional options described so far, Solar-MACH provides some plotting settings that enable users to have several options of visualization customization:

- Deactivate plotting of the heliospheric magnetic field lines.
- Add a straight line from the Sun to each observer, indicating the line of sight.
FIGURE 4
Detailed view of Earth’s magnetic connection from the PFSS backmapping example shown in Figure 3. (A) Two-dimensional side view with the point of view (POV) at Carrington longitude 270° (i.e., the observer is located at six o’clock in Figure 3), (B) freely rotatable three-dimensional view. The two colors of the field lines indicate the corresponding magnetic polarity.

- Use numbered markers for the spacecraft to help with color vision deficiency issues or if the plot needs to be converted to grayscale.
- Provide a longitudinal offset for the polar plot to define where the Earth is located (by default 270°, i.e., at six o’clock).
- Make the background of the plot transparent.

2.2 Python package solarmach

The Solar-MACH Python code is made available as an installable Python package called solarmach (Gieseler et al., 2022b) that is listed on PyPI and conda-forge, i.e., it can be installed using either the pip or the conda command line tool, which are by far the most widely used means of installing Python packages. Installing the package in this way ensures that all other required packages are also installed.

An example of the standard workflow for the package is shown in a Jupyter Notebook in Figure 2. This specific Notebook is also available from the subfolder examples in the corresponding GitHub repository. The workflow consists of the following steps (each step is represented by one code cell in the example Notebook in Figure 2):

1. Import the SolarMACH class
2. Provide necessary options and optional parameters
3. Initialize the SolarMACH object with these settings
4. Generate a plot from the SolarMACH object (optionally)
5. Obtain the data as a Pandas DataFrame from the SolarMACH object (optionally)

The Pandas DataFrame obtained in the last step does not only display the dataset in a tabular form, but also provides the user with a ready-to-use dataset for further analyses, ultimately enabling fast diagnostic pipelines (McKinney, 2010). Furthermore, through the optional settings in step 2, it is possible to either provide a file name and ending under which the plot will then be saved (where the file type is automatically determined, e.g., “file.png” or “file.pdf”), or to return the matplotlib (Hunter, 2007) plot object in order to further manipulate it.

A Jupyter Notebook with more detailed descriptions and various use cases is available in the SERPENTINE software repository (Gieseler et al., 2022c). This Notebook contains examples on how to connect the Solar-MACH outputs with other Python functionalities. One example is the creation of an animated GIF file that shows a time-lapse of the spacecraft constellation over a given time period (see here for an example). In addition, this extended Notebook provides the functionality to continue the interplanetary magnetic connections to the solar corona with a PFSS model, which is described in more detail in Section 2.3.

2.3 Further backmapping with PFSS

The magnetic field configuration close to the Sun in the corona differs drastically from the heliospheric magnetic field (HMF) and cannot be described using the simple Parker spiral approach given by Eq. 1. In order to further extend the magnetic connectivity from an observer towards the solar surface, a
FIGURE 5
The Streamlit web version at solar-mach.github.io.
2.4 Streamlit web tool

While all the functionality of Solar-MACH depends fully on Python, a web tool is provided online at solar-mach.github.io that completely runs in the web browser, features a fully graphical user interface (GUI), and requires neither Python knowledge nor installation. It is implemented using the open-source Python package streamlit. This package provides an easy way to set up a GUI to Python functions that is made available through a web server. It is possible to run it locally on a computer (if Python and all other requirements are installed), offering a full GUI to the user instead of running the code in a Jupyter Notebook or terminal. But the main purpose is to deploy the Streamlit application to a web hosting service, because only then it provides all the underlying Python functionalities without the need to install any kind of (Python) software. A screenshot of the web tool is shown in Figure 5. We emphasize that the possibility to use the web tool is invaluable for getting started quickly in a non-code environment, which is especially important for new users.

In addition to quickly obtaining an overview of the observer constellation when a SEP event just took place, another important functionality of the web tool is to bookmark or share a given combination of options that has been set through the GUI. After all settings have been done, a specific URL for this configuration is shown at the bottom of the web page (blue box in Figure 5). An option is provided next to it to generate a short link of this URL because it contains all settings and is thus rather long. An advantage of this configuration is that almost all settings can be provided through the URL, which thus can be used as a simple API. This API is not documented yet, but should be self-explanatory using a given example.

For further use of the Solar-MACH results, the user can save the plot by clicking with the right mouse button on it and using the web browser functionality, exactly like in the Jupyter Notebook version mentioned in Section 2.2 and Section 2.3. In addition, the web tool provides a download button for the plot that might be helpful, especially when using mobile devices like smartphones or tablets. The web tool also automatically displays a table with all ephemeris, magnetic footpoint, and separation information, which can be downloaded as a comma-separated values (CSV) file.

The Streamlit application itself consists of a rather small code base that is open source (Gieseler et al., 2022a). It collects all necessary parameters from the user through the GUI, hands them over to the solarmach package described in Section 2.2, and presents the returned results. The online version provided at solar-mach.github.io is automatically redeployed whenever the corresponding GitHub repository is updated. In this process, the latest solarmach version available at PyPI is installed. Because the GUI needs to be adjusted manually to any new functionality, the Streamlit version may not always offer the latest additions to the solarmach package.
3 Discussion

This paper introduces the Solar-MACH tool to the scientific community. Solar-MACH automatically obtains and visualizes the constellation of the heliospheric spacecraft fleet for a given time and provides a first estimation of the magnetic connectivity of all observers. Here, we illustrated the general workflow of the tool, presented the corresponding Python package solarmach and its PFSS extension, and described the important web version of it provided through the streamlit library. While the tool is intended to have a very simple interface, it is important to point out that for the magnetic backmapping several assumptions are made in the background. Those results should therefore only be interpreted by users with sufficient knowledge of the topic.

The development process of Solar-MACH started at the University of Kiel, Germany, and further discussion took place within the ESA HelioPhysics Archives USeR (HAUS) group. Now, the main development continues within the SERPENTINE EU Horizon 2020 project, which has been presented in the beginning of this paper. It is important to note that the project is, and remains, fully open source (under the BSD 3-Clause License) and very much encourages everyone to contribute, either by actual code or by any other kind of feedback.

The software is completely developed in Python and uses established open-source Python libraries such as pfsspy, streamlit, and sunpy. The development takes place continuously in public GitHub repositories (separated for the Python package solarmach and the web application), which are also the main entry point for any kind of participation. We encourage users to give feedback on bugs or possible new features by submitting GitHub issues. This should also be the starting point for code contributions. But of course we are also open to feedback through direct communication. All code versions released on GitHub are automatically archived at Zenodo with a DOI, so that each version (or the latest) can be cited independently.

Solar-MACH is already used widely in the scientific community (e.g., Badman et al., 2022; Hu et al., 2022; Mierla et al., 2022; Papaioannou et al., 2022; Rodríguez-García et al., 2022), but the development is still ongoing with the introduction of additional functionalities. Some important future plans have already been mentioned in this paper, such as fully integrating the PFSS extension into Solar-MACH. Further ideas are listed in the GitHub repository’s issues section and contain, among others, using dynamically obtained solar wind speeds for the HMF lines, adding trajectories of spacecraft, or introducing features related to CME analyses, such as providing a reference cone instead of an arrow. Furthermore, it is planned to incorporate functions that have already been demonstrated in the Jupyter Notebooks into the core solarmach package, for example, the creation of animated GIF files with time-lapses of the spacecraft constellation. For this, like for all other Solar-MACH functions, one important aspect is to provide them in a way such that they remain accessible to all levels of Python users.

Data availability statement

The source code used in this study can be found online:
- Python package code in GitHub repository solarmach, archived at https://doi.org/10.5281/zenodo.7311178
- Streamlit code in GitHub repository Solar-MACH, archived at https://doi.org/10.5281/zenodo.7311215
- Extended example Jupyter Notebook in GitHub repository serpentline, archived at https://zenodo.org/record/7104866.

Author contributions

ND proposed the initial idea of Solar-MACH and discussed it with VG, AM, MR, and AV. Based on code by ND, the initial software version was designed by ND and JvF. JG continued the development. JG developed the Python package and the Streamlit version. JG, ND, CP, and RV designed further functionalities. CP and ND developed the pfsspy integration. AK, DP, LR-G, and DT tested the software and provided critical comments for the development. JG and RV prepared the first paper draft, and all authors were involved in the preparation of the final manuscript. All authors revised the manuscript before submission.

Funding

This study has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 101004159 (SERPENTINE). JG, CP, RV, and DP acknowledge the support of Academy of Finland (FORESAIL, grants 312357, 336809, and 336807). ND acknowledges the support of Academy of Finland (SHOCKSEE, grant 346902). JvF thanks the German Space Agency (Deutsches Zentrum für Luft- und Raumfahrt e.V., DLR) for their support of his work on the Solar Orbiter EPD team at the University of Kiel under grant 50OT2002. AK acknowledges financial support from NASAs NNN06AAA01C (SO-SIS Phase-E) contract. LR-G acknowledges the financial support by the Spanish Ministerio de Ciencia, Innovación y Universidades FEDER/MCIU/AEI Projects ESP2017-88436-R and PID2019-104863RB-I00/AEI/10.13039/501100011033.
Acknowledgments

The authors acknowledge the Heliophysics Archives User (HAUS) group at ESA, where the idea for this tool was initially discussed. The authors would like to thank everyone who helped to improve Solar-MACH by providing feedback or contributing to the various open-source projects that it utilizes. The authors acknowledge the free cloud hosting provided by Streamlit Inc. The authors thank the two reviewers for their valuable feedback.

Conflict of interest

Author JvF was employed by the company Paradox Cat GmbH since 2021. This article is based on his research conducted prior to that at the University of Kiel and unrelated to his current commercial affiliation.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L., et al. (2022). The Astropy Project Sustaining and growing a community-oriented open-source project and the latest major release (v5.0) of the core package. ApJ, 935, 167. doi:10.3847/1538-4357/ac7c74

Astropy Collaboration, Price-Whelan, A. M., Simplicio, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., et al. (2018). The Astropy Project: Building an open-science project and status of the v2.0 core package. AJ, 156, 123. doi:10.3847/1538-3881/aabc4f

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., et al. (2013). Astropy: A community Python package for astronomy. AAS. doi:10.1051/0004-6361/201322068

Badman, S. T., Carley, E. P., Caslares, L. A., Dressing, N., Jian, L. K., Lario, D., et al. (2022). Tracking a beam of electrons from the low solar corona into interplanetary space with the Low Frequency Array, Parker Solar Probe and 1 au spacecraft. doi:10.3847/1538-4357/ac90c2

Benkhoff, J., Murakami, G., Baumjohann, W., Besse, S., Bunce, E., Casale, M., et al. (2021). BePColombo - mission overview and science goals. Space Sci. Rev., 217, 90. doi:10.1007/s11214-021-00866-1

Daoui, M., and Giacalone, J. (2016). Large gradual solar energetic particle events. Living Rev. Sol. Phys. 13, 3. doi:10.1007/s41116-016-0002-5

Dressing, N., Gómez-Herrero, R., Heber, B., Klassen, A., Malandraki, O., Droge, W., et al. (2014). Statistical survey of widely spread out solar electron events observed with STEREO and ACE with special attention to anisotropies. ApJ, 862, 127. doi:10.1088/0004-637X/862/1/127

Dresing, N., Gómez-Herrero, R., Klassen, A., Heber, B., Kartavykh, Y., and Droge, W. (2012). The large longitudinal spread of solar energetic particles during the 17 January 2010 solar event. Sol. Phys. 281, 281–300. doi:10.1007/s11207-012-0849-y

Fox, N. J., Velli, M. C., Bale, S. D., Decker, R., Driesman, A., Howard, R. A., et al. (2016). The Solar Probe Plus Mission: Humanity's first visit to our star. Space Sci. Rev. 204, 7–48. doi:10.1007/978-94-017-9657-5

Gieseler et al. 10.3389/fspas.2022.1058810

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. Comput. Sci. Eng. 9, 90–95. doi:10.1109/MCSE.2007.55

Kouklouvakos, A., Rodriguez-Garcia, L., Gieseler, J., Price, D., Vourlidas, A., and Vainio, R. (2022). PyThea: An open-source software package to perform 3D reconstruction of coronal mass ejections and shock waves. Front. Astronomy Space Sci. 9. doi:10.3389/fas.2022.974137

Lario, D., Kwon, R.-Y., Vourlidas, A., Raouafi, N. E., Haggerty, D. K., Ho, G. C., et al. (2016). Longitudinal properties of a widespread solar energetic particle event on 2014 February 25: Evolution of the associated CME shock. ApJL, 819, 72. doi:10.3847/2041-8205/819/1/72

Mackay, D. H., and Yeates, A. R. (2012). The Sun's global photospheric and coronal magnetic fields: Observations and models. Living Rev. Sol. Phys. 9, 6. doi:10.12942/lrsp-2012-6

McKinney, W. (2010). “Data structures for statistical computing in Python,” in Proceedings of the 9th Python in science conference. Editors Stefan van der Walt, and Jarrod Millman, 56–61. doi:10.5281/Majora-928f1922-00a

Mierla, M., Zhukov, A. N., Berghmans, D., Parenti, S., Aschere, F., Heinzel, P., et al. (2022). Prominence eruption observed in He II 304 Å up to >6 R⊙ by EUV/PSI aboard Solar Orbiter. A&A, 662, 1, L5. doi:10.1051/0004-6361/202244020

Müller, D., Cyr, O. C., Zouganelis, I., Gilbert, H. R., Marsden, R., Nieves-Chinchilla, T., et al. (2020). The Solar Orbiter Mission. Science overview. A&A, 642, 1. doi:10.1051/0004-6361/202038467

Mumford, S. J., Freij, N., Stanbury, D., Christie, S., Ireland, J., Mayer, F., et al. (2022). SunPy. doi:10.5281/zenodo.7074315

Palmroos, C., Gieseler, J., Dressing, N., Morosan, D. E., Asvestari, E., Valkila, S., et al. (2022). Solar energetic particle time series analysis with Python. Front. Astronomy Space Phys. 9. doi:10.3389/fspas.2022.1053578

Papaioannou, A., Kouklouvakos, A., Mishev, A., Vainio, R., Usoskin, I., Herbst, K., et al. (2022). The first ground-level enhancement of solar cycle 25 on 28 October 2021. A&A, 660, L1. doi:10.1051/0004-6361/20214855

Parker, E. N. (1958). Dynamics of the interplanetary gas and magnetic fields. ApJ, 660, 287. doi:10.1086/146579

Poljančić Beljan, I., Jurdana-Šepić, R., Brajša, R., Sudar, D., Rudičjak, D., Hržina, D., et al. (2017). Solar differential rotation in the period 1964-2016 determined by the Kanzelhöhe data set. A&A, 606, A72. doi:10.1051/0004-6361/201731047

Reames, D. V. (1999). Particle acceleration at the Sun and in the heliosphere. Space Sci. Rev., 90, 413–491. doi:10.1023/A:105010831781

Reames, D. V. (2013). The two sources of solar energetic particles. Space Sci. Rev., 175, 53–92. doi:10.1007/s11214-013-9958-9
